

LO1: Explore the use of game engines in real-world projects.

CUBETHON:

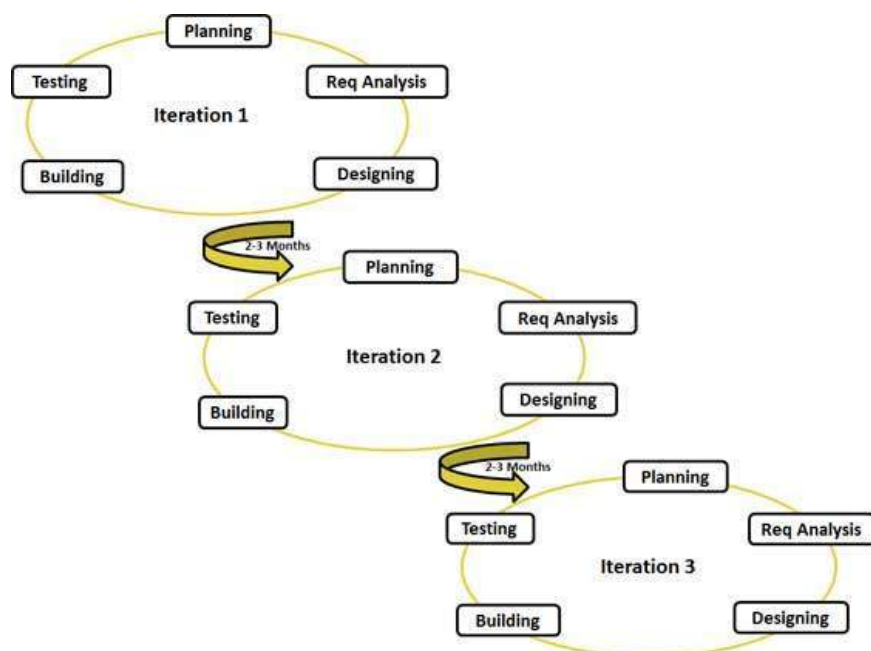
The game I will be making a replica of is CUBETHON. To make it work I will be following a tutorial made by Brackeys on YouTube. Trying to expand it by adding Level Select, Pause Menu, Music and Mute Button to mute the sound. To succeed in this game, you need good reflexes and a rhythm. I chose this type of game as I really enjoy playing arcade games.

Part 1: Agile Model:

The Agile model believes that every project needs to be managed differently and the existing methods need to be created to fit best the project requirements. In the agile model, tasks are divided to time boxes "smalltimeframes" to deliver definite features for a release.

Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

This is a graphical illustration of the Agile Model:



The Agile principles are:

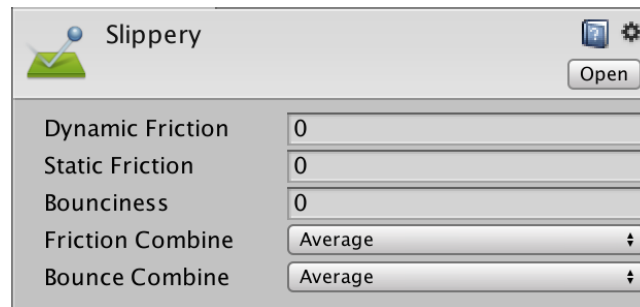
- **Individuals and interactions** - in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- **Working software** - Demo working software is considered to be the best means of communication with the customer to understand their requirement, rather than that of just depending on documentation.
- **Customer collaboration** - As the requirements cannot be collected completely in the beginning of the project due to different factors, continuous customer interaction is very important to get appropriate product requirements.
- **Responding to change** - agile development is focused on quick responses to change and continuous development.

The agile model has both advantages and disadvantages. I think that the developers of CUBETHON game used Agile model as it can be developed rapidly and demonstrated not too much complications. I think they started to build the game by simply planning some few points about what the basic game mechanics are. For example, moving left or right, avoid obstacles and complete the level. The resource requirements are to the minimum. It is a simple game as using only controls to move to the left or right and to try to avoid obstacles. The purpose of the agile is that you can fix or change requirements during the construction of the game. It is a good model for environments that change steadily as you can create environments how much as you want during the game. It also gives a good flexibility to the developers, so they can work more comfortable on the game. On the other hand, the disadvantages are that is very individual dependency as there is a minimum documentation generated. For example, if you are making the game with someone else or you want the lecturer to check the game for you it may be quite challenging because of the lack of the documentation. There is more risk of sustainability, maintainability and extensibility while creating the game.

