Review

Integrating Localization into a Video Game

Sanja Seljan
Department of Information and Communication Sciences
Faculty of Humanities and Social Sciences. University of Zagreb
Ivana Lucica 3, Zagreb, Croatia
sanja.seljan@ffzg.hr

Josip Katalinić
Department of Information and Communication Sciences
Faculty of Humanities and Social Sciences. University of Zagreb
Ivana Lučića 3, Zagreb, Croatia
josipkatalinicandroid@gmail.com

Summary

The aim of the paper is to analyse the integration of the localization process within a video game from the engineering perspective, paying attention to tools and resources, processes and project management. Localization is perceived in the context of the global market with regard to language, market needs, time, monetization and project organization. In the paper, the integration of localization into project setting is analysed, then methods of automation of localization, quality assurance and time aspect. The authors suggest several stages of automation in the localization process, such as key-value pairs imported from spreadsheets, dynamic retrieval in the context, separation of code from the content, modularity and point out the importance of quality assurance. Investing in quality assurance of the localization is essential and it is achieved through a number of methods and storage structures where a systematic access to text ensures the visibility.

Key words: localization, video game, automation, project organization, quality assurance

Introduction

Video games, as an interactive media, are growing every day, weather through entertainment, education or business ventures. By expanding popularity, video game localization has become an indispensable part of the game production cycle and approaches to the global marketplace. According to Erbil (2016), media industry has significant influence on the evolution of digital product, comprising movies and video services, digital music, electronic publishing and video games, the last one showing the biggest revenue. Among the mentioned digital products, video games show the tendency of growth (67% of household in USA play video games; revenue from video games and virtual goods in 2012 are eight times bigger than in 2009). Czech (2013) elaborates on video game industry which is seen as the fastest growing industry in the last five years, due to, among others, development of video game concept and the increase of cultural values in interactive entertainment.

Interactivity of video game media offers us unique experiences that can be perceived through localization, linking it to other cultures. Although localization is perceived from the wider perspective which requires technical, linguistic, project and marketing knowledge and skills, not many researches have elaborated an integrated and a technical perspective. Important question is what language to localize in? How to organize localization? How to implement automation process in localization? Publishing platforms like Google Play, App Store and Steam are also going to be used for exploring tendency of language and its connection to a particular platform. Monetization factor is also important in language selection, as certain markets and languages have greater profitability than others which is derived via market analysis.

Methods that are explained in paper describe systematic integration of the localization, supported via scripts for automation based on industry experience. Current industry is experiencing many shortcomings that are directly connected to the inability of creating reusable system that would have all variables stored in a separate resource file, so nobody needs to go through source code to add translated text into the game.

In the paper, the integration of the localization process within the video game is analysed, from the engineering perspective. In the second chapter the role of the localization process inside of GILT sector is pointed out. In the third chapter, localization is perceived in the context of the global market with regard to language, market needs, time, monetization and project organization. Levels of video game localization are discussed, as well as use of different tools, processes and the role of project management. In the fourth chapter, the authors analyse aspects of the integration of the localization into project setting, methods of automation of localization, quality assurance and time aspect. In the conclusion suggestions for further research and implementation are given.

GILT (Globalization, Localization, Internationalization, Translation) sector

According to W3C, "*localization* refers to the adaptation of a product, application or document content to meet the language, cultural and other requirements of a specific target market (a locale)". Localization refers to, as stated in LISA (2007), linguistic issues, physical modifications, business and cultural issues. Localization often assumes adaptation of settings that usually include number format, date and time zone formats, currency usage, keyboard usage, case conversion, collation and sorting, then symbols, icons and colours, paper size, adaption of text, abbreviations, graphics and ideas that may cause misinterpretation, physical product adaptation, shortcuts, forms, etc. Localization may require a comprehensive review of logic, visual design or presentation if translation is significantly different from the culture of origin. Localization process can also include change of design, marketing message, object or colour, or paradigm in order to adapt the product to the target culture. Localization goes beyond text structure, including also technical, cultural and marketing strategies, as it is oriented to the final product, "without taking any standpoint on the source text or the relation between the source text and the target text", as in Czech (2013). Localization is often referred to as 'l10n' indicating 10 letters between 'l' and 'n'.

