Blaine Burke – G00354397

Mobile Applications project

My game

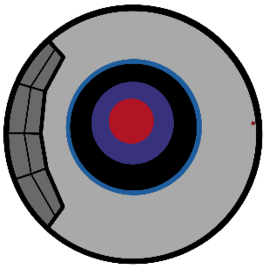
My game is a 2d top down shooter in which you fight hordes of enemies. There are three levels that are all different difficulties and in order to win you must collect 1000 coins and purchase the teleporter to win. To collect coins the user must kill enemies and to help with this process the user can purchase damage and bullet speed upgrades that can be found in the levels.

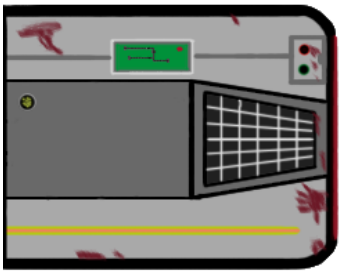
The enemies also do various amounts of damage, the small enemies doing 10, the normal size doing 15 and the big ones doing 20. In order to get health back when lost is to retrieve health drops from enemies when killed. You also have a small chance of getting a rapid-fire drop which uses a coroutine function in order to allow the player to hold left mouse button and provide many bullets for 5 seconds.

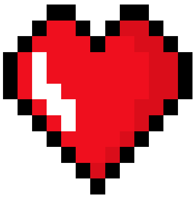
When in game, the user presses the ‘Escape’ key to pause the game. This allows the user to access the volume settings and to quit to the main menu.

If the user wins a game a Victory screen appears showing the players number of kills and allows them to leave to the main menu.

If the user dies in game a game over screen appears showing the players number of kills and allows them to leave to the main menu.







Mobile Controls

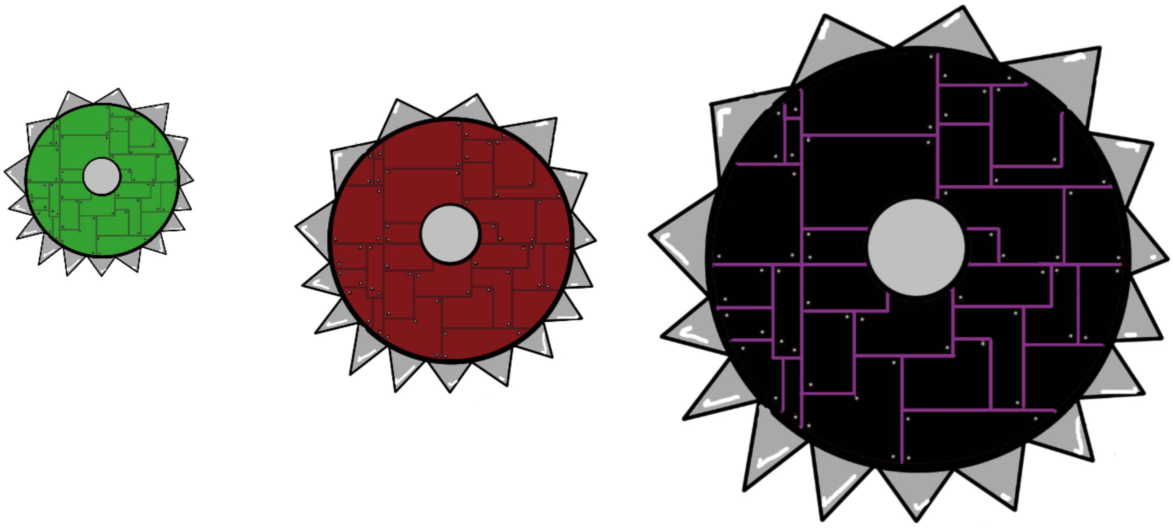
When trying to implement mobile controls I attempted using both my lecturer, Damien Costello's tutorial and a YouTube tutorial by Brackys. I could not get either to function correctly, so I left it absent from my project.

Varying Difficulty

In my game the difficulty increases throughout my levels. Level 1 features only the regular enemies and a generally open map compared the following two.

In level 2 the map is much tighter making it harder for the player to keep distance from enemies. It also introduces a new enemy which are smaller and faster making them hard to hit. To balance this, they are a one shot kill and only do 10 damage compared to the 15 of the regular enemies.

In level 3 space is pushed to its limit as larger enemies are introduced. They are slower, have high health and deal 20 damage. They are notorious for blocking paths due to their sheer size but also Lava tiles are introduced dealing 5 damage on touch making it easy for the players to accidently bump into one and take damage.

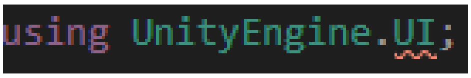


My drawings

For my project I drew a lot of my sprites such as my character , enemies and my menu background

Problems I encountered

During the coding process I encountered several issues. My HUD or heads up display would not display the health, bullet speed and damage. Over time I figured it out. I also kept getting an error for ‘using UnityEngine.UI;’. I never found a fix to this as it merely said it was an error but when ran it ran with no problems.



```
using UnityEngine.UI;
```

What have I learned?

During the time it took me to develop my game I became more efficient in c#. It also enhanced my problem-solving skills due to fixing errors and finding ways to work around limitations of my project. I also am more confident and efficient using Unity.

Bibliography

WaveSpawner.cs → From Brackys(https://www.youtube.com/user/Brackeys)

WaveSpawner.cs controls the waves of enemies in my game. It ensures a steady flow of enemies that I set myself to keep the game progressing. I adapted the code I received from Brackys and incorporated it into my game.

My Coroutine → Damien Costello

In my Shooting.cs script, I have a FireCoroutine function. This is used for when the player gets a rapid-fire power up in the game allowing them to hold left mouse button to rapidly shoot bullets. I took Damien's script from one of his tutorials and incorporated it into my code.

My Music from Level 3→ Doom

My song used in level 3 is sourced from the popular video game Doom. It is titled 'At Doom's Gate'. I was inspired by doom when making the game and I felt it fit to include a song from it which also turned out to fit the level well.

Pathfinding → arongranberg.com/Astar(https://arongranberg.com/astar/)

My pathfinding algorithm was sourced form Astar. It functions by making the enemies calculate the shortest distance to the player taking any obstacles into account. This removes the issue of enemies flying directly at the player and into walls and getting stuck.

Tiles and Decals → Tiny RPG forest

For my second level I used Tiny RPG Forests assets to create my tile maps.