Sean Farrell

CPSC 2150 section 2 Re

12/15/2023

CPSC 2150 project 6 Report

Functional Requirements:

- As a player, I can choose what column I choose to place my marker in.
- As a player my token can be one of the following: {'X', 'O', 'H', 'L', 'W', 'C', 'V', 'E', 'M', 'Z'}
- As a player, I can correct what column I want to place my marker in if the chosen column is full.
- As a player, I can choose to exit the game when I don’t want to play any connect 4 games.
- As a player, I can choose to play again If I want to play another ConnectX game after a victory.
- As a player, I can choose to play again If I want to play another ConnectX game after a tie.
- As a player, I can see the board visually before my turn.
- As a player, I can see the board visually after my turn.
- As a player, I can view the completed board after a win condition occurs.
- As a player, I can view the completed board after a tie condition occurs.
- As either player, I can place the “numToWin” token consecutively vertical to win the game.
- As either player, I can place the “numToWin” token consecutively horizontal to win the game.
- As either player, I can place the “numToWin” token consecutively diagonal to win the game.
- As a player, if I fill the entire game board results in a tie.
- As a player, I can choose the number of players in the game.
- As a player, I correct the number of player in the game if the input was invalid.
- As a player, I can enter the amount of rows to be on the game board.
- As a player, I can enter the amount of rows to be on the game board again if the previous input was invalid.
- As a player, I can enter the amount of columns to be on the game board.
- As a player, I can enter the amount of columns to be on the game board again if the previous input was invalid.
- As a player, I can enter the amount of tokens needed to win.
- As a player, I can enter the amount of tokens needed to win again if I enter invalid input.
- As a player, I can have from 2 to 10 players inclusively.
- As a player, my piece must be placed at the bottom most available position in the selected column.
-

Nonfunctional requirements:

- The machine running must have java installed.
- The machine running must have IntelliJ installed for GUI to work.
- Program should be portable and be able to run on any system when unzipped.
- Player 1 must always go first as ‘X’.
- (0,0) is the bottom leftmost position of the gameboard.
- The number of tokens in a row must be between 3 and 20 inclusive.

- The number of rows must be between 3 and 20 inclusive.
- The number of columns must be between 3 and 20 inclusive.
- The number of players must be between 2 and 10 inclusive.
- The program decides to play a memory saving game if the area of the board is greater than 100.
- The program decides to play a fast game if the area of the board are less than 101 .
-

ConnectXController Class Diagram

ConnectXController
-curGame: IGameboard[1] -screen: ConnectXView[1] +MAX_PLAYERS: int[1] -numPLayers: int[1] -gameConidition: Boolean[1] -turns: int[1] -tokens char[10]
+ConnectXController(IgameBoard model, ConnectXView view, int np) +processButtionClick(int col): void -newGame(): void