Quake Inspired Setup

Group number: 3

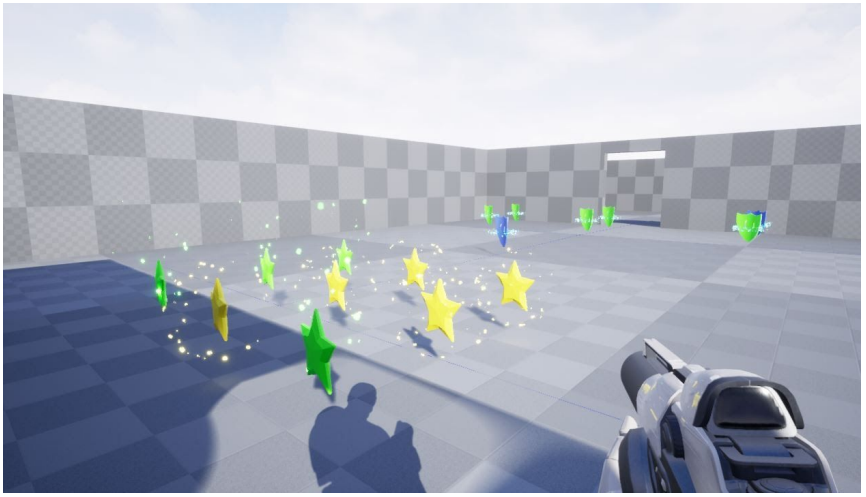
Ivan Zheng - 13536901
Danny Jiang - 13550417

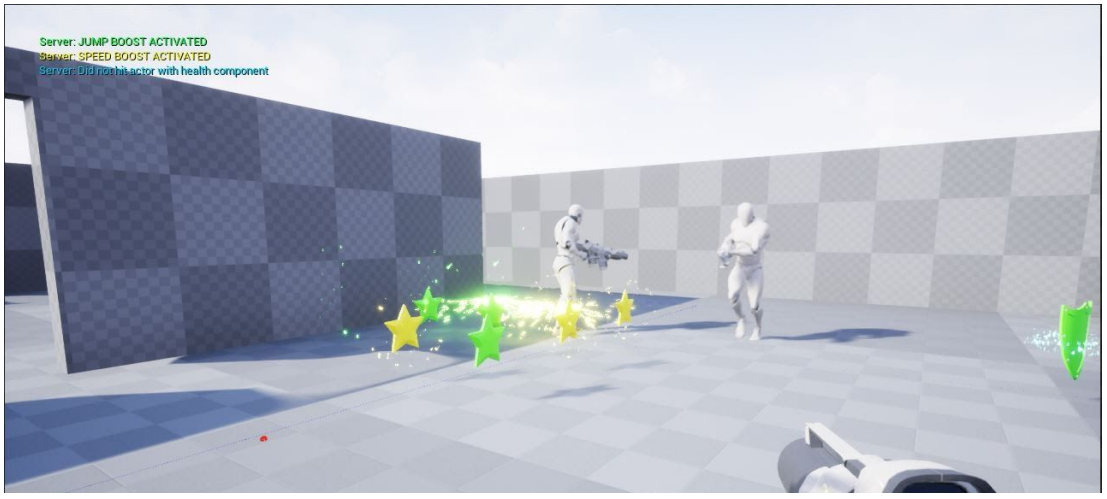
Summary

The plan for Assessment 4 presented in Assessment 3 was to have a Quake inspired project with power ups and shield buffs implemented and working in a multiplayer environment. The power ups consist of a speed boost powerup which increases your sprint speed, as well as a jump boost powerup that increases the Jump Z Velocity built into the player character. The shield is implemented as an extra layer of health. We wanted both the power ups and shields to work in multiplayer. We also planned to implement a basic respawn method so there was an aspect of competitiveness to it.

In terms of what has been implemented since then, we have made most of the existing project functional in multiplayer for a dedicated server, including the powerup and shield behaviors. However, instead of a complete game inspired by Quake, we've made more of a setup that features behaviours similar to Quake. Additionally, we have added a respawn system that destroys the player character when their health reaches zero.