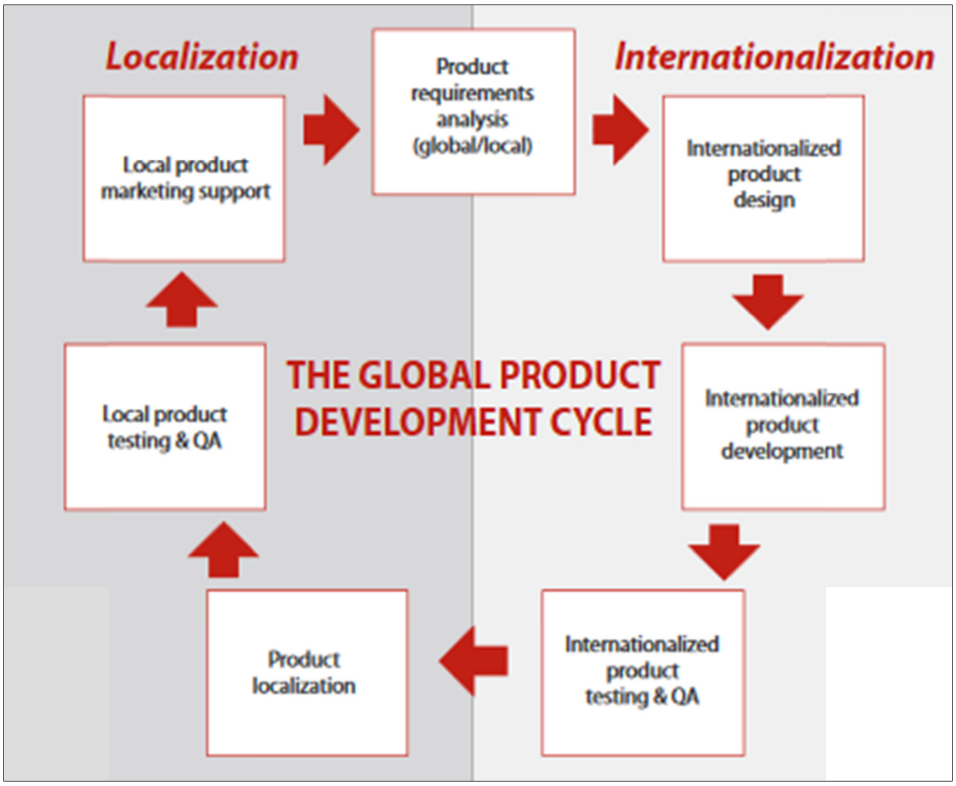
According to Esselink (2003), localization has moved from in-house localization in 80's to internationalization in 90's, focusing on project management, programming and publishing environment using XML, Java and .NET. At the beginning, localization was separated from the products, while today developers create software with source files created for internationalization process. Later on, the expansion of software and documentation has asked for the process of internationalization, which precedes the localization process.

According to W3C, the term *internationalization* refers to the "design and development of a product, application or a document content that allows a localization for a target audience that is different from source product culture, region or language" or, according to LISA, "the process of enabling a product at a technical level for localization". In that sense, it can be said that internationalization precedes and facilitates the task of localization. For internationalization, the numeronym 'i8n' is frequently used. Internationalization assumes the level of abstraction from the specific culture, language or the market. It is performed only once in the development process and can influence the localization process, which impacts ease of technical issues, costs and business processes.

While translation is the process transferring content from one language (original language) to another language (target language) by preserving the context, *localization* process is more specialized and oriented to a final product in order to meet expectations of the global market, as indicated on web pages of Lionbridge. Pym (2005) defines localization as translation + factor X, where factor X includes internationalization, adaptation, controlled writing, use of translation memories, integration of machine translation, project management, i.e. technical knowledge and skills, management and marketing issues.

