

Term Game Project

ROULETTE

Milestone 4: Application Logic



Link to Github repository:

<https://github.com/sfsu-csc-667-fall-2017/term-project-laturkar-litwin-echevarria-pise-tait>

Created By:

NAME	SFSU ID	EMAIL
Aishwarya Laturkar	917927184	alaturkar@mail.sfsu.edu
Gayatri Pise	917922296	gpise@mail.sfsu.edu
Leire Litwin Echevarria	917969837	llitwinechevarria@mail.sfsu.edu
Sophie Tait	912710024	stait@mail.sfsu.edu

Table of contents

- 1) User Actions
- 2) Events
- 3) API

User Actions

Player logs into account

Inputs: username, password

Output: login successful and player enters Game Lobby

Player registers for account

Inputs: email, username, password

Output: registration successful and player enters Login page

Player forgets password

Inputs: email

Output: a link is sent to player's email and player enters new password on Password Reset page

Player resets password

Inputs: new password

Output: player enters Login page

Player joins a game room

Inputs: player, game room

Output: player assigned to game room

This event will only occur from Game Lobby

Player adds credit

Inputs: positive integer (min 1, max 1,000)

Output: amount added to player's balance and player enters Game Lobby

Player views Game Rules

Inputs: player

Output: rules displayed

Player sends a message in Game Lobby

Inputs: player, lobby, message

Output: message broadcast to all players in the Game Lobby

Player exits application

Inputs: player

Output: player logs out of application

Player places a bet

Inputs: player, game room, pocket, bet

Output: bet placed on a selected pocket

Player ends their turn

Inputs: player, game room

Output: check for events and process accordingly. The next player is given a turn

Player sends message in Game Room

Inputs: player, game room, message

Output: message broadcasts to all players in the Game Room

Player exits Game Room

Inputs: player

Output: player exits Game Room and enters Game Lobby

Events

Player enters incorrect username/password

Conditions: incorrect username/password entered

Outcome: message pop-up “Incorrect username/password entered”

Player successfully registers

Conditions: player successfully enters username and password

Outcome: player enters Login page and message pop-up “Registration successful. Please sign in again”

Player changes password

Conditions: player successfully changed password

Outcome: player enters Login page and message pop-up “Password changes successfully”

Game Room receives all players

Conditions: Game Room required player count is met after a player joins the game room

Outcome: first player is given a turn

Player is given a turn

Conditions: the game state indicates that the current player

Outcome: UI updated if player is current to show play options. If player is not current, UI updated to remove play options

Roulette displays winning number

Conditions: all players have placed their bets

Outcome: A message will be displayed with the winning number to all players and then their balances will be updated.

Game broadcasts place bets

Condition: Application logic to place bets

Outcome: Players can place bets during this window

Game broadcasts hold bets

Condition: Application logic to hold bets

Outcome: Players cannot place bets during this window

Player receives individual game score

Condition: Game finished calculating player score

Outcome: Players receives their scores as per the score calculation of the game.

API

Action: Player logs in to account

POST /player/:playerId/password
{ playerId: int, password: string }

Action: Player registers for account

POST /player/:email/username/password
{ email:string, username: string, password: string }

Action: Player forgets password

POST /player/:email
{ email: string }

Action: Player resets password

POST /player/:password
{ password: string }

Action: Player joins a game room

POST /game/:gameId
{ playerId: int }

Action: Player adds credit

POST /player/:balance
{ playerId: int, balance: int }

Action: Player views Game Rules

GET /gamerules
{ game rules: string }

Action: Player sends a message in the lobby

POST /lobby/message
{ playerId: int, message: string }

Action: Player places a bet

POST /GameBets/:playerId/gameId/bet
{ playerId: int, gameId: int, bet: int }

Action: Player ends their turn

POST /player/:playerId/next playerId/pocketId
{ playerId: int, nextPlayerId: int }

Action: Player sends a message in a Game Room

POST /player/:playerId/game/:gameId/message
{ playerId: int, message: string }

Action: Player exits Game Room

POST /player/:playerId
{ playerId: int }

Action: Player exits application

POST /player/:playerId
{ playerId: int }