ProgrammingTechnology Documentation

Naveed Abdullah Task 04 OXRMEB

oxrmeb@inf.elte.hu

Task:

Create a game, which is a variant of the well-known five-in-a-row game. The two players can play on a board consists of n x n fields. Players put their signs alternately (X and O) on the board. A sign can be put only onto a free field. The game ends, when the board is full, or a player won by having five adjacent signs in a row, column or diagonal. The program should show during the game who turns.

The trick in this variant is that if a player makes 3 adjacent signs (in a row, column or diagonal), then one of his signs is removed randomly (not necessary from this 3 signs). Similar happens, when the player makes 4 adjacent signs, but in this case two of his signs are removed. Implement this game, and let the board size be selectable (6x6, 10x10, 14x14).

The game should recognize if it is ended, and it has to show in a message box which player won (if the game is not ended with draw), and automatically begin a new game.

Analysis:

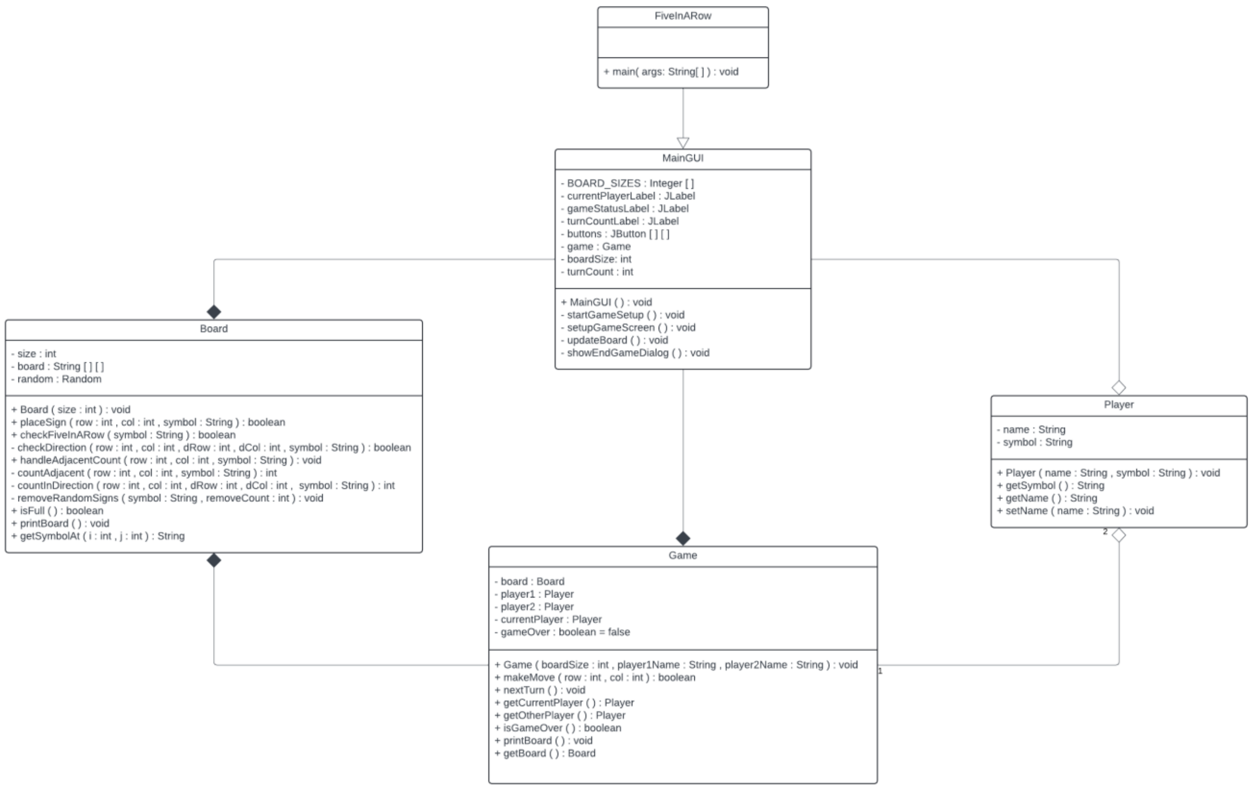
The "Five-In-A-Row" game is a two-player strategy game where each player takes turns placing their symbol ('X' or 'O') on a square board of either 6x6, 10x10, or 14x14 dimensions. The game’s objective is to achieve five consecutive symbols in any direction (horizontal, vertical, or diagonal) without any interruptions, winning the game. However, if a player aligns three or four symbols consecutively, they risk losing a random selection of one or two symbols, respectively, adding a layer of strategic complexity. This design keeps players from focusing solely on offense and requires careful planning to maintain their alignments.

