Project Report - Asymmetrical Escape Room

Jonathan Collaud, Grégoire Hirt, and Lucio Romerio

CSS-444 Virtual Reality, Group 2, EPFL

Spring 2018

1 Introduction

The goal of this report is to present the game for *HTC Vive* that we developed, using Unity and *SteamVR*, in the context of the EPFL course *CSS-444 Virtual Reality*. In the following we will try to explain the game itself, while pointing out the main features we implemented.

2 Game Description

Our project consists in an asymmetrical escape room game. The scenario is a spy mission and it involves two players: the *VR player* and the *Oracle player*. The first is inside the VR world (using the *HTC Vive*) and acts as the spy. The second - the so called *Oracle player* - act has the guy behind the scenes, which guides the spy through its mission. Obviously, in order to accomplish the mission and reach their goal, the two players have to collaborate and communicate.

2.1 Commands & Interactions

VR player

Figure 1 resumes the commands allowing the *VR player* to interact with the VR world.

Some objects - like books for example - can be grabbed using the *Trigger* button. In order to provide a visual feedback, we added a little laser on the top of the controllers, and we used a simple ray-cast mechanism to determinate whether an object is in range of this laser. In case it is, the object becomes highlighted and can be grabbed, once grabbed it will smoothly move on the laser until reaching its edge.

A subset of objects, beside being grabbed, can also be stored. Actually, by pressing the left *Trackpad* the *VR player* can open its inventory, which comes with five slots. The slots are shown as transparent spheres, to insert an object into the inventory it suffices to release it inside one of such spheres. To extract an object from the inventory, the player simply has to grab it.

Obviously there are some objects which cannot be grabbed (such as tables, closets, or even walls). Nevertheless we thought it was important to give a feedback to the user, to let him know he is touching/hurting something. For this reason we introduced an haptic feedback: whenever the laser described above collides with a physical object, the concerned controller vibrates. In case the colliding object can be grabbed, this vibration blends nicely with the highlighting, giving an additional feedback to the user.

Another key interaction in the game is the movement, which is done through a teleportation system. We tried to make this system as intuitive as possible: pressing the right *Grip button* the *VR player* can use a laser to target the place where he want to teleport, when releasing the button it will immediately move to this location. In order to provide the best visual feedback possible, together with the laser ray comes a circle which hopefully makes it easier to visualize the teleport location. Also it changes color indicating whether the selected location is a valid one (obviously the player cannot teleport on walls, tables, etc).

Finally, by pressing the *Menu button*, the player can transform its controller into a plier and use the *Trigger* button to cut (there are some cables to cut in the game).

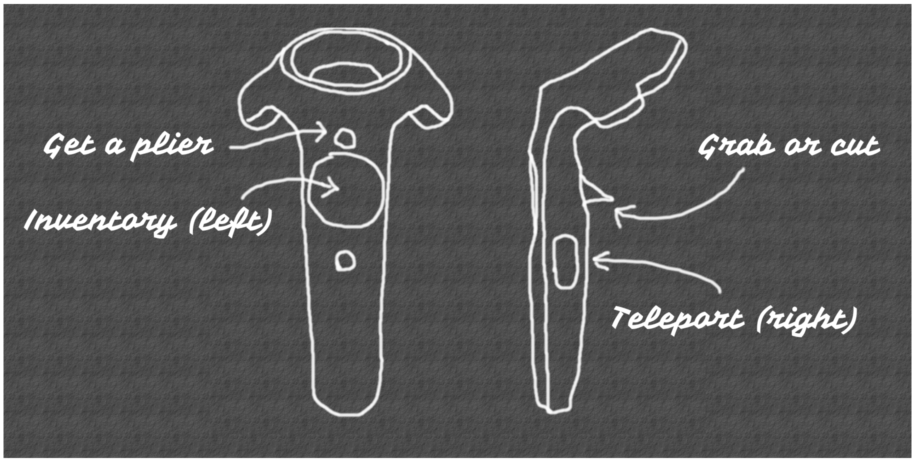

Figure 1: Commands available to the *VR player* through the *HTC Vive* controllers

Oracle player

The *Oracle player* has a top view of the *VR player* and a console at his disposal. Thanks to his point of view he can help the other player moving in the VR world, with which he can interact using the console. More precisely he can execute the following two commands:

- **pc <id> hack** , where <id> is the ID of the computer he wants to hack;
- **ligth <id> <action>** , where <id> is the ID of the light and <action> *on* or *off* depending on his intent.

In order to enhance the user experience we added: a simple command history (which can be navigated thanks to the *UP* and *DOWN* arrows); a color feedback, green for correct commands and red for wrong ones; an hint mechanism which will either show the command syntax or signal wrong arguments in case an invalid command is inserted.

2.2 Scenario & Plot

When the game starts, the *VR player* find himself in a "tutorial room". There is Figure 1 on a blackboard so that he can try the commands and understand how to interact with the VR world before the real game starts. Once he is ready he can leave the room, as soon as he does that, the timer will start. The goal of the game is to find the secret room before the time expires, to do that the two players need to solve a series of puzzles. In the following lines we basically present a complete walkthrough of the game, thus consider jumping directly to the conclusion section and come back later if you plan to try the game.

First the *VR player* has to find a computer and communicate the ID of the latter to the *Oracle*, so that he can hack it. Hacking the computer (there is only one in the whole scenario), will result in printing the light specifications. There is a room with a printer: the light specifications will appear there. Those consist in a simple paper in which are written both the command that the oracle has to use to interact with the lights, and the lights IDs. After turning off all the light in the hallway, an hidden message, visible only by the *Oracle*, will appear. This message indicates where to find the alarm cables, and which cables to cut. At this point the *VR player* has to find the key of the cable box, so that he can open it and follow the instructions of the *Oracle*. In the hallway there are a few chest of drawers, above every chest there is a poster. The poster indicates which object can be found inside the drawers of the chest. Thus, the *VR player* has to found the poster in which a key is illustrated, and look in the drawers under it. Once the key is found, he can finally cut the alarm cables. Cutting the correct cables will reveal the hidden room: you won. Cutting a wrong cable or running out of time will fire the alarm: you lost and you have to restart from the beginning. Note that, in order to keep track of the remaining time, there are timers a bit every where in the building.

3 Conclusion

We really enjoyed coding this game and preparing the puzzles.

All of us were new to Unity, thus we often had to check the documentation but fortunately Unity is quite intuitive. In particular we enjoyed the physic system, which allowed us to easily obtain a realistic result and made quite straightforward to insert all the appropriate sounds (ambiance sounds, but also the noise of objects hitting the floor and so on). Another point in favor of Unity is that it lets you import your Blender models into your application. We took advantage of this aspect in different occasions; for example the building, the plier and the cables are all Blender models done by us.

One of the most delicate aspect of such a project, is that cybersickness is always behind the corner. We really tried our best, exploring different solutions, to make the user experience the most enjoyable as possible and to avoid falling in this phenomena.