

# Ue5 First Person Shooter

Kevin Huang

# Contribution

- Enemy: Navigation, Patrol, Animation, Blueprint, Spawner, Stats, and Behavior
- UI: Player, Main Menu, Death Menu, Options Menu, Enemy damage, Menu Navigation, and Pause Screen
- Bullets: Player, Enemy bullets, and Damage type
- Map: DemoMap
- Player: Firing, Score, Stats

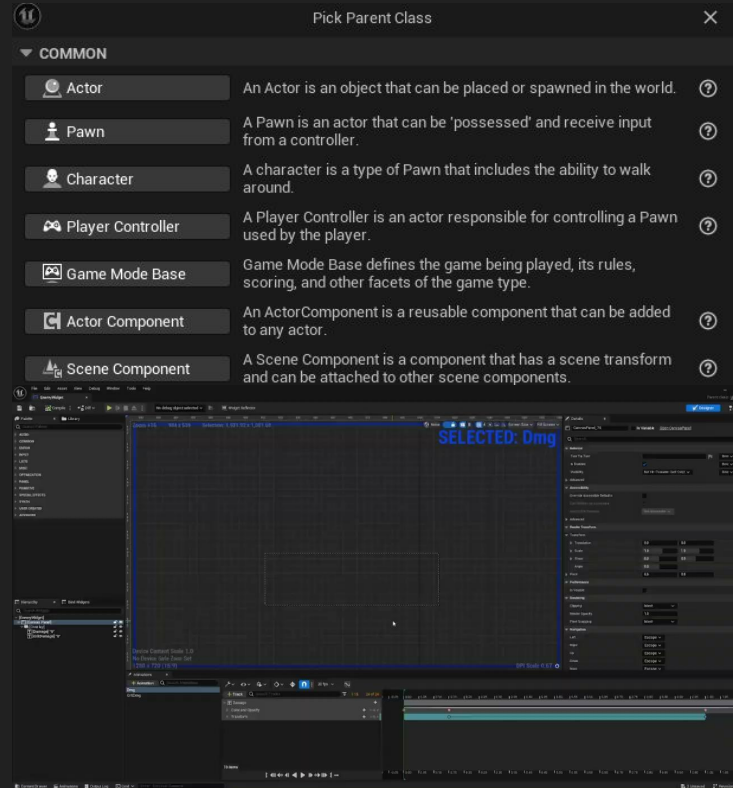
# Learning Unreal Engine

Learning Unreal Engine is tough on your own and we started out researching and looking into tutorials. We had zero information coming into this engine and being able to visualize and listen to tutorials assisted us on our journey to seek out more specific questions we had for Unreal Engine.



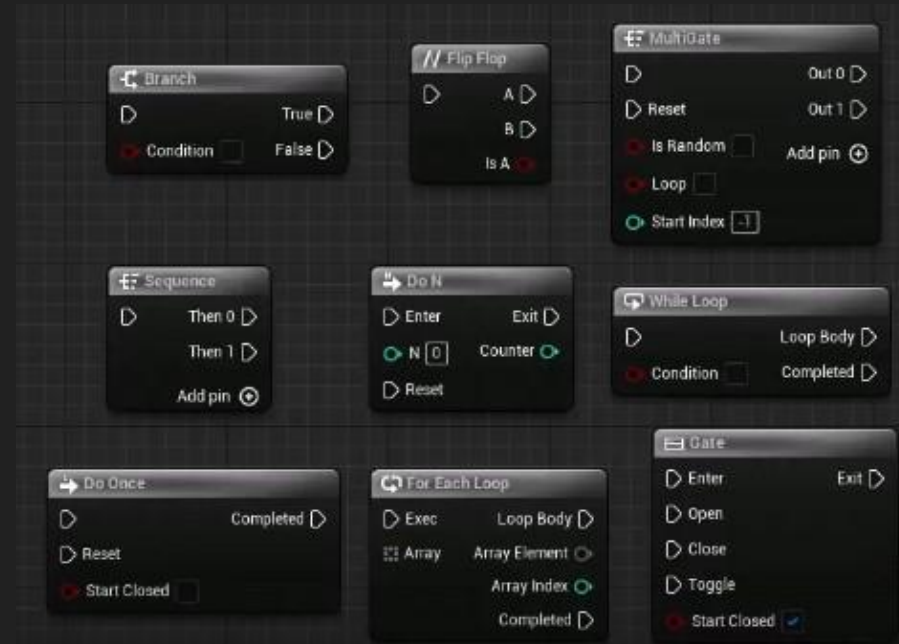
# Blueprints

Blueprints is the most important component of Unreal. Because this is how you code. I started out with enemies and enemy blueprints because we needed a target to experiment with. We had to learn the various kinds of blueprints and how to modify them to assist us in making our game.



