You can set the hp at here: Common/config.js

The main principle to make online multi-player game smooth is reduce the amount of communication to minimum.

When it comes to a real time game, we want to run the game logic itself on the server and the client.

This is due to the fact that the server needs to be sync all the client side at all times.

But the client also needs to run the game locally too.

Each frame on the server, input from the network will be processed and applied to players, and that change is sent to the other players at a fixed rate.

On the client, input will be collected and sent to the server, and positions can be updated while awaiting for the messages to come back from the server(Client prediction).

The approach we will be implementing works as follows:

Client presses the right key, client moves immediately right.

Server receives the input message and stores it for the next update.

The next server update, the player input is applied, moving him right on the server state.

All changes in state are sent to all clients

Client receives the message immediately setting clients positions

Client can smoothly correct mistakes in prediction from the first step.

**Game Server Set up**

At this moment, the server updates every 40ms.

One of the most important principle to send data with socket.io is to minify the data and send at a once.

I as you know the server-side speed will be changed when the user number increases.

The time will be longer when the user increases because it has many sockets.

Because of that, I updated the speed of movement dynamically.

And used broadcast way to send server-side data to all clients.

* Server-side

Server/global.js

Data structure to save Player, Mob, Bullet data.

Server/Component/Entity.js

Basic structure of components

Server/Component/Player.js

Player structure.

Server/Component/Bullet.js

Bullet structure

Server/Component/Mob.js

Mob structure: stone, cow, food

Common/config.js

Have basic constant values.

Server.js

Process incoming sockets and send sockets

**Client side set up**

On the client we also run multiple loops, once in every 20ms.

Clear canvas

Draw info/status

Handle input (sends input messages to server)

Update position (using client prediction)

Move the other clients based on the server position(interpolation)

Draw players on canvas

To use client prediction, I added just the same movement to client side.

And players’ positions are updated every 20ms.

So when user input, it acts immediately.

But the position of players in the client side will be different from the server side.

So I interpolate the position when data arrives from the server.

Both of them are implemented at a once.

Player’s position is updated at every 20ms.

And server data will be arrived every 40ms.

To make interpolate and make the movement smooth, I gave weight to these two elements.

0.2&0.8

And as the time passes, the difference between client-side and server-side will be reduced.

The linear interpolation between two elements are used.

Client/js/game.js