ECE 421 Assignment 5 Part 2

Winter2019_Group4:

Nathan Klapstein (1449872)

Tony Qian (1396109)

Thomas Lorincz (1461567)

Zach Drever (1446384)

Table of Contents

server.rb **1**

app_model.rb **2**

app_presenter.rb **3**

main_menu_presenter.rb **4**

main_menu_view.rb **5**

game_board_presenter.rb **5**

game_board_view.rb **5**

server.rb

create_game(game_type, game_mode)

Pre:

- Game phase is MENU
- “game_type” is CONNECT_4 or TOOT_AND_OTTO
- “game_mode” is PLAYER_PLAYER, PLAYER_CPU, CPU_PLAYER, or CPU_CPU

Post:

- Game phase is IN_PROGRESS with selected game mode and type.
- A unique game ID is created to reference the game.

get_games()

Pre:

- None

Post:

- Existing gamesID’s with available spots are provided

get_games(username)

Pre:

- username has been used previously

Post:

- Existing gamesID’s where <username> is a player are returned

update_game(game_id, board_data)

Pre:

- Game phase is IN_PROGRESS

Post:

- Game data is updated with the provided board_data
- Updated game data is returned to the caller

load_game(game_id)

Pre:

- None

Post:

- Return game_data for provided game_id if it exists
- Return an error if no game_data corresponds to the provided game_id

post_move(game_id, state)

Pre:

- “state” conforms to app_model invariants

Post:

- The game state is saved in the server's database

get_league_standings()

Pre:

- None

Post:

- All information about previously played league games is returned as a JSON payload

---

app_model.rb

app_model Invariants:

- Game types: CONNECT_4 or TOOT_AND_OTTO
- Game modes: PLAYER_PLAYER, PLAYER_CPU, CPU_PLAYER, or CPU_CPU
- Player turn: 1 or 2
- Interface type: GUI or CLI
- Game phases: MENU, IN_PROGRESS, or GAME_OVER
- Game result: NO_RESULT_YET, PLAYER_1_WINS, PLAYER_2_WINS, or TIE
- Active token (for Toot and Otto): T_TOKEN or O_TOKEN
- abs(tokens(player_1) - tokens(player_2)) ≤ 1

update_turn(turn)

Pre:

- Game phase is IN_PROGRESS
- "turn" is 1 or 2

Post:

- "turn" has been swapped (1 => 2, 2 => 1)

update_game_type(type)

Pre:

- "type" is CONNECT_4 or TOOT_AND_OTTO
- Game phase is MENU

Post:

- The game type has been updated to "type"

update_game_mode(mode)

Pre:

- "mode" is PLAYER_PLAYER, PLAYER_CPU, CPU_PLAYER, CPU_CPU
- Game phase is MENU

Post:

- The game mode has been updated to "mode"

update_active_token(token)

Pre:

- The game type is TOOT_AND_OTTO
- “token” is T_TOKEN or O_TOKEN

Post:

- The active_token is set to “token”

place_token()

Pre:

- The game phase is IN_PROGRESS

Post:

- The players move has been applied to the app_model. The internal representation of the game board is updated based on the player’s move.

cpu_attempt()

Pre:

- It is the CPU’s turn

Post:

- If there is a winning token to play, the CPU plays it and wins the game. If not, cpu_prevent() is tried.

cpu_prevent()

Pre:

- cpu_attempt() is unsuccessful

Post:

- The CPU searches for a 3-chain that the other player has made. If one exists, a token is played to block the imminent 4-chain. Otherise, cpu_progress() is called.

cpu_progress()

Pre:

- cpu_prevent() is unsuccessful

Post:

- The CPU searches for its longest chain and adds to it. If there is no chain, a random move is played.

---

app_presenter.rb

update(signal, *data)

Pre:

- “signal” is “turn updated”, “game_phase_updated”, “game_type_updated”, or “game_mode_updated”

- *data is [model], [state], or [window, state]

Post:

- The appropriate method is called based on the signal

turn_updated(state)

Pre:

- “state” conforms to the app_model invariants
- Game phase is IN_PROGRESS

Post:

- The game board is redrawn with notifications for the next player turn

game_phase_updated(state)

Pre:

- “state” conforms to the app_model invariants

Post:

- The main menu view is drawn if game_phase is MENU
- The game board view is drawn if game_phase is IN_PROGRESS
- The game board view is drawn with winner notifications if game_phase is GAME_OVER

game_type_updated(state)

Pre:

- “state” conforms to the app_model invariants
- Game phase is MENU

Post:

- Main menu view is redrawn

game_mode_updated(state)

Pre:

- “state” conforms to the app_model invariants
- Game phase is MENU

Post:

- Main menu view is redrawn

---

main_menu_presenter.rb

update(signal, *data)

Pre:

- “signal” is “game_type_changed”, “game_mode”changed”, or “start_game”
- “data” is [state] or empty

Post:

- The appropriate app_model method is called based on the signal

---

main_menu_view.rb

draw(state)

Pre:

- “state” conforms to the app_model invariants

Post:

- The main menu is drawn
- Each button is clickable

---

game_board_presenter.rb

update(signal, *data)

Pre:

- “signal” is “column_clicked”, “main_menu_clicked”, “t_clicked”, “o_clicked”
- “data” is [column_index] or empty

Post:

- The appropriate app_model method is called based on the signal

---

game_board_view.rb

draw(state)

Pre:

- “state” conforms to the app_model invariants

Post:

- The game board is drawn
- All interactive elements are clickable