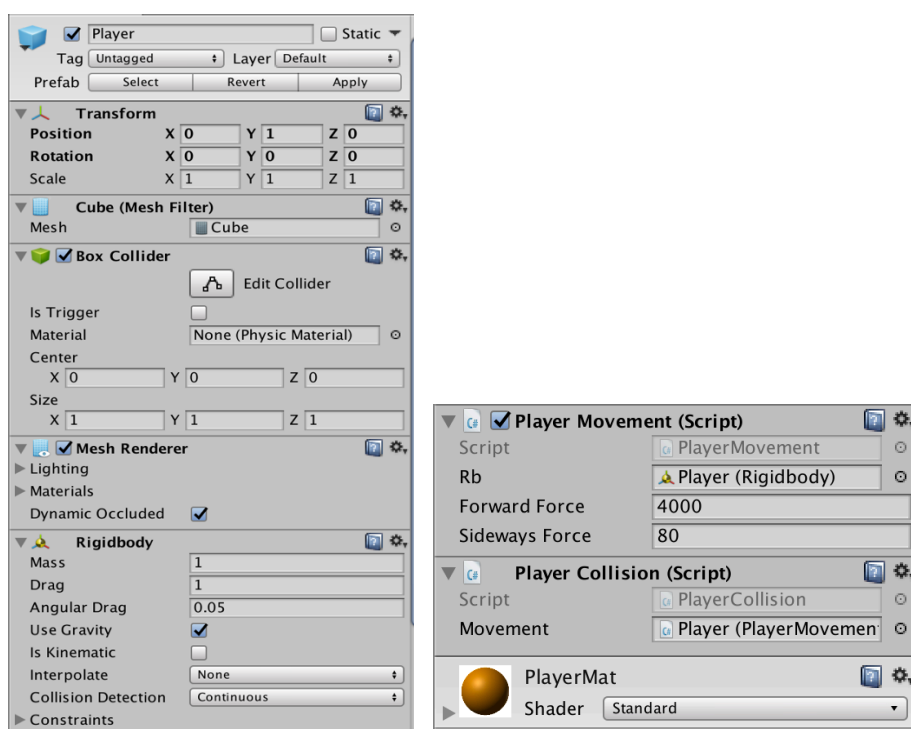
Part 2:

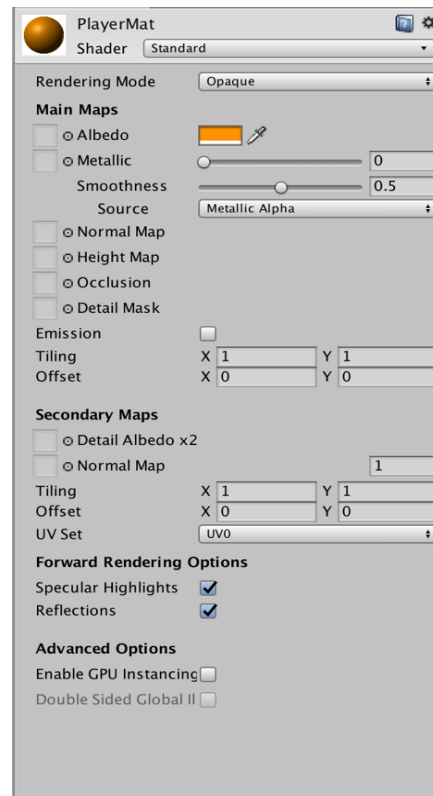
The components and code that I need to build this game are:

1. A **Ground** where the game is going to be played on. Give it a Physic material so the player cube that is going to be created will move very smooth on the ground.

