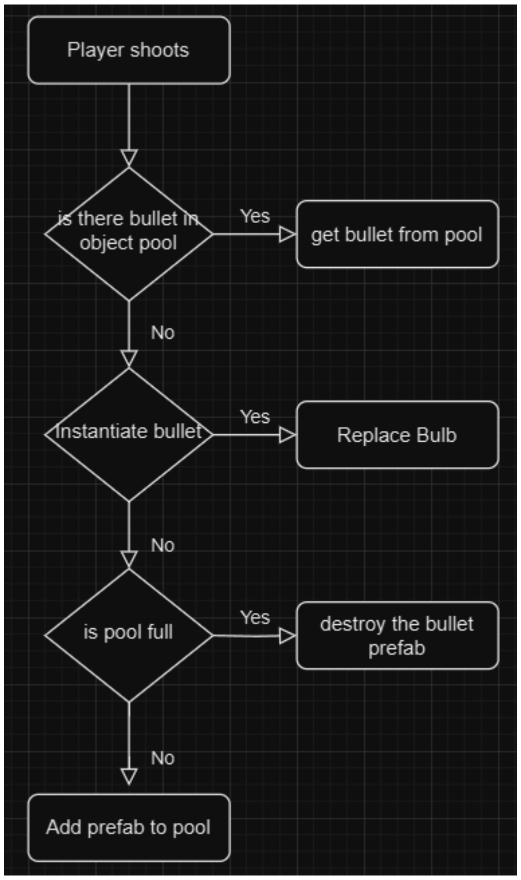
UML Diagram and explanation of implementation
Chia Lo Chen

Improvements from the previous assignment

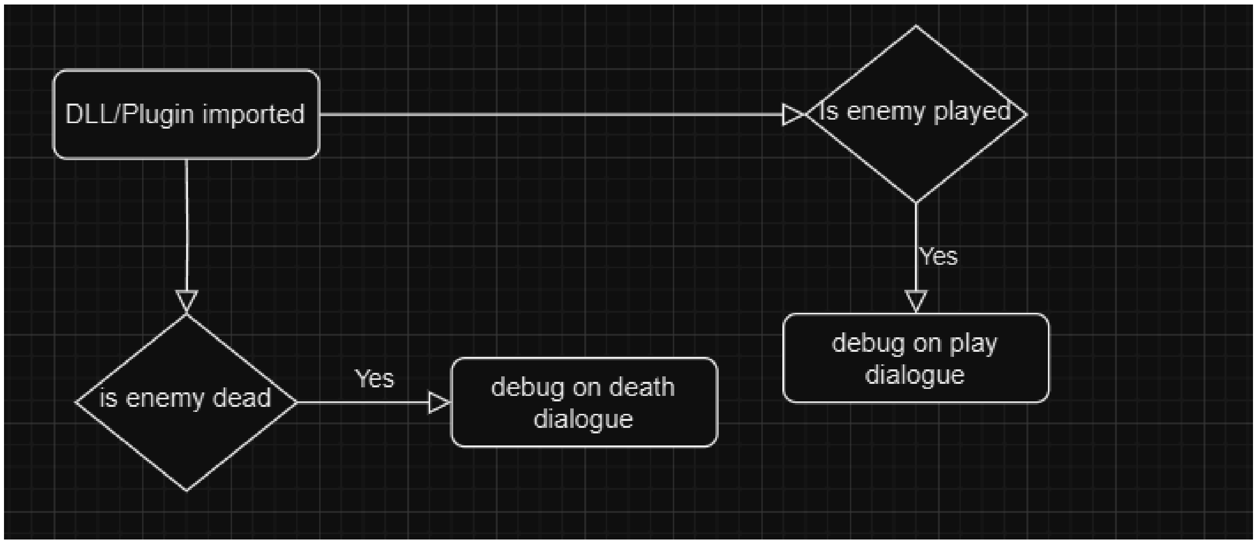
In the improvement from the previous assignment, I originally wanted to improve the command pattern for it to record mouse movements as well but I didn't plan out the time well so I ended up improving the visual and the labyrinth generator code. I add it in some materials for the wall and floor for better visualization compared to the old scene, which is pure white. Then for the labyrinth code, I decide to add a 50% chance of instantiating the wall in a vertical direction giving the scene a more versatile look.