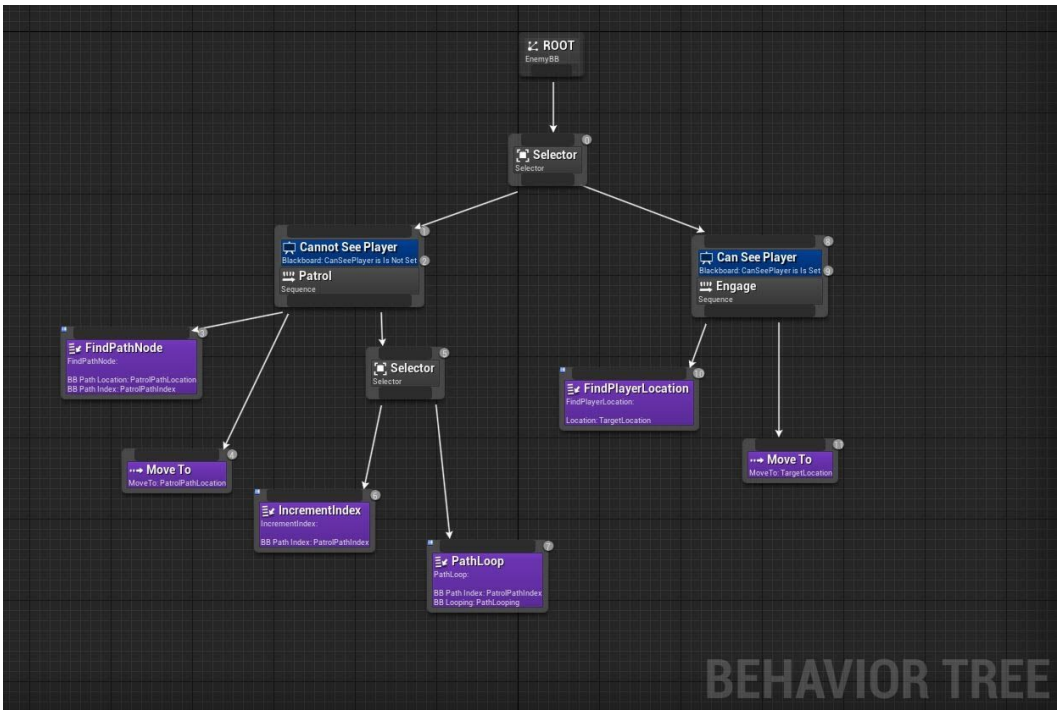
On top of that, we have added particles for most of the core behaviours of our project, such as the power ups, shields, hitting a target, and picking up power ups



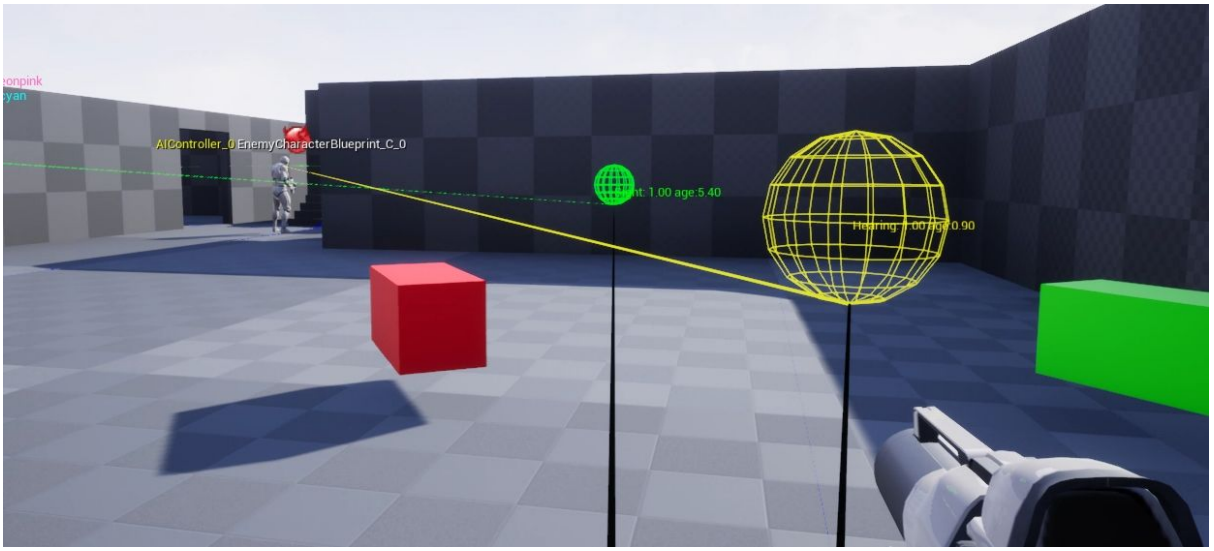


AI Contribution

For AI contribution, Ivan contributed to the creation of a behaviour tree to replace the existing finite state machine to control the behaviour of the AI. The behaviour tree was made using the ‘behaviour tree’ component built in the Unreal Engine. It features a ‘patrol’ state and an ‘engage’ state.