Šiaučiūnė and Liubinienė (2011) point out that "*localization* encompasses not only activities of traditional translation (terminology research, editing, page layout, proofreading), but also includes multilingual project management, software and online help engineering and testing, conversion of translated documentation to other formats, translation memory alignment and management, multilingual product support, and translation strategy support. One more important difference between translation and localisation is that the former is performed after the product or document is finished and released, and the latter usually runs in parallel with the development of the source document or product."

According to Erbil (2016), *internationalization* is related with developers and copywriters of the content who are engaged in design and development step. In the localization process, various experts engaged in product adaptation are involved, such as engineers, translators, project managers and test engineers. Globalization is related with sales and marketing activities in order to present the product at international market. Translation is often done with the help of software where it is commonly used in form of machine translation (MT) or computer-assisted translation (CAT), requiring background knowledge of these technologies for successful implementation. Software localization tools include the whole range of commercial and free software created for Windows or Macintosh platform. CAT tools provide stable platform for immediate sharing and updating translation memories (Brkić et al., 2009), which can be used integrated with MT technology and terminology management software.


Localization
Product requirements analysis (global/local)
Internationalization
Local product marketing support
Internationalized product design
THE GLOBAL PRODUCT DEVELOPMENT CYCLE
Local product testing & QA
Internationalized product development
Product localization
Internationalized product testing & QA

Source: LISA. The Globalization Industry Primer, 2nd edition, 2007.

Figure 1. The global product development cycle

Localization of Web sites often require use of content management systems (CMS) in order to create, manage, store and publish complex sites that often contain functionalities that require an engineering knowledge. Localization also includes other types of software, such as tools to track changes on Web sites, word counting tools, currency converters, etc. Publishing is often performed using XML, Java and.NET environment. XLIFF (XML Localization Interchange File Format) is XML-based format used to store content extracted form original file and the translated file, TMX (Translation Memory Exchange) for the storage and interchange of translation memories and TBX (Term Base Exchange) as a model for terminological databases.

According to LISA, the final stage of this process is globalization which is associated with “all issues to make a product or service really global, integrating also business processes, marketing, sales, customer support, etc. The whole process in known as GILT (Globalization, Internationalization, Localization, Translation) process where various departments (development and planning, design, production and technical department, translation, marketing) and experts are included.

Video game localization

The problem with video games, unlike for e.g. documentation or technical texts containing repetition and being suitable for CAT, is that video games mostly consists of creative texts (e.g. artistic descriptions of landscapes). Video games also do not use controlled languages (except programming language) or any scheme in the text creation phase. Moreover, game makers often employ professional writers who create a story for a particular title. The nature of video game texts is one of the reasons why they are not easily translated using the CAT tools or machine translation, where sublanguage is often used (Seljan, 2000). The second difference between video games and technical translation is the medium. Video games are embedded in information technology. As in Mateusz (2014), video game translation is interwoven with the source code of a particular product. As presented in Šiaučiūnė and Liubinienė (2011), Chroust (2007) elaborates localization as multi-layered process in the form of pyramid performed at different levels:

- technological infrastructure – as the basic level, including all technological and organizational preparation before and during the localization
- grammatical and semantic layer – as the second and third layers, including textual translation and language expressiveness
- graphic and iconic representation layer – which can be changed when localizing software, games or web sites
- business conventions and practice layer – important for e.g. for contractors or the company who can buy the rights to perform localization

- social and communication layer and cultural layers – the last two layers related to specific cultural aspects or specific market.

In today's work, there are numerous solutions to localization, but outsourced localization team is often a solution. Numerous companies and associations offer their services on the market for localization, although sometimes it seems as a major and perhaps unnecessary investment, localization is today considerably important process for increased competitiveness in the global marketplace. Cooperation with the localization can start from the preparation of the software that will use localization, continuing in regular intervals in order to avoid pauses in production, also in Soh et al. (2016). Software preparation through internationalization greatly facilitates the localization team to see and test the context of use, thus increasing the ability to test the integration of localization and the quality of the translation. Quality control through communication with the localization team within the development process, and clear use of unambiguous words leads to successful localization project, as in Cem et al. (2016).

