4 The Translation of Video Games

This chapter contains three different sections dealing with TMIES, i.e. the translation of the linguistic contents included within most video games, the unique characteristics of MIES texts and the playfulness natural to games that has a direct impact on the translator’s task as well as the business constraints around the whole process. In the planning stages of a new project, game designers already have a clear idea of the kind of experience they want to provide for players, which in turn allows them to create a product with a particular target audience in mind, normally related to a particular country of residence. Even when teams are multinational, which is increasingly the case, a single approach must prevail in an attempt to focus resolutely on the creation of the imaginative worlds, storylines, characters and features, which result in the ‘perfect’ game.

To date, the academic literature written on this subject remains limited, and in many cases articles are superficial or repeat what has already been written. Fortunately, there have been several conferences to remedy the situation, such as the Localization Summit within IGDA, which is central to the video game industry and the Game Localization Roundtable within LW the leading event in the software localisation industry. These have enabled professionals to come together and share their ideas on the best practices for game localisation.

Due to the high costs involved in the development of Triple A titles, a budget “in excess of £20 million” (Wilson 2011: online), and the competitive nature of today’s global market, it has become almost mandatory to release all new games in as many countries as possible in order to cover the investment and to maximise profits. This approach calls for a full I18n (internationalisation) of the product, signifying that attempts must be made to incorporate within the game design itself all the potential changes importing countries might require. This customisation, which is made in order to accommodate different national tastes and preferences, is normally referred to as ‘localisation’ (Maxwell-Chandler and O’Malley-Deming 2012). It aims to cover the costs of production as well as to generate further profits within the highly competitive MIES market. From a technical point of view, it includes, among other things, the implementation of Unicode to guarantee...
that all the characters in all writing systems are properly displayed; the adjustment to different hardware requirements such as the analogue television signals PAL and NTSC, and more recently the wide screen high definition (HD) upgraded to 1,980 × 1080 pixels with, in addition, a multi-network configuration to enable lag-free, simultaneous online playability in all corners of the world. If they wish to maximise distribution potential, in addition to the obvious technical adaptations, game publishers also need to look into the legal framework, cultural preferences and age ratings current in each country in which they intend to commercialise the game.

The linguistic and cultural aspects of this customisation are not issues in which the industry has been very interested until rather recently, particularly in the US, the UK and Canada, where only a knowledge of English is needed to reach their respective domestic markets. Officially, bilingual countries (Switzerland, Belgium, Canada) combine their language offer based on disc space (DVD or Blu-ray Disc) and regional preferences so, for example, Quebec requires French if it is available, but the rest of Canada sells the US English version. If there are no disc storage issues because it is a small game or a high capacity Blu-ray Disc, several languages will be included in the disc and the gaming system will default to the predefined language. In these cases, the text for the box will be reduced to allow for the two or three languages officially spoken in countries such as Canada or Switzerland.

English has also been the dominant language used for distribution in Japan as well as for non-English game developers, because it remains the lingua franca of international commerce and mass entertainment, making it the most profitable market option from the language point of view if only one version is to be released. Indeed, most game publishers show some detachment from localisation issues and tend to outsource this part of the process to specialised companies, while retaining the control and final vetting rights. However, this detached attitude with regards to the translation and localisation of video games is gradually changing due to the fact that sales in foreign markets are growing relatively fast, and can sometimes represent more than half of worldwide sales.

In 2007, PricewaterhouseCoopers (www.pwc.com) were already forecasting sales of $50 billion by 2011, but the revolutionary changes that have taken place in online, social and mobile gaming meant that by the end of 2011, the market had in fact reached over $180 billion (Merel 2011: online). While in the US, there was a decrease in the rate of growth in the game market during the first decade of the twenty-first century due to the virtual saturation of its domestic market; in most other countries around the world consumption nearly trebled, turning localisation into a justifiable and profitable investment.

Another factor contributing to the change in attitudes towards translation is connected with adverse public opinion, and the consequent game brand and company image damage caused by products that lack the necessary localisation awareness and fail to accommodate the sensibilities of other
cultures. Some prime examples directly concern two of the biggest companies operating within the sector. In the first case, Sony/PlayStation was seen as having been highly irresponsible for including Manchester Cathedral as one of the battle grounds in its first-person shooter game Resistance: Fall of Man (BBC 2007). Microsoft, the other video game giant, lost most of the money spent on Kakuto Chojin after having to recall all its stocks because, in order to create an atmospheric soundscape, a fragment of the traditional Muslim call to prayer was included. It was considered blasphemous by members of Muslim communities, not only in Arab countries but all around the world, prompting the recall of the game (P. Brown 2004). Another interesting localisation setback involved Sony/PlayStation and concerned the music used in LittleBigPlanet (Miller 2008). On this occasion, Sony/PlayStation acted very early in the process, deciding to postpone the official launch of the game so that the musical theme could be deleted. It was considered to be disrespectful to the Muslim faith because, despite being a modern Arab song, it contained some verses from the Qur’an (these issues will be analysed and illustrated further in Section 5.3).

The game industry is currently highly technical and design oriented and is primarily controlled by mass-market forces and profitability. This particular focus means that linguistics, translation and even communication are rarely taken into consideration. As a result, and perhaps unwittingly, game development and publishing companies seem to be wasting both the time and the money spent on the translation of their products for foreign markets due to an essential lack of understanding of what the process entails and the bad planning that ensues. This in turn affects language professionals because they are forced to work under unnecessarily extreme time constraints, to unrealistic deadlines. It is worth noting, however, that the situation is gradually improving and that discussions concerning cultural and language topics are being included at some of the international conferences held by the game industry, such as the ‘Localization Summit’, which is part of the annual Game Developers Conference taking place in San Francisco (www.gdconf.com). Highlighting the importance of the cultural and linguistic dimensions can benefit the game and the localisation industries, as well as being of interest within the academic field of Translation Studies. This lack of co-operation between the two is one of the existing gaps that it is the intention of the present research to fill, in an attempt to open the possibility of a fruitful collaboration between all stakeholders, based on a cross-referenced knowledge.

4.1 ARTEFACTS AND PRODUCTS

A video game is both a product designed for mass consumption and an artistic creation at the same time. In this respect, and due to the youth of the industry itself and its link to computer technology, video games can be perceived as being somewhat different from more established and respected
art forms such as sculptures, paintings, novels or films, even though all of these may be considered as mass-media products belonging to culture (Gambier and Gottlieb 2001a; Peeren 2008; Ricardo 2009). The perception of the video games industry as an outsider or a newcomer to arts plays a key role when analysing the games and their translation for the different international markets (see Section 5.3). This is particularly so because of the importance that linguistic localisation has in the playability of the game and, consequently, in the penetration and sales of the versions being commercialised in different countries or locales.

Before discussing internationalisation in depth, it may be useful firstly to differentiate between the concepts of ‘work of art’ and ‘product’ in order to better understand the nature of video games and their translation. According to the Merriam-Webster dictionary (www.merriam-webster.com), a work of art is “Something giving high aesthetic satisfaction to the viewer or listener” often associated with specific historical periods. In this sense, it could be argued that works of art and products occupy the two extremes of the same scale. In the epistemological sense, art can be considered to represent a profession since it is obviously capable of generating money, but the creation and its value are linked to the artist in every case. Copies will never be worth the same as an original so that an original is always regarded as constituting a work of art per se. A painting by Goya, a sculpture by Michelangelo, an opus by Bach, or a novel by James Joyce, as works of art, remain unique throughout their existence, even if copies and variations of them can be made. They need no justification other than the pleasure of creating them, and the author remains largely independent, despite the influence of ‘Maecenas’ if, indeed, there is one.

In contrast, products are normally understood as “something (as a service) that is marketed or sold as a commodity”, as constituting merchandise for mass consumption and, as commodities for the wider population to buy and enjoy, they are generated in their thousands and all share the same commercial value and qualities, however many copies are sold. Indeed, the number of copies sold adds value to these creations because their popularity is understood as an indicator of their quality. Because they are the result of industrialisation and chain production processes, products are, in fact, merchandise. This is not to say that there is no artistic value or skill involved in producing them but, as a result of the industrial process that allows for mass production, these items are universally available. In the words of Benjamin (1936: online):

that which withers in the age of mechanical reproduction is the aura of the work of art. This is a symptomatic process whose significance points beyond the realm of art. One might generalize by saying: the technique of reproduction detaches the reproduced object from the domain of tradition. By making many reproductions it substitutes a plurality of copies for a unique existence.
Companies may try to highlight uniqueness in their products with labels such as ‘first copy’, ‘special edition’, or ‘autographed edition’, but it can easily be argued that these are somewhat gimmicky sales strategies. Modifications to actual works of arts are normally considered illicit, and the only option is reinterpretation by another artist. In these cases, a new work of art is created, inevitably linked to a new artist. A clear example of this approach is *Las Meninas [Maids of Honour]* painted by Velázquez in 1619 and the interpretation by Picasso created in 1957.

The difference between artefacts and products can, however, be one of perception more than an objective one, for it is the eye of the beholder, the one that assigns value, often in agreement with society and peer groups but also in daring discord. The genius of many individuals have only been recognised posthumously, such as Van Gogh, Kafka, Bach, el Greco, Poe, Goddard, Vermeer, Mendel, Thoreau to name but a few. Their work is greatly valued today socially but, at the same time, many would be more likely to identify and pay high prices for a puppet used in the first *Star Wars* film, a Kojima’s autographed edition of *Metal Gear Solid*, or the first Barbie doll ever created. The artefact and the product, the outstanding and the average are in the eye of the beholder.

The mechanical copying or reproduction of works of art renders them consumable products that can be cloned and multiplied as many times as desired, but none of those copies will ever have the same value as the original. Yet, if we look at this issue from the product point of view, its value resides in the fact that it can be manufactured in millions, fostering an economy of scale whilst keeping the same quality in all the copies. Mass production is one of the main pillars of modern society and, although artistry is still part of it, it is usually only of importance at the beginning of an industrial process—the designing stage—which also requires the intellect and skills of other craftsmen in order to fuel the mass production, adding to the cost-effective nature of the product.

The actual creation and design of video games can be related to the ‘art’ dimension described above, as is borne out by the fact that they have been recognised since 2006 by BAFTA. This is because video games require the same talent for storytelling found in literature, the cinematic realisation of filmmakers and the craftsmanship involved in the manufacture of toys. Owing to a decision made by game developers and publishers, translation and localisation have, until recently, been placed firmly in the post-production stage of the process, together with packaging. This is because there seems to be no doubt that making the product usable by speakers of other languages is purely practical (Bartelt-Krantz 2011). In this approach, however, translation seems to involve the simplistic task of the word-pairing typical of glossaries, valid perhaps for the translation of basic menus, but totally unsuitable for the translation of narration and dialogue in video games, in which a similar level of creativity is required as for literature or films.

Picasso’s version of *Las Meninas* is, in fact, an excellent illustration of the process of translation, represented in graphic terms. The re-processing of
artistic creations cannot produce sameness, and when it does it is considered to be plagiarism, because it is directly linked to the cultural and personal interpretation of the mediator, in this particular case, the translator, who works in a given time and within a given culture and society. Benjamin (1923: 80) analyses this idea, stating that:

fidelity in the translation of individual words can almost never fully reproduce the meaning they have in the original. For sense in its poetic significance is not limited to meaning but derives from the connotations conveyed by the word chosen to express it. We say of words that they have emotional connotations. A literal rendering of the syntax completely demolishes the theory of reproduction of meaning and is a direct threat to comprehensibility.

Yet somehow ‘sameness’ is what is often expected of translation, whether the original text is of an artistic or merely pragmatic nature. Seen from this perspective, it can be argued that the only justification for the commercialisation of a product like a car or a video game is to generate revenue by appealing to buyers, so that, in order to cater for the specific likes and needs of a different country, substantial modifications to the actual design and functionality of the product, are commonplace, even expected. One of the reasons for this attitude may be found in the fact that commodities do not have an author in the traditional sense, but rather rely on shared-authorship because the only relevant matter is the product itself and its appeal to buyers.

