Essay

The Practical Aspects of Video Game Localization

Ryszard Chojnowski
Albion Localisations, Poland
ryszard.chojnowski@gmail.com

I have nearly 20 years of experience in the field of professional translation and software localization – particularly in the localization of video games. I worked on more than 200 titles, and it would obviously be impossible to list them all here, but the *Diablo* series[1], *Warcraft 3*[2], *Hearthstone*[3], *Baldur's Gate 2*[4], *Neverwinter Nights*[5], *Dragon Age*[6], and *Battlefield: Bad Company 2*[7] are certainly worth mentioning.

The terms "translation and localization" have been used deliberately, because they are neither synonymous, nor interchangeable. Localization involves the process of translation from the source language to target language, but also many other elements. Theoretically, it would seem that this should not be a complicated process, but reality shows that it is quite the opposite, and that producing a high-quality localization is a path fraught with peril, pitfalls, blind alleys, and unexpected obstacles – and involves a struggle with matter itself. The path may be difficult, but arriving at the destination brings immense satisfaction, and that is what I would like to touch upon in this article dedicated to the practical aspects of video game localization. Inevitably, it is going to be only a brief outline of the problem, presenting the challenges that professionals involved in this field are facing in their work.

In today's world, video games surround us at every turn. Only 30 years ago, they were considered as entertainment aimed at a fairly limited group of computer and new technologies fans, but later on games left their narrow niche and evolved into true electronic entertainment industry, generating more income than Hollywood movies[8]. In addition to standard video games for consoles and computers, there are games meant to be played on mobile devices (smartphones and tablets), or in web browsers, not to mention the increasingly popular board games in all shapes, sizes and types, though

[1] *Diablo* (1996) . Blizzard Entertainment, CD Projekt.
[2] *Warcraft 3* (2002). Blizzard Entertainment, CD Projekt.
[3] *Hearthstone* (2014-2016). Blizzard Entertainment.
[4] *Baldur's Gate 2* (2000). BioWare, CD Projekt.
[5] *Neverwinter Nights* (2002). BioWare, Infogrames, CD Projekt.
[6] *Dragon Age: Origins*. (2009). BioWare, Electronic Arts.
[7] *Battlefield: Bad Company 2*. (2010). DICE, Electronic Arts.
[8] http://www.theguardian.com/technology/gamesblog/2009/sep/27/videogames-hollywood. Retrieved on March 12, 2016.

they have been present on the market for many years now. The process called "gamification" is becoming ubiquitous in more and more aspects of our everyday lives. Gamification employs game design elements meant to make mundane activities more attractive, and is often associated with advertising or marketing campaigns. We receive points for purchases and services, unlock achievements for physical activities tracked by wearable electronic devices, and are able to interact (and compete) in various ways with people from all around the world. All of those factors have resulted in video games growing increasingly popular, emerging from a kind of ghetto to which they were still consigned a couple of years ago, and becoming an extremely influential medium with millions of players who do not necessarily speak the language in which a game was developed.

What is Localization?

The vast majority of games available in Poland have been created in English, but other languages include German, Japanese, and Chinese (China is becoming a more and more important market every year). If we want to fully understand the game, to fully assimilate the message conveyed by its creators, it is necessary for the games to be adapted to the particular needs of Polish customers. This, in turn, necessitates the translation of the entirety of the game's text – and it must be noted that video games are getting more voluminous, often reaching a million words. In comparison, J.R.R. Tolkien's famous novel – *The Lord of the Rings* – is "only" 470,000 words long[1]. But that adaptation to the needs of the market does not merely involve translation. The process is much more complex and encompasses many varied elements, which can be collectively referred to as "localization".

Video games are not the only medium that undergoes the process of localization. Generally speaking, websites, software, and other sources that require not only translation of the text itself, but also adaptation of the text to the local market, are localized (taking into consideration a number of aspects: from translation itself, through adaptation, font choice, numerical and date notation, text layout, production of necessary recordings and images, to thorough testing of the finished product). Various elements of localization will be discussed later in the article, but at this point a definition of the notion of localization should be provided. We already know that it is something more than a translation. In order not to complicate matters further, we should adopt the following definition: localization is an adaptation of the product to the linguistic, regional, and technical needs of a given market. The Holy Grail of localization is a situation where the reader does not notice any artificiality in the final text and treats it as if it were originally created in the target language – in short, when a localization is consistent and "transparent" to the end user. (Chojnowski, 2013)

[1] http://loopingworld.com/2009/03/06/wordcount/. Retrieved on March 12, 2016.

Elements and Types of Localization

Accordingly, localization should enable its potential recipient to use the end product (in our case – a video game) correctly and without problems. Usually, we are dealing with a source product created in a very different socio-cultural environment, utilizing other sets of notions than the local products. Today, we live in a "global village" and many (broadly understood) cultural phenomena are instantly recognizable throughout the world (for example: *Star Wars*), but a good number of references remain clear primarily to players from the game's country of origin. Hence the need for adaptation, translocation, and sometimes complete transformation of source material into a form that can be understood by the target audience. Achieving such a state requires a labor-intensive process consisting of many stages (Chojnowski, 2013), such as:

- translation of the source text itself, including its cultural adaptation;
- creation of graphics and multimedia elements (e.g. dubbing);
- preparation of printed materials;
- composition of marketing texts associated with the product, taking into account the uniqueness of the local market;
- preparation of an appropriate font set, in order to preserve the Polish diacritical marks;
- modification of the text – especially important when locating Asian comics (e.g. Japanese mangas), which are read "backwards" and from right to left;
- the use of local standards for the notation of dates, numerical values, etc.;
- through testing of the finished product.

While most of the above points are quite clear, the mention of the need to prepare a set of fonts can be surprising. However, it is often the case that Polish special characters are problematic, because developers do not always remember to implement the "Eastern European" standard of character encoding known as ISO Latin 2. And even if the standard is supported by the software, an appropriate set of fonts still needs to be adjusted to it. Ideally, the appearance of such fonts should correspond closely to the appearance of the original set, encoded in the "Western European" ISO Latin 1 standard. It sometimes happens that the Polish version of a game utilizes system fonts, such as Arial, due to the unavailability of appropriate fonts. Substituting other fonts for the original ones can lead to a variety of unexpected problems. In the case of *Diablo* 2[1], the original version of the game used a font called Exocet. During the localization process, it soon became clear that there was no version of Exocet with Polish diacritics. A replacement was found that, at first glance, seemed to look quite similar, and the

[1] *Diablo 2* (2000). Blizzard Entertainment, CD Projekt

localization team decided to use it. As it belatedly turned out, despite the new font's overall similar appearance to the original, its line spacing was twice as wide as that in Exocet. As a result, the tooltips and descriptions for items in the Polish version of the game often take up the entire screen space or even more. Figure 1 shows the description of the same item in the original version and in Polish. As shown below, even seemingly simple elements can become a source of serious problems.



Fig. 1

Font comparison between Polish and English version of Diablo 2

Now that some of the components related to localization have been presented, let us take a closer look at the individual elements of the process. There are different levels of video game localization that may be undertaken, and thus one may come across the terms ‘full localization’ and ‘partial localization’. In the case of a full localization, all of the game's linguistic assets are rendered into the target language, including text, voiceover (recorded dialogues and narration), graphics and manuals. *StarCraft 2*[1] is an example of such a localization. Quite literally everything in the game has been rendered into the Polish language – including posters on the walls in the virtual world (Fig. 2) and news tickers displayed during TV shows.

