Unreal Engine 5 First Person Shooter

Kevin Huang

Introduction

"*It's Not the Result That's Important, It's Who You Become in The Process of Achieving the Result*", Jack Canfield.

This project was created as an attempt to translate our knowledge from Unity to Unreal Engine utilizing the knowledge we have learned throughout the semesters. I am working with a fellow student from the New York City College of Technology. The goal of this project is creating a demo of a first-person shooter game using the engine. The project idea was originally inspired by single player shooter games. We will be creating a game to learn about Unreal Engine. To learn more about Unreal Engine we will learn and try out new tools that Unreal Engine provides. The process of experimenting will be a heavy emphasis in our project because we will be creating a game without any knowledge of this system.

Description

Our project will be a bare bone Unreal Engine game map which takes assets from the First-Person Template provided by Unreal Engine. We will be pulling and utilizing online resources and attempt to minimize the cost of the game production as much as possible(Preferably Keeping the game Free besides already owned assets). We will be learning Unreal Engine and understand the process and methods of making a shooter game.

Methods

We will be utilizing online tutorials to improve upon our knowledge and learn about blueprints and utilizing the blueprints and actors. We will learn more about the in-built in Unreal Engine properties and events to create our game.

Process

I looked through starter tutorials of Unreal Engine as a start to gather information on the things we will need to learn. Unity relies more on creating C# scripts and prefabs while I have found that Unreal Engine relies more on the blueprints and already made actors. I begin to delve more specifically into subjects that I want to add into my game from the player actor to enemy and environment. I look for tutorials that have some similarities to the blueprints I want to create which I will attempt to modify them based on my understanding of what was shown in the tutorials. Most of the blueprints shown in the tutorials does not work as the code I am looking for often deviates from what I wanted from the code. I experiment with the online resources until I find a method that works with my code and work upon that structure.

