Seth Messier V00975362
Dario Monette V00983158

Biweekly Update On Peer to Peer Game

A fair amount has been implemented into the game/project since the last update, most important of those things being the ability for the game to establish WebRTC connections and send player data through one-to-all broadcasts! But more changes have also been made to the backend to allow for the facilitation of WebRTC connections and the use of more than one type of netcode between lobbies.

Multiplayer

When a peer starts up their version of the game, it first attempts to connect to a signaling server, this also being the server used to serve the game itself to users. Doing this allows for us to grab a list of other peers we must connect to, and for the server to prevent players from joining full lobbies. Once all these other peers are obtained, we initialize a set of WebRTC connections: we only need one for broadcasting messages, but we need one separate connection for every other peer we wish to get data from (O(n) connections). As long as all these connections are established with no issues, the peer is connected, and we can begin to send data using a standard format to keep other players updated.

Finished Lobby Browser

To make testing different types of netcode easier, I have added the ability to swap what netcode is used when creating a lobby. At this point, we have all the functionality we need from it, and no more needs to be done with it in the front or backend for it to work.

Security Measures

We have implemented multiple checks to prevent mass joining of lobbies and strictly enforce a maximum player count. This is done by having a hard-maximum in our signaling server that denies any WebSocket connections if too many are being made for a single lobby. This is not needed for testing, but would be necessary for any WebRTC based game.

Limitations

The game can comfortably run with 4 players playing the game concurrently, but once more than that joins the webRTC connections start to lag and updates become slow and start to glitch around. Having 4 players is still plenty for testing different lag compensation strategies on the network side, but if this game were to expand to more players other optimization strategies would need to be used.

Next Steps

Next we plan to fully implement different netcode strategies that the lobby selector can choose between. We also plan to add some monitoring to track network performance to see the effectiveness of the different lag compensation techniques under various conditions and with varying amounts of players.

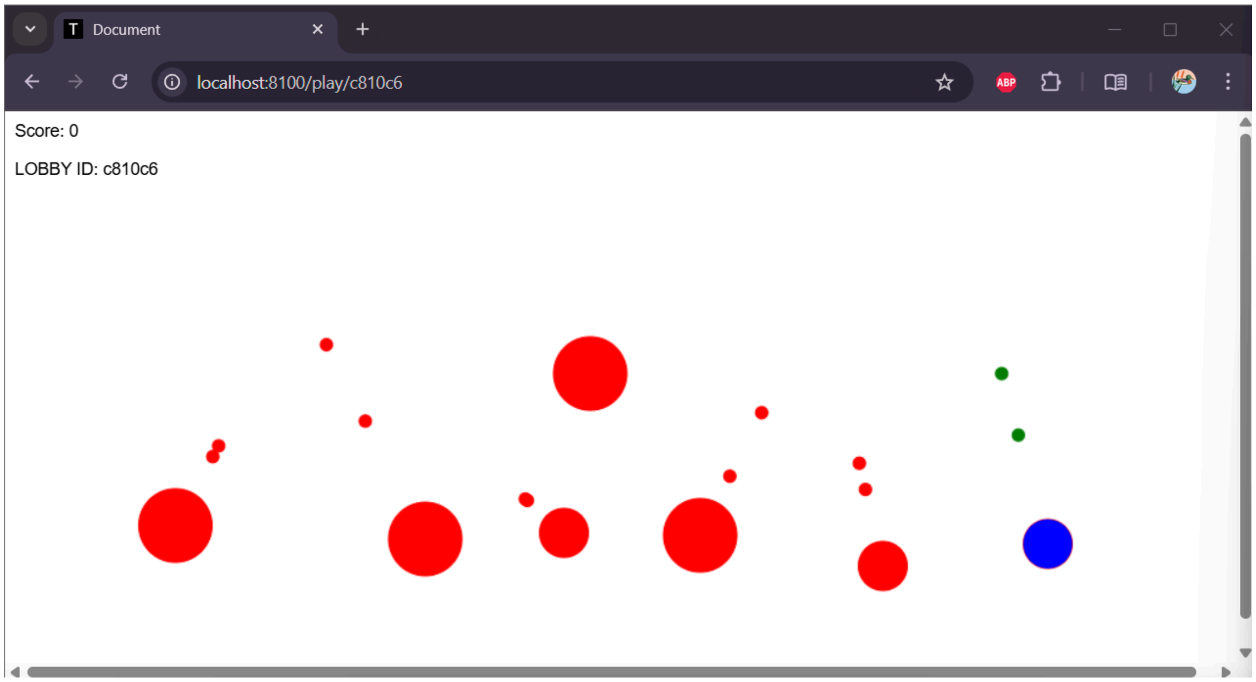

Figure 1. One players view of 8 other players in a single lobby

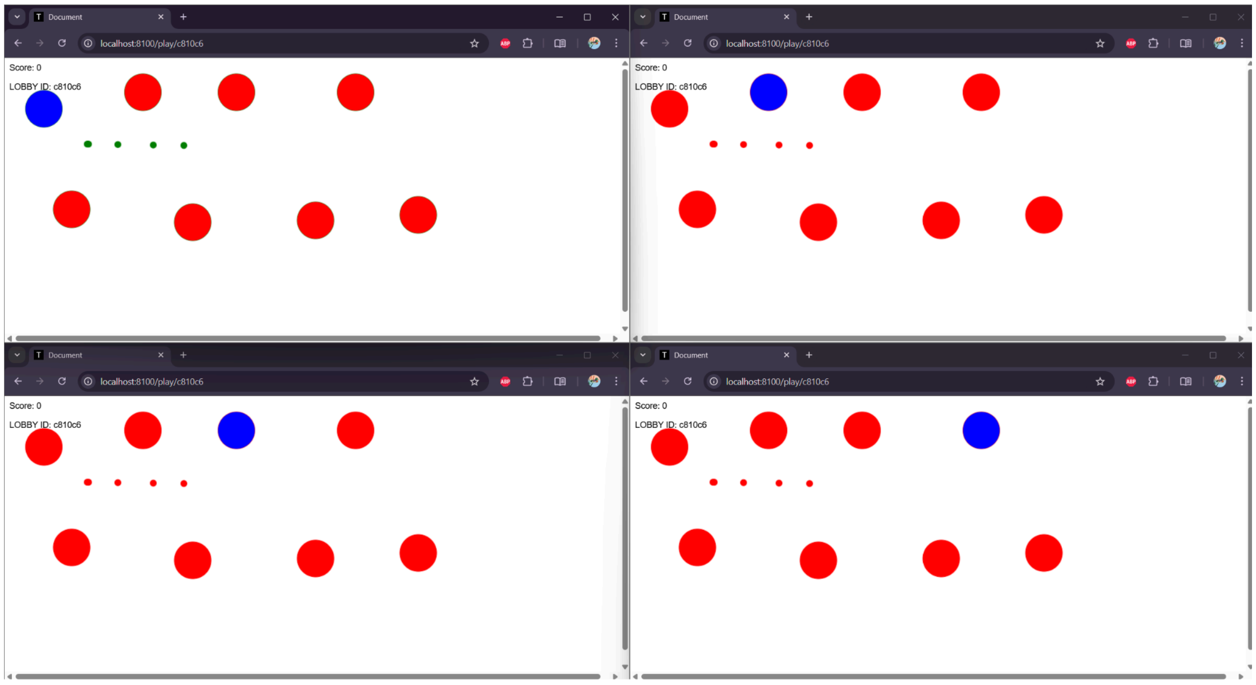

Figure 2. 4 players viewing an 8 player lobby