Cardenosa (2006) explains three main approaches in the architecture of the internationalization process:

- when messages, menus and other culturally sensitive factors are contained in the source code of the application – for this case it is necessary to develop different version for each language, which multiplies costs and time
- when messages are extracted into external library, the application is generated from the common code and linked to the specific language library
- the third approach consisting of core application with all functionalities, culturally independent and linked dynamically to external localization resources containing all specific information.

According to Erbil (2016), different game companies follow today different levels of localization, pointing out four levels of localization:

- the first level when *localization is not performed* because of cutting costs. At this level introductory text, content and documentation are not localized. It can be expect that from fan groups to translate the game.
- the second level, called "*box and documentation localization*", assumes that mostly introductory text, cover text and game manual are translated, while in-game text, visual content and dubbing are in foreign language
- the third level is *partial localization*, which includes all in-game texts and visual text translated and localized, while dubbing and effects are in original language
- the fourth and last level is *full localization*, which includes all elements localized (introductory text, visuals on platform when buying the product, text content, documentation and rich content of the game, e.g. dubbing, effects, videos).

Another area of relevant development are online games, which often incorporate chat among players while the game is being played. This implies a need for simultaneous translation of text chat and chat based on speech, as online games have no regard for linguistic boundaries. This raises the scope for research into the application of natural language processing systems.

Benefits of localization

According to data presented by ICT Facts & Figures (2015) development of the mobile gaming market and the adoption of the Internet show that localization contributes to sales in other markets. In addition to monetizing localization, it also enables the convergence of creativity writing to many users who can enjoy localized texts, which would not be possible if the game was released in a language they do not know.

By publishing a localized game to a new market, it is possible to acquire a new community that monitors our product and becomes a future buyer if the video game has multiple sequels or downloadable content. If the budget is limited and does not allow localizing the entire game, this problem can be approached by localizing only the application keywords and thus accessing more actively global market where users potentially use the language of the application, but search in their native language. Keyword localization is a far cheaper and faster solution then localizing the entire game, but a research is required for each language as well as statistics of localization, using tools such as GKWT, Google Trends and App Store Search Suggest, etc.

Users tend to buy more if content is written in their first language and in that way localization is contributing to monetization by opening up markets to numerous customers as potential buyers. According to data on Statista web pages, Asia and especially China rapidly enter the global video game market, which is expected with regard to the population and the economic growth of the middle class. Sales of smartphones are also on the rise, a sales growth of iPhone devices in China exceeds sales in the US and each iPhone owner is a potential buyer of a video game.

Integrating localization

Challenges that localization poses through integration within the game are not negligible and the localization process should be considered when setting up a project. Localization process can start together with the development of the software product and online documentation. For this part, translation memory technology can be used, as it enables cooperative work, sharing and updating and interface localization tools to translate resource files. Late integration of localization within the game is leading to increased pressure on production, since it is necessary to integrate, check the integration accuracy as well as the localization itself, within the short time that is set in the production cycle.

Localization integration is possible to implement in different ways, e.g. a *key-value pair* or ID–localized text. If there is a game containing texts that needs to be localized, it is suggested to use a central place (e.g. Google Sheets) where it is possible to enter IDs of text to be localized. For example, the option is to create for e.g. ID DOCTOR_SPEECH with the value 'Hello' in English, and to have a look at ID DOCTOR_SPEECH as a reference to value that can be variable. In the next step, it is possible to send for translation using a file containing IDs pointing to a text that needs to be localized. The value of the ID varies depending on the language settings within the video game: if the language is set up for the German, DOCTOR_SPEECH will reference the value 'Hallo'.