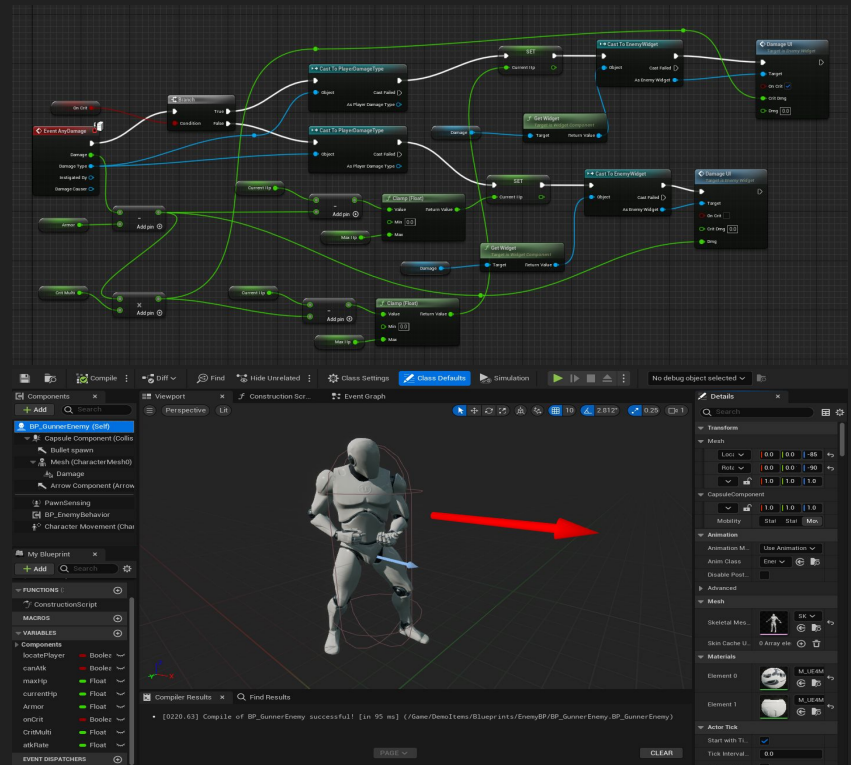
# Blueprints

The struggle about learning Unreal Engine is learning how to code in Unreal. Most of the code I learned from Unity is just through C# scripting which triggers the code in sequence from top to bottom. I am still learning things that would still be helpful many months down the line and the final weeks such as sequences within blueprints.



# Enemy

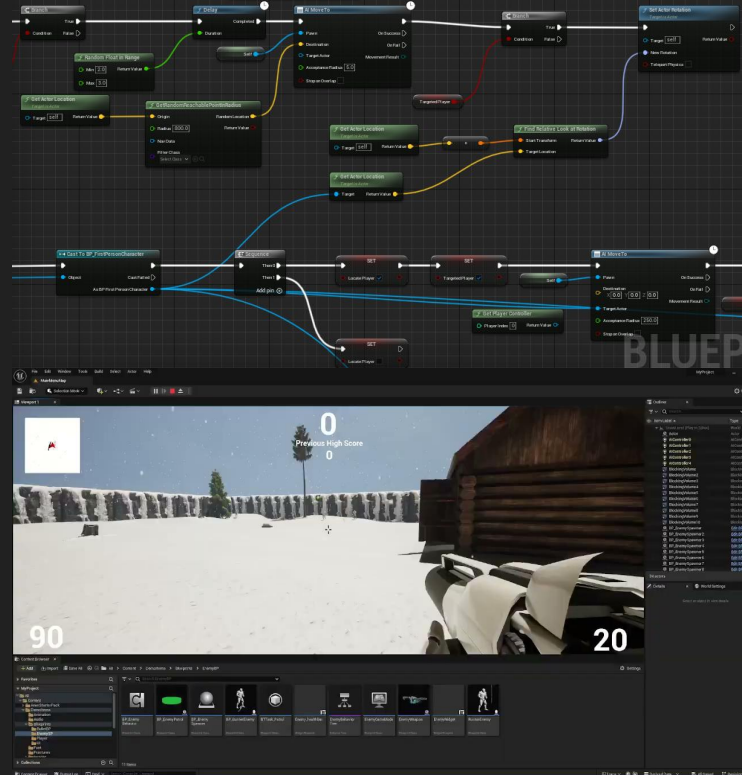
The enemy required a blueprints for animation, spawning bullets, UI widget, and behavior. The addition of everything together created a basic enemy which targeted the player actor and fired projectiles at the player. Animation is important because you can spawn additional projectiles based on the animations.