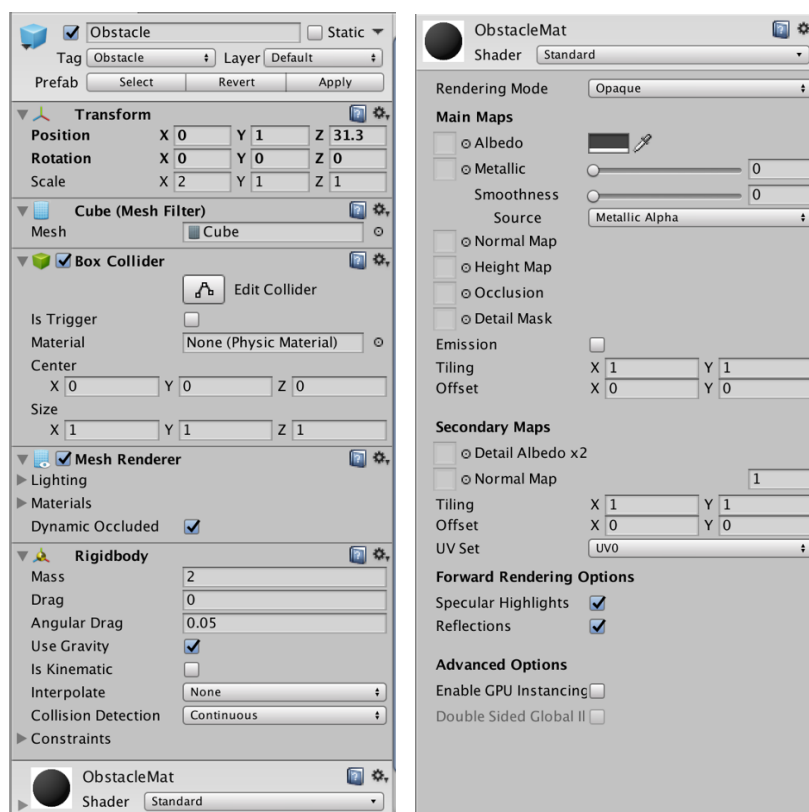


2. A player will be moving to the left or right when pressing A key and D key or Left Arrow key and Right key. A script named PlayerMovement has to be created so you insert the code to make player(cube) move left or right. The player must have a material, so you can change the colour of the shape for example red, orange blue, etc. Another script that is going to be named PlayerCollision has to be created so when the player hits the obstacles or fall down from the ground the game will stop and restart again that level. A forward force is needed to move the player forward with a constant speed and a sideways force to have control on moving the player from left or right.

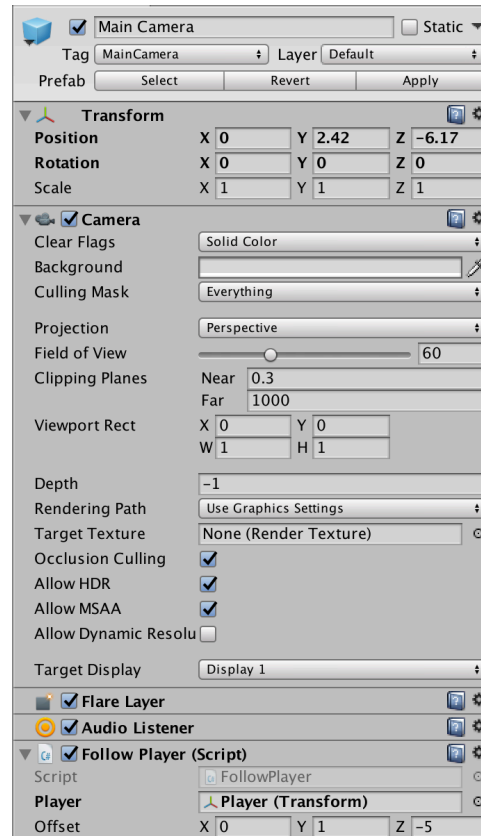




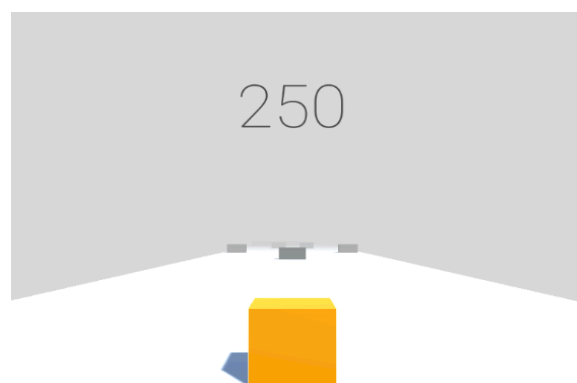
3. The Obstacles need to be created as a cube shape so when the player hits them the game will stop immediately. To make your life easier it is better to create a layer for the obstacles as you will have many of them.

