

Seth Messier V00975362
Dario Monette V00983158

Biweekly Update On Peer to Peer Game

We first did some research about what different libraries we could use for peer-to-peer connections, and settled on using a form of WebRTC with the Sono library (sono.land). This library allows us to easily establish WebRTC connections, put sets of users in virtual “lobbies” that allow us to filter what peers send to what other peers, and handles STUN/TURN server usage. It connects users in a mesh topology, using a persistent channel list to know what peers are a part of what channels, and with that, what peers to send messages to. With this library, we made a basic server that is capable of managing these lobbies and facilitating WebRTC connections using the Sono library.

We then started developing a simple web game that would allow multiple people to play together. The game concept would need to be designed so that lag is noticeable so we decided on one that would have projectiles that could hit players. We found a basic tutorial that demonstrates the creation of a simple javascript web game [1], we then found an expanded tutorial that builds on this by modifying the game to be multiplayer[2]. The multiplayer game uses client server architecture which we will adapt to use peer-to-peer connections using the Sono library.

The next thing we plan on doing is finishing up the multiplayer game and merging it with the WebRTC connections to allow the game to be playable by multiple players. Once the game is working we would like to add the different networking strategies of Delay based and Rollback to see how they perform compared to no strategy and compared to each other.

Bibliography

[1] Chris Courses, "HTML5 Canvas and JavaScript Game Tutorial," *www.youtube.com*, Sep. 21, 2020. <https://www.youtube.com/watch?v=el9idPTT0c4> (accessed Feb. 20, 2025).

[2] Chris Courses, "Online Multiplayer JavaScript Game Tutorial - Full Course," *YouTube*, Aug. 07, 2023. <https://www.youtube.com/watch?v=HXquxWtE5vA> (accessed Mar. 04, 2025).