Authors: Adnan Al-Ramadan, Polo Song

Overview:

Our project aims to create a website for users to play Gomoku, also known as Five in a row . The website will work based on the gamelobby style mechanism. After login, the user will enter the lobby page. In that page, the user could choose either to create a game room that allows other players to join in or join others' game room by room id. One can also choose to play with AI. Besides the basic game functionality, our platform will record the statistical information of every player. So the players could check how many games they win or lose.

And the following list, is a summary of what we plan to implement in our game:

- Users will be required to register so they can play.
- Users will be able to play against AI, if they wish to.
- Users will be able to create rooms where they can play against each other.
- There will be a leaderboard displaying the users with the highest scores.
- When a game is played, the status of the game will be updated dynamically, depending on the move of the players.

Frontend:

There will be 3 web pages, login page, game lobby page, and game room page.

In the home page, there will be two forms, one for logging in and the other one for registration. If the player doesn't have an account, then they will be able to create a new one. Otherwise, they can log in, using their existing account, which will redirect them to the gamelobby page. And will assign session ID to them, so that they don't need to login again in every page on the website.

In the gamelobby page, there will be two buttons for creating or joining a room. If the join room button is clicked, then they will be able to view the currently available rooms on the server. Otherwise, if they choose to create a room, they will be prompted to enter the details of the room. Such as, the room name, and if they want the room to be private or not. If they choose for the room to be private, then they will be asked to enter a password for it.

In the game room page, there will be a chess board and a ready button. After both players have pressed the ready button, players can make moves on the chess board.

Backend:

We'll be using ajax, so that the users don't have to load a new html page, every time a change happens.

Also, we will be using the following npm modules:

- express: the server will be hosted using the express module.
- mongoose: because we are planning on storing data such as users, passwords and ranks on a mongodb database.
- multer: because it will allow us to POST and GET files, instead of just plain text.
- cookie-parser: because we don't what the users to have to login every time they refresh any page on the website.
- Body-parser: because it gives us access to the request body in a POST request.

Database Design:

```
User {
    Username: String
    Password: String // should be hashed and salted
    Status: String // “Free”, “In Room”, “In Game”
    Room: null or Room id
}
Room {
    JoinId: String
    Password: String
    PlayerBlack: {id: UserId, readyState: String}
    PlayerWhite: {id: UserId, readyState: String}
    CurrGame: GameId
}
Game{
    Next: String // Black or White
    ChessBoard: [{type: [{type: String}]}]
}
```

Routes:

1. createUser: creates a new user, and adds it to the database of the server.
2. Login: handles login requests, by searching for the user by name. And then it matches the entered password by the user, to the password stored in the database of the server.
3. A game lobby page: In this page, players will be able to choose if they want to join an existing lobby or create a new one.

4. GameLobbyCreation:page: In this page, players will be able to choose if they want to play against an AI or create a PVP game.
5. MakeAIGame: it allows players to make a game against an AI.
6. MakePVPGame: it allows players to make a game that other players can join.
7. The game page: After a player joins/creates a game lobby, this is the page where they will be playing the game.
8. Makemove: for when players make a move in the game.

Timeline:

1. We'll do the html pages.
2. Style the html pages with CSS.
3. Work on the server and client sides of code. Which will include the following:
   - Work on the login/create user functions.
   - Implementing all the functionalities in the game lobby page, so that they choose if they want to join an existing lobby or create a new one.
   - Implementing the game lobby creation functions, so that players can choose if they want to play against an AI or create a PVP game.
   - Make the functions that will handle the games that will be played against an AI.
   - Make the functions that will handle the games that will be played against another player.
   - Work on the functions that will display the state of the game.
   - Implement all Gomoku rules.
4. Tweaking and debugging the code.