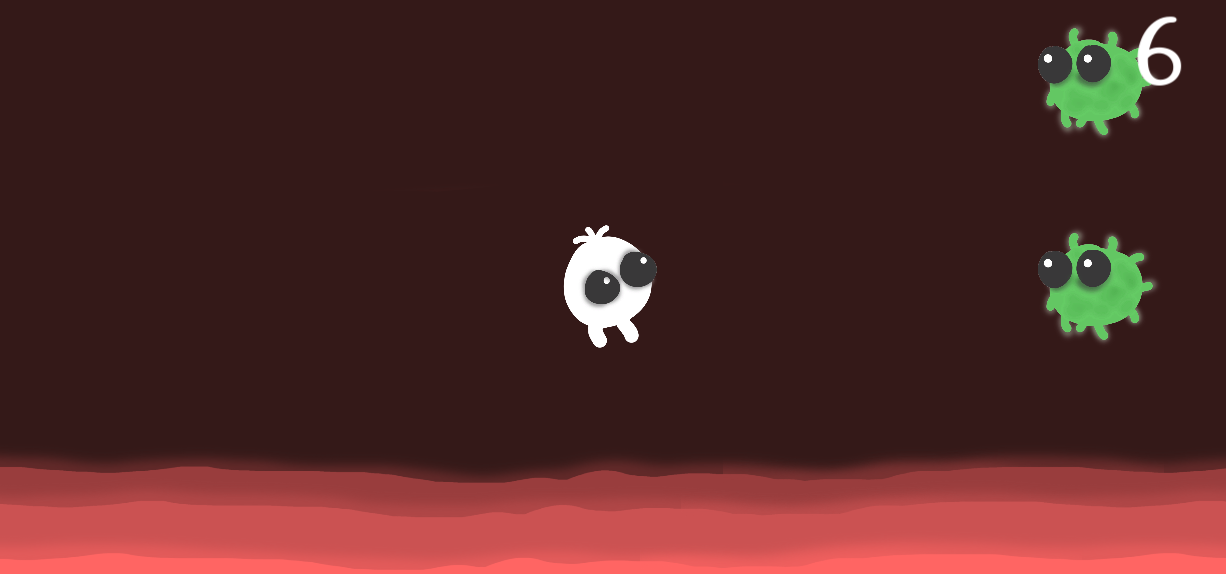
# Gesture Based Documentation

### Ethan Horrigan & Dylan Loftus

### <https://github.com/ethanhorrigan/Gesture-Based-Project>



Contents

[Cover 1](#_Toc27073534)

[Contents 2](#_Toc27073536)

[Background Movement 2](#_Toc27073537)

[Player 4](#_Toc27073538)

[Enemy 5](#_Toc27073539)

[Score Handler 6](#_Toc27073540)

[Screen Effects 9](#_Toc27073541)

[Particle Effects 10](#_Toc27073542)

[Myo 11](#_Toc27073543)

[Speech Recognition 12](#_Toc27073544)

UI [12](#_Toc27073545)

[12](#_Toc27073546)

[12](#_Toc27073547)

[13](#_Toc27073548)

[13](#_Toc27073549)

[References 14](#_Toc27073550)

### 

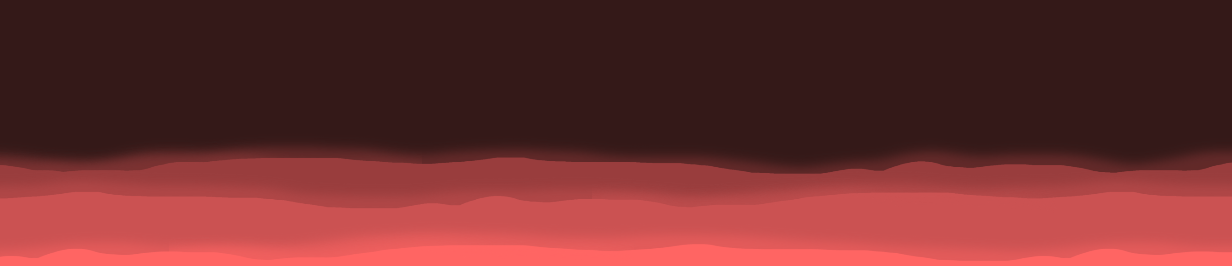
### Background Movement

### Ethan

Since the genre of this game is an Endless Side Scroller, the background moves instead of the player moving, giving the illusion that the player is moving.

The background sprites have a Starting X position and an End X position, the background is then translated on the X-axis. When the End of the sprite reaches the End X position, it is reset back to its Starting X position. To ensure this worked correctly, the starting X and ending X must meet at the same point or else the movement would not be smooth.

I created 3 “Wave” sprites using this movement at different opacity and speed to give the game a sense of depth.