[1] *StarCraft 2* (2010). Blizzard Entertainment, LEM



Fig. 2

Localized poster comparison between English and Polish versions of StarCraft

On the other hand, a partial localization involves the translation (and, possibly, adaptation) of the textual layer of the game only, with the exclusion of voiceover and graphics. This type of localization is much more common due to lower costs and lighter workload involved. Full localizations are far costlier. Recording the voiceover alone can be multiple times as expensive as the translation of the text. The general assumptions behind the localization process, however, remain the same when it comes to both types of localization mentioned before.

The Localization Process

Market development and the need for simultaneous release of games in various countries have strongly influenced the process of game localization, which is different today from what it used to be several years ago, when localization teams would work on a finished product, which had already been released in Western markets. Then, the game would first be delivered to programmers, who data-mined it for all the necessary materials, such as the texts, images and audio files, as well as fonts, and then the localization team would begin their work on the Polish version. By that time, they would be familiar with the full version of the game and use it as a model and a source of contextual information. Often, the work would last several months, and the Polish-language version of the game would be released long after the original's release date. For example, the global release of *Planescape: Torment*[1] was December 10, 1999, while in Poland the game was published on May 31, 2000 – almost half a year after the English version.

[1] *Planescape: Torment* (1999 [2000]). Black Isle, Interplay, CD Projekt

Currently, the entire localization process usually begins well before the release of the game. Sometimes, it can be as early as ten or more months beforehand. The first step to be taken by the localization team when a localization gets the green light is to familiarize themselves with the source product. It did not use to be a problem in the past, when localizations were based on finished games. Nowadays, access to an early version of a game in development is nearly impossible. That is the case mainly for security reasons – the developers do not wish to share any information concerning the game for fear of data leakage, which may have extremely negative commercial results. Therefore, localization teams are often affected by the lack of contextual information – in fact, they receive very limited information regarding the project they are working on. Thus, when we compare game localization process to the process of translating a book, the fundamental difference is that the translator of a book is dealing with an already finished product, and is often in direct contact with the author – something members of localization teams can only dream about most of the time. As a result, the first step in the process of localization is, in fact, reading the text provided, but without accessing the game itself. It could be compared to reading the subtitles for a movie without actually watching it.

The next step is to create a so-called loc-kit, or localization kit. Ideally, it would include all the necessary materials from the game (texts, sounds, textures, images, fonts, background information), supplemented by glossaries and style guides need to localize the product – all collected in a single package. Unfortunately, as a rule, the localization team can only count on being provided with texts from the game, often without any additional information that would facilitate their work, which tends to lead to a number of complications during translation (this will be touched upon later in the article).

After receiving texts from the game's developers, it is advisable to start with the creation of glossaries and style guides, especially if several people are involved in the localization project. Nowadays, game localization is so complex a process that the situations where one translator works on creating the entire localized product have become increasingly rare. Most often, there are teams made up of translators, editors, graphic artists, recording specialists, and programmers. That is why such a strong emphasis is placed on the use of uniform terminology by all members of the team.

Having laid the foundations for glossaries and style guides to be used during the translation of particular elements of the game, the most important part of work can commence – the translation itself. It takes up to several months, especially considering the fact that the source texts can be modified all the time by the developers, and are susceptible to change almost until the end of the game development process.

If we are dealing with a full localization (i.e. text, voiceover and graphics are localized), the translation of the initial batch of dialogue script is usually followed by the beginning of recording sessions. At the recording stage, the translation is subject to additional changes and to so-called "dialogization", which involves making the spoken

lines sound more natural, synchronizing the timing with the original, and matching the lip movements of characters in the game with new sounds they produce (lip sync, or lip synchronization).

While the translation and recording of texts from the game proceed, translation of graphic elements also takes place, wherein it is necessary to ensure that the target text is as close to the original as possible in terms of the number of characters. Graphic elements are usually subject to strict restrictions on the amount of space they can take up. Since messages in English are usually quite succinct, the translation must also reflect that quality, sometimes at the cost of a slight change in the meaning. For example, in the game *Heroes of the Storm*[1] a small graphical element – doormats – was added by the developers at the doors of houses. Each doormat was decorated with a caption: the first read "WELCOME", the second "KEEP OUT". While the Polish word "WITAJ" is shorter than its English counterpart, and can be easily used, we would encounter issues if we wanted to translate "KEEP OUT" as "NIE PODCHODŹ" or "TRZYMAJ SIĘ Z DALA", which theoretically are the correct equivalents but exceed the character limit. Hence, the idea to use the slightly frivolous "SPADAJ" (Eng. "GET LOST"), which is a perfectly suitable choice considering the game's register (Fig. 3)



Fig. 3

Localized graphical elements in Heroes of the Storm

During the translation process, finished texts are regularly sent back to the developers, who implement them in the game and, whenever possible, provide a partially localized version for testing. Currently, testing is usually performed by specialized external teams whose task is to find all the errors, and then submit appropriate reports to the translators, requesting any necessary corrections. Sometimes, the translators themselves also become testers, and since they are already familiar with the text, they are far more likely to notice any errors and shortcomings. The situation is even better

[1] *Heroes of the Storm*. (2015-2016). Blizzard Entertainment, CD Projekt.

when the authors make the tools for implementing translated elements in the game available to the localization team, so that the quality of localization can be continuously monitored in the target environment.

The Tools

Of course, work on modern technologies requires the use of appropriate tools. It's hard to work with video games without the assistance of computers and specialized software, which greatly facilitate the process, and sometimes are the only reason localization is possible at all. Gone are the times of analogue translation. Today, the translator's work is performed entirely within the digital environment. Although it would seem that the basic translation tools are word processors and editors, that is not always the case. As far as video games are concerned, a portion of the source material is, in fact, stored as text in TXT or DOC formats, which correspond to 'clean' text files or MS Word files, but most often the developers' preferred method of text storage for games are markup languages such as XML[1] (Extensible Markup Language), which enable the presentation of data in a structured way. XML is a language that is platform-independent and very often it is used to create webpages or databases containing text strings from the translated applications or software.

