THESE DE DOCTORAT DE L'ETABLISSEMENT UNIVERSITE BOURGOGNE FRANCHE-COMTE

PREPAREE A L'UNIVERSITE DE BOURGOGNE

Ecole doctorale n° 592

ÉCOLE DOCTORALE LECLA

Doctorat de Sciences du langage

Label Européen

Par

Mme Rivas Ginel María Isabel

THE ERGONOMICS OF CAT TOOLS FOR VIDEO GAME LOCALISATION

Thèse présentée et soutenue à Dijon, le 21/06/2023

Composition du Jury :

M., Froeliger, Nicolas	Professeur à l'Université Paris Cité	Président
Mme, Seghiri, Miriam	Professeure à l'Université de Málaga	Rapportrice
Mme, Monti, Johanna	Professeure à l'Université degli Studi di Napoli	Rapportrice
M., De Faria Pires, Loïc	Maître de Conférences à l'Université de Mons	Examinateur
M., Gautier, Laurent	Professeur à l'Université de Bourgogne	Directeur de thèse
Mme, Corpas Pastor, Gloria	Professeure à l'Université de Valladolid	Codirectrice de thèse

LECLA
ÉCOLE DOCTORALE

Titre : L'ergonomie des outils pour la traduction de jeux vidéo

Mots clés : Jeux vidéo, Localisation, Traduction, Enquête, outils TAO

Résumé : La localisation revêt une importance capitale dans l'industrie vidéoludique car le traducteur doit continuellement se consacrer à un travail d'adaptation culturelle du contenu, à la fois au niveau non textuel (unités de mesure, monnaies, lois) et culturel (références, blagues, etc.). Toutefois, le traducteur de jeux vidéo est également confronté à des phénomènes typiques de ce secteur, tels que travailler sur un document Excel sans avoir accès au jeu pour visualiser les contraintes spatiales, ou encore sur un document non linéaire pour connaître l'identité du personnage qui parle (ou à qui il s'adresse). De plus, ils doivent faire face aux erreurs commises lors de la phase d'internationalisation, telles que l'oubli de prendre en compte le coefficient de foisonnement des langues. Toutes ces contraintes peuvent entraîner de nombreux *bugs* linguistiques. Cette thèse vise à trouver des solutions technologiques qui pourraient permettre aux traducteurs d'avoir des indices visuels, des références, ou potentiellement le jeu lui-même, afin d'éviter de travailler à l'aveugle. La base de cette étude est la conception, la diffusion et l'analyse de trois enquêtes en ligne adressées à différents professionnels travaillant dans la localisation et le développement de jeux vidéo. Les enquêtes ont été rédigées en anglais sans limitation de langues de travail ou de nationalités, afin d'atteindre un nombre maximum de participants. La première enquête s'est concentrée sur les traducteurs pour analyser les pratiques commerciales actuelles et les outils qu'ils utilisent. La deuxième enquête a été adressée aux testeurs linguistiques pour recueillir des données sur les bugs linguistiques qu'ils rencontrent. Enfin, la dernière enquête, créée pour les développeurs de jeux vidéo, a été utilisée pour collecter des données techniques sur les formats de fichier, les processus et les pratiques commerciales.

Title: The ergonomics of CAT tools for video game localisation

Keywords: Videogames, Localisation, Translation, Survey, CAT tools

Abstract: Localisation is of capital importance in the gaming industry and localisers must continually adapt the content both at a non-textual level (measure units, currencies, laws) and a cultural level (pop culture references, jokes, puns, etc.). In addition, they must face complications inherent to the sector since, for instance, the strings are usually sent in Excel files that lack internal coherence and omit the images linked to the text that must be translated. Another main obstacle is deficiencies in the internationalisation phase of the project where developers may forget, for example, to leave enough space in the menus for the translated items. This is closely linked to the language expansion rate and the lack of a WYSIWYG (what you see is what you get) environment, resulting in numerous linguistic bugs. This thesis aims to find technological solutions that could allow localisers to access visual cues, references, or potentially the game itself to avoid working blind. The basis of this study is the conception, diffusion, and analysis of three online surveys addressed to different professionals working in localisation and video game development. The surveys were drafted in English without any limitations to working languages or nationalities in order to reach a maximum number of participants. The first survey focused on translators to analyse current business practices and the tools they used. The second survey was addressed to linguistic testers to gather data about the linguistic bugs they encounter. Finally, the last survey, created for video game developers, was used to collect technical data about file formats, processes, and business practices.

UBFC
UNIVERSITE BOURGOGNE FRANCHE-COMTE

Université Bourgogne Franche-Comté
32, avenue de l'Observatoire
25000 Besançon

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à Madame le Professeur Gloria Corpas Pastor et à Monsieur le Professeur Laurent Gautier, mes deux co-directeurs de thèse. Leur rigueur, leurs qualités scientifiques et humaines exceptionnelles, ainsi que leur disponibilité, leur patience, leur soutien et surtout leurs encouragements m'ont motivée tout au long de ce travail. Je suis honorée de pouvoir également remercier sincèrement Mesdames et Messieurs le Professeur Nicolas Froeliger, Miriam Seghiri, Johanna Monti et Loïc De Faria Pires pour avoir accepté de faire partie de mon jury.

J'exprime tous mes remerciements à Monsieur Will Noonan, Maître de Conférences à l'Université de Bourgogne, qui a mené à mes côtés la bataille contre les coquilles typographiques, grammaticales et orthographiques, et m'a suggéré de meilleures reformulations et tournures de phrases. Je remercie très chaleureusement les traducteurs, testeurs linguistiques et développeurs ayant participé aux enquêtes.

Mes remerciements vont également à ma famille et en particulier à mon père, pour leur soutien et leur présence malgré mon absence ces dernières années. Sans leur aide précieuse et leur support moral, cette thèse n'aurait jamais vu le jour. Je souhaite également remercier Andrew Pitt pour son soutien et son aide scientifique précieuse (informatique), ainsi que sa patience infinie pendant mes moments de fatigue et de détresse.

Enfin, j'adresse mes derniers mots à mes amis, en particulier à Camille Metin, Clément Lepert, Pauline Dionne, Eleonora Montagner et Dorota Kamiska. Ils sont restés à mes côtés malgré le manque de variété de mes conversations, et ont toujours su comment m'encourager et me soutenir dans les moments difficiles.

TABLE OF CONTENTS

List of Figures

List of Images

List of Tables

INTRODUCTION

The advent of computers, video games, and the appearance on the market of new technologies for translation purposes go hand in hand. Thus, the first video game that appeared, *Spacewar!* (1962), was created by hobbyists in a smaller version of the giants that were the first computers to be ever created. During the ‘70s, while computers were being improved and the use of microprocessors enabled engineers to gradually reduce their size, the video game industry started taking its first steps aided by the popularity of arcade cabinets and a rapid succession of game consoles that would be later considered as the first generation. The ‘80s saw the arrival of personal computers, productivity software, and computer games into people’s workplaces and homes. These advances were quickly exploited by software companies that discovered the potential of international markets and fuelled the beginning of localisation practices, described as “taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold” (Esselink, 2000, p. 3). The evolution of localisation and the gradual improvements and adjustments it underwent were mostly characterised by a trial-and-error approach that resulted in the process we know today. In the meantime, less than 25 years after *Spacewar!*, the video game industry survived a market crash that almost cut its life short due to a combination of a general lack of quality in games and fierce competition in terms of consoles. Once the clear victors resurfaced, the sector began an unstoppable economic growth that would practically triple the revenue produced in four decades going from an estimated 59 billion dollars[1] in the ‘80s before the market crash to 196.8 billion[2] in 2022.

During those four decades, as part of the abovementioned trial-and-error approach, companies started refining the process of developing and launching multilingual products by adding a preliminary step. This phase became known as internationalisation and was described by Esselink (2000, p. 2) as “the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for re-design. Internationalization takes place at the level of program design and document development”. With time, it became a common practice that is almost always

[1] https://www.visualcapitalist.com/50-years-gaming-history-revenue-stream/
[2] https://newzoo.com/key-numbers

automatically implemented in order to facilitate the creation of multilingual products and avoid *post hoc* modifications to the source code of a piece of software. Incidentally, during these decades, language service providers seized the opportunities offered by the increasing popularity of multimedia products and either specialised in localisation or added localisation as part of their services. Additionally, a wide variety of translation technologies appeared on the market in order to reduce costs, improve translators' productivity, and supply competitive prices.

Nowadays, we can observe the democratisation of video games due mostly to the ever-rising popularity of mobile gaming and the success of casual games, which have turned smartphones into another game platform that can be found in everybody's pockets. Consequently, the profile of gamers has drastically evolved, going from what was considered geeks and teenagers to players of all ages, social statuses, and nationalities. "[V]ideo games are a multi-billion dollar industry catering for home entertainment market […] It is no longer an option to offer English-only games" (Bernal-Merino, 2013, p. 2) as video game companies often did in the very beginning. Therefore, localisation has become a crucial part of the video game development process that "can account for more than 50% of total sales [thus], international markets are not something to be ignored" (Chandler, 2020, p. 231). As a result, the steady increase in the number of video games being released to the market every year has boosted the need for professional localisation. Currently, many developers automatically plan for the simultaneous shipment of games in more than 10 different languages, which is the case of Triple-A game companies—also known as AAA, companies that create games "with the biggest budgets (often in excess of £10 million) which are expected to become blockbusters" (Bernal-Merino, 2013, p. 286). Nevertheless, video game localisation is a remarkably complex task due to simultaneous shipment practices, the multimedia nature of the product, the wide variety of text types that can be found in games and the functions they play, the lack of text linearity and visual environment, the presence of code mixed in with the source text, and all the cultural components involved.

Localisers are asked to provide creative yet accurate translations that maintain the immersion of the player and the suspension of disbelief while adapting the product to fabricate the illusion that it was created specifically for the locale in question. However, business practices reported by professionals in the field in the first survey that will be presented and discussed in this thesis as well as papers published by multiple scholars underline the complexity of working "in a double-blind process (no audiovisual context, no text linearity) that is bound to produce more linguistic and culturalisation mistakes that they otherwise would" (Bernal-Merino, 2013, p. 119). As technology evolves and new programmes are created for video game development, testing, and localisation it becomes crucial to research the possibilities offered by these new tools with a view to finding potential features that could increase localisers' productivity and translation quality while reducing the level of complexity involved in the process and solving one of the main issues in the field: the lack of visual environment. Therefore, it is essential to identify current business practices, programmes, and processes related to the three main actors in the video game localisation chain: developers (or engineers), localisers, and linguistic testers. Furthermore, in order to analyse the ergonomics of the tools used, it is necessary to ascertain localisers' needs and the consequences of current practices encountered by testers during the linguistic quality assurance phase.

The starting hypothesis is that it is possible to offer tools to video game localisers that cater for their needs. And this can be done by (i) discovering tools on the market that could be adapted (or interconnected) to meet the demanding needs of the field in terms of localisation, or, else, (ii) by developing a comprehensive localisation tool specifically created for the video game localisation process. Furthermore, due to the rapid evolution of technology and the constant appearance of new programmes, we speculated that these new systems should be able to grant visual access to localisers to a certain extent. Additionally, due to the lack of visual context in such a complex process, we hypothesised that the main type of linguistic bug encountered would be a direct consequence of this particular constraint, which would also play a major role in the causes of any other type of bug. Therefore, due to the necessity of obtaining a sufficient sample of data from as many sources and backgrounds as possible to ensure representativeness, the methodology favoured to confirm or refute our initial hypotheses was the implementation of three user surveys focused on localisers, linguistic testers, and developers, respectively. This decision was also instilled by the remarkably reduced number of studies that used surveys addressed to translators in the field of video game localisation, the lack of quantitative data about video game linguistic testing, and the reduced number of studies taking a multidisciplinary approach in the field. Furthermore, the implementation of surveys would provide the answers to a series of key questions, including but not limited to:

1. Are there significant differences in the access to reference material depending on the localiser’s place or type of employment?
2. What are the current business practices in video game localisation and how do they affect localisers?
3. What features and functionalities should include the perfect tool (or suite of tools) for video game localisation?
4. What are video game localisers’ attitudes towards these features and functionalities?
5. What tools do localisers use and are there any new technologies more suitable for the task at hand?
6. Since linguistic testers have to review and amend translations, do they have an educational background in language or translation studies?
7. What are the consequences of the lack of a visual environment in terms of linguistic bugs?
8. What are the causes of those linguistic bugs according to testers?
9. When does the localisation phase actually start in video game development?
10. Do the current development methods provide potential solutions that could be taken advantage of to grant visual access to localisers?
11. Can the tools or testing methods used by developers allow localisers to see changes to the game “on the fly”?
12. Is there a way to cover the localisers’ needs with a specific tool or suite of tools that encompass the whole development process?

This dissertation is composed of 5 chapters, besides the introduction and the conclusion, under the titles: the video game industry, video game localisation, first survey, second and third surveys, and the analysis and discussion of the results. The first chapter presents an overview of the technological milestones and the historical advances in video games in order to create a comprehensive picture of the industry. Afterwards, we discuss the roles of the main actors involved in the development of a video game in order to illustrate the different positions that can be found both inside and outside the development team proper, their characteristics, the interconnections between them, and their impact on the production chain. These descriptions should allow the reader to better understand the sometimes blurred line between the different job titles that can be found in the industry, clarify the definition of the term "video game developer", and contextualise some of the questions included in the third survey. Finally, we will present some of the tools used in the creation of a video game—especially those included in the surveys—and explain the video game development life cycle. Overall, the main goal of the first chapter is to lay the foundations of this dissertation and provide essential information in order to allow the reader to better grasp the complexity of the field, offer an insight into how video games are developed and the impact of dependencies between tasks, and understand key concepts that will be developed further in the following chapters.

The second chapter offers a cursory glance at the field of video game localisation from both a historical and academic perspective. First, we will begin by defining the scholarly framework of this particular research field by situating it in relation to translation and localisation studies. Subsequently, we will present an overview of the papers, conferences, dissertations, etc. published on video game localisation proper throughout time as well as their contribution to the topic. Afterwards, the chapter provides a historical sketch of the birth of localisation practices, translation technologies, and their evolution throughout time. This section, in combination with a description of the different localisation types that can be found nowadays and their characteristics, will serve as the base of a subsequent analysis that aims to portray the inherent traits of video game localisation. Therefore, the final sections of this chapter focus on the history of game localisation, the common aspects it shares with other types of localisation, its specificities, and the localisation process itself. This chapter aims at completing the picture of the production of a multilingual product from the localisation point of view as well as situating this dissertation in the academic frame of translation and localisation studies.

The third chapter presents the raw results of the first survey without engaging in an in-depth study as the data will be subsequently cross-analysed with those from the other surveys in the final chapter. Thus, it briefly describes the methodology used for the survey's implementation, its design, and the results, which are divided into fourteen sub-sections depending on their nature: personal information; languages; professional information; business practices; assets, access, and linearity; attitudes towards features and functionalities; asset extraction and integration tools, content management tools, and project management (PM) tools; linguistic testing tools; tree-based tools for dialogues; resources; corpora compilation tools; terminology extraction and management tools; computer-assisted translation tools; and machine translation tools.

The fourth chapter includes the results of the remainder of the surveys as they were shorter than the previous one and more focused on technical data. Therefore, the chapter is divided into two main sections: video game linguistic testing and video game development tools. Each section begins with an overview of the implementation and design of the survey in question and the following sub-sections follow as closely as possible the order established in the previous chapter in order to maintain consistency. Finally, the fifth chapter provides the cross-analysis of all the data collected as well as a detailed examination of certain particular aspects such the access to resources and reference material depending on the type of employment in localisation, the results from the key questions included in each survey, tools, etc. Thus, the chapter is divided into five main sections: respondents' profiles, business practices, key questions and technical data, and tools analysis.

Additionally, this dissertation includes a series of relevant documents in the form of appendices: the configuration of the three first surveys with all the questions and the different options provided (Appendices 1 to 3), the unedited complete list of comments left by localisers about the reference material they receive (Appendix 4), the unedited comments about missing features that should be included in the ideal programme for video game localisation according to professional localisers (Appendix 5), the complete and unedited list of extra comments left by linguistics testers at the end of the second survey (Appendix 6), and an example of a bogus answer left by a developer to illustrate the issues encountered during the implementation of the third survey and the subsequent decision of reviewing one by one all the answers provided by the participants (Appendix 7).

CHAPTER 1. THE VIDEO GAME INDUSTRY

The aim of this chapter is to provide an overview of the video game industry that will serve as an introduction to the inner workings of the field and present its characteristics, particularities, and the complexity of its processes. Furthermore, it will complement the information provided in the second chapter and create a link between the technical questions included in all three surveys. Consequently, the first section of this chapter will serve as a historical sketch of some of the most important advancements in technology and how they contributed to paving video games' road to success. Therefore, after defining the concept of both games and video games, we will review the industry's history starting from the appearance of computers and arcade games and subsequently examine the evolution of consoles, the rise and fall of companies and devices, and briefly discuss some of the most iconic video games and the changes they sparked off.

The second section of the chapter is devoted to reviewing the main actors involved in the video game industry in order to better understand their roles and the place engineers have within the team—as well as their tasks. For this purpose, we will start with platform holders, current trends in terms of devices, and present concisely some of their specific requirements. Afterwards, we will introduce the role publishers play in the industry and the relationships they build with developers (both individuals and companies) in order to release a game. Then, we will analyse the organisation chart of a video game development studio, observe the different positions that can be found in a team, and describe the basic functions of each team member. The third section will focus on presenting the characteristics of the most important tools (or combination of tools) for our purposes: the video game production pipeline, the configuration of a game engine, and tree-based dialogue tools. More specifically, whereas the first one englobes the whole process, the last two are the object of some of the questions included in the first and third surveys as they are key in our research.

Finally, the last section of the chapter examines the game development life cycle by presenting the stages included in every phase and the team members involved. Therefore, besides illustrating the characteristics of each step, the different subsections will introduce the various builds of a game following an iterative development method and the recommended milestones that should be achieved in them classified by discipline. Afterwards, the final two subsections will present the different game development models and a brief introduction to linguistic testing, a largely ignored but crucial step in the process and the object of our second survey.

1.1 Video games: a historical overview

The video game industry has achieved great success in its relatively short life span; born around 60 years ago, it started gathering momentum in the '80s, became a worldwide phenomenon in the '90s and has already changed the way we think about entertainment. As Juul wrote more than fifteen years ago, compared to other forms of digital entertainment, "multimedia interactive software" (Bernal-Merino, 2013) is still young (Juul, 2005, p. 15):

> The video game is thus a little more than forty years old, and it has been part of popular culture for around thirty years. Compare this to the roughly seventy-five years of television, a hundred years of film, and five hundred years of the printing press. Therefore, video games are a comparatively new cultural form, intimately linked to the appearance of computers, postdating literature, cinema, and television. However, if we think of video games as games, they are not successors of cinema, print literature, or new media, but continuations of a history of games that predate these by millennia. The Egyptian board game, senet […], found in the 2686 BC tomb of Hesy-re is a precursor of contemporary backgammon and Parcheesi, games that are commonly played using computers today. Therefore, the question is not whether video games are old or new, but how video games are games, how they borrow from non-electronic games, and how they depart from traditional game forms.

First and foremost, the hypernym "game" refers to a myriad of recreational activities with deep roots in history and society. Games may be enjoyed individually or with multiple players, might be competitive or not, might entail physical activity or be mostly focused on mental effort. “There are many types of games, for example cards, football, billiards, catch, charades, marbles, I spy, dice, connect 4, grownups, video games and many more, each involving different rules and props” (Bernal-Merino, 2013, p. 15). However, if “the act of playing is universal, the themes and activities themselves may not be” (Bernal-Merino, 2013, p. 15), as games are profoundly affected by the ethos of each community. Therefore, the definition and description of games has proven to be a complex endeavour due to cultural differences and disparate perceptions about the essence of games and the difficulty in finding common ground in such a vast ocean. One of the first attempts to define the concept was made by Huizinga (1950, p. 15; cited in Bernal-Merino, 2013, p.16):

> A game is a free activity standing quite consciously outside ‘ordinary’ life as being ‘not serious’, but at the same time absorbing the player intensely and utterly. It is an activity connected with no material interest, and no profit can be gained by it [sic]. It proceeds within its own proper boundaries of time and space according to fixed rules and in an orderly manner. It promotes the formation of social groupings which tend to surround themselves with secrecy and to stress their difference from the common world by disguise or other means.

Additionally, other authors such as Juul (2005) analysed the topic and concentrated on the issue of defining video games proper. The author stated that all games share a common core of similarities that can be extrapolated to create a universal model of what a game is and the features that all of them share. Juul's list of common elements, further developed by Zackariasson and Wilson, comprises 6 characteristics (Zackariasson and Wilson, 2012, p. 5):

1. *Rules*: Games are rule-based.
2. *Variable, quantifiable outcome*: Games have variable, quantifiable outcomes.
3. *Valorization of outcome*: The different potential outcomes of the game are assigned different values, some positive and some negative.
4. *Player effort:* The player exerts effort in order to influence the outcome. (Games are challenging.)

5. *Player attached outcome:* The player is emotionally attached to the outcome of the game in the sense that a player will be [a] winner and "happy" in [the] case of a positive outcome, but a loser and "unhappy" in [the] case of a negative outcome.
6. *Negotiable consequences:* The same game [set of rules] can be played with or without real-life consequences.

Therefore, when we analyse these 6 intrinsic characteristics and contrast them with Huizinga's definition, we can observe many similarities that can easily be extended to video games themselves. Another topic that requires attention is the denomination of this form of entertainment. As Bernal-Merino discusses, due to the industry's popularity and its impact on society, game "has also become a shortened form used to refer to the young multimedia interactive entertainment software industry […] It is a frequent occurrence that, nowadays, many people use the superordinate or hypernym 'game' to refer to 'video game'" (Bernal-Merino, 2013, p. 17). The author goes on to describe the different possible denominations until concluding that the most accurate term would be "multimedia interactive entertainment software". However, similarly to him and due to the length of the nomenclature, we will either use "game" or "video game".

1.1.1 From computers to consoles

Even though games are well rooted in history, the appearance of video games would not have been possible without the invention of computers, the democratisation of personal computers, and the apparition of arcade machines. There is significant debate about determining the exact device that could be considered the first computer in history and it is beyond the scope of this thesis to cover the issue in great detail. Therefore, for the sake of simplicity, only computers that appeared in or after 1938 will be considered due to the crucial role of the Second World War in their development. The first programmable binary computer, the Z1, was created by Konrad Zuse in Germany between 1936 and 1938. Built in his parent's living room, it only had purely mechanical components and used punched tape to store the sequence of instructions it was capable of executing (Rojas, 1997). Afterwards, between 1938 and 1942, John Atanasoff built a more sophisticated binary machine alongside with his assistant Clifford Berry. The machine was named ABC (Atanasoff Berry Computer) and used switches to store numbers as digits—similarly to the Z1—although the main difference was that up until then, numbers had been stored via wheels and rods' positions; thus moving from analogue machines to digital computers. Image 1, Z1 on the left and ABC on the right, portrays both computers in order to illustrate these differences.

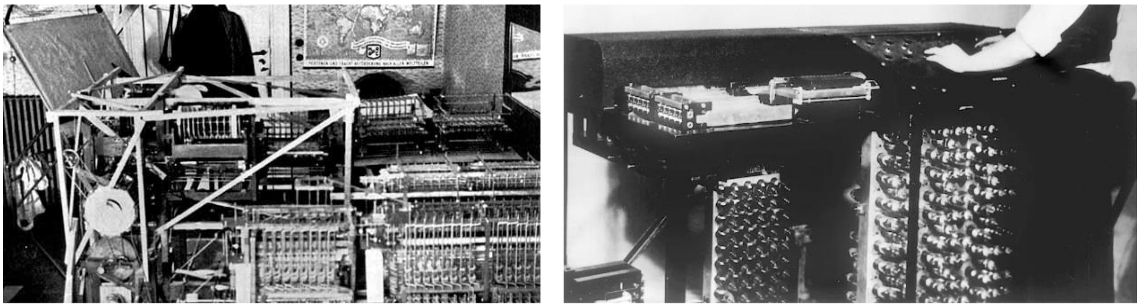


Image 1: The original Z1 [3](left) and the Atanasoff Berry Computer[4] (right)

The next step towards the creation of modern computers was the use of electromagnetic relays in order to process and store numbers. This method was first used by Howard Aiken in the Harvard Mark I and resulted in a digital computer that was almost 16 metres long and more than 2 metres high due to the size of the aforementioned relays—which were particularly slow and consumed high amounts of energy. The Harvard Mark I or the IBM Automatic Sequence Controlled Calculator (ASCC) was built in 1944 at Harvard University and ran repetitive mathematical calculations in order to create tables that were used by the Navy during the war. Around the same time, two machines, the Colossus and the Electronic Numerical Integrator And Calculator (ENIAC), adopted a different approach and started to use vacuum tubes (or valves) to increase the speed and reduce the energy needed to power the device. Although vacuum tubes had already been used in the ABC—albeit in combination with electromagnetic relays—and the Colossus started operating in 1944, the ENIAC is considered by many as the pioneer in their use as the Colossus remained a secret until the end of the war. Furthermore, the Colossus' only purpose was to crack the code created by the Tunny machine used in German transmissions and could not perform other tasks. Built in London at Bletchley Park by a team led by Tommy Flowers, it was fully electronic, used 2500 vacuum tubes and weighed 5 tons. The ENIAC (Image 2), on the other hand, used more than 17000 valves, weighed almost 30 tons and was multi-purposed.

[3] https://dcmlr.inf.fu-berlin.de/rojas/reconstruction-of-the-z1-computer/

[4] https://www.britannica.com/technology/Atanasoff-Berry-Computer

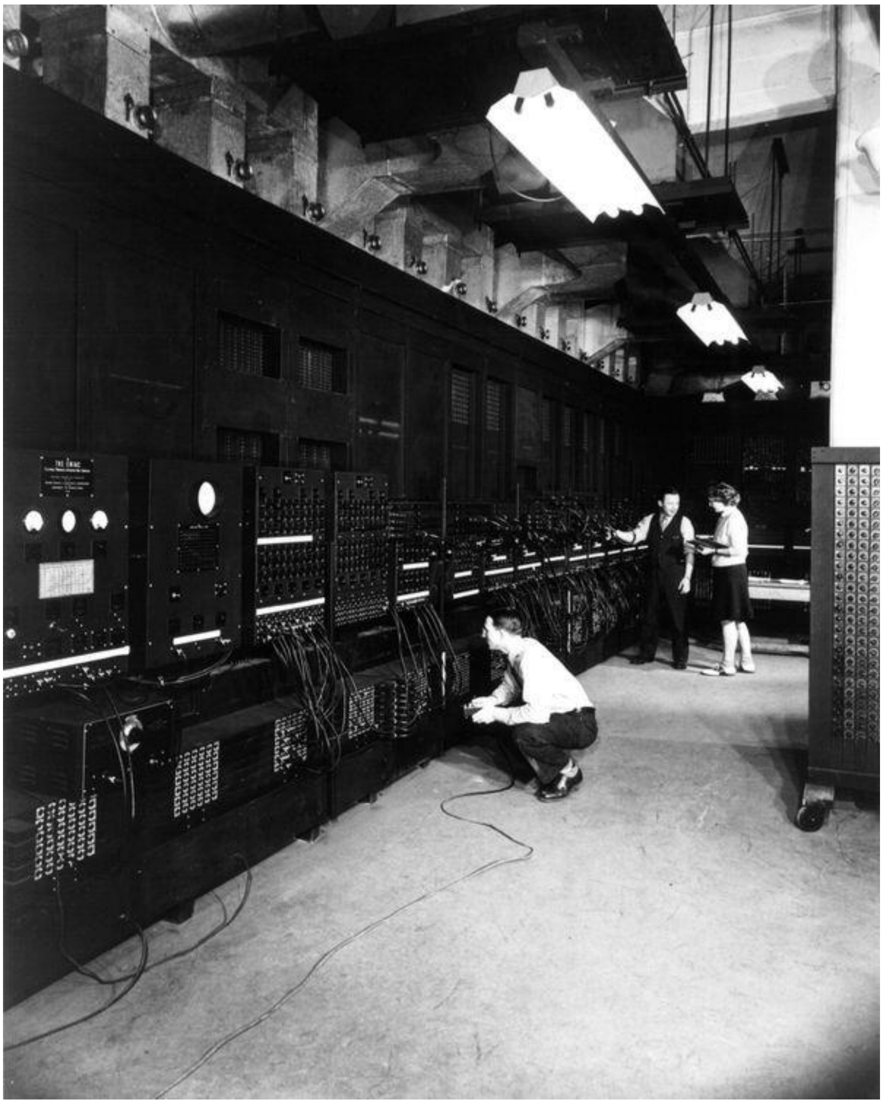


Image 2. The ENIAC being set up[5]

Therefore, between the ‘30s and throughout the ‘60s computers were characterised by their size and it would not be until the development of microprocessors that their dimensions could be reduced, which led to the appearance of personal computers. Nevertheless, it was during those decades of room-sized computers that the history of video games begins. As a matter of fact, the first two prototypes of video games—*Tennis for two* (1958) and *Spacewar!* (1962)—were created during those days of early development. *Tennis for two* (Image 3) was conceived by William Higinbotham for an exhibition and consisted of the representation of a tennis court displayed on an oscilloscope, which led to a debate on whether it can be considered a computer game or not.

[5] https://www.computerhistory.org/revolution/birth-of-the-computer/4/78/316

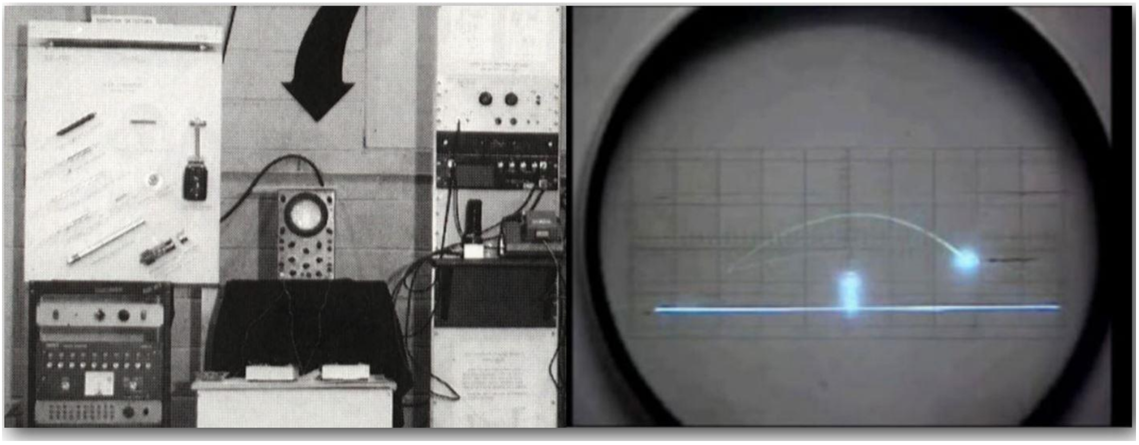


Image 3. Tennis for two[6]

Some years later Steve Russell, a student at MIT, decided to hack a computer that had been recently donated to the university in order to create an interactive digital game. The computer was the PDP-1 (Image 4) and was smaller than the other computers available at MIT at the time—an IBM 709 and a TX-O. The former was referred to as “the Hulking Giant” and the latter, much smaller and equipped with a monitor, “still required 15 tons of air-conditioning equipment for cooling” (Kent, 2001, p. 17). Steve Russell and other members of the MIT also designed a specific controller and never patented their work or aspired to any financial gain even though he spent “nearly six months and 200 hours to complete the first version of the game: a simple duel between rocket ships. Using toggle switches built into the PDP-I, players controlled the speed and direction of both ships and fired torpedoes at each other” (Kent, 2001, p. 18).

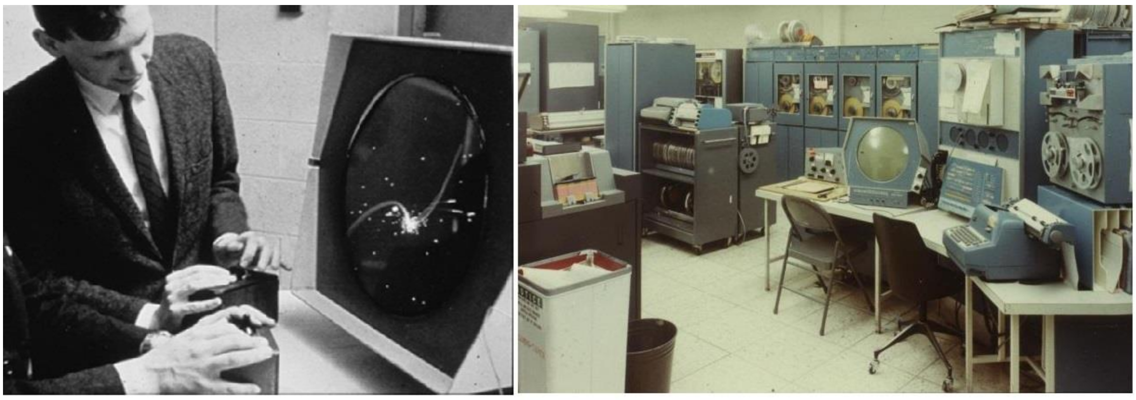


Image 4. The PDP-1 or Programmable Data Processor-1[7] (right) and *Spacewar!*[8] (left)

[6] https://maker.uvic.ca/tennis/

[7] https://www.computer-history.info/Page4.dir/pages/PDP.1.dir/

[8] https://www.computerhistory.org/pdp-1/spacewar/

However, due to the fact that computers remained rare and expensive devices, the successors of these first games were created and commercialised for arcade machines, consoles, and some home computers in the late '70s. This decade sees the rise of arcade games developed and commercialised by Atari, a firm established in 1972 by Nolan Bushnell and Ted Dabney who "founded their company with an initial investment of $250 each. Within ten years, Atari would grow into a $2-billion-a-year entertainment giant, making it the fastest-growing company in U.S. history" (Kent, 2001, p. 38). Before starting the company, Bushnell developed *Computer Space* (1971) and, "[i]nstead of building a general-purpose computer, he designed a specialized device capable of only one thing—playing his game" (Kent, 2001, p. 31). This previous experience proved the potential success of cabinets and served as an inspiration for Atari's first arcade game *Pong* (1972), which became a grand success (Image 5).

Image 5. Atari's first arcade game *Pong*[9]

Alongside arcade games, 1972 will bring games into people's homes with the first video game console: the *Odyssey*. Commercialised by Magnavox, the console "featured no

[9] https://www.arcade-museum.com/game_detail.php?game_id=9074

audio and was sold with plastic overlays that the player attached to the television screen" (Chandler and Chandler, 2010, p. 4). The device was not a success due to its high price and the lack of publicity; it was followed by *Home Pong* (1974) by Atari, the *Channel F* console (1976) by Fairchild Semiconductor, and *Atari VCS* or *Atari 2600* (1977) which "soon became a must-buy holiday gift, and over the next few years, the word "Atari" would become synonymous with "video games"' (Chandler and Chandler, 2010, p. 4). The most iconic games of the decade—besides *Pong—*were *Space Invaders* (1978) and *Asteroids* (1979) which were among the first released for arcade coin-operated machines and subsequently adapted for the *Atari 2600*. Table 1 (Mangiron Hevia and O'Hagan, 2013, p. 45) summarises the developments that took place during the '70s and includes the platforms, milestone games and technological achievements.

Time	Platform	Sample of milestone games	Technological milestone
1970s	Following the US development of modern video games, Atari dominated the game scene with early arcade games and brought them into the home with game consoles.		
1972	1st generation console Magnavox Odyssey – no sound	1972 *Pong* (US)[22]	Apple II computer supporting a library of games Use of microprocessors instead of integrated circuits for better quality animation
1976	Channel F – first cartridge home video game system		
1977	2nd generation console Atari VCS (Atari 2600) launched	1978 Space Invaders (J) 1979 *Asteroids* (US)	

Table 1. Technical milestones of game console evolution during the '70s (Mangiron Hevia and O'Hagan, 2013, p. 45)

1.1.2 From the 1980s to the 2000s

The '80s proved to be extremely fruitful for arcade games regardless of the fact that personal computers had also entered the scene. Thus, we find devices "including the *Commodore 64* (C64) in the US and the *Sinclair Spectrum* in the UK, as well as the earlier *Apple II*, allowing for a range of games to be played on the same machine as opposed to the one-game-only hardwired consoles" (Mangiron Hevia and O'Hagan, 2013, p. 50-51). Furthermore, this decade was scarred by what "is generally known as "the game industry market crash" in the US, often referred to as the 'Atari crash'" (Mangiron Hevia and O'Hagan, 2013, p. 50). The stage for the crash was set when a group of former employees of Atari decided to create Activision in 1979 and started producing "third-party titles (games developed by companies other than the console manufacturer) for the 2600" (Chandler and Chandler, 2010, p. 5). The first three years of the '80s saw the release of remarkably popular games to the market such as the iconic *Pac-Man* (1980) and the first platform game *Donkey-Kong* (1981). However, even though *Donkey-Kong*'s port to the 2600 was a success, *Pac-Man*'s was a complete failure, which added to the lack of popularity of the game *E.T. the Extra-Terrestrial*. These misadventures, coupled with the “glut of systems and games” (Chandler and Chandler, 2010, p. 6), games’ low quality and the rise of multifunctional personal computers, led to the crash of the industry in 1983. The sudden collapse of the US company Atari created the perfect ground for the birth of Nintendo and Sega, two Japanese manufacturers that released by the end of the decade 16-bit consoles with better graphics and sound. Sega and Nintendo would enter a fierce competition in the second half of the decade known as the “platform war” and both companies continuously released new devices and games in order to surpass each other (Table 2).

1980s	This decade saw the dramatic decline of the game industry in the US with the demise of Atari, along with the rise of Japanese console manufacturers such as Nintendo and later Sega, with a shift to 16-bit consoles.		
1982	Commodore 64 (C64); Sinclair ZX Spectrum	1980 *Pac-Man* (J) 1981 *Donkey Kong* (J) 1982 *Microsoft Flight Simulator* (US) 1983 *Mario Bros.* (J) 1985 *Tetris* (USSR); *Habitat* (US) 1987 *The Legend of Zelda* (J) 1988 *Ninja Ryukenden* (J) 1989 *SimCity* (US)	Use of tape and floppy disks in addition to cartridges for C64; PC sound cards 8-bit machine with cartridges
1983/ 1985	3rd generation console Nintendo Famicom released in Japan in 1983; Nintendo Entertainment System (NES) released elsewhere in 1985		
1987/ 1989	Nintendo Game Boy worldwide release in 1989; Sega MegaDrive unveiled in 1987 in Japan (released as Genesis in US in 1989); Atari’s handheld Lynx		Handheld game consoles with communication cables 16-bit machines with better graphics and sound

Table 2. Technical milestones of game console evolution during the ‘80s (Mangiron Hevia and O’Hagan, 2013, p. 45)

The video game industry during the ‘90s was changed once more by “the advent of 3D graphics and new genres” (Chandler and Chandler, 2010, p. 7). In terms of devices’ popularity, we can obverse a decline in arcade games, a remarkable growth for computer games, and the continuation of the console wars. Nintendo had already established itself as a household name in Asia, America and Europe. Nevertheless, as Izushi and Aoyama (2006, p. 1847) explain, “[t]echnological development in the 1990s altered the logic of competition, however, which undermined the monopoly of Nintendo and increased the share of PlayStation, made by Sony Computer Entertainment (SCE) [was] CD-ROM technology”. This power change was mostly due to Nintendo’s decision to continue developing games using cartridges which ultimately ended up damaging their reputation as “consumers saw cartridges as archaic” (Chandler and Chandler, 2010, p. 8). Additionally, computer games also continued to grow with the release of titles such as *Doom* (1993), a first-person shooter game that allowed players to create their own levels using mods and that had an innovative method of distribution where “users could purchase part of the game for a small amount, then pay to play the rest of the game” (Chandler and Chandler, 2010, p. 7). Image 6 provides a visual representation of the evolution of the market during its first three decades of existence and portrays the fluctuations in terms of revenue per platform as well as milestone games and consoles.

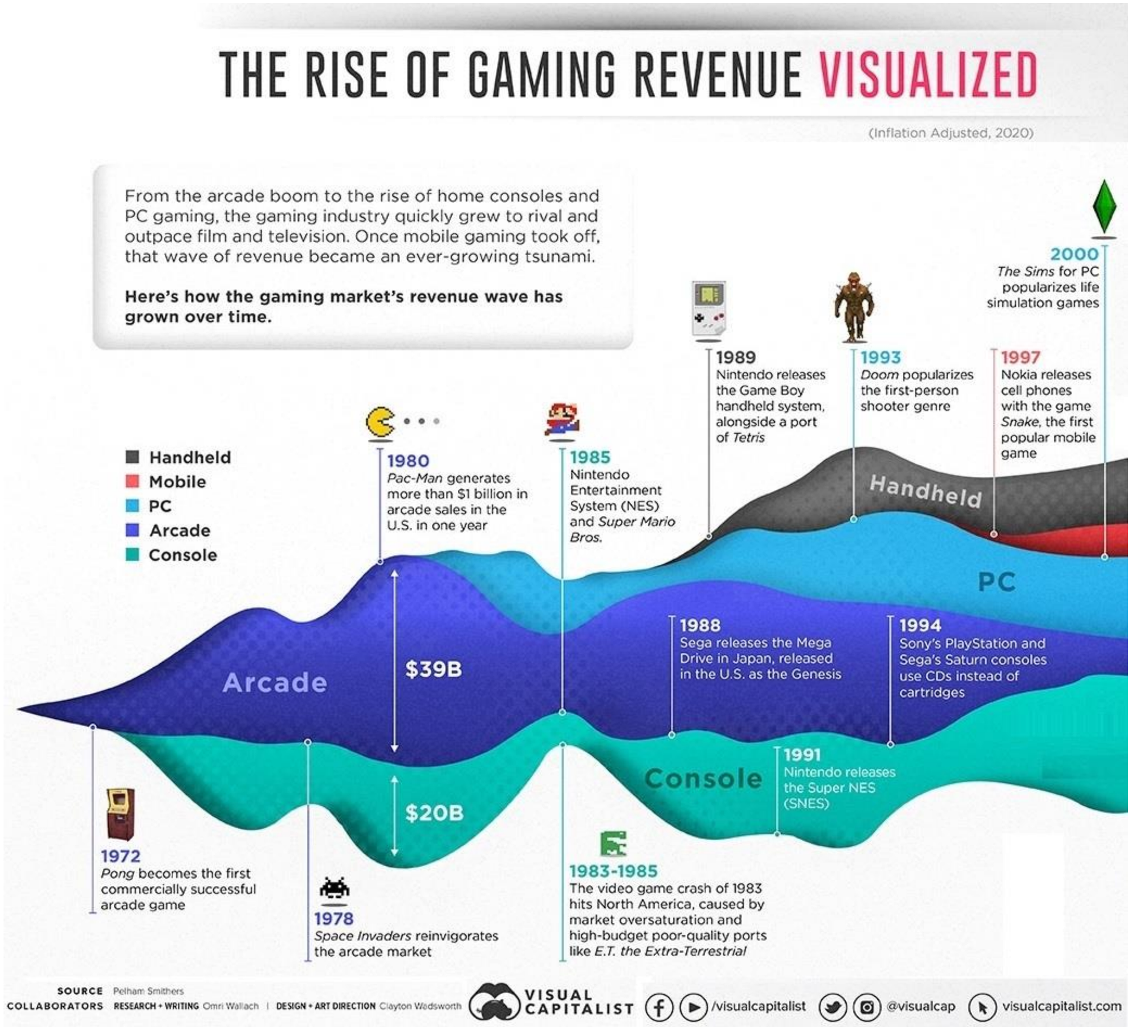


Image 6. The rise of revenue visualised from the '70s until the 2000s[10]

Therefore, during the '90s, the market saw the appearance of the 4th and 5th generation consoles. Additionally, the industry moved to CD-ROMs, which enabled games to include better soundtracks and ultimately, allowed to mainstream the use of human voices. Games also evolved technologically and smash hits such as *Sonic the Hedgehog* (1991), *Mortal Kombat* (1993), *Tomb Raider* (1996), *Grand Theft Auto* (1997) and *Final Fantasy VII* (1997) were released (Table 3). Furthermore, as Image 6 showed, mobile phones entered the scene in the late '90s and included in-built games such as the famous *Snake* (1997).

[10] https://www.visualcapitalist.com/50-years-gaming-history-revenue-stream/

1990s	The **platform** war begins to die down with the release of Sony *PlayStation* moving to CD-ROM.		
1990/ 1991	4th generation console Nintendo Super Famicom released in Japan in 1990 and as Super Nintendo Entertainment System (SNES) for North America in 1991 (UK in 1992)	1990 *Super Mario Bros. 3* (J) 1991 *Civilization* (US); *Sonic the Hedgehog* (J) 1993 *Mortal Kombat* (US) 1993 *Doom* (US)	SNES still using cartridges 64-bit machines with CPU/GPU; Cartridges replaced by CD-ROM; 2D to 3D; richer soundtracks
Time	**Platform**	**Sample of milestone games**	**Technological milestone**
1994/ 1995	5th generation console Sony PlayStation released in Japan in 1994 and in North America in 1995; Nintendo 64 released in Japan in 1995; Sega Saturn released in Japan in 1994 and in North America in 1995	1994 *Myst* (US) 1996 *Tomb Raider* (UK); *Pocket Monster* (J) 1997 *Grand Theft Auto* (UK); *Final Fantasy VII* (J); *The Legend of Zelda: Ocarina of Time* (J) 1998 *Dance Dance Revolution* (J); *Metal Gear Solid* (J) 1999 *EverQuest* (US)	Inclusion of (unsynthesized) human voice in games becoming more common Windows 95

Table 3. Technical milestones of game console evolution during the ‘90s (Mangiron Hevia and O’Hagan, 2013, p. 45-46)

1.1.3 From the 2000s onwards

The 2000s “saw the major Japanese game company Sega withdrawing from console manufacturing while Microsoft entered the market, leaving Sony, Nintendo, and Microsoft as the three console platform holders” (Mangiron Hevia and O’Hagan, 2013, p. 58). One of the biggest landmarks of the new century was the move to DVDs which provided an even larger storage capacity, supported the ASCII format, and provided better audio capacities. These advancements contributed to increasing the number of features that could be included in games and, thus, cut-scenes or cinematics became common and their quality increased during the subsequent decades. The 6th generation of consoles, which included the *PlayStation 2*, was launched and the device quickly became the “fastest-selling console in history” (Chandler and Chandler, 2010, p. 8) due to its vast library of games and its compatibility with the previous model. Microsoft’s *Xbox* was launched in 2001 and became extremely popular as an “online-focused console for the mature, tech-savvy gamer” (Chandler and Chandler, 2010, p. 9). Among the most iconic games, we find *The Sims* (2000) and the newly founded popularity of simulation games as well as the appearance of *World of Warcraft* (2004), a massively multiplayer online role-playing game (MMORPG) that will be followed by 8 expansion packs throughout the years to come. Other milestones that were achieved during the first half of the 2000s were voice recognition and motion sensor technology (Table 4).

Beyond 2000	A move from CD-ROM to DVD, *PlayStation2* further facilitates inclusion of voiced dialogue. Microsoft enters the game scene alongside Japanese console manufacturers Nintendo and Sony, while Sega withdraws from platform manufacturing. With advanced global networking as well as increased computing power, consoles allow online modes and function as multimedia entertainment centres.		
2000	6th generation console Sony PlayStation 2 launched in Japan, followed by North America	2000 *The Sims* (US)	DVD; integration of games, movies, music Xbox with internal hard drive Touch-sensitive screen; voice recognition Radio sensor controller; online networking mode; multimedia storage Blu-ray and High Definition TV technologies
2001	Microsoft Xbox; Nintendo GameCube & Game Boy Advance		
2004	Nintendo DS	2004 *World of Warcraft* (US) 2004 *Half Life 2* (US)	
2005	7th generation console Nintendo Wii; Microsoft Xbox360; Sony PSP		
2007	PlayStation3	2007 *Halo 3* (US) 2009 *Call of Duty: Modern Warfare 2* (US)	
2010	PlayStation3 motion controller Move; Xbox360 interface Kinect	2010 *Heavy Rain* (J)	Advanced motion-sensor technologies emerge
2011	Nintendo 3DS; Sony EricssonXperia Play Sony PlayStationVita	2011 *LA Noire* (AUS) 2011 *Child of Eden* (J) 2011 *Call of Duty: Modern Warfare 3* (US)	Stereoscopic three dimensional effects without requiring additional accessories PlayStation certified smartphone
2012	Nintendo Wii U	2012 *Assassin's Creed III* (CAN) 2012 *Nintendo Land* (J)	Tablet controller known as "GamePad" Games can be played offline with GamePad

Table 4. Technical milestones of game console evolution beyond the 2000s (Mangiron Hevia and O'Hagan, 2013, p. 46)

Furthermore, 2005 saw the release of the 7th generation of consoles which include the *Xbox 360* (2005), *PlayStation 3* (2006), and the *Nintendo Wii* (2006). The latter used the aforementioned motion sensor technology instead of traditional controllers and "focused on family-friendly games and casual games" (Chandler and Chandler, 2010, p. 9). Additionally, the game *Skylanders: Spyro's Adventure* (2011) provided the first example of augmented virtual reality, although the technology would not consolidate until the release of the mobile phone game *Pokémon Go* from Nintendo (2016). Despite the fact that consoles seemed to be dominating the video game market, one of the most important milestones during the first decade of the 20th century would be the entry of smartphones into the market and the appearance of app stores, which created the perfect opportunity for gaming to become mainstream. Smartphones popularised casual games with titles such as *Angry Birds* (2009) or *Candy Crash Saga* (2012) and we have since witnessed an explosion of mobile gaming that has opened the market to companies such as Apple, Google, or Amazon and now constitutes more than half of the industry's revenue (Image 7). Nowadays, after the appearance of 3D, the most interesting novelties are virtual reality (VR), augmented reality (AR), and mixed reality (MR).

> VR (Virtual Reality) replaces the real world with a simulated experience (virtual world). AR (Augmented Reality) allows a virtual world to be experienced while also experiencing the real world at the

same time. Mixed Reality provides blends that interpolate between real and virtual worlds in various proportions. (Mann *et al*., 2018)

Even though VR appeared in the '80s and Dr Jonathan Waldern presented *Virtuality* in 1990 at the Computer Graphics 90 exhibition at London's Alexandra Palace, it is only now that the technology has evolved enough to be accessible and affordable to households and is expected to continue growing.

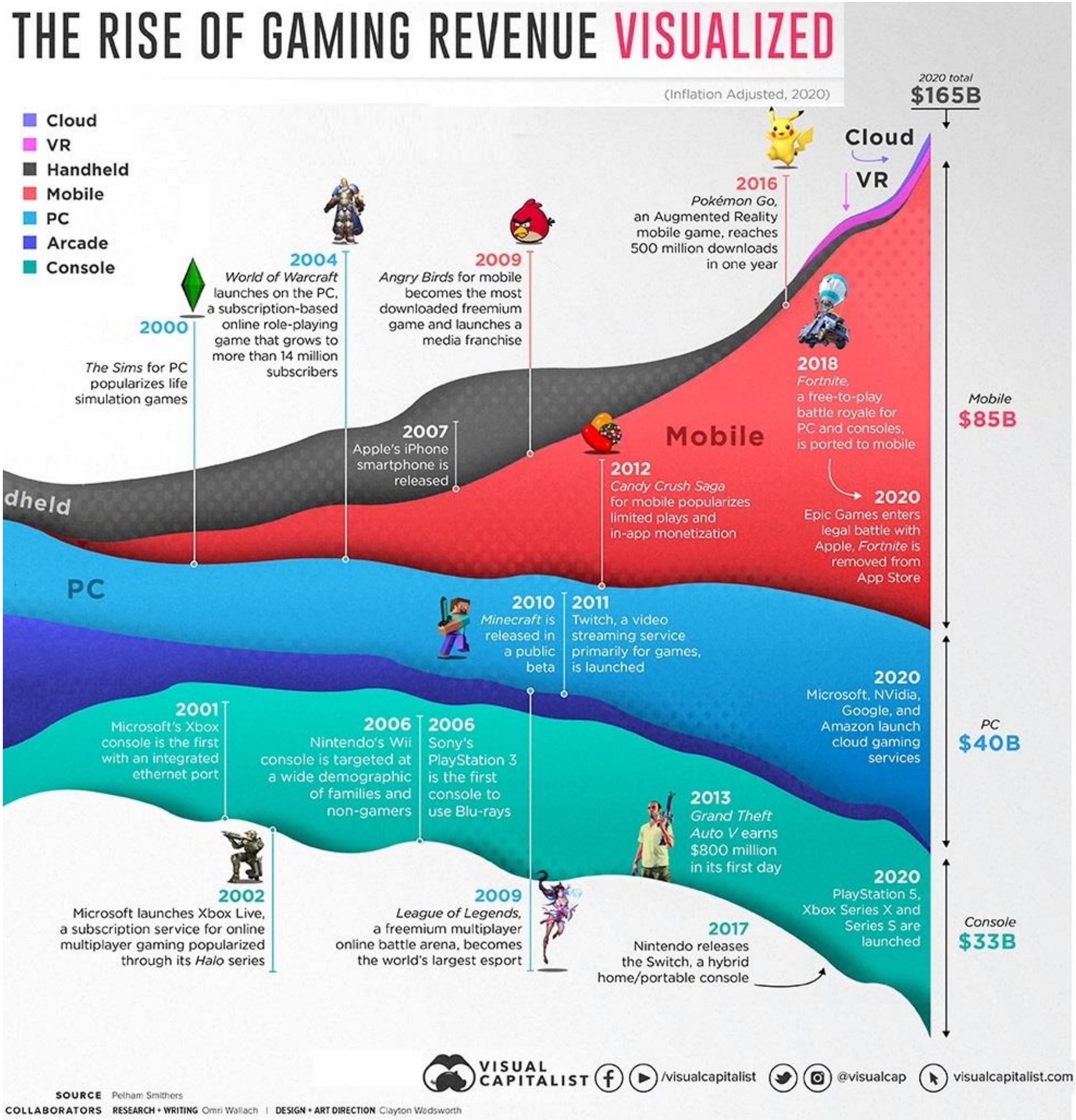


Image 7. The rise of revenue visualised from the 2000s until 2020[11]

[11] https://www.visualcapitalist.com/50-years-gaming-history-revenue-stream/

1.2 Actors

In order to understand the complexity of the processes involved in the creation of a video game, it is essential to review the different actors involved. Therefore, the present section introduces the role of platform holders, publishers, and developers as well as the impact they have on the final product. First, we will discuss the different types of devices that can be found in the market as well as the specifications imposed by platform holders. Subsequently, we will analyse the role of publishers, their responsibilities, and their involvement in the process. Afterwards, we will examine the characteristics of the various relationships that may appear in the industry and their repercussions. Finally, we will briefly review the members of a typical development team and their functions.

1.2.1 Platform holders or first-parties

The concept of platform holder or manufacturer refers to the hardware that is going to be used to play the game itself. Nowadays there are four platforms due to the disappearance of arcade machines and the advent of virtual reality: computers, consoles, mobile, and extended reality. As Chandler explains (2020, p. 18) "[e]ach platform type has differences that will impact the design and monetization of your game, such as controller inputs, technical limitations, and screen size". Furthermore, there are also differences depending on the operating system of the platform in question due to the diversity of hardware. Therefore, each platform has a vast number of standards although, for developers "three central types of standards apply to: i) hardware architectures, ii) operating systems, and iii) software development environments, including both compatibility and compliance guidelines to match the platform holder's requirements" (Laakso and Nyman, 2014, p. 16).

Throughout the history of video games, technological advances have heavily impacted the evolution of the platform themselves, as seen in the previous section. These developments are also linked to market trends and show the increasing weight of the mobile gaming market. Figure 1 shows the evolution of the revenues per device from 2012 until 2018 and the estimates for the following three years, which proved to be remarkably accurate. The graphic clearly portrays the current shift towards mobile gaming to the detriment of the other types of devices. Nowadays almost all platforms use online stores "Apple, Google, Microsoft, Sony, and Nintendo all support digital storefronts that cater to their specific platform. Like Steam, they provide a full infrastructure for marketing and distributing games. The difference is that you must apply to become a licensed developer" (Chandler, 2020, p. 21). Nevertheless, different developments in the industry seem to have created a bridge between divisions due to hardware and systems. As a result, we find browser games that can be played using any platform that grants access to a web browser; or video game streaming services that considerably reduce hardware requirements (Laakso and Nyman, 2014).

Figure 1: Revenues per segment 2012-2021[12]

Additionally, nowadays developers have access to tools such as Unity and Unreal Engine, the two most widely used game engines according to the results of our third survey, that help teams manage multi-platform releases by simply selecting the desired platform (Image 8). Therefore, whereas before they had to choose the platforms in advance and "were subsequently more or less locked into that platform" (Laakso and

[12] https://venturebeat.com/2018/04/30/newzoo-global-games-expected-to-hit-180-1-billion-in-revenues-2021/

Nyman, 2014, p. 19) now they are more independent. Nonetheless, other specifications such as the quality of the graphics and animations or how to map actions must be taken into account and need to be planned for in the pre-production phase. For example, a keyboard "allows for a lot of player-input control since there are more buttons to map actions, and the keyboard and mouse configuration allow for more accurate targeting and shooting" (Chandler, 2020, p. 19). Therefore, computer games provide more options in terms of the variety of actions that a player can perform compared to mobile devices. Additionally, as the author explains (Chandler, 2020, p. 19):

> When games are released on multiple platforms, think about what design elements work best with a given platform, and design accordingly. The game should provide a consistent experience on all platforms but also take into account the platform differences in order to provide the best experience.

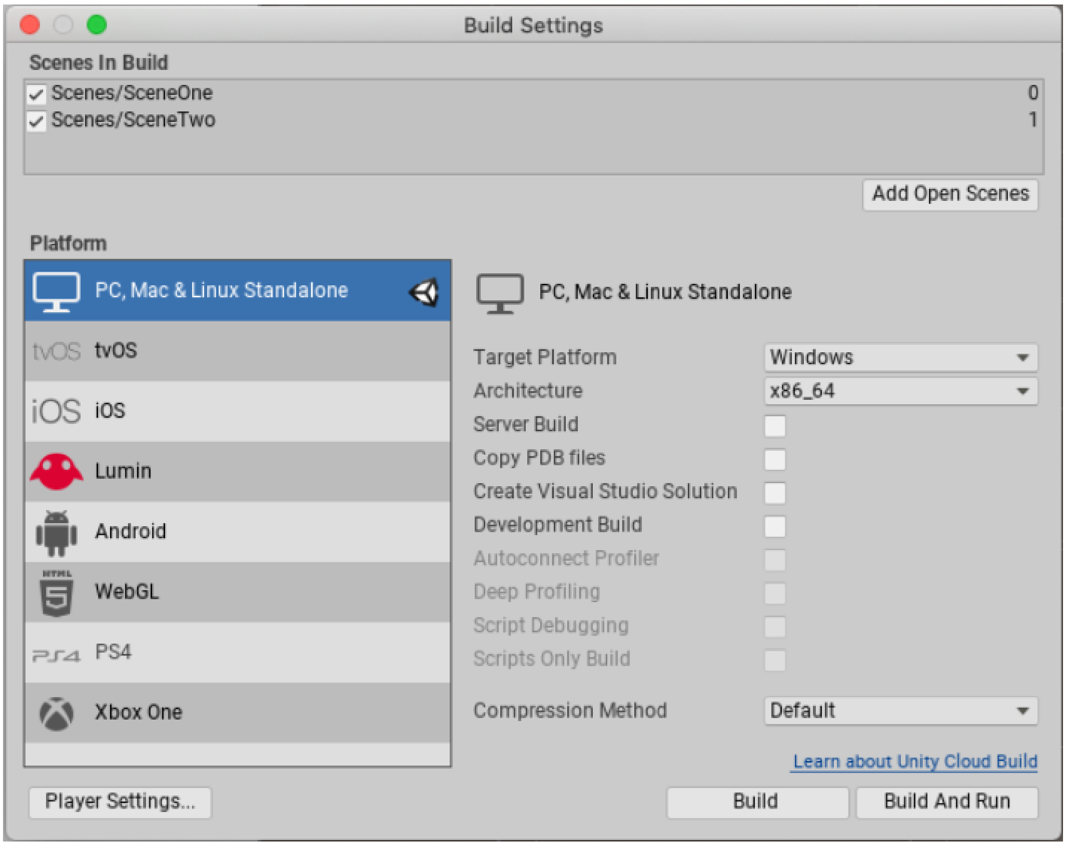


Image 8. Unity's standalone build settings[13]

[13] https://docs.unity3d.com/Manual/BuildSettingsStandalone.html

Furthermore, some platform holders such as Sony and Microsoft require a proposal of the game's concept even before the production itself and "[i]f the developer skips this step in the process, they might find that the fully developed title is rejected out-right before it is even submitted for final approval and manufacturing" (Chandler and Chandler, 2010, p. 37). The abovementioned list of technical requirements is usually provided in the form of documentation "and cover[s] all aspects of the game, such as how to word specific pop-up messages, how to set up the friends lists, and so on" (Chandler and Chandler, 2010, p. 37). In order to improve the communication between developers and manufacturers, the latter tend to appoint an account manager in order to "help the developer navigate the submission process" (Chandler and Chandler, 2010, p. 37) since all the requirements must be met and implemented. Additionally, the platform holder may also provide feedback on the game and suggest some modifications for an optimal user experience.

1.2.1.1 Personal computers

The first section showed how the first video game was specifically designed for a computer even though it was not until de democratisation of personal computers and the standardisation of operating systems that computer games started to gain popularity. Additionally, if we combine the results for PC games and browser PC games issued from our third survey, they are the most commonly developed type of games by the participants with 41.91% of the total (see Chapter 4 section 4.2.5). "As a platform, computers are the most flexible when it comes to choice, because there are multiple viable producers for each major platform standard (i.e., hardware, operating system, software development environment)" (Laakso and Nyman, 2014, p. 16). In terms of current operating systems, Windows seems to be the most popular for games, although many are also released for Mac (and some for Linux) or simply created in order to work on all three systems. Video game developers need to plan in advance if the game in question is to support all three systems in order to accommodate the necessary modifications during the engineering phase. Other considerations include the fact that "PCs have a lot of processing power, random access memory (RAM), and hard drive storage, which allows PC games to display more realistic graphics and animation" (Chandler, 2020, p.19). Conversely, due to the fact that the technology involved in developing games is capable of creating products of remarkably high quality, gamers need to continually upgrade their devices in order to make sure that they meet the requirements of the latest games. Additionally, "the wide variety of PC configurations makes it difficult for developers to test the game to ensure that it works correctly on all computer configurations" (Chandler and Chandler, 2010, p. 3). However, in terms of entry barriers, "they are easily accessible and don't require permission from a third party to develop on it" (Chandler, 2020, p.19).

1.2.1.2 Consoles

The evolution of consoles since their appearance in the '70s has also been significant, going from a remarkably low number of in-built games to cartridges, CDs, DVDs, Blu-ray, and online storefronts. New consoles are routinely released to the market in development cycles that are known as "generations" the latest being the Ninth Generation, which reached the stores in November 2020. The main manufacturers are Sony (PlayStation), Microsoft (Xbox), and Nintendo; these companies control the production and distribution of games for their devices and compete fiercely to dominate the market. These characteristics make developing video games for consoles relatively more difficult for independent developers (or indie devs) and small companies, a fact that, when combined with the prevalence of Non-Disclosure Agreements (NDAs), may explain their lower representation in the results of our third survey with only 19.59% of the total (see Chapter 4 section 4.2.5).

Nevertheless, throughout the years, the number of sales has fluctuated depending on the quality of the games released and the introduction of new technologies (Figure 2). Basically, "[a] console is hardware that plugs into the television and utilizes a controller for the main gameplay input" (Chandler, 2020, p.19). Therefore, one of the main considerations that developers need to take into account is the fact that controllers have a reduced number of buttons and that players cannot point and click as they would on a computer. Furthermore, "[p]rocessing power is more limited on consoles, so a fair amount of engineering work is required to create a game that runs within the memory limits, has goods graphics, and runs smoothly at a high frame rate" (Chandler, 2020, p. 19). Additionally, handheld consoles are also part of this category and are characterised by even simpler controllers and less power, which increases the number of things developers need to keep in mind when choosing this particular support.

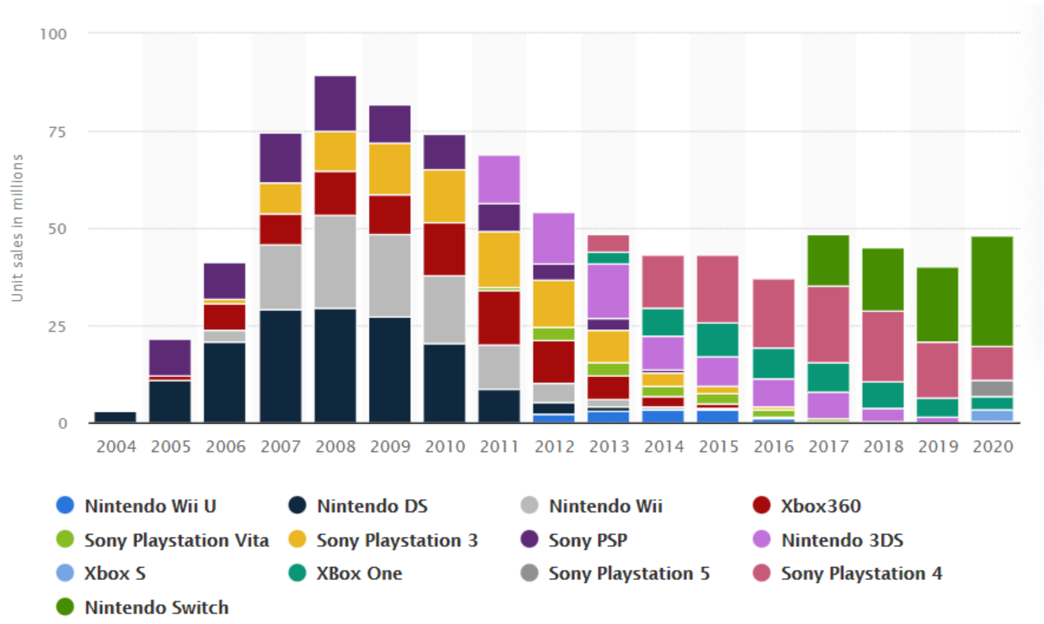


Figure 2. Units sold per console per year[14]

1.2.1.3 Mobile devices

Mobile phones, similarly to consoles, appeared in the market with built-in games such as *Snake* in 1997 and evolved until becoming the most promising device in the gaming industry. Even though the most crucial technological milestone was the emergence of smartphones in 2007, it was not until the appearance of the first App store (2008) that mobile gaming exploded. The mobile gaming segment includes both smartphones and tablets and is mostly dominated by two operating systems: iOS for Apple and Android for Google. These two companies have their proprietary storefronts and compliance with their requirements and terminology is necessary in order to distribute games through them. In mobile devices (Chandler, 2020, p. 19-20):

> As with consoles, the technology limitations impact graphics and performance. You may need to have lower resolution art assets, so they are small enough to be used with a mobile game. The touch interface comes with its own set of challenges—you have to rely on taps, swipes, and tilts to interact with the game interface. The mobile user interface (UI) must be more [sic] simplified so that players can

---

[14] https://www.statista.com/statistics/276768/global-unit-sales-of-video-game-consoles/

engage with it more intuitively. The small screen size also needs to be accounted for, which means reducing the UI elements as much as possible. In addition, mobile games are designed around shorter play sessions, so the player can pick up and play a few minutes and put it down just as quickly.

1.2.1.4 Extended reality

Extended reality, as previously mentioned, is a new umbrella term used for all types of immersive technologies such as virtual reality, augmented reality, and mixed reality. These technologies are a new addition to the gaming market and reports are remarkably optimistic about their future prospects (Figure 3). The main characteristic is the need for a special type of headset to provide full immersion (and space requirements) so players can interact with the game. There are also differences according to the hardware used as well as limitations in terms of memory and performance. Among the most popular devices, we can find *Oculus Quest 2*, *Sony PlayStation VR*, *Valve Index VR*, and *HTC Vive Pro 2*. There are many design challenges such as changes in the functions of the controller, the fully immersive environment, or the way the player interacts with the game. Conversely, augmented reality games "overlay audio and visual elements onto a real-world setting to create an interactive experience […] tend to enhance the player's real-world experience with the addition of gameplay" (Chandler, 2020, p. 20).

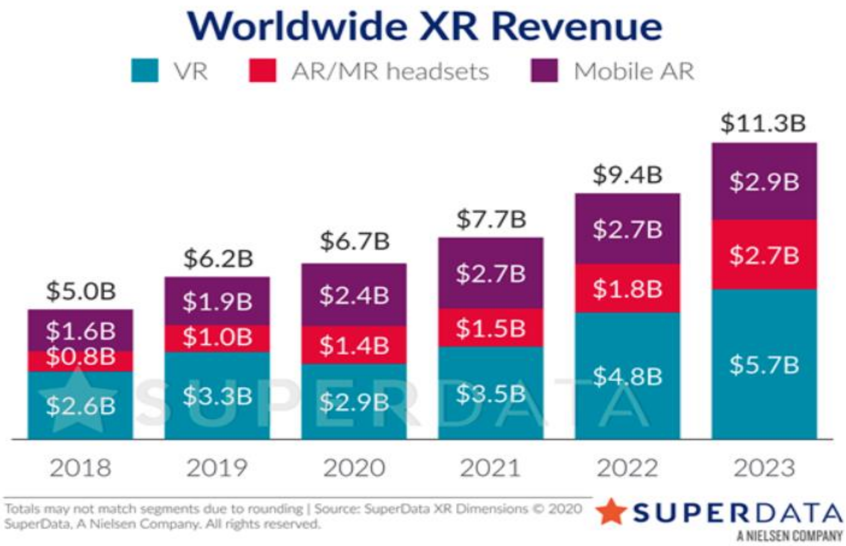


Figure 3. Sales and estimates for XR[15]

[15] https://www.statista.com/statistics/276768/global-unit-sales-of-video-game-consoles/

1.2.2 Publishers

Although years ago publishers were the only solution in terms of funding the development of a game, nowadays there are other alternatives besides self-funding (or bank loans) such as crowdfunding, angel investors, venture capitalists, grants or even releasing the game in early access (Chandler 2020). “Publishers like Activision Blizzard and Electronic Arts have defined their release calendar years in advance—much like movie studios. They have a set number of major releases in any given year, along with smaller releases throughout the year” (Chandler, 2020, p. 15). These major releases are usually handled by their internal teams and they fund smaller projects from independent developers. “The publisher’s responsibilities run the gamut from providing money and resources to manufacturing the actual boxed games that appear on store shelves. […] [I]f the game is distributed online, the publisher may provide the actual network resources needed for this type of distribution” (Chandler and Chandler, 2010, p. 14). Therefore, the publisher’s responsibilities include funding, distribution, marketing and public relations, production support, product management, live operations, community management, and customer support (Chandler, 2020, p. 35-38) As a result, a percentage of the revenue that results from the sales of the game will go directly to the publisher. As Lee Jacobson explains (Lee Jacobson, Vice President of Bussiness Development and Acquisitions *Midway Entertainment* cited in Chandler and Chandler, 2010, p. 21):

> The typical deal is the standard developer-publisher model in which the publisher funds 100 percent of the game’s development as an advance against future royalties and sales of the game. In this deal, the publisher typically provides the third-party commercial software, the tools, and the development kits. The developer is required to fulfil monthly milestones that are evaluated on a regular basis by the publisher. Various royalty structures can be brought to bear on this. As the risk profile changes from the publisher to the developer, the deal can change. There are also copublishing deals. In this instance, the game is usually fully funded by the development studio, and they are looking for a publisher who can package the game and distribute it.

Therefore, due to the importance of the role played by the publishers, they are highly involved during the entirety of the process and might propose changes to the game or veto certain features, while regularly communicating with the development team. Thus, "[a] publisher will assign a producer to work directly with the developer and represent the publisher's interests on the project" (Chandler, 2020, p. 41). This might result in the presence of two producers, the publisher's producer (PP) and the developer's producer (DP), which might create confusion if their respective roles are not clearly defined. In this dynamic, the PP will be in charge of ensuring compliance with deadlines (or milestones), validating expenses, and coordinating the tasks that fall under the publisher's responsibilities. Conversely, the DP is responsible for managing the day-to-day operations related to the development of the game, "creating the game development plan and making sure that this plan is completed during the production cycle, […] human resource (HR) issues within the team, equipment requests, and anything else that directly affects people on the development team" (Chandler and Chandler, 2010, p. 22).

1.2.3 Developers

Video game developers are in charge of all the different technical aspects of the creation of a game: art, design, engineering, audio, user experience, and quality assurance. Therefore, their role is at the very heart of the industry as their responsibility is to materialise a concept into a playable game by conceiving the story and the setting, developing all the graphics and features, and implementing all the assets. Nowadays we can observe various categories in the industry depending on their type of contract. Firstly, due to the current popularity of the industry, we find hobbyists and generalist individual developers who work on their projects in their free time and sometimes collaborate among themselves to produce small games that will be distributed for free or for a small sum of money via online storefronts such as itch.io or Steam. We can also observe an increment in the number of freelance individual developers who collaborate with studios that outsource part of the tasks or hire talent for a project in particular. These freelance developers also tend to work on their own or in a small team and develop smaller games. Afterwards, we find small independent studios that have a production team in-house and, once they have a game concept, might try to pitch it to a publisher (or fund their project by other means). According to the results of the third survey, this seems to be the most common job position in the industry (see Chapter 4 section 4.2.3.3). Figure 4 shows the stereotypical structure of a small studio, although the configurations are extremely varied and change from one company to another.

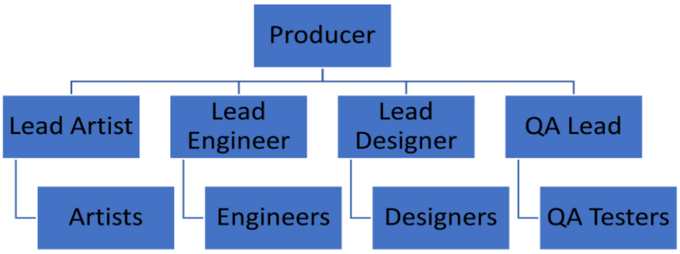


Figure 4. Small team with producer/lead structure (Chandler and Chandler, 2010, p. 59)

As previously mentioned, the economic relationship with the publisher will crucially impact the degree of freedom the developer has as well as the funding they receive. Furthermore, "[a]nother factor that influences this relationship is who brings the intellectual property (IP) to the table. For instance, the publisher will feel more strongly about a project when they provide the IP" (Jeff Matsushita, Executive Producer *Microsoft*, cited in Chandler and Chandler, 2010, p. 24). This scenario may occur when a publisher has a minor project and decides to outsource its development. "Publishers commonly send out a formalized Request for Proposal (REP), especially for licensed properties or smaller projects" (Don Daglow, President and CEO *Stormfront Studios* cited in Chandler and Chandler, 2010, p. 19). Alternatively, the developer might provide the concept and "the publisher will focus on working with the developer to ensure that there is a strong marketing effort to support the developer's vision of the game" (Jeff Matsushita, Executive Producer *Microsoft*, cited in Chandler and Chandler, 2010, p. 24). Moreover, as explained in the previous sub-section, publishers may have their own studios or in-house teams of developers who will be in charge of the larger projects.

> Wholly owned developers usually have direct access to the people making decisions about which games to develop and thus are not under as much pressure to create and pitch a game idea. If a wholly owned developer does not have an idea for a game, it is likely that the publisher will have a game in mind for the developer. (Chandler and Chandler, 2010, p. 17-18)

In-house teams are usually larger and include more specialists instead of generalist developers. Figure 5, adapted from Chandler's latest book on how to produce a video game and a previous one on video game development (2010, 2020), depicts the stereotypical organisation chart of a large development team and follows the structure of a unit overseen by an executive producer. However, the names given for each position might change depending on the company and, as usual, each corporation will have different organisation charts. Finally, platform holders also have their own in-house development teams or studios they fully own in order to completely control the production of their AAA titles. Their structure is similar to the one portrayed in Figure 5 although, once again, each company differs from one another.

Figure 5. Large development team with executive producer structure (adapted from Chandler 2020, and Chandler and Chandler, 2010, p. 60)

In order to fully illustrate the complexity of the development of a AAA game and the number of people involved, Toftedah and Engström (2019) decided to analyse the credits list of the game *Assassin's Creed: Odyssey* (2018) and draw up an inventory. According to their results, by the end of the credits "4388 persons have rolled by where 3355 are developers in one of 692 development roles […]. The development roles in the project are divided on 29 different development studios all over the world" (Toftedah and Engström, 2019, p. 3). In order to further analyse the development roles, they created a word cloud using MAXQDA (Image 9) to analyse their prevalence by multiplying "the occurrence of each role by the amount of people associated with it (i.e. if 3 persons was credited as 3D Artist the data would look like 3D Artist, 3D Artist, 3D Artist)" (Toftedah and Engström, 2019, p. 4). Their word count of the top ten most used words contained: 415 occurrences for tester, 402 for programmer, 208 for manager, 154 for designer, 127 for technical, 111 for artist, 106 for gameplay, 96 for level, 77 for localisation, and 69 for translation (Toftedah and Engström, 2019, p. 5).

Image 9. Word cloud regarding the roles credited in *Assassin's Creed: Odyssey.* (Toftedah and Engström, 2019, p. 4)

1.2.3.1 Production team

As explained in the section about publishers, producers are in charge of ensuring that the project runs smoothly, remains on schedule, and oversee the entire process. "They are also the information hub for any questions" (Chandler, 2020, p. 30). There are a variety of production positions in the industry depending on the size of the company although the most common ones are (Chandler, 2020, p. 33-34):

- **Executive Producer (EP):** Responsibilities for this role may vary. Some EPs are focused on managing the multi-year development and release plan for a game franchise, like Call of Duty. They determine the strategies for growing and maintaining the franchise […] Other Eps might oversee the process of game development and focus on broader development tasks, such as establishing employee training programs, evaluating external vendors, improving processes, determining the needs of the project, and mentoring other producers.
- **Producer:** This person usually manages an entire development team […] Their focus is on executing the plan. If the team is especially large, several producers may split responsibilities across it. […] They are responsible for keeping the team on track, solving problems, ensuring the work hits the quality bar, and facilitating the production pipeline. They also anticipate, define, and mitigate risks. […]
- **Associate Producer:** They may be tasked with producing specific parts of the game, such as the localizations or voiceover recordings. […] They are mostly focused on what is needed that week or over the next few weeks, so they will have a set of regular responsibilities, such as running daily stand-ups, doing risk analysis, and setting up production pipelines. […]

1.2.3.2 Art team

Artists are in charge of creating all the assets related to art such as “concept art, 3D models, 2D textures, and any other graphic elements in the game” (Chandler, 2020, p. 26). In order to do so, they must collaborate closely with the designers to create the representation of the characters and objects and “with engineering to determine how to utilize de technology most effectively in the art production pipeline” (Chandler and Chandler, 2010, p. 47). The number of members of the art team will vary depending on the company’s size, budget, and the needs of the project. Smaller teams will be mostly composed of generalist artists that can work on different types of assets such as being able to “create a character concept and have the skills necessary to take it from an image to a fully animated 3D character” (Chandler, 2020, p. 26). However, in larger teams, artists tend to specialise in one particular area. Chandler briefly describes the most common roles that can be found in an art team (2020, p. 26-27):

- **Art Director:** They create and manage the artistic vision. They work with other project leads or directors to shape the overall scope and requirements for the game from an artistic standpoint.
- **Lead Artist:** They manage the day-to-day work of the art team. […]
- **Concept Artist:** They create concepts for game objects, environments, and characters that the other artists use as a reference when making the game assets.
- **World Builder:** They build the game world that the characters and objects inhabit. This is sometimes considered a design position […]
- **2D Artist or Texture Artist:** They focus on creating all the 2D art or textures for the 3D models. […]
- **3D Artist or Modeler:** They build all the 3D models in the game. […]
- **Animator:** They focus on creating all the animations in the game, including fully rendered animations (highest quality) and in-engine animations (used for interactive sequences). […]
- **UI Artist:** They create the art needed for the user interface (UI), including buttons, boxes, drop-down lists, and other elements that appear in the game.
- **Technical Artist:** They focus on the technical side of asset creation. They work closely with the engineers to push the technology so that more impressive art can be included in the game. They also help the engineers build art tools that can streamline an artist’s workflow.

- **Marketing Artist:** They focus on creating assets used by publishing and marketing, including logos, website design, key art, gameplay video, and anything else marketing needs.

1.2.3.2 Design team

Designers are in charge of conceiving an immersive and engaging game for the players by devising "the control scheme, game systems (combat, trading, levelling up, etc.) narrative and story, character backgrounds and personalities, missions and objectives, level layouts, and so on" (Chandler 2020, p. 27). If the artists create the resulting image, designers "are responsible for creating all the "verbs" in the game, that is, what the player can do and interact with" (Chandler 2020, p. 27). Designers are involved in the game from the very beginning of the project and are essential during the pre-production phase where "they are brainstorming and prototyping potential gameplay ideas and then documenting the ones that work best […]. During production, they are implementing the game design, which includes scripting missions, writing dialogue, and playtesting" (Chandler and Chandler, 2010, p. 53). Additionally, they may have to redesign certain parts of the game or features to improve the game experience. The most common roles that can be found in the design team as described by Chandler are (2020, p. 28):

- **Creative Director:** They create and manage the overall vision for the player experience. […] They work with other project leads or directors to shape the overall scope and requirements for the game from a design standpoint.
- **Lead Designer:** They manage the design team and their day-to-day work. […]
- **Level Designer:** They create the level layout, mission, and objectives that the player experiences in the game. They work closely with the World Builder to bring all the art and design pieces together.
- **Narrative Designer:** They create the characters, settings, and story. They work closely with the Level Designer on the game missions to ensure that everything is narratively cohesive.

- **Writer:** They specifically write all the dialogues and in-game text. Sometimes, the Narrative Designer is also a writer (and vice versa).
- **Systems Designer:** They design any game-wide systems, such as combat, scoring, character skill progression, and AI design.

1.2.3.3 Engineering team

Engineers (also known as programmers) are the members of the team who create the code that makes everything else work—although some tools such as Unity now provide visual scripting options (a method that allows creating a game without using code). Thus, they were the subjects of the third survey as they "are responsible for creating the technology for every aspect of the game, including physics, performance, AI, graphics, scripting tools, audio, lighting, player movement, and so on" (Chandler, 2020, p. 28). Nevertheless, depending on their place of employment and their status, many of the respondents of our third survey may have had several roles simultaneously. Engineers must collaborate closely with all the members of the team in order to refine the technology and provide the means to materialise a concept or an idea. "For example, they create the technology that makes it possible for game characters to leave real-time footprints in a snowy environment" (Chandler, 2020, p. 28). The most common roles in the engineering team and their description are (Chandler, 2020, p. 28-29):

- **Technical Director:** They define and communicate the technical goals and standards for the game. […]
- **Lead Engineer:** They direct and manage the engineering team. […]
- **Network Engineer:** They focus on networking and multiplayer features.
- **Graphics Engineer:** They specialize in getting the most out of the game graphics from a performance and pipeline creation standpoint. They work closely with technical artists.
- **AI Engineer:** They create the AI behaviors of the NPCs. They work closely with system designers to define this behavior.
- **UI Engineer:** They create the functional UI screens. They work closely with the UI artists and UX designers.

- **Tools Engineer:** They work with the development team to create different tools for the development pipeline: for example, a tool that designers can use to create stats for an in-game character and import these into the game.
- **Build Engineer:** They create the tools and pipelines for checking things into the game and then compile game builds. They will also automate the build creation process as much as possible.

1.2.3.4 Audio team

The sounds included in a game are crucial in order to create an immersive experience for the player. Besides being essential for storytelling and setting the ambience of the scenes, audio can be used to provide instructions to the player and vital clues. Although large studios tend to hire a permanent team, smaller companies might outsource part of the work to service providers. Chandler describes some of the most common positions in an audio team (2020, p. 29):

- **Audio Engineer:** They focus on the technical aspects of sound design and implementation. They work closely with the audio designer.
- **Sound Designer:** They design sound effects, music, and voiceover for the game. They work closely with the writer and the narrative and level designers. They also process and implement audio assets in the game.
- **Composer:** They compose music for the game. They may also be responsible for licensing music instead of creating original content.

1.2.3.5 User Experience (UX) team

Even though not all studios have a separate UX team, UX processes are vital for the success of a video game. Whenever the company has a team, it will be composed of UX designers and UX researchers. Chandler describes UX as a mindset that focuses on ensuring "that the design and the business intentions are experienced the way they are intended by the target audience of a product, system, or service" (Chandler, 2020, p. 29). Therefore, the team will make use "of cognitive science and psychology and apply user research methodologies (e.g., playtests and analytics)" (Chandler, 2020, p. 206) in order to guarantee the immersion and engagement of the player. In other words, "looking at how a user understands and interacts with a system without the guidance of the humans who designed it." (Chandler, 2020, p. 206). She continues describing the different phases or steps that are involved in the UX process "hypotheses, planning, testing, reporting, iterating, and retesting" (Chandler, 2020, p. 210):

- **Hypotheses:** This is the first step and consists of deciding alongside the designers and engineers what are the key questions that need to be answered. Once these are determined, the UX team will start creating a protocol and defining the sources that are necessary to collect the data.
- **Planning:** The UX team starts scheduling the tests that were agreed upon, as well as the protocol to follow, which builds will be tested, etc.
- **Testing:** Participants will be recruited to carry out the tests. Those tests could be guided or walked through depending on how the tests were conceived. They will also be asked to fill out a document to provide information about game habits and other data that might be considered relevant.
- **Reporting:** The team will then analyse the data and create a report that will be reviewed in a meeting to classify the issues. Some might not be addressed whereas others might have already been solved or are being solved as the team was aware of them.
- **Iterating:** Iteration consists of making small changes and then testing them again to check for bugs or issues and then continuing to work on top of them once they are approved.
- **Retesting:** Once all the changes that resulted from the tests are implemented, the game will need to be retested.

1.2.3.6 Quality Assurance (QA) team

Testers are in charge of making sure everything works as intended since they are the last bastion before the release of the game. Even though their role is usually associated with the end of the production cycle, they should also be involved in playtesting the prototype to provide valuable feedback in order to improve the user experience. Subsequently, they also perform functionality testing on completed features of the game in order to help the development team. Finally, once the alpha build is ready, they will begin playtesting again and will continue throughout both Beta and Release candidates. The QA lead, test lead or lead tester manages the team of QA testers and is in charge of the whole debugging process. Their functions include looking for:

- **Functionality bugs:** Problems that arise when an action does not produce the expected reaction. Some examples would be when the system freezes, hangs or crashes, etc.
- **Graphic bugs:** Normally images missing, not being displayed properly, or simply not being the latest version.
- **Audio bugs:** Issues such as implementation problems, synchronisation, or quality deficiencies.
- **System bugs:** Problems such as font issues, wrong text implementations, or missing translations.
- **Linguistic bugs:** Found in both the original version and each localised version of the game. These types of bugs are subdivided into truncations, overlaps or overflows; mistranslations; terminology inconsistencies; grammatical and typographical errors; subtitling errors; confusing instructions; or style issues. (The topic will be further developed in section 1.5.5)
- **Compliance testing:** Testers need to make sure the game follows all the strict requirements imposed by the platform holders. As previously explained, these requirements vary from one platform to another, and any error could get the game sent back and cause a bottleneck in the development process.

Finally, the QA team also performs other types of tests such as compatibility testing, soak testing, regression testing, load testing, multiplayer testing, etc.

1.3 Video game development tools

This section will present the tools that were included in the first and third surveys due to their importance in the process and aims at providing a basic understanding of how they work and what they are used for. Given the wide variety of tools available in the market, we will only briefly describe game engines and tree-based dialogue tools as well as supply a rudimentary introduction to game production pipelines. As presented in the previous section, video game development teams are composed of various team members working separately—albeit collaborating closely—in different aspects of the game at the same time, creating the assets that will later constitute the game and that need to be integrated into the build itself (Image 10).

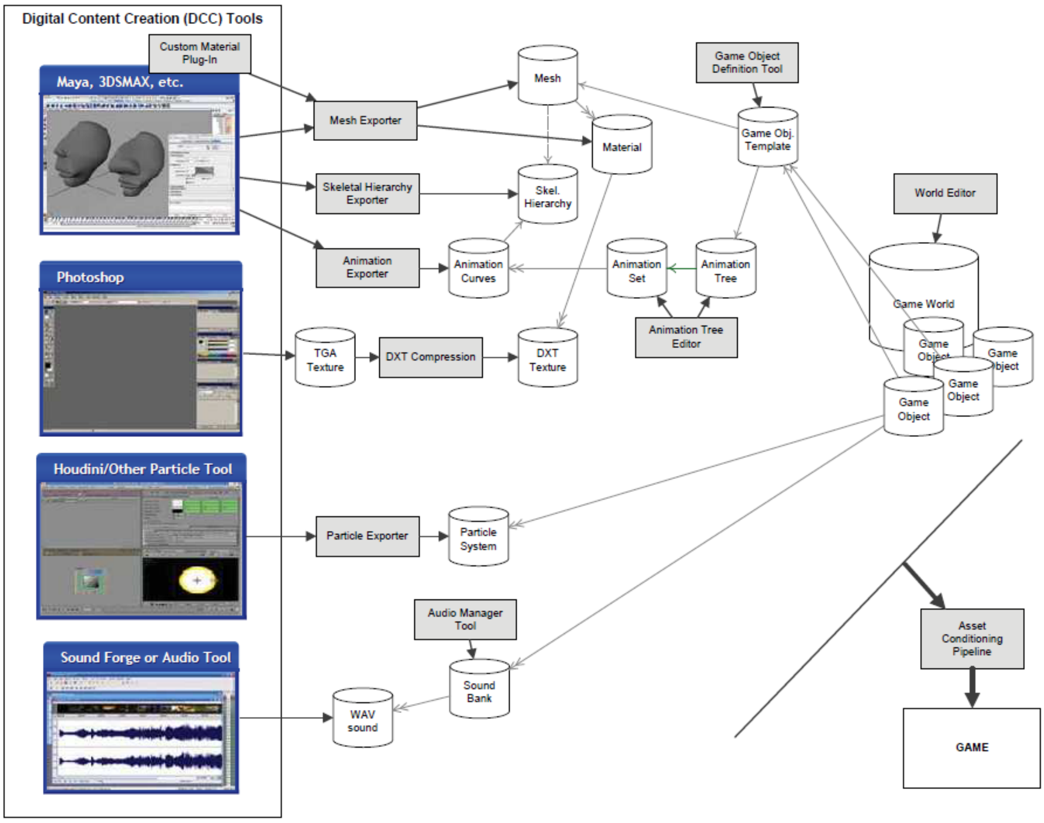


Image 10. Tools and the asset pipeline (Gregory, 2018, p. 60)

“Since games are not only an entertainment product but also a complex technical system […], game development is a complex task where system engineering and creative competences in art and design must be handled in the same project infrastructure” (Toftedah and Engström, 2019, p. 2). The asset pipeline (or production pipeline) is simply “the sequence of processes that take assets from their source form (usually the direct output of whatever package the artist created them in) to the final data […] to form part of the finished game” (Carter, 2004, p. 6). In order words, the pipeline is a series of automated processes that can be used to integrate already finished assets into the game to allow developers to view the said assets in the game itself and test or examine them in order to iterate as fast as possible. “[S]ince the vast bulk of the asset creation process is a iterative process (a succession of alterations and “tweaks” until the desired result is achieved)” (Carter, 2004, p. 6-7). Additionally (Carter, 2004, p. 8-9):

> The assets used in the construction of a game fall into many categories—sounds, artwork, music, maps, and so on, but they all share certain common features. In general, each asset forms an individual element of the game experience, and there is a hierarchy in which assets get combined to form one "gameplay unit" or a single scenario, level, or such, which constitutes a coherent chunk of game experience. […] Of course, not all games break up neatly into separate chunks like this. In particular, games that offer a free roaming world without fixed levels often cannot be effectively divided into sections along "gameplay units" boundaries. Instead the most commonly used approach is to break the game into functional units, separating assets by type, so for example all of the characters form one package […].

Game production pipelines are characterised by the fact that they can be modified in order to accommodate the needs of the project and the game's genre. Toftedah and Engström (2019, p. 12) classified the tools used in video game development into two groups, pipeline tools and non-pipeline tools, and proposed a taxonomy in order to better categorise them. Their proposal (Image 11) divides pipeline tools into 3 groups: product-facing tools, user-facing tools, and tool-facing tools (Toftedah and Engström, 2019, p. 12). As the authors explain, "[t]he product facing tools constitutes the core engine that handles the game simulation and compiles the game for a target platform. […] they typically handle complex tasks such as rendering, physics and AI." (Toftedah and Engström, 2019, p. 13). They define user-facing tools (Toftedah and Engström, 2019) as those used in order to create the content that will be included in the game itself and explain that these tools are specifically "designed to support human developers to create game content" (Toftedah and Engström, 2019, p. 13). Although some of these tools may be integrated into the game engine, others can be used separately and the assets produced will be implemented either manually or automatically. Due to the complexity of video games, there is a myriad of user-facing tools utilised for the creation of digital content that include "writing editors, 2D drawing tools, 3D modelling software, Integrated Development Environments (IDE) for programmers, audio mixers, etc" (ibid). Finally, the authors describe the last type of pipeline tools as those that "create bridges between different tools in the production pipeline or to add functionality with a middleware" (Toftedah and Engström, 2019, p. 13).

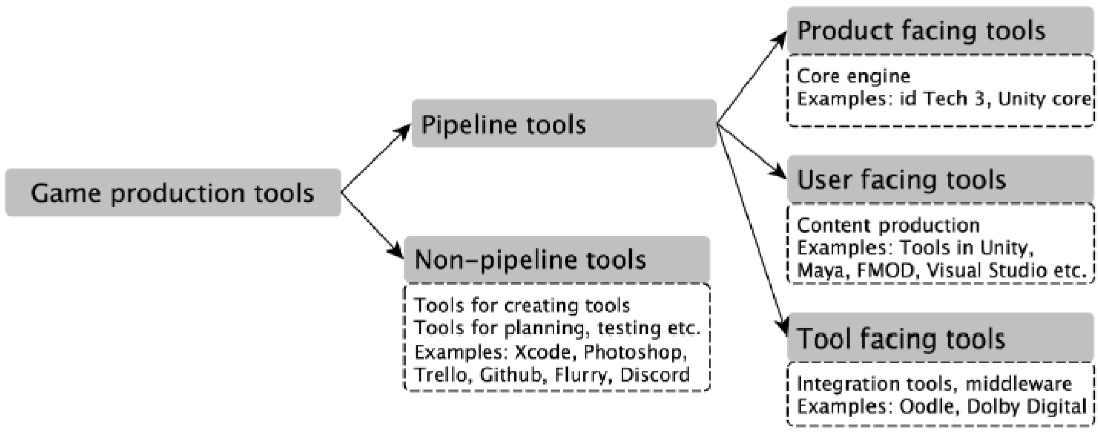


Image 11. A proposed taxonomy of game production tools (Toftedah and Engström, 2019, p. 12)

Middleware refers to software created by third-parties "that is situated between OS and device drivers and end-user application" (De Prato *et al.*, 2012, p. 229). In the case of video game development, the term applies to all commercial software such as game engines that can be customised in order to be used during the production process. "Other third-party tools used for modelling, texturing, bug tracking, and project management could be also considered middleware" (Chandler, 2020, p. 188). Similarly to the productivity software industry, these sets of tools are usually found in the form of Software Development Kits (SDKs), adopted by developers in order to reduce production costs while accelerating the process. As De Prato *et al.* explain, "a first generation of third-party separated middleware modules (graphics engines or renderers) appeared between the late 1980s and the early 1990s" (2012, p. 229). Eventually, during the mid-90s, technological advances along with the normalisation of the use of 3D in games and "the increasing complexity of applications pushed the development of what were starting to be called game engines further" (De Prato *et al.*, 2012, p. 229). According to a survey carried out by Wang and Nordmark in 2015, the use of middleware gained popularity over the years. In their article, the authors presented the results of a survey they conducted in order to analyse game developers' attitudes towards the software architecture they use to develop video games. Their findings include the fact that the game genre affects the developer's choice of tools, that developers tend to modify and customise game engines, and an increase in the use of middleware (Table 5) which was confirmed by the results yielded by our third survey (see Chapter 4, section 4.2.6).

ID	Statement	Agree	Neutral	Disagree	N/A
Q17	Today our company uses more 3rd-party modules than 3 years ago	46%	15%	8%	31%
Q18	It is easier to develop games today than it was 5 years ago	77%	8%	15%	0%
Q19	Middleware is more important to our company today than 3 years ago	55%	15%	15%	15%

Table 5. Evolution of the use of middleware (Wang and Nordmark, 2015, p. 281)

Therefore, developers must evaluate what software architecture will be used for the creation of the game when establishing the game requirements and decide about the game engine, the tools used for art assets, those used for scripting, or any other elements (Chandler and Chandler, 2010, p. 163). "The vast majority of all game assets creation tools fall into one of two categories. They are either […] commercial "off the shelf" tools […] or they are a custom tool, created in-house for a game-specific or platform-specific task" (Carter, 2004, p. 17). However, many video games are developed using a combination of both types in order to leverage the benefits offered by each option. On the one hand, in-house technology "has no licensing fee; in-house experts are readily available to fix bugs and add feature enhancements; and the technology can be specifically tailored to the game " (Chandler and Chandler, 2010, p. 163). Nevertheless, creating an in-house tool might become more expensive than a commercial tool and turn into a time-consuming endeavour. On the other hand, middleware reduces the effort and time spent developing tools, it is reusable, and providers usually have technical support in order to guide the development team if there are issues when they integrate the tool into the production pipeline (Chandler, 2020, p. 188). Conversely, besides the potential learning curve involved in the use of a new tool, the licensing fee may dramatically impact the budget. However, "[m]any middleware providers price their products competitively and provide excellent technical support to make it easier for the developers to decide to use it" (Chandler, 2020, p. 188).

1.3.1 Game engines

The term "game engine" is used in the industry to refer to a myriad of tools used in video game development and sometimes can lead to confusion since an "engine generally consists of a tool suite and a runtime component" (Gregory, 2018, p.38). Figure 6 illustrates some of the main runtime components of a 3D engine and shows the complexity of these systems. This image demystifies the concept of game engines being simple "pieces of software" and this particular representation does not include every available tool that could be part of an engine.

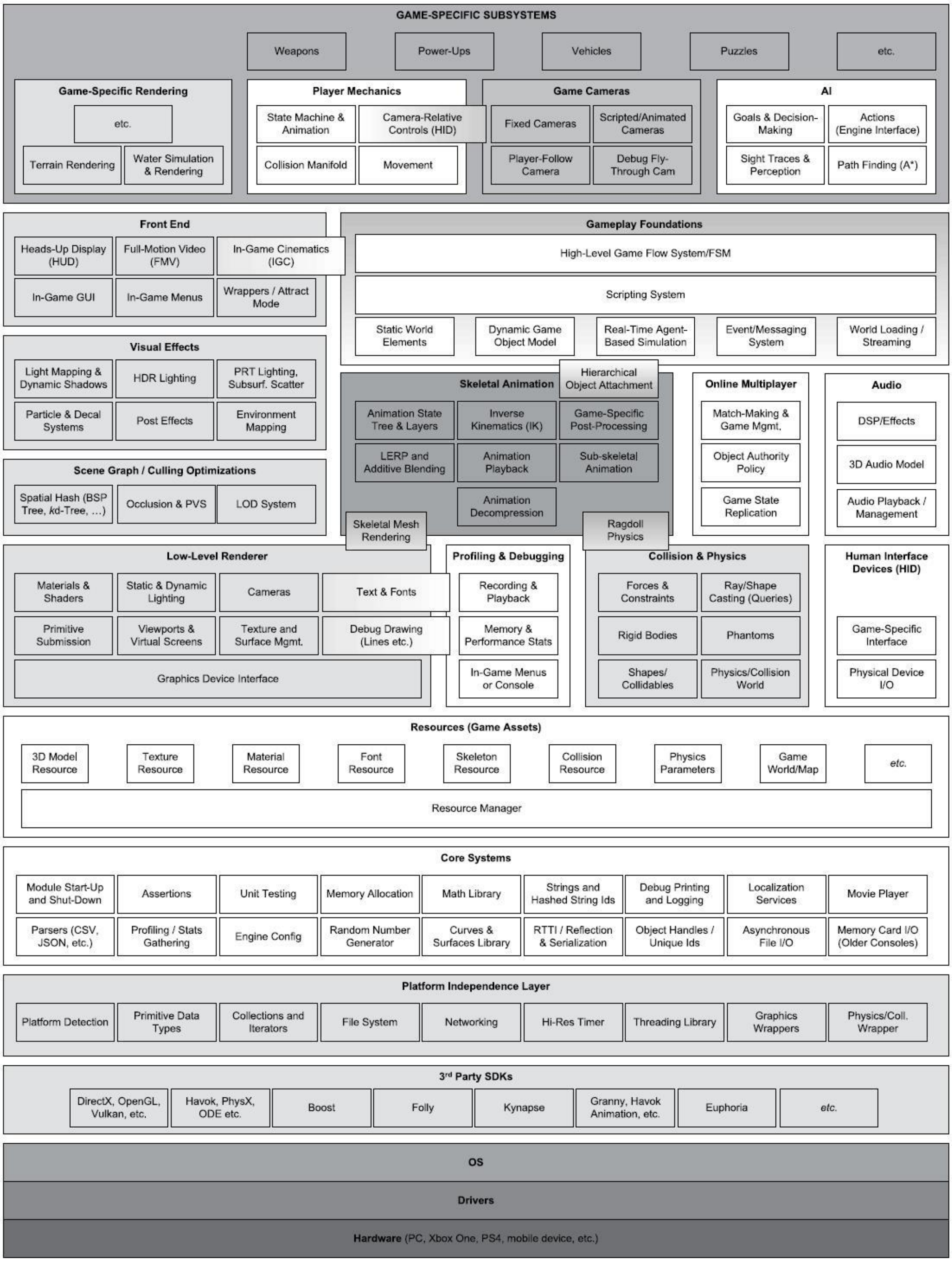


Figure 6. Runtime game engine architecture (Gregory, 2018, p. 39)

“From a software architecture perspective, a game engine is a complex system of intertwined layers relating to hardware and other software” (Toftedah and Engström, 2019, p. 8). Each layer depends on the layer situated right beneath it and the lower layer should not depend on the upper layer to avoid circular dependency, which is undesirable in software architecture (Gregory, 2018, p. 38). In Figure 6, the first lower layer represents the platform on which the game is supposed to run; then, the drivers “are low-level software components provided by the operating system or hardware vendor [that] manage hardware resources and shield the operating system and upper engine layers from the details of communicating with the myriad variants of hardware devices available” (Gregory, 2018, p. 38). The next three layers are the OS of the computer being used to develop the game, any middleware and third-party SDK used to design the game, and a layer that allows creating a game capable of running on multiple platforms as it “shields the rest of the engine from the majority of knowledge of the underlying platform by “wrapping” certain interface functions in custom functions over which […] the game developer, will have control on every target platform” (Gregory, 2018, p. 43). The core systems are a collection of utility software necessary to run the application and the resource manager is used in order to access game assets (or other input data) by providing an interface.

On the left side (in grey) directly on top of the resource manager, we can observe the components of the rendering engine, which is used to generate 2D or 3D images. Among the other elements present in that series of layers we can observe those used to analyse the performance and deal with bugs; those used to animate the characters; others to deal with audio; those to allow for multiple players; or “to process input from the player, obtained from various human interface devices (HIDs) including the keyboard and mouse, a joypad, […] etc.” (Gregory, 2018, p. 53). Another category that can be found in that area of Figure 6 is collision and physics, which are essential as the game needs to detect what could be called “movement and contact”. Otherwise, “objects would interpenetrate, and it would be impossible to interact with the virtual world in any reasonable way. Some games also include a realistic or semi-realistic dynamics simulation. We call this the “physics system” in the game industry” (Gregory, 2018, p. 51). The final category would be gameplay foundations, which encompass the actions of the game, the rules of the virtual world, what the characters can do, and the missions or objectives (Gregory, 2018, p. 55-56). Including (Gregory, 2018, p. 56-58):

> Game Worlds and Object Models: The contents of the world are usually modeled in an object-oriented manner (often, but not always, using an object-oriented programming language). In this book, the collection of object types that make up a game is called the game object model. […] Event System: Game objects invariably need to communicate with one another. This can be accomplished in all sorts of ways. For example, the object sending the message might simply call a member function of the receiver object. […] Scripting System: Many game engines employ a scripting language in order to make development of game-specific gameplay rules and content easier and more rapid. […] Artificial Intelligence Foundations.

The final layer at the top is where all the members of the development team collaborate in order to implement the game's features and "these systems include, but are certainly not limited to the mechanics of the player character, various in-game camera systems, artificial intelligence for the control of non-player characters, weapon systems, vehicles and the list goes on" (Gregory, 2018, p. 58). Additionally, another misunderstanding related to the term "game engine" derives from the blurry line that exists between the engine and the game it was created with said engine. As Gregory explains (2018, p. 11-12), the term was first used around the mid-90s as a result of the launch of *Doom*. The game was designed to provide a relatively clear separation between the core components and the individual assets, which allowed to modify it in order to create other games. Thus "developers began licensing games and retooling them into new products by creating new art, world layouts, weapons, characters, vehicles and game rules with only minimal changes to the "engine" software" (Gregory, 2018, p. 11). These practices gave birth to the creation of games that used remarkably customisable engines that could be subsequentially licensed to create other games. Therefore, in order to clarify the topic and provide a comprehensive definition that includes all the categories in Image 8, Toftedah and Engström proposed the following (2019, p. 13):

- A **core engine** is a collection of product facing tools used to compile games to be executed on target platforms. (Examples: id Tech 3, Unity core etc.)

- A **game engine** is a piece of software that contains a core engine and an arbitrary number of user facing tools.
- A **general purpose game engine** is a game engine targeted at a broad range of game genres (Examples: Unity, Unreal).
- A **special purpose game engine** is a game engine targeted at specific game genres. (Examples: GameMaker, Construct, Twine etc.)

