CSC 667/867 Fall 2017
Term Project

Milestone 4: Application Logic

Game Name: Scrabble

Team Name: term-project-667-project-alavi-chalke-mazumdar-shah-vora

Table of Contents

Team Information

Team Name: alavi-chalke-mazumdar-shah-vora
Game Name: Scrabble
Team Members:

- Ajinkya Chalke
- Neharika Mazumdar
- Ali Alavi
- Rushabh Vora
- Risha Shah

Repository Link

https://github.com/sfsu-csc-667-fall-2017/term-project-667-project-alavi-chalke-mazumdar-shah-vora

User Actions

1. **Player logs in**
   Inputs: player
   Outcome: User successfully logged in and redirected to the Lobby page

2. **Player signs up**
   Inputs: new player
   Outcome: Account created for the user and redirected to the lobby page

3. **Player requests access after forgetting password**
   Inputs: player
   Outcome: Password successfully changed

4. **Player creates a game room**
   Inputs: player
   Outcome: New game room created, with the player assigned to that game room. The game is in a pending state. Note that this event will only occur from the application lobby.

5. **Player joins a game room**
   Inputs: player, game room

Outcome: Player added to the game room Note that this event will occur from within the application lobby.

6. **Player joins game room as a spectator**
   Inputs: player, game room
   Outcome: Player added to the game room as a spectator

7. **Player sends a message in game room**
   Inputs: player, game room, message
   Outcome: Message broadcast to all players in the room

8. **Player sends a message in lobby**
   Inputs: player, message, lobby
   Outcome: Message broadcast to all players in the lobby

9. **Player creates a word**
   Inputs: player, game room
   Outcome: word placed on the board and the user's score updated

10. **Player swaps tile(s)**
    Inputs: player, game room
    Outcome: player's rack updated

11. **Player passes turn**
    Inputs: player, game room
    Outcome: turn given to next user

12. **Player quits game**
    Inputs: player, game room
    Outcome: user deactivated from current game and their tiles returned to the tile bag

13. **Player updates account information**
    Inputs: player
    Outcome: player's information updated successfully

14. **Player logs out**
    Inputs: player
    Outcome: player logged out of system

System Events

1. **System receives invalid login credentials**
   Conditions: The user provides login information which is incorrect
   Outcome: User is denied access and an error message is displayed the user.

2. **Game room receives all required players**
   Conditions: Game room required player count is met after a player joins the room
   Outcome: Players are allocated rack of tiles and board is initialized

3. **Player given a turn**
   Conditions: The game state indicates that the player is the current player
   Outcome: UI updated if player is current to show play options. If player not current, UI updated to remove play options.

4. **Valid word formed**
   Conditions: Player forms a valid word on the board
   Outcome: Player score updated accordingly and player is given tiles from the tile bag, up to 7 (depending on tiles remaining)

5. **Invalid word formed**
   Conditions: Player forms an invalid word on the board
   Outcome: Player notifies about invalidity of the word formed and asked to play again

6. **One player uses up their tiles**
   Conditions: An event has occurred that changes the state of the player's rack, and the player has no tiles remaining in their rack and the tile bag is empty
   Outcome: Game ends, win/loss scores updated for all players in the game room

API Documentation

1. **Action**: Player logs in
   POST /login
   { username: string, password: string }

2. **Action**: Player enters information
   POST /register
   { username: string, email: string, password: string, name:string }

3. **Action**: Player request reset (forgot password)
   POST /forgotpassword
   { email: string }

4. **Action**: Player creates a game room
   POST /game
   { playerId: int }

5. **Action**: Player joins a game room
   POST /game/:gameId
   { playerId: int }

6. **Action**: Player sends a message in a game room
   POST /game/:gameId/message
   { playerId: int, message: string }

7. **Action**: Player sends a message in the lobby
   POST /lobby/message
   { playerId: int, message: string }

8. **Action**: Player joins a game room to spectate
   POST /game/:gameId
   { playerId: int , isSpectator: int }

9. **Action**: Player creates a word
   POST /game/:gameId/word
   { playerId:int, word: object }

10. **Action**: Player swaps tile(s)
    POST /game/:gameId/swap
    { playerId:int, tiles: string }

11. **Action**: Player passes a turn
    POST /game/:gameId/pass
    { playerId:int }

12. **Action**: Player quits
    POST /game/:gameId/quit
    { playerId:int }

13. **Action**: Update account
    POST /account/:userId
    { userId: int }

14. **Action**: Player logs out
    POST /logout
    { playerId: int }