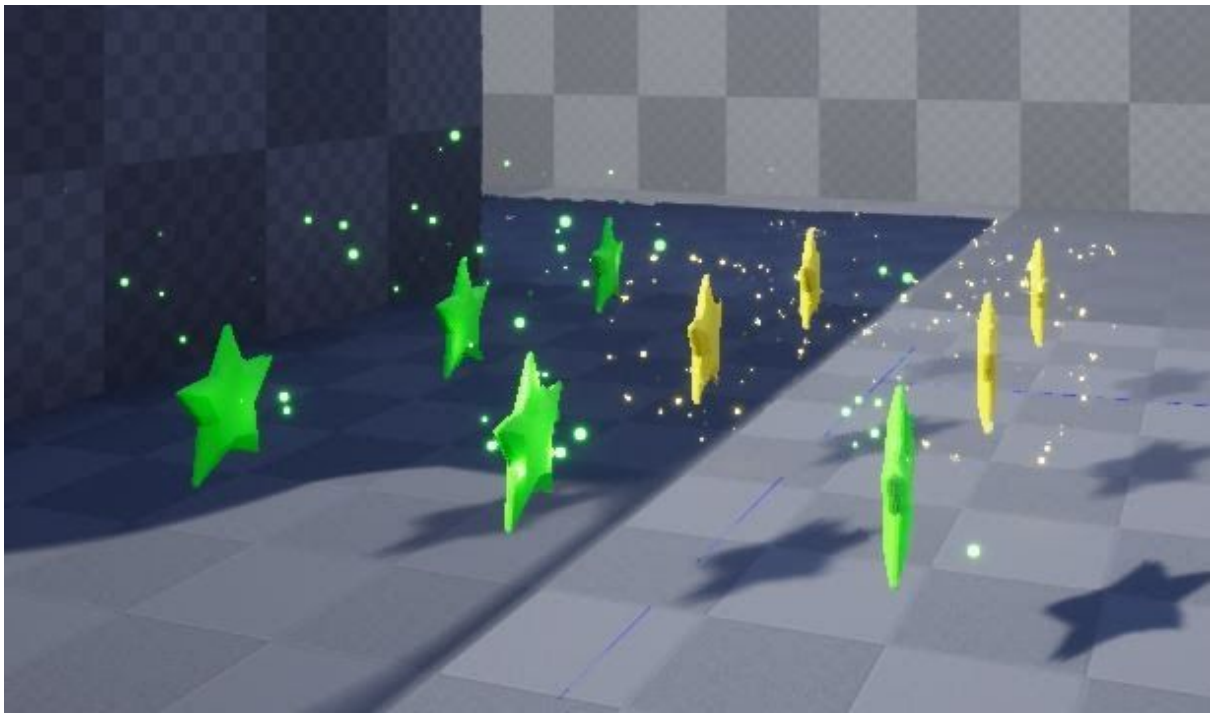


For AI contribution, Danny contributed to the creation of the AI hearing perception on the enemy character where if they hear a sound such as the distract whistle he added, the enemy character will hear that and mark the location of where did the enemy character hear it from.

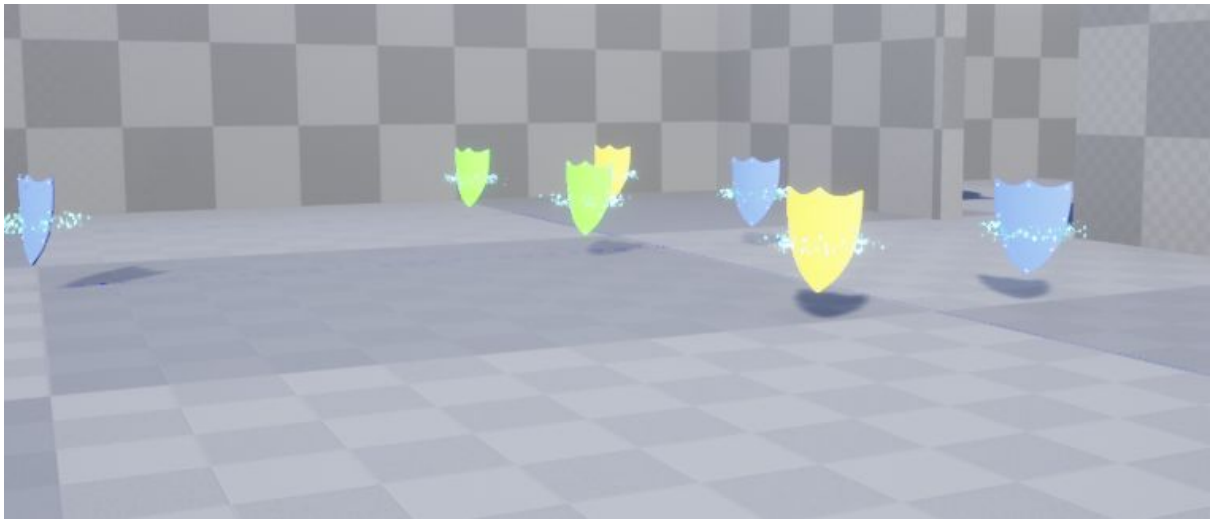


PCG Contribution

Ivan contributed to the creation of the powerups in the PCG contribution. He made the speed boost powerup and the jump boost powerup, which increase your sprint speed and jump height respectively. The speed boost powerup increases your max walk speed from 600.0f (default) to 1350.0f. The jump boost powerup increases your Jump Z Velocity from 420.0f (default) to 700.0f.