The program features a graphical interface to display the board, indicate the current turn, and allow players to choose the board size. Upon winning, drawing, or filling the board, a message is displayed, and players have the option to start a new game or exit.

Plan:

- **FiveInARow (Main Class)**
  - Launches the application and initializes the main game GUI (MainGUI).
- **MainGUI (Graphical Interface)**
  - Handles user interface:
    - Prompts for board size (6x6, 10x10, 14x14) and player names.
    - Displays game board and updates moves.
    - Shows game status (turn info, win/draw).
    - Allows restarting or exiting after the game ends.
- **Game (Game Logic Controller)**
  - Controls game flow:
    - Initializes board and players.
    - Manages turns and tracks the current player.
    - Checks for win/draw conditions.
    - Applies rule to remove symbols if a player aligns three or four in a row.
- **Board (Game Board Representation)**
  - Manages board state:
    - Places symbols if cell is empty.
    - Checks five-in-a-row for winning condition.
    - Counts adjacent symbols and removes random symbols if three or four are aligned.
    - Checks if the board is full for a draw.
- **Player (Player Representation)**
  - Stores player attributes:
    - Holds player name and symbol (X or O).
    - Used by Game for turn switching and display purposes.

UML Class Diagram:


FiveInARow
+ main(args: String[ ]) : void
MainGUI
- BOARD_SIZES : Integer [ ]
- currentPlayerLabel : JLabel
- gameStatusLabel : JLabel
- turnCountLabel : JLabel
- buttons : JButton [ ] [ ]
- game : Game
- boardSize: int
- turnCount : int
+ MainGUI ( ) : void
- startGameSetup ( ) : void
- setupGameScreen ( ) : void
- updateBoard ( ) : void
- showEndGameDialog ( ) : void
Board
- size : int
- board : String [ ] [ ]
- random : Random
+ Board (size : int) : void
+ placeSign (row : int , col : int , symbol : String) : boolean
+ checkFiveInARow (symbol : String) : boolean
- checkDirection (row : int , col : int , dRow : int , dCol : int , symbol : String) : boolean
+ handleAdjacentCount (row : int , col : int , symbol : String) : void
- countAdjacent (row : int , col : int , symbol : String) : int
- countInDirection (row : int , col : int , dRow : int , dCol : int , symbol : String) : int
- removeRandomSigns (symbol : String , removeCount : int) : void
+ isFull ( ) : boolean
+ printBoard ( ) : void
+ getSymbolAt (i : int , j : int) : String
Player
- name : String
- symbol : String
+ Player (name : String , symbol : String) : void
+ getSymbol ( ) : String
+ getName ( ) : String
+ setName (name : String) : void
2
Game
- board : Board
- player1 : Player
- player2 : Player
- currentPlayer : Player
- gameOver : boolean = false
+ Game (boardSize : int , player1Name : String , player2Name : String) : void
+ makeMove (row : int , col : int) : boolean
+ nextTurn ( ) : void
+ getCurrentPlayer ( ) : Player
+ getOtherPlayer ( ) : Player
+ isGameOver ( ) : boolean
+ printBoard ( ) : void
+ getBoard ( ) : Board
1

Description of Methods

- FiveInARow
  - **main**
    Makes an instance of the GUI Class and sets its visible to true to start the game.
- Game
  - **makeMove**
    Invokes the move on the board on the specific tile of the board. It also handles every event that should be checked after the move is made.
  - **nextTurn**
    Changes the turn and moves to the next player.
- Board
  - **checkFiveInARow**
    Checks whether the game has ended or not by checking the five in a row property of any symbol in the game.
  - **checkDirection**
    Checks for consecutive five symbols along every positive direction along the board.

