Intelligent Interfaces Assignment Paper

Deborah Vella
Artificial Intelligence Department
University of Malta
Malta
deborah.vella.17@um.edu.mt

Valerija Holomjova
Artificial Intelligence Department
University of Malta
Malta
valerija.holomjova.17@um.edu.mt

ABSTRACT

The following report describes the project we created and presented at the MRO for the Intelligent Interfaces study unit. For our project, a 3D runner game, called 'Run Box Run', was implemented using Unity which utilizes the hand detection features of the Leap Motion device. The device allows the user to navigate through the game and interface using hand gestures.

1. INTRODUCTION

For the assignment, a 3D runner game was implemented using Unity. The base of the game was built from the following tutorial [5]. Additional features were then added to the game such as randomly generated unique obstacles, a point system, an interactive and user-friendly UI, a tutorial section and more. The player interacts with the game using the Leap Motion device [3]. A Leap Motion device is a computer hardware sensor device that supports hand and finger motions as input. This device was used so that the player could interact and navigate through the game in a natural manner so even users who are not experienced in gaming could be able to play the game.

1.1 Aims and Goals

The goal of our presentation was the following:

- To create a unique and interactive user experience.
- To create a UI that is easy to comprehend and navigate through.
- To implement natural hand motions that are understandable for a wide variety of users - for instance when the user moves their hand to the right the player should move to the right.

The target audience of our game is mostly directed towards children but the game can be appealing to different age groups as well.

1.2 Limitations

The following limitations of our game are as follows:

- The game was implemented as a single player game, hence, only one user can play at a time.
- Certain hand gestures are not recognizable by the device - hence there was a limit to the hand gestures we could use and implement on our own.
- Sometimes the hand gesture detection of the device can be inaccurate, however, this is rarely the case.

2. BACKGROUND RESEARCH

2.1 Similar Works

The Leap Motion Controller is commonly paired with the Oculus Rift in the creation of VR Games. The implementation of VR in games can make the game more exciting and immersive. Some top games which support Leap Motion are:

- Hollow - uses Oculus VR and Leap Motion hand gestures.
- Pin Pon - requires Leap Motion controller
- Gourmet Quest - uses Oculus Rift and Leap Motion device [6]

2.2 Technologies Used

Unity [4] is a cross-platform game engine and was used to build the actual game along with the User Interface.To use the features of the Leap Motion device, the Leap Motion driver had to be installed [1]. To integrate the Leap Motion device with Unity we had to install a Unity Module from the following website [7]. Please note that an old version was used because newer versions did not support hand gesture recognition. An asset package was also taken from the same site that provided us with the hand models and hand controllers that we used in our game. The hand controller would render the hand models as predicted by the Leap Motion device into the game. Scripts were attached to the hand controller to retrieve data from the Leap Motion device such as the position of the fingers, the distance of the hand, any detected hand gestures, etc. The hand input data was then used to interact with objects within the game such as moving the player and selecting buttons.

3. IMPLEMENTATION AND TESTING

3.1 The Main Menu

The user is welcomed by a start menu providing a choice to either start the game or start the tutorial. Hence, whoever already knows the game controls and rules can start playing immediately, while those who are new to it can become familiar to the rules by following a tutorial. The user has to choose between the two options by pointing towards the button they want to choose and pressing it.

Figure 1. Main Menu, having the user about to choose the start button.

3.2 The Tutorial

The tutorial contains of 4 different screens, explaining the movement controls, jumping, the appearance of obstacles, and the appearance of collectibles. The user may transition from one screen to another by either swiping left to go back, or swiping right to go to the next screen. Each tutorial scene contains some animations to help portray the actions visually for the player to understand thoroughly. When the tutorial is finished, the game scene is loaded so that the user can start playing whenever they are ready.