However, the works of art/product and the single/shared authorship dichotomies are not, in themselves, enough to differentiate entertainment software from other audiovisual entertainment products, such as television series or quiz shows. The third characteristic that sets video games apart is an emphasis on a customisable experience as opposed to the unchangeable one of books and films: the tailoring of the game product to what is desired by a given player. It is not only that each player has a different perception and experience of the game, as with a book or a film, but that the game offers a self-adaptable, virtual world that responds to the actions of the players, adapting to their decisions. For example, during a first play through, the player might decide to enter a locked room by talking to the guard in front of it; in the second play through, by killing the guard and picking the lock and, in the third, by casting a spell on the guard to make him sleep and using the key found in his secret pocket. The fact that interactivity is part of the very essence of video games means that a basic game design can also accommodate technical, cultural and linguistic transformation in order to bring the product closer to the local taste and expectations of the territory of release. According to this principle, almost everything in a video game is open to change in order to meet the needs of specific territories if a potential increase in sales is at stake. In other words,
games may be played differently and look (slightly) different depending on the country where they are purchased. Amateur game localisation and romhacking practices (Muñoz-Sánchez 2008) have not yet had a direct impact on commercial translations of video games, but they do convey a clear message to publishers to the effect that gamers care enough about the translation process and think that playability can be improved in their own languages if more care is taken. These issues are explored further in the following sections, but the size and power of player communities, that is to say the buyers of the product, must be taken into account in order to understand why the translation of these complex, multitextual products deserves special attention from the industry itself, as well as from academics.

4.2 THE MULTITEXTUAL REALITY OF A COMPLEX PRODUCT

The quantity and variety of the translatable assets generated by each video game may come as a surprise to people who are unacquainted with these products. Whether in combination with the cinema or the book industries, or on their own, many video games will require the translation of thousands, or even hundreds of thousands of words, including, for example, manuals, game dialogue and technical and legal documentation. The workload for translation agencies increases exponentially depending on the number of languages into which a game will be translated, as well as on the number of platforms for which it is developed. They can be broadly grouped into two categories: desktop devices (PCs, PS3, PS4, Xbox 360, Xbox One, Nintendo Wii, Wii U), and portable devices (PlayStation Portable, Nintendo 3DS, mobile phones, smart phones, and tablet computers). Since they all have different hardware and software specifications, programming and localisation testing needs to be carried out in each of them. The linguistic and cultural translation of a game can be a highly creative undertaking that adds to the complexities of merely functional and technical adaptations (Fernández Costales 2012). In addition, translators in this field are usually asked to expand their horizons and act as expert terminologists and copywriters when dealing with the different brand-specific glossaries concerning the trademarked, copyrighted and legal texts of the project.

Linguistic assets will be utilised in a variety of ways at different times throughout the creation, development and launch of the game. They can be found in different file formats, which makes working with them challenging for translators as they need to own and familiarise themselves with many different software programs, and sometimes they even have to visit the premises because the right software is only available in-house (see Section 5.9). The following paragraphs concern details of the most common
linguistic assets to accompany a video game, whether in the box itself or in the associated web services and materials:

- The game itself, which is made up of a variety of texts in need of translation, encoded in a variety of formats depending on what the text is going to be used for. These include packaging and manual documentation (usually produced with software applications such as Word or Page-maker); installer programs (designed with proprietary tools in XML or binary code); ‘ReadMe’ files and end user agreements (encoded in the standard text file .txt format); user interface and pop-up help captions (often designed with proprietary tools in XML or binary code); audio files (encoded in .wav, .mp3 or a unique proprietary format) and video files (encoded in .avi, .mov, .mp4, or with more restricted proprietary tools).
- The official website of the game, which will normally use HTML or Java Script. Many video games websites use content management programs, which may be productive tools when regular updates are necessary.
- Promotional articles and merchandising in general, which can be distributed in analogue, electronic or paper formats, such as television commercials, interactive banners, and game magazines.
- Game patches: these small downloadable programs fix existing bugs missed in the testing process when, owing to time constraints, they could not be fixed for the gold release date.
- Game updates: periodical downloadable modular augmentation of game chapters, map-packs, features, storylines, and characters.

With this volume of texts appearing in such a considerable variety of components, workflows need to be planned carefully and well in advance so that nothing is forgotten in the translation process, and consistency in terminology and style are maintained throughout the entire video game. Streamlined workflows become even more important when the information needs to be released simultaneously in more than a dozen languages in different countries around the world (see Chapter 5).

The following paragraphs include a comprehensive catalogue concerning the translatable assets in video games. These can be found accompanying the release of most video games, and all of them have their own textual characteristics and communicative purpose. These texts are created separately and stored in different areas of the game code and in different formats, depending not only on their purpose and on how they are meant to be used by the player but also on how they are integrated into the computer application: the virtual machine that enables the interactivity enjoyed by the users. Although there is no official list of localisation assets—indeed the relative youth of the game industry has meant that it does not even have a standardised list of game assets at all (Carter 2004)—localisation producers and managers (Maxwell-Chandler and O’Malley-Deming 2012: 267) typically seem to organise the translatable files they deal with into
five game asset categories: (1) text, including all the text displayed in-game such as narration, tutorials, installer strings, help files, and error messages; (2) voice-over and cinematics; (3) art, including game logo and in-game language embedded textures; (4) localisation, which refers to branding and technical glossaries to maintain the company image and products across language versions and (5) box and docs, which refers to the packaging and manual.

The translation of all these assets, an important part of the entire localisation process, although not the only one, requires a translator who is able to deal with both the rigors of terminology and the fresh and subtle literary nuances of in-game dialogue (Bernal-Merino 2008c). As mentioned earlier, there are many texts that accompany a game, even if buyers do not make full use of them or are not even aware of their relevance and how they are connected with the game at all. The concept of text types discussed by Sager (1997: 30), and based upon a text typology proposal put forward by Reiss (1981: 126), has proved to be particularly useful in this debate, especially owing to the fact that the author draws attention to the fact that text variety is not confined to one particular language or culture. Of additional significance is her emphasis on the importance of this knowledge to translators, so that they are able to ensure the functional equivalence of the texts in different languages:

I meanwhile define text variety as super-individual acts of speech or writing, which are linked to recurrent actions of communications and in which particular patterns of language and structure have developed because of their recurrence in similar communicative constellations. The phenomenon of text variety is not confined to one language. The various kinds of text variety are partly not confined to one language or one culture, but the habits of textualization, the patterns of language and structure often differ from one another to a considerable extent. Hence, the establishment of the text variety is of decisive importance for the translator, so that he may not endanger the functional equivalence of the TL text by naively adopting SL conventions.

In this sense, the text types generated for a video game can be grouped into seven main categories, as initially suggested by Bernal-Merino (2006, 2007c) and Vela-Valido (2011): (1) narrative: heard or displayed, it carries the information about the game world and its characters; (2) oral/dialogic: heard or displayed, it represents the transcription of characters speaking to themselves of others in the game world; (3) technical: displayed or printed, containing detailed information about the software and hardware required to enjoy the game; (4) functional: displayed as part of the menus and enabling players to choose between different game options; (5) didactic: displayed, printed or heard, to train players to use the game application; (6) promotional: printed or displayed, to encourage users to buy more products, and (7) legal: printed or displayed, to advise buyers of their rights and duties.
as owners of the game product. These assets are organised in Table 4.1, according to the text types with which they comply.

The following paragraphs refer to each one of the components that make up the game asset groups mentioned above, relating them to the text types with which they comply and analysing their role in order to highlight their characteristics and illustrate the different writing skills required of translators. This review of assets and characteristics draws clear parallels with those of previous media as seen in Chapter 3 when describing multichannel texts, while highlighting at the same time the essential fact that they are all combined in a single product, the video game.

### 4.2.1 In-game Text

In-game text refers to the text strings found in the UI (user interface), the OS (operating system) messages and the game installer.

#### 4.2.1.1 User Interface (UI)

The user interface (UI) is also often referred to as the menu. This is made up of short text labels that trigger specific computer behaviours. This is a functional text type that can take the form of a reader-friendly version such as ‘play’, or of a more aseptic computer command such as ‘IP override’ for the network settings options. The UI is used to control the hardware preferences as well as the many aspects of game play. As with utility software (Esselink 2000: 26), video games make use of very detailed and often complex menu options with which to control the different features of the game,

<table>
<thead>
<tr>
<th>Game asset</th>
<th>Text type</th>
<th>Narrative</th>
<th>Oral/Dialogic</th>
<th>Technical</th>
<th>Functional</th>
<th>Didactic</th>
<th>Promotional</th>
<th>Legal</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-game text: UI, system messages, game installers</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Voiceover and Cinematics: audio and video scripts</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Art: game logo, in-game texture embedded words</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>-</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Glossaries and TMs</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td>Packaging and promotion: box, manual, EULA, guarantee, ‘ReadMe’, help files, official website</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
such as level of difficulty, graphic display selection and controller sensitivity. Figure 4.1 from the ‘Controls’ page of the UI of Batman: Arkham City for PS3, shows two of the main challenges encountered when localising UIs: available space and terminology. The labels describing the control functions in Spanish are often longer than in English, for example: “Detective Mode / Hold to Scan” becomes Modo Detective / Mantener para escanear, and “Run / Double Tap to Evade” becomes Correr / Doble toque para esquivar. Thanks to the fact that this UI was designed with plenty of available space, the Spanish labels fit without any problem, despite being visually longer. The terminological appropriateness and accuracy is also illustrated here by “Cape Stun” which has been translated as Aturdir and not Aturdir con capa, so Spanish players would not know by reading the Spanish UI that the cape is used to do the ‘stunning’. Finally, this example is also illustrative of a translation issue that concerns orthotypography and formatting consistency in terms of capitalisation. In the English UI, all nouns and verbs are capitalised, even when used in explicative phrases such as “Double Tap to Evade”, but this is not the case for the equivalent Spanish phrase Doble toque para esquivar, although the Spanish version does capitalise Toque para Disparo Rápido de Batarang, making the Spanish UI inconsistent in its use of capitalisation.

Unfortunately, the text found in menus is presented to translators in tables or lengthy spreadsheets, which, together with the fact that the game is not actually available for them to contextualise, means that a reasonably simple translation task is turned into an error-prone guesswork exercise. This part of the translation of video games can prove very problematic due to the brevity and condensation of some of the concepts, the random order in which they appear on the spreadsheets and the lack of a WYSIWYG (What You See Is What You Get) translation work environment. The space available in menus, pop-up windows, and hint captions is at a premium
Translation and Localisation in Video Games

and redesigning these windows is rarely an option because it impinges on engineering time. Translators have, therefore, to come up with solutions that contain a number of characters comparable to that of the original label.

A less frequently used option involves a reduction in the font size, but this affects the legibility of the text displayed. The margin recommended by experienced localisers when translating from English is to allow for a +30 percent increase for Roman-alphabet based languages, and to reduce the length of the original text by 50 per cent when translating into languages which use ideograms (Maxwell-Chandler 2005a: 9).

The UI text needs to be short, clear and precise in terms of the information displayed, because its main purpose is to facilitate play without interrupting immersion or breaking the suspension of disbelief. Icons are often used instead of words in UIs, since the former are, generally speaking, more universally understood, and sometimes the solution involves the use of graphic textures with embedded text, which, of course, requires extra time for localisation into other languages. Besides having to find counterparts that respect the informative and functional nature of the original text, an ingenuity which conforms with the spirit of the game world itself is a valuable asset for translators of game texts. Figure 4.2 is an example from Viva Piñata showing the store where players can buy their gardening supplies. The humour in the name of the store ‘Costolot’, a phonetic reference to ‘costs a lot’, may not be very obvious if translated from a spreadsheet with no contextual image or information. ‘Costolot’ was not translated in the Spanish version of the game.

4.2.1.2 System Messages

Whether a computer or a console is being used as the video game platform, system messages contain technical information in the form of official error
reporting messages that have been approved by platform manufacturers such as Microsoft, Nintendo and Sony Computer Entertainment, as well as promotional information that the game publishers, the console manufacturers or any of the stakeholders wish to promote. As with the translation of documentation, these messages contain official terminology and trademarks that do not allow for translation errors or variations; these are illustrated in Figure 4.3, from the system messages of PlayStation Store. The content and wording of these messages have to follow style, formatting and terminology guidelines within each language and across all localised versions. No variation is permitted unless it has been formally validated by the platform holders. In the example shown above, ‘PlayStation®Store’ and ‘PS3™’ must remain unaltered across all languages.