The blue arrow is the direction the capsule/actor is facing(Located at the center of the capsule) and the red is the arrow to spawn bullets

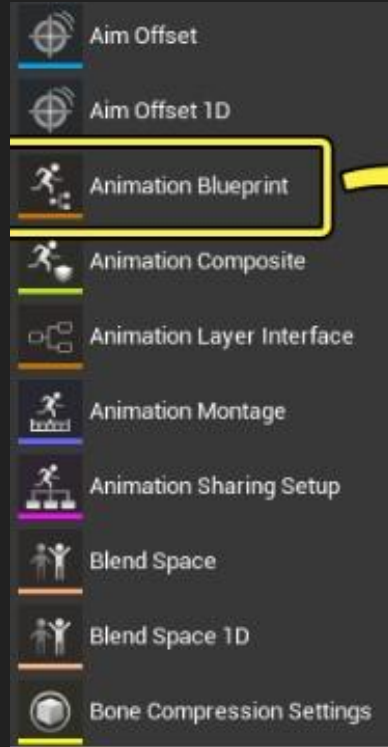
# Enemy Movement

Enemy movement has been an issue as it would break the animation most of the time or just fire bullets without the enemy looking at the player. The solution I utilized is through random patrols every few seconds to simulate dodging and it will occasionally look for the player after.



# Animation

Animating both the player and the enemy has stumped me for a very long time. When it came to first person, I have tried to change the model to a full body character and tested for 10+ hours but ended up needing to scrap the idea because of the bugs that came with it. Many of the bugs confused me as someone new to Unreal Engine because I did not know what to look at in the blueprints and all of the different type of animation blueprints.

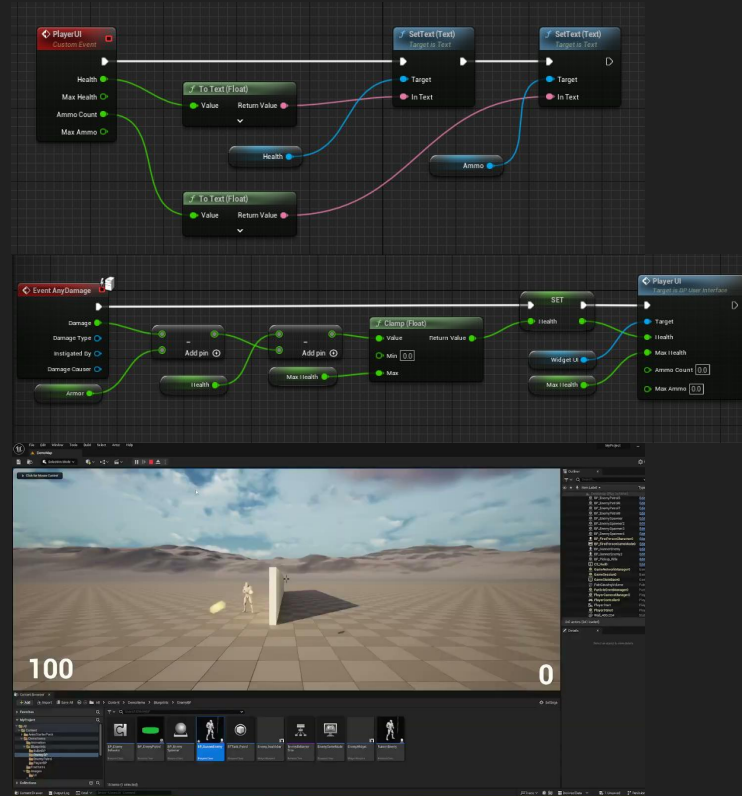




# Player

The additions that I have contributed towards the player is the enemy bullets damaging the player and the player UI.

