VIRTUALSIGN TRANSLATOR AS A BASE FOR A SERIOUS GAME

Paula Maria de Sá, Oliveira Escudeiro, Nuno Escudeiro, Marcelo Norberto, Jorge Lopes

pmo@isep.ipp.pt, nfe@isep.ipp.pt, matno@isep.ipp.pt, jolbl@isep.ipp.pt

Case study

DOI: **10.7251/JIT16012S** UDC: **004.738.5**

Abstract: The goal of this paper is to present the development of a game aimed at making the process of learning sign language enjoyable and interactive, using the VirtualSign Translator. This game aims to make the process of learning sign language easier and enjoyable. In the game the player can control an avatar and interact with several objects and Non-player characters in order to obtain signs. Through the connection with VirtualSign Translator the data gloves and Kinect support, this interaction and the gestures can then be represented by the character. This allows for the user to visualize and learn or train the various existing configurations of gestures. To improve the interactivity and to make the game more interesting and motivating, several checkpoints were placed along game levels. The game has as an inventory system where the signs are kept and can be checked allowing for the user to visualize and learn or train the various existing configurations of gestures. A High Scores system was also created, as well as a History option, to ensure that the game is a continuous and motivating learning process.

Keywords: VirtualSign, Serious Games, Portuguese Sign Language, Kinect.

Categories and Subject Descriptors

H.5.2 [User Interfaces]: *Graphical user interfaces (GUI); Interaction styles; Theory and methods; User interface management systems (UIMS)*

General Terms

Information Interfaces and Presentation

Conference'10, Month 1–2, 2010, City, State, Country.

DOI: http://dx.doi.org/10.1145/2808580.2808618

INTRODUCTION

Currently, games and simulations create remarkable opportunities to overcome the scarcity of educational digital content available for the hearing impaired community. Playing a game as the name suggest has a great leisure aspect that cannot be found in conventional educational means. Educational game researcher James Gee [3] shows how good game designers manage to get new players to learn their long, complex, and difficult games. A well-designed game entices players into the "reality" of the game world and keeps them there until the goals of the game have been met [10]. Making these opportunities available to those who endure handicap and disabilities is a core concern in today's society and a must to promote equity and inclusion. In this work we propose a new approach by using a game to make the process of learning sign language enjoyable and interactive. In this game the player controls a character that interacts with various objects and non-player

characters with the aim of collecting several gestures from the Portuguese Sign Language. The game is played in first person view in which the player controls a character on one of several maps. Each map represents a game level each of which has several objects that represent signs scattered through the map for the player to interact with. All objects gathered by the player will be stored in his inventory and can be accessed at any point during the game. With these objects the players can then mimic the gestures at the checkpoints using the VirtualSign translator. In case the players chose to play without the translator the avatar will be the one performing the gestures.

State of the art

Our research showed that there are some projects related to this theme/area but none of them implies an automatic bi-directional translation process in a game as this does, therefore, making this project very innovative. There is a rising number of serious games projects. Some of the most relevant related work within the sign language scope is described below.

The game CopyCat is the most similar project in comparison to ours. It consists of a game where sign language gestures need to be executed properly in order to proceed. The movement analysis is done through gloves with sensors. However, the researchers from the CopyCat project have published a video where they show their intention to use Kinect for movement detection. Their current research platform is a custom system that uses computer vision, colored gloves and wrist-mounted 3-axis accelerometers to collect data as users sign and machine learning to recognize the signs for game play. That system was built on top of Ubuntu Linux and uses the Kinect system as input for the computer vision, which replaces the gloves and sensors.

ProDeaf is an application that does the translation of Portuguese text or voice to Brazilian gesture language. This project is not a serious game but it is very similar to one of the main components used on the VirtualSign game, which is the text to gesture translation. The objective of the ProDeaf is to make the communication between mute and deaf people easier by making digital content accessible in Brazilian gesture language. The translation is done using a 3D avatar that performs the gestures. This software is already used by over 130 000 users.

Kinect Sign Language Translator is another project that is similar to the VirtualSign translator. The project was a result of collaboration, facilitated by Microsoft Research, between the Chinese Academy of Sciences, Beijing Union University, and Microsoft Research Asia, each of which made crucial contributions. Dedicated researchers in China have created the Kinect Sign Language Translator, a prototype system that understands the gestures of sign language and converts them to spoken and written language—and vice versa. The system captures a conversation from both sides: it displays the signer and renders a written and spoken translation of the sign language in real-time, and it also takes the non-signer's spoken words and turns them into accurate, understandable sign language. An avatar on the screen represents the non-signer and makes the appropriate sign languages gestures.