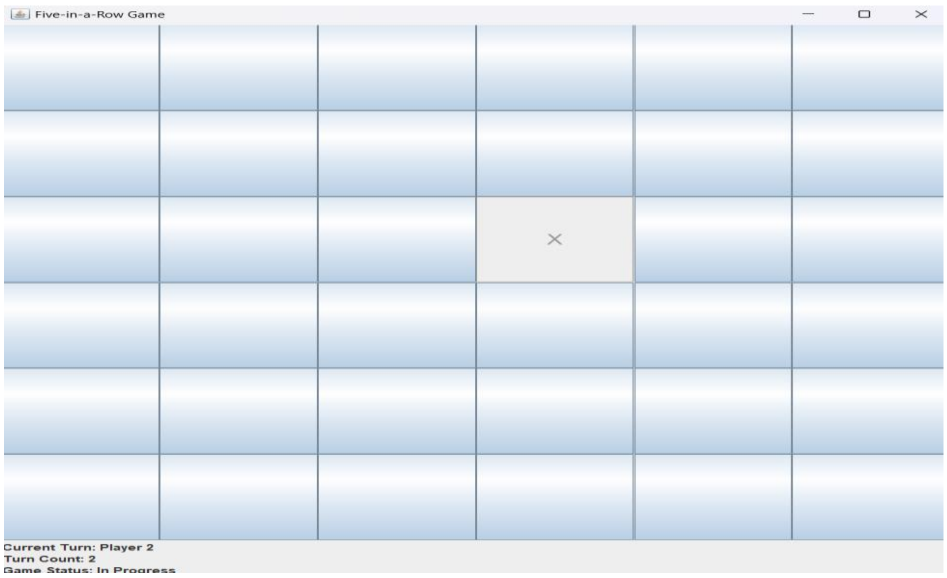
- ○ **handleAdjacentCount**
  Checks the special strategy of the game and invokes the functionality when the adjacent count is 3 and when it is 4.
- ○ **removeRandomSigns**
  Removes signs randomly from the player's move based on the number given as parameter.

- MainGUI
  - ○ **setupGameScreen**
    Sets up the game screen in the GUI by initializing the board, making the button, adding the action listeners, and making the game.
  - ○ **actionPerformed**
    Handles the logic of the player's move, changing the state of the board, moving to the next player.

White Box Testing:

1. Invalid Move Test:

AS a player
I WANT TO be not able to place a symbol in an occupied cell
GIVEN the cell is already occupied
WHEN I click on the occupied cell
THEN the game should reject the move