Modularity can be achieved using IDs characters with defined values that can be replaced inside the programming language. For example, in order to create modular talk with a doctor that depends on his mood, it is possible to create within the Google Sheet ID such as DOCTOR_SPEECH_MOOD with value 'Today I am %mood%'. Using Python programming language with the replace method we can easily replace the first argument with another argument. The mood that is placed through the programming logic can then be replaced by %mood%. We also have to note that translation columns within the Google Sheet need to have brief context of message so translators can use that message and the name of defined values inside IDs to make better understanding of what could be placed inside those values.

```
current_mood = DOCTOR_SPEECH_MOOD.replace('%mood%', 'sleepy')
```

In this case the variable current_mood contains the value of 'Today I am sleepy'.

Another positive aspect to IDs access is that it considerably facilitates localization by *separating content and code*, and thus *enables integration of modularity*. There are different localization techniques as (Holovaty 2009) explains, with usage of *gettext* via the standard gettext module that comes with Python in context of Django web framework, showing that it is possible to add translation strings to tell Django that "This text should be translated into the end user's language, if a translation for this text is available in that language." Philips (2015) provides example of index.html Django page with three strings that require localization "Log in", "Username" and "Password". To start with, there is a list of language which the site support inside settings.py with:

```
LANGUAGES = (
    ('en', _('English')),
    ('fr', _('French')),
)
```

After that translation template tags are required, which are accessible by putting `{% load i18n %}` at the top of the template files. There are two i18n translation template tags: trans and blocktrans, where `trans` tag marks a string for translation while `blocktrans` is used if translation require variables (placeholders). Once the text is put inside trans tag, our index.html template will look like:

```
{% load i18n %}
<h1>{% trans 'Log in' %}</h1>
<label>{% trans 'Username' %}</label>
<input id='password' type='text'/>
<label>{% trans 'Password' %}</label>
<input id='password' type='password'/>
```

After the text is wrapped in `trans` tags, it is possible to proceed to creating translation files with

```
python manage.py makemessages -l 'fr'
```

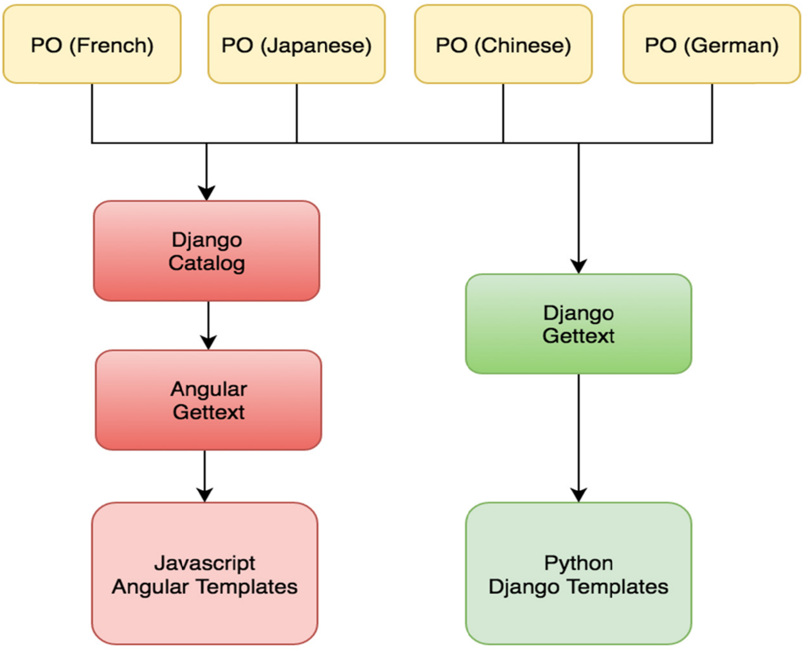
by which .po file is created, which contains actual translations. Each language has its own .po file, usually saved inside of application/locale/<lang_code>/ LC_MESSAGES/django.po. The following information is presented:

```
#: login.html:2
msgid "Log in"
msgstr ""
#: login.html:4
msgid "Username"
msgstr ""
#: login.html:7
msgid "Password"
msgstr ""
```

After replacing msgstr with translation, it is possible to proceed to creating machine object (.mo) that contains object data referenced by a program:

```
> python manage.py compilemessages
```

The files .mo and .po, after changing the language setting, are presented. Gettext can be used in other web frameworks, like AngularJS as shown in Figure 2 where we see implementation of Angular Gettext with JavaScript Angular Templates based on Django Catalog. Here Django Catalog means Django Message Catalog that corresponds to directories that are created and organized by language with their .po and after compiling .mo files. Standardization of .po files enables us this power, so we can use it with different web frameworks, Figure 2 also shows on the right side more traditional way of getting translations strings from .po files in the manner what we already discuss previously.