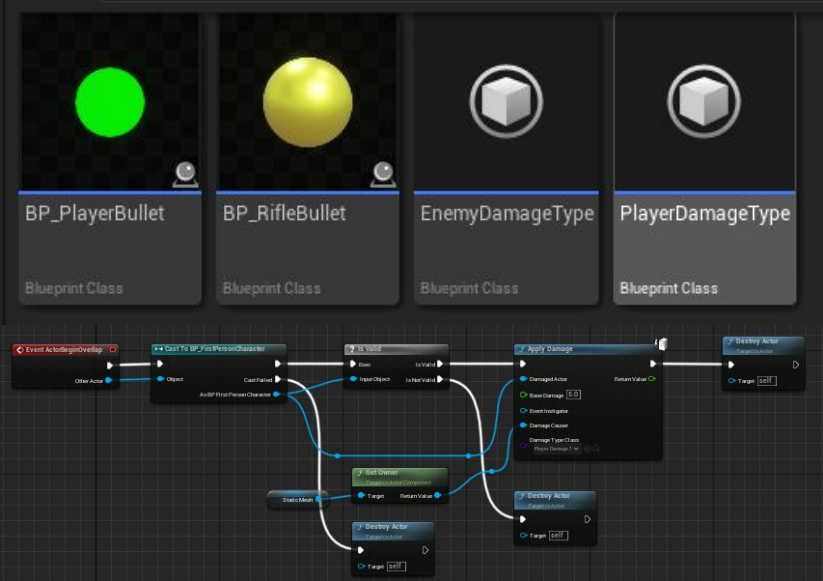
Setting up the ammo system was an issue as I needed to learn how to connect the ammo count and needed to learn what Suraj has done with the ammo system. I decided to remake the reload and ammo system with an input system.



# Damage

The damage system I have settled upon is by settling up damage types so the player or enemy will only take damage from bullets with a certain damage type set.

I have utilized the overlap system instead of the on hit system. I did not want to touch Ue5's baseline system and learned off the overlap system when the actor object touches and overlaps with another object.



This bullet blueprint would try to find if the other object is the player to deal damage. Otherwise it will destroy itself when the cast fails.

# Menus

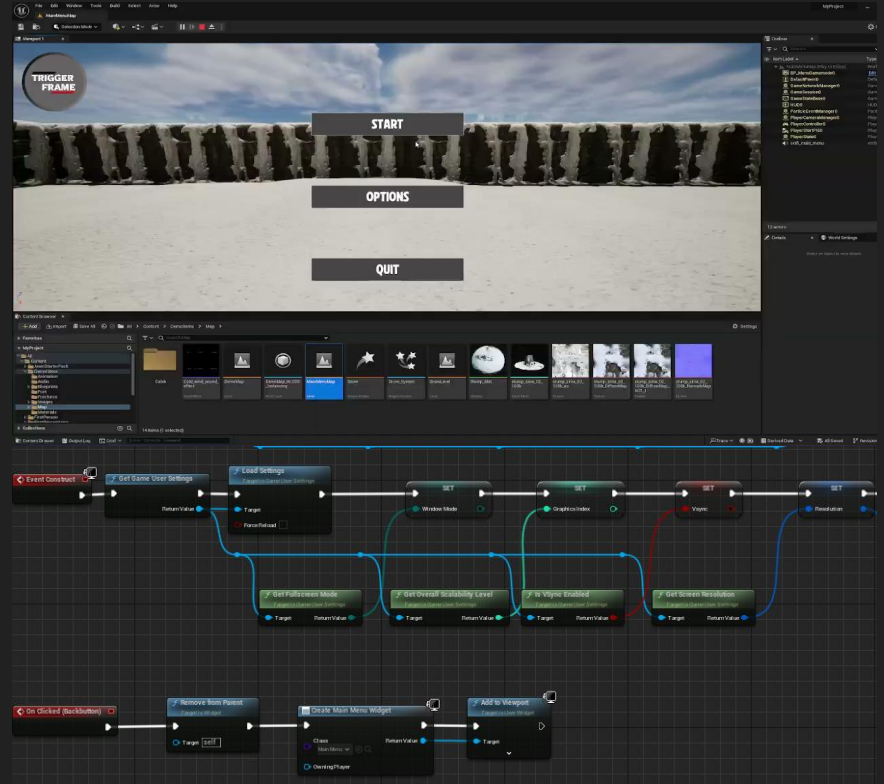
The menu system required a large amount of logic and took around 12 hours to finish the menus. I needed to learn how to create the menu however, the problem was that I needed to learn how to save the data from the options menus across game. This was also the issue I had to learn with the high score system(The radar was done by Suraj).



# Menus

I needed to learn whether to send the user to another scene or whether it is better to remove a widget and simply add another widget into the scene.

I needed to research the high score system because there was a save blueprint in UE5. I needed to learn how to save information into game instead of just the widgets.



# Additions if there was more time

- Fix the High score system as it is still bugged.
- Sound additions and fixes
- Learn if there is post processing
- Learn enemy AI

# Conclusion

The experience I have received practicing Unreal Engine enables me to understand the basic functions of Unreal Engine 5. There is a large amount of things that I would still be experimenting with such as Fractures, Behavioral Trees, Tree events and other types of blueprints. The level of nuances in Unreal Engine compared to Unity3D is much more refined and provides us with greater control over our game in comparison. The greatest gains I have gathered is the process of creating a demo in this engine.