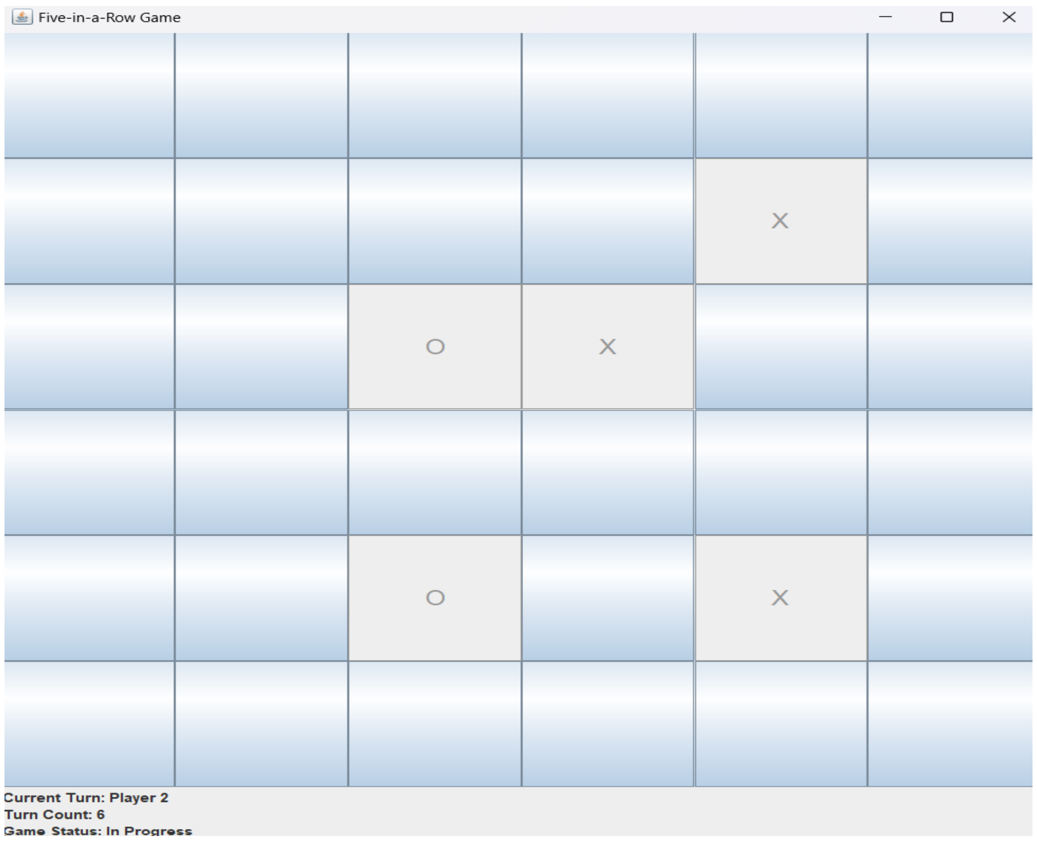


```
Current Turn: Player 2
Turn Count: 2
Game Status: In Progress
```

Here, the game does not move forward if I keep on placing the element on the same cell, it just remains here.

2. Valid Move Test:

AS a player
I WANT TO place a symbol on the board
GIVEN the selected cell is empty
WHEN I click on the cell
THEN my symbol should appear in that cell