There is an official version for each system message, and it is compiled and translated by means of a spreadsheet, which often breaks the natural order in which system dialogue exchanges are sequenced, decontextualizing its meaning and introducing uncertainty in the translation process. More recently, some companies have started to work with translation memory tools, which can make style, terminological consistency and system version updating more efficient, although there is still some way to go before the natural order of the dialogue flux becomes clear to translators. Maintaining the official terminology of each device and company is the main concern when translating these texts, but there is an added difficulty when operating systems are updated within the same hardware generation and glossaries or TMs have not been forward to translation agencies or have not been kept in line with the new versions of the documentation.

**FREE** to access, PlayStation®Store is the only place to download new and exclusive PS3™ games, FREE playable demos, add-on packs, and high-definition videos.

Get more for your PS3™ and visit PlayStation®Store today.

* Broadband Internet connection required.

PlayStation®Store, de acceso GRATUITO*, es el lugar ideal para descargar juegos nuevos y exclusivos de PS3™, packs complementarios, videos de alta definición y demos jugables gratis*.

Saca el máximo partido a tu PS3™ y visita ya PlayStation®Store.

* Se necesita conexión a Internet de banda ancha.

Figure 4.3  System message for PS3. © Sony Computer Entertainment. All rights reserved.
4.2.1.3 Game Installers
These are short programs that transfer the game files into the user’s hardware in an organised and precise manner, creating the right paths to enable the game to run smoothly on the gaming platform in question. Some commercially available installers may already come with localised versions for many languages, but if game publishers are targeting new markets for which the installers have not been translated, then new languages have to be included. Some developers may create their own game installer, such as Microsoft for Xbox 360 (Figure 4.4) which, unlike the off-the-shelf version, makes Microsoft responsible for its localisation into all the required languages. As in the previous two sections, the text to be translated in game installer applications contains technical information and system dialogue strings that have been approved by platform manufacturers and must always be expressed and displayed in the same manner. These texts are short and syntactically simple, but it is precisely this succinctness coupled with the need for specific terminology that makes the task difficult for translators. The problem is exacerbated by the need to work from the ubiquitous and context-less spreadsheet.

4.2.2 Voiceover and Cinematics
All issues involved with the translation of strings for audio files and video sequences are discussed in this section.

4.2.2.1 Text for Voiceover
All audio texts recorded for video games, which are referred to as voiceover, come in three degrees of complexity, or in the language of game localisation producers, three different prices:
1. Lip-synchronisation, known as lip-synching, is found in trailers, introductory cinematics and game-engine animations. It is the most costly and time-consuming type of voiceover because the synchronisation of facial movement is required across localised versions. Whether animations (rendered in-game by the engine when playing), or video clips (pre-rendered and ready for playing when triggered by the player) are involved, these files tend to be high-quality sequences, in imitation of cinema standards, and therefore, requiring articulatory accuracy to fit in with the images, both in the original and in the translation.

2. Dubbing, also found in cinematics and in-game cut scenes, but where only the total duration has to be matched since the characters’ mouths are not visible. The duration of the translated audio file is the most relevant aspect of this option. It has to be matched as closely as possible to the original, so that the amount of data storage allocated to each language is kept within the capacity of the game discs.

3. Voiceover, audio-only files used in different parts of the game where the speakers’ faces are not visible, meaning that the localised audio files can be longer or shorter than the original. They can easily be substituted by the localised ones. Ambient dialogue and random inarticulate sounds uttered by unimportant characters and creatures in the game are included in these files.

These are related to dubbing practices commonly used in cinema and television. Chaume (2004a: 72–73) writes about the three synchronies needed in dubbing: phonetic synchrony, kinetic synchrony and isochrony, which involve achieving a similar duration of ST and TT lines uttered by the characters on the screen to maintained the revoicing illusion.

Voiceover text is therefore made up of all the audio scripts (Maxwell-Chandler and O’Malley-Deming 2012: 10), including dialogue and narration monologue, and these are presented to translators on excel spreadsheets. As explained in Section 3.2.3.2, the revoicing of video games is rather different from the television and cinema industries. The information contained in these files is organised in various columns (Table 4.2). When the original script has been prepared for revoicing in the translation, rather than subtitled, there will be information for all the professionals involved in the revoicing process, such as the name of the character, the cue or actual text that needs to be translated, the context to which the utterance belongs, the inflection used by the character, the location or place where the exchange is taking place, the area within that location, the effect given to the sound file, and the name of the file. If the original strings have been extracted directly from the game code, instead of a dialogue list (Díaz-Cintas 2001), only columns 1, 2 and 8 in Table 4.2 are likely to be included on the spreadsheet sent to translators (Chandler 2006), with the obvious risk of not providing enough contextual information for a successful transfer.
<table>
<thead>
<tr>
<th>Character</th>
<th>Cue</th>
<th>Context</th>
<th>Inflection</th>
<th>Location</th>
<th>Area</th>
<th>Effect</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mordred</td>
<td>Aw, snaps! Busted</td>
<td>Arthur is all in Lancelot’s face.</td>
<td>Sarcastic</td>
<td>Throne Room, Camelot</td>
<td>1.2</td>
<td>None</td>
<td>M01-a02-mor01</td>
</tr>
<tr>
<td>King Arthur</td>
<td>Shut up. Come on, Lancelot. Let’s hear it.</td>
<td>Mordred’s such a damn clown.</td>
<td>Fed up</td>
<td>Throne Room, Camelot</td>
<td>1.2</td>
<td>None</td>
<td>M01-a02-art04</td>
</tr>
</tbody>
</table>
The dubbing of video files tends to present the most difficulties, especially when lip-synching is required in close ups where the mouths of the characters can be seen, in which case bilabial (b, p, m) and fricative consonants (f, v), as well as open and closed vowels will have to match the original video animation (Chaume 2004a, 2004b), as in cinema and television productions (see Section 3.2.3.2), so that the illusion is maintained. Nonetheless, depending on the type of game, accurate lip synchronisation might not be required even in the case of close-ups, because the characters may be wearing masks or because of the cartoon style of the graphics (see Figure 4.5).

The translation of audio-only files, or of text that will be voiced over instead of dubbed, tends to allow for less complicated translation and adjustment, because the target text does not have to match the articulatory movements of the original, and space and time constraints are less stringent.

In a similar fashion to characters appearing in films, game characters often display a colourful array of registers, accents and idiosyncrasies that contribute to rounding their personalities and making them unique: a stylistic device that is really no different from the emphasis on oral discourse found in stage plays and films (Bernal-Merino 2008c). These features have to be accurately translated and dubbed in order to convey all the different nuances in the target text. Of course, in addition to translating the dialogue exchanges, translators may have to provide relevant information for the sound engineers about the filters they may use, the actors, the performance or the expected inflexion. They may need to liaise with the localisation engineers concerning the location of a particular file and its trigger within the game code and inform the dubbing directors concerning the manner in which each utterance is connected to the storyline and the characters in the game, as can be seen in Table 4.2.

The image in Figure 4.5, from the video game The Witcher 2, based on the popular novels by Polish writer Andrzej Sapkowski, is a good example of characterisation based on the idiosyncratic use of language. The original video game content in Polish, as well as the English and Spanish original translations, are used as an demonstration of the way in which writers and translators create an idiosyncratic use of language and spelling for the characters in their stories.

**Original Polish:** Ludź widzi trolla i do zabijania się bierze. Zabiliby. Na pewno.

[My back translation: “Humans troll see and to kill intends. Surely kill troll they would.”]

**Original English translation:** “Humies troll see, to kill humies itchy. Would kill troll, sure.”

**Original Spanish translation:** Trol ve humanos, ganas de matarlos. Ellos matarían a trol, seguro.

[My back translation: “Troll sees humans, feels like killing them. They’d kill troll, surely.”]
All three versions incorporate altered syntax and paratactical constructions, as well as creativity in the use of some nouns such as Ludź and Humies. The Spanish version seems to be the least imaginative; it is almost syntactically correct and it does not play with the lexis since humanos is the standard form. These strategies are not very different from those elaborated in Chapter 3, in which translations of non-standard language are shown in the Figure above, in which unconventional spelling is used to enhance characterisation.

Apart from the dialogue exchanges that take place between the characters, the game may also include other types of utterance to fulfil different roles. Most games feature characters, who talk to themselves in lengthy monologues; to each other, allowing players to listen in or as a response to the players’ actions. Although some of these characters may have little relevance to the main plot, their inclusion contributes to the immersion of the players in the virtual world by creating a credible and enjoyable experience. As in stage plays and films, accomplished creations manage to break through the fourth wall (Bell 2008), the imaginary boundary between fiction and reality, so that the spectators empathise with the characters. This is not the full extent of the situation, as video gaming includes players in the action itself. The fourth wall is broken, and a third space is created in between the game machine and the user, creating a reality beyond what is achieved in contemporary theatrical performances (Stevenson 1995). The game players become the protagonists, occupying centre stage with the spotlights turned on them.

Linearity is crucial to storytelling, and it is therefore essential for the translator to know how the story develops. However, when handling the voiceover scripts for video games, the narrative linearity of each dialogue is not always
easy to follow because of the many paths that interactive conversations can generate, as is shown in Figure 4.6, which is taken from Christou et al. (2011: 44), where the authors explain the very complex and lengthy localisation of Mass Effect 2. In Bioware’s case, translators could be given access to the game dialogue tool, which helped with contextualising partially its translation. The image shown in Figure 4.6 is displayed at the top of the dialogue tree window; in other words, the script where all the utterances of each character and all the different paths the conversation may follow, depending on players’ choices, appear in the correct order. The window at the bottom left allows for a sentence-by-sentence control, where writers, engineers and managers can edit the text and the scripts (or computer behaviours) it triggers and record and listen to the voiceover files. The bottom part of this window gives the word and the letter count, and it has tick-boxes to indicate the need for translation and voiceover. A small window at the bottom centre of the image allows for the type of comments that voiceover actors and directors would find useful. Finally, the window at the bottom right contains general comments on each of the conversation pathways; in this case, the highlighted dialogue option follows the ‘paragon path’ embodying the archetypical hero, as opposed to the aggressive or egotistical paths, which affects the subsequent events, options and dialogue exchanges in the games.

Figure 4.6 Interactive dialogue tool to help writing and translating keeping track of all possibilities © Bioware. All right reserved.
Most companies are much less prepared than Bioware to deal with the localisation of audio files, and they continue extracting translatable strings directly from the game code into Microsoft Excel files rather than preparing detailed dialogue lists as is standard practice in the film industry (Díaz-Cintas 2001). As seen in Table 4.2, the chronological development of the sequence in the dialogue may be contextualised and clarified on spreadsheets through the actual filename, by using compound alphanumeric filenames, such as “M01-a02-gue01” or “M01-a02-art03”, which can be very puzzling unless instructions are provided. However, the logic behind the filenames and the numerical or alphabetical way in which computers organise the content of a spreadsheet conflict with the narrative pathways of the game composed by the scriptwriters. It is no surprise, therefore, that coherence and cohesion can be seriously compromised by this way of working. Without the right sequence of events and a clear order in which dialogue exchanges take place, translations are bound to be wrong, with errors, for example, in who and how games address non-playing characters, meaning that words with a deictic function lose their antecedents, whether endophoric or exophoric. Additionally, games risk ending up misleading players in their immediate goals and overarching game quests. In contrast, in dubbing for television and cinema, the natural continuity of the programme is maintained in the dialogue list provided for translators to enable them to follow the logic in the way the dialogue exchanges have been interwoven. Video game translators have no dialogue list or script, only a spreadsheet with the text arranged in hundreds of cells.

As explained in Section 3.2.3.2, once the translation has been carried out, the dialogue writers for dubbing (who are sometimes also the translators) divide and number the script in takes, depending on the number of characters taking part in the conversation and the number of lines they have to revoice (Chaume 2004b). When it comes to the actual recording, many studios prefer to work with all the dubbing actors taking part in a given scene at the same time, in an attempt to encourage communication between them and to enhance the credibility of their performance. This is not the usual process as far as video game dubbing is concerned, and many “studios won’t use sequential order to record but rather proceed by characters, one after the other” (Le Dour 2007: online). In many cases, dubbing actors may only receive their lines, as opposed to the whole script, which can make their work unnecessarily difficult due to the lack of context. The original actors benefit from working with the game writers and designers, as well as with the freedom to improvise.

