Andor Attila Nás
HF3CJW
2017.12.08

Basic Workings of 5inaRow Project Front-End

- load_header_out.js (running behind every page on which the user is not logged in on):
    - displays the header, which contains the name of the application
- load_header_in.js (running behind every page on which the user is logged in on):
    - displays the header, which contains the name of the application and a button that, when clicked, logs the user out
    - clicking on the "LOG OUT" button will send a GET request containing the token in the authorization header
    - on successful logging out, the token and username are removed from local storage and the user is redirected to index.html
- register.js (running behind registration.html):
    - clicking on the "REGISTER" button will send a POST request containing the username and password entered by the user
    - on a successful registration, the user is redirected to the index.html
- login.js (running behind index.html):
    - clicking on the "LOG IN" button will send a POST request containing the username and password entered by the user
    - on a successful login, the token in the response along with the entered username is stored locally and the user is redirected to find_game.html
- find_game.js (running behind find_game.html):
    - clicking on the "FIND GAME" button will send a POST request with the saved token in the authorization header

- upon successfully finding a game, the game ID in the response is stored locally and the user to redirected to the game area: game_area.html
- also in find_game.html, the leaderboard is presented with leaderboard.js (running behind leaderboard.html):
  - sends a GET request with the token in the authorization header to retrieve scores, which are displayed

- game_play.js (runs behind game_area.html):
  - the game board is drawn on the screen
  - when it is the user's turn, clicking on the board will send a PUT request containing the coordinates of where the user clicked on the game board and the game ID along with the token in the authentication header; the following codes may be received in the JSON response if the request was successful:
    - 2: there is already a piece where the user clicked, so the user is alerted
    - 6: the user's move was recorded, so the page starts polling to know when the other player has made a move, during which GET requests are sent every 500 milliseconds; the following codes may be received in the JSON response if the request was successful (polling is stopped in all cases except 9):
      - 7: the game ended in a draw, so the user is redirected to game_draw.html
      - 8: the game ended and the user lost, so the user is redirected to game_lost.html

        - 9: there has been no change in the game's state, so nothing happens
        - 10: the other player made a move, so the user's board is update accordingly and the user is notified to make a move
    - 7: the game ended in a draw, so the user is redirected to game_draw.html
    - 8: the game ended and the user won, so the user is redirected to game_won.html
    - 9: it is not the user's turn to make a move, so the user is alerted