VirtualSign

VirtualSign aims to contribute to a greater social inclusion for the deaf. Its main motivation comes from a team of university teachers that have realized the difficulties in communicating with deaf students in the context of a class. The creation of a real time bi-directional translator between PSL and text is expected to facilitate the communication with students who have hearing disabilities. In addition to the bi-directional translator, this paper also presents a serious game directed to assist in the learning of the Portuguese Sign Language.

The project bundles three interlinked modules:

Translator of PSL to Text (Figure 1): module responsible for the capture, interpretation and translation of PSL gestures to text. A pair of sensors gloves (5DT Data Gloves) provides input about the configuration of the hands while the Microsoft Kinect provides information about the orientation and movement of the hands.

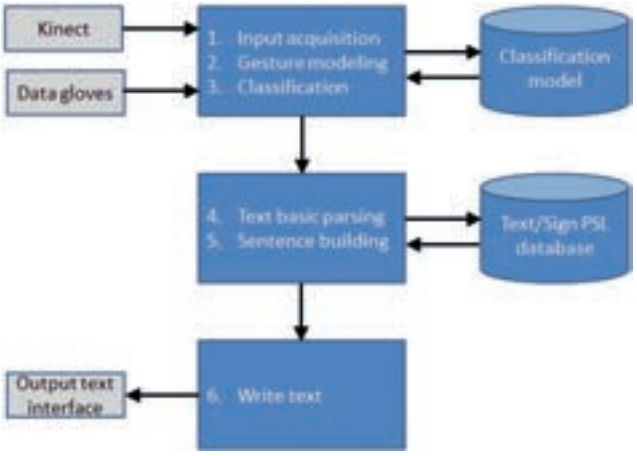

Figure 1. PSL to text translator

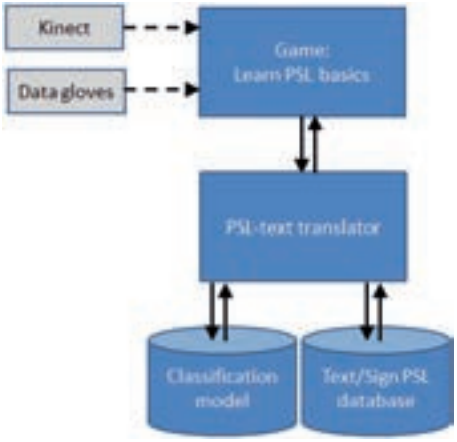

Figure 3. VirtualSign game and translator

Translator of Text to PSL (Figure 2): this module is responsible for the translation of text to PSL. The 3D models and animations used in this application to mime PSL were created in Blender. A MySQL database is used to store animation data. The main code is written in C# and all features are merged together with Unity.

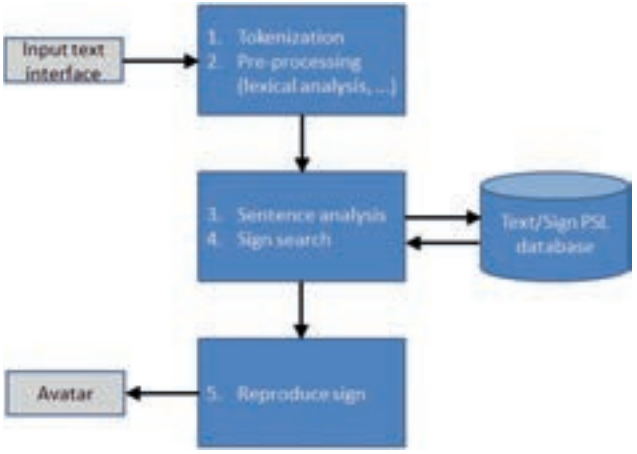

Figure 2. Text to PSL translator

Serious Game (Figure 3): Module responsible for the didactic aspects which integrates the two modules described above into a serious game. This adventure game has several challenges that bring the basics of PSL to the scene introducing the player to the PSL alphabet, commonly used words and sentences.

Gesture Classification

Once having ensured stability of the data, we proceed with the classification of the configuration. During a preparatory stage we have compared the performance of six classification algorithms, namely Random Trees (RT), Boost Cascade (BC), Neural Networks (NN), K-Nearest Neighbours (KNN), Naive Bayes (NB) and Support Vector Machine (SVM). For all these algorithms we have used the default configuration of the respective implementation available in the Open Source Computer Vision Library (OpenCV). To evaluate their performance we have used a dataset composed of 40 samples for each hand configuration (1680 samples in total). To reduce the variance of our estimates we have used 10-fold cross validation. In Table 1 and Table 2, the results of the evaluation for each glove (right and left glove) are presented.

