Plot4_flow

17 February 2024 17:30

app.py

Server starts at 8001
(python app.py)

Handler fn -> calls start function.
Start fn -> initialse a game, create join, watch key,
Calls play fn.
Play fn -> process moves, check if valid. Broadcast the data to all
Clients via receivemoves fn. Also check if game has any winner.

Handler fn -> calls join fn.
Join fn -> extracts join key, checks if valid, connects
with existing game. waits for Replay fn to get called in case previous
Move has been made. Calls play fn.
Replay fn -> sends all previous game moves to client.

Play fn -> Loops through message received on websocket.
process moves, check if valid. Broadcast the data to all
clients via receivemoves fn. Also check if game has any winner.

Handler fn -> Calls watch fn.
Join fn -> extracts watch key, checks if valid,
connects with existing game. waits for Replay fn to get called
in case previous moves has been made. Keeps the connection
open until its closed.
Replay fn -> sends all previous game moves to client.
The spectator will receive all further moves since the play fn
Broadcast the data to all clients which is received via received fn.
Sendmoves fn -> returns if the client is spectator.

main.js

Game page is loaded, websocket
Connection opens at client side.
First player starts new game.

Receivemoves fn -> adds join, watch link to dom element.
Player 1 send join and watch link to respective people.(through 3rd party).
Player 2 load page with join link, opens websocket
Connection. Calls init.
Init fn -> check url, sends data to server that player is joining.

Receivemoves fn -> play the received moves on the board or show error or win message.
Sendmoves fn -> Next player(Either player 1 or 2) sends moves to server.

Receivemoves fn -> plays the move on the board or show error
or win message.

Spectator loads page with watch link, opens websocket. Calls init fn.
Init fn -> checks url, send data to server that player is spectator.