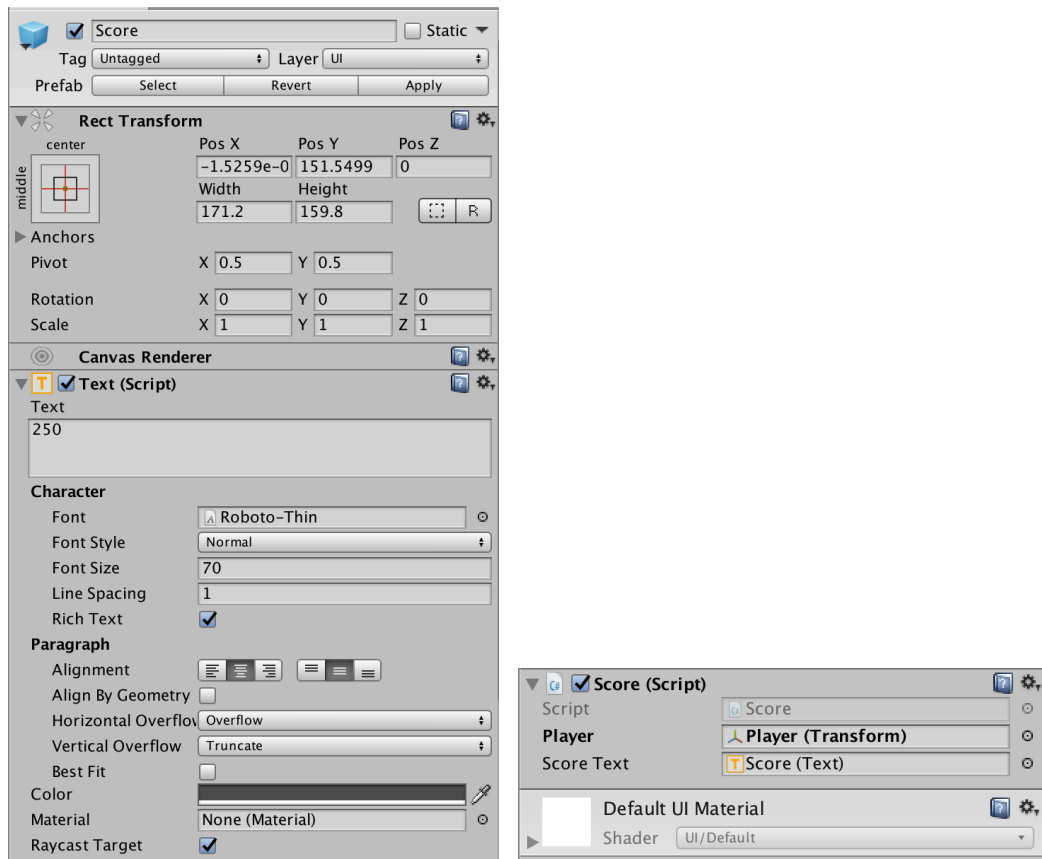


4. A camera is needed to follow the player while playing the game. The camera needs to be set following the player in each and every action moving left or right, falling down from the ground or winning the level. A script has to be created to tell the camera to follow the player.