Danny contributed with the creation of shields for the PCG contribution. He made a shield with 3 types small, medium and large with the corresponding values 25, 50 and 100. These are generated randomly according to their percentages 60% for small, 30% for medium and 10% for large. I added materials to the pickups so that it would be easier to tell them apart such as green for 25, blue for 50 and yellow for 100. Also made the max shields to be 100 and it won't add any more on to the player if they pick up more.

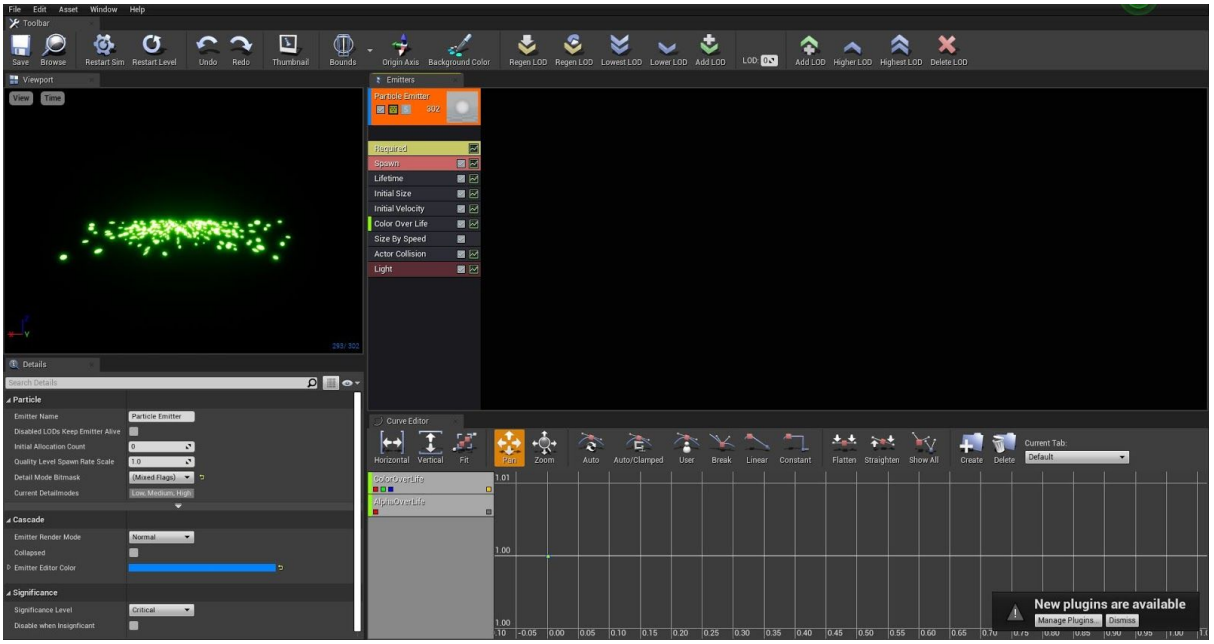


Polish Contribution

Ivan - For the particles, the learning materials I used include the:

- ‘Intro to Cascade’ youtube series.
- Unreal Documentation
- Unreal Forums

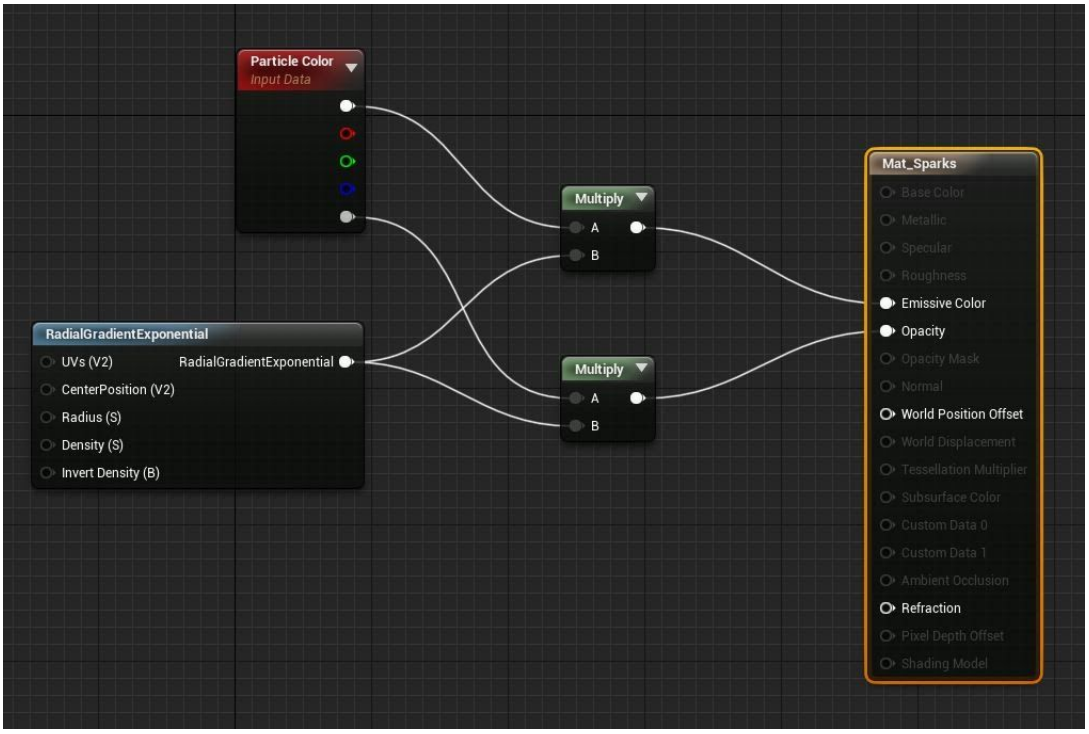
I started off by watching the videos, and then from there, I experimented with various particle effects and then implemented them with the appropriate behaviours. I primarily used Cascade in Unreal Engine to create the particles themselves, then I implemented the particles in the respective blueprints.