Furthermore, game engines are, to some extent, specific to a particular game genre (as we can observe from the previous definitions). Among the engines considered “general purpose”, we find Unreal Engine and Unity. Although Unreal Engine was initially developed for first-person shooter games (FPS), it has now “become known for its extensive feature set and cohesive, easy-to-use tools” (Gregory, 2018, p. 32) and holds an important position in the industry (see Chapter 4 section 4.2.3.3). Unity is the most widely used tool according to our research and its strongest selling point is its cross-platform production capacities and the myriad of tools available in the assets store that enable developers to create an extremely customisable system. Other game engines include those that use the Quake technology (with some of them providing the source code for free), DICE’s Frostbite, CRYENGINE, etc. However, regardless of the tool and its origin, due to their complexity (Toftedah and Engström, 2019, p. 14):

> [A] game engine is not likely to include the whole production pipeline. Only games developed in small teams in special purpose engines can handle the whole pipeline in a single application (for instance GameMaker). Even Unity that includes a lot of user facing tools, e.g. for animation and audio processing, depends on external tools such as Visual Studio for script editing.

1.3.2 Tree-based dialogue tools

Depending on the game genre, dialogues play an essential role and can be used to add up to the players’ immersion by simulating spontaneous responses. Due to the impact of dialogues in video game localisation practices, this subsection will provide a cursory examination of some of the tools available as well as images in order to illustrate their usefulness from a localiser’s point of view. As Domsch (2017) explains, dialogues in video games can be either text-based and appear static inside a dialogue box, or spoken and recorded by voice actors. Regardless, the author points out that the “major structural difference is whether the game system uses language merely as invariable content to fill what is in essence a branching multiple-choice structure, or actually tries to process the player’s language input” (Domsch, 2017, p. 258). The latter is achieved by parsing (analysing and simplifying) the player’s input in order to reduce the number of possibilities from the game’s perspective and simulate endless options. Conversely, the former entails the creation of dialogue branches and their subsequent implementation in the game. Thus, the resulting dialogue tree can become remarkably extensive and complicated to follow without visual aid. Therefore, many tools specialising in video game development, game engines, or simply used to create interactive stories, provide a visual representation of the branching options. Additionally, these tools can help in order to create a flowchart for non-linear stories. The following image (Image 12) contains screenshots from one of Unity’s tools developed by Pixel Crushers[16] (top left), Unreal Engine (top right), and Twine[17], a tool specifically created for interactive stories.

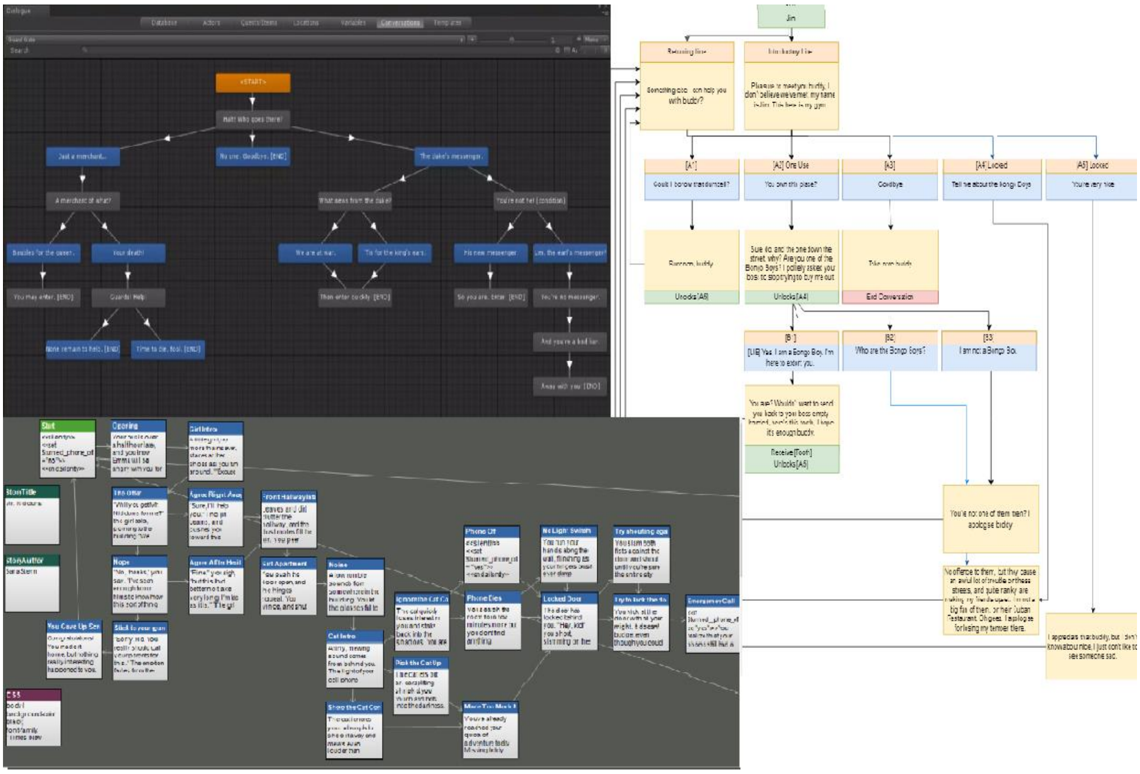


Image 12. Examples of tree-based dialogue tools

[16] https://assetstore.unity.com/packages/tools/ai/dialogue-system-for-unity-11672
[17] https://pixelkin.org/wp-content/uploads/2014/05/TwineHeader.jpg

1.4 The development life cycle

Due to the scope of this paper, this section will review the basic steps of a game development life cycle (GDLC) from a general point of view, present the different builds of a video game as well as its milestones, and introduce the different development models or processes. Depending on the company's practices or the scholar, the video game development cycle can be divided into a different number of distinct phases that range from 3 extremely basic steps—pre-production, production, and post-production (Aleem *et al.*, 2016)—to 6 stages that consider the pitching, the initiation of the project or testing as separate. Table 6 provides an overview of some of the different GDLCs that have been proposed: Blitz Game Studios' was composed of 6 phases (2011), Hendrick (2014) proposed 5 stages, Mcgrath's (2014) had 6 steps, and Chandler and Chandler's (2010) included four.

Blitz Games Studios	Arnold Hendrick	Doopler Interactive	Heather Chandler
Pitching			
Pre-production	Prototype	Design	Pre-production
	Pre-production		
Main production	Production	Develop/ Redevelop	Production
		Evaluate	
Alpha	Beta	Test	Testing
Beta		Review release	
Master	Live	Release	Post-production

Table 6. Overview of various Game Development Life Cycles (Ramadan and Widyani, 2013, p. 97)

Regardless of the denomination, the phase (or phases) situated before production include the creation of the concept, finding funding when applicable, the creation of the team, and the prototype. The production stage (or stages) includes the alpha and beta builds as well as testing and creating the release candidate. Finally, everything related to the release and distribution of the game will be done during the post-production part of the game cycle. Ramadan and Widyani analysed the previous processes and concluded that they all had in common three fundamental activities **"(1) Design and prototype:** the process of creating initial game design, game concept, and put [sic] it into a form of playable prototype, **(2) Production:** the process of making the source code, creating the assets, and integrating them as one, **(3) Testing:** the process of playtesting" (2013, p. 97). Subsequently, the authors proposed a detailed GDLC with an iterative approach that encompasses all the processes (Figure 7).

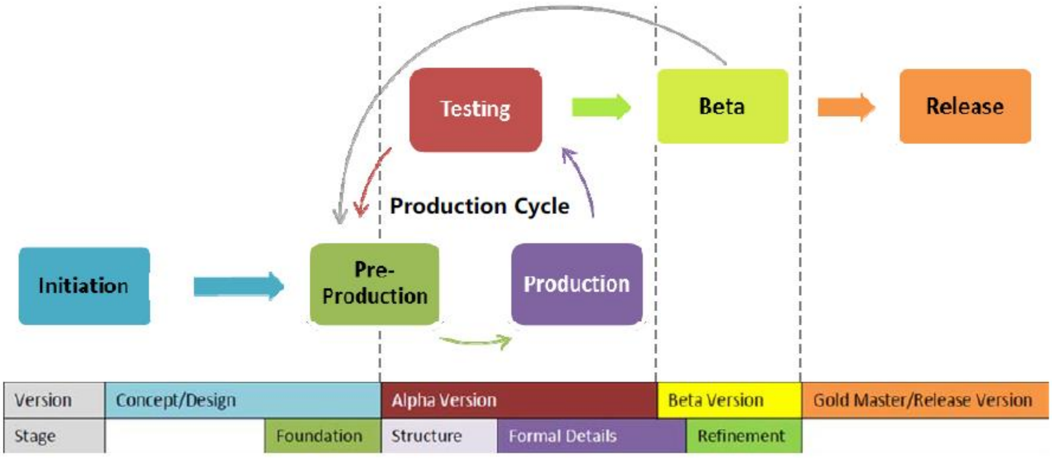


Figure 7. Ramadan and Widyani's GDLC (2013, p. 98)

1.4.1 Initiation and pre-production phases

This sub-section introduces the steps that need to be taken before commencing the production of a video game and include: creating the concept of the game, prototyping, laying out the requirements, creating the documentation, (in some cases) pitching the game, and assembling the team if necessary. As Chandler explains (2020, p. 81-87), the concept is the basic idea of the game, which usually begins with a very broad topic, concept, or question that will be subsequently fleshed out in order to create a more defined idea that includes the genre, the potential game mechanics, and the features among other things. The team in charge of creating the initial concept will gradually develop said ideas until obtaining the core design of the game, which should include: the goals, the hook (used to attract the players to the game), the core game loop (the actions the player can perform in the game), the genre, the target platform (or platforms), the target audience, and the game's revenue model (one-time payment, subscription, downloadable content, or free-to-play) (Chandler, 2020, p. 81-87).

Once the concept of the game is as detailed as possible, the team will create a prototype that includes the main idea and the desired key game mechanics to study the project's viability. Prototypes can also be created at any time of the game's production life cycle in order to assess certain levels or ideas. "Prototypes can also be used to test out new content creation pipelines, different art techniques and styles, or anything else that is not well-defined that needs to be communicated to people" (Chandler, 2020, p. 97). Therefore, the main goals can be to explore new possibilities, experiment with features or systems, iterate by creating a first version to improve on, or even create a vertical slice (a polished preview) that can be used for pitching the game to publishers. Additionally, the audience greatly impacts the approach (analogue or digital) and the degree of detail included in the prototype. Whereas other developers might quickly understand game mechanics, the publisher will ask for a more polished version of the game. Furthermore, if the game is released in Early Access, the developers should release a game where players are "able to interact with it and make progress, and there should be limited times where it crashes or has bugs" (Chandler, 2020, p. 100).

Table 7 represents the milestones (or quantifiable goals) that should be accomplished in the builds that have been discussed so far according to Chandler (2020, p. 71). Therefore, the prototype build (or First Playable) will include placeholder assets (both art and audio) in order to illustrate the game concept and imitate the "look and feel" of the game, even though the assets will be subsequently replaced by the final ones. Additionally, the essential game loop should be in place and functional to allow QA to playtest the build and provide actionable feedback that will be implemented along with the findings of the UX team. Conversely, the vertical slice (or beautiful corner) will provide a more detailed version of a specific area of the game for display purposes. Thus, besides the elements already present in the previous build, the game should include some key features and definitive art assets as well as sample audio. Once the features have been defined, the build should also contain basic documentation, a test localisation pipeline, and initial test plans. Finally, the production team should have the initial estimation of the costs and a budget plan, as well as a schedule and the basic game requirements (ibid).

	Prototype	Vertical Slice
Engineering	Basic functionality for a key game play loop is functioning.	Basic functionality for a few key features are in to demonstrate very basic game play.
Art	Art assets are in the prototype, but they are placeholder and not representative of the final game.	Two or three key art assets are created and viewable in the build. The assets demonstrate the look and feel of the final version of the game.
Design	Key game loop is defined and can be implemented by engineering.	Basic features are defined, key game play mechanics have basic documentation and a playable prototype if possible.
UX	First pass of UX flows are designed and implemented into the prototype.	Initial UX testing done on the game. Development team can start iteration on UX experience for key game play mechanics.
Audio	If time allows, placeholder sounds are available in the prototype. Minimal audio is needed at this stage, unless it is required to prove out a key gameplay loop.	The sound of the game is determined, including voiceover, music, sound effects. Samples are available to communicate the sound vision of the game.
Localization	Start planning localization friendly pipeline.	Set up and test localization pipeline for translation, integration, and testing. Content checks done to ensure that appropriate content can be created for each territory.
Production	Initial plan for creating and playtesting the prototype.	Basic game requirements are defined. Initial development plan, budget, schedule are completed. Initial financial forecasts are completed. Build pipeline is established.
QA	Playtesting the prototype and providing feedback.	Write initial test plans and automation checks for the game. Focus on helping development team test core functionality as it is implemented in the game. Some functional testing begins on completed features.

Table 7. Milestone plan prototype and vertical slice (Chandler, 2020, p.71)

The game requirements “include the basic art, design, and engineering features that must be supported, any constraints on the project, and basic technical and design documentation” (Chandler and Chandler, 2010, p. 101). Some examples would be creating a list of essential features and desired features with a view to defining the priorities and establishing the technology needed. Once these are determined and the documentation becomes more refined, the final files should be drafted and acknowledged by all the members of the development team. The basic requirements (or game plan) include the budget, the staffing plan and establishing a schedule as detailed as possible that includes time estimates, the allocated resources, and the dependencies. The latter is crucial as, for example, the artists will need to wait for the approval of the initial layout by management in order to start creating the prototype. In the same manner, before the QA team begins checking the textures, the objects must be created first and then the textures need to be applied (Chandler, 2020, p.177).

Moreover, before the beginning of the production phase, two more aspects need to be final: the documentation and the production pipeline. The documentation and its configuration vary immensely from one development team to another, and each discipline will use different types. Artists will need documents such as a style guide (types of fonts, colour schemes, etc.), a complete asset list, or instructions about the tools. Designers require character bibles (that describe each character in the game), story bibles, or the script of the game whereas engineers must have technical documents that include coding standards and the technical design of the game. QA uses the documentation created by the design team in order to formulate testing plans, etc. Finally, the production pipeline must be fully functional to start implementing assets into the game, validating the said assets, and start planning for dependencies. As Chris Schweitzer explains (Chris Schweitzer, producer of Interactive Entertainment cited in Chandler, 2020, p. 194):

> Let me use the Weapon Creation pipeline, from one of the games I worked on, as an example. We realized that in order to finalize a weapon, we needed QA to do a pass. So, QA testing is the end of the pipeline. Before QA starts, audio will need to be finished. Before audio can start, visual effects (VFX) needs to be finished. Before VFX, we need animation to be complete, and so on… We identified the pipeline steps needed and represented the dependencies; the next step was to identify the owner.

1.4.2 Production, testing, and Beta phases

The production phase entails the creation of the assets, the implementation of the game plan, and the completion of the tasks that had been stipulated. Nowadays, games are developed following agile development (see section 1.5.4 of this chapter for more information about the models and Chapter 4 section 4.2.8 for the survey's results) which consists of iterating by testing, modifying, and then building upon the previous element. Therefore, the first build created during the production stage is an improved version of the prototype (or the vertical slice). "Builds are typically used for demonstration purposes, more comprehensive testing, or as submissions to become a final version of the game" (Carter, 2004, p. 10). Table 8 portrays the milestone plan for the Alpha and the Beta builds, both of them part of the production stage of the video game development life cycle. As the table shows, for the Alpha, around half of the art and audio assets are final and have replaced the initial placeholders, the game has already been adapted for the target platform and all the features are functional (albeit not definitive). Half of the tasks of the design team are completed, the QA team can start playtesting, and the UX team is trying out the features to ensure that they produce the desired effect in the intended public. Additionally, before finalising the compilation of the Alpha build and moving towards Beta, the localisation phase should be completed and the localised strings implemented to start linguistic testing (ibid). Once Alpha is done and the completed assets and features have been reviewed and tested, the development team will move towards releasing the Beta build. However, as shown in Ramadan and Widyani's GDLC, "[t]esting is ongoing during the production process, as the quality assurance (QA) department will check milestone builds, new functionality, and new assets as they become available in the game" (Chandler and Chandler, 2010, p. 107). Once in the Beta phase, all art, design, and audio assets should be final and implemented, as the build may need to be submitted at the manufacturer's request. Beta is mostly devoted to fixing the bugs encountered by the QA team and potential minor changes to apply the feedback produced by the UX team. Thus, during this stage, the QA team will put in place the validation plan created during the pre-production phase by checking and validating all areas and assets as well as "regressing bugs that the development team has fixed" (Chandler and Chandler, 2010, p. 107). Additionally, "[a] game in beta might also be released as an Early Access title. […] The game should be at Beta around 75%-85% through development" (Chandler, 2020, p. 68).

	Alpha	Beta
Engineering	Key game play functionality is in for all game features. Features work as designed, but may be adjusted and changed based on feedback. Game runs on target hardware platform.	Code complete, only bug fixing from this point forward.
Art	Assets are 40 – 50% final, with placeholder assets for the rest of the game.	All art assets are final and working in game. Only major bug-fixes from this point forward.
Design	All design documentation is completed. Feature implementation is in progress. 40 - 50% of design production tasks are completed. Major areas of game are playable as designed.	All design assets are final and working in the game. Only major bug fixes from this point forward. Minor game play tweaks can be done, based on playtest feedback.
UX	Continued UX testing on features as they are added and iterated on in the game.	Final UX testing on the game. Final UX feedback and polish are implemented.
Audio	40 – 50% of sound effects are in and working. Voiceover design is in progress, placeholder VO files are recorded. Music in progress of being composed.	All final sound assets are in and working in the game.
Localization	Text is finalized and sent for translations. Translations are integrated into the game. Initial linguistic testing can begin.	Localization is complete, only bug fixes from this point forward.
Production	Full production has begun. The game requirements and game plan are fully completed and approved. If working with licenses, all licenses are secured and an approval process is in place.	Development team is done with tasks. Production starts bug triage and focuses on burning down bugs until there is a suitable release candidate.
QA	Game is now playable a full game, although there are some rough edges and holes in some of the functionality. Playtesting can begin. Can test against the alpha deliverables expected for this milestone.	All aspects of game can be fully tested and bugged. Some playtesting continues in order for design to put the final polish on the game.

Table 8. Milestone plan Alpha and Beta (Chandler, 2020, p.71)

1.4.3 Release phase and other post-production activities

The last step in the process is the creation of the Release Candidate (RC) or Post-Gold version and "product launching, project documentation, knowledge sharing, post-mortems, and planning for maintenance and game expansion" (Ramadan and Widyani, 2013, 99). However, in order to create the final build, the development team must initiate the code release process, which consists of a final and more profound check to "confirm that it is ready to be shipped to the manufacturer" (Chandler and Chandler, 2010, p. 107). As Table 9 shows, during this final phase, the testers start performing specific tests (compliance testing) to ensure that the game complies with the manufacturer's requirements and only critical and crash bugs will be fixed at this stage. Once the RC build is final, "the release version must be submitted to the appropriate platform holder for approval before it can be officially released. The platform holder may find a few issues that need to be addressed by the development team" (Chandler, 2020, p. 69). If the game is sent back due to these issues, the team will have to resolve them and submit the game once more to the manufacturer for approval. For this purpose, "the console manufacturer will generate a report detailing what caused the game to fail" (Chandler and Chandler, 2010, p. 275).

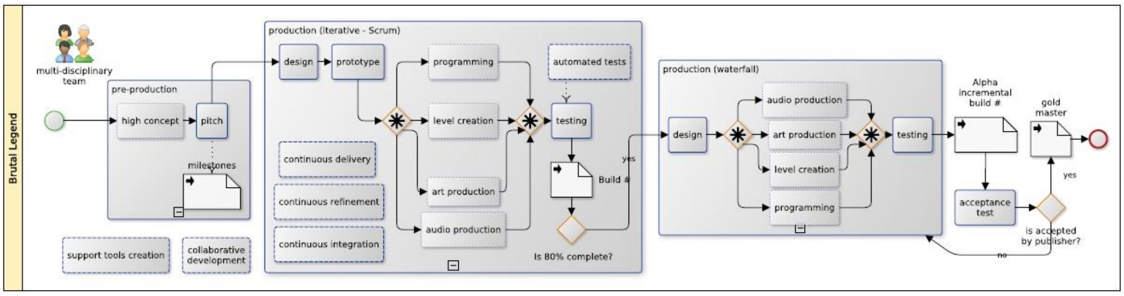
	Release
Engineering	Full code freeze. During this phase only crash bugs can be fixed. Critical bugs can be fixed with approval.
Art	Full art freeze. No art fixes, unless it is to fix a crash bug.
Design	Full design freeze. No design fixes, unless it is to fix a crash bug.
UX	Full UX freeze. Nothing fixed unless a critical UX issue.
Audio	Full audio freeze. Nothing fixed unless a critical audio issue.
Localization	International software ratings are finalized. Localized versions are approved by appropriate territories.
Production	Release pipeline is finalized. Release plan is in place. LiveOps, community management, and publishing are ready for release and post-release activities.
QA	Begin testing RC candidates. Run checks for third party technical requirements. Testing patching and live services pipelines.

Table 9. Milestone plan Release (Chandler, 2020, p.71)

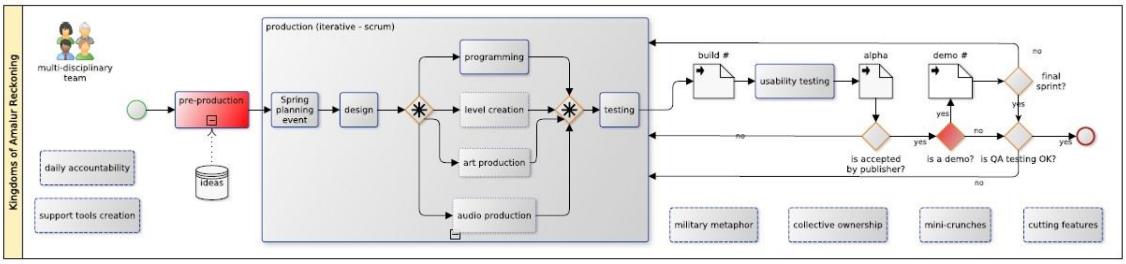
1.4.4 Game development processes

The processes used in game development are similar to those used in software development and can be classified into: waterfall or predictive, iterative or agile, hybrid, and *ad hoc* (Politowski et al., 2016). As previously explained, the market has shifted towards iterative (or scrum) practices as opposed to the traditional waterfall methods. Thus, the waterfall or predictive method is characterised by "a sequential process in which a next phase is started only if the previous phase is completely finished, delivering business value all at once" (Politowski *et al*., 2016, p. 1-2). Conversely, agile practices consist of developing a game "by repeating short-cycles to deliver a ready-to-use feature each time" (Politowski *et al*., 2016, p. 2). As its name indicates, using the hybrid method entails a mix of waterfall and agile practices, usually relegating the latter to the development phase and using the former for the remainder of the GDLC. Finally, *ad hoc* models are specifically created for a project on demand. In order to study current trends in the use of these models, Politowski *et al*. analysed a series of post-mortems published on the website Gamasutra to extract their processes and model them. Their findings showed that "**Iterative practices** are increasing and are applied to at least **65% of projects** in which **45% of this projects** explicitly adopted Agile practices. However, **waterfall** process is still applied at least [sic] **30% of projects**" (Politowski *et al*., 2016, p. 1). These were also confirmed by the results of our third survey, where 47.84% of the respondents preferred iterative development (see Chapter 4, section 4.2.8).

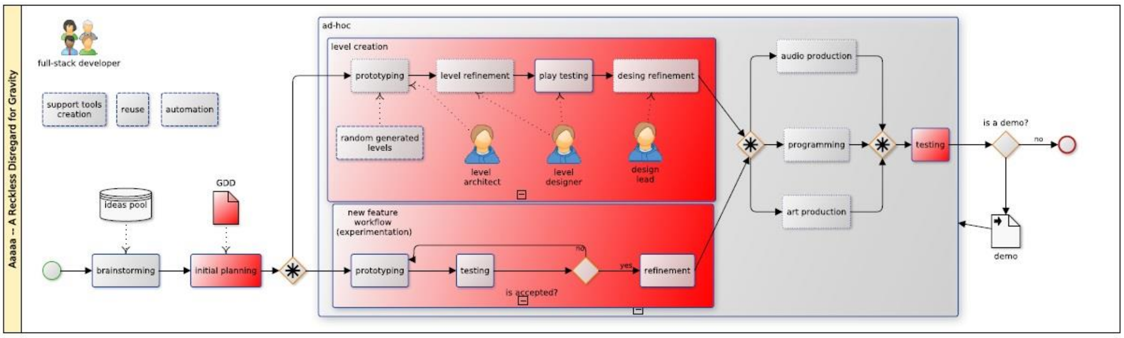
Figures 8-11 show the development model of 4 different games in order to represent each category. The first model, a hybrid process, has a clear division between the different stages of the GDLC: the pre-production phase follows a waterfall model, the production phase is iterative, and then it reverts to waterfall practices (Politowski *et al*., 2016, p. 3-4). The second game, *Kingdoms of Amalur: Reckonin*g, exemplifies the iterative model with a very short pre-production phase that is followed by a series of development cycles (ibid). The third figure represents the *ad hoc* approach, which tends to be used mostly by small teams and "production can be separated by level creation, experimentation and testing. Tasks are distributed by role, following a contextual sequence" (Politowski *et al*., 2016, p. 4). Finally, the example of a video game developed following a waterfall process. As the image shows, the different steps in the sequence are well-defined: first, the pre-production phase that leads to a design, followed by the creation and implementation of all the assets and the conception of a single build which is subsequently tested and then delivered (ibid).



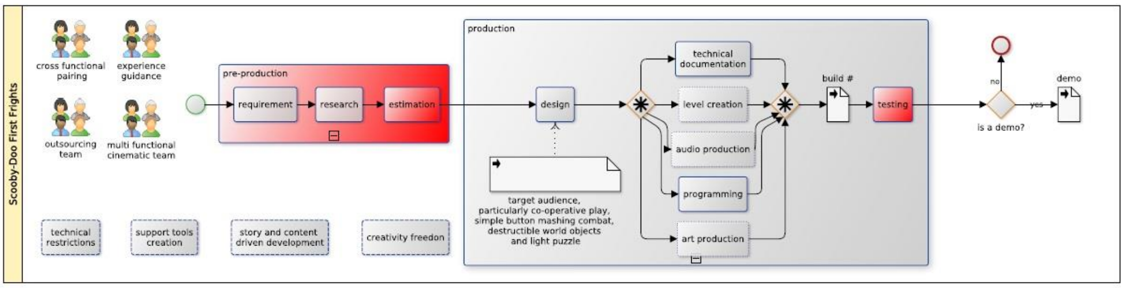
Hydrid process - Brutal Legend model



Iterative process - Kingdoms of Amalur: Reckoning model



Ad-hoc process: Aaaa! A Reckless Disregard for Gravity model



Waterfall process: Scooby-Doo model

Figures 8-11 examples of game development models (Politowski *et al.*, 2016, p. 5)

1.4.5 The LQA process

Linguistic Quality Assurance (LQA) or simply linguistic testing in video games is a crucial part of the development process. Although there are different types of testing throughout the entirety of the GDLC, it is beyond the scope of this paper to describe them all and the current subsection will only deal with testing related to language matters. Thus, we will briefly cover functionality testing linked to the localised versions of the game and subsequently focus on the LQA process proper. In the industry, two distinct QA departments will usually intervene in the testing process of the localised version of the game as "[f]unctionality testing is done to ensure that no additional code bugs have been introduced […]. Linguistic testing is necessary to confirm that the language assets are displaying properly, there are no typos, the translations are correct, and all necessary assets have been translated" (Chandler, 2005, p. 197). Developers should allocate enough time in the schedule for both testing and fixing bugs with a sufficient margin to deal with previous delays and thus, the schedule should include several rounds of LQA to eliminate as many bugs as possible. Chandler (2005, p. 199) recommends planning for at least three rounds and then adding more if the game contains copious strings. The author explains that (Chandler, 2005, p. 199-200):

> The initial round will uncover a large number of linguistic bugs. Some functionality bugs may also be uncovered that need to be addressed before the tester can check all the localized assets in the game. The first round will also uncover text, voiceover, and other assets that have not been translated. […] The second round of linguistic testing allows the tester to check all the bugs fixes and provide additional information on bugs that have not yet been fixed. They will also uncover a few more bugs. […] The third round of linguistic testing should focus primarily on verifying all the linguistic bug fixes and less on finding additional bugs to be fixed.

Therefore, during the pre-production phase, the QA team should design the LQA testing plans to be used as a checklist. Said list can be an adaptation of the plans used for the source version of the game as "[l]inguistic testers will not follow the English functionality test plan to the letter, but it will provide them information about what areas need to be accessed and checked for linguistic bugs" (Chandler, 2005, p. 203). Regardless of the type of testing, the QA team will either use a tool specialised in reporting bugs or an Excel sheet (or any other file) to inform of the bugs, keep track of their status, and verify that they have been fixed. Image 13 is an example of a bug report from a version modified *ad hoc* of the game *Fortune Winds: Ancient Trader* (2012) (Muñoz Sánchez, 2017, p. 173). The different elements that appear in the said report include the identification number of the bug (usually generated automatically), the date of the report, the build where the bug was found, the tester who reported the bug, the language affected, whether it is a system (functional) bug or not and, in some cases, who is in charge of fixing the bug. Other important aspects are the priority or severity of the bug, the area or level of the game where it can be found, and a detailed description of the bug that includes the steps to reproduce it, a possible fix, and a screenshot or a video (Muñoz Sánchez, 2017, p. 166-168). According to Muñoz Sánchez, the different types of bug statuses are (2017, p. 170):

- **Open or accepted:** The report has been made and the person in charge of fixing the issue has been informed.
- **Duplicated:** Another tester has already reported the issue. This can be avoided by searching in the database first to avoid redundancy.
- **Obsolete:** The string affected has been modified or simply deleted from the game.
- **No bug (NB):** A status that should be avoided as much as possible; it means that the person responsible for fixing it does not consider that there is a bug. Frequently related to the localiser’s style or what the tester considers to be a misspelling.
- **Working as intended (WAI):** It is a variation of NB and has coined the expression “It’s not a bug, it’s a feature” (Muñoz Sánchez, 2017, p. 170).
- **Will not fix (WNF):** Some bugs might not be fixed even though it is technically possible due to time, monetary, or difficulty reasons.
- **Will fix later (WFL) or Fix later (FL):** Due to time constraints or the team’s priorities, the issue will be addressed later on.
- **Fixed:** The issue has been solved.
- **Pending verification or to verify:** The tester needs to verify that the issue has been solved.
- **Not fixed:** Due to various reasons, the bug has not been fixed and it will be sent back to the person in question.

Bug ID: 31438	**Date:** 20/02/2016
Status: Accepted	**Build #:** 0.99RC1
Reported by: pmunoz	**Languages:** ES
System *bug*?: No	**Priority:** P2
Category: Term inconsistency	**Area:** Missions menu
Description:	

File: mis_menu.xls
ID: PortAngelo

When the player arrives at Port Nova, he/she can select several options. If "MISIÓN" (MISSION) is selected, the following message appears:

"Un desconocido misterioso quiere
viajar a Puerto Ángelo."

However, according to the glossary, the correct term should be "Puerto Angelo", without an accent in the "A". In fact, this appears correctly in other instances of the game. Therefore, the string PortAngelo in mis_menu.xls should be:

"Un desconocido misterioso quiere
viajar a Puerto Angelo."

Steps to reproduce:
1. After completing the tutorial mission (you can skip it), fight Melchior.
2. After Melchior is defeated, go to Port Nova.
3. Select "MISIÓN" and the corresponding message will appear.

Screenshot:

Image 13. Example of a bug report for *Fortune Winds: Ancient Trader* (Muñoz Sánchez, 2017, p. 173).

Finally, the last element in Image 10 is the bug category. Some categories of linguistic bugs might be closely related to functional or system issues and will require minor code changes. For example, a character that cannot be displayed properly results from a problem with the code and, in some cases, those bugs might not get fixed since "even minor code changes to fix a problem with the display of special characters can cause incompatibility between the [localised] versions" (Chandler, 2005, p. 213). Although each company will have different denominations that must be used to facilitate communication within the team, the nature of the bugs themselves tends to remain the same. According to Muñoz Sánchez, the different categories are (2017, p. 154-166):

- **Font issues:** Problems displaying special characters, differences in the size of the characters, etc.
- **Wrong text implementations:** This bug appears when the developer implemented a string in another language.
- **Missing localisations/Unlocalized string:** When a particular string has not been localised.
- **Typo/Misspellings**
- **Grammar issues**
- **Mistranslations:** Some mistranslations might be more difficult to detect than others and may arise from the localiser's lack of a visual environment.
- **Overflow/Text overflows:** Or text overlaps may happen due to the lack of instructions, lack of space, or human error.
- **Truncation/Truncated texts:** Occur due to issues similar to the previous category.
- **Term/Terminology issues:** Either due to the lack of glossaries or human error.
- **Inconsistencies:** often included in the previous type and creating a category called "term/terminological inconsistency".
- **Instructions/Guidelines issues:** Such as not following the developer's instructions or when the system messages do not follow the manufacturer's guidelines.
- **Style/Readability issues:** Readability issues or confusing instructions are extremely subjective and should only be reported if it is strictly necessary.
- **Subtitle issues:** Either the subtitles do not appear, they present synchronisation problems, they are not displayed long enough, have segmentation issues, or do not correspond with the audio.
- **Audio issues:** Synchronisation problems, when the audio does not correspond with the context, or when it appears in the original language.
- **Cultural issues:** The use of images or gestures that might be considered offensive in the target culture.

CHAPTER 2. VIDEO GAME LOCALISATION

The present chapter provides an overview of video game localisation from an academic, historical and practical point of view in order to better contextualise the field and its practices. As such, it aims at introducing the characteristics of this practice to better understand the importance of the questions included in the surveys and interpret the results presented in the remainder of this thesis. The chapter will be divided into two sections—localisation and video game localisation—to present a holistic and situated view of the sector, the state of the art, and its evolution throughout time. Firstly, we will review some of the attempts that have been made by scholars and the industry to define localisation and localisation studies. Then, we will study the genesis of localisation from a historical point of view presenting the technological advances that contributed to its birth and rise. Thus, we will present the evolution of localisation practices and how the working methods were adapted in order to overcome the obstacles resulting from such a complex field. Furthermore, as part of the goal of this thesis is to focus on the degree of adoption of digital tools involved in the localisation process as well as to study video game localisers' needs and attitudes towards them, we will briefly review the origins of those tools and previous findings about their degree of adoption. Finally, we will discuss the characteristics and evolution of the different localisation types that appeared following technological developments.

The second part of the chapter focuses on the origins of video game localisation academic research and the topics that have been covered. We will mostly concentrate on the practitioners and the scholars who built the field's foundations and the increasing momentum it has gained over the past five years. Afterwards, we will move on to the video game localisation industry and present its origins and evolution as well as its characteristics, textual types, the specific constraints of the field, and translation approaches. We will also analyse the assets that are usually handed down for localisation and usual formats in order to create a base for later comparison with the surveys' results. The last subsection of this chapter starts by briefly reviewing the main actors that take part in the localisation process and their impact. Afterwards, we will describe the video game localisation process depending on the release model used in order to introduce the basic differences between them.

2.1 Localisation

This section presents the origins of localisation as a practice as well as the historical changes that took place in the industry and how localisation evolved to adapt to them. In order to understand video game localisation, it is necessary to study the context in which it appeared as well as the attempts that have been made in order to create an academic framework for this practice. As we will see in the first sub-section, many scholars and practitioners consider the use of technology as an inherent part of the process. Therefore, we will also cover the appearance of different electronic tools specifically designed to help translators improve their performance and cover their evolution and the development of integrated translation environments. Finally, we will concentrate on the main characteristics of other localisation types individually in order to analyse the similarities and differences between each other to provide a basis for later comparison with video game localisation.

2.1.1 Defining localisation: industry versus academia

Holmes' contribution to the field of Translation Studies and the classification he devised to outline the discipline created a solid foundation that still stands despite "multiple revisions, criticisms and additions" (Jiménez-Crespo, 2013, p. 136). In his paper, (Holmes, 1988) he named and described Translation Studies (TS) as empirical in its most basic nature and divided the field into two branches: Pure TS and Applied TS. However, not long after the appearance of TS, the global trend towards digitalisation and the driving force of the localisation industry pushed "the boundaries of pre-digital understandings of what is meant by translation" (Jiménez-Crespo, 2019, p. 26) thus commencing the contentious topic of the definition of localisation. This issue remains open to debate nowadays as "definitions of localization tend to be contextually bound, reflecting the perspectives of those who formulate them" (Folaron, 2006, p. 197). It is beyond the scope of this thesis to offer anything beyond a brief introduction and we will simply present some definitions both from academic and industrial sources chronologically. First and foremost, the term localisation stems from "locale" which, even though in its origins solely referred to "a place, a locality, a scene,"[18], became an industrial notion described in the ISO 17100:2015(en)[19] as a “set of characteristics, information, or conventions specific to the linguistic, cultural, technical, and geographical conventions of a target audience”. Therefore, due to the very origin of the denomination itself, the definition of localisation is heavily influenced by the industry in general. Throughout the discipline’s history, one of the most common methods used has been defining the concept of localisation by directly comparing it with translation. For example, as a practitioner, Esselink (2000, p. 4) explained that:

> Traditionally, translation is only one of the activities in a project where material is transferred from one language to another. Other activities in traditional translation projects include terminology research, editing, proofreading, and page layout. In localization, many more activities have been added to this list. Examples of activities in localization which are not necessarily part of traditional translation include multilingual project management, software and online help engineering and testing, conversion of translated documentation to other formats, translation memory alignment and management, multilingual product support, and translation strategy consulting.

Esselink would later add that (2003, p. 4) “in a nutshell, localization revolves around combining language and technology to produce a product that can cross cultural and language barriers. No more, no less”. Localisation is described by Pym (2004, p. 127) as a concept that “generally refers to the processes by which a generic (“international”) product is adapted to the requirements of a “locale”, a place with a specific union of cultural and linguistic features.” Dunne provides a complete definition of localisation in the framework of Translation Studies that encompasses the full process and many of the previous definitions (2006, p. 4):

> The processes by which digital content and products developed in one locale (defined in terms of geographical area, language and culture) are adapted for sale and use in another locale. Localization involves:

[18] https://www.etymonline.com/word/locale
[19] https://www.iso.org/obp/ui/#iso:std:iso:17100:ed-1:v1:en

> (a) translation of textual content into the language and textual conventions of the target locale; and (b) adaptation of nontextual content (from colors, icons and bitmaps, to packaging, form factors, etc.) as well as input, output and delivery mechanisms to take into account the cultural, technical and regulatory requirements of that locale. In sum, localization is not so much about specific tasks as much as it is about the processes by which products are adapted. Moreover, localization is but one of a number of interdependent processes and cannot be fully (or correctly) understood without being contextualized in reference to them. These processes are referred to collectively by the acronym GILT (Globalization, Internationalization, Localization, Translation).

When we analyse the definitions provided by the industry, we can observe that localisation is mostly described as a process comprising multiple steps that include adapting the product, management tasks and technical steps such as character encoding, which have turned localisation as a service the preserve of specialists. The common denominators are the fact that the professional works with products instead of texts, the division between linguistic matters and cultural matters, the use of the nomenclature “locale” instead of “language”, and the fact that they tend to avoid the term “translation” as much as possible (Jiménez-Crespo, 2013, p. 13). The now-defunct Localization Industry Standard Association (LISA) provided several definitions that changed over time to include the new aspects brought by new technological developments and the addition of new services. In 2003 the LISA described localisation as a practice that “involves taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold” (LISA, 2003, p. 13). Later, it added (LISA, 2007, p. 11):

> Localization involves the adaptation of any aspect of a product or service that is needed for a product to be sold or used in another market. [...] While there is overlap between translation and localization, localization generally addresses significant, non-textual components of products or services in addition to strict translation.

The current definition that can be found nowadays on the Globalization and Localization Association (GALA Global) website, which succeeded LISA, also highlights the fact that localisation is a process that consists of "adapting a product or service to a specific locale"[20] and that translation is merely one of the steps included in a process that also encompasses:

> Adapting design and layout to properly display translated text in the language of the locale. Adapting sorting functions to the alphabetical order of a specific locale. Changing formats for date and time, addresses, numbers, currencies, etc. for specific target locales. Adapting graphics to suit the expectations and tastes of a target locale. Modifying content to suit the tastes and consumption habits of a target locale. The aim of localization is to give a product or service the look and feel of having been created specifically for a target market, no matter their language, cultural preferences, or location.[21]

In response to the definitions that appeared following the industry's developments indicating "that the industry consolidated without relying on the body of knowledge of TS" (Jiménez-Crespo, 2013, p. 14), Jiménez-Crespo tried to refine GALA's effort while highlighting the relationship with academic TS and emphasising the concept of interactivity, which written translation does not take into account. Therefore, the author defines localisation as "a complex, technological, textual, communicative, and cognitive process by which source interactive digital texts (i.e. software, websites, and video games) undergo modifications so that they may be used in linguistic and sociocultural contexts other than those of production." (Jiménez-Crespo, 2019, p. 26). Conversely, other scholars in Translation Studies criticise that the industry seems to consider that the inclusion of a cultural dimension that needs to be adapted is a new addition to the field of translation and argue that "in the translation studies community, this is simply a commonly accepted definition of translation itself" (Hartley, 2009, p. 107).

[20] https://www.gala-global.org/knowledge-center/about-the-industry/language-services

[21] https://www.gala-global.org/knowledge-center/about-the-industry/language-services

However, scholars such as Odacıoğlu highlight a profound shift that extends to the source text itself due to the fact that localisation cannot be performed without translation and localisation tools and specify that "the role of [the] source text is replaced by internationalization, and as a rule, vertical translation method has begun to be adopted, in the same way as linguistic translation approaches" (Odacıoğlu, 2017, p. 26). The same author also adds that in localisation, the localiser is part of a team and that those members (Odacıoğlu, 2017, p. 26):

> [H]ave become integrated to the [localization] process, and although indirectly, to the translation practice and localization project by using various technological tools (see computer programming, computer engineering, graphic design tools, translation management systems, project management systems, etc.). In addition to these, the translator may also assume different roles (e.g. project management, post-editing, marketing consultancy, localization engineering, language engineering, etc.) in the localization process, and use his 'expertise in translation practice' in integration with different fields. Thus, translators may also be termed localizer or localization expert. This gives the impression that the role of translators in the localization team is not only translation.

On the whole, the definition of localisation remains a controversial topic[22] in Translation Studies due to the gap between the industry and Translation Studies mostly caused by the former's rapid development and its multidisciplinary nature. Therefore, those definitions depend on "the differentiating criteria chosen – adaptation, management, culture –" (Jiménez-Crespo, 2013, p. 23) and each individual's affinities and background. Following Odacıoğlu's example, we will refer to professional translators working in video game localisation as "localisers" (using the British English spelling) due to this paper's focus on the video game industry and, more specifically, on the use of technology as part of a process that goes from video game development to linguistic quality assurance including asset extraction and integration, management, translation technologies, etc.

---

[22] Resulting as well from the fact that localisation is the object of study of other fields outside of TS.

On the whole, the concept of "Localization Studies" or LS (Munday, 2012) gained notoriety and scholars such as Jiménez-Crespo described LS "as a sub-branch of general TS interfacing with and feeding off all the three branches of research but also incorporating connections with a number of new disciplines not previously connected to TS, such as information management or international business strategies" (Jiménez-Crespo, 2013, p. 139). The author divides the discipline into "Pure and Applied LS" (Jiménez-Crespo, 2013, p. 139) with both theoretical and descriptive branches and explains that Theoretical LS "would cover general theories of localization […] as well as partial theories restricted to a localization type (software, web localization, videogame localization, small device and app localization), rank, problem, text type or genre, etc." (Jiménez-Crespo, 2013, p. 139). Conversely, "[d]escriptive LS could be divided between product-based, function-based and process-oriented" (Jiménez-Crespo, 2013, p. 139). Figure 12 is Jiménez-Crespo's adaptation of LS following Holmes' classification (Holmes, 1988) and Toury's map (Toury, 1995).

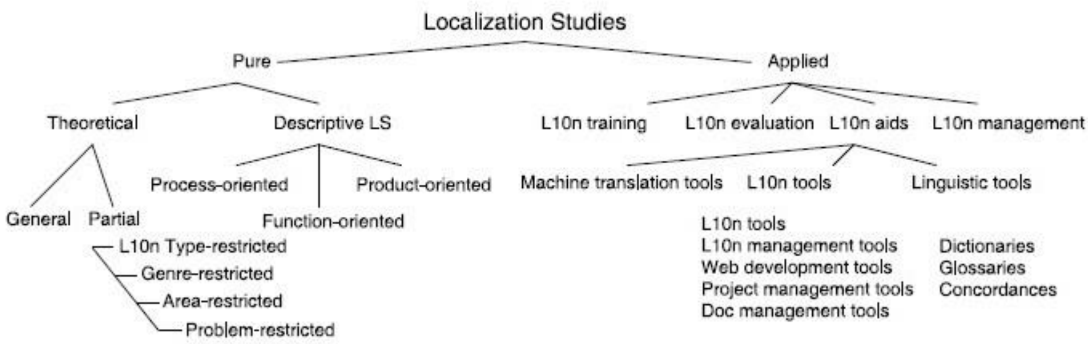


Figure 12. Map of Localization Studies (Jiménez-Crespo, 2013, p. 141)

Nowadays the controversial definition of LS and its place within TS gains notoriety as the thin differentiating line between them is becoming even thinner due to the expansion of the use of the term "localisation" to non-digital texts as "localisation has brought to the forefront of TS and TS theorisations a number of specific models, constructs, and perspectives. These have been subsequently adopted to extend the notion of localisation to non-digital texts such as comics, news, and advertising." (Jiménez-Crespo, 2019, p. 41). This tendency towards erasing the borders, pushing the limits and converging models and notions is bound to continue as technology keeps evolving in the future, eroding the frontiers even more.

2.1.2 Localisation throughout history

Localisation as a practice did not become a reality until the democratisation of personal computers and the commercialisation of software as an international product, thus crossing the frontier of English-speaking countries and targeting a non-English-speaking clientele. Even though the first computers appeared in the '40s and '50s, as explained in Chapter 1, they were mostly used for military purposes or in universities (to further research and development) due to the sheer size of the equipment as well as the price of the components. It was not until the mid-seventies that the advancements related to microprocessors allowed to reduce their size and computers started to become more common and evolved into a tool also used in big companies. Although the first microprocessor to be commercially produced, the Intel 4004, was released in 1971, personal computers did not enter the market until 1974 and, as Dunne explains (2014, p. 147-148), before the 1980s the word "computer" was largely understood and used as another way to refer to mainframes. Mainframes were the predecessors of personal computers and, although considerably smaller than the first computers, they were still mostly used by large organisations due to their elevated price and how complex they were, being handled only by programmers or developers.

However, the '80s brought new advances to microprocessors which culminated in the democratisation of personal computers, also favoured by the reduction of production costs for manufacturing computer hardware. Thus, "[a]s computers, especially personal computers, became more and more common, the typical users were no longer professional computer programmers, software engineers, or hardware engineers" (Uren *et al*., 1993, p. ix). This particular shift towards non-computer professionals was rapidly taken advantage of by American companies that realised that there was a new niche for their products. Therefore, they started manufacturing software more suited for non-specialists which entailed the need to create "user-friendly" programmes that could be produced for a mass market with products that were adapted to be easier to use and that contained fewer functional bugs to troubleshoot. As Uren *et al*. (1993, p. x) explained:

> To this end, documentation and on-line help were both improved, along with the software itself. While experienced professionals had become adept in detecting bugs and working around them, the new

users expected, indeed demanded, that the software they bought operate exactly as described in the manuals.

Therefore, US-based computing companies started to concentrate their efforts on manufacturing programmes for desktop computers such as word processors focused on improving the efficiency of workers as well as their productivity. Although these businesses targeted their domestic market first, once they had successfully made their products popular at home, they soon realised the potential increment in the number of clients if they started to propose their products in other languages which "automatically triggered the need to localize the products for international markets" (Esselink, 2003, p. 4). And thus, localisation as a practice began driven by economic reasons as well as by the growing number of PC programs, developed mostly by Microsoft, that reached the international market and crossed the borders of English-speaking countries.

As Jiménez-Crespo explains (2013, p. 8), "the initial targets were Japan and the so-called FIGS countries (France, Italy, Germany and Spain)" and, in order to establish themselves in non-English-speaking countries, manufacturers promptly understood that "it was essential to convert the software so that users saw a product in their own language and firmly based in their own culture" (Uren *et al*., 1993, p. x). Even though initially companies thought that they could improve their return on investment (ROI) by simply translating the aforementioned manuals, help documentation and the text that was displayed on the programmes' user interfaces, they swiftly grasped that the process was much more complicated and had to "include all target market requirements for culturally dependent representation of data" (Dunne, 2014, p. 148). These requirements went from time and date formats to "character sets for the digital representation of writing systems, encodings to enable the storage and retrieval of data in languages other than English, collation rules, [...] as well as calendars and decimal separators (period or comma)" (Dunne, 2014, p. 148). The manufacturers adopted various strategies in terms of localisation at the beginning, and some software and hardware developers decided to deal with the process by creating an in-house specialised department in their own company that would take care of the translation and support their international endeavours. Other companies, however, would simply deal with the matter by sending the finished product to the office branch they had in the country in question and letting them take care of the issue. Whichever strategy they used, there was always a division between development and localisation and, as Esselink explains (2003, p. 4):

> This separation of development and localization proved troublesome in many respects. Microsoft, for example, asked its then-distributor ASCII in Japan to localize Multiplan (predecessor of Excel) into Japanese. According to a Microsoft director responsible for localization at that time, "we'd finish the product, ship it in the United States, and then turn over the source code library to the folks in Japan, wish them luck and go on vacation".

The practice of finalising the English version of the programme first and then localising it into different languages afterwards, as described by Esselink (2003, p. 4), proved to be extremely problematic. The fact that the original piece of software was developed in English, following only the requirements for that precise language and that it was not prepared in advance to allow for a certain number of necessary modifications was the main reason. As Dunne explains (2014, p. 149) the English version "lacked certain critical capabilities, such as the ability to display the necessary target-language scripts and writing systems." Therefore, once the localisation teams or the office branches in charge of performing the localisation had identified all the translatable text among the code, which was a daunting endeavour to begin with, then they would have to ask the development team to perform modifications *post hoc* to the software in order to implement the necessary changes to display the text. Moreover, due to the need for specific source code modifications linked to the different requirements of each individual language, instead of dealing with a single version of the programme, companies were forced to deal with as many versions as languages used. Furthermore (Dunne, 2014, p. 150):

> This multiplier effect was compounded by the "localization-as-afterthought" approach whereby localization bugs were not discovered until after the development of the source products had been completed. Further complicating the localization process was the fundamental problem of the duplication of effort required to manage

multiple sets of source code in parallel: changes to one version of source code would potentially need to be made in all other versions.

Additionally, after the localisation process, each version had to be compiled and then tested individually. Whenever bugs were found, those would need to be fixed and the version would need to be updated subsequently (and potentially re-tested to ensure that the bugs had been dealt with). This approach proved to be extremely expensive and time-consuming and the task was only worsened by the complexity in terms of engineering and the wide array of texts and formats. Austermühl (2006, p. 80) provides a comprehensive overview of the different types of texts present in a localisation project in order to better illustrate the scope of the task and its complexity. Figure 13 shows that he distinguished seven different categories: software UI, development documents, reviews, secondary manuals, training material, marketing text, and documentation (although nowadays printed material has become rare).

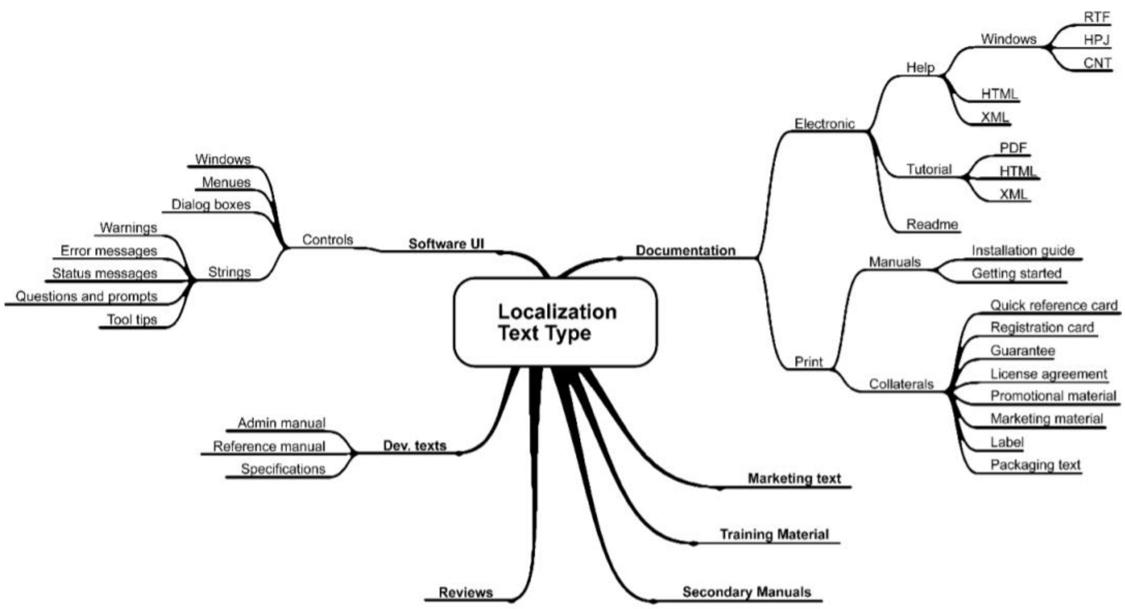


Figure 13. Localisation text types (Austermühl, 2006, p. 80)

As it can be easily deduced, finding all the translatable strings among the programming code in such a wide array of documents was, as mentioned above, an extremely time-consuming task and, without proper standards, there was little space for code reusability. The word "string" refers to a suite of characters that is recognised as a complete unit and goes from one word to several sentences, they are characteristic of localisation and compose the source text. Among the most common issues they encountered when faced with localisation, we can mention space constraints related to the length of the translated strings or the presence of code "(the so called 'hard-coded strings') that could not be translated when target locales required specific number, gender or declension agreements; dealing with these types of textual strings required a basic understanding of programming, etc." (Jiménez-Crespo, 2013, p. 9). Consequently, companies soon realised that there were "upstream measures that could be taken to facilitate the process" (Dunne, 2006, p. 5) to reduce the costs of localisation and to lessen the complexity it involved. Thus, internationalisation was created, according to Esselink (2003, p. 4):

> Internationalization refers to the adaptation of products to support or enable localization for international markets. Key features of internationalization have always been the support of international natural language character sets, separation of locale-specific features such as translatable strings from the software code base and the addition of functionality or features specific to foreign markets.

In other words, the effective implementation of internationalisation "eliminated the need to maintain a separate set of source code for each supported locale" (Dunne, 2014, p. 152) and allowed companies to simply replace the set of resources that contained the source language with those that had the localised version by the means of specialised editors. During these decades, a new term was introduced: GILT, which is a combination of the processes of globalisation, internationalisation, localisation, and translation. These concepts were defined as follows by LISA:

- **Globalisation (G11N): "**Globalization addresses the business issues associated with taking a product global. In the globalization of high-tech products this involves integrating localization throughout a company, after proper internationalization and product design, as well as marketing, sales, and support in the world market." (Esselink, 2000, p. 4)

- **Internationalisation (I18N):** "Internationalization is the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for re-design. Internationalization takes place at the level of program design and document development.” (Esselink, 2000, p. 2)

- **Localisation (L10N):** “Localization involves taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold.” (Esselink, 2000, p. 3)

- **Translation (T9N):** “Translation is only one of the activities in localization; in addition to translation, a localization project includes many other tasks such as project management, software engineering, and desktop publishing.” (Esselink, 2000, p. 4)

The GILT cycle resulted from the awareness that separating development and localisation causes undesirable constraints due to the interrelation of these processes. Another historical adjustment that became a particularity of the field of localisation and added importance to the implementation of GILT practices while highlighting the necessity of planning localisation at the beginning of the project was the shift in the industry towards simultaneous shipment or sim-ship. Contrary to traditional translation where the source text tends to be final before the translation process starts, localisation normally occurs when the product is still under development.

This particular release model allows to commercialise a fully multilingual product and reduces potential production and marketing costs. However, the practice adds another layer of complexity to an already intricate procedure and relies on constant communication and collaboration between each link of the chain. Therefore, “[t]he GILT process represents a collaborative endeavour and is considered an interactive bottom-up and top-down process” (Jiménez-Crespo, 2013, p. 26). In other words, even though globalisation and internationalisation must take place before localisation and translation in order to prepare de product in advance (Figure 14), localisers will inform the team about the potential issues they encounter to create solutions (Jiménez-Crespo, 2013, p. 26).

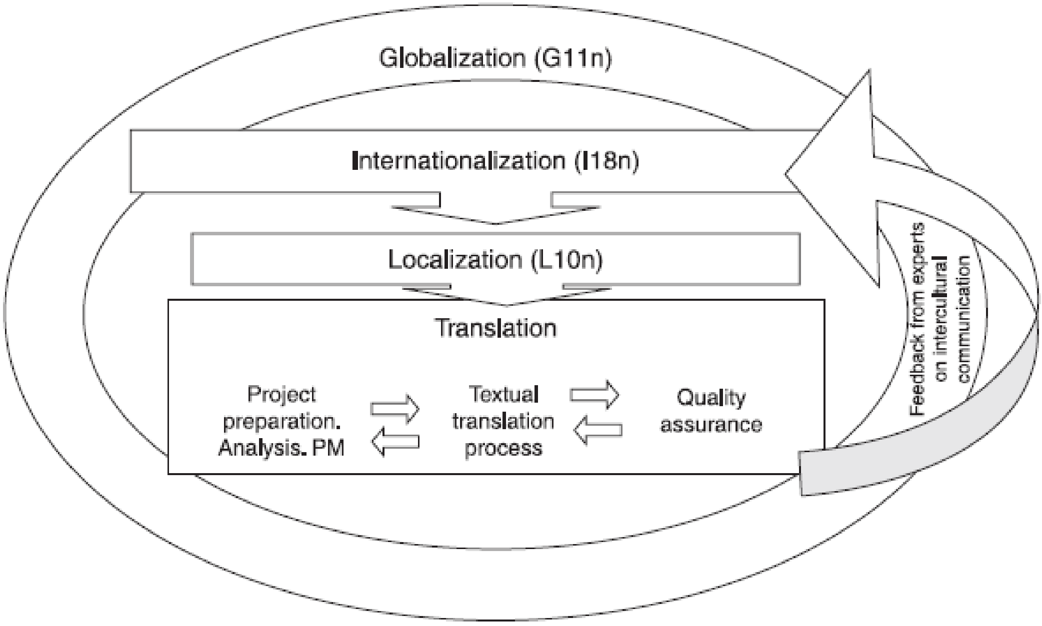


Figure 14. GILT cycle and its dependencies (Jiménez-Crespo, 2013, p. 27)

During the ‘80s, several prominent US companies decided to use Ireland as a base of operations when it came to distributing their products and the country became the main hub for localisation in the ‘90s. This move was driven by the fact that, once the country’s efforts to create a cheap manufacturing sector failed, the “Irish government switched its focus to research and development and the high-tech, blue-chip companies, that is, a more long-term strategy” (Esselink, 2003, p. 4-5). Consequently, the majority of companies specialised in software and the web established a base “in Ireland, with the bulk of their localization being managed from there, including Microsoft, Oracle, Lotus Development, Visio International, Sun Microsystems, Siebel and FileNET” (Esselink, 2003, p. 5). Having their European headquarters in Ireland supposed the reduction of labour costs (compared to the US) as well as VAT exemptions for sales in Europe and thus, the country became a very desirable place to be for localisation experts (Esselink, 2003, p. 5).

Eventually, as the complexity of computers (and therefore software) increased and the volume of work augmented, a separate and independent industry focused solely on providing localisation services appeared. Consequently, there was a change in the market and some publishers decided to shift towards an outsourcing model as they lacked the means to manage multilingual translation processes on their own (Esselink, 2000). This resulted in the apparition of multi-language vendors (MLVs) such as INK or IDOC (both now divisions of Lionbridge) or specific departments in already existing companies that became specialised in software localisation and all the components involved (Esselink, 2000, p. 5-6). However, testing was still carried out by software publishers, which resulted in hold-ups as they had to test multiple versions in different languages and, thus, MLVs extended their services to include "engineering, testing, desktop publishing, printing, and support services" (Esselink, 2000, p. 6). Proof of the establishment of the discipline was the founding in 1990 of the now-defunct Localisation Industry Standards Association which described itself as:

> [An organisation that r]ather than acting as a traditional standards body per se, members view themselves as a results oriented implementation group focusing on improving the bottom line of their company's localization business. Meeting four times each year, The LISA Forum motivates localization professionals to discuss non-proprietary information about product and translation costs, QA testing and product internationalization procedures among other key business issues.[23]

As Esselink (2000, p. 6) explains, the industry consolidated during the second half of the '90s and there were important mergers that resulted in about a dozen of MLVs across the world. These vendors tended to have programmes for internal training since, at the time, it was not common to find linguists with computer skills and one of the main characteristics of software localisation is the number of different components that are localised and the technical complexity that it entails. This system evolved and those big MLVs "took on multilanguage, multiservice projects, outsourcing the core translation services to single-language vendors (SLVs), one in each target country" (Esselink, 2003, p. 6). These first decades and the birth and boom of the industry resulted in the apparition of translation and localisation tools to reduce costs as the increasing number of companies providing localisation services resulted in ferocious competition. We can also observe that during the '90s "the localization industry further

[23] https://www.w3.org/International/O-LISA-intro.html

professionalized, including industry organizations, conferences, publications, academic interest and generally increased visibility" (Esselink, 2003, p. 6).

2.1.3 Translation tools and integrated translation environments

The appearance of computers also had a great impact on the tools used and translators went from typewriters and printed dictionaries to translation memories, machine translation, terminology management tools, localisation tools, etc. The idea of machine translation dates back to the 17th century (Hutchins, 2005) although it did not become a reality until the 1950s "following a demonstration in 1954 by IBM and a research team at Georgetown University. By 1965 American government agencies are estimated to have spent some 20 million dollars in supporting machine translation research at 17 different institutions" (Arthern, 1979, p. 77). Throughout the following decades, the enthusiasm and interest of institutions fluctuated greatly mainly due to the report released in 1966 by the Automated Language Processing Advisory Committee (ALPAC) which argued that machine translation increased the costs, slowed the process and was far less accurate than human translation. Therefore, the field went from a period of high expectations during the '50s and '60s to a quieter period in the '70s and the release of the first commercial systems in the '80s (Hutchins, 2005). However (Zaretskaya, 2017, p. 16):

> At the same time, as research in MT was discouraged, there was a shift in research direction, and it was proposed to focus instead on the development of computer programs that would assist translators. Thus, the ALPAC report includes a description of a system for 'automatic dictionary look-up with context' (ALPAC 1966, 34) […]. This system was intended for terminological research and included tasks such as text alignment and term retrieval, which are still present in today's tools.

As Christensen and Schjoldager (2010) explain, Arthern (1978) can be considered one of the first to conceive the idea of a translation memory system that reuses already translated texts. In his paper, the author speculates about the impact of Machine Translation on the future of translators (translation quality and the need for pre-editing and post-editing), describes Systran and analyses the quality of its output, takes a look into terminology data banks, etc. The latter started being created during the ‘70s and included both definitions and translations as well as concordance searches (Hutchins, 1998). Most importantly, in his paper, Arthern proposes creating a system that processes texts with enough available memory to store both source texts and target texts in different languages; “in such a way that any given portion of text in any of the languages involved can be located immediately, simply from the configuration of the words, without any intermediate coding, together with its translation into any or all of the other languages” (Arthern, 1978, p. 94-95). Two years later, Martin Kay (1980) suggested a system that would include “multilingual word processor, dictionary look up, and a possibility to consult previous translations. It also included an automatic translation component, which would work under translator’s control” (Zaretskaya, 2017, p. 17).

However, commercially available TM tools did not appear until the development of the appropriate technology. This development was facilitated by the appearance of the Automated Language Processing System (ALPS) which allowed users to extract repetitions and display both the source and target texts, as well as the invention of bi-text, tools used for aligning texts, and the creation of bilingual databases (Christensen and Schjoldager, 2010, p. 90). “Translation technology INK was also one of the first companies to create desktop translation support tools, called the INK TextTools, the first technology commercially developed to support translators” (Esselink, 2003, p. 5). The tool provided terminology management, included the possibility to create bilingual dictionaries by analysing the source text to extract terms, and provided sample dictionaries (Lewis, 1991, p. 35). Shortly after, translation memory technology entered the market (Esselink, 2003, p. 5):

> In 1987, a German translation company called TRADOS was reselling the INK TextTools and a year later released TED, the Translation Editor plug-in for TextTools. Shortly thereafter, TRADOS released the first version of its Translator’s Workbench translation memory (TM) product. TRADOS continued to establish itself as the industry leader in TM technology throughout the 1990s, boosted by Microsoft taking a 20% stake in 1997.

Nonetheless, early translation memory tools could not support other formats besides text files and most software manufacturers would develop in-house tools used to deal with the localisation of user interfaces, thus creating their own localisation tools (Esselink, 2003, p. 5). These tools were specifically designed to suit each company's standards and source code, were developed *ad hoc*, would crash often, and contained multiple bugs (ibid). Eventually, commercial localisation tools such as SDL Passolo or Alchemy Catalyst appeared in the market. Their main benefit was the possibility to grant localisers a WYSIWYG (What You See Is What You Get) environment which allowed them to see changes "on the fly" (in real-time). Furthermore, the tool could "facilitate the translation of natural languages strings (words, phrases and sentences) embedded in artificial languages programming commands (HTML, Java, C++, Linux, etc.) for software products by leaving game code out-of-bounds" (Bernal-Merino, 2013, p. 118). Therefore, localisers could directly access the assets that required localisation including menus and buttons and a window would display the final appearance of the product. However, it should be noted that (Bernal-Merino, 2013, p. 119):

> Video games share part of the above-mentioned menu mechanics to control the game but the translation part of the process can rarely benefit from visual localisation tools such as those available for utility software and websites for the mere reason that the latter are designed to a different coding and file formatting standard (such as '.NET' or '.XML') and do not prioritise rich multimedia worlds nor entertainment like games do; in other words, they are completely incompatible.

TM technology has been gaining terrain consistently among translators since its origins until becoming the most widely used translation tool nowadays. When we analyse the results obtained by previous studies about the degree of adoption of these systems, we can observe that in 2005 a study carried out by Fulford and Granell-Zafra showed that only 28% of the 591 UK-based freelancers that took part used CAT tools (Fulford and Granell-Zafra, 2005, p. 10). Only one year later, Lagoudaki found that among the 699 participants that took part in her survey, 82.5% used TM systems (Lagoudaki, 2006, p. 11). Conversely, the results obtained and published by Zaretskaya *et al.* in 2018 revealed a drop and that 76% of the 736 participants used it (Zaretskaya *et al.*, 2018, p.

46). Additionally, these studies showed that translators prefer comprehensive solutions, a fact that can also be observed in the low degree of adoption of standalone corpus compilation tools and terminology extraction and management tools. In their study, Fulford and Granell-Zafra found that only 24% of the respondents used terminology management systems (Fulford and Granell-Zafra, 2005, p. 9) whereas Zaretskaya *et al.* reported 58% for terminology management tools and 25% for terminology extraction tools (Zaretskaya *et al*., 2017, p. 46).

Therefore, most systems have steadily increased the number of features offered and included terminology extraction tools, terminology management, and even corpus compilation tools in order to become comprehensive systems that cover as many of the translators' needs as possible. In the field of MT, the appearance of neural machine translation seems to have renewed the interest in the topic and its degree of adoption has been increasing steadily (see Chapter 5 section 5.4.6) attracting the attention of scholars, practitioners, and the industry. Therefore, MT has also been steadily incorporated into comprehensive solutions thus further developing the features offered by what could now be called integrated translation environments. Nowadays, there are multiple programmes available in the market both downloadable and web-based going from what was called initially "globalization management systems (GMS)" which later became "translation management systems (TMS)", to the increasing popularity of "content management systems (CMS)" in the world of localisation, which also integrate the abovementioned features.

The main benefits of translation memory-based tools in the field of localisation are, on the one hand, the fact that they can protect tags, code and variables, a vital feature in the field as "any accidental tampering could lead to a malfunction in the localized software" (Mangiron Hevia and O'Hagan, 2013, p. 95). On the other hand, they lessened the impact of simultaneous shipment practices by memorising previous translations and recognising ulterior changes to the text. Regarding video game localisation, these changes are mostly due to the constant modifications that occur during the development phase and, in the other types of localisation, due to subsequent updates as "after general release, most software products are updated at least once a year. These updates usually just add features onto a stable base platform, making it all the more important to be able to reuse—or leverage—previously produced content and translations" (Esselink, 2003, p. 6). However (Mangiron Hevia and O'Hagan, 2013, p. 101):

> The restricted role of translators which is commonly assumed in localization may indeed be justified on the basis of the techno-centric workflow adopted in localization. For example, Pym (2004) has noted how the working environment in localization was configured on the basis of CAT tools. Even the very design of earlier TM products seemed to have assumed that the translator did not need to see the wider context beyond the sentence-based segment currently being translated. Such a design forced the translator to work at sentence level rather than according to a more intuitive segmentation at a bigger unit than a sentence. This is a long way from the now more acknowledged vision of translation as taking place on the broader levels of textual units even between cultures. The failure of some CAT tools to factor in many translators' needs has been highlighted in the literature (e.g. Lagoudaki 2008) and indeed has led to various currents of process-oriented research focused on the impact of tools on human translators (Christensen 2011).

2.1.4 Localisation types

As we have seen, new localisation types appeared as a direct result of technological advances such as the democratisation of personal computers, the appearance of digital games, the invention of the World Wide Web, and the creation of smartphones. Each type of localisation was driven by the industry's needs and the field evolved accordingly, moving from including only software (and video games) to web localisation or apps for small devices. As Jiménez-Crespo explains (2013, p. 28):

> During the 2000s the different localization types consolidated into distinctive categories that required specific translation and technical skills from the agents involved, and, although the basic localization types still exist, new emerging modalities are now blending these types and continue to redefine them. For example, cloud computing

> software applications are directly accessed online via the browser – the so-called ‘web-based applications’. This combines traits of web and software localization into a single process. This combination also appears in small device localization.

Although each type shares multiple characteristics such as the digital nature, the use of a screen, or interactivity; there are also significant technical contrasts such as programming languages, the methods used to store and present the strings, textual types and genres. Software, whether for productivity purposes or small devices, tends to follow a “relatively standardized textual genre” (Jiménez-Crespo, 2013, p. 28) whereas video games contain a wider variety of assets and textual types. Another characteristic of video games is the diversity of genres and their immersive nature, an immersion that is also sought after in website localisation which, conversely, has to deal with an infinite number of potential genres.

2.1.4.1 Software localisation

As Figure 15 shows, the assets that require localisation are not limited to the “programming code assembled in an executable file” (Sandrini, 2008, p. 169) and include tutorials, marketing collateral, sample files, printed and online user documentation, and online help (Dunne, 2014, p. 150). The resources that are normally part of the application include accelerators (keyboard shortcuts); dialogue boxes, icons, menus, the toolbar, and string tables (Dunne, 2014, p. 151). These strings “include menu items, command button captions, dialog box titles, tool tips, error messages, status messages, and so forth” (Dunne, 2014, p. 151). Additionally, due to the development of the previously mentioned localisation tools, “[m]enu and dialog box strings can often be visually represented in a WYSIWYG (what you see is what you get) editor during localization, whereas string tables typically cannot” (Dunne, 2014, p. 151). Finally, the compiled section of online help contains hypertext documents in different formats whereas the web-based type is likely to operate following the characteristics of a website.

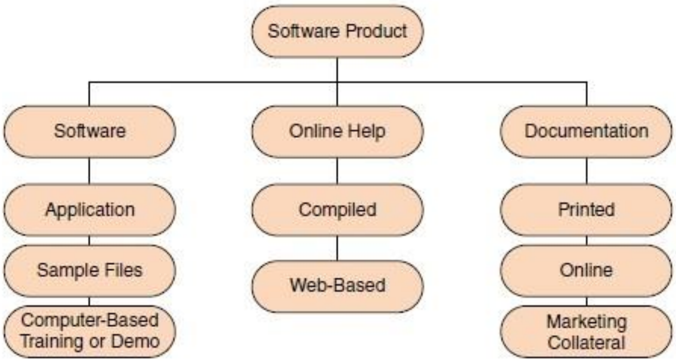


Figure 15. Assets in a traditional software localization project (Dunne, 2014, p. 150 adapted from Esselink, 2000, p. 10)

As Dunne explains (2014, p. 154), the development of object-oriented programming was the milestone that facilitated internationalisation and allowed companies to represent functions in a standardised way. As its name shows, this paradigm is based on objects which are created using blueprints called classes—reusable pieces of code that determine the object's data or properties. In other words, they can become templates that can be later used to create other objects, therefore allowing to build a logical structure and helping to develop new programmes faster while making them easier to modify, maintain and reuse. C++, Python and Java are some examples of object-oriented programming languages and, in a programme, a command button or a menu would be considered an object. Therefore, "internationalization can be thought of as a standardized way of representing, categorizing, and storing all of the objects that comprise the user interface as classes of resources" (Dunne, 2014, p. 154) and localising is just a matter of changing the object's internals. As a result (Dunne, 2014, p. 155-156):

> If a software application has been properly internationalized and no translatable text is embedded in the source code, the scope of localization is confined exclusively to resource files and translators have neither access to nor any ability to alter the functional code of the program. Consequently, the hands-on work of software localization now consists essentially of the translation of strings in menus, dialog boxes, and string tables. Depending on the source and target languages involved, localization may also require that dialog

> boxes and their constituent controls (such as command buttons) be resized to account for translation-related string expansion.

As previously mentioned, there are WYSIWYG localisation tools that allow localisers to work within the visual environment and see the changes "on the fly" to cope with the space constraints that appear mostly in menus or user interfaces. Although, new business practices are moving towards web-based systems and less complicated file formats thus making those tools obsolete. Space constraints are a common issue that arises from language expansion due to the fact that programmes tend to be developed in English and when localising into other languages, the resulting string is longer than the original causing overlaps, truncations, or overflows. However, nowadays many programmes have incorporated responsive design in the internationalisation phase in order to allow menus, for example, to adapt their size to the string and any screen size, something that has become customary due to the wide variety of devices currently used. Normally, localisers receive a localisation kit that contains all the files and documents needed such as "source files, built environment, guidelines, available glossaries and translation notes" (Sandrini, 2008, p. 170). As for glossaries, besides maintaining consistency throughout the programme, there are existing requirements depending on the publisher or the operating system that need to be respected in order to commercialise the product.

"The source files are usually programming code files — e.g. instructions in the programming language C++ — and resource files (with the extension .rc), which include user interface elements such as menus and dialog boxes" (Sandrini, 2008, p. 170). The first criterion in order to choose the necessary tool used to be the file format since not all source files were kept separated from the rest and some were compiled along with the code in binary executable programme files and binary resource files (.dll or .exe). If the .exe and .dll files were standard, then the only way to proceed was to use a localisation tool. Non-standard binary files would have required localising directly within the source code and the format .rc could be handled by either a localisation tool or the native development environment (Sandrini, 2008, p. 171). Nowadays, localisers usually work in Excel or Google Spreadsheets and rarely have access to the programme itself, getting closer to the current system for software application localisation in small devices and video games. They might also receive other formats such as .po or .json files that can be handled by regular computer-assisted translation tools, which is also the case for the rest of the documents, either printed or online, as most of the tools also work with HTML files and can also provide a WYSIWYG environment.

2.1.4.2 Website localisation

The appearance of the World Wide Web in the early ‘90s resulted in the boom of e-commerce, and website localisation did not take long to follow. As Dunne explains (2006, p. 3) the “spectacular explosion of e-commerce, especially B2B e-commerce, has in turn fueled the proliferation of an ever-expanding variety of “content” to be localized in an ever-increasing number of formats for an increasingly diverse set of users and locales.” Sandrini (2008, p. 179) lists three main differences between software localisation and website localisation: the frequency of content updates, the fact that this dynamism results in a continuous relationship between the localiser and the customer, and the need for a higher degree of expertise due to the wide range of text types and strategies since a single website might contain “marketing texts, product descriptions, legal information, manuals, listings, etc.” (ibid). The content (constituted of assets) can be static or dynamic and can be categorised as:

1. Textual elements: such as the text content *per se*, the name of the website, all the elements of the menu, hyperlinks, downloadable files, AdWords, text embedded in images, and keywords used for SEO (Search Engine Optimisation).
2. Multimedia: audio files, video files, or GIFs.
3. “Community assets: dynamic content of discussion forums and chat rooms” (Sandrini, 2008, p. 179).

Among the assets that need to be localised, we can find multiple file formats such as HTM/HTML, XML, XSL, JS, ASP, PHP, JSP, or XLIFF (created specifically for localisation tools based on XML) that can be handled directly by computer-assisted translation tools in order to protect tags and to avoid accidental modifications. The process of website localisation, similarly to software localisation, requires a number of technical steps which start by making sure that the website has been properly internationalised. Other tasks that have to be performed before localising (once more similarly to software) include setting up the environment and fetching the contents, identifying third-party components, deciding the level of localisation and the assets that must be localised, identifying what needs to be adapted depending on the markets targeted, the creation of the localisation kit, and creating the appropriate structure to hold the localised versions (Jiménez-Crespo, 2013, p. 30). The pertinent localisation level is usually decided depending on the company's expected ROI in order to be as cost-efficient as possible. Singh and Pereira identified 5 different levels (Singh and Pereira, 2005; cited in Jiménez-Crespo, 2013, p. 34-35):

> 1. Standardized websites: in which a multinational company simply offers a site in one language for all countries/markets.
> 2. Semi-localized websites: in which the only locale/specific content is a contact page in the target language with information about local branches, contacts, etc.
> 3. Localized websites: in which most content and pages are localized, but the original functionalities and back end are not modified.
> 4. Extensively localized websites: in which there is a global localization and all content and site structure/functionalities are fully adapted to the target locale.
> 5. Culturally adapted websites. This is the most advanced level of localization, the one that the authors advocate, and in which there is a total immersion in the target locale.

Therefore, besides the typical cultural adaptations necessary in software localisation (icons, date and time formats, currencies, addresses, measurement units, etc.), localisers must pay special attention to other aspects such as symbols and colours or even the images used. In terms of terminology consistency, besides the browser-related terms, it is essential to comply with the guidelines of the client and the corporation's image as well as with the website's topic and the terms used by third parties. Another aspect that needs to be taken into account during the authoring phase is the source texts, as Sandrini explains (2008, p. 182):

> Texts have to be also adapted in order to be suitable for new cultures or to be at least culturally neutral. In comparison to software localization, website localization involves a slight shift of priorities.

> Software localization concentrates on functionality – the application must work in the target language – and language quality concerns are less critical to a certain extent. For websites, however, language quality is crucial as it is the medium to convey a specific content to the target audience.

Finally, new technological developments such as the impact of the use of small devices and the subsequent reduction of the size of the screen must be considered during the internationalisation phase of the development process. The wide variety of different screen sizes and devices can potentially worsen space constraints if the developer does not consider incorporating responsive design to mitigate the problem and can force the localiser to resort to unnatural solutions that may hinder the quality of the text.

2.1.4.3 App localisation

When it comes to mobile phones and other touch-screen devices such as tablets, we must differentiate between the inbuilt software of the phone itself and the downloadable apps, although they share many similarities in terms of localisation processes. Mobile phones entered the markets and became an everyday commodity in the late '90s early 2000s but it was not until the second half of the 2000s that apps (short for software applications) appeared, which entailed the development of iOS and Android as user interfaces with integrated app store-type ecosystems. Even though the software integrated into the phone has to be localised following very strict terminology rules, the bulk of the work related to small devices has to do with apps. In terms of the general characteristics of the field, this localisation type shares many of the specificities of software, game and website localisation since the product needs to undergo internationalisation in order to create a localisation-friendly environment, to eliminate (or modify) elements that are culture-related and to implement responsive design. Localisers must pay attention to the Graphical User Interface (GUI), make sure that the icons and graphics are internationally recognised and rule out potential issues later on, ensure that different date and time formats are supported, etc. Icons are of special importance in these types of devices since most apps favour images instead of text due to current trends and space restrictions.

Similarly to software localisation, the text should be separated from the code and organised in resource files and localisers should be provided with a complete localisation kit. In this particular localisation type, as in games, there are many specifications in terms of the terminology related to the operating system of the device and non-compliance can cause delays in the app's release, thus stressing the importance of guidelines and glossaries. The most common format of resource files is XML, which can be easily localised using regular computer-assisted translation tools and localisers are rarely given access to the app itself, similarly to current practices in software and game localisation. Other common practices are sending the strings in multilingual Excel files (or Google Spreadsheets) that include a column with the string's ID, the source strings, a column for the target languages and another with comments from the developers, although the latter depends on the developers' goodwill. Other typical formats are .json files, which can be dealt with in the same manner as XML files and directly uploaded to a CAT tool. Normally, apps do not tend to contain much text (around 500 to 1000 words) but they are constantly updated and the descriptions in the different stores are also part of the project, thus increasing the word count and creating a hybrid of app and website localisation. Furthermore, due to the popularity of smartphones and apps, many companies propose web-based services that extract the strings that need to be localised and outsource the process to freelancers, a practice that has been extended to software in general and seems to be reaching the gaming market as well.

2.2 Video game localisation

The following section will be devoted to video game academic research and video game localisation, the fourth and final category of localisation types. As the main subject of this thesis, we will commence with a cursory examination of the academic research carried out in the field in its origins and then continue by reviewing the historical evolution of the field. Then, we will go through the characteristics that are specific to video game localisation as well as those they share with other localisation types, textual types, translation strategies, and the assets that are commonly handed down to localisers. Finally, as mentioned in the introduction, we will review the localisation process itself and the different actors involved.

2.2.1 Video game localisation research

Even though video game localisation has greatly contributed to the success of the gaming industry, the field has been largely ignored until recently when it comes to Translation Studies in general. As an example, when we looked up "game localisation" on the Translation Studies Bibliography (TSB) website[24] (a site specialised in academic papers in the field of TS) in November of 2021 we obtained 41 hits (81 just a year later) including papers published in multiple languages. However, the same research in Google Scholar yielded 1370 results only in English (1610 in November 2022), a stark difference in the number of results that derives from "the inherently interdisciplinary nature of game localization research and particularly its close ties with the industry" (O'Hagan and Mangiron, 2013, p. 26) and shows that the field includes many papers that do not belong to the field of TS or LS. Beyond peer-reviewed academic publications, the already growing number of articles, blogs, game reviews, and personal accounts published on localisation-related websites or social media has more than doubled during the health crisis aided by an ever-increasing number of webinars and different types of online conferences.

In the past years, the number of events led by the video game development industry, video game localisers or Language Service Providers (LSPs) has grown exponentially as well, raising awareness about the importance of video game localisation for developers and putting the profession in the limelight. Consequently, the same research in Google Scholar only for articles published after 2018 shows that 587 out of those 1610 have been published in the last five years. Therefore, this section will mostly concentrate on the beginnings of research in video game localisation due to the late boom of the field since it is beyond the scope of this work to provide a comprehensive analysis of every publication. Moreover, we will mostly focus on papers, articles, conferences, monographs, chapters, and dissertations published in Spanish, French, and English directly related to video game localisation. The division of the following subsections is based on Mangiron Hevia's article "Research in game localisation" (2017).

[24] https://benjamins.com/online/tsb/

2.2.1.1 Origins: 1999 – 2006

In order to establish the framework in which the present thesis finds its place among game localisation research, it becomes necessary to review the origins of this particular academic field and the topics that have been covered so far. As such, the first articles to appear about video game localisation were written by professionals mostly depicting their personal experience and the way the industry worked following a descriptive approach that covered processes, functions, and final product. The first article about video game localisation was "Beyond PacMan: Translating for the Computer Game Industry" published in 1999 by Frank Dietz—an English into German translator—where he reflected on his work from a personal perspective. In 2001, Flavia Timiani Grant wrote "A Leisure Industry but a Serious Business" about the issues encountered when localising some games to other cultures, age ratings and legal issues in Germany. In 2002, Koichi Iwabuchi published the book *Recentering Globalization: Power Culture and Japanese Transnationalism* and wrote about translation approaches as he analysed the globalisation process in place when translating a video game from Japanese into other languages. The same year, an article entitled "Localización de videojuegos" was written by Michael Scholand (2002), another English into German translator, who described the specificities of the field and its main features. Three more papers appeared the next year; the Master's dissertation *Video Games Localization: Constraints and Choices in the Industry* (Roturier, 2003); the article "Games localization: production and testing" (Trainor, 2003), and a second article by Dietz (2003) "A Translator's Perspective on Games Localization". These are the seven first papers to ever be published on the topic and, although mostly descriptive, create the base for the appearance of the founders on the discipline in terms of research.

The years 2004 and 2005 mark a milestone for video game localisation research as three of the main figures—Minako O'Hagan, Carme Mangiron Hevia and Heather Maxwell Chandler—start publishing and contributing to the field regularly, thus creating the pillars of the discipline. Among their first contributions, we find "Localizing Final Fantasy: bringing fantasy to reality" (Mangiron Hevia, 2004), "Translating into the Digital Age: The expanding horizons of localization" (O'Hagan, 2004), and a joint presentation entitled "Games localisation: when Arigato gets lost in translation" (Mangiron Hevia and O'Hagan, 2004) at the *New Zealand Game Developers'*

Conference. Following this first collaboration, both authors continued working together regularly and published several papers on video game localisation. Additionally, in 2005 Heather Maxwell Chandler wrote *The Game Localization Handbook* which became the first monograph about video game localisation and contributed to the discipline with localisation guidelines and a large number of interviews with developers and publishers. Chandler, a professional from the video game industry with more than 23 years of interactive product development experience nowadays, has since published numerous books in game production and development targeting practitioners. One of the author's most relevant contributions is the abundance of interviews and anecdotes included in each publication since, as Mangiron Hevia points out, (2017, p. 85), academic research in video game localisation has:

> [A] limited number of participant-oriented studies using tools such as interviews and questionnaires. Chandler (2005) and Chandler and Deming (2012) include several interviews to different agents in the game localisation process, such as developers, localisation vendors, and localisation producers. Jayemanne (2009) interviewed three professional translators, while Díaz Montón (2011) interviewed eight fan translators. O'Hagan and Mangiron (2013) also interviewed a renowned game translator from Japanese into English, and O'Hagan and Chandler (2016) interviewed a US-based game localisation specialist.

The year 2006 is called by Mangiron "Year one" (2017, p. 77) for game localisation research in the field of Translation Studies due to the number of academic papers that were published on the topic. During that year Miguel Ángel Bernal-Merino, another of the most renowned researchers in game localisation, published an article in the 6th edition of the *Journal of Specialised Translation (JoSTrans)* exploring the terminology used by the industry called "On the Translation of Video Games". Another article by Carme Mangiron Hevia and Minako O'Hagan appeared in the same issue of JoSTrans under the title "Game Localisation: Unleashing Imagination with "Restricted Translation" where they discussed the term "transcreation" and set the basis for what would become "one of the key strategies in game localisation" (Mangiron, 2017, p. 77). Another article by Mangiron Hevia, "Video Games Localisation: Posing New

Challenges to the Translator", explored localisation models and assets, the importance of training translators, and some examples of cultural adaptation. O'Hagan also published an article called "Manga, Anime and Video Games: Globalizing Japanese Cultural Production." On the whole, this thorough review of most of the presentations, articles, monographs, etc. allows us to observe how the discipline took off based on empirical data, a constant throughout the years to come thus categorising the majority of the research as descriptive LS. Furthermore, cataloguing what was created during these first years shows the variety of topics covered and illustrates what Mangiron Hevia explains (2017, p. 84) about video game localisation research where "qualitative methods are predominant, and most of the research is based on first-hand industry experience and self-reflection of the authors or on textual analysis and case studies focusing on particular aspects".

2.2.1.2 Growth phase: 2007-2011

As part of the consolidation of the field and the attention it started to attract, the *International Game Developers Association* (IGDA) created the Localization Special Interest Group (Loc Sig) in 2007. This highly active body has become a driving force in the industry and hosts numerous conferences yearly, thus, its interest in localisation in particular marked a historical milestone. During that same year, the number of articles and academic papers increased exponentially and, for the first time, an issue of a journal, *Tradumàtica*, entirely devoted to game localisation edited by Mangiron Hevia was published. The authors who contributed to the said issue were O'Riada, a journalist; Dietz, Muñoz-Sánchez, and Torres in their role as experienced localisers; Loureiro, a localisation manager; and 4 lecturers from different universities: Bernal-Merino, Di Marco, Fernández, and O'Hagan. Bernal-Merino (2013, p. 354-355) later stated that even though several articles included content that was more "anecdotal than a thorough analysis of the subject, this dedicated issue of the translation journal hosted by the Universitat Autònoma de Barcelona has heralded the formal inclusion of video game localisation as an area of research within translation studies". Furthermore, the first paper based on experimental research was published in 2009, a topic that has remained unexplored in general due to the field's descriptive and qualitative approach, as explained by Mangiron Hevia (2017, p. 85):

> It should also be mentioned that experimental research, such as reception studies, is still quite uncommon. Despite the fact that the main skopos of game localisation is to reproduce the gameplay experience of the original, there are no large-scale studies confirming whether this hypothesis, which has become one of the pillars of game localisation theory, holds true. The only reception studies about player experience available are those by O'Hagan (2009a, 2016). […] Such studies combine both quantitative and qualitative data for data triangulation purposes. Mangiron (2016a) also carried out an experiment on the reception of game subtitles using questionnaires and eye tracking technology.

The year 2011 is a particularly prolific one with more than thirty articles, conferences, chapters, or similar publications on video game localisation. We observe, for example, another issue completely devoted to video game localisation, the 15th edition of *Trans: Revista de Traductología*, a journal published by the University of Málaga and edited by Miguel Ángel Bernal-Merino which offered "a comprehensive cross-section of the many aspects and decision-making processes concerned with successful video game localisation" (Bernal-Merino, 2013, p. 355). The issue included articles from scholars with professional experience such as Carme Mangiron and Inmaculada Serón-Ordóñez, and professional localisers who wanted to publicise good practices in the field such as "Enabling the Localization of Large Role-Playing Games" by Chris Christou, Jenny McKearney, and Ryan Warden. In addition, we can mention a series of presentations at high-profile conferences and events such as the *Game Localization Round Table* of Barcelona or the *Localization Summit of the Game Developers Conference* held in San Francisco.

2.2.1.3 Strengthening phase: 2012 - until today

From this point on, the number of articles published in academic journals, specialised websites, book chapters, books, PhD dissertations, undergraduate dissertations, master dissertations and conference presentations triples and providing an exhaustive review becomes practically impossible. Among the increasing number of papers and most importantly, monographs, 2012 represents another milestone for the video game localisation industry as Richard Honeywood and Jon Fung released *Best Practices for Game Localization* as part of the *IGDA Localization SIG*. As previously mentioned, Chandler and O'Malley Deming published a revised version of *The Game Localization Handbook* which included contributions in the form of chapters from various game localisation professionals. In the year 2013, Minako O'Hagan and Carme Mangiron Hevia published the second specialised monograph on video game localisation *Introduction to Video Game Localization: Translating for the Global Digital Entertainment Industry*. Their work became one of the most influential books in the field and provides extensive information about game localisation from an academic, professional, and prescriptive perspective. This period will see the appearance of a great number of books specifically devoted to video game localisation since Bernal-Merino's PhD dissertation, *The Localisation of Video Games* at the Imperial College London, would later be revised and improved, thus becoming the third monograph on game localisation under the title *Translation and Localisation in Video Games: Making Entertainment Software Global* (2015).

Bernal-Merino's dissertation and monograph focused "on game localisation from a descriptive translation studies perspective, aiming to investigate the game localisation process, mapping professional practice and discussing the implications of this new translation practice in Translation Studies" (Mangiron Hevia, 2017, p. 80-81). Adding to this trend we find that, again in 2017, "a monograph by Granell, Mangiron and Vidal was published in Spanish, which describes the main features of game localisation and also includes practical exercises" (Mangiron Hevia, 2017, p. 81). Pablo Muñoz-Sánchez (2017) published a fifth monograph about video game localisation called *La localización de videojuegos* which concentrates on video game localisation training for professionals and provides numerous examples, tips and strategies for translators including an entire chapter on transcreation.

In the case of trends related to research topics, it is necessary to mention the publication of a volume called *Fun for All: Translation and Accessibility Practices in Video Games and Virtual Worlds* in 2014 by Mangiron Hevia, Orero and O'Hagan. This work covered issues on video game localisation and featured chapters about training, cultural issues, fan translation, and accessibility. The latter has been gaining popularity in the past years in the industry, which has led to symposiums specialised in game accessibility both from the perspective of localisation and game development. Additionally, following the current trend of gender-inclusive language that has been gaining representation in publications in the last years, Cristina Pérez and Leticia Sáenz (2019) published a paper about inclusive language called "Gender-inclusive language in games and its localization challenges". Additionally, following the trend of analysing gender bias in machine translation, Rivas Ginel and Theorine studied the performance of neural machine translation in video game localisation in the presentations "Jeux vidéo, localisation, traduction automatique et problèmes de genre" and "Machine Translation and Gender biases in video game localisation: a corpus-based analysis" (Rivas Ginel and Theroine 2021a, 2021b).

In the last three years, we can mention the issue nº 244 of the French journal *Traduire*, entirely dedicated to video game localisation and published under the title *Des jeux et des mots*. The journal, usually characterised by its conservative approach, published 9 articles (and 1 table game) covering a wide array of topics ranging from the authors' perspectives and experiences to the lack of terminology resources (Theroine, *et al.*, 2021). Nevertheless and as previously mentioned, research in video game localisation has covered many aspects of the field from a predominantly descriptive perspective including:

> [M]ain features, challenges, priorities and constraints, […] the localisation process, the agents involved in it and the different localisation models, […] the relationship of game localisation with other types of translation, […] the study of different assets, text-types and paratexts, […] Japanese games and the cultural adaptation, […] translation strategies, the translation of humour, the use of terminology, […] culturalisation, […] fan translation, pedagogical and training issues, […] the topic of research itself, […] usability and accessibility. (Mangiron Hevia, 2017, p. 82-84)

Furthermore, the use of questionnaires as a method to collect information about video game localisation practices remains limited to less than a handful of papers. Moreover, both tools' usage and tools' ergonomics in this particular field are two topics that have remained largely unexplored as pointed out by Mangiron Hevia, a topic that the author labelled as a likely venue of future research stating that "the application of new technologies and their impact on localisation processes, such as an increasing use of CAT tools and localisation management tools, as well as the potential application of machine translation to certain game genres or assets." (Mangiron Hevia, 2017, p. 88). Therefore, Rivas Ginel has published a series of papers as well as some presentations on video game localisation tools that aim to assess the ergonomics of the tools used by professionals in the field. The present thesis endeavours to further research in an aspect of video game localisation that has not been extensively covered so far as there have only been two studies based on the use of a questionnaire (Mangiron Hevia, 2017, p. 85):

> Carreira and Arrés (2014) did a study with 30 game translators about their training background. Fernández Costales (2016) carried out a large-scale survey with 94 participants by means of a questionnaire about the translation of video games, language preferences, and users' habits regarding video game websites, official videos, and advertising.

2.2.2 Video game localisation throughout history

Even though video game localisation can be classified as a sub-type of software localisation and has close ties to AVT due to the presence of movie-like scenes, localisation in general was "developed initially for productivity applications and later extending to include the localization of websites, games localization presents added dimensions arising from the interactive nature of games" (Mangiron Hevia and O'Hagan, 2013, p. 19). One of the field's most distinctive characteristics is the fact that games can be considered as "multimedia interactive texts" (Bernal-Merino, 2013, p. 143) since "modern games take full advantage of multimedia and multimodality to engage the player" (Mangiron Hevia and O'Hagan, 2013, p. 19). This section presents the historical developments and changes in video game localisation practices from the origins of the field until nowadays. Therefore, we will analyse early game localisation practices that set the basis of cultural adaptation in the industry, how the process evolved, and the appearance of different levels of localisation. The last part of the first subsection is devoted to transcreation and the field's emphasis on cultural adaption, a technique that pushes the definition of functional translation and brings the player into the spotlight. The second and final part of the section focuses on the characteristics of video game localisation and the strategies to deal with text types and assets, as well as the process of localisation itself.

The development of the game localisation industry broadly follows the evolution of software and web localisation practices described in the previous section. Thus, its birth is a direct consequence of the genesis of the video game industry and the gaming platforms used. Chapter 1 provided an overview of the history of the field and showed that the first commercial games appeared in the ‘70s, however, video game localisation as such was not born until 1980. The first video games that were released into the market and achieved international fame contained short sentences or single words (or no text at all) and were mostly produced in English. Those first games relied upon simple mechanics that were easy to grasp by the players or, on some occasions, user manuals that would be translated as any other manual if necessary. Therefore, during its first decade of existence, the industry did not need to use localisation in order to sell its products abroad. The prevalence of English was due to the fact that the first video games were produced in the US and, in games such as *Space Invaders*, the Japanese company was forced to use English as technology was not advanced enough to display Japanese characters. Indeed, it was not until the invention of Unicode in 1991 that Japanese characters would be properly displayed (Bernal-Merino, 2013, p. 226):

> Japanese script was never displayed in the original version. As the game was developed in the very early years of the game industry (and even of the software industry), the only solution was to program it in English, displaying only the Roman characters utilised in English writing, since Japanese characters could not be easily shown for technical reasons. The only possibility was to treat them as images which had the inherent problem of using too much of the little memory available.

It was not until 1980 and the appearance in the US market of *Pac-Man*, the world-famous Japanese game, that what is believed to be the first example of localisation took place. Although the game itself was published in English due to the aforementioned impossibility to display Japanese characters without occupying too much memory space, the name of the game and both the names and nicknames of the ghosts were adapted in the version released in the United States. As Image 11 shows, the original

Japanese name of the game was *Puck-Man*, which is derived from /paku paku taberu/, an onomatopoeic that means "to gobble greedily" in Japanese. However, it was subsequently modified by the US company that published it, Midway, as they deemed that "the word 'Puck' would likely tempt vandals in the US to slightly alter the first letter" (Mangiron Hevia and O'Hagan, 2013, p. 49). The publisher decided to change the names and nicknames of the characters as well and ended up opting to add a humorous touch in order to create something catchy instead of descriptive (which was the case of the original version) in an effort to build on the "fun" aspect of video games. Thus, "the original Japanese nicknames of the key characters (four ghosts) were based mainly on colours, plus the demeanour of the last one, i.e.アカベイ ["Reddie"], ピンキー ["Pinky"], アオスケ ["Bluey"] and グズタ["Slowy"] became Blinky, Pinky, Inky, and Clyde in the official English translation" (Mangiron Hevia and O'Hagan, 2013, p. 49).

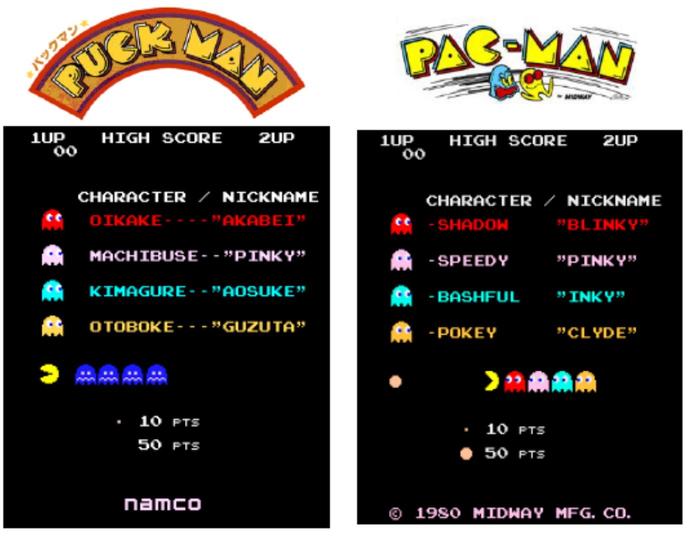


Image 11. Japanese and US versions of Pac-man (Bernal-Merino, 2013, p. 226)

Even though the game itself did not contain a large number of strings to localise, these changes set the foundations of the practices subsequently used by the video game localisation industry. Indeed, the choices they made showed "the importance of pithy and punchy-sounding renditions, even to the point of choosing entirely new names in the target text" (Mangiron Hevia and O'Hagan, 2013, p. 49) and gave localisers more freedom in terms of making decisions to improve the resulting product and making it more attractive. The need to undertake these changes goes beyond cosmetic purposes and illustrates the cultural differences between the Japanese and the US markets. Furthermore, it highlights the depth of cultural adaptation and sets a trend towards localising the names of the characters by default as Nintendo of America (NOA) realised very soon that Japanese games do not emphasise that particular aspect, contrary to Western games. Image 12 shows the original version of the instructions of the game *Donkey Kong* (1981) that were stuck to the arcade machines next to the control. In this particularly world-famous game, Nintendo simply referred to the characters as "Jumpman" and "The Lady". Therefore, Nintendo of America decided to rename them Mario and Polly in order to create something that sounded more edgy and attractive and in tune with Western culture.

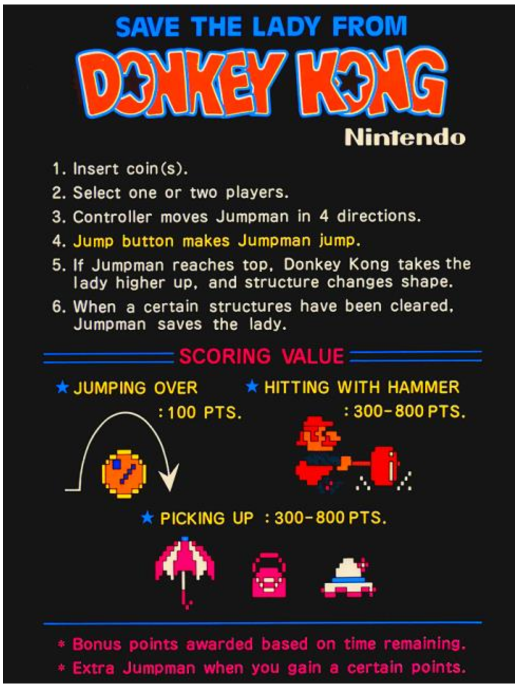


Image 12. Original Donkey Kong instructions from the arcade cabinet[25]

Consequently, the '80s are considered the early days of video game localisation, a practice that was first used by Japanese companies that became aware of the potential of foreign markets and, similarly to the software localisation industry, was characterised by a trial-and-error development process. Thus, games were localised once the original version had been released into the market and dealt with in-house by first locating the strings within the source code and then localising or adapting them (as oftentimes the game was already in English). The process was usually carried out without proper planning and as an afterthought by anybody in the Japanese department with some command of English, which resulted in numerous mistranslations and grammar mistakes. Moreover, the lack of specialised tools designed for translation purposes such

[25] http://www.dizionariovideogiochi.it/doku.php?id=donkey_kong

as translation memories or quality assurance features (as they were still being developed and fine-tuned), contributed to the presence of consistency issues and the lack of overall quality. The game *Pro Wrestling* (1986) from Nintendo contains a good example in the final sentence displayed on the screen after beating the game, which reads “a winner is you”. This grammatical error in particular became so famous that it ended up becoming an Easter egg (a message, feature or image hidden in a game) in numerous games as a tribute to the original.

Other well-known errors in games released during the ‘80s are the sentence “I would express my sincere. Thanks to You. Take a good rest!” from *Ikari Warriors* (1986); or the message at the end of the game *Ghostbusters* (1988): “Conglaturation !!! You have completed a great game. And proved the justice of our culture. Now go and rest our heroes!”. In addition to mistranslations and grammatical errors, we can also observe poor translation choices such as “I am Error.” from *Zelda II: The Adventure of Link* published in 1987. Image 13 shows a side-by-side comparison of both the Japanese and the official translation where, as the specialised website Legends of Localization explains, the literal translation should have been “my name is Error”. The sentence was an intentional pun inserted in the game by the developers as another character was named “Bug”. However, the translator in question missed the connection between “error” and “bug” and translated the latter as Bagu, thus losing the intended humorous touch[26].

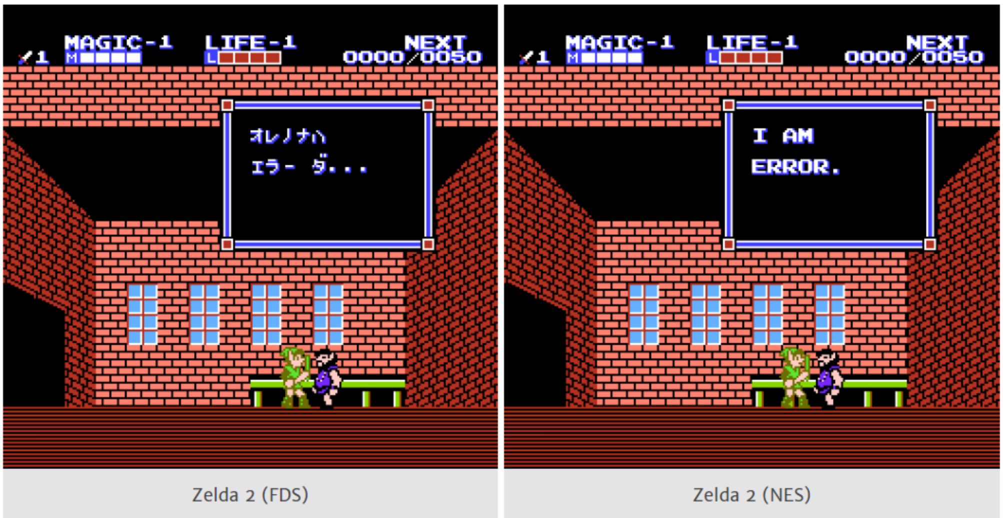


Image 13: *Zelda II: The Adventure of Link*, I am error[27].