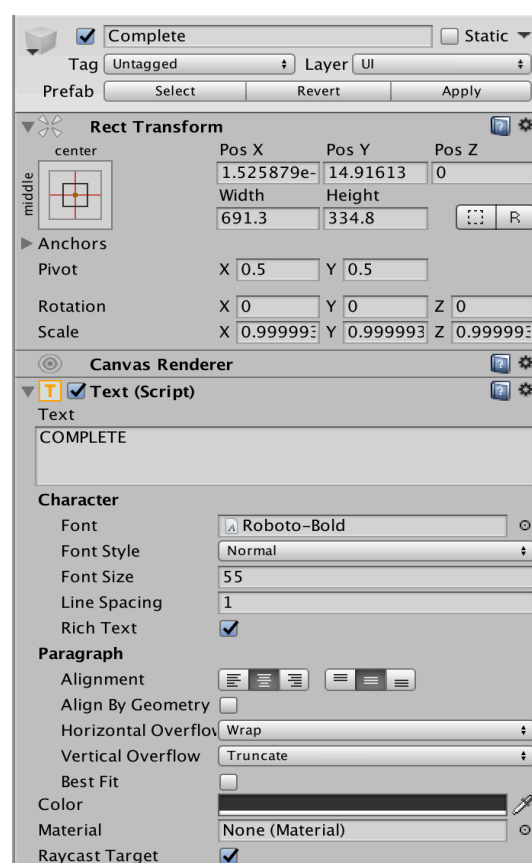
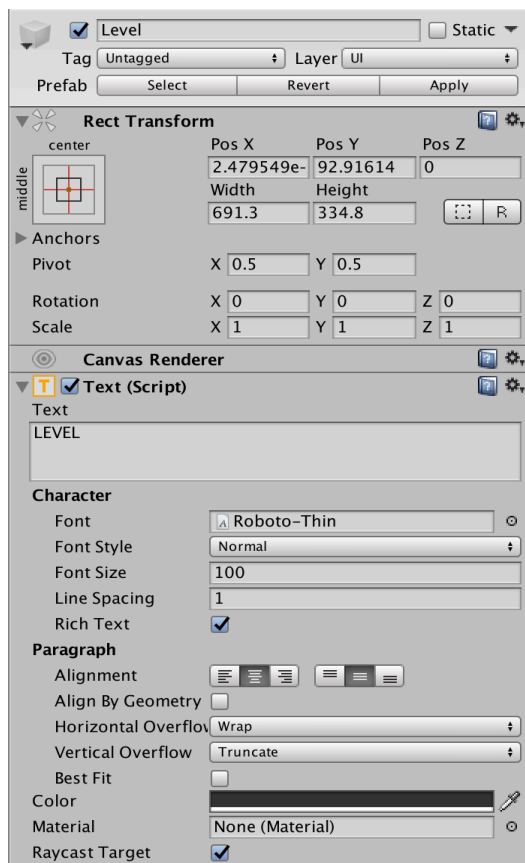
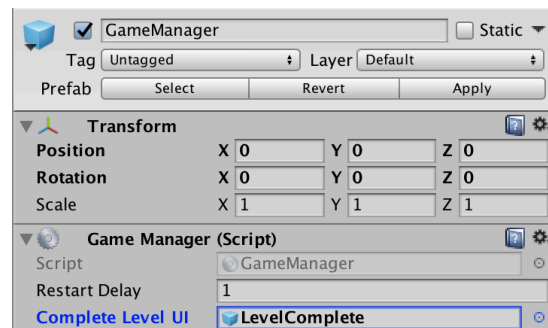
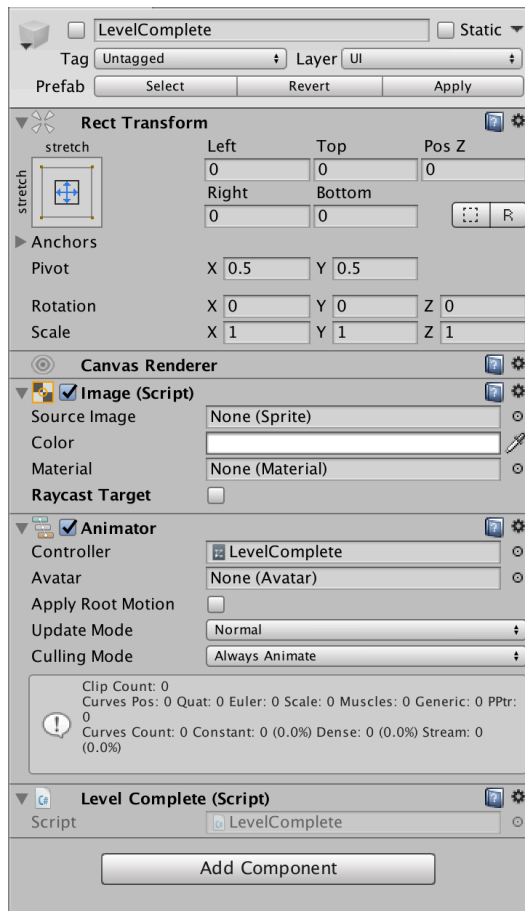
5. To create the score (Distance), you have to use UI canvas with an inside text. To fit the score appropriately in the screen it is best to select the mode to Scale with Screen Size and increase the Match to 1. A script named Score has to be created to start the score (Distance) from 0 when the game starts. As a **font** I downloaded Roboto from google and used it as my text.





In the same canvas you have to create a new UI text that is the LevelComplete and a script named LevelComplete has to be created to load the next level when your done by the previous one. An animation has to be created after so as you finish the level, a Level Complete text will flash up showing that you passed the level. Game Manager script will manage the state of the game.





6. I need UI buttons for the Start Menu, Paused Menu and at the End of the game.

a. Start Menu:

- i. **Start Button**
- ii. **Level Button** which contains Level **1, 2, 3, 4, 5, 6** and a **Back Button** that takes you back in the main menu.
- iii. **Mute Button** to mute the song if you want or to play the game with sound.
- iv. **Quit Button** to quit the game.

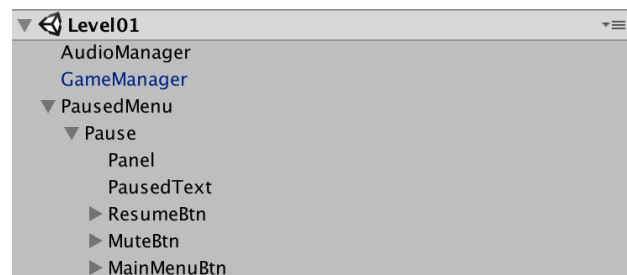
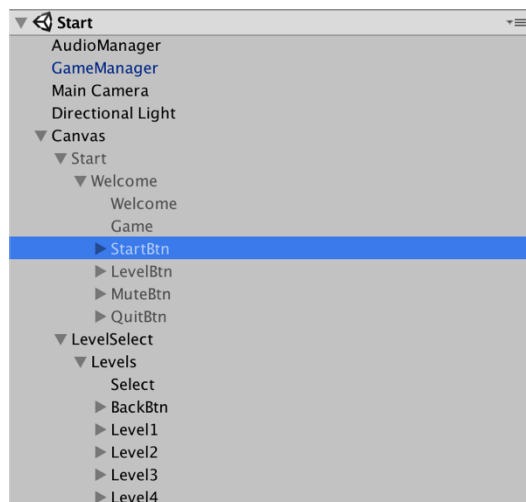
b. Paused Menu:

