Milestone 1: Server-Based Asset Loading Implementation

For the first milestone, I implemented server-based loading of game assets, shifting from local disk storage to network-based asset retrieval. This upgrade enhances the flexibility and scalability of the asset management system by allowing centralized control over game assets, facilitating updates and modifications without needing to redistribute local files.

To achieve this, I developed an HTTP server to host files from the AssetsOut folder, making them accessible over the network. Previously, the game engine's asset loading depended on the FileReader.cpp component within the PrimeEngine FileSystem, which retrieved files directly from the local disk. I modified FileReader.cpp to perform HTTP GET requests to the server, enabling the game engine to fetch necessary files dynamically.

With these adjustments, the game engine can now request and download required asset files from the server, storing them in a memory buffer for seamless access. This approach ensures compatibility with the existing file I/O system while integrating network-based asset management, improving both convenience and efficiency by allowing assets to be managed remotely without altering existing workflows.