[26] https://legendsoflocalization.com/whats-up-with-the-i-am-error-guy-in-zelda-ii/
[27] https://legendsoflocalization.com/whats-up-with-the-i-am-error-guy-in-zelda-ii/

During the ‘90s, video game localisation practices remain unpolished, originating in more mistranslations, grammatical errors, and poor-quality renditions in general. One of the most iconic mistranslations of this decade is “All your base are belong to us” from the game *Zero Wing* (1990), a sentence that became so famous that it was turned into a meme, songs, T-shirts, and many other merchandising items including face masks[28]. In addition, the game’s introduction contained a large number of mistranslations and grammatical issues and included other problematic sentences such as “Somebody set up us the bomb” and “You have no chance to survive to make your time.” Furthermore, these two first decades are characterised by the predominance of space constraints derived from technological limitations. As presented in the previous chapter, it was not until 1990 that some platforms started to use CD-ROMs instead of cartridges, which had severe limitations in terms of storage. However, some consoles such as the Super Nintendo Entertainment System (SNES) continued using cartridges until the mid-90s, thus perpetuating the issue. Due to these constraints, the text needed to be reviewed and shortened constantly to cope with the limited storage capacity of those systems. In a podcast aimed at fans and professionals, Ted Woolsey, who used to work as a translator from Japanese into English at Squaresoft back in those years, explains how hard it was to work around space constraints:

> Part of it I think was just trying to get the squeezing and squeezing space. It's one of those things where you translate a bunch of stuff and then you're told that you're at 125% capacity and you go back in and you start shortening everything. I didn't think much about it. In some cases, I would just go through and I would just run through sheet after sheet after sheet trying to squeeze stuff down to get things in. I'd give it back to the engineers and they would compile it. Then I'd be 104% over and they'd give it back to me. I'd go back again and squeeze and cut and shape and finally would go in and it was like, my god, it's finally in there. That was when we'd try to do some final polishing on it.[29]

[28] https://www.redbubble.com/i/mask/All-Your-Base-Are-Belong-to-Us-by-FlashmanBiscuit/36568589.9G0D8

[29] http://www.playeronepodcast.com/forum/index.php?/topic/145-transcript-of-ted-woolsey-interview/

The technological move from CD-ROMs to DVDs in game consoles that took place at the beginning of the 2000s increased considerably the storage capacity and alleviated some of the issues related to space constraints. The limitations that remained in spite of the bigger storage capacity derived from the character restrictions in user interfaces and menus will persist as a constant issue in the discipline. As storage and audio capacities improved and 3D graphics became common, video games increasingly adopted cut-scenes or cinematics for functional and aesthetic purposes even though throughout the '80s and the '90s "[d]espite the relative limitations of the technology, there were early signs of cinematic techniques being used in games" (Mangiron Hevia and O'Hagan, 2013, p. 51). These scenes are used to move the plot forward or introduce new characters and situations, as well as to showcase the quality of the game and create a movie-like atmosphere. Even though, as mentioned above, this technique started to be used in the late '80s early '90s, the animations and the dialogues became more realistic in order to improve the gameplay experience, which heavily impacted the localisation industry adding an extra layer of complexity. Thus, (Mangiron Hevia and O'Hagan, 2013, p. 59):

> From the point of view of translation, cut-scenes gave rise to the explicit use of subtitles and dubbing techniques similar, but not identical, to those used in AVT. Subtitle-like techniques had already been used in games for more rudimentary cinematic sequences as mentioned earlier […], and over time these developed into something closer to subtitles used in cinema.

All these technological advances required more attention to details such as fine-tuning lip-sync which led to "the entry onto the market of small localization companies enabled by the increased scope for making profits out of localization, in turn increasing competition while reducing the pricing of localization" (Mangiron Hevia and O'Hagan, 2013, p. 59). Additionally, similarly to software localisation, companies shifted towards simultaneous shipment in multiple languages in order to increase their ROI, use a single marketing campaign to create momentum, and reduce the impact of piracy. Whereas the first markets to adopt this new model were North America and the UK, Japanese companies continued using the "post-gold" model until very recently—they would first release the games to their domestic market before undertaking internationalisation and localisation, often "using English as a pivot language: all the text was translated into the target language, but the game audio was kept in English and subtitles in FIGS were provided" (Mangiron, 2021, p. 6).

Nowadays, even though simultaneous shipment has become predominant in the industry "Asian and Middle Eastern languages are also routinely released but sometimes take more time due to technical constraints with displaying Asian or Middle Eastern fonts and other features that must be added to comply with government requirements" (Chandler, 2020, p. 231). These issues tend to result in a hybrid between sim-ship and post-gold. Furthermore, the use of English is predominant due to the appearance of companies all over the world that first develop the game in their native language, then localise it into English and subsequently use the latter as the pivot language. Localize Direct, an LSPs specialised in the field, explains in its game localisation report that "[f]or non-English games, the most common practice is an initial translation into a pivot language (usually English) which then serves as the source for the rest of the languages" (Localize Direct, 2021, p. 15). This trend is reflected in the results of our first survey as we can observe that, even though the questionnaire received answers from participants originating from 56 countries and with 45 different native languages, 543 out of 620 answered that English was their main source language. If we contrast these figures with those collected in the third survey about video game developers' nationalities and the results published by the European Games Developer Federation (EGDF, 2019, p. 9-10) we can observe stark differences between the figures about countries of origin and source language. Localize Direct also highlights that this practice is used as a means to reduce costs for less common languages or with low resources while warning users that:

> Still, whenever possible, developers can request direct translations from their source language without using English as a bridge - to keep the translated content as close to the original as possible, reduce the risk of errors and start localization earlier, without waiting for the bridge language translations (Localize Direct, 2021, p. 15).

Other developments that are likely to considerably affect the localisation process in the future are the use of augmented reality and virtual reality in video games in terms of terminology or the matter of subtitles. In the case of subtitles, the position, size, and display features require special attention since "the presentation of contents is no longer purely time-based, but involves a spatial dimension, determined by both the free user's exploration and the dynamic positions where the main actions are taking place" (Hughes *et al.*, 2019, p. 221). Finally, we can observe that in recent years the market has been slowly moving towards mobile gaming. As Figure 16 shows, more than half of the revenues generated by the video game industry come from mobile games. Due to the size of the screens of these devices, the limitations of characters are more stringent for video games released for these platforms even if the developer uses responsive design.

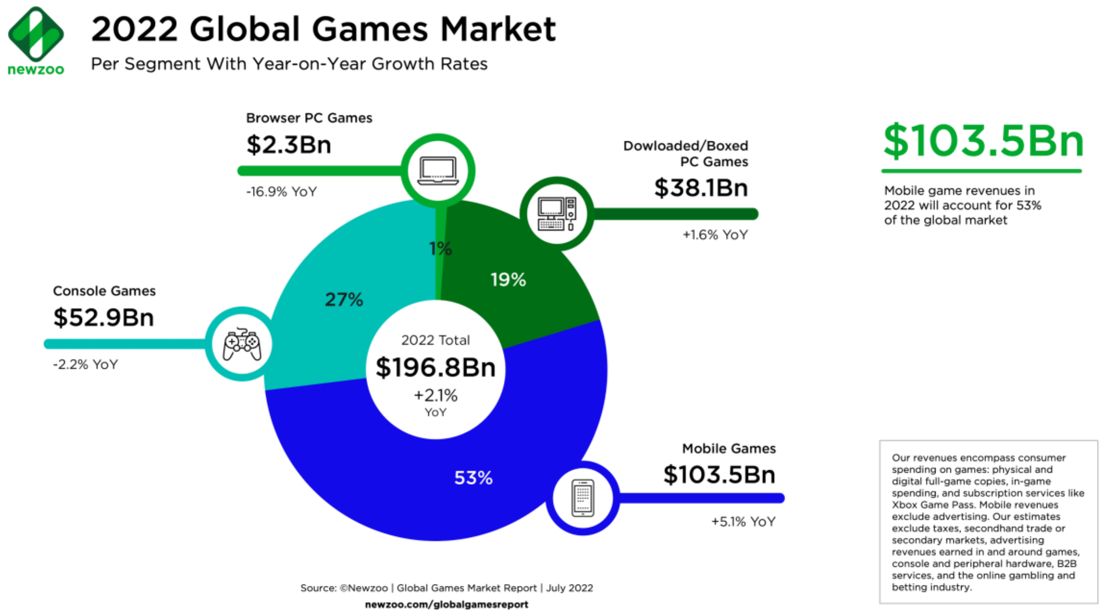


Figure 16. 2022 Global Games Market [30]

2.2.1.1 Localisation levels

During the '70s and '80s, besides the already mentioned convention of leaving the game in the original language, another common practice that prevailed when it came to localising the game into other languages, was the so-called "box and docs" (Chandler, 2005, p. 13). Similarly to the software localisation industry, the target languages were French, Italian, German and Spanish (FIGS)—or E-FIGS if the game was originally developed in Japanese. This level of localisation, still in practice nowadays, consists of localising all the printed documents that accompany the game, the packaging, and any other document that might be included in the box, leaving the rest of the user interface (UI), the dialogues, and in-game assets in English. More recently, as technologies and platforms have moved away from physical copies, the practice entails the localisation of

[30] https://newzoo.com/key-numbers

the description of the game in the various online gaming platforms or app stores, the instructions and part of the marketing material. As Chandler explains (2005, p. 13):

> "Box and docs" localization is low risk for the developer since no game code is altered. The developer might have to assist the translator with understanding the game-specific terms to be translated for the packaging. Additionally, on PC version, the developer may have to double-check the functionality of international keyboards to ensure the English keyboards commands are carried over correctly to international keyboards.

In the '90s, the trend moves towards what Chandler (2005, p. 14) calls "partial localisation". This level of localisation includes the packaging, the documents in the box, the manual, all in-game assets and a subtitled version of the dialogues and all the online material. Whereas the previous technique is used to achieve a limited number of extra sales in small secondary markets, "partial localisation" will be used for important secondary markets with an interesting ROI. The only difference with a completely localised game is the fact that the audio files are left in the original language and subtitles are added instead. This level is efficient from an economic point of view as it saves the costs of voiceover actors, lip-sync efforts, assets integration, etc. "Additionally, properly integrating localized voiceover files into the game can be challenging, specifically if lip-sync is used for the in-game characters. When lip-sync is used, facial animations might have to be redone" Chandler (2005, p. 14).

During the 2000s, AAA games (due to the budget requirement of this practice) established "full localisation" (Chandler, 2005, p. 14) into 10+ languages as the norm in the field. In this level of localisation, similar in practice to "extensively localized websites" (Jiménez-Crespo, 2013, p. 34), all the assets and materials available to the player are localised: in-game text, audio files, manual, website, etc. "Every aspect of the game has to be thoroughly reviewed to make sure that all the text and voiceover files are localized. This can be costly and challenging if the game code is not localization-friendly and the assets are not well organized within the code" (Chandler, 2005, p. 14). However, it is not uncommon to combine "full localisation" with "partial localisation" for the main secondary markets in games with a smaller budget. Additionally, although "publishers automatically plan for French, German, Italian, and Spanish localizations,

[…] it is becoming increasingly popular for developers to translate games into Russian, Japanese, Korean, and Hebrew” (Chandler, 2005, p. 3). Thus, in the present day, besides FIGS (or E-FIGS) the market has also seen the appearance of the acronyms BRIC (Brazil, Russia, India, and China) and CJK (China, Japan, and Korea) due to the emergence of these markets in the video game international scene.

Furthermore, individual game developers (indie devs) and hobbyists either release their video game in English regardless of their country of origin and opt for a “box & docs” approach or use “partial localisation”. The industry has also seen a resurgence of crowdsourcing and the use of machine translation in order to provide a multilingual product while reducing costs. Finally, Bernal-Merino adds a fourth level of localisation called deep or enhanced localisation, arguably similar to “culturally adapted websites” (Jiménez-Crespo, 2013, p. 35) but that in practice goes beyond most web or desktop software localisation due to the interactive, immersive, and multimodal nature of most modern games. Bernal-Merino describes enhanced localisation as (2013, p. 243):

> [A]ny amendment that does not run counter to the game-world itself, and is capable of increasing the immersion of players through familiarity with gameplay features and specific story preferences can be reconsidered and adapted to cater successfully to a particular local market. It may seem that there is nothing novel in this approach, but whilst previous efforts had aimed at breaking into markets by simply eliminating language barriers, deep or enhanced localisation staves off competition from other top games also offering a fully translated content by presenting a product catering directly to local tastes and sensitivities in a systematised way.

2.2.1.2 Transcreation

The 2000s also see the birth of transcreation or, at least, the use of this specific term in the context of video game localisation since, as Bernal-Merino (2013, p. 129-130) points out, the term had already been used in Translation Studies before, once in 1964 by Lal to talk about the translation of Indian plays, and a second time in 1998 by Vieira to talk about poetry translation. Nowadays it’s a commonly used concept for translation in marketing, among others. Bernal-Merino describes it as “a translation that completely tilts the balance towards the target audience but claims to be the same product, despite those differences" (2006, p. 34). In his doctoral thesis, he provides an example of transcreation applied to character names (Bernal-Merino, 2013, p. 129-130):

> Pedro Picapiedra [Peter Chipstone] is the Spanish name for 'Fred Flinstone', the protagonist of the popular cartoon TV series The Flintstones. Translators not only maintained the reference to the original 'stone' but they also came up with a new alliteration of the phoneme /p/ present in the Spanish lexical family of piedra [stone] and reinforced by the verb picar [to chip] and the proper name given to the character, Pedro. The end result is both a meaningful and playful name that is certainly very creative. In this sense, it seems rather unclear where translation finishes and transcreation begins.

Di Giovanni (2008, p. 33) adds that "[t]he transcreated text had to be entirely fluent and, most importantly, it had to be fully understandable to its target audience". The practice of transcreation in the field stems from the fact that one of the priorities in video game localisation is to achieve immersion. As Christou, McKearney, and Warden (2011, p. 40) point out "[t]he mark of perfect localization would see a player considering a video game to have been created in his culture, for his culture. That is, the ideal localization would engender a complete suspension of disbelief". In this sense and compared to other localisation types, video games push even further the concept of functional translation found in the Skopos Theory of Vermeer and the Theory of Translatorial Action of Holz-Mänttäri. As Odacıoğlu (2017, p. 24) explains:

> It is accepted that the Skopos Theory and Mänttari's Theory of Translatorial Action are prospective theories advocating a target-oriented approach. Consequently, these theories attach importance to the expectations of the target group, which is more relevant for the category of cultural transformation/turn, rather than sentence-based approaches and the seeking of (linguistic) equivalence, which is generally analyzed under linguistic transformation/turn. Therefore, according to these theories, the prime objective of a translator is not to

> seek equivalence but to produce a translated text by making translation decisions that are most relevant for the target audience.

To this end, game localisers are allowed to take more liberties, as Mangiron Hevia and O'Hagan (2006, p. 6) explain "localisers are granted *quasi* absolute freedom to modify, omit, and even add any elements which they deem necessary to bring the game closer to the players and to convey the original feel of gameplay". Bernal-Merino (2006, p. 34) adds that "from the translational point of view, this is the only product in which the linguistic transfer is part of the development process and can, therefore, affect the actual creation of the video game". In other words, transcreation is at the heart of game localisation and is encouraged by companies in order to further immerse the player and improve the game experience. Anecdotally, another reason for the use of transcreation during the '80s and the '90s was pure necessity, as explained by Minako O'Hagan and Carme Mangiron (2013, p. 54):

> Such liberties were sometimes also taken out of desperation rather than as a creative addition. In the case of the Japanese RPG (J-RPG) Story of Thor (1994), re-titled as Beyond Oasis in English for Sega Genesis, the poorly translated story and dialogue, which did not make sense to the English editor, were completely re-written, simply using plot points.

2.2.3 Characteristics, text types, strategies, and assets

In general, most of the technical constraints present in video game localisation resemble those found in the other types of localisation. First and foremost, the game must undergo internationalisation and follow a series of guidelines in order to prepare the programme to support multiple languages. Thus, the game needs to be capable of displaying special characters and different time and date formats. Other recommended internationalisation practices are creating menus and dialogue boxes with enough space to account for language expansion, including responsive design, and using fonts with variable width in order to save space. Additionally, developers need to separate the translatable strings from the code and make sure that nothing has been hard-coded, that the game can include subtitles both for partial localisation purposes and accessibility and, in PC games, the system must support international keyboards. Developers should also avoid embedding text in art assets, create a standard naming convention that is well-defined for audio and cut-scenes, and create a localisation kit as complete as possible. Many experts recommend game developers to plan for video game localisation from the beginning of the project and to create a localisation-friendly game because (Chandler, 2020, p. 233):

> If developers plan ahead in the development process, they can create localization-friendly code, which will help them to avoid obstacles and delays later, when the game needs to be localized. Localization-friendly code takes into account technical, translation, integration, and testing needs.

Due to the fact that video games are pieces of software (and similarly to the rest of the localisation types), once the localisation phase has been completed, the resulting product will need to be compiled and undergo a specific LQA process (linguistic quality assurance) in order to check for the presence of implementation errors or linguistic bugs, as explained in Chapter 1 section 1.4.5. Therefore, in this sense, we can observe the aforementioned integration of the localiser in a team that handles a complex localisation process requiring expertise in fields that range from programming and engineering to translation and content management and might force them to play various roles (Odacıoğlu, 2017, p. 26). Similarly to software and app localisation, nowadays video game localisers usually receive Excel files or Google Spreadsheets (which creates a scission between the actual visual environment and the source text) that contain the string's ID, the source text, and the developer's comments. However, contrary to the limited word count that characterises productivity software or apps, some games can easily surpass the hundreds of thousands of words: "*Mass Effect* 2 (2010) contained 440,000 words with 30,000 VO lines while the MMORPG title *Dragon Age: Origins* (2009) had around one million words with 56,000 lines of VO" (Mangiron Hevia and O'Hagan, 2013, p. 143-144). Furthermore, video games comprise a wide variety of assets that require different approaches in terms of localisation strategies depending on their function and whether they contribute to the storytelling or not.

Relationship to the game world	Translation assets	Text function and description	Characteristics / translation brief	Translation priorities and strategies
In-game text assets				
Non-diegetic	User Interface (UI)	Informative function for smooth navigation and gameplay. Typically contains short text fragments, such as menu items and also help messages.	Brevity due to space constraints; user-friendliness of text; clarity of text.	Pragmatic and functional choice to address space constraints; creative solutions to overcome space constraints and also to reflect an edgy feel often imbued in game text in terms of expressions and naming of certain items.
Non-diegetic	System messages	Informative function for instructive pragmatic purposes. Messages generated by the system, such as warning messages, instructions, and confirmation messages.	Platform-specific terminology needs to be used.	Prescriptive, conforming to the existing terminology and phraseology of the **platform holder**
Diegetic	Narrative text	Expressive / informative function for imparting certain information in a dramatic manner. Literary passages used to engage the player in the game world or to a new level within the game. They contextualize and provide information about the game story, including a backstory.	Often formal and literary style; natural flowing writing style often asked and separate rewriting may be applied.	Fluency in TL with appropriate register and style.
Non-diegetic	Exposition / tutorial	Informative function with instructive and didactic messages. In-game tutorials may be used to explain game mechanics by way of demonstration and the player practice. Passages describing characters, monsters, animals, geographical locations, etc.	Clarity and informativity are stressed.	Functional while remaining faithful to the instructive intention and the original characterization of main game characters.
Diegetic	Unvoiced dialogue scripts	Informative / expressive function mainly to provide information and elicit a certain action by the player. Dialogue which appears only in written form, commonly used for Non-Playable Characters (NPCs).	Speech expressed in written text with colloquial style; natural flowing style may be asked.	Fluency in TL typically with casual register to reflect a conversational style.
Art assets (textual graphics)				
Diegetic	Text in images	Informative / expressive function to give the player certain information such as clues in an authentic atmosphere. Any in-game art assets containing text (poster, billboards, maps, etc.).	Varying styles with some space constraints; informativity in case of providing clues is stressed.	Informative function must be prioritized in the case of crucial clues being given; visual / aesthetic dimensions also need to be considered.
Non-diegetic	Text in images	Informative / some persuasive function to provide the player with information not related to the game and to raise brand awareness and loyalty. Game logo art which may need to be translated and re-designed.	Consideration of space constraints and consistency in case of precedence where a certain translation is previously used / officially registered.	Prescriptive approach to conform to official recommendation or prior translation which may be legally binding.
Audio and cinematic assets				
Diegetic	Lip-synch voiceover	Informative / expressive function to provide a clue or a backstory in a dramatized manner.	Oral text with character-specific idiosyncrasies; natural flowing writing style is often called for; dubbing actor/director may suggest changes to the translated script.	Prioritizing lip-synch / space constraints; fluency in TL; characterization may involve the use of linguistic variation and may involve rewriting.
Diegetic	Non lip-synch voiceover			Fluency in TL with correct register, style.
Diegetic/non-diegetic	Songs performed by game characters/ theme songs	Expressive function. Lyrics of songs in the game soundtrack may be translated and re-recorded by a TL singer.	(Cont. from previous page.)	Retaining appropriate thematic feel; may involve rewriting lyrics by involving a TL musician.
Diegetic	Environmental sound	Expressive/informative function for realism and for dramatizing. Various sound effects to enhance the atmosphere.	Socio-culturally appropriate choice must be made for the given sound source in case of cultural differences.	Socio-cultural considerations.
Printed materials				
Non-diegetic	Manual	Informative function for instructions. A hardcopy manual contains information and instructions to get started with the game, whether or not the player actually uses it. This may also include a booklet which may function as a bonus material.	Varying text types, ranging from informative and technical to promotional. When translated by different translators the translation of terms and names must be consistent with the relevant in-game text.	Informativity with pragmatic, functional orientation; may involve re-ordered layout (see Figure 3.2).
	Strategy books	Informative function for instructions. Strategy books functions as a comprehensive walkthrough.		
Non-diegetic	Box	Persuasive/informative function to appeal to the prospective customer while providing product information. Relevant text on packaging. Minimum level of localization, so-called "box and docs", only involving translation of manual and packaging.		Fluent TL, right feel, advertising/ marketing oriented language use prioritized.
Non-diegetic	Other associated paratext, including advertising text (e.g. posters) and strategy books published separately	Persuasive/ informative function to appeal to prospective consumers and to provide information such as the game's release date, content and playing guidance. Texts of a varied nature used for legal, marketing, promotional purposes, such as press releases, health and safety precautions, etc.	(Cont. from previous page.)	Free marketing style writing to appeal to users; consistency with similar text used elsewhere within the product; prescriptive with some legal and technical information.
Online/screen materials				
Non-diegetic	Other associated paratext, including the game's official websites and TV ads.	Persuasive / informative function to whet appetite of prospective consumers with some informative content. Texts mainly for marketing and promotional purposes (including TV ads), such as press releases, health and safety precautions, etc.	Natural flowing style to appeal to the audience required; in case of references to names and key terms must correspond to those used in the game.	Free marketing style writing; consistent with similar text used elsewhere within the game; prescriptive with some legal and technical information.

Table 9. Text taxonomy (Mangiron Hevia and O'Hagan, 2013, p. 155-158)

Mangiron Hevia and O’Hagan’s Table (Table 9) provides a comprehensive taxonomy of the different types of texts that can be found in a “story-oriented console game text” (Mangiron Hevia and O’Hagan, 2013, p. 153) in order to portray the function of the text and provide a short introduction to the strategies and assets. The authors distinguish five different types of assets: in-game text assets, art assets, audio and cinematic assets, printed materials, and online/screen materials (Mangiron Hevia and O’Hagan, 2013, p. 155-158). Nowadays, due to the popularity of online distribution methods, printed materials have become rare and games tend to be sold directly online with only a minority of games releasing physical copies. In both cases, as the table shows, these assets are either used to provide instructions or to promote the game and are closely related to technical translation, marketing translation, AVT translation, and website localisation depending on the content and the support.

Although the inclusion of cinematics and the use of dialogues suggests proximity with AVT translation practices “[e]ven games developed by the same company or distributed by the same publisher lack a consistent approach to subtitling” (Bernal-Merino, 2013, p. 174). Consequently, video games do not follow standardised conventions in terms of layouts and, among the discrepancies in style and position, “they can combine monochrome and polychrome fonts; they use different types of fonts; they can be static or dynamic, and they can be presented in different parts of the screen: bottom, top, sides, and inside speech bubbles” (Bartoll, 2008; cited in Mangiron Hevia, 2013, p. 48). The industry seems to rely on the player’s ability to pause the game if necessary and does not follow the recommendations in terms of the number of lines, characters per line, or on-screen display time. Regarding the number of lines and the number of characters, we can observe further inconsistencies and modifications due to the different screen sizes. Concerning on-screen display time, it is not uncommon for developers to hard-code entry and exit timestamps for subtitles based on the source language, not taking into account language expansion and ignoring what could be considered a comfortable reading speed.

Video game localisation gets once again closer to AVT for voiceover and dubbing techniques as localisers must pay, now more than ever, special attention to lip-sync due to the improvement in the quality of graphics and their movie-like style. Therefore, the text might need to be rewritten in order to match the character’s lips once localised.

Anecdotally, the adaptation can be as profound as the famous transformation (along with other cultural considerations) of "arigato" into "I love you" in *Final Fantasy X* (Mangiron Hevia and O'Hagan, 2013, p. 173). However, recent technological developments such as the plug-in for Unreal Engine "Oculus Lipsync" might simplify the task. Although such tools have been around for a while, as presented in Bernal-Merino's Master's dissertation their use "is not however standard practice and lip synchronisation remains a challenge in the field of video games" (2013, p. 115). "Oculus Lipsync":

> [A]nalyzes the audio input stream from microphone input or an audio file and predicts a set of values called visemes, which are gestures or expressions of the lips and face that correspond to a particular speech sound […] Oculus Lipsync uses a repertoire of visemes to modify avatars based on a specified audio input stream. [31]

As technology advances and games become more sophisticated "the macrostructure of game text has also become more complex and non-linear" (O'Hagan and Mangiron 2013: 150). Nowadays, as video games move away from physical copies and the mobile gaming market gains terrain, the borders with other types of localisation seem to fade. For instance, the appearance of browser games with constant content updates brings the discipline closer to website localisation, thus creating a long-term association between the developer and the localiser. Moreover, the late boom of mobile games and their app-like nature in combination with the shift towards a Games-as-a-Service (GaaS) business model, is pushing towards monetisation and frequent game updates—a prerogative of app localisation. Nonetheless, one differentiating characteristic of video games is the combination of system messages, user interfaces, a ludic dimension and storytelling; a mix where the last two become essential for the immersion of the player. Bissell (2010, p. 37) explains that video games that rely on a narrative structure employ two different types of storytelling where one would be the narrative proper, represented by cinematics where the player has no control over the unfolding of the story and the second one:

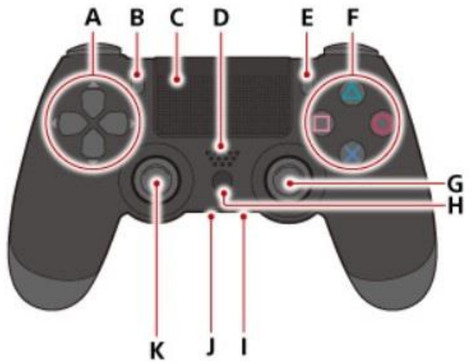
[31] https://developer.oculus.com/documentation/unreal/audio-ovrlipsync-unreal/

[W]hich some game designers and theoreticians refer to as the "ludonarrative," is unscripted and gamer-determined--the "fun" portions of the "played" game--and usually amounts to some frenetic reconception of getting from point A to point B. The differences between the framed narrative and the ludonarrative are what make story in games so unmanageable: One is fixed, the other is fluid, and yet they are intended, however notionally, to work together (ibid).

The combination of all these aspects is exclusive to video games and essential for the player's immersion which in turn, puts a greater emphasis on the localisation and adaptation of the content to achieve the aforementioned "complete suspension of disbelief" (Christou *et al.*, 2011, p. 40). Conversely, non-diegetic assets also play a major role in video game localisation due to the strict terminology requirements from platform holders since, if the guidelines of the hardware manufacturers are not followed, the game could suffer significant delays if it fails the submission process (Mangiron Hevia and O'Hagan, 2013, p. 123). The authors explain that "[u]sing 'analogue stick' in a game that is going to be published for Microsoft Xbox would mean that the game would be rejected and would have to go back to the developer, who would have to make the necessary changes and resubmit it" (ibid). In order to prevent these issues, localisers should be provided with updated and well-maintained terminology databases as platforms are continually being improved. Additionally, many games are released simultaneously for multiple platforms, increasing the likelihood of terminology inconsistencies (Bernal-Merino, 2013, p. 336):

For example, game controllers for the three main desktop consoles have small joysticks which players manipulate with their thumbs in order to play the game. This joystick is called the 'analog stick' [Joystick analógico] for Sony's Playstation, 'thumbstick' [Stick] for Microsoft's Xbox, and 'control stick' [palanca de control] for Nintendo's Wii.

The following image (Image 14) further shows the differences between the terminology used by Sony and Microsoft in their respective controllers "DUALSHOCK™4 wireless controller”[32] and “Xbox One Wireless Controller”[33]. This example portrays the great difficulty to maintain consistency without appropriate resources.



A) Directional buttons
B) SHARE button
C) Touch pad/Touch pad button
Press the touch pad to use the touch pad button.
D) Speaker
E) OPTIONS button
F) △ button/○ button/✕ button/□ button
G) Right stick/R3 button
Press down on the stick to use it as the R3 button.
H) PS button
I) Stereo headset jack
J) Extension port
K) Left stick/L3 button
Press down on the stick to use it as the L3 button.

Original Xbox One Wireless Controller front

1 Left stick	**9** Expansion port
2 Left bumper	**10** Right stick
3 View button	**16** 3.5mm port
4 USB charge port	**X** X button
5 Xbox button	**Y** Y button
6 Menu button	**A** A button
7 Right bumper	**B** B button
8 Directional pad (D-pad)	

Image 14. Comparison of the terminology “DUALSHOCK™4 wireless controller” and “Xbox One Wireless Controller”.

Bernal-Merino (2013, p. 192) identifies three main factors that differentiate video games from other localisation types: “story-building interactivity, the fragmentation of the source text, and the translation of variables” (ibid). In order to recreate interactivity without breaking the player’s immersion, the developers provide NPCs (non-playable characters) with multiple lines and options following a tree-based dialogue format that allows the NPC to react to the player’s actions or choices to simulate spontaneity. This approach poses important challenges to the localiser for various reasons that range from the complexity of the structure to the inability to discern the gender of the addressee without context or access to the visual environment. The fragmentation of the source

[32] https://manuals.playstation.net/document/en/ps4/basic/pn_controller.html
[33] https://support.xbox.com/en-US/help/hardware-network/controller/xbox-one-wireless-controller

text is an intrinsic characteristic of localisation practices due to sim-ship approaches, the dynamic nature of some of the products and the development process itself. In other words, it is not uncommon in the field to work with Excel files containing strings that do not follow a linear exposition of the events. However, "[s]tripping the text from its inherent linearity does not constitute much of a problem as far as the functionality of the software is concerned because of the practical and pertinent nature of these texts" (Bernal-Merino, 2013, p. 197). Conversely, this particular practice in video game localisation can heavily impact storytelling and create unexpected issues. Finally, the third factor is the presence of variables (Bernal-Merino, 2013, p. 205):

> In games, variables can stand for a number of points or coins gained, as well as for a word such as 'elf', 'stupid', or 'northerner'. As has already been pointed out, this is directly relevant to the interactivity in games because it allows players to choose a wide number of attributes for their characters such as: name, gender, profession, nationality, and religion. For this approach to work successfully, a text for translation, which addresses users, will need to be linguistically unfinished so that the game can allude to those characteristics as players select them.

Even though variables (or placeholders) are not exclusive to video games and can also be used in software and apps, video games' requirements in terms of linguistic quality stress the importance to find creative and free-flowing solutions. Some examples of the use of variables are the sentences "%s, welcome to the game", "you found %s", or "you earned %d experience points". The programme subsequently substitutes "%s" with the name of the player in the first case, the corresponding name of the object chosen from a pre-established list in the second case, and a number in the last case. In the first two cases, the main obstacles are taking into account potential changes to the syntax of the sentence and finding a way to account for all gender and number possibilities. Even though the most elegant solution would be to create various strings during the development phase of the game, it is not uncommon for localisers to be forced to use controlled language. In other words, over-simplifying grammar and vocabulary to avoid issues, an undesirable technique "for narration and dialogue, because these types of text should be creative and varied if their ultimate goal is to heighten the enjoyment of the gaming experience" (Bernal-Merino, 2013, p. 208). Finally, some games use concatenated strings (multiple variables in a single sentence) which can become a major challenge for many languages, one example would be "'<ADJ> <NOUN> from <BAND> at <VENUE>'. The game code includes lists of variables where each 'adjective', 'noun', name of 'band', and name of 'venue' is allocated a linguistic value" (Bernal-Merino, 2013, p. 209). As the author explains, even though this works for languages such as English, it would cause many errors in others such as Spanish or French (ibid).

2.2.4 The localisation process

As Chapter 1 showed, from publishers to platform holders passing through developers and, in some cases, language service providers, there are many actors involved in the development of a video game that influence the localisation process in different ways. Their influence might depend on the decisions made by the entity in charge of the commercialisation of the game, the expected return on investment, and the targeted countries. Furthermore, as explained in Chapter 1, whereas some platform holders may have their own localisation department—thus controlling the whole process—some publishers may include the localisation of the game as part of the deal with the development studio. Regardless, the first link between the developer and the localiser arises from the internationalisation process where the developer's task is to prepare the game for translation allowing for special characters, leaving extra space in the UI, etc. Then, we can observe a second connection emerging from the assets' extraction and implementation as well as the support developers—or the person appointed for this purpose—provide to localisers in terms of reference material, comments, or simply answers to possible questions that may spring up during localisation. Finally, although indirectly, platform holders also play a major role in localisation decisions as they impose terminology requirements in order to publish the game and non-compliance can cause bottlenecks in the process. As Mangiron Hevia and O'Hagan (2013, p. 80) phrase it "localisation is subject to decisions made by publishers while technically its process is closely linked to game development which in turn is influenced by the various specifications of the particular platform". The diagram (Figure 17) proposed by Mangiron Hevia and O'Hagan (2013, p. 129) provides a comprehensive overview of the development, pre-localisation, localisation, post-localisation and production phases of a video game and the different stages involved in the process.

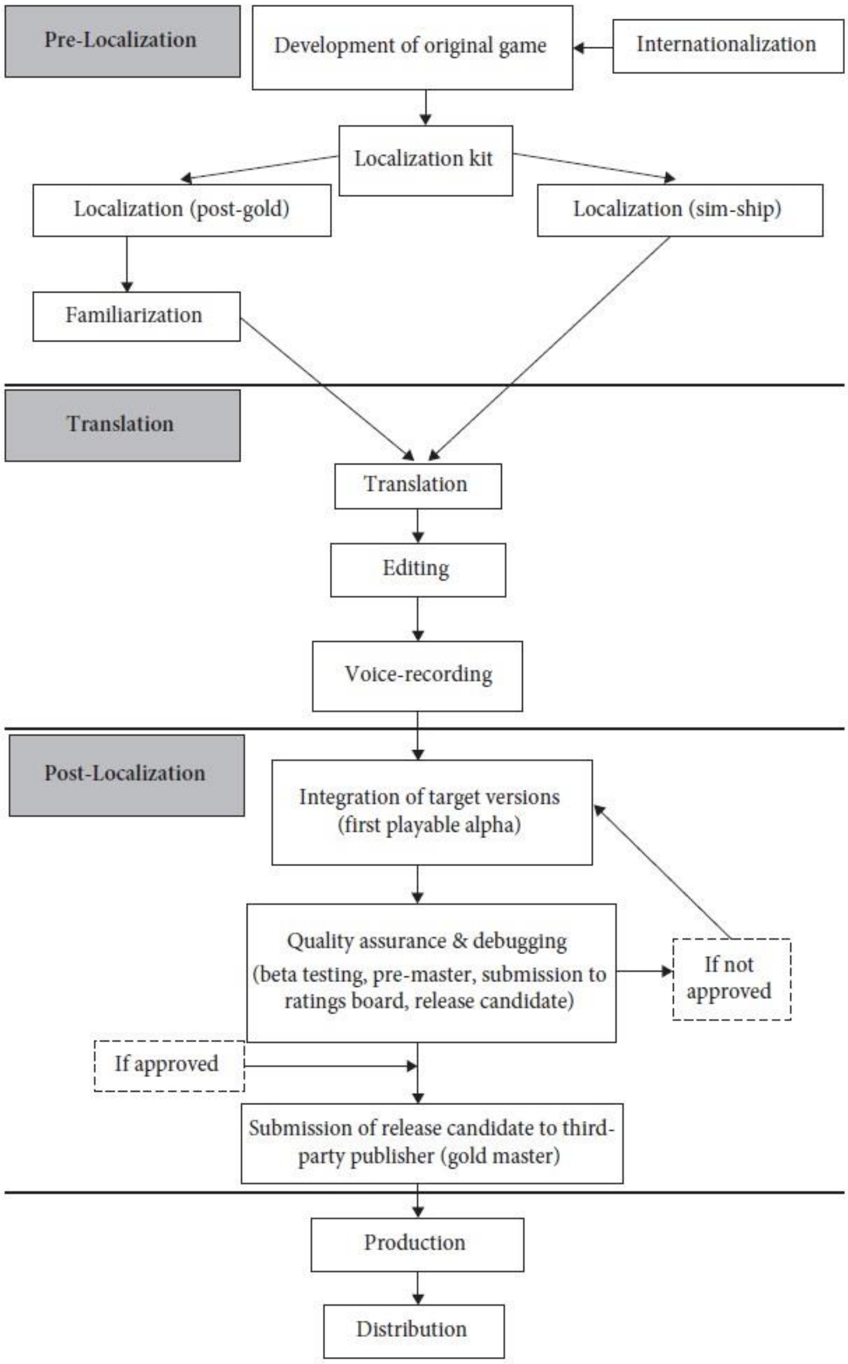


Figure 17. Steps in the localisation process (Mangiron Hevia and O'Hagan, 2013, p. 129)

As previously mentioned in Chapter 1 section 1.2.3, the number of professionals involved in the development, translation, production and distribution of a AAA video game can include up to "4388 persons […] 29 different development studios all over the world" (Toftedah and Engström, 2019, p.3). The diagram includes the different steps from a simultaneous shipment release point of view and a post-gold release point of view. Although in the case of some games and markets, these two release models may be used at the same time (i.e. in the case of Asian languages), for the most part, companies have been moving towards sim-ship practices (as can be seen in Chapter 3 section 3.3.4) and abandoned the post-gold model. As explained in Chapter 1, whereas the former entails the distribution of all the versions of the game at the same time, the latter consists in finishing the original version of the game first and subsequently localising it. The main differences from a localisation point of view are the lack of access to the video game for familiarisation, the impact of constant modifications due to the fact that the source text is not final, and the reduced amount of reference material as many assets are still under development. Following Figure 17, we can therefore observe that the first phase of the localisation process, the pre-localisation phase, is dealt with by the development team.

In the second phase, which involves the actual translation of the game, we can observe the evolution of two distinct options that have prevailed throughout history. The localisation process in the video game industry, similarly to translation in general, broadly follows two models: outsourcing and in-house. The industry evolved and, although in the '80s and '90s, companies created localisation departments and dealt with the process in-house, nowadays outsourcing has become the norm. The first difference between the outsourcing and the in-house model is the reduction of costs since maintaining a team of translators tends to be more expensive than working with an LSP that will subsequently hire freelancers for each language pair. In this outsourcing model, freelancers usually do not have any type of contact with other members of the localisation team and must go through a project manager appointed by the vendor. This draws a parallel with how publishers will sometimes impose their own producers on a development studio (Chapter 1 section 1.2.2) complicating communication tasks.

Furthermore, oftentimes the LSP's project manager's sole connection is a localisation coordinator selected by the developing company who acts as the link between the development team and the vendor's project manager. "The project manager supervises the work of the translators throughout the project, collates their queries, sends them to the developer or publisher and liaises between the translation team and the client" (Mangiron Hevia and O'Hagan, 2013, p. 128). This extra link in the chain adds to the complexity of the exchanges and can significantly slow the communicative process.

Another particularity of the outsourcing model is the lack of access to early builds of the game, which forces localisers to do the preparatory work by researching the genre of the game, playing similar ones, watching trailers, etc. The previously mentioned lack of linearity is worsened due to tight deadlines, a high word count, and the act of dividing the source documents among different freelancers that might not be able to communicate among themselves and only have access to their part. Finally, depending on the services proposed by the LSP, the contract might also include voiceover recording and linguistic quality assurance (linguistic testing services).

Conversely, in-house models usually have a small localisation department with various localisers who resort to freelancers if necessary, thus reducing the number of links in the chain and potentially increasing the localiser's access to resources and reference materials, a fact that we will be able to study in-depth in Chapter 5 section 5.2. These freelancers will be in direct contact with the localisation coordinator, "who will manage the project in the different languages, answer queries, solve any problems that may arise and ensure that the deadlines for the project are met" (Mangiron Hevia and O'Hagan, 2013, p. 128). Other responsibilities of the coordinator are acting as a liaison between the freelancers and the development team and, on some occasions, selecting the localisers involved in the project (Mangiron Hevia and O'Hagan, 2013, p. 128).

On the whole and following Figure 17, localisation processes can be divided into 4 distinct phases: pre-localisation, localisation, post-localisation, and production and distribution (Mangiron Hevia and O'Hagan, 2013, p. 129). The pre-localisation phase includes creating the localisation kit, a responsibility that falls on either the developer or the publisher (depending on the contract specifications), appointing the coordinator and selecting the localisers, and any preparatory work—either playing the actual game (post-gold), going through the localisation kit, or reviewing past translations (Mangiron Hevia and O'Hagan, 2013, p. 130). Other tasks included in the localisation phase are editing and voice recording. Editing is basically proofreading and reviewing the localised assets and:

> In the in-house model if there is a team of translators they usually review each other's work. After that, editors employed by the developer or publisher (or, time and budget permitting, an external

> vendor) may carry out a thorough review of the translated material to ensure that there are no errors and that the team's translation is coherent and consistent. In the outsourcing model it is usually the vendor who performs the editing. Reviewers may make the appropriate changes to unify the style and the terminology used in the game in order to guarantee the quality of the localized product or they may simply indicate the suggested changes and corrections to the translators, who will then implement them in their files. (Mangiron Hevia and O'Hagan, 2013, p. 134)

Voiceover requires prioritising the localisation of certain strings, thus further increasing the lack of linearity, as well as more in-depth adaptations for lip-synching. Regardless of the process being carried out in-house or outsourced to either an LSP or a recording studio, the strings will need to be localised in advance and then edited to allow enough time for the recordings to be made and the audio files to be integrated into the localised versions. There are different requirements depending on the types of recordings (besides lip-sync) as some of them will not be subjected to any type of synchronisation constraints, others will have time limitations based on the duration of the original audio, and others will need to be adapted to match pauses without paying special attention to lip-sync (Mangiron Hevia and O'Hagan, 2013, p. 135). Another difference between the translation of traditional audiovisual products and video game localisation is the presence of "stitches" which are "short audio files containing utterances made by game characters, segmented and recorded separately, so that they can be used at different stages of the game as appropriate, with variables inserted in run-time" (Mangiron Hevia and O'Hagan, 2013, p. 136).

The third phase of the localisation process starts by integrating the localised assets into the game, a process usually done thanks to the asset pipeline (Chapter 1 section 1.3) and the creation of the first playable alpha. This version will be subsequently playtested in search of both functionality and linguistic bugs as described in the previous chapter. This extra step included in the process constitutes another difference with other translation fields that simply undergo a proofreading pass done either by the translator itself or an external person whereas (Mangiron Hevia and O'Hagan, 2013, p. 137):

> [I]n localization there is another type of check after the editing has been finalized. This is due to the nature of localization, namely that: (a) the TT is embedded in electronic form; (b) the ST is often unstable; (c) sometimes there is no access to the original game and no contextual information, and (d) several translators and reviewers participate in the localization process (although this is not unique to localization).

This quote summarises the complexity of the field itself and justifies the necessity of including questions about business practices (and tools' usage) in all three of our surveys in order to obtain quantitative data that will allow us to: (a) analyse current practices and evaluate their impact on the final product from multiple angles and (b) examine the solutions in place to deal with these constraints and improve localisers' working conditions.

CHAPTER 3. FIRST SURVEY

The previous two chapters, included in order to better understand the context within which our three surveys were created, have also provided a historical overview of the video game development and localisation industry as well as a cursory glance at the actors, processes, and main difficulties involved in developing a video game. Thus, we have observed how the increasing complexity of the technology used—and the content included in the games themselves—throughout time has forced the industry and its workers to specialise further. We have also reviewed the improvements brought up by the refinement of localisation practices, the democratisation of internationalisation, and the shift from post-gold production methods towards simultaneous shipment. Additionally, we have briefly introduced game engines, development cycles, milestones, localisation processes, and linguistic testing methods as described by practitioners and scholars. However, as Chapter 2 explained, academia has not explored all the possibilities offered by questionnaires in this multidisciplinary field and we are not able to objectively analyse and quantify the practices described in the previous two chapters.

The present chapter is the first of two focusing on presenting the raw results of the three surveys that can be found at the heart of this thesis and Chapter 5 will present the cross-analysis of all the results. Therefore, the data from the questionnaires themselves will be presented in chronological order of implementation and subsequently commented on, contrasted, and studied in a separate section due to the strong dependencies between the localisation process and the development and testing phases. Our three surveys were created to specifically target three strategic job positions in the industry that highly impact the video game localisation industry as well as to gather information about current business practices or the job market that can be contrasted to obtain a multidimensional point of view. In other words, the choice of tools and current business practices—or the deadlines in place—for developers will highly impact localisers' work conditions and their own schedules which, in turn, greatly affects linguistic testers. Therefore, it is paramount to gather current information from as many links in the production chain as possible with the ultimate aim of trying to identify the problems they might encounter, simplify the translation process, and reduce the prevalence of linguistics bugs during the testing phase. Due to the author's background, the first survey was addressed to localisers as it was necessary to further study the other fields before drafting the questions to avoid redundancy or omitting critical information.

The present chapter begins by briefly discussing the initial implementation method of the first survey, the means used to distribute the collector, and how the methodology was modified twice during the first two months until finding the best solution in order to improve the completion rate and increase the number of complete answers. Subsequently, before discussing the survey's design itself, we will introduce the methodology, advantages and shortcomings related to the use of surveys and some related previous studies. Afterwards, we will comment on the different sections and types of questions that were included as well as some of the drawbacks caused by them, such as the number of participants that left the questionnaire halfway through and the potential causes for it. The main body of the chapter will consist of the presentation of the raw results and the data that was collected. The information will be organised following the previously mentioned survey's sections and subdivided by topic; each subsection will include a short introduction with an overview of the content and the motivations behind those questions when necessary.

3.1 Survey's implementation

The definitive version of the first survey was finished during the first week of April 2020 and the link was officially published on the 10th of April of the same year. The initial approach was to distribute said link via email to every available contact in the university related to translation and to all multilingual language vendors that indicated on their website that they provided video game localisation services. Individual posts were also created in Spanish, French, and English in a great number of specialised groups that had numerous members on Facebook and LinkedIn and that were particularly active and continuously posted about game localisation. However, this approach did not yield the expected results and was characterised by a high response rate followed by a high abandon rate. Thus, from the first day until the 13th of May, out of the 170 answers provided by said means, only 79 were complete and most of the participants had stopped after a few minutes into the survey. One of the reasons was the length of the questionnaire, an average of 8 minutes and 23 seconds if we consider all of the respondents and 11 minutes if we only consider the average time of those who reached the final page.

Another factor that played a key role in these results was the fact that the distribution method did not target specifically professional localisers and, as it was not a personal message but a post, many of them would miss it if they did not check their social media regularly. Therefore, with a view to increasing the number of complete answers, the method of distribution was modified in order to target professionals individually via LinkedIn. During the rest of May, all of the employees listed as a translator in the most popular video game development studios or companies were contacted one by one—as well as those in numerous language service providers that worked in the localisation industry. Although this approach provided an extra 196 complete answers out of 295, the most effective method proved to be directly using the research parameters offered by LinkedIn. LinkedIn allows filtering the results by specifying a series of current or past companies as well as keywords, which helps to retrieve a substantial number of profiles and reduces the screening process. This system, less time-consuming than the second one, resulted in 248 complete answers out of 401 for June only and was the definitive method adopted for the rest of the implementation period. The link used as a collector was officially closed on the 3rd of August and the questionnaire received a total of 620 complete answers out of 1000 in just under 4 months (Figure 18).

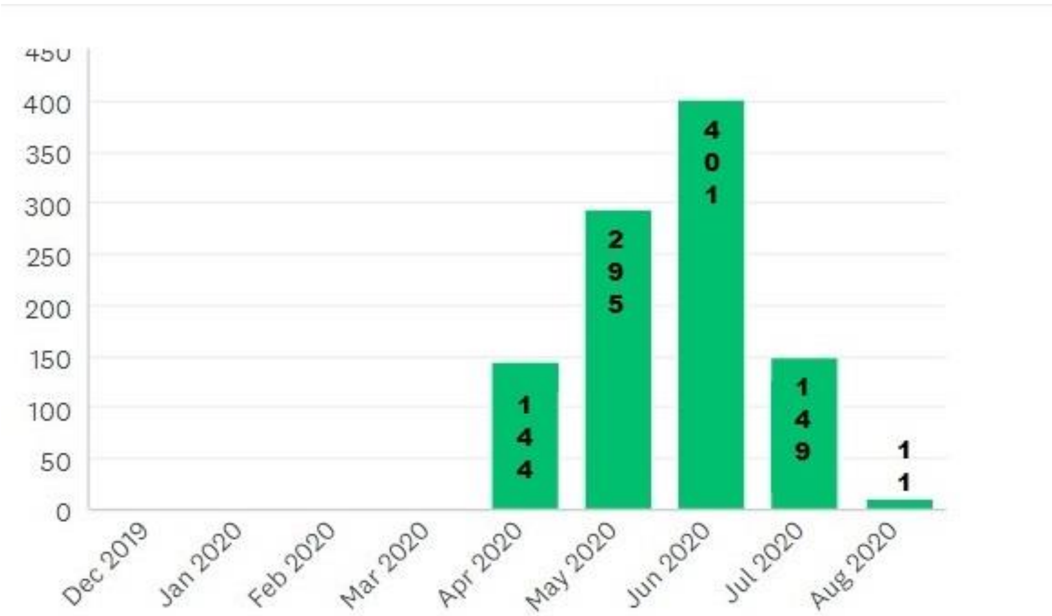


Figure 18. Total of answers per month (localisers)

3.2 Survey's methodology and design

As previously mentioned, the aim of this thesis was to gather a high quantity of information about business practices, tools, and processes directly from the users' point of view due to the lack of previous studies in the different fields in question. The existing techniques used for data collection include direct observation, discussion groups, interviews and surveys; where the latter "represent the most appropriate method for use in studies that aim to collect extensive amounts of information from large populations, where statistical representativeness and the mathematical processing of data are sought" (Kuznik *et al.*, 2010, p. 2). Therefore, the implementation of user surveys was deemed to be the optimal method as this non-experimental technique provides an extensive quantity of empirical data that is already structured and quantifiable with results that can be easily extrapolated to the whole community in question. Although surveys offer many advantages such as their versatility, "simplification of reality, the wide range of possible ways of processing data and the known degree of representativeness" (Kuznik *et al*., 2010, p. 2); there have also several shortcomings. These include the oversimplification and decontextualisation of the collected data, the impossibility of modifying the questionnaire once the survey has been launched, the importance of the representativeness of the sample group, and the degree of implication they require (Kuznik *et al*., 2010). Indeed, as the previous section shows, (Kuznik *et al*., 2010, p. 5):

> Surveys are a costly, time-consuming and laborious method of research. All the operations involved in carrying them out have a very high cost, i.e. designing and collecting data for a reliable database; designing and creating or purchasing databases about the chosen population; and finally maintaining and keeping them up-to-date. Sending out the surveys, receiving the answers, reading and analysing the data, and writing up the final report is also usually a laborious and costly procedure.

Regardless of those disadvantages, the field of video game localisation has a remarkably low number of surveys addressed to localisers and, besides the two surveys mentioned in Chapter 2 section 2.2.1.3, we have not been able to find any paper about questionnaires on linguistic testing or video game development tools (either addressed to professionals or players). The first and only survey addressed to localisers among those mentioned, “Video Game Localization Training on Offer in Spanish Universities at an Undergraduate Level”, was carried out by Olivier Carreira and Eugenia Arrés in 2014, had a total of 7 questions and received 30 answers. The majority of the questions were directly related to training (5 questions) and inquired about the universities attended and the content of the courses. The final two questions gathered information about the respondents’ professional profiles and whether they worked in-house or as freelancers. Conversely, the number of surveys that have been carried out in translation studies in general about translation technologies is much higher and could be categorised into two different groups: surveys about translation technologies in general and surveys about either translation memories or machine translation. Regardless of the group, the studies cover diverse topics such as training, preferences, concerns, the quality of the output, degree of adoption, etc. Due to the number and diversity of papers, it is beyond the scope of this thesis to provide anything beyond an introduction to some of those that will be later on used in Chapter 5 as a counter-point to compare the data collected by this first survey and contrast our results with those that apply to translation in general.

Therefore, among the numerous user surveys about translation technologies, we will begin by mentioning a study on translation tools in general carried out in 2004 and published in 2005 by Fulford and Granell-Zafra which received 391 responses from freelance translators based in the UK. In 2011, Blancafort *et al.* presented at a conference the results of a survey about terminology extraction tools and their integration into MT and CAT tools, the questionnaire received responses from 139 different specialists in translation. Additionally, Torres Domínguez carried out and analysed the results of another survey about different types of translation tools in 2012, said survey received 509 responses. Finally, in 2014, Zaretskaya launched a questionnaire in the framework of her PhD thesis and the EXPERT project that received 738 complete responses. Furthermore, among the user surveys related to either Machine Translation or Translation memories exclusively, we find the survey created by Lagoudaki in 2006 about users’ perspectives on the use of TM technologies or the findings of the QT LaunchPad survey in 2013.

3.2.1 Survey's design

The survey was created using the online service provider SurveyMonkey as it offers an easy, user-friendly, and comprehensive solution for creating, distributing, and managing questionnaires. The web-based tool allows users to design the questions from scratch or to upload the elements in bulk, the content can be divided into different pages and provides the possibility of skipping some of them if necessary. This "page skip option" (Image 18) helps reduce the length of the questionnaire and increases completion rates while improving the accuracy of the answers. By marking all questions as compulsory but allowing respondents to avoid certain parts, none of the questions is ignored or forgotten and, at the same time, they do not need to forcibly answer a question if they do not use that precise tool or method, thus reducing the risk of receiving conflicting or potentially bogus results. The system also provides simple means for analysing data and allows for filtering the results following a wide range of customisable criteria, including specific answers to a specific question. This feature in particular allowed us to analyse, for example, access to reference material and different business practices depending on the respondents' type of employment (Chapter 5 section 5.2).

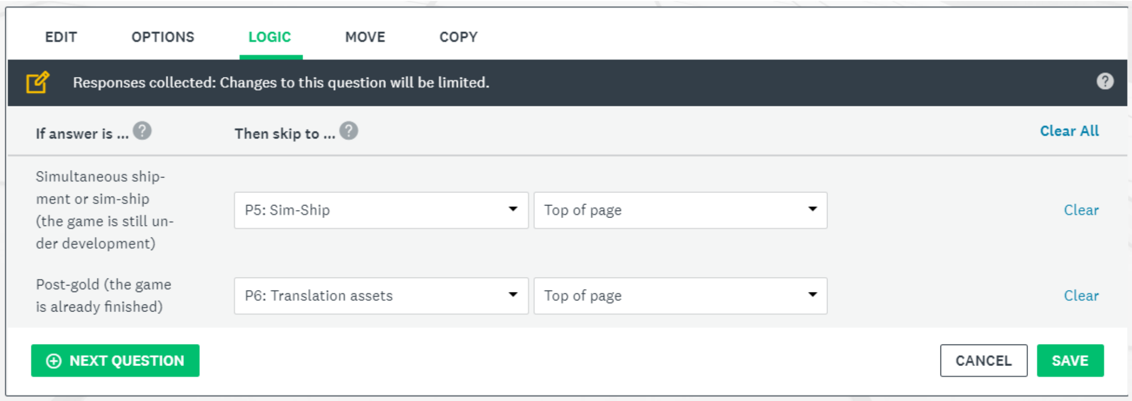


Image 18. Example of the page skip option (localisers)

The survey was divided into 14 different sections that were spread over 20 pages and contained a total of 46 questions. The first page only had a welcoming message with the instructions and the description of the project and, once the participants clicked on the "submit" button at the end of the survey, there was a pre-programmed message thanking them for their help—the latter is not taken into account when counting the pages as it is not a page of the survey *per se*. The following table (Table 11) shows the general purpose of each section and the different topics that were covered in each of them.

Additionally, it is necessary to point out that although some of the topics only required a single question, others needed a series of them which explains the fact that the list contains fewer topics than questions.

SECTION	TOPICS
1. Personal information	Age & nationality
2. Languages	Native language, main source and target language & other languages
3. Professional information	Studies, professional experience, main source of income, workload & type of employment
4. Business practices	Teamwork & release model
5. Assets, access, and linearity	File formats, reference material, video files, visual environment, text linearity & use of controlled language
6. Attitudes	Features and functionalities
7. Asset extraction and integration, content management tools, project management tools	Familiarity & adoption; asset extraction and integration, content management tools; project management tools
8. Testing	Testing tools
9. Tree-based tools	Familiarity
10. Resources	Web resources
11. Corpora	Usage & corpus compilation tools
12. Terminology extraction and management tools	Familiarity, standalone tools, integrated into corpus tools & integrated into CAT tools
13. Computer-assisted translation tools	Familiarity & adoption, CAT tools & localisation tools
14. Machine translation	Familiarity & adoption, commercial MT & free MT systems

Table 11. Sections and topics (localisers)

Before describing the design of the survey, it is also worth noting that the number of pages does not match the number of sections due to the aforementioned "skip option"; some sections are spread over several pages while others share the same one. The initial section comprised two questions to establish the respondents' age and nationality. The first one was presented following a multiple-choice format and had 7 different options that grouped them in different ranges starting with "17 or less" and finishing with "65 or older”. The second question was a drop-down list that included all the nationalities provided in the CSV file that can be downloaded directly from the United Kingdom’s website[34]. In order to allow respondents to select multiple options, as some might be bilingual or trilingual, the question about native languages was in a checkbox format (Image 19). However, since SurveyMonkey does not allow to have more than 200 options in this format, the last box was reserved for other languages and the localisers concerned could write said language down if they did not find it among the given options.

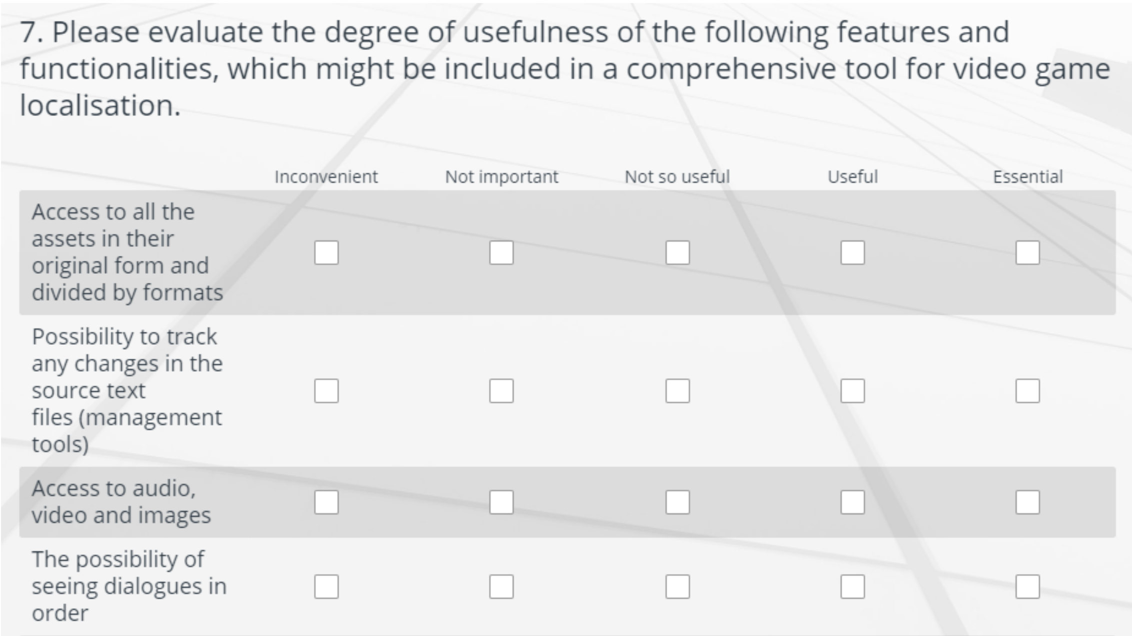
Image 19. Preview of the question about native languages from the user’s perspective

The following questions—those about the main source and target languages—reverted to a dropdown format since it allows to include 500 options and, in this case, only a single answer was required. Finally, the second section included a question about whether the respondents had other working languages; this would either lead to a different page with another checkbox list of languages, or the third section if the response was negative. Sections 3 to 5—both included—only contained multiple-choice questions. They were spread over 3 different pages and there was a specific page with an extra set of two single questions only addressed to those who worked following a sim-ship model.

Section 6 can be found at the end of page 6 and contains one of the questions that were at the heart of the survey. This question in particular was created with the sole purpose of measuring the professionals’ attitudes towards a set of features and functionalities

34
https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/664133/CH_Nationality_List_20171130_v1.csv/preview

that should be integrated into a comprehensive solution for video game localisation. After researching the different options provided by SurveyMonkey, the most viable solution was to use the “Matrix / Rating Scale” format (Image 20) as any other option would have made the questionnaire considerably longer. The main drawback of this format is that, if all the rows are marked as compulsory and respondents must provide an answer to be able to continue, they are automatically allowed to provide multiple answers per row, a fact that must be taken into account when analysing the results as they may not correlate with the number of participants. This particular question included a list of 14 items and the participants had to rate them as “inconvenient”, “not important”, “not so useful”, “useful”, and “essential”.


7. Please evaluate the degree of usefulness of the following features and functionalities, which might be included in a comprehensive tool for video game localisation.

	Inconvenient	Not important	Not so useful	Useful	Essential
Access to all the assets in their original form and divided by formats	☐	☐	☐	☐	☐
Possibility to track any changes in the source text files (management tools)	☐	☐	☐	☐	☐
Access to audio, video and images	☐	☐	☐	☐	☐
The possibility of seeing dialogues in order	☐	☐	☐	☐	☐

Image 20. Example of the matrix format from the respondent's point of view (localisers)

The remaining sections were devoted to analysing the level of familiarity and degree of adoption of a number of technological solutions on the market as well as the specific name of the programme the respondents used. Almost all of them followed the same structure; the introductory question was about whether the participants had heard about the tools, followed by a second one where they were asked if they used them. Those who marked either “yes”, “regularly” or “sometimes” would be directed to a different page with a list of names of programmes and were allowed to select multiple options in each of them. To shorten the list, only the most known tools would appear and the final option would always be “other”, then the respondents would be asked to provide the name of the tool in a comment. Three exceptions were made, one in the case of tree-based tools where the participants were only asked about whether they had heard about them before and if they thought they could be useful, as they are mostly used by developers and are not well-known. The second exception was that the section about resources only contained one question with a list of options (and the possibility to add others that were not provided). Finally, in the section about terminology extraction tools, all questions were compulsory and there was no skip logic—although each of them had the option “none” to avoid false results.

3.3 Results

During the preliminary analysis of the data collected, one of the first steps was to study the behaviour of the respondents in order to find patterns to improve the effectiveness of the questionnaires for the subsequent surveys. Therefore, due to the high number of participants who abandoned the survey halfway through, we decided to look into the matter closely. First, we observed that all 1000 respondents completed the questions about age, nationality, native language, the main source language, and the main target language that were located on page number 2 (Table 12). However, 123 abandoned the survey after reaching the first question that followed the skip system and being redirected to a new page (either page 3 or 4). The remaining 877 answered the questions about studies, experience, type of employment as well as those about teamwork and production model on page 4. Once again, as they reached the end of the page and moved to either the extra set of questions about simultaneous shipment or to the section related to assets, 198 participants stopped answering the survey (126 out of those who chose sim-ship and 72 out of those who chose post-gold).

The numbers remain once again unchanged until the end of page 6, including the long question about the participant’s attitudes presented in a matrix format. However, out of the remaining 679 localisers, 6 gave up and did not answer the questions about asset extraction and integration, content management tools and project management tools located on page 7. This trend continues throughout the rest of the sections and respondents only abandoned when they were redirected to a new page, always answering the questions displayed on the same page before quitting. We can observe 3 more desertions when they moved towards the page about linguistic testing, 4 more when they progressed towards tree-based tools, 7 once they reached the question about web resources and 8 when they were asked about corpora. Out of the 651 who reached page number thirteen, 22 decided to leave and did not answer any of the questions about terminology. Only one participant gave up after page number 15 leaving 628 to respond to the degree of adoption of CAT tools. On the last pages, 6 abandoned the survey when they were redirected towards the section about machine translation and 2 when they were led to the questions about the specific tools used in machine translation, only two questions away from the end of the questionnaire.

PAGE NUMBER	NUMBER OF RESPONSES
PAGE 1: WELCOME MESSAGE	-
Page 2: Personal Information	1000
(Skip question) Page 3: Other languages	453 (skipped 547)
Page 4: Professional information	877
(Skip question) Page 5: Sim-ship	582 (skipped 418)
Page 6: Translation assets	679
Page 7: Asset extraction/integration, etc.	673
(Skip question) Page 8: Tools	162 (skipped 838)
Page 9: Linguistic testing tools	670
(Skip question) Page 10: Tools	406 (skipped 594)
Page 11: Tree-based tools for dialogues	666
Page 12: Web resources	659
Page 13: Corpus compilation tools	651
(Skip question) Page 14: Tools	55 (945)
Page 15: Terminology extraction and management	629
Page 16: CAT Tools	628
(Skip question) Page 17: Tools	577 (skipped 423)
Page 18: Machine Translation (MT)	622
(Skip question) Page 19: Tools	305 (skipped 695)
FINAL MESSAGE	620

Table 12. Survey’s pages and number of total answers per page (localisers)

Among those who decided to stop answering the questions right after the sections that covered the topics about professional information and assets, some sent a private message specifying where they had stopped and explained that they were not allowed to continue with the survey due to confidentiality issues. Others, however, wrote back as well and said that they did not have sufficient knowledge about the specific technology or current business practices. In the case of those who abandoned later in the survey, there were many complaints about the length of the questionnaire and the lack of a progress bar, which might explain the case of the respondents who left the survey right before the last two questions about machine translation tools as well as those who left right before the questions about the last type of tool. Thus, the results presented will only take into account the 620 participants who completed the survey, submitted their answers and reached the acknowledgement message. This decision was taken to avoid inconsistencies when analysing the data as trying to calculate each section according to the number of active respondents would be too complicated and undoubtedly lead to statistical errors. Furthermore, the design of the tables and figures created in order to display the results will vary depending on the content in order to provide the most readable option.

3.3.1 Personal information

This initial and considerably short section only contains two questions as the survey was conceived taking into account the need to avoid references to the name of the company or personal names due to confidentiality reasons and NDAs. One of the main inconveniences of the industry is the high degree of secrecy and the fact that there is almost always a non-disclosure agreement in place. Thus, it was necessary to provide a safe space and ensure the anonymity of the respondents. Therefore, the first question directly covered the participants' age and followed a multiple-choice format that displayed 7 possible answers: 17 or less, 18 to 24, 25 to 34, 35 to 44, 45 to 54, 55 to 64, and 65 or older.

The option "17 or less" was provided due to the well-known and proven impact that fan translation has in the industry—which is also behind the creation of an "18 to 24" group. In addition, since the practice of video game localisation was born about 40 years ago, the category of "65 or older" had to be included for statistical reasons as well. However, as Figure 19 shows, none of the 620 participants was concerned by the first or last options, and less than 1%—5 of them to be more exact—were over 55 years old. Only 28 were in the "45 to 54" category and 61 belonged to the second group of age. Therefore, the most represented group is "25 to 34" with 346 answers (or 55.81%), followed by the participants between 35 and 44 years of age—with 29.03% or 180 localisers.

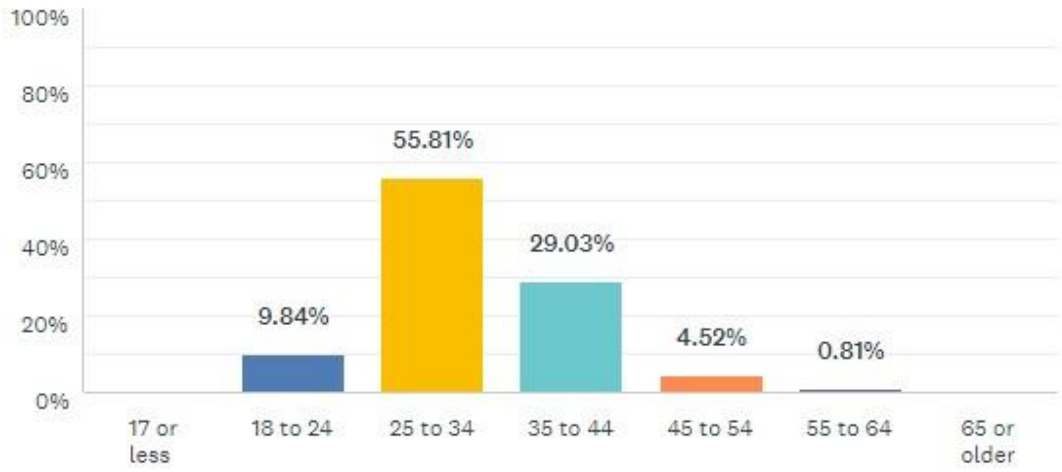


Figure 19. Respondents' age (localisers)

The second question—and last—of the section was created to retrieve information about the respondents' nationalities for statistical reasons as well as to have a reference point for comparing the countries of origin with native languages, source languages and target languages in Chapter 5. The dropdown question listed 226 different nationalities and was created via the "bulk upload" option from a CSV document downloaded from an official source as mentioned before. Among all the countries on the list, only 56 had at least one representative and, out of them, 13 received 10 answers or more. As shown in Table 13, we can observe that Spain, France, and Italy were at the top of the list (Table 13).

Indonesian was the nationality of 8 participants; Taiwanese of 7 respondents; Chinese, Czech, Japanese, and Mexican had 6 representatives each; Colombian and Hungarian were chosen 5 times; Austrian and Chilean appeared 4 times each; Canadian, Danish, Dutch, English, Finnish, Greek, Portuguese, Romanian, and Thai were chosen 3 times each; and Algerian, Australian, Belgian, Bulgarian, Croatian, Hong Konger, Rwandan, Serbian, South Korean, and Venezuelan had 2 respondents each. Finally, the nationalities with only one representative were: Armenian, Belarusian, Ecuadorean, Indian, Malaysian, Moldovan, Norwegian, Peruvian, Senegalese, Singaporean, Slovak, Sri Lankan, Swiss, and Syrian.

NATIONALITIES	RESPONSES
Spanish	113
French	107
Italian	69
Brazilian	46
German	43
Turkish	23
Russian	22
Argentine	20
American	16
Polish	13
British	10
Egyptian	10
Ukrainian	10

Table 13. Nationalities with 10 respondents or more (localisers)

3.3.2 Languages

With a view to create a clearer structure and minimise the number of items per section, all the questions related to languages were considered as a separate section from those that have to do with other professional information. As part of the second section of the survey, the participants were asked a series of questions about their native language, their main source language, and their main target language. As language specialists sometimes have multiple linguistic combinations, they were also asked whether they had other working languages and, for those who did have other language pairs, there was an extra question on a separate page where they could specify them by selecting multiple options. However, this last question only provided a list that did not discriminate between source or target languages.

3.3.2.1 Native languages

In order to allow respondents to select multiple languages, as they might be bilingual or even trilingual, this question was set up following a checkbox format as mentioned before. One of the main inconveniences is that the number of options is limited by default in SurveyMonkey to 200 and some of the languages could not be included. Instead, they could choose the option “other” and specify their native language, which was the case for 19 localisers. In total, if we take into account those provided in the option “other”, there were 45 native languages that had a least one speaker. The most represented native language was Spanish with 158 answers followed by French with 121, Italian with 72, English with 63, and German with 49. There were also a consistent number of Portuguese and Russian native speakers. Table 14 shows the first 10 results, although it is also worth mentioning that the option “other” appeared right after Mandarin as the eleventh choice. If we take a look into the most represented languages among those that were not listed, the most popular one was Galician with 9 responses followed by Cantonese with 3.

NATIVE LANGUAGE	
Spanish	158
French	121
Italian	72
English	63
German	49
Portuguese	47
Russian	36
Turkish	25
Catalan	24
Mandarin	22

Table 14. Top 10 native languages (localisers)

3.3.2.2 Main source languages

In the case of main source languages, only 17 received at least one response and the participants were limited to only one choice as the question is specifically for the main source language solely. Figure 20 includes the languages that were selected more than once and shows that English, with 543 answers, is clearly the most represented one followed by Japanese and Mandarin—that only received 28 and 14 answers, respectively. The other languages that were not included in the graphic as they only had a single representative are: Arabic, Danish, Dutch, Italian, Persian, Polish, Portuguese, and Serbian.

Although most video games are directly developed in English, these numbers are not representative of the industry. In reality, as explained in Chapter 2, many video games are created in other languages, but companies tend to use English as a pivot language as professionals with combinations such as Chinese into French are more difficult to find and have higher rates. As a result of the increasing weight of the video game industry worldwide and its expansion to other countries, this practice is gaining popularity and adds to the complexity of the localisation schedule as all the content has to be previously translated into English before sending the text to the rest of the translators.

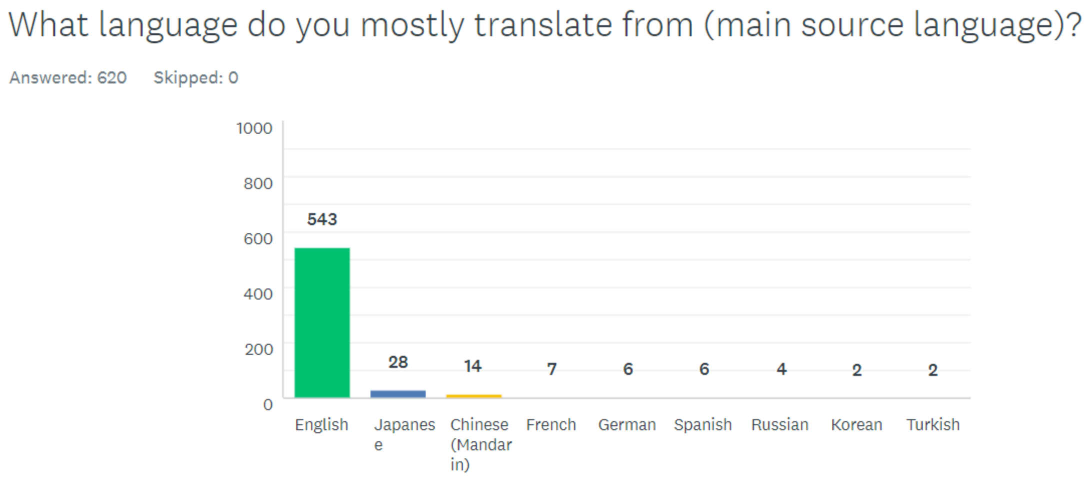


Figure 20. Main source languages (localisers)

3.3.2.3 Main target languages

The following question had the same format as the previous one and, once again, the participants could only provide a single answer as they were asked about their main target language. As expected, the number of languages was higher than in the previous question—albeit lower than the 45 native languages that received an answer in the first place. In this case, 33 languages made the list and the three most represented ones were Spanish with 149 responses, French with 111, and Italian with 68. If we take a look at the languages that appear in the top five, we can clearly observe that the E-FIGS combination still has a big place in the market.

Table 15 shows the comparison between the ten most-represented target languages and the previous results about native languages. Even though there are some minor discrepancies in the number of responses—mostly due to the presence of bilingual or trilingual translators—the eight first languages remain the same in both lists. The appearance of Catalan as a common native language—as well as Galician—is a direct consequence of the high percentage of Spanish respondents. The languages that were mentioned as having native speakers but were not marked as main target languages were: Telugu, Persian, Javanese, Icelandic, Basque, Croatian, Austrian, Galician, Cantonese, Belarusian, Wolof, and Brazilian Portuguese.

NATIVE LANGUAGE			**MAIN TARGET LANGUAGE**	
Spanish	158		Spanish	149
French	121		French	111
Italian	72		Italian	68
English	63		English	52
German	49		German	47
Portuguese	47		Portuguese	46
Russian	36		Russian	33
Turkish	25		Turkish	22
Catalan	24		Mandarin	14
Mandarin	22		Polish	14

Table 15. Comparison between native languages and main target languages (localisers)

3.3.2.4 Multiple language combinations

As previously mentioned in the section about the survey's design, the last question on page 2 was about multiple language combinations. The participants were asked whether they worked with other languages besides those they had already provided. Therefore, a simple multiple-choice question was added with only two options and, depending on their answer, the respondents would be directed towards page 4 if they chose the option “no” or to page 3 if they answered “yes”. This decision was taken in order to avoid inserting a compulsory extra question for those who did not have other languages—and receiving a high number of negative answers—or adding the question and not marking it as compulsory—and risking receiving fewer answers as some might miss it and others might ignore it. As Figure 21 shows, 356 of the respondents did not work with other languages, although the percentage of those who do—42.58%—remains fairly high.

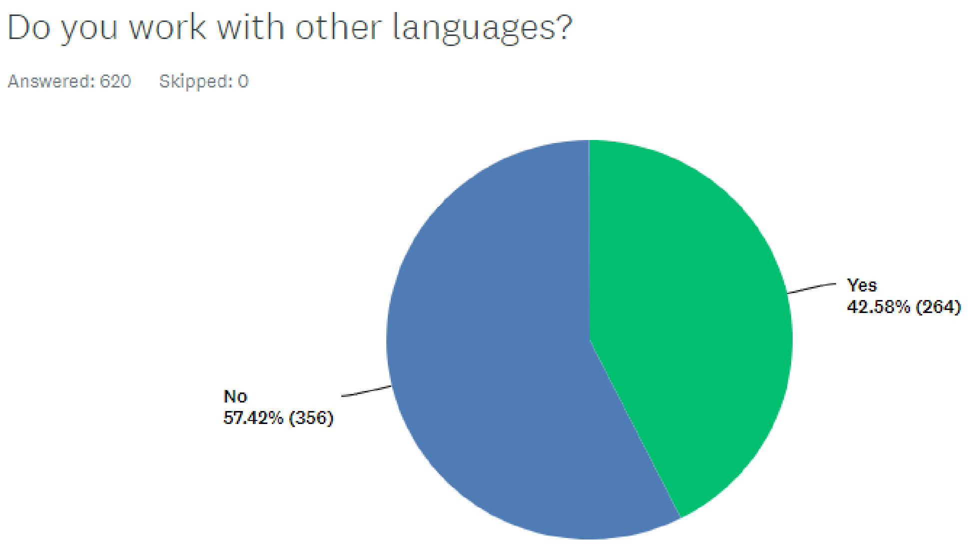


Figure 21. Multiple languages combinations (localisers)

Page 3 contained one single question in checkbox format marked as compulsory that led to page 4 once answered. The format was chosen, once again, to allow multiple answers and there was no distinction between source or target languages as the main idea was to try to reduce the length of the questionnaire as much as possible. Out of the 264 participants who reached this section, the question received 267 answers and 43 languages had a least one response. Table 16 puts together all the languages that received 10 responses or more. Danish received 9 answers; Arabic, Finnish and Thai received 7; Indonesian and Norwegian 6 each; and Vietnamese was chosen 5 times.

LANGUAGE	RESPONSES	LANGUAGE	RESPONSES
French	81	Russian	20
Spanish	70	Portuguese	19
German	58	Korean	15
Japanese	43	Swedish	15
English	36	Polish	13
Italian	30	Dutch	11
Mandarin	24	Turkish	11
Catalan	20	Ukrainian	10

Table 16. Other working languages (localisers)

3.3.3 Professional information

The section about professional information was created to collect data that would be later used in order to evaluate, among other things, the impact of university studies on the degree of adoption of different types of technology or how the type of contract determines access to reference material. Although some of the questions could be also labelled as personal information or as business practices—in the cases of education or type of employment—a different category was created as they directly impact their professional careers, and they remain personal choices to a certain extent. All of the questions are located on page number 4, and they follow a multiple-choice format where all the options are provided. They were only asked to add a comment—although it was optional—to clarify their answer on one single occasion and only if they happened to respond that their main source of income was not related to translation.

3.3.3.1 Educational background

The participants were asked whether they had formal education or training specifically in translation—the question was purposely phrased to avoid including any reference to other types of studies besides translation. Even though in some countries university degrees in language studies do include some courses in their curriculum that provide a certain background in translation in general, they tend to lack specific classes on translation tools and, as previously mentioned, one of the main reasons behind this question was to contrast the results with the degree of adoption of new technologies. Thus, some of the respondents who marked “none” as an answer might have had a university degree in other fields including, but not limited to, languages in general.

The question provided 5 different options: none; master’s degree; bachelor’s degree; PhD; and specialised courses, seminars, workshops, etc. Figure 22 portrays the results sorted in descending order and shows that more than half of the respondents—62.57% or 388 to be more exact—had, indeed, obtained some kind of university degree in translation. Nevertheless, if we take a closer look at those who did not follow an academic path specifically in translation and combine the results of those two options, the percentage is quite high and remarkably close to those who had a master’s degree: 37.42% or 232 participants. In addition, the difference in the results between those with a bachelor's degree and those who selected the option "none" is negligible and only amounts to two responses.

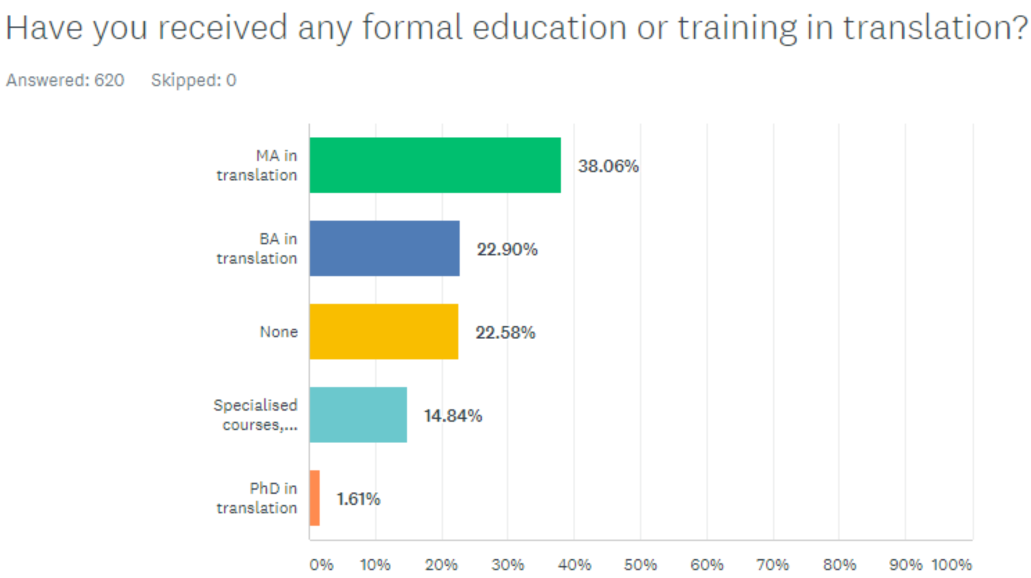


Figure 22. Educational background (localisers)

3.3.3.2 Professional experience

The participants were also asked about how many years of experience they had as a translator. Even though the survey could have included a question about the length of their careers specifically in video game localisation, this might have proven challenging to answer as translators might specialise in different fields or start working in multiple disciplines until choosing a definitive one. Thus, other methods were used as a solution to assess their experience in the game industry in particular and two more questions were added to this subsection in order to clarify. Consequently, the professionals were first asked whether translation was their main employment or primary source of revenue and, secondly, the percentage of video game localisation they usually dealt with in their workload on a regular basis.

The question about their professional experience proposed 5 different options that ranged from "less than 1 year" to "more than 20 years". The numbers related to those two categories are fairly similar, with 35 participants in the first case (5.65%) and 31 in the second case (5%), making them the two less common options. As Figure 23 illustrates, although followed closely by "5 to under 10", the largest group of respondents seemed to have between 1 to under 5 years of experience—211 out of 620. In absolute numbers, this means that the difference between the two groups is only 12 (199 responses for 5 to under 10). In total, 144 translators had more than 10 years of experience—but less than 20. These results point towards the increasing popularity of the field both in terms of development and localisation.

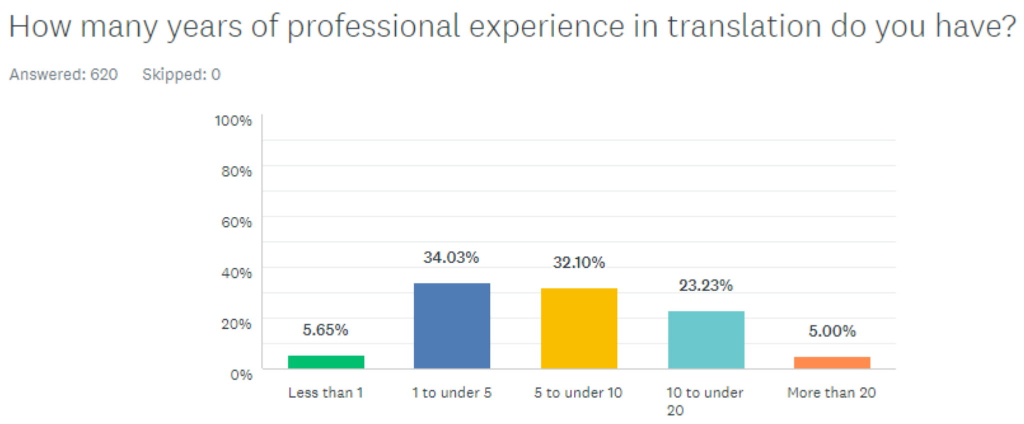


Figure 23. Years of professional experience (localisers)

Translation is a profession characterised by a high number of freelancers, which also tends to mean combining professions at one point or another or translating as a means to supplement the revenue—such as working both in education (teaching languages) and as a translator. In order to determine whether the respondents had solid experience in the field, they were asked if translating was their main type of employment. The question was formulated in a simple “yes” or “no” manner and those who selected the latter were invited to leave a comment to clarify the nature of their main source of income. Additionally, the answers will allow filtering the information collected in other sections to cross-reference it and add to the solidity of future statements if necessary.

The majority of the participants—530 out of 620 or 85.48%—confirmed that their main profession was translation, which only leaves 14.52% working in the field as a secondary source of revenue. Out of the 90 who marked the latter, 76 provided an explanation in the comments section, although in many cases they did not provide actionable data. Among the participants who wrote a clear and complete message, some of them were directly linked to the industry: 10 respondents worked as a project manager either in translation, localisation or testing; and 8 had a position in LQA (linguistic testing). There were also 3 localisation engineers, 6 specialised mostly in different types of writing or copywriting, and 9 who worked teaching languages. Surprisingly, some of them had professions that had nothing to do with languages at all, such as one software engineer, another participant working in diplomacy and even somebody who worked moving furniture.

The third question of this subsection had to do with their workload in video game localisation. Formulated once again in a multiple-choice format, they were asked to indicate how much of their activity was in game localisation. They were provided with 4 options ranging from "less than 25%" to "more than 75%" in order to assess their level of specialisation. The results show that more than half of them—348 or 56.13%—normally dealt with more than 75% of this sort of translation in their workload (Figure 24). This was followed by those who marked "less than 25%" as an answer, which received 145 responses (23.39%).

The initial assumption was that most of those who selected "less than 25%" would be the participants who replied that they had a different primary activity, therefore we decided to analyse the results by only taking them into account. After filtering the data we can observe that out of the 90 participants who had translation as a second activity, 39 indicated dealing with less than 25% of video game localisation in their activity and 37 selected "more than 75%". In other words, almost 27% of the 145 participants in the group who had less work related to video games did not actually work full-time as a translator. This outcome does not validate nor dismiss the initial assumption as the percentage remains relatively low but does procure an interesting angle.

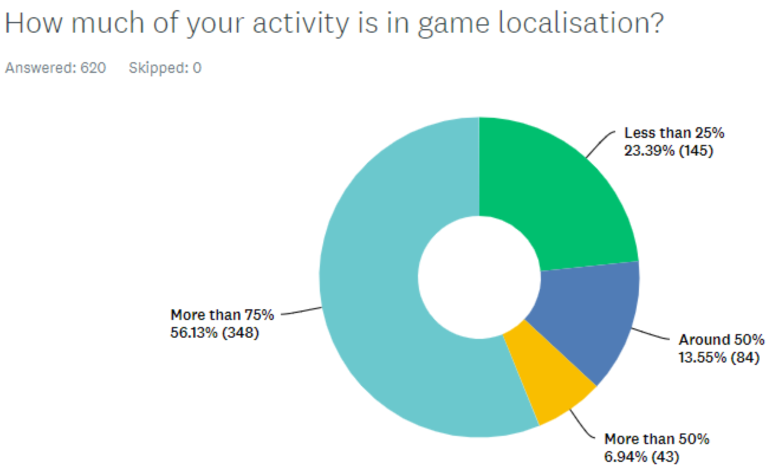


Figure 24. Percentage of activity in game localisation (localisers)

3.3.3.3 Type of employment

The following question about the type of employment provides crucial information in order to analyse many aspects of the industry. First and foremost, it provides a clear view of the reality of the video game localisation employment market as it shows the actual percentage of in-house translators in development companies in comparison to different types of freelancers. Furthermore, using the filtering options provided by SurveyMonkey, the data will provide an insight into how different types of contracts might impact access to resources. Additionally, it could supply revealing information related to professionals' attitudes towards certain tools or functionalities as well as their degree of adoption—especially in the case of machine translation. These last two possibilities will be discussed in depth in Chapter 5 as they require the cross-analysis of multiple sections that have not been discussed yet.

The seven different options that were provided in this particular question included students and fan translators to avoid false results or having to process data left in the comments. In addition, the category "in-house translator in a non-translation company" was created in order to refer to video game development companies as well as studios and platform holders as, essentially, they all work in-house in a company specialised in the game industry. We did not create a different option for each of those possibilities as market trends in general pointed towards a low number of in-house localisers and the percentages could have been too low to be useful. However, the reason behind including a separate option for freelancers working both independently and with an agency—albeit still as a freelancer—was to measure the different degrees of impact agencies have, for example, on the access to reference material. Evidently, independent freelancers, freelancers working solely with agencies, and in-house translators with an in-house contract at a language service provider were included as well.

Table 17 displays the results sorted in descending order and contains both the percentages and the absolute numbers that were obtained. Although not exclusive to the field of video game localisation, it is clear that working as a freelancer is the norm: the combined results of all three categories amount to 70.16% of the responses (435 total) against 24.52% who work in-house in both types of companies. Furthermore, the percentage of localisers working internally in a non-translation company is low compared to the other types of contracts and points to either a low number of job positions available in the market or a high impact of non-disclosure agreements—as some of those who abandoned the survey brought up in private messages. Thus, we decided to analyse the number of respondents who chose that option by reviewing all 1000 answers and found that it was the case for 37 of them. If we turn this number into a percentage taking all categories and all responses into account, the changes are negligible—a mere extra 0.35—although the category does move up one place and takes the central place of the list. Even though these results do not rule out the possibility that some in-house localisers might have abandoned the survey before this point or that they did not even attempt to answer it due to the aforementioned NDAs, it is safe to assume that working in-house in the video game industry as a localiser is not the most common job position in the market.

TYPE OF EMPLOYMENT	RESPONSES	
Independent freelancer	28.06%	174
Freelancer working both independently and with an agency	25.32%	157
Freelancer working with an agency	16.77%	104
In-house translator in a translation company	12.42%	77
In-house translator in a non-translation company	12.10%	75
Student	2.90%	18
Fan translator	2.42%	15

Table 17. Type of employment (localisers)

3.3.4 Business practices

The purpose of this section was to collect data about whether localisers tended to work alone or in teams and the release model that companies usually follow. These two points are of great importance when it comes to managing projects and the impact of terminology inconsistencies—in the case of teamwork—or tracking changes in the source text in the case of simultaneous shipment. This part of the survey was located both on pages 4 and 5 as the second question led to an extra page in case of a positive answer. For those participants who worked following a post-gold model, a negative answer would lead them directly to page 6.

The participants were first asked if they usually worked alone or in teams. They were provided with three different options: mostly alone, mostly in teams, and both. When we analyse the three options separately, 39.84%—a total of 247—of them selected the option "mostly alone", followed by 30.97% (192) that chose "both", and 29.19%—or 181—who answered "mostly in teams". If we consider the combination of the second and third options, 373 of them were accustomed to teamwork on a regular (or semi-regular) basis. These results, although somewhat ambivalent, point to a tendency towards working in teams and the necessity of a solid communication pipeline in place and a proofreading system—or glossaries—to deal with terminology-related issues.

Nonetheless, the responses collected via the question about the production model are more contrasted and show a clear preference towards the simultaneous shipment method. As Figure 25 illustrates, almost 70% of the respondents worked following the aforementioned model, which amounts to a total of 432 localisers who would be redirected to a separate page with two extra questions to collect more details. We can find the highest concentration of participants who stated being students or fan translators among those who worked following the post-gold production model, although unfortunately, their numbers are not representative enough to allow us to analyse the issue any further.

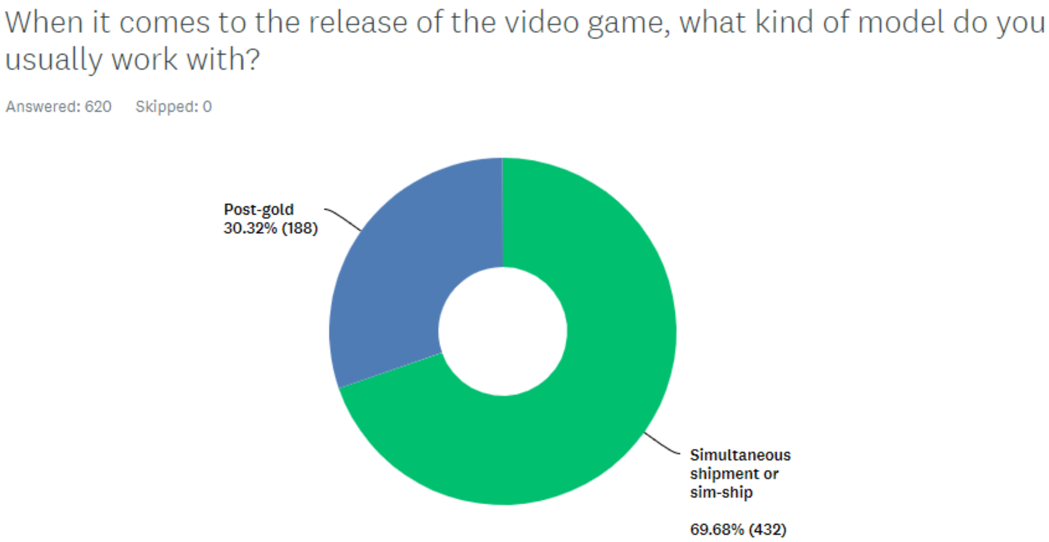


Figure 25. Release model (localisers)

Those who were sent to page 5 had to answer two more questions about changes in the source text in the case of the sim-ship release model. In the first case, they were asked if they constantly received modifications of the source text and, in the second case, whether there was a system in place—to the best of their knowledge—to keep track of those changes. Thus, 80.09% of the 432 respondents who were concerned replied that they did receive constant changes to the documents they were working on (Figure 26) and 60.64% said that there was a system keeping track of them as far as they knew.

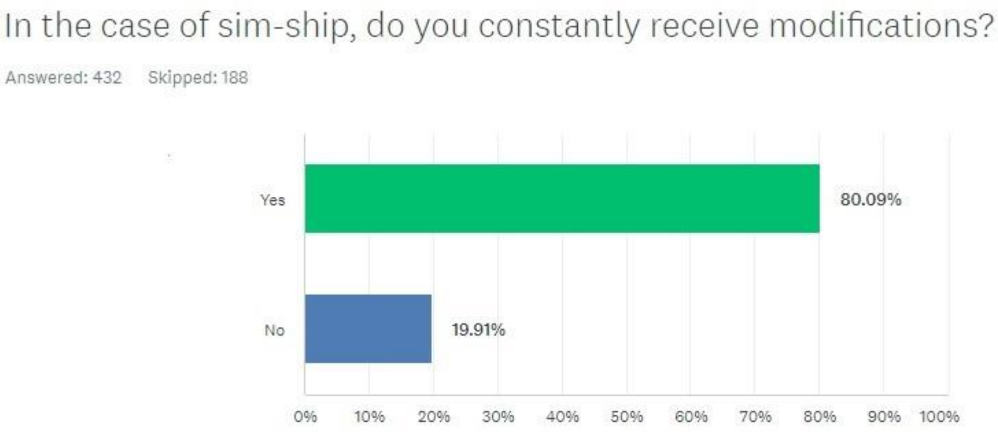


Figure 26. Frequency of changes in the source text (localisers)

3.3.5 Assets, access, and linearity

This section was located on page number 6 and dealt with file formats, access to reference material, access to video files, access to the visual environment, text linearity and the use of controlled language. All of the questions but one were formulated in a multiple-choice format and the first one allowed the participants to add different options—in the case of formats. Two more included the possibility of leaving a comment if they wanted to provide extra information related to the question in particular. The common thread among these questions is the type of context that localisers have at their disposal in order to evaluate to what extent they are forced to deal with what Bernal-Merino calls a "double-blind" process (2013, p. 119).

3.3.5.1 File formats

The first question provided a list of different types of file formats that are common in the translation industry in general. In order to allow multiple answers, as different clients will have in place different working methods, it followed a checkbox format and included a box for "other" where the respondents could add any missing type. Figure 27 shows the results arranged in descending order confirming that the most commonly used document are Excel files with 85.16% of all of the responses—528 occurrences in total. Word was chosen 297 times, XML 238 times and TMX appeared 202 times.

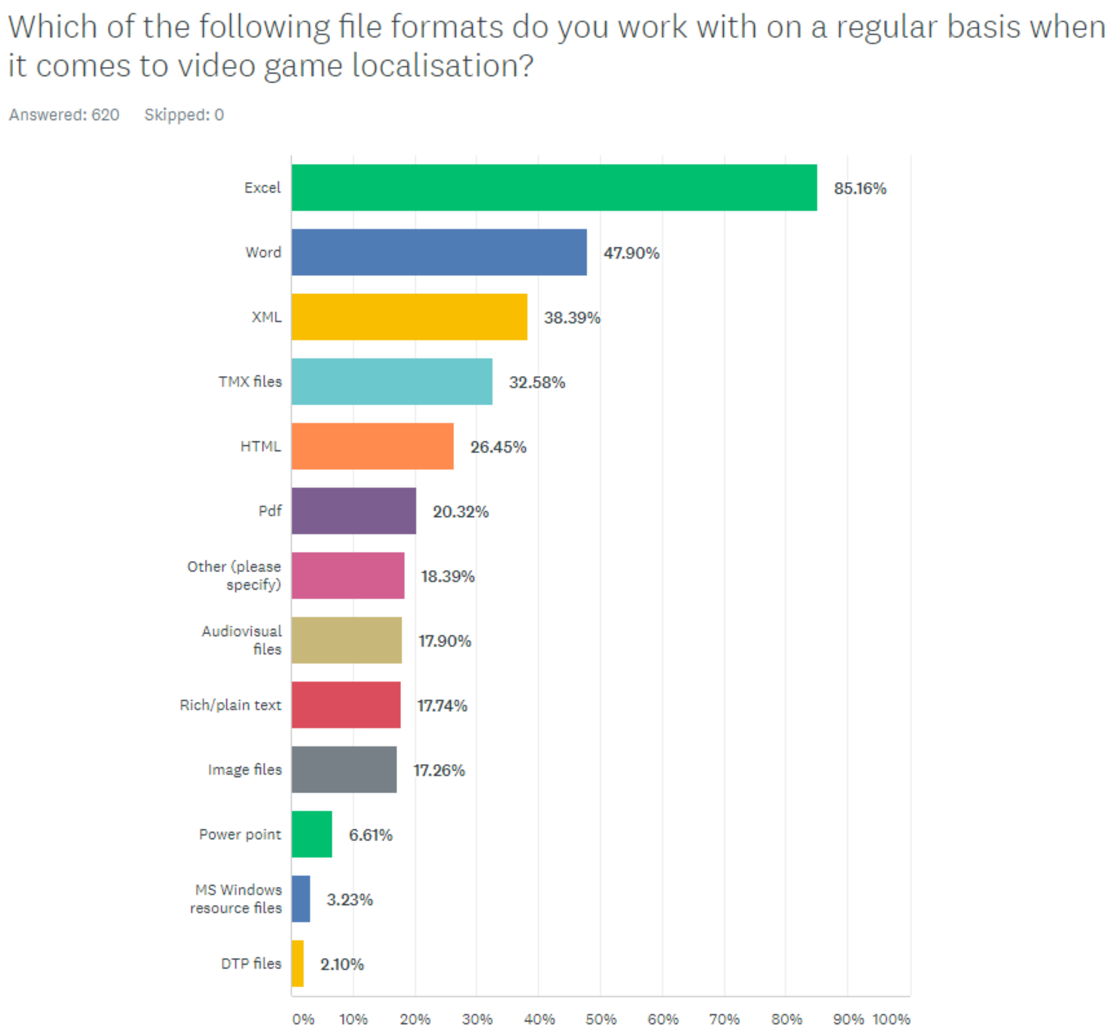


Figure 27. File formats (localisers)

Among the 114 localisers who worked with other types of documents as well, 17 did not disclose the name of the format as it was the result of a proprietary tool that created a proprietary format, and they could not specify more. CAT files in general, once again without specifying, were mentioned 13 times; memoQ either as a server or as a specific file format was mentioned 27 times; Trados appeared 12 times; and there were also references to SmartCat, XTM, Crowdin, Wordbee, and XLOC. Other formats that were specified in the comments without any reference to CAT tools were: .xliff files (13); .po files (11); .json (7); CSV (3); Drive or google docs (4); .srt (2); .slp (2); .properties; .STRING ; .TXT; and java resource files.

3.3.5.2 Access

This subsection comprises three questions about access to reference material, video files and the visual environment itself. In the case of reference material, localisers were simply asked whether they were provided with it or not and, at the end of the question, there was the option to leave a comment. Although 145—23.39%—stated that they did not receive any type of material, the majority of the participants—475 or 76.61%—gave a positive answer. 273 respondents left a comment where, in most cases, they wrote a list of the reference material they usually received. The ones that were mentioned the most in their comments were: Images or screenshots on at least 80 occasions; videos appeared not less than 57 times; glossaries in a minimum of 38 comments; characters' descriptions, bibles or bios were brought up at least 37 times; different types of files with descriptions and game builds (or the game itself) were mentioned at least 35 times each, and both design documents and either past translations or translation memories appeared in at least 15 occasions each. There was also a considerable percentage of localisers who provided a more complete explanation on top of the material they usually received. Table 18 presents some of the unedited comments as an example, they are sorted in chronological order. The complete sample—also in the same order and unedited as well—can be found in Appendix 4. Those provided as an example in this chapter show a stark contrast between localisers who work in-house—in the case of the fifth comment—and those who work as freelancers.

7/30/2020 4:11 PM	Game development files, sometimes a few pictures and some context about the game, but not much and most of the time only the official website.
7/6/2020 8:30 AM	Glossaries, some videos, the occasional beta ROM. For some game (sic), reference files for the variables system. My main client is usually happy to pay for a few hours' (sic) worth of familiarisation, which makes a huge difference.
6/23/2020 11:56 AM	Depends on the project. For some RPGs I've received character descriptions and world/lore descriptions. Sometimes devs also send pictures/screenshots if asked about specific scenes/items. In all this time I've only received a steam key once.

6/11/2020 2:44 PM	Images of characters, objects and some screenshots from the game, characters background information in order to adapt their tone of voice and a few lines about the plot of the game.
6/9/2020 1:11 PM	As inhouse (sic) localization specialist, we mostly have access to everything we need (through in-development-versions of the games), we also provide as much reference material to vendors that translate languages which are not done in-house.
5/22/2020 12:25 PM	Some WIP images sometimes, as well as content bible for some projects. Which is not to say there is never a case of games where we receive no reference material at all and must localize "to the best of our abilities".

Table 18. Example of comments about reference material (localisers)

The participants were also specifically asked about access to videos when they were creating subtitles or translating dialogues that would be dubbed for voice-over later on. Although it might seem logical that this would always be the case for better synchronisation, as explained in Chapter 2, subtitles in video games do not tend to follow the technical and linguistic norms for AVT as used by traditional (post)production houses and television broadcasters and sometimes the time codes are hard-coded into the game and cannot be modified. Having access to videos is also essential in order to solve issues related to ambiguity or knowing the gender of the character that is talking or that of the addressee. In this case, the results are mitigated as 55% (341) of the respondents stated receiving the videos for the aforementioned purposes and 45%—the remaining 279 localisers—selected the option “no”.

The third and last question aimed at collecting data about common practices when it comes to access to the visual environment for interfaces, menus, and text inside images to avoid text overflows or truncations. The participants could choose between “yes” and “no”, and they were also provided with an option about a common solution in the industry: providing the maximum number of characters that can be displayed. While the said solution is not ideal and access to the game itself or the ability to see changes “on the fly” remains the most desirable option, providing a digit should help to reduce the number of overflows or truncations.