Concerning lip-synching, there is a wide spectrum of different needs in the case of games. This ranges from what could be described as ‘rough dubbing’, consisting in a basic open and closed mouth shot without much articulation (Figure 4.7, left) seen in LEGO Indiana Jones: The Original Adventure, to an almost life-like lip movement such as appears in Heavy Rain, which is described by the developers as the perfect combination between gaming and cinema (Figure 4.7, right).
Video game audio engineers have come up with some practical solutions to compensate for the time and money constraints experienced by the first games to have received audio localisation. In many cases, when videos are not available to the translators (for the wide variety of reasons explored in Section 5.7) but revoicing is required, the recording of the target text is adjusted \textit{a posteriori} rather than asking translators to work with pre-spotted documents, where the times are included. As opposed to the latter approach, more common in dubbing for the television and cinema, the audio engineers of video games match the translation given to them to the length of, and gaps in, the clip, guided by a graphical representation of the sound wave. Figure 4.8 shows the original English recording on the top and the translated performance below (Le Dour 2007: online).
Clear evidence of the movement within the video game industry to match the translation quality current in the film industry, and to conquer part of the traditional audiovisual market, can be seen in the development of software applications such as FaceFX, with which facial animations are matched to a combination of text strings and their audio files. Figure 4.9 shows FaceFX, one of the tools widely available in the market, where the sentence “Bandits, thieves and brawlers: by order of the Ministry Guard, you’re all under arrest” is analysed by breaking it down into its constituent phonemes. As the screen capture shows, the text has been matched against the voice sound wave frequency analysis and the automatic facial animation natural to the pronunciation of each of those letters and phonemes. This technology allows companies to recreate a 3-D facial animation of the game characters specific to each language they are localising into, producing a very high-quality lip-synching for all versions and reducing the time and costs associated to the hiring of dubbing studios and actors.

This technology has been used in leading games, such as Max Payne 3, Elder Scrolls V: Skyrim, Gears of War 3, Assassins Creed: Revelations, Battlefield, Call of Duty, and it is being increasingly adopted when adding sound to both the original and the localised versions since it is able to batch-process thousands of audio files, making 3-D graphic lip-synchronisation automatic. It has been heralded by some (Barnes 2012) as a likely cost-effective option for localisation managers, who have to deal with costly and time-consuming multilingual projects, because the sound wave analysis is language-agnostic, which means that it recognises phonemes and assigns to each of them the corresponding articulatory movements in the 3-D character model producing those sounds.

Despite the progress that has been made with regard to automatic lip-synching technology, it could still be said that the process of voiceover in games could benefit from some of the practices followed in the audiovisual translation industry. Something resembling a dialogue list used in film dubbing would elicit better performances than it does if the information is

Figure 4.9  The automation of lip synchronisation for games with 3-D facial animations. © FaceFX. All rights reserved.
arranged on a spreadsheet. Such an approach would save both time and money in the rehearsing, revision and recording of the dialogue exchanges.

4.2.2.2 Text for Subtitling
As discussed in Chapter 3, video game subtitles do not have very much in common with subtitles for television or cinema. They are not usually given any special or systematic thought in the video games industry beyond fitting the words on the screen to the overall visual design of the game. This can be seen in Figure 4.10, from Mass Effect. This is a science fiction action role-playing game in which the positioning of the subtitles is very flexible, and they can appear both at the top and at the bottom of the screen, or within the images themselves. The font type departs from the standard Times or Arial in different sizes, to embrace completely new and unique designs more in line with the look and feel of the game story, although potentially affecting the legibility of the text. In the example given, the font is bluish and modern, and there are two lines with up to 110 characters per line, more than double the number usually recommended for cinema and television subtitles (D’Ydewalle et al. 1987; Díaz-Cintas 2001; Neves 2005).

The subtitles used in video games do not seem to follow strict guidelines as far as positioning, layout or time considerations are concerned. Concerning the positioning, as we have seen, game dialogue may appear in different areas of the screen rather than being restricted to a place at the bottom of the screen, as is usual in the subtitling of audiovisual media. This approach has the effect of turning the screen into something more like a comic-book vignette than a traditional audiovisual product (illustrated in Figure 4.11, from Dragon Age II).

![Figure 4.10](image)

Figure 4.10 Very high number of characters per line in video game subtitles. Mass Effect © Bioware/EA. All rights reserved.
Regarding the number of lines, for instance, there might be one long line or five shorter lines per subtitle without much apparent logic in the line breaks. The font type and size of the text sometimes vary within the same game, presumably in an attempt to enhance the aesthetic complementarity between text and image, and some subtitles may contain text in a big font followed by another subtitle written in a smaller font. Subtitles often stay on screen until the next character speaks, and line breaks follow no apparent syntactical logic. Even games developed by the same company or distributed by the same publisher lack a consistent approach to subtitling. Regarding the actual content, intralingual subtitles tend often to follow the verbatim written rendition of the voices (dialogue and narration) that can be heard in the audio track. There is no text condensation as is the case with intralingual and interlingual subtitling in films (see Section 3.2.3.1). It is simply not considered. The audio script translation is often utilised in the sound studio and for the subtitles, there is no difference made between the two purposes.

The text used in the subtitles created for video game cut scenes (pre-rendered videos) and cinematics (in-game engine animations) can sometimes be hard-coded, which means that it is actually mixed with the program command strings that create everything in the game; this makes its retrieval, edition, translation and reintegration a rather time-consuming task, both cryptic and prone to error. Fortunately, this is occurring less and less as regards the translation process, and, nowadays, most developers have their own internal semi-automatic programs (not available for public use or scrutiny) which extract translatable strings for subtitles into spreadsheets, a procedure which has the benefit of making the text very easy to edit and to reintegrate in the target version. In this approach, information concerning the context to which the subtitle belongs is lacking and translators are, necessarily, left to their own devices.

As regards the temporal dimension of subtitling in games, the subtitles are only roughly synchronised with the soundtrack when compared with the
practice usually followed for cinema, television and DVDs. They may disappear from screen too early or stay on screen until the next subtitle comes in, which tends to coincide with the turn of the next character to speak. The so-called ‘6-second rule’ applied in standard subtitling, whereby a two-line subtitle of some eighty characters remains on screen for a maximum of six seconds to allow for the comfortable reading of the text (Díaz-Cintas and Remael 2007: 26), is not followed in the case of subtitling for video games. The result is that, especially when the actor’s delivery is fast, there is not enough time to read the subtitles that appear on screen at speeds exceeding 200 words per minute (wpm) or twenty characters per second (cps), figures that exceed what would be considered a comfortable reading speed in the subtitling industry, which usually ranges from between 160 to 180 wpm, or fifteen to seventeen cps (ibid.). There are, in fact, no real conventions or parameters that are applied systematically beyond what programmers consider appropriate for any particular scene in the game. This lack of real consideration for an appropriate reading speed for the subtitles is exacerbated when the translation process involves languages that are more long-winded than the source (such as English when compared to Spanish), because all language versions follow a master template where subtitle timing is programmed into the game code following the original, which often means that they are displayed for too little time for players to read them comfortably while playing.

To add to the problems, subtitle display often changes depending on the screen resolution used by the player. Indeed, the font may actually appear bigger or smaller, and occupy a different percentage of the screen as is shown in the screenshot from The Lord of the Rings: The Fellowship of the Ring (Figure 4.12), where the same subtitle can take two or three lines depending on the graphic settings enabled by the computer hardware or chosen by the players. When subtitles are translated, the readability may be compromised when languages that are more long-winded than English (such as German) are involved, or those in which diacritics are employed (such as Spanish), or those which use ideograms (such as Japanese). Rather than adhering to syntactical criteria, line breaks follow spatial considerations, measured in pixels. The example given in Figure 4.12 contains the sentence in Spanish: “Los Anillos mágicos, como tú los llamas, fueron forjados por los Elfos. Pero creo que este Anillo fue hecho por otro . . . Dámelo.” In low screen resolution, this game allows for a maximum of sixty characters per line, which means that ‘por’ is separated from the noun it accompanies, ‘los Elfos’. In high screen resolution, the first line allows for 120 characters, which means that, in many cases, the second subtitle is comparatively short, in this case only seven characters. This type of discrepancy in the length of subtitle lines, occurring due to changes made to layout settings, is always discouraged in television and cinema (Ivarsson and Carroll 1998a; Díaz-Cintas 2001; Karamitroglou 2000). This is because more reading time is required and the result is, aesthetically speaking, unsatisfactory. The spatial aspect of subtitles has, indeed, to be considered, but their communicative function, essential in the creation of the game experience, can best be achieved when syntactical and semantic criteria are adhered to.
High-definition television sets and new graphics cards enhance the picture, though, unfortunately, they create more pixels per inch, which means that in order to achieve a higher resolution, an image is made up of smaller dots, a fact which does not necessarily correspond with better legibility in terms of the subtitles. In practice, higher graphics settings often mean that the onscreen text takes up less space. This occurs not only because proportional rather than the monospaced lettering typical of teletext subtitling is used, but because the pixels making up the image are closer together. This technical advance had a great impact on professional cinema and television subtitling practices, allowing for a more rational use of the screen space available for the translation (Ivarsson and Carroll 1998a). There is also the risk of worsening the legibility, especially in video games, if the right font type, size and shadowing is not chosen, as can be seen in Figure 4.13, from *Batman: Arkham Asylum*, where the subtitle “Oh no, I forgot. Doctor Kellerman was in the Patient Observatory Room and Dr. Chen went to Surgery” can be barely read. Since players using subtitles do this out of need or personal preference, their legibility should be a prime requirement, and the aesthetic evaluation should probably be based on their adequacy to the game art style and colour palette.

Sometimes, the text displayed in a video game to indicate interaction between characters, system messages and gameplay hints is part of the
whole game design and must conform to the overall aesthetic concept. There is no specific font type, size, colour or standard way of signalling who the speaker is, as there is in subtitling for the deaf and the hard-of-hearing (Neves, 2005). The use of character labels in video game subtitling is a growing trend in lengthy story-based games, because of the number of characters with whom players must interact and because publishers want to encourage the deaf and the hard-of-hearing community to become gamers. Individual games are characterised by different methods of conveying this kind of information. The following example in Figure 4.14 illustrates this point. It comes from the medieval fantasy role-playing game Elder Scrolls IV: Oblivion, in which the names of the characters, Varnado in this case, are displayed at the bottom right-hand side corner of the screen, below the subtitles, sharing the same colour (white). In this instance, there is a relatively large number of lines, which is very different from what is normally done in subtitling for television or cinema. There are up to five centre-justified lines of forty-five characters for the dialogue with an additional line for the name of the speaker. The ample variation in game subtitling comes from the infinite possibilities that game programming gives. However, bearing in mind that we are used to thirty-five to fifty characters per line from reading newspapers, books and film subtitles, it would make sense to follow these parameters.

Although the wealth of possible options from which to select a style of subtitle can be justified from the point of view of graphic design, the variety of choice is questionable in terms of readability and legibility, which should,
in fact, constitute the two guiding principles for subtitling. As described by Mueller (2001: 146), the former “may include changing time codes, additional condensing or even pointing out nuances of meaning (from visual cues) which the subtitler may have missed”, whereas the latter is determined by aspects such as the “visibility of the titles, the colour, the size and the type of characters” (Dewolf 2001: 179).