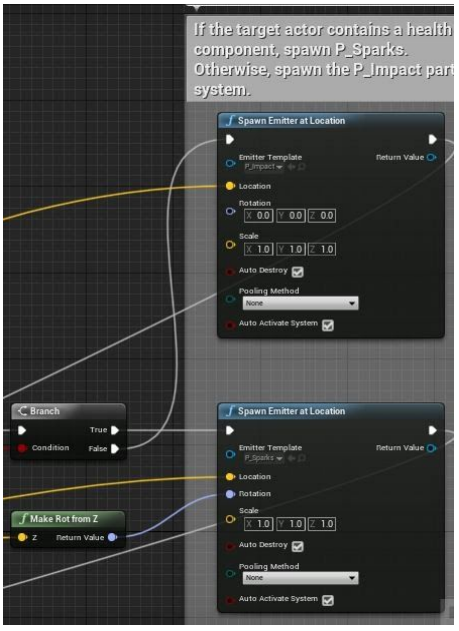
The Jump Boost explosion particle in Cascade

Sparks on hit, Jump Boost, Speed Boost (left to right)

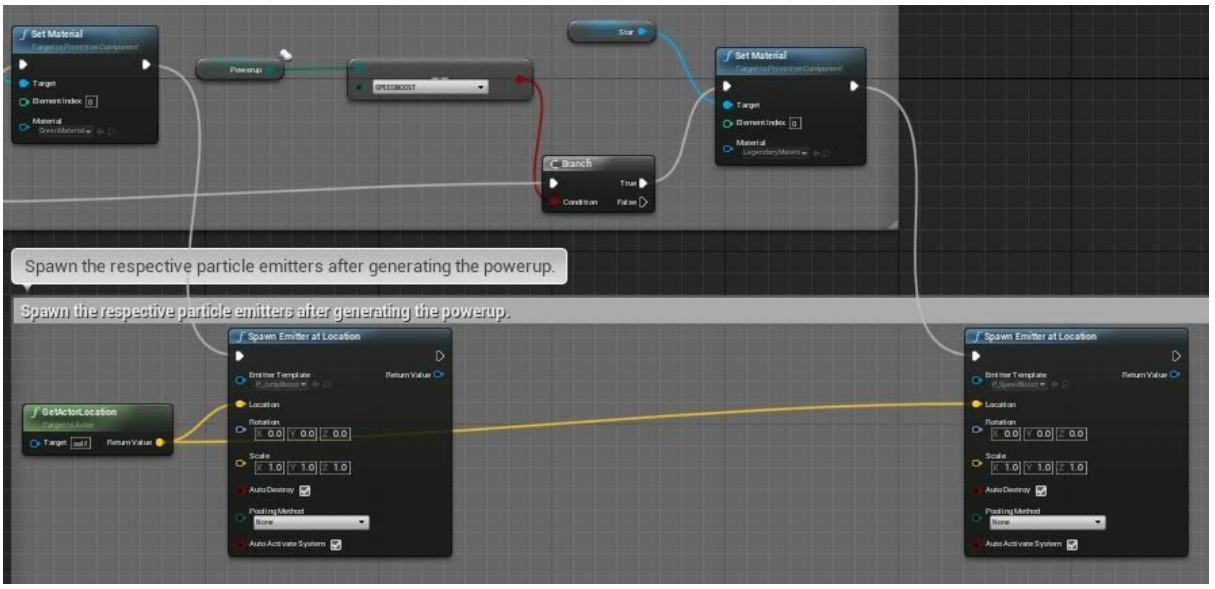
To start off, I made a material in blueprints called Mat_Sparks and assigned the ‘translucent’ blend mode and the ‘unlit’ shading model. This served as the main material used in most of the particles I made. It creates a circular shape, similar to a sphere.