If we analyse the results separately, we can observe that the most common result is not having access to the game and only receiving the character limitations with almost 47% of the answers (Table 19). In the second place, we find that 145 participants did not receive any kind of instructions or access and had to work blindly without knowing if their translation would fit in the space available. However, if we combine the results of those who did have access to the visual environment only and those who had both the game (or any other type of visual support) and character limitations at their disposal, the resulting figure rises up to 184 participants or 29.68%. On the other hand, if we combine the results of the localisers working completely outside of any sort of WYSIWYG environment, we confirm that 70.32% of the respondents (a total of 436) do work in near-blind conditions.

ACCESS TO THE VISUAL ENVIRONMENT	**RESPONSES**	
They provide the maximum amount of characters	46.94%	291
No	23.39%	145
Yes	14.84%	92
Both visuals and maximum amount of characters	14.84%	92

Table 19. Access to the visual environment (localisers)

3.3.5.3 Text linearity and controlled language

The lack of text linearity in the files that localisers receive is mostly due to developers' practices as it is faster to organise the strings by area or character for text extraction and integration. The participants were asked to choose between 6 different options ranging from "always" to "never". The results showed that only 37 of the participants marked "always"; 220 chose "most of the time"; and 81 "often". These 3 possible responses—the most positive ones—when put together, amounted to 338 or 54.52% of the total. The second most popular answer, "sometimes", received 192 replies and, when combined with the remaining two options—the most negative ones—represented 45.48% of the cases (Figure 28).

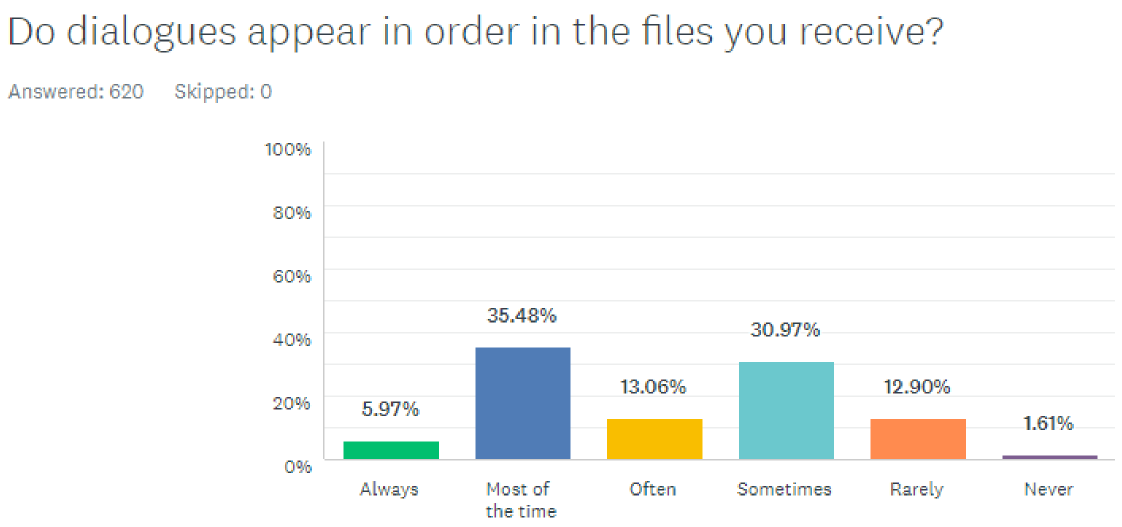


Figure 28. Order of the dialogues

Finally, the respondents were asked if they had to resort to controlled language because of variables and internationalisation issues. The main problem with the last question of this section was the concept of "controlled languages" that, although known in the academic world, is not common in the professional sphere. Therefore, the results might not be as representative as we would have liked since we thought that including the words "variables" would be enough. The respondents were given the same options as in the previous question although, this time, they could provide a comment if they wished to. The results show that "sometimes" received the highest number of answers (179), followed by "most of the time" with 167 responses, and "often" in the third position with 136 (Table 20).

USE OF CONTROLLED LANGUAGE	RESPONSES	
Always	9.19 %	57
Most of the time	26.94 %	167
Often	21.94 %	136
Sometimes	28.87 %	179
Rarely	7.74 %	48
Never	5.32 %	33

Table 20. Use of controlled languages (localisers)

As the option of leaving a message was not compulsory, only 100 out of the 620 respondents wrote one. Among the comments left by the participants, many were about the concept of controlled language, as one localiser put it: “this is the first time I come across the concept of controlled language in my career”. Other comments were about the use of swear words or censorship such as the following one (which remains relevant to a certain extent): “Due to China’s censorship policy, we’re not allowed to use certain sonograms”. Fortunately, there were also numerous comments where they explained how they worked around variables or the impact they had on their work. Table 21 includes four that are particularly interesting, with the last one providing a very complete explanation of the issue.

8/1/2020 10:22 AM	It (sic) exceptional to have declination issues handled. More often there are more and more placeholders for some names or even common words which don't fit to target grammar rules and lower the overall quality.
7/30/2020 7:36 PM	English being mostly gender-neutral, I have to find a way to be as impersonal as possible in French
6/25/2020 1:54 PM	It is often needed to omit the gender either of the subject, object or addressee.
6/24/2020 11:42 AM	It depends on the project and the style guide if available. I don't do "wooden" translations even if I have to usesl (sic) controlled language
6/22/2020 2:58 PM	Most of the time I have access to all the information I need, so I can adjust accordingly
6/12/2020 2:58 PM	Programmers in companies I work with, being mostly monolingual English speakers, often think that whatever works in English will also work in other languages. The more complex a given case, the more disastrous the final effect is: I have come across a complete sentence builder in which every word (or cluster of them) is a variable. This, of course, makes it impossible to be properly localized due to it ignoring a number of factors like nouns used as verbs and adjectives, agreements between adjectives, nouns, verbs etc., register/mood, multiple plural forms, phrasal verbs and collocations not present in the target language, and so on.

Table 21. A small sample of comments about controlled language use (localisers)

3.3.6 Attitudes towards features and functionalities

This section, located at the end of page 6, contains one of the most important and central questions of the survey as its results provide key data in order to evaluate the ergonomics of the tools that are currently on the market. Although the list of options was adapted to the field of video game localisation, the model and inspiration for the question came from the article “User Perspectives on Translation Tools: findings of a User Survey” written by Zaretskaya *et al.* (2018, p. 48). The participants were asked to evaluate a 14 items-long list of different features and functionalities presented to them in a closed preferences question. Additionally, they were offered the possibility of adding other features and functionalities—or anything they could think of—that they thought were missing in the current tools available on the market (Image 21). The comments from that option were, subsequently, divided into different categories and some of the results will be presented in this subsection, the complete list of all the unedited comments can be found in Appendix 5.

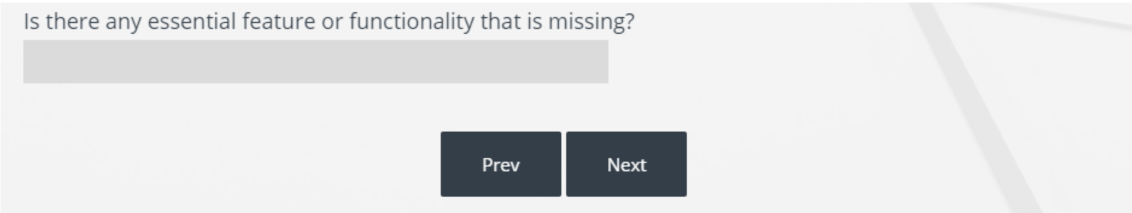


Image 21. Question about missing features or functionalities (localisers)

Therefore, the participants were asked to mark the different options as “inconvenient”, “not important”, “not so useful”, “useful”, and “essential” to measure their attitudes towards them. As previously mentioned in the section about the survey’s design, they had the possibility—by default—of providing multiple answers if they thought it was necessary. The 14 items that were in the list included both tools and features (or a mix of both) and were: access to all the assets in their original form and divided by formats; the possibility to track any changes in the source text files (management tools); access to audio, video and images; the possibility of seeing dialogues in order; terminology management and extraction tools to ensure consistency; being able to use corpora; being able to compile corpora; computer-assisted translation tools; machine translation; the possibility to combine corpora, CAT tools and machine translation; subtitling tools for spoken dialogues and cinematics; quality assurance tools; bug reporting tools; and finally, work on online CAT tools that allow working in teams.

3.3.6.1 Essential features and functionalities

If we analyse the top five features and functionalities that were considered essential without combining the results (Figure 29), we observe that “the possibility of seeing dialogues in order” appeared in the first place with 421 responses; it was followed by “terminology management and extraction tools to ensure consistency” with 403 answers; then “access to audio, video and images” with 314; “to include quality assurance tools” (278); and finally "computer-assisted translation tools” with 273. Furthermore, if we decide to combine the data from the options that were considered both essential and useful, we can obtain a clearer view with a list that is more comprehensive and complete. Thus, “the possibility of seeing dialogues in order” would remain first; “access to audio, video and images” climbs up to the second position; “terminology management and extraction tools to ensure consistency” would go down and become the third option; “possibility to track any changes in the source text files (management tools)” would appear in the fourth position; and finally “access to all the assets in their original form and divided by formats” would also appear fifth.

3.3.6.2 Inconvenient features and functionalities

When we analyse the results on the complete opposite side of the list—only those marked as “inconvenient”—“machine translation” would be at the very bottom of the list as it received 170 responses, followed far behind by the option “the possibility to combine corpora, CAT tools and machine translation” with 34 answers. In the third place from the bottom, we find “Online CAT tools that allow working in teams” with just 13 responses, “computer-assisted translation tools” is in the fourth place with 9 answers and finally “being able to compile corpora” with 6. One result that catches the eye is the fact that the fifth feature considered essential is also the fourth less liked one: computer-assisted tools. The common thread among these functionalities is the fact that they all refer to either machine translation, corpora, or CAT tools. Still on this opposite side of the list—now adding up the results for “inconvenient”, “not important” and “not so useful”—we find that “machine translation” remains the least liked feature, it is still followed by “the possibility to combine corpora, CAT tools and machine translation”; “being able to compile corpora” goes from the fifth position to number three; “being able to use corpora" appears in the fourth position, and finally we find "subtitling tools for spoken dialogues and cinematics" as the fifth option.

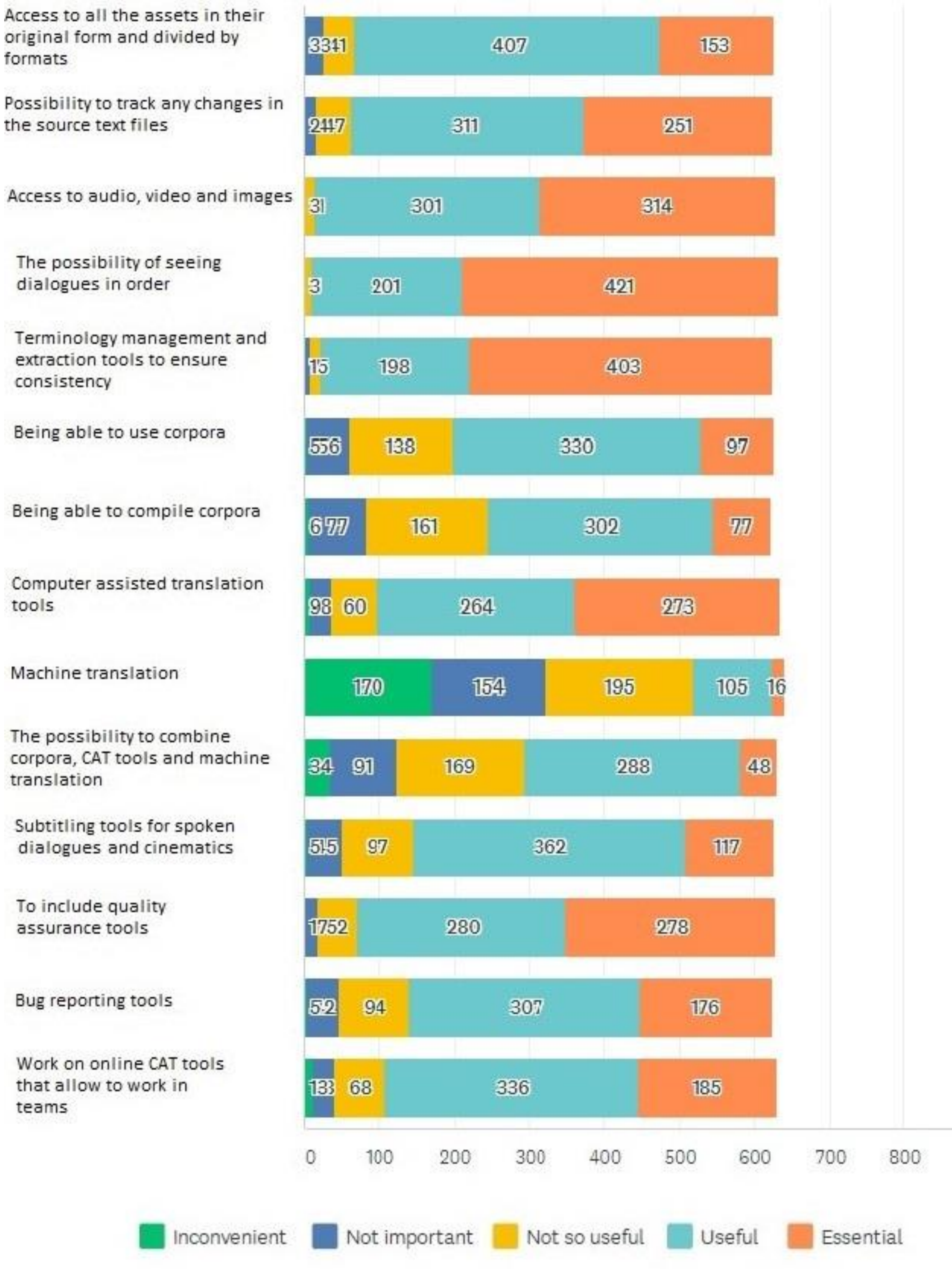


Figure 29. Features and functionalities (localisers)

3.3.6.3 Missing features and functionalities

As previously mentioned, the question also included a comments box in order to provide the participants with the opportunity to add any feature, functionality, tool or combination that had not been included among the 14 proposed options. The text box, as it could not be made compulsory, did not require an answer in order to move on and only 99 comments were left in total. Furthermore, among the few comments left, many participants simply wrote "no", "I don't think so", or that they could not think about anything at the time. Once those comments—along with others that did not provide any actionable information—were put aside, the remainder were classified manually into 6 different categories: quality assurance, communication, preview mode or access to the game, more visual access, character limitation, and miscellaneous tools (Table 22). These categories were created according to the most common topics addressed in the comments if the thread in question appeared more than 5 times, if not they were classified as "miscellaneous". Additionally, whenever one of the responses had elements that belonged to two categories or more, it was copied and pasted into all of the pertinent columns.

QUALITY ASSURANCE	COMMUNICATION	PREVIEW MODE OR ACCESS TO THE GAME	MORE VISUAL ACCESS	CHARACTER LIMITATION	MISCELLANEOUS TOOLS
Real time grammar engines to deal with variables. 7/22/2020 6:29 PM	When sending queries to the team, having them display next to the relevant string can be useful. 7/27/2020 9:09 PM	Previsualisation tools to see how the translation looks ingame (in the case where a dev build is provided) 7/21/2020 11:40 AM	A way to see what variables stand for. / Being able to see how a segment has been translated in another language. 7/30/2020 4:02 PM	An easy way of setting character limits for multiple strings at once, both for PM and the translators/proofreader. 6/21/2020 7:10 PM	Filtering with Regex, creating views a la MemoQ, comments 6/24/2020 12:11 PM
Spellcheck and if possible, but not essential, a platform for communication with members of the team. 5/21/2020 11:44 PM	An easy way to ask questions about specific strings (for example, working on an online platform with a comment feature in each segment) 6/24/2020 11:49 AM	Being able to see where the translated text belongs in a game 6/26/2020 9:38 AM	Feature displaying the context often added by developpers in Excel files (for example memoQ's View pane) 6/29/2020 10:49 AM	Character count tool 6/4/2020 4:10 PM	Friendliness to dictation (correct capitalization, spacing), for example to Mac dictation. 7/30/2020 9:25 PM

Table 22. Example of comments about missing features (localisers)

Evidently, as a result of the methodology used to classify the comments, the category that had the highest number of elements (23 total) was "miscellaneous" and some of the topics included in the said section were related to tag management, controlling tags or the possibility to tagging (4 times); better search capacities (3); modifiable translation memories (2); and the possibility of working with pivot languages and a way of tracking changes in those (2). The category that had to do with better means of communication also had 23 elements and included solutions ranging from "Q&A forms" to comments sections attached to the string in question for better comprehension. “More visual access” comes third with 12 references that vary from being able to export and view the column that some developers add with comments from Excel into memoQ, the possibility of screenshots and being able to see the translations provided in other languages or more context in general. Having a preview option in-game that shows the context, as some CAT tools are able to do for websites or utility software, or access to the game itself was mentioned 10 times, then we find different comments about quality assurance solutions such as spellcheckers or grammar engines (9 references), and finally, a feature that allows an automatic character count to help with space constraints in the case of user interfaces, menus or inventories (mentioned 5 different times).

3.3.7 Asset extraction and integration tools, content management tools, and project management (PM) tools

This section is the first of 8 dedicated to the degree of adoption of tools and resources. The first question is located on page 7 and the extra set of questions (following the aforementioned “skip logic format”) for the participants who chose the options that imply either regular or occasional use could be found on page 8. Therefore, localisers were simply asked if they were familiar with asset extraction and integration tools, content management tools, and project management (PM) tools. They could choose between four options: “Use regularly”, “use sometimes”, “have heard of, but do not use”, and “never heard of”. The results showed that 413 (66.61%) of the participants had heard of them but did not use them; this was followed by those who used them sometimes (13.23% or 82 in total); then we find “use regularly” with 10.32% (64), and finally “never heard of” with 9.84% of the answers (61).

Those who marked either “regularly” or “sometimes” were directed to an extra page with two different questions created in order to inquire about the specific programmes the respondents might use. Consequently, the first question was about asset extraction and integration tools and content management tools and offered 6 different options including “other” if they could not find their tool and “none” in case their previous positive answer was related to PM tools. Figure 30 illustrates the results obtained for the first question and demonstrates that most of the labour is still carried out manually using

Excel or Word and visual basic macros—with a total of 65.10% or 97 answers. XLOC received only 33 responses, the same that those who selected "other".

When we analyse the comments left by the participants who chose "other", which would be in a shared second position, 15 of the respondents could not provide the name of the system due to non-disclosure agreements or simply did not provide a specific name. The rest of them specified anything ranging from a single name to multiple names in their comments. The programmes that received multiple references were: memoQ (mentioned 8 different times); Trados appeared 4 times; and Sisulizer was mentioned twice. The rest of the programmes were: Crowdin; Heartsome TMX Editor; IBM Aspera; Memsource; Python scripts; Ubisoft proprietary tool called Oasis; Unreal Engine 4; XTM; and LAMS.

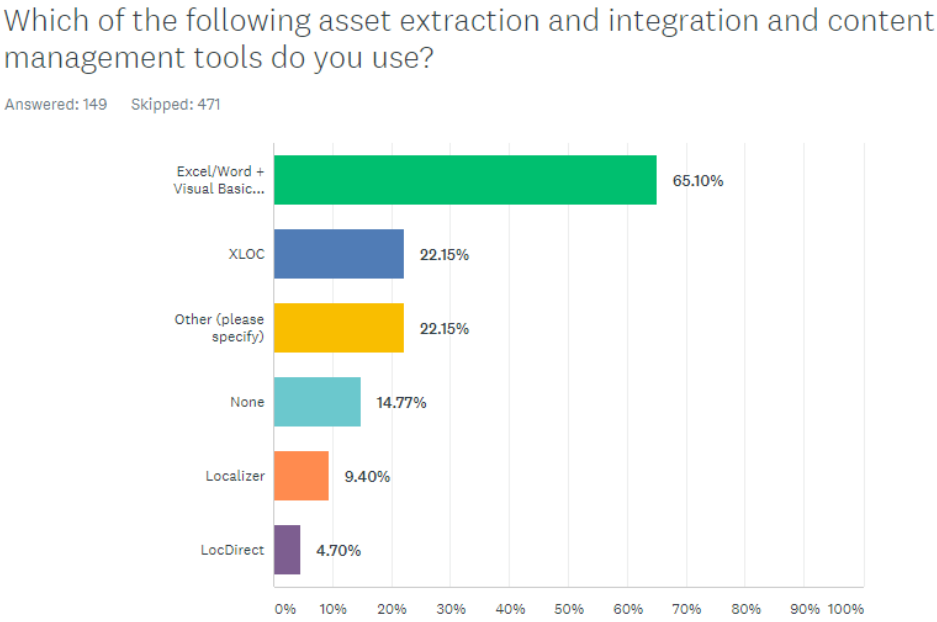


Figure 30 Asset extraction and integration tools and content management tools (localisers)

The second question in this "extra page" was about project management tools, it provided the names of 14 different systems, the 15th option was "none" and the 16th was "other" so they could specify the name of any missing tool they used. Table 23 includes the programmes that received more than 10 answers and shows Plunet and Microsoft leading the ranking. Among the responses for the 37 participants who marked "other", we find: Confidential or not specified (13); Jira (5); memoQ (5); Memsource (3);

Crowdin (2); Trados (2); Trello (2); 1c ERP; BaccS; BPS; excel-based; Favro; Lokalise; Project director; Protemos; Redmine; Testrail; Smartcat; Wordbee; and WORDUX.

PM SYSTEMS	RESPONSES	
Plunet	30.20%	45
Microsoft	25.50%	38
Other please specify	24.83%	37
None	22.8%	34
XLOC	18.12%	27
XTRF	15.44%	23
Transifex	13.42%	20

Table 23. Project management tools (localisers)

3.3.8 Linguistic testing tools

Although the survey was conceived for video game localisers, we decided to include a question about linguistic testing in order to evaluate the percentage of those who wore more than one hat and also participated in these processes. Therefore, the respondents were first asked whether they also worked as linguistic testers using a multiple-choice question with three options—yes, no, and sometimes—that either sent the participants to a question about tools or to the page with questions about tree-based tools. As Figure 31 shows, a consistent number of the participants did actually work as linguistic testers either regularly or occasionally, bringing the total to 378 (almost 62%).

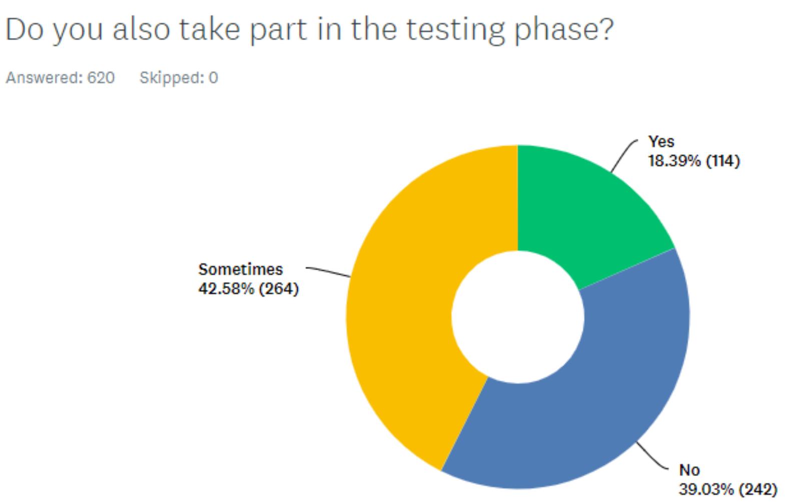


Figure 31. Incidence of linguistic testing (localisers)

Among those who were directed to the specific question about testing tools, 207 selected Jira as the most commonly used programme (Table 24), 109 said that they did not use any kind of tool and 67 chose the option “other”. The methods or programmes mentioned in the comments from the section “other” that appeared more than once were: Confidential or not specified (25); Excel, spreadsheet or google docs (18); Redmine (11); Azure (2), and e-mails (2).

BUG TESTING TOOLS	RESPONSES	
Jira	54.62%	207
None	28.76%	109
Other (please specify)	17.68%	67
Mantis	12.66%	48
Bugzilla	9.50%	36
DevTrack	8.71%	33
Bugtracker	7.92%	30

Table 24. Bug testing tools (localisers)

3.3.9 Tree-based tools for dialogues

The section about tree-based tools for dialogues only contained two questions and was located on page 11. In this case, instead of choosing from a list of different tools available in the market, the participants were only asked whether they knew about this type of tool and if they thought it could be a useful addition to the ones they already used. The reason behind this was that tree-based tools are mostly used during the video game development phase and are usually integrated into the game engine—as explained in Chapter 1 section 1.3.2. Normally, those used in-house are either proprietary software or developed ad-hoc, although there are some options available for the general public.

As expected, the number of regular users was very low and represented less than 2% of the respondents—1.94% or 12 in total—and those who sometimes used this kind of technology were also rare (12.42% or 77). Furthermore, the difference between the respondents who had heard of tree-based tools but did not use them and those who had never heard of them was minimal—42.58% or 264 in the first case and 43.06% or 267 in the second place (Figure 32). Besides the limited availability, we must also consider the fact that these tools—as explained in Chapter 2—are mostly used in the case of "branch dialogues", a common occurrence in RPGs and graphic novels, which are not always the most popular genre.

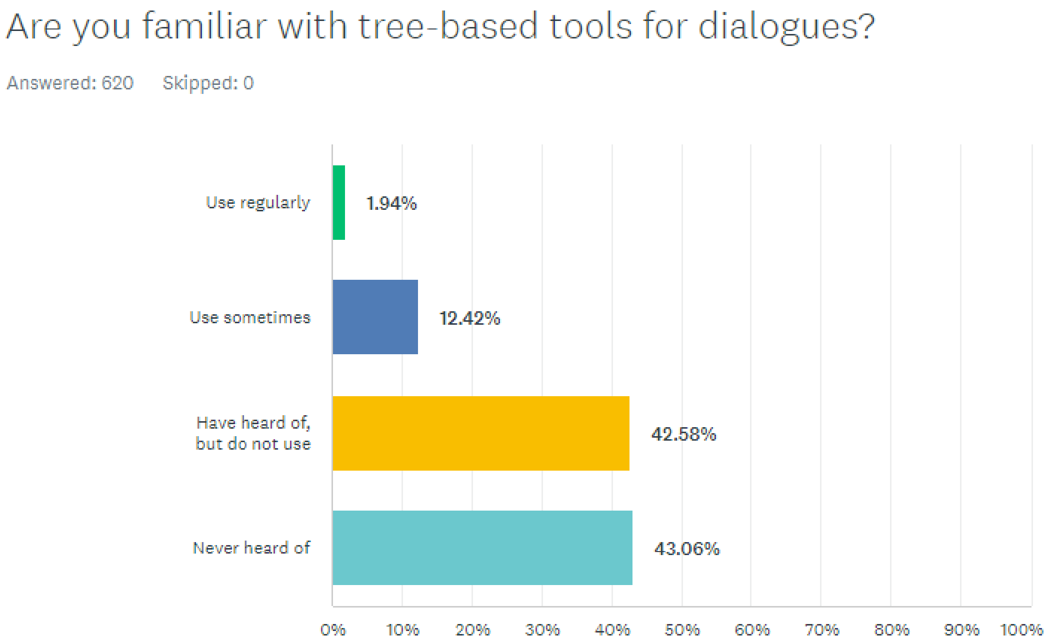


Figure 32. Familiarity with tree-based tools for dialogues (localisers)

The second question of this section aimed at evaluating the perceived value and attitude towards this type of tool and the respondents were provided with the possibility of adding a comment to clarify their answer if they wished to. The question simply read "Do you think you could benefit from them?" and provided two options, "yes" or "no". The results show that 490 localisers (79.03%) selected "yes" and 130 chose "no". The possibility of using the comments box was not subjected to a specific answer and, among all the participants, 144 decided to leave a comment. However, many were simply to state that they could not truthfully provide an answer because they did not know the tool in question. Others added something along the lines of "the more data and tools we have the better". The following table (Table 25) puts together some of the messages that were left by the participants (unedited). The sample includes examples of the previous comments as well as 4 messages left by users who had actual experience with this type of tool and explained how it helped or hindered their performance.

7/30/2020 7:17 PM	I used only internal tools provided by game studios, so I'm not familiar with any public ones (if any). But for my perspective, it helped a lot, especially when it came to player choice-driven dialogs, because the structure is clear.
7/22/2020 10:46 PM	I have had accessed (sic) to (non interactible) (sic) dialogue trees for a recent project, and they were very useful. I can only wish we had those for every project.
7/22/2020 4:35 PM	I can decide when I try.
7/5/2020 9:38 AM	Any tool that helps the translator could be beneficial.
6/23/2020 7:39 PM	I've used this once in a previous project, and it helped a lot in that case, specially because of that particular game mechanic (sic) (different replies triggered different dialogues)
6/2/2020 7:35 PM	Could be very useful for e.g. translation (sic) very text-heavy and especially dialogue-heavy (roleplaying) games, where the course of the dialogue may vary a lot depending on the dialogue options chosen by the player.

Table 25. Comments about tree-based tools (localisers)

3.3.10 Resources

The question about the resources the participants tended to use on a regular basis was once again created following the checkbox format to allow the respondents to provide multiple answers. It was also conceived to include as many options as possible to reduce the number of people choosing the option “other” in order to decrease the quantity of data to process manually. The decision to increase the number of options proved to be a success since out of all the participants only 37 localisers found that the list was missing a resource. As Table 26 illustrates, the 620 respondents provided almost 4000 answers in total (due to the possibility of multiple answers)—3985 to be more exact. The results placed bilingual dictionaries first with 591 responses, followed by monolingual dictionaries (486) and Wikipedia, which reached the third place (482 answers). The table also contains all the resources that were mentioned in the section “other” grouped by categories and with the number of occurrences written next to them.

RESOURCES	RESPONSES		RESOURCES	RESPONSES	
Bilingual dictionaries	95.32%	591	Consultation with experts	33.71%	209
Monolingual dictionaries	78.39%	486	Comparable corpora	24.52%	152
Wikipedia	77.74%	482	Termbank portals	20.00%	124
Terminology databases	73.23%	454	Parallel corpora	17.90%	111
Web forums	63.23%	392	Bilingual visual dictionaries	11.61%	72
Image search engines	53.55%	332	Monolingual visual dictionaries	9.19%	57
Thesauri	42.90%	266	Other (please specify):	5.48%	34
Specialised search and metasearch engines	35.97%	223	Wikis (8); Synonym and collocation tools (6); Google (5); Other games (4); Official websites (3); YouTube (3); Other dictionaries (3); Fan websites (2); Antidote (2); Translation memories (2); Proz (2); Grammar sites (2); Reverso; Linguee; and Microsoft terminology tool.		

Table 26. Resources (localisers)

3.3.11 Corpus compilation tools

In order to contrast the familiarity with corpora and the usage of the tools for corpus compilation, the participants were first asked what were the different types of corpora they usually consulted when facing a localisation project. They could choose more than one option if they needed to and the list of answers included all the possible types of corpora available: bilingual parallel corpora, multilingual parallel corpora, monolingual comparable corpora, bilingual comparable corpora, multilingual comparable corpora, and "none". As Figure 33 shows, bilingual parallel corpora were the most consulted type with 284 responses (45.81%) directly followed by "none" with 235 (37.90%), and bilingual comparable corpora in third place with 139 answers (22.42%).

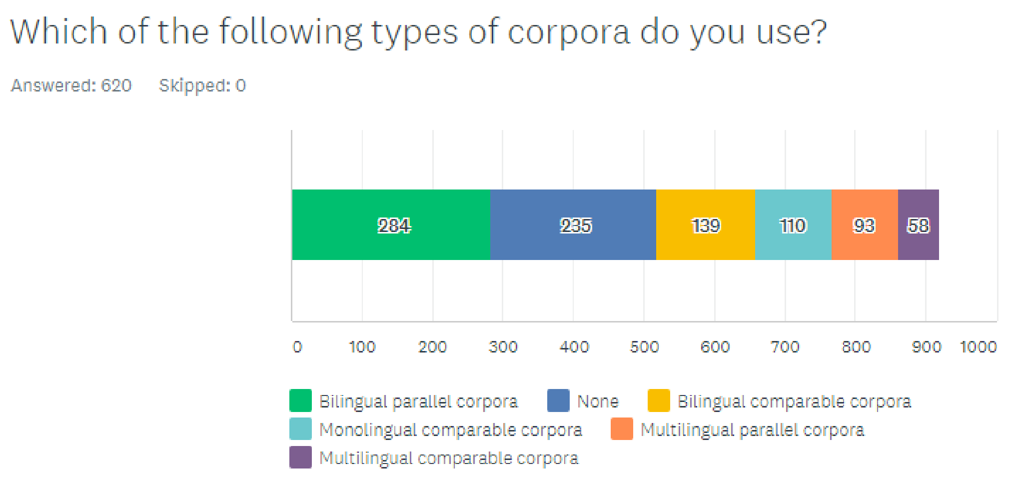


Figure 33. Corpora usage (localisers)

However, when the respondents were asked if they compiled their own corpora, the vast majority stated that they did not—91.61% or 568 in total—and skipped the question about specific tools. The participants that did create them (Figure 34), were asked about the tools they used and, among their answers, the first option was "other" with 45.28% responses, SketchEngine was second (28.30%), and AntCorGen appeared in the third position (22.64%) followed by BootCat (20.75%) and CLaRK (5.66%). The tools—or explanations—included in the comments along with the number of occurrences were: memoQ (9), confidential or not specified (5), integrated into CAT tool (3), AntConc (2), Built-in memoQ LiveDocs (2), Translation memories, and Python.

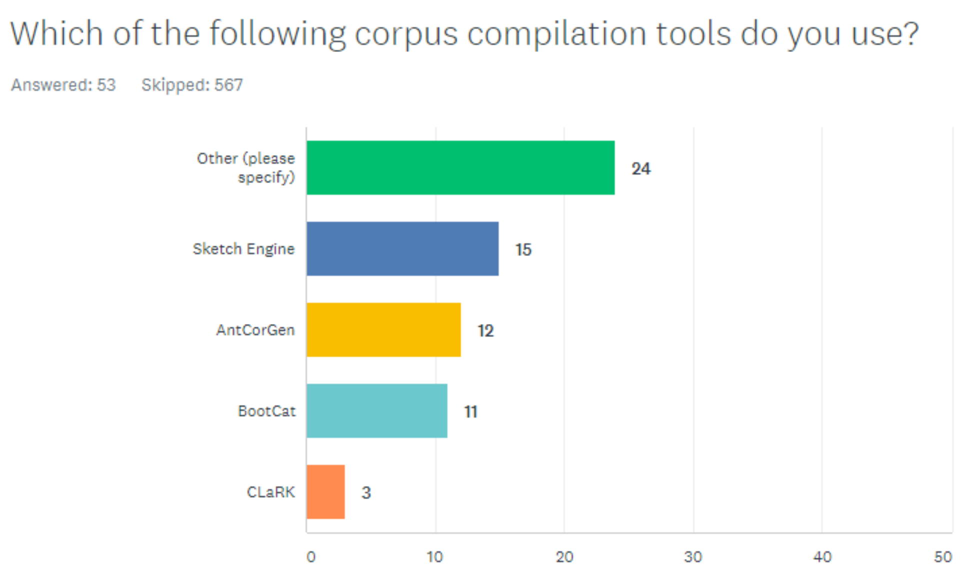


Figure 34. Corpus compilation tools (localisers)

3.3.12 Terminology extraction and management tools

Terminology extraction and management tools—similarly to tools for compiling corpora—usually tend to be among those with a lower degree of adoption. This is mainly due to the fact that translators normally prefer the features integrated into computer-assisted tools and favour comprehensive systems. This section had four different questions and all of them were compulsory with a view to increasing the number of participants answering them and making the figures more representative. Once more, the first question aimed at analysing the degree of familiarity and the degree of adoption of these systems in general. Figure 35 shows that even though the majority of the respondents had heard about them, they did not use these tools (64.35% or 399 in total), 80 participants had never heard of them; 39 localisers (6.29%) reported using them regularly and 16.45% stated that they only used them sometimes.

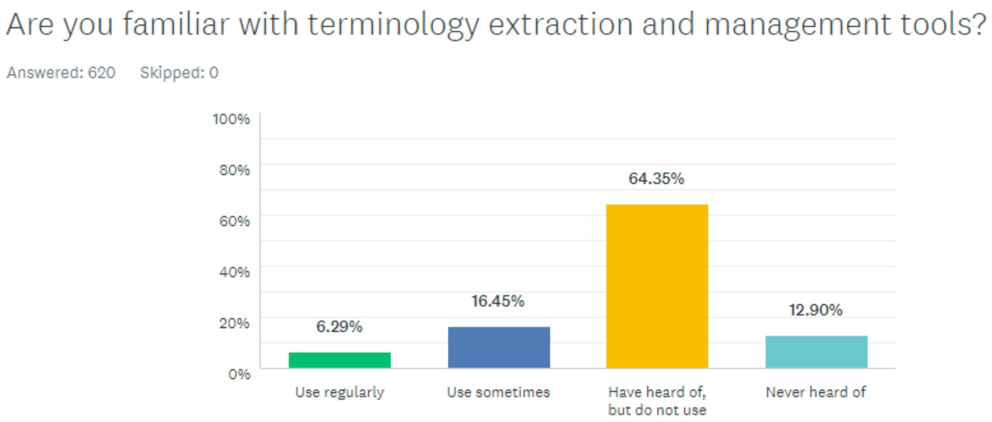


Figure 35. Familiarity with tools for terminology extraction and management (localisers)

The remainder of the questions were subsequently divided into types of tools differentiating between standalone systems, programmes that were designed specifically as a means for extracting and managing terminology from corpora or finally, those that had been integrated into CAT tools. Unsurprisingly, more than 86% (538 participants to be more exact) did not use standalone systems, the second most popular option was "other" with 37 answers, LogiTerm appeared third with 24, then Term-Web (22), and Termologic (13). The complete list of programmes or comments in the section "other" sorted by frequency were: memoQ (12), confidential or not specified (6), integrated into CAT tool (3), SDL Multiterm or Trados (4), Acrolinx, Crowdin, Memsource, Sketch Engine, SynchroTerm, Term Morphology Editor, TemExtract, Excel, and Xbench.

In the case of terminology extraction and management tools only used for corpora (either standalone or integrated into tools), more than 94% of the participants chose the option “none” as an answer—as expected considering the results for corpus compilation tools. The second most common option was, once more, “other” with a mere 2.74% of the responses, followed by Intragloss (1.61%), Lingvo.Pro (1.45%) and OneClick Terms 1.29% (provided by SketchEngine). The content of the section “other”, very similar to that of the preceding question, included: memoQ (7), confidential or not specified (5), integrated into CAT tool, and Crowdin.

The final question of this section inquired about the solutions that had been integrated directly into CAT tools. One of the main issues was the fact that many of the respondents did not know that qTerm™ is the name of the system that manages terminology in memoQ, thus they chose “other” instead to provide the desired answers. Therefore, Table 27 puts together all the results and includes those from the comments as well, combining the results from the participants who marked qTerm™ in the first place and those who wrote memoQ in the section “other” for more clarity.

RESPONSES	OCCURRENCES
None	393
SDL MultiTerm Extract	171
qTerm™ (memoQ)	50
QuickTerm	24
crossTerm	7
Memsource	3
Crowdin	1
Wordfast	1
Lokalise	1
DejaVu	1
Xbench	1
Under NDA or not specified	9

Table 27. Programmes integrated into CAT tools (localisers)

3.3.13 Computer-assisted translation tools

According to previous studies in the field of translation in general (Zaretskaya *et al.*, 2018, p. 47), CAT tools are the technology that tends to be the most widely adopted. The survey's section about these systems had three different questions, the first one was included to analyse the degree of familiarity and use of these programmes and the rest were about the tools themselves. The answers to the first question show (Figure 36) that 81.45% of the participants—505 to be more precise—used them on a regular basis and 11.13% (or 69) made use of them sometimes. These numbers, when combined, amount to a total of 92.58% of the 620 respondents, in other words, 574 users.

On the other hand, 3 participants (0.48%) had never heard of CAT tools despite their widespread use. If we analyse their background and filter all the results to only show their responses, we observe that they came from either the US, Brazil, or France. None of them had any kind of educational background in translation; two of them had "1 to 5" years of professional experience and one had "5 to under 10" years of experience. Translation was the main source of income for two of them and, in the case of the third participant, IT engineering was his or her main activity. Two of them had a workload of "more than 75%" in game localisation and the remainder had "less than 25%". Finally, we find two freelancers and one who worked in-house in a non-translation company.

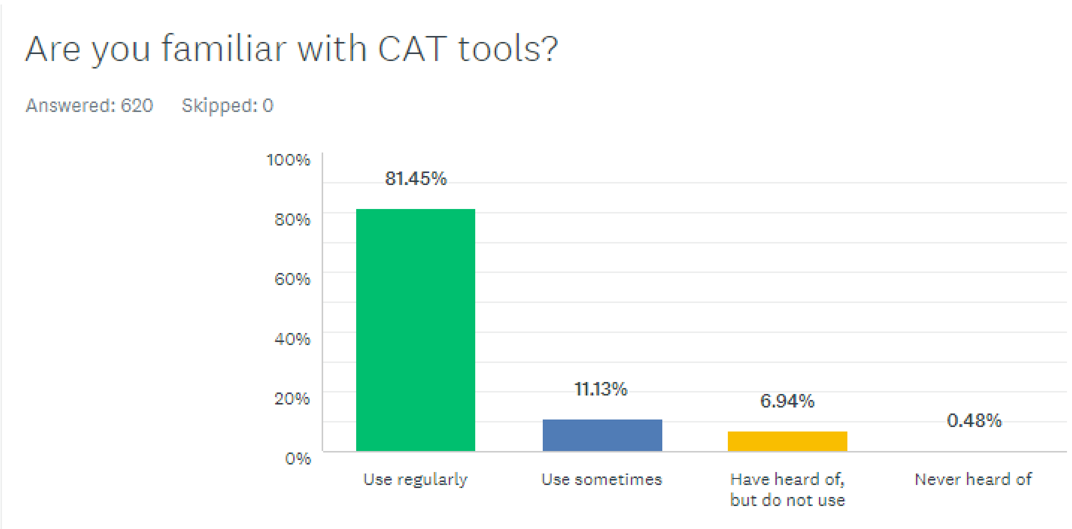


Figure 36. Familiarity and adoption of CAT tools (localisers)

Those who answered either “regularly” or “sometimes” were redirected to page 17 where they were asked two extra questions about the name of the programmes. The first question provided a list of regular CAT tools and the second one had systems specialised in localisation. Table 28 puts together the answers to both questions including all the tools that were mentioned in the section “other”. Once more, the respondents could select several options if they wanted to. The most commonly used CAT tool, according to the results, is memoQ with 496 responses, followed by SDL Trados with 352 and Memsource with 236. SmartCat and Wordfast also received a respectable number of answers.

In the case of localisation tools, the most common answer was “none”, which obtained 229 responses. The main reason behind this result is the fact that the best-known tools were usually designed specifically for utility software, the localisation of websites or apps and, even though they do provide a visual environment for those cases, they are incompatible with video games due to the coding languages used in them. However, with the appearance of new systems as well as new methods of video game development, the question about localisation tools was included as it could yield interesting results.

Among the options provided in the list, SDL Passolo received 161 responses and was followed by Poedit with 102 and Catalyst with 49. A fairly high number of participants chose MultiTrans and Lokalize from the list, while the last provided option, Gtranslator, only received 13 answers. Among those who chose “other”, 24 of them did not specify the name of the system or could not do so because of non-disclosure agreements. Crowdin appears to be the system that was mentioned the most in the comments, a company that has recently started creating a solid presence in the field of video game localisation and that will be discussed in Chapter 5. Additionally, there were also some references to Sisulizer, Wordbee, Localize direct and LEAF.

Surprisingly, the first question about CAT tools in general was the one that received the highest number of answers mentioning new systems that could potentially be friendlier towards video game localisation. The answers included numerous tools that are web-based and tend to be more focused on localisation as an activity on the whole. Some of them, such as XTM and Wordbee, even advertise their services specifically for the video game development industry. Other tools that were only mentioned once in the first questions and were not included in the table were: DejaVu, Sfera, ATMS, Lilt, Lokalise, Matecat, MemSource, Lingohub, PhraseApp, GTT, Polyglot, Idiom, HMI Linguist, Weblate, Transit, and CTE.

CAT tools		**Localisation tools**	
MemoQ	496	None	229
SDL Trados	352	SDL Passolo	161
Memsource	236	Poedit	102
SmartCAT	118	Catalyst	49
Wordfast	103	MultiTrans	40
Omega T	72	Lokalize	28
MateCat	41	Confidential / not specified	24
Déjà Vu	28	Gtranslator	13
XTM	21	Crowdin	10
Wordbee	16	Sisulizer	5
Crowdin	15	Localize Direct	2
Confidential / not specified	11	Wordbee	2
LEAF	8	LEAF	2
Smartling	8	DejaVu	1
CafeTran	6	Transifex	1
Across	4	Verifika	1
Lingotek, TWS, Transifex	3	Xbench	1
Transtool, Polyglot, Sisulizer, Message Studio, Similis	2	XTM	1

Table 28. Computer-assisted translation and localisation tools (localisers)

3.3.14 Machine translation tools

The first question of the survey's section about Machine Translation was, once again, about the familiarity and degree of adoption of these systems. The results show that among the respondents (Table 29), 315—or 50.81%—reported knowing about the existence of these tools but not using them, 37.10% said that they did use them sometimes, 11.61% used MT regularly and, surprisingly, 3 of the participants had never heard about it. If we analyse the background of those 3 participants and, as we did in the previous section, and filter all the results to only show theirs, we observe that they were either Danish, Taiwanese, or Turkish. Two of them had taken specialised courses or seminars in translation and one did not have any kind of studies in the field; one of them had "5 to 10" years of professional experience and two had "10 to under 20" years of experience. Translation was the main source of income for two of them and, in the case of the third participant, the comment only said: "freelance". Two of them had a workload of "more than 75%" in game localisation and the remainder had "less than 25%". Finally, we find two freelancers and one who worked in-house in a non-translation company.

FAMILIARITY WITH MT TOOLS	RESPONSES	
Have heard of, but do not use	50.81%	315
Use sometimes	37.10%	230
Use regularly	11.61%	72
Never heard of	0.48%	3

Table 29. Familiarity and adoption of MT tools (localisers)

The participants who either chose "never heard of" or "have heard of, but do not use" reached the end of the survey, whereas those who marked the options "sometimes" or "regularly" were sent to a final page. Page 19 was the last one of the survey and had two questions to specify the names of the tools the respondents usually worked with. In order to receive more detailed results, the tools were divided according to whether they were paid MT systems or free systems. If we analyse the results of the first question (Figure 37), we observe that almost 47% (or 143 of them) did not use commercial tools at all, 120 participants used Google Translate API, and 35 selected “other” either as well or as their sole answer (they were allowed to provide multiple answers). Among those who chose the third option, we find: Confidential or not specified (13); DeepL (8); SDL (6); Amazon (2); ATS; Lilt; MemSource; and a Microsoft base model customised for our purposes.

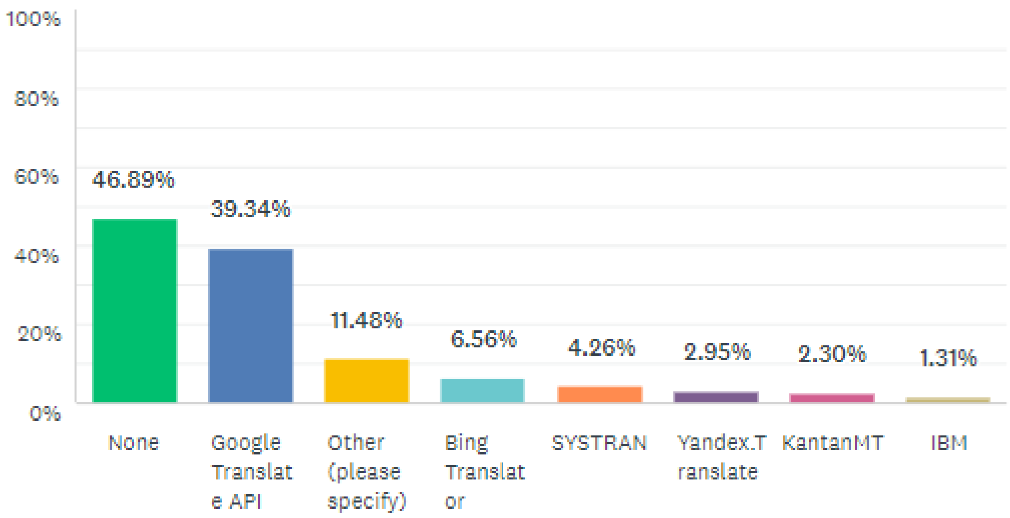


Figure 37. Commercial MT systems (localisers)

Google translate was in the first position with 191 responses if we analyse the results for free MT systems (Figure 38) and was followed by DeepL (128); the third option was “none” with 56; then Babylon (22) and, “other” was in fifth place with references to confidential or not specified (2); Reverso (4); Crowdin MT (3); Linguee (3); WordReference (2); Katò; Lingohub; Lokalise; MyMemory; Naver translation app; SmarCat; Weblio; and Yandex.

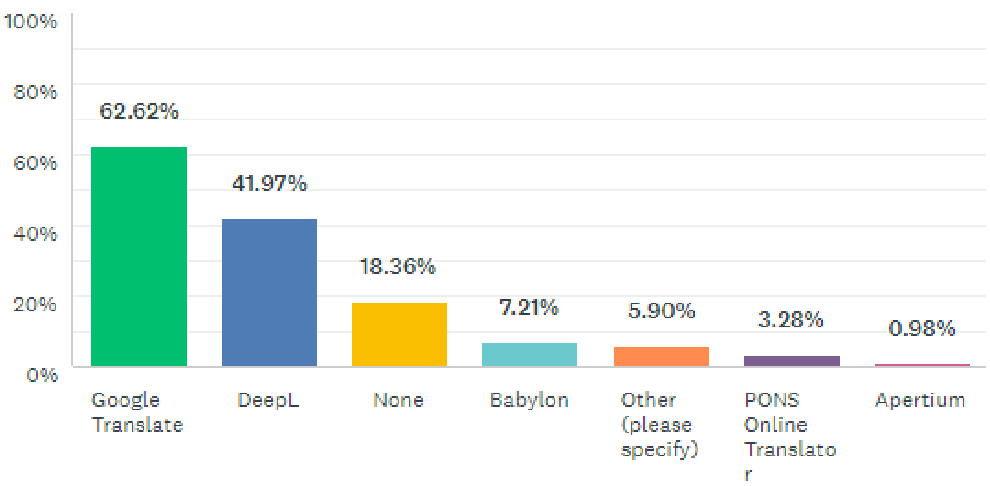


Figure 38. Free MT systems (localisers)

CHAPTER 4. SECOND AND THIRD SURVEYS

Chapter 1 introduced the video game development industry, the complex number of dependencies involved in creating a multilingual product and how crucial it is to carefully plan each step to avoid setbacks that can send the whole project tumbling down. Chapter 2 provided a cursory examination of the intricacies of the field of localisation and presented the context in which our first survey was developed in order to better understand the relevance of the questions that it contained. Chapter 3 provided the raw results from the said survey in terms of business practices in the video game localisation industry, the availability of reference material, localisers' access to the game itself, the tools most commonly used and the participants' attitudes towards them as well as the features and functionalities they need. Chapter 4 deals with the processes that chronologically precede and succeed video game localisation and were presented in the section about video game development. Therefore, the present chapter contains the results of two more surveys that were created to (i) investigate the consequences of localisers' limited access to content in the final product, (ii) gather technical data about video game development, and (iii) to better understand video game business practices. The surveys, which were significantly shorter than the first one, will be presented following the sequential order of implementation and the different subsections will be organised following the pattern of the previous chapter for consistency.

Thus, the first section of this chapter presents the results gathered in the second survey, which was created mainly to determine the most common types of bugs encountered in the LQA process. This endeavour is of particular significance due to the lack of studies in the field and allowed us to create a ranking of these bugs in order of importance. These results, along with those from the series of questions specifically designed to discern the causes of the bugs identified, will allow us to collect information from a different angle and shed new light on the linguistic issues encountered. Additionally, the data will also offer an insight into the current practices in LQA, the situation of the job market, and the tools they normally use. The second half of the chapter will be devoted to presenting the results from the third and final survey. Created to collect technical data about coding languages, cross-compatibility between systems, and attitudes towards different types of testing, the survey included questions about localisation and production models and game engines as well. In addition, the results about file formats used to isolate game text and some of the comments left by the participant will provide essential data in the analysis of technological solutions currently on the market.

4.1 Video game linguistic testing

The main purpose of this survey was to collect data about the final step in the linguistic process of the creation of a multilingual video game. As presented in Chapter 1, testers are faced with finding the errors that may occur both during the development phase and after the localisation process and must ensure that the product is as free of bugs as possible. Although some of them are fully specialised in LQA, many also work in other aspects of quality assurance or functional testing, thus gaining a comprehensive perspective of a product that they spend hours analysing and inspecting closely. Therefore, their insights in discerning the most common types of bugs or the causes behind their appearance will prove crucial and add another dimension to the topic. Following the example of the previous chapter, the design of the tables and figures included in both main sections will vary depending on the content in order to provide the most readable option.

4.1.1 Survey's implementation and design

The definitive version of the survey addressed to linguistic testers was finalised during the last week of July 2020 and the link used as a collector was created on the 31st of the same month. Given the positive results yielded by the improved implementation method of the previous questionnaire, it was also distributed via individual messages using the research parameters of LinkedIn and the link was also posted on Facebook to maximise the possibilities. Therefore, the completion rate rose from 62% (in the case of the first survey) to 71% and the time spent contacting professionals in the field was reduced considerably. The collector was officially closed on the 16th of September of the same year and the questionnaire received a total of 770 answers. In other words, the improved methodology allowed us to gather a high number of responses in the span of 48 days and included 550 complete answers. If we analyse the number of responses by calendar months (Figure 39), it received 9 complete answers out of 14 during the last day of July only, 471 out of an extra 643 in August, and 70 out of 113 during the first weeks of September. Another change that improved the completion rate and reduced the number of participants leaving the survey halfway through was the addition of a progress bar to the survey's design.

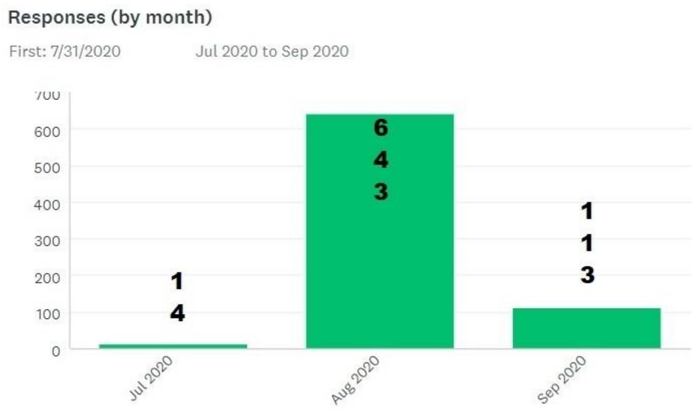


Figure 39. Total responses to S2 per month (testers)

The second survey was also created using SurveyMonkey and, as previously mentioned, the length was reduced considerably in order to increment the completion rate and only one of the questions followed the "page skip option" format. The full questionnaire was divided into 7 sections that were organised into 4 pages—although similarly to the first survey, there was an introductory page with a message explaining the purpose and thanking the participants, which brings the total to 5 pages. Consequently, with only 20 questions, the average time spent was 5 minutes for those who reached the final page and 4 minutes if we take into account all the respondents. Table 30 puts together the different sections along with the different topics covered by the questions therein. Similarly to the previous survey, the number of questions does not match the number of topics as some were more complex and required various queries.

SECTION	TOPICS
15. Personal information	Age & nationality
16. Languages	Native language
17. Professional information	Studies, professional experience, main source of income, workload & type of employment
18. Business practices	Teamwork & tasks
19. Tools	Bug reporting tools
20. Linguistic bugs	Prevalence of linguistic bugs & causes
21. Additional comments	Additional comments

Table 30. Sections and topics for LQA (testers)

Additionally, the number of pages does not match the number of sections as the questions were increasingly grouped together to reduce the number of page changes. This decision was taken in order to reduce the percentage of respondents who abandoned the survey halfway through since the analysis of the previous questionnaire indicated that most participants usually gave up when they were directed from one page to another. The hypothesis proved to be correct and, aided by the abovementioned progress bar, the reduction in the number of abandons was significant. Thus, with only 5 pages containing questions and the last page being completely optional, 157 participants left the survey when they advanced from page number 2 to page number 3 and 63 did not answer the questions on page 4. The last page was only for those who wanted to add additional comments, which was the case for 34 testers (Table 31).

PAGE NUMBER	NUMBER OF RESPONSES
PAGE 1: WELCOME MESSAGE	-
Page 2: Personal Information	770
Page 3: Professional information	613
Page 4: Linguistic testing tools	550
(Skip question) Page 5: extra comments	34 (skipped 736)
FINAL MESSAGE	550

Table 31. Number of pages and total number of answers (testers)

Similarly to the survey addressed to localisers, this questionnaire's first section only contained two questions designed to collect data about the respondents' age and nationality—said questions followed the same format and were created in the same manner. The participants were also asked about their native language and, once again, the question was formulated using a checkbox format that, although limited in the number of options (only 200), allowed them to select multiple choices in the case of bilingual or trilingual respondents. The following section, about professional information, was also identical to the first survey and the questions (which had minimal differences) followed a multiple-choice format. Located on page 3, two more questions about teamwork and the tasks involved in linguistic testing were added at the end of the said page in order to know if they only had to report the issues or also fix the errors they found. Section 5 was on page 4 and had a single question and respondents were asked to choose from a list of different tools or methods used for reporting bugs. Section 6 contains the most important questions of the survey, as they aimed at analysing the prevalence of the different types of bugs by first creating a ranking and subsequently discerning the causes of those linguistic bugs. SurveyMonkey provides the possibility to create rankings that are easy to answer as the respondents are only asked to either drag the different elements or directly select the number. Image 22 illustrates this question's appearance from the respondents' point of view.

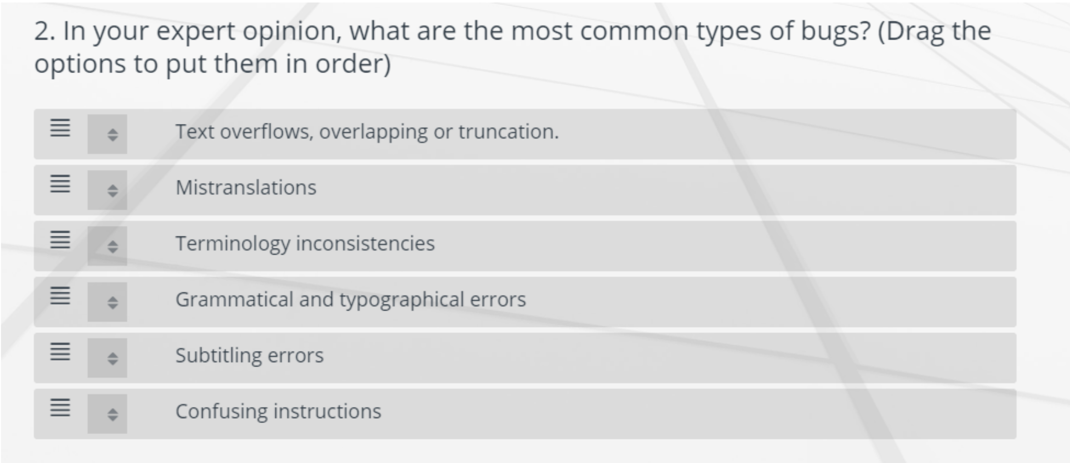


Image 22. Example of the ranking format from the respondent's point of view (testers)

The rest of the section contained multiple-choice questions where the participants were asked to select the cause of every type of linguistic bug—excluding grammatical and typographical errors—from a pre-existing list of options. Additionally, they could provide another reason if they could not find the root of the problem among the proposed items. The last question on the page followed the "skip logic format" and would either send the respondents to the end of the questionnaire—if they did not want to leave a message with extra information—or to the fifth and final page of the survey that contained a comment box.

4.1.2 Personal information

As previously explained, this section only had two questions and once again, all references to the name of the company (or personal names) were avoided for confidentiality reasons. The first question was about the age of the participants and, as in the previous survey, it had 7 possible answers: 17 or less, 18 to 24, 25 to 34, 35 to 44, 45 to 54, 55 to 64, and 65 or older. Once again, it included "17 or less" to evaluate the impact of students in the industry and to maintain consistency with the other surveys. Figure 40 shows the results and allows us to observe that, once more, none of the 550 participants who completed the survey was part of the first or last age groups. The largest group is that of testers with an average age of “25 to 34” who amounted to a total of 358 participants; it was followed by those between the ages of 35 and 44, although the difference between the numbers is stark (114 respondents). Only 58 participants were concerned by the second group of age, 17 were in the “45 to 54” category, and 3 selected the sixth option.

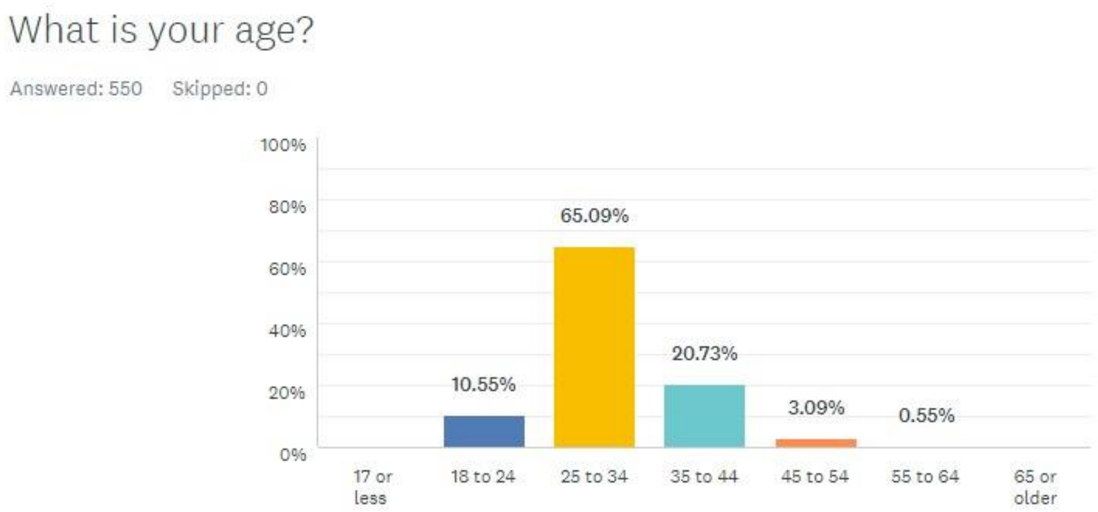


Figure 40. Respondents’ age (testers)

The last question of the section was designed to collect data about the respondents’ nationalities which will be subsequently contrasted in Chapter 5 with the results about their native languages, their educational background, and the tasks they have to carry out as linguistic testers. The question was created as in the previous survey and, out of the 226 different nationalities listed, 63 had at least one representative and 21 of them were selected 10 or more times. Table 32 puts together those 21 nationalities and shows that France, Italy, and Spain were found at the top of the list, following the trend that appeared in the previous survey. The other nationalities grouped by the number of occurrences were: Portuguese, which received 9 responses; Danish, Egyptian, Indonesian, and Taiwanese were selected 8 times; Hungarian appeared 7 times; Greek and Indian had 6 representatives each; Austrian, Chilean, English, Finnish, and Venezuelan were chosen as the nationality of 5 participants in each case; 4 respondents selected Belarusian, Slovak, or Thai; Costa Rican, Hong Konger, Malaysian, Romanian, and Vietnamese received 3 answers each; and Afghan, Colombian, Croatian, Ecuadorean, Lithuanian, and Swedish were selected twice in each case. Additionally, there were 21 nationalities that were only chosen once: Algerian, Armenian, Australian, Azerbaijani, Bahamian, Bolivian, Cameroonian, Estonian, Greenlandic, Iraqi, Irish, Kuwaiti, Latvian, Malagasy, Nigerian, Salvadorean, Singaporean, Surinamese, Syrian, Uruguayan, and Yemeni.

NATIONALITIES	RESPONSES	
French	12.60%	97
Italian	10.52%	81
Spanish	10.52%	81
Polish	8.31%	64
Brazilian	7.27%	56
German	4.55%	35
Russian	2.73%	21
Turkish	2.47%	19
American	2.34%	18
Canadian	2.21%	17
British	2.08%	16
Argentine	1.95%	15
Mexican	1.82%	14
Chinese	1.69%	13
South Korean	1.69%	13
Ukrainian	1.56%	12
Czech	1.43%	11
Dutch	1.43%	11
Japanese	1.43%	11
Belgian	1.30%	10
Norwegian	1.30%	10

Table 32. Nationalities with more than 10 respondents (testers)

4.1.3 Native and working languages

In order to provide a clearer structure and follow the pattern of the first survey, the question about the respondents' native language will be included and discussed apart from the questions included in the sections that dealt with personal and other professional information. The participants were only asked about their native or working languages and, as previously mentioned, they were given the possibility of choosing various options if they were either bilingual or trilingual. The question was identical to the one included in the first survey and, to compensate for the limitations in the number of options, the respondents were provided with the possibility of specifying other languages that might have not been included in the pre-existing list. When we take into account all the answers including all the languages that were mentioned in the comments section, we can observe that 54 different languages received at least one response. Furthermore, one of the participants left a comment (unedited) explaining that, even though he or she did not speak or know the languages:

> Actually, I can work in any language, I don't "judge" if the localization is precise or suitable, I only report standarized processes like spelling issues, instructions unfollowed, DNT, glossaries unfollowed, standard grammar, etc. I do the QA and Testing of any language.
>
> 8/26/2020 5:12 PM

This comment certainly explains the fact that, even though less than 10% of the respondents reported coming from an English-speaking country, that language was at the top of the list with 579 responses. The order changes in comparison with the list of nationalities and we find Spanish in the second position with 165 answers, followed closely by French with 153, and Italian with 89. Table 33 shows, once again, the list of the languages with more than 10 representatives although it is worth mentioning that the option "other" was chosen 17 times and appeared in the 13th position. The languages that did not make the table (and the number of times they were selected) were: Catalan and Indonesian with 9 responses each; Finnish and Greek were chosen 6 times; Hindi, Hungarian, Slovak, Swedish, and Thai were the native or working languages of 4 professionals each; Malay, Romanian, and Vietnamese appeared 3 times each; Albanian, Bulgarian, Croatian, Javanese, Lithuanian, and Marathi were selected twice each. Those that only had one representative were: Afrikaans, Armenian, Fiji, Icelandic, Latvian, Mongolian, Persian, Quechua, and Urdu. In the section "other", besides the abovementioned comment, the remaining languages and dialects were: Latin American

Spanish (mentioned 3 times), Brazilian Portuguese (specified twice), Cantonese (twice as well), French Canadian (also twice), Malagasy, Mandarin (traditional), Marathi, Northern Sami, Pashto, Hausa, Yoruba, and Tagalog.

NATIVE LANGUAGES	RESPONSES	
English	75.19%	579
Spanish	21.43%	165
French	19.87%	153
Italian	11.56%	89
Portuguese	9.09%	70
German	7.66%	59
Polish	7.40%	57
Russian	5.32%	41
Japanese	4.68%	36
Chinese (Mandarin)	3.77%	29
Dutch	2.47%	19
Turkish	2.21%	17
Arabic	2.08%	16
Korean	2.08%	16
Czech	1.69%	13
Ukrainian	1.43%	11
Danish	1.30%	10
Norwegian	1.30%	10

Table 33. Native languages with 10 or more responses (testers)

4.1.4 Professional information

The questions included in this section were almost identical to those that were used in the first survey in order to allow us to contrast the results between surveys. Additionally, these questions portray the respondents' education in language-related studies thus enabling us to compare the results with their daily tasks as linguistic testers as well as to gather information about the degree of specialisation of the respondents. Once again, the questions were grouped under the label "professional information" to maintain the internal structure of the present study. The 5 questions were all located on page number 3 and, as in the previous questionnaire, all of them followed a multiple-choice format. Although all the options were directly provided, the respondents could, once again, clarify their answer if their main source of income was not linguistic testing or if they had a low percentage of LQA in their day-to-day workload.

4.1.4.1 Educational background

The respondents were asked whether they had formal education or training specifically in either translation or other types of language-related studies besides translation. Even though testers might come from a wide variety of backgrounds, the question was specifically phrased in this particular way to analyse their studies in languages and contrast the results with the percentage of respondents fixing linguistic issues. Although most of them work in their native language when performing LQA tasks, we find that a background in languages might play an important role in better understanding the causes of the bugs and pinpointing potential solutions.

The question provided the same 5 options included in the first survey: none; master's degree; bachelor's degree; PhD; and specialised courses, seminars, workshops, etc. Figure 41 shows their answers sorted in descending order and we can observe that 287 (or 52.18% of the participants) did not have any sort of university degree in either translation or language-related studies—a fact that does not exclude higher education. Conversely, 35.09% of them—or 193 respondents—did have either a MA, a BA or a PhD in the domains specified in the question. Only 70 of them had followed specialised courses, seminars, workshops or similar.

As explained before, the cause for these results lies in the fact that they were specifically asked about linguistic training and testers tend to come from diverse educational backgrounds. Although some of them might be translators in their first paid job in the industry awaiting an in-house position or trying to get into the field (or simply supplementing their income by diversifying their services), others are developers, producers, or designers in an entry-level position hoping for an opening in their respective fields.

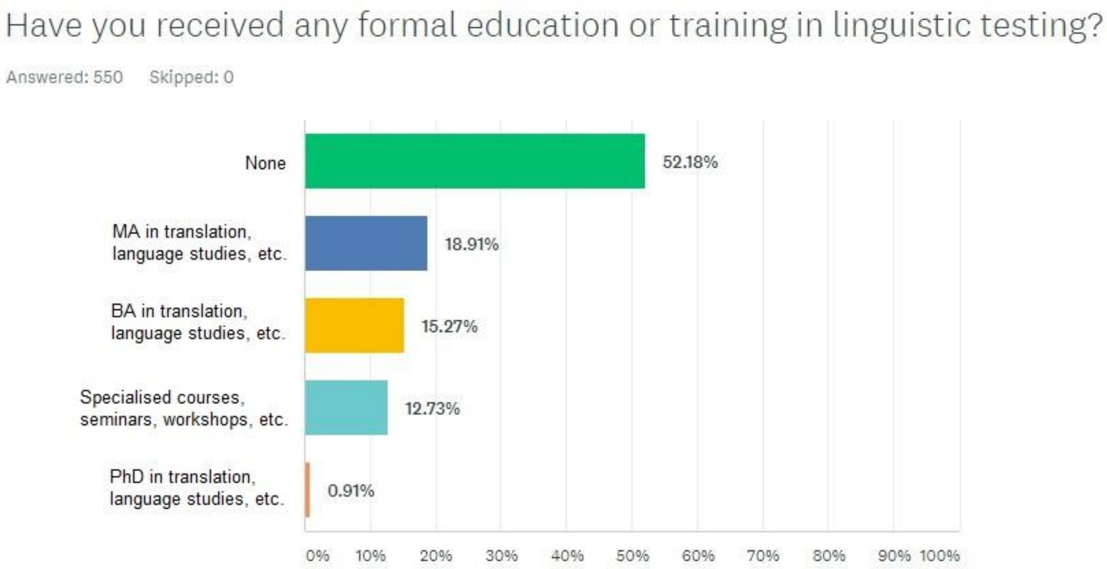


Figure 41. Educational background (testers)

4.1.4.2 Professional experience

Similarly to the previous survey, the respondents were asked a series of questions about their professional experience as linguistic testers. The first question focused on the length of their professional careers in this particular field whereas the following two questions aimed at establishing their expertise as linguistic testers as opposed to other types of testing or other activities. Therefore, the participants were also asked if linguistic testing was their main source of income and, subsequently, the percentage of LQA they normally dealt with in their daily workload.

For these reasons, the first question included 5 different options that ranged from “less than 1 year” to “more than 20 years”, reproducing once more the model used in the first survey. As Figure 42 illustrates, 287 participants—or 52.28%—had a career in linguistic testing with a total duration of 1 to under 5 years of professional experience. The second most common answer was respondents with less than 1 year of experience in the field since the category received 132 responses. The third group in the top three received almost 17% of the answers and was formed by participants with under 10 years of experience in the field of LQA. In light of these figures, we can clearly observe the reality of the market in question as the results confirm that most linguistic testers move up the ladder as soon as they can. Additionally, the growing number of new testers also shows, as in the previous survey, the increasing popularity of the field.

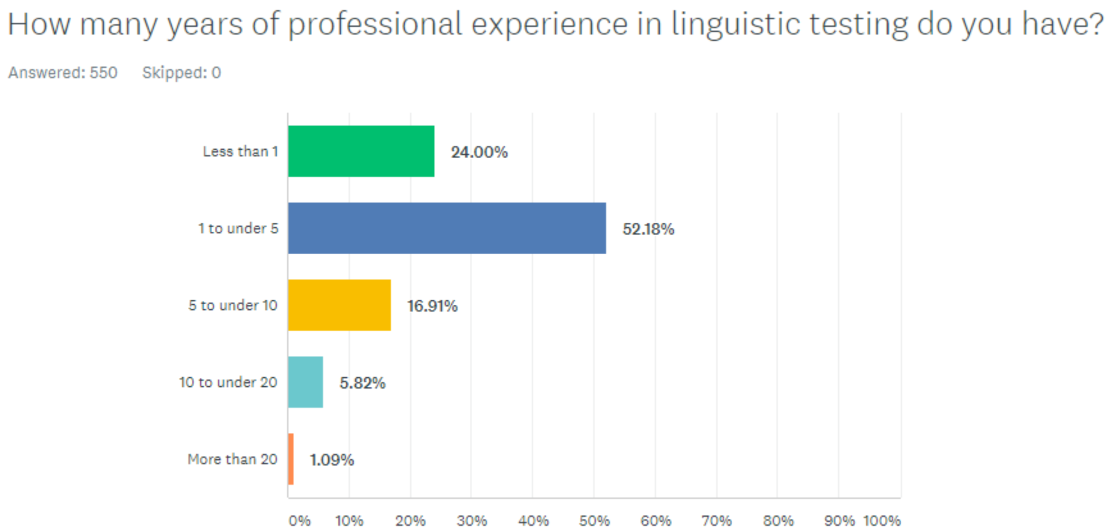


Figure 42. Years of professional experience (testers)

When the participants were asked if linguistic testing was their main source of income, 333 of them (or 60.55% of the total) selected the option “yes” and the remainder marked “no” as their answer (39.45% or 217). This question included the option of explaining what their main employment was (if they wished to) and 168 participants left a comment. Among the most common answers, we find those who worked mostly in either translation or localisation (51 out of those who left a message) and 41 participants who stated that they performed various types of testing including QA in general or functional testing. There were also 13 comments that explained that they mostly worked in project management, coordinating tasks or as team leads and 11 participants said that they mostly worked in proofreading or reviewing content. Other comments with positions related to either the game industry or the language industry were: localisation editing, engineering and IT in general (mentioned 8 times each); development, design, analyst (7 times each); writing, copywriting, creative writing (5 times); and teaching (5 respondents). Others were in completely different fields, such as marketing and psychology (which were mentioned twice each); an engineer of production in a refinery; someone who worked at a municipality; or a nurse.

The third and final question was created to inquire about how much of their professional activity was in linguistic testing for video games specifically. Similarly to the previous survey, the questions included 4 different options ranging from “less than 25%” to “more than 75%” and the respondents could also leave a message. As Figure 43 shows, 48.55% (or 267) of them stated that their activity was mainly LQA. Conversely, the second most common answer was “less than 25%” with 195 answers (35.45%). Few of them, compared with the results of the previous categories, had a percentage of LQA tasks that occupied around 50% of their time (48 participants) or more than 50% but less than 75% (40 respondents). On this occasion, the number of comments left in the “please specify” section included 36 out of 101 messages that read something along the lines of “100% LQA”, thus proving that not only those who selected the “less than 25%” option left a comment. The rest of the comments were subsequently grouped by topic in order to provide a clearer image. The topics and number of occurrences were: other types of testing (11); project management, coordination or team lead (10); and translating, proofreading, or reviewing (7).

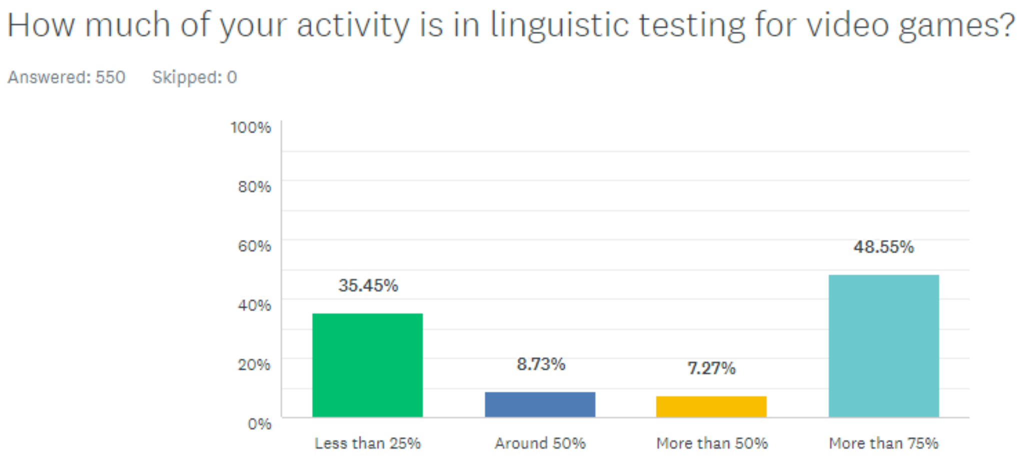


Figure 43. Percentage of activity (testers)

4.1.4.3 Type of employment

The question about the type of employment was included in order to evaluate the impact of subcontracting services for linguistic testing and to retrieve quantitative data about the percentages of in-house positions in video game development companies. The respondents were provided with the same options that appeared in the first survey due to the potential similarity in business practices in both fields. However, this particular questionnaire only contained 7 different options as it omitted the reference to fan testers due to the apparent inexistence of said phenomenon and the fact that the survey was mostly addressed to professionals (or at the very least students in a part-time job). Thus, the categories included were: testers working with in-house contracts either at a non-translation company or a translation company; independent freelancers; independent freelancers working exclusively for an agency; for freelancers working both independently and for an agency and, finally, there was an option for students. The results displayed in Table 34 are sorted in descending order and contain both the percentages and the absolute numbers.

Contrary to the previous survey, more than 70% of the respondents (71,27% to be more exact) worked in-house either in a non-translation company (in all likelihood a game development company) or in a translation agency. Additionally, the first position was for testers with a contract with a non-translation company and the difference between those at development companies and those working for translation companies was almost half. These results prove that even though the industry of video game development tends to externalise localisation services in general, linguistic quality assurance seems to be performed mostly by the company itself. The combined results of all the participants who chose any of the three options that entailed freelancing—26.18% or 144—is barely higher than those who worked in-house in a translation company—only 12 more responses. Finally, there was a very low percentage of students, the category only received 14 responses, not even reaching 3% of the total. These results are consistent with the fact that testers need to have access to the game and companies are reluctant to share the game when it comes to external providers due to potential confidentiality issues.

TYPE OF EMPLOYMENT	RESPONSES	
In-house in a non-translation company	47.27%	260
In-house in a translation company	24%	132
Independent freelancer	11.27%	62
Freelancer working with an agency	7.64%	42
Freelancer working both independently and with an agency	7.27%	40
Student	2.55%	14

Table 34. Type of employment (testers)

4.1.5 Business practices

The purpose of this section was twofold, first to determine whether linguistic testers usually worked in teams or if it was something that depended on the project. Secondly, the final question aimed at establishing the tasks that LQA usually involves, namely if they also had to provide the correct translation or solution for the linguistic bugs they reported. The first point was included both for consistency issues and to avoid pre-conceptions due to the current lack of studies or surveys addressed to linguistic testers in academia. Furthermore, the results gathered by the second question will be correlated, cross-referenced, and compared in Chapter 5 with the respondents' nationalities, native or working languages, and their educational backgrounds for a more comprehensive perspective. As explained in section 4.1.1, both questions were located at the end of page 3 and, although they were answered by 613 respondents, only the results of those who completed the survey will be taken into consideration.

In the first question, the participants were provided with three different options: mostly alone, mostly in teams, and both. On the one hand, if we analyse all three options separately, "mostly in teams" received the highest number of answers with 65.27%—a total of 359. Additionally, 20.73% (114 participants) chose "both" as their response, and 14%—or 77—selected "mostly alone". On the other hand, if we consider the combination of the two options that included the presence of teamwork practices, 475 of the 550 (86%) participants were used to working in teams on a regular (or semi-regular) basis. These results stem from the fact that depending on the size of a video game and following the level of attention to detail LQA normally entails, it might be impossible for a single person to thoroughly test all the levels, branching options of dialogues, etc. in a reasonable amount of time. Therefore, companies tend to rely on teams working together in order to comply with the tight deadlines that characterise this last phase of the development process.

Furthermore, the results collected about the tasks involved in LQA also show a clear tendency towards fixing the linguistic bugs detected or, at the very least, suggesting a possible solution to the issue that might be later evaluated by a linguist (in some cases) and implemented by the person in charge of fixing the bugs. These practices will, evidently, depend on the type of bug, as sometimes adjusting the size of the menu might suffice. As Figure 44 illustrates, almost 54% of the respondents stated that they had to find a solution or suggestion for the bugs and, in the case of 191 participants, it depended on the project. Only 11.45% (or 63) of them were only tasked with reporting them without taking any other measures.

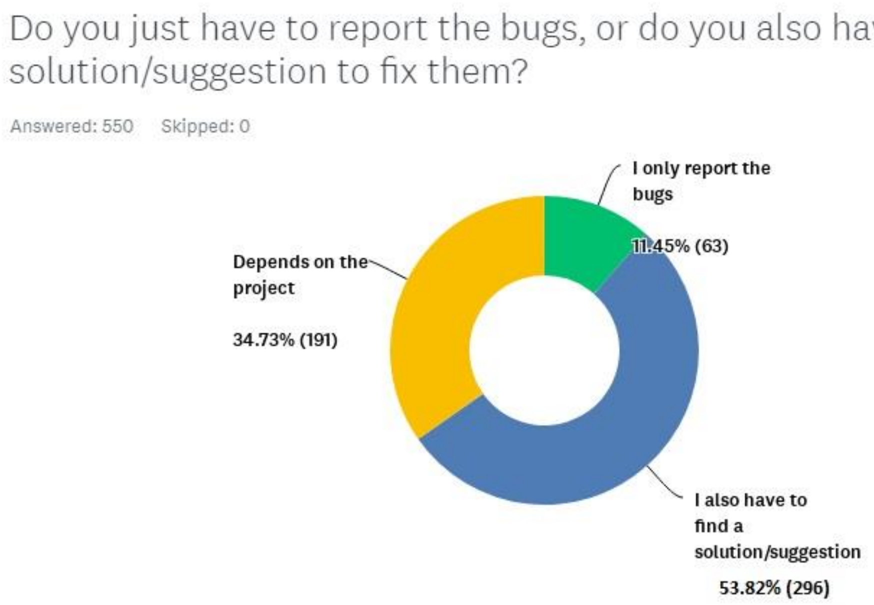


Figure 44. LQA tasks (testers)

4.1.6 Linguistic testing tools

The participants were also asked about the tools they used for reporting the bugs they encountered in order to analyse whether the systems used were compatible with other programmes used in video game localisation. The main reason was to evaluate the tools' integration capacities in general and the possibility of conceiving a comprehensive solution for the full linguistic process by combining current solutions on the market. The question was also located on page 4 and followed a checkbox format to allow multiple answers. Therefore, respondents were able to choose the option "other" if the tool—or method, as some might not employ a programme—they used was not listed. As Table 35 shows, similarly to the first survey, the most widely used tool was Jira with 439 answers followed by DevTrack which received 136, and "proprietary software" in case of an NDA. Among the comments left by the 85 participants who selected "other", 18 specified that it was either confidential or they did not want to provide the name and 22 wrote down Redmine, which seems to be the open-source version of Jira according to their website[35]. The rest of the programmes or systems mentioned in that section more than once were: Excel or google spreadsheets (14); Azure DevOps or Visual Studio (11); Bonsai (9); Helix (5); Github (2). Those that only

[35] https://www.atlassian.com/software/jira/comparison/jira-vs-redmine

appeared once were: ADO, Asana, Backlog, Devon, FogBugz, memoQ, Nintendo Marking Tool, Qubo, Radar, TapD, Trello, and TTP.

BUG TESTING TOOLS	RESPONSES	
Jira	79.82%	439
DevTrack	24.73%	136
Proprietary software	19.09%	105
Other (please specify)	15.45%	85
Mantis	14.73%	80
Hansoft	8.91%	49
None	8.73%	48
Bugtracker	8.36%	46
Bugzilla	7.82%	43
Test Track Pro	5.64%	31
PR Tracker	0.36%	2

Table 35. Bug testing tools (testers)

4.1.7 Linguistic bugs

As previously mentioned, the main purpose of this survey was to try to create a ranking in order to identify the most common type of linguistic bug encountered during the LQA phase as well as to assess the prevalence of the rest of them and discern their causes. This step was necessary due to the lack of papers, presentations, or studies about this precise topic besides a brief comment on a website[36]. Therefore, in order to tackle those linguistic issues, the first action consisted in identifying the most common bugs and their frequency in order to concentrate on the most important ones. The denomination given to each category follows the names used in Muñoz Sánchez's monograph about video game localisation (2017, p. 8-9) although only the bugs that were directly linked to linguistic issues were included since the rest of them can be considered as being caused mostly by implementation errors. In his monograph, the author devotes a chapter to the specificities of linguistic testing from a professionalising point of view, providing tips and examples in a descriptive manner, although he does not mention the frequency of appearance.

[36] https://www.gamesindustry.biz/articles/2021-04-21-the-keys-to-multilingual-game-development

Thus, the survey contained seven different questions where the participants were first asked to sort the bugs in order of frequency and subsequently, they had to answer a series of questions that aimed at finding the most common cause for each of those bugs. Figure 45 supplies a simplified version of the results gathered and shows that according to the respondents, the most frequent type of linguistic bug testers have to deal with is "Text overflows, overlapping or truncation". Furthermore, we can observe that there is a significant distance between the first and the second most common type, that is, "Mistranslations". "Grammatical and typographical errors" appeared in third place and was followed closely by "Terminology inconsistencies", then "Subtitling errors" (far behind the previous ones) and finally, "Confusing instructions".

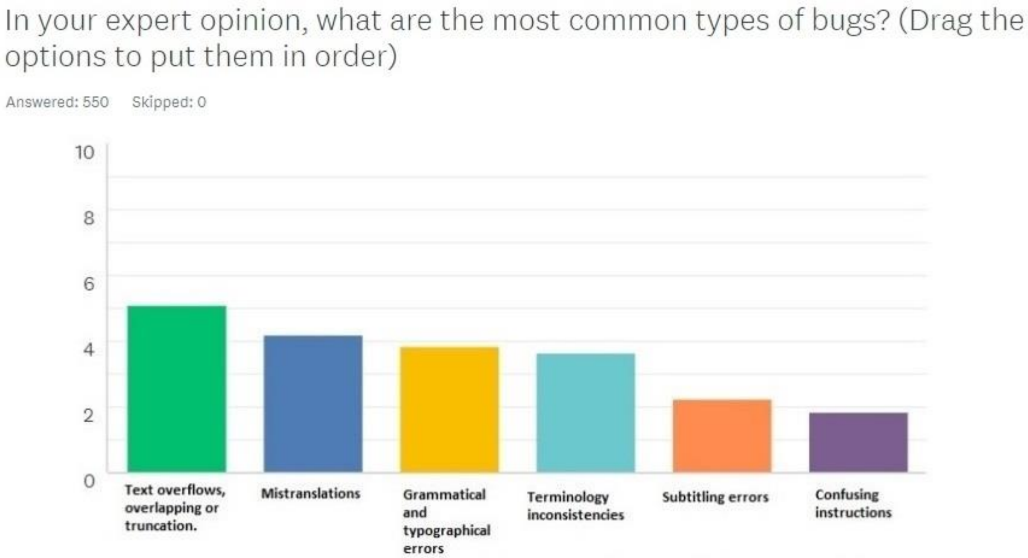


Figure 45. Types of bugs sorted by frequency (testers)

Additionally, SurveyMonkey allows us to analyse in more detail these results by breaking down each category in order to observe the number of participants that ranked each bug type first, second, third, fourth, fifth or sixth. These details will allow us to identify potential grey areas in the overall position of an option and better understand the results. As Figure 46 illustrates, the detailed results for either "Text overflows, overlapping or truncation" were fairly definitive and leave little space for doubts. Bugs or issues related to the appearance of the text on the graphical interface of the game were chosen 318 times as the most common type of bug and 95 as the second most common, consolidating it as the most frequent by far. Conversely, the results for "confusing instructions" show that the category was ranked last on 300 occasions and fifth 127 times, clearly marking this type of bug as the least common.

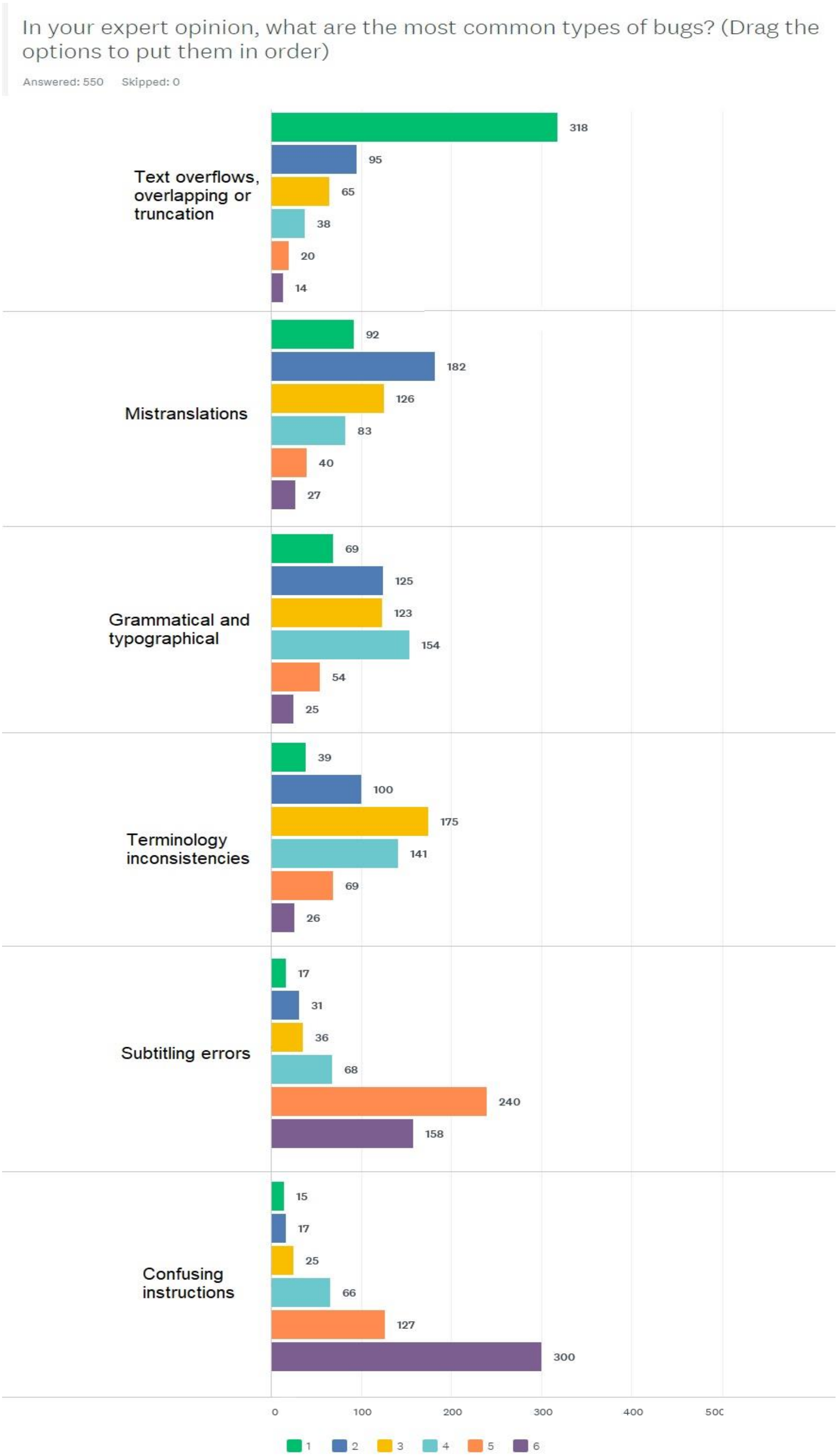


Figure 46. Detailed results of the linguistic bugs ranking (testers)

Although slightly less definitive, in the case of "subtitling errors" we find a somewhat similarly clear result as 240 respondents selected it as the fifth option—even though a fairly high number of participants (158) also thought it should appear in the last position. However, once we analyse the results for the second, third and fourth types of bugs; we can see that the total results are much closer together and the respondents were less polarised. This is the case for the answers received about the frequency of "grammatical and typographical errors" and "terminology inconsistencies" where, although the position with the largest number of responses suggested that they should inverse their spot in the ranking, it is the average weight of the respondents who chose otherwise that ended up deciding their final place in the list with minimal differences. In the case of "mistranslations", 182 participants said that it was the second most common bug and 126 ranked the issue as the third most common which, although more mitigated than the first, fifth and sixth types of bugs, remains a relatively solid result.

Once the participants had identified the prevalence of bugs, the survey included six questions to inquire about the causes of each type according to the testers' opinions and experience. The decision to include these questions stems from the testers' unique perspective as they have access to the game itself, which gives them a comprehensive vantage point. Thus, for the category "overflows, overlapping or truncation", as Table 36 shows, 284 participants said that "lack of visual environment for the translators" was the main reason. The second cause was deemed to be "insufficient instructions for the translator or failure to follow the instructions" although it was only chosen by 90 of them, this second option includes supplying information about the maximum number of characters.

In your expert opinion, what is the main reason for text overflows, overlapping or truncation?	**RESPONSES**	
Lack of visual environment for the translators	51.64%	284
Insufficient instructions for the translators or failure to follow the instructions	16.36%	90
All of the above	15.09%	83
Other (please specify)	13.09%	72
Don't know	2.18%	12
Lack of translation quality	1.64%	9

Table 36. Causes of overflows, overlapping or truncation (testers)

The 72 respondents who chose the option "other" were asked to leave a comment, and the majority explained that it was due to the fact that video games were mostly developed by English native speakers that do not consider that other languages need more space in the UI. Table 37 shows two comments that summarise the rest of the messages left, including not informing the translator about character limitations.

8/24/2020 4:32 PM	Lack of developer awareness of how much longer text will become in translation. Lack of clarity for translators regarding character limits, auto-rendering, etc.
8/3/2020 3:37 PM	Actually in the case for (sic) the company I work for it is more a design issue than the translator's fault. Usually if they are given a character limit they will respect it, but most of the time they don't have specific instructions and the design doesn't always consider that other languages need more space than EN.

Table 37. Comments about causes of overflows, overlapping or truncation (unedited)

According to the respondents' answers in the case of "mistranslations" (Table 38), the main cause was the "lack of in-text context", which was selected by 224 of the participants. The second option was "all of the above" (92 responses) which included all the causes that were provided (lack of visual environment as well, which ranks third). Although only 22 testers chose "other", some of the conclusions confirm that not having access to the game itself causes a high degree of ambiguity and that localisers lack resources. One of them also points out that some companies use machine translation without proofreading which causes translation issues as well (Table 39).

In your expert opinion, what is the main reason for mistranslations?	**RESPONSES**	
Lack of in-text context (e.g., the strings aren't in order, etc.)	40.73%	224
All of the above	16.73%	92
Lack of visual environment for the translators	16.00%	88
Insufficient instructions for the translators or failure to follow the instructions	11.82%	65
Lack of translation quality	9.64%	53
Other (please specify)	4.00%	22
Don't know	1.09%	6

Table 38: Causes of mistranslations (testers)

8/24/2020 10:38 PM	In one way or the other, definitely lack of context. Be it visual or in-text context or just general context about the game mechanics.
8/24/2020 4:32 PM	Ambiguity in the source, failure of developer to answer questions in a timely manner
8/17/2020 4:51 PM	Lack of in-text context but also lack of general context such as glossaries, character bibles, etc.
8/2/2020 1:24 PM	The translators don't know the context. Also, some companies shamelessly use machine translations which (sic) are almost never accurate.

Table 39. Comments about causes of mistranslations (unedited)

In the case of terminology inconsistencies, the first cause was "lack of resources such as glossaries, character bibles, etc." (183) and once again, with 133 answers, "all of the above". The third option, "too many translators" also received a consistent number of responses (115) (Table 40). This lack of resources, which seems to be endemic to the industry, has also been already analysed from the localisers' point of view in the first survey. Additionally, among those who left a comment after selecting the option "other", we find references to constant changes in the game itself resulting from the sim-ship model or the lack of continuity when it comes to hiring the same translators. The latter can result in a high number of issues if they do not provide translation memories or if the document is split between localisers without providing them access to the segments they are not supposed to be working on (Table 41).

In your expert opinion, what is the main reason for terminology inconsistencies?	**RESPONSES**	
Lack of resources such as glossaries, character bibles, etc.	33.27%	183
All of the above	24.18%	133
Too many translators working on the same project	20.91%	115
Insufficient instructions for the translators or failure to follow the instructions	14.55%	80
Other (please specify)	3.64%	20
Don't know	3.45%	19

Table 40. Causes of terminology inconsistencies (testers)

8/31/2020 8:45 PM	The dev team constantly changing its mind about the glossaries, late implementations (sic) resulting in translators missing the latest updates, trying to “modernise” the translations and not sticking to the translations of the previous games of the same license.
8/8/2020 11:37 PM	Most of the time, issues are present in the extensions (translation made after several months and therefore potentially another person)
8/5/2020 8:00 PM	All of the above, plus, when it comes to Certification Terminology, many times freelance translators are not trained or not sufficiently trained to handle terminology. They are just told "follow this list", without a proper training. Hence sometimes translators would ignore terminology because "it's ugly" and fail to realize how strict this task is. For the Training (sic) issue, the fault lies in management of translation vendors most of the time. It's important to note that anyone with the minimal knowledge on (sic) how Terminology works can verify any certification-risky string before delivering to a Developer, and it's unfortunately too often that this type of issues is found during a LQA cert pass.

Table 41. Comments about causes of terminology inconsistencies (unedited)

“IT-related issues” with 150 responses and “lack of visual environment” (chosen 125 times) were the first and second main causes of subtitling errors (Table 42). Only 32 participants chose the option “other” and explained that these types of issues had more to do with mismatches unrelated to the localisers or lack of access to the video itself although the category had to be included as it could stem from localisers (Table 43).

In your expert opinion, what is the main reason for subtitle issues?	**RESPONSES**	
IT-related issues	27.27%	150
Lack of visual environment for the translators	22.73%	125
Don't know	14.36%	79
All of the above	11.27%	62
Insufficient instructions for the translators or failure to follow the instructions	10.36%	57
Lack of translation quality	8.18%	45
Other (please specify)	5.82%	32

Table 42. Causes of subtitle issues (testers)

8/21/2020 9:10 PM	Subtitle issues may not be down to the translator. I am referring mainly to VO/text mismatches (missing subtitles, repetitions, incorrect timing, etc.)
8/10/2020 5:40 PM	Might be lack of IT support (special characters, problems with re-sizing, truncations, and anything related), timing, sync.
8/7/2020 11:26 AM	Most subtitle related issues tend to come from subtitles mismatching the audio or disappearing too fast. Again the source of this problem is is that English is a shorter languages and therefore does not need to be displayed as long or doesn't need as many lines causing awkward breaks in sentences.

Table 43. Comments about causes of subtitle issues (unedited)

Finally, confusing instructions were deemed to arise once more from "lack of context in-text" (162) and, as usual in the video game industry, "lack of visual environment" was second with 119 responses (Table 44). However, the name of the category—chosen following those used by Muñoz Sánchez (2017)—proved to be opaque for some of the respondents who selected "other" and, as a consequence few of the comments were relevant. Table 45 includes two that provided actionable data and refer to the fact that the instructions given to the player are not clear enough and might compromise the gameplay.

In your expert opinion, what is the main reason for confusing instructions?	**RESPONSES**	
Lack of context in-text	29.45%	162
Lack of visual environment for the translators	21.64%	119
Insufficient instructions for the translators or failure to follow the instructions	14.73%	81
All of the above	14.00%	77
Don't know	7.27%	40
Lack of translation quality	7.09%	39
Other (please specify)	5.82%	32

Table 44. Causes of confusing instructions (testers)

8/31/2020 8:45 PM	Lack of time and will from the dev teams, the infos (sic) are usually scattered everywhere and the translators don't have time to check everything before the deadline.
8/4/2020 11:02 PM	Poor quality of the source text, which leads to guesswork by translators and which sometimes results in confusing or incorrect instructions

Table 45. Comments about causes of confusing instructions (unedited)

4.1.8 Extra comments from LQA

The final question on page 4 was included in order to allow the participants to add a comment about anything they thought was worth including or talking about in more depth. Therefore, as a means to shorten the length of the questionnaire while also giving them this chance, a simple "yes" or "no" question was created and they were directly asked if there was anything else they wanted to add. Those who did not would be directly sent to the end of the survey (93.45% of them or a total of 514) and the remaining 36 were redirected to another page that contained a comment box, although only 34 left a message. The complete list of unedited comments can be found in Appendix 6 and Table 46 provides some examples.

9/1/2020 3:12 PM	The second question on (sic) last page is difficult to answer generally, as it will depends (sic) on several factors such as the intended market, the budget or the country of origin. For example, mobile games will have more linguistic issues (grammar, typo, mistranslations), due to hiring cheap translators; Japanese games will have more UI issues because they rarely consider other languages during development; AAA games will end up with more inconsistencies due to having several translators on the project, including freelancers who rarely have all the resources of the in-house translators.
8/24/2020 7:54 PM	In general when it comes to video game testing and the linguistic issues found, the main issue seems to be that there is a disconnect between the translators/linguists and the actual game environment. I find that if translators were provided with more documentation, including visual documentation about the game, its characters and features, and especially for them to know the size of text boxes and the likes, half of the mistakes found would no longer occur.

8/10/2020 2:44 PM	I believe time constraints play a very big role in both translation and technical issues arising. Moreover, in a good 50% or more of the projects I've worked on, the character limit wasn't specified, which led to overlapping and/or cuts once implemented in the game.
8/5/2020 4:00 PM	One of the biggest issues in my opinion is that translators and QA testers are mostly on zero hour contracts and considered as disposable by the companies. They are underpaid and usually (sic) move on to better paid jobs and are quickly replaced. This means that a lot (sic) of translators/testers will move on mid project and new ones will be brought in. I worked for two years as a localisation tester and was on a zero hour contract the whole time. I eventually got tired of the job and am now looking elsewhere for work. The pay was also not sustainable.
8/4/2020 3:55 PM	As a Localisation QA tester, I think one of the main hurdles to overcome is a lack of communication. I rarely have the opportunity to speak directly with the translators to ask questions about choices they've made and I know translators often don't get much communication from dev (sic) either. I think a lot of issues that arise in the localisation process could be avoided by improving communications between developers, translators and localisation testers.

Table 46. Extra comments from LQA (unedited)

4.2 Video game development tools

As previously mentioned in the introduction of the chapter, the third survey was mainly created to collect technical information in order to better understand some specific points about video game development and answer some key questions. Furthermore, this phase precedes, succeeds, and runs parallel to both the localisation and linguistic testing phases, critically affecting their outcome. Indeed, developers are in charge of preparing the product before the beginning of localisation and subsequently implementing the localised strings as well as fixing the bugs reported by the testers—although not all bugs will be fixed due to either time or budgetary constraints. Therefore, due to the crucial nature of their work, comprehending the creation of a video game from their point of view and understanding in more detail some of the technical aspects as well as knowing more about the development processes, will help to subsequently evaluate the suitability of possible solutions currently available on the market taking their needs into consideration.

4.2.1 Survey's implementation and design

Although a semi-definitive version of the survey was ready by the first week of August 2020, the link used as a collector was created—and some adjustments were made—on the 12th of January 2021 as some extra time was necessary in order to evaluate the need to add further questions and to analyse the data from the previous surveys. The implementation method followed was the same as in the previous surveys, using mainly LinkedIn for individual messages and Facebook for posts on specialised pages. In the case of this survey, the completion rate rose again and reached 79% even though the questionnaire was only actively sent to potential participants during the first two months. The short implementation period resulted from inconsistencies that appeared while reviewing the preliminary data collected at the end of February as a closer inspection revealed that some participants had left bogus or prank answers (see Appendix 7 for an example) even though many did actually reach the final page of the questionnaire. Therefore, part of the time that was initially dedicated to distributing the link had to be reallocated towards screening individual answers and looking for inconsistencies in all three surveys. Eventually, only the answers of 3 participants—all from the survey addressed to video game developers—were deemed unusable and were subsequently eliminated, thus not being taken into consideration at all in any of the future references to the number of participants. However, this process proved to be extremely time-consuming and took the better part of two full months, reducing the number of potential participants and pushing back the project's deadlines (Figure 47).

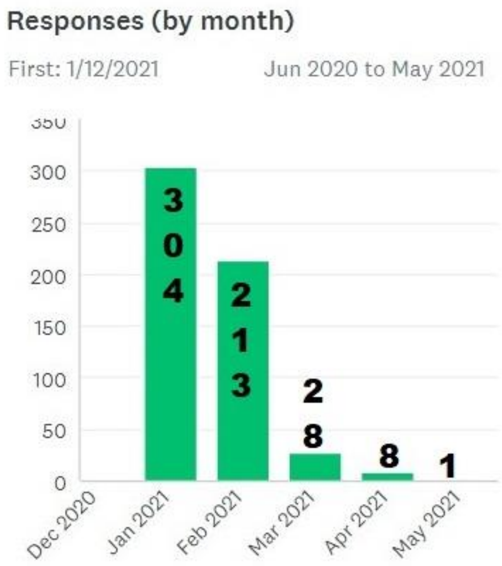


Figure 47. Number of total responses to S3 per month (developers)

As a result, the survey was officially closed on the 5th of May 2021, once the screening process was over and the number of responses that arrived per day had dwindled considerably. The questionnaire received 554 answers in total, although only 439 participants reached the final page. If we analyse the results on a monthly basis, as shown in the previous figure (Figure 47), there were 247 complete responses out of 304 in January and an extra 166 out of 213 during February. Once it was not being actively sent, the numbers went down drastically and it only received 20 full answers out of 28 in March, 5 out of 8 in April and a single answer in May. The fact of including a progress bar and reducing the number of pages helped to lessen the impact of participants abandoning in the middle of the questionnaire. Therefore, in the case of our final survey, only 95 respondents stopped after completing the second page and 20 left once they had finished answering all the questions included on page number 4 (Table 47).

