Kevin Huang
Professor Ryoya
Culmination
11 December 2023

Final Reflections

This project that I have embarked with Suraj over the course of this semester has taught me a lot about Unreal Engine 5. From a beginner with zero knowledge of Unreal Engine, I currently have the experience to work with a sizable amount of resources that Unreal Engine provides us. The project is made in Unreal Engine version 5.3.1 but, we were unable to delve too deeply or know what Unreal Engine 5 is capable of compared to Unreal Engine 4 in our goal to create a demo first person shooter game. The first steps of this journey have been confusing.

The progress to learn Unreal Engine is not beginner friendly. Both me and Suraj decided to host watch parties for several hours at night or during our first few weeks of the project. We were used to coding in C# and we were ready to jump into Unreal Engine and excited to code in C++. It shouldn't be too large of a jump in our opinion until we got into the coding side of Unreal Engine. We were paralyzed; not just by choice but from visual programming altogether. The blueprint system was intimidating and that made us more susceptible to the tutorials. We needed to differentiate between usage tutorials, good tutorials, and tutorials that will waste our time. Thinking back to these moments, having a direction or asking a forum for assistance may have assisted us through this journey more than looking through online resources on our own.

I have learned how to incorporate and utilize online resources and how to insert them into our code in a neat fashion. If I had more foresight, I would have had our group settle upon a solid foundation on how to start our code. I have created my blueprints in rows with reroute nodes but I did not create comments in my blueprint. With the amount of code, it would be useful having comments for anyone else looking at my code to understand. In the future, in group problems I would add some comments for people to navigate my code. I have looked into Suraj's code which has nodes going back and forth in confusing matters and the requirement of asking him how his blueprint works should not be needed if I wanted to edit a certain aspect of our game.

We created a workflow between the two of us. The workflow is important because we only had one day we could meet every week and some of the times, that meeting period has to be canceled for any issues that come up on Suraj's side whether it is Internships or emergencies. We communicate through text and discord and we list the things we are doing for the week in case we are unable to meet up. This may not always work out because sometimes we may do more work than we stated and we work on some of the same things. There is also a communication issue if these weeks persist because sometimes, Suraj would work on something I have completed weeks prior only for his work to be scrapped.

The learning process could be simplified by learning through longer tutorials of people working through longer sessions. The resources used are targeted towards very specific issues that we had and I think in the future, learning from a broader overview could have informed us of new information easier and in a step by step process that would make sense. Instead of going through a tutorial that is fragmented videos that changes their code and not utilizing a large portion of their code as their tutorials go on. The benefits of segmented and short tutorials is a fast and rapid way to understand the problem we want to solve and a potential solution we can utilize. The benefits of long tutorials can give us an understanding of what it takes to code a FPS game. It can provide us with a list of things to do and allow us to check back on things we may have missed along the way. We looked for shorter tutorials as we wanted to keep the sanctity of our work as much as possible without blindly following someone else's work.

After understanding the learning process on how we have learned Ue5, we jumped straight into Ue5 and had to continue to use online resources. With the sheer amount of in-built Ue5 tools, I was only able to scratch the surface and if I would like to learn more about. From my understanding, just utilizing the actors, widgets, and regular blueprint classes are all we need to create the core of a game. The things I haven't gotten to understand are the various other blueprint types and behavioral trees.

I have seen tutorials involving a behavioral tree for a random patrol. I decided to not utilize behavioral trees given the time I had left in the project because I wanted to make sure I can make the movement AI less buggy and weird. I decided to settle with random patrols that occur no matter what to simulate dodging and occasionally, if the enemy has found the player; It will turn around to find the player. If I had more time, I would look into behavioral trees because from my understanding, it would refine what my AI would be doing. My enemy AI could only choose to do one thing at a time or things will overlap and look weird/break. Learning behavioral trees will be able to make it much smoother and operate much better overall.

I would love to improve upon the map and environment by adding fractures into the game to create destructible terrain. So far, health is an important resource due to the inability to recover it. I would refine the health system, create new weapons, enemies, and overall gameplay loop. There are a lot of missing basic gameplay qualities to change our demo into a game and quality of life upgrades for the player. As a medium to boost my Ue5 knowledge, this project has assisted me understand the ins and outs of what a fundamental game needs.

I have learned a lot about saving data for the options within Unreal. Unreal already has a system that does system settings built in and a save load system for the settings. We had to import the code but otherwise it was straight forward. I would love to learn how to create an actual save and load system that is separated from the in-built Unreal Engine events to learn more about this to use it in my own way.

In summary, this project has taught me the fundamental knowledge needed to code for Unreal Engine 5. I have learned how to animate widgets, apply animations onto actors, and create basic behaviors for my enemies. I gained insight on how events and input systems work within Unreal. I would create a new project while using the logic that is good and makes sense from this code in my new project. This has been a good placeholder project although there is still some visual code confusion due to the fact that there is very little code clean up for me to understand Suraj's blueprints. Finally, TriggerFrame has given me the information needed to make what I have planned to make in the future in Unreal Engine.