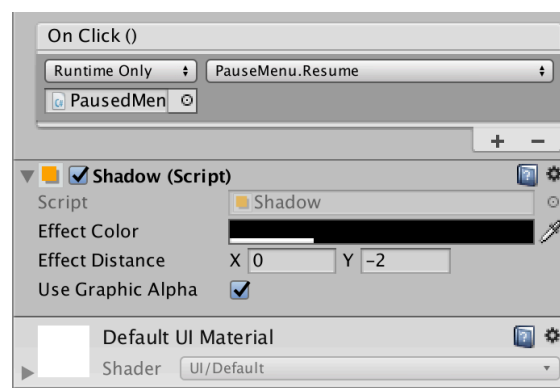
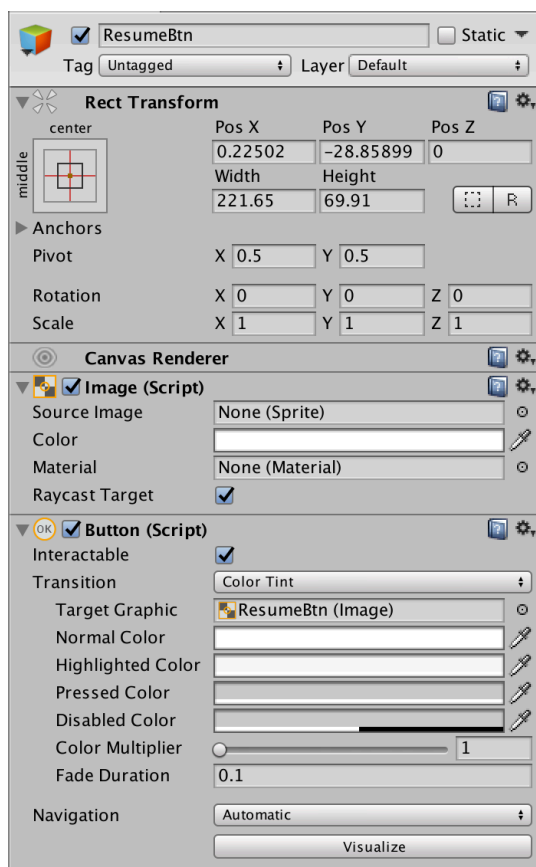
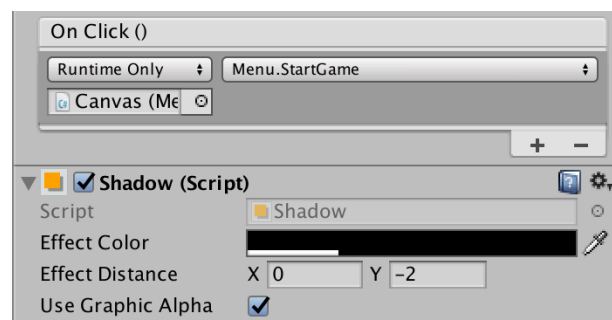
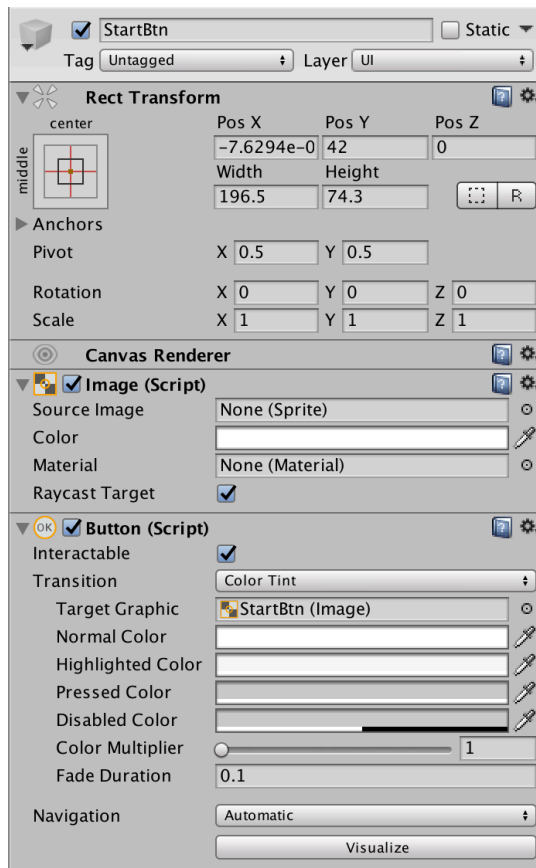
- i. Resume Button to complete the game.
- ii. **Mute Button** to mute the song if you want or to play the game with sound.
- iii. **Menu Button** which will take you to the Start Menu.