Some of the text presented on screen in video games differs from the text used in the subtitles used in audiovisual products, which tend to relate to the translation of dialogue, whereas in video games, the purpose of the text is to enhance interactivity and, as such, resembles the text which is used in the menus of any software application. Figure 4.15 shows a screenshot from *Mass Effect 3* displaying what appears to be a subtitle reflecting Joker’s words, but followed by a set of four possible responses from which the player must choose in order to continue the dialogue (and contributing to the overall autobiographical and cinematic experience offered by game). However, these interactive lines cannot be considered to be subtitles as such, even though they share the same informative function and turn the aural into visual. Rather, they become a new hybrid by fulfilling the communication act of a dialogue exchange whilst including UI characteristics, which have long been in use in utility software and website applications. An appropriate term by which to refer to them would be ‘interactive subtitles’ to account for their informative value, as well as their interactive function, allowing players to shape the progress and outcome of the game adventure.
As with all the other characteristics associated with subtitles used in television and cinema, colours do not have a standardised role in video game subtitling. In games, a colour is rarely assigned for the purpose of character identification as the user interface usually contains information relating not only to the story but to the progress of the players, indicating their statistics and scores, so that colours are needed to differentiate one type of information from another. Figure 4.16 contains an example taken from Diablo III, in which a light-yellow font is used for all character utterances (far left); white indicates treasures found in the quest (centre); purple indicates magic items (centre); grey indicates items of low value (centre, right); a capitalised green font is used for information about player skills (centre, top), and a bigger, cream coloured font is used for information regarding the experience and bonus points achieved with each new task completed (centre, bottom).

As explained in Section 3.2.3.1, despite all the idiosyncrasies highlighted in these pages concerning the nature and function of the subtitles used in video games—displaced positioning, varied use of fonts types and sizes, mix of colours, increased number of lines and indication of characters—the generic term ‘subtitle’ continues to be widely used in the game industry to refer to these very diverse types of text. A rival term might be ‘caption’, but this is mostly used in the US and Japan to refer to intralingual subtitling for the deaf and hard-of-hearing community. It may well be that the time has come for the use of both terms, ‘subtitle’ and ‘caption’, to be reconsidered in view of these new practices and usages encountered in video games, but for the purposes of this research subtitling remains the preferred term because of its widespread use across countries and languages.
4.2.3 Art

Artwork with words can be part of the packaging and merchandising, the UI, and the graphic textures of maps that appear in the game. A multilayered graphic file format is needed in order to be able to edit the text neatly without altering the original art style. This can be achieved with image editing programs that can seamlessly apply multiple layers to an image. One of the layers may contain the graphic art used in the game, while other layers will be created for texts, such as the title in Figure 4.17, from the main menu screen of The Lord of the Rings: The Fellowship of the Ring. In this example, the letter ‘r’ in ‘lord’ and ‘rings’ was modified to create a circular shape evoking a ring, while in the Spanish version, the equivalent effect was achieved in the words señor and anillos modifying the ‘r’ and the ‘a’.

This procedure does not apply only to game logos; graphic-embedded text contributing to making the game world both credible and enjoyable may appear anywhere in a game, regardless of the genre, and it certainly requires translation in order to keep players engrossed in the adventure and not to alienate them with texts in languages that are not required by...
The Translation of Video Games

131

the story. Immersion can be further enhanced by quality 3-D animations, inspired voice acting and appropriate sound effects and music, but there can be little doubt that most of the credibility of each localised version depends on the quality of the target text, as in the example given from the pirate school found in *Escape from Monkey Island* (Figure 4.18).

In this example everything has been translated, even the name of the teacher which is changed from ‘Miss Rivers’ to *Srta. Ríos*, whereas the drawing, writing and colours of words embedded in the graphics have been retained in the Spanish version, replicating the style of the original exactly. The only text that creates a problem is the sign ‘Learning is Good!’ because it has been translated as ¡El Aprendizaje es Bueno! which can barely be read during play because of the space constraints of the graphics in which the words are embedded. The fact that translators work from spreadsheets, which is almost always the case, with no access to the actual graphics or to the moment in the gameplay where these graphic-embedded words can be seen, makes it more difficult to produce the ideal translation. In this case, for example, with a knowledge of the available space in the image, ¡Aprender es Bueno! would have seemed a better translation since it is shorter than the official translation by six letters. Some constraints remain undetected until the localisation process is finalised and rushed for gold copy (the master copy) and international release.

More pronounced examples of graphic changes during localization can be seen in Japanese games such as *Nier Replicant, Nier Gestalt* in the rest of the world, where Nier changes from an adolescent hero looking for his sister to a muscly, masked father looking for his daughter (Figure 4.19). Another good example is *Zero* known as *Project Zero* in Europe and *Fatal Frame* in the US. In the original Japanese version the female protagonist, Miku, is a pale, thin adolescent Japanese girl looking for her brother Mafuyu who has disappeared into a haunted mansion. In the western version, she is a rather voluptuous western woman that wears a modern outfit instead of the original Japanese school uniform (Figure 4.19). Interestingly, developers did not apply the same westernising principle to her brother, so when players do find Mafuyu at the end of the game these characters do not seem to be blood-related.

![Figure 4.18 Localisation of linguistic graphic art in game textures. Escape from Monkey Island © Lucas Arts. All rights reserved.](image)
4.2.4 Glossaries and TMs

Although these would normally be classed as reference material and not text to be translated for the game as such, agencies and translators have to create or add to these files so as to enable the whole localisation process. Traditional bilingual glossaries are still being used by translators, particularly for the translation of the more obscure aspects of medical and engineering research, new glossaries have been developed, becoming powerful databases capable of automated searches and the translation of specific terms. Some modern glossaries and terminology databases used in games include all kinds of multimedia information, as is seen in Figure 4.20, a mock-up from the localisation of Resident Evil 5, helping the translator to make informed decisions based on the game lore and to maintain consistency throughout the game. The cross-referenced, multilingual capabilities and the rationalised use of space in the UI of the TM can also help with the checking and simultaneous updating of all the language versions for technical and trademarked terms, as well as ensuring consistency in spelling, game feature labels and character names, as is illustrated in Figure 4.20.

Localisation companies working on a PlayStation title benefit from having an official terminology list, which is compiled and updated internally by a dedicated team. Tom Boerger, Senior Manager of Product Information Design and Development at Sony, highlighted (Boerger 2010) that terminology databases (TDs) used in conjunction with translation management tools provide a solid foundation from which the platform conventions (so important in game localisation projects) can be referenced, tracked and updated. However, even with a good TD, there are still many challenges, which include linguistic variation (gender and case), space constraints, branding
requirements (capitalisation), duplications, abbreviations and acronyms (*ibid.*,). The team at Sony Computer Entertainment (SCE) decided that the consistency of the localisation required for all SCE products was on such a scale that it had to be tackled from three angles. These included the employment of specialised staff, the design of a dedicated terminology related tool, and the ultimate refinement of the process (*ibid.*). Boerger’s team was not just referring to the linguistic pairing of terms used by SCE in their manuals, but to all the instances in which any SCE-related terms, logo or image appeared on their products—that is, in print, on screen, on the hardware

Figure 4.20 Multimedia terminology database. Mock-up from *Resident Evil 5* © Capcom. All rights reserved.
and on boxing and promotional material. Their database had to include the familiar terminological information, as well as icons such as the PlayStation logo, information concerning official colours and relief description depending on the hardware on which they were meant to be displayed.

It is worth bearing in mind that most game products require translation to be carried out in parallel, simultaneously in order to meet deadlines, so terminological and syntactical consistency is by no means guaranteed if not planned for. Terminology management tools are explored further in Section 5.9, where the various tools used by the localisation industry are discussed.

4.2.5 Packaging and Promotion

There is a mixture of text types in evidence on the game boxes. One of the main differences between them lies in the space provided, which is limited not only by the physical size of the box but also by the inclusion of images pertaining to the game, company logos, age-rating labelling and technical hardware and software requirements. The game box often includes an alluring promotional text, using an exhortative second-person pronoun and the imperative form of the verb, as shown in Figure 4.21, from the back cover of Mass Effect where, for example, the original US English text “Customize your character and embark on an epic adventure in an immersive open-ended storyline” becomes in the French version “Personnalisez votre héros et plongez dans une aventure passionnante aux multiples issues”. Age rating and copyright notices are also revised in accordance with the cultural and legal framework of a particular country, as is seen in the bottom part of the back covers shown in Figure 4.21. All these correspondences are marked by directional green arrows. An important piece of information for players of localised versions, in this case French speakers, is that the manual, UI- and audio are in French, as highlighted with a yellow oval in Figure 4.21.

One of the main aims of this type of text is to set the product clearly within a given genre and brand, as part of a strategy to attract potential buyers. It is therefore a highly promotional text, interwoven with essential technical jargon about the hardware requirements of the game, as well as legal copyright notices and age rating warnings clearly displayed for parents and children to see. This text type, which is so often found on the packaging of the video games, can also be found on the official website, in promotional articles and in some parts of the manual.

4.2.5.1 Manual

Although it may also contain some engaging creative writing, partly promotional and partly literary, most sections of the manual (also called the instruction manual) would normally consist of didactic texts containing simple instructions so that players can focus on enjoying the game rather than on deciphering the information. The main objective of these texts is to
explain to players how to play, irrespective of their knowledge of technology and video game culture. For this reason, they have to be written in such a way that their content is straightforward and welcoming enough to enable both experts and newcomers to discover and enjoy the intricacies of the game. This is even more important because levels of difficulty are adjusted to suit the skill of the player, so that games can be played on easy, normal and difficult settings. This enables players to improve at their own pace rather than becoming so frustrated by the game that they then abandon it.

Texts of this didactic nature can sometimes be reutilised for the online help section of the UI, as well as appearing as context-specific pop-up windows and system dialogues the aim of which is to remind players about basic story information and game mechanics, as in the example given (Figure 4.22), from the English and Spanish manuals of Viva Piñata.

Manuals are part of the official documentation with which games have to comply, so they often include technical texts with information concerning the appropriate hardware and software specifications needed for the application to run successfully, as well as sections on after-sales customer services and online connectivity. Even when simplified for users of all ages, they are often rich in technical jargon and trademarked terminology to which special attention needs to be paid, because it is an important part of the platform compliance approval process and the amendment of any errors or mistakes in
the translation will be time-consuming and might also have costly legal implications. As shown in Section 4.2.4, many companies send their glossaries in spreadsheet format to translators, but many have already incorporated TM tools with terminology extraction, management and updating, such as SDL Trados Multiterm application. Translators are expected to use the tools and to flag up any discrepancies. Even professionals can easily mix up the naming conventions of each of the three main gaming platforms, for example: ‘thumb-stick®’ (for Xbox) is not to be confused with ‘analog stick®’ (for PlayStation) or ‘control stick®’ (for Nintendo), although for most users all three of them look and feel very much the same when playing. It is not only the technical terminology used but also the concepts talked about and the rather formal syntactic construction of the sentences themselves that set these texts apart from any other text in the game. The following example is from the technical section of the manual of *Halo*, which is rich in specialist jargon: “Connection issues are typically caused by one or more ports being blocked by a firewall, router, or even your Internet Service Provider to help maintain security”.

4.2.5.2 EULA and Guarantee Files

The End User License Agreement (EULA) and guarantee files are the main legal documents shipped with every single game. They need to be acknowledged and accepted by the player when installing or initiating the game for the first time, and they are often printed out in the manual, as well as being stored separately in the main game folder in highly compatible file formats. The text type utilised here is of a legal nature as can be seen in the Table 4.3 from the copyright notice of the game *Viva Piñata*.

The syntax and the terminology employed in such texts tends to be very formal, as can be expected from legal documents (Mayoral 2003), and to follow law enforcement contract legislation appropriate for each country in which the game is legally distributed. Legal texts are normally drafted by the right teams or a management professional with the appropriate legislative background.
knowledge and responsibility to validate the final, original version. The translation of this kind of register requires a very specific type of writing and a knowledge of legal jargon and phraseology (Bestué-Salinas 2009) that does not come easily to non-specialists.