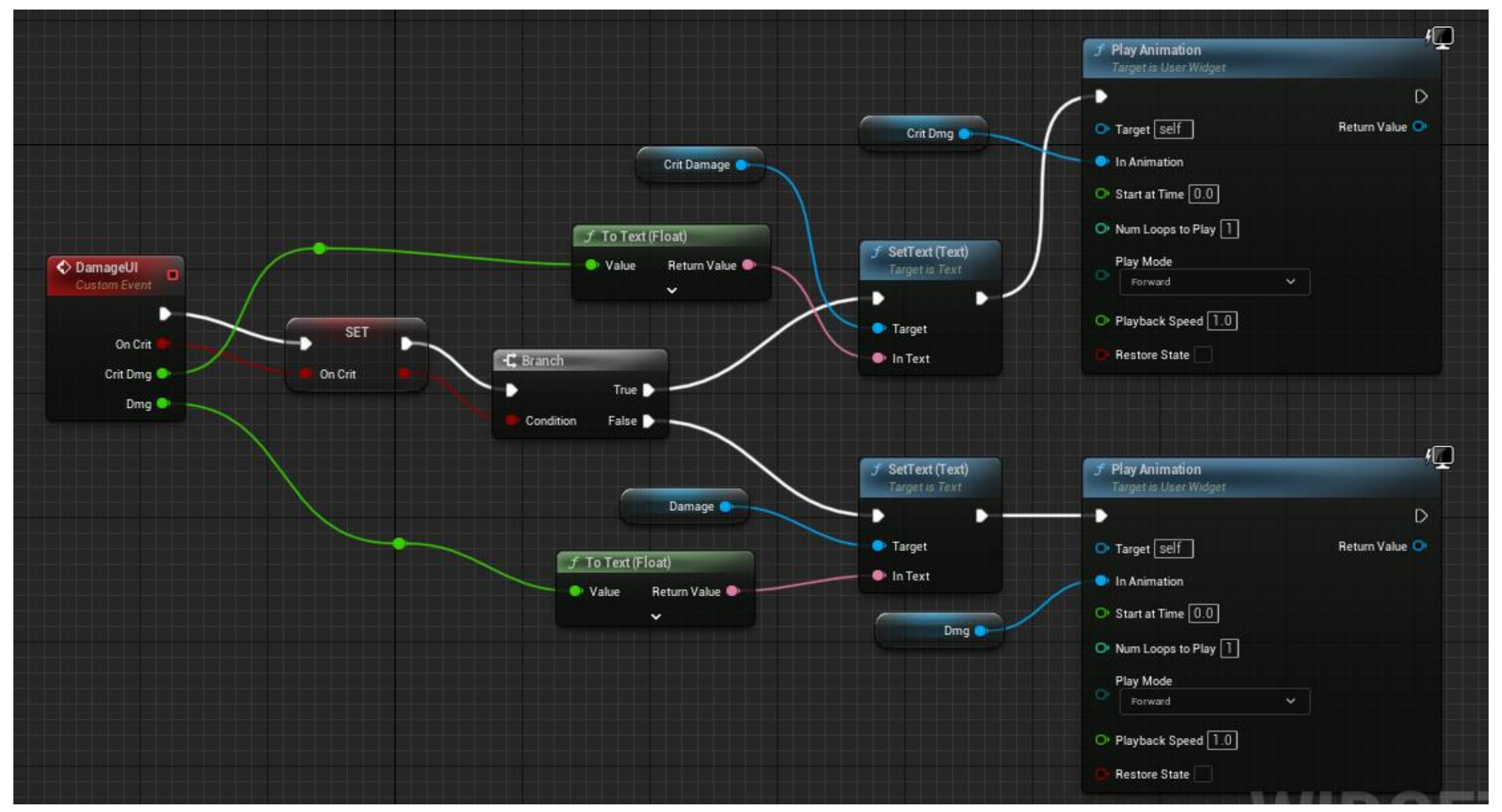
Budget

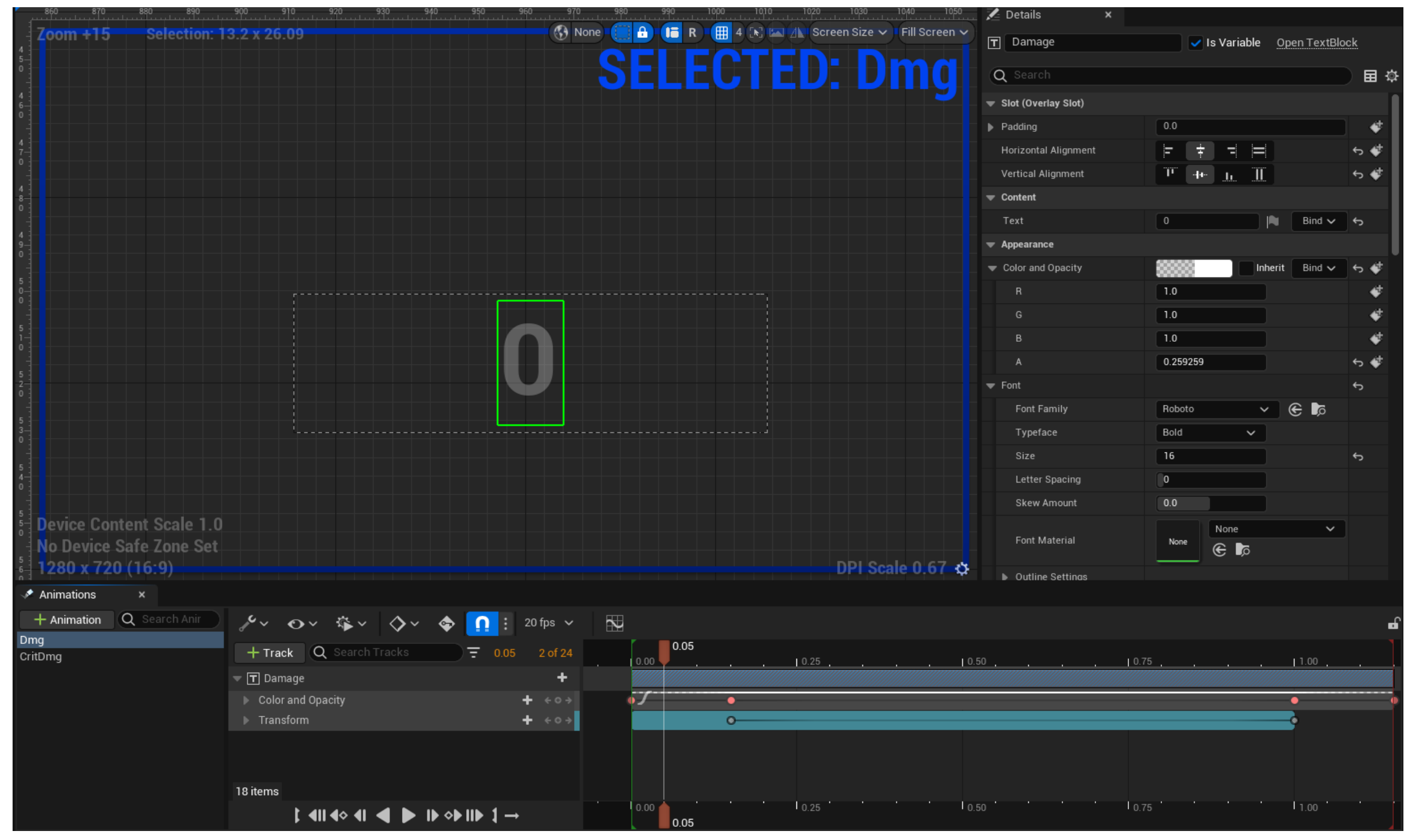
Item	Price
LG 27GL83A-B 27 Inch Ultragear QHD Gaming Monitor(Owned)	$358.34
Redragon S101 Wired Gaming Keyboard and Mouse Combo(Owned)	$31.95
ASUS ROG Gaming PC i7 11700KF 64GB RAM 512GB PC Win 11 Pro(Owned)	$1259.00
Poster	$39.99
Git LFS storage	$5.00
Total	1694.28

