

VirtualSign in Serious Games

Paula Escudeiro^(✉), Nuno Escudeiro, Marcelo Norberto, and Jorge Lopes

Departamento de Engenharia Informática, Instituto Superior de Engenharia do Porto,
Porto, Portugal
{pmo,nfe,matno,jolbl}@isep.ipp.pt

Abstract. This paper presents the game developed within the VirtualSign project. 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. To obtain those signs the player will have to overcome challenges and use the VirtualSign Translator. The player then has to perform the gesture himself and information is sent from the translator to the game. This improves the interactivity and makes the game more interesting and motivating. The game has as an inventory system where the signs are kept and can be checked. This lets the user visualize and learn or train the various existing configurations of gestures. There are also various checkpoints to check the player acquired knowledge, a highscore list and a continuous storyline.

Keywords: VirtualSign · Serious games · Portuguese Sign Language · Kinect

1 Introduction

Games have become a recognized way of providing rich content to its players, especially digital games. Digital games provide a remarkable opportunity to overcome the lack of educational digital content available for the hearing impaired community. Playing a game as the name suggest has a great leisure aspect that can't be found in conventional educational means. Educational game researcher Gee [1] 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 [2]. 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 the map. Each map represents a level and each level has several objects that represent signs scattered through the map for the player to interact with. All objects obtained by the player will be stored in his inventory and can be accessed at any point during the game. With these objects the player can then perform the gesture at the

checkpoint using the VirtualSign Translator or they can chose to play without the translator in which case the avatar will be the one performing the gestures.

2 State of the Art

There are some projects related to this theme/area but none of them implies an automatic bidirectional translation process in a game as this does, therefor, making this project very innovative. There is a rising number of serious games projects. Some of the most relevant related works within the sign language scope are described below.

The game *CopyCat* [3] 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. These are used to collect data as users sign and machine learning to recognize the signs for game play. The 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.

3 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, how the gameplay should be to reach this goal and which structure the game should have. Below are the most important aspects of the game.

3.1 Gameplay

The gameplay is the most important part of the game. This aspect can increase or decrease how much the user will to 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.* [4]

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 the 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 [5].

The game starts by showing the user an interface explaining the basics of the game and introduces the story. The character is told by a friendly NPC that he must find someone and in order to do so he will need to collect all the gestures. Then after introducing his name the player can start his adventure in an open world scene. All the 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 world but never being clear enough so that he loses interest in the story [6]. 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. The avatar is capable of



Fig. 1. Avatar in the first level, the score can be seen on the top, and on 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.

representing all the hand configurations of Portuguese Sign Language, meaning all the letters and numbers as well as a selected set of words and sentences. 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's 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 (Fig. 1).

3.2 Scenarios

The game has three main scenarios as well as three evaluation ones. The scenarios were built in accordance to the game plot [7]. 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, on 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 the 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 (Fig. 2).

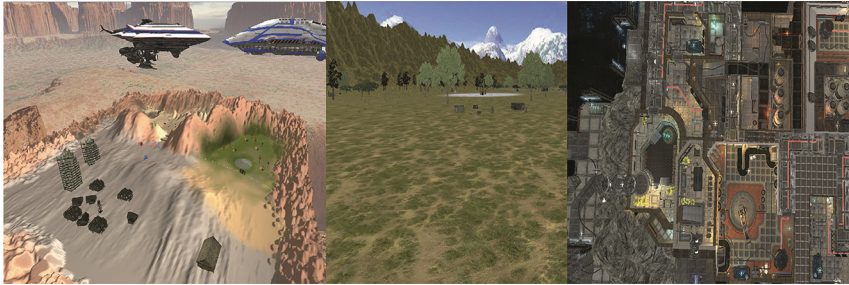


Fig. 2. Three main scenarios of the game. The alphabet (*left*), word (*middle*) and sentences (*right*) scenarios generic view.