PAGE NUMBER	NUMBER OF RESPONSES
PAGE 1: WELCOME MESSAGE	-
Page 2: Personal and professional information	554
(Skip question) Page 3: Sim-ship	383 (skipped 171)
Page 4: Technical data and tools	459
(Skip question) Page 5: External files	391 (skipped 163)
Page 6: Game development process	439
(Skip question) Page 7: Extra comments	19 (skipped 535)
FINAL MESSAGE	439

Table 47. S3 different pages and total number of answers (developers)

This survey, also created with SurveyMonkey, had three questions of a total of 23 that followed the “skip page option” and was divided into 8 sections split into 7 pages—including the introductory page with the message. Additionally, the average time spent was 5 minutes and 24 seconds for those who reached the final page and 4 minutes and 55 seconds if we take into account all the respondents, further improving its completion rate in comparison to the previous two surveys. Once again, the number of pages does not match the number of sections—nor does the list of topics equate to the number of questions—as those who did not work with a simultaneous shipment release model were redirected to the fourth page and did not reach page number 3. Additionally, the participants who did not isolate the text in external files went directly to the 6th page. Finally, as in the second survey, the last page was only for the respondents who wanted to add something else, which was the case for 19 developers. Table 48 shows the different sections and topics covered in them although, in this particular table, the order of the questions had to be rearranged and differs from the actual survey. These changes were applied as a means to maintain some sense of continuity and consistency with the previous surveys, even if in this case many of the questions are of a different nature and cannot be compared with the ones included in the previous questionnaires.

SECTION	TOPICS
1. Personal information	Age & nationality
2. Professional information	Studies, professional experience, main source of income, workload & type of employment
3. Business practices in localisation	Release model, start of the localisation process, isolating game text & types of external files
4. Technical data	Platforms, OS, cross-compatibility issues, coding languages
5. Tools	Game engine & XLOC
6. Attitudes towards testing	Attitudes & play-mode testing
7. Game development process	Development processes
8. Additional comments	Additional comments

Table 48. Sections and topics (developers)

Similarly to the other surveys, this questionnaire also avoided references to company names and the first two questions—respondents' age and nationality—mirrored the format and content of those included in S1 and S2. Similarly, the respondents were asked on the second page the same questions about their professional information that were included in the previous surveys. The first "skip option" question can be found at the end of page 2, where the participants were asked about the release model and, those who worked with a sim-ship system were sent to page number 3. This page contained a question designed to gather information about when the localisation process normally started in their companies. Otherwise, respondents would be sent to the fourth page, which had either checkbox questions to allow multiple answers or multiple-choice questions. This particular page was devoted to gathering technical details as well as inquiring about the game engines used for video game development and the respondents' specific attitudes (or the companies' practices) towards different types of testing.

The final question on the fourth page followed, once again, the "skip logic" format and sent the respondents to page number 5 if they used external files to isolate the translatable strings or to the following page if they did not. The respondents who indicated the latter had to leave a comment and specify what they did instead of using external files in order to be able to move on. Page number 6 contained two questions, the first one followed a multiple-choice format and inquired about the development process used in the participants' companies. The second question, similarly to the second survey, followed a simple "yes" or "no" format and if the participants wanted to leave an extra comment. Those who did want to add something were sent to the last page (page number 7) which contained a comment box (Image 23) and those who did not, reached the end of the survey.

Image 23. Comment box from S3 (developers)

4.2.2 Personal information

The first question about the age of the participants included the same possible answers as the previous surveys for three reasons (i) consistency, (ii) to collect data about underaged developers starting in the industry, and (iii) to try to correlate the results of all three surveys in the next chapter. Figure 48 shows the results and allows us to observe that none of the 439 participants who completed the survey (similarly to the previous questionnaires) was part of the last group of age and that, conversely, 3 participants were part of the category devoted to respondents who were in the less than 17 years old group. The same number of respondents (3) were part of the “55 to 64” group and only 11 of them (2.51%) were between 45 and 54 years of age. Once again, the largest group was the one formed by developers with an average age of “25 to 34” with 55.13% of the results (242 participants). The second highest result was for the respondents that were in the “18 to 24” age group with 26.42% or 116 representatives and, finally, “35 to 44” with 64 answers.

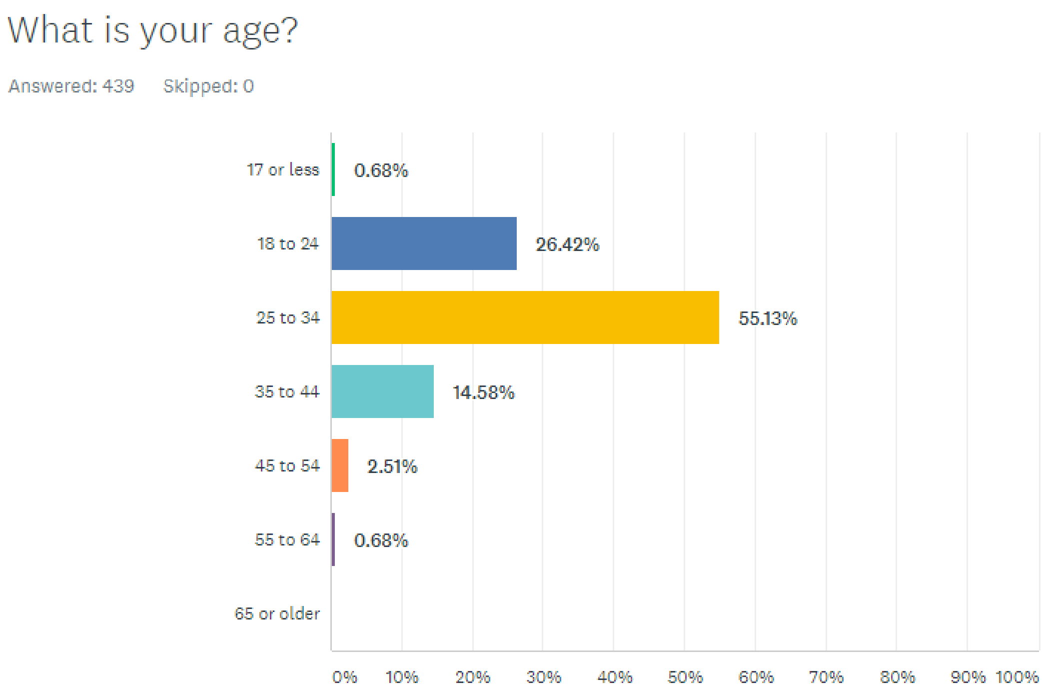


Figure 48.: Respondents’ age (developers)

Additionally, the participants were also asked about their nationality for statistical reasons as well as to maintain consistency. The question was phrased and designed following the same methods used in the other surveys and remains identical to the ones from the previous questionnaires. The results show that out of the 226 different nationalities listed, 57 had at least one representative and 10 of them had been selected 10 or more times. Anecdotally, the findings had to be adjusted as one of the participants did not seem to be able to find “British” and chose “Vatican City” instead, as he explained in the final comment. Table 49 puts together those 10 nationalities and shows that France, Turkey, and India can be found at the top of the list.

NATIONALITIES	RESPONSES	
French	26.36%	116
Turkish	9.32%	41
Indian	8.86%	39
Spanish	8.41%	37
American	4.09%	18
Italian	3.86%	17
Iranian	3.64%	16
Pakistani	3.18%	14
German	2.50%	11
Brazilian	2.28%	10

Table 49. Nationalities with more than 10 respondents (developers)

The other nationalities grouped by the number of occurrences were: Ukrainian and British, selected 8 times each; Canadian and Egyptian were the nationalities of 7 participants each; Argentine was chosen 5 times; Dutch, Greek, Mexican, Polish, Romanian, Serbian and Tunisian appeared 4 times each; Algerian, Australian, Bangladeshi, Belgian, English, Hungarian, Indonesian, Swedish, and Syrian were selected 3 times; and Chilean, Chinese, Malaysian, South African were chosen twice each. The following countries only had one representative: Armenian, Azerbaijani, Bulgarian, Colombian, Cuban, Cypriot, Czech, Dominican, Filipino, Georgian, Hong Konger, Irish, Lao, Lebanese, Lithuanian, Nepalese, New Zealander, Peruvian, Russian, Singaporean, Slovak, and Vietnamese.

4.2.3 Professional information

This section’s content remains identical to those included in the previous surveys with the exception of some minor modifications in the options provided. Thus, the different possible answers to the questions about their level of education and the types of contracts were adapted to suit the specificities of the video game development market. Consequently, the first question covered the topic of their studies and was created to evaluate their educational background and contrast their other answers with their degree of specialisation in video game development. The answers to these two questions will be correlated and cross-referenced, in a similar manner, in Chapter 5. As previously explained, the order of the questions in the actual survey and the analysis differ and, in this case, due to a technical error that could not be corrected once the collector was published, the question about their percentage of workload in development was located after the question about their type of employment instead before it; we will follow the structure from the previous questionnaires for consistency and clarity.

4.2.3.1 Educational background

The respondents were asked about their educational background in general to cross-reference their level of education and the data collected about technical details. The options included were, once again: none; master's degree; bachelor's degree; PhD; and specialised courses, seminars, workshops, etc. Figure 49 portrays the results sorted in descending order and shows that 198 of the participants (or 45.10%) had a bachelor's degree, which was followed closely by those with a master's degree—181 respondents or 41.23%. The percentage of them who did not have university studies was very low compared to the other surveys, although the main reason for these results would be the fact that we did not specify a particular field of study.

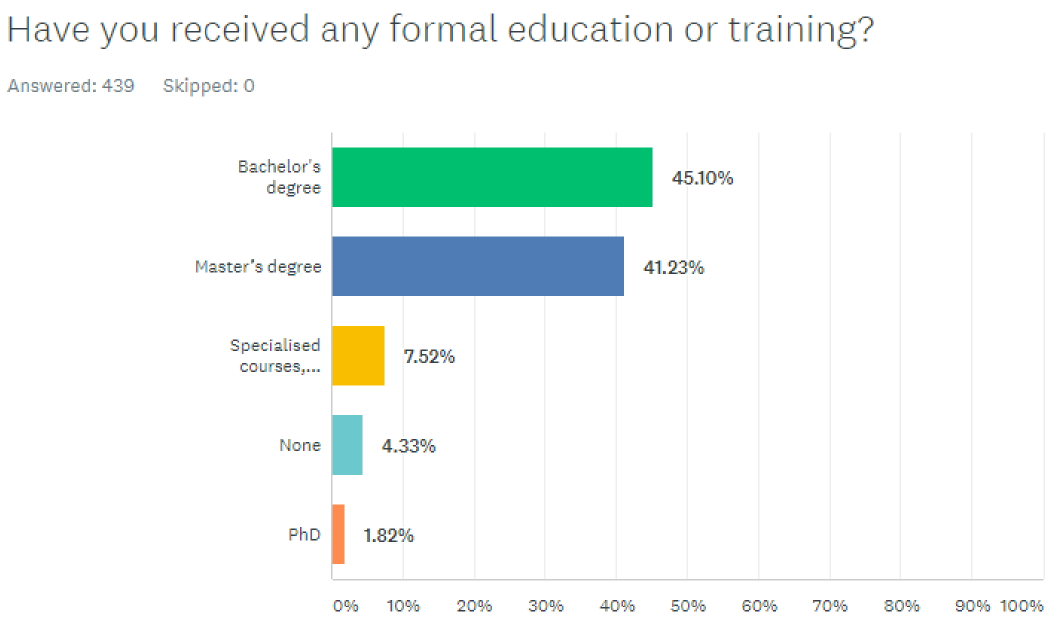


Figure 49. Educational background (developers)

4.2.3.2 Professional experience

The first question of this subsection was, as in the previous surveys, about their professional experience in video game development and provided the same options ranging from “less than 1 year” to “more than 20 years”. Similarly to the previous surveys, the option that received more answers was “1 to under 5” with 210 responses (Figure 50)—or 47.84%—followed by “5 to under 10” with 24.15% (106 participants) and, in the third position “less than 1 year” with 73 responses. The last two options received 44 and 6 answers, respectively.

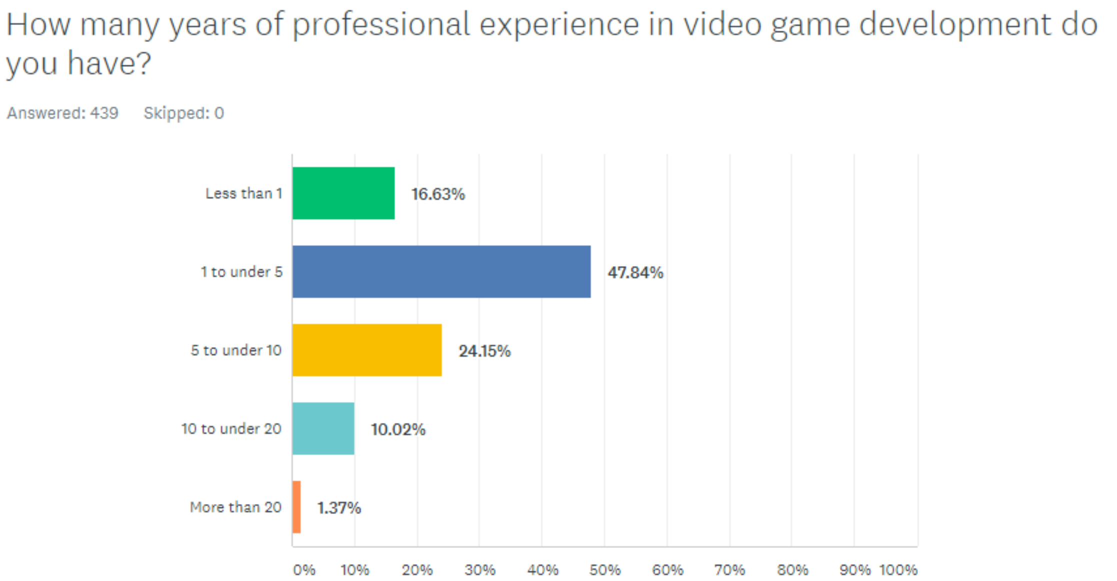


Figure 50. Years of professional experience (developers)

In order to assess their degree of specialisation in the field, they were once again asked about their main type of employment. In this case, 387 of them (or 88.15% of the total number of participants) stated that video game development was their main activity thus confirming that the majority of the respondents were experts in the field. The remainder, those who selected “no” as an answer—11.85% or 52—, were given the possibility of explaining their professional situation if they wanted to. Out of those 52 participants, 46 left a message in the section where they could state the nature of their main source of income. Anecdotally, there were 49 messages since 3 participants who chose the option “yes” felt the need to leave a comment as well. Among the most common answers, we can find 15 participants who worked in various types of development as well as 4 who specified that they were web developers and 9 software engineers. There were also 3 who stated that it was a hobby, 2 students, 1 professor and 1 teacher. Among the rest of the comments, we find “QA”, “UX”, “designer”, “military”, “sales”, “photography”, “translation”, “undergraduate chemical engineer”, and “vocal consultant”.

The results from the third and final question about how much of their professional activity was specifically in video game development show that, in this survey, the majority of the respondents had a workload of at least 50% in the field. As Figure 51 shows, 300 of them (or 68.34%) said that their main activity was video game development. Additionally, among the 44 comments left, 19 said that it occupied 100% of their time. The second most common answer was “more than 50%” with 53 answers, 51 of them chose “around 50%” and 35 selected “less than 25%”.

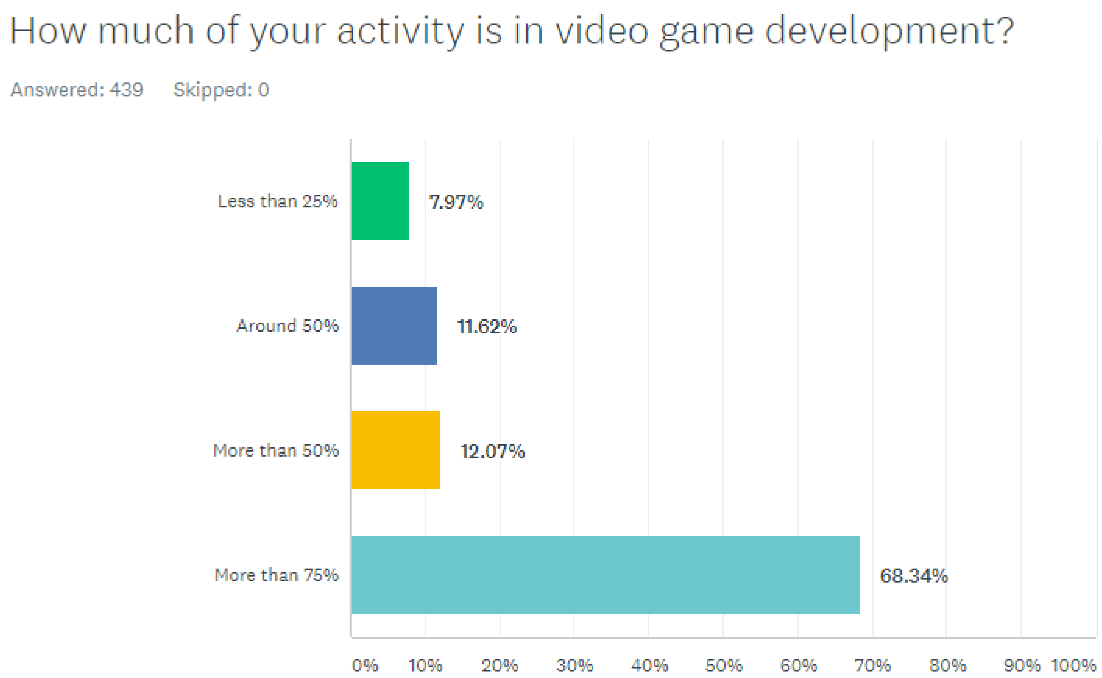


Figure 51. Percentage of activity in game development (developers)

4.2.3.3 Type of employment

In order to investigate more about the situation of the video game development market, the survey provided the option of leaving a comment following the addition of the option “other” to the list of potential types of contracts. Consequently, the question provided 6 different options: in-house developer in an independent studio, independent developer, in-house developer in a publisher-owned studio, in-house developer in a platform holder-owned studio, student and other (and the message of “please specify”). Table 50 displays the results sorted in descending order with the percentages and the absolute numbers. Upon closer analysis, the results show that 43.96% of the respondents worked as in-house developers in an independent studio (193) and 21.41% (or 94) were independent developers.

Following closely in the third position (19.13%) we find in-house developers in publisher-owned studios and, with less than 8%, students. Finally, there was a remarkably low percentage of in-house developers in platform holder-owned studios, the category only received 15 responses, less than 4% of the total. Among those whose type of employment was not among the options provided (4.78% or 21 participants), we can find 2 freelancers working for an independent studio, 4 who only wrote freelancer without commenting any further, a developer in an educational company for educational games, an owner of a game development studio, a localisation project manager in a major video game editor and publisher, an in-house developer in a studio that does contract work for other studios, an in-house developer in a work-for-hire studio and an in-house developer in an advertising company.

TYPE OF EMPLOYMENT	RESPONSES	
In-house developer in an independent studio	43.96%	193
Independent developer	21.41%	94
In-house developer in a publisher-owned studio	19.13%	84
Student	7.29%	32
Other (please specify)	4.78%	21
In-house developer in a platform holder-owned studio	3.42%	15

Table 50. Type of employment (developers)

4.2.4 Business practices

The section about business practices comprises four different questions which were not presented consecutively in the survey. Those four questions, which covered two topics, included two that followed the “skip logic system” and therefore they could not be located on the same page. In order to reduce the size of the subsections in this chapter, the question about video game development processes as well as others labelled as technical data will be covered in different sections although arguably they could also be categorised as part of this section. Thus, only the answers related to the release model and those about external files will be discussed as business practices.

The first question was located at the end of page 2 and inquired about the release model when it came to the localisation of a video game. As in the first survey, there were only two options: “simultaneous shipment” and “post-gold”. Once again, the difference between the results is stark and while only 28.02% of the participants followed the post-gold system, almost 72% worked with a sim-ship model (71.98% to be more exact). Figure 52 shows the responses of those developers who followed the simultaneous shipment release mode and reached the extra page that displayed an additional question. In this case, the respondents were asked to specify the actual build when the localisation phase tended to begin in the company or studio they worked for. The graphic corroborates the information provided in Chapter 1 section 1.4.2 as well as Chapter 2 about the localisation phase and shows that the most common moment to begin the process is during the Alpha build—this option was selected by 111 participants—followed by the Beta stage (with 74 responses or 23.34%). Additionally, 63 respondents selected “prototype build or first playable” and 41 marked “vertical slice or beautiful corner”.

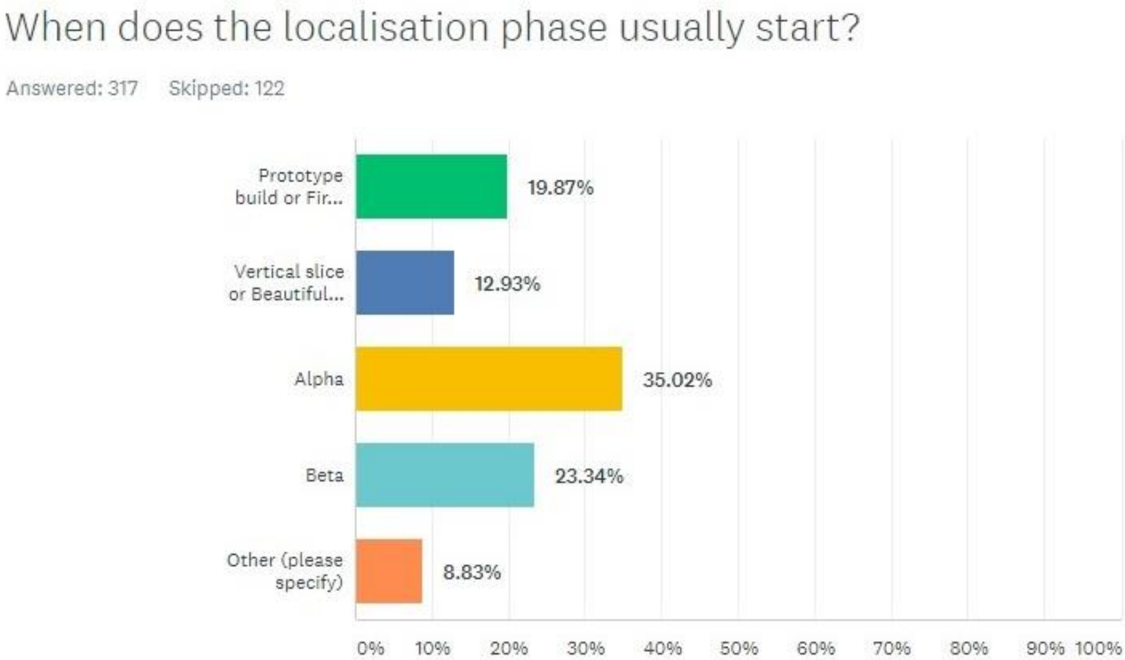


Figure 52. Beginning of the localisation phase (developers)

Those who chose the option "other" were asked to leave a message to explain their situation. All of them did, and the most representative messages (unedited) have been gathered in Table 51 as an example. In general, most of the messages left, including many not included in the table, expressed the fact that the beginning of the localisation phase would vary greatly depending on each individual project and it was difficult to provide a definitive answer. Others pointed out that it was usually late, or even very late in the development process, almost an afterthought. Finally, there is a remarkably high number of respondents who stated that they actually did not know at all when it usually started, were not sure about it or could not provide a proper answer.

4/12/2021 1:34 AM	I am not sure I think some of it happens before alpha but the bulk of it is after alpha
3/3/2021 3:05 AM	It depends on the product, usually between Alpha and Beta. We get strings translated daily for our live product. For VO, the process usually begins with alpha but can also depend on if we have non-English speaking characters.
2/19/2021 3:15 PM	Depends on the project size, sometimes alpha, sometimes almost in gold
2/18/2021 9:59 AM	Depends on how public the versions are gonna be. If it is going to be open Alpha, maybe there, but if not, maybe it will be done later. Or if the Vertical Slice needs to be shown to different publishers it may need localization too. It depends a lot on the needs of the project.
2/14/2021 10:45 AM	We're developing mobile titles and we always continue (sic) develop our games. So the answer is anytime. So answer is maybe in (sic) first year maybe 3 years later.
2/10/2021 9:49 PM	on (sic) first playable but only with 2 lenguages (sic) to check everything works just fine
2/7/2021 12:29 PM	At (sic) the polishing phase of the final product.
2/2/2021 2:12 PM	Too late. Depending on the production, it ranges from Beta to Production. This should be in Pre-production alongside other pipelines.
1/22/2021 12:39 PM	It's not always Beta, but it's late in the process

1/22/2021 12:20 PM	It's an MMORPG, so it varies. Usually, there is a "pretranslation" process while the next update is still in progress.
1/20/2021 10:03 PM	During development, before alpha.
1/18/2021 3:31 PM	Largely depends on the short-term goals of the company/project. Usually around vertical slice or Alpha though.
1/15/2021 11:33 PM	whenever (sic) a narrative (sic) is finished and validated. Usually start around Alpha but can shift up to the beta.
1/12/2021 12:49 PM	Game as a service. Loca is updated before each update (usually every 1-2 months)

Table 51. Comments about the beginning of the localisation phase (developers)

The second topic covered in this subsection was the methods used in order to isolate the translatable strings from the game code to facilitate localisation and avoid mistakes. The initial question was located at the end of page 4 and the second with the “skip logic” to clarify the type of files, was on page 5. The participants were asked whether they either used external files or data storage systems to isolate the text. Those who chose the latter option were asked to specify what types of methods whereas those who selected the former were sent to an extra page with a list of file formats. Only 60 participants (13.67%) used data storage systems, and, among their comments, we find numerous references to external files, which implies that the question might have proved to be opaque for them. Their responses were: Confidential or do not specify (20), Google Drive Sheets (10), Unity Scriptable Objects (7), UE4 system and tables (7), I2 Localisation (2), Amazon CloudFront, CSV, XML, Perforce, SQL, .txt files, BBDOC files, HeidiSQL, and YMAL.

Additionally, if we analyse the replies provided by 86.33% (379) of the participants who stated using external files to isolate game-play text from code (and were directed to page 5) we can observe that the majority selected .json files. Indeed, as Figure 53 shows, 65.17% or 247 participants selected .json files, 36.68% (139) marked XML and 29.82% (or 113) chose Excel. Among the 45 respondents that used other types of files, there were many who said that they did not know and 11 said that it was confidential (or did not actually specify). The rest of the comments mentioned: CSV (14), .po (4), TSV files (3), LUA files (2), JS, and RCK bank files.

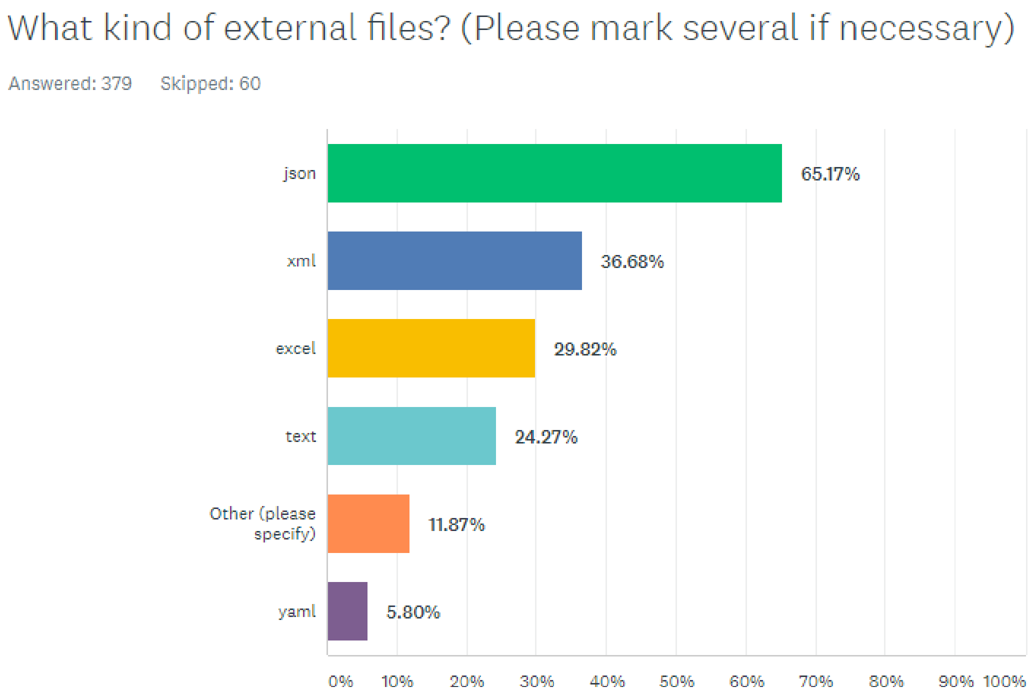


Figure 53 Types of external files (developers)

4.2.5 Technical data

As previously mentioned in the introduction of this thesis, one of the initial hypotheses is the possibility of developing an all-encompassing and comprehensive tool specifically designed for video game localisation that takes into account every step in the process, from development to production. Thus, the following questions were mostly designed in order to evaluate that possibility or to use the responses gathered to analyse already existing tools if necessary. Consequently, developers were asked a series of key questions to identify the most common platforms, their OS system, if they used emulation programmes, and the programming languages most commonly used.

The first question of this section was about the platforms used to play the video games they created. The respondents had to choose from a list of the current types of platforms available. Some of the respondents expressed their dissatisfaction (in private messages) with the fact that they were not allowed to provide multiple choices but, unfortunately, it was not possible to change the format of the question once launched. Figure 54 shows that the option with the highest number of answers was "mobile games" with 35.54% of the responses, closely followed by "downloaded/boxed PC games" with 32.57% or 143 responses. The third option was console games with almost 20% of the answers.

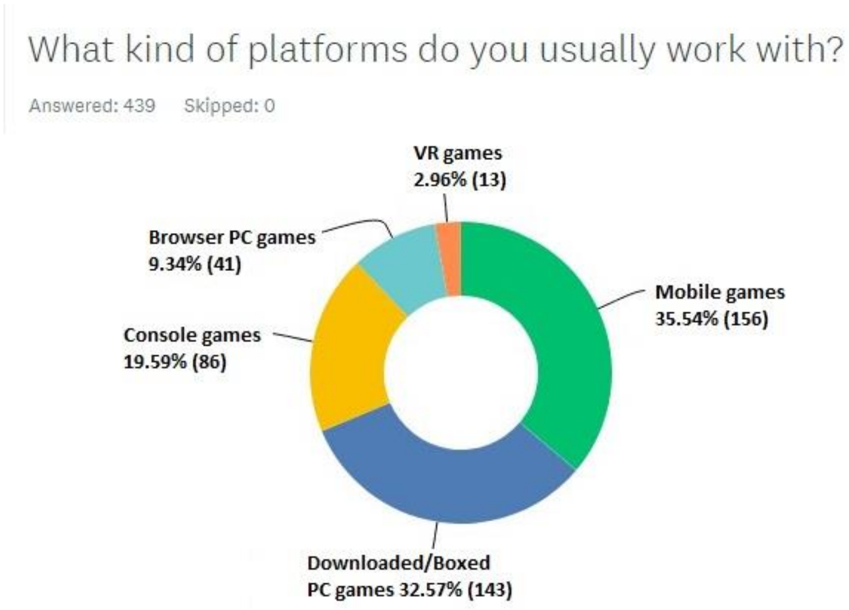


Figure 54. Main game platforms (developers)

The participants were also asked about the operating system that they used on a regular basis. Unsurprisingly, Windows was the most widely used OS with 88.15% of the total (387 responses), followed by Mac OS with 9.34% (or 41), and Linux (1.14% or 5). Among those who chose "other" we find two that stated using both Windows and Mac; one replied "Windows at work, Linux for personal projects, MacOS occasionally"; another one said, "all of them". There was a participant that answered "html5" although when analysing the rest of the responses provided in the other questions it proved to be a one-time mistake. Additionally, there was also a comment about a previous question in the survey unrelated to the one being analysed.

Furthermore, the developers were asked if they needed to use system emulation to facilitate cross-compatibility within the development team. Table 52 includes all the answers collected both by the question itself and those in the "other" section—as 25 participants selected it only to add "none" in the comments and 2 of them wrote "custom under NDA" instead of choosing the appropriate option from the list. Some of the participants specified the name of the game engine they used to indicate that they did not need to use these kinds of systems as they all worked using the same tool: 4 said they used Unity, 1 used Roblox, another one CoreGames and one said they used Sony Development Kit. There was also a participant who said that they worked with Google Sheets (also a simple one-time mistake).

EMULATION TOOLS	RESPONSES
None	338
Other (please specify)	38
A proprietary tool I can’t name for NDA reasons	35
Virtual Box	23
Docker	19
VM Ware Workstation	13
Android emulator	1
QEMU	1
WSL	1

Table 52. System emulation tools (developers)

In order to investigate the programming languages video game developers used on a regular basis, the participants were provided with a multiple-choice question with some of the most common ones and they could either choose one of them or select “other” and then leave a comment. Since the question was not created to allow multiple answers and many respondents left a comment instead, the following list presents the results from both the question and the comments: C# (259), followed by C++ (153), Html5 & JavaScript (14), Java (5) and Python (4). Among the rest of the comments that were left in the option “other” we find 3 that did not know, 4 references to Haxe, 2 mentioned the visual scripting option of Unreal Engine 4, and 2 more for Lua. Go, Objective-C, PHP, a proprietary language, Actionscript 3, JS, Flash and Bolt were only mentioned once.

4.2.6 Video game development tools

The question about the tools used was crucial in order to assess the possibility of using plug-ins or CMS systems such as XLOC. Therefore, the participants were asked, firstly about the game development engine they used and secondly, if they knew about XLOC and what they thought of it. Figure 55 shows that the most widely used engine is Unity with 312 responses (or 71.07%) of the total. Unreal Engine received 32.35% with 142 users and, as is common in the industry, 18.22% of the respondents were not able to provide the name of the engine used due to NDAs. The last three options were: CryEngine with 7 responses, Source with 3 and Unigine with 2. The question included Frostbite, Fox Engine and Creation Engine as well, but they were not selected at all.

Among those who chose "other", 11 of them did not want to disclose or simply specify the name of the tool they used even though they were not subject to confidentiality clauses. Other programs that were mentioned more than once were: Construct 2/3 which appeared 5 times, Godot was also mentioned 5, Cocos2d-x (3 times), Heaps-Haxe 3 times as well, and Phaser also 3 times. Those that were mentioned only once were: CoreGames, CoronaSDK, Gamemaker Studio 2 Engine, Hide, Irrlitch, JavaScript Canvas, JNGL (own C++ Engine), Libgdx, Love2D, Lumberyard, Monogame, Pixi.js, Ren'py, RGL, Roblox, Snowdrop, Three.js, UIKit, WebGL Frameworks, Acknex Engine, Adobe Air, amethyst (Rust), Asura, BabylonJS, BigWorld, Buildbox, and Cicos2D.

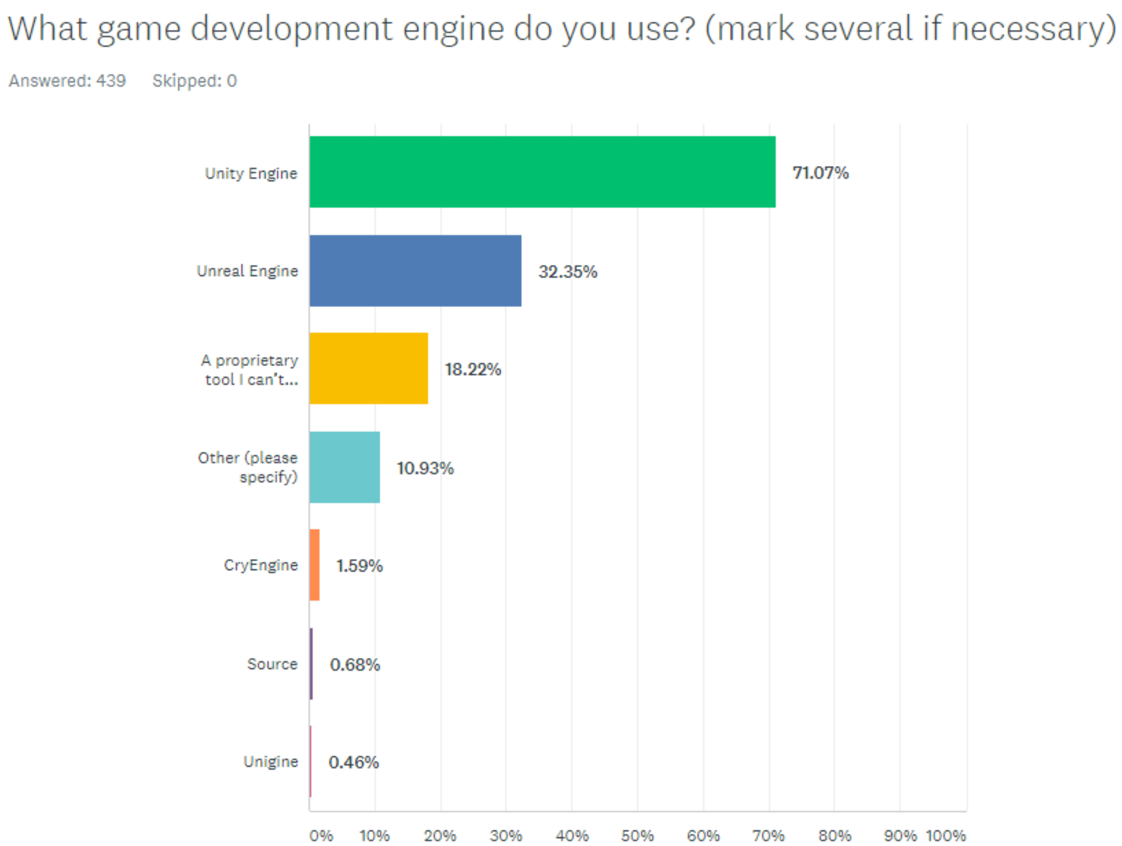


Figure 55. Game development engines (developers)

The first survey only included XLOC as an asset extraction and integration tool, a content management tool, or a project management tool. However, while analysing the results of the first and second surveys, some of the comments implied that it was more than a simple CMS and that it included features related to localisation, playing a more important role in the industry than initially attributed to it. Therefore, right before launching the collector of the present survey, we decided to include a last-minute question about it in order to collect some information about the degree of adoption of this tool as well as the developers' opinion about it. Thus, the respondents were provided with 5 options: “Never heard of it”, “I've heard of it but I never use it”, “I use it whenever I can”, “I use it occasionally”, and “I've heard of it but I don't like/want to use it”. Surprisingly after researching more about the tool and its features and as Table 53 shows, the vast majority of the participants (388 of them or 88.38%) had never heard of XLOC. The percentage of those who had heard about it but never used it (or had never used it) was very low (less than 10.5%) and only 5 participants used it either regularly or occasionally. The last provided option did not receive a single response.

ANSWER CHOICES	RESPONSES	
Never heard of it	88.38%	388
I've heard of it but I never use it	10.48%	46
I use it whenever I can	0.68%	3
I use it occasionally	0.46%	2
I've heard of it but I don't like/want to use it	0.00%	0

Table 53. Degree of adoption and attitudes towards XLOC (developers)

Additionally, the respondents were invited to leave a comment about the tool but following the low percentage of adoption, the participants only left 7 comments in total. Table 54 puts together all of them including the messages that were not relevant. Nevertheless, those who had used it and left a message with actionable content seemed to be pleased with the system although such a reduced sample does not allow us to draw conclusions as it seems to be a tool mostly utilised by the localisation department.

4/12/2021 1:39 AM	I worked at EA localization and I remember one game using xloc 9 years ago... and they did not want to use our toolset... to be fair xloc was certainly good
3/26/2021 2:56 PM	Because I translate myself
3/3/2021 3:10 AM	I have heard of it. I think our localization team uses it, but I do not personally for my position.
2/23/2021 11:10 AM	Use in-house publisher solutions
2/14/2021 2:07 AM	Interesting, I've worked with Keyword Studios, but never heard of this.

2/12/2021 11:31 AM	I've never published a commercial game that needed languages other than Turkish & English
1/15/2021 2:31 PM	Makes things a lot easier to manage text databases in several languages and follow-up textu pdates (sic) and additions. However this is handled by the localsiation Department, many development teams have never heard of it.

Table 54. Comments about XLOC (developers)

4.2.7 Attitudes towards testing

This section was created in order to evaluate the existence of testing methods or systems already in place (besides those for LQA) that could be taken advantage of to provide localisers with a WYSIWYG environment. Therefore, the first question was about their attitudes towards testing the code and, as Table 55 shows, it seems to be more of an afterthought in the case of 41.69% of the respondents whereas 146 respondents do not use unit or functional tests. Only 19 participants state that even though they do not cover all the code, they create tests that are very "defensive" (that test for all possible errors instead of those that are likely) and 20.73% of them try to cover 100% of the code.

ATTITUDES TOWARDS TESTING	RESPONSES	
We try to add testing but it's more of an afterthought	41.69%	183
We don't use unit/functional tests	33.26%	146
We aim for 100% code coverage	20.73%	91
We write very defensive tests	4.33%	19

Table 55. Developers' attitudes towards testing

The second question focused on play-mode or screen comparison style testing, which would be the most useful testing type for localisers since they would be able to see their translations in-game. Although games tend to have many functional bugs during the first builds, having access to user interfaces and (potentially) menus would prove to be of great help and save a lot of time for the industry. In order words, being able to see the translated string in-game would help to drastically reduce the impact of overflows, overlapping and truncations, as well as decrease the time localisers spend coming up with solutions to shorten the length of strings. Therefore, the participants were asked about the importance they put on play-mode or screen comparison style testing in a multiple-choice question that had 4 options: We don't use them, we cover 100% of gameplay, we use them in vital or complex areas, and our existing testing methods do not allow for the possibility. Figure 56 shows that 38.72% of them (or 170) did not use them at all, 25.74% said that they covered 100% of the gameplay, 22.78% only used it in vital or complex areas, and 12.76% stated that their existing testing methods did not allow them to use them.

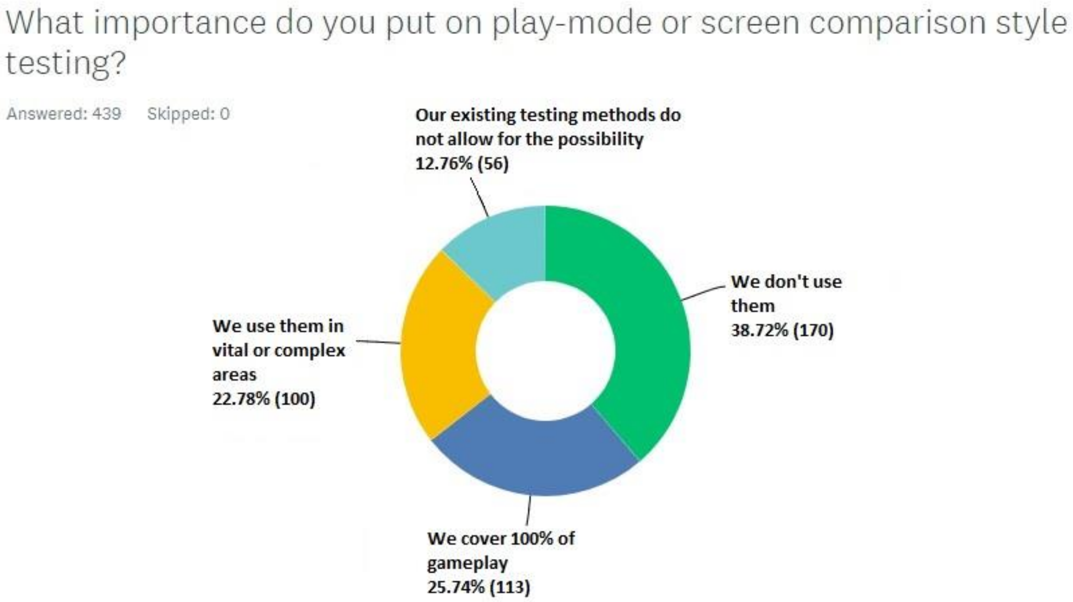


Figure 56. Play-mode testing (developers)

4.2.8 Game development process

The last question before reaching the end of the survey and asking the participants whether they had something else to add was about the game development process. This particular question was mainly included in the questionnaire in order to assess the possibility of creating testing environments with visual access, a scenario that becomes more likely if the respondents used a method that would rely on ready-to-use features and short development cycles. Consequently, the participants were asked—by the means of a multiple-choice question—if they used an iterative method, a hybrid method, a waterfall method or if their methodology was ad-hoc. To decrease the chances of confusion and limit the number of opaque answers, we also included the descriptions of each type as presented in Chapter 1 section 1.4.4. The rest of the options were (as usual and for consistency reasons): “I don’t know”, “can’t say because of an

NDA", and "other". Table 56 displays the results—with the full description for better reference—sorted in descending order and includes both the percentages and the absolute numbers.

GAME DEVELOPMENT PROCESS	RESPONSES	
Iterative: is a process that consists of developing software by repeating short cycles to deliver a ready-to-use feature each time. Agile software methodology follows this iterative approach, improving continuously and systematically its processes and practices.	47.84%	210
Hybrid: is a combination of waterfall and iterative processes in the same project. Typically, the waterfall strategy is used during pre/post-production and the iterative is applied during the production phase.	27.56%	121
Waterfall or predictive: is a sequential process in which the next phase is started only if the previous phase is completely finished, delivering business value all at once. This is the traditional game development process, requiring explicit requirement assessments followed by orderly and precise problem-solving procedures.	10.71%	47
I don't know	10.71%	47
Can't say because on an NDA	9.11%	40
Ad-Hoc: is a process that is created only for a specific project, without a previous definition. In the ad-hoc process, activities are defined on demand and the process changes to respond to punctual and contextual issues.	7.52%	33
Other (please specify)	1.59%	7

Table 56. Game development process (developers)

Following the trends in development in utility software, 47.84% of the respondents (or 210) stated that their game development process was iterative (described in Table 54) and 27.56% replied that it was hybrid. Only 10.71% of them selected waterfall, the same percentage that chose "I don't know". The 7 participants that chose "other" left messages of diverse nature: one said that they did a bit of everything depending on the size of the project, another that they tried to follow a Scrum mindset, and another specified that it was agile with scrum framework, and the last one said “mostly agile” (all different ways of saying iterative). One only said “2” instead of choosing the second option (we assume), another wrote: “I think the answer is Hybrid, but looks like Chaos sometimes” and the last one said: “Do not want to precise it”.

4.2.9 Extra comments from the developers

The final question located on page 6, as in the survey addressed to linguistic testers, was included in order to allow the respondents to add a comment if they thought there was something that needed to be explained in further detail. Thus, the participants were directly asked if there was anything else they wanted to add and they only had to choose between “yes” or “no”. Those who did not have anything else to say were directed to the end of the survey (which was the case for 95.67% of them or a total of 420) and the remaining 19 were sent to another page that contained a comment box. Table 57 shows the messages that provided actionable information as many just left their e-mail offering to help more or to answer any other future questions.

2/22/2021 11:08 AM	Since I work at a AAA(A) game developer and publisher we use a proprietary tool that makes it easier to manage translations. The tool works with our source control and Jira as well.
2/18/2021 10:53 AM	Solo projects are managed in a different way than team ones. Also, the scope and the type (and the budget) of the project dicate (sic) how (and if) it's localized. Projects that are heavy in localization needs, such as entire genres like point and click games or generally narrative games, tend to not be localized because they are niche games and the costs are too high because of the nature of the game, even when the team would like to do it.
2/14/2021 3:32 PM	We tried using tools, such as Weblate, which simplify both our worlkflow and the translators'. However, the localisation companies (we outsource localisation) we work with, they don't want to hear about tools. They use csv (or tsv) files and that's it. They have it how they like it. So regardless of how much time and effort they would save, we had to revert to using good ol' csv files.

2/5/2021 6:34 AM	Now as a (sic) overall gaming industry is moving towards VR/AR/MR. Which I feel is not as creative as gaming. As mostly we do port or do some simple stuff to project it on this platform instead of focusing more on creating new. Hoping for the game industry stands (sic) on its own sooner or later
1/28/2021 11:39 AM	Used the tool Localizor for most of the translation part with the help of the community.
1/26/2021 10:58 AM	As a contractor I'm working with a lot of different studios and I noticed the localization tools are often really basic and error prone (often just an excel or Google Doc). That's why I decided to create my own tool (somewhat similar to XLoc), with UE4 integration. I'm planning to open-source it soon. Contact me if you want to know more.
1/13/2021 6:36 PM	One thing to keep in mind is that studios may have internal staff performing localization or they may be outsourcing it to experts which may affect tool usage. At a previous job we used an internal database system that could export to excel files that would be used to work with outsourced localizers. Good luck on your phd! If you need more info feel free to contact me. I'm not an expert but have worked closely with ui and localization in the past.
1/12/2021 12:59 PM	My nationality (British/UK) was not listed, so I put Vatican instead. I work full-time for a mobile game studio, and in the evenings have my own indie games studio. Some of the games I work on are for both console and pc and mobile, but the question did not allow multiple answers. I use i2 localisation plugin for Unity both at work and for my indie games. It works great! Bit fiddly to set up by (sic) once working, it is ideal for a small and medium sized projects, which is all I ever work on.

Table 57: Extra comments from the developers (unedited)

CHAPTER 5. ANALYSIS OF THE RESULTS AND DISCUSSION

Although Chapters 3 and 4 displayed the raw results from the surveys, it is necessary to make a side-by-side comparison to identify patterns. Therefore, this chapter will present the cross-analysis of the results of all three surveys and classify them into 4 different categories: respondents' profiles, business practices, key questions and technical data, and tools analysis. The first section will contrast the data related to the participants' profiles, both personal and professional, including their ages in order to identify patterns, a comparison between the respondents' nationalities and their native languages (and main source and target languages when applicable). Afterwards, we will analyse their educational background, the duration of their professional careers, study the differences in the percentage of workload depending on the respondents' main source of revenue and finally, compare the results about the types of contracts.

In the case of the cross-analysis of the information about business practices, we will proceed to study their working conditions and identify problematic situations that may have an impact on the incidence of linguistic bugs. Thus, the section will provide an overview of how different types of contracts affect the localisers' access to reference material and the game itself. The third section will analyse in-depth the questions that were at the heart of each survey: localisers' attitudes towards features and functionalities, the prevalence of linguistic bugs and their causes, the development process, the beginning of the localisation phase in the case of developers as well as how the technical data collected in the third survey about technology may be utilised to improve the situation of localisers and testers. Finally, we will study the impact of the educational background and the type of contract on the degree of adoption of new technologies and compare our findings with those obtained by other research in the field of translation in general. Furthermore, we will review the features included in the most widely used tools, their integration capabilities, and carry out a cross-analysis of said features with the results extracted from the key questions. Section 5.5 will provide an overview of any additional findings that stemmed from our research and could not be included in any of the previous categories.

5.1 Respondents’ profiles

The current section will present the analysis of all the data collected from all three surveys about the respondents’ personal and professional information and it will be divided into three subsections: respondents’ age, nationality and working languages; respondents’ studies and professional experience; and type of employment. In the case of working languages, we will only proceed to compare the results obtained in the first two surveys since the respondents of the third questionnaire were not concerned with linguistic matters.

5.1.1 Respondents’ age, nationality and working languages

When we analyse the results about all the respondents’ ages (Table 58), we can observe numerous similarities despite the fact that they come from different fields. In all three surveys, the figures show that the group with the highest number of participants is always the one that brings together those between the ages of “25 to 34”. Furthermore, when we observe the differences in numbers with the age group that appears in second place, we can see that the gap is considerable and constitutes almost half of the figures obtained for the first group (in the case of the linguistic testers and developers) or more than half in the case of video game localisers. Therefore, the total number of participants from all three surveys between the ages of “25 to 34” represents 946 out of the 1609 respondents that were taken into account as they had reached the end of their respective surveys.

Chapters 1 and 2 provided us with an overview of the video game industry from a historical perspective that helps to understand this data. Although *Pac-man*—released in 1980—is believed to be the first case of localisation, the prevalence of this age group results from the fact that the professionalisation of the field and the adoption of localisation practices as a recognised part of the process (albeit still an afterthought) did not become a common occurrence until the end of the ‘90s (see Chapter 2 section 2.3.1). Therefore, chronologically, the figures obtained by localisers and by extent, linguistic testers, are consistent with the development of the video game localisation industry since the “25 to 34” group includes those individuals that grew up and received their education while the industry was getting established. The same principle applies to the second most common group in both the first and the second survey, the respondents that were between “35 and 44” years old. In the case of developers, even though the first group follows the same pattern and is easily explained by historical developments, we observe that the results for the second and third positions are reversed. Thus, in comparison to the other two surveys, we observe that the second most common age range among the respondents to the third survey is “18 to 24”. This change in the pattern that resulted from the previous two surveys is a clear reflection of the growing popularity of the industry worldwide and the increase in the number of video game companies. This popularity also leaves a mark in the results of the other two surveys and places the respondents that belong to the “18 to 24” years of age category in the third position, which represents half, or nearly half, of the results from the category immediately above.

The rest of the options included in the questions all share the same place in the ranking regardless of the survey, showing a low percentage of participants that belonged to the “45 to 54” category. In order to see if they were just individuals who specialised in the field late in their careers or part of what we could call “the pioneering group” (at least in the case of localisers and localisation testers), we filtered the results about their professional experience. In the case of localisation, 15 participants out of the 28 had more than 20 years of experience (and 11 stated “10 to under 20”) thus confirming that they were among the first in their field. There were no respondents with the same amount of experience in the second survey: 5 selected “10 to under 20” and 7 had between “5 and 10”. Only 2 developers had chosen “more than 20” and 5 indicated “10 to under 20”. In all three surveys, the percentage of participants in the “55 to 64” age group is less than 1%, the same number of developers that chose “less than 17” (that survey is the only one with respondents of that age who reached the final page of the questionnaire). Even though the number of participants was remarkably low, in order to see if the surveys received answers from what could be considered the “groundbreakers” of the three aspects of the industry that are being covered by this thesis, we decided to filter the results once more. Almost all of the localisers (4) had more than 20 years of experience (the remaining participant chose “5 to under 10”); all three testers had more than 20 years of experience and, in the case of developers, two selected “5 to under 10” and only one “more than 20”.

LOCALISERS				TESTERS				DEVELOPERS		
25 to 34	55.81%	346		25 to 34	65.09%	358		25 to 34	55.13%	242
35 to 44	29.03%	180		35 to 44	20.73%	114		18 to 24	26.42%	116
18 to 24	9.84%	61		18 to 24	10.55%	58		35 to 44	14.58%	64
45 to 54	4.52%	28		45 to 54	3.09%	17		45 to 54	2.51%	11
55 to 64	0.81%	5		55 to 64	0.55%	3		55 to 64	0.68%	3
-17	0%	0		-17	0%	0		-17	0.68%	3
+65	0%	0		+65	0%	0		+65	0%	0

Table 58. Comparison of all respondents' ages

The rest of the subsection will deal with the remainder of the questions about personal data for localisers and linguistic testers in order to contrast information about languages and the country of origin of the respondents. First, Table 59 portrays the combined results of the localisers' nationality, native language, and target language to obtain a clearer view. We decided to omit the column for source languages due to the fact that the answers mainly showed that English was the predominant language (either as a source language or as a pivot language) with 543 responses and the results obtained by the rest of the languages in the list—with Japanese (28) and Mandarin (14) as the only ones with more than 10 answers—were too low to add value to the comparison. Therefore, the first position is occupied by Spanish in all three categories with slightly lower figures in the column for nationalities due to the fact that part of the respondents also came from Latin-American countries, thus selecting them instead of Spain but choosing Spanish as a native language. The same principle can be applied to the second position, as French is the official language of various countries and all nationalities were included in the list. Additionally, we cannot ignore the presence of bilingual respondents which would explain, for example, the slight increase in native speakers of Italian in comparison with the number of nationals.

The first discrepancy appears once we reach the fourth position, where we can find Brazilian as a nationality but English as both native and target language, this is most likely due to the presence of respondents that were either bilingual or trilingual as well as the fact that there are various countries with English as an official language. The results for German remain consistent although, from this point forward, we observe a significant decline in the numbers related to nationalities (compared to languages) that increases the difficulty in finding correlations and analysing the causes of these discrepancies in some cases. As mentioned in Chapter 3, the reason for the presence of Catalan in the list is the high percentage of Spanish respondents. Following the emergence of other localisation practices besides the E-FIGS combination, we can also observe the presence of languages such as Russian, Mandarin, Polish and Turkish. In the case of this last language, we can perceive a connection between these results and the fact that the second most common nationality among developers was Turkish, which may well hint towards a growing market that might become increasingly important in the near future.

NATIONALITIES		NATIVE LANGUAGE		TARGET LANGUAGE	
Spanish	113	Spanish	158	Spanish	149
French	107	French	121	French	111
Italian	69	Italian	72	Italian	68
Brazilian	46	English	63	English	52
German	43	German	49	German	47
Turkish	23	Portuguese	47	Portuguese	46
Russian	22	Russian	36	Russian	33
Argentine	20	Turkish	25	Turkish	22
American	16	Catalan	24	Mandarin	14
Polish	13	Mandarin	22	Polish	14

Table 59. Comparison of localisers' nationality, native language and target language

Finally, when we carry out the same comparison using this time the results of the second survey and contrast linguistic testers' nationalities and native or working languages, we only find a concordance between the results of both columns for two languages: Brazilian-Portuguese and German. The former is due to the fact that those who chose Brazilian as a nationality chose Portuguese as a native or working language as we did not provide the option of choosing Brazilian as a language. The remaining elements hold different places in the ranking depending on the column and the tendency is to have more results for the nationalities than for the language speakers. In the case of Spanish and Russian, the difference in their position in the ranking's different columns is only one, although Spanish is the only language that had more speakers than nationals in the top ten, thus becoming the exception to the above-mentioned trend (for the top ten only). The difference between the positions in the two columns widens and becomes two positions in the cases of French and Italian (and follow the trend of counting with more nationals than speakers) and three positions in the case of Polish (also more nationals than speakers).

English appears undisputedly as the first native or working language with 579 responses, although the first nationality from an English-speaking country is in the 10th place, a result that can only be explained by the fact that the participants were allowed to provide multiple answers. This was confirmed when we decided to filter the results using only those who chose “English” (as either their sole answer or one of them) and saw that it was selected by 412 participants. When we analysed the results for their country of origin, in the first 5 positions for nationalities we find that 11.41% were French (47), 11.41% were Italian (47), 8.75% were Polish (36), 8.01% were from Brazil (33) and 7.52% were Spanish (31). Only 90 testers stated that they were from a country that has English as an official language (or one of them). To conclude, the last two working or native languages on the list are Japanese and Chinese (Mandarin) even though the number of nationals was not high enough to enter the top ten (they were both chosen only 8 times each as a nationality) demonstrating once more the high impact of bilingual or trilingual respondents (Table 60).

NATIONALITIES		NATIVE OR WORKING LANGUAGES	
French	97	English	579
Italian	81	Spanish	165
Spanish	81	French	153
Polish	64	Italian	89
Brazilian	56	Portuguese	70
German	35	German	59
Russian	21	Polish	57
Turkish	19	Russian	41
American	18	Japanese	36
Canadian	17	Chinese (Mandarin)	29

Table 60. Comparison of testers’ nationality and native language

5.1.2 Respondents’ studies and professional experience

When we analyse the respondents’ educational background, we can observe that the majority of the localisers and the developers had either a master’s degree or a bachelor’s degree. In the case of linguistic testers, since the question specified the type of studies as well, the highest-ranked option was "none". Nevertheless, this result does not exclude the possession of other types of university diplomas as they were only questioned about a degree in either translation or languages and they were not given any other option for the rest of the possibilities. As explained in Chapter 4, these results are due to the fact that LQA testing tends to be used as an entry-level job in the industry for developers or even producers. Another contributor is that there seems to be no specific university curriculum available for the position at the moment. Additionally, as explained in a participant's comment included in the presentation of the results in Chapter 4, even though they work in LQA, in many cases they either review the translations according to a pre-existing list of conditions without actually being experts in the specific language or they work with their native languages without specific linguistic training. Table 61 shows all the results from all three surveys and allows us to observe that 388 localisers and 387 developers stated having gone through university (in translation in the case of localisers) compared to 193 participants in the case of linguistic testers (for translation or language studies).

Localisers		**Testers**		**Developers**	
MA in translation	236	None	287	BA	198
BA in translation	142	MA in translation, language studies, etc.	104	MA	181
None	140	BA in translation, language studies, etc.	84	Specialised courses, seminars, workshops, etc.	33
Specialised courses, seminars, workshops, etc.	92	Specialised courses, seminars, workshops, etc.	70	None	19
PhD in translation	10	PhD in translation, language studies, etc.	5	PhD	8

Table 61. Educational background of all the participants

The analysis of the combined results about the respondents' years of professional experience (Table 62) shows that, once we compare the figures side by side, the most common answer was "1 to under 5 years". This result proves to be slightly inconsistent with the answers about the participants' ages at first glance if we consider that graduates usually enter the market at age 23 on average and in localisation and LQA most participants were either “25 to 34” or “35 to 44” years old. However, in the case of the first age group, it suggests that the respondents were mostly in the lower half of the age span and, when we consider both groups together, the results can also be explained by the fact that it remains fairly difficult to enter the industry at first and that it takes time to fully specialise in it.

When we move on to the second row of the table, we find that the correlation and progression from this point forward remain consistent with the participants’ ages in the case of localisers as the number of them that were between “18 to 24” years of age is low. However, in the case of LQA, the cross-analysis shows that a high percentage of them just started in the field since, after filtering the results, most of them were part of the “25 to 34” age group. Among those who stated having “less than 1 year” of experience, we can also find that a high percentage of them did not work exclusively in linguistic quality assurance and had a different main source of revenue. This situation was actually the case for 43.18% (or 57) out of the 92 with less than one year of experience, which could point out to a shift in their role or a change of their careers in recent months.

The fact that “5 to under 10” years of experience appears in the second position for developers and that “less than 1” appears in the third position also shows a discrepancy with the respondents’ ages if we consider that the category of “18 to 24” was second in the participants’ age ranking. Therefore, we filtered the results to clarify the answers and found that most of the participants in the “1 to under 5” category were between “25 and 34” years old (131) followed by “18 to 24” (64) which points to difficulties to get into the industry in the first case (there were also 15 respondents above 35 years of age) and a fairly high degree of developers starting early in the industry in the second case. This also explains why “5 to under 10” appears second, as it included 84 participants from the “25 to 34” group, 15 above 35 years of age and 7 under the age of 24; two of whom chose “less than 17”, which means that they started very young, during their teen years (and there was no indication of bogus answers). Finally, the majority of those who selected “less than 1 year” were “18 to 24” years old (46), followed by “25 to 34” (22 participants), then “35 to 44” (4) and “less than 17” (1).

LOCALISERS			TESTERS			DEVELOPERS		
1 to under 5	34.03%	211	1 to under 5	52.43%	216	1 to under 5	47.84%	210
5 to under 10	32.10%	199	Less than 1	22.33%	92	5 to under 10	24.15%	106
10 to under 20	23.23%	144	5 to under 10	16.99%	70	Less than 1	16.63%	73
Less than 1	5.65%	35	10 to under 20	7.28%	30	10 to under 20	10.02%	44
More than 20	5.00%	31	More than 20	0.97%	4	More than 20	1.37%	6

Table 62. Comparison of professionals' experience

Unsurprisingly, when we filter the results about the participants' percentage of workload in their corresponding field using their main employment as the base (Table 63), those who selected that either localising, linguistic testing or video game development was their main source of income also stated that they mostly worked in those areas more than 75% of the time. In the case of LQA and video game development, the resulting percentage of specialists is higher than in the case of video game localisation. Indeed, we can observe 72.07% and 76.23% of participants selecting this option in comparison with only 58.68% in the case of localisation. As it happens, the results are consistently more mitigated both for the respondents whose main source of revenue was video game localisation and for those who expressed otherwise—the respondents who selected less than 25%.

These results concur with the high percentage of localisers working as freelancers in contrast with the prevalence of in-house contracts in the other fields since, as independent workers, they are more prone to accepting other types of translation projects. When we only analyse the figures for those who had a different main activity, we can clearly see a high percentage in the results for "less than 25%" in the case of LQA (almost identical to the opposite side of the scale). In the case of localisers, the difference between "less than 25%" and "more than 75%" is almost negligible as it amounts to only 2 participants. As for video game developers, even though the percentage of those who selected "less than 25%" is above 50%, the results are less polarised than in the case of linguistic testers (the rest of the options were almost always under 10%) and we can observe a slightly higher percentage of non-specialised developers working in games around 50% of their time.

	LOCALISERS			TESTERS			DEVELOPERS	
Main source of income	Yes 530	No 90		Yes 333	No 217		Yes 387	No 52
Less than 25%	20.00% 106	43.33% 39		11.41% 38	72.35% 157		1.81% 7	53.85% 28
Around 50%	13.77% 73	12.22% 11		7.21% 24	11.06% 24		9.56% 37	26.92% 14
More than 50%	7.55% 40	3.33% 3		9.31% 31	4.15% 9		12.40% 48	9.62% 5
More than 75%	58.68% 311	41.11% 37		72.07% 240	12.44% 27		76.23% 295	9.62% 5

Table 63. Workload by main source of income

5.1.3 Type of employment

As mentioned in Chapter 2 section 2.2.4, the video game industry (following the trends already present in translation in general) tends to favour freelance work and outsourcing models in the case of game localisation. Thus, 435 participants belonged to one of the three categories related to freelancing that were provided in the first survey against 152 who worked in-house either at a language provider or a development company (Table 64). Conversely, the results for both linguistic testers and video game developers show that the trend is to work in-house. In the case of LQA, we can find 392 participants working for either a development company or a translation company and 144 who were freelancers (divided into the same categories provided to localisers).

Furthermore, after analysing the comments left by the developers who selected "other" as an answer, labelling those messages as either "freelance" or "in-house" and adding the results to the rest of the options provided to them, we find 300 developers working in-house in any of the multiple types of companies present among the replies whereas only 102 worked independently. In the case of students, we find the highest numbers among developers, corroborating once more the increasing popularity of the field in recent years which goes along with the fact that it was the only survey that actually had participants in the "less than 17" category. However, even though students appear in the fourth position (instead of last in the case of the other two surveys) the percentage remains remarkably low compared to the figures obtained by the options that were in the top 3. Nevertheless, the result is still higher than the number of "in-house developers in a platform holder-owned studio" (15) either pointing to the scarcity of those positions or the prevalence of NDAs forbidding the respondents to take part in this research.

LOCALISERS		TESTERS		DEVELOPERS	
Independent freelancer	28.06% 174	In-house in a non-translation company	47.27% 260	In-house developer in an independent studio	43.96% 193
Freelancer working both independently and with an agency	25.32% 157	In-house in a translation company	24% 132	Independent developer	21.41% 94
Freelancer working with an agency	16.77% 104	Independent freelancer	11.27% 62	In-house developer in a publisher-owned studio	19.13% 84
In-house translator in a translation company	12.42% 77	Freelancer working with an agency	7.64% 42	Student	7.29% 32
In-house translator in a non-translation company	12.10% 75	Freelancer working both independently and with an agency	7.27% 40	Other (please specify)	4.78% 21
Student	2.90% 18	Student	2.55% 14	In-house developer in a platform holder-owned studio	3.42% 15

Table 64. Comparison of all respondent's types of contracts

5.2 Business practices

This section will analyse data related to the working conditions of all the participants involved in this study. To begin with, in the case of the first two surveys, the respondents were asked about teamwork to evaluate their need for project management systems, communication tools, etc. The results of both questionnaires show that linguistic testers mostly work in teams and as explained in Chapter 3, teamwork is also common among localisers if we combine the two results that included the possibility of collaboration (although the answers were varied if they are analysed separately). In the case of video game developers, these practices will mainly depend on the size of the project (although it is extremely rare to cover all aspects of a high-end game single-handedly) and whether they work independently or not. However, as explained in Chapters 1 and 2, the process of developing a game usually involves many actors that cooperate closely in order to achieve the final product and some games may require the collaboration of multiple studios. Therefore, communication within the team, compatibility of both formats and tools, version control in the case of assets and game builds, as well as simply tracking changes are crucial to the success of the project. Furthermore, as all the surveys concentrated on different phases of the same process, communication among all the partakers regardless of the field is essential as well, even more in the case of the involvement of publishers, LSPs and the subsequent addition of more links to the production chain.

The need for tracking changes and version control only gets more pressing when we cross-reference the results of the first and third surveys regarding release models. We observe that 69.68% of the localisers and 71.98% of the developers stated working with a simultaneous shipment model and beginning localisation during the development phase. Therefore, avoiding sending outdated strings to be localised to reduce both costs and extra work or providing localisers with a fast way of detecting changes in the source (or pivot) language becomes essential. Working with translation memories (if they are properly maintained) does solve part of the issue by means of fuzzy matches. However, if the modifications are not properly indicated in the document (or database) that contains the text to begin with, those strings might not even reach the translator as, depending on the game, the quantity of text can reach "440,000 words with 30,000 VO lines" (Mangiron Hevia and O'Hagan, 2013, p. 143) as explained in Chapter 2. If we add pivot languages to the mix, a new layer of complexity appears that calls for an even tighter control of all text changes in order to avoid surprises.

When we analyse this particular aspect from a technical point of view and study the methods used by developers to isolate in-game text and compare said methods to the types of files localisers tend to receive, we can identify another problematic step in the process that might lead to human error: the need to manually copy and paste. Chapter 4 showed that 86.33% of the developers (379 of them) used external files to isolate game-play text from code and when they were asked about the type of files they used, 65.17% (or 247 participants) selected .json files. However, as presented in Chapter 3, only 7 participants from the first survey mentioned receiving .json files when performing video game localisation (and those only appeared in the comments section), whereas Excel files received a total of 528 responses. Therefore, these results suggest that the content of those files, once they are "cleaned" to remove the code and extract the translatable strings (albeit leaving the tags and variables), gets copied and pasted into Excel files. This practice increases the chances of implementation errors (among other potential mistakes) and adds a cumbersome task to the localisation phase that can be easily removed by improving tools' connectivity and taking advantage of APIs and plug-ins.

Furthermore, as presented in Chapters 3 and 4, localisers do not tend to have access to the video game itself, contrary to linguistic testers and, evidently, developers. In their line of work, they are mostly only allowed access to an Excel file that will contain the string ID (which sometimes can provide valuable information), the source text, the column for their translation, and an extra column with comments or instructions provided by the person in charge of managing the localisation process or the developers themselves. This practice creates a scission between the environment and the text itself, as mentioned repeatedly throughout Chapter 2, which is only worsened by the lack of linearity in the text and the absence of in-game context. Although in the case of text linearity the only way to help localisers would be to add a comment in the file, reference material can solve the lack of context. As presented in Chapter 3, the reference material the client may provide might include: images or screenshots, videos, glossaries, characters' descriptions, bibles, bios, descriptions, design documents, past translations, translation memories, and sometimes previous game builds.

Localisers specified in their comments that the availability and quality of the material itself depended on the project itself. However, another crucial variable would be their status within the work chain. Therefore, in order to have a comprehensive view of how the type of contract impacts the availability of reference material as well as access to the game itself for localisers, we decided to filter the results using the different contract options as a base. Furthermore, besides analysing the results obtained for reference material, we decided to include the data collected about the availability of video files and the different options included in the case of localising menus or user interfaces: visual access to the game itself, visual access and the maximum number of characters, just the maximum number of characters, or simply no indication at all (Table 65).

	Reference material	Video files	Visual access	Visual and characters	Number of characters	Nothing
All types of contracts	76.61%	55%	14.84%	14.84%	46.94%	23.39%
Independent freelancer	77.59%	48.28%	15.52%	9.20%	47.70%	27.59%
Freelancer working with an agency	71.15%	45.19%	7.69%	11.54%	59.62%	21.15%
Freelancer working both independently and with an agency	76.43%	41.40%	7.64%	13.38%	52.23%	26.75%
In-house in a non-translation company	92.02%	89.33%	42.67%	29.33%	18.67%	9.33%
In-house in a translation company	84.42%	81.82%	9.09%	19.48%	50.65%	20.78%

Table 65. Localisers' access to material, videos, or visuals according to employment

Thus, Table 65 presents the combined results for access to reference material, access to video files and all the different options included in the specific question about access to the visual environment in the case of menus or user interfaces to avoid overflows, overlaps and truncations. All this data has been classified considering every type of contract and the overall results presented in Chapter 3 have also been included to provide more context (referred to as "all types of contracts"). These will be called from this point forward the "average" to facilitate the analysis and for an easier way to compare each individual result. Therefore, the average result for access to reference material suggests that it is a common practice to provide some type of help to localisers, even though the quality and quantity vary depending on the project (see Appendix 4 for all the comments left by the participants about the topic). We can observe that independent freelancers are slightly above the average (almost an extra 1%), freelancers working both independently and with an agency are also above said percentage, although the difference is minimal. Unsurprisingly, the highest scores are for the participants who work in-house: 92.02% for those who work directly for the developer and 84.42% for those working in-house in a language vendor.

Conversely, the participants who work as freelancers for an agency—which according to the results receives a fair amount of reference material (84.42%)—fare the worst and are barely above 70% (even though the result remains reasonable). In the case of video files for subtitling purposes, the average result shows that in general, the practice is less common than the previous one, and only those participants who work in-house are above the average (55%) This result is not to be confused with video files in general as the comments that mentioned videos as reference material also included walkthroughs and trailers. In addition, we can observe that the differences between the results are stark, ranging from 41.40% to 89.33%. Thus, the percentage of access to video files in the case of localisers working in a non-translation company more than doubles the results of those at the other end—freelancers working both independently and with an agency, once again obtaining the lower figures.

The average result for visual access—which means access to the game itself but not necessarily access to viewing changes “on the fly”—is remarkably low and does not even reach 15%. In this case, the only two categories above said average are in-house localisers in a development company (with 42.67%) and independent freelancers (15.52%). Thus, access to the game itself does not seem to be quite common as even those with the highest score do not reach the 50% mark. The differences among the rest of the categories are almost inexistent (in the case of freelancers) and in the case of in-house translators working for language vendors, the percentage is below 10%. Additionally, the average result for “the best-case scenario”, which would be having access to the game itself and knowing the maximum number of characters that can be displayed, is identical to the previous one. However, once we observe the individual results for each category, the differences among types of contracts become very clear. As we can observe, although the professionals who work for the development company still maintain the highest percentage, the resulting percentage falls under 30% and loses more than 10% (in comparison with the previous column). Conversely, those working in-house for an agency add 10% to their result (19.84%) and the localisers working both independently and for an agency almost double their access and score 13.38%. Furthermore, those working as freelancers for an agency gain almost an extra 4% (11.54%) and, in this case, the worst results are those obtained by independent freelancers with only 9.20%.

All the categories but one are above the average when it comes to the practice of receiving the maximum number of characters. In this case, the exception is the localisers who work directly for the developer—a potentially direct consequence of having access to the actual game for reference. The best result belongs to the category of freelancers working with an agency, followed by those working both for an agency and independently and, in the third position, those who work in-house in a language vendor. Independent freelancers are just above the average with less than 48%. Finally, the average of localisers not receiving any kind of access or instructions when it comes to menus or user interfaces is 23.39% and the only two categories that are above that figure (which in this case would mean obtaining the worst results) are freelancers who work both independently and with an agency (26.75%) and independent freelancers (27.59%). The best results are those obtained by in-house localisers who work for the developer (9.33%), followed by those who work also in-house but in a language vendor (20.78%) and, thirdly, freelancers who work exclusively for translation agencies (21.15%).

5.3 Key questions and technical data

This section will present the questions that provided crucial information for our research and constituted the heart of each survey. First and foremost, the question conceived to analyse the localisers' attitudes towards features and functionalities will allow us to evaluate and establish what they consider essential in order to carry out their work as well as detrimental. Secondly, the cross-analysis of the most common linguistic bugs and their causes compared with business practices in both localisation and video game development will let us pinpoint potential courses of action as well as sources of friction. Finally, the results from the questions of the third survey about the game development process, the beginning of the localisation phase as well as the developers' attitudes towards testing will help us to investigate more comprehensive solutions available on the market. Additionally, they will provide data that will enable us to research different means to procure more context for localisers and reduce the impact of certain types of bugs, especially those that can be avoided by simply including some sort of "preview" feature that would almost immediately eliminate overflows, overlaps and truncations.

5.3.1 Localisers' attitudes towards features and functionalities

As previously mentioned, this question was central to the first survey as it provides us with a unique insight into video game localisers' needs in terms of tools' features, their opinions on those already present and what they consider to be lacking. Furthermore, in combination with the comments left about missing features, we will be able to elaborate a list that will allow us to analyse the tools that are currently available on the market. Additionally, thanks to the filtering options, we will try to identify differences in said preferences according to their type of employment.

5.3.1.1 Essential and useful features and functionalities

In the case of the features and functionalities that were considered essential by the localisers and presented in Chapter 3, we can observe that the first position is occupied by "the possibility of seeing dialogues in order" which received a total of 421 responses. The results from the first survey confirmed that, even though it is crucial to have access to the context that surrounds a sentence in order to translate it appropriately, only 37 participants stated that the strings they received were always in order thus justifying the need to select this option as the most essential one. The combined results of "always", "most of the time" and "often" are only slightly above 50% (54.52%) leaving a high percentage of localisers (45.48%) having to work out where the sentence belongs on top of finding the most suitable translation. These practices also explain the fact that, as mentioned in Chapter 3, 79.03% of the localisers indicated that tree-based tools for dialogues would be a useful addition to the tools they already use. The inclusion of these types of systems would dimmish the effect of this intrinsic characteristic of video game localisation explained in Chapter 2 section 2.2.3 and its impact on storytelling.

The second feature and functionality selected as "essential" was "terminology management and extraction tools to ensure consistency" with 403 answers. This result proves to be more complex than it seems at first glance. On the one hand, the fact that localisers, testers, and developers work in teams most of the time plays a considerable part in the need to use tools to ensure the consistency of the translation of the document. The fact that the industry has fully adopted the simultaneous shipment release method (as well as the increasing popularity of the "game as a service" system as mentioned in

Chapter 2) only complicates the task of maintaining said consistency. Another factor, as explained by at least one linguistic tester, is the fact that the localisers might change before the end of the project and, in the section about missing features, some participants stated that the translation memories they received were not modifiable or kept up-to-date.

On the other hand, when we analyse the degree of adoption of terminology extraction and management tools that results from the first survey, we observe that these systems are the second less popular ones among localisers either as standalone programmes, in the case of corpora or even integrated into the computer-assisted translation tool. The results showed that only 141 participants used them at least occasionally (39 selected "always" and 102 chose "sometimes"). However, this contradiction is not exclusive to the field of video game localisation. As explained in Chapter 2, previous research in the field of technology adoption (Zaretskaya *et al*., 2018, p. 46; 2015, p. 250) has already established that these tools tend to remain largely unused. Conversely, O'Brien *et al.* (O'Brien *et al*., 2017, p. 145-162), while presenting the results from the ErgoTrans survey, received 18 comments categorised as "missing features" about "TM/Terminology", implying once more the desire for these features. Therefore, our results confirm those obtained by both studies and show a pronounced dichotomy between what professionals want and the actual use of the features provided by the market.

In the third position with 314 responses we find "access to audio, video and images", once again a clear statement about the need for more context in order to carry out their work. As analysed in the section about business practices from this chapter, although localisers tend to have access to reference material often (76.61% of them do) we only found around 80 comments (273) that mentioned either images or screenshots, 50 about videos and 1 about audio. Additionally, the majority of those comments explained that it would vary depending on the project, thus indicating a potential longing for guidelines standardising these practices. In the case of video files for subtitling purposes in particular, we saw that only 55% of the respondents stated receiving them regularly. This need for visual access is not exclusive to the field of video game localisation as it extends to all types of audiovisual translation as well as translation in general. In the same article written by O'Brien *et al.* (2017, p. 145-162), the authors also explained that out of the 329 comments that the respondents left as an answer to the features they found missing from CAT tools in general, 12% of them "mentioned the need for a preview feature to see the final layout of the target text while working on the translation" (O'Brien *et al*. 2017, p. 159). Furthermore, in the comments about missing features left by the participants of this study, as presented in chapter 3, we found 12 references to the need for "more visual access" and 10 more that explicitly mentioned "access to the game itself". Thus, both our research and O'Brien's confirm the crucial importance of avoiding the scission between the text and the product itself.

"To include quality assurance tools" received 278 and ended up in the fourth position and, in the comments section about what the current tools in the market were missing, 9 participants specified that they needed spellcheckers or grammar engines in particular. In the above-mentioned article, 12% of the respondents also included comments about lacking quality assurance features and the authors concluded that "these features are already implemented in some of the tools, but as these results show, they are not a standard feature in every CAT tool" (O'Brien *et al*. 2017, p. 159). The fifth result considered essential with 273 responses was "computer-assisted translation tools" which have proven to be the translation technology most widely adopted in the results of the present study. These results are backed by those gathered about translation in general included in Chapter 2 section 2.1.3 as well as those presented in the 2020 European Language Survey report that stated that "TM still [is the] most popular tool" and specified that "Memsource, memoQ and SmartCAT [are the] most frequently mentioned CAT products" (LIND 2020, p. 51-52).

When we analysed the combination of the features and functionalities that were considered "essential" and "useful" in Chapter 3, we observed some changes in the order of the elements on the list as well as some additions to it. We saw that "access to audio, video and images" switched places with "terminology management and extraction tools to ensure consistency" and became second. These results highlight the localisers' need for more access and reflect what Bernal-Merino said about video game localisation being a "double-blind process" (2013, p. 119). The last two items, that were not present when we only considered "essential" features, are the "possibility to track any changes in the source text files (management tools)" and, in fifth place, "access to all the assets in their original form and divided by formats". Although the fourth feature is mostly related to the constant changes brought up by simultaneous shipment (as indicated by 80.09% of the respondent to the first survey), the latter is also due to the lack of context and access that characterises the video game localisation industry.

5.3.1.2 Inconvenient, not important, and not so useful features

At the top of the list of the features that were marked as "inconvenient" stands "machine translation" with 170 responses (136 more than the second feature on the ranking). This high percentage, compared to the results obtained by the rest of the features and functionalities listed, shows a predominantly negative opinion about machine translation. Thus, we decided to look into the rest of the possible answers for this particular tool and observed that it was also chosen as "not important" by 154 participants and "not so useful" by 195 (a total of 519). These figures are also reflected in the fairly high percentage of participants that stated "having heard about MT but not using it" (50.81% or 315) and some of the comments left by the 305 participants that did use it. Among those comments, we find one respondent that said that the system he or she employed "depends on what my clients use as it is linked to the CAT tools. Most of the time is just a hindrance". Another one said "proprietary FAANG MT systems, but they still suck" and a third one just said: "EA's piece of shit software". There were also numerous comments stating that they only performed post-editing and they did not know the name of the tool since the translation was directly done by the agency while others said that they had to use MT because the client demanded it. O'Brien *et al.* labelled this as "imposed usage" (O'Brien *et al*. 2017, p. 152), a phenomenon confirmed by our research that tends to result in a negative opinion of the tool since the professionals are not able to choose if they want to use it or not and are forced to comply.

The second less popular feature was "the possibility to combine corpora, CAT tools and machine translation" with 34 answers. Although it received exactly a fifth of the responses that MT did in the case of "inconvenient features", it still ended up in second place. When we observe the rest of the results it was also selected as "not important" by 91 participants and as "not so useful" by 169; this brings the total of negative answers to 294. These results derive from the presence of machine translation among the options and the fact that tools for corpus compilation in general remain a largely unadopted technology as shown in Chapter 3. “Online CAT tools that allow working in teams” was selected as “inconvenient” 13 times, 28 times as “not important” and 68 as “not so useful” (a total of 109 negative answers). Although CAT tools are progressively becoming a web-based service and new systems such as SmartCat, Crowdin, XTM or Wordbee gain popularity in the market, personal experience has shown that certain online CAT tools suffer from connection issues often. These issues may range from being unable to sign up to having problems validating translated segments, which can become painfully slow depending on the number of members on the project.

“Computer-assisted translation tools” was selected 9 times as “inconvenient” and became the fourth option on the list. Additionally, 28 respondents said that it was “not important” and 60 that it was “not so useful” (97 in total). These results place the fifth feature that was selected as “essential” as the fourth less liked one as well. Although the numbers are relatively low compared to the rest of the options and translation memory systems are consolidated in the market as the most widely used translation technology, “they are still a source of irritation to their users” (O’Brien *et al*. 2017, p. 145). Among the reasons for this negative attitude towards them we can enumerate that they tend to be technology-driven, sometimes lacking user-friendly interfaces, and characterised by steep learning curves. Other possibilities could be, once more, the imposed use that we mentioned in the case of machine translation.

The fifth option considered “inconvenient” was “being able to compile corpora” although it only received 6 responses. Additionally, 77 respondents marked this feature as “not important” and 161 considered that it was “not so useful” which brings the total to 244 respondents who did not quite care for this functionality in particular. Among the reasons for these results stands the fact that the video game industry is characterised by a lack of official resources online in general as well as the lack of access to in-game content both before and after the release of the game. Besides non-disclosure dispositions, the contracts tend to include clauses to specify that, once the translation is finished, the company will own the translation memory and the localiser will not be able to take advantage of it for future work. Theroine *et al*. (2021) analysed the availability of suitable material online in order to evaluate the possibility of compiling corpora to create a non-specialised bilingual comparable corpus in English and French. Their research proved the lack of readily available resources when they tried to create a corpus specifically for video games using the web as a source, which yielded a high number of repeated results in the case of the second language.

Thus, obtaining pertinent material (and without too much code) for a corpus might become a time-consuming task in this field, an issue that is connected to another problem in the industry: the impact of time constraints. Freelancers must pay special attention to the time they spend on each project in order to make a profit and to try to compensate for the late decrease in rates brought by the health crisis as well as trends in the market. This is not exclusive to translation and extends to every field that deals with a high percentage of freelancers (which proved to be the case for the respondents of the first survey). However, in the case of video game localisation, localisers' capacity to dedicate time to the project at hand is also hindered by the tight deadlines that are characteristic of a simultaneous shipment release model, thus reducing the time that can be allocated to tasks such as compiling a corpus or creating glossaries from previous translations using terminology extraction and management tools.

5.3.1.3 Variations in the results depending on types of contracts

In the case of the 174 participants who worked as independent freelancers, there were no changes in either the order or the elements that were chosen as "essential": "The possibility of seeing dialogues in order" (110), "Terminology management and extraction tools to ensure consistency" (102), "Access to audio, video and images" (75), "To include quality assurance tools" (71) and "Computer-assisted translation tools" (69). However, when we observe their results for "inconvenient" features and functionalities, "Online CAT tools that allow working in teams" does not appear in the top five and is replaced by "being able to use Corpora". "Machine translation" remains first with 42 responses, followed by "The possibility to combine corpora, CAT tools and machine translation" with 12. "Being able to use Corpora", "Being able to compile Corpora", and "Computer-assisted translation tools all received 4 answers each. These results mean that almost half of those who selected "the possibility to combine corpora, CAT tools and machine translation" or "being able to use Corpora" were independent freelancers. Finally, they represent 4 out of the 6 that selected "Being able to compile Corpora" as "inconvenient", thus strengthening the previous statements about time constraints and freelancers in the case of video game localisation. In other words, the previous hypothesis about time constraints in the case of working with corpora for the participants that are paid depending on the number of words present in the source text (as opposed to those who work in-house) is confirmed by these findings.

Among the 157 freelancers working both independently and with an agency, in the case of the "essential" results, CAT tools jumped to the third position: "The possibility of seeing dialogues in order" (122), "Terminology management and extraction tools to ensure consistency" (112), "Computer-assisted translation tools" (76), "Access to audio, video and images" (74), and "To include quality assurance tools" (70). On the opposite side of the list, we find that "Subtitling tools for spoken dialogues and cinematics" and "Bug reporting tools", both with 3 responses, appeared in the fifth position. The rest is similar to the combined results: "Machine translation" (52), "The possibility to combine corpora, CAT tools and machine translation" (12), "Work on online CAT tools that allow to work in teams" (6), and "Computer-assisted translation tools" (4). Once again, we find almost half of the responses about "The possibility to combine corpora, CAT tools and machine translation", "Work on online CAT tools that allow to work in teams" and "Computer-assisted translation tools" in this group. In this case, we find the reflection of both the need to allocate resources to maximise the profit and the potential impact of the "imposed usage" (O'Brien *et al*. 2017, p. 152) of certain tools by the language vendors the localisers work with.

The 104 participants that were freelancers working with an agency also had similar results for the "essential" features although CAT tools are in fourth position instead of fifth: "The possibility of seeing dialogues in order" (76), "Terminology management and extraction tools to ensure consistency" (74), "Access to audio, video and images" (64), "Computer-assisted translation tools" (54), and "To include quality assurance tools" (50). In the case of "inconvenient features", we find that "Access to all the assets in their original form and divided by formats", "Possibility to track any changes in the source text files", and "Bug reporting tools" received 1 response and made the top five. The rest of the list remains the same: "Machine translation" (31), "The possibility to combine corpora, CAT tools and machine translation" (7), and "Work on online CAT tools that allow to work in teams" (2). The analysis of these results reveals many similarities with the previous freelance categories that have been discussed above and shows the need to save time and the impact of not being able to choose the tool they have to work with.

The changes become more apparent once we start analysing the results for the 77 localisers who work in-house in a translation company. Terminology management becomes first, and quality assurance takes the third place: “Terminology management and extraction tools to ensure consistency” (55), “The possibility of seeing dialogues in order” (49), “To include quality assurance tools” (41), “Access to audio, video and images” (40), and “Computer-assisted translation tools” (34). On the other side of the ranking, only 4 features received negative responses: “Machine translation” (29), “The possibility to combine corpora, CAT tools and machine translation” (2), “Computer-assisted translation tools” (1), and “Work on online CAT tools that allow to work in teams” (1). In this case, the figures for MT are lower than in the previous categories and, since not many features were marked as “inconvenient”, the results are not representative enough to discern clear patterns.

Finally, the results for the 75 localisers who work in-house in a non-translation company show that 55 of them chose the “Possibility to track any changes in the source text files” as essential and it became the first one on the list. Additionally, QA disappeared from the top five: “Possibility to track any changes in the source text files (management tools)” (55), “The possibility of seeing dialogues in order” (45), “Terminology management and extraction tools to ensure consistency” (44), “Access to audio, video and images” (38), and “Computer-assisted translation tools” (31). In this case, even fewer features were deemed “inconvenient”: “Machine translation” was only selected 13 times in total, “Access to all the assets in their original form and divided by formats” received 1 response, and “The possibility to combine corpora, CAT tools and machine translation” was also only chosen once. In this case, the most interesting findings are among the features that were deemed as “essential” and show that, since they start working on the project the moment parts of the text are made available, their most pressing need is to be able to see all the changes and act accordingly. Furthermore, having more time to familiarise themselves with the game itself and the technocentric aspect of the company they work for might also have an impact on their adoption of translation technologies, a fact that will be studied in-depth later on in this chapter when we analyse tools’ adoption individually.

5.3.2 Linguistic bugs

This subsection will contrast the results provided by the key questions included in the second survey with those obtained in the first survey to add weight to the findings and observe the consequences that localisation business practices have in LQA. Table 66 shows both the linguistic bugs in order of prevalence (from left to right) and the causes the testers identified for each one of them sorted in descending order.

Overflows, overlapping, truncations	**Mistranslations**	**Terminology inconsistencies**	**Subtitling errors**	**Confusing instructions**
Lack of visual environment (284)	Lack of in-text context (224)	Lack of resources (183)	IT-related issues (150)	Lack of context in-text (162)
Insufficient instructions or failure to follow them (90)	All of the above (92)	All of the above (133)	Lack of visual environment (125)	Lack of visual environment (119)
All of the above (83)	Lack of visual environment (88)	Too many translators on the same project (115)	Don't know (79)	Insufficient instructions or failure to follow them (81)
Other (72)	Insufficient instructions or failure to follow them (65)	Insufficient instructions or failure to follow them (80)	All of the above (62)	All of the above (77)
Don't know (12)	Lack of translation quality (53)	Other (20)	Insufficient instructions or failure to follow them (57)	Don't know (40)
Lack of translation quality (9)	Other (22)	Don't know (19)	Lack of translation quality (45)	Lack of translation quality (39)

Table 66. Causes of all linguistic bugs present in the survey

When we analyse the cause that appears at the top of the list for each bug, we find that "lack of in-text context" appears twice, both as the main cause for "mistranslations" and for "confusing instructions". This issue derives from the practice of sending the strings that need to be localised in Excel files without internal order (as well as the lack of visual cues from the game to deal with ambiguity) that has been presented all along this thesis consistently. Furthermore, as examined in the previous subsection about localisers' attitudes as well as in Chapter 3, the number of localisers that face this situation remains fairly high (282 or 45.48%). One solution to this issue would be to take advantage of the possibilities that many tools provide (including Excel files) to filter the segments during the extraction and integration process or to change the practice of initially arranging the strings in the files by sections or characters. The second possibility, and the reason for the particular interest of this study in integration capacities, would be to automate the asset extraction and integration process via APIs (application programming interface, some sort of intermediary to connect two applications and exchange data) or plug-ins. In other words, setting up a system that allows the different tools in the chain to "pull" data directly from one programme to the other and to "push" it back afterwards, will reduce manual labour and might help change the developers' approach to arranging strings.

"Lack of visual environment" appeared as the main cause for "Overflows, overlapping, and truncations", the third in the case of "Mistranslations", second for "Subtitling errors", and second as well for "Confusing instructions". In total it appeared four times in the top three causes of every type of bug the respondents were asked about except in the case of terminology inconsistencies (since it was not part of the options provided for that question). As shown in the previous chapters and sections, access to the game itself is rare in general regardless of the type of contract. The results showed that it was made available only for 42.67% of the localisers who worked in-house in a video game development company, even though the initial assumption was that they would always be able to access the game. Moreover, the combined results for access to the game and "both visuals and maximum amount of characters" were below 30% for all types of contracts (29.68%). Having said this, it is important to take into account that access to the game itself does not mean being able to see the translated string in context. In order to be able to see translations "on the fly", localisers would need to have the means to translate the segment, validate it, integrate it into the game and then find the precise location of that particular line in the game, unless there is an automated system in place as explained before. Nevertheless, either access to the game for reference or being able to integrate strings to check for bugs could be counterproductive timewise if the localiser has to go through the entire game to solve a doubt about a single sentence. The following sections will explore the potential solutions to this problem by analysing the features offered by game engines, among other things.

In the case of terminology inconsistencies, we find that according to linguistic testers the main cause is the "lack of resources". Moreover, among the testers that selected the option "other", some of them explained that many inconsistencies that appear in the translated text might actually be due to problems in the source text and originate from inconsistencies created by constant last-minute changes or additions to the text itself, a direct consequence of simultaneous shipment practices. From the localisers' point of view, even though they confirmed in the first questionnaire that they do receive resources often, their comments in that particular question only included 39 references to glossaries in particular and 20 to either past translations or translation memories. In these two particular cases, if the resources are well maintained and up-to-date (another issue in the translation industry in general), they are the best solution as they can be easily processed and added to the computer-assisted tool for reference. Furthermore, localisers tend to depend on their client to provide documentation since, as presented in the previous subsection, due to the secrecy that characterises the sector, finding information online from official sources can become difficult (and cumbersome).

Among the other causes in the list, we find that "Insufficient instructions or failure to follow them" was the second cause for "Overflows, overlapping, and truncations", the fourth cause of "Mistranslations", the fourth as well for "Terminology inconsistencies", fifth in the case of "Subtitling errors", and third in "Confusing instructions". In the case of "Overflows, overlapping and truncations", the results of the first survey showed that 23.39% (or 145) localisers did not receive any type of information in relation to the number of characters that can be displayed without creating a GUI bug (graphical user interface bug). Furthermore, some of their comments about missing features mentioned 5 times that they needed a solution for counting the number of characters of the translated segment, an existing option in some CAT tools as we will see in section 5.4.5.

Additionally, others asked for the possibility to import the comments column of an Excel document into the computer-assisted tool. This feature, also supported by several computer-assisted translation tools, would reduce the number of times the localiser has to switch between the CAT tool and the document itself, thus saving time and reducing the possibility of overlooking (or misreading) a comment. In the case of the rest of the linguistic bugs, the quality and quantity of the instructions and comments provided by the client will vary depending on the company, the client's experience with localisation processes and, evidently, each project as a whole. Another crucial factor is the presence of a robust Q&A system in place to ensure fluid communication to both solve the localisers' doubts and share last-minute instructions. This proved to be the most requested feature in the comments left when the localisers were asked about missing functionalities. Furthermore, in the messages left by the testers in the case of terminology, we found one in particular (see Chapter 4 for the complete comment) about problems following the terminology necessary for the certification process because localisers were not aware of the importance of using the requested terms.

In the case of "Too many translators on the same project", as we saw in Chapter 3 and the section about business practices of the present chapter, localisers tend to work in teams if the size of the project is too big to handle by a single person in the time allocated for the task. Video game development is characterised by tight deadlines, which only become even tighter in a simultaneous shipment release model as they are subject to dependencies (which also impact text linearity as the strings that will be used in voiceover will be prioritised). Thus, these time constraints force developers to hire various localisers and the text will need to be divided among them, however, if the text already lacks linearity, the document is split and the localisers only receive their part of the file, the potential for errors only increases as they will not be able to check the entirety of the document. When we move on to the main cause of "subtitling errors", we observe that it was deemed to be "IT-related issues" that do not have a direct connection with the localisers' work. When we look into the format of the subtitles, we can find numerous differences if we compare them with AVT translation—although it is beyond the scope of this study to analyse them—and these differences can be perceived as errors during the LQA phase. Another possibility that causes IT-related issues, would be implementation errors which explain the cases where the audio does not match (at all) the subtitles or the subtitles are missing entirely. Finally, as explained by some testers in the comments they left (and included in Chapter 4), late changes in the audio scripts and/or additions or alterations during the recording of the audio file, are the likeliest cause of partial mismatches between audio and subtitles.

“Lack of translation quality” appeared sixth in the case of “Overflows, overlapping, and truncations”, fifth for “Mistranslations”, and also in the sixth position for both “Subtitling errors” and “Confusing instructions”. Therefore, we can observe that it was chosen as the last option (or almost) whenever it was available in the list of causes. As part of the obvious explanation for this result, we need to consider the impact of crowdsourcing translation in the industry as an inexpensive solution to provide a multilingual product. Another event that can be seen often in social media, is translators that want to start building a curriculum in video game localisation and offer their services for free or much cheaper than the rest. In the case of professionals, we also need to consider the impact of all the constraints involved in video game localisation since translating without references, enough space, and having to deal with gender/number-related issues due to variables do not mix well with fluid and natural texts. Another crucial factor that was mentioned in the comments left by the testers is the use of machine translation to further reduce costs and the time spent on localisation. Although the quality of MT output has improved (and continues to improve) in the last decades, the text still needs to undergo a post-editing process performed by a professional translator or at the very least, by a native speaker with profound knowledge of the game. If these conditions are met, there should not be translation-quality issues, however, as the tester in question mentioned, it is not uncommon in the industry to find that the output has been directly integrated into the game and the company relies solely upon the revision performed during the LQA phase. This practice is due to a confluence of factors that include the rise of popularity in MT due to the apparition of neural machine translation and the misplaced trust of non-linguists in its quality. Secondly, we can observe the growing awareness in companies of the importance of localisation in a globalised market that lack the monetary means to ensure the quality of the translation. Thirdly, the popularity of Unity, which provides tools such as l2 localization or One Click Localization that translate the strings without even leaving Unity’s interface. Finally, the late boom of the video game industry has also played against the profit margin of smaller developers as they move to a game-as-a-service method that depends on monetisation techniques and the continuous addition of new content.

5.3.3 Development process, the localisation phase, and technical data

The results from the third survey showed that nowadays the most widely used video game development process is “iterative”, which received 210 responses (47.84%). However, 3 participants left a message in the comments (instead of selecting “iterative”) to say that they used “agile” or “scrum” methods, which are other ways to refer to the same system as explained in Chapter 1 section 1.3.1, and 1 wrote “hybrid” also without choosing the option from the list (they were allowed multiple answers). These additions would bring the total to 48.52% (or 213) for “iterative” and 27.79% (or 122) for “hybrid”, the second most common process from the questionnaire and which is a mix between “iterative” and “waterfall”. In the case of those 4 participants, 2 of them were in-house developers in an independent studio, one of them was a student and the last one was an in-house developer in a publisher-owned studio (the participant who specified “hybrid”). In order to study if there were significant differences in the development processes adopted by the team depending on the type of contract (and by extent the place of work), we decided to filter the results of the top three methods (“iterative”, “hybrid” and “waterfall”) to obtain a clearer view. Table 67 puts together said processes, as well as the updated average percentage and number of responses, the 6 different options in terms of types of contracts, and the last row displays the number of participants per contract. As the table shows, the majority of the respondents that indicated following an “iterative” method were in-house developers in an independent studio (52.85% or 102) and constitute almost half of the responses for said method (48.57% if we analyse the responses using iterative as the condition). The second highest percentage belongs to those working in-house in a publisher-owned studio (48.81% or 41) although, given the difference between the total number of participants in each group, they only account for 19.52% of the overall results for “iterative”.

When we analyse the figures for the “hybrid” method, we observe that in the case of independent developers (38.30% or 36) the percentage was similar to their results for “iterative” (40.43%) making them the second group in the list of participants following a hybrid method (29.75%). Evidently, due to the size of the sample, in-house developers who worked in an independent studio appeared first with 37.19% of the total amount for “hybrid”. In conclusion, even though some of the groups had a low number of representatives, we can see that the preferred development process for video games is “iterative” with the only exception of the participants who worked in-house in a platform holder-owned studio. In their case, “hybrid” received the highest number of responses although the sample is so small compared to the rest that we cannot make definitive assumptions.

	Independent developer	**In-house in an independent studio**	**In-house in a publisher-owned studio**	**In-house in a platform holder-owned studio**	**Student**	**Other**
Iterative 48.52% (213)	40.43% 38	53.89% 104	48.81% 41	46.67% 7	46.88% 15	38.10% 8
Hybrid 27.79% (122)	38.30% 36	23.32% 45	25% 21	53.33% 8	21.88% 7	23.81% 5
Waterfall 10.71% (47)	14.89% 14	8.29% 16	8.33% 7	13.33% 2	15.63% 5	14.29% 3
Total: 439	94	193	84	15	32	21

Table 67. Development process per type of contract

On the whole, we can observe that nowadays regardless of the type of contract and the place of work, iteration is the norm. In other words, the development team works using short cycles in order to create a feature that is ready to use even though it is not yet final; once they create a base, they continue building on top of what they already have, improving the product until it is deemed finished. This technique will be applied to the mechanics of the game, the art assets, and the different levels (or locations) thus increasing the chances of having usable content for reference in the case of video game localisation. The second more common method, the hybrid approach, is the combination of the waterfall approach (a sequential process in which the next phase is started only if the previous phase is completely finished) at the beginning and the end of the production phase (pre/post-production) and an iterative approach during the actual production phase. This development process, much like the previous one, offers the possibility of reference material to the localisers as well as potential access to the game.

In the case of the beginning of the localisation phase, the build that received the highest number of responses was Alpha (111 or 35.02%) followed by Beta (74 or 23.34%) and Prototype (63 or 19.87%). Table 68 displays all the answers classified following the type of contract and includes the overall results for each build sorted in descending order. The final row also includes the total number and percentage of participants who worked with a simultaneous shipment model for localisation as well as the different figures and percentages for each contract. The first figure that catches the eye is the relatively low number of independent developers working with a sim-ship model (59.57%) compared with the other groups and the survey's average. In-house developers working in an independent studio are also below the average, but the difference is practically negligible (72.02%). In the case of each individual build, we can see that almost all the groups indicated that the localisation phase started once the Alpha build was ready, with "in-house developers in an independent studio" at the top (41.73%). The only two exceptions were the developers who worked in-house in a platform holder-owned studio who chose Beta instead (although the sample is small) and the students, who selected Prototype. The results of these two groups for the second most common build used to begin the localisation phase were Vertical slice and Alpha, respectively. In the case of independent developers, they selected Prototype second instead of Beta.

	Independent developer	**In-house in an independent studio**	**In-house in a publisher-owned studio**	**In-house in a platform holder-owned studio**	**Student**	**Other**
Alpha 35.02% (111)	35.71% 20	41.73% 58	33.33% 22	15.38% 2	20.83% 5	21.05% 4
Beta 23.34% (74)	21.43% 12	23.02% 32	25.76% 17	38.46% 5	16.67% 4	21.05% 4
Prototype 19.87% (63)	26.79% 15	14.39% 20	21.21% 14	7.69% 1	45.83% 11	10.53% 2
Vertical slice 12.93% (41)	14.29% 8	10.79% 15	7.58% 5	30.77% 4	16.67% 4	26.32% 5
Total: 317 (72.21%)	56/94 (59.57%)	139/193 (72.02%)	66/84 (78.57%)	13/15 (86.67%)	24/32 (75%)	19/21 (90.48%)

Table 68. Beginning of the localisation phase per type of contract

When we analyse the advancement of the game in terms of tasks that should be completed in order to have the Alpha build from the development point of view, we can observe that a high percentage of the work is not yet final. Table 69 is a reduced version of Chandler's milestone plan included in Chapter 1 (Chandler, 2020, p. 71) that only displays Alpha and Beta and the main development actors: engineering, art, design, and audio. The best-case scenario in engineering and design in the Alpha build shows that, even though the features and the majority of the areas of the game are playable, the game suffers from numerous functionality issues that will hinder the gameplay experience or even prevent the players to advance in the game (critical bugs). In the case of art and audio, only half of the assets are final and voiceover is still under development. The bulk of the LQA work will be undertaken once Beta is ready, although they tend to start as soon as there are enough localised strings integrated into the game. However, when we observe the column with the information about Beta, we can see that everything is almost final (including localisation if it starts during the previous build) and only the critical bugs will be addressed, providing better working conditions for the localisers who work with studios that start localisation during this build.

