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. After this updates are sent to to both clients around half a second apart but this time varies slightly. These updates include both snakes’ positions the position of the food and a time stamp. Every time one of the clients makes a move this info is sent to the server. We implemented artificial latency user a que with a random number and calling a continuous loops decrementing the number when we hit zero a message in the que is sent or received depending on if it’s from the server or client and a new random number is generated. The effects we saw from this was lots of stutter and synchronization issues where both clients where at different points in the game. We fixed this by

Member participation by milestone

Milestone 1

* Kinsey made the Clinet run a game and only use the sever connection for score.
* Kevin made server game logic.
* Alex made server connection logic and helped kisney make and debug client connection logic both using websocket.
* Andres wrote out what packets we should send between server and cient.

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 wrote out new packets to send.

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 told us how to make time stamps for the client.

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 for milestone 5.

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.