

ECE 421 Assignment 4 Part 2

Winter2019_Group4:

Nathan Klapstein (1449872)

Tony Qian (1396109)

Thomas Lorincz (1461567)

Zach Drever (1446384)

Table of Contents

App.model.rb	1
App.presenter.rb	1
ComputerOpponent.model.rb	1
MainMenu.presenter.rb	2
MainMenu.view.rb	2
GameBoard.presenter.rb	2
GameBoard.view.rb	3

App.model.rb

App.model Invariants:

- Game modes: "Connect 4" or "OTTO TOOT"
- Game types: "Player vs. Player", "Player vs. CPU", or "CPU vs. CPU"
- Player turn: 1 or 2

updateGameType(elementData)

Pre: Game is not started.

Post: The game type has been updated to the chosen type (either "connect 4" or "OTTO TOOT").

updateGameMode(elementData)

Pre: Game is not started.

Post: The game mode has been updated to the chosen mode (either "Player vs. Player", "Player vs. CPU", or "CPU vs. CPU").

applyPlayerMove(elementData)

Pre: The game state 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.

initiateNextTurn()

Pre: The game state is "In Progress".

Post: Update player turn from 1->2 or 2->1.

App.presenter.rb

updateMainMenu(state)

Pre: App.model state has been updated.

Post: Calls draw function in MainMenu.view.

updateGameBoard(state)

Pre: App.model state has been updated.

Post: Calls draw function in GameBoard.view.

ComputerOpponent.model.rb

ComputerOpponent.model Invariants:

- Strategy: "Random", "Thwart" or "Chain"
- Difficulty: "Easy", "Medium" or "Hard"

determineStrategy(state)

Pre: Game mode is Player vs. CPU or CPU vs. CPU.

Post: One of several strategies has been determined for playing the next move (random move, blocking another player's chain, adding to own chain, etc.)

emulateMistake(difficulty)

Pre: Strategy has been determined.

Post: Strategy has been randomly changed based on the difficulty level. For example, "easy" might have a 50% chance of changing the calculated strategy to a randomly selected strategy while "hard" might have a 0% chance of doing this.

setDifficulty(difficulty)

Pre: None

Post: The computer opponent's difficulty is set according to the input parameter.

applyStrategy()

Pre: Strategy for playing has been determined. A mistake has optionally been applied.

Post: A call to App.model.applyPlayerMove() has been made.

MainMenu.presenter.rb

onClick(elementData)

Pre: The MainMenu.presenter is listening for MainMenu.view "on click" events. An element in MainMenu.view has been clicked.

Post: A function of the App Model is called in order to update app data and state.

MainMenu.view.rb

draw()

Pre: None

Post: The main menu will be drawn. Each button is clickable.

onClick(element)

Pre: The MainMenu.view has been drawn.

Post: A signal is sent to the Main Menu Presenter with the payload being information about what element was clicked.

GameBoard.presenter.rb

onClick(elementData)

Pre: GameBoard.presenter is listening for GameBoard.view "on click" events. An element in GameBoard.view has been clicked.

Post: A function of the App Model is called in order to update app data and state.

GameBoard.view.rb

draw(state)

Pre: None

Post: GameBoard.view has been drawn.

onClick(element)

Pre: The GameBoard.view has been drawn.

Post: A signal is sent to the GameBoard.presenter with the payload being information about which element was clicked.