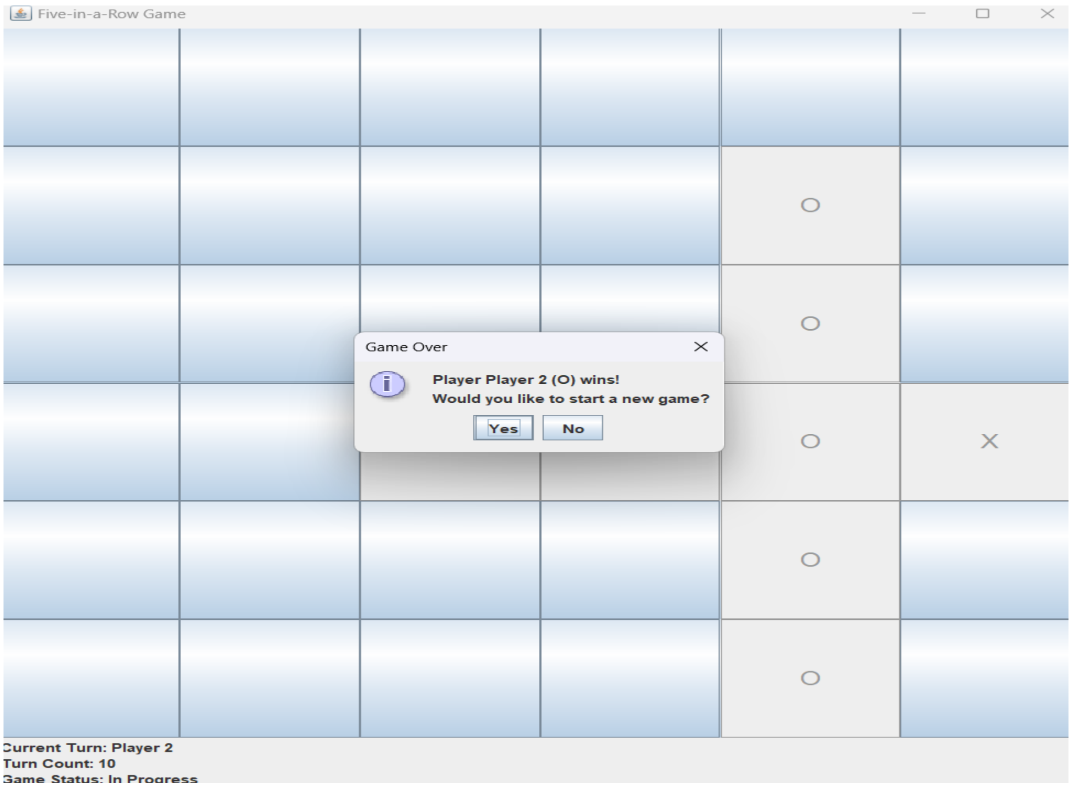
Current Turn: Player 2
Turn Count: 6
Game Status: In Progress

Now it is the sixth move and there are five signs on the Board which means that there are 5 signs correctly placed On the board.

3. Win Condition (Five in a Row):

AS a player
I WANT TO win the game by creating five symbols in a

row
GIVEN I have four of my symbols in a consecutive row, column or diagonal.
WHEN I place the fifth symbol in the sequence
THEN the game should declare me as the winner