c. End:

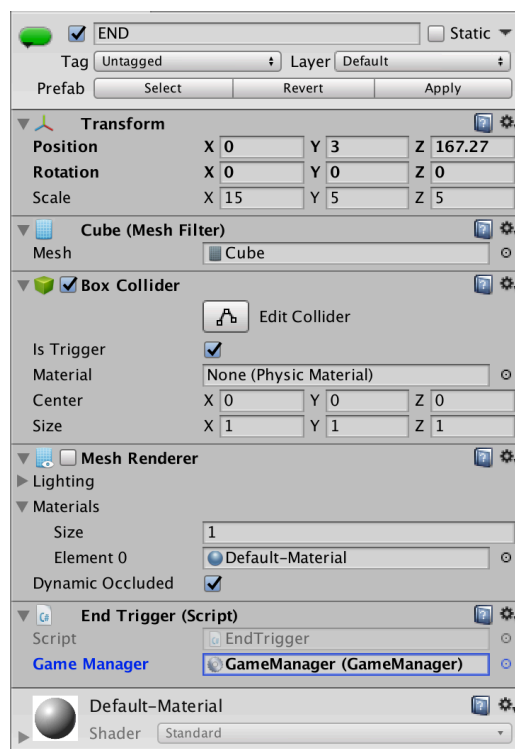
- i. **Quit Button** to quit the game.

All these buttons need to be placed in a script how they are going to work. For each menu you have to create a script listing the buttons that I mentioned above.

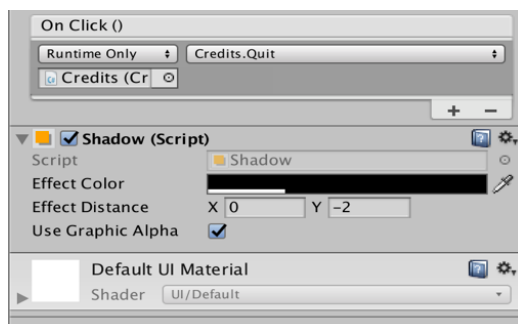
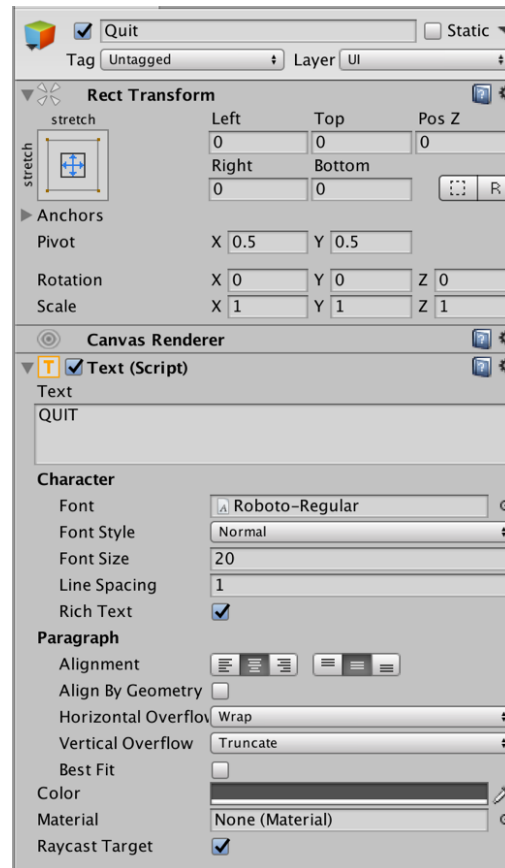
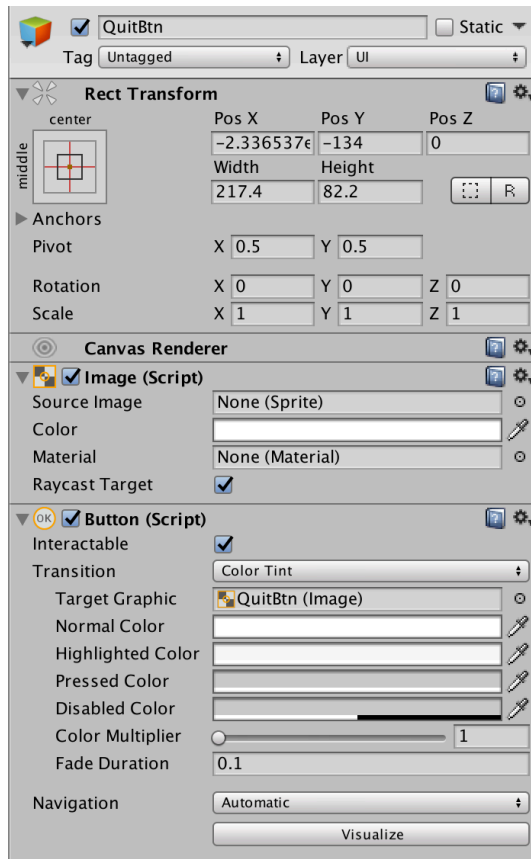