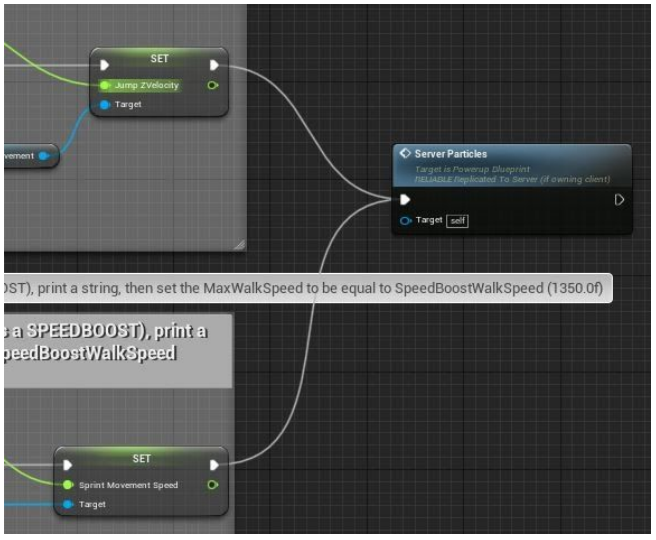
From there, the material can be used in the Cascade Particle System. This particular material emits a slight glow when colour is applied to it. The first few particles I made was for the on hit particles. In the gun blueprint, I made it so that it would spawn different particles depending on whether the actor has a health component (a player or AI)

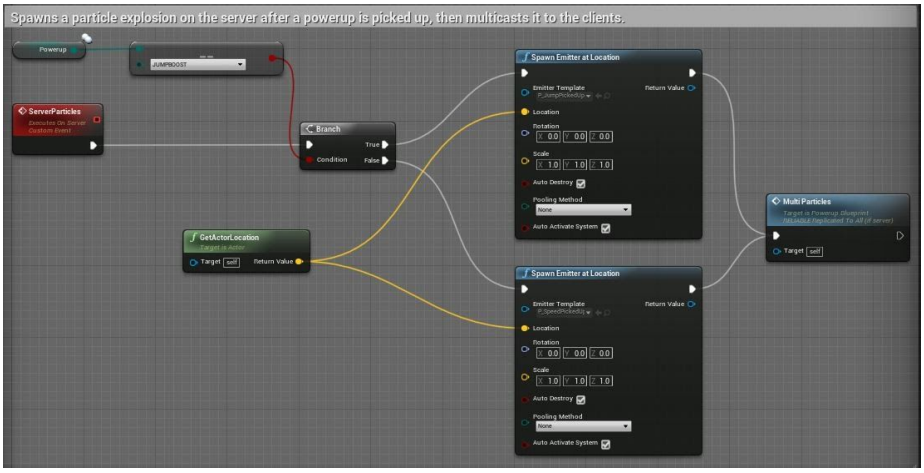


For the power ups and shields, I attached a ‘Spawn Emitter At Location’ node after setting the material for each pickup, which spawns the respective particle systems on the pickup’s location when it is generated



I also created particle explosions when a powerup is picked up, which are separated into a server function and a multicast function. This initiates when a powerup is picked up by a player character, as seen below. The particles are created on the server then replicated to the clients.



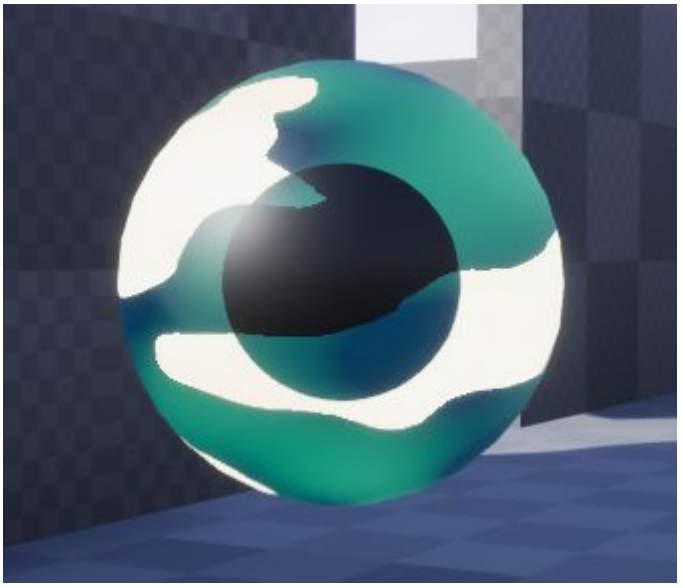


Overall, the benefits of implementing particle systems into our project include making the game feel nicer and less bland. It makes the objects look more visually appealing and it also makes it more realistic in a sense, as seen from the bullet particles. From my perspective, it adds more life into the project and provides more visual reactions from specific behaviours.

