Aiden Hamade and Brock Kessinger

CS 371 Project Writeup

17 November 2023

Background

The goal of this project was to implement client/server interactions on a game on classic Pong. The client code would contain a running instance of a pong game, and it was the server's job to communicate to each client what the other client's game state was. The server is also responsible for making sure that the client games are synced up while they are running concurrently.

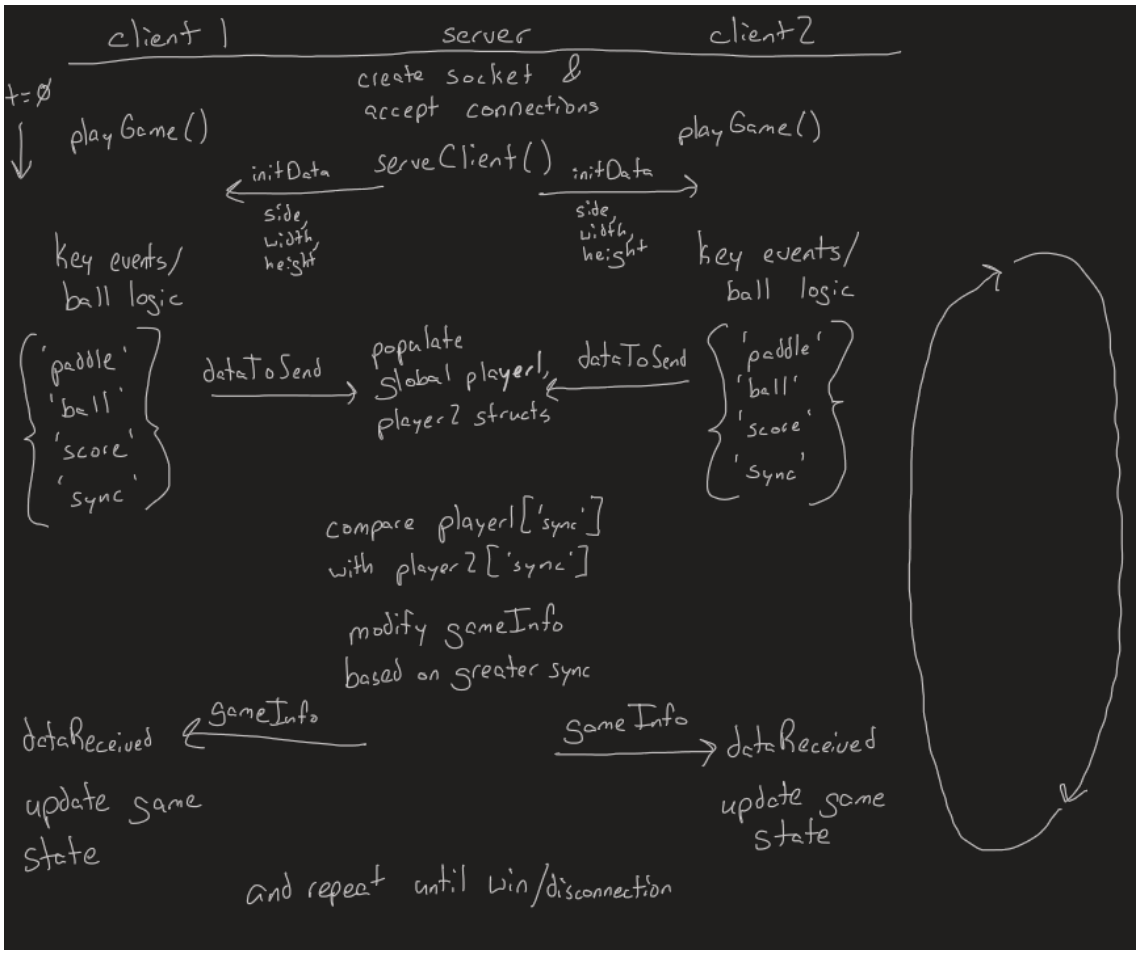
Design

We planned to have two files to handle communication between clients and the server. One file, titled pongServer.py is responsible for first creating the server and allowing clients to connect, then sending initial data to the clients so that they can load their game. Once the clients load into the game, the server is then responsible for receiving updates from clients and sending the most updated information to both clients. We planned that the server would start a thread for each client upon connection; the thread would then run a function with an infinite loop with arguments for the client socket and a boolean. The boolean was used so that each client's thread on the server would know which player's information it is carrying. All of the syncing would be done in this loop after the server has received data from each client.

On the client side, the client first attempts to connect to the server, then it receives initial data and runs the playGame function. During playGame, it enters an infinite loop where it is responsible for receiving game state information from the server and updating its game state as

well as sending information about its game state to the server. It sends its game state to the server at the end of the loop. We utilized json strings to send and receive the game state dictionary.

Implementation



Challenges

After we got our clients and server working on localhost, we attempted to connect clients from separate machines to the same game. We started a server and client on one laptop and another client on a different laptop. We could get the client that was on the same machine as the running server to connect, but we could not manage to get the separate client to join. After about a long time of debugging and trying lots of different things, Aiden and I connected to the campus

vpn and tried. The connection immediately worked, and we realized something on our house wifi was probably throwing the connection off.

Lessons Learned

Before working on this project, we had no idea how to establish a connection between two different computers. After getting the pong game to work from two clients on the same computer, we struggled with connecting clients from different computers. We had to spend a lot of time learning about the socket library in our efforts to debug.

Additionally, we had heard of json strings in the past, but didn't have any experience actually using them. We also didn't really understand their purpose, but we ended up using the json library in our code and it actually makes a lot of sense now. We also looked into the pickle library, but it seemed like a more complicated attempt at what json does, which is why we moved forward with json.

Known Bugs

One bug that we experienced while testing on the campus vpn was that the client on a different machine from the server moves slower than the client that is on the same machine. We experimented with changing the buffer size when receiving a transmission, but this did not change much. We concluded this behavior is expected with running one client on the same machine as the server and the other not.

Conclusions

The pong project was excellent hands-on experience with networking. Before this project, we thought that connecting two devices was a highly complicated process, but it seems as if it's actually quite simple. The most difficult aspect of the project was ensuring that the clients were seeing the same thing, whereas establishing the connection was really only a few

lines of code. GitHub also made this project more interesting because this has felt like a genuine team coding experience. Creating a markdown file to explain how to run the code and thorough commenting are honestly different expectations than we've had on a lot of other coding assignments in our undergrad, so we appreciate this project a lot.