### 4.2.5.3 ‘ReadMe’ File

This ‘.txt’ file is usually the last document to be created in the game development process and is mostly found in versions of the game run on PCs. Its objective is to inform users of all the last-minute adjustments as well as to advise on how to make sure that the product runs smoothly on a home PC. This is because developers and publishers are aware of the possible incompatibilities between home PCs, which are due to their being highly customisable, as opposed to game consoles, the hardware of which is static. This file is also designed to inform players about possible mistakes and typing errors occurring in the printed material, such as in the manual and packaging, and were noticed too late to be corrected in the hard copies before the release of

<table>
<thead>
<tr>
<th>Unauthorized copying, reverse engineering, transmission, public performance, rental, pay for play, or circumvention of copy protection is strictly prohibited.</th>
<th>Queda totalmente prohibido realizar cualquier acto no autorizado de copia, ingeniería inversa, transmisión, comunicación pública, alquiler, pago por jugar o elusión de la protección contra la copia.</th>
</tr>
</thead>
<tbody>
<tr>
<td>© &amp; p 2006 Microsoft Corporation. All rights reserved.</td>
<td>© &amp; p 2006 Microsoft Corporation. Todos los derechos reservados.</td>
</tr>
</tbody>
</table>

**Figure 4.23** Technical text type used for ‘ReadMe’ files.
the video game. ‘ReadMe’ files are written in a rather technical and direct style, as illustrated in Figure 4.23.

They can also offer direct instructions on how to install a new patch, to fix minor problems particular to a given type of hardware, or to adjust software for optimum game performance. ‘ReadMe’ files also provide advice on peripheral compatibility to ensure the best gaming experience possible.

4.2.5.4 Official Websites
The texts used on official websites tend to be a combination of a promotional text type and a technical one containing details such as minimum system requirements and the like. Most of the information offered on the official website is similar to the one that is shipped with the game, usually included in the manual and the packaging. Nonetheless, websites tend to include additional information such as previews and reviews of the product, notice boards, forums, customer support details, downloadable files to fix specific problems and expansion packs with new chapters and language versions as well as screenshots, gameplay clips of the actual video game, developer interviews, merchandising, and even links to external fan blogs. The websites of the most popular games tend to be very busy, and language-specific community managers may be employed to learn about the likes and dislikes of players, with a view to improving future versions of the game.

The coexistence of so many different text types in video games requires the specific training and preparation of a new type of translator, with rather a different professional profile (see Chapter 6).

4.3 IN A CLASS OF ITS OWN

As explained in the previous section, in addition to the characteristics centred on game assets and text types, there are four differentiating factors that set the translation of video games apart from that of other products. Although they have been briefly mentioned previously, they are so essential and unique to video games that it is worth discussing each one separately. The four factors are: story-building interactivity, the fragmentation of the source text, the translation of variables and the localisation of voice commands and gestures.

4.3.1 Story-Building Interactivity

One of the most important characteristics of video games is the way in which they engage with the users, immersing them in a virtual reality where the incredible is made not only possible, but also controllable. This gives players a feeling of empowerment because they know they can choose their own line of action and ultimately beat the game (Wardrip-Fruin and Harrigan 2004). As with any other software product, this is partly due to the functionality of the virtual machine although a considerable part of the immersion, and
The empowerment originates from the way in which language is used. In an attempt to heighten the feeling of engagement, for instance, video games address the player directly in the second person, initiating a relationship with users, which differs from some other media (Svanæs 1999). Three different styles of writing are used for this interaction so as to position players in three different roles:

1. As a client of the developer and the publishing companies. These texts are normally written with an input from marketing departments, and they are pitched so as to attract customers to the product. They are used for packaging, promotional articles, websites, advertisements and radio and television commercials, as seen in Section 4.2.6.

2. As a legal owner and responsible user. This type of content and writing style can be found in the installer program, the EULA, and the ‘ReadMe’ files. These texts notify buyers of their legal rights and duties, at the same time informing them how to use the product responsibly and how to get the best out of the experience without damaging their health. The text type used here pivots between legal and technical, as highlighted in Section 4.2.6.

3. As the protagonist and champion of the challenge ahead. This particular text type is probably what ultimately convinces players to buy the products, as it is responsible for selling the promise of an immersion in a virtual world where they can live out extraordinary stories. Normally, players become more immersed in the game through these texts, and it is through their quality and tone that an engaging and memorable experience is created. Excerpts from these texts can sometimes be found on the packaging, in the manual and on the website, but most of it is accessible only while playing (or in-game), for example through video introductions, interactive dialogue and linking cut-scenes.

The advent of software ushered in a different relationship between consumers and products, changing a situation where consumers are idealised and often remain in the background to a situation where they are addressed directly through the video gaming experience. In a broad sense, interactivity has always been a part of human culture: humans interact with each other, with the environment, with the works of art and with machines they create, but it has only been possible to experience interactivity to the full now that computers, machines that can store, retrieve and process data, have increased their processing power more than a thousand times (Crawford 2004). In video games, this enormous processing potential is used in order to generate 3-D worlds in real time, play music and sound effects on a 5:1 system, and reproduce dialogue options with game characters so as to enhance immersion. Computers can be programmed to display specific information as players advance through the virtual game world, meeting NPCs and creating illusory conversations. Players can choose what they want to say, thus
influencing the conversations which follow. Game designers, scriptwriters and programmers work together to create a dialogic mechanic so that the characters in the story behave naturally according to their literary persona depending on what players decide to say, as is seen in Section 4.2.2. When all these possible conversation paths have to be translated, there are only two options: a spreadsheet document with hundreds of lines organised in a way that does not respect the logic of the story, or a proprietary tool such as Bioware (shown in Figure 4.6) that offers helpful metadata and an online ‘questions to the developer’ query system but is rarely available for translators and is, in fact, a tool the operation of which involves a steep learning curve (Warden 2008). All this means that the translation of all the possible conversation paths into multiple languages poses two different, but equally relevant challenges for translators:

- **Notional challenge**: the way people address each other in different languages and cultures may differ depending, for example, on age, professional profile, level of kinship and gender. This does not necessarily mean that some languages cannot differentiate between these modes of address but simply that they may do it through other means, and this is relevant to the understanding of the story and its game world.

- **Linguistic challenge**: the way these modes of address manifest in each language may differ in terms of grammatical rules. The typical problem when translating the English ‘you’ into Spanish is deciding if it is a polite (usted singular and ustedes plural) or a familiar (tú singular and vosotros/vosotras plural) mode of address. When translating into Japanese, as explained by Lau (2012), there are four possibilities: respectful-distant (meshiagarimasu), respectful-familiar (meshiagaru), distant-rude (taberu), familiar-rude (kuu).

Of course, these issues exist in other media, such as novels, comics and films, but the problem is compounded by the degree of interactivity required, not only in terms of mere mechanics but also in terms of storytelling and plot immersion. The text and narrative have to acknowledge the player, as well as his or her actions and dialogue responses. For many years, video game developers would decide on a prototypical player, often a male teenager, and write the text accordingly. This approach proved to be inappropriate once interactive entertainment started attracting a more varied fandom, including male and female players, young and old. In order to capitalise on the greater financial opportunities thus available, the industry experienced greater challenges in terms of writing and translating. For most of the twentieth century, the modus operandi was to write and translate all the sentences referring to the player both as male and female, and with both the polite and the familiar modes of address; a rather lengthy, expensive and time-consuming approach. Thanks to the development of computational linguistics (Roark and Sproat 2004), some professionals, with a knowledge of both computing and translation, realised that grammar rules could be programmed so that
one single sentence would automatically produce all the necessary variations. This breakthrough helped to increase the interactive possibilities offered to the players and meant the introduction and development of localised macros for game translation, that is, single computer instructions which stand for a whole sequence of operations. Based on the variable concept (see Section 4.3.2) found in mathematics, physics and computer programming, and working with an element the value of which is unknown, video games writers can write in variables that are replaced automatically (or ‘on-the-fly’ as it is often referred to in computer programming) by the game engine. For example, the string “You take the $ITEM$!” can be used for any of the hundreds of items likely to appear in the game, simply by picking it up from the items table (Heimburg 2003: online), so that the string could become “You take the sword!”, “You take the gold!” or “You take the sack of flour!” However, the system was initially still too simple to be able to account for the basic grammatical and morphological changes found in natural languages, so macros received a linguistic makeover in order to enable greater immersion and dialogic interactivity, a point that is further explained in Section 4.3.2.

Encoding interactive conversation into the source code requires the breaking up of the texts that make up the dialogue into shorter, independently named strings, so that the game engine can track and parse them correctly, coming up with sentences that create the illusion of direct interaction with the characters and their virtual reality. What this means in practical terms for translators is that the text that conjures up the game world, and all the stories within it, is presented to them in a fragmented form, so that they have to adapt to the organisation and input procedures of the virtual machine and, as a consequence, lose out on most of the contextual information, which is actually the most valuable source for them after the string itself. This point is further explored in the next section.

4.3.2 The Fragmentation of Interactive Text

For most of translation history, the written word reigned the page unchallenged by any other source of information, and texts were an inevitably linear exposition of content. With the proliferation of audiovisual technologies in the second half of the twentieth century, translators either had to learn to work from written scripts without access to the images present in the audiovisual product itself, or they had to cope with (poor) copies of the actual audiovisual programme and no script at all. The challenge translators faced was that they had to produce a final written text that did not conflict with the overall audiovisual filmic product. Having to translate text without taking into consideration the semiotic context of the product risks affecting the act of communication adversely (Cutting 2002: 1–14) because isolated linguistic items tend to have multiple possible meanings and, hence, give rise to an unnecessary ambiguity which slows down the translation process. However regrettable, this approach to translation is still current, although, on the positive side, language professionals can still count on receiving the
actual text to be translated in its natural order, in the same way in which it is going to be received by the user.

As already explained, (see Section 3.2.4.), software localisation can impede the task performed by interlingual communication professionals by introducing an additional issue, that of fragmentation. Pym (2011: 2) argues that “technology [. . .] disrupts linearity”, referring to the way in which we read websites. This statement, however, can be easily extended to include the field of software, which is understood as a multitextual medium, not only in the way information is accessed, but also in the way it is created and translated. According to the same author, this development tends to hide translation as, “this peculiarly technological movement is not especially away from the text as such, but away from linearity. The more technology, the less easy it is to make decisions in terms of linearity, and the less we tend to see translation as communicating between people” (ibid.: 4). Stripping 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. For the storytelling part of entertainment software, however, the implications can be both serious and far reaching.

In video games things are triggered through the players’ choices or actions. In fact, this relative freedom to resolve situations in the way and at the pace chosen by the players is part of the appeal of the games. It does not mean that there is no story or that games have a random and chaotic sequence of events but rather that the unravelling of the story is dependent on the individual choices and performance of the player. In order to achieve this expected interactivity, linguistic fragmentation becomes a permanent feature of the texts characteristic of entertainment software, since it is the underlying structure provided by the game code that makes interactivity possible, creating the illusion that it is the players who influence the virtual storyline. This particular feature of entertainment software products has great influence on the way scripts are written, as Chandler (2005: online), an experienced game writer, explains:

Oddly enough, an accounting spreadsheet can be a writer’s most effective tool. I use Excel to keep track of my dialogue, as do many writers. It’s particularly useful when preparing “active format” dialogue (any dialogue taking place in-game, where multiple variables can make it a challenge to keep track of all possible dialogue threads).

Everything in a video game has to be programmed through the game code, which is basically an artificial language used to give instructions to the computer. Programming languages are constantly being optimised to produce the best results with a minimum of commands, but still, virtual worlds have to be created or rather programmed into existence, as nothing ‘is’ there, and this applies to language, its display and its meaning as well. Figure 4.24 shows an example from the tutorials accompanying the Garage Games engines (www.garagegames.com), illustrating how even a simple UI requires thousands of...
lines of code to tell the hardware what and when to do things and these are often rather confusing for the translator. Figure 4.24 displays just a few lines of the code used to create the accompanying UI for a game in the middle of other lines of code used to control various behaviours and settings. A more exhaustive list can be found in Appendix 3 (from www.garagegames.com). It should be noted here that Figure 4.24 is just a brief example used to illustrate game code and how working with it directly would be problematic for both translators and engineers. Words appearing as part of the UI are in boldface. The indicator “[... ]” has been necessary in order to shorten the sample, since the original source code for the game would take hundreds of pages.