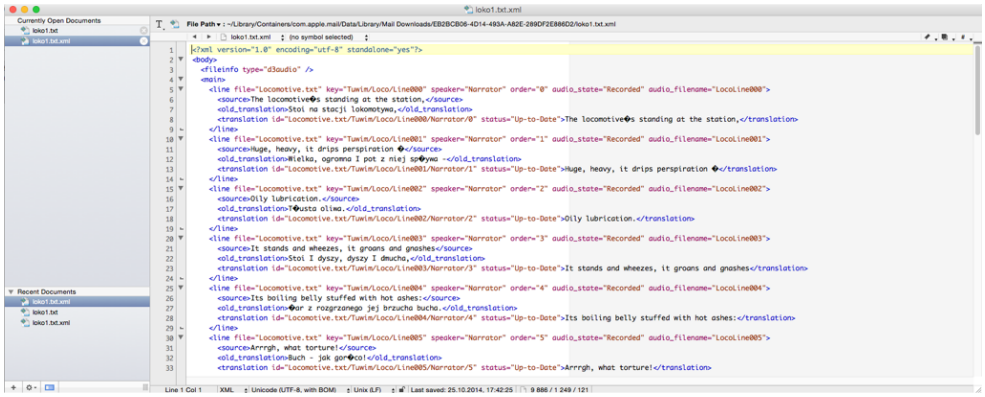


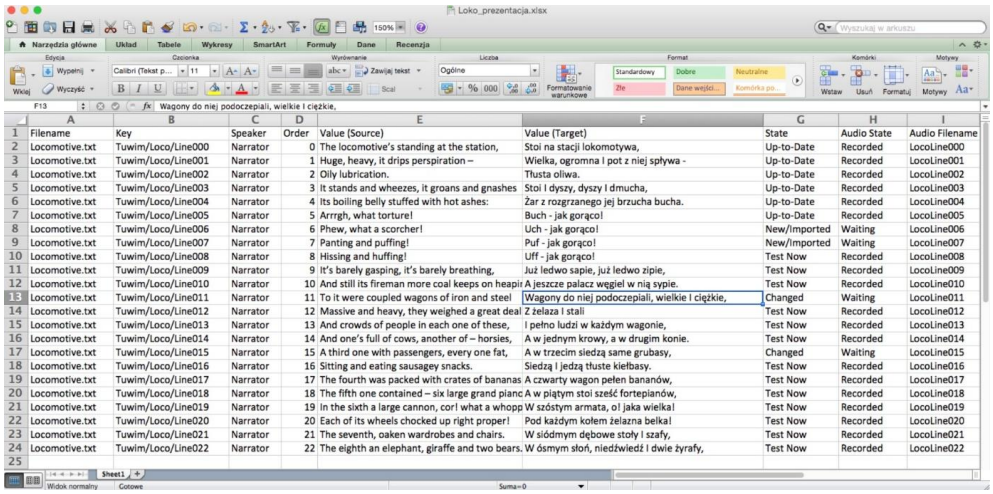
Fig. 4

Sample text from a game stored in XML format

Text stored in this format (as seen in Fig. 4) is not easy to translate because of the huge number of various markers and the need to preserve the characteristic syntax. As such, text is often submitted for translation as an XLS spreadsheet. This type of spreadsheet

[1] XML Reference - http://www.w3.org/XML/

can take the form of a very simple two-column table with the source text and space provided for a translation. But sometimes the spreadsheet contains also context-sensitive information (Fig. 5), for instance: the number and gender of the speakers or their tone of voice, a character's background, a plot synopsis, or a description of variables.



	A	B	C	D	E	F	G	H	I
1	Filename	Key	Speaker	Order	Value (Source)	Value (Target)	State	Audio State	Audio Filename
2	Locomotive.txt	Tuwim/Loco/Line000	Narrator	0	The locomotive's standing at the station,	Stoi na stacji lokomotywa,	Up-to-Date	Recorded	LocoLine000
3	Locomotive.txt	Tuwim/Loco/Line001	Narrator	1	Huge, heavy, it drips perspiration –	Wielka, ogromna I pot z niej spływa -	Up-to-Date	Recorded	LocoLine001
4	Locomotive.txt	Tuwim/Loco/Line002	Narrator	2	Oily lubrication.	Tłusta oliwa.	Up-to-Date	Recorded	LocoLine002
5	Locomotive.txt	Tuwim/Loco/Line003	Narrator	3	It stands and wheezes, it groans and gnashes	Stoi I dyszy, dyszy I dmucha,	Up-to-Date	Recorded	LocoLine003
6	Locomotive.txt	Tuwim/Loco/Line004	Narrator	4	Its boiling belly stuffed with hot ashes:	Żar z rozgrzanego jej brzucha bucha.	Up-to-Date	Recorded	LocoLine004
7	Locomotive.txt	Tuwim/Loco/Line005	Narrator	5	Arrrgh, what torture!	Buch - jak gorąco!	Up-to-Date	Recorded	LocoLine005
8	Locomotive.txt	Tuwim/Loco/Line006	Narrator	6	Phew, what a scorcher!	Uch - jak gorąco!	New/Imported	Waiting	LocoLine006
9	Locomotive.txt	Tuwim/Loco/Line007	Narrator	7	Panting and puffing!	Puf - jak gorąco!	New/Imported	Waiting	LocoLine007
10	Locomotive.txt	Tuwim/Loco/Line008	Narrator	8	Hissing and huffing!	Uff - jak gorąco!	Test Now	Recorded	LocoLine008
11	Locomotive.txt	Tuwim/Loco/Line009	Narrator	9	It's barely gasping, it's barely breathing,	Już ledwo sapie, już ledwo zipie,	Test Now	Recorded	LocoLine009
12	Locomotive.txt	Tuwim/Loco/Line010	Narrator	10	And still its fireman more coal keeps on heapir	A jeszcze palacz węgiel w nią sypie.	Test Now	Recorded	LocoLine010
13	Locomotive.txt	Tuwim/Loco/Line011	Narrator	11	To it were coupled wagons of iron and steel	Wagony do niej podoczepiali, wielkie I ciężkie,	Changed	Waiting	LocoLine011
14	Locomotive.txt	Tuwim/Loco/Line012	Narrator	12	Massive and heavy, they weighed a great deal	Z żelaza I stali	Test Now	Recorded	LocoLine012
15	Locomotive.txt	Tuwim/Loco/Line013	Narrator	13	And crowds of people in each one of these,	I pełno ludzi w każdym wagonie,	Test Now	Recorded	LocoLine013
16	Locomotive.txt	Tuwim/Loco/Line014	Narrator	14	And one's full of cows, another of – horsies,	A w jednym krowy, a w drugim konie.	Test Now	Recorded	LocoLine014
17	Locomotive.txt	Tuwim/Loco/Line015	Narrator	15	A third one with passengers, every one fat,	A w trzecim siedzą same grubasy,	Changed	Waiting	LocoLine015
18	Locomotive.txt	Tuwim/Loco/Line016	Narrator	16	Sitting and eating sausagey snacks.	Siedzą I jedzą tłuste kiełbasy.	Test Now	Recorded	LocoLine016
19	Locomotive.txt	Tuwim/Loco/Line017	Narrator	17	The fourth was packed with crates of bananas	A czwarty wagon pełen bananów,	Test Now	Recorded	LocoLine017
20	Locomotive.txt	Tuwim/Loco/Line018	Narrator	18	The fifth one contained – six large grand pianc	A w piątym stoi sześć fortepianów,	Test Now	Recorded	LocoLine018
21	Locomotive.txt	Tuwim/Loco/Line019	Narrator	19	In the sixth a large cannon, cor! what a whopp	W szóstym armata, o! jaka wielka!	Test Now	Recorded	LocoLine019
22	Locomotive.txt	Tuwim/Loco/Line020	Narrator	20	Each of its wheels chocked up right proper!	Pod każdym kołem żelazna belka!	Test Now	Recorded	LocoLine020
23	Locomotive.txt	Tuwim/Loco/Line021	Narrator	21	The seventh, oaken wardrobes and chairs.	W siódmym dębowe stoły I szafy,	Test Now	Recorded	LocoLine021
24	Locomotive.txt	Tuwim/Loco/Line022	Narrator	22	The eighth an elephant, giraffe and two bears.	W ósmym słoń, niedźwiedź I dwie żyrafy,	Test Now	Recorded	LocoLine022
25									