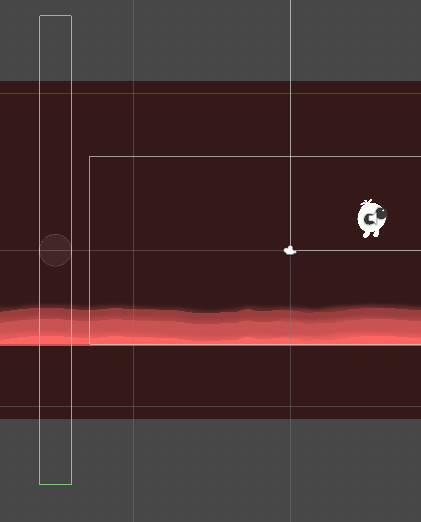
Adapted from: <https://answers.unity.com/questions/19848/making-textures-scroll-animate-textures.html>

### Score Handler

### Ethan and Dylan

Originally, Dylan had implemented the score handler which incremented based on how long the player was alive, we both agreed that refactoring it to an enemies-defeated based score handler would be better suited.

This was achieved by adding an invisible object off-camera with a collider, each time an enemy collided with this object, the score would increment.



### Artwork

### Ethan

The artwork was made using Adobe Photoshop with a Drawing Tablet.

To create Sprites, Tile sets & Animations.

### Sound Design

### Ethan and Dylan

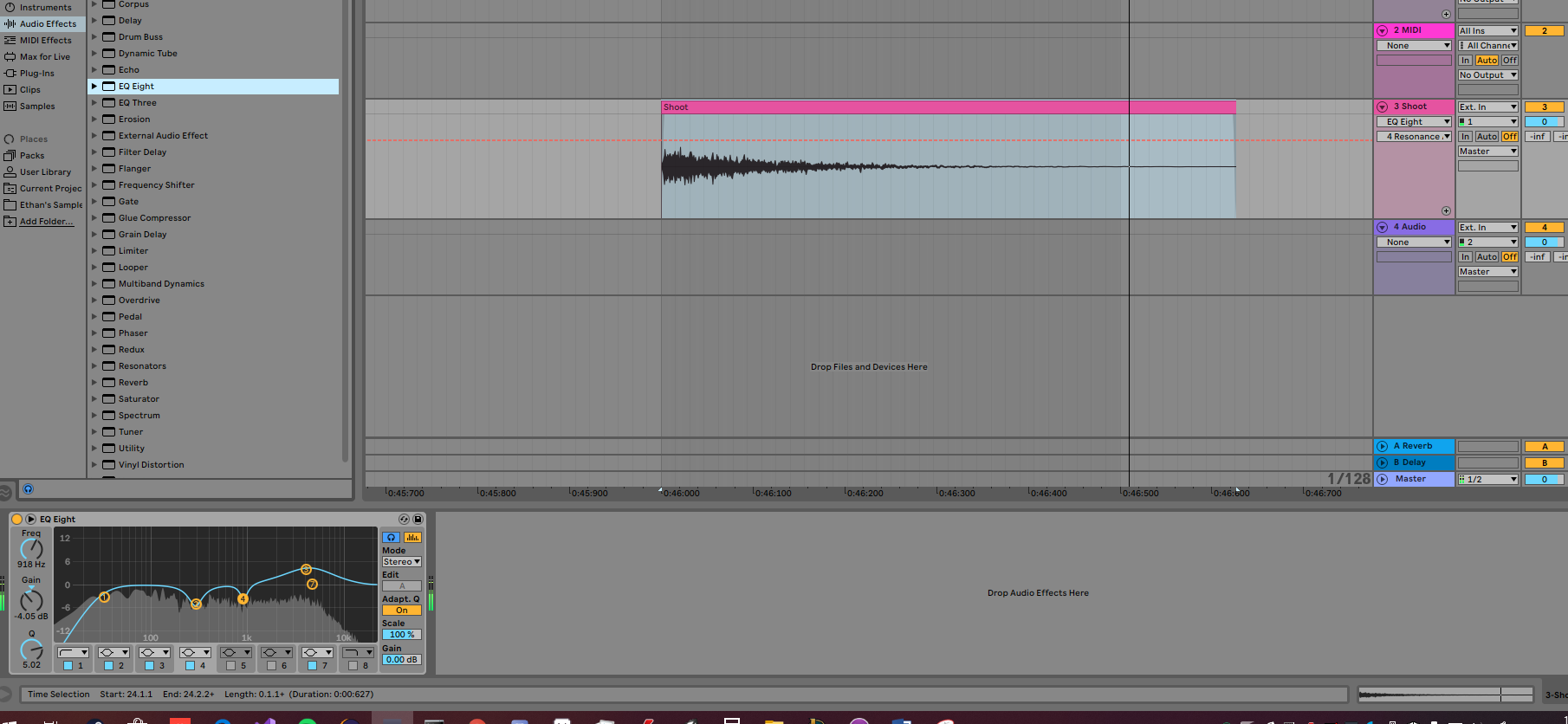
I used <https://freesound.org/> to obtain sound effects and then I imported them into Ableton for further manipulation and processing.

I did this because some of the sound’s quality were not great and I could also process them in any way with the use of equalizers, compression, reverb etc.

I compressed each sound effect with similar settings so the range of volume between sound effects were relatively similar.

This allowed for coherency between sound effects within the game.

Program: Ableton



### References

[SoundFX] <https://freesound.org/>

[Font] <https://www.fontsmarket.com/font-download/burbank-big-condensed-bold>

[Music Playing Throughout] <https://answers.unity.com/questions/1260393/make-music-continue-playing-through-scenes.html>