Table 1. Classification results of the 1680 samples, obtained with the use of the left glove

%	RT	BC	NN	KNN	NB	SVM
Precision	98,6	82,0	98,1	98,8	97,5	98,6
Accuracy	85,5	95,4	78,1	97,3	97,1	100,0

Table 2. Classification results of the 1680 samples, obtained with the use of the right glove

%	RT	BC	NN	KNN	NB	SVM
Precision	98,8	86,1	97,2	98,0	98,0	98,1
Accuracy	87,3	96,6	80,4	98,2	96,8	100,0

From these results, Boost Cascade algorithm was discarded, by far the worst of all. Neural Network was

also discarded due to the high computational cost when compared to the rest. This is a serious drawback since we need a classifier to use in real time. The remaining four algorithms, present a high precision and accuracy. Based on these results we have opted to use SVM classifiers. For each configuration we have kept the top three instances and their associated probability, meaning that the application will take into consideration the tree configurations with the highest probability and their probability will be used in the classification to increase the accuracy. These instances were used later to build the classification model for word recognition.

The game

The VirtualSign game aims to teach Portuguese Sign language (PSL) in the most enjoyable and motivational way. Therefore the creation of the game had a series of factors to consider on how the gameplay should be in order to this goal and which structure the game should have. Below we describe the most important aspects of the game.

Gameplay

The gameplay is the most important part of the game. This aspect can increase or decrease how much the user will play the game, as the game designer David Perry said: *Keep the gameplay challenging, but don't let players get lost or blame the game for their problems. A good game designer always knows what the player is thinking and looking over their shoulder every step of the way*. [9]

In order to keep the game challenging there is a score system as well as a storyline so that the player feels motivated when performing a task. The score is based on the time user spends to achieve a goal, which may be clearing a checkpoint or finding a sign. The signs are spread out around the scenes and the player will need to find them with the help from the Non-Player Characters (NPC) and a map with hints. There is also always a minimap in the corner of the screen to help the player stay oriented. There are two types of checkpoints, the first type is the one required to gain access to the next area of the scene and usually requires a set of signs. The other type only requires a single sign and is used to gain another different sign. The second type of checkpoint is required to finish a level but the user can choose the order in which to clear it. Some of the gestures will also trigger minigames with score systems as well where the user will acquire a gesture depending on his score [2].

The game starts by showing the user an interface explaining the basics of the game and introduces the story where our character is told by a friendly NPC that he must find someone and in order to do so he will need to collect all gestures. Then, after introducing his name the player can start his adventure in an open world scene. All interactive items always give feedback and this feedback is always either hints for the player or a part of the story. The story has a relevant mysterious aspect to it. We tried to make sure that the story gives hints so that the player starts to realize what is happening in the virtual word but never being clear enough so that he loses interest in the story [11]. When the player acquires a sign, that sign is added to his inventory, this inventory is where the user can access and see all his gestures at any time. To see the avatar performing the gesture, the player simply needs to click the desired gesture. After each level, there is an evaluation scenario where the player knowledge will be evaluated to check if the information is being retained by the player. In those scenes, there is a NPC that will ask the player to perform certain signs, however, contrary to the checkpoints in the level, the user will not be able to access his inventory and check which gesture represents what.

Figure 1. Avatar in the first level, the score can be seen in the top, and in the corners there is the timer ***(top)*** and the minimap ***(bottom).*** The image also shows the inventory containing three gestures ***(three square images with hand configurations)*** and the avatar is performing the C sign

Scenarios

The game has three main scenarios as well as three evaluation ones. The scenarios were built in accordance to the game plot [8]. The first one is the one where the player can obtain letters and numbers and

it takes place in a desert. There are three main checkpoints where the player must perform the gestures and only after passing those checkpoints the player can access the next area of the scenario.

The second scenario is where the player starts to learn words and has to use them to acquire new gestures. It also has checkpoints, however, unlike the first where there were three checkpoints where the player had to perform six gestures in each, in this second level the player only has to perform one word per checkpoint.

As for the third and last scenario, we used the scenario provided by one of Unity assets and added what NPCs and objects needed, the logic of this scene is just as the second level except that instead using words we are now using full sentences.

Other than these three main scenes there is the starting menu, the evaluation levels and mini-games.