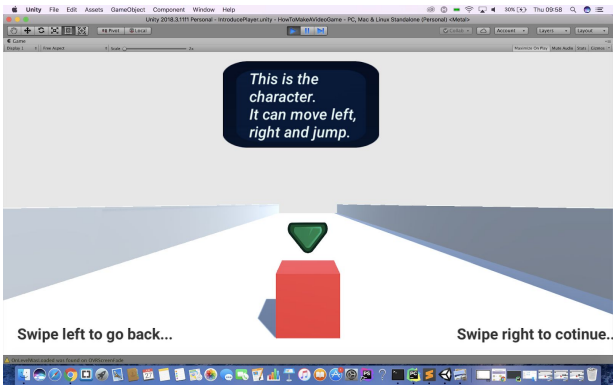

Figure 2. Screenshot of the character description.

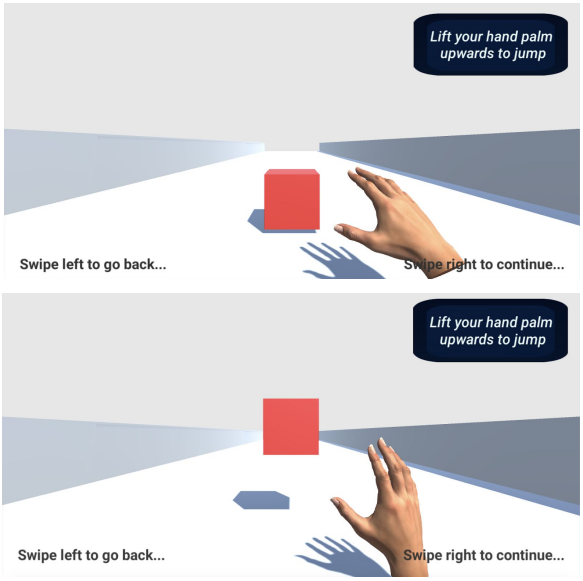

Figure 3. This is the jumping tutorial. In the top screen there is the beginning of the animation. On the bottom screen there is the hand moving upwards and the player moving with it.

3.3 The Game

3.3.1 Player Movement

At the start of the game the player is prompted to place their hand at the center of the device and swipe upwards to start. Once the player completes the specified actions the game starts. The user can move the player left and right by moving their hand left and right respectively above the Leap Motion device. Additionally, if the user swipes up the player will jump. These hand movements were chosen so that the game is played in the most natural way. Walls were created in the game to notify the user on how far the player can move horizontally.

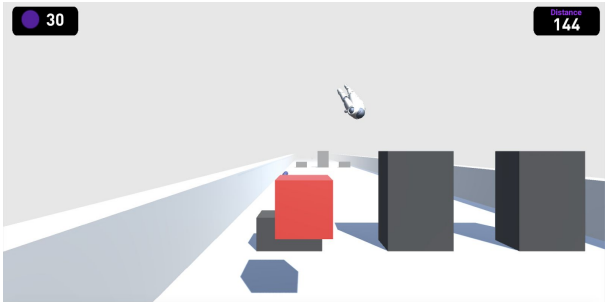

Figure 4. Screenshot of the Player jumping after a Swipe Up Motion

3.3.2 Obstacles and score system

The player has to dodge incoming obstacles, which can take one of two predefined shapes, and collect as many points as possible.

The obstacles in the game are generated randomly each time the game starts, as well as, the locations of the generated points. Object pooling was implemented in the game to increase its speed and performance. Object Pooling plays a huge role in creating a more dynamic game environment, rather than having static obstacles and points locations which never change.

3.4 In-Game UI elements

UI elements were added to the game so the players could keep track of the points they collected (see Figure 4 in the top left corner) and the distance they have travelled (see Figure 4 in the top right corner).

3.4.1 Game Paused

The user can pause the game by removing his hand from the Leap Motion's detection range. During this time, the player character stops moving forward. This was implemented in cases for when the player requires a break, or when the Leap Motion device is having issues detecting the hand. The game state is saved without allowing the character to potentially hit obstacles while the user is not playing. Once the hand is detected again, the game will resume from its previous paused state.