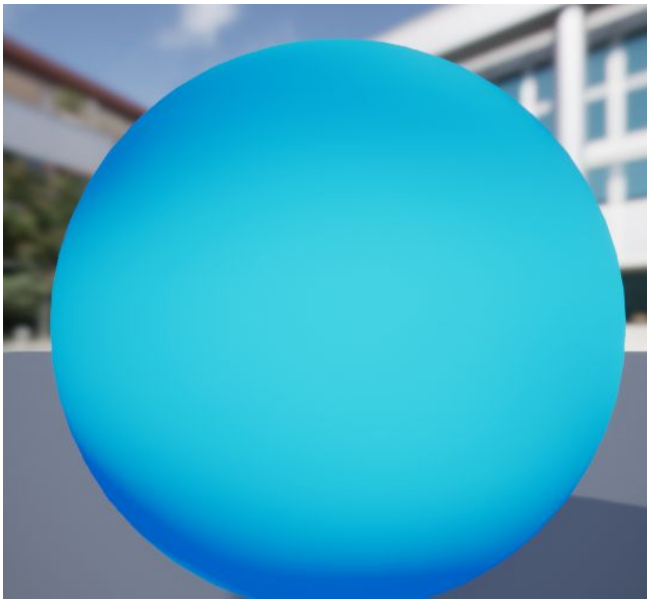
Danny - Shaders and other post-processing, the learning material i used:

- Shaders tutorial-https://youtu.be/mig6EF17mR8
- Unreal Documentation
- Unreal Forums

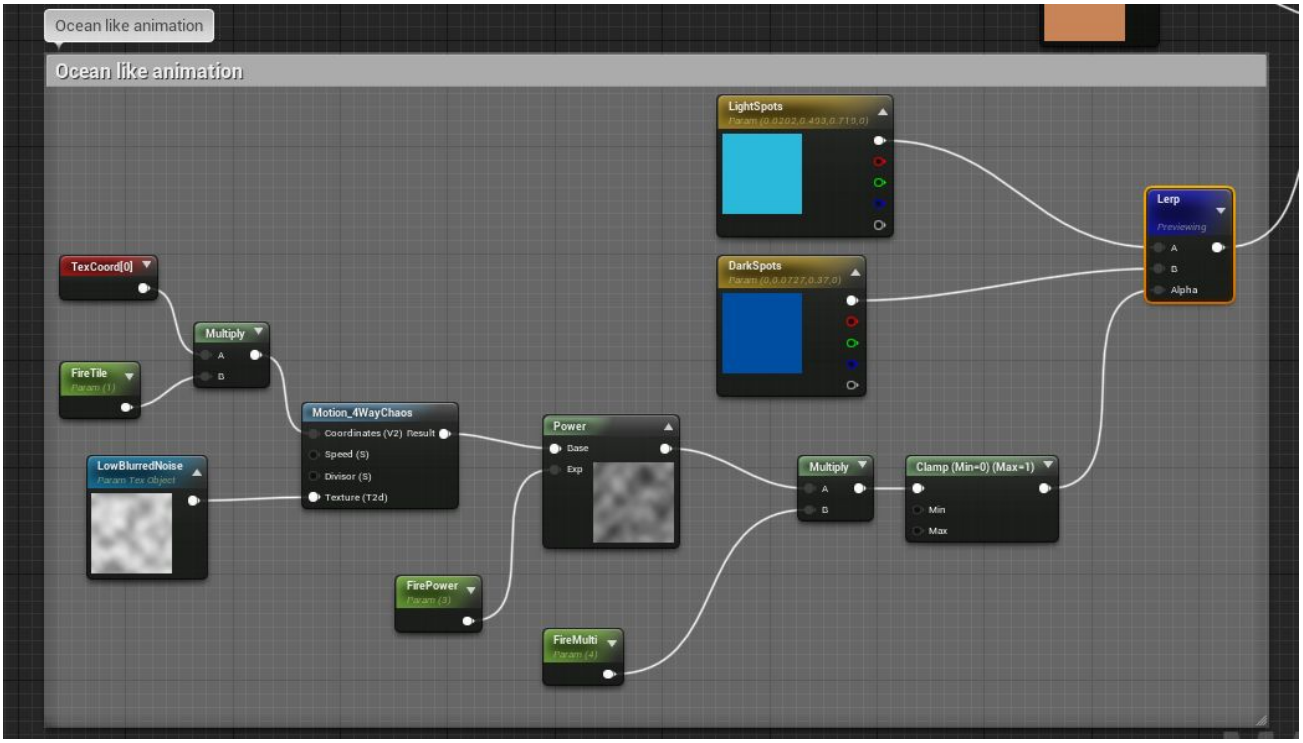
I started off with watching videos on tutorials on how to make a post-processing material and decided to go with a shader. I started off with something similar to the tutorial and then started experimenting with different colours and textures to create something cool and interesting. The material changes with the player is looking at it from different angles.