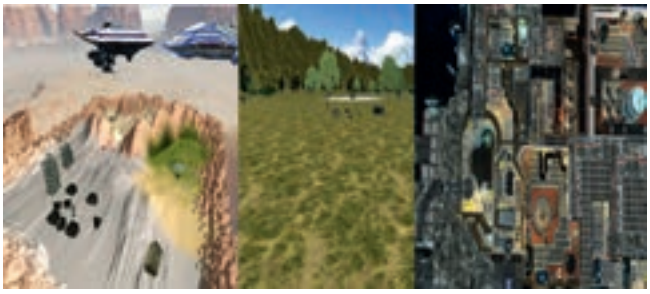

Figure 2. Three main scenarios of the game. The alphabet ***(left),*** word ***(middle)*** and sentences ***(right)*** scenarios generic view

Minigames

As mentioned before, minigames were developed within this game in order to keep the user interested and assure his motivation. Minigames are known for having a short period of playing; it has a simple set and reduced content compared to larger games. These factors make them very useful within the scope of serious games. For instance, the fact that the game has less content also means the user has fewer distractions and can focus on the content being displayed which may ease the process of learning. However, there is the need of finding the balance between the serious and leisure aspect of the game. Therefore, several mini games were developed within the main game of this project, some of those like the following memory game:

Figure 3. Gameplay of the memory minigame, the cards will turn when clicked and when a matching pair is turned the score will be increased and the cards will stay turned

This memory game has educational content as the cards represent gestures the user has previously acquired during his game and these cards have the associated letter on them. However, having educational minigames within a serious game could be overwhelming to the user. Therefore some minigames that are meant fully for leisure were added to the game, such as the following 2D shooter.

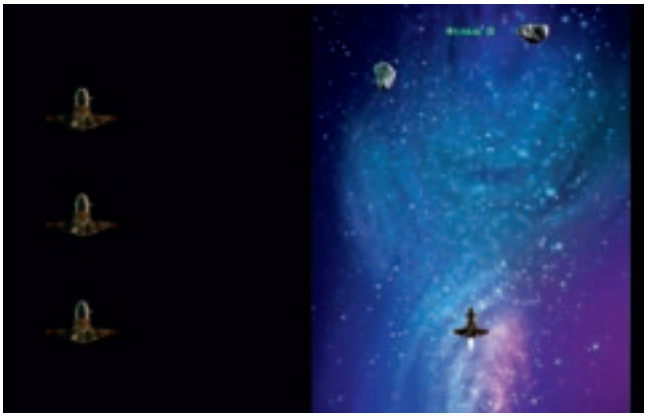

Figure 4. Gameplay of the shooter minigame, the user has to destroy objects and survive as long as he can. With each object destroyed, points to the user score are added

Technical aspects

Several technical aspects had to be taken into consideration to successfully develop this game as it involves a number of different technologies and a connection to the VirtualSign translator application. Below, there is a more deeply insight on those aspects.

Requirements

The functional requirements identified early in the project were:

- Start Menu where the player can choose the type of game he wants (with or without Kinect), consult the options or exit;
- Menu options where you can change the graphics quality, volume, save or load the game and see the table of high scores.

The game requirements within the levels are:
- Handling and control of the character;
- Interact with NPCs (Non-Player Characters);
- Consult the inventory and use the items in it;
- Interaction with map objects;
- Access to the above menu options;
- Access the mini-games.

As for the Non-Functional requirements, the usability [5], adaptability and performance were the main focus. This project aims to be fairly intuitive, allowing easy adaptation and learning. The interfaces were developed with the care to enable a pleasant interaction. The character controls were also structured to present a simple usage. Along the game, there are several short explanations of how the player should act to fulfil the objectives and surpass the levels. Every interactive object or NPC gives the player some kind of feedback. The gaming performance is a factor of the utmost importance, any perceptive delay can affect the gameplay making the game annoying rather than fun. To maintain the performance, this game was tested to never run less than 60 frames per second on an optimal computer. The ideal frame rate for a game must be around 40 frames per second [1]. The essential functions are constantly executed and the code is optimized to avoid the waste of resources.

Besides the code, all factors that constrain the performance of the game are taken into account, such as textures, bumps, number of vertices of the 3D models, among others. As for the gesture recognition, the VirtualSign translator works in real time with no delay. There is also no delay in the connection between the translator and the game, so the input from the player performing the gestures is instantaneous and can be seen on the interface at the checkpoints. In terms of adaptability, the scripting was considered and created as templates so that they can be changed in case there is the need to increase the sign count or even adding a new language. The only issue with the adaptability is the animations for the new words and new language that must be created and imported. However, this was also considered so the avatar animations are separate from the avatar itself and animation created with a similar skeleton can be played by it.

