Final Project Reflection

Jiangwen Nong, Mingliang Wang

For both of us, this was our first time developing a VR experience. We used Unreal Engine to create a game prototype, and I was surprised to find that the workflow for building a VR project wasn't significantly different from developing a regular game project, such as an FPS or RPG game. The primary difference lies in the input setup. In traditional video games, inputs are handled through keyboards, mice, and touchscreen panels. However, VR games use an entirely new input approach that focuses on the natural movement of the human body. The design of VR devices reflects this shift. For this project, we used Meta's latest VR device, the Quest 3, which includes two controllers and a headset. Players hold the controllers in their hands and wear the headset on their heads. In this setup, the headset serves as the in-game camera, which receives the rendered view of the game world. Unlike traditional games where players rotate the view using a mouse, VR headsets allow players to look around freely to explore different areas of the scene. The two controllers serve as input devices for hand-based interactions. These inputs can vary widely depending on the player's actions, such as moving their thumbs, index fingers, or other fingers. Based on the specific input setup, the game can produce different responses, like allowing players to grab objects or drop them. This design allows players to interact with the virtual world using their eyes and hands, creating an embodied and tangible experience.

One of the major components of this project is rendering. We aimed to explore the rendering limitations of the device, so we developed a relatively complex shader for volumetric clouds and an advanced visual effect known as Gate of Babylon. Interactive weapons were another key focus of the project. Our initial goal was to create a variety of weapons with unique mechanisms, allowing users to hold these weapons in the scene and activate different effects. However, during tests, we discovered that the graphical computation was extremely demanding due to the volumetric cloud shader. While the project ran smoothly without additional interactive elements, adding NPCs and weapons significantly decreased the frame rate (FPS), causing it to drop to an unplayable level. As a result, we were forced to include only one enemy NPC and a very small number of weapons in the scene.

Almost all video games follow the input-output design pattern, which is why I mentioned earlier that the workflow for VR projects and regular projects is not fundamentally different. In both cases, input sent to the game is processed by algorithms built in the game system. Then it will generate output. At its core, input is simply data — data that triggers specific outputs and feedback. The primary difference in VR development is the use of new devices for input. When setting up input for a VR project, we need to pay close attention to the design and functionality of VR devices. However, once the input system is configured, we no longer need to worry about the data transmission.