Fig. 5

The same text as an XLS spreadsheet

Once the translator has at their disposal the "workable" source material, they can finally put their hand to the metaphorical (or rather – digital) plough, using standard office software, such as Word or Excel. However, their job can be made a little easier. In recent years, specialized software called CAT tools (Computer-assisted translation) has grown in importance, popularity, and functionality[1]. There are many such programs, each of which offers a slightly different approach to the issue of translation, with a rather similar general approach in terms of their methods of operation. Such software makes it possible to import the source file, divides the text into smaller segments the size of which depends on the user – those can be paragraphs, sentences, and even single words – and stores all finished translations in the so-called translation memory (TM). That memory is growing with each entry, and if the translator encounters a passage that is similar to or identical with what is stored in the memory, the program

[1] memoQ (2007-2016) Kilgray, https://www.memoq.com; SDL Trados (2000-2016) SDL Trados, http://www.translationzone.com/trados.html; XTM (2002-2016) XTM-International, http://xtm-intl.com.

suggests the existing translation. This is especially useful when it comes to translating user interfaces, technical descriptions, ways in which skills work etc. What is more, a good CAT tool enables the storage in the TM of several translations for one segment, so that the translator has options when they wish to diversify different characters' lines. Even when dealing with texts that contain few repetitions, CAT programs allow for a clear segmentation of fragments (Fig. 6), making the text more comfortable to work with. Additionally, through the use of CAT tools, several translators can simultaneously work on a single translation project, each enjoying access to databases created by others, and providing one another with editing remarks, comments, or new items for glossaries.

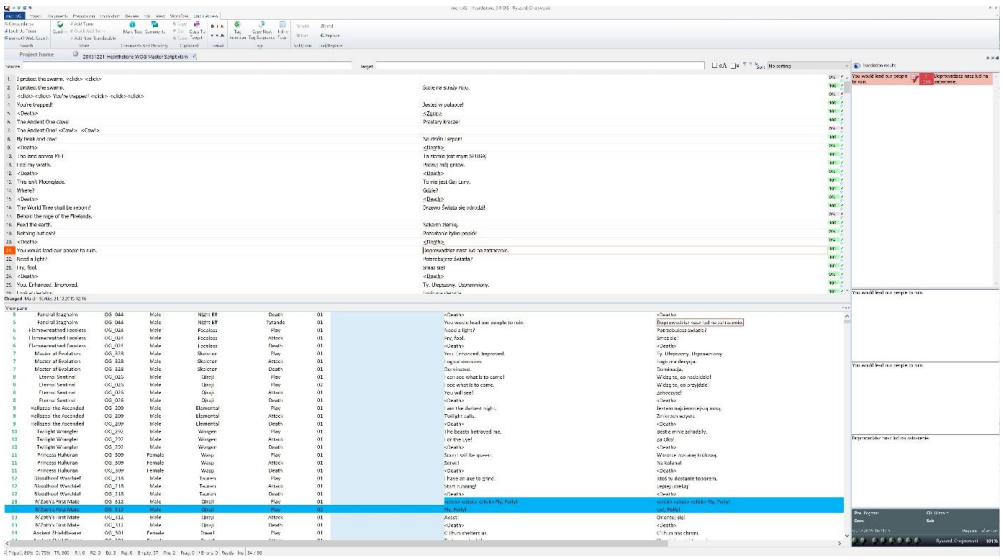


Fig. 6

Sample text presented in memoQ CAT tool

In addition, translation memories are excellent for tracking changes made to the original text, and applying all the necessary updates. Currently, most games are being continuously developed after their release – there are add-ons, upgrades, bug fixes, modifications, etc. Video games, especially those designed for multiple players, are subject to constant changes and improvements, which often impact existing textual components. When all the texts are stored in the translation memory, introducing updates to the translation becomes significantly easier because the program selects pieces that have been changed, removed, or added. There is no need to translate the entire entry from scratch – the translator only applies amendments as necessary. When there are several hundred such updated skills, a CAT took becomes truly indispensable.

After the translation has been completed, it can be assigned for edition (often multiple times) in the same CAT tool. Finally, when the entirety of the text has been translated, proofread, and edited, it is then exported into to the source format. For instance, if we received the text as an XLS spreadsheet, we need to submit it to the customer in the same spreadsheet – of course, supplemented with the translation. CAT programs offer the option to export and validate the text, thanks to which a clean, complete and error-free file can be compiled. The file is then sent back to the customer.

In addition, teams of translators often develop their own tools for specialized tasks, such as text converters from one format into another. For example, if the game developers provide source texts stored in XML format, which is convenient for them, then – before work on the translation can commence – the text needs to be converted into a more "pliable" format that facilitates segmentation and will be clearer for translators. Most often, the converted text is stored in XLS spreadsheets, then is uploaded to the CAT program, where translation takes place, after which the conversion process is reversed, and ultimately the customer receives a properly compiled file that can be implemented in the game.

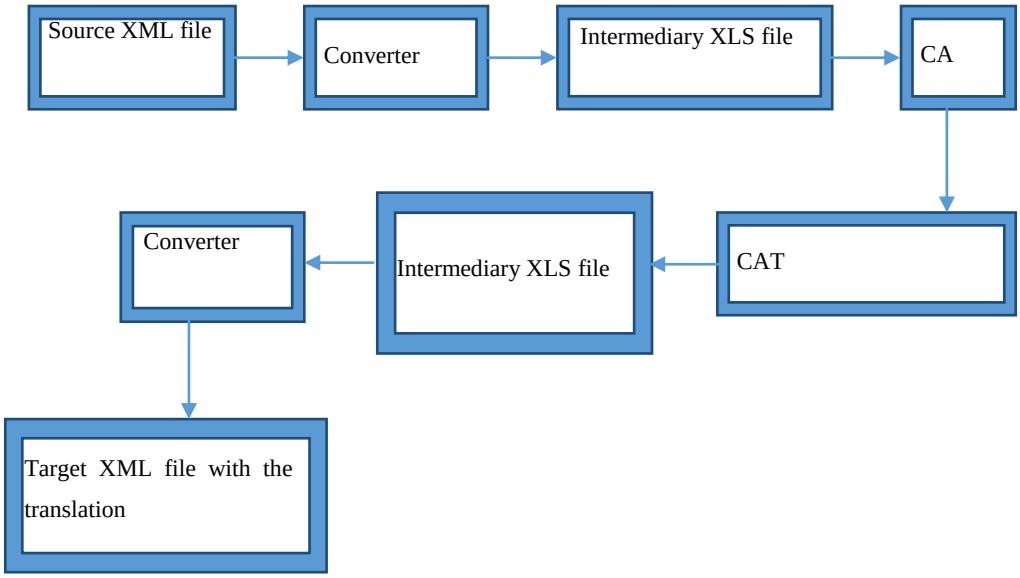


Fig. 7

A diagram showing an outline of the translation process

Sometimes tools for the creation and management of glossaries are also utilized. There are special applications dedicated to that, such as Xbench[1], but this role can be also fulfilled by CAT programs, which allow the user to create and edit glossaries from translation memories. Often, ordinary spreadsheets are used for this purpose (Fig. 8), where the source text is in one column, and next to it we enter its translation, commentary, and the identification of the field to which the entry pertains – for example, the name of skill, the name of location, the type of the item, etc.