Game Layers

Given its high degree of complexity, the game was divided into layers. At the top level there is the interface. All functionalities of the project can be accessed through this layer by the user. This layer is responsible for the transmission of actions of the user for the following layers. On the lower level, there are three layers: the sockets layer which is responsible for linking Unity to the Virtualsign translator application, the game engine layer that represents Unity [4] and the business layer.

Figure 5. Application architecture layers

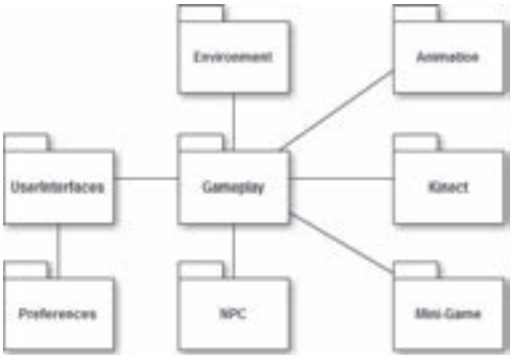

Figure 6. Package Diagram

As shown in Figure 6, the project has 7 main packages that contain the scripting for each section of the game. The environment represents the scripts for the random events within the scene, the animation as the name suggests has the base for all animations, the Interface has the user interfaces such as the main menus, inventory and quest windows. The Kinect is where the connection scripts based on sockets are stored. The NPC (non-player character) contains the scripting for the events that each NPC triggers. The

gameplay is where the basic functions are, such as the inventory, movement, interaction and highscore. The preference is very linear as it is where the audio, graphics and saves are stored and finally the Mini-games which contains all information and scripts.

Scripting

All scripts were developed considering the performance and adaptability [7] of the game. Some of the main functionalities are described below. One of the most important scripts is the inventory one, the inventory contains a list of signs, each sign contains an id and the string of the sign meaning. The inventory icons on the interface are obtained based on the id of each item therefore making the inventory adjustable. Example of the method used to add items to the inventory:

```
public void addItem(int ID, string gesture){
Item item_ = new Item(ID,textures[ID],gesture);
GameObject item = (GameObject)
Instantiate(inventoryItem);
Button b = item.GetComponent<Button>();
b.onClick.AddListener(() => ItemClicked(ID));
RawImage raw = item.GetComponent<RawImage>
();
raw.texture = textures[ID];
itens_gui.Add (item);
itens.Add(item_); inv.score.setScore(inv.
CheckTimer,timer.getTimer());
inv.CheckTimer.setTimeC(timer.getTimer());}
```

The score is incremented when the user finds a gesture as it can be seen on the code above. The information containing the current time (timer) and the time where the last gesture was found (CheckTimer) are sent to a setScore method that will calculate the score based on the difference of times. After the score is set, the checkTimer is updated to the current time.

Another crucial part of the scripting is the connection to the VirtualSign translator.

The following script shows how the text is received from the translator into the game and how it is kept and used at the checkpoints.

```
if(kinectChars.Length<passcode.Length){
kinectAtempt=kinectChars;
}else{
if(passcode==kinectChars){
correct=true;
setUnlock(true);
kinectChars="";
}else{
kinectAtempt=kinectChars;
kinectChars="";}
}
```

Since the gesture to text translator is a separate and complex application developed in C, it was easier to connect to it than to integrate it in the game as the game is being developed in Unity 3D using C#. Therefore a socket based connection was created between the game and the translator. The connection is only created at checkpoints where the player will be asked to perform the gesture. The translator sends the translated information to the game in real time, there was never any noticeable delay. On the game side, as soon as the string containing the detected translation arrives it is shown in the checkpoint interface.

The other of the main components is the translation from text, which in the game corresponds to the items the player acquires, into gestures. To assure the adaptability of the game, the avatar has a list of animations and those animations are sorted by the id of the corresponding sign. The animations are separate from the avatar and they can be simply dragged into the list of animation of the avatar and it will have access to them. The only restriction on this matter is that the animation must be created with a similar skeleton structure to the one from the avatar. As for the interaction within the game with the virtual world, all the interactive items and NPCs have colliders and when the player enters them it will show an option to interact by pressing the E key. If the players press the key, there will be always some kind of feedback. This happens because we developed template scripts that can be adjusted so new information can be added easily.