Technical Issues

The technical issues include animations, enemy rotation, navigation, and projectiles.

Animations has been a constant presence in our project because we needed to fall back on already made animations. We have messed for hours in attempting change animations and the player character which we have settled on the original skeletal mesh with only arms. The enemy did not rotate correctly towards to player to fire the projectiles and we had to set up a system which rotates the enemy to point their weapon towards the player.





Widget
The Widget is a blueprint used for creating UI like health bars and be used as an event trigger such as taking damage from enemies or displaying health and ammo on the screen.

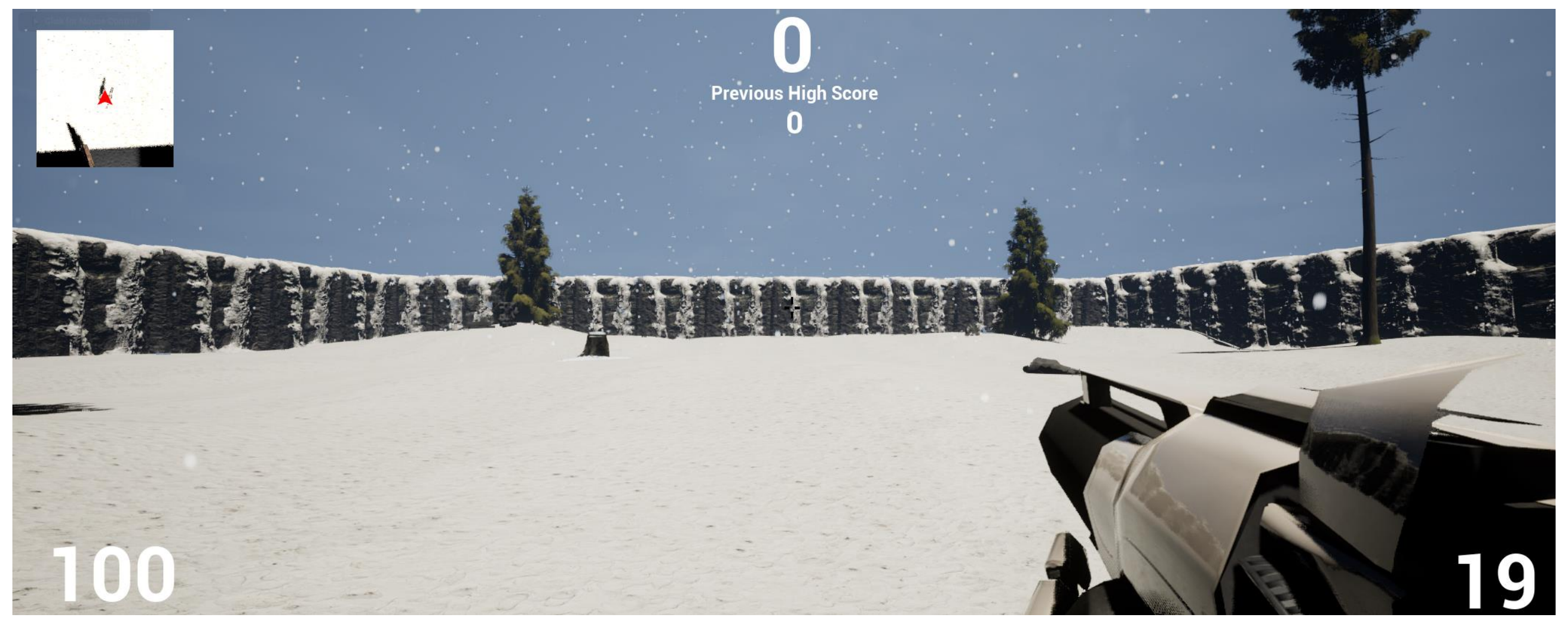
Calendar

- Utilizing online resources to understand Unreal Engine | 8/28-11/20
- Meet with Suraj | Every Friday 8/28-10/30
- Meeting with Hosni | 10/9
- Unreal Engine tests 8/28-9/20
- Enemy 9/10-9/30
- Environment 9/20-9/30
- Player 10/1-10/15
- Bullets 10/15-10/30
- UI 11/1-11/14
- Menu 11/10-11/26

UI

The UI was a system that required knowledge on how to utilize code from different blueprints. I had to research how to create code that synchronizes blueprints and widgets together and how to utilize code from other blueprints. This created a series of issues because blueprints needs to know the objects that the other blueprint is on and at times creates issues.

