

Engine System Update #2

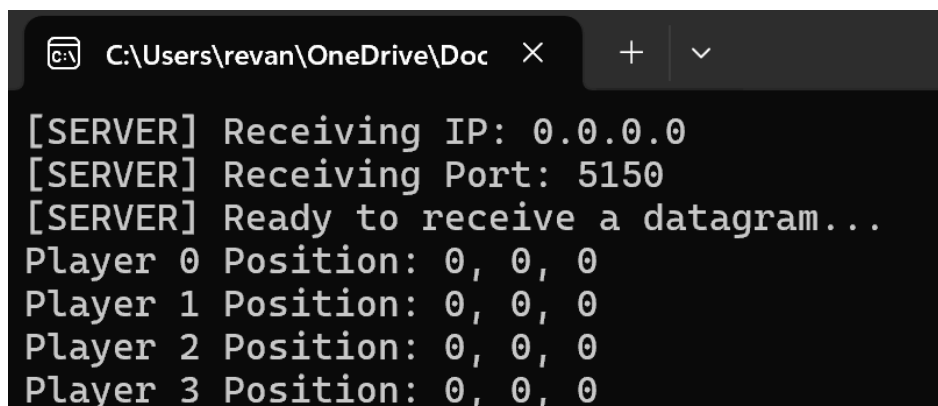
Project - GAMES6320

During this second week, I continued progressing on my networking system and implemented the client-side code for the UDP model. This involved plenty of debugging, where I learned a lot about the communication between server and client in such a system. Although I haven't got everything working as expected, I still believe I'm in a good enough place going into the last week.

Game Client Implementation

Just like the implementation for the game server last week, I created a platform-independent interface for the Game Client, with platform-specific (Windows) implementation being private. During this process, I had a few problems with the structure of Windows sockets, such as receiving and sending data to and from the game server. I kind of had to dig deep into how these sockets are initialized, cleaned up, and talked to each other in a UDP model.

Eventually, I managed to get the Game Client implementation working, where each player can start a new instance of the Game Client, and join the game while everyone is connected to the Game Server. However, I realized that currently, a new 'instance' does not show anything, since it is not integrated with a 'game' yet.

A screenshot of a Windows command prompt window. The title bar shows the file path 'C:\Users\revan\OneDrive\Doc' and standard window controls. The command prompt displays several lines of text: '[SERVER] Receiving IP: 0.0.0.0', '[SERVER] Receiving Port: 5150', '[SERVER] Ready to receive a datagram...', 'Player 0 Position: 0, 0, 0', 'Player 1 Position: 0, 0, 0', 'Player 2 Position: 0, 0, 0', and 'Player 3 Position: 0, 0, 0'.

```
C:\Users\revan\OneDrive\Doc > [SERVER] Receiving IP: 0.0.0.0
[SERVER] Receiving Port: 5150
[SERVER] Ready to receive a datagram...
Player 0 Position: 0, 0, 0
Player 1 Position: 0, 0, 0
Player 2 Position: 0, 0, 0
Player 3 Position: 0, 0, 0
```

The image above is the output and debug statements that are printed when running the Game Server. As you can see, currently it recognizes the number of players in the game (I started 4 instances of Game Client in this case). However, since I have not setup any positions yet, they are all just initialized to zero.

Plans for the Next Week

As the server and clients are now talking to each other and exchanging data between themselves, the plan for the last week is to actually integrate the system with the MyGame project, and show multiple instances of the .exe file running simultaneously.

Technically, I would like to have the Game Client implementation fully integrated with MyGame, where each instance of MyGame can be opened by a player to join the game, and meshes, effects, and positions are all updated for all the players while playing.

I have already been trying to incorporate the Graphics system with the networking, but I have not managed to figure out the dependencies yet. This is what I will be focusing on for the next week leading up to the presentation. Essentially I want to show a proof of concept of my networking system, and data being visibly exchanged between different clients.