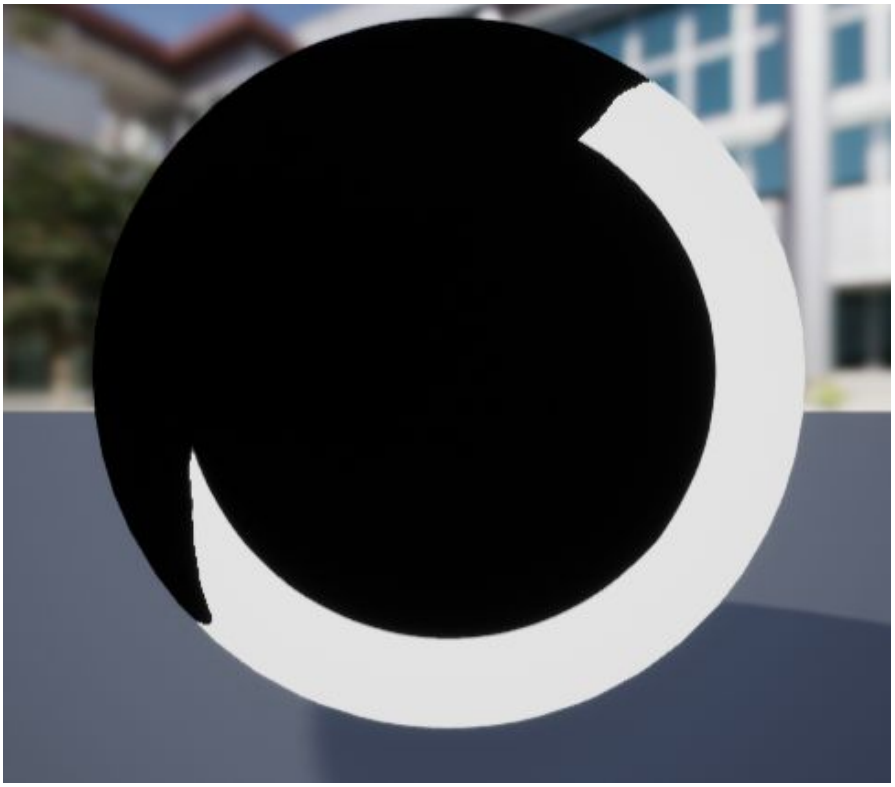
To start off I made a new material called PostProcessingShaderMaterial and started off with ocean waves animation. This was just something after I experimented with some different colours and finally came to the decision of making it ocean like.



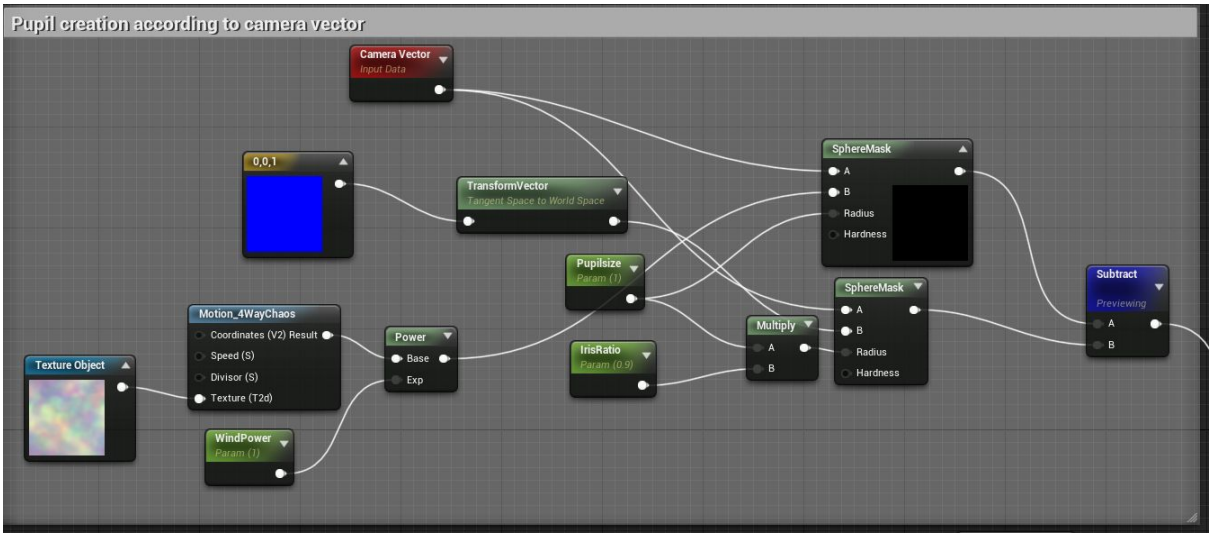
For what i did was taken the LowBlurredNoise from the engine content and added the motion 4 way chaos which samples different parts of the LowBlurredNoise and makes it move randomly. Then I increased the power of it and multiplying by the power. The 2 colours that are lerped are the light and dark spots of the blurred noise.



Then I started with the next part with the material with the camera. This part allows the material to change or look at the camera vector which also changes the look of the material.



I first made 2 sphere masks which is for the pupil of the shader. Then add the camera vector to both of the sphere masks. I added a texture with the motion 4 way chaos to make it change as u look at it from different angles for the outer ring. While the inner part is made to be all black.



Overall, I found it cool and interesting to create although it might not have much to improve the project. It could be used for many different things such as a portal or an eye that watches the player but I decided to just make it as itself on a sphere object because it looked cool. This can help me in the future with different implementations that I could use this for.