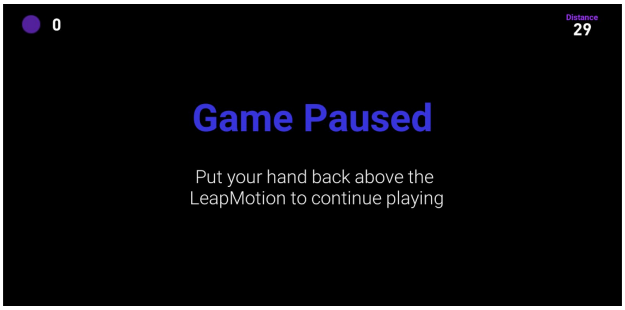

Figure 5. Screenshot of the game's Pause Screen when the User's hand is undetected

3.4.2 Game Over

If the player fails to avoid an obstacle, the game finishes resulting and a game over screen is displayed, accompanied with a game over sound. Here the user can see his score and total distance travelled to encourage him to get a better score the next time he plays. This game over screen later transitions back to the start menu.

Figure 6. Screenshot of the game's Game Over Screen after the player collides with an obstacle.

3.4.3 Music and Particle system

For a more enjoyable user experience, background music and a particle system were added to the game. The particle system comes into action whenever the player collects points followed by a happy sound, stimulating the user into collecting more points. The particle system is accompanied by a change of color in the score UI (see Figure 7. Top left corner).

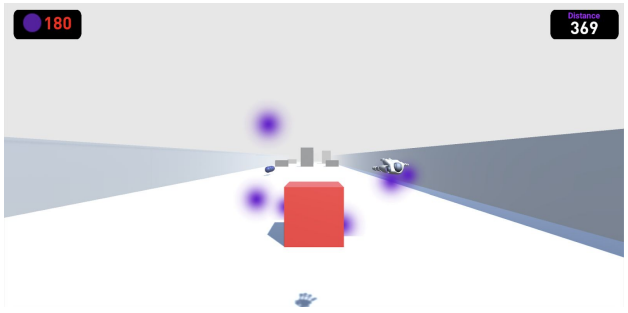

Figure 7. Screenshot of the player collecting points.

3.4.4 Fonts

Every text that is shown in every scene has a font applied to it. Text is vital to make the game as user friendly as possible while the fonts are necessary to convey the game's atmosphere and character. The users usually are not aware that the fonts are a stimulus that play a huge role in the game whether they are immersed in the game or bored by it. Therefore, for this runner game, it was made sure that the fonts chosen bring out the theme we wanted to create and fitting aesthetically in our game world. Different fonts have been used to give out instructions, distribute scores and provide the required information. Above all, all fonts had to be legible across different platforms, allowing the user to read everything with ease. The fonts used were downloaded from dafont website[2].

3.3.5 Level Complete

The player movement becomes faster each time a certain distance is reached, indulging the user more into the game, by increasing his focus and desire to win. If the user manages to

avoid all the obstacles until the endpoint is reached, the game has been won, resulting in a level complete screen to be displayed, complimented by a level complete sound. The total score and distance travelled are shown for the user to evaluate how good his performance was. This screen, then transitions back to the start menu.

Figure 8. Screenshot of the Level Complete scene when the final endpoint is reached.

3.4 Testing

Each aspect of the game was tested as soon as it was implemented and the development of the game resumed when the prior part were verified to be working properly. First the basis of the game was implemented, controlling the player with normal arrow keys. When this was tested and was working perfectly fine, the hand movement using the Leap Motion controller replaced the arrow keys controls. This hand movement was tested over and over again and perfected after each bug that was encountered. When all of this was functioning as it should, we moved on to implementing a better user interface adding sounds, particle systems, different screens, scores and more. Each UI element was also tested for its functionality and updated accordingly. The main menu scene was also analysed to make sure the buttons are transitioning to the right screen when pressed by the human gesture. Last but not least, the tutorial scenes were evaluated one by one, ensuring the animations were played as they should. Simultaneously, the transitions between tutorial scenes using hand gestures were tested.