It is not impossible for the untrained eye to understand programming languages, but it is certainly very different from reading a novel, a screenplay

```cpp
[...]
// OrbitObject mode requires an object to orbit
// @client is the LocalClientConnection
getClient().camera.setOrbitObject(4this.player, mDegToRad(60) @ "0 0", 0, 30, 30);
[...]
singleton GuiControlProfile (InvList)
{
    opaque = true;
    fontType = "Arial";
    fontSize = 16;
    fillColor = [150 150 158]; // selection color
    fontColor = [255 255 255];
    justify = "left";
};
[...]
function inventoryGui::btn2()
{
    $lbInvTitle.setValue("SPELLS");
    $lbInventory.clearItems();
    for($i = 0;$i < 5;$i++)
    {
        $lbInventory.addItem($aInv[ $SPELLS,$i ]);}
}(...)
$FOOD = 0;
$SPELLS = 1;
$WEAPON = 2;
$ARMOUR = 3;
[...]
$aInv[$SPELLS,0] = "Fall From Grace";
$aInv[$SPELLS,1] = "Ice Call";
$aInv[$SPELLS,2] = "Water Wish";
$aInv[$SPELLS,3] = "Fire Storm";
$aInv[$SPELLS,4] = "Healing Heart";
(...)
ConsoleFunction(echo, void, 2, 0, "echo(${... })")
{
    U32 len = 0;
    S32 i;
    for(i = 1; i < argc; i++)
    {
        len += strlen(argv[i]);
    }
    char *ret = Con::getReturnBuffer(len + 1);
    ret[0] = 0;
    for(i = 1; i < argc; i++)
    {
        dStrcat(ret, argv[i]);
    }
    Con::printf("%s", ret);
    ret[0] = 0;
}(...)
```

*Figure 4.24* Coding for UI and UI screenshot. © Garagegames. All rights reserved.
or a set of subtitles. Game source code is too cryptic for game scriptwriters, so engineers have to create tools for scriptwriters in order allow for storytelling. The logic of the story is secondary to the way in which everything contained in the game is programmed, including: display, graphics, sounds, music, simulated physics, character behaviours and their utterances. If translators are given access to thousands of lines of programming language such as those shown in Figure 4.24, which was the case when localisation companies first appeared in the 1980s, the task of translation task is slowed down considerably and many functionally problems are accidentally created because of the game code changes inadvertently made by translators.

This affects the tasks of both writers and translators because they have to adapt to the logic of the computer so that the game machine works in the way it is supposed to. This is the reason why localisation engineers extract all the linguistic assets from the game and present them to translators in a format that is useful for all the parties involved in the team, which primarily includes the localisation, the programming and the quality assurance departments. The preferred format used to present this type of material is the spreadsheet (Chandler 2005: online) because it is a simple, rational structure that can be reduced to numerical values: the true language of machine code. Informative texts and characters’ dialogue are presented in columns, a fragmented but easy and effective way in which everybody involved in the project is able to locate specific details. By allocating a separate line and column to each piece of information (Table 4.4), the whole team is able to work with a more comprehensible source, and programmers can then automatically and safely insert the relevant strings back into the game code, avoiding the potential creation of errors in the source code by non-engineers.

Prioritising functional over linguistic needs may, at first glance, appear to be mistaken, particularly from a translator’s point of view, but whilst players are able compensate for some shortcomings in terms of communication, for example, computers are unable to counteract the coding mistakes which will ultimately cause the game to crash.

Owing to the fact that thousands of strings in thousands of files are used in the development of video games, clear naming conventions need to be followed and the files need to be correctly compiled and referenced for localisation purposes: a complex process which has to mirror all those strings, files and folders

<table>
<thead>
<tr>
<th>Line</th>
<th>Resource file ID</th>
<th>Original string</th>
<th>Spanish version</th>
<th>US English revision</th>
<th>Spanish revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>108</td>
<td>VO_vmiss5.wav</td>
<td>Oh, so close</td>
<td>Huy, casi.</td>
<td>Oh, so close.</td>
<td>Uy, casi.</td>
</tr>
<tr>
<td>109</td>
<td>VO_vmiss6.wav</td>
<td>Almost had it</td>
<td>Casi te sale.</td>
<td>You almost had it.</td>
<td>Lo tenías casi.</td>
</tr>
<tr>
<td>110</td>
<td>VO_vmiss7.wav</td>
<td>Don’t give up</td>
<td>No te rindas.</td>
<td>Don’t give up.</td>
<td>No te rindas.</td>
</tr>
</tbody>
</table>
into each of the languages of the project. Tables normally have a column for the resource file ID/name, one for the original string, and another one for the version (Table 4.4), but in each project the naming and organisation of assets may be different. As we have seen in Table 3.5, these tables also include a column in which the localisation engineer is able to add extra information to help voice actors, translators and other professionals. Although this may be of little consolation for those having to deal with thousands of lines and multiple columns, this procedure is considerably better than the method used in the last two decades of the twentieth century, where the text requiring translation was directly written into the game code, creating a complicated situation for both translators and engineers who were forced to puzzle over thousands of game source code lines. Translation memory applications have helped somewhat with the consistency of terminology, but visual localisation tools such as Catalyst or Passolo are still unable to offer much assistance in video game translation.

Tables are, no doubt, the ideal tools to order data that respond well to numeric organisation, but not to tell the stories, which are central to the game. They are somewhat more useful for non-linear storytelling because they enable the storage and retrieval of information by the game engine, but it is the translator who has to adapt to the tool and build a visual picture of characters, places and events, so as to be able to tell the story to the target audience in the same, engaging manner as the source narrative.

Even when all the possible precautions have been taken and files have been organised following an intuitive naming convention, translating games from spreadsheets can be confusing and time-consuming. Some of the information contained in each cell might not need to be translated because it is a variable part of the game code, although this may not be immediately apparent. As a rule, any text preceded by the ‘$’ sign, between curved brackets or square brackets (Table 4.5) is part of the game code and, as such, it is out of bounds for translators.

In the latter example, the text ‘[Title]’ will automatically be replaced by the name of the product, which may be undecided, changed several times during development or require a different font. Table 4.5 also shows other words that appear before an equal sign (NEW_STRING4 =), none of which

<table>
<thead>
<tr>
<th>US English</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW_STRING1=Play [Title]</td>
<td>NEW_STRING1=Jugar a [Title]</td>
</tr>
<tr>
<td>NEW_STRING2=Uninstall [Title]</td>
<td>NEW_STRING2=Desinstalar [Title]</td>
</tr>
<tr>
<td>NEW_STRING3=Red Storm on the Web</td>
<td>NEW_STRING3=Red Storm en la Red</td>
</tr>
<tr>
<td>NEW_STRING4=[Title] requires DirectX 8 to run\nDo you wish to install DirectX 8 now?</td>
<td>NEW_STRING4=[Title] Necesita DirectX 8 para funcionar\n¿Quiere instalar DirectX 8 ahora?</td>
</tr>
<tr>
<td>NEW_STRING5=Install DirectX 8</td>
<td>NEW_STRING5=Instalar DirectX 8</td>
</tr>
</tbody>
</table>
are to be altered in any way either. Line 4 shows a command between ‘run’ and ‘Do’ with no spaces in between (run\nDo). This ‘\n’ command simply tells the game engine to display the subsequent sentence in the line below, and it is important because it controls the position of text on the screen, directing the players’ attention to essential information. In some cases, this can be a question of design or personal preference on the part of the engineer, but in others it may mean that the text might otherwise overrun the space available and go beyond the safe area, or bleed out of its textbox with the risk that players will not be able to read the information. Similarly, blank spaces have to be left in exactly the same place, even when found in places that make no linguistic or syntactical sense.

As illustrated in Section 3.2.4, some of the problems likely to arise from inadvertently modifying out-of-bounds command strings can be partially solved by using visual localisation tools such as SDL’s Passolo (Figure 4.25) or Alchemy’s Catalyst because the code is either hidden or non-editable, and they present everything in a WYSIWYG environment, allowing translators to see and edit the linguistic component that needs to be translated and the text box where it is embedded in the right graphical context as it will appear in the final product.

These tools are, however, designed primarily for the translation of utility software and they can rarely be used for entertainment software because games do not follow the same coding practices. Some developers create their own in-house tools but, since the game industry does not adhere to any particular file format, let alone a standard, every game project is unique with regard to its programming. The reality is that translators have to accept what they receive, however complex or disorganised it may seem.

The following section deals with the third aspect that sets video games and their translation apart from other professional practices. The need for a player-driven story-building that enhances immersion and the diverse linguistic variables that need to be taken into account for a successful localisation are discussed.

Figure 4.25  SDL Passolo, a visual localisation tool. © SDL. All rights reserved.
4.3.3 The Translation of Linguistic Variables

There are various ways in which textual cohesion can be established. Pronouns (he, them, hers) and deictics (this, those) are used in a similar way once their value has been established by the context in which the communication takes place (Halliday and Hasan 1976). These are naturally occurring linguistic devices common to all languages owing to the fact that they create an efficient way in which to build up information without repeating what has already been stated, and they are of equal importance in translation (Guillemin-Flescher 1993). Textual cohesion is challenged in the translation of video games because of the need to leave deictic items referring to players and their actions as unknown, and computer engineers use variables in order to do this. A variable is an incognitum, the exact value (or meaning) of which is unspecified, although the range of possible meanings usually is (www.oed.com). Variables are commonly used in mathematics, physics, propositional logic, computing and statistics in order to calculate values from the numerical information available. In games, variables can stand for a number for 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, as will be explained in the following paragraphs.

Variables are used in many complex ways in order to trigger players’ suspension of disbelief, but it is not easy for programmers to devise strategies to take into account the various syntactical and morphological rules of their mother tongue, let alone those of the many different languages into which the game will be translated. The most common variable, and the easiest to explain, relates to the player’s name. For example, the winning message after the completion of a part of the game normally states: ‘/n player1 /n wins !’ The string between the ‘/n’ markers is the variable (Player1), which the program will substitute depending on the particular choice made. When playing an original video game in English, and in the case of a player named ‘Miguel’, the phrase displayed on the computer screen would read ‘Miguel wins !’. The Spanish equivalent would be: ‘¡ /n player1 /n ha ganado !’ [¡ player1 /n has won !], so the phrase displayed by the computer would read ‘¡ Miguel ha ganado !’

Several issues come to the fore. As is usual when translating from English into Spanish, the translation is longer than the original text, which could pose a problem depending on the number of characters allowed for in the message window. The opening exclamation mark, mandatory in Spanish, needs to be incorporated into the final text, and the tense of the verb has to be changed from the present to the present perfect in order to make it sound more natural and to coincide with what Spanish players would say in a similar context. This example shows how the easiest variable that corresponds
perfectly with English grammatical rules might prove not to be as straightforward when it is transferred into another language.

In games, variables may be used when dealing with substantives which, when they function as the subject of the sentence, influence the conjugation of the verb as well as any other words that need to agree with them. If the game code does not take into account the grammar of the languages covered by the project, many mistakes will occur. Although mistakes are often attributed to translators, they may actually derive from the difficulties encountered by computers when having to deal with the complex and varying grammar of natural languages (Heimburg 2003: online), and this can happen even when they have been programmed correctly. Strategy games, for instance, allow players to choose among different nations that they may want to conquer. When someone attacks the player, a message says: ‘/n nameofnation /n is attacking you !’. Names of nations change widely from one language to another: they may carry an article or not; they can be singular or plural; or they can be masculine, feminine or neuter. Therefore, translators not only have to take care concerning the syntax of the sentence and the possible relocation of the variable, but they also need to be aware of how these variable formulae may trigger potential morphology changes affecting each linguistic item. The previous formula could generate the Spanish sentence ‘¡ Francia te está atacando !’ [France is attacking you!], but it could also produce the ungrammatical ‘¡ Los Países Bajos te está atacando !’ [Netherlands is attacking you!] because, in Spanish, a plural subject requires a plural verb and the correct sentence should be ‘¡ Los Países Bajos te están atacando !’.

Whenever possible, programmers and designers opt for rephrasing to avoid grammatical hurdles. The above message, for example, could be rewritten in the passive voice as ‘You are being attacked by /n nameofnation /n !’ [Estás siendo atacado por /n nameofnation /n !], but this might not be an option for all target languages because of the different usage of the passive voice. There is also the danger that if the toponyms list from which the formula draws is a basic glossary containing equations such as ‘Netherlands = Países Bajos’ or ‘Brussels = Bruselas’, incorrect sentences may result because, in Spanish, some country names need an article when they appear as part of a sentence, but not when they are part of a list, or because some names are singular despite finishing in ‘-s’, exceptions that have to be programmed into computers if the formulae for translation depend on a relative degree of automation.