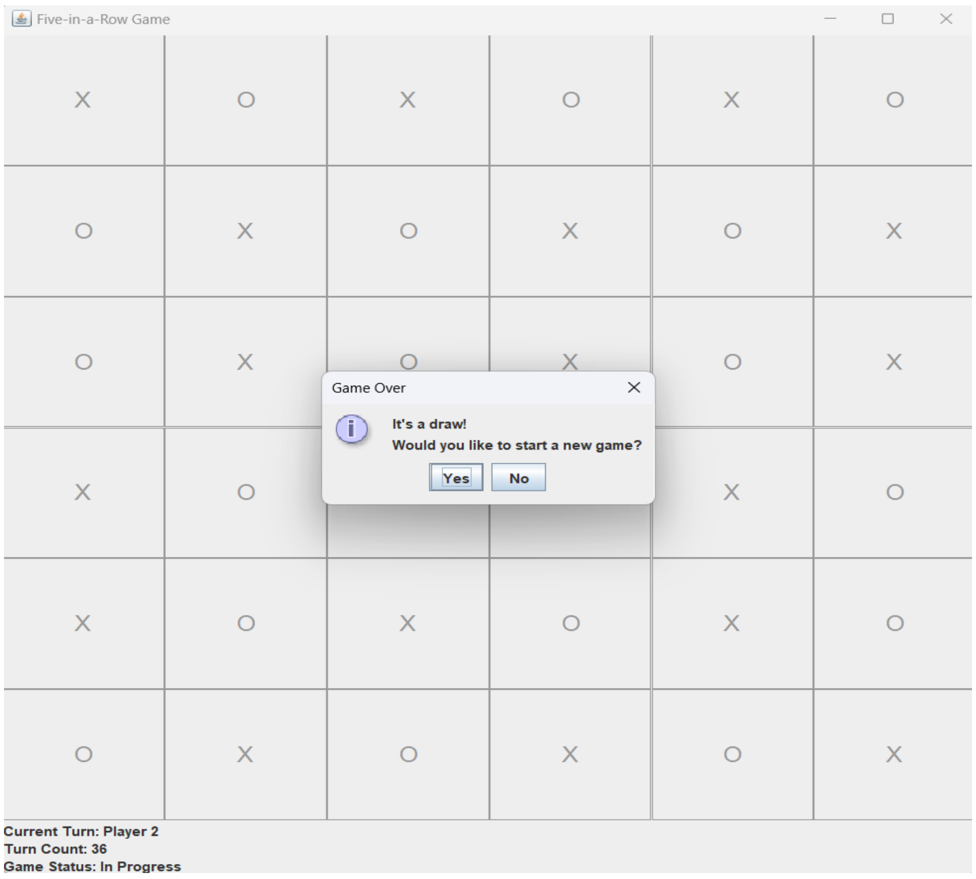


Player Player 2 (O) wins!
Would you like to start a new game?

Current Turn: Player 2
Turn Count: 10
Game Status: In Progress

4. Draw Condition:

AS the players
I WANT TO have the game end in a draw if the board is full without a winner
GIVEN all cells are occupied and no player has five symbols in a row
WHEN the last cell is filled
THEN the game a draw and end



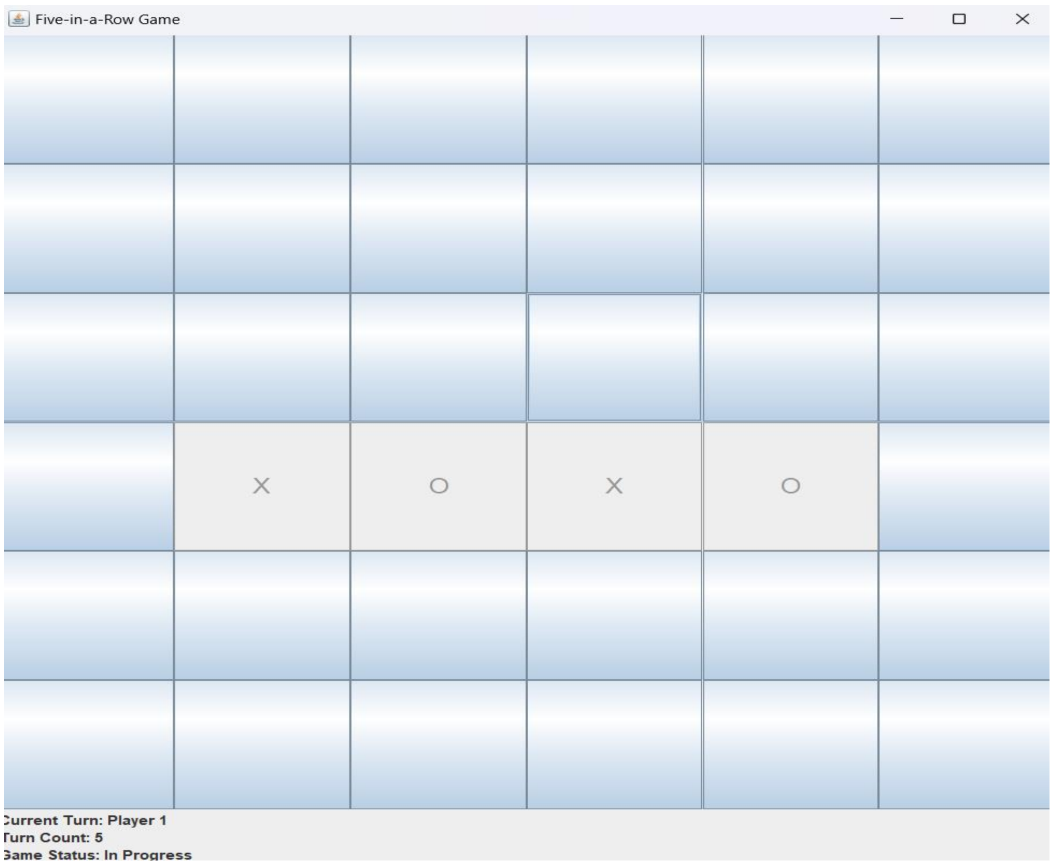
It's a draw!
Would you like to start a new game?

