Lab 3: Report on managing game states

This report describes the basics of how I implemented game state management primarily within the client side. Additionally, the server side is used to relay messages. I faced a few difficulties throughout this lab and found this to work out which is why I was able to use both sides.

Client-Side:

- The basic game states such as orientation and ship placements are done in app.js. Starting from the function startSinglePlayer() on line 122, all the basics from Lab 01 are done here. Additionally, single-player moves are all done on the client side.
- The game progress and mode are controlled here as well. This is seen throughout the function startMultiPlayer(), this is where the game states are managed through both. With regards to the client side, messages are received on the client side. This is seen with the ws.on functionality being applied for player connection, checking players, shots, etc.

Server-Side:

- The main aspects of the messages are relayed here as seen in server.js file. Here sessions are managed as shown starting line 19.
- Actions are synchronized so that the actions between the two players are coordinated. This is seen with the various replies to the actions in app.js such as checking players, firing shots, and players connected.

Finally, the lab was challenging to work with the client side and sockets in the server. I believe working more on the server side and enhancing it to perform more of the function of the client-side would further secure the game. As far as this lab goes, I worked on the various aspects of managing game states mainly in the client side.