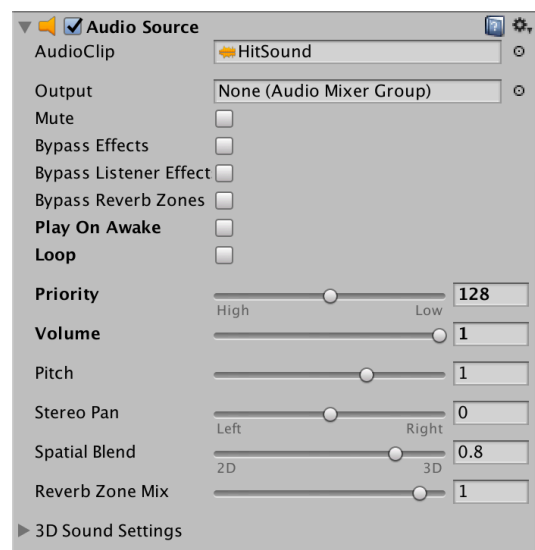
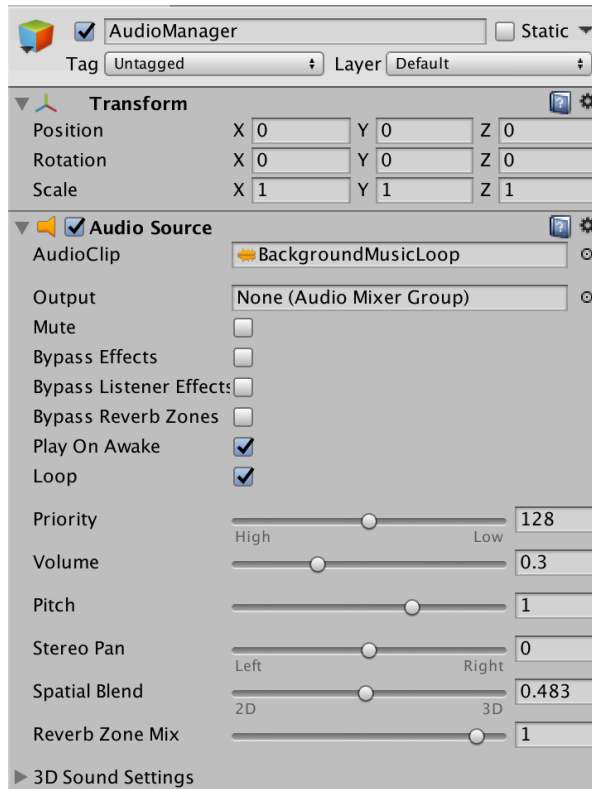
7. Game Over and Restart need to be done so when you lose the game the game will restart after 1 second.
8. A script named PlayerPrefs needs to be created to delete the player prefs.
9. End trigger has to be used so when the player passes the level it will collide with an invisible cube that has to be created from side to side of the ground, so you can pass to the next level. Then a canvas need to be created to input an image at the end of the level. Then a script named EndTrigger needs to be created so when the object hits the EndTrigger it will take you to the next level.



- To create the credits at the end of the game when you pass all the levels a screen will pop up saying “Thank You For Playing!” and by who the game was made. A script named Credits need to be created to tell the application to quit. A Quit Button needs to be created to quit the game when you finish it all.



11. The last thing that I have to make is the Sound. There is going to be an introduction audio at the beginning of the game in the start menu, a background audio while playing the game and an audio when the player hits the obstacles.



References:

Powell-Morse, A. (2018). Agile Model: What Is It And How Do You Use It?. [online] Airbrake Blog. Available at: <https://airbrake.io/blog/sdlc/agile-model> [Accessed 12 Apr. 2018].

Tutorialspoint.com. (2018). [online] Available at: https://www.tutorialspoint.com/sdlc/pdf/sdlc_agile_model.pdf [Accessed 12 Apr. 2018].

www.tutorialspoint.com. (2018). SDLC Agile Model. [online] Available at: https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm [Accessed 12 Apr. 2018].

YouTube. (2018). How to make a Video Game - Getting Started. [online] Available at: <https://www.youtube.com/watch?v=j48LtUkZRjU> [Accessed 12 Apr. 2018].