What was hard

- The pointer that points to the closest enemy. The enemy array was not emptying correctly when an enemy died, therefore we had to rework the OnEnemyDied delegate to also pass the enemy index in the array.
- Getting the enemy to rotate towards the player. I wanted it to lean towards the player, not just look at player and lean forward. Required some finicky vector math.

What was easy

- Enemy logic was hard until we switched to using enums for each state the enemy can be in, then it got a lot more manageable and easier to customize.
- The projectile onOverlap logic was relatively strait forward

Buttons to play the game:

- WASD – Move
- Mouse pointer – Aim
- Left Mouse Button – Fire
- SHIFT – Dash
- R – Reload
- Space – Slow down

References

Creating an InGame User interface in Unreal 4 and C++ | by BlueBubbleBee | The Startup | Medium. Retrieved from: https://medium.com/swlh/create-an-ingame-user-interface-in-unreal-4-and-c-edf3311b14c9

Physics Mass Override. - Development Discussion / Physics - Unreal Engine Forums. Retrieved from: https://forums.unrealengine.com/t/physics-mass-override/277555/2

UE4 Tutorial: Character Health Bar UI Using C++ - YouTube. Retrieved from: https://www.youtube.com/watch?v=nNe-NSrtYUk

Unreal Engine 4 Particles - Cascade - Intro & Fireball – YouTube. Retrieved from: https://www.youtube.com/watch?v=UJJxhW9jciM

Unreal Engine C++ Developer: Learn C++ and Make Video Games | Udemy. Retrieved from: https://www.udemy.com/course/unrealcourse

Unreal Engine C++ Fundamentals - AHUD, UUserWidget & UWidgetAnimation – YouTube. Retrieved from: https://www.youtube.com/watch?v=lYfXkxlInVI

All 3D models and textures were made by Mathias M. Mørch.

All sound effects are from 1001 Sound FX (2004) by Sony Creative Media, which used to come bundled with Sony Movie Studio and Sony Movie Studio Platinum.