English	Polish	Category	Notes	Date changed	Appearance
Soulbound	Przypisany do konta	web			General Glossary
Patch	Aktualizacja	web			General Glossary
PTR	PST	web			General Glossary
Public Test Realm	Publiczny Serwer Testowy	web			General Glossary
Patch Notes	informacje o aktualizacji	web			General Glossary
Friends	znajomi	web			General Glossary
redeem code	zrealizuj kod	web	also: wykorzystaj kod (battle.net account -> Gry i kody -> Dodaj klucz gry -> Wykorzystaj kod)		General Glossary
livestream	transmisja na żywo/stream	web	also: live stream		General Glossary
BattleTag	BattleTag	web			General Glossary
Authenticator	Authenticator	web			General Glossary
Now Available	już dostępny	web			General Glossary
LMB	LPM	in game			General Glossary
RMB	PPM	in game			General Glossary
-	-	in game	ASCII 45. It should be used as connector between two words (i.e. beta–testy).		General Glossary
—	–	in game	ASCII 151. It should be used as punctuation mark (i.e. LOKALIZACJA – MENADŻEROWIE).		General Glossary
"	"	in game	ASCII 34. It should be used only in variables (i.e. <s val="StandardTooltip">).		General Glossary
"	„"	in game	ASCII 132, ASCII 148. Both should be used in every case when quotation marks are needed except in variables.		General Glossary
'	'	in game	ASCII 39. It's a change for Heroes of the Storm for .09 (i.e.: Anub'arak, not Anub'arak).		General Glossary
real money	prawdziwe środki pieniężne	web			General Glossary
real-money	prawdziwe środki pieniężne	web			General Glossary
Offline	Niedostępny	web			General Glossary
Closed Beta	Zamknięte beta-testy	web	ASCII 45 should be used. Changed from "Welcome to Closed Beta"/Witamy w zamkniętych	20150301	General Glossary
New User Experience	Nowe wrażenia z rozgrywki	web			General Glossary
New Content	Nowa zawartość	web			General Glossary
Veteran Edition	Edycja Weterana	web	Added on 20150203		General Glossary
New features	Nowe funkcje	web	Added on 20150203		General Glossary
Balance	Stan konta	web	Added on 20150213		General Glossary
Privacy Policy	Polityka prywatności	web	Added on 20150203		General Glossary
Terms of Sale	Warunki Sprzedaży	web	Added on 20150213		General Glossary
Demo	wersja demonstracyjna	web	Added on 20150223		General Glossary
Accept	Akceptuję	web	UI		General Glossary
Cancel	Anuluj	web	account		General Glossary
Account	konto	web	account		General Glossary
Account in good standing	konto o nieposzlakowanej opinii	web	account		General Glossary

Fig. 8

Example of a glossary window

Types of Text

Those not familiar with the topic might think that there is not too much text to be translated in video games – that the only things to be found there are score values, the amount of energy or magic at the player's disposal, etc. However, as I mentioned in the introduction, there are often games that contain more than a million words. There are also games like the *World of Warcraft*[2], the textual components of which amount to more than 6 million words (mostly due to the fact that the game has been developed continuously for over 10 years now)[3]. This begs the question – what kind of texts can we find in video games? The answer is simple: all of them. The following are some examples of different types of texts found in games:

[1] XBench (2015) ApSIC, S.L., http://www.xbench.net/
[2] *World of Warcraft* (2004-2016). Blizzard Entertainment, CD Projekt
[3] http://us.battle.net/wow/en/blog/12346804/world-of-warcraft-azeroth-by-the-numbers-1-28-2014. Retrieved March 12, 2016.

- Dialogues – they appear in almost every game and, depending on the subject matter and the interlocutors, may concern virtually everything and make use of every register: from solemn speeches, through sublime discussions of aristocrats, to vulgar and coarse taunts of street urchins.

- One-liners – texts of this kind include all brief statements that are not dialogues (e.g. battle cries, comments uttered by the characters, jokes, taunts, warnings or requests). Their characteristic feature is that they must usually be appropriate for many speakers and situations, because video games quite often make use of recycled lines. Developers frequently assume that it is unnecessary to write lines for multiple characters, since one line can be used by a few speakers. This practice definitely saves time, which is extremely valuable during the creative process, but it also causes a lot of problems during translation, which will be discussed later in the article.

- Interface – all the elements of the game aimed at communication with the player – descriptions of the options, settings, on-screen prompts and notifications, and much more. Such texts often have a character limit, since it is strictly defined where they appear, and the text cannot exceed a set number of characters.

- Names – a very broad category, which includes both geographic names, character names, items, areas, and all the objects appearing in the game. It often happens that translation of names requires the creation of neologisms and application of various word-formation processes, especially when in the original we have to deal with adjective-and-noun clusters and complex nouns, such as *Stormwind, Ramshield* or *Puzzleworth*, etc.

- Descriptions – again, a very broad category, which may include, for example, skills, instructions, information on quests, descriptions of objects, places, characters, enemies, as well as bestiaries and databases. Depending on the type of information described, it is necessary to use a different register for the translation – a completely different approach should be taken to the descriptions of combat units than to ancient tomes of magical spells.

- Journals, diaries, logs, books and short stories (no, this is not a mistake on the Author's part) – in video games one can often find journals penned by various characters, as well as literary works that can be both rooted in the fantasy world and famous among its denizens. Therefore, the translator is dealing with a sort of Chinese-box storytelling: they translate the game, constituting in itself a story, which also includes other stories. For instance, in the games from *The Elder Scrolls*[1] series there are approximately 100 various volumes of stories that can be found and read by the player [19].

[1] *Elder Scrolls, The: Skyrim* (2011). Bethesda Studios, Cenega; *Elder Scrolls, The: Oblivion* (2006). Bethesda Studios, Cenega.

- Songs, poems, and rhymes – those appear most often in adventure and role-playing games. For example, in the *Bard's Tale* [20] there are a dozen different songs. Many other games feature various rhyming riddles, and even poems.

- System messages – texts that are not part of the interface, which inform the user of potential errors, updates, progress in the downloading of new data, the quality of Internet connection, etc. Often, such messages must fit a well-defined form, depending on the requirements of hardware manufacturers. The slightest deviation from the designated templates can lead to the rejection of the translation, and even delay the release of the game. Hence the need for the use of corporate glossaries and style guides during the translation of such texts. However, even with strict adherence to the guidelines, problems may still arise. As an example, let us have a look at the translation of the word "friend" in Polish. According to the guidelines of Sony, in the case of system messages the word should be translated as "przyjaciel". And according to the guidelines of Microsoft, the correct translation is "znajomy". The word occurs in many different messages and terms, such as the "friend list", "add friend", "remove friend", etc. When translating those expressions as "lista znajomych", "dodaj znajomego", and "usuń znajomego", we are acting in accordance with the requirements of Microsoft, but contrary to the requirements of Sony, which may delay the release of the game on the consoles of the latter company. In such a case, the game's developers need to prepare text strings that contain the word "friend" separately for the consoles of both companies, allowing the translators to provide two different translations of the term and thus meet the guidelines of both manufacturers.

