Our server is authoritative with clients whose role is sending input and rendering graphics using info sent by the sever. An initial packet is sent to the server from each client with player ID then the server sends both clients back an init packet with the snakes starting location for both the player and their opponent as well as the location of the food. The server then sends update messages to both clients every logic update (~500ms). These updates include both snakes’ positions the position of the food. Every time one of the clients makes a move this info is sent to the server. We implemented artificial latency using queues on both send and receive with a random number (normalized) and calling a continuous loops which decrements and sends the message when it is <= 0. This caused stutter and synchronization issues where both clients where at different points in the game. We fixed this by implementing a predictive algorithm on the client side, as well as balancing the frame rate with the latency.

Milestone 1

* Kinsey made the client run a game and only use the sever connection for score.
* Kevin made server game logic.
* Alex made server connection logic and helped debug client connection logic with websocket.
* Andres designed initial client/server communication and initial scoring packets.

Milestone 2

* Kinsey made the Client stop running game logic and instead only send input to the server and render the games graphics based on info connected to the server.
* Alex and Kevin working together using paired programming worked together to couple our server and game code together and to make it accept Kinsey’s input sent from the client.
* Andres designed the info sent between server/client to minimalize bandwidth consumption

Milestone 3

* Kinsey made the client send the server timestamps and receive time stamps and then calculated the average latency and displayed it on the screen.
* Kevin and Alex paired programming made the artificial latency and time stamps.
* Andres debugged server code and designed timestamp packet communication

Milestone 4

* Alex and Kevin did paired programming on the server and fixed a couple of bugs that had went unnoticed for the previous milestones.
* Kinsey attempted writing a predictive algorithm for the snake game but It made the game crash and we were unable to figure out why so that algorithm was removed and reintroduced later.
* Andres attempted to debug the issue in client but also couldn't figure it out

Milestone 5

* Alex wrote the report and cleaned up the server code a little.
* Kinsey made the client use prediction to reduce the effects of lag.
* Andres debugged client-side prediction issues
* Kevin helped debug the client