INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO
GALWAY-MAYO INSTITUTE OF TECHNOLOGY

Gesture Based UI

Blaine Burke, Arnas Steponavicius
G00354397, G00361891

https://github.com/BurkeBlaine1999/Jungle-Gym

1 Introduction

For the Gesture Based UI Development project we have created a Virtual Reality (VR) game. The game incorporates gestures such as grabbing, pointing, pressing and voice recognition using the VR controllers and microphone. We were hoping on using VR and voice recognition technologies to create a fun family game.

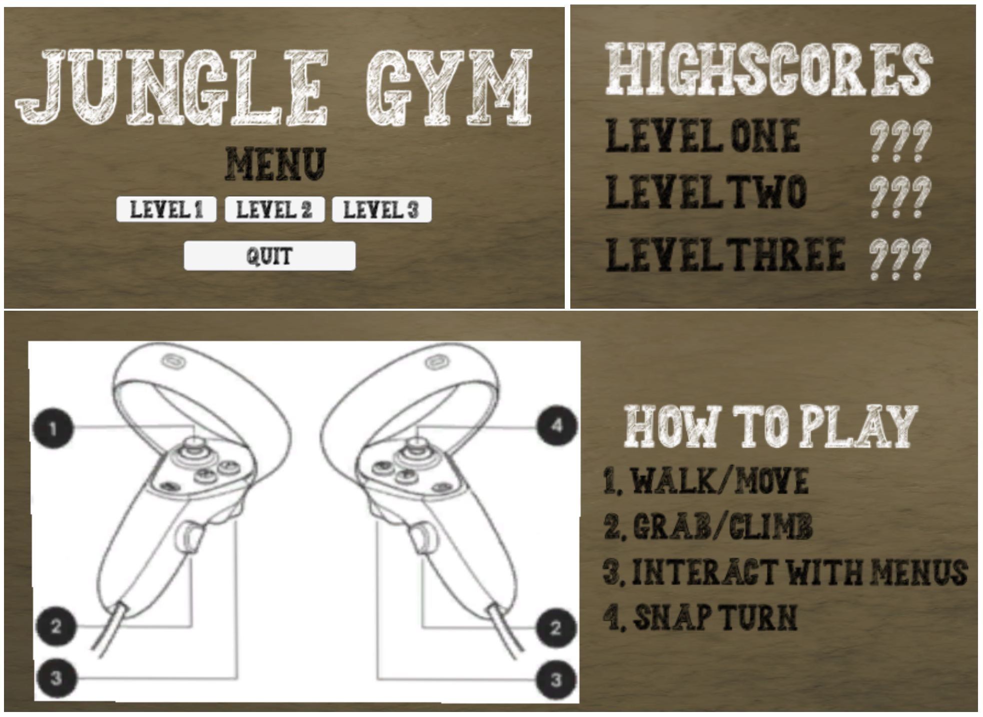
2 Purpose of the application

The game is called 'Jungle Gym', it is a family friendly virtual reality climbing game. The player must race against the clock to reach the end buzzer as fast as possible in order to achieve the high score. The game aims to incorporate gestures that would be used in climbing through the use of virtual reality technology such as the Oculus Quest 5 and the use of voice commands for certain scenarios.

2.1 Starting Area

The starter area is an open room for the player to explore and practice the game mechanics. It serves as a tutorial without having the need to go through a predefined route, allowing the player to just get stuck in to the game. There are several different interfaces displayed in the starting area that explains to the player the controls and shows the menus.

- The Main Menu, for the player to choose levels and exit the game.
- The high score interface, for the player to view their high scores.
- The controls interface where the player can see all controls of the game.



This area also allows the player to hone their skills, as the area is created using all the different obstacles that can be found throughout the game. We also decided to put the menu systems in this area to make them more accessible. This also eliminates the need to use canvases.

2.2 In-game interfaces

There are many different interfaces in the game such as the menus that allow the player to return to the main menu area but also the timers displayed on the walls. The end timer also changes colour depending on if the player has beaten the high score or not.

3 Appropriate Gestures for this application

There are a few gestures incorporated into this game that can be considered appropriate. The gestures include grabbing, pointing, pressing and using voice to go through menus or the start the game if the user desires. Each hand gestures has its own animation such as clenching their fist for grabbing.

3.1 Grabbing

Grabbing is the primary gesture in this game. When interacting with climbable objects they must press down the grip button while their hand is in reach of the object to latch onto it. They can then manipulate their body position with their hands by moving them in any direction. The animation used here is a fist animation to simulate the player grabbing onto the wall holds.

3.2 Pointing

The pointing gesture is showcased in this game by its use in interface navigation. The player uses the controller to point to a button on the interface. They will see a blue laser appear from the hand while pointing at interfaces whether its an options screen or selecting a level. We decided on the pointing gesture to act as a way to interact with interfaces, as humans use pointing as a way to let their intentions be known. Therefore it only made sense to use pointing as a gesture to select buttons on the interface.

3.3 Selecting

The selecting gesture is showcased by pressing the index trigger button on the right or left hand controller. When the player is pointing at a button on a menu, they use the trigger button on the pointing controller to select that option. The gesture used when pressing the trigger is a pinching gesture to simulate the player picking something.

3.4 Voice Recognition

We decided to incorporate voice recognition into this project to give the player a hands off experience when interacting with the game. While the voice recognition is not a part of the main aspect which is climbing, it does allow the user to restart the level or quit the game without having to return to any menus.

