**COMP3064 – Game Programming Assignment 1**

Student: Darren Lung

ID: 100937566

Date: Thursday, October 19, 2017

**Table Of Contents**

Detailed Game Description -----------------------------------------------------3

Controls Description -------------------------------------------------------------4

Interface Sketches----------------------------------------------------------------4-5

Screen Descriptions--------------------------------------------------------------5-6

Enemies-----------------------------------------------------------------------------6

Scoring------------------------------------------------------------------------------7

Sound Index-----------------------------------------------------------------------7

Art/ Multimedia Index----------------------------------------------------------8

**Detailed Game Description**

Robo Shooter works by having multiple scenes. The start scene works by having a empty game canvas that previews the player with how the game is ran. When the play button is pressed, the user is then sent to the main scene where the game runs.

When the game is started, the player is set on the scene and enemies start spawning from the right-hand side of the screen and moves to the left simulating flight. This is done with setting the rigid body of the enemy to have a -X speed.

When an enemy or enemies bullet collides with an object with the player tag it will call the damage function in the player's class and subtract their health, which starts from 3. In addition, the player will flash red and play a hit noise that is implemented in the class. The player will explode and shoot out red particle effects when the user reaches 0 health.

The player gets points by destroying enemies with their bullet. When the bullet collides with an enemy, the enemy calls a similar function to the player where they are damaged. Every bullet subtracts 1 life from the enemy and when the enemy dies, an orange particle effect is shown on the screen and the player is rewarded with points. The amount of points set is up to the person controlling the game to set. In addition with enemies, the user can pick up red health potions which give them 1 extra life as well as gold point packs which will give them an extra 200 points.

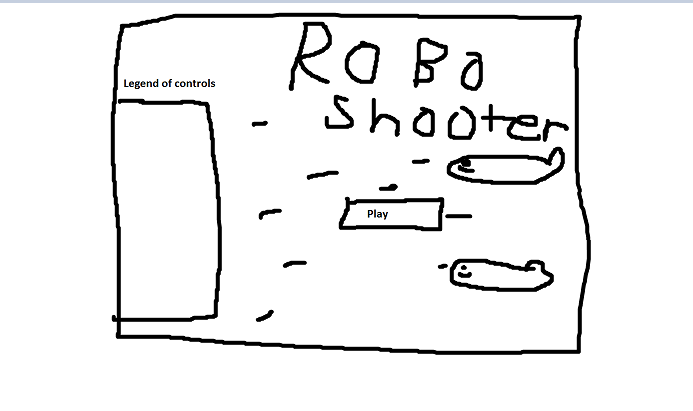
When the player dies, a game over screen will show after a brief delay. The game over screen displays their score and high score. These scores are stored within the player prefs. The end screen will compare the current score to the high score, by default it is zero. When a user gets a score higher than the high score, it will save the current score to the highscore playerpref and will save. In the game over screen, there are three options. One to go to the main screen, which switches the scene to zero. One to replay, which resets the score player pref to zero and send them to scene one. Lastly, one to clear the game. This button function makes the user go to scene zero and resets the score and highscore player pref.

**Controls Description**

Controls are fairly simple, the WASD and ^<v> arrow keys are used to control the rigid body of the player. When a user presses the A and D keys I use a new add force on the rigid body and add a new Vector 2 which uses Input.GetAxis which determines whether or not the player is moving left or right. After the input is gotten, I apply a speed of 35 to the player as well as setting the Y coordinate to zero.

The up and down and W and S keys work similar to the left and right controls. The only difference is that I set the Input.GetAxis to vertical and i set the X coordinate to 0 and implement the input.getaxis \* speed on the Y coordinate instead.