Current Turn: Player 2
Turn Count: 36
Game Status: In Progress

5. Turn Switching:

AS a player
I WANT TO switch turns after making a move
GIVEN it is my turn and I have placed my symbol on the board
WHEN my move is valid
THEN the turn switches to the other player, showing their name as the current player

Here, the player1 has the current move and since both players have played two moves and now the fifth move is underway and it shows that the turn switching is working.

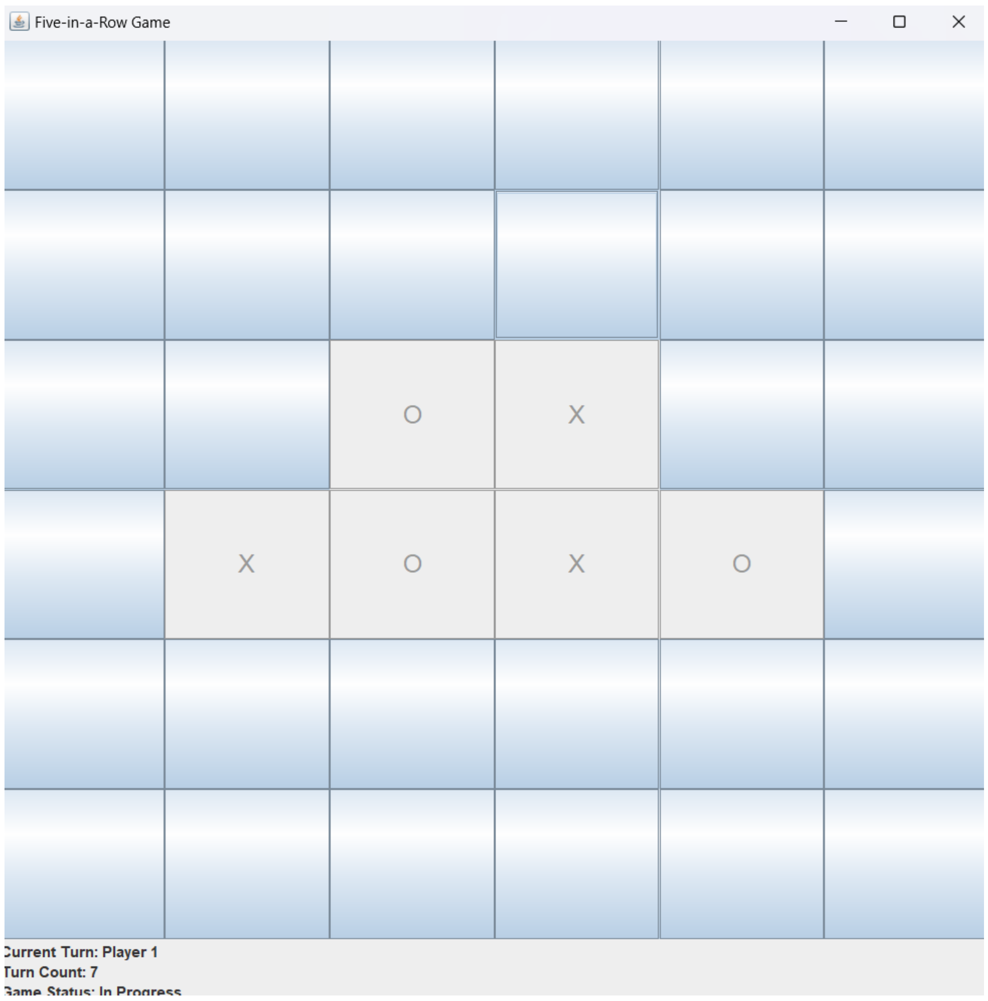


Current Turn: Player 1
Turn Count: 5
Game Status: In Progress