4 Hardware used in creating the application

We looked at many different types of hardware and what we could use them for. There were many things we looked at but the main two were the Xbox Kinect and the Oculus quest 2.

4.1 Oculus Quest 2

Virtual reality is a relatively new technology with a lot of future potential. It allows users to fully immerse themselves into the game world. Although they are still expensive starting at 350 euro for the quest which cannot run certain games properly due to it not being powerful enough therefore you would require a VR ready computer which would cost approximately 700 euro.

Virtual reality games are not simple to make however as you must take in many factors when developing it such as the acceleration of the input devices, rotations and positions constantly. This factor alone would deter some from engaging in creating a virtual reality game however we are both interested in VR and its capabilities. There is also a lot of useful documentation and Unity packages that aid in the creation of VR programs in Unity.

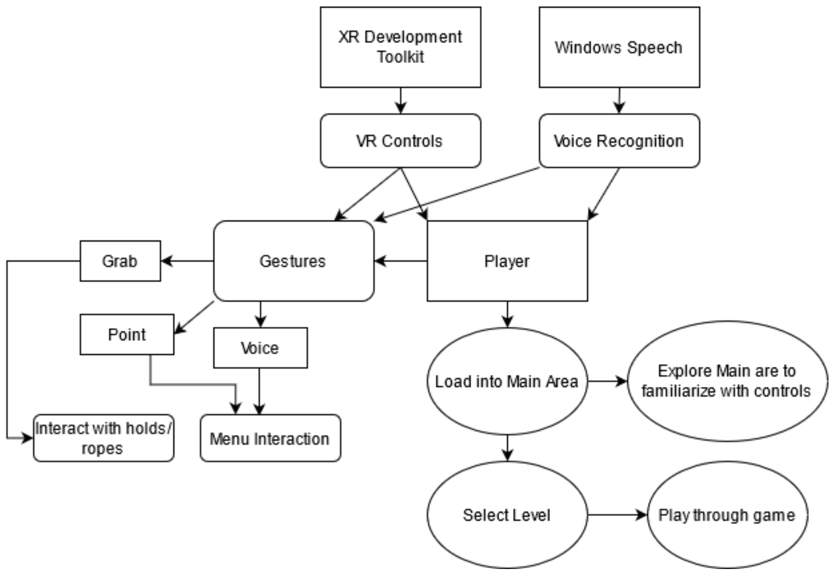
4.2 Xbox Kinect

The Xbox Kinect was revolutionary at its time of release as it was the first of its kind. It was the closest users felt to being immersed in a game by following their body movements. Although it has recently been replaced by its successor the Kinect for the Xbox One. We had many ideas for the Kinect such as a goal keeper game where the player must block balls from entering the goals. However, we were swayed against using the Kinect for this project as it is a considerably older technology compared to VR. Since the technology is older, We researched as to whether the documentation is still being maintained and did not find any recent posts regarding the Kinects upkeep. Adding to our decision of not using this technology and opting for VR instead.

4.3 Windows Speech Engine

The Windows Speech engine that is built into Unity was another technology we incorporated into the project. We used speech as an alternative way to navigate between levels in this game. Our reasoning was to add another more convenient way of navigating between levels through voice since the user would not have to run back to checkpoints if they desired to restart the levels.

5 Architecture for the solution



The main technology used in this project was the Oculus Quest 2 and the Unity package called XR Interaction version 0.9.4. The XR Interaction package version has to be 0.9.4 for the game to compile inside the Unity editor. Initially we researched the Oculus Integration SDK and the XR Development Toolkit for implementing virtual reality into our game. However, we ultimately decided on using the XR Development Toolkit as we found the toolkit to be easier to use and understand when reading documentation or following tutorials compared to the Oculus Integration SDK.

The diagram above displays the relationship between the different components of the game. We used the XR Development Kit and the Unity Windows Speech engine to control the VR and voice controls. The controls are attached to the player meaning they are always looking for the next input. The Player can use the defined gestures which are grab, point and voice recognition. The grab gesture is used to interact with the wall holds and the ropes in the game. While grabbing the player is also able to shift around their body to manoeuvre into a better position. The point and voice gesture are used to control the interaction with the Menu System. By pointing the player can click on the trigger to select their desired option or they can use their voice to activate a function that is otherwise activated on pressing a button.

The controls are displayed in the area the player loads into initially. We decided on removing a tutorial which we initially planned to have and instead we replaced it with a wide open area for the player to explore and familiarise themselves with the controls at their own pace.

6 Conclusions & Recommendations

To conclude this document, there were multiple artifacts of information we learned through the making of this project. The most enjoyable part of this project was being able to work in a group together. This allowed us to discuss and delegate different ideas or tasks amongst each other. It also allowed us to consult each other on issues we were facing whether it was the coding aspect of the project, level design or general issues. It was also a very good learning experience of working with virtual reality and cutting edge technologies such as the XR Development toolkit. It made us have to do more research and critical thinking throughout the course of the project. We had to think of different ways of overcoming obstacles and issues we were encountering. An aspect we would have changed for the project, is we would look into a better voice recognition library for Unity. As the built-in Windows Speech technology can be slow to react and execute functions when a voice input is given. There are also issues with the speech engine not being able to detect voice commands if the player says a sentence and then says the command a second later or if the player has an unfamiliar accent due to it using the United States language pack.