- Internal term databases – such as dictionaries of profanities, designed to filter out such content from messages typed by players, and censor them by replacing parts of words (e.g. with asterisks: k***a [f**k]). Interestingly, these databases also contain misspelled words, taking into account the fact that potential users may not be able to spell correctly or want to bypass the profanity filter by intentionally misspelling words. In principle, creating such glossaries is not translation as such, since profanities in different languages are not translatable as one-to-one counterparts. The source texts then become a kind of model, based on which the translator creates a set of phrases that are similar in meaning in the target language. And as the colloquial language is constantly undergoing changes, it is also necessary to update such glossaries with new terms.

- Legal information – a game is not just a source of entertainment. This much is clear when one attempts to read different (and usually verbose) end-user license agreements, and various terms and conditions for the provision of services included with the software, as well as copyright information, and other similar legal documents. Those are standard legal texts, which also must be translated according to the appropriate templates.

If games contain so many wildly different texts, it becomes necessary for the translator to compile and use as large a database of source materials as possible, and have at their disposal any translation aids that would facilitate the process of translation and preparation of the target language version. Video games developed nowadays concern all areas of life and entertainment – hence it would be advisable for a translator to have wide interests and be accustomed to using sources.

Source Materials

First of all, the primary source materials are always other games, which makes it necessary for video game translators to also be active players. Knowledge of video games is extremely helpful when translating the user interface and system messages, as well as any situations where the translator is faced with the lack of context. For example: knowing what randomly generated item names look like in *Diablo 2*[1] can help understand the mechanisms behind the creation of such names in the next installment of the series. Having played strategy games before can make it easier for a translator to understand the concepts of the 'fog of war', or 'rush' and so on.

Another very important source material is broadly understood culture, especially popular culture, because of the ubiquitous references to various films, TV shows, theater, music, books, graphic novels, other video games, and even Internet memes. In many games one can also come across the so-called *Easter eggs* – hidden jokes, messages, and references to other works, the knowledge of which is useful in the process of translation. For example, in the game *Diablo 3* [22] the player can find pages from a log, which is a reference to the television drama series *Lost*, and after arranging them in the correct order, it turns out that the first letters of each of the entries form the word 'DHARMA', which is the name of one of the organizations in the series. Had the translator not been familiar with *Lost*, they might overlook this hidden joke, which in turn would impoverish the translation.

When discussing the source materials, we cannot overlook specialist literature from various fields, as well as dictionaries and encyclopedias. Many reference materials are currently available in the digital form on various websites, which definitely makes the search for information much easier. The Web can also be a veritable treasure trove of various dictionaries and thesauri, which are constantly updated and expanded with new entries. Particularly useful are the dictionaries of slang, idioms, and colloquial speech, like *www.urbandictionary.com* . When all else fails, and the translator comes across a term that escapes any classification, assistance may be provided by the authors themselves, who sometimes do respond to questions submitted by the localization team. Above all, online search engines might prove to be true life-savers. The Internet is unquestionably a gold mine of information, so mastering the art of the efficient use

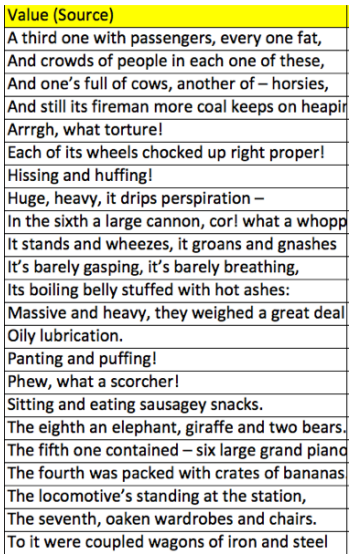
[1] *Diablo 2* (2000). Blizzard Entertainment, CD Projekt.

of search engines is now becoming one of the most important skills in the translator's research toolbox.

Real Life Examples

The Order of Things

After this insight into the localization process, let us move on to more practical considerations and look at the potential difficulties and traps that await translators during the translation of a video game. The first potential pitfall might materialize itself immediately upon receiving the files for translation. Sometimes, the localization package consists of an XLS spreadsheet containing individual phrases from the game, one after another, like in the figure below:



Value (Source)
A third one with passengers, every one fat,
And crowds of people in each one of these,
And one's full of cows, another of – horsies,
And still its fireman more coal keeps on heapir
Arrrgh, what torture!
Each of its wheels chocked up right proper!
Hissing and huffing!
Huge, heavy, it drips perspiration –
In the sixth a large cannon, cor! what a whopp
It stands and wheezes, it groans and gnashes
It's barely gasping, it's barely breathing,
Its boiling belly stuffed with hot ashes:
Massive and heavy, they weighed a great deal
Oily lubrication.
Panting and puffing!
Phew, what a scorcher!
Sitting and eating sausagey snacks.
The eighth an elephant, giraffe and two bears.
The fifth one contained – six large grand piano
The fourth was packed with crates of bananas
The locomotive's standing at the station,
The seventh, oaken wardrobes and chairs.
To it were coupled wagons of iron and steel

Figure 9.

An example of a XLS spreadsheet

The problem is that someone sorted the strings alphabetically, thus making it impossible to determine the order in which they appear in-game, or to understand what their broader meaning and context is. What is missing is some sort of a string ID, which would indicate the element to which a particular line of text refers, and where it should be located. This particular situation actually occurred during the translation of the game *Enter the Matrix* [1]. The lines were sorted alphabetically – a few thousand

[1] *Enter the Matrix* (2003). Atari. CD Projekt

segments of text arranged in such a way that attempts at their correct identification and translation were utterly futile. Let us imagine any book submitted for translation in such a state. Translation is rendered practically impossible when the original layout of the text is disturbed.

It therefore becomes clear that the textual layer of the game cannot be simply uploaded anywhere into it. Texts become an integral part of the software, and as such must be assigned to the relevant items in the database and have their own IDs which, in addition to making it possible to determine the order of strings or the place where the text is to be displayed, often carry with them the much-needed context information.

Let us have a look at the same text with added information about the identifier of each string.