CONCLUSION

With this project, a larger support to the deaf and mute community was created and the sign language learning process now has an extra tool to support it. Since there are no many completed projects in this field, this project becomes even a greater asset for the development of Portuguese sign language. The implications for understanding the relationship between

games and learning are that games need not to be defined as an essential instrument or a type of content, but as contemporary human creations whose forms and meanings are strategic for education, more specifically concerning the hearing impaired community. The selection of this target population is due to the growing number of students with special needs, who complete the elementary and high school and enroll at the universities of higher education. This situation demands for new means that allow these individuals to have easy access to educational digital content. In order to motivate them towards the learning process, we have created a game that combines the sign language learning process with the pleasant feeling of playing a digital game. We believe this is a great time to take on the challenge of adopting new digital media, serious games and interactive simulations. However, the development of a game is always a complex task and many adversities were faced along the way. A lot of effort and time were needed to face challenges and solve problems, but a fair amount of knowledge was acquired during this process. As for future work, the game can be adapted for mobile platforms [6]. Also, as the game is in its final stage of development, a quantitative evaluation framework has been created and the next step will be to fill it based on the tests from users.

ACKNOWLEDGMENTS

This work was supported by Engineering Institute of Oporto and GILT (Graphics, Interaction and Learning Technologies) under Grant Nos. 60973089.

Paula Maria de Sá Oliveira Escudeiro

PhD in Computer Science -Information Systems and Technologies in Education Multimedia Systems, Adjunct Professor in Instituto Superior de Engenharia do Porto, in the Department of Computer Science, where she teaches disciplines of the Degree's course in Computer Engineering and Master's course in Computer Engineering - specialization in Graphics and Multimedia. Director of the Post-graduate course in Technology and Informatics Applied to Education.

Nuno Escudeiro

Teacher at IPP-ISEP and a researcher at LIAAD INESC Tech - Laboratory of Artificial Intelligence and Decision Support. PhD in Informatics from the Faculty of Engineering of the University of Porto and a MSc in Data Analysis and Decision Support Systems from the University of Economics of the University of Porto. Research activity focused on text mining and active learning.

Marcelo Norberto

Degree in Computer Engineering from the Instituto Superior de Engenharia do Porto, and attending the Masters in Graphics Systems and Multimedia at ISEP. With general interests in Game Developing and 3D Applications. Currently working in the VirtualSign more particularly in the game design.

Jorge Lopes

Degree in Computer Engineering from the Instituto Superior de Engenharia do Porto, and attending the Masters in Graphics Systems and Multimedia at ISEP. With general interests in video-games development. Currently working in the VirtualSign project with special focus in the modeling and animation.

References

[1] Claypool, K. T., Claypool, M. (2007). On frame rate and player performance in first person shooter games. Multimedia systems, vol. 13(1), pp. 3-17

[2] Creighton, R. H. (2010). Unity 3D Game Development by Example: A Seat-of-Your-Pants Manual for Building Fun, Groovy Little Games Quickly. Packt Publishing Ltd.

[3] Gee, J. P (2003). What video games have to teach us about learning and literacy, New York: Palgrave Macmillan

[4] Goldstone, W. (2009). Unity Game Development Essentials. Packt Publishing Ltd.

[5] Holm, I. (2006). Ideas and Beliefs in Architecture and Industrial design: How attitudes, orientations, and underlying assumptions shape the built environment, Oslo School of Architecture and Design.

[6] Jie, J., Yang, K., Haihui, S. (2011, October). Research on the 3D game scene optimization of mobile phone based on the Unity 3D engine. In: Computational and Information Sciences (ICCIS), 2011 International Conference on pp. 875-877. IEEE

[7] Ketfi, A., & Belkhatir, N. (2003, July). Dynamic interface adaptability in service oriented software. In: 8th International Workshop on Component-Oriented Programming (WCOP'03), Darmstadt, Germany

[8] Mariais, C., Michau, F., & Pernin, J. P. (2010). The use of game principles in the design of learning role-playing game scenarios. In: ECGBL 2010 Proceedings, pp. 462-469

[9] Perry, D. (1999). Saltzman p. 24

[10] Salen, K., Zimmerman, E. (2004). Rules of play: Game design fundamentals. Cambridge, MA: MIT Press

[11] Schell, J. (2005). Understanding entertainment: story and gameplay are one. Computers in Entertainment (CIE), vol. 3(1), pp. 6-6

Submitted: December 8, 2015.
Accepted: May 10, 2016.