After the game was tested and finalised, we continued the testing by allowing other students to test the game and give us feedback about it. Furthermore, we gained more feedback from people who used it at the MRO and continued to make changes to the game based on the users' opinions. One example of this is that some children found the player speed to be rather slow, which we immediately changed it to increase by some units after a specific distance is travelled. When this improvement was added, more children started to enjoy the game experience.

4. EVALUATION AND CRITICAL ANALYSIS

The interaction between the user's hand and the game worked very smoothly. The only issues we had in terms of precision was the fact that sometimes the selection gesture in start menu was undetected but this was due to the Leap Motion device having difficulty registering the complicated gesture. This was a similar issue with the player jump in the game (when the hand is moving up). However, if precise gesture movements are made by the user the performance of the Leap Motion detection was improved.

5. CONCLUSION

Overall, the project we implemented managed to achieve all of its goals and tasks. Some of the improvements of the game were made after feedback from participants that played our game at the MRO. One of the improvements they requested was to make the player move faster since initially he moved at a constant pace. Hence, we decided to make the player speed up slightly at certain distances to make the game more challenging but not impossible for younger users.

Other improvements for future versions of our project that we considered but haven't managed to implement are the following:

- Implementing the game as a multiplayer game so that two participants could race each other at once.
- Creating different fun game modes where the user could try different hand interactions such as picking up objects.

REFERENCES

1.Desktop setup - Leap Motion. *Leap Motion*, 2019. https://www.leapmotion.com/setup/desktop/.

2.Fancy fonts | dafont.com. Dafont.com, 2019. https://www.dafont.com/mtheme.php?id=1.

3.LeapMotion.LeapMotion,2019. https://www.leapmotion.com/.

4.Technologies, U. Unity - Unity. *Unity*, 2019. https://unity.com/.

5.Thirslund, A. How To Make A Video Game | Dev Assets.Devassets.com,2019.http://devassets.com/assets/how-to-make-a-video-game/.

6.Top games with Leap Motion support. itch.io, 2019. https://itch.io/games/leap-motion.

7.Unity — Leap Motion Developer. *Leap Motion Developer*,2019.https://developer.leapmotion.com/unity.