	Alpha	**Beta**
Engineering	Key game play functionality is in for all game features. Features work as designed, but may be adjusted and changed based on feedback. Game runs on target hardware platform.	Code complete, only bug fixing from this point forward.
Art	Assets are 40 – 50% final, with placeholder assets for the rest of the game.	All art assets are final and working in game. Only major bug-fixes from this point forward.
Design	All design documentation is completed. Feature implementation is in progress. 40 - 50% of design production tasks are completed. Major areas of game are playable as designed.	All design assets are final and working in the game. Only major bug fixes from this point forward. Minor game play tweaks can be done, based on playtest feedback.
Audio	40 – 50% of sound effects are in and working. Voiceover design is in progress, placeholder VO files are recorded. Music in progress of being composed.	All final sound assets are in and working in the game.

Table 69. Adapted milestone plan (Chandler, 2020, p. 71)

For these reasons, although many of the features of the game are playable in the Alpha build, the numerous functionality issues will cause the game itself to be highly unstable and prone to contain bugs that will interfere with gameplay. Therefore, providing localisers with access to the main build in order to grant them a visual environment to work with might prove difficult as well as counterproductive timewise as they would spend too long trying to reach the area where the problematic string might appear. Furthermore, depending on the game's mechanics, it is necessary to know how to trigger certain events in order to unlock certain dialogues or retrieve certain objects, which would only complicate the task even more. In addition, in between the two builds being discussed, there are multiple versions that will be released and worked upon until reaching the next deliverable build (the main characteristic of an iterative development process). Consequently, it is not realistic to expect to be able to play the full game at such an early stage and access all the scenarios the way they are supposed to look like in the final version (due to the presence of placeholder art assets). However, in the case of menus and user interfaces, even though the graphic design might not be final, it might be possible to access the current version (or the placeholder one) in order to see how the text fits and observe changes "on the fly". This would drastically reduce the impact of "overflows, overlapping and truncations", the most common bug as we have seen in the previous section.

Therefore, we decided to include two questions about developers' testing attitudes in order to analyse if they had systems in place to carry out something similar to "visual regression testing" on top of the options provided by game engines. This type of testing is the generic name of the methods used in programming in order to ensure that the later changes in the source code do not cause unexpected issues in the visual interface or final look of websites, installable programmes, and apps. The possibility of using these methods without having to grant localisers access to the game engine itself would provide the sought-after "preview mode" without compromising the game's integrity. However, when we analyse the developers' responses in terms of both regular testing and play-mode testing presented in Chapter 4, the results show that they mostly rely on the QA phase to identify display or functional bugs. Thus, the only possibility seems to be to use the game engine's preview settings (when available) or any other features and take advantage of those using plug-ins or APIs to create some sort of visual environment in a protected setting that will allow introducing changes safely in the game while protecting the current build from accidental changes or creating different extra builds for testing purposes.

5.4 Tools analysis

This section will be devoted to contrasting the information about tools, each different subsection will begin by studying the impact of the educational background on the level of adoption and knowledge about localisation tools by filtering the results using the questions that were included in the first survey for that purpose. In the case of linguistic testers and developers, said questions were not included in order to shorten the questionnaires and we will directly start with the tools' analysis. Furthermore, we will examine the tools that appeared at the top of the list of each question from all three surveys and study the differences in usage depending on the participants' type of contract. Subsequently, we will focus on their features in general and how they can be integrated or connected to other tools to create a video game development workflow as seamless as possible. Thus, as one of the goals of the present study is to try to identify comprehensive solutions that eliminate the need for extra manual work, we will pay special attention to their capabilities in terms of integration via plug-ins or APIs.

Due to the scope of this thesis and the topic at hand, we will only perform an in-depth analysis of the tool that received the highest number of responses and, whenever the tool appeared in more than one survey, we will contrast all the results simultaneously. In the case of those questions where "none" or "other" was the most popular answer, we will study the second element in the list and, if said element is not a specific tool, in other words, if it happens to be the counterpart of those options, we will evaluate the representativeness of the third tool and decide whether to include it or not. An exception has been made in the case of "asset extraction and integration, content management tools" since the second element in the list was specifically conceived for the video game industry and it was more promising in terms of versatility and perspectives of integration. Finally, we will contrast the features that each tool offers with the results from the previous section when applicable to cross-examine the proposed features and localisers' needs.

5.4.1 Asset extraction and integration, content management tools and project management tools: XLOC and Plunet

First and foremost, we will analyse the degree of adoption of these types of tools depending on the educational background in order to evaluate the impact of an official translation diploma. Table 70 shows all the results organised following the options that were provided to localisers in the question about their educational background and the last row indicates the number of participants in each group. Due to the reduced number of participants with a PhD, their figures will not be displayed and will be replaced by a column with the average results presented in Chapter 3 for easier analysis. In the case of "asset extraction and integration, content management tools and project management tools", we can observe that there are no major variations between the results displayed and that the degree of adoption remains fairly low. The average result in the case of regular users of these systems is low since these tools are mostly reserved for translation project managers or agencies. The highest percentage appears in the case of the participants that did not have any kind of studies in translation (although followed closely by those with a master's degree). When we study the figures for the occasional users, the participants with a bachelor's degree appear first and, once again, those who only attended specialised courses had the worst result. Among those who had heard about these tools but did not use them, (the majority in general) we find the highest percentages in the same two groups that were mentioned before: 69.07% had a master's degree and 68.57% did not have any studies in translation. Finally, in the case of those who had never heard of the tools, the lowest percentage appears in the column of the participants that had a master's degree. Once more, the second group is that of the localisers who did not have a university career in translation studies. These results do not show any relevant differences, even more so if we acknowledge the fact that these tools will be chosen and used by either the developer, the publisher, the platform holder or the LSP.

	None	**Specialised courses, seminars, etc.**	**BA in translation**	**MA in translation**	**All types (average)**
Use regularly	10.71% 15	8.70% 8	9.86% 14	10.59% 25	10.32% 64
Use sometimes	10.71% 15	9.78% 9	16.90% 24	13.56% 32	13.23% 82
Have heard of, but do not use	68.57% 96	66.30% 61	61.27% 87	69.07% 163	66.61% 413
Never heard of	10.00% 14	15.22% 14	11.97% 17	6.78% 16	9.84% 61
TOTAL	140	92	142	236	620

Table 70. Degree of adoption of asset extraction and integration, content management tools and project management tools by type of education

The second part of this analysis will focus on the two results that appeared at the top of the list in the two questions about the tools the participants used by filtering them this time according to the type of contract. Table 71 displays the resulting tools and the average responses they received, the last row provides an overview of the degree of adoption of the tools depending on the type of contract and, due to space constraints and the low number of participants, the categories for students and fan translators are not displayed but will be referred to whenever it is pertinent. First, in terms of the general degree of adoption, we can see that the figures for all three types of freelancers are below the average and the lower result is for freelancers who only work with an agency. Conversely, the result for in-house translators who work for a language vendor almost doubles the average, thus proving that these tools are mostly used by in-house personnel and LSPs. The individual results for asset extraction and integration and content management tools that will be mentioned in this section show that all three categories of freelancers are above the average in the use of Excel/Word for these purposes whereas the participants who work in-house have the lowest percentages. In addition, 3 students or fan translators also marked Excel/Word as their preferred method.

In the case of XLOC, the highest percentage of users is for in-house translators who work in a translation agency and the second category above the average are freelancers who work both independently and with an agency. Surprisingly, localisers who worked in-house for a video game development company had the second-worst result and 2 students or fan translators selected XLOC as well. When we contrast the results that XLOC received in the third survey with those from the localisers, we can observe that out of the 46 developers who had heard about the tool but did not use it, the majority worked in-house in an independent studio (47.83% or 22) followed by independent freelancers (28.26% or 13). The 5 participants that used the tool either regularly or occasionally worked also as in-house developers in an independent studio (3), or in a platform holder-owned studio (1) and the last one was a project manager in a video game editor and publisher. Therefore, we can assume that either XLOC is mostly reserved for project managers in the localisation team who work directly with language vendors that later outsource the work, or that the tool is not widely used outside the company that owns it. Finally, the results for Plunet show that it was mostly used by freelancers and the highest percentage was for those who worked both independently and with an agency (54.55%).

	Freelancer	Freelancer with an agency	Freelancer working both independently and with an agency	In-house translator in a translation company	In-house translator in a non-translation company
Excel/Word 65.10% (97)	67.65% 23	75.00% 12	66.67% 22	62.86% 22	57.69% 15
XLOC 22.15% (33)	14.71% 5	6.25% 1	24.24% 8	42.86% 15	7.69% 2
[redacted]					
Plunet 30.20% (45)	38.24% 13	25.00% 4	54.55% 18	20.00% 7	11.54% 3
Total: 149 (24.03%)	34/174 (19.54%)	16/104 (15.38%)	33/157 (21.02%)	35/77 (45.45%)	26/75 (34.67%)

Table 71. Top asset extraction and integration, content management tools and project management tools by type of contract

5.4.1.1 XLOC

In the case of asset extraction and integration tools as well as content management tools, the most common answer was "Excel/Word + Visual Basic macros" followed by XLOC. However, as explained in the introduction, we will concentrate on the second option as the aim was to study promising technological solutions for video game localisation that might be able to reduce the need for copying and pasting and automate as much of the process as possible. XLOC is a company founded in 2000 specialised in the video game industry, is part of the Keywords Studios group and provides web-based Content Management System (CMS) services. As mentioned in Chapters 3 and 4, although we thought initially that the tool was only used for basic content management, after the first analysis we realised that it provided a wide array of functionalities especially conceived to deal with the complexity of the video game localisation industry. According to their website:

> XLOC is a key production tool link to enable all global team professionals involved in the development process to collaborate and visualise the status of multilingual assets at any time. As a result, the localization cycle can be seamlessly connected to the source content production cycle, pushing to translation just the new and updated

content, saving precious time and resources, and eliminating repetitive tasks and errors caused by the manual tracking of assets.[37]

The initial confusion stems from the fact that the main problem when trying to gather information about this system was the scarcity of data available in comparison with the rest of the tools that will be presented later on. Additionally, the number of participants from both the first and the second survey that either worked with it or at least knew about its existence was remarkably low, as presented at the beginning of the subsection. However, XLOC seems to be a well-established service provider that collaborates with important actors in the industry such as Capcom, Activision, Neowiz, GLU Mobile, Milestone, Bethesda, and Keywords Studios. Nevertheless, accessing technical information about how the tool actually works has proven to be a challenge due to the abovementioned scarcity of resources. As a matter of fact, the only sources of information available are several blog articles, their website, two videos on Youtube and less than 90 actionable results on Google—once we remove from the list similar entries—which mostly lead to either their own website, to Keyword studios or their social media accounts. The most recent blog entries provide information about localisation processes or improvements to the product. They also propose the option to schedule a meeting with them for more information but, despite numerous attempts since we learned about their existence, we have never received a reply to our messages to this day.

For this reason, we will only be able to use their website, the videos, and the blog entries to analyse how the tool works and the features it provides without access to a manual or the means to test it. The company's site shows the primary benefits and the key features offered by the latest version of its product. The system seems to provide a shared location for all the assets that will need to undergo localisation and it seems to be focused on how said assets move through a team going from each stage of the translation process. Therefore, it offers a centralised solution to import all the assets from the game and store them and, in the case of the localisation proper, the operation seems to be carried out on a language-by-language basis by the means of an individual tab dedicated to each target language. In the case of asset types, XLOC supports images,

[37] http://xloc.com/overview.html

videos, audio, and art; providing a seamless connection with all the programmes involved. This leads us to believe that it could technically allow checking the translated string in its visual environment if the localisers were given access to the programme itself and the game engine had either a preview mode or the possibility of creating "testing builds". Other interesting features are the fact that they have a Q&A function, they seem to be able to closely track changes in every asset and they also have the possibility to track tasks.

When we analyse the options that the tool provides in terms of integration, the system is "compatible with all major 3D engine technologies, social networking games and vast MMO environments, and can be leveraged as stand-alone products or be combined with XLOC's consulting and support services".[38] Additionally, in the specific case of computer-assisted translation tools, the company recently launched in 2020 a memoQ connector that enables users to "push" the strings directly to memoQ to further reduce the number of manual tasks. Unfortunately, when we search "XLOC" on memoQ's website there is only one false result even though their website usually has extensive information about all the integrations they provide or, at least, a blog entry in the case of new additions. This only confirms how secretive the company seems to be and how much control it exercises when it comes to information about its services. In order to find information about the connector itself we had to investigate several blog entries and found an article published in December 2020 where Carlos Garcia-Shelton (Product Manager at XLOC) explained:

> With projects that take advantage of memoQ's TM, the connector will solve for the first time two recurrent but different problems: string status tracking and strings recycling. This is obtained through a seamless back and forth interaction between content management (XLOC) and translation management (memoQ). Instead of being in a state of grabbing from XLOC and then going to memoQ to import a new translation job, rebuilding in memoQ and back to XLOC to upload that translation work when completed, the XLOC memoQ connector will allow for an easy push to memoQ. There will be a

[38] http://xloc.com/about.html

unified page to display and track all current memoQ jobs, along with their status.[39]

Finally, we will present the cross-analysis of the features that XLOC provides and those deemed as “essential” and “useful” by the localisers as well as the results from the most common bugs. In the case of “the possibility of seeing dialogues in order” XLOC does not seem to have a feature specifically created in order to deal with it (unless it does not appear on their website). The courses of action to deal with this particular issue would be changing business practices in general and arranging Excel documents in chronological order (or the closest thing to it) and integrating tree-based tools in the case of branching dialogues; pre-processing the document before uploading it, or modifying the exporting options (and providing dialogue trees as well). In order to mitigate the impact of the lack of linearity, localisers could use XLOC’s search features although this may also prove to be difficult if they do not have access to the entirety of the text and might take too much time. As presented in the subsection with the causes of the linguistic bugs, the lack of linearity or, in other words, in-game context, was deemed to be the main cause for “mistranslations”, which was the second most common linguistic bug, and for “confusing instructions”.

The new connector would provide memoQ’s terminology management and extraction tools to ensure consistency, its quality assurance options, and the computer-assisted translation tool. Those three features were marked as “essential” and occupied the second, fourth and fifth positions, respectively. To avoid redundancy, we will study them in the specific subsection dedicated to memoQ. In the case of XLOC’s own translation tool, there is not much information available on their website besides a reference about in-line editing and this statement: “Centralized Translation & Fuzzy Matching: Reference, link, reuse and search translation terms. Leverage Fuzzy Matching on a string-specific level, as well as global population.”[40] However, in a video[41] that can be found online on Youtube, there is an image that specifies “translation memory” and there are several screenshots from the tool itself. The presence of a

[39] https://www.gamespress.com/XLOC-streamlines-video-games-localization-process-with-new-XLOC-memoQ-

[40] http://xloc.com/feature.html

[41] https://www.youtube.com/watch?v=wl_WKLlEhag&ab_channel=creativeillusionsProductions

translation memory would help to mitigate the impact of "terminology inconsistencies" which was deemed as the fourth most common linguistic bug as well as improve the localiser's capacity to spot changes in the source file and keep track of modifications.

The third feature considered essential in the first survey was "access to audio, video and images", it subsequently appeared in the second position when we included the results for the "useful" features. In this case, as explained above, XLOC provides access to audio, video, and image files directly from the game engine via API or will do so shortly according to their website (see Image 24) and they specify that "Audio and Art assets can be uploaded and stored, and Audio assets can be linked and played with corresponding string"[42]. Therefore, at the moment, the availability of these resources seems to depend on the developers' (or project managers') willingness to supply the reference material. However, if the process becomes automatic (or as automatic as possible) and every manual task is reduced to the bare minimum, the potential of including more assets for reference would increase as it would require less time to set up the process. As we saw in the subsection about linguistic bugs, the absence of a visual environment and resources appears among the top three causes of all linguistic bugs besides grammatical and typographical errors.

The last two features and functionalities that appeared from the combination of "essential" and "useful" were the "possibility to track any changes in the source text files (management tools)" and "access to all the assets in their original form and divided by formats" (fourth and fifth positions). In the first case, we can see that the main selling point of XLOC is its capacity to track changes, processes, and builds in order to maintain everything under control and, for example, they propose "String history and status" and "dynamic synchronization"[43]. In the second case, the system stores game assets that will need to be localised, thus enabling access to them, and they also claim to be able to support "any Game file structure and Formats" and that "XLOC can take any source file from the game engine and display the contents in the standardized environment with no further conversion or tweaking".[44]

---

42 http://xloc.com/feature.html
43 http://xloc.com/feature.html
44 http://xloc.com/feature.html

Primary benefits of XLOC 6.2

- XLOC is the **key production tool** to enable all global team professionals involved in the development process to collaborate and visualize the status of multilingual assets at any time; LPM, LQA, Translators, Dev.
- **XLOC becomes the linking connector tool between service lines.**
- XLOC perfects the process of **storing, analyzing and tracking the evolution of source and localized** game assets during development in a **centralized and standardized environment** for all your projects.
- XLOC can take **any source file** from the game engine and display the contents in the standardized environment with no further conversion or tweaking.
- The localization cycle can be seamlessly connected to the source content production cycle, pushing to translation just the new and updated content, saving precious time and resources, and eliminating repetitive tasks and errors caused by the manual tracking of assets.
- **String history and status** allows various service lines to track, test and approve localization efforts.
- **Advanced search and bulk processes** simplifies workflows.
- Robust **Questions and Answers** features allows for team knowledge share.
- **Error checking and alignment** of data to ensure the integrity of your localized assets.
- **Audio and Art** assets can be uploaded and stored, and Audio assets can be linked and played with corresponding string.

Key Features of XLOC 6.2

- Continued **Enhanced API integration** with Game Engine and TM hooks coming shortly
- Supports all **game platforms and languages.**
- Supports **any Game file structure and Formats.**
- **Powerful Extended search, Commenting and Error checking features**, to ensure the integrity of the data for all users. Dependable central repository.
- **Customized Interface.**
- **Interface localized** into English, Chinese, Japanese and Korean.
- **String history and status** allows various service lines to track, test and approve localization efforts.
- **Powerful search features**, including multi-choice options.
- **In-Line editing** for quick changes to strings.
- **Saved searches** for creating customised workflows.
- **Robust messaging** for production autonomy and troubleshooting.

Image 24. Features of XLOC 6.2

Finally, if we also take into account the categories that were included in the “missing features” section, we can see that XLOC and its connection to memoQ seem to cover a substantial number of them. The combination of these two systems would provide memoQ’s quality assurance options and character count (that will be discussed later), and XLOC also covers the most requested feature, a method to communicate with developers and/or project managers. Specifically, it provides a tab for questions and answers, according to the video. When we analyse closely their website, we find two sections in particular that lead us to believe that XLOC could also potentially provide the possibility of seeing changes “on the fly” although we have not been able to confirm this hypothesis. The first simply reads: “Immediate Testing. Create prototype builds prior to translation and instantly generate game files for localized drop-ins or builds.” And the second reference—besides presenting the possibility to perform LQA and see changes “on the fly”—suggests that XLOC can be used as a bug-reporting tool as well:

> For LQA. Cut QA lead and test time by 50 percent. Do linguistic testing on the fly while maintaining the highest standards of quality assurance. Pseudo-builds speed early testing, checklists verify

translation progress and XLOC becomes your centralized source for linguistic bug reference data.[45]

5.4.1.2 Plunet

Plunet is a translation management system designed mostly for LSPs and was not exclusively created for the video game industry. The tool allows to receive and organise translation requests including the source text and the instructions for the project. Once the translation project has been added to the system, the tool also simplifies the process of creating quotes and "the price lists for your customers are linked to the CAT analyses in the project and correct prices are automatically calculated"[46]. The system also allows to create workflows, allocates translators to the project, and compares them "based on their prices, availability, job feedback"[47] and, finally, it has an accounting feature. In terms of integration capacities, Plunet seems to work with the most popular computer-assisted tools and, in the case of memoQ, the interface they use is called memoQManager. Although management and quotes are essential in the trade, it is beyond the scope of this work to go into further detail. According to the website, there are two versions (standard and advanced) and the features of the standard version include:

> Import of CSV files under Quotes, Orders and Jobs. Assignment of all price units for translation, proofreading, quality checks, DTP, engineering via memoQ analysis. Automatic calculation of translation services and other services such as proofreading, working with DTP programs, and quality checking. Assignment of all price units to projected target times. Calculation of prices and estimation of deadlines for even complex quotes and orders with a single click (Requirement: WorkflowResourceManager Module). Automatic calculation of job costs for translators, proofreaders, quality checkers.[48]

---

45 http://xloc.com/feature.html
46 https://www.plunet.com/en/translation-management-software/
47 https://www.plunet.com/en/translation-management-software/
48 https://www.memoq.com/integrations/business-management/plunet-businessmanager

5.4.2 Tree-based tools

This subsection will only deal with the degree of adoption since the first survey did not include a question with a list of potential programmes even though some examples were provided in Chapter 1 section 1.3.2. As explained in Chapter 3, the average level of adoption of this type of tool was already suspected to be low at the moment of the conception of the survey and the results only confirmed the hypothesis. When we analyse the studies for the 12 participants that stated using tree-based tools regularly, we find that 8 of them had a master's degree in translation whereas the rest of the groups had either 2 or 0 representatives (Table 72). Once again, in the case of occasional users, we find the highest percentage among the participants with a master's degree, followed by those without university studies. Among the results for those who had heard about tree-based tools but did not use them, we find that the participants with a bachelor's degree appeared first with 49.30%, followed by the participants that did not have a degree in translation. Conversely, these two categories also had the lowest percentages when it came to the last option, "never heard of". In this subsection, the worst results have been consistently those obtained by the participants that only had "Specialised courses, seminars, workshops, etc." with the exception of "Have heard of, but do not use" although the difference with the lowest percentage is almost negligible.

	None	**Specialised courses, seminars, etc.**	**BA in translation**	**MA in translation**	**All types (average)**
Use regularly	1.43% 2	0.00% 0	1.41% 2	3.39% 8	1.94% 12
Use sometimes	12.86% 18	10.87% 10	10.56% 15	13.56% 32	12.42% 77
Have heard of, but do not use	43.57% 61	38.04% 35	49.30% 70	39.41% 93	42.58% 264
Never heard of	42.14% 59	51.09% 47	38.73% 55	43.64% 103	43.06% 267
TOTAL	140	92	142	236	620

Table 72. Degree of adoption of tree-based tools by type of education

5.4.3 Corpus usage and compilation tools: Sketch Engine

In the case of corpora, we will first analyse the usage of the different types depending on the respondents' educational background, then we will move on to the adoption of corpus compilation tools and, finally, we will analyse the second option that appeared on the list of tools (as the first option was "other"). Table 73 puts together the results for each type of corpus and the percentage of users depending on their studies, it is also necessary to mention that the respondents were allowed to provide multiple answers if they used more than one type of corpus. The average results presented in Chapter 3 already showed that “bilingual parallel corpora” was the most widely used type and the present table confirms that it was also the case for almost every category with the exception of those without university studies or specialised courses. “Multilingual parallel corpora” is the second less used type on the list and the highest percentage of users can be found among the respondents with a bachelor’s degree, whereas the results of the previous type of corpus put those with a master’s degree first. The only moment when the respondents who had only taken “specialised courses, seminars, workshops, etc.” diverged noticeably from the average (17.74%) and came up first with 27.7%, was in the case of “monolingual comparable corpora”. The participants with a master’s degree had the best result in the case of “bilingual comparable corpora” and those with a bachelor’s degree had the highest percentage of use of “multilingual comparable corpora”. To conclude, we can observe that those who did not have any type of university studies in translation had the lowest percentages in every option except for “none” and that, at times, the differences were substantial. Conversely, those with university studies had the best results (even though the participants that had attended specialised courses were not far behind them).

	None	**Specialised courses, seminars, etc.**	**BA in translation**	**MA in translation**	**All types (average)**
Bilingual parallel corpora	30.71% 43	50.00% 46	50.00% 71	51.27% 121	45.81% 284
Multilingual parallel corpora	9.29% 13	14.13% 13	17.61% 25	17.37% 41	15.00% 93
Monolingual comparable corpora	10.00% 14	27.17% 25	18.31% 26	17.37% 41	17.74% 110
Bilingual comparable corpora	18.57% 26	22.83% 21	23.24% 33	23.73% 56	22.42% 139
Multilingual comparable corpora	6.43% 9	9.78% 9	10.56% 15	10.17% 24	9.35% 58
None	57.14% 80	33.70% 31	30.28% 43	32.63% 77	37.90% 235
TOTAL	140	92	142	236	620

Table 73. Corpus usage by type of education

In the case of the use of corpus compilation tools, as presented in Chapter 3, we can observe that they remain a largely unadopted technology. The filtered results presented in Table 74 show that, even though the number of users is remarkably low, the highest percentage can be found among those with a master's degree, followed by those with specialised courses and, in the third position, the participants without university studies. Although the result is not representative due to the fact that the number of participants was only 10, none of those who had a PhD compiled their own corpora. As explained in Chapter 2, previous research in the field of translation in general carried out by Zaretskaya *et al*. (Zaretskaya *et al.* 2015, p. 250; 2017, p. 46) found that only 17% of those who worked with corpora also used compilation tools.

	None	**Specialised courses, etc.**	**BA in translation**	**MA in translation**	**PhD in translation**	**All types (average)**
Yes	6.43% 9	7.61% 7	5.63% 8	11.86% 28	0.00% 0	8.39% 52
No	93.57% 131	92.39% 85	94.37% 134	88.14% 208	100.00% 10	91.61% 568
Total	140	92	142	236	10	620

Table 74. Degree of adoption of corpus compilation tools by type of education

In the present study, the total percentage of users of corpus compilation tools is 8.39% however, once we remove the respondents who stated not using any type of corpus (for a better comparison), our result rises to 12.44%. Furthermore, the authors studied the impact of the education level on the degree of adoption of the tools and their results also proved to be higher than those obtained by the present study: 66.7% for the respondents without any type of studies in translation, 30.4% for those who had attended specialised courses, 13.3% had a bachelor's degree and 28% had a masters' degree (Zaretskaya *et al.* 2017, p. 48). Since the authors calculated their results using corpora users as their base, we decided to adapt our results as well to check for similarities. In our case, after modifying the calculations, 13.33% did not have formal studies in translation, 11.48% only had specialised courses, 8.00% had a bachelor's degree, and 15.72% had a master's degree. Although in both our study and Zaretskaya's *et al*. the sample is small (52 and 32 participants respectively) and not very representative, the differences in the results are noticeable and would derive from the lack of resources that characterises the video game localisation industry as explained in section 5.3.1.2 of the present chapter.

The tool that received the highest number of responses among the few localisers that did compile their own corpora was Sketch Engine, which appeared in the second position after the option "other". When we analyse the results classified by type of contract, we observe that, even though they were not included in Table 75, 53,33% of the total number of participants who selected Sketch Engine were students (44.44% in the case of the total number of corpora compilation tools' users). The second highest percentage can be found among freelancers and neither in-house translators working in a development company nor fan translators chose Sketch Engine. These results, although once again not very representative, point to the presence of university courses in corpus compilation since 7 of those students were following a master's degree in translation.

	Freelancer	**Freelancer with an agency**	**Freelancer working both independently and with an agency**	**In-house translator in a translation company**	**In-house translator in a non-translation company**
Other 45.28% (24)	33.33% 5	85.71% 6	37.50% 3	50.00% 3	75.00% 6
Sketch Engine 28.30% (15)	26.67% 4	14.29% 1	12.50% 1	16.67% 1	0.00% 0
Total: 53	15/174 (8,62%)	7/104 (5%)	8/157 (5,1%)	6/77 (7,79%)	8/75 (10,67%)

Table 75. Corpus compilation tool by type of contract

Sketch Engine is an online corpus compilation tool that allows creating a corpus either from documents uploaded into the system, from an already selected number of websites or even searching the internet in order to find relevant results that can be used later (webcrawling). Once the corpus is created, it provides different options to exploit it depending on its type. These range from simply extracting terms and being able to analyse the way they are used in context, to contrasting the different translations of a word in multiple languages and documents. As the company explains on its website:

> Sketch Engine is the ultimate tool to explore how language works. Its algorithms analyze authentic texts of billions of words (text corpora) to identify instantly what is typical in language and what is rare,

unusual or emerging usage. It is also designed for text analysis or text mining applications.[49]

In terms of integration capabilities, the tool has a plug-in specifically designed for SDL Trados Studio (Image 25) although, besides this extension, it seems to rely upon exporting files in formats compatible with CAT tools so they can be uploaded and used directly by them. The first issues are the current preference towards comprehensive solutions and the fact that SDL Trados Studio seems to be losing ground in the computer-assisted translation landscape. Another characteristic of the industry that proves to be problematic for the use of corpus compilation tools is the lack of resources besides those provided by the client and the difficulty in finding official resources online. In the case of professional localisers, the most useful type are “bilingual parallel corpora”, however, unless they are given access to past translations (in which case they tend to receive a translation memory), the quantity and quality of bilingual information online is highly limited. Whenever there are resources online such as bilingual wikis that can be used either as parallel or comparable corpora, the documents need to be aligned, which requires another tool on top of Sketch Engine and adds another step to the process. However, once the multilingual corpus (as called in Sketch Engine) is compiled and aligned, it can be downloaded in TMX format and leveraged by computer-assisted translation tools such as memoQ as if it was a translation memory.

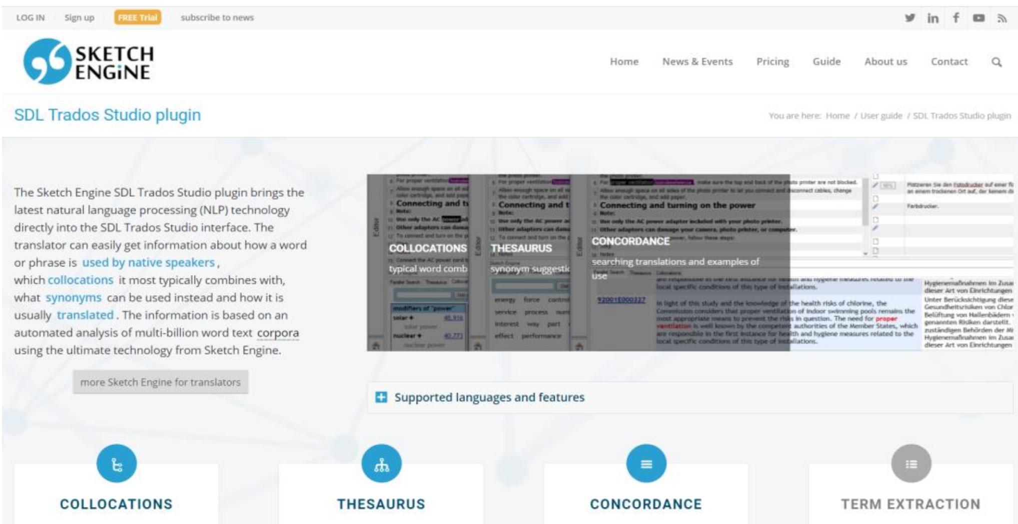


Image 25. Sketch Engine and SDL Trados Studio plugin[50]

49 https://www.sketchengine.eu/

5.4.4 Terminology extraction and management tools: SDL MultiTerm

In the case of terminology extraction and management tools, we can see an improvement in the number of users (141 participants) compared to the previous tool although the percentage remains low, as explained in Chapter 2 and the previous section (Table 76). Among the respondents that use these tools regularly we find that the highest percentage has a bachelor's degree (albeit the result remains under 9%) and the lowest percentage belongs to those with specialised courses only. Localisers with a master's degree have the best results in terms of occasional users and, once again, the lowest number is for those with specialised courses, who also have the highest percentage of having heard about terminology extraction and management tools but not using them. Finally, the participants with a bachelor's degree were those with the highest result for the last option on the list. Overall, we can observe that the respondents who did not have a university degree in translation remained fairly close to the average and only surpassed it in the case of "never heard of", where they had the second worst result. The same can be said about those with a master's degree, who were slightly above the average in every option except for the last one, where they had the best result. The results for the other two groups were less stable and do not show discernible patterns.

In the article published by Zaretskaya *et al.* (Zaretskaya *et al.* 2017, p. 46), the authors divided these tools between terminology management tools and terminology extraction tools (as explained in Chapter 2), therefore, the comparison becomes more difficult. However, the results from the present study seem to be closer to those obtained in the case of "terminology extraction tools" both in terms of the average percentages and those classified by educational background. The authors' results combining all the responses from regular and occasional users of terminology extraction tools were 25% (and 58% for terminology management tools) whereas the results from our study show 22.74% (Zaretskaya *et al.* 2017, p. 46). These findings lead us to believe that the localisers mostly took into consideration the extraction feature when they answered our survey. The individual results for each type of educational background obtained by Zaretskaya *et al.* (Zaretskaya *et al.* 2017, p. 48) were: 23.2% for "none", 28.8% for

[50] https://www.sketchengine.eu/

specialised courses, 26.8% for those with a BA, and 25.7% for those with an MA. Our study obtained a result slightly higher in the case of the respondents with a master's degree (25.85%) and around 1% lower in the case of BA holders (25.35%) and those without official studies in translation (22.14%). The only clear divergence is the noticeably lower percentage of participants who had followed specialised courses and used this type of tool (11.96%).

	None	**Specialised courses, seminars, etc.**	**BA in translation**	**MA in translation**	**All types (average)**
Use regularly	5.71% 8	3.26% 3	8.45% 12	6.78% 16	6.29% 39
Use sometimes	16.43% 23	8.70% 8	16.90% 24	19.07% 45	16.45% 102
Have heard of, but do not use	61.43% 86	75.00% 69	57.75% 82	65.25% 154	64.35% 399
Never heard of	16.43% 23	13.04% 12	16.90% 24	8.90% 21	12.90% 80
TOTAL	140	92	142	236	620

Table 76. Degree of adoption of terminology extraction and management tools by type of education

The results about the tools used for standalone terminology extraction and management and those used for working with corpora were not representative enough as the first option on each list was "none" followed by "other". Nevertheless, the question about the tools integrated into computer-assisted translation systems yielded better results and, even though the first option was still "none", the second on the list was SDL MultiTerm. Table 77 presents the results classified by type of contract and shows that in-house translators who work in a video game development company do not usually work with the terminology extraction and management features integrated into the CAT tool and, consequently, had the worst result for SDL MultiTerm. In addition, 6 students and 5 fans translators also selected SDL MultiTerm as the tool they used for terminology extraction and management. Freelancers working both independently and with a language vendor have the best result for the tool in question, followed by independent freelancers and in-house translators working in a translation agency. Among the already mentioned low degree of adoption of these tools, we can observe the fact that localisers seemed to prefer memoQ, which could also explain these results.

	Freelancer	**Freelancer with an agency**	**Freelancer working both independently and with an agency**	**In-house translator in a translation company**	**In-house translator in a non-translation company**
None 63.39% (393)	63.22% 110	63.46% 66	57.96% 91	66.23% 51	76.00% 57
SDL MultiTerm 27.42% (170)	28.74% 50	25.96% 27	35.03% 55	27.27% 21	8.00% 6
Total	174	104	157	77	75

Table 77. Terminology extraction and management tools by type of contract

SDL MultiTerm belongs to RWS, the same company that owns Trados Studio, which describes itself as "the world's leading provider of technology-enabled language, content management and intellectual property services"[51]. MultiTerm can be utilised as a standalone programme, but it is normally used in combination with Trados Studio. The tool provides a centralised environment for terminology management in order to allow easy access for a team of translators, it supports different file formats, the entries can be modified easily, and it has a feature that allows users to look up the meaning of a term from any application.[52]

Although any of the tools mentioned can provide means of creating glossaries and dealing with terminology issues is essential in video game localisation, the fact that it can only work in combination with SDL Trados Studios and the abovementioned success of other computer-assisted tools, hinder the integration capabilities of the system. As mentioned in the previous subsection, translators in general tend to favour

[51] https://www.rws.com/about/
[52] https://www.rws.com/translation/software/multiterm/

comprehensive systems and given the lack of resources, time constraints and potential NDAs, it might become a matter of the ratio between the time invested and the money earned. Furthermore, when localisers have access to glossaries, those tend to be created and maintained by either the developer or the language vendor and, if they are not given access to the entirety of the document and have no means to contact the rest of the translators in the group, their chances of taking advantage of terminology extraction and management tools will be diminished.

5.4.5 Computer-assisted translation tools: memoQ

In the case of the degree of adoption of computer-assisted translation tools, we can clearly see that the best results are those obtained by the participants who had either a bachelor's degree or a master's degree, with the latter in the first position (Table 78). On the opposite side, we find that the participants who did not have a university degree in translation had the worst results and the category "none" included the only 3 participants that had never heard about CAT tools. The lowest percentage of regular users can be found among those who only had specialised courses in translation, the group that had the highest percentage of occasional users as well as the highest percentage of those not using computer-assisted tools by choice.

Chapter 2 showed that computer-assisted translation tools have been fully integrated into the translation workflow and our research has proven that they are considered essential for the trade. The results obtained by Zaretskaya *et al.* (Zaretskaya *et al.* 2017, p. 46) revealed that at the time 76% of the participants used this type of technology whereas 3 years later, in the video game localisation domain, the percentage of users rises to 92.58% thus confirming the 2020 European Language Survey report's findings (2020, p. 51-52). Once the authors classified their results by type of educational background, the lowest percentage of users was found among translators without any kind of translation studies (66.9%), followed by those who only had specialised courses (78.4%), and those with a bachelor's degree (78.1%). The participants with a master's degree proved to have the best result with 86.7% (Zaretskaya *et al.* 2017, p. 48).

In the case of the present study, nearly all the percentages are above those results and the lowest belongs to localisers who had only attended specialised courses (84.78%).

The participants who did not have any type of certification in translation obtained 85.71%, those with a bachelor's degree in translation scored 95.79% and those with a master's degree appeared at the top with 97.46%. We can therefore conclude that computer-assisted translation tools have been becoming increasingly popular due to their systematic inclusion in universities' curricula as well as the increasing demand from both clients and language vendors alike.

	None	Specialised courses, seminars, etc.	BA in translation	MA in translation	All types (average)
Use regularly	75.00% 105	67.39% 62	85.92% 122	87.71% 207	81.45% 505
Use sometimes	10.71% 15	17.39% 16	9.86% 14	9.75% 23	11.13% 69
Have heard of, but do not use	12.14% 17	15.22% 14	4.23% 6	2.54% 6	6.94% 43
Never heard of	2.14% 3	0.00% 0	0.00% 0	0.00% 0	0.48% 3
TOTAL	140	92	142	236	620

Table 78. Degree of adoption of computer-assisted translation tools by type of education

Furthermore, when we analyse the differences in the use of computer-assisted translation tools depending on the type of contract of the participants (Table 79), we can observe in the last row that the percentage of overall users of this type of technology drops by 10% compared to the average in the case of in-house translators who work in a video game development company (82,67%). The participants from that group are the only ones below the average degree of adoption along with independent freelancers (90,8%), although the difference between the figures is smaller in the second case.

Freelancers who worked both independently and with a translation agency had the highest percentage of overall users (98,09%) followed closely by freelancers who worked exclusively with an agency (97,12%) whereas the participants who worked in-house in a translation agency are just above the average result (93,51%). In the case of the specific tools used, we can observe that localisers who work directly for the video game developer do not follow the pattern established by the other groups and their results diverge noticeably from the average. In the case of memoQ and Trados, the gap between the average and the figures obtained by in-house translators in a non-translation company is 15,44% and 37,13% respectively (the participants also chose Memsource over Trados), whereas 69.35% of them chose "other" for localisation tools with many references to memoQ, proprietary tools and new web-based translation systems. Therefore, we will concentrate on memoQ and its features given the lack of specific localisation options.

	Freelancer	Freelancer with an agency	Freelancer working both independently and with an agency	In-house translator in a translation company	In-house translator in a non-translation company
MemoQ 86.41% (496)	86.08% 136	88.12% 89	93.51% 144	84.72% 61	70.97% 44
Trados 61.32% (352)	60.76% 96	69.31% 70	70.13% 108	56.94% 41	24.19% 15
Other 48.61% (279)	51.27% 81	42.57% 43	49.35% 76	41.67% 30	69.35% 43
Passolo 28.05% (161)	26.58% 42	34.65% 35	25.97% 40	30.56% 22	12.90% 8
Total: 574 (92,58%)	158/174 (90,8%)	101/104 (97,12%)	154/157 (98,09%)	72/77 (93,51%)	62/75 (82,67%)

Table 79. CAT tools by type of contract

MemoQ provides a comprehensive solution in terms of computer-assisted tools in general and has been gaining popularity among translators during the past years. Furthermore, the company launched in 2019 the "memoQ Gaming Unit"[53] to stand out from the competition and position itself as one of the main technological solutions in the gaming industry while highlighting how its tool fits in this particular sector. Following the trend in the market, memoQ provides an increasing number of extensions

[53] https://www.memoq.com/solutions/game-localization

and connections with other systems to reduce as much as possible the need for standalone options. On their website we can observe that the company provides an extensive array of integration options classified into "Extensions: Qterm, memoQWeb, Customer Portal, and Language Terminal"; "Integrations: Machine Translation, Business Management, Content Management, Preview Tools, Term/memory Banks, Quality Assurance, and Marketing Automation"; "APIs: memoQ server Web Service API, memoQ server Resources API, and memoQ server CMS API"; "SDKs: Quality Assurance SDK, Machine Translation SDK, Translation Memory SDK, Preview SDK, and Term Base SDK"[54]. In addition, they provide services in order to set up customised workflows and offer tailored advice and solutions in video game localisation.

The company has also published an eBook on video game localisation (*LocLand: the Land of Game Localization*[55]) in order to provide video game developers with some information about the localisation process as well as to present how memoQ's features can be leveraged by developers. The booklet presents LiveDocs as an alternative to translation memories, the presence of an automated quality assurance system, how to work with multilingual Excel filters, memoQ's project management features, and how to collaborate on translations in real-time or the system's communications options (Image 26). LiveDocs is a feature in memoQ that allows users to create a corpus that "can contain monolingual and bilingual documents, alignment pairs and binary reference materials"[56], it aligns the segments automatically and provides fuzzy matches as a translation memory. The automated QA process can be used at the end of the translation process in order to find "missing segments, forbidden terms, and consistent translation of same sentences"[57]. In the case of the Excel filters, memoQ allows managing the columns that will be displayed when the Excel document is uploaded and the possibilities include showing multiple source languages as well as comments and context.[58] The tool also has a dashboard for project management, the possibility of creating projects based on a server in the case of teamwork in order to synchronise the translations "to leverage from the translation memories they and other translators update

[54] https://www.memoq.com/integrations
[55] https://www.memoq.com/resources/ebooks/locland-game-localization
[56] https://www.memoq.com/tools/livedocs
[57] https://www.memoq.com/resources/ebooks/locland-game-localization
[58] https://docs.memoq.com/current/en/Places/multilingual-excel-and-delimit.html

in real-time"[59], and instant messaging among the members of the teams. Finally, according to their eBook, "Native Prime (a multilingual game localization company) has created a plugin called Loclink that makes localization between Unity and memoQ easy and simple."[60]

TABLE OF CONTENTS

Image 26. LocLand: the Land of Game Localization's table of contents

Due to the number of options proposed and the topic at hand, we will move on to the cross-analysis of the features that memoQ provides and the results from the previous section and the key questions. In the case of the first essential feature "the possibility of seeing dialogues in order" memoQ does not propose a specific solution besides those included in XLOC's analysis. The second feature considered essential was "terminology management and extraction tools to ensure consistency". However, as previously discussed in the section specifically devoted to the degree of adoption of this particular tool, these systems remain largely unused despite being considered essential due to time constraints, scarcity of resources, NDAs, etc. On one hand, computer-assisted translation tools usually provide features specifically created for ensuring consistency that derive from the tool's translation memory and the possibility of using parallel corpora. Image 27 is a screenshot directly taken from memoQ's QA settings that shows the different alternatives it includes in order to ensure consistency in the translation.

59 https://files.memoq.com/hubfs/eBooks/memoq_locland.pdf
60 https://files.memoq.com/hubfs/eBooks/memoq_locland.pdf

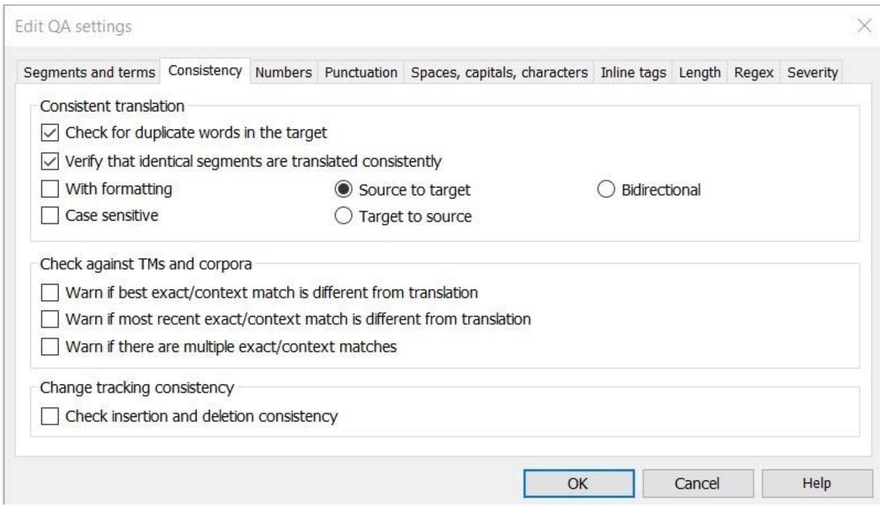


Image 27. MemoQ's consistency features

On the other hand, memoQ provides an extension called Qterm which was also included in the first survey as part of the options provided in terms of tools and received a fairly high number of responses ending up in the third position. QTerm works in a similar manner to SDL MultiTerm and, even though it is a web-based extension, it provides seamless integration capabilities with memoQ. Qterm's basic features are the possibility of adding entries with a single click, including definitions, setting up terms that are forbidden, including images along with the term, a semi-automatic terminology extraction feature, sublanguages, fuzzy matches for terminology to avoid redundancy and using prefixes to search[61]. Among the key and advanced features, they propose the possibility of adding a discussion for each term, a merging option for duplicates, and supporting "Graphics, video, audio or other media in descriptive fields: You can upload media files or reference documents to each entry. These are then shown or played in the default viewer."[62]

Therefore, as the last option clearly explains, Qterm also provides "access to audio, video and images", thus covering the third feature that appeared in the list of "essential functionalities". Nevertheless, the availability of these resources entirely depends on the client's "goodwill" and their implication in the localisation process. Once again, the best solution is to eliminate as much manual labour as possible in order to automate processes while modifying business practices. In this case, connecting memoQ with a content management system such as XLOC could be leveraged for these purposes as the

61 https://www.memoq.com/extensions#qterm
62 https://www.memoq.com/extensions#qterm

project manager would only need to link the assets and the terms present in the glossary while he or she prepares the localisation kit. The fourth feature was "to include quality assurance tools", and two comments specifically mentioned "Antidote" (from French participants). As previously explained, memoQ has different quality assurance options integrated, although it is beyond the scope of this thesis to analyse the quality of its performance. As explained on their website:

> MemoQ offers two kinds of quality assurance. The Automatic QA finds machine-detectable mistakes during and after translation. Also, the Linguistic QA module categorizes errors and provides detailed feedback to translators. Additionally, Antidote's corrector is also available in memoQ 9.1 and versions above.[63]

In the case of the fourth and fifth features resulting from the combination of "essential" and "useful", memoQ's integration capabilities with content management systems such as XLOC would help to track changes in assets and to access them if the localisers working in the project are provided access to the tool in question. Furthermore, the results from the second survey have proven the fact that overflows, overlaps and truncations are the most common bug encountered during the LQA phase. Currently, this issue is dealt with during the testing process and, although providing the maximum number of characters that can be displayed is essential, the results from the questions about business practices have proved that this is not always the case. As Jenny McKearney from XLOC explains:

> Another issue we see on a consistent basis is that a large percentage of localization bugs are graphical user interface (GUI) related. Overruns, overlapping and cut-offs are bothersome and difficult to fix the closer to final you get. By performing localization QA (LQA) on the GUI early on in the process, you can see where your potential issues lie and get them fixed before they become bugs that could delay the project or market launch.[64]

[63] https://www.memoq.com/integrations/quality-assurance

[64] https://www.gamesindustry.biz/articles/2021-04-21-the-keys-to-multilingual-game-development

The comments left by the localisers about missing features included 5 messages that asked for a tool to count the number of characters to be able to tackle this issue from its root. Excel provides sophisticated macros that can even count the number of characters per line inside a cell although this particular feature does not seem to be available in memoQ. However, the column with the number of characters can be imported from Excel and displayed for reference as previously mentioned and the system provides the possibility of checking the number of source and target characters in the status bar. Furthermore, memoQ provides the possibility to change the QA settings to establish an error notification when the translation goes over a pre-established character limit. As Image 28 shows, the tool can use the source text to set the length of the translated sentence using characters, words, or both. Other options are to set an average ratio and deviation and to set up warnings. MemoQ also provides more advanced options such as "pixel-based length check" to better calculate the space available depending on font size, style, etc.

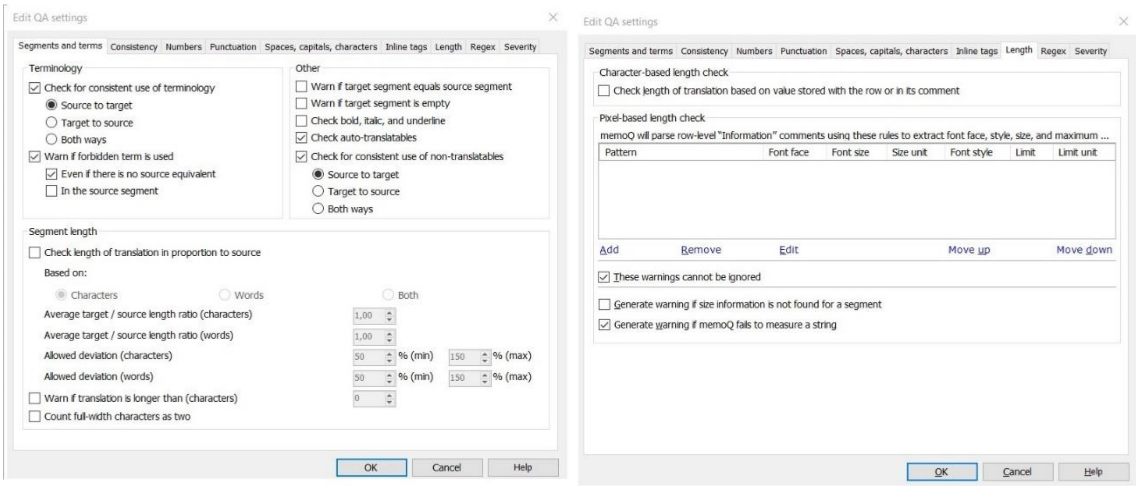


Image 28. Character limitation in memoQ

Once the QA settings have been updated, they can be applied to the document in question so that at the end of the translation process, it would suffice to run the QA check option and the report will show the segments that do not comply with the rules. During the translation, a warning will appear (if the box is activated when setting the parameters); said warning can be modified to become an error that will not allow the translator to validate the segment if necessary. Overall, the possibility of using the length of the source text to set up the maximum length for the target text (and specifying an acceptable deviation) might help reduce the impact of the lack of instructions in terms of the number of characters that can be displayed. Additionally, the most demanded feature in the "missing features section", communication options, is covered by the instant messaging options of memoQ although in this case, it would only include the translation team and the project manager. The tool's display options would also reduce the time spent switching between memoQ and the Excel file in order to see the comments. Finally, the "content connector" feature helps to closely monitor any new updates to the game as:

> There is a new update to a game. The content is pushed to a file folder, FTP site or similar. memoQ Content Connectors monitor this folder. When a new or modified file is detected, memoQ then automatically puts this into the relevant translation project and the localization phase of the update starts very soon after new content is created.[65]

5.4.6 Machine translation tools: Google translate

In the case of machine translation (Table 80), the results show that the highest percentage of regular users can be found in the group that did not have any type of official studies in translation followed by those with a master's degree. The lowest percentage belongs to the participants who only had specialised courses in translation. Among the participants with a bachelor's degree, we find the exact same percentage of occasional users and of those who have heard of machine translation but do not use it. In the first case is the highest percentage and, in the second case, the lowest. The 3 participants who did not know about the existence of these tools were in the groups that did not have any formal university education. In terms of patterns, we can observe that the respondents with a master's degree always have a percentage that is slightly above the average (although never more than an extra +0.25%).

Overall, our results show a gradual increase in the use of machine translation in comparison with the percentages obtained by previous researchers. The results of the QTLaunchPad survey (Doherty *et al.*, 2013, p. 9) showed that only 34% of the 500 respondents were using machine translation at the time. Zaretskaya's paper (Zaretskaya,

---

[65] https://files.memoq.com/hubfs/eBooks/memoq_locland.pdf

2015, p. 4) showed a slight increase in the number of participants who were using machine translation at the time of the survey, with a total of 36% of the 736 participants. Our findings show that 48.71% of the 620 localisers who took part in the first survey were either regular or occasional MT users thus proving the increasing acceptance of these systems and backing the trend signalled by the 2020 European Language Survey report's findings: "66% of agencies and 44% of in-house translation teams expect to invest in it in 2020" (2020, p. 50). In terms of the differences in the degree of adoption depending on the participants' educational background, Zaretskaya *et al.* (2017: 48) found the highest percentage of users among the participants with specialised courses (51.9%), followed by those with a bachelor's degree (46.5%), then the participants with a master's degree (46.6%) and finally those without official studies in translation (43.1%). However, the first survey showed that in the case of video game localisers, the participants with a bachelor's degree in translation appeared first with 55.64%, then those with a master's degree (49.55%), followed by those with specialised courses (47.83%) and finally those without any official studies in translation with 42.14%.

	None	**Specialised courses, seminars, etc.**	**BA in translation**	**MA in translation**	**All types (average)**
Use regularly	13.57% 19	8.70% 8	11.27% 16	11.86% 28	11.61% 72
Use sometimes	28.57% 40	39.13% 36	44.37% 63	37.29% 88	37.10% 230
Have heard of, but do not use	57.14% 80	50.00% 46	44.37% 63	50.85% 120	50.81% 315
Never heard of	0.71% 1	2.17% 2	0.00% 0	0.00% 0	0.48% 3
TOTAL	140	92	142	236	620

Table 80. Degree of adoption of machine translation tools by type of education

When we analyse the degree of adoption of MT systems by type of contract, the bottom row of Table 81, we can observe that the highest percentage of users belongs to independent freelancers (58.62%) followed by in-house translators in a translation agency (50.65%), even though the gap between the two results is noticeable. The rest of the groups are all below the average although once more, the difference in the case of freelancers working with an agency is negligible. It’s more than 5% in the case of freelancers working both independently and with an agency (43.31%) and 14,52% in the case of in-house localisers who work for a video game development company (34.67%). In the case of each option or tool, we can see that freelancers tend to avoid paid systems and “none” was their first choice whereas in-house localisers selected Google Translate API first. In the case of free machine translation tools, Google Translate appeared at the top of the list for every group and the results remain relatively similar with the highest percentage among independent freelancers (69.61%) and the lowest among freelancers working both independently and with an agency (52.94%). Finally, in the case of DeepL, the results follow the same pattern and are higher in the case of independent freelancers (48.04%) and lower for freelancers working independently and with agencies (32.35%).

	Freelancer	**Freelancer with an agency**	**Freelancer working both independently and with an agency**	**In-house translator in a translation company**	**In-house translator in a non-translation company**
None 46.89% (143)	50.98% 52	58.82% 30	44.12% 30	38.46% 15	30.77% 8
Google Translate API 39.34% (120)	36.27% 37	37.25% 19	39.71% 27	41.03% 16	57.69% 15
Google Translate 62.62% (191)	69.61% 71	60.78% 31	52.94% 36	66.67% 26	65.38% 17
DeepL 41.97% (128)	48.04% 49	43.14% 22	32.35% 22	38.46% 15	34.62% 9
Total: 305 (49,19%)	102/174 (58,62%)	51/104 (49,04%)	68/157 (43,31%)	39/77 (50,65%)	26/75 (34,67%)

Table 81. Machine translation tools by type of contract

Google Translate appeared as the most widely used machine translation tool in both cases (although in the case of paid tools, the first option was “none”) and it includes a different variety of features depending on the pricing. All the solutions are based on neural machine translation unless the language combination is not supported (in which case it will use Phrase-Based Machine Translation) and Google also provides the possibility of training the tool’s artificial intelligence to improve the quality of the translation and create a custom model (AutoML translation). The pricing is based on the number of characters and the company has 4 different plans: Translation API Basic, Translation API Advanced, AutoML translation, and Media Translation API. In terms of integration capabilities, Google Translate works using API keys and:

> Some Google APIs charge for usage, and you need to enable billing before you can start using these APIs. Enabling billing for the APIs that your projects use also has other advantages: Some APIs allow free usage up to a courtesy usage limit, and in some cases this free limit is increased when you enable billing.[66]

Once the key has been created, it can be used to directly “push” the strings into Google translate and then retrieve them. This system is used by numerous computer-assisted tools including memoQ, which includes 18 different options in terms of machine translation systems that can be set up from the resource console (Image 29). Furthermore, there are numerous tools available in Unity that can connect with Google translate in order to automatically translate the game’s content. Although it is beyond the scope of this paper to analyse the characteristics of neural machine translation and the quality of its output, it is not uncommon to be used by small developers who are not able to perform post-editing and are not able to pay for professional translation services (or do not see the need). These practices were mentioned in the LQA comments about linguistic bugs in order to explain the reason behind some “mistranslations” and the lack of translation quality that they may find. Besides, we must mention the numerous examples of games where the LQA phase was directly omitted as well and displayed flagrant mistranslations. Additionally, the potential lack of text linearity might hinder an NMT tool’s capacity of analysing the context in order to choose the appropriate term or

[66] https://support.google.com/googleapi/answer/6158867

gender causing even more linguistic bugs, to which we need to add the fact that the presence of variables in the text only complicates the task further. As mentioned before in section 5.3.2 and examined in a more detailed manner in section 5.4.8, game engines such as Unity even provide machine translation options that do need require to extract the string at all.

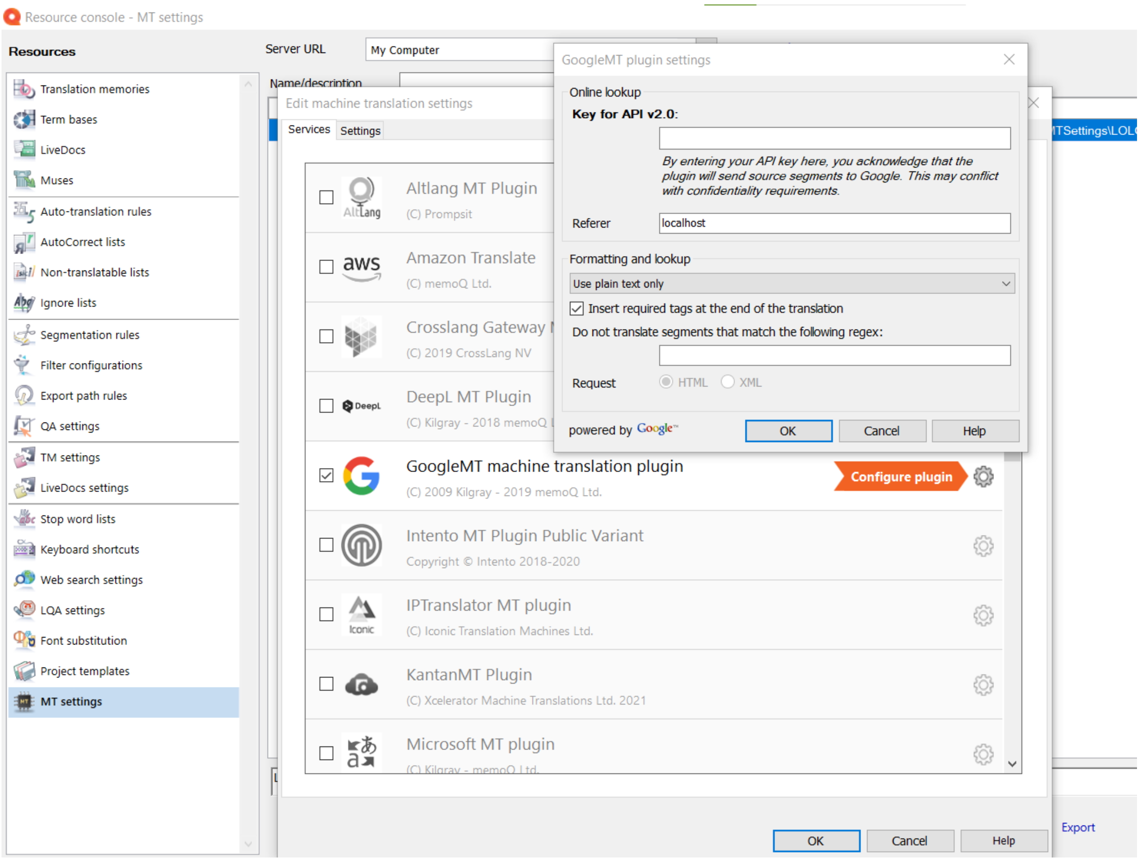


Image 29. Google Translate API in memoQ

5.4.7 Testing tools: Jira

Chapter 4 presented the results about the most widely used tools linguistic testers work with for reporting bugs and confirmed the prevalence of Jira, which had already appeared as the preferred tool for those localisers who also took part in LQA processes. In the case of the first survey, 54.62% (207) out of the 379 participants that undertook linguistic testing either regularly or occasionally used Jira, whereas 28.76% (109) did not use any kind of software. Table 82 puts together the tools' results from the second survey filtered by type of contract and shows that regardless of the group, the tool with the highest percentage of users remains Jira. We can observe that the highest results are those obtained by linguistic testers who work in-house in a translation company followed by those who work in-house in a video game development company. The percentages are lower in the case of all three types of freelancing options and the difference with the average result becomes noticeable in the case of independent freelancers (54.84% vs 79.82%). DevTrack seems to be relatively popular in the case of in-house testers whereas all three types of freelancers seem to favour “none” as their second most common answer (besides freelancers who work with an agency who chose “none” in third place). Overall, we find relatively similar results for the testers who work in-house either in a language vendor or a video game development company, which impacts slightly the results of the freelancers who work exclusively with an agency.

	Freelancer	**Freelancer with an agency**	**Freelancer working both independently and with an agency**	**In-house translator in a translation company**	**In-house translator in a non-translation company**
Jira 79.82% (439)	54.84% 34	69.05% 29	67.50% 27	89.39% 118	85.38% 222
DevTrack 24.73% (136)	20.97% 13	16.67% 7	12.50% 5	25.76% 34	29.23% 76
Proprietary software 19.09% (105)	14.52% 9	21.43% 9	15.00% 6	20.45% 27	20.77% 54
Other 15.45% (85)	14.52% 9	14.29% 6	2.50% 1	21.97% 29	15.00% 39
Mantis 14.73% (81)	16.13% 10	9.52% 4	5.00% 2	14.39% 19	17.31% 45
Hansoft 8.91% (49)	3.23% 2	2.38% 1	2.50% 1	9.09% 12	12.69% 33
None 8.73% (48)	27.42% 17	19.05% 8	22.50% 9	3.03% 4	2.31% 6
Total: 550	62	42	40	132	260

Table 82. Bug reporting tools by type of contract

Jira is a tool developed by Atlassian, the company that also owns Trello and Redmine (which appeared among the results) as well as various other tools and focuses on teamwork from a software development point of view. Jira is described on its website as a tool for software development used in agile processes that focuses on project and issue tracking. In an article published in their blog in 2018, they described 5 highlights of their product[67]:

> Build-your-own-boards: Customize your own workflow, issue types, and fields for the board you want and need. […] Redesigned issues: […] In the brand new issue design, it's now easier to understand what's most important and how to take action that will move work forward. All the Agile, with more agility: […] Teams can begin with a lightweight approach and then progressively add more features like backlogs, estimations, sprints, and more. All in a single click. Out-of-the-box filters: Generate a custom view of your next-gen board in seconds, with out-of-the-box filters by issue owner, epic, label, and more. […]

Although Jira has more applications than simply reporting bugs, it is beyond the scope of this work to analyse all of them. Therefore, we will now move on to the integration capabilities of this system with development tools (or any other tools), one of the main features showcased on their website, as they seem to provide integrations with many of the company's other tools[68]. However, when researching how Jira can be used specifically in video game development, we found a plug-in for Unity called Easy Jira available at the Unity Asset Store and created by a small studio of independent developers called Unreal Byte games. According to the product's description in the asset store, the tool allows users to create and assign issues as well as add comments to those issues. Customisable log files can also be attached to the issue in question and testers can "capture and attach a screenshot when the player sends the feedback form."[69] Image 30 is a screenshot that was provided on the same website and shows the appearance of the tool in Unity.

67 https://www.atlassian.com/blog/jira-software/the-new-jira-begins-now
68 https://www.atlassian.com/software/jira
69 https://assetstore.unity.com/packages/tools/integration/easy-jira-111744

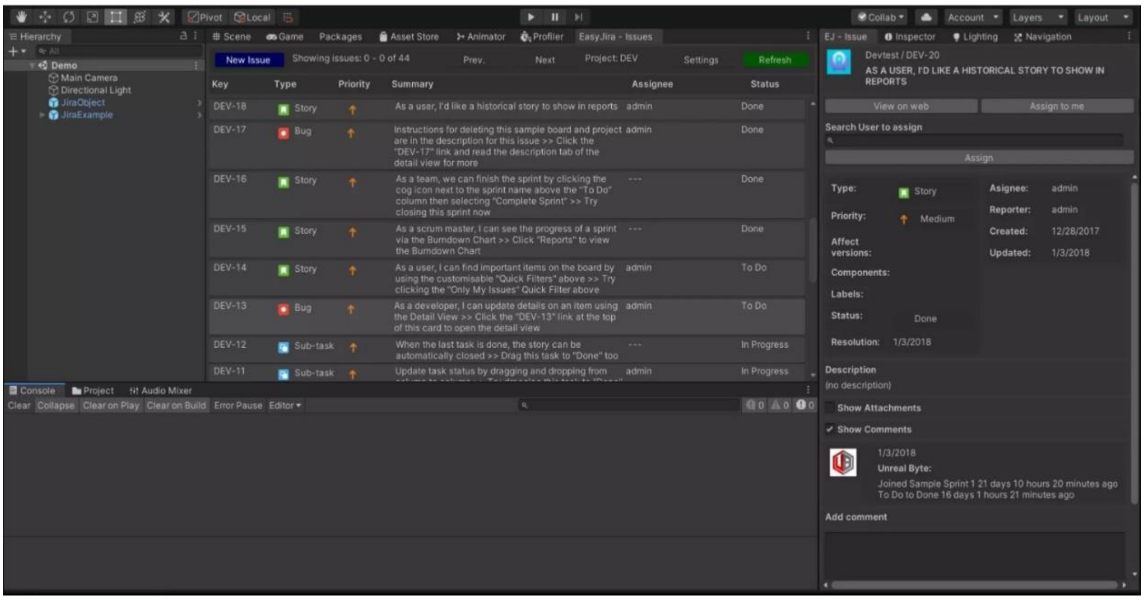


Image 30. Easy Jira in Unity[70]

5.4.8 Video game development tools: Unity

The following section will analyse the results in the case of the tools used for video game development and presented in Chapter 4 section 4.2.6. As we were able to observe, the overall results seemed to privilege “general purpose game engines” (Toftedah and Engström 2019, p. 13) and the use of middleware (see Chapter 1 section 1.3.1 for a detailed definition). We will begin therefore by studying the impact of the different actors involved in video game development when it comes to the degree of adoption of different tools. When we analyse the differences in tools’ usage depending on the type of contract (Table 83) we can observe that, once again, every group chose Unity as their first option although, in the case of in-house developers who work for a publisher-owned studio, the percentage for “a proprietary tool under NDA” is also high. Unreal Engine was the second option for independent developers (together with “other”), the participants who worked for an independent studio, students, and “other”. The respondents who worked in either publisher or platform holder-owned studios selected “proprietary tool” in second place and Unreal appeared third. Therefore, it seems safe to assume that at the moment using Unity for video game development is the norm and that analysing this tool for the purposes of this thesis is essential even though there might be many other options in the market.

70 https://assetstore.unity.com/packages/tools/integration/easy-jira-111744

	Independent developer	In-house in an independent studio	In-house in a publisher -owned studio	In-house in a platform holder-owned studio	Student	Other
Unity 71.07% (312)	85.11% 80	66.84% 129	53.57% 45	60.00% 9	93.75% 30	90.48% 19
Unreal Engine 32.35% (142)	19.15% 18	37.31% 72	27.38% 23	33.33% 5	50.00% 16	38.10% 8
Proprietary under NDA 18.22% (80)	4.26% 4	17.62% 34	40.48% 34	40.00% 6	3.13% 1	4.76% 1
Other 10.93% (48)	19.15% 18	7.25% 14	8.33% 7	13.33% 2	9.38% 3	19.05% 4
CryEngine 1.59% (7)	1.06% 1	1.04% 2	1.19% 1	0.00% 0	9.38% 3	0.00% 0
Total: 439	94	193	84	15	32	21

Table 83. Video game development tools by type of contract

Unity is a platform mostly known for the services they provide in terms of video game development although, according to their website, the system also provides solutions for other industries such as "Automotive, Transportation & Manufacturing; Architecture, Engineering & Construction; Film, Animation & Cinematics"[71]. Unity includes an immense variety of tools thanks to the different products that can be found on their website and all the additions that can be bought in the assets store. Therefore, the tool provides a comprehensive environment that allows users to create a video game with very little need for other programmes outside its interface, which eliminates compatibility issues. As we have seen in the analysis of the previous tools, Unity has

71 https://unity.com/

integration capabilities with almost all the specific systems that have been mentioned in this chapter. In the case of the two content management systems specifically created for video game localisation services, XLOC and Gridly (which will be analysed in the next section), the content can be integrated via API. Unity also provides several tree-based dialogue tools in the asset store that can be purchased and added to the system. Among the possibilities, the most simple one would be to use the tool to create the conversation and then simply take a screenshot of the resulting dialogue tree that can be included among the documents the localisers receive as reference material. Although extremely simplistic, this approach would exponentially simplify the task of grasping the gist of a conversation in video games that include complicated branching dialogues.

In terms of corpus usage and compilation, or terminology extraction and management tools, these features can be integrated into the workflow taking advantage of the content management systems' connection with computer-assisted translation tools that already include these solutions as built-in options or connected via plug-ins or APIs, etc. Another possibility is to directly connect Unity with the computer-assisted translation tool and then take advantage of all the options the tool itself provides. As briefly mentioned in the subsection about memoQ, there is a solution available at the Unity Assets Store called "Loclink – One-click Localization" that connects both systems using a plug-in. According to their website, among other things, this tool cuts down drastically manual labour as it allows to send the strings to memoQ and back once translated with a single click, has the possibility of specifying and setting limitations for the length of the translated strings, has a feature to track the progress of the localisation for each language, has a system for creating quotes and, most importantly, "[a]llows to provide context from Unity to a localization team"[72]. However, as this tool belongs to a video game localisation company, Native Prime, using their services seems to be the only way of taking advantage of the tool, which does not provide the possibility of streamlining its use. There is also the possibility of using the memoQ Content connector for this task, although it is necessary to create content connections that must include some sort of container or folder with the documents (or items) the source language, the specific source and target languages, the type of documents and the specific path to reach said folder:

[72] https://assetstore.unity.com/packages/tools/localization/loclink-one-click-localization-75295

> In memoQ server, you set up content-connected projects to work with the content source. When you set up a content-connected project, you choose a specific content connection using a particular content source. The content source is essentially a folder (or another type of container) in the content provider, containing items of source-language text. When a content-connected project is created, memoQ automatically picks up new and updated items in the content source. If there are new items in the content source, they are automatically added to the content-connected memoQ project. If some items are changed, the updated items are re-imported in memoQ, and pre-translated with translation that might already be written for the previous version of the same document. In the content connector, you specify content connections.[73]

In terms of other tools in the assets store that can be used for localisation purposes (besides Unity’s base features), we can find solutions such as I2 Localization. This system helps synchronise the strings with Google spreadsheets for fast and automated content updates, those spreadsheets can be subsequently downloaded and sent for localisation or accessed remotely by the localisers themselves. It also includes the possibility of exporting the text using CSV files and has an automatic translation option that uses Google Translate[74]. In the I2LocalizationManual they provide extensive information about this solution’s features as well as a link to a video on Youtube that shows how the tool works[75]. The main drawback of this tool from a localiser’s point of view is that it is based on spreadsheets and does not have any type of translation memory options and would require uploading the project manually to the tool and constantly checking for updates. However, in the case of memoQ, this system can be used to create a connection with its content connector. As the manual explains:

73 https://docs.memoq.com/ggl-tst/memoQ-server-deployment-tool/deptool-what-is-the-memoq-content-conn.html
74 https://assetstore.unity.com/packages/tools/localization/i2-localization-14884
75 https://www.youtube.com/watch?v=3J38Lf8W32k&ab_channel=InterIllusion

> While keeping the internal localization data in optimized custom files, spreadsheets can be linked as external sources for easy editing and sharing of translations. Google Spreadsheets, CSV, Excel and Open Office files can be synchronized with just one click.[76]

Due to the number of options, features and tools that Unity provides, we will concentrate on those that cover the specific needs that the localisers expressed in the section about “essential and useful features” as well as those from the “missing features” section. In the case of “the possibility of seeing dialogues in order” we have not been able to find any other solutions besides those presented in the subsection about XLOC and the possibility of using tree-based dialogue tools. In the cases of “terminology management and extraction tools to ensure consistency”, “to include quality assurance tools”, and "computer-assisted translation tools” these features depend on the connection of Unity to a computer-assisted translation tool or a content management system that would, later on, be connected to a CAT tool that includes all these options. Said content management tool will also provide the “possibility to track any changes in the source text files (management tools)”, “access to audio, video and images”, and “access to all the assets in their original form and divided by formats”.

When we analyse the possibilities for granting access to the actual visual environment of the game, Unity provides several options. First of all, its project’s user interface supplies a window and a preview option that allows displaying the scene as it would appear in the game itself. However, this can only be leveraged if the localisers are given access to Unity once the translated strings have been integrated into the system. This solution can cause numerous issues from the developer’s point of view such as the fact that the translator in question would need to have a certain degree of knowledge about Unity and the game to navigate the tool, the existing risk of potential unexpected issues if something is unintentionally modified, confidentiality concerns, etc. Therefore, in order to avoid having to grant the localisers access to Unity itself and leveraging the automated assets extraction and integration features, a build containing the localised strings could be created for testing purposes.

76 http://inter-illusion.com/assets/I2LocalizationManual/Features.html

As explained in the subsection about video game development processes and the advancement of the game depending on the build in question, Alpha still contains functionality issues that might impact gameplay and many of the assets are not yet final. Furthermore, reaching the area in question may prove a long and difficult endeavour and, depending on the game mechanics, the localiser will need to be able to trigger the exact combination of events as well. However, one of the options provided by Unity is the possibility to choose the scenes before compiling a build, thus allowing users to preview the specific part of the game in question without having to go through every scenario or level. This also implies the possibility of creating mini-builds that only contain menus or inventories in order to effectively avoid overflows, overlaps and truncations. As explained in the Unity manual:

> Scenes contain the objects of your game. They can be used to create a main menu, individual levels, and anything else. Think of each unique Scene file as a unique level. In each Scene, you will place your environments, obstacles, and decorations, essentially designing and building your game in pieces.[77]

This can be done directly from the "Build Settings window"[78] by simply not checking the box of the scene that will not be of use (Image 31). Among the other options that are included in the same window we find that, before compiling the build, the user has to select the target platform. Nowadays we can observe in the market a trend towards creating multiplatform games and, although this is not always the case, the differences in how each system displays the content might be noticeable. For instance, the number of characters that can be displayed in a menu created for a computer game will consistently be higher than in the case of a smartphone due to the size of the screen. Therefore, in the case of multiplatform games, ideally, the localiser would be provided with a partial build for each platform (or the platform with the higher number of restrictions) in order to be able to verify the correct implementation of the localised text. The system of creating different builds for each localised version and testing them is the current method used in LQA and adapting this practice to provide a WYSIWYG environment during the localisation phase can only be a possibility if the process is

[77] https://docs.unity3d.com/410/Documentation/Manual/CreatingScenes.html
[78] https://docs.unity3d.com/Manual/BuildSettings.html

automated to avoid impacting the production schedule and if menus or inventories are saved as separate scenes.

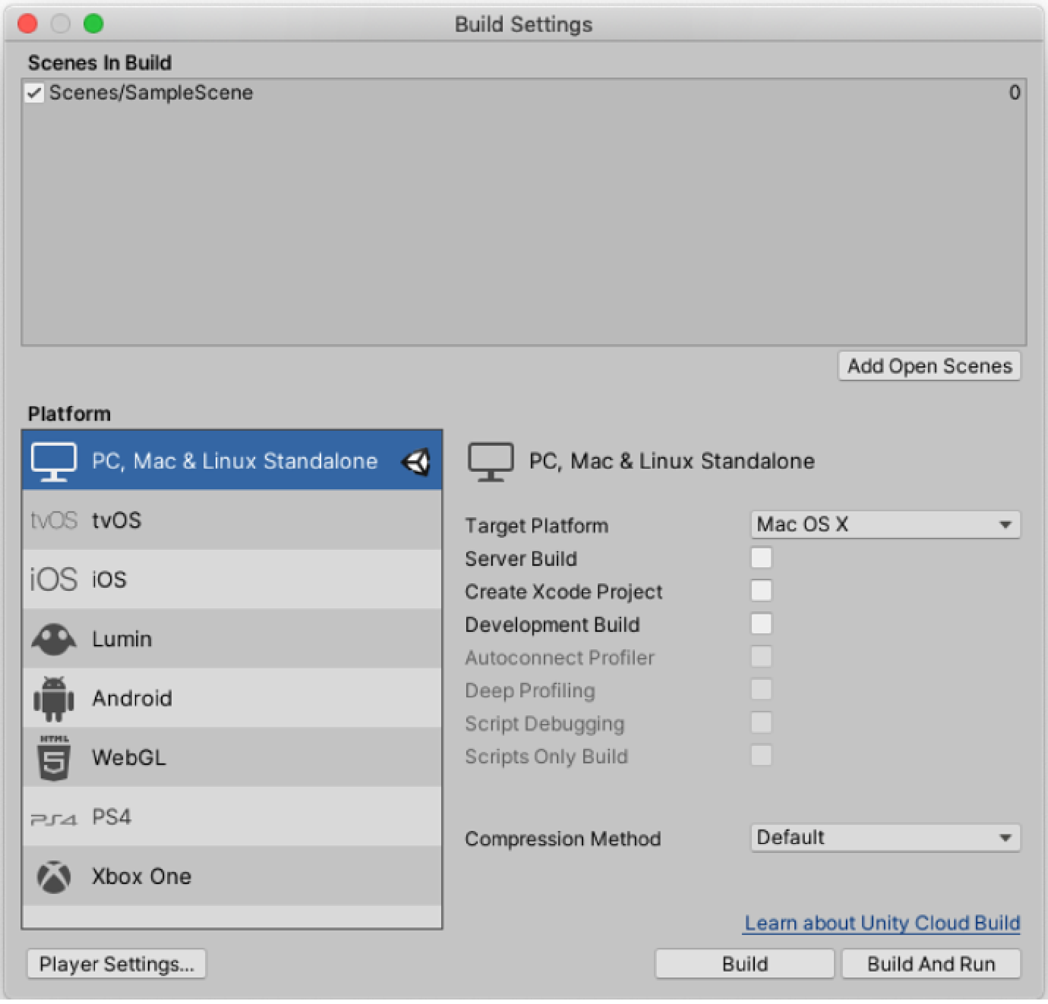


Image 31. Scene selection for build in Unity[79]

5.5 Additional findings

During our research, we came across another web-based content management system named Gridly that was launched into the market in late 2020 and has been steadily adding new features and releasing updates. Besides the novelty of the tool, the reason why it was not included at least in the third survey derives from the fact that we learned about its existence during a webinar that was held on the 14th of April 2021, less than a month before officially closing the questionnaire. The webinar, "Scaling game localization", was organised by Gridly in collaboration with Memsource in order to

79 https://docs.unity3d.com/Manual/BuildSettings.html

present the CMS system, how it works, and the release of a plug-in specifically designed for Memsource. Since then, the company has developed another plug-in for memoQ and has also created recently two more for Unity[80] and Unreal Engine[81] (in the beginning the integration was done via APIs). We were also able to schedule an online meeting with a representative of the company for an individual demonstration in order to learn more about how the tool worked and the possibilities it offered.

Gridly's user interface is designed to resemble a spreadsheet with multiple and customisable columns in order to display various items such as source languages, every target language as well as pivot languages (Image 32). They use what they call "dependencies"[82] to keep track of any modification in the source text and its translation. Using a colour system, the tool signals the need to update the corresponding translation by highlighting the cell in green if it is up-to-date, yellow if there was a change in the source language, or red if the translation is missing. By means of this system, they can also create various levels of dependencies by connecting the source language with a pivot language and, subsequently, said pivot language to every target language. Additionally, the content and the number of columns can be modified depending on the needs of each project and the tool also provides "granular user access control" to:

> Create individualized views for different user groups. Only expose a subset of the data to external vendors. Always track who changed your data and when.[83]

Other information that can be displayed in the columns includes the strings' IDs and the status of each segment so that project managers can track whether the translation of a string is pending, done, waiting for feedback, or if the string has been removed. Furthermore, Gridly has the possibility of using a column to specifically upload text documents, images, videos, and audio files as reference material for the localisation team. The procedure is simply carried out by adding a new column, choosing the type "files" and then dragging each element into the corresponding cell. Moreover, Gridly

[80] https://www.gridly.com/integrations/unity/
[81] https://www.gridly.com/integrations/unreal-engine/
[82] https://help.gridly.com/hc/en-us/articles/360013250777-Dependencies
[83] https://www.gridly.com/

allows quick in-line editing and has a built-in translation memory—besides those provided by the abovementioned add-ons for Memsource and memoQ. In terms of teamwork and collaboration, it is possible to create comments in each cell to communicate among the members. The specific window for the comments looks similar to those used in social media and allows users to keep track of all the questions and answers left for each string. In order to read the comments, you can either click on the symbol that appears in the right-hand corner of the cell or simply go to the menu on the right side of the project's window.

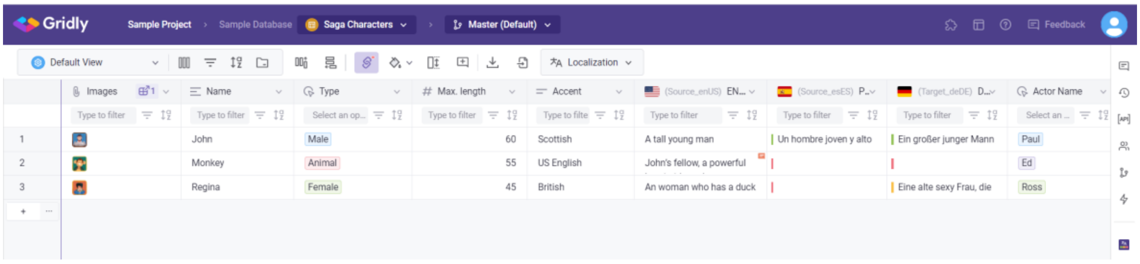


Image 32. Gridly's user interface

In the case of video game developers and producers, Gridly provides a centralised system for all the data from the game, works with APIs and also provides webhooks (also known as a "push API", a way to deliver data instantly to other applications). These can be used in combination with other programmes such as Slack (or similar ones) to, for instance, automatically generate a placeholder voiceover while waiting for the official recording. Gridly also allows updating the game in real-time by instantly publishing the changes in the game or website, therefore making LQA more agile and cutting down manual labour. In terms of compatibility, it can be used to work with games that will run on any type of platform and proposes the possibility of creating a localisation project directly by uploading a .json file. Gridly will simply extract the text and classify it in different columns including metadata such as the ID of the string or the path tag. However, the system seems to have difficulties if the .json file follows non-standard configurations (a very common occurrence in video game development) and the number of file formats accepted does not match the wide variety that can be found in the field, two facts that somehow put a damper to this novelty. In the case of tracking changes and testing new environments, the developers can use branching and version control to create an "alternative version" of the project to test it before merging the new branch or to "tweak project data in isolation or have branches translated independently."[84] In addition to the already mentioned features, and especially in the case of producers, Gridly's granular user access can be used to protect the content from unintentional changes which would encourage granting access to more team members. Other important tasks the tool can help with are:

> Track last-minute changes to script and voice-over in multiple languages. Track, audit, and undo changes if necessary in your browser. Push changes after LQA checks straight into code in one click.[85]

We will now proceed to contrast Gridly with the data obtained and presented in the previous chapters about business practices, localisers' attitudes towards features and functionalities, the prevalence and causes of linguistic bugs, and video game development processes to analyse what the tool offers to this specific industry. We can observe that this system has a variety of features to facilitate and automate the process in the case of current business practices and the potential problems derived from them. The direct connection between Gridly and the game engine (and CAT tools) considerably reduces the necessity of manual labour and processing.

Additionally, the possibility of directly uploading .json files with useful metadata and automatically creating a spreadsheet only contributes to further automating the process. The tool also allows users to directly upload from folders, which reduces the effort of providing reference material and classifying it, as the files can be displayed in the corresponding column. Moreover, the impact of a simultaneous shipment model and the constant changes that it entails is considerably diminished if we take into account a seamless connection for the updates and Gridly's dependencies system that highlights changes in the text. This last feature also simplifies the task of working with pivot languages, a widely extended practice as presented at the beginning of the chapter and in Chapter 2.

---

84 https://www.gridly.com/for-developers/
85 https://www.gridly.com/for-producers/

Gridly does not seem to have any specific features to provide "the possibility of seeing dialogues in order" besides the option of filtering the content (at the top of each column). Other solutions would be external and have already been presented in XLOC's analysis (section 5.4.1.1) and Unity's tree-based dialogue tools (section 5.4.8). "Terminology management and extraction tools to ensure consistency", "quality assurance tools", and "computer-assisted translation tools" are provided by the plug-ins Gridly proposes, which cost 99 euros each. In order to be able to use those plug-ins, the users need to have an account with either one of the CAT tools and the set-up process can be completed in under 5 minutes.

Once the connection is created, it suffices to select the content from Gridly that will be sent for localisation using the filters to only show the strings that have not been localised yet or those that need to be updated. Afterwards, the user must click on the Memsource or memoQ trigger button in the right-hand menu, click on push and then specify the project's settings (Image 33). The settings allow selecting an ongoing project to use the same parameters for consistency, use templates or create a new project. Afterwards, the project manager will specify the languages and will also be able to add additional columns besides target and source, add notes and set (manually) the maximum length for the translated strings. The text then gets sent to either Memsource or memoQ and, once the localisation process is finalised, there is another button marked as "deliver" that sends back the strings to Gridly.

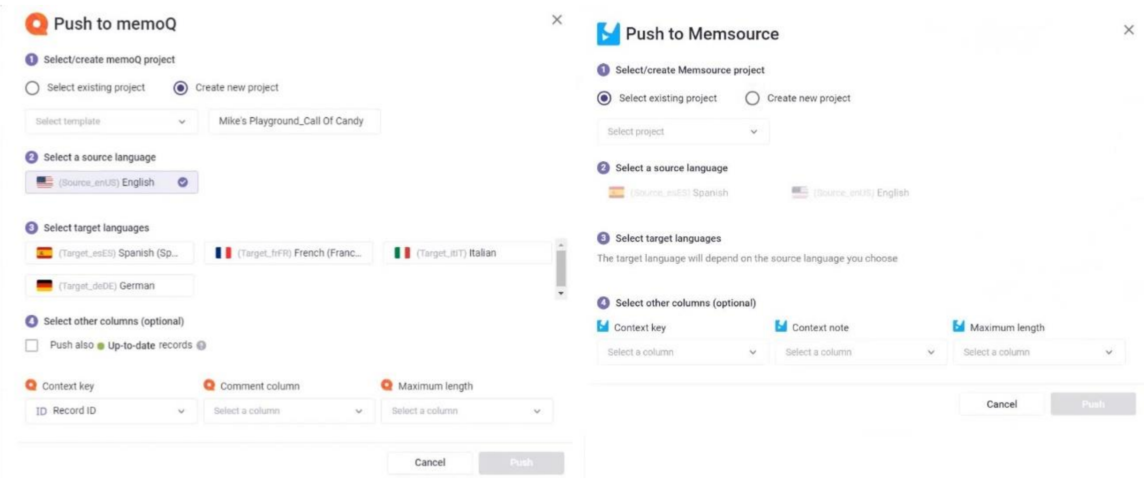


Image 33. Gridly's options for pushing strings

As mentioned in the description of the system, Gridly supports and displays images, video files and audio files (text files must be downloaded to be visualised) that can be directly added to an extra column for reference thus covering the third feature and functionality marked as essential by the localisers, “access to audio, video and images” (Image 34). However, even though the process in Gridly has been simplified as much as possible, the availability of resources ultimately depends on the developers’ or project managers’ willingness (or capacity) to supply them and the localisers’ access to this tool in particular instead of solely to the memoQ or Memsource project. In the case of the “possibility to track any changes in the source text files (management tools)”, Gridly has also proven to have a simple method to track changes in source (or pivot) languages. Moreover, each string also provides a “record history” that includes every modification (Image 32) in comparison to the original version as well as the name of the user who made the changes and the time and date which would cover the feature that appeared in the fourth place when we combined the results for “essential” and “useful”. However, the fifth and final feature that resulted from the said combination, “access to all the assets in their original form and divided by formats”, does not appear to be specifically addressed as everything is included directly in the spreadsheet and the only folders are those used for path tags (which basically put together similar types of strings). Finally, although it does not provide the same features as a specialised computer-assisted translation tool, Gridly has a translation memory and allows to upload TMX files in order to improve its output.

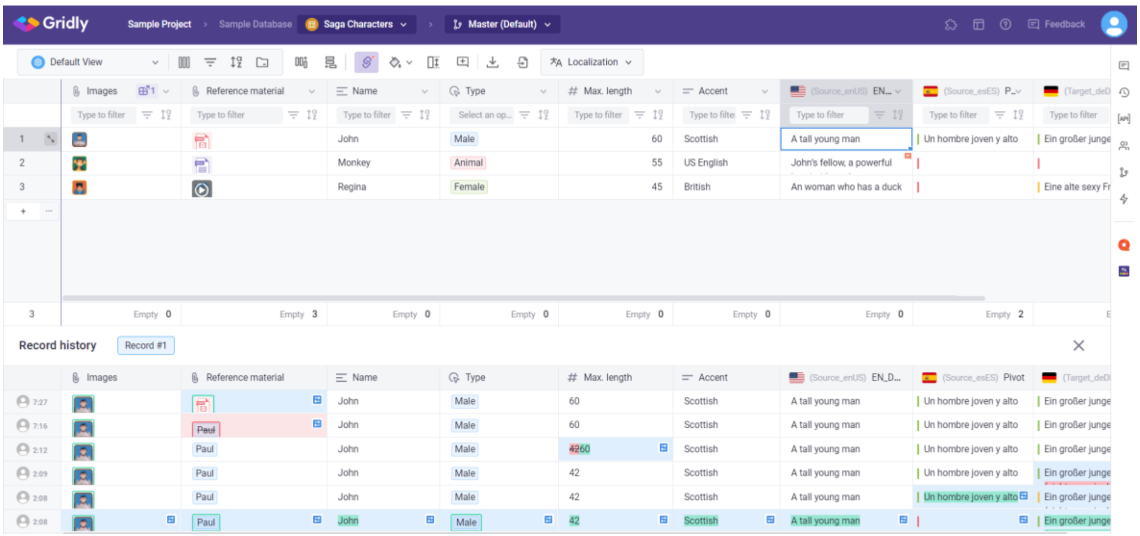


Image 34. Gridly’s options for reference material and tracking changes

To summarise, we have already gone through the features resulting from the combination of “essential” and “useful” that Gridly covers via the CMS itself or add-ons—namely “terminology management and extraction tools to ensure consistency”, “access to audio, video and images”, “to include quality assurance tools”, “computer-assisted translation tools”, and “the possibility to track any changes in the source text files”. We have also seen that the tool does not provide means to “seeing the dialogues in order” or “access to all the assets in their original form and divided by formats”. Even more, the tool seems to provide all the features included as options in the question presented in Chapter 3 section 3.3.6 besides the previous two. When we cross-reference the features offered by the tool with those that appeared in the different categories of the section for missing features we can observe, once again, that it seems to provide some type of solution or feature to address the majority of them. The most requested functionality, a feature to communicate with other team members, has been dealt with the possibility of creating conversations that are pinned to the cell in question, Gridly’s integration with Slack, and memoQ’s own features. The most common requests in the category “more visual access” were to be able to see the columns for other target languages, being able to export the extra comments left by developers or IDs into the CAT tool, or to have the reference material next to the text; all of them are possible either directly in Gridly or in combination with memoQ. The same can be said about the availability of quality assurance tools, as well as almost all of those included in the “miscellaneous” category (Appendix 5): filtering with regex, segmentations tools, tag processing, wordcount tool, etc.

During the two webinar sessions we were able to attend and a private online meeting we were granted, they performed a localisation task in real time and showed how the changes could be instantly seen on a website as a demonstration of its integration capabilities. Therefore, we decided to ask directly if it was possible to preview translation changes “on the fly” in a video game, something far more difficult to perform. In response, they explained that it could be possible if the developers were to set the appropriate environment by using “branch control”, although we were not shown an example. In the private demo (without exactly specifying how), they explained that it could be achieved by creating an alternative branch for testing purposes, the content could then be integrated into said “testing build” and displayed without affecting the main master branch. Finally, in the case of character limitations, besides all the features provided by Memsource and memoQ, they explained that they were currently working on a solution to automate the process. As previously shown in Image 32, at the moment the availability of a set number in Gridly depends on the developer or the project manager entering the figure manually or adding an extra column for reference, much like for the other tools already presented in this chapter. Overall, Gridly already has or is in the process of putting in place in the near future many of the necessary features to tackle the linguistic bugs usually encountered during the linguistic testing phase. The main issue, as it tends to happen all too often in the video game industry, is the client's willingness to provide more access as the possibility of leveraging all these features depends on how the project is set before the text gets to the localisers.

CONCLUSIONS AND FUTURE LINES OF WORK

The preliminary analysis of the data obtained by our third survey resulted in the realisation that it was necessary to review all the responses individually to counter potential bogus answers. The decision was taken for quality and consistency reasons given the lack of quantitative data in the field and with full knowledge of the time constraints that it will cause. Therefore, in addition to the fact that creating in-house tools can be expensive as mentioned in Chapter 1, we decided to concentrate on "off-the-shelf" tools or middleware. Consequently, once the idea of creating a comprehensive tool for video game localisation was discarded, the resulting aim of this dissertation was threefold; first, it sought to identify localisers' needs in terms of tools or suite of tools for video game localisation based on the professionals' attitudes towards them and the degree of adoption of the solutions available in the market. Secondly, it endeavoured to identify current business practices in terms of localisation, linguistic quality assurance, and video game development as well as the prevalence of linguistic bugs and the reasons behind them. Finally, by contrasting the previously collected data and creating a list of necessary requirements based on localisers' preferences, it attempted to determine possible technological solutions that may provide visual access to the game as well as to analyse the ergonomics of said tools.

As mentioned in the introduction and presented throughout this dissertation, the motivations for this project were the scarcity of papers covering these topics in academia, the lack of quantifiable data in the form of surveys centred on professionals in the field of video game localisation compared to other fields, the rapid technological evolution of the market in terms of developments in platforms, the apparition of new features in game engines, and the release of new content management systems focused on games. Although the approach used to distribute the surveys had to be modified twice initially in order to attain the expected results, these improvements provided a high number of answers from a large sample of professionals in a short period of time. Thus, after extracting the incomplete contributions and eliminating the responses that were deemed bogus, a total of 1609 answers were taken into account if we combine the results from the three surveys. Furthermore, the implementation of questionnaires allowed us to gather a high number of responses from a remarkably high number of different sources scattered all over the world. Given the unique opportunity the availability of this data offered, four multiple correspondence analyses (MCA) were carried out to identify hidden connections in addition to the results presented in the previous chapters: three individual MCAs per profession and one MCA using the common data of all surveys.

In order to obtain readable results, all the questions that contained more than 10 propositions—those related to nationalities, languages, resources, files, tools, and user preferences—could not be taken into account without reducing the number of options offered. In the particular case of the questions included in the personal information sections, we initially considered creating larger categories such as classifying the options depending on economic regions or continents in the case of nationalities. However, we soon realised that the task was too time-consuming and would reduce the significance of the data obtained from some of the already-mentioned questions. Due to the nature of the questions themselves, they had to be removed from the variables taken into account along with all the questions where respondents could provide multiple answers and those that could be skipped as they did not have an answer per participant. In the three individual MCAs, the previously mentioned conditions significantly reduced the number of variables that could be used and the resulting outcome simply reproduced the results presented in the previous chapters without providing any additional findings.

Finally, with a view to comparing the data collected from all the surveys in order to perform a common multiple correspondence analysis, besides removing the abovementioned variables, all the questions that were not identical in all three surveys had to be withdrawn and the number of options was dramatically reduced. Consequently, only 5 aspects could be studied in the common MCA: the age, the level of education, the years of professional experience, whether it was their main source of revenue or not, and the percentage of work related to their respective fields. Due to the restrictions and the nature of the variables, the results merely showed strong correlations between the respondents' age and the number of years of professional experience as well as between their workload and their main source of revenue. These results simply corroborate the importance of including those questions in all three surveys and the fact that they could be used to determine the respondents' level of expertise and add weight to the results presented in this thesis.

Even though these four multiple correspondence analyses did not provide any actionable insights, the implementation of surveys and the use of identical questions in the sections about the respondents' personal and professional lives have shown many similarities that result in a series of new contributions due to the novelty of this interdisciplinary approach. First and foremost, we can observe that the first question in every survey shows correspondences in the groups of age regardless of their activity, which derives from the relative youth of the field and its increasing popularity. The results about the participants' source, target and working languages when combined with the respondents' nationalities from the first two surveys have also confirmed the prevalence of English as a source language which, in the case of LQA, underlines its importance globally and, for localisers reflects the impact of the common practice of using pivot languages to reduce the costs of hiring professionals with unusual linguistic pairs. Furthermore, even though the findings corroborate the relevance of the E-FIGS language combination in the current market, there is a noticeable influence of other languages such as Russian, Portuguese, Mandarin, and Turkish. The presence of the latter, if combined with the fact that Turkey appeared as the third most common nationality for developers, may hint towards the birth of a future market.

The results obtained in the sections of the surveys created to collect data about professional information confirmed the diversity of educational backgrounds in the field of linguistic testing as well as a considerably low number of participants with studies related to languages. Conversely, the majority of the respondents who worked as localisers had some type of degree in translation and the vast majority of the developers had pursued Higher Education. Additionally, the questions included in these sections allowed us to define a common profile for the participants regardless of their field of expertise: a professional between the ages of 25 and 34 with 1 to 5 years of experience in their respective fields whose main source of revenue is either localisation, LQA, or video game development and spends more than 75% of his or her working hours in tasks related to the said professional field. This profile adds value to the data collected due to the high percentage of specialists and also contributes to outlining the characteristics of the actors that are part of the industry. However, the respondents' employment situation remains a distinct trait of each field and shows that whereas translation services are almost always automatically outsourced, development is performed in-house and linguistic testing remains in the middle with a slighter higher tendency towards outsourcing (52.73% versus 47.27%).

Furthermore, one of the key contributions of this dissertation related to current practices in the field was obtained by filtering localisers' access to reference material, videos, the game itself, and the maximum number of characters depending on their type of employment. This particular approach and its analysis have provided for the first time a quantifiable overview of the impact of different business practices resulting from the place of work. Thus, the results have shown that working as a freelancer with an agency tends to influence negatively the availability of reference material, videos, or the game itself whereas it increases the chances of having instructions such as the maximum number of characters that will display correctly on screen. Conversely, localisers working in-house in either type of company will have a much higher chance of accessing any type of reference material. Although the result itself was not completely unexpected, the contrast between the chances of receiving material in the case of an LSP and the material the LSP itself relays to the freelancer is sizable. Among the other discoveries that stem from this analysis, we can observe that providing video files for subtitling or localising the strings that will be subsequently dubbed seems to be a rare occurrence in general. Nevertheless, once we analyse the results per type of contract, the differences between in-house localisers and freelancers are even more pronounced.

Additionally, the percentage of localisers who continue working blind remains too high if we take into account the complexity of the task, all the technological solutions available to remediate it, and the consequences of this lack of context. Moreover, the combined findings of these surveys have confirmed the prevalence of simultaneous-shipment methods along with the constant changes that this release method entails. On top of that, the results related to the files that localisers received compared to those used by developers point to the persistence of conceivably unnecessary manual tasks. These tasks could potentially be automated in order to save time that could be used to, for example, set up procedures that provide more context to localisers. As we have seen, video game developers could take advantage of plug-ins to directly send the strings to CAT tools instead of manually extracting them from .json files into Excel files.

Although this will certainly change with time as the number of content management systems and their seamless connections increases, this systematic reduction of steps in the process and the time saved should be used to indicate—at least—the maximum number of characters that can be displayed in the case of strings located in areas of the game that suffer from space constraints.

The third section included in the surveys was devoted to the questions that were essential to achieve a considerable part of our main goals: identifying localisers' needs in terms of tools, establishing what was the most prevalent bug, and assessing technological solutions that could both reduce the impact of said bug and meet the required needs of translators. These "key questions" were used to analyse a topic that has not been covered previously in the field of video game localisation to combine information from localisers, linguistic testers—largely ignored until presently—and developers. Thus, the multidisciplinary perspective provides a holistic view of the processes and tools involved in conceiving, developing, translating, and producing multilingual multimedia interactive entertainment software. The third survey proved that the use of iterative development practices—or hybrid—results in a high probability of creating various milestone builds as well as smaller builds used for iterating on levels, features, or areas that will be playtested and then modified until attaining the desired outcome. These smaller builds could potentially grant visual access to localisers if they were made available, especially in the case where they show static assets or environments. Furthermore, the developers' answers about the beginning of the localisation phase indicate relatively early planning in terms of internationalisation in many cases—although other participants indicated otherwise in their comments. The combination of all these factors demonstrates that this lack of visual access is mostly based on business practices that derive from confidentiality concerns, even though localisers are asked to sign NDAs and the percentage of agencies receiving the game is also fairly high. Additionally, Unity has proved to be able to produce even smaller builds based on scenes, as we showed in Chapter 5 section 5.4.8.

This dissertation has also provided the first quantifiable data in the case of the prevalence of linguistic bugs encountered during the testing phases of the localised versions, and an insight into what the respondents considered their most likely causes. This data, which was not available until presently, portrays the unique perspective of those in charge of the final quality assessment and demonstrates the consequences of the lack of a visual environment and the lack of pertinent reference material. Thus, the second survey confirmed the hypothesis that the most common type of bug encountered by linguistic testers was “overflows, overlapping and truncations”, which was deemed to be the result of the scission between the text and the game itself. This was also confirmed by linguistic testers themselves and proves that the main constraint in the field is the impossibility of having visual access to the game and verifying changes “on the fly” or simply observing the string in context. This same issue was identified as one of the probable causes of many other linguistic bugs and underlines the necessity of finding alternative working methods to reduce its impact, thus justifying the end goal of this thesis. Additionally, the open question situated at the end of the survey where the participants could leave a supplementary comment has allowed us to better understand their working conditions, identify other issues they encounter, and observe organisational problems among other things. This leads to an accessory contribution made by this research: the quantity of raw data about reference material, missing features in translation tools, and LQA practices that will be made available in the form of appendices for future study.

The final aspect evaluated by the means of “key questions” included the query created following the matrix/rating scale format conceived to assess localisers’ attitudes towards features, functionalities, and suites of tools. It also included the possibility of specifying missing ones, which provided a framework that allowed us to evaluate the ergonomics of the tools available in the market according to their needs and preferences. Furthermore, depending on their place of work, these parameters can be fine-tuned to address their specific requirements in particular. As a matter of fact, one particularly interesting finding was the stark differences in preferences depending on the type of contract and the fact that freelancers showed more reluctance towards certain options whereas in-house localisers seemed to be more open to all types of solutions on the market. Regardless of their place of work, the participants selected “the possibility of seeing dialogues in order”, “terminology management and extraction tools to ensure consistency”, “access to audio, video and images”, “to include quality assurance tools”, and "computer-assisted translation tools” as the top five essential features and functionalities in this order. These options, once combined with the results of those deemed useful and the requests specified in the section about missing features (communication, character count, better QA, etc) provided the basis that constituted the parameters employed to analyse the tools studied subsequently. Moreover, these requirements could be extrapolated and used in future research in order to study the suitability of new technological solutions and their adaptability to the market.

The last sections of each questionnaire were devoted to the degree of adoption of various technological solutions available on the market and were fairly short in the case of the second and third surveys. The results of these two have shown the popularity of Jira as a bug reporting tool in the case of testers (and localisers as well) and the fact that developers do not work with XLOC and the tools seems to be only used by the localisation department. In the first survey, we could also observe a certain enthusiasm towards the potential advantages resulting from the implementation of tree-based dialogue trees to mitigate the effects of the lack of text linearity. Conversely, we were faced with the lack of adoption of tools such as those used for corpus compilation and terminology extraction and management, a fact that reflects a trend that has already been established by previous researchers in translation studies as presented in Chapter 5. Although these percentages echo the realities of translation in general, where they are seen as time-consuming processes, they are also the consequence of a common issue in the video game development industry: the lack of resources due to confidentiality agreements and business practices in general. Nevertheless, the resulting percentage of participants who took advantage of already compiled corpora (62.10%) is higher than what has already been reported in previous studies in TS, which might result from the gradual inclusion of corpora in university courses or the fact that some tools such as memoQ have added features related to corpus compilation such as LiveDocs. Additionally, the filtering options provided by SurveyMonkey allowed us to study in detail the degree of adoption of all the technological solutions depending on the educational background as well as examine each particular tool according to the type of contract. The latter remains an important contribution to the field of localisation studies and translation studies in general due to the scarcity of data on the topic; the former has allowed us to observe changes in time and the reality of the field.