Key	Value (Source)
Tuwim/Loco/Line000	The locomotive's standing at the station,
Tuwim/Loco/Line001	Huge, heavy, it drips perspiration –
Tuwim/Loco/Line002	Oily lubrication.
Tuwim/Loco/Line003	It stands and wheezes, it groans and gnashes
Tuwim/Loco/Line004	Its boiling belly stuffed with hot ashes:
Tuwim/Loco/Line005	Arrrgh, what torture!
Tuwim/Loco/Line006	Phew, what a scorcher!
Tuwim/Loco/Line007	Panting and puffing!
Tuwim/Loco/Line008	Hissing and huffing!
Tuwim/Loco/Line009	It's barely gasping, it's barely breathing,
Tuwim/Loco/Line010	And still its fireman more coal keeps on heapir
Tuwim/Loco/Line011	To it were coupled wagons of iron and steel
Tuwim/Loco/Line012	Massive and heavy, they weighed a great deal
Tuwim/Loco/Line013	And crowds of people in each one of these,
Tuwim/Loco/Line014	And one's full of cows, another of – horsies,
Tuwim/Loco/Line015	A third one with passengers, every one fat,
Tuwim/Loco/Line016	Sitting and eating sausagey snacks.
Tuwim/Loco/Line017	The fourth was packed with crates of bananas
Tuwim/Loco/Line018	The fifth one contained – six large grand pianc
Tuwim/Loco/Line019	In the sixth a large cannon, cor! what a whopp
Tuwim/Loco/Line020	Each of its wheels chocked up right proper!
Tuwim/Loco/Line021	The seventh, oaken wardrobes and chairs.
Tuwim/Loco/Line022	The eighth an elephant, giraffe and two bears.

Fig.10

An example of a XLS spreadsheet with line identifiers

Everything is now sorted in the correct order, and we receive contextual information describing the origin of the text. Here, for example, in addition to numbering the lines, which enables proper sorting, we are presented with information on the author of the poem, which can later facilitate the search for an existing translation. It is precisely the lack of similar contextual information that is one of the biggest problems when translating software.

Desperately Looking for Context

While working on the translation of novels, poems, articles, movies, or comic books, the translator almost always has all the information at their disposal. They know who is speaking to whom and what the gender of the speaker is, they are aware of the context in which the words have been used and what register to apply. It is not quite as straightforward in the case of video games, and there are several reasons for that. First of all, video games are an interactive medium in which the user can influence the development of the story. In addition, the textual layer is only one of many elements of the whole, preventing the translator from forming a complete overview of the situation. The translation spreadsheets often contain individual words or expressions that have multiple meanings depending on the context. Only after implementing the texts in the game can one truly see if they are properly translated. Here are a few excerpts from a variety of games that will help analyze some of the problems faced by translators:

String ID	CharName	English
CS_02_L11	Engine Driver	There… Up there, patrol
D_PlaneEsc	Merlin	You have to leave the plane!
Str_10	Tuesday	I did not know you killed it!

Fig.11

Examples of in-game strings without context

In the first example, taken from *Battlefield: Bad Company 2*[1], the string ID tells us nothing, apart from showing the possible sequence of expressions. We know who the speaker is (Engine Driver), but lack information on their gender. However, we can quite safely assume that the speaker is a man. Unfortunately, there is a problem concerning the phrase "up there". It can mean "ahead" or "on top". If we do not have any additional information, we have to take the risk and choose one of those options. The translator decides to pick "Tam.. przed nami, patrol" [patrol ahead], because this version seems more probable, and similar texts have appeared in other games in the series, almost always denoting something that is ahead of the speaker. Ultimately,

[1] *Battlefield: Bad Company 2*. (2010). DICE, Electronic Arts.

though, after verifying this line in-game, it becomes clear that the patrol in question is located quite literally above the speaker. The characters are sailing in a dinghy and passing under a bridge which is patrolled by enemy soldiers. The translator could not have guessed that from the original text, and needed the situational context, which was unavailable at the time of translation.

The second example seems deceptively simple. We know the name and gender of the speaker, and the phrase they utter – "You have to leave the plane!" – apparently does not contain any potential traps. But when we look closely, it turns out that we do not know to whom Merlin is speaking. When reading this sentence out of context we cannot determine whether he is addressing a single person or a group of people. In this case, it is safest to assume that he is speaking to one person. There remains the matter of the word "plane". Although the initial assumption would be to translate it as "samolot" [aircraft], Merlin's name, as well as the game's fantasy setting seem to point to the alternative option: "płaszczyzna" or "sfera egzystencji" [plane of existence]. Very similar wording appears in the game *Icewind Dale*[1], where it would be quite difficult to find any aircraft, but it is far easier to travel between worlds. However, without knowing the context for this line, the translator might be tempted to make the mistake of translating "plane" as aircraft, despite the fact that the final version should be: "Musisz opuścić tę sferę!" [You must leave this realm!] or even "Musisz opuścić ten wymiar!" [You must leave this dimension!], depending on the adopted convention.

Gender Problems

The translator's task becomes a little more difficult in the third example. The string ID does not contain any additional information and the name of the character – Tuesday – does not help determine their gender. Also, similarly to the previous example, we do not know who is addressing whom. It can be assumed that the addressee is the character controlled by the player, and that Tuesday is one of the cast of NPCs (non-player characters). In many games, the gender of the protagonist can be chosen by the player. Therefore, the translation must be appropriate for both male and female versions. The phrase "killed it" alone can also be problematic: if taken literally, it means "to kill some creatures", but as an idiom it denotes the successful conclusion of some activity. So, if the translator does not have any additional information (and such is often the case), the sentence "I did not know you killed it!" can be translated in several ways. Let us assume, though, that we know from other sources that Tuesday is a woman, and she is addressing the player's character. But the question of the addressee's gender still remains. Thankfully, sentences in the present tense are not problematic in Polish. Unfortunately, in the case of past and the future tenses, Polish verb forms are different for women and men. Assuming that Tuesday is a woman, we

[1] *Icewind Dale* (2000). Black Isle, Interplay, CD Projekt

can translate "I did not know" as "Nie wiedziałam", while the rest of the sentence can be rendered in two different ways:

"Nie wiedziałam, że go **zabiłeś**." (female subject, male object)

"Nie wiedziałam, że go **zabiłaś**." (female subject and object)

The point is that regardless of the selected version, there will be situations in the game where the chosen translation will be incorrect. That is one of the many problems encountered in game localization that stem from the fact that games are an interactive medium and, ultimately, it is the player who decides the final form of many of its aspects, including, for example, the gender of their protagonist.

There are at least three ways to solve this issue, although some require intervention on the part of the programmers, as well as adequate preparation of the source material and creation of additional tools. The simplest method is to duplicate all the texts in the game (or, at least, dialogues and descriptions), so that they can be translated in two versions – male and female. Those alternative versions need to have different identifiers, so that the software "knows" which one to choose, depending on the player-controlled character. This method is used, among other games, in *Deus Ex: Invisible War* [1] and *Diablo III* [2]. However, it is generally rarely applied because it causes a considerable increase in the amount and size of textual data, and above all – an increase in the localization costs.

The second method is a little more complicated, and its application also requires the support of the game's programmers. It involves placing special tags specifying gender within the translated text. When the program encounters a tag, it appropriately adjusts the text that follows, for instance – by assigning to it a particular gender. Such translated sentences containing markup elements might look like this:

"Nie wiedziałam, że go [m:{zabiłeś}][f:{zabiłaś}]."

Depending on the character's gender, the program will display the initial text (not containing tags) normally, and then supplement it with the words found in tag brackets "[m: {}]" or "[f: {}]". Unfortunately, apart from the obvious advantages of the tag system, tags have a fundamental flaw: it is very easy to make the mistake of not closing the brackets properly or using the wrong syntax, which, at best, will only lead to the game displaying some strange-looking text (like: "Nie wiedziałam, że go zabiłeśzabiłaś."), and at worst – to crashing the game. During the translation of such texts it is necessary to exercise extreme caution or use scripts that check the translation's syntax after its completion. The tag method is quite widespread and can be encountered in games such as *Guild Wars*[3], or the *Dragon Age*[1] series.