6. Appendix

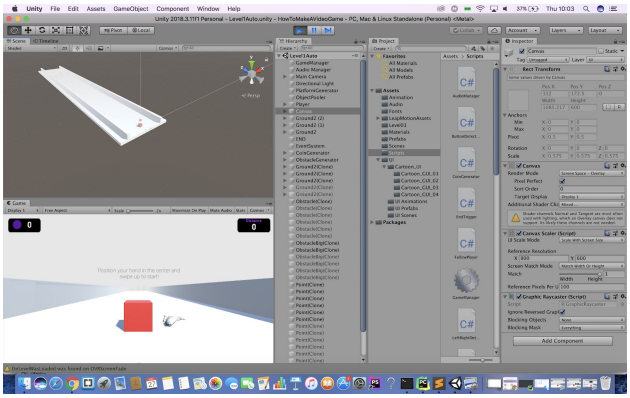

Figure 9. Object Pooling during running game

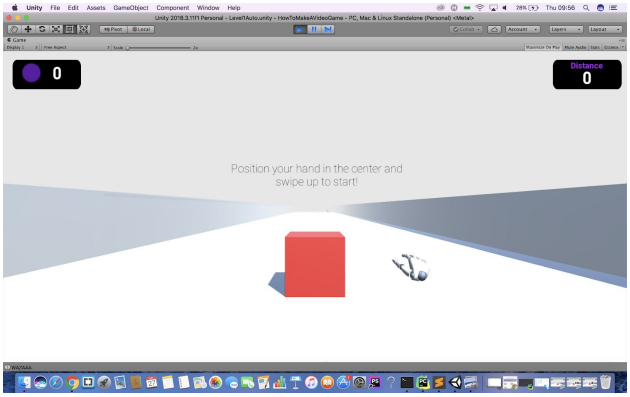

Figure 10. Instructing the users on how to start the game

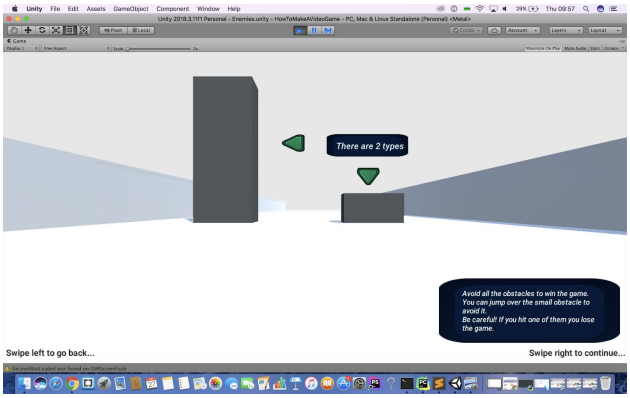

Figure 11. A tutorial scene explaining the obstacles.

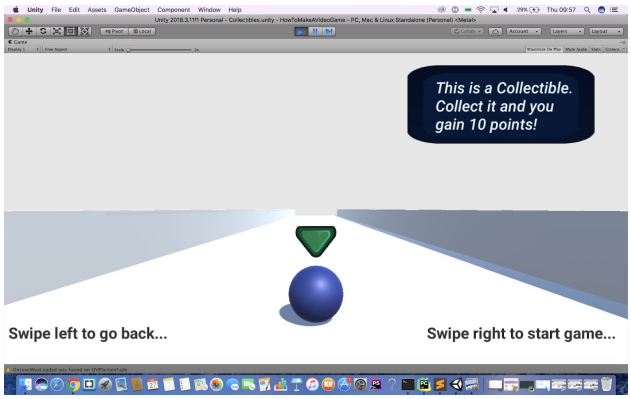

Figure 12. A tutorial scene explaining the collectible.

Figure 13. A tutorial animation that is triggered when the user swipes to the next scene. In this case it shows that the next part will deal with the explanation of obstacles

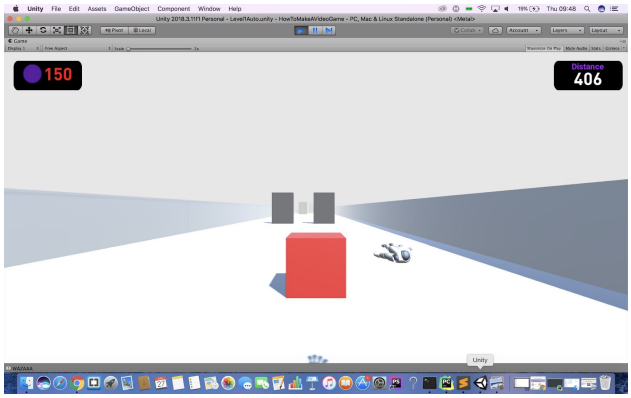

Figure 14. A screenshot taken during gameplay, having the point score being 150 and the distance travelled being 406.