This type of linguistic issue has prompted the software industry to explore other more efficient and less expensive options in order to deal with translations, causing the concept of controlled language to be born (White 2000).

A controlled language is obtained by restricting the grammar and vocabulary in order to reduce ambiguity and complexity (O’Brien 2003). Traditionally, controlled languages fall into two major types: those that improve readability for children or non-native speakers, and those that enable realisable computer processing such as in the case of automatic translation. This approach is supposed to facilitate automatic translation and speed up part of the process.
(Nyberg and Mitamura 2003: 245–81). However, it relies on the simplification of syntax and lexis and, while the use of controlled language may benefit the production and translation of utility software, it is certainly less appropriate when dealing with entertainment products where creative language usually plays a greater role than in instructive or technical texts (Quah 2006). Arguably, video games could perhaps make use of controlled language for their instruction manuals and basic documentation, but not 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.

**Concatenated** strings with several variables are sometimes used, pushing the boundaries of the capabilities of linguistic variables to their utmost. In the *Guitar Hero* series (Harmonix 2006–present), for instance, players are presented with feedback in the form of original newspaper headlines directly related to the number of points accumulated at a given stage of the game. After each concert challenge, the player is presented with the cover of a newspaper with headlines such as, for example, ‘Outstanding show from Cyclone at the Garden!’ (Figure 4.26). The coded string needed to generate this sentence has four variables: ‘<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. The resulting headlines will then vary according to the quality of the player’s performance and, depending on the success level attained at the end of certain tasks, the headline will be phrased in the form of the rather neutral ‘Decent effort from Mongoozer2k at Rock City Theatre’ (Figure 4.26, left) or the hyperbolic ‘Outstanding show from Cyclone at the Garden’, (Figure 4.26, right).

This type of formula works relatively well for analytic languages such as English. However, this approach is prone to error when dealing with inflected languages such as Spanish, due to the degree of concordance required between the articles, pronouns, substantives, adjectives and verbs in a sentence. In the examples cited above, the most problematic part relates

![Figure 4.26 Variables in linguistic formulae create varied headlines based on player performance.](https://example.com/dailypose.png)
to the adjective and noun combination at the beginning of the sentence. ‘Show’, ‘effort’, and ‘performance’ would normally be translated into Spanish as espectáculo, esfuerzo and actuación, respectively. However, actuación is feminine while the other two nouns are masculine, which means that the adjective will need to change from único to única in order to agree with the substantive. If the game code does not take these morphological and syntactical agreement rules into account, translators will need to limit their options to one gender and one number, which would result in an obviously unnatural discourse. The other option would be to resort to controlled language, reducing lexical variety and choosing terms which share only one gender, a solution which would not only be restrictive and semantically inappropriate, but also inadequate from the point of view of the game experience intended by the game designers.

Of course, game programmers do not need to know the grammatical intricacies of all the languages into which the game is to be translated, but the whole process would benefit from an initial awareness of the various complexities related to working with natural languages and their inclusion in the game code. When discussing the use of variables in video games, Heimburg (2006: 136) notes that “people don’t even notice when the grammar is good, but they certainly notice when the grammar is bad”. Indeed, many players are able to detect an incorrect use of language in the text of a (translated) game, and sloppy localisation bugs can result in a negative impression of the game. This, in turn, can hinder the sales and even loyalty to the brand, leaving many potential buyers inclined to make their purchases in second-hand game shops, or even prompting them to acquire a copy on the black market (Díaz-Montón 2011).

One of the latest developments in this particular area has been the design of grammar engines for MMORPGs. According to Mitchell (2007), a grammar engine is an adaptive translation system that can process nouns and adjectives modifying their number and gender; conjugate verbs; assign correct articles, pronouns and prepositions and keep track of which characters are speaking and listening at all times. Game writers and designers employ metadata tags in their writing in order to direct the grammar engine to the value of these linguistic variables regarding the gender or profession of a player, enabling the game to adapt to that player and his/her choices. This will be explained in the following paragraphs.

In any grammatically correct sentence, subject and object pronouns are closely interlinked, and a successful grammar engine must be able to deal with these relationships in order to guarantee grammatical consistency in the target language. To achieve this goal, grammar engines work with macros, single computer instructions that stand for a sequence of operations, for all the deictic information referring to players and their choices. The example provided illustrates how grammar engines deal with the translation of sentences where the third-person singular can be used as a subject (he, she, it) and as an object pronoun (him, her).
4.3.3.1 Subject Pronoun Macro for Third-person Singular: #he()

This macro generates ‘he’, ‘she’ (or even ‘it’ although rarely), depending on the gender of the character the player has chosen. In the string: ‘Don’t wait for [*player*] because #he([*player*]) is away’

#he([*player*]) will yield ‘he’ if the player is masculine, ‘she’ if the player is feminine, and ‘it’ if the player is neuter. Thus, if the player is ‘Miguel’ the grammar engine will produce ‘Don’t wait for Miguel because he is away’, and if the player is ‘Mary’ it will generate ‘Don’t wait for Mary because she is away’.

4.3.3.2 Object Pronoun Macro for Third-person Singular: #him()

This macro generates ‘him’, ‘her’ or ‘it’ depending on the gender of the noun to which it refers. Therefore, ‘Don’t wait for #him([*player*]) because #he([*player*]) is away’ would generate: ‘Don’t wait for him because he is away’, ‘Don’t wait for her because she is away’, and ‘Don’t wait for it because it is away’, depending on whether the [*player*] is Miguel, Mary or a robot guard for example. Figure 4.27 shows some in-game examples from Free Realms of how these macros generate completely natural sentences (underlined) once the grammar engine has been fed with the right information for each variable for each character. This means that companies do not have to translate all the various options every single time because the localisation grammar engine can render the correct sentence.

The pronoun and gender macros that allow the grammar engine to render complex sentences, such as the ones in Figure 4.27, are embedded in the actual dialogue text, and writers and translators need to learn how to deal with them. The following are the US English, French and Spanish sentences and macros that make the game deal with grammar automatically on the fly:

**US English:** Wow . . . I’m certainly impressed. Everyone, take a good look at [*player*]! There’s no doubt in my mind that #he([*player*])’ll be famous! <BR><BR>I’m honored to pass on my hero card to #him([*player*]).

**French:** Wah . . . Je suis très #ms(‘impressionné’)#fs(‘impressionnée”). Que tout le monde regarde bien [*player*] ! Je suis #ms(‘convaincu’)

![Figure 4.27 Grammar engine taking care of pronouns automatically. Free Realms © Sony Online Entertainment. All rights reserved.](image)
Translation and Localisation in Video Games

Spanish: Guau . . . Estoy #ms('impresionado')#fs('impresionada'). ¡Quedaos con la cara de [*player*]! ¡No tengo la menor duda de que se hará #mp('famoso')#fp('famosa')! Estoy muy #ms('orgulloso')#fs('orgullosa') de darle mi carta de héroe.

Grammar engine rules and formulae are not the most intuitive option or layout used to present the material that needs to be translated, but in terms of speeding up the process by enabling the automation of interactive dialogue, and of reducing the translation bill for multilingual localisation projects, they seem to be a valid solution for the fast-paced video game industry. It therefore comes as no surprise that the idea came from developer companies responsible for creating standalone or online role-playing games, such as Sony Online Entertainment, Square Enix, Blizzard and the like, because of the sheer amount of text with which they have to deal and because they are aware that much of the players’ immersion in the virtual world depends on their being addressed as individuals. Grammar engines are still in their infancy but, over the coming years, there will no doubt be a considerable improvement in the way they interpret grammatical logic as well as in the practical way in which writers and translators will be able to interact with them.

4.3.4 The Localisation of Voice Commands and Gestures

The latest generation of game consoles has pushed previous technology even further by including voice and motion recognition, enabling players to interact with video games by speaking commands into a microphone and making gestures while holding the game controllers or moving in front of the console cameras. The translation of games such as SingStar Dance (Figure 4.28) and Dance Central, a game where players have to move their arms and legs following onscreen prompts, effectively using their bodies as controllers, has presented a new challenge for localisation managers and agencies because of the constant use of movements that could be considered rude in other cultures (Thompson et al. 2011). SingStar Dance is a game designed for Sony’s PlayStation Move which builds on the success of SingStar, its flagship casual gaming series, by introducing the PlayStation Move to the game, allowing up to two people to sing while up to two others dance while holding the PS3 Move controller. The game uses advanced voice recognition technology, together with pitch and rhythm, comparing the players’ performances to that of the original artist of the song. Kinectimals (Figure 4.28), a game where children interact with virtual mascots by simply speaking and gesturing, requires a similar treatment in order to guarantee a smooth operation in the requisite languages. This is not only so for the content of the ‘trigger’ words spoken, acting as the pressing of the control button, but also for the possible utterances from the different players (young and old, male and
female), which prove to be particularly problematic in the case of children’s voices due to the laws concerning childhood labour (Brown 2011: online).

This is because the voice recognition software must include sample voice packages for each of the locales, allowing for some leeway in order to deal with the accents and pitch relevant to each language involved in the release of the game in question (Thompson et al. 2011). Microsoft’s head of Xbox One product development, Albert Penello, explained in August 2003 that there would be a delay in some European countries due to the complexities of localising the voice-control features (Hicks 2003).

Voice recognition software is breaking new ground in language services, localisation and indeed, translation theory, because it is the first time that the need for automating audio-interaction for a mass entertainment product has been so pressing and profitable. Voice recognition has already been used for some years now related to accessibility, for live subtitling of public TV, because of the legal requirements on TV channels to cater for the deaf and hard of hearing community. New technology combined with respeaking research (Romero-Fresco 2011) meant that live programmes can be broadcast with subtitles even if there is an unavoidable delay of three to five seconds and some inaccuracies.

Although voice commands and gestures have only recently come into the arena for video gaming, it is clear that the localisation of games utilising this technology will generate a lot of research concerning translation in the coming years, research which will explore the multichannel nature of communication (written language, spoken language and body language), how it is brought into a ludic multimedia interactive experience and what it means to localise all these communicative elements for other countries.

The first four chapters of this book have focused mainly on the linguistic aspects of the translation of the texts found in video games. Game localisation as an industrial process will be explored in Chapter 5 because of its inevitable impact on translators’ working conditions, in terms of the way

---

Figure 4.28 Voice and gestures used as controllers also need localising (arrows added). SingStar © Sony Computer Entertainment, Kinectimals © Microsoft Corporation. All rights reserved.

---
source and target texts are written, translated and proofread, and in the final quality of the product delivered to players.

RESEARCH PROJECTS

1. What is different in production and distribution terms between the first video games of the 1970s and the latest ones, and how do they affect localisation?

2. Section 4.2 describes video games as a complex, multitextual product but how do they compare to other consumer products such as DIY manuals, children’s toys, home appliances, corporation videos, European Union treaties?

3. Which of the four main characteristics of video game text analysed in Section 4.3 can you find in other products?

4. What are the implications of each of them for the translation and localisation of video games?

NOTES

1. Romhacking is described by Muñoz-Sánchez (2008) as a process in which game fans and hackers get together with the aim of cracking and modifying games belonging to older generations with the intention of localising them, or improving them in any way they see fit, for the benefit of their particular locale or community.

2. Complete video examples can be found in http://facefx.com/content/english-un-declaration-human-rights.


5. The game localisation industry and its processes are elaborated in Chapter 5, as are some of the new strategies and tools which have been designed to improve the non-linearity and version tracking which are common challenges of game translation.

6. Please note that conventions in orthotypography are often ignored in video games in order to have visually impressive text. This is why there is a space between ‘wins’ and the exclamation mark.

7. Please note that conventions in orthotypography are often ignored in video game programming. The capitalisation found in this example represents the personal preference of the programmer, undertaken in order to make the variables stand out for the benefit of all those involved in the programming process.

8. Represented by Nintendo Wii™ (www.nintendo.com/wii), Kinect®Xbox 360 (www.kinect-xbox-360.co.uk) and PlayStation®Move (http://uk.playstation.com/ps3system).