Optimization



I chose performance profiling for optimization because I believe a lot of CPU usage goes into instantiating the bullet prefab. The object pooling would set the bullet prefab active to false and add it to a list of pool objects and whenever ever player shoots again, object pooling would set the bullet prefab active to true and shoot out the bullets. If the player shoots more than the object pooling size then the code will still temporarily instantiate a new bullet but will destroy any bullet prefab that exceeds the pooling size.

Plugin/DLL

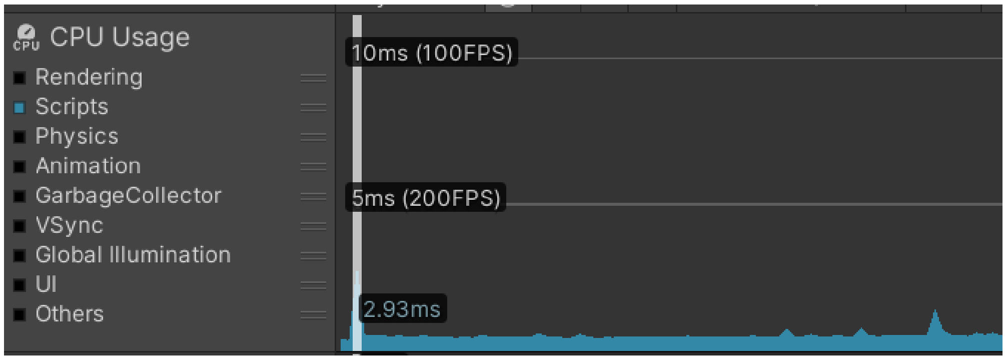


For plugins, I decided to make a simple dialogue for the enemy when they are spawned and killed. The plugins are quite complicated for me, I tried to do something that gives effects to the player but failed to do so. What the plugins do is add in additional dialogue and have a trigger to trigger the dialogue, the user would need to add in the DLL import to link to the plugins and then set up the corresponding dialogue to where the user wants it to trigger.

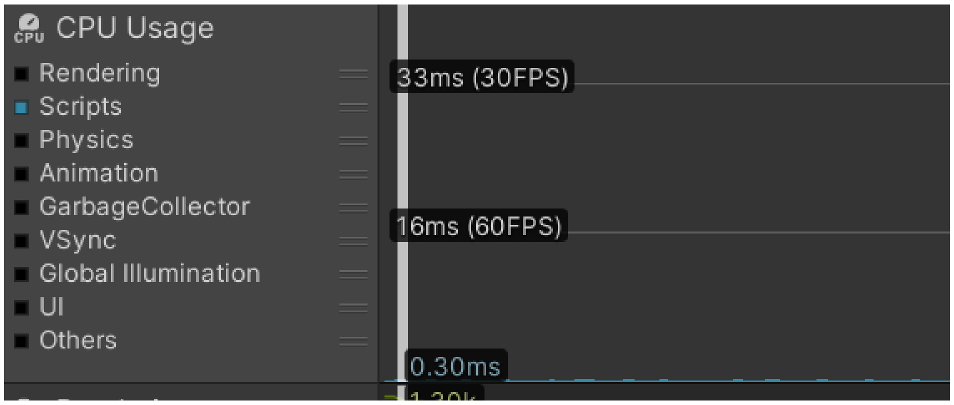
Performance profiling

The performance profiling is implemented to indicate where the optimization improves the overall performance of the game. In this case, object pooling decreases the CPU usage being used during gameplay. It is helpful because, in a multiplayer scenario where there would be more bullet prefab being instantiated, too high CPU usage would create huge lag issues.

Before the object pooling implementation



After the object pooling implementation


CPU Usage
Rendering
Scripts
Physics
Animation
GarbageCollector
VSync
Global Illumination
UI
Others
33ms (30FPS)
16ms (60FPS)
0.30ms