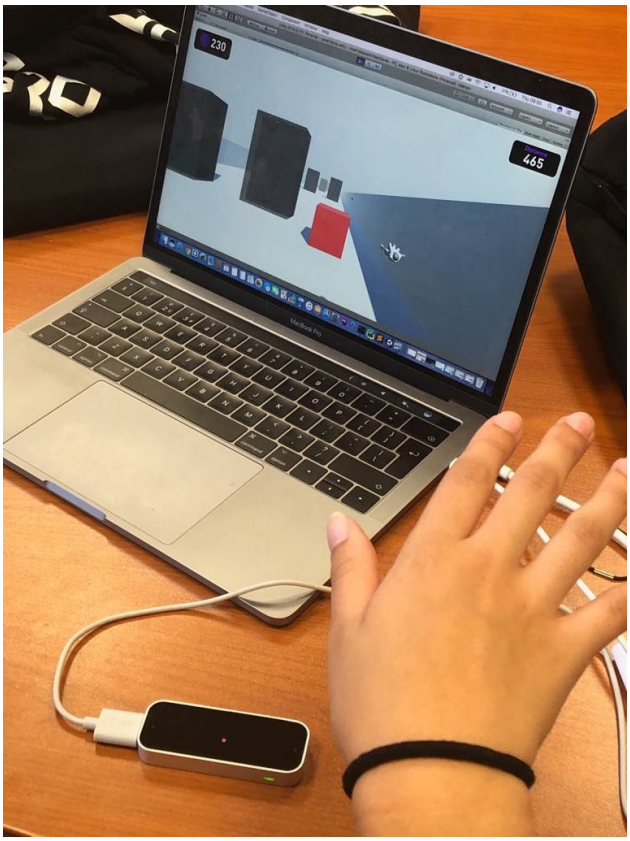

Figure 15. A demonstration of a user interacting with the game - the position of the player in the game matches the position of the hand (towards the right).

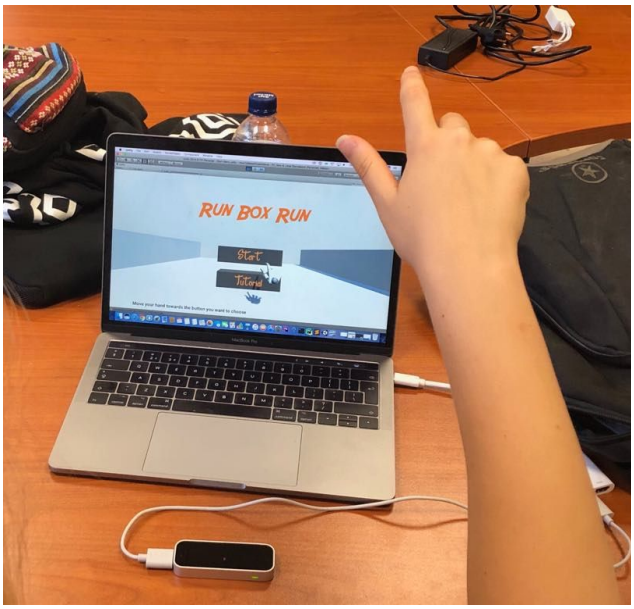

Figure 16. An example of a user interacting with the start menu by selecting a button.

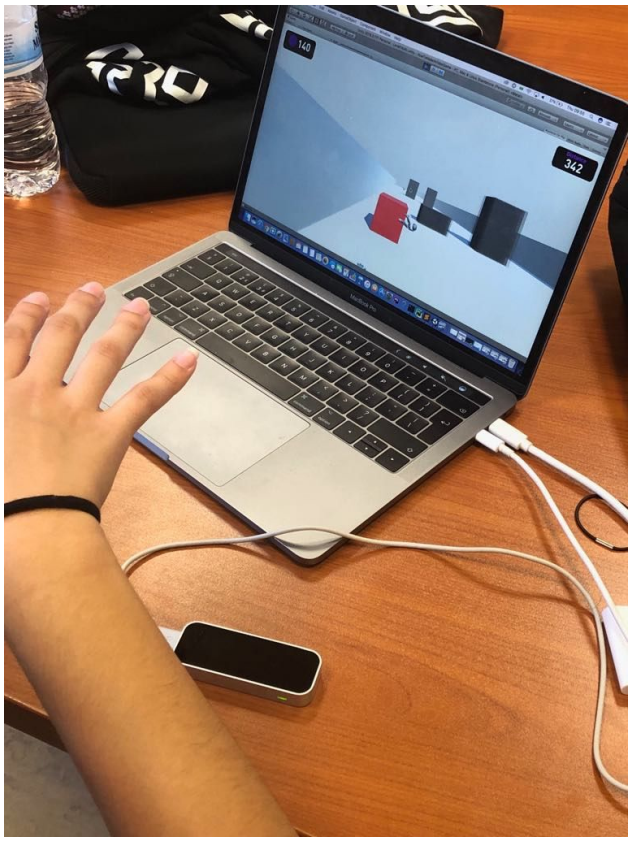

Figure 17. A demonstration of a user interacting with the game - the position of the player in the game matches the position of the hand (towards the left this time).