[1] *Deus Ex: Invisible War* (2003 [2004]). Square-Enix, Cenega
[2] *Diablo 3* (2012). Blizzard Entertainment, CD Projekt
[3] *Guild Wars*. (2006-2012). NCSoft

Still, even today there are game developers who do not provide any supporting tools for translators. In such cases, the only available option is to create a target text that will be suitable for both male and female versions. Handling the endings of Polish verbs (which are marked for gender) alone can cause quite a headache, and it often can result in sacrificing the characteristic register of the original in order to convey the content of the message. The passive form or expressions such as "udało się" [someone managed to do something – in Polish this expression does not require a subject], or "zdarzyło się" [it happened that...] are used. When utilizing this approach, our example would look like this:

"Nie wiedziałam, że udało ci się go zabić!"

or

"Nie wiedziałam, że zginął z twojej ręki!"

"Unisex" translation has many limitations. Although it most often manages to convey the meaning of the original, this usually happens at the expense of style. Such translation strategy should be applied only as a last resort measure, when there are no other options available. Its use was required in the case of the more "voluminous" titles such as *The Elder Scrolls: Oblivion* or *Fallout 3* [2](each contained over 1,000 pages of text), because the engine of these games did not support other methods.

Variables

Of course, the search for context and solving the problem of gender are not the only challenges facing translators. There are numerous pitfalls to be found in gaming-related texts, which often take the form of the already-mentioned tags and variables. Variables are used in programming languages. They are inserted into the text to provide a reference to the software's database. They can replace any part of the text, and are used primarily for the sake of convenience, and also to save time and space. It is easier to write "In order to prepare this spell, find %s", where "%s" stands for one of the many ingredients needed to cast spells, than to repeat the same sentence a few or dozens of times, replacing the individual ingredients. Variables can be stored in many different ways – the simplest is "%n" (n stands for "number") for numerical variables, and the aforementioned "%s" ("string") for textual variables. One might ask: what, then, is so difficult to translate when dealing with texts containing variables? The answer is simple – the problem is caused by inflection in Polish (and the lack thereof in English). As a rule, words taken from the database have only the nominative case, and it is difficult to create a sentence in the Polish language without applying inflection. It will be easier to demonstrate this phenomenon by using an example:

[1] *Dragon Age: Origins*. (2009). BioWare, Electronic Arts
[2] *Fallout 3* (2008). Bethesda Studios, Cenega

Gather %n1 %s1, %s2 and %s3 and then use them to craft %s4.

Some textual variables and one numerical variable can be seen here. Let us then translate this sentence as if there were nothing special about it.

Zbierz %n1 %s1, %s2 oraz %s3 i użyj ich, aby wykonać %s4.

The layout of the text and the original order of the variables have been preserved. To see how the sentence would appear in the target form, we need to substitute the variables with words. This is a possible result:

Zbierz **2 materiał**, **dratwa** oraz **igła** i użyj ich, aby wykonać **czapka kolejarza**.

Although we more or less understand what the sentence means, it is completely ungrammatical because of the absence of inflected noun endings. It becomes clear that we need to adopt a different approach to avoid this problem.

The best solution to this conundrum would once again require intervention on the part of programmers, and also translators. Developers must prepare additional variables for inflection, and translators have to translate each entry in the database using the most commonly-occurring Polish cases. As a rule, however, developer assistance is not possible in similar cases. The problem of inflection has to be dealt with differently – by modifying the syntax of the sentence, so that nominative-case nouns can be used. For example:

Zbierz te składniki: %s1 (%n1), %s2, %s3. Następnie użyj ich, aby stworzyć następujący przedmiot: %s4.

Here, the variables have been rearranged in such a way as to form a kind of list, and the numerical value is shown in parentheses to avoid compatibility issues with different cases. After substituting the variables with words, the text reads as follows:

Zbierz te składniki: **materiał (2)**, **dratwa**, **igła**. Użyj ich, aby stworzyć następujący przedmiot: **czapka kolejarza**.

The grammatical correctness and the meaning of the original have been preserved. Such an approach works well in the case of purely utilitarian, informative texts. When technical restrictions make it impossible to introduce elements of inflection into the translation of variables, the translation in the form of a list appears to be the most practical solution. However, there are situations when the translator needs to count on their intuition and rely on blind luck, because it often happens that texts from video games consist of nothing BUT variables. For example: %s1 %d % s2 %n %v. When the translator encounters such a passage, their first order of business is to determine where the message is displayed in the game, and only then should they try to rearrange the order of variables. During the translation process, one must also be very careful not to make a mistake while typing in the variables, as this can cause problems with the operation of the program. Fortunately, most CAT tools have options to disable the ability to edit selected elements of the source text, which is especially useful for long descriptions that combine the use of variables and tags whose role is, for instance, to change the color of the displayed text, like in the following example:

Super Pistons
{c_gold}Cost:{/c_gold} {c_green}{Resource Cost}{/c_green} Coal^n{c_gold}Cooldown:{/c_gold} {c_green}{Script Formula 23}{/c_green} seconds^n^nPut extra pistons in your locomotive. The pistons add {c_green}[{Script Formula 2}*100%]% {/c_green} to you speed. Lasts {c_green}{Script Formula 17}{/c_green} seconds. You may have {c_green} {Script Formula 22}{/c_green}extra pistons active at a time.

This description is more like a math equation or HTML code, which is used in webpages. The passage is not particularly clear, and consists of a multitude of variables and tags, such as "c_gold {}", which stands for the color of the letters. Also visible are "^n" characters, which are used to determine the beginning of a new line. Once translated, the description reads as follows:

Super Tłoki
{c_gold}Koszt:{/c_gold} {c_green}{Resource Cost}{/c_green} węgla.^n{c_gold}Czas odnowienia:{/c_gold} {c_green}{Script Formula 23}{/c_green} s.^n^nUmieszczasz w lokomotywie dodatkowe tłoki, która zwiększają twoją szybkość o {c_green}[{Script Formula 2}*100%]%{/c_green}. Czas działania: {c_green}{Script Formula 17}{/c_green} s. Liczba dodatkowych tłoków, jakie możesz mieć zamontowane jednocześnie wynosi {c_green} {Script Formula 22}{/c_green}.

Only after implementing the texts in the game can one really see the final layout of text and verify the correctness of the translation (Fig. 12).



Figure 12

Onscreen version of translated text

Conclusion

In the light of the above examples, it is clear that the localization process differs significantly from other branches of the art of translation. It requires a different approach and presents the translator with a completely new set of challenges. I have touched upon only a handful of aspects included in the process – there still are plenty of issues to be discussed, such as the question of translating proper names, concatenation – or creating texts from random elements, and many other topics. I hope this article will help the Readers understand the essence of localization process, and increase their interest in this subject. Localization of video games is a very interesting area that is constantly changing and evolving. As a craft, it demands creativity, a passion for discovery and innovation, and often requires the adoption of the unorthodox approach.