**Interface Sketch**

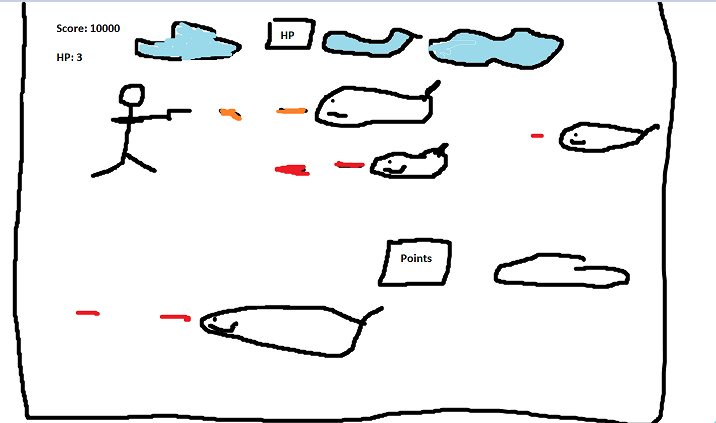
****

**-Start scene**

-Preview of game shown

-Play button is runs the second scene

-Legend of controls on the left

****

**-Play scene**

-User is able to shoot

-Hp and points packs available

-Life and Score is shown

**-Game Over**

-Score and high score is shown

-3 buttons user can press

-main menu, replay, cleargame

**Screen Descriptions**

****

-**Start State**

-Legend on the left showing

what pickups are and keys

-Play button is in the middle

-preview of the game is

available

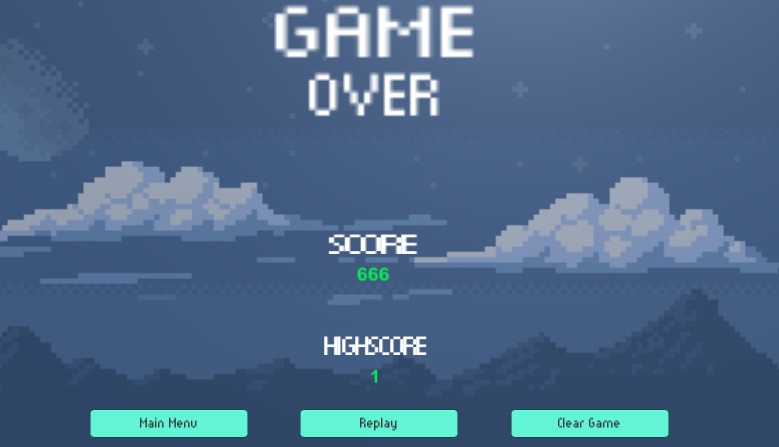
****

**Game State**

-Enemies shoot, HP/points packs available

-User can fire bullets

-Score starts from 0 and lives are default to 3.

**End Game State**

-Score and high score is shown

-3 buttons, Main menu, replay and clear game is shown.

**Enemies**

Enemies in Robo shooter work by having the Game dev set a health them. In addition, the game dev can choose whether or not the enemy fires bullets by selecting the checkbox in the enemy script properties in the unity editor. This is implemented to the enemy by having a canFire if statement to check if the boolean equals true. If it is true, set a fire rate and loop it to have a enemy shoot a bullet.

Enemies spawn in the spawner script. This spawner script has a rate of spawning that repeats as soon as the game is started. In addition, the spawner script has an enemies array. This array is used to hold many different enemies so that the game developer can add as many enemies as they wish to in the future.

The spawn method itself grabs all of the enemies within the array and spawns them within -4.29 to 4.29 using a random range. This makes it so that enemies can spawn anywhere within the camera range, which is -4.29 to 4.29.

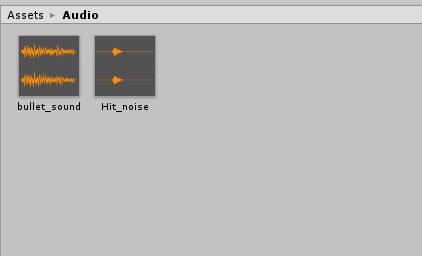
The game developer can also set a score for each individual enemy. When an enemy dies it will grab this score number and add it to the player prefs "Score", which is accurately displayed at the end of the game and during the game.

**Scoring**

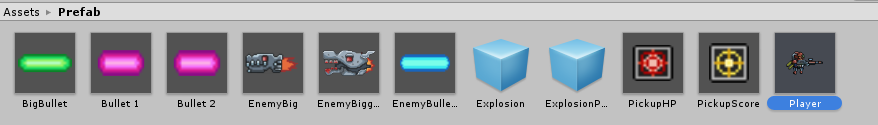
Players can earn points by either grabbing the points pack (200points) or shooting down an enemy. The points are calculated in the Enemy script of my game. Specifically in the damage function, when an enemy reaches a health of 0, the enemy will explode, and set and int to the player prefs, which is saved under the "Score".

The points pack works very similar to the enemy, but the only difference is whenever it collides with the player it will add the points to the player, which is a stock value of 200. The method for adding points is within the playerS script under addpoints(), and method for calling this addpoints() is within the pickup points script under onCollisionEnter2D.

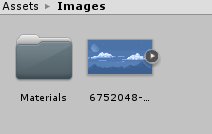
**Sound Index**

I only have 2 audio sources, one for when the player gets hit and one for when a player shoots a bullet.****

**Art/Multimedia Index**



There are currently two enemies within my game and a player. The game also has two pickup items as shown and 4 different bullets to choose from.



**ASSET > IMAGES**

-In the images folder, my background is saved there