The degree of adoption of computer-assisted translation tools shown in the results of the first survey, contrary to the previous tools, is remarkably high and scores above 90% if we combine regular and occasional users. In this particular case, by filtering the results according to the educational background, we were able to observe the positive impact of holding a degree in translation studies and the effectiveness of integrating these tools into the universities' curricula. Additionally, by dividing the results per type of contract, we can quantify the impact of business practices related to the imposed use of certain tools by LSPs. Conversely, the results also reflected the fact that in-house translators working in a video game development company may have access to proprietary tools instead of using other commercially available options. Regardless of their education or type of contract, the participants' answers about computer-assisted translation tools and the comments left in the section "other" allowed us to widen our research in terms of tools and pointed towards the increasing popularity of web-based localisation solutions such as Crowdin, another CMS that has been steadily diversifying its services and integrating video game localisation among its specialities. Notwithstanding, memoQ remained the most favoured tool by users of all types of contracts and educational backgrounds. Moreover, after analysing memoQ closely, it proved to meet many of the needs expressed by the respondents, including but not limited to, quality assurance capabilities, options to count characters and control the resulting strings' length, LiveDocs for corpus compilation and exploitation, the possibility of displaying various columns of an Excel file, etc. Furthermore, the combination of memoQ with tools such as Gridly could potentially cover all the requirements of the field if the developers upload enough reference material to the latter and localisers are given access to Gridly as well.

Finally, the third survey's results about game engines indicated the widespread use of Unity in the field of video game development in general followed by Unreal Engine, although we were able to observe that developers who work in publisher-owned studios have a stronger tendency to utilise proprietary software than the others (although their numbers were less representative than other categories). These findings allowed us to focus on Unity and the possibilities it offers both from integration capabilities and means to provide visual access to localisers. Among the proposals to address the latter, we put forward taking advantage of the tree-based dialogues tools that Unity provides for complex dialogues and subsequently adding screenshots to the localisers' reference material of the said dialogues to mitigate the effects of the lack of linearity. Furthermore, generalising the practice of making use of the tool's ability to create smaller builds that only contain previously selected scenes of user interfaces, menus, and inventories would dramatically reduce the percentage of text overflows, overlaps, and truncations encountered during the LQA phase. Moreover, seamlessly connecting Unity with Gridly—or any other future content management system—and memoQ would automatically reduce the amount of manual labour involved in the process and facilitate the task of providing the previously mentioned reference material. This connection would provide developers with all the advantages involved in the use of computer-assisted translation tools for consistency and the creation of translation memories that can be leveraged later on in the case of sequels of the game. Additionally, thanks to Gridly, this combination would provide a central hub to improve the communication between the different actors, keep track of the constant changes of both source and pivot languages, ensure that all assets are localised, and add an extra layer of protection against the integration of localised strings that have not been validated.

To summarise, the list of features that were deemed essential, useful or missing from current tools in the market if considered individually is: “seeing the dialogues in order”, “terminology management and extraction tools to ensure consistency”, “access to audio, video and images”, “to include quality assurance tools”, “computer-assisted translation tools”, “possibility to track any changes in the source text files”, “access to all the assets in their original form and divided by formats”, “a feature to communicate with other team members”, “more visual access”, “having the character limitations”, and those included in the “miscellaneous” category (Appendix 5)—filtering with regex, segmentations tools, tag processing, wordcount tool, etc. The combination of Unity, Gridly, and memoQ using add-ons covers all but four of them automatically and the company is launching new features on a monthly basis. According to the information they gave us during our exchanges, Gridly is working on integrating automatic character detection/limitation features soon and is capable of providing access to the game to see translations “on the fly”. Unfortunately, as explained at the end of Chapter 5, we have not been able to personally witness these two features in action. As for the remaining requirements that are not met by the combined use of these three tools—seeing dialogues in order and access to all the assets in their original form and divided by formats—the first could be solved by using tree-based dialogues tools and the second by simply uploading as much material as possible to Gridly via it “drag and drop folder” option.

In the case of the limitations of the present dissertation and future lines of work, we must start by acknowledging that due to the scope of the present dissertation and the time-consuming task of implementing and analysing three surveys, some links involved in the chain of video game localisation could not be examined. As presented in Chapter one, the process of developing a video game involves many actors both in the production and the localisation team such as producers, project managers, localisation project managers, outside service directors, managers of global production, or simply localisation engineers (in charge of implementing the localised strings). Therefore, their role and impact on the task at hand should be further researched in order to complete the picture of the localisation process.

The industry, and especially AAA studios, has a wide variety of positions that are involved in planning the creation of a multilingual product, the resulting files sent to localisers, and the subsequent implementation of the resulting translations. These actors will have different requirements in terms of the necessary features that should be included in a comprehensive suite of tools and insights about the consequences of current business practices. These needs may vary depending on the position and might encompass options such as prioritising version control, file format control, or the availability of metadata that allows to better understand the location of the string in question. Furthermore, due to the lack of academic research in the form of questionnaires and the fairly low number of interviews (compared to other types of translation) that have been carried out by scholars in this particular field at the time of this research, it becomes almost impossible to do so in the given span of time. Therefore, widening the scope of this research even more in order to take all the actors involved in the localisation chain process into consideration and fully conceptualise the project remains an interesting avenue of research.

Another limitation is the tendency towards secrecy that stems from non-disclosure agreements, the weight of proprietary tools, and the industry-driven approach of the field, which translated into the impossibility of fully examining tools such as XLOC, for example. Additionally, due to the lack of knowledge of the different possibilities offered by XLOC initially and the impossibility of including questions about its use in the first two surveys, asking localisers and localisation project managers about its degree of adoption in comparison to Gridly would have provided insightful data. The latter, which is rapidly and constantly developing new features in order to cater to the needs of the industry and the company's impact in the market should also be investigated and followed closely with a view to assessing its future prospects as Gridly was created recently and seems to be gaining popularity swiftly. Moreover, the implementation of a pilot study following the full process from development to production of a game and a subsequent follow-up with the respondents in order to study the benefits of our resulting combination of tools would complement our findings in the future.

Another obstacle encountered during this research is the rapid and constant evolution of the field, which results in the continuous appearance of new systems in the market. These advancements have a twofold impact: (i) the impossibility of keeping track of them as it is too time-consuming and, often, the information available about them is not sufficient for a complete analysis; (ii) the lack of adoption of said tools due to their novelty, which hinders the capacity to examine their performance from a user's perspective. Among these emerging systems we should mention Crowdin, which recently released its Unity plug-in and has started publicising its capabilities as a localisation management system specialised in agile video game development following the current trend of continuous localisation. This plug-in, similarly to the one developed by Gridly, allows to push and pull data directly from and to Unity and provides a central hub for strings and assets, including the possibility of localising images such as flags, etc. Furthermore, the tool seems to include translation memory, machine translation, and many of the features proposed by computer-assisted translation tools. Conversely, Crowdin seems to rely mostly upon crowdsourcing services and using fans as translators (which was the case for one of the *Minecraft* versions) by sharing the project with the community and allowing the fans to translate graciously. Nevertheless, the company offers the possibility of creating private projects and managing groups of localisers with an innovative approach that allows professionals to propose their translations and vote for their preferred options. Therefore, analysing all the new technological solutions that appear on the market, especially Crowdin, and comparing the features they propose to the list of requirements issued from our research should be considered in future studies once they are better established and have slightly reduced the number of new updates they release.

Besides budgetary constraints and the complexity of the process of developing, localising and producing a game, it is necessary to mention that the market is—or has been—shifting towards continuous localisation as a result of the increasing popularity of the GaaS (Game as a service) model. Due to the novelty of this model, there is no research in terms of its impact in the field of video game localisation or video game localisation practices. Contrary to traditional single shipment practices, game as a service relies on the monetisation of a game in order to generate revenue in the long run by constantly updating the content and offering new missions, assets, levels, etc. Often referred to as cloud gaming or gaming on demand, it is the model used by games such as *World of Warcraft* or *Candy Crush*. The need for constant updates and the frequency of new content releases accelerates the production and the LQA processes and increases the pressure on the localisation team. Consequently, companies such as Gridly and Crowdin are now advertising agile localisation services and pushing content management systems to seamlessly connect all the tools involved in the process to speed up localisation. Analysing this practice in particular and the effects it has on all the actors involved is an important avenue for future research. Additionally, as a consequence of this accelerated pace, the use of raw machine translation output by individual game developers and self-funded small studios needs to be considered in the future due to the rising popularity of NMT and the public's overly optimistic perception of its capabilities.

BIBLIOGRAPHY

Aleem, Saiqa, Luiz Fernando Capretz, and Faheem Ahmedb. 2016. “Game Development Software Engineering Process Life Cycle: A Systematic Review.” *Journal of Software Engineering Research and Development* 4(6):1–30.

ALPAC. 1966. *Languages and Machines: Computers in Translation and Linguistics*. Washington, D.C.

Arthern, Peter J. 1978. “Machine Translation and Computerized Terminology Systems - a Translator’s Viewpoint.” Pp. 77–108 in *Translating and the Computer*, edited by B. M. Snell. London: Aslib Proceedings.

Austermühl, Frank. 2006. “Training Translators to Localize.” Pp. 69–82 in *Translation Technology and its Teaching*, edited by A. Pym, A. Perekrestenko, and B. Starink. Tarragona: Intercultural Studies Group Universitat Rovira i Virgili.

Bartoll, Eduard. 2008. “Paràmetres per a Una Taxonomia de La Subtitulació.” POMPEU FABRA.

Bernal-Merino, Miguel Ángel. 2006. “On the Translation of Video Games.” *The Journal of Specialised Translation* (6):22–36.

Bernal-Merino, Miguel Ángel. 2013. “The Localisation of Video Games.” Imperial College, London.

Bernal-Merino, Miguel Ángel. 2015. *Translation and Localisation in Video Games: Making Entertainment Software Global*. New York: Routledge.

Bissell, Tom. 2010. *Extra Lives: Why Video Games Matter*. New York: Pantheon Books.

Blancafort, Helena, Ulrich Heid, Tatiana Gornostay, Claude Méchoulam, Béatrice Daille, and Serge Sharoff. 2011. “User-Centred Views on Terminology Extraction Tools: Usage Scenarios and Integration into MT and CAT Tools.” P. 10 in *Tralogy I. Métiers et technologies de la traduction: quelles convergences pour l’avenir?* Paris.

Carreira, Oliver, and Eugenia Arrés. 2014. “Video Game Localization Training on Offer in Spanish Universities at an Undergraduate Level.” Pp. 243–68 in *FUN FOR ALL*

Translation and Accessibility Practices in Video Games, edited by C. Mangiron Hevia, P. Orero, and M. O'Hagan. Switzerland: Peter Lang.

Carter, Ben. 2004. *The Game Asset Pipeline*. Massachusetts: CHARLES RIVER MEDIA, INC.

Chandler, Heather Maxwell. 2005. *The Game Localization Handbook*. 1st editio. CHARLES RIVER MEDIA, INC.

Chandler, Heather Maxwell. 2020. *The Game Production Toolbox*. 1st Editio. Boca Raton: CRC Press Taylor & Francis Group.

Chandler, Heather Maxwell, and Rafael Chandler. 2010. *Fundamentals of Game Development*. Jones & Bartlett Learning.

Chandler, Heather Maxwell, and Stephanie O'Malley Deming. 2012. *The Game Localization Handbook*. 2nd ed. Sudbury, MA: Jones & Bartlett Learning.

Chris Christou, Jenny McKearney, and Ryan Warden. 2011. "Enabling the Localization of Large Role-Playing Games." *Trans. Revista de Traductología* 15:39–51.

Christensen, Tina Paulsen. 2011. "Studies on the Mental Processes in Translation Memory-Assisted Translation: The State of the Art." *Trans-Kom* 4(2):137–60.

Christensen, Tina Paulsen, and Anne Schjoldager. 2010. "Translation-Memory (TM) Research: What Do We Know and How Do We Know It?" *HERMES - Journal of Language and Communication in Business* 23(44):89–101.

Christou, Chris, Jenny McKearney, and Ryan Warden. 2011. "Enabling the Localization of Large Role-Playing Games." *TRANS (Revista de Traductología)* 15:39–51.

Díaz Montón, Diana. 2011. "La Traducción Amateur de Videojuegos Al Español." *TRANS. Revista de Traductología* 15(Special issue on games localization):69–82.

Dietz, Frank. 1999. "Beyond PacMan: Translating for the Computer Game Industry." *ATA Chronicle* 28 (9): 57.

Dietz, Frank. 2003. "A Translator's Perspective on Games Localization." *MultiLingual Computing & Technology Published by MultiLingual Computing, Inc* 14(5):21–25.

Doherty, Stephen, Federico Gaspari, Declan Groves, Josef van Genabith, Lucia Specia, Aljoscha Burchard, Arle Lommel, and Hans Uszkoreit. 2013. *QTLaunchPad -*

Mapping the Industry I: Findings on Translation Technologies and Quality Assessment.

Domsch, Sebastian. 2017. “Dialogue in Video Games.” in *Dialogue across Media*, edited by J. Mildorf and B. Thomas. John Benjamins Publishing Company.

Dunne, Keiran J. 2006. “Introduction: A Copernican Revolution.” Pp. 1–11 in *Perspectives on Localization*, edited by K. J. Dunne. John Benjamins Publishing Company.

Dunne, Keiran J. 2014. “Localization and the (R)Evolution of Translation.” Pp. 147–62 in *A Companion to Translation Studies*, edited by S. Bermann and C. Porter. John Wiley & Sons, Ltd.

EGDF. 2019. *European Video Games Industry Insights Report*.

Esselink, Bert. 2000. *A Practical Guide to Localization*. John Benjamins Publishing Company.

Esselink, Bert. 2003. “The Evolution of Localization.” *The Guide from MultiLingual Computing & Technology*, 4–7.

Folaron, Deborah A. 2006. “A Discipline Coming of Age in the Digital Age.” Pp. 195–219 in *Perspectives on Localization*, edited by K. J. Dunne. John Benjamins Publishing Company.

Fulford, Heather, and Joaquín Granell-Zafra. 2005. “Translation and Technology: A Study of UK Freelance Translators.” *The Journal of Specialised Translation Issue* (4):2–17.

Di Giovanni, Elena. 2008. “Translations, Transcreations and Transrepresentations of India in the Italian Media.” *Meta: Journal Des Traducteurs = Translators’ Journal* 53(1):26–43.

Granell, Ximo, Carmen Mangiron Hevia, and Núria Vidal. 2017. *La Traducción de Videojuegos*. Sevilla: Bienza.

Gregory, Jason. 2018. *Game Engine Architecture*. Third Edit. edited by J. Lander and M. Whiting. CRC Press Taylor & Francis Group.

Hartley, Tony. 2009. “Technology and Translation.” Pp. 106–27 in *The Routledge*

Companion to Translation Studies, edited by J. Munday. London and New York: Routledge.

Hendrick, Arnold. 2014. “Project Management for Game Development.”

Holmes, James S. 1988. “The Name and Nature of Translation Studies.” Pp. 66–80 in *Translated!: Papers on Literary Translation and Translation Studies*. Brill Rodopi.

Holz-Mänttäri, Justa. 1984. *Translatorisches Handeln: Theorie Und Methode*. Helsinki: Suomalainen tiedeakatemia.

Honeywood, Richard, and Jon Fung. 2012. *Best Practices for Game Localization*.

Hughes, Chris, Mario Montagud Climent, and Peter tho Peschand. 2019. “Disruptive Approaches for Subtitling in Immersive Environments.” Pp. 216–229 in *Proceedings of the 2019 ACM International Conference on Interactive Experiences for TV and Online Video*. New York: Association for Computing Machinery.

Huizinga, Joahn. 1950. *Homo Ludens*. Boston: Beacon Press.

Hutchins, John. 1998. “The Origins of the Translator’s Workstation.” *Machine Translation* 13(4):287–307.

Hutchins, John. 2005. “Machine Translation: General Overview.” in *The Oxford Handbook of Computational Linguistics*, edited by R. Mitkov. Oxford Handbooks.

Iwabuchi, Koichi. 2002. *Recentering Globalization: Popular Culture and Japanese Transnationalism*. Durham, NC: Duke University Press.

Izushi, Hiro, and Yuko Aoyama. 2006. “Industry Evolution and Cross-Sectoral Skill Transfers: A Comparative Analysis of the Video Game Industry in Japan, the United States, and the United Kingdom.” *Environment and Planning A* 38:1843–61.

Jayemanne, Darshana. 2009. “Generations and Game Localization. An Interview with Alexander O. Smith, Steven Anderson and Matthew Alt.” *The Journal for Computer Game Culture* 3(2):135–147.

Jiménez-Crespo, Miguel Ángel. 2013. *Translation and Web Localization*. 1st Editio. Routledge.

Jiménez-Crespo, Miguel Ángel. 2019. "Localisation Research in Translation Studies: Expanding the Limits or Blurring the Lines?" P. 238 in *Moving Boundaries in Translation Studies*, edited by H. Dam, M. Brøgger, and K. Zethsen. London and New York: Routledge.

Juul, Jesper. 2005. *Half-Real Video Games between Real Rules and Fictional Worlds*. 1st ed. Cambridge, Mass, MIT Press.

Kent, Steven L. 2001. *The Ultimate History of Video Games*. New York: Three rivers press.

Kuznik, Anna, Amparo Albir Hurtado, and Anna Espinal. 2010. "The Use of Social Surveys in Translation Studies. Methodological Characteristics." *MonTI : Monografías de Traducción e Interpretación* (2):1–20.

Laakso, Mikael, and Linus Nyman. 2014. "Innovation Opportunities: An Overview of Standards and Platforms in the Video Game Industry." *Technology Innovation Management Review* (July):15–21.

Lagoudaki, Elina. 2006. "Translation Memories Survey 2006: Users' Perceptions around TM Use." Pp. 1–29 in *Translating and the Computer 28*. London: Aslib Proceedings.

Lagoudaki, Elina. 2008. "Expanding the Possibilities of Translation Memory Systems: From the Translator's Wishlist to the Developer's Design." Imperial College London.

Lal, P. 1964. *Great Sanskrit Plays in Modern Translation*. New York: New Directions.

Language industry platform (LIND). 2020. *2020 Language Industry Survey*.

Lewis, Derek Robert. 1991. "INK TextTools, Version 1.5." *ReCALL* 3(4):35–37.

LISA. 2003. *The Localization Industry Primer*. 1st Editio. edited by D. Fry. Switzerland : LISA.

LISA. 2007. *The Globalization Industry Primer*. edited by A. Lommel and R. Ray. Romainmôtier, Switzerland : Localization Industry Standards Association.

Localize Direct. 2021. *The Most Popular Languages for Game Translation Translation and LQA*.

Mangiron, Carme. 2021. “Found in Translation: Evolving Approaches for the Localization of Japanese Video Games.” *Arts* 10(1):9.

Mangiron Hevia, Carmen. 2004. “Localizing Final Fantasy – Bringing Fantasy to Reality.” *LISA Newsletter Global Insider* 13((1.3)):5.

Mangiron Hevia, Carmen. 2006. “Video Games Localisation: Posing New Challenges to the Translator.” *Perspectives: Studies in Translatology* 14(4):306–23.

Mangiron Hevia, Carmen. 2013. “Subtitling in Game Localisation: A Descriptive Study.” *Studies in Translation Theory and Practice* 21(1):42–56.

Mangiron Hevia, Carmen. 2016. “Reception of Game Subtitles: An Empirical Study.” *The Translator* 22(1):72–93.

Mangiron Hevia, Carmen. 2017. “Research in Game Localisation: An Overview.” *The Journal of Internationalization and Localization* 4(2):74–99.

Mangiron Hevia, Carmen, and Minako O’Hagan. 2004. “Games Localisation: When Arigato Gets Lost in Translation.” Pp. 57–62 in *New Zealand Game Developers Conference*. Otago: University of Otago.

Mangiron Hevia, Carmen, and Minako O’Hagan. 2006a. “Game Localisation: Unleashing Imagination with “Restricted Translation".” *The Journal of Specialised Translation* (6):10–21.

Mangiron Hevia, Carmen, and Minako O’Hagan. 2006b. “Game Localisation: Unleashing Imagination with “Restricted Translation.” *The Journal of Specialised Translation* (06):10–21.

Mangiron Hevia, Carmen, and Minako O’Hagan. 2013. *Game Localization Translating for the Global Digital Entertainment Industry*. 1st ed. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Mangiron Hevia, Carmen, Pilar Orero, and Minako O’Hagan. 2014. *Fun for All: Translation and Accessibility Practices in Video Games and Virtual Worlds*. Bern: Peter Lang.

Mann, Steve, Tom Furness, Yu Yuan, Jay Iorio, and Zixin Wang. 2018. “All Reality: Virtual, Augmented, Mixed (X), Mediated (X,Y), and Multimediated Reality.”

Mcgrath, Joshua. 2014. “The Game Development Lifecycle - A Theory for the Extension of the Agile Project Methodology.”

Munday, Jeremy. 2012. *Introducing Translation Studies*. 3rd editio. New York-London: Routledge.

Muñoz Sánchez, Pablo. 2017. *Localización de Videojuegos*. Seville: Síntesis, S. A.

O’Brien, Sharon, Maureen Ehrensberger-Dow, Megan Connolly, and Marcel Hasler. 2017. “Irritating CAT Tool Features That Matter to Translators.” *Hermes - Journal of Language and Communication in Business* (56):145–162.

O’Hagan, Minako. 2004. “Translating into the Digital Age: The Expanding Horizons of Localization.” in *The 9th Annual International Localisation Conference*. University of Limerick, Limerick, Ireland.

O’Hagan, Minako. 2006. “Manga, Anime and Video Games: Globalizing Japanese Cultural Production.” *Perspectives* 14(4):242–47.

O’Hagan, Minako. 2009. “Towards a Cross-Cultural Game Design: An Explorative Study in Understanding the Player Experience of a Localized Japanese Video Game.” *Jostrans: The Journal of Specialised Translation* 11:211–233.

O’Hagan, Minako. 2016. “Game Localisation as Emotion Engineering: Methodological Exploration.” Pp. 81–102 in *Conflict and Communication: A Changing Asia in a Globalising World*, edited by M. O’Hagan and Q. Zhang. New York: Nova.

O’Hagan, Minako, and Heather Maxwell Chandler. 2016. “Game Localization Research and Translation Studies: Loss and Gain under an Interdisciplinary Lens.” Pp. 309–30 in *Border Crossings: Translation Studies and other disciplines*, edited by Y. G. and L. van Doorslaer. Amsterdam: John Benjamins.

Odacıoğlu, Mehmet Cem. 2017. “Integrated Localization Theory of Translation Studies 1.” *International Journal of Comparative Literature and Translation Studies* 5(4):24–30.

Pérez, Cristina, and Leticia Sáenz. 2019. “Gender-Inclusive Language in Games and Its Localization Challenges.” *Multilingual* (May/June):32–35.

Politowski, Cristiano, Politowski Lisandra, Fabio Petrillo, and Yann-Gaël Guéhéneuc.

2016. “Are the Old Days Gone? A Survey on Actual Software Engineering Processes in Video Game Industry.” in *GAS16*. Austin, TX, USA.

De Prato, Giuditta, Sven Lindmark, and Jean-Paul Simon. 2012. “The Evolving European Video Game Software Ecosystem.” Pp. 221–43 in *The Video Game Industry. Formation, Present State, and Future*, edited by P. Zackariasson and T. L. Wilson. New York-London.

Pym, Anthony. 2004a. “Localization and the Training of Linguistic Mediators for the Third Millennium.” *FORUM. Revue Internationale d’interprétation et de Traduction / International Journal of Interpretation and Translation* 2(1):125–35.

Pym, Anthony. 2004b. *The Moving Text: Localization, Translation, and Distribution*. Amsterdam and Philadelphia: John Benjamins.

Ramadan, Rido, and Yani Widyani. 2013. “Game Development Life Cycle Guidelines.” Pp. 95–100 in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*.

Rivas Ginel, María Isabel, and Sarah Theroine. 2021a. “Jeux Vidéo, Localisation, Traduction Automatique et Problèmes de Genre.” in *Hispanisme et jeu vidéo*. Paris.

Rivas Ginel, María Isabel, and Sarah Theroine. 2021b. “Machine Translation and Gender Biases in Video Game Localisation: A Corpus-Based Analysis.” in *Colloque interdisciplinaire « Vers une robotique du traduire ? »*. Strasbourg.

Rojas, Raúl. 1997. “Konrad Zuse’s Legacy: The Architecture of the Z1 and Z3.” *IEEE Annals of the History of Computing* 19(2):5–16.

Roturier, Johann. 2003. “Video Games Localization: Constraints and Choices in the Industry.” Dublin.

Sandrini, Peter. 2008. “Localization and Translation.” *MuTra Journal LSP Translation Scenarios. Selected Contributions to the EU Marie Curie Conference Vienna 2007* 2:167–91.

Scholand, Michael. 2002. “Localización de Videojuegos (Translated by Lidia Cámara).” *Revista Tradumàtica: Traducció i Tecnologies de La Informació i La Comunicació* 1.

Singh, Nitish, and Arun Pereira. 2005. *The Culturally Customized Website: Customizing Websites for the Global Marketplace*. Oxford: Elsevier.

Studios, Blitz Games. 2011. “Project Lifecycle.”

Theroine, Sarah, María Isabel Rivas Ginel, and Aurélie Perrin. 2021. “Au Cœur de La Terminologie Du Jeu Vidéo. L’absence de Ressources, Frein Majeur Pour Les Traducteurs.” *Traduire* (244):27–40.

Timiani Grant, Flavia. 2001. “A Leisure Industry but a Serious Business.” *Language International* 13(5):16–19.

Toftedah, Marcus, and Henrik Engström. 2019. “A Taxonomy of Game Engines and the Tools That Drive the Industry.” in *DiGRA ’19 - Proceedings of the 2019 DiGRA International Conference: Game, Play and the Emerging Ludo-Mix*. Kyoto.

Torres Domínguez, Ruth. 2012. “2012 Use Of Translation Technologies Survey.” *Mozgorilla*.

Toury, Gideon. 1995. *Descriptive Translation Studies – and Beyond*. Revised ed. John Benjamins Publishing Company.

Trainor, Helen. 2003. “Games Localization: Production and Testing.” *Multilingual Computing & Technology* 14(5):17–20.

Uren, Emmanuel, Robert Howard, and Tiziana Perinotti. 1993. *Software Internationalization and Localization: An Introduction*. 1st ed. New York: Van Nostrand Reinhold.

Vermeer, Hans J. 1989. “Skopos and Commission in Translational Action.” Pp. 173–200 in *Readings in Translation Theory*, edited by A. Chesterman. Helsinki: Finnlectura.

Vieira, Else Ribeiro Pires. 1998. “Liberating Calibans: Readings of Antropofagia and Haroldo de Campos’ Poetics of Transcreation.” in *Post-Colonial translation Theory and Practice*, edited by S. (S E. Bassnett and H. Trivedi. London: Routledge.

Wang, Alf Inge, and Njål Nordmark. 2015. “Software Architectures and the Creative Processes in Game Development.” Pp. 272–85 in *International Conference on*

Entertainment Computing.

Zackariasson, Peter, and Timothy L. Wilson. 2012. *The Video Game Industry Formation, Present State, and Future - Introduction*. edited by P. Zackariasson and T. L. Wilson. Routledge.

Zaretskaya, Anna. 2015. “The Use of Machine Translation among Professional Translators.” Pp. 1–12 in *EXPERT Scientific and Technological Workshop*. Malaga.

Zaretskaya, Anna. 2017. “Translators’ Requirements for Translation Technologies: User Study on Translation Tools.” Universidad de Málaga.

Zaretskaya, Anna, Gloria Corpas Pastor, and Miriam Seghiri. 2015. “Translators’ Requirements for Translation Technologies: A User Survey.” Pp. 247–54 in *AIETI7: New Horizons in Translation and Interpreting Studies*, edited by G. Corpas Pastor, M. Seghiri Domínguez, R. Gutiérrez Florido, and M. Urbano Mendaña. Geneva: Editions Tradulex.

Zaretskaya, Anna, Gloria Corpas Pastor, and Miriam Seghiri. 2018. “User Perspective on Translation Tools: Findings of a User Survey.” Pp. 37–56 in *Trends in E-Tools and Resources for Translators and Interpreters*, edited by G. Corpas Pastor and I. Durán-Muñoz. Leiden/Boston: Brill | Rodopi.

APPENDIX 1 – VIDEO GAME LOCALISATION TOOLS

PERSONAL INFORMATION

What is your age? 17 or less 18 to 24 25 to 34 35 to 44 45 to 54 55 to 64 65 or older

What is your nationality?

What is your native language? (Please mark several if you happen to be bilingual or trilingual)

What language do you mostly translate from (main source language)?

What language do you mostly translate into (main target language)?

Do you work with other languages? Yes No

What other languages do you work with?

Have you received any formal education or training in translation?

BA in translation MA in translation PhD in translation

Specialised courses, seminars, workshops, etc. None

How many years of professional experience in translation do you have?

Less than 1 1 to under 5 5 to under 10 10 to under 20 More than 20

Is translation your main type of employment? Yes No If no please specify:

How much of your activity is in game localisation?

Less than 25% Around 50% More than 50% More than 75%

What is your type of employment as a video game translator?

Independent freelancer Freelancer working with an agency

Freelancer working both independently and with an agency Student Fan translator

In-house translator in a non-translation company In-house translator in a translation company

Do you usually work alone or in teams? Alone Teams Both

When it comes to the release of the video game, what kind of model do you usually work with? Simultaneous shipment or sim-ship (the game is still under development)

Post-gold (the game is already finished)

In the case of sim-ship, do you constantly receive modifications? Yes No

Is there any system in place, to your knowledge, that keeps track of these modifications? Yes No

TRANSLATION ASSETS

Which of the following file formats do you work with on a regular basis when it comes to video game localisation?

Pdf DTP files TMX files Excel MS Windows resource files

Word XML HTML Image files Rich/plain text

Audiovisual files Power point Other (please specify)

Do you usually have access to reference material? Yes No If yes please specify:

When it comes to subtitling or dubbing cinematics or spoken dialogues, do you usually have access to the video file? Yes No

Do you have any access to the visual environment when it comes to user interfaces, menus, text inside images; to avoid text overflows or truncations?

Yes No They provide the maximum amount of characters

Both visuals and maximum amount of characters

Do dialogues appear in order in the files you receive?

Always Most of the time Often Sometimes Rarely Never

Do you have to resort to controlled language because of variables and internationalisation issues? Always Most of the time Often Sometimes Rarely Never Please comment:

Please evaluate the degree of usefulness of the following features and functionalities, which might be included in a comprehensive tool for video game localisation.

	Essential	Useful	Not so useful	Not important	Inconvenient
Access to all the assets in their original form and divided by formats					
Possibility to track any changes in the source text files (management tools)					
Access to audio, video, and images					
The possibility of seeing dialogues in order					
Terminology management and extraction tools to ensure consistency					
Being able to use corpora					
Being able to compile corpora					
Computer-assisted translation tools					
Machine translation					
The possibility to combine corpora, CAT tools and machine translation					
Subtitling tools for spoken dialogues and cinematics					
To include quality assurance tools					
Bug reporting tools					
Work on online CAT tools that allow to work in teams					

Is there an essential feature or functionality missing?

ASSET EXTRACTION AND INTEGRATION, CONTENT MANAGEMENT TOOLS, PROJECT MANAGEMENT (PM) TOOLS, AND TESTING TOOLS.

Are you familiar with asset extraction and integration tools, content management tools, and project management (PM) tools?

Use regularly Use sometimes Have heard of, but do not use Never heard of

Which of the following asset extraction and integration, and content management tools do you use? Excel/Word + Visual Basic macros Localizer LocDirect

XLOC None Other (please specify)

Which of the following Project Management (PM) tools do you use?

GlobalSight Hansoft LTC Worx Microsoft Prudle I18N

Pairapharase Phrase Plunet Projetex Transifex

XLOC XTRF Star translation cloud None Other (please specify)

Do you also take part in the testing phase? Yes No Sometimes

Which of the following Bug reporting tools do you use? Mantis Jira Hansoft BugZilla

Test Track Pro DevTrack PR Tracker Bugtracker None Other (please specify)

TREE-BASED TOOLS FOR DIALOGUES

Are you familiar with tree-based tools for dialogues?

Use regularly Use sometimes Have heard of, but do not use Never heard of

Do you think you could benefit from them? Yes No Because:

WEB RESOURCES

What resources do you use on a regular basis? Bilingual dictionaries

Monolingual dictionaries Bilingual visual dictionaries Monolingual visual dictionaries

Comparable corpora Parallel corpora Thesauri Terminology databases

Image search engines Wikipedia Web forums Termbank portals

Consultation with experts Specialised search and metasearch engines

Other (Please specify)

CORPUS COMPILATION TOOLS

Which of the following types of corpora do you use?

Bilingual parallel corpora Multilingual parallel corpora

Monolingual comparable corpora Bilingual comparable corpora

Multilingual comparable corpora None

Do you use a tool for corpus compilation? Yes No

Which of the following corpus compilation tools do you use?

AntCorGen BootCat CLaRK Sketch Engine Other (please specify)

TERMINOLOGY EXTRACTION AND MANAGEMENT TOOLS

Are you familiar with terminology extraction and management tools?

Use regularly Use sometimes Have heard of, but do not use Never heard of

Which of the following terminology extraction and management tools do you use?

1. STANDALONE TERMINOLOGY EXTRACTION AND MANAGEMENT TOOLS

Anylexic LogiTerm Termologic TermWeb

TEXT Tilde None Other (please specify)

2. TERMINOLOGY EXTRACTION AND MANAGEMENT TOOLS FOR CORPUS

Intragloss OneClick Terms Lingvo.Pro None Other (please specify)

3. TERMINOLOGY EXTRACTION AND MANAGEMENT TOOLS FOR CAT TOOLS

crossTerm qTerm™ SDL MultiTerm Extract

QuickTerm None Other (please specify)

COMPUTER ASSISTED TRANSLATION (CAT) TOOLS

Are you familiar with computer-assisted translation tools?

Use regularly Use sometimes Have heard of, but do not use Never heard of

Which of the following CAT tools do you use? Déjà Vu MateCat MemoQ Memsource Omega T SDL Trados Similis SmartCAT Wordfast Other (please specify)

Which of the following localisation tools do you use? Catalyst Gtranslator Lokalize

MultiTrans Poedit SDL Passolo Other (please specify)

MACHINE TRANSLATION

Are you familiar with machine translation (MT) tools?

Use regularly Use sometimes Have heard of, but do not use Never heard of

Which of the following commercial (paid) MT tools do you use?

Google Translate API Bing Translator GramTrans IBM IdiomaX

KantanMT ModernMT SYSTRAN

Yandex.Translate Other (please specify)

Which of the following free MT tools do you use?

Apertium Babylon DeepL Google Translate Moses NiuTrans

OpenLogos PONS Online Translator Prompt Other (please specify)

APPENDIX 2 – VIDEO GAME LINGUISTIC TESTING

PERSONAL INFORMATION

What is your age? 17 or less 18 to 24 25 to 34 35 to 44 45 to 54 55 to 64 65 or older

What is your nationality?

What is your native or working language? (Please mark several if you happen to be bilingual or trilingual)

Have you received any formal education or training in linguistic testing?
BA in translation, language studies, etc. MA in translation, language studies, etc.
PhD in translation, language studies, etc.
Specialised courses, seminars, workshops, etc. None

How many years of professional experience in linguistic testing do you have?
Less than 1 1 to under 5 5 to under 10 10 to under 20 More than 20

Is linguistic testing your main type of employment? Yes No If no, please specify:

How much of your activity is in linguistic testing for video games?
Less than 25% Around 50% More than 50% More than 75%

What is your type of employment as a video game linguistic tester?
Independent freelancer Freelancer working with an agency
Freelancer working both independently and with an agency
In-house in a non-translation company In-house in a translation company

Do you usually work alone or in teams? Alone Teams Both

Do you just have to report the bugs, or do you also have to find a solution/suggestion to fix them? I only report the bugs I also have to find a solution/suggestion

Depends on the project

Which of the following Bug reporting tools do you use? Mantis Jira Test Track Pro
DevTrack Hansoft Bugzilla PR Tracker Bugtracker Proprietary software
None Other (please specify)
In your expert opinion, what are the most common types of bugs? (Drag the options to put them in order) Text overflows, overlapping or truncation. Mistranslations

Terminology inconsistencies Grammatical and typographical errors

Subtitling errors Confusing instructions

In your expert opinion, what is the main reason for text overflows, overlapping or truncation? Lack of visual environment for the translators

Insufficient instructions for the translators or failure to follow the instructions

Lack of translation quality All of the above

Don't know Other: (please specify)

In your expert opinion, what is the main reason for mistranslations?

Lack of visual environment for the translators

Insufficient instructions for the translators or failure to follow the instructions

Lack of translation quality Lack of context e.g. the strings aren't in order, etc.

All of the above Don't know Other: (please specify)

In your expert opinion, what is the main reason for terminology inconsistencies?

Too many translators working on the same project

Insufficient instructions for the translators or failure to follow the instructions

Lack of glossaries, character bibles, etc. All of the above

Don't know Other: (please specify)

In your expert opinion, what is the main reason for confusing instructions?

Lack of visual environment for the translators

Insufficient instructions for the translators or failure to follow the instructions

Lack of translation quality Lack of context in-text

All of the above Don't know Other: (please specify)

In your expert opinion, what is the main reason for subtitle issues?

Lack of visual environment for the translators

Insufficient instructions for the translators or failure to follow the instructions

Lack of translation quality IT-related issues

All of the above Don't know Other: (please specify)

Is there anything else you would like to add?

Yes No

Please leave your comment here

APPENDIX 3 – VIDEO GAME DEVELOPMENT TOOLS

PERSONAL INFORMATION

What is your age? 17 or less 18 to 24 25 to 34 35 to 44 45 to 54 55 to 64 65 or older

What is your nationality?

Have you received any formal education or training?

Bachelor's degree (3 years) Master's degree (3+ years) PhD

Specialised courses, seminars, workshops, etc. None

How many years of professional experience in video game development do you have? Less than 1 1 to under 5 5 to under 10 10 to under 20 More than 20

Is video game development your main type of employment? Yes No If no, please specify:

What is your type of employment as a video game developer?

Independent developer In-house developer in an independent studio

In-house developer in a publisher-owned studio Student

In-house developer in a platform holder-owned studio Other (please specify)

How much of your activity is in video game development?

Less than 25% Around 50% More than 50% More than 75%

When it comes to the localisation phase of the video game, what kind of model do you usually work with? Simultaneous shipment or sim-ship Post-gold

When does the localisation phase start?

Prototype build or First playable Vertical slice of Beautiful Corner

Alpha Beta Other (Please specify)

What kind of platforms do you usually work with?

Browser PC games Console games Mobile games

Downloaded/Boxed PC games VR games

What OS do you use day to day?

Windows Mac OS Linux Something else (please specify)

To facilitate cross-compatibility within the devteam do you use any kind of system emulation? Docker Openshift VM Ware Workstation

Virtual Box QEMU None Other (please specify)

What game development engine do you use? (mark several if necessary)

Unity Engine Unreal Engine CryEngine Source

Frostbite Fox Engine Unigine Creation Engine

A proprietary tool I can't name for NDA reasons Other (please specify)

What do you think of XLOC?

Never heard of it I've heard of it but I never use it I use it whenever I can

I use it occasionally I've heard of it but I don't like/want to use it

What's the attitude to testing within your current devteam?

We don't use unit/functional tests We try to add testing but it's more of an afterthought

We aim for 100% code coverage We write very defensive tests

What importance do you put on play-mode or screen comparison style testing?

Our existing testing methods do not allow for the possibility

We don't use them We use them in vital or complex areas

We cover 100% of gameplay

What are your primary day-to-day programming languages for game development? C++ C# Java Html5 & JavaScript Swift (Mac) Python Other (Please specify)

How do you isolate gameplay text from code? External files Data storage system (please specify)

What kind of external files? (Please mark several if necessary)

Json xml yaml txt other (please specify)

What is the game development process that you usually use?

Hybrid: is a combination of waterfall and iterative processes in the same project. Typically, the waterfall strategy is used during pre/post-production and the iterative is applied during the production phase.

Iterative: is a process that consists of developing software by repeating short cycles to deliver a ready-to-use feature each time. Agile software methodology follows this iterative approach, improving continuously and systematically its processes and practices.

Ad-Hoc: is a process that is created only for a specific project, without a previous definition. In the ad-hoc process, activities are defined on demand and the process changes to respond to punctual and contextual issues.

Waterfall (or predictive): is a sequential process in which the next phase is started only if the previous phase is completely finished, delivering business value all at once. This is the traditional game development process, requiring explicit requirement assessments followed by orderly and precise problem-solving procedures.

Don’t know Can’t say because on an NDA Other (please specify)

Is there anything else you would like to add? Yes No

Please leave your comment here

APPENDIX 4 – COMMENTS ABOUT REFERENCE MATERIAL S1

8/1/2020 2:02 PM	Only for AAA games
8/1/2020 10:22 AM	Style guides, glossaries, TMs from previous versions, sometimes design docs
7/31/2020 9:49 AM	TMs, screenshots, character descriptions, etc.
7/30/2020 10:10 PM	Word files, pdf files
7/30/2020 9:25 PM	It depends. Sometimes there is even access to the game. Some reference materials are always available. But as a rule there is not that much of it as I'd like to have. A usual practice are questions and answers (e.g. in a Google sheet).
7/30/2020 7:36 PM	Source Excel file
7/30/2020 7:12 PM	Public info like Store page, trailers and let's plays. Sometimes refs are provided by the devs: lore, character, mechanics descriptions, item pictures, etc.
7/30/2020 4:11 PM	Game development files, sometimes a few pictures and some context about the game, but not much and most of the time only the official website
7/30/2020 4:02 PM	Well prepared localization projects gives some images/videos as reference, as well as a word file explaining the backstory of the game. The best is when they provide an excel file with all the text in different tabs, in the right order. With the name and gender of the speaker as well as the name and gender of the person they are talking too. That is very rare though.
7/29/2020 8:08 PM	Scripts, images, videos, wikis
7/28/2020 9:18 AM	Developer notes, glossaries, demo/test version of the game.
7/27/2020 9:09 PM	Always, but not always useful, or have to ask for it.
7/27/2020 3:42 PM	Depending on the project, I have access to the Game Design Document, the game itself (usually a Beta version), translations of earlier games in case the game is part of a series, etc.
7/27/2020 2:13 PM	Query files, guidelines, files with string descriptions, etc.
7/23/2020 8:22 PM	glossaries, wikis, guides
7/23/2020 5:35 PM	Game Design Documents, character bios, screenshots, videos...
7/23/2020 10:03 AM	source files, videos, pictures, characters bios, SOPs and such
7/22/2020 10:44 PM	Glossaries, translation memories, various background information on the game, query logs, sometimes screenshots and videos
7/22/2020 6:29 PM	Glossary, sometimes some game overview, rarely anything related to game design documents
7/22/2020 6:18 PM	Images, full export, guidelines...
7/22/2020 5:08 PM	Sometimes, although clients normally are strict to share reference for confidentiality.
7/22/2020	Not much, usually just vague “style guides” put together by the PM

4:58 PM	
7/22/2020 4:32 PM	Client style guide
7/22/2020 3:02 PM	Character bios, game plot, etc,.
7/22/2020 1:30 PM	Game design docs, loc kits, character bios, etc.
7/22/2020 9:05 AM	Mostly Google Docs are used for glossaries and QA
7/21/2020 9:25 AM	I have access to images and videos of the game as well as specific guidelines.
7/20/2020 9:09 PM	sometimes beta version on Steam, query sheet, images. That's the ideal case of course.
7/19/2020 4:38 PM	Glossaries, style guides, videos, character sheets, etc.
7/18/2020 8:08 PM	access to translated light novels game was based on
7/17/2020 10:10 PM	Character sheets, summaries
7/17/2020 1:09 PM	Trailers, videos, screenshots, etc.
7/17/2020 11:13 AM	Pictures, Q&A answers
7/17/2020 8:39 AM	TMs, glossary
7/16/2020 3:35 PM	Legacy files, characters...
7/16/2020 8:52 AM	Images
7/14/2020 10:37 AM	Query sheets, screenshots
7/13/2020 10:40 PM	Screenshot, bios, etc
7/12/2020 5:59 PM	Occasionally, the game itself; usually, Steam page, online videos, screencaps
7/9/2020 10:07 AM	General design docs mostly. Reference material specifically for text not so much.
7/8/2020 1:48 PM	Older translation files
7/7/2020 11:53 AM	GDDs, character descriptions
7/6/2020 11:50 PM	images, videos, ref docs
7/6/2020 10:55 PM	Screenshots, character limitations (not always right), ROMs of the game demo
7/6/2020 4:10 PM	PDF with references + VPN access to client tools + queries
7/6/2020 2:28 PM	Glossaries, previous translations
7/6/2020 2:16 PM	MT and glossaries
7/6/2020 8:30 AM	Glossaries, some videos, the occasional beta ROM. For some game, reference files for the variables system. My main client is usually happy to pay for a few hours' worth of familiarisation, which makes a huge difference.

7/5/2020 1:02 PM	Screenshots, background information, images, videos
7/5/2020 9:36 AM	Clients usually send us files with information regarding the characters, the plot of the game and also when there is an update of the game, information about what is new and how the update affects what was already in the game.
7/5/2020 5:15 AM	mainly screenshots
7/4/2020 5:45 PM	Depends, glossaries or previous translation memories, for instance.
7/4/2020 2:34 PM	Text as dropin, images, orig. Games, source text, tm, previous games
7/2/2020 10:12 AM	Game APK, client provided data, existing released game
7/1/2020 2:27 PM	Some strings are contextualized with screenshots to have an idea. It's very grateful for the translators.
6/30/2020 4:35 PM	Working in house with developers and attending their meetings, full access to everything
6/29/2020 10:49 AM	Style guides, screenshots, character bios, synopses...
6/28/2020 8:16 PM	Game sumary, videos, bios, etc.
6/28/2020 7:39 PM	Images
6/28/2020 3:54 PM	Glossaries
6/28/2020 3:11 PM	Termbases - Glossaries - Style Guides
6/27/2020 1:47 PM	Design docs, glossaries.
6/27/2020 1:15 PM	about 60% of the time
6/27/2020 8:33 AM	press kits, TMs, documents, presentations
6/26/2020 1:38 PM	The additional file which the client provided.
6/25/2020 4:06 PM	Guidelines to the game, written or in video format.
6/25/2020 2:26 PM	Pre-release versions have of the game
6/25/2020 11:57 AM	Game scripts, videos, and I am able to play the games even prior to their release.
6/25/2020 10:49 AM	Dev comments, pictures, game builds, lore details, style guides
6/24/2020 9:04 PM	Old translation files
6/24/2020	Some visual assets, maybe some worldbuilding references. This is usually light

1:11 PM	and game knowledge usually comes from directly working on it.
6/24/2020 12:11 PM	It's usually not much, but I get at least a lockit
6/24/2020 11:49 AM	Style Guides
6/23/2020 7:36 PM	Images, videos, query sheets etc.
6/23/2020 6:37 PM	Texts explaining context and background of the characters. Also visuals of the characters, weapons, accesories.
6/23/2020 4:08 PM	Clients always send context information, some images, some videos, character bios... All this in order to provide some context to us, translators.
6/23/2020 12:31 PM	Terminology files by 3rd parties (Such as Sony, Microsoft, Google). Database online or in excel files with all the stringe related to a game and games of the past. Glossaries.
6/23/2020 11:56 AM	Depends on the project. For some RPGs I've received character descriptions and world/lore descriptions. Somtimes devs also send pictures/screenshots if asked about specific scenes/items. In all this time I've only received a steam key once.
6/22/2020 11:43 PM	glossary, style guide
6/22/2020 11:50 AM	website, glossaries, character charts, game
6/22/2020 9:14 AM	scripts, books, images...
6/21/2020 4:20 PM	glossaries, TM files, pdf
6/21/2020 11:12 AM	Termbases/Translation Memories of previous games, character descriptions
6/21/2020 3:31 AM	Original file, sometimes screenshots
6/21/2020 12:02 AM	Game builds, if ready. Not always the case.
6/20/2020 11:32 PM	Images, context info, characters info, mechanics info
6/20/2020 4:09 PM	Documentation regarding setting, plot, characters and suggestions from the publisher
6/18/2020 4:41 PM	Termbase, glossaries and style guides.
6/18/2020 1:57 PM	Game itself
6/18/2020 11:03 AM	Wikia related to game or YouTube videos
6/17/2020	Lore, wiki

10:45 PM	
6/17/2020 5:11 PM	Game source code via perforce, build tool/in-game console commands, design/QA docs, direct access to devs and sprint planning meetings
6/16/2020 2:31 PM	Style guides, in-game images, reference videos, world guides
6/16/2020 11:00 AM	GDDs, Q&A logs, game assets, game builds
6/15/2020 9:51 PM	Usually screencaps or video from the game in question
6/15/2020 6:27 PM	Game design or narrative
6/15/2020 4:02 PM	It depends from the developer.
6/15/2020 2:11 PM	Test builds, visuals, videos
6/15/2020 11:38 AM	Text, still images
6/14/2020 8:09 PM	Descriptions, comments, video about the game, sometimes the game itself.
6/14/2020 6:03 PM	lots of them: pics, video, comments and loc docs
6/14/2020 3:47 PM	Glossary, Style Guide
6/14/2020 11:28 AM	Most of the time I have access to the debug manual, notes from the French translator, explanations for some strings etc
6/14/2020 11:16 AM	Images, Betas, Steam codes, videos, glossaries
6/14/2020 10:35 AM	Scenario references, game's lore and some designs (characters/enemies sketches and conceptual designs of some stages)
6/13/2020 7:07 PM	Images, beta version
6/13/2020 6:54 PM	I often happen to have screenshots from the developer. Sometimes even plot overviews or game footage videos, but those are more rare cases.
6/12/2020 4:00 PM	Glossaries, game design documents, query sheets
6/12/2020 3:29 PM	everything we need
6/12/2020 2:04 PM	Client's reference material
6/12/2020 1:59 PM	Context provided in most cases
6/12/2020	Guidelines, glossary, etc

11:33 AM	
6/12/2020 9:11 AM	Reference material is often limited to links to product websites and videos e.g. on youtube.
6/11/2020 8:52 PM	Depends
6/11/2020 7:39 PM	Screenshots, short videos
6/11/2020 5:48 PM	For about 50% of game localization projects I receive some sort or reference material, usually screenshots or game design documents (Word/Power Point).
6/11/2020 2:44 PM	Images of characters, objects and some screenshots from the game, characters background information in order to adapt their tone of voice and a few lines about the plot of the game.
6/11/2020 8:48 AM	I usually have a dev build of the game
6/10/2020 7:29 PM	Images, EN text, EN game build
6/9/2020 1:11 PM	As inhouse localization specialist, we mostly have access to everything we need (through in-development-versions of the games), we also provide as much reference material to vendors that translate languages which are not done in-house
6/9/2020 11:52 AM	Access to dev server, brand/marketing briefings related to original content, ad assets, anything we need
6/9/2020 10:52 AM	Most of the time. But in many cases devs don't provide enough reference. Sometimes they don't answer or too late to answer queries.
6/8/2020 6:13 PM	video clips, explanatory documents
6/8/2020 5:15 PM	screenshots and/or videos.
6/8/2020 3:13 PM	Info of the game
6/8/2020 2:58 PM	sometimes ^^
6/8/2020 1:48 PM	Game summaries, glossaries, story boards, video support, etc.
6/7/2020 10:46 PM	Mainly graphics assets and videos
6/6/2020 3:56 PM	images, vocabulary etc.
6/6/2020 11:42 AM	PDFs , game specs etc
6/6/2020 9:22 AM	Files of other localizations; ingame builds on closed test server; dev contacts that supply required information upon request

6/5/2020 6:25 PM	Design documents, gameplay videos, game builds
6/5/2020 11:39 AM	Images of the game content, artworks etc, playable ROM
6/5/2020 11:34 AM	We make the reference material while translating to use for future reference.
6/5/2020 10:30 AM	PDF, images, Word documents, Excel glossaries or references directly in Excel
6/5/2020 10:06 AM	Translations of the same text in other languages
6/4/2020 6:00 PM	Internal project database
6/4/2020 5:30 PM	I'm taking reference materials directly from the game engine
6/4/2020 4:48 PM	Videos, images, summaries
6/4/2020 4:10 PM	Characters description, plot summary etc
6/4/2020 10:14 AM	Screenshots, comments
6/4/2020 8:57 AM	Client seldom provides references, I personally search for references as a normal step in my workflow
6/4/2020 8:33 AM	Design documents
6/4/2020 4:16 AM	I often need to ask, but most clients will then offer screenshots or playthrough videos.
6/3/2020 11:26 PM	Usually just source files, but sometimes the client provides a reference guide with characters, plot...
6/3/2020 4:26 PM	LocKits specilly designed for the work itself.
6/3/2020 1:12 PM	glossary, style guide
6/3/2020 12:18 PM	Access to development servers
6/3/2020 11:19 AM	It can be images, lore, descriptions, or access to the game itself (before release)
6/3/2020 3:09 AM	We usually have some basic documents in the beginning and will request more during the translation process
6/2/2020 10:36 PM	Excel files with glossaries and PDF files with guidelines

6/2/2020 7:33 PM	Depending on the task performed this may range from files with former translations, to glossaries or actual Translation Memories.
6/2/2020 11:24 AM	Translation memories in general referred to previous game or previous translations
6/2/2020 12:22 AM	game playthroughs or
6/1/2020 8:17 PM	screenshots
6/1/2020 8:06 PM	It's not very common, but sometimes you get lucky and the client sends some.
6/1/2020 6:34 PM	Guidelines and TMs
6/1/2020 4:09 PM	As I mostly work on the video games which were already released, I can find many online reference materials. Also, the vendor/client often provides additional reference materials such as descriptions or images when requested.
6/1/2020 12:25 PM	Existing translations, translation memories, graphics/art files
6/1/2020 11:08 AM	character bios, screenshots, walkthrough videos, the game itself, if ready for beta
5/31/2020 10:05 PM	It is usually provided by the client as it is no secret that providing reference material does maximise the possibilities of achieving the desired result quality-wise.
5/31/2020 8:04 PM	sometimes visuals, scripts, videos
5/31/2020 1:17 PM	Often we get a build of the game direct from devs
5/31/2020 12:09 PM	Not always, but I can usually request it
5/31/2020 11:26 AM	Video game itself
5/31/2020 11:23 AM	Builds of the game
5/31/2020 10:15 AM	Game test builds
5/30/2020 6:24 PM	I have access to a development build of the games I am working on.
5/30/2020 12:41 AM	Most of time the client only sends the source text and then if asked they send a few references
5/29/2020 1:56 PM	It depends on the client/agency. It could be original source files (with or without comments) or screenshots.
5/29/2020 12:07 PM	visuals
5/29/2020	Working in house of course I have access to the game so I can easily check the

10:39 AM	language in context.
5/29/2020 9:09 AM	list of characters, images
5/29/2020 6:40 AM	I usually have access to the game itself (I work with mobile games)
5/28/2020 4:34 PM	style guide; reference screenshots; dialogue architecture (rarely); build of the game (very rarely)
5/28/2020 4:31 PM	query sheets and terminology sheets
5/27/2020 2:59 PM	Translation guidelines, list of variables
5/27/2020 1:41 PM	Screenshots of gameplay
5/27/2020 11:59 AM	Glossaries, complete past projects, TMs.
5/27/2020 1:52 AM	Mainly PDFs, but overall little to none
5/26/2020 2:21 PM	Previous deliveries, previous games or specific documentation provided by the client
5/25/2020 3:49 PM	A few projects provides it
5/25/2020 2:39 PM	Images, videos or text files that may be used as guides
5/25/2020 1:47 PM	Normally they are images or PDF text in order to see how the original file looks like
5/25/2020 11:42 AM	Glossaries and criteria files mostly
5/25/2020 11:40 AM	Screen shots, term bases, translation memories with legacy translations
5/25/2020 11:37 AM	Either server based TM/glossary or a folder full of reference material.
5/25/2020 11:03 AM	It depends on the clients, some clients don't give any information other than the summary of the game, but we can always send queries.
5/25/2020 7:12 AM	Normally provided 50% and upon request
5/24/2020 11:07 PM	Videos, images, client communication and for mobile games, apks
5/24/2020 6:27 PM	Narrative content, sometimes even a test build

5/24/2020 5:15 PM	Most of the time, I have reference material and can understand the context.
5/24/2020 5:14 PM	Design documents provided by developers
5/24/2020 4:36 PM	Depending on what sort of content I'm working on, I might get text files explaining major plot points or game mechanics or character backgrounds or even client specific instructions in regards to formatting and the like . Sometimes we would get in-game videos to show combat or game environment. Maybe An Excel file that gives context to the translated strings; speaker, addressee, emotion or general context for the situation.
5/24/2020 2:17 PM	Style guides, pictures, screenshots, builds of the game, communication with devs
5/24/2020 2:05 PM	Documents and videos or access to game
5/22/2020 1:03 PM	Sometimes the client shares style guides, glossaries, screenshots or videos or the game or even a build of the game, in some (unfortunately) exceptional cases
5/22/2020 12:25 PM	Some WIP images sometimes, as well as content bible for some projects. Which is not to say there is never a case of games where we receive no reference material at all and must localize "to the best of our abilities".
5/21/2020 11:44 PM	Reference files with background on characters, images of items and even audio files and video files for subtitling and dubbing.
5/21/2020 1:44 PM	Usually, contextual information of the game, like descriptions of characters, places, skills, etc. Sometimes (but rarely) I have been provided with some script and visual support.
5/21/2020 12:45 PM	I usually get access to reference material regarding game setting/features/characters, sometimes there's localization guides depending on the client (this is more rare), and for small marketing requests ("Hey! New reward available!", "New content roadmap", "New trailer/sale/news", "Check out the Collector's Edition content", "Watch this developer interview", etc.) it's very common to get an image or video for reference. However, it's not common at all to have access to cinematics and/or gameplay footage in sim-ship projects.
5/21/2020 10:35 AM	Information about characters, game playing, etc.
5/20/2020 9:27 AM	I have access to reference material, but it's often not enough.
5/20/2020 6:55 AM	Mostly Word files, if I'm very lucky some images or even videos.
5/19/2020 10:57 AM	Rarely
5/18/2020 7:20 PM	It ranges from just the link to a youtube video or Steam page (which is like having nothing) to detailed reference files like pictures of the game, character background, historical content and access to the game itself.
5/18/2020	Background information, screenplays, sometime game softwares

4:42 PM	
5/18/2020 12:19 AM	Very limited reference material, like some PDFs about the game concept or with information on previous games in a series.
5/17/2020 1:38 PM	Images, videos, game development documents, access to full gamr
5/17/2020 1:33 PM	It actually depends on the developper...
5/17/2020 11:29 AM	Previous lore, videos, images
5/17/2020 11:05 AM	Marketing content, video references, developer files
5/16/2020 8:18 PM	Criteria files, images, character descriptions, story
5/16/2020 12:12 PM	Steam key, screenshots when asked, playthrough videos, information about the games
5/16/2020 10:56 AM	Access to previously translated content of the game on question plus glossaries
5/15/2020 11:06 PM	Glossary, old text file
5/15/2020 7:41 PM	BD and MD notes
5/15/2020 3:48 PM	Excel files, screenshots, game build
5/15/2020 11:11 AM	Speach pattern, relationship details, screenshots when available
5/15/2020 10:53 AM	Glossaries and past translations
5/15/2020 10:51 AM	Client can send us pictures ofthe characters and places, and also can provide some basic concepts of the game as a Word file. We can also ask questions to the client via a Google Doc query sheet.
5/15/2020 5:36 AM	in-house created spreadsheets, information directly from developers.
5/14/2020 9:12 PM	Mostly images, but sometimes videos.
5/14/2020 3:17 PM	There are some instructions or some information in string IDs
5/14/2020 2:41 PM	But it's not that common: build of the game, screenshots...
5/14/2020 12:42 PM	The game itself

5/14/2020 12:39 PM	Test builds, concept art, briefing and guidance materials, platforms and channels to communicate with developers
5/14/2020 10:22 AM	Yes, but not always
5/14/2020 8:36 AM	Design documents, legacy glossaries if any, internal presentations of new patch content, etc
5/13/2020 10:13 PM	Client provides the references
5/13/2020 5:30 PM	images, videos, power point presentations, character profiles
5/13/2020 5:02 PM	Character bios, VO scripts, style guides, videos
5/13/2020 2:57 PM	Style Guide, images, previous translations
5/13/2020 2:31 PM	Other than information easily found on a Google search, reference materials are rare.
5/13/2020 2:02 P	Manuals, images, source texts
5/13/2020 1:58 PM	When the game is out, accessible reference material are aplenty. Otherwise there'll be a query sheet to discuss and confirm things with the developers.
5/13/2020 1:57 PM	Mostly on bigger projects, but not very often
5/13/2020 1:40 PM	I have only worked on a single project (with different deliveries during the last few months) and they always provided me context of the scene as well as screenshots so I could imagine how the translated text would look like.
5/13/2020 12:29 PM	Just websites available publicly
5/13/2020 11:55 AM	Not always, but I often get game builds or at least briefs with screenshots.
5/11/2020 2:11 PM	Since I work for a company that also develops, I have easy access to confidential developing materials.
5/11/2020 8:02 AM	Videos, general information about new content, comments added to individual segments in CAT tools, Q&A files.
5/10/2020 1:14 PM	Images, videos, glossaries, legacy text and the video game ROM
5/10/2020 12:32 PM	Character bible, visual references, context... but the amount of reference is usually not enough
5/10/2020 11:51 AM	Game studios provide all sorts of reference material.
5/2/2020	Some clients have provided a copy of the game, or comments on the source text, sometimes additional documents that provide more info on their writing

5:27 PM	intentions
4/30/2020 6:55 PM	Relevant art, UI screenshots, reference images, sometimes a test copy of the game.
4/27/2020 3:16 PM	I always try to know what is the context of phrase in-game.
4/26/2020 5:27 AM	Screenshots of the game (images)
4/25/2020 4:23 PM	Game build, game descriptions, pictures, walkthroughs, videos
4/22/2020 11:02 AM	We develop our own games
4/17/2020 7:06 PM	I seldom receive reference material.
4/17/2020 4:41 PM	Usually a general description of the game, sometimes images or videos
4/15/2020 11:54 AM	Well it depends on what the devs give us! But usually, we have something! Of course, the bigger the game/studio, the better the reference material.
4/15/2020 8:14 AM	Older versions
4/14/2020 5:50 PM	It depends on the project. Usually, it is barely enough.
4/14/2020 5:06 PM	With some (rare) clients, images, quests/characters description
4/14/2020 5:05 PM	It really depends on the Devs team and the client
4/14/2020 3:04 PM	Character profiles etc.
4/14/2020 2:49 PM	Usually have access to images from the game, character bios and occasionally movies or test builds.
4/14/2020 2:19 P	TM, glossaries, past translations
4/14/2020 2:14 PM	Lore books, info about characters, original documents and so on
4/14/2020 1:48 PM	Chinese videogames and foreign videogames are reviewed before release in China by the General Administration of Press and Publication (GAPP)
4/13/2020 11:58 PM	Game
4/12/2020 8:29 AM	Terminology, style guide

4/11/2020 10:45 PM	Varies from client to client and project to project; can be anything from a basic style guide to extense GDDs, character references, screenshots etc.
4/11/2020 3:52 PM	I've only translated games that are already released, so I have access to the full game. Also, some text files have context if necessary.
4/11/2020 4:43 AM	Usually reference material will be provided but how useful it is is another question.
4/10/2020 6:48 PM	translator toolkits by developers, Q&A sheet, in-game descriptions
4/10/2020 1:55 PM	sometimes the general info of a game, but rarely very specific details
4/10/2020 11:34 AM	It really depends on the project.
4/10/2020 11:32 AM	It depends on the client.
4/10/2020 11:27 AM	Shared spreadsheets, word documents, queries sheets

APPENDIX 5 – COMMENTS ABOUT MISSING FEATURES S1

QUALITY ASSURANCE	
7/22/2020 6:29 PM	Real time grammar engines to deal with variables.
6/23/2020 6:37 PM	a tool tocheck and correct ortography/spelling/grammar
5/29/2020 12:07 PM	sophisticated spell-check tool so that you don't have to export the file as a text file to use Word's spell-check tool (which is a good one)
5/22/2020 3:40 PM	I think that a proper spelling check function is missing in CAT tools. You can use them, but they are very slow, tedious and usually wrong suggestions.
5/21/2020 11:44 PM	Spellcheck and if possible, but not essential, a platform for communication with members of the team.
5/14/2020 10:22 AM	Include a quality spellchecker as Antidote, a regex tagger module, a good recognition of numbers for similar strings where just the numbers change
5/13/2020 10:13 PM	Spellcheck, QA
5/13/2020 11:55 AM	Spell check, easy way to extract and send (share) files, competitive pricing and accessability.
4/10/2020 1:57 PM	Possibility to add spelling check tools such as Antidote

	COMMUNICATION
7/30/2020 7:12 PM	I'm not sure whether by "corpora" you mean public translation databases or a translation memory for a particular project. If the former, then a project-specific TM is essential to ensure consistency. Also, I find tag inlining tools and a comment/discussion section essential.
7/27/2020 9:09 PM	When sending queries to the team, having them display next to the relevant string can be useful.
7/23/2020 10:03 AM	encourage/enable/facilitate clients to add comments to anything they'd like to (and should) elaborate on to give access to as much info to the translator as possible
7/6/2020 4:10 PM	Queries platform for translators.
7/5/2020 1:02 PM	Q&A form
6/27/2020 5:56 PM	Commenting/communication system when working in teams.
6/24/2020 12:11 PM	Filtering with Regex, creating views a la MemoQ, comments
6/24/2020 11:49 AM	An easy way to ask questions about specific strings (for example, working on an online platform with a comment feature in each segment)
6/23/2020 11:56 AM	Some kind of chat function that allows for communication with devs and other translators.
6/13/2020 6:54 PM	A direct link for developers to the working files so that forwarding queries is more immediate.
6/12/2020 2:58 PM	The ability to speak with the devs or, at the very least, people in the localization team in a given company, who then talk to the devs and relay the translators' questions.
6/12/2020 9:11 AM	Integrate tools to manage queries accessible for the whole team, across all languages if applicable..
6/5/2020 11:34 AM	The opportunity to interact with other teams to ask for further information or discuss the design etc.
6/1/2020 8:06 PM	A tool to communicate safely and directly with the rest of the team if working remotely.
5/31/2020 10:05 PM	Any functionality that facilitates communication between PMs and translators within the software may be of some interest.
5/21/2020 11:44 PM	Spellcheck and if possible, but not essential, a platform for communication with members of the team.
5/21/2020 12:45 PM	A line of communication with someone heavily involved in the project on the client's side. It's important to use Q&A files to […].
5/17/2020 1:38 PM	Direct real-time communication both in-team and with developers
5/14/2020 1:19 PM	Would there be any chance to implement comments to strings that get directly extracted and forwarded to the developers so that they can reply to queries without having to fill an external query spreadsheet?
5/13/2020 5:02 PM	A tool to manage and answer queries (something better than an Excel sheet with hundreds of rows and columns)
5/13/2020 2:57 PM	Feedback sharing when working in team with the role of Proofreader
4/15/2020 11:54 AM	Well, Skype is pretty useful! As for myself, the agency I work with created an online Q&A platform, so everyone working on a project can check and ask questions directly to the devs, which is obviously very helpful!
4/14/2020 5:50 PM	Possibility to communicate easily with dev team

	PREVIEW MODE OR ACCESS TO THE GAME
7/21/2020 11:40 AM	Previsualisation tools to see how the translation looks ingame (in the case where a dev build is provided)
6/26/2020 9:38 AM	Being able to see where the translated text belongs in a game
6/23/2020 7:36 PM	This probably goes into the first item, but it would be great to have a preview function that actually works, especially for UI and OST.
6/15/2020 4:02 PM	On CAT tools, it would be great to have a preview on how a string appears in game. This was a feature of a client tool that I used in the past but it's under NDA unfortunately.
6/8/2020 2:58 PM	See what you are translating in its context (in-game)
6/5/2020 6:25 PM	Some sort of plug-in allowing to see text as it will be displayed in-game would be fantastic.
5/15/2020 10:51 AM	View pane that allows me to see the final looks of the translation (ex. HTML files).
4/14/2020 1:26 PM	The possibility to see the text in context, like displaying a screenshot in a column next to the text
6/13/2020 2:26 PM	Being able to play a game yourself is essential for any good translation.
5/29/2020 1:38 PM	Being able to play the scenes of the game as they are being translated

	MORE VISUAL ACCESS
7/30/2020 4:02 PM	A way to see what variables stand for. / Being able to see how a segment has been translated in another language.
7/6/2020 8:30 AM	When working with Excel, the ability to import additional columns beyond source and target as reference; my main client often provides a lot of contextual info in a "comments" column in the Excel, and being able to import this into the memoQ environment saves a ton of time.
6/29/2020 10:49 AM	Feature displaying the context often added by developpers in Excel files (for example memoQ's View pane)
6/26/2020 12:24 PM	The possibility of seeing who says what and to whom in a dialogue
6/24/2020 12:11 PM	Filtering with Regex, creating views a la MemoQ, comments
6/18/2020 1:57 PM	String IDs displayed somewhere would be nice for some context.
6/5/2020 10:30 AM	built-in context information about each quest or dialogue (tone of speech etc.)
5/31/2020 1:17 PM	To be able to view multiple languages in the same system, with a seed language change pushing notifications for branch languages
4/16/2020 7:35 PM	Broader context about the franchise when applicable
4/22/2020 11:02 AM	Multi-language simultaneous editor for linguists who can do multiple languages at once or who can read multiple source languages to translate to a target language.
4/14/2020 1:26 PM	The possibility to see the text in context, like displaying a screenshot in a column next to the text
6/11/2020 8:48 AM	Shared text files in order to apply changes on the fly

	CHARACTER LIMITATION
6/21/2020 7:10 PM	An easy way of setting character limits for multiple strings at once, both for PM and the translators/proofreader.
6/9/2020 4:56 PM	Character limits should be displayed
6/4/2020 4:10 PM	Character count tool
5/28/2020 4:34 PM	Length checkers
5/15/2020 11:11 AM	Automatic caracter count

	MISCELLANEOUS TOOLS
7/30/2020 7:12 PM	I'm not sure whether by "corpora" you mean public translation databases or a translation memory for a particular project. If the former, then a project-specific TM is essential to ensure consistency. Also, I find tag inlining tools and a comment/discussion section essential.
6/24/2020 12:11 PM	Filtering with Regex, creating views a la MemoQ, comments
7/30/2020 9:25 PM	Friendliness to dictation (correct capitalization, spacing), for example to Mac dictation.
7/22/2020 3:05 PM	Segmentation tools like divide/join, encoding issues report
6/15/2020 8:51 PM	Autonomous management of TMs without having to depend on PMs and the ability to work offline and mniipulate files and TMs as neesed.
6/14/2020 6:03 PM	processing tags is missing. Say, in MemoQ you have tags, but you cannot search for tags in TM. There are many other features I usually need for tags processing and have to export the text into xlsx to proceed.
6/14/2020 10:35 AM	A powerful text search engine that allows the translator (if possible) to look for a particular character interventions/text translated by a particular member of the team/multiple segments with different IDs.
6/10/2020 8:12 PM	Not exactly essential, but being able to track changes made during review/revision would be nice.
6/10/2020 7:29 PM	Ability to split/combine files for multiple translators
5/31/2020 9:08 PM	This may be included in the terminology management, but specifically a translation memory and term base that is editable by the translator is a very important part of any translation tool.
5/31/2020 12:09 PM	HTML/variables support to simplify work with code-heavy strings, Search system (SQL or else)
5/26/2020 2:21 PM	Wordcount tool, glossary tool
5/31/2020 1:17 PM	To be able to view multiple languages in the same system, with a seed language change pushing notifications for branch languages
5/25/2020 7:12 AM	Synchronized pivot languages
5/24/2020 3:34 PM	The possibility to combine CAT Tools, updated glossaries as termbases, with source files (for reference) always up to date. And, in case the client enters any changes in the project, the tool would send automated messages to the linguists involved.
5/14/2020 9:12 PM	Smart search and replace features, including advanced regex. Smart term base features, such as affixes, usage, definition, image references and alternatives with context categories.
5/14/2020 10:22 AM	Include a quality spellchecker as Antidote, a regex tagger module, a good recognition of numbers for similar strings where just the numbers change
5/13/2020 1:58 PM	Ensuring tools are as lightweight as possible and even more so when it comes to online ones

5/13/2020 11:55 AM	Spell check, easy way to extract and send (share) files, competitive pricing and accessability.
5/13/2020 11:44 AM	Dark/night mode
4/11/2020 10:45 PM	Being able to see and filter by string IDs, as these CAN be helpful for context (depending on how the developer has structured their text strings and IDs)
4/10/2020 2:35 PM	Suggestion : an ability to check how target translation matches with the corpora so that it can build a consistent Translation Memory for each type of game (a TM for FPS, or even one for a series of game)
4/10/2020 1:28 PM	Tagging of code

APPENDIX 6 – EXTRA COMMENTS S2

9/15/2020 6:00 AM	Corrupted characters are very commonly spotted issues as well (I'd place this the 3rd in the ranking,) especially with Asian languages that use unique non-Alphabet-based characters, like my language Japanese. I think "Lack of visual environment for the translators" is the biggest factor that is hindering the efficiency of videogame localization, and thus the factor that can leave the biggest (positive) impact to it if resolved/improved.
9/14/2020 3:37 PM	Please keep in mind that every project is different, I worked in some projects that we had no UI issues (overlaps/truncation) and the translation was poor, while other projects the UI was perfect, but the translation quality not so much. It's difficult to pin-point issues in a general manner.
9/14/2020 1:07 PM	As a localization tester there have been lots of problems with outdated and incomplete test cases that we had to fix and fill ourselves. This should be better provided by developer party.
9/10/2020 5:41 PM	The answer to the following question: "2. In your expert opinion, what are the most common types of bugs? (Drag the options to put them in order)" really depends on the language that is being tested. Languages with long words (like German) or long sentences (like Spanish) tend to have a lot of overlap/cut-off-related issues. The same issues are relatively rare for example in Chinese because of the character-based writing system which results in much shorter sentences.
9/1/2020 3:12 PM	The second question on last page is difficult to answer generally, as it will depends on several factors such as the intended market, the budget or the country of origin. For example, mobile games will have more linguistic issues (grammar, typo, mistranslations), due to hiring cheap translators; Japanese games will have more UI issues because they rarely consider other languages

	during development; AAA games will end up with more inconsistencies due to having several translators on the project, including freelancers who rarely have all the resources of the in-house translators.
8/31/2020 9:03 PM	When I entered the industry of video games, I thought it would a great experience. It turned out to be a horrible way to show what bad organisation is: - Translators have absolutely no information about the game, they usually translate standalone strings with a few notes here and there, and the same goes for the voice actors. - The devs are bullied by the editors who want to remove everything that would not be PC for different countries, creating new content for both the testers and translators to struggle with, and delaying the projects - Lack of instructions from the companies to the testers, we were given a computer and then were expected to know the whole process because we played a lot of games - English being the main base language for all translations, its genderless articles and pronouns cause a lot of issues for many languages like French and Spanish - The censorship of all the different countries and hypocrisy running in this industry complicates a lot the process developing a game, a country bans sex, the other bans violence, and you're stuck trying to find a common ground between your different markets for months What would solve this whole issue? A respect of the ESRB (accept that a game is -18 and allow drugs, sex and violence instead of creating a game rated "All" to make more money), remove censorship, have the test and other teams participate to meetings where they could have a glimpse at what they are expected to do, create trainings for all parties (a lot of translators and testers are newcomers because everyone wants cheaper and cheaper staff members, thus they fire and rehire every week), have at least 2 people by language look at the game to avoid mistakes and allow efficient proofreads, stop wasting all the budget on stand-by because of late implementations and updates.
8/28/2020 12:23 AM	I realise that many issues could be avoided if the LSPs had access to in game screenshots or in game context. Also, I believe is important to work with a team that supports and takes localisation seriously, as usually the budget for this is quite limited. Line break issues for Asian languages tend to be very abundant due to a missing autosizing tool that works properly.
8/27/2020 10:43 AM	If the translators had access to a character limitation on every string, this would limit the number of cut-off or overlap issues we encounter.
8/24/2020 11:11 PM	About the terminology, large projects seem to have too many hands on same piano as the main issue, but on small projects, it's usually because the translator

	goes back and forth through the translation fixing client's requests, and eventually skips it here and there, creating inconsistency. As reviewer and LQA of projects, I see this happen quite often.
8/24/2020 7:54 PM	In general when it comes to video game testing and the linguistic issues found, the main issue seems to be that there is a disconnect between the translators/linguists and the actual game environment. I find that if translators were provided with more documentation, including visual documentation about the game, its characters and features, and especially for them to know the size of text boxes and the likes, half of the mistakes found would no longer occur.
8/24/2020 4:35 PM	Some of my more recent projects have implemented a back-end tool that allows me to both toggle the game build to show key string identifiers in addition to text AND make revisions to the in-game script in real time and refresh it into the build. This is a MAJOR improvement over making revisions in a script and returning it to see if the revisions are fixed next round, especially as many of my developers have asked that truncation and overflow issues be fixed in-script where possible. The best possible localization tool for me would be an industry standard version of this translation management tool which allows testers to view script changes in the build in real time.
8/21/2020 1:16 AM	I would like to add that it is very common in the industry to have on-call contracts. You are not a freelance but you are no in house either. You are paid by the hours you work. You have a contract that attaches you to the company but you are just paid for the hours done.
8/17/2020 5:13 PM	Payment in the localisation of videogames is too low, be it for testers or translators. Companies are not willing to pay for experience. Cheap will most likely always win.
8/15/2020 7:52 PM	Video games are often created without localization in mind (often the source of display or grammar issues). Time is often a factor too (speed almost always prevails in video game development).
8/15/2020 7:47 PM	Working in AAA structure with vertical hierarchy made it more difficult to communicate with translators and voice talents from outside agencies. In my opinion this is the main hurdle to overcome regarding LQA and ensuring a better localization quality for video games. Committing mistakes is a thing, but having the ability to be reactive enough to correct said mistakes is just as much if not more important.
8/14/2020	I would like to mention that there was no suitable answer for me in this

7:14 PM	question "What is your type of employment as a video game linguistic tester?". I was a Full-time employee in a company that has translation branch
8/12/2020 10:56 PM	In my experience, I have noticed many times that often translators are not familiar with common videogame terminology or common settings like sci-fi, fantasy, military and so on. All of this together with the lack of context and visual lead to a translation that has to be almost completely corrected. It would be easier to directly translate from the videogame.
8/11/2020 6:03 PM	Context and guidance for the translation team is key - in my experience, translators that have had exposure to a game are able to deliver a much higher base line of quality which in turn results in a much better end result. If that's not possible though, anything helps from general information and design documents to character limits and hard restrictions. Linguistic Quality Assurance will improve upon the base that they receive, but to maximize the quality of the localized end result we want to start with a high quality translation.
8/10/2020 4:32 PM	Although it might go in a different category, voice over acting has a strong presence in video-games nowadays. So often I dare to say it probably represents 40% of the linguistic issues.
8/10/2020 2:44 PM	I believe time constraints play a very big role in both translation and technical issues arising. Moreover, in a good 50% or more of the projects I've worked on, the character limit wasn't specified, which led to overlapping and/or cuts once implemented in the game.
8/9/2020 2:43 PM	I cannot add too much due to NDA, but deadlines are usually a problem.
8/8/2020 10:54 PM	Lack of product knowledge also result in major Linguistic issue
8/8/2020 4:05 PM	Companies should stop 1) weaseling their way out of a proper wage and tossing everything to contractors (whom they know nothing about, and somehow pass tests with gootle translated crap) b) stop being racist. I've been refused jobs simply because I'm white, even though I'm fluent in Japanese. Emails suddenly stop when they find out, interviews are instantly hung up and such... Pathetic. And they're left with terrible translations that make little to no sense, and murdering of characters (most of them either sound absolutely dumb or egotistical)
8/8/2020 1:43 PM	Self-debugging on the dev/translation side before the start of LQA can go a long way to catching many of the issues described in this survey. If time and

	budget can be set aside for self-debug, then it will reduce the number of bug reports during QA and result in a smoother QA process where resources can be better spent solving more serious issues.
8/7/2020 11:20 AM	The main problem I have encountered is that European Spanish translators translate for Mexican Spanish; AND that there is no official dictionary for Mexican Spanish, so most translators need to follow European Spanish rules, but don’t really think about the end-user, so they’d rather use official accepted words by the Spain dictionary, rather than using a local word that is more user friendly.
8/6/2020 10:33 AM	Bugs are only opened for issues such as overlaps, cutoffs and visual issues with texts (basically, only when changing the text is not enough to fix the issue, or the text cannot be shortened). Inconsistencies, mistranslations, typos, ecc are usually handled just by the loc tester editing the text through the database.
8/5/2020 4:00 PM	One of the biggest issues in my opinion is that translators and QA testers are mostly on zero hour contracts and considered as disposable by the companies. They are underpaid and usualy move on to better paid jobs and are quickly replaced. This means that alot of translators/testers will move on mid project and new ones will be brought in. I worked for two years as a localisation tester and was on a zero hour contract the whole time. I eventually got tired of the job and am now looking elsewhere for work. The pay was also not sustainable.
8/5/2020 1:02 PM	The issues detected during loc testing could be treated better if none of the parts would be pushed into unrealistically tight deadlines: less the time and resources, worst the results. This regards everyone, though, from devs to loc testers and translators.
8/4/2020 3:55 PM	As a Localisation QA tester, I think one of the main hurdles to overcome is a lack of communication. I rarely have the opportunity to speak directly with the translators to ask questions about choices they've made and I know translators often don't get much communication from dev either. I think a lot of issues that arise in the localisation process could be avoided by improving communications between developers, translators and localisation testers.

APPENDIX 7 – EXAMPLE OF BOGUS ANSWER S3

VIDEO GAME DEVELOPMENT TOOLS

#413

COMPLETE

Collector:	Social Media Post 1 (Facebook Link)
Started:	Sunday, February 28, 2021 1:46:47 AM
Last Modified:	Sunday, February 28, 2021 1:50:44 AM
Time Spent:	00:03:56

Page 2: Personal Information

Q1

What is your age?

17 or less

Q2

What is your nationality?

Brazilian

Q3

Have you received any formal education or training?

PhD

Q4

How many years of professional experience in video game development do you have?

More than 20

Q5

Is video game development your main type of employment?

No,
If no please specify::
Pickup sticks for mommy and daddy

Q6

What is your type of employment as a video game developer?

Other (please specify):
i make game with sticks

Q7

How much of your activity is in video game development?

More than 75%,
Please comment:
i spend all day make game with sticks

Q8

When it comes to the localisation of the video game, what kind of model do you usually work with?

Post-gold (the game is already finished)

Page 3: Sim-ship

1 / 3

Q9 When does the localisation phase usually start?	Respondent skipped this question

Page 4: Professional information

Q10 What kind of platforms do you usually work with?	**Mobile games**
Q11 What OS do you use day to day?	Other (please specify): dirt with sticks
Q12 To facilitate cross compatibility within the devteam do you use any kind of system emulation?	Other (please specify): you can play on dirt, table, chair, bed. anywhere
Q13 What game development engine do you use? (mark several if necessary)	Other (please specify): dirt and sticks
Q14 What do you think of XLOC?	**I've heard of it but I don't like/want to use it,** Please specify why you use it or avoid using it:: My mommy said its scary
Q15 What's the attitude to testing within your current devteam?	**We aim for 100% code coverage**
Q16 What importance do you put on play-mode or screen comparison style testing?	**We cover 100% of gameplay**
Q17 What are your primary day to day programming languages for game development?	Other (please specify): gravity, dirt, sticks
Q18 How do you isolate gameplay text from code?	Data storage system (please specify): i put all sticks in stick pile and all dirt in dirt pile

Page 5: External files options

Q19

What kind of external files? (Please mark several if necessary)

Respondent skipped this question

Page 6: Game development process

Q20

What is the game development process that you usually use?

Other (please specify):

first i take all sticks and pile them on top of each others, then put dirt on all the sticks

Q21

Is there anything you would like to add?

Yes

Page 7

Q22

Please leave you comment here!

I love sticks and dirt

APPENDIX 8 – RÉSUMÉ EN LANGUE FRANÇAISE

L'avènement des ordinateurs, des jeux vidéo et l'apparition sur le marché de nouvelles technologies à des fins de traduction allèrent de pair. Le premier jeu vidéo, Spacewar! (1962), fut développé dans une version réduite des ordinateurs géants de l'époque. L'industrie du jeu vidéo commença à prendre son essor grâce aux bornes d'arcade et à une succession rapide de consoles de jeu dans les années 70, tandis que la taille des ordinateurs diminuait grâce à l'utilisation de microprocesseurs. Les années 80 virent l'arrivée des ordinateurs personnels, des logiciels de productivité et des jeux dans les lieux de travail et les foyers. Ces avancées furent rapidement exploitées par les entreprises qui découvrirent le potentiel des marchés internationaux, donnant ainsi naissance aux pratiques de localisation.

Le secteur a connu une croissance économique imparable qui a pratiquement triplé les revenus générés en l'espace de quatre décennies, atteignant 196,8 milliards[86] en 2022. La localisation est désormais essentielle au développement des jeux vidéo, amplifiée par leur popularité croissante et le besoin accru de professionnels. Actuellement, de nombreux développeurs prévoient la sortie simultanée de jeux dans plus de 10 langues différentes, une tâche remarquablement complexe en raison de la nature multimédia du produit, de la grande variété de types de texte présents dans les jeux et des rôles qu'ils jouent, de l'absence de linéarité du texte et de l'environnement visuel, de la présence de code mélangé au texte source, ainsi que de tous les éléments culturels impliqués.

À mesure que la technologie évolue et que de nouveaux programmes sont créés pour le développement, les tests et la localisation des jeux vidéo, il est crucial de rechercher les opportunités offertes par ces nouveaux outils afin d'identifier les fonctionnalités potentielles qui pourraient améliorer la productivité des traducteurs et la qualité des traductions, tout en résolvant l'un des principaux problèmes du domaine : le manque d'environnement visuel. Il est donc essentiel d'analyser les pratiques commerciales, les programmes et les processus actuels impliquant trois acteurs clés de la chaîne de localisation : les développeurs, les traducteurs et les testeurs linguistiques.

1. L'industrie du jeu vidéo

Le premier ordinateur binaire programmable, le Z1, fut créé par Konrad Zuse en Allemagne entre 1936 et 1938. Cependant, ce n'est qu'aux alentours de 1944 que deux machines, le Colossus et l'Electronic Numerical Integrator And Calculator (ENIAC), commencèrent à utiliser des tubes à vide (ou lampes) pour accélérer les calculs et réduire la consommation d'énergie. L'ENIAC était équipé de plus de 17 000 lampes et pesait près de 30 tonnes. Ainsi, des années 30 aux années 60, les ordinateurs se caractérisaient par leur taille imposante, et ce n'est qu'avec le développement des microprocesseurs qu'ils purent être réduits en taille, donnant ainsi naissance aux ordinateurs personnels. Toutefois, c'est pendant ces décennies des géants que l'histoire des jeux vidéo débute. Les deux premiers prototypes de jeux vidéo, Tennis for Two (1958) et Spacewar! (1962), virent le jour lors de ces premières phases de

[86] https ://newzoo.com/key-numbers

développement. Leurs successeurs, quant à eux, furent créés et commercialisés pour les bornes d'arcade, les consoles et certains ordinateurs domestiques à la fin des années 70.

Les années 80 furent une période extrêmement fructueuse pour les jeux d'arcade et se caractérisèrent par l'effondrement soudain de la société américaine Atari, ouvrant ainsi la voie à l'émergence de Nintendo et Sega, qui se lancèrent dans une concurrence féroce connue sous le nom de "guerre des plates-formes". Durant les années 90, de nouvelles avancées technologiques transformèrent le marché avec l'avènement de la 3D et de nouveaux genres de jeux (Chandler et Chandler, 2010). En termes de popularité, on assista à un déclin des jeux d'arcade, une croissance remarquable des jeux informatiques et la poursuite de la guerre des consoles avec l'introduction de la technologie CD-ROM.

Un des grands jalons du nouveau siècle fut le passage aux DVD, offrant une plus grande capacité de stockage, prenant en charge le format ASCII et offrant une meilleure qualité audio. Ces avancées permirent d'inclure davantage de fonctionnalités et la généralisation des cinématiques de meilleure qualité au fil des décennies. D'autres étapes importantes furent franchies au cours des premières années 2000, telles que la reconnaissance vocale et la technologie de capteur de mouvement (Mangiron Hevia et O'Hagan, 2013). Aujourd'hui, après l'avènement de la 3D, les nouveautés les plus intéressantes sont la réalité virtuelle (RV), la réalité augmentée (RA) et la réalité mixte (RM). En ce qui concerne les appareils, les jeux sur mobile ont pris le dessus sur le marché et génèrent presque autant de revenus que toutes les autres plateformes combinées[87].

1.2 Acteurs

Pour comprendre la complexité des processus impliqués dans la création d'un jeu vidéo, il est essentiel d'examiner les différents acteurs impliqués : les plateformes, les éditeurs, les développeurs et les testeurs. Le concept de plateforme ou fabricant fait référence au matériel utilisé pour jouer au jeu et aujourd'hui, il existe quatre plateformes principales en raison de la disparition des bornes d'arcade et de l'avènement de la réalité virtuelle : les ordinateurs, les consoles, les appareils mobiles et la réalité étendue. Selon Chandler (2020, p. 18) : "[c]haque type de plateforme présente des différences qui impactent la conception et la monétisation du jeu, comme les contrôleurs, les limitations techniques et la taille de l'écran". De plus, les systèmes d'exploitation de chaque plateforme ont leurs propres spécificités en raison de la diversité matérielle. Les développeurs doivent donc se conformer à ces normes matérielles, aux systèmes d'exploitation et aux environnements de développement logiciel, en suivant les directives de compatibilité et de conformité des détenteurs de plateformes (Laakso et Nyman, 2014). Il est également important de noter que certains fabricants, tels que Sony et Microsoft, exigent une proposition de concept de jeu même avant le début de la production. Ne pas respecter cette étape peut entraîner le rejet du jeu avant même sa soumission finale pour approbation et fabrication (Chandler et Chandler, 2010).

[87] https ://helplama.com/game-industry-usage-revenue-statistics/

“Des éditeurs tels qu'Activision Blizzard et Electronic Arts planifient leur calendrier de sorties plusieurs années à l'avance, à l'instar des studios de cinéma. Chaque année, ils prévoient un certain nombre de grosses sorties, ainsi que des sorties plus modestes tout au long de l'année” (Chandler, 2020, p. 15). Les grosses sorties sont généralement prises en charge par leurs équipes internes, tandis qu'ils financent également des projets plus petits de développeurs indépendants. Les éditeurs ont un large éventail de responsabilités, notamment le financement, la distribution, le marketing, le soutien à la production, la gestion des produits, les opérations en direct, la gestion de la communauté et le service client (Chandler, 2020, p. 35-38). Ils jouent un rôle crucial tout au long du processus de développement et peuvent proposer des modifications au jeu ou rejeter certaines fonctionnalités. Parfois, il peut y avoir deux producteurs impliqués, le producteur de l'éditeur (PE) et le producteur du développeur (PD), et il est important que leurs rôles respectifs soient clairement définis pour éviter toute confusion.

Les développeurs de jeux vidéo sont responsables de tous les aspects techniques de la création d'un jeu : l'art, le design, l'ingénierie, l'audio, l'expérience utilisateur et l'assurance qualité. Aujourd'hui, l'industrie du jeu vidéo connaît une grande popularité, ce qui se traduit par la présence de développeurs indépendants amateurs et polyvalents qui travaillent sur leurs projets pendant leur temps libre et collaborent parfois entre eux pour produire de petits jeux distribués gratuitement ou à un prix abordable sur des plateformes en ligne comme itch.io ou Steam. De plus en plus de développeurs indépendants travaillent en freelance, collaborant avec des studios qui externalisent certaines tâches ou engagent des talents pour des projets spécifiques. Par ailleurs, de petits studios indépendants ont une équipe de production interne et cherchent à présenter leur concept de jeu à un éditeur ou à financer leur projet par d'autres moyens. Enfin, les plateformes et les éditeurs possèdent aussi leurs propres équipes internes ou des studios affiliés pour exercer un contrôle total sur la production de leurs titres AAA.

Les testeurs s'assurent du bon fonctionnement et, bien que leur rôle soit souvent associé à la fin de la production, ils devraient également participer aux tests du prototype pour améliorer l'expérience utilisateur. Ils s’occupent notamment de :

- ***Bugs* de fonctionnalité :** Problèmes d'action-réaction non attendue : figement, blocage, fermeture inattendue du système, etc.
- ***Bugs* graphiques :** Images manquantes, affichage incorrect, etc.
- ***Bugs* d’audio :** Problèmes d'implémentation, de synchronisation ou de qualité.
- ***Bugs* de système :** Problèmes de police, mauvaise implémentation, etc.
- **Bugs linguistiques :** Fautes d’orthographe, erreurs de traduction, débordements de texte, textes tronqués et problèmes de terminologie, de grammaire, incohérences, d'instructions, de lisibilité, de sous-titres, d’audio et culturels (Muñoz Sánchez, 2017).
- **Tests de conformité :** Les testeurs doivent s'assurer que le jeu respecte toutes les exigences imposées par les plateformes.

1.3 Outils de développement

Une équipe de développement de jeux vidéo est composée de membres travaillant simultanément sur différents aspects du jeu sur plusieurs outils, créant les éléments qui seront intégrés plus tard. Ainsi, le pipeline d'assets devient essentiel. Il s'agit d'une série de processus automatisés permettant d'intégrer les éléments déjà finalisés dans le jeu, offrant aux développeurs la possibilité de les visualiser, de les tester et de les examiner pour itérer rapidement. Les différents outils peuvent être adaptés aux besoins du projet et du genre du jeu et il existe un grand-nombre des logiciels intermédiaires tels que des éditeurs de texte, des outils de dessin 2D, des logiciels de modélisation 3D, etc. Ces logiciels intermédiaires prêts à l'emploi sont souvent combinés avec des outils internes et le “moteur du jeu” pour profiter des avantages de chaque option. Le terme "moteur de jeu" est utilisé dans l'industrie pour désigner une multitude d'outils utilisés dans le développement de jeux vidéo et peut parfois prêter à confusion, car un "moteur consiste généralement en une suite d'outils et un composant d'exécution" (Gregory, 2018, p.38). L'idée est que, au lieu d'un seul logiciel, "un moteur de jeu est un système complexe de couches interconnectées liées au matériel et à d'autres logiciels" (Toftedah et Engström, 2019, p. 8). On peut trouver deux types principaux de moteurs de jeu : à usage général (comme Unity ou Unreal) et à usage spécifique (comme GameMaker ou Twine) (Toftedah et Engström, 2019).

Selon le genre de jeu, les dialogues jouent un rôle essentiel et peuvent être utilisés pour renforcer l'immersion des joueurs en simulant des réponses spontanées. Comme l'explique Domsch, "la différence structurelle majeure réside dans l'utilisation du langage par le système de jeu : soit comme contenu invariable pour remplir une structure de choix multiples, soit pour réellement traiter l'entrée langagière des joueurs" (Domsch, 2017, p. 258). Dans ce dernier cas, le système analyse et simplifie le texte afin de réduire le nombre de possibilités du point de vue du jeu et de simuler des options infinies. En revanche, la première option implique la création de branches de dialogue et leur mise en œuvre ultérieure dans le jeu. Cette arborescence de dialogues peut rapidement devenir complexe sans support visuel. C'est pourquoi de nombreux outils spécialisés dans le développement de jeux vidéo, ainsi que ceux utilisés pour créer des histoires interactives, proposent une représentation visuelle des options de branches sous forme d'outils de dialogue en arborescence.

1.4 Les phases et processus du développement de jeux

Les processus utilisés dans le développement de jeux, de manière similaire au développement de logiciels, peuvent être classés en : méthode en cascade ou prédictive, itérative ou agile, hybride et ad hoc (Politowski et al., 2016). Le marché s'est orienté vers des pratiques itératives par opposition aux méthodes traditionnelles en cascade. Alors que la méthode en cascade ou prédictive se caractérise par le développement d'un programme en suivant une séquence de phases, les pratiques agiles consistent à développer un jeu "en répétant des cycles courts pour fournir une fonctionnalité prête à être utilisée à chaque fois" (Politowski et al., 2016, p. 2). De plus, en fonction des pratiques de l'entreprise, le cycle complet de développement de jeux vidéo peut être divisé en un nombre différent de phases distinctes, allant de 3 étapes extrêmement basiques - préproduction, production et post-production (Aleem et al., 2016) - à 6 étapes qui considèrent la présentation du projet, l'initiation du projet ou les tests comme des étapes distinctes.

La phase de préproduction englobe plusieurs aspects, tels que la création du concept du jeu, la réalisation de prototypes, la documentation et, le cas échéant, la présentation du jeu et la constitution de l'équipe. L'équipe chargée de développer le concept initial travaillera progressivement sur les idées jusqu'à parvenir à la base essentielle du jeu, comprenant les objectifs, l'accroche pour attirer les joueurs, la boucle de jeu principale décrivant les actions possibles, le genre du jeu, les plateformes cibles, le public cible et le modèle économique (Chandler, 2020, p. 81-87). Un prototype sera ensuite créé, intégrant l'idée centrale et les mécanismes de jeu clés, afin d'évaluer la viabilité du projet. Ce prototype, également appelé "First Playable", comportera des éléments temporaires (graphiques et audio) pour illustrer le concept et reproduire l'apparence et les sensations du jeu, bien que ces éléments soient ultérieurement remplacés par des versions finales. De plus, la boucle de jeu essentielle devra être fonctionnelle pour permettre aux équipes de contrôle qualité de tester le prototype et d'apporter des commentaires utiles, qui seront pris en compte en collaboration avec les résultats de l'équipe d'expérience utilisateur.

La documentation (ou plan de jeu) comprend des éléments essentiels tels que le budget, le personnel et l'établissement d'un calendrier détaillé comprenant des estimations de temps, l'allocation des ressources et les dépendances. Les dépendances jouent un rôle crucial car, par exemple, les artistes doivent attendre l'approbation de la direction avant de commencer à créer le prototype. De même, avant que l'équipe de contrôle qualité ne puisse vérifier les textures, les objets doivent être créés et les textures appliquées (Chandler, 2020, p. 177). En outre, avant le début de la phase de production, deux autres aspects doivent être finalisés : la documentation et le pipeline de production. La documentation, qui varie d'une équipe de développement à l'autre, comprend différents types de documents selon les disciplines. Les artistes auront besoin d'un guide de style (polices, palettes de couleurs, etc.), d'une liste complète des éléments graphiques ou d'instructions sur les outils. Les concepteurs nécessiteront des "bibles" de personnages décrivant chaque personnage du jeu, des "bibles" d'histoire ou le script du jeu. Les ingénieurs auront besoin de documents techniques incluant les normes de codage et la conception technique du jeu. L'équipe de contrôle qualité utilisera la documentation créée par l'équipe de conception pour élaborer les plans de test, etc.

La phase de production consiste à créer les éléments graphiques, à mettre en œuvre le plan de jeu et à accomplir les tâches prévues. Ainsi, la première version développée pendant cette phase, appelée Alpha, constitue une amélioration du prototype). Environ la moitié des éléments graphiques et audio sont définitifs et ont remplacé les éléments temporaires initiaux. Le jeu a été adapté à la plateforme cible et toutes les fonctionnalités sont opérationnelles, bien qu'elles ne soient pas encore définitives. La moitié des tâches de l'équipe de conception sont terminées, l'équipe de contrôle qualité peut commencer les tests de jeu, et l'équipe d'expérience utilisateur vérifie les fonctionnalités afin de s'assurer qu'elles produisent l'effet souhaité sur le public cible. Avant de finaliser la compilation de la version Alpha et de passer à la version Beta, la phase de localisation doit être terminée et les chaînes localisées doivent être intégrées pour effectuer les tests linguistiques. En phase Beta, tous les éléments graphiques, de conception et audio doivent être définitifs et intégrés, car la version peut être soumise à la demande du fabricant. La phase Beta est principalement dédiée à la résolution des bugs identifiés par l'équipe de contrôle qualité et à d'éventuels ajustements mineurs pour tenir compte des commentaires de l'équipe d'expérience utilisateur (Chandler et Chandler, 2010, p. 107). De plus, "[u]n jeu en version bêta peut également être publié sous le titre d'Early Access. […] Le jeu devrait être en version Beta à environ 75 %-85 % du développement" (Chandler, 2020, p. 68).

La dernière étape du processus est la création de la version finale, la “Release Candidate” (RC) et comprend "le lancement du produit, la documentation du projet, le partage des connaissances, les bilans après-projet et la planification de la maintenance et de l'expansion du jeu" (Ramadan et Widyani, 2013, p. 99). Cependant, pour créer la version finale, l'équipe de développement doit initier le processus de diffusion du code, qui consiste en une vérification finale et plus approfondie pour "confirmer qu'il est prêt à être expédié au fabricant" (Chandler et Chandler, 2010, p. 107), ce qui comprend principalement les tests de conformité.

2. Localisation : définition, histoire, outils et types

La contribution de Holmes au domaine des études de traduction et la classification qu'il élabora pour définir la discipline posèrent des fondations solides qui persistent malgré "les multiples révisions, critiques et ajouts" (Jiménez-Crespo, 2013, p. 136). Dans son article (Holmes, 1988), il nomma et décrivit les études de traduction (Translation Studies ou TS) comme étant essentiellement empiriques et divisa le domaine en deux branches : les TS pures et les TS appliquées. Cependant, peu de temps après l'apparition des TS, la tendance mondiale à la numérisation et la force motrice de l'industrie de la localisation repoussèrent "les limites des compréhensions pré-numériques de ce que l'on entendait par traduction" (Jiménez-Crespo, 2019, p. 26), ouvrant ainsi le débat controversé sur la définition de la localisation. Cette question demeure ouverte à ce jour, car "les définitions de la localisation tendent à être liées au contexte, reflétant les perspectives de ceux qui les formulent" (Folaron, 2006, p. 197).

Lorsqu'on examine les définitions fournies par l'industrie, on constate que la localisation est principalement décrite comme un processus comprenant plusieurs étapes, telles que l'adaptation du produit, les tâches de gestion et les étapes techniques comme l'encodage des caractères, ce qui en fait un service réservé aux spécialistes. Selon Jiménez-Crespo (2013, p. 13), les professionnels travaillent avec des produits plutôt qu'avec des textes, ils distinguent les aspects linguistiques des aspects culturels, utilisent le terme "locale" plutôt que "langue" et évitent autant que possible le terme "traduction". Cependant, les chercheurs en études de traduction, comme Hartley (2009, p. 107), critiquent le fait que l'industrie considère l'inclusion d'une dimension culturelle comme une nouveauté, alors que cela fait partie de la définition couramment acceptée de la traduction au sein de leur communauté. Aujourd'hui, la définition controversée de la localisation et sa place dans les études de traduction gagnent en notoriété, car la frontière différenciatrice entre les deux devient de plus en plus mince avec l'extension de l'utilisation du terme "localisation" aux textes non numériques. Cette tendance à effacer les frontières, repousser les limites et converger vers des modèles et des notions est appelée à se poursuivre avec l'évolution de la technologie, effaçant les frontières encore davantage.

La naissance de l'industrie de la localisation fut rendue possible grâce à la démocratisation des ordinateurs personnels et à la commercialisation de logiciels à l'échelle internationale, permettant ainsi de cibler une clientèle non anglophone et de dépasser les frontières des pays anglophones. Cette transition vers un public composé de professionnels non informaticiens fut rapidement saisie par les entreprises américaines, qui réalisèrent qu'une nouvelle opportunité s'offrait à elles. Comme le souligne Jiménez-Crespo (2013, p. 8), "les premières cibles furent le Japon et les pays FIGS (France, Italie, Allemagne et Espagne)", et les fabricants comprirent rapidement qu'il était essentiel de "convertir le logiciel de manière à ce que les utilisateurs puissent voir un produit dans leur propre langue, profondément ancré dans leur propre culture" (Uren et al., 1993, p. x). Cela incluait des ajustements tels que les formats de date et d'heure, les jeux de caractères pour la représentation numérique des systèmes d'écriture, les calendriers et les séparateurs décimaux (Dunne, 2014, p. 148).

Initialement, les fabricants adoptèrent différentes stratégies pour la localisation. Certains développeurs de logiciels et de matériel créèrent des services spécialisés en interne pour gérer la traduction et soutenir leurs efforts internationaux, tandis que d'autres entreprises préféraient confier la localisation à leurs bureaux locaux dans les pays concernés. Cependant, la pratique consistant à finaliser d'abord la version anglaise du programme avant de le localiser dans différentes langues se révéla problématique. Esselink (2003, p. 4) explique que cette approche était "extrêmement problématique". Selon Dunne (2014, p. 149), la version anglaise présentait des lacunes majeures, notamment l'incapacité d'afficher les scripts et systèmes d'écriture nécessaires dans la langue cible. Par conséquent, les équipes de localisation devaient identifier les parties du code pouvant être traduites, une tâche complexe, et demander aux développeurs d'effectuer des ajustements pour afficher correctement le texte traduit. Cette approche engendrait des coûts supplémentaires et des retards dans le processus de localisation.