Source: Translation in Horizon. 07.09.2017

Figure 2. Message Substitution

Organization of localization in multiple documents is desirable, in order to avoid creating a large single file of the localization document. For example, it is possible to store the menu localization in one document and button texts in another. By subdividing content into categories, it is easier to find specific localized data and referencing within the code. Separation of the content from the code makes communication easier. The localization process including various experts entails structured and organized workflow in order to obtain good quality product localized on time.

Quality assurance

Ensuring localization quality is a multi-dimensional problem that can be approached through the software design, here video game, and by dividing the product into two concept blocks:

- a block containing all the elements of the user interface and
- the block containing the execution code.

The user interface block contains only localized elements such as text, error messages, windows, menus and so on. The execution code block contains only the application code that will support the localization. Such structure allows testers, as well as programmers, to quickly and systematically test and implement localized video game, saving time and money in the process. In the perfect situation, an original speaker would perform testing of implemented localization. Every problem needs to be described, recorded, evaluated and finally corrected. After performing all corrections, the new version is sent for verification. As this process takes a long time, it is important to plan for it. A crucial factor in ensuring quality of localization is the time, since the localization team should start working together with the development of video game as early as possible, thus avoiding the production delays that may arise with the arrival of a large number of localized texts. An estimation of the required project resources is typically carried out by a project manager in a localization service company. The localization team activity through questioning within the production cycle is a key factor in quality assurance, making the context of localization clearer.

One of the possible acceleration of localization testing within the game can be achieved by setting and grouping localized text within the Google Sheet so that each spreadsheet denotes a particular entity. Therefore, it is possible to have spreadsheet of user interface, text buttons and windows. The search could be facilitated by setting up description of each localized text, for example if we have original English text in column A, then we can have description in column B and localized text in column C. Sanchez and Lopez (2016) propose the following *methods for quality assessment* in localization testing, where testers can choose one or more methods, depending on time and budget:

- step-by-step testing,
- ad hoc testing,
- screen testing,
- emulation.

Testing step-by-step is the most advanced method. It basically consists of the step-by-step testing plan presented in the specific order (e.g. testing dialogues, menus, descriptions, etc.). Testers should follow this plan, trying not to skip any of steps.

Ad hoc testing complements the previous technique. The testers leave the test plan and checklist through playing the game, doing the actions that each standard player would. This type of testing can be useful for finding problems at locations that were not included in the test plan or to focus on details in a similar thinking style as the player.

Testing the screen is done by sending a screen instead of the game with which testers interact. However, this method is only useful if the number of screenshots is not too large, since screen capture for each language takes a time (although sometimes the process can be automated).

Emulation though not a method as such is a useful resource in certain situations. Emulation is a set of additional debug options inside the game that is currently in production, so testers can easily simulate different aspects of game without playing through whole game. The emulation method is frequently used because it saves time and enables checking (e.g. level 5 out of 7). Testers must be able to use these application testing options correctly.

Conclusion

The aim of the paper was to analyse the integration of the localization process within into a video game from the engineering perspective, paying attention to tools and resources, processes, project management and quality assessment. Localization is perceived as a multi-layered process, where attention is given to technical and organizational layers. The authors suggest some aspects of automation in the localization process, such as key-value pairs imported from spreadsheets, dynamic retrieval in the context, separation of the code from the content, modularity and importance of the quality assurance. Investing in quality assurance of the localization is essential and it is achieved through a number of methods and storage structures where a systematic access to text ensures the visibility. The authors also point out simultaneous collaborative work on the same project and need for integration of the localization process from the beginning of the project setting. Localization integration is an important factor that will have to be standardized for fast and efficient integration through the process of internationalization that precedes the localization. The authors suggest continuing the research on standardization of the technical and organizational layers in the domain of localization engineering (also in Soh et al., 2016) and correlating it with output efficiency.