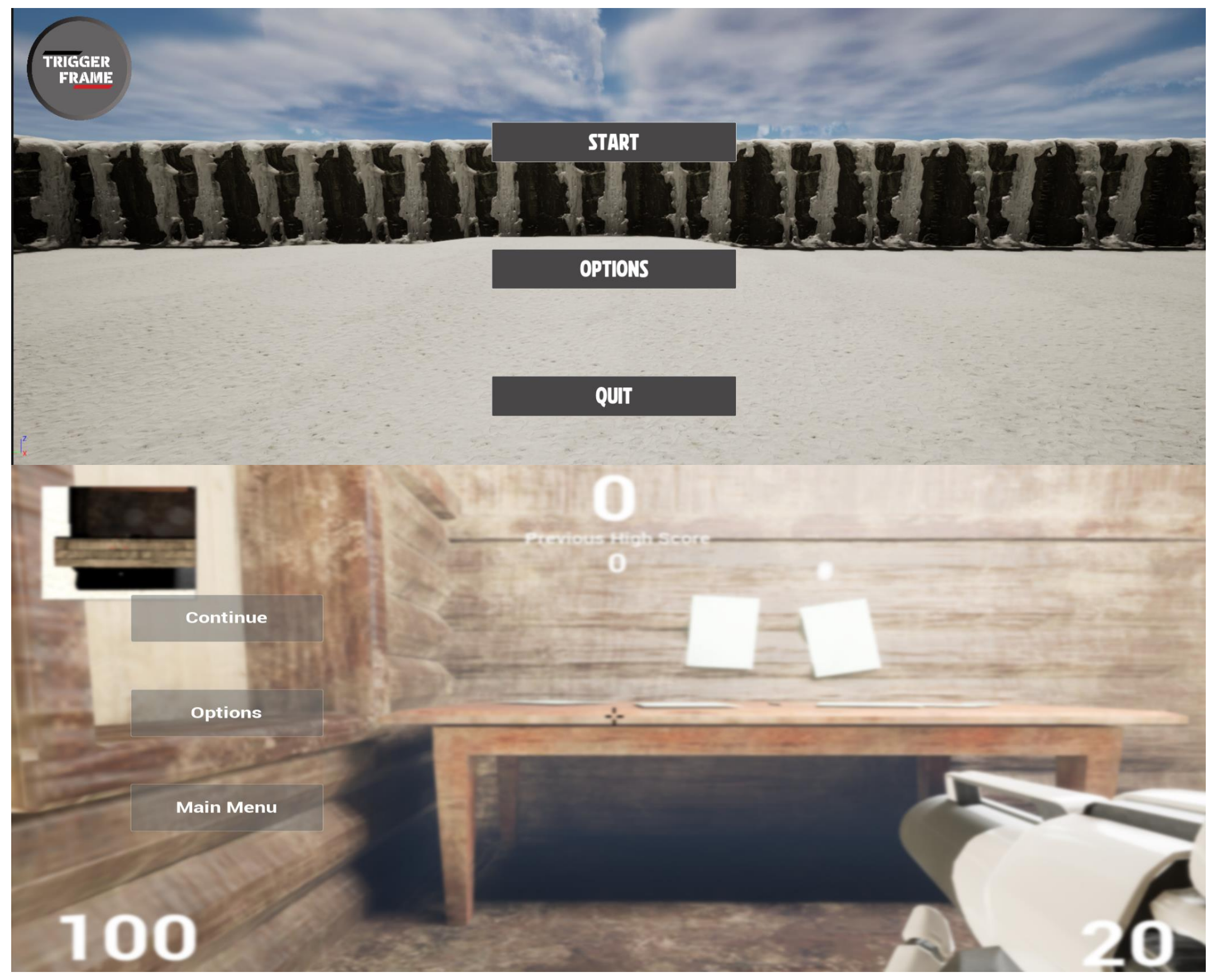
The variables are created in the UI widget for the score system, health and ammo count. They are called upon by the enemy, player, and the weapon component blueprint. The ammo system utilizes code from the weapon component. The Score utilizes and mixture of the enemy, player, and a score widget. The health/stat system uses the player blueprint.



Player UI
The player UI contains the score, health, ammo, and radar. The radar was created by Suraj. The score system is stored inside the player and changes whenever the enemy's health is zero. There is ammo and reload for the weapon however, it currently contains no reload animations.

Menus

Menus utilized the binding system within the widget. We change the code within the widget and bind events to buttons. The main menu is a separate scene that either brings you into the play level or removes itself and bringing forward the options menu. There are two options menu, one for the main menu and one for in-game to create a background blur. Unreal engine has an in-built save and load system for options. The pause menu and the death menu uses the pause event from Unreal Engine. All of the menu systems alternate between the UI input mode to allow the user to interact with the menus. The only exceptions are the death menu and the pause menu that utilize both the UI and game input mode because the widgets are triggered within the game level while the player is playing.



Menu
The menu utilizes the widget blueprint. It contains bindings for buttons and the code is done within the event graph of the menu.

Results

The contributions I have made upon our project include enemies, bullets, player UI, menus, bullets, enemy and player attacks. and navigation. I have utilized the animations within Unreal Engine as a free animation asset and learned how to align items along with animations. I have used Navmesh to create the enemy navigation of this game. Enemy spawning has been added by spawning them through spawner actors created in the map. The demo map was created through a series of assets such as walls and ramps. I created the enemies, enemy widget animations for taking damage, and added their animations. The menu system, UI widgets, pausing, and options taught me how to utilize widgets and how to connect code with widgets.

Summary

The goal of learning Unreal Engine has been productive as we have learned how to utilize the blueprints and animation graphs within Unreal Engine. We learned about the foliage and fracture system, but we did not include fractures due to a more condensed map. We understand how to use input systems and event graphs within the blueprints. Overall, the experience that I have received from this project gave me the knowledge on how to code in Unreal Engine.

References

Tutorial Channel References

Gorka Games

Unreal University

Buvesa Game Development

Special Thanks to

Professor Hosni Auji