La complexité croissante des ordinateurs et des logiciels, ainsi que l'augmentation du volume de travail, ont donné lieu à une industrie indépendante dédiée exclusivement à la localisation. Les entreprises se sont retrouvées confrontées à la nécessité de gérer plusieurs versions du programme en raison des modifications spécifiques du code source requises pour chaque langue. Après le processus de localisation, chaque version était compilée et testée individuellement, nécessitant des corrections et des mises à jour en cas de découverte de *bugs*. Cette approche s'est avérée à la fois coûteuse et chronophage, étant donné la complexité en termes d'ingénierie et la diversité des textes et des formats à gérer. Cependant, les entreprises ont rapidement compris qu'il était possible de faciliter le processus en prenant des mesures en amont, ce qui a conduit à la création de l'internationalisation.

Cette approche a éliminé la nécessité de maintenir un ensemble distinct de code source pour chaque localisation et a permis aux entreprises de remplacer simplement les ressources contenant la langue source par celles contenant la version localisée à l'aide d'éditeurs spécialisés. Parallèlement, le concept de GILT, qui regroupe les processus de mondialisation, d'internationalisation, de localisation et de traduction, est apparu pour décrire cette convergence. L'industrie de la localisation a également adopté le modèle d'expédition simultanée ou "sim-ship", où la localisation se déroule pendant la phase de développement du produit, permettant une commercialisation plus rapide et réduisant les coûts potentiels de production et de marketing. En parallèle, l'externalisation des processus de traduction multilingues a donné naissance à des prestataires multilingues et à des départements spécialisés au sein des entreprises existantes. Ces développements ont entraîné une professionnalisation de l'industrie, avec la création d'organisations professionnelles, de conférences, de publications et une visibilité accrue au cours des années 90. Cette croissance a également été accompagnée du développement d'outils de traduction et de localisation visant à réduire les coûts et à faire face à la concurrence intense dans le secteur (Esselink, 2000, 2003).

2.1 Outils de traduction et environnements de traduction intégrés

L'apparition des ordinateurs a révolutionné les outils utilisés par les traducteurs, remplaçant les machines à écrire et les dictionnaires imprimés par des mémoires de traduction, des systèmes de traduction automatique, des outils de gestion terminologique, des outils de localisation, et bien d'autres encore. Malgré que l'idée de la traduction automatique remonte au XVIIe siècle (Hutchins, 2005), elle n'a réellement pris forme qu'à partir des années 1950. Toutefois, l'enthousiasme et l'intérêt des institutions ont connu des fluctuations importantes en raison du rapport publié en 1966 par le Comité consultatif sur le traitement automatique des langues (ALPAC), qui soulignait les coûts élevés, la lenteur du processus et la moindre précision de la traduction automatique par rapport à la traduction humaine.

Face à ces constats, d'autres chercheurs ont orienté leurs travaux dans une autre direction. En 1978, Arthern a proposé la création d'un système capable de traiter les textes avec une mémoire suffisamment grande pour stocker à la fois les textes sources et les textes cibles dans différentes langues (Arthern, 1978, p. 94-95). Deux ans plus tard, Martin Kay (1980) a suggéré un système comprenant un "traitement de texte multilingue, un dictionnaire et la possibilité de consulter des traductions antérieures. Il inclurait également un composant de traduction automatique, fonctionnant sous le contrôle du traducteur" (Zaretskaya, 2017, p. 17). Ces avancées ont ouvert de nouvelles perspectives pour l'industrie de la traduction, contribuant ainsi à l'évolution des outils et des méthodes utilisés par les traducteurs.

"INK, une société spécialisée dans les technologies de traduction, fut pionnière dans le développement commercial d'outils de support de traduction pour les ordinateurs de bureau, tels que les INK TextTools" (Esselink, 2003, p. 5). Ces outils permettaient la gestion terminologique, la création de dictionnaires bilingues en analysant le texte source et des dictionnaires (Lewis, 1991, p. 35). Peu après, les outils de traduction assistée par ordinateur (TAO) firent leur apparition sur le marché (Esselink, 2003, p. 5)

> En 1987, la société allemande de traduction TRADOS revendait les INK TextTools et, un an plus tard, lançait TED, le plug-in Translation Editor pour TextTools. Peu de temps après, TRADOS sortit la première version de son produit Translator's Workbench, une mémoire de traduction (TM).

Les premiers programmes TAO étaient limités aux fichiers texte, et la plupart des fabricants de logiciels développaient leurs propres outils de localisation pour gérer les interfaces utilisateur (Esselink, 2003, p. 5). Par la suite, des outils de localisation commerciaux tels que SDL Passolo ou Alchemy Catalyst sont apparus, offrant aux traducteurs un environnement WYSIWYG pour voir les modifications en temps réel. Cependant, les jeux vidéo diffèrent des logiciels car ils sont basés sur des normes de codage et de formatage de fichiers différentes (Bernal-Merino, 2013, p. 119).

Les outils de TM ont progressivement gagné en popularité parmi les traducteurs, devenant l'outil de traduction le plus largement utilisé de nos jours. Des études antérieures sur l'adoption de ces systèmes ont révélé que seuls 28 % des 591 participants utilisaient des outils de TAO au Royaume-Uni (Fulford et Granell-Zafra, 2005, p. 10). Un an plus tard, Lagoudaki a constaté que parmi les 699 participants à son enquête, 82,5 % utilisaient des systèmes de TM (Lagoudaki, 2006, p. 11). En revanche, les résultats publiés par Zaretskaya et al. en 2018 ont révélé que 76 % des 736 participants utilisaient cette technologie (Zaretskaya et al., 2018, p. 46). Ces études ont également mis en évidence une préférence pour des solutions globales, Fulford et Granell-Zafra ont constaté que seulement 24 % des répondants utilisaient des systèmes de gestion terminologique (Fulford et Granell-Zafra, 2005, p. 9), tandis que Zaretskaya et al. ont rapporté des taux de 58 % pour les outils de gestion terminologique et de 25 % pour les outils d'extraction terminologique (Zaretskaya et al., 2017, p. 46).

Ainsi, la plupart des systèmes ont constamment enrichi leur offre en intégrant des outils d'extraction terminologique, de gestion terminologique, voire de compilation de corpus, afin de devenir des systèmes complets répondant au mieux aux besoins des traducteurs. Par ailleurs, la traduction automatique neuronale a ravivé l'intérêt pour ce domaine et a été progressivement intégrée dans ce que l'on pourrait désormais appeler des environnements de traduction intégrés. De nos jours, il existe de nombreux programmes disponibles sur le marché, à télécharger ou basés sur le web, allant des "systèmes de gestion de la traduction (TMS)" initiaux à la popularité croissante des "systèmes de gestion de contenu (CMS)" dans le domaine de la localisation, qui intègrent également les fonctionnalités mentionnées précédemment.

2.2 Types de localisation

De nouveaux types de localisation sont également apparus en conséquence directe des avancées technologiques telles que la démocratisation des ordinateurs personnels, l'apparition des jeux vidéo numériques, l'invention du World Wide Web et la création des smartphones.

Localisation de logiciels : englobe plus que le simple "code de programmation assemblé dans un fichier exécutable" (Sandrini, 2008, p. 169). Elle concerne les tutoriels, les supports marketing, les fichiers d'exemple, la documentation utilisateur imprimée et en ligne, ainsi que l'aide en ligne (Dunne, 2014, p. 150). Les ressources à localiser incluent les accélérateurs, les boîtes de dialogue, les icônes, les menus, et les chaînes de caractères (*string tables*), "qui regroupent les éléments de menu, les libellés des boutons de commande, les titres des boîtes de dialogue, les infobulles, les messages d'erreur et les messages d'état" (Dunne, 2014, p. 151). Certains outils de localisation permettent une représentation visuelle en temps réel des de menus et de boîtes de dialogue, mais les *string tables* sont généralement exclus de cette fonctionnalité (Dunne, 2014, p. 151). De plus, l'aide en ligne comporte des documents hypertexte compilés dans différents formats, tandis que les contenus basés sur le web adoptent les caractéristiques d'un site web. Les outils de localisation WYSIWYG permettent aux traducteurs de visualiser les changements "en temps réel" pour gérer les contraintes d'espace, notamment dans les menus et les interfaces utilisateur. Les kits de localisation fournissent les fichiers source, les directives, les glossaires et les notes de traduction nécessaires (Sandrini, 2008, p. 170). Il est également essentiel de respecter les exigences spécifiques des éditeurs ou des systèmes d'exploitation en matière de glossaires pour assurer la commercialisation du produit.

La localisation de sites web : a connu un essor considérable avec l'avènement du World Wide Web au début des années 90, stimulant l'essor du commerce électronique. Sandrini (2008, p. 179) souligne trois différences clés entre la localisation de logiciels et de sites web : la fréquence des mises à jour de contenu, la relation continue entre les traducteurs et les clients en raison de la nature dynamique des sites web, et la nécessité d'une expertise accrue dans la gestion de divers types de texte et de stratégies. Parmi les éléments à localiser, on trouve plusieurs formats de fichiers tels que HTM/HTML, XML, XSL, JS, ASP, PHP, JSP ou XLIFF (créé spécifiquement pour les outils de localisation basés sur XML) qui peuvent être directement traités par les outils de traduction assistée par ordinateur afin de protéger les balises et d'éviter les modifications accidentelles. Le niveau de localisation dépend du retour sur investissement souhaité par l'entreprise, allant de "sites web standardisés, semi-localisés, localisés, largement localisés et adaptés culturellement" (Singh et Pereira, 2005). De plus, lors de la phase d'internationalisation, les développeurs doivent prendre en compte l'impact des petits appareils et des tailles d'écran limitées, en intégrant un design adaptatif pour atténuer les contraintes d'espace et garantir la qualité du texte localisé.

Localisation d'applications : pour des téléphones mobiles et des appareils tactiles tels que les tablettes, il convient de distinguer le logiciel intégré de l'appareil et les applications téléchargeables, qui présentent des similitudes dans les procédures de localisation. Alors que le logiciel intégré exige le respect strict des règles terminologiques, la majeure partie du travail se concentre sur les applications. Ce type de localisation présente des caractéristiques communes à la localisation de logiciels, de jeux et de sites web, nécessitant une internationalisation pour créer un environnement propice à la localisation, prendre en compte les éléments culturellement spécifiques et mettre en œuvre un design adaptatif. Les traducteurs doivent tenir compte de l'interface utilisateur graphique (GUI), veiller à ce que les icônes soient internationalement reconnaissables, traiter les problèmes liés aux formats de date et d'heure, et s'adapter à la préférence locale pour les icônes. Le format de fichier le plus courant est XML, qui peut être facilement localisé à l'aide d'outils TAO, tandis que l'accès à l'application elle-même est souvent limité pour les traducteurs. Une autre pratique courante consiste à envoyer le texte dans des fichiers Excel ou Google Spreadsheet multilingues.

3. La localisation des jeux vidéo : recherche, histoire, caractéristiques et processus

Bien que la localisation de jeux vidéo ait largement contribué au succès de l'industrie du jeu, ce domaine a été largement ignoré dans les études de traduction jusqu'à récemment. Outre les publications académiques évaluées par des pairs, le nombre croissant d'articles, de blogs et de témoignages personnels publiés sur des sites web ou des médias sociaux liés à la localisation a plus que doublé pendant la crise sanitaire, soutenu par un nombre croissant de webinaires et de différentes types de conférences en ligne. Au cours des dernières années, le nombre d'événements organisés par l'industrie a également connu une croissance exponentielle, sensibilisant les développeurs à l'importance de la localisation des jeux vidéo et mettant la profession en lumière.

L'article pionnier est "Beyond PacMan : Translating for the Computer Game Industry", publié en 1999 par Frank Dietz. Tout comme les articles sur la localisation de jeux vidéo qui suivront, il a été écrit principalement par un professionnel décrivant son expérience personnelle et la manière dont l'industrie fonctionnait, en adoptant une approche descriptive qui couvrait les processus, les fonctions et le produit final. Les années 2004 et 2005 marquent une étape importante dans la recherche sur la localisation de jeux vidéo, avec la publication régulière de trois figures principales - Minako O'Hagan, Carme Mangiron Hevia et Heather Maxwell Chandler - qui ont contribué à établir les bases de la discipline. De plus, en 2005, Heather Maxwell Chandler a écrit *The Game Localization Handbook*, qui est devenu la première monographie sur la localisation de jeux vidéo et a contribué à la discipline en proposant des lignes directrices de localisation et un grand nombre d'entretiens avec des développeurs et des éditeurs.

L'année 2006 est considérée comme "Year one" (Mangiron, 2017, p. 77) pour la recherche sur la localisation de jeux vidéo dans le domaine des études de traduction en raison du nombre croissant d'articles académiques publiés sur le sujet. Pendant cette période “les méthodes qualitatives sont prédominantes et la plupart des recherches sont basées sur l'expérience de première main et l'autoréflexion des auteurs, ainsi que sur l'analyse textuelle et des études de cas se focalisant sur des aspects particuliers” (Mangiron Hevia, 2017, p. 84). En 2011, on observe une grande productivité avec plus de trente articles, conférences et chapitres consacrés à la localisation de jeux vidéo. À partir de 2012, le nombre d'articles publiés dans des revues académiques, des sites web spécialisés et d'autres publications a triplé. Une étape importante pour l'industrie de la localisation de jeux vidéo a été marquée en 2012 par la publication du livre *Best Practices for Game Localization* par Richard Honeywood et Jon Fung, dans le cadre de l'IGDA Localization SIG.

En 2013, Minako O'Hagan et Carme Mangiron Hevia ont publié le deuxième ouvrage monographique, *Introduction to Video Game Localization : Translating for the Global Digital Entertainment Industry*. Leur travail est devenu l'un des livres les plus influents dans le domaine et fournit des informations détaillées sur la localisation de jeux à partir d'une perspective académique, professionnelle et prescriptive. De plus, la thèse de doctorat de Bernal-Merino, *The Localisation of Video Games*, présentée à l'Imperial College London en 2013, a été révisée et améliorée, devenant ainsi le troisième ouvrage monographique sur la localisation de jeux sous le titre *Translation and Localisation in Video Games : Making Entertainment Software Global* (2015). En ce qui concerne les tendances liées aux sujets de recherche, l'accessibilité gagne en popularité ces dernières années dans l'industrie, de même que l'utilisation d'un langage inclusif sur le plan du genre. Néanmoins, la recherche en localisation de jeux vidéo a couvert de nombreux aspects du domaine d'un point de vue principalement descriptif, et l'utilisation de questionnaires comme méthode de collecte d'informations sur les pratiques de localisation de jeux vidéo reste limitée à moins d'une poignée d'articles.

3.1 L'évolution de la localisation des jeux vidéo au fil du temps

La localisation des jeux vidéo remonte à 1980 avec la sortie de Pac-Man aux États-Unis quand le nom du jeu et les noms des fantômes ont été adaptés pour attirer les joueurs américains, posant ainsi les bases des pratiques de localisation futures. Ces choix ont souligné l'importance de créer des adaptations accrocheuses et ont donné aux traducteurs plus de liberté pour améliorer l'attrait du produit (Mangiron Hevia et O'Hagan, 2013). Les années 1980 ont marqué les débuts de la localisation de jeux vidéo, initiée par des entreprises japonaises qui reconnaissaient le potentiel des marchés étrangers. Tout comme la localisation de logiciels, cela impliquait un processus d'essais et d'erreurs, souvent réalisé en interne après la sortie de la version originale. La localisation consistait à identifier et à adapter les chaînes de texte dans le code source, ce qui entraînait de nombreuses erreurs de traduction et de grammaire. De plus, l'absence d'outils de traduction spécialisés et d'assurance qualité contribuait aux problèmes de cohérence et de qualité globale. Pendant les années 1970 et 1980, en dehors de laisser le jeu dans sa langue d'origine, une pratique courante pour localiser les jeux dans d'autres langues était l'approche "box and docs" (Chandler, 2005, p. 13) visant les FIGS (français, italien, allemand, espagnol), ou E-FIGS pour les jeux développés à l'origine en japonais. Cela impliquait de traduire les documents imprimés accompagnant le jeu, tels que l'emballage et les documents à l'intérieur de la boîte, tout en laissant l'interface utilisateur (IU), les dialogues et les éléments du jeu en anglais. Ce niveau de localisation est encore utilisé aujourd'hui. Avec le passage aux plateformes numériques, la pratique "box and docs" a évolué pour inclure la traduction des descriptions de jeu en ligne, les instructions et une partie du matériel marketing.

Pendant les années 1990, les contraintes d'espace constituaient un problème majeur en raison des limitations technologiques, bien que l'industrie ait trouvé un certain soulagement avec la transition vers les CD-ROM et plus tard les DVD au début des années 2000. Cependant, des limitations d'espace persistent en raison des restrictions de caractères dans les interfaces utilisateur et les menus. Dans les années 1990, on a observé une tendance vers la "localisation partielle", qui implique la traduction des emballages, des documents, des manuels, des éléments du jeu, ainsi que la fourniture de versions sous-titrées des dialogues et des supports en ligne (Chandler, 2005). Ce niveau de localisation vise à générer des ventes supplémentaires sur des marchés secondaires plus petits avec un retour sur investissement raisonnable. Cette approche économiquement efficace permet de réduire les coûts liés aux acteurs de doublage, aux efforts de synchronisation labiale et à l'intégration des éléments du jeu. Dans les années 2000, la "localisation complète" est devenue la norme pour les jeux AAA, impliquant la localisation de tous les éléments et matériaux disponibles pour le joueur, y compris le texte du jeu, les fichiers audio, les manuels et les sites web. Cependant, certains jeux avec des budgets plus restreints combinent la localisation complète avec une localisation partielle pour les principaux marchés secondaires. Les avancées technologiques, telles que l'amélioration des capacités de stockage et audio ainsi que l'utilisation de graphismes 3D, exigent une attention accrue aux détails, notamment dans des domaines tels que la synchronisation labiale. Ces avancées ont également donné lieu à l'émergence de petites sociétés de localisation, augmentant la concurrence et réduisant les prix de la localisation. De manière similaire à la localisation de logiciels, les sorties simultanées sont devenues plus courantes, avec l'anglais souvent utilisé comme langue pivot, où le texte du jeu est traduit dans la langue cible tandis que l'audio reste en anglais avec des sous-titres proposés dans les FIGS (Mangiron, 2021).

De nos jours, bien que les sorties simultanées soient la norme, Chandler souligne que "les langues asiatiques et moyen-orientales sont également régulièrement publiées mais prennent parfois plus de temps en raison de contraintes techniques et d'autres fonctionnalités pour se conformer aux exigences gouvernementales" (2020, p. 231), ce qui entraîne une combinaison *sim-ship* et *post-gold*. Dans le futur, l'utilisation de la réalité augmentée et de la réalité virtuelle pourrait avoir un impact considérable sur le processus de localisation, Hughes et al. soulignent que pour les sous-titres, des considérations spéciales doivent être prises en compte en ce qui concerne la position, la taille et les caractéristiques d'affichage, car "la présentation des contenus n'est plus uniquement basée sur le temps, mais implique une dimension spatiale, déterminée à la fois par l'exploration libre de l'utilisateur et les positions dynamiques" (2019, p. 221).

En termes de niveaux de localisation, Bernal-Merino introduit un quatrième niveau appelé localisation approfondie ou améliorée, qui est similaire aux "sites web culturellement adaptés" (Jiménez-Crespo, 2013, p. 35), mais va au-delà des localisations traditionnelles en raison de la nature interactive, immersive et multimodale des jeux modernes. Cette évolution est étroitement liée à la normalisation de l'utilisation de la transcréation dans la localisation des jeux vidéo, un concept qui s'est étendu à d'autres domaines tels que le marketing.

Bernal-Merino le décrit comme "une traduction qui penche entièrement en faveur du public cible, tout en prétendant être le même produit malgré ces différences" (2006, p. 34). Di Giovanni (2008, p. 33) ajoute que le texte transcréé doit être "entièrement fluide et, surtout, pleinement compréhensible pour son public cible". La pratique de la transcréation découle du fait qu'une des priorités de la localisation des jeux vidéo est d'atteindre l'immersion. Comme l'indiquent Christou, McKearney et Warden (2011, p. 40), "la marque d'une localisation parfaite serait que le joueur considère que le jeu vidéo a été créé dans sa propre culture, pour sa propre culture. En d'autres termes, la localisation idéale susciterait une suspension totale de l'incrédulité". Comparé à d'autres types de localisation, les jeux vidéo poussent encore plus loin le concept de traduction fonctionnelle présent dans la théorie du Skopos. Ainsi, les traducteurs de jeux sont autorisés à prendre plus de libertés, comme le soulignent Mangiron Hevia et O'Hagan (2006, p. 6), qui expliquent que les traducteurs bénéficient d'une "liberté quasi absolue pour modifier, omettre, voire ajouter tous les éléments qu'ils jugent nécessaires afin de rapprocher le jeu des joueurs et de transmettre l'essence même du gameplay original". Bernal-Merino (2006, p. 34) ajoute que "d'un point de vue traductionnel, il s'agit du seul produit où le transfert linguistique fait partie intégrante du processus de développement et peut donc influencer la création réelle du jeu vidéo". En d'autres termes, la transcréation est au cœur de la localisation des jeux et est encouragée par les entreprises afin d'immerger davantage les joueurs et d'améliorer leur expérience de jeu.

3.2 Localisation des jeux vidéo : caractéristiques et types de texte

Bien que la localisation des jeux vidéo puisse être classée comme une sous-catégorie de la localisation de logiciels et qu'elle entretienne des liens étroits avec la traduction audiovisuelle, l'une des caractéristiques les plus distinctives de ce domaine est le fait que les jeux peuvent être considérés comme des "textes multimédias interactifs" (Bernal-Merino, 2013, p. 143). En général, la plupart des contraintes techniques présentes dans la localisation de jeux vidéo ressemblent à celles que l'on retrouve dans d'autres types de localisation : on recommande utiliser d'un design adaptatif, séparer les chaînes traduisibles du code, activer les sous-titres, éviter d'incorporer du texte dans les éléments graphiques, garantir la prise en charge des claviers internationaux et un kit de localisation complet est indispensable. Une fois la phase de localisation terminée, le produit final fait l'objet d'un processus spécifique d'assurance qualité linguistique (LQA) pour identifier les erreurs de mise en œuvre et les anomalies linguistiques. Les traducteurs de jeux vidéo reçoivent généralement des fichiers Excel contenant des identifiants de chaque *string*, le texte source et des commentaires des développeurs.

Cependant, les jeux vidéo peuvent contenir un grand volume de mots, dépassant souvent plusieurs centaines de milliers. De plus, la localisation de divers éléments au sein des jeux vidéo nécessite différentes stratégies en fonction de leur rôle dans le récit et leur fonctionnalité.

Mangiron Hevia et O'Hagan proposent une taxonomie complète des différents types de textes et distinguent cinq types d'éléments : les textes intégrés au jeu, les éléments graphiques, les éléments audio et cinématiques, les supports imprimés et les éléments en ligne (Mangiron Hevia et O'Hagan, 2013, p. 155-158). En ce qui concerne les éléments intégrés au jeu, "la macrostructure du texte du jeu est devenue plus complexe et non linéaire" (O'Hagan et Mangiron, 2013, p. 150) avec le temps. De plus, l'essor des jeux mobiles et le passage à un modèle économique de Jeu vidéo en tant que service (Games-as-a-Service) implique des mises à jour fréquentes des jeux - une prérogative de la localisation d'applications. Néanmoins, une caractéristique différenciatrice des jeux vidéo est la combinaison des messages système et des interfaces utilisateur avec une dimension ludique et de la narration. Bissell (2010, p. 37) explique que les jeux vidéo reposant sur une structure narrative utilisent deux types différents de narration : la narration proprement dite (représentée par des cinématiques où le joueur n'a aucun contrôle sur le déroulement de l'histoire) et "la 'ludonarration', […] non scénarisée et déterminée par le joueur - les parties 'amusantes' du jeu 'joué'". De plus, les éléments non diégétiques jouent également un rôle majeur en raison des exigences terminologiques des fabricants. En effet, si leurs directives ne sont pas suivies, le jeu pourrait subir des retards importants s'il échoue au processus de soumission car "utiliser un 'joystick analogique' dans un jeu destiné à être publié sur Microsoft Xbox signifierait que le jeu serait rejeté et devrait être renvoyé au développeur, qui devrait effectuer les modifications nécessaires et le soumettre à nouveau" (Mangiron Hevia et O'Hagan, 2013, p. 123).

Bernal-Merino (2013, p. 192) identifie trois facteurs principaux qui différencient les jeux vidéo des autres types de localisation : "l'interactivité de la construction de l'histoire, la fragmentation du texte source et la traduction des variables" (ibid). Afin de recréer l'interactivité sans rompre l'immersion du joueur, les développeurs fournissent aux PNJ plusieurs lignes et options suivant un format de dialogue arborescent qui leur permet de réagir aux actions ou aux choix du joueur pour simuler la spontanéité. Cette approche pose d'importants défis pour le traducteur, notamment en raison de la complexité de la structure. La fragmentation du texte source est une caractéristique intrinsèque des pratiques de localisation en raison des pratiques de sortie simultanée, de la nature dynamique de certains produits et du processus de développement lui-même, ce qui peut engendrer l'incapacité de discerner le genre de l'interlocuteur sans contexte ou accès à l'environnement visuel. Enfin, le troisième facteur est la présence de variables et, bien qu'elles ne soient pas exclusives aux jeux vidéo, les exigences des jeux vidéo en termes de qualité linguistique soulignent l'importance de trouver des solutions créatives et fluides. Même si la solution la plus élégante consisterait à créer plusieurs chaînes de caractères pendant la phase de développement du jeu, il n'est pas rare que les traducteurs soient contraints d'utiliser un langage contrôlé. Enfin, certains jeux utilisent des chaînes de caractères concaténées (plusieurs variables dans une seule phrase), ce qui peut représenter un défi majeur pour de nombreuses langues.

3.3 Le processus de localisation

Dans l'ensemble, les processus de localisation peuvent être divisés en quatre phases distinctes : la pré-localisation, la localisation, la post-localisation et la production et distribution (Mangiron Hevia et O'Hagan, 2013, p. 129). La phase de pré-localisation comprend la création du kit de localisation, la nomination du coordinateur et la sélection des traducteurs, ainsi que tout travail préparatoire (Mangiron Hevia et O'Hagan, 2013, p. 130). Dans le cas de la deuxième phase, la traduction propre du jeu, nous pouvons également observer que, de manière similaire à la traduction en général, il existe deux modèles : l'externalisation et l'interne. La première différence entre le modèle d'externalisation et le modèle interne réside dans la réduction des coûts, car maintenir une équipe de traducteurs tend à être plus cher que travailler avec un prestataire qui engagera ensuite des traducteurs indépendants pour chaque paire de langues. Dans ce modèle d'externalisation, les traducteurs indépendants n'ont généralement aucun contact avec les autres membres de l'équipe de localisation et doivent passer par un chef de projet désigné par le prestataire. De plus, souvent, le seul lien entre le chef de projet du prestataire et l'équipe de développement est un coordinateur de localisation sélectionné par l'entreprise de développement, qui fait le lien entre l'équipe de développement et le chef de projet du prestataire.

Une autre particularité du modèle d'externalisation est l'absence d'accès aux premières versions du jeu, ce qui oblige les traducteurs à effectuer un travail préparatoire en recherchant le genre du jeu, en jouant à des jeux similaires, en regardant des bandes-annonces, etc. Le manque de linéarité précédemment mentionné est aggravé par des délais serrés, un nombre élevé de mots et la répartition des documents source entre différents traducteurs indépendants qui pourraient ne pas pouvoir communiquer entre eux et n'avoir accès qu'à leur propre partie. Enfin, selon les services proposés par le prestataire de services linguistiques, le contrat peut également inclure l'enregistrement des voix et l'assurance qualité linguistique (services de tests linguistiques). En revanche, les modèles internes disposent généralement d'un petit service de localisation avec divers traducteurs qui font appel à des traducteurs indépendants si nécessaire.

La phase de localisation comprend également des tâches telles que l'édition et l'enregistrement vocal. Le doublage nécessite de donner la priorité à certains segments et de réaliser des adaptations plus approfondies pour la synchronisation labiale. Que le processus soit réalisé en interne ou externalisé à un prestataire ou à un studio d'enregistrement, les *strings* devront être localisés à l'avance puis éditées pour permettre suffisamment de temps pour les enregistrements et l'intégration des fichiers audio dans les versions localisées. Les exigences varient en fonction du type d'enregistrements, certains n'étant pas soumis à des contraintes de synchronisation, d'autres ayant des limitations de temps basées sur la durée de l'audio original, et d'autres devant être adaptés pour correspondre aux pauses sans une attention particulière à la synchronisation labiale (Mangiron Hevia et O'Hagan, 2013, p. 135). La troisième phase du processus de localisation commence par l'intégration des éléments localisés dans le jeu et à la création de la première version jouable. Cette version sera ensuite soumise à des tests à la recherche de *bugs* fonctionnels et linguistiques. Cette étape supplémentaire incluse dans le processus constitue une autre différence par rapport à d'autres domaines de traduction qui se contentent généralement d'une relecture effectuée soit par le traducteur lui-même, soit par une personne externe.

4. Résultats de l'enquête pour les traducteurs

La version définitive de la première enquête a été achevée au cours de la première semaine d'avril 2020, et le lien a été officiellement publié le 10 avril de la même année. Le lien utilisé comme collecteur a été officiellement fermé le 3 août, et le questionnaire a reçu un total de 620 réponses complètes sur 1000 en moins de 4 mois. La méthode la plus efficace pour distribuer l'enquête, après plusieurs tentatives, a été de cibler individuellement les professionnels via LinkedIn en utilisant directement les paramètres de recherche qu'il offrait via un message privé contenant le lien. L'enquête a été créée sur SurveyMonkey, car il offre une solution facile et complète pour créer, distribuer et gérer des questionnaires, avec la possibilité de sauter certaines questions si nécessaire.

4.1 Informations personnelles

Aucun des 620 participants n'a choisi les options "17 ans ou moins" et "65 ans ou plus", et moins de 1 % avait plus de 55 ans. Seulement 28 participants étaient dans la catégorie "45 à 54 ans", et 61 appartenaient au deuxième groupe d'âge. Par conséquent, le groupe le plus représenté est celui des "25 à 34 ans" avec 346 réponses (soit 55,81 %), suivi des participants âgés de 35 à 44 ans, représentant 29,03 % ou 180 traducteurs. En ce qui concerne les nationalités, seuls 56 pays avaient au moins un représentant, dont 13 ont reçu 10 réponses ou plus (Tableau 1).

NATIONALITÉS	RÉPONSES
Espagnole	113
Française	107
Italienne	69
Brésilienne	46
Allemande	43
Turque	23
Russe	22
Argentine	20
Américaine	16
Polonaise	13
Britannique	10
Égyptienne	10
Ukrainienne	10

Tableau 1. Nationalités avec 10 répondants ou plus (traducteurs)

4.2 Langues

Au total, si l'on prend en compte ceux qui ont choisi l'option "autre", il y avait 45 langues maternelles qui avaient au moins un locuteur. La langue maternelle la plus représentée était l'espagnol avec 158 réponses, suivie du français avec 121, de l'italien avec 72, de l'anglais avec 63 et de l'allemand avec 49. Il y avait également un nombre important de locuteurs natifs du portugais (47), du russe (36), du turc (25), du catalan (24) et du mandarin (22). En ce qui concerne la langue source principale, les résultats montrent que l'anglais, avec 543 réponses, est nettement la plus représentée, suivie du japonais et du mandarin qui n'ont reçu que 28 et 14 réponses respectivement. Enfin, 356 des participants ne travaillaient pas avec d'autres langues, bien que le pourcentage de ceux qui le font, soit de 42,58 %, reste assez élevé et parmi ces 264 participants, 43 langues ont reçu au moins une réponse.

4.3 Informations professionnelles

Plus de la moitié des répondants (62,57 %) avaient obtenu un diplôme universitaire en traduction. Néanmoins, si l'on examine de plus près ceux qui n'ont pas suivi de formation académique spécifique en traduction et que l'on combine les résultats de ces deux options, le pourcentage est assez élevé et remarquablement proche de ceux qui détiennent une maîtrise : 37,42 % ou 232 participants. De plus, la différence de résultats entre ceux qui ont obtenu une licence et ceux qui ont sélectionné l'option "aucun" est négligeable et ne représente que deux réponses.

Les participants ont également été interrogés sur leur nombre d'années d'expérience en tant que traducteur. Le pourcentage de participants ayant "moins d'un an" d'expérience est de 5,65 % et de 5 % pour ceux ayant "plus de 20 ans". Bien que suivi de près par la catégorie "5 à moins de 10 ans", la majorité des répondants avaient entre 1 et moins de 5 ans d'expérience, soit 211 sur 620. De plus, 530 participants sur 620, soit 85,48 %, ont confirmé que leur principale profession était la traduction. Les résultats montrent que plus de la moitié d'entre eux, soit 348 ou 56,13 %, s'occupaient normalement de plus de 75 % de ce type de traduction dans leur charge de travail. Viennent ensuite ceux qui ont indiqué "moins de 25 %" comme réponse, ce qui a recueilli 145 réponses (23,39 %). Enfin, le Tableau 2 présente les résultats en termes de type d'emploi.

TYPE D'EMPLOI	RÉPONSES	
Traducteur indépendant	28.06 %	174
Indépendant travaillant aussi avec des agences	25.32 %	157
Traducteur indépendant travaillant seulement avec des agences	16.77 %	104
Traducteur en interne dans une entreprise de traduction	12.42 %	77
Traducteur en interne dans une entreprise non spécialisée en traduction	12.10 %	75
Étudiant	2.90 %	18
Traducteur amateur (Fan translator)	2.42 %	15

Tableau 2. Type d'emploi (traducteurs)

4.4 Pratiques commerciales

La première question portait sur le fait de travailler seul ou en équipe. Si nous analysons les trois options séparément, 39,84 % (247) d'entre eux ont choisi l'option "principalement seul", suivis de 30,97 % (192) qui ont opté pour "les deux", et 29,19 % (181) ont répondu "principalement en équipe". Si nous combinons les deuxième et troisième options, 373 d'entre eux avaient l'habitude de travailler en équipe régulièrement. En ce qui concerne le modèle de production, près de 70 % (432) des répondants travaillaient en sim-ship et 80,09 % d'entre eux ont répondu qu'ils recevaient des changements constants aux documents sur lesquels ils travaillaient et 60,64 % ont déclaré qu'un système suivait ces changements.

4.5 Actifs, accès et linéarité

La première question proposait une liste de différents formats de fichiers couramment utilisés dans l'industrie de la traduction en général. Les résultats ont confirmé que les fichiers Excel sont les plus couramment utilisés, avec 85,16 % (528) de toutes les réponses, suivi de Word avec 297 choix, XML avec 238 choix et TMX avec 202 choix. En ce qui concerne les documents de référence, 23,39 % (145) ont déclaré ne pas en recevoir, mais la majorité des participants, soit 76,61 % (475), ont répondu positivement. 273 répondants ont laissé un commentaire où, dans la plupart des cas, ils ont énuméré les documents de référence qu'ils recevaient généralement (Annexe 4). Les participants ont également été spécifiquement interrogés sur l'accès aux vidéos lorsqu'ils créaient des sous-titres ou traduisaient des dialogues qui seraient doublés ultérieurement. Les résultats montrent que 55 % (341) des répondants reçoivent les vidéos, tandis que 45 % (279) ont sélectionné l'option "non".

En ce qui concerne l'accès au jeu lui-même, le résultat le plus courant est de ne pas avoir accès au jeu et de ne recevoir que les limitations des caractères, avec près de 47 % des réponses. En deuxième place, 145 participants n'ont reçu aucune sorte d'instructions ou d'accès. Cependant, si nous combinons les résultats de ceux qui n'ont eu accès qu'à l'environnement visuel et de ceux qui avaient à la fois le jeu (ou tout autre support visuel) et les limitations des caractères à leur disposition, le chiffre obtenu atteint 184 participants, soit 29,68 %. D'autre part, si nous combinons les résultats des traducteurs travaillant complètement en dehors de tout environnement visuel, nous retrouvons 70,32 % des répondants (436). Enfin, en ce qui concerne la réception du texte avec les dialogues dans l'ordre, seuls 37 participants ont indiqué "toujours"; 220 ont choisi "la plupart du temps"; et 81 "souvent". Ces 3 réponses possibles (les plus positives) combinées, représentent 338 ou 54,52 % du total.

4.6 Attitudes envers les fonctionnalités et les caractéristiques

Les participants ont également évalué une liste de 14 fonctionnalités et caractéristiques, en choisissant soit "inconvénient", "non important", "peu utile", "utile" ou "essentiel" (Zaretskaya et al., 2018, p. 48). Les cinq fonctionnalités et caractéristiques jugées essentielles étaient : "la possibilité de voir les dialogues dans l'ordre" (421 réponses), suivie de "la gestion des termes et les outils d'extraction pour assurer la cohérence" (403 réponses), puis "l'accès aux fichiers audio, vidéo et images" (314 réponses) ; "l'inclusion d'outils d'assurance qualité" (278 réponses) ; et enfin "les outils de traduction assistée par ordinateur" (273 réponses). En ce qui concerne les éléments considérés “inconvénients", "la traduction automatique" a reçu 170 réponses, suivi par "la possibilité de combiner des corpus, des outils de TAO et la traduction automatique" (34), "les outils de TAO en ligne permettant de travailler en équipe" (13), "les outils de traduction assistée par ordinateur" (9) et "la possibilité de compiler des corpus" (6).

Les 99 commentaires laissés pour les fonctionnalités manquantes ont été classés manuellement dans différentes catégories en fonction de leur contenu. La catégorie intitulée "communication" comptait 23 éléments avec des solutions allant des "formulaires de questions-réponses" aux sections de commentaires attachées aux segments en question. "Plus d'accès visuel" a été mentionné 12 fois avec des commentaires variant de la possibilité d'exporter et de visualiser la colonne que certains développeurs ajoutent avec des commentaires d'Excel dans memoQ, à la possibilité d’ajouter des captures d'écran ou de voir les traductions fournies dans d'autres langues ou plus de contexte en général. Avoir une option de prévisualisation dans le jeu qui montre le contexte ou l'accès au jeu lui-même a été mentionné 10 fois. Nous trouvons également différents commentaires sur des solutions d'assurance qualité telles que des correcteurs d'orthographe ou des moteurs de grammaire (9 références), et enfin, une fonctionnalité permettant de compter automatiquement les caractères est apparue 5 fois. Le reste des commentaires a été classé comme "divers" et peut être consulté dans l'annexe 5.

4.7 Outils d'extraction et d'intégration des ressources, de gestion de contenu et de gestion de projet.

Les traducteurs ont été interrogés sur leur usage des outils d'extraction et d'intégration des ressources, de gestion de contenu et de gestion de projet. Les résultats ont montré que 66,61 % (413) des participants en avaient entendu parler mais ne les utilisaient pas ; suivi de "parfois" avec 13,23 % (82) ; "utilisation régulière" avec 10,32 % (64), et enfin "jamais entendu parler" avec 9,84 % des réponses (61). Les résultats obtenus pour la question sur les outils d'extraction et d'intégration des ressources montrent que la plupart du travail est encore effectué manuellement à l'aide d'Excel ou de Word, avec un total de 65,10 % ou 97 réponses. XLOC n'a reçu que 33 réponses, tout comme ceux qui ont choisi "autre". Pour la question sur les outils de gestion de projet, Plunet et Microsoft sont en tête du classement et parmi les réponses des 37 participants qui ont coché "autre", nous trouvons : confidentiel ou non spécifié (13) ; Jira (5) ; memoQ (5) ; Memsource (3) ; Crowdin (2) ; Trados (2) ; Trello (2) ; 1c ERP ; BaccS ; BPS ; basé sur Excel ; Favro ; Lokalise ; Project director ; Protemos ; Redmine ; Testrail ; Smartcat ; Wordbee ; et WORDUX.

4.8 Outils de test linguistique

Près de 62 % des participants (378) ont déclaré avoir travaillé en tant que testeurs linguistiques, soit régulièrement, soit occasionnellement. Parmi eux, 207 ont choisi Jira comme leur programme le plus couramment utilisé, 109 n'utilisaient aucun outil spécifique, et 67 ont opté pour l'option "autre".

4.9 Outils basés sur des arbres à dialogues

La section sur les outils basés sur des arbres à dialogues contenait deux questions : s'ils connaissaient ce type d'outil et s'ils pensaient que cela pourrait être utile. Comme prévu, le nombre d'utilisateurs réguliers était très faible, représentant moins de 2 % des répondants, soit 1,94 %, soit 12 au total, et ceux qui utilisaient parfois ce type de technologie étaient également rares (12,42 %, soit 77). La deuxième question était simplement : "Pensez-vous que vous pourriez en bénéficier ?" et proposait deux options : "oui" ou "non". Les résultats montrent que 79,03 % traducteurs (490) ont choisi "oui" et 130 ont choisi "non". 144 ont décidé de laisser un commentaire, mais beaucoup ont simplement indiqué qu'ils ne pouvaient pas fournir une réponse sincère car ils ne connaissaient pas l'outil en question.

4.10 Ressources

La question sur les ressources que les participants avaient tendance à utiliser régulièrement a reçu 3985 réponses différentes (Tableau 3).

RESSOURCES	RÉPONSES		RESSOURCES	RÉPONSES	
Dictionnaires bilingues	95.32 %	591	Consultation avec des experts	33.71 %	209
Dictionnaires monolingues	78.39 %	486	Corpus comparables	24.52 %	152
Wikipédia	77.74 %	482	Portails de bases terminologiques	20.00 %	124
Bases de données terminologiques	73.23 %	454	Corpus parallèles	17.90 %	111
Forums en ligne	63.23 %	392	Dictionnaires visuels bilingues	11.61 %	72
Moteurs de recherche d'images	53.55 %	332	Dictionnaires visuels monolingues	9.19 %	57
Thésaurus	42.90 %	266	Autre (veuillez préciser) :	5.48 %	34
Moteurs de recherche spécialisés et méta-moteurs de recherche	35.97 %	223	Wikis (8) ; Outils de synonymes et de collocations (6) ; Google (5) ; Autres jeux (4) ; Sites web officiels (3) ; YouTube (3) ; Autres dictionnaires (3) ; Sites web de fans (2) ; Antidote (2) ; Mémoires de traduction (2) ; Proz (2) ; Sites de grammaire (2) ; Reverso ; Linguee ; et outil terminologique de Microsoft.		

Tableau 3. Ressources (traducteurs)

4.11 Outils de compilation de corpus

Les résultats montrent que les corpus parallèles bilingues étaient les plus consultés avec 45,81 % (284), suivi de "aucun" avec 37,90 % (235), et les corpus comparables bilingues en troisième place avec 139 réponses (22,42 %). Cependant, la grande majorité a déclaré qu'ils ne compilent pas leurs propres corpus (91,61 % ou 568). Les participants qui le faisaient ont été interrogés sur les outils qu'ils utilisaient, et parmi leurs réponses, la première option était "autre" avec 45,28 % des réponses, suivie par SketchEngine (28,30 %), et AntCorGen en troisième position (22,64 %), suivi de BootCat (20,75 %) et CLaRK (5,66 %).

4.12 Outils d'extraction et de gestion de terminologie

La majorité des répondants en avaient entendu parler, mais n'utilisaient pas ces outils (64,35 % soit 399), 80 participants n'en avaient jamais entendu parler ; 39 (6,29 %) ont déclaré les utiliser régulièrement et 16,45 % ont dit les utiliser occasionnellement. En ce qui concerne les systèmes autonomes, plus de 86 % (538) ne les utilisaient pas, suivi de "autre" avec 37 réponses. Plus de 94 % des participants ont choisi l'option "aucun" pour les outils d'extraction et de gestion de terminologie utilisés uniquement pour les corpus (soit autonomes, soit intégrés aux outils). La deuxième option la plus courante était, une fois de plus, "autre" avec seulement 2,74 % des réponses. Le tableau 4 présente les résultats pour les solutions intégrées directement dans les outils TAO.

RÉPONSES	FRÉQUENCE
Aucun	393
SDL MultiTerm Extract	171
qTerm™ (memoQ)	50
QuickTerm	24
crossTerm	7
Memsource	3
Crowdin, Wordfast, Lokalise, Déjà Vu, Xbench	1
Sous NDA ou non spécifié	9

Tableau 4. Programmes intégrées dans les outils TAO (traducteurs)

4.13 Outils de traduction assistée par ordinateur

Les réponses montrent que 81,45 % des participants (505) les utilisaient régulièrement et 11,13 % (69) de manière occasionnelle. Ces chiffres, combinés, représentent un total de 92,58 % des 620 répondants, soit 574 utilisateurs. Le tableau 5 regroupe les réponses aux questions sur des outils spécifiques.

OUTILS TAO		OUTILS DE LOCALISATION	
MemoQ	496	Aucun	229
SDL Trados	352	SDL Passolo	161
Memsource	236	Poedit	102
SmartCAT	118	Catalyst	49
Wordfast	103	MultiTrans	40
Omega T	72	Lokalize	28
MateCat	41	Confidentiel / non spécifié	24
Déjà Vu	28	Gtranslator	13
XTM	21	Crowdin	10
Wordbee	16	Sisulizer	5
Crowdin	15	Localize Direct, Wordbee, LEAF	2
Confidentiel / non spécifié	11	DejaVu, Transifex, Verifika, Xbench, XTM	1
LEAF, Smartling	8		
CafeTran	6		
Across	4		
Lingotek, TWS, Transifex	3		
Transtool, Polyglot, Sisulizer, Message Studio, Similis	2		

Tableau 5. Outils TAO et de localisation (traducteurs)

4.14 Outils de traduction automatique

50,81 % des répondants (315) ont déclaré connaître l'existence de ces outils mais ne pas les utiliser, 37,10 % ont déclaré les utiliser parfois, 11,61 % les utilisaient régulièrement et 3 des participants n'en avaient jamais entendu parler. Près de 47 % (143) n'utilisaient pas du tout d'outils commerciaux, 120 participants utilisaient Google Translate API et 35 ont choisi "autre". En ce qui concerne les systèmes gratuits, Google Translate était en première position avec 191 réponses, suivi de DeepL (128) ; la troisième option était "aucun" avec 56 ; puis Babylon (22) et "autre" était en cinquième position.

5. Résultats de l'enquête pour les testeurs

La version définitive de l'enquête adressée aux testeurs linguistiques a été finalisée au cours de la dernière semaine de juillet 2020 et le lien utilisé comme collecteur a été créé le 31 du même mois. Le collecteur a été officiellement fermé le 16 septembre de la même année et le questionnaire a reçu un total de 770 réponses, dont 550 réponses complètes. Également créé à l'aide de SurveyMonkey, le questionnaire complet était divisé en 7 sections organisées sur 4 pages et comportait seulement 20 questions.

5.1 Informations personnelles

Aucun des 550 participants ayant rempli l'enquête n'appartenait aux groupes "17 ans ou moins" ou "65 ans et plus". Le groupe le plus représenté est celui des testeurs âgés de "25 à 34 ans" qui compte un total de 358 participants ; il est suivi par ceux âgés de 35 à 44 ans, bien que la différence entre les chiffres soit marquée (114 répondants). Seuls 58 participants étaient concernés par le deuxième groupe d'âge, 17 se trouvaient dans la catégorie "45 à 54 ans" et 3 ont sélectionné la sixième option. Parmi les 226 nationalités différentes répertoriées, 63 avaient au moins un représentant et 21 d'entre elles ont été choisies 10 fois ou plus. Le Tableau 6 rassemble les 10 principales nationalités.

NATIONALITÉS	RÉPONSES	
Française	12.60 %	97
Italienne	10.52 %	81
Espagnole	10.52 %	81
Polonaise	8.31 %	64
Brésilienne	7.27 %	56
Allemande	4.55 %	35
Russe	2.73 %	21
Turque	2.47 %	19
Américaine	2.34 %	18
Canadienne	2.21 %	17

Tableau 6. Nationalités avec plus de 10 répondantes (testeurs)

5.2. Informations professionnelles

Les résultats concernant leurs langues maternelles ou de travail placent l'anglais en tête de liste avec 579 réponses, suivi de l'espagnol (165), du français (153) et de l'italien (89). Nous constatons que 287 participants (52,18 %) n'avaient aucun diplôme universitaire en traduction ou en études linguistiques, mais cela n'exclut pas une éducation supérieure. En revanche, 35,09 % d'entre eux, soit 193 répondants, possédaient une maîtrise, une licence ou un doctorat dans ces domaines. Seuls 70 d'entre eux avaient suivi des cours spécialisés, des séminaires, des ateliers ou des formations similaires. 287 participants (52,28 %) avaient "1 à moins de 5 ans” d'expérience professionnelle et la deuxième réponse la plus courante était "moins d'un an" avec 132 réponses. Le troisième groupe du top trois a reçu près de 17 % des réponses et était composé de participants ayant moins de 10 ans d'expérience dans le domaine du LQA. De plus, 333 d'entre eux (soit 60,55 % du total) ont déclaré que le test linguistique était leur principale source de revenus, tandis que les autres ont répondu "non" (39,45 % ou 217). Enfin, 48,55 % (soit 267) d'entre eux ont déclaré effectuer "plus de 75 %" de leur travail en LQA, peu d'entre eux consacrant environ 50 % de leur temps à des tâches de LQA (48 participants) ou plus de 50 % mais moins de 75 % (40 répondants). En revanche, la deuxième réponse la plus courante était "moins de 25 %" avec 195 réponses (35,45 %). Le Tableau 7 présente les résultats pour le type de contrat.

TYPE DE CONTRAT	RÉPONSES	
En interne dans une entreprise non spécialisée en traduction	47.27 %	260
En interne dans une entreprise de traduction	24 %	132
Indépendant	11.27 %	62
Indépendant travaillant seulement avec des agences	7.64 %	42
Indépendant travaillant aussi avec des agences	7.27 %	40
Étudiant	2.55 %	14

Table 7. Type de contrat (testeurs)

5.3 Pratiques professionnelles

Dans le cas de travail en équipe, "principalement en équipe" a reçu le plus grand nombre de réponses avec 65,27 % (359), 20,73 % (114) ont choisi "les deux" et 14 % (77), ont sélectionné "principalement seul". En ce qui concerne les tâches liées à l'assurance qualité linguistique, les résultats montrent également une tendance à résoudre les problèmes linguistiques détectés (53,8 %). Dans le cas de 34,73 % des répondants, cela dépendait du projet, et seulement 11,45 % d'entre eux étaient chargés de les signaler sans prendre d'autres mesures. De manière similaire à la première enquête, l'outil le plus largement utilisé était Jira (439), suivi de DevTrack qui a reçu 136 réponses, et de "logiciel propriétaire".

5.4 *Bugs* : Problèmes linguistiques

Le type le plus fréquent de problème linguistique auquel les testeurs sont confrontés est le "Débordement, chevauchement ou troncature de texte", choisi 318 fois comme étant le plus courant et 95 fois comme le deuxième plus courant, selon les résultats détaillés. En revanche, les résultats concernant les "Instructions confuses" montrent que cette catégorie a été classée dernière à 300 reprises et cinquième à 127 reprises, ce qui en fait clairement le type de problème le moins courant. De même, en ce qui concerne les "Erreurs de sous-titrage", 240 participants l'ont sélectionné comme étant la cinquième option, bien qu'un nombre assez élevé de participants (158) aient également pensé qu'elle devrait figurer en dernière position. Les "Mauvaises traductions" ont été identifiées comme le deuxième type le plus courant et ont été choisies comme telles par 182 participants. Elles ont été suivies des "Erreurs grammaticales et typographiques" puis des "Incohérences terminologiques". Dans ces cas, les résultats détaillés sont moins tranchants car, bien que la position ayant reçu le plus grand nombre de réponses suggérât un échange de place dans le classement, c'est finalement le poids moyen des répondants qui a déterminé leur position finale dans la liste, avec des différences minimes entre elles. Les tableaux suivants présentent ce que les répondants ont jugé être les causes de ces problèmes en fonction de leur expérience (tableaux 8 à 12).

Selon votre expertise, quelle est la principale raison des débordements, chevauchements ou troncatures de texte ?	**RÉPONSES**	
Manque d'environnement visuel pour les traducteurs.	51.64 %	284
Instructions insuffisantes pour les traducteurs ou non-respect des instructions.	16.36 %	90
Toutes les raisons ci-dessus.	15.09 %	83
Autre (veuillez préciser).	13.09 %	72
Je ne sais pas.	2.18 %	12
Manque de qualité de traduction.	1.64 %	9

Tableau 8. Causes des débordements, chevauchements ou troncatures (testeurs)

Selon votre expertise, quelle est la principale raison des erreurs de traduction ?	**RÉPONSES**	
Manque de contexte dans le texte (par exemple, les segments ne sont pas dans l'ordre, etc.)	40.73 %	224
Toutes les raisons ci-dessus.	16.73 %	92
Manque d'environnement visuel pour les traducteurs.	16.00 %	88
Instructions insuffisantes pour les traducteurs ou non-respect des instructions.	11.82 %	65
Manque de qualité de traduction.	9.64 %	53
Autre (veuillez préciser).	4.00 %	22
Je ne sais pas.	1.09 %	6

Tableau 9 : Causes des erreurs de traduction (testeurs)

Selon votre expertise, quelle est la principale raison des incohérences terminologiques ?	**RÉPONSES**	
Manque de ressources telles que des glossaires, des bibles de personnages, etc.	33.27 %	183
Toutes les raisons ci-dessus.	24.18 %	133
Trop de traducteurs travaillant sur le même projet.	20.91 %	115
Instructions insuffisantes pour les traducteurs ou non-respect des instructions.	14.55 %	80
Autre (veuillez préciser).	3.64 %	20
Je ne sais pas.	3.45 %	19

Tableau 10 : Causes des incohérences terminologiques (testeurs)

Selon votre expertise, quelle est la principale raison des problèmes de sous-titres ?	**RÉPONSES**	
Problèmes informatiques.	27.27 %	150
Manque d'environnement visuel pour les traducteurs.	22.73 %	125
Je ne sais pas.	14.36 %	79
Toutes les raisons ci-dessus.	11.27 %	62
Instructions insuffisantes pour les traducteurs ou non-respect des instructions.	10.36 %	57
Manque de qualité de traduction.	8.18 %	45
Autre (veuillez préciser).	5.82 %	32

Tableau 11. Causes des problèmes de sous-titres (testeurs)

Selon votre expertise, quelle est la principale raison des instructions confuses ?	**RÉPONSES**	
Manque de contexte dans le texte.	29.45 %	162
Manque d'environnement visuel pour les traducteurs.	21.64 %	119
Instructions insuffisantes pour les traducteurs ou non-respect des instructions.	14.73 %	81
Toutes les raisons ci-dessus.	14.00 %	77
Je ne sais pas	7.27 %	40
Manque de qualité de traduction.	7.09 %	39
Autre (veuillez préciser).	5.82 %	32

Tableau 12. Causes d'instructions confuses (testeurs)

6. Résultats de l'enquête pour les développeurs

Le lien utilisé en tant que collecteur a été créé le 12 janvier 2021, et l'enquête a été officiellement clôturée le 5 mai 2021. Cependant, le questionnaire n'a été activement envoyé qu'au cours des deux premiers mois en raison des incohérences repérées lors de l'examen des données préliminaires. Une inspection plus approfondie a révélé que certains participants avaient laissé des réponses fausses (voir l'Annexe 7 pour un exemple). Par conséquent, une partie du temps initialement consacré à la distribution du lien a dû être réaffectée à l'examen individuel des réponses et à la recherche d'incohérences dans les trois enquêtes. Ainsi, seuls 439 participants sur un total de 554 ont atteint la dernière page et fourni des informations exploitables. Cette enquête comportait 23 questions et était divisée en 8 sections réparties sur 7 pages.

6.1 Informations personnelles

Aucun des 439 participants n'avait 65 ans ou plus, mais 3 participants se trouvaient dans le groupe des moins de 17 ans. Le même nombre de répondants (3) faisait partie du groupe des "55 à 64 ans" et seuls 11 d'entre eux (2,51 %) avaient entre 45 et 54 ans. Encore une fois, le plus grand groupe était celui des développeurs avec une moyenne d'âge de "25 à 34 ans", représentant 55,13 % des résultats (242). Le deuxième résultat le plus élevé était pour "18 à 24 ans", avec 26,42 % (116), et enfin "35 à 44 ans" avec 64 réponses. Sur les 226 nationalités différentes répertoriées, 57 en avaient au moins un représentant et 10 d'entre elles avaient été sélectionnées au moins 10 fois (Tableau 13).

NATIONALITÉS	RÉPONSES	
Française	26.36 %	116
Turque	9.32 %	41
Indienne	8.86 %	39
Espagnole	8.41 %	37
Américaine	4.09 %	18
Italienne	3.86 %	17
Iranienne	3.64 %	16
Pakistanaise	3.18 %	14
Allemande	2.50 %	11
Brésilienne	2.28 %	10

Tableau 13. Nationalités avec plus de 10 répondants (développeurs)

6.2 Informations professionnelles

198 des participants (45,10 %) étaient titulaires d'une licence, suivi de près par ceux ayant un master (181 ou 41,23 %). Le pourcentage de ceux qui n'avaient pas suivi d'études universitaires était très faible par rapport aux autres enquêtes, bien que la principale raison de ces résultats soit le fait que nous n'avons pas précisé de domaine d'études particulier. Quant à l'expérience, l'option qui a reçu le plus de réponses était "1 à moins de 5 ans" avec 210 réponses (47,84 %), suivie de "5 à moins de 10 ans" avec 24,15 % (106 participants) et "moins d'un an" avec 73. Dans cette enquête, 387 d'entre eux (soit 88,15 %) ont déclaré que le développement de jeux vidéo était leur principale activité, confirmant ainsi que la majorité des répondants étaient des experts dans ce domaine. Dans cette enquête, la majorité des répondants avaient une charge de travail d'au moins 50 % dans le domaine : 300 d'entre eux (soit 68,34 %) ont choisi "plus de 75 %" et "plus de 50 %" a reçu 53 réponses (12,07 %). De plus, parmi les 44 commentaires laissés, 19 ont déclaré que cela occupait 100 % de leur temps. Le tableau 14 présente les résultats en termes de type d'emploi.

TYPE DE CONTRAT	RÉPONSES	
Développeur en interne dans un studio indépendant	43.96 %	193
Développeur indépendant	21.41 %	94
Développeur en interne dans un studio détenu par un éditeur	19.13 %	84
Étudiant	7.29 %	32
Autre (veuillez préciser)	4.78 %	21
Développeur en interne dans un studio détenu par une plateforme	3.42 %	15

Tableau 14. Type de contrat (développeurs)

6.3 Pratiques commerciales

Seulement 28,02 % des participants suivaient le système post-gold, et près de 72 % travaillaient avec un modèle sim-ship (71,98 % pour être plus précis). Lorsqu'on a demandé aux participants à quel moment commençait la phase de localisation dans le modèle sim-ship, 111 participants (35,02 %) ont choisi Alpha, suivi par Beta avec 74 réponses (23,34 %), 63 répondants (19,87 %) ont sélectionné "prototype ou first playable" et 41 (12,93 %) ont marqué "vertical slice ou beautiful corner". Seulement 60 participants (13,67 %) utilisaient des systèmes de stockage de données, et 86,33 % (379) des participants ont déclaré utiliser des fichiers externes pour isoler le texte de jeu du code : 65,17 % ou 247 participants ont choisi des fichiers .json, 36,68 % (139 XML et 29,82 % (ou 113) ont choisi Excel.

6.4 Données techniques

La première question concernait les plateformes utilisées pour jouer aux jeux vidéo qu'ils créaient, et l'option ayant reçu le plus grand nombre de réponses était "jeux mobiles" avec 35,54 % des réponses, suivie des "jeux PC téléchargés/boîtiers" avec 32,57 %, et des "jeux sur console" avec près de 20 % des réponses. En ce qui concerne le système d'exploitation qu'ils utilisaient régulièrement, Windows arrivait en tête avec 88,15 % du total (387 réponses), suivi de Mac OS avec 9,34 % (ou 41), et Linux (1,14 % ou 5). Enfin, en ce qui concerne les langages de programmation utilisés, nous trouvons C# (259), suivi de C++ (153), Html5 & JavaScript (14), Java (5) et Python (4).

6.5 Outils de développement de jeux vidéo

Les résultats montrent que le moteur le plus largement utilisé est Unity avec 312 réponses (soit 71,07 % du total). Unreal Engine a obtenu 32,35 % avec 142 utilisateurs et, comme c'est courant dans l'industrie, 18,22 % des répondants n'ont pas pu fournir le nom du moteur utilisé en raison de clauses de confidentialité. Les participants ont également été interrogés sur leur connaissance de XLOC et ce qu'ils en pensaient. La grande majorité des participants (388 d'entre eux, soit 88,38 %) n'avaient jamais entendu parler de XLOC, le pourcentage de ceux qui en avaient entendu parler mais ne l'avaient jamais utilisé était très faible (moins de 10,5 %) et seuls 5 participants l'utilisaient régulièrement ou occasionnellement.

6.6 Les attitudes envers les tests

41.69 % des répondants semblent considérer les tests comme une réflexion après coup, tandis que 146 participants déclarent ne pas utiliser de tests unitaires ou fonctionnels du tout. Seulement 19 participants déclarent qu'ils créent des tests très "défensifs" (qui testent toutes les erreurs possibles plutôt que celles qui sont probables), même s'ils ne couvrent pas l'intégralité du code, et 20.73 % d'entre eux essaient de couvrir 100 % du code. En ce qui concerne les tests de type “play-mode”, 38.72 % d'entre eux (soit 170 participants) ne les utilisent pas du tout, 25.74 % affirment couvrir 100 % du gameplay, 22.78 % les utilisent uniquement dans les zones vitales ou complexes, et 12.76 % déclarent que leurs méthodes de test actuelles ne leur permettent pas de les utiliser.

6.7 Le processus de développement de jeux vidéo

La dernière question était sur le processus de développement utilisé (Tableau 15).

PROCESSUS DE DÉVELOPPEMENT DE JEUX VIDÉO	RÉPONSES	
Itératif : c'est un processus qui consiste à développer un logiciel en répétant de courts cycles pour livrer une fonctionnalité prête à l'emploi à chaque fois. La méthodologie agile de développement de logiciels suit cette approche itérative, améliorant continuellement et systématiquement ses processus et ses pratiques.	47.84 %	210
Hybride : c'est une combinaison des processus en cascade et itératifs dans le même projet. En général, la stratégie en cascade est utilisée pendant la phase de pré-production/post-production et l'approche itérative est appliquée pendant la phase de production.	27.56 %	121
En cascade ou prédictif : c'est un processus séquentiel dans lequel la phase suivante démarre uniquement si la phase précédente est complètement terminée, livrant ainsi la valeur commerciale en une seule fois. Il s'agit du processus traditionnel de développement de jeux, exigeant des évaluations explicites des exigences, suivies de procédures de résolution de problèmes ordonnées et précises.	10.71 %	47
Je ne sais pas.	10.71 %	47
Je ne peux pas le dire car soumis à une clause de non-divulgation (NDA)	9.11 %	40
Adhoc : c'est un processus créé uniquement pour un projet spécifique, sans définition préalable. Dans le processus ad-hoc, les activités sont définies selon les besoins et le processus change pour répondre à des problèmes ponctuels et contextuels.	7.52 %	33
Autre (veuillez préciser).	1.59 %	7

Tableau 15. Processus de développement de jeux (développeurs)

7. Analyse des résultats et discussion : Âge, nationalité et langues de travail

Le nombre total de participants des trois enquêtes âgés de "25 à 34" ans représente 946 des 1609 répondants, un groupe qui comprend les personnes qui ont grandi et reçu leur éducation pendant que l'industrie était en train de se développer. Le même principe s'applique au deuxième groupe le plus commun dans les première et deuxième enquêtes, les répondants âgés de "35 à 44" ans. Cependant, en comparaison avec les deux autres enquêtes, nous observons que le deuxième groupe le plus commun parmi les répondants de la troisième enquête est celui des "18 à 24" ans. Ce changement dans le schéma est le reflet clair de la popularité croissante de l'industrie dans le monde entier et de l'augmentation du nombre de sociétés de jeux vidéo. Cette popularité se reflète également dans les résultats des deux autres enquêtes et place les répondants appartenant au groupe "18 à 24" ans en deuxième position.

Le Tableau 16 présente les résultats combinés de la nationalité, de la langue maternelle et de la langue cible des traducteurs afin d'obtenir une vue plus claire. La première différence apparaît lorsque nous atteignons la quatrième position, où l'on trouve "Brésilien" en tant que nationalité, mais "Anglais" à la fois comme langue maternelle et langue cible. Cela est dû à la présence de répondants qui étaient soit bilingues, soit trilingues. De plus, avec l'émergence d'autres pratiques de localisation en plus de la combinaison E-FIGS, nous pouvons également observer la présence de langues telles que le russe, le mandarin, le polonais et le turc.

NATIONALITÉS		LANGUE MATERNELLE		LANGUE CIBLE	
Espagnole	113	Espagnol	158	Espagnol	149
Française	107	Français	121	Français	111
Italienne	69	Italien	72	Italien	68
Brésilienne	46	Anglais	63	Anglais	52
Allemande	43	Allemand	49	Allemand	47
Turque	23	Portugais	47	Portugais	46
Russe	22	Russe	36	Russe	33
Argentine	20	Turc	25	Turc	22
Américaine	16	Catalan	24	Mandarin	14
Polonaise	13	Mandarin	22	Polonais	14

Tableau 16. Comparaison nationalité, langue maternelle et langue cible des traducteurs

Enfin, lorsque nous effectuons la même comparaison pour les testeurs linguistiques, l'anglais est la première langue maternelle ou de travail avec 579 réponses, bien que la première nationalité d'un pays anglophone se situe en 10e position. La présence de bilingues a été confirmée lorsque nous avons filtré les résultats en utilisant uniquement ceux qui ont choisi "anglais", et nous avons constaté qu'il a été sélectionné par 412 participants. En analysant les nationalités de ces 412 personnes, nous avons observé que 47 étaient français, 47 étaient italiens, 36 étaient polonais, 33 étaient brésiliens et 31 étaient espagnols. Seulement 90 testeurs ont déclaré venir d'un pays où l'anglais est une langue officielle (ou l'une des langues officielles).

7.2 Études et expérience professionnelle des participants

Lorsque nous examinons le niveau d'éducation des participants, nous constatons que la plupart des traducteurs et des développeurs ont soit une maîtrise soit une licence. Pour les testeurs linguistiques, l'option la plus courante était "aucune", car la question portait spécifiquement sur les diplômes en traduction ou en langues. En ce qui concerne l'expérience professionnelle, la réponse la plus fréquente était "1 à moins de 5 ans". Ce résultat peut sembler légèrement incohérent au premier abord, si nous le comparons aux réponses sur l'âge des participants, car la plupart d'entre eux étaient soit dans la tranche d'âge "25 à 34 ans", soit dans la tranche d'âge "35 à 44 ans". Cependant, en ce qui concerne le premier groupe d'âge, cela suggère que les répondants étaient principalement dans la moitié inférieure de la tranche d'âge et, lorsque nous combinons les deux groupes, les résultats peuvent également s'expliquer par le fait qu'il est assez difficile d'entrer dans l'industrie au début et qu'il faut du temps pour se spécialiser pleinement.

La deuxième réponse la plus fréquente pour les traducteurs et les développeurs est "5 à moins de 10 ans", ce qui est cohérent avec l'âge des participants. Cependant, pour les testeurs linguistiques, la réponse la plus courante est "moins d'un an" et l'analyse croisée montre qu'un pourcentage élevé d'entre eux vient de commencer dans le domaine, car la plupart appartenaient au groupe d'âge "25 à 34 ans". De plus, un pourcentage élevé d'entre eux (43,18 %) ne travaillaient pas exclusivement dans l'assurance qualité linguistique. Finalement, lorsque nous filtrons les résultats concernant la charge de travail des participants dans leur domaine principal d'emploi, ceux qui ont indiqué que la localisation, le test linguistique ou le développement de jeux vidéo étaient leur principale source de revenus ont également déclaré qu'ils travaillaient principalement dans ces domaines à plus de 75 % du temps. Dans le cas de l'assurance qualité linguistique et du développement de jeux vidéo, le pourcentage de spécialistes est plus élevé que dans le cas de la localisation de jeux vidéo (72,07 % et 76,23 % versus 58,68 %). Les résultats montrent que les traducteurs travaillant en freelance sont plus nombreux que dans les autres domaines où les contrats internes sont plus répandus. En tant que travailleurs indépendants, les traducteurs sont plus susceptibles d'accepter d'autres types de projets de traduction en complément de leur activité principale.

7.3 Pratiques commerciales contrastées

Nous constatons que 69,68 % des traducteurs et 71,98 % des développeurs ont déclaré travailler avec le modèle sim-ship et commencer la localisation pendant la phase de développement, ce qui accentue la nécessité de suivre les modifications. Par conséquent, éviter l'envoi de chaînes obsolètes à localiser pour réduire à la fois les coûts et le travail supplémentaire, ou fournir aux traducteurs un moyen rapide de détecter les changements dans la langue source (ou pivot) devient essentiel. Lorsque nous examinons cet aspect spécifique d'un point de vue technique et comparons les méthodes utilisées par les développeurs pour isoler le texte du jeu avec les types de fichiers que les traducteurs reçoivent généralement, nous pouvons identifier une autre étape problématique du processus qui pourrait entraîner des erreurs humaines : la nécessité de copier-coller manuellement. Les résultats ont montré que 65,17 % (soit 247 participants) ont sélectionné les fichiers .json comme type de fichier pour isoler le texte. Cependant, seuls 7 traducteurs ont mentionné avoir reçu des fichiers .json, alors que les fichiers Excel ont reçu un total de 528 réponses. Enfin, le Tableau 17 rassemble les résultats concernant le matériel de référence, les fichiers vidéo, l'accès au jeu et les différentes options incluses pour la localisation des menus ou des interfaces utilisateur, afin d'avoir une vue complète de l'impact du type de contrat sur ces pratiques.

	Matériel de référence	Fichiers vidéo	Accès visuel	Visuel et caractères	Nombre de caractères	Rien
Tout contrat	76.61 %	55 %	14.84 %	14.84 %	46.94 %	23.39 %
Indépendant	77.59 %	48.28 %	15.52 %	9.20 %	47.70 %	27.59 %
Indépendant travaillant seulement pour des agences	71.15 %	45.19 %	7.69 %	11.54 %	59.62 %	21.15 %
Indépendant travaillant aussi pour des agences	76.43 %	41.40 %	7.64 %	13.38 %	52.23 %	26.75 %
Employé chez un développeur	92.02 %	89.33 %	42.67 %	29.33 %	18.67 %	9.33 %
Employé dans une agence traduction	84.42 %	81.82 %	9.09 %	19.48 %	50.65 %	20.78 %

Tableau 17. Accès aux ressources, vidéos ou éléments visuels en fonction de l'emploi.

7.4 Questions clés et données techniques

7.4.1 Attitudes des traducteurs en fonction des types de contrats

Dans le cas des 174 traducteurs indépendants, il n'y a eu aucun changement ni dans l'ordre ni dans les éléments choisis comme "essentiels" : "La possibilité de voir les dialogues dans l'ordre" (110), "Outils de gestion et d'extraction de terminologie pour assurer la cohérence" (102), "Accès à l'audio, à la vidéo et aux images" (75), "Inclure des outils d'assurance qualité" (71) et "Outils de traduction assistée par ordinateur" (69). En ce qui concerne leurs réponses concernant les fonctionnalités "inconvénients", "Traduction automatique" reste en première position (42), suivi de "La possibilité de combiner des corpus, des outils CAT et la traduction automatique" (12). "Pouvoir utiliser des corpus", "Pouvoir compiler des corpus" et "Outils de traduction assistée par ordinateur" ont chacun reçu 4 réponses.

Parmi les 157 travailleurs indépendants travaillant aussi avec des agences, nous trouvons les outils TAO en troisième place les fonctionnalités "essentielles" : "La possibilité de voir les dialogues dans l'ordre" (122), "Outils de gestion et d'extraction de terminologie pour assurer la cohérence" (112), "Outils de traduction assistée par ordinateur" (76), "Accès à l'audio, à la vidéo et aux images" (74) et "Inclure des outils d'assurance qualité" (70). En revanche, en bas de la liste, nous trouvons que "Outils de sous-titrage pour les dialogues parlés et les cinématiques" et "Outils de signalement de *bugs*", tous deux avec 3 réponses, occupent la cinquième position : "Traduction automatique" (52), "La possibilité de combiner des corpus, des outils TAO et la traduction automatique" (12), "Travailler sur des outils TAO en ligne permettant de travailler en équipe" (6) et "Outils de traduction assistée par ordinateur" (4).

Les 104 indépendants qui ne travaillaient que avec des agences ont également obtenu des résultats similaires pour les fonctionnalités "essentielles", bien que les outils TAO soient à la quatrième position au lieu de la cinquième : "La possibilité de voir les dialogues dans l'ordre" (76), "Outils de gestion et d'extraction de terminologie pour assurer la cohérence" (74), "Accès à l'audio, à la vidéo et aux images" (64), "Outils de traduction assistée par ordinateur" (54) et "Inclure des outils d'assurance qualité" (50). En ce qui concerne les fonctionnalités "inconvenantes", "Accès à tous les éléments dans leur forme originale et divisés par formats", "Possibilité de suivre toutes les modifications apportées aux fichiers source" et "Outils de signalement de *bugs*" ont chacun reçu 1 réponse et sont dans le top cinq, puis : "Traduction automatique" (31), "La possibilité de combiner des corpus, des outils CAT et la traduction automatique" (7) et "Travailler sur des outils TAO en ligne permettant de travailler en équipe" (2).

Pour les 77 traducteurs travaillant dans une entreprise de traduction, la gestion de la terminologie devient la première priorité, et l'assurance qualité prend la troisième place : "Outils de gestion et d'extraction de terminologie pour assurer la cohérence" (55), "La possibilité de voir les dialogues dans l'ordre" (49), "Inclure des outils d'assurance qualité" (41), "Accès à l'audio, à la vidéo et aux images" (40) et "Outils de traduction assistée par ordinateur" (34). De l'autre côté du classement, seules 4 fonctionnalités ont reçu des réponses négatives : "Traduction automatique" (29), "La possibilité de combiner des corpus, des outils TAO et la traduction automatique" (2), "Outils de traduction assistée par ordinateur" (1) et "Travailler sur des outils TAO en ligne permettant de travailler en équipe" (1).

Enfin, les résultats pour les 75 traducteurs travaillant dans une entreprise non spécialisée dans la traduction montrent que 55 d'entre eux ont choisi la "Possibilité de suivre les modifications dans les fichiers source" comme essentielle et elle est devenue la première de la liste. De plus, l'assurance qualité a disparu du top cinq : "Possibilité de suivre les modifications dans les fichiers source (outils de gestion)" (55), "La possibilité de voir les dialogues dans l'ordre" (45), "Outils de gestion et d'extraction de terminologie pour assurer la cohérence" (44), "Accès à l'audio, à la vidéo et aux images" (38) et "Outils de traduction assistée par ordinateur" (31). Dans ce cas, encore moins de fonctionnalités ont été jugées "inconvenantes" : "Traduction automatique" (13), "Accès à tous les éléments dans leur forme originale et divisés par formats" (1) et "La possibilité de combiner des corpus, des outils TAO et la traduction automatique" (1).

7.4.2 Les *bugs* linguistiques

Lorsque nous analysons la principale cause en tête de liste pour chaque *bug*, nous constatons que le "manque de contexte dans le texte" apparaît deux fois, pour les erreur de traductions et les "instructions confuses". C'est le résultat de la pratique d'envoyer le texte dans des fichiers Excel sans contexte (45,48 % du temps selon la première enquête). Le "manque d'environnement visuel" est apparu comme la principale cause des "débordements, chevauchements et troncatures", en troisième position pour les "erreurs de traduction", en deuxième position pour les "erreurs de sous-titrage" et également en deuxième position pour les "instructions confuses". En tout, il est apparu quatre fois car, comme mentionné précédemment, l'accès au jeu lui-même est rare en général, quel que soit le type de contrat. Parmi les autres causes de la liste, nous trouvons que "Instructions insuffisantes pour les traducteurs ou non-respect des instructions" était la deuxième cause des "débordements, chevauchements et troncatures", la quatrième des "erreur des traduction", la quatrième pour les "incohérences terminologiques", la cinquième en cas d'"erreurs de sous-titrage" et la troisième pour les "instructions confuses". Les résultats de la première enquête ont montré que 23,39 % (ou 145) des traducteurs n'avaient aucune information concernant le nombre de caractères maximale.

7.4.3 Processus de développement, phase de localisation et données techniques.

Afin d'étudier s'il existait des différences significatives dans les processus de développement adoptés par l'équipe en fonction du type de contrat, nous avons décidé de filtrer les résultats des trois principales méthodes (Tableau 18).

	Développeur indépendant	En interne dans un studio indépendant	En interne dans un studio d’un éditeur	En interne dans un studio de plateforme	Étudiant	Autre
Itératif 48.52 % (213)	40.43 % 38	53.89 % 104	48.81 % 41	46.67 % 7	46.88 % 15	38.10 % 8
Hybride 27.79 % (122)	38.30 % 36	23.32 % 45	25 % 21	53.33 % 8	21.88 % 7	23.81 % 5
Cascade 10.71 % (47)	14.89 % 14	8.29 % 16	8.33 % 7	13.33 % 2	15.63 % 5	14.29 % 3
Total : 439	94	193	84	15	32	21

Tableau 18. Processus de développement par type de contrat

Dans le cas du début de la phase de localisation, le Tableau 19 affiche toutes les réponses classées selon le type de contrat et inclut les résultats globaux pour chaque version, triés par ordre décroissant. La dernière ligne comprend également le nombre total et le pourcentage de participants ayant travaillé avec un modèle sim-ship, ainsi que les différentes données et pourcentages pour chaque type de contrat.

	Développeur indépendant	En interne dans un studio indépendant	En interne dans un studio d’un éditeur	En interne dans un studio de plateforme	Étudiant	Autre
Alpha 35.02 % (111)	35.71 % 20	41.73 % 58	33.33 % 22	15.38 % 2	20.83 % 5	21.05 % 4
Beta 23.34 % (74)	21.43 % 12	23.02 % 32	25.76 % 17	38.46 % 5	16.67 % 4	21.05 % 4
Prototype 19.87 % (63)	26.79 % 15	14.39 % 20	21.21 % 14	7.69 % 1	45.83 % 11	10.53 % 2
Vertical slice 12.93 % (41)	14.29 % 8	10.79 % 15	7.58 % 5	30.77 % 4	16.67 % 4	26.32 % 5
Total : 317 (72.21 %)	56/94 (59.57 %)	139/193 (72.02 %)	66/84 (78.57 %)	13/15 (86.67 %)	24/32 (75 %)	19/21 (90.48 %)

Table 19. Début de la phase de localisation par type de contrat

Dans le processus de développement, une partie significative du jeu n'est pas encore finalisée pour la version Alpha et la plupart du travail d'assurance qualité de localisation (LQA) est prévu pour la version Beta car elle est presque finale, y compris en ce qui concerne la traduction, et seules les erreurs critiques seront corrigées. Cela signifie que Alpha a de nombreux problèmes de fonctionnalité qui rendent le jeu très instable et sujet aux *bugs*, ce qui affecte le gameplay. Accorder aux traducteurs accès à cette version s'avère donc difficile et chronophage car ils peuvent avoir du mal à accéder aux parties problématiques. Cependant, les développeurs peuvent autoriser l'accès à des versions actuelles ou de substitution des menus et des interfaces utilisateur pour réduire les "débordements, les chevauchements et les troncatures". C'est pourquoi nous avons inclus des questions sur les attitudes de test des développeurs et sur l'utilisation de "tests de régression visuelle". Cette méthode de test peut être utilisée pour créer un "mode de prévisualisation" pour les traducteurs sans leur donner accès au moteur de jeu lui-même. Cependant, l'analyse des réponses des développeurs indique qu'ils s'appuient principalement sur la phase d'assurance qualité pour identifier les *bugs*.

7.5 Analyse des outils

7.5.1 Outils d’extraction et intégration des ressources, gestion de contenu et gestion de projet : XLOC.

Le Tableau 20 présente les résultats organisés en fonction au niveau d’étude des traducteurs, la dernière ligne indique le nombre de participants dans chaque groupe.

	Aucun	**Formations spécialisées, séminaires, etc.**	**Licence en traduction**	**Master en traduction**	**Tous (moyenne)**
Régulièrement	10.71 % 15	8.70 % 8	9.86 % 14	10.59 % 25	10.32 % 64
Parfois	10.71 % 15	9.78 % 9	16.90 % 24	13.56 % 32	13.23 % 82
Ont entendu parler, mais n'utilisent pas	68.57 % 96	66.30 % 61	61.27 % 87	69.07 % 163	66.61 % 413
Jamais entendu parler	10.00 % 14	15.22 % 14	11.97 % 17	6.78 % 16	9.84 % 61
TOTAL	140	92	142	236	620

Tableau 20. Degré d'adoption des outils d'extraction et d'intégration des ressources, gestion de contenu et de gestion de projet selon le type de formation

Le Tableau 21 présente les outils et les réponses moyennes qu'ils ont reçues, et la dernière ligne fournit un aperçu du degré d'adoption des outils en fonction du type de contrat. En raison des contraintes d'espace et du faible nombre de participants, les catégories pour les étudiants et les traducteurs bénévoles ne sont pas affichées.

	Indépendant	**Indépendant travaillant seulement pour des agences**	**Indépendant travaillant aussi pour des agences**	**Employé dans une entreprise de traduction**	**Employé dans une entreprise non spécialisée en traduction**
Excel/Word 65.10 % (97)	67.65 % 23	75.00 % 12	66.67 % 22	62.86 % 22	57.69 % 15
XLOC 22.15 % (33)	14.71 % 5	6.25 % 1	24.24 % 8	42.86 % 15	7.69 % 2
[redacted]					
Plunet 30.20 % (45)	38.24 % 13	25.00 % 4	54.55 % 18	20.00 % 7	11.54 % 3
Total : 149 (24.03 %)	34/174 (19.54 %)	16/104 (15.38 %)	33/157 (21.02 %)	35/77 (45.45 %)	26/75 (34.67 %)

Tableau 21. Principaux outils d'extraction et d'intégration d'actifs, de gestion de contenu et de gestion de projet selon le type de contrat.

XLOC

XLOC est un prestataire de services bien établi, en collaboration avec d'importants acteurs de l'industrie tels que Capcom, Activision et Keywords Studios. Cependant, il est difficile d'obtenir des informations techniques sur le fonctionnement de l'outil en raison de ressources limitées. L'outil offre des options d'intégration avec toutes les principales technologies de moteur 3D. En 2020, XLOC a lancé un "connecteur memoQ" pour les outils de traduction assistée par ordinateur, mais il semble n'y avoir qu'un seul résultat incorrect lors de la recherche de "XLOC" sur le site web de memoQ. Le nouveau connecteur fournirait les outils de gestion de terminologie et d'extraction de memoQ pour assurer la cohérence, ses options d'assurance qualité et l'outil de traduction assistée par ordinateur. Ces trois fonctionnalités ont été marquées comme "essentielles" et occupent respectivement les deuxième, quatrième et cinquième positions. Pour éviter les redondances, nous les étudierons dans la sous-section spécifique dédiée à memoQ.

L'analyse croisée des fonctionnalités de XLOC, révèle le manque d'une fonctionnalité spécifique pour visualiser les dialogues dans l'ordre. Des solutions possibles incluent des changements dans les pratiques commerciales et l'organisation des documents Excel de manière chronologique, l'utilisation d'outils à base de hiérarchie pour les dialogues en branches, le prétraitement des documents avant leur téléchargement ou la modification des options d'exportation pour inclure des arbres de dialogues. Cependant, il n'est pas certain que XLOC propose une telle fonctionnalité, car elle n'est pas clairement mentionnée sur leur site web. Selon l'enquête, "l'accès à l'audio, à la vidéo et aux images" a été considéré comme une fonctionnalité essentielle, se classant deuxième parmi les fonctionnalités utiles. XLOC offre un accès direct à ces ressources depuis le moteur de jeu via une API ou le fera prochainement, comme indiqué sur leur site web. Cependant, la disponibilité de ces ressources dépend de la volonté des développeurs ou des responsables de projet de fournir le matériel de référence. Les quatrième et cinquième positions dans la liste des fonctionnalités essentielles et utiles étaient "la possibilité de suivre les modifications dans les fichiers texte source" et "l'accès à tous les actifs dans leur forme originale et divisés par formats". Le principal avantage de XLOC réside dans sa capacité à suivre les modifications, les processus et les builds, garantissant le contrôle. De plus, le système stocke les actifs de jeu pour la localisation, permettant d'y accéder, et prend en charge diverses structures et formats de fichiers de jeu sans nécessiter de conversion ou d'ajustement supplémentaire.

Lorsque l'on considère la section "fonctionnalités manquantes", il apparaît que XLOC, en association avec memoQ, couvre un nombre significatif de ces fonctionnalités. La combinaison offre les options d'assurance qualité de memoQ, le comptage des caractères et répond à la demande d'un moyen de communiquer avec les développeurs et les chefs de projet. Bien qu'il reste incertain si XLOC permet de voir les modifications "en temps réel", leur site web laisse entendre la possibilité de cette fonctionnalité. Dans l'ensemble, l'intégration de XLOC et de memoQ semble prometteuse pour répondre aux besoins de localisation et améliorer la collaboration entre les équipes.

7.5.2 Corpus usage and compilation tools : Sketch Engine

Le Tableau 22 regroupe les résultats pour chaque type de corpus et le pourcentage d'utilisateurs en fonction de leurs études. Il est également nécessaire de mentionner que les répondants ont été autorisés à fournir plusieurs réponses. En ce qui concerne l'utilisation d'outils de compilation de corpus, les résultats filtrés montrent que, bien que le nombre d'utilisateurs soit remarquablement faible, le pourcentage le plus élevé se trouve parmi ceux ayant un diplôme de master, suivi par ceux ayant suivi des cours spécialisés, et en troisième position, les participants sans études universitaires.

	Aucun	Formations spécialisées, séminaires, etc.	Licence en traduction	Master en traduction	Tous (moyenne)
Corpus parallèles bilingues	30.71 % 43	50.00 % 46	50.00 % 71	51.27 % 121	45.81 % 284
Corpus parallèles multilingues	9.29 % 13	14.13 % 13	17.61 % 25	17.37 % 41	15.00 % 93
Corpus comparables monolingues	10.00 % 14	27.17 % 25	18.31 % 26	17.37 % 41	17.74 % 110
Corpus comparables bilingues	18.57 % 26	22.83 % 21	23.24 % 33	23.73 % 56	22.42 % 139
Corpus comparables multilingues	6.43 % 9	9.78 % 9	10.56 % 15	10.17 % 24	9.35 % 58
Aucun	57.14 % 80	33.70 % 31	30.28 % 43	32.63 % 77	37.90 % 235
TOTAL	140	92	142	236	620

Tableau 22. Utilisation des corpus selon le type d'éducation

Des études dans le domaine de la traduction en général menés par Zaretskaya et al. (Zaretskaya et al. 2015 : 250; 2017 : 46), ont constaté que seulement 17 % de ceux qui travaillaient avec des corpus utilisaient également des outils de compilation. Dans la présente étude, le pourcentage total d'utilisateurs d'outils de compilation de corpus est de 8,39 %. Cependant, une fois que nous éliminons les répondants qui ont déclaré ne pas utiliser de corpus du tout (pour une meilleure comparaison), notre résultat atteint 12,44 %. De plus, les auteurs ont étudié l'impact du niveau d'éducation sur le degré d'adoption des outils et leurs résultats se sont également avérés plus élevés que ceux obtenus dans la présente étude : 66,7 % pour les répondants sans aucune étude formelle en traduction, 30,4 % pour ceux qui avaient suivi des cours spécialisés, 13,3 % avaient une licence et 28 % avaient une maîtrise (Zaretskaya et al. 2017 : 48). Dans notre cas, après avoir modifié les calculs en fonction des utilisateurs de corpus, 13,33 % n'avaient pas d'études formelles en traduction, 11,48 % avaient suivi des cours spécialisés, 8,00 % avaient une licence et 15,72 % avaient une maîtrise.

Parmi ceux qui compilent leurs propres corpus, Sketch Engine a reçu le plus grand nombre de réponses, après l'option "autre". Sketch Engine est un outil en ligne de compilation de corpus offrant diverses façons de créer un corpus, notamment en téléchargeant des documents, en sélectionnant des sites Web et en utilisant le web crawling. Une fois le corpus créé, les utilisateurs peuvent l'analyser, extraire des termes et comprendre leur utilisation contextuelle ou comparer les traductions de mots dans plusieurs langues et documents. En ce qui concerne l'intégration, Sketch Engine dispose d'un plug-in dédié à SDL Trados Studio, mais il repose principalement sur l'exportation de fichiers dans des formats compatibles avec les outils TAO pour une utilisation directe.

7.5.4 Outils d'extraction et de gestion de terminologie : SDL MultiTerm

Dans l'article publié par Zaretskaya et al. (Zaretskaya et al. 2017 : 46), les auteurs ont divisé ces outils en outils de gestion de terminologie et en outils d'extraction de terminologie. Cependant, les résultats de la présente étude semblent être plus proches de ceux obtenus pour les "outils d'extraction de terminologie". Les résultats combinés des utilisateurs réguliers et occasionnels d'outils d'extraction de terminologie obtenus par les auteurs étaient de 25 % (et de 58 % pour les outils de gestion de terminologie), tandis que les résultats de notre étude montrent 22,74 % (Zaretskaya et al. 2017 : 46). Les résultats individuels pour chaque type de formation obtenus par Zaretskaya et al. (Zaretskaya et al. 2017 : 48) étaient les suivants : 23,2 % pour "aucun", 28,8 % pour les cours spécialisés, 26,8 % pour les détenteurs d'une licence et 25,7 % pour les détenteurs d'une maîtrise. Notre étude a obtenu un résultat légèrement plus élevé pour les répondants ayant une maîtrise (25,85 %) et environ 1 % de moins pour les détenteurs d'une licence (25,35 %) et ceux sans études officielles en traduction (22,14 %). La seule divergence claire concerne le pourcentage nettement inférieur de participants ayant suivi des cours spécialisés et utilisé ce type d'outil (11,96 %). Le Tableau 23 présente les résultats concernant les outils classés par type de contrat.

	Indépendant	**Indépendant travaillant seulement pour des agences**	**Indépendant travaillant aussi pour des agences**	**Employé dans une entreprise de traduction**	**Employé dans une entreprise non spécialisée en traduction**
Aucun 63.39 % (393)	63.22 % 110	63.46 % 66	57.96 % 91	66.23 % 51	76.00 % 57
SDL MultiTerm 27.42 % (170)	28.74 % 50	25.96 % 27	35.03 % 55	27.27 % 21	8.00 % 6
Total	174	104	157	77	75

Tableau 23. Outils d'extraction et de gestion de terminologie par type de contrat

SDL MultiTerm est un produit de RWS, une entreprise connue pour être propriétaire de Trados Studio et spécialisée dans les services de gestion linguistique, de contenu et de propriété intellectuelle. Bien qu'il puisse être utilisé comme un programme indépendant, il est couramment utilisé en association avec Trados Studio. L'outil offre un environnement centralisé pour la gestion de la terminologie, facilitant ainsi l'accès aux équipes de traduction. Il prend en charge divers formats de fichiers, permet une modification facile des entrées et inclut une fonction qui permet aux utilisateurs de rechercher la signification des termes à partir de n'importe quelle application.

7.5.5 Outils de traduction assistée par ordinateur : memoQ

Les résultats obtenus par Zaretskaya et al. (Zaretskaya et al. 2017 : 46) ont révélé qu'à l'époque, 76 % des participants utilisaient ce type de technologie, tandis que trois ans plus tard, dans le domaine de la localisation de jeux vidéo, le pourcentage d'utilisateurs passe à 92,58 %. Lorsque les auteurs ont classé leurs résultats par type de formation, le pourcentage le plus bas d'utilisateurs a été trouvé parmi les traducteurs sans aucune formation en traduction (66,9 %), suivi de ceux qui avaient suivi des cours spécialisés (78,4 %), et ceux avec une licence en traduction (78,1 %). Les participants titulaires d'une maîtrise ont obtenu le meilleur résultat avec 86,7 % (Zaretskaya et al. 2017 : 48). Dans le cas de la présente étude, presque tous les pourcentages sont supérieurs à ces résultats et le plus bas appartient aux traducteurs qui avaient suivi uniquement des cours spécialisés (84,78 %). Les participants qui n'avaient aucune certification en traduction ont obtenu 85,71 %, ceux avec une licence en traduction ont obtenu 95,79 % et ceux avec une maîtrise se sont classés en tête avec 97,46 %. Nous pouvons donc conclure que les outils de traduction assistée par ordinateur sont de plus en plus populaires en raison de leur inclusion systématique dans les programmes universitaires ainsi que de la demande croissante tant des clients que des prestataires de services linguistiques. Le Tableau 24 analyse les différences dans l'utilisation des outils de traduction assistée par ordinateur en fonction du type de contrat des participants.

	Indépendant	**Indépendant travaillant seulement pour des agences**	**Indépendant travaillant aussi pour des agences**	**Employé dans une entreprise de traduction**	**Employé dans une entreprise non spécialisée en traduction**
MemoQ 86.41 % (496)	86.08 % 136	88.12 % 89	93.51 % 144	84.72 % 61	70.97 % 44
Trados 61.32 % (352)	60.76 % 96	69.31 % 70	70.13 % 108	56.94 % 41	24.19 % 15
Autre 48.61 % (279)	51.27 % 81	42.57 % 43	49.35 % 76	41.67 % 30	69.35 % 43
Passolo 28.05 % (161)	26.58 % 42	34.65 % 35	25.97 % 40	30.56 % 22	12.90 % 8
Total : 574 (92,58 %)	158/174 (90,8 %)	101/104 (97,12 %)	154/157 (98,09 %)	72/77 (93,51 %)	62/75 (82,67 %)

Tableau 24. Outils TAO par type de contrat

MemoQ offre une solution complète et a gagné en popularité auprès des traducteurs ces dernières années. Suivant la tendance du marché, memoQ propose un nombre croissant d'extensions et de connexions avec d'autres systèmes pour réduire autant que possible le besoin d'options autonomes. La société a également publié un livre électronique sur la localisation de jeux vidéo (*LocLand : the Land of Game Localization*[88]) afin de fournir aux développeurs de jeux vidéo des informations sur le processus de localisation ainsi que pour présenter comment exploiter les fonctionnalités de memoQ. L'analyse croisée des fonctionnalités et les résultats des questions clés révèle que, pour la première fonction essentielle, "la possibilité de voir les dialogues dans l'ordre", memoQ ne dispose pas d'une solution spécifique. La deuxième fonction essentielle, "gestion et extraction terminologique", est prise en charge par l'extension Qterm de memoQ, qui garanti aussi des capacités d'accès à des fichiers audio, vidéo et aux images. Cependant, la disponibilité de ces ressources dépend de la collaboration du client dans le processus de localisation. En ce qui concerne les quatrième et cinquième fonctionnalités, les capacités d'intégration de memoQ avec les systèmes de gestion de contenu tels que XLOC facilitent le suivi des modifications apportées aux ressources et leur accès si les traducteurs ont accès à l'outil.

Les traducteurs ont exprimé le besoin d'un outil pour compter les caractères, ce qui n'est pas directement disponible dans memoQ comme c'est le cas dans Excel. Cependant, le système propose l'affichage du nombre de caractères dans la barre d'état et la configuration de notifications d'erreur lorsque les traductions dépassent des limites spécifiées. Des options avancées telles que "la vérification de la longueur basée sur les pixels" assurent des calculs précis de l'espace. Le service de messagerie instantanée de memoQ facilite la communication au sein de l'équipe de traduction, bien qu'il soit limité à l'équipe de traduction et au gestionnaire de projet. Les options d'affichage réduisent les allers-retours entre memoQ et Excel pour les commentaires et le connecteur de contenu surveille de près les mises à jour du jeu.

7.5.6 Outils de traduction automatique : Google Translate

Dans l'ensemble, nos résultats montrent une augmentation progressive de l'utilisation de la traduction automatique par rapport aux pourcentages obtenus par les chercheurs précédents. Les résultats de l'enquête QTLaunchPad (Doherty et al., 2013, p. 9) ont montré qu'à l'époque, seuls 34 % des 500 répondants utilisaient la traduction automatique. L'étude de Zaretskaya (Zaretskaya, 2015, p. 4) a montré une légère augmentation au moment de l'enquête, avec un total de 36 % des 736 participants. Nos résultats montrent que 48,71 % des 620 traducteurs ayant participé à la première enquête étaient des utilisateurs réguliers ou occasionnels de la traduction automatique. En ce qui concerne les différences dans le degré d'adoption en fonction du parcours éducatif des participants, Zaretskaya et al. (2017, p. 48) ont trouvé le pourcentage le plus élevé d'utilisateurs parmi les participants ayant suivi des cours spécialisés (51,9 %), suivis de ceux ayant un baccalauréat (46,5 %), puis les participants ayant une maîtrise (46,6 %) et enfin ceux sans études officielles en traduction (43,1 %). Cependant, la première enquête a montré que dans le cas des traducteurs de jeux vidéo, les participants ayant un baccalauréat en traduction sont en tête avec 55,64 %, suivis de ceux ayant une

[88] https ://www.memoq.com/resources/ebooks/locland-game-localization

maîtrise (49,55 %), puis ceux ayant suivi des cours spécialisés (47,83 %) et enfin ceux sans aucune étude officielle en traduction avec 42,14 %.

Lorsque nous analysons le degré d'adoption des systèmes de traduction automatique en fonction du type de contrat, nous pouvons observer que le pourcentage le plus élevé d'utilisateurs appartient aux travailleurs indépendants (58,62 %), suivi de ceux travaillant dans une agence de traduction (50,65 %), puis des indépendants travaillant exclusivement avec des agences (49,04 %), des travailleurs indépendants travaillant à la fois avec des agences et de manière indépendante (43,31 %) et ceux travaillant pour une société de développement de jeux vidéo (34,67 %). En ce qui concerne chaque outil, nous pouvons constater que les indépendants ont tendance à éviter les systèmes payants et ont choisi "aucun" en premier lieu, tandis que ceux embauchés ont sélectionné l'API de Google Translate en premier. Quant aux outils gratuits, Google Translate arrive en tête de liste pour chaque groupe avec des résultats relativement similaires. Google Translate propose une variété différente de fonctionnalités selon le tarif et toutes les solutions sont basées sur la traduction automatique neuronale, sauf si la combinaison de langues n'est pas prise en charge. En ce qui concerne les capacités d'intégration, Google Translate fonctionne à l'aide de clés API et, une fois la clé créée, elle peut être utilisée pour "pousser" directement le texte dans Google Translate, puis récupérer leur traduction. Ce système est utilisé par de nombreux outils TAO, notamment memoQ. De plus, il existe de nombreux outils disponibles dans Unity qui peuvent se connecter à Google Translate afin de traduire automatiquement le contenu du jeu.

7.5.7 Outils de test : Jira

Jira est un outil développé par Atlassian, la même société qui possède Trello et Redmine, ainsi que divers autres outils axés sur le travail d'équipe dans le domaine du développement de logiciels. Sur son site web, Jira est décrit comme un outil de développement logiciel utilisé dans des processus agiles, axé sur le suivi de projets et de problèmes. Lors de nos recherches sur la manière dont Jira peut être utilisé spécifiquement dans le développement de jeux vidéo, nous avons trouvé un plug-in pour Unity appelé "Easy Jira", disponible sur la boutique d'actifs Unity et créé par un petit studio de développeurs indépendants appelé "Unreal Byte Games" et qui permet aux utilisateurs de créer et d'assigner des problèmes, ainsi que d'ajouter des commentaires à ces problèmes.

7.5.8 Outils de développement de jeux vidéo : Unity

Unity offre un environnement complet qui permet aux utilisateurs de créer un jeu vidéo avec très peu besoin d'autres programmes en dehors de son interface, ce qui élimine les problèmes de compatibilité. Comme nous l'avons déjà vu, Unity possède des capacités d'intégration avec presque tous les systèmes mentionnés. Cela signifie que le système pourrait couvrir toutes les exigences qui ont été analysés dans les sections précédentes via des plug-ins ou des API. Unity propose également plusieurs outils basés sur des arbres à dialogues dans sa boutique d'actifs qui peuvent être achetés et ajoutés au système. Parmi les possibilités, la plus simple serait d'utiliser ledit outil pour créer la conversation, puis de prendre simplement une capture d'écran de l'arbre résultant et l'inclure en tant que matériel de référence.

Lorsque nous analysons les possibilités d'accès à l'environnement visuel réel du jeu, Unity offre plusieurs options. Tout d'abord, son interface utilisateur de projet fournit une fenêtre et une option de prévisualisation qui permet d'afficher la scène telle qu'elle apparaîtrait dans le jeu lui-même. Cependant, cela ne peut être utilisé que si les traducteurs ont accès à Unity une fois que les textes traduits ont été intégrés dans le système. Deuxièmement, l'une des options proposées par Unity est la possibilité de choisir les scènes avant de compiler une version du jeu (build), ce qui permet aux utilisateurs de prévisualiser la partie spécifique du jeu en question sans avoir à parcourir chaque scénario ou niveau. Cela implique également la possibilité de créer des mini-builds qui ne contiennent que des menus ou des inventaires afin d'éviter efficacement les débordements, les chevauchements et les coupures.

7.6 Découvertes supplémentaires

Gridly a été lancé sur le marché fin 2020 et n'a cessé d'ajouter de nouvelles fonctionnalités et de publier des mises à jour. La société a développé deux plug-ins pour Memsource et memoQ, ainsi que deux autres pour Unity et Unreal Engine. Le système dispose d'une variété de fonctionnalités pour automatiser le processus et la connexion directe entre Gridly, le moteur de jeu et les outils de TAO réduit considérablement la nécessité de travail manuel et de traitement. Gridly ne semble pas avoir de fonctionnalités spécifiques pour fournir "la possibilité de voir les dialogues dans l'ordre", en dehors de celles déjà présentées pour XLOC. Cependant, Gridly couvre les fonctionnalités résultant de la combinaison des éléments "essentiels" et "utiles" via le CMS lui-même ou des add-ons, à savoir "la gestion des terminologique", "l'accès à l'audio, la vidéo et les images", "l'inclusion d'outils d'assurance qualité", "les outils de traduction assistée par ordinateur", et "la possibilité de suivre tous les changements dans les fichiers sources".

La fonctionnalité la plus demandée, la possibilité de communiquer avec d'autres membres de l'équipe, a été traitée grâce à la possibilité de créer des conversations épinglées à la cellule en question, à l'intégration de Gridly avec Slack et aux fonctionnalités propres de memoQ. Toutes le demandes les plus courantes dans la catégorie "plus d'accès visuel" sont possibles soit directement dans Gridly, soit en combinaison avec memoQ : pouvoir voire les colonnes pour les autres langues cibles, être en mesure d'exporter les commentaires supplémentaires des développeurs ou les ID dans l'outil TAO, et avoir le matériel de référence à côté du texte. Il en va de même pour la disponibilité des outils d'assurance qualité, ainsi que pour presque tous ceux inclus dans la catégorie "divers" (Annexe 5). Lors des webinaires et d'une réunion privée avec memoQ, ils ont démontré des modifications de localisation en temps réel sur un site web pour mettre en valeur leurs capacités d'intégration. Nous avons demandé si cela pouvait être réalisé aussi dans les jeux vidéo, et ils ont expliqué que ça serait possible grâce au "contrôle de branche", en créant une branche alternative à des fins de test, en intégrant le contenu dans une "version de test" et en l'affichant sans affecter la branche principale. Cependant, aucun exemple spécifique n'a été montré pendant la démonstration privée. Enfin, en ce qui concerne les limitations de caractères, en plus de toutes les fonctionnalités fournies par Memsource et memoQ, ils ont expliqué qu'ils travaillaient actuellement sur une solution pour automatiser le processus.

8. Conclusions et perspectives futures de travail

La présente dissertation présente des limites et ouvre des perspectives futures de recherche. En raison de sa portée, certains maillons de la chaîne de localisation de jeux vidéo n'ont pas pu être pleinement examinés, rendant nécessaire des études complémentaires pour mieux comprendre leur rôle et leur impact dans le processus global. De plus, la nature confidentielle des accords de non-divulgation et l'utilisation d'outils propriétaires dans l'industrie ont restreint notre accès à certaines informations, notamment concernant des outils tels que XLOC. Pour enrichir nos conclusions, il serait opportun de réaliser une étude pilote couvrant tout le processus de développement à la production d'un jeu, avec un suivi des participants pour évaluer les avantages de la combinaison d'outils proposée. Ces futures recherches permettront de mieux appréhender les dynamiques de la localisation de jeux vidéo et d'explorer de nouvelles opportunités pour optimiser ce processus complexe.

Un défi supplémentaire auquel nous avons été confrontés dans cette recherche est la constante de l'évolution de ce domaine, qui entraîne l'émergence continue de nouveaux systèmes sur le marché. Parmi ces systèmes émergents, Crowdin se distingue particulièrement avec son récent plug-in pour Unity et sa promotion en tant que système de gestion de localisation spécialisé dans le développement agile de jeux vidéo, suivant ainsi la tendance croissante de la localisation continue. Cette évolution du marché vers la localisation continue est en partie alimentée par l'essor du modèle GaaS (Game as a Service). Cependant, étant donné le caractère relativement récent de ce modèle, il existe encore peu de recherches sur son impact dans le domaine de la localisation de jeux vidéo ou sur les pratiques de localisation associées. Il serait donc essentiel d'explorer davantage cette pratique et d'étudier ses effets sur tous les acteurs impliqués pour mieux comprendre comment la localisation peut s'adapter à cette nouvelle approche.

Un autre aspect important à considérer est l'utilisation croissante de la traduction automatique neuronale (TAN) par les développeurs de jeux individuels et les petits studios autofinancés, qui sont souvent attirés par les perspectives prometteuses de cette technologie. Avec la perception généralement optimiste du public concernant les capacités de la TAN, il devient essentiel d'examiner comment l’utilisation de la traduction automatique brute peut influencer les processus de localisation et la qualité des jeux. En effet, l'emploi de la TAN sans intervention humaine peut entraîner des erreurs de traduction et des incohérences qui pourraient nuire à l'expérience utilisateur et à la réputation des jeux. Par conséquent, des recherches futures doivent se concentrer sur l'évaluation objective de l'efficacité et des limites de la TAN dans le contexte spécifique de la localisation de jeux vidéo.