4 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 Virtual-Sign Translator application. The game was developed in Unity 3D which is a game

development software and some of the models as well as the main avatar were created and animated on Blender. Below there is a more deeply insight on those aspects.

4.1 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 options menu;
- Access the mini-games;

As for the Non-Functional requirements the usability [8], 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 [9]. The essential functions are constantly executed and the code are optimized to avoid the waste of resources.

Besides the code all the 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 nor is there any 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 thought and created as templates so that they can be changed in case there's 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 thought so the avatar animations are separate from the avatar itself and the animation created with a similar skeleton can be played by it.

4.2 Game Layers

Given its high degree of complexity the game was divided into layers. At the top level there is the interface. All the functionalities of the project can be accessed through this layer by the user. This layer is responsible for the transmission of the 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 [10] and the business layer (Fig. 3).

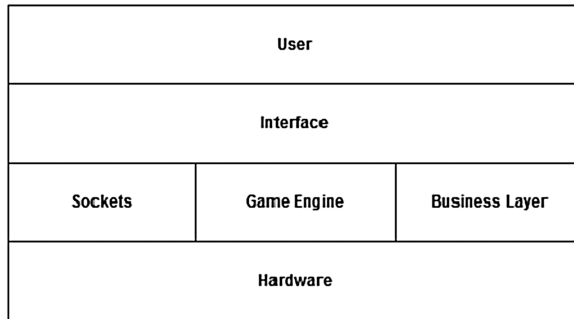


Fig. 3. Application architecture layers.

4.3 Scripting

All the scripts were developed considering the performance and adaptability [11] of the game. Below there are descriptions of some of the main functionalities. One of the most important scripts is the inventory one, the inventory contains a list of signs, each sign contains an id and 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.

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 main component 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 he 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 player presses the key there will always be some kind of feedback. This happens because we developed template scripts that can be adjusted so new information can be added easily.

5 Conclusion

The implications for understanding the relationship between games and learning are that games need not 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 both elementary and high school and come to 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 was 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 [12]. Also as the game is on its final stage of development a quantitative evaluation framework has been created and the next step will be to fill it based on tests from users.

References

1. Gee, J.P.: What Video Games Have to Teach Us About Learning and Literacy. Palgrave Macmillan, New York (2003)
2. Salen, K., Zimmerman, E.: Rules of Play: Game Design Fundamentals. MIT Press, Cambridge, MA (2004)
3. Perry, D.: Saltzman, p. 24 (1999)
4. Georgia Institute of Technology. <http://cats.gatech.edu/content/copycat>
5. Creighton, R.H.: Unity 3D Game Development by Example: A Seat-of-Your-Pants Manual for Building Fun, Groovy Little Games Quickly. Packt Publishing Ltd, Birmingham (2010)
6. Schell, J.: Understanding entertainment: story and gameplay are one. *Comput. Entertainment (CIE)* **3**(1), 6 (2005)
7. Mariais, C., Michau, F., Pernin, J.P.: The use of game principles in the design of learning role-playing game scenarios. In: Proceedings of the ECGBL 2010, pp. 462–469 (2010)
8. Holm, I.: Ideas and Beliefs in Architecture and Industrial Design: How Attitudes, Orientations, and Underlying Assumptions Shape the Built Environment. Oslo School of Architecture and Design, Oslo (2006)
9. Claypool, K.T., Claypool, M.: On frame rate and player performance in first person shooter games. *Multimedia Syst.* **13**(1), 3–17 (2007)
10. Goldstone, W.: Unity Game Development Essentials. Packt Publishing Ltd., Birmingham (2009)
11. Ketfi, A., Belkhatir, N.: Dynamic interface adaptability in service oriented software. In: 8th International Workshop on Component-Oriented Programming (WCOP 2003), Darmstadt, Germany, July 2003
12. Jie, J., Yang, K., Haihui, S.: Research on the 3D game scene optimization of mobile phone based on the unity 3D engine. In: 2011 International Conference on Computational and Information Sciences (ICIS), pp. 875–877. IEEE, October 2011