References

Brkić, M.; Seljan, S.; Bašić Mikulić, B. Using Translation Memory to Speed up Translation Process. // INFuture2009 - Digital Resources and Knowledge. Zagreb: Department of Information Sciences, 2009.

Cardenosa, Jesus; Gallardo, Carolina; Martin Alvaro. Internationalization and localization after system development: a practical case. // International Journal Information Theories and Applications, 2006.

Cem, Odacıoğhu Mehmet; Loi, Kim Chek; Köktürk, Şaban; Uysal, Müge Nazan. The Position of Game Localization Training within Academic Translation Teaching. // Journal of Language Teaching and Research, Vol. 7 (2016.), No. 4

Chroust, G. Software Like a Courteous Butler Issues of Localization Under Cultural Diversity. // Proceedings of the ISSS. 5th Annual meeting and Conference for the System Sciences, 2007.

Czech, Dawid. Challenges in video game localization: An integrated perspective. Explorations. // Journal of Language and Literature, Vol. 1 (2013.), No. 1; 3-25

Erbil, Mert. Video Game Localization Factors and Impacts on Digital Purchasing Behavior, Dissertation, 2016.

Esselink, Bert. The evolution of localization. // The Guide from Multilingual Computing & Technology: Localization, (2003.), No. 57 Supplement; 4-7

Holovaty, Adrian; Kalpan-Moss, Jacob. The Definitive Guide to Django. New York: Apress, 2008.

Krammer, Connor. Persona 5: Phantoms of Translation 02.05.2017. http://www.personaproblems.com/ (07.09.2017.)

LISA. The Globalization Industry Primer, 2nd edition, 2007.Mateusz, Sanja. Computer-assisted translation tools and video game rendition. // The Translator and the Computer 2, 2014

Muñoz Sánchez, Pablo; Sánchez López, Rafael. The ins and outs of the video game localization process for mobile devices. // Revista Tradumàtica: tecnologies de la traducció, (2016.), No. 14

Nichols, Brian. The Difference BetweenTranslation and Localization for Multilingual Website Projects. 28.07.2015. http://content.lionbridge.com/the-difference-between-translation-and-localization-for-multilingual-website-projects-definitions/ (25.04.2017.)

Philips, Noal. Translating your site with Django 1.8 03.10.2015. https://medium.com/@nolanphillips/a-short-intro-to-translating-your-site-with-django-1-8-343ea839c89b (06.09.2017.)

Pym, Anthony. Translation and Localization. 21.10.2005. http://www.fut.es/~apym/on-line/talks/localization_bergen_2005.ppt (08.06.2017.)

Sanou, Brahima. ICT Facts & Figures. May 2015. http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf (29.04.2017.)

Seljan, Sanja. Sublanguage in Machine Translation. // Proceedings of 23rd Int. Convention MIPRO 2000: Computers in Intelligent Systems CIS + CTS. Rijeka: Liniavera, 2000.

Šiaučiūnė, Vaida; Liubiniene, Vilmante. Video Game Localization: the Analysis of In-Game Text. // Studies about Languages, (2011.), No. 19.

Soh, Mathurin; Nkenlifack, Marcellin; Fotso, Laure P. A New Hybrid Process for Software Development and Localisation. // International Journal of Scientific and Engineering Research, Vol. 7 (2016.), No. 6; 945-953

Steam Hardware & Software Survey. April 2017. http://store.steampowered.com/hwsurvey/ (02.05.2017.)

Translation in Horizon. 07.09.2017. https://docs.openstack.org/horizon/latest/contributor/topics/translation.html (07.09.2017)

W3C. Localization vs. Internationalization. https://www.w3.org/International/questions/qa-i18n (12.6.2017.)