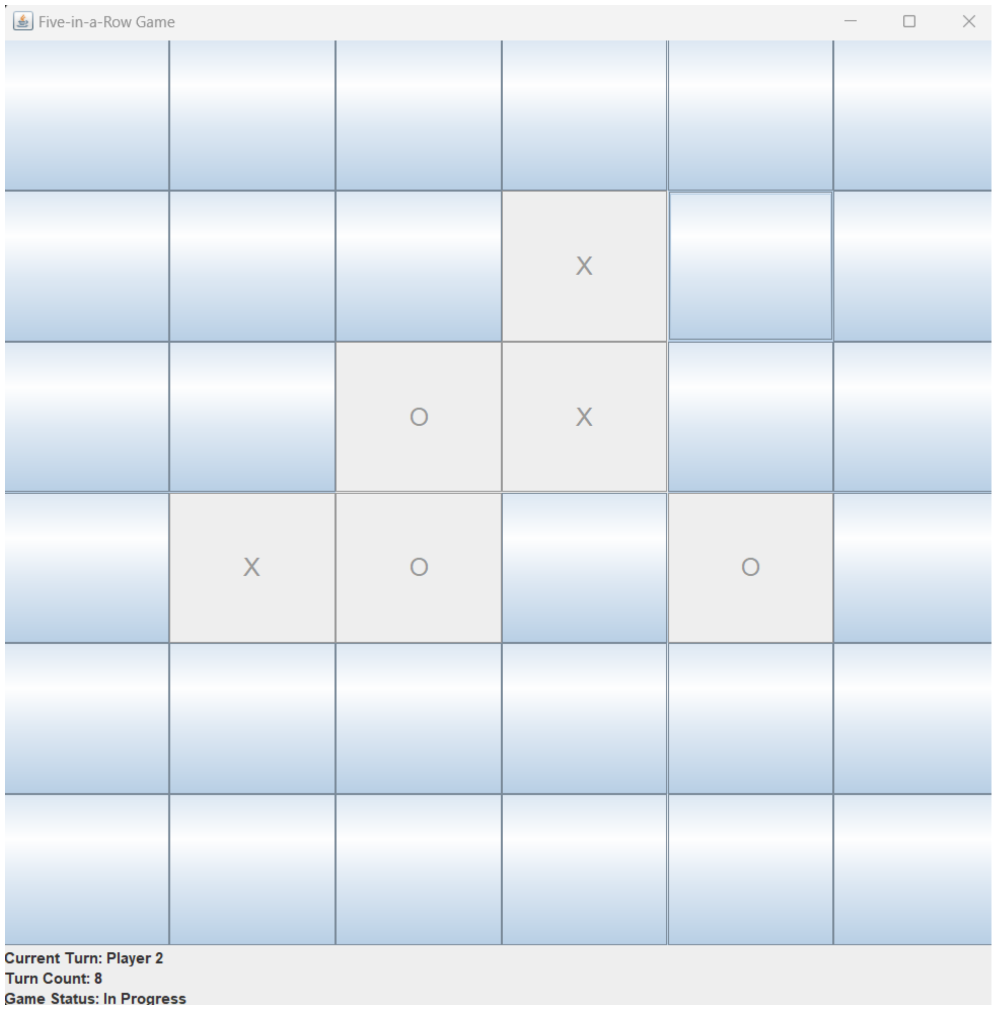
6. Symbol Removal on Three in Row:

AS a player
I WANT TO lose one of my symbols if I create exactly three consecutive symbols
GIVEN I have placed my symbol on the board creating exactly three consecutive symbols
WHEN my turns ends
THEN one of my symbols in that sequence should be removed randomly, and the game should proceed

After placing X for the third time, the number of X on the screen should have been 4 but instead they are 3 which shows that the symbol removal is working.



```
Current Turn: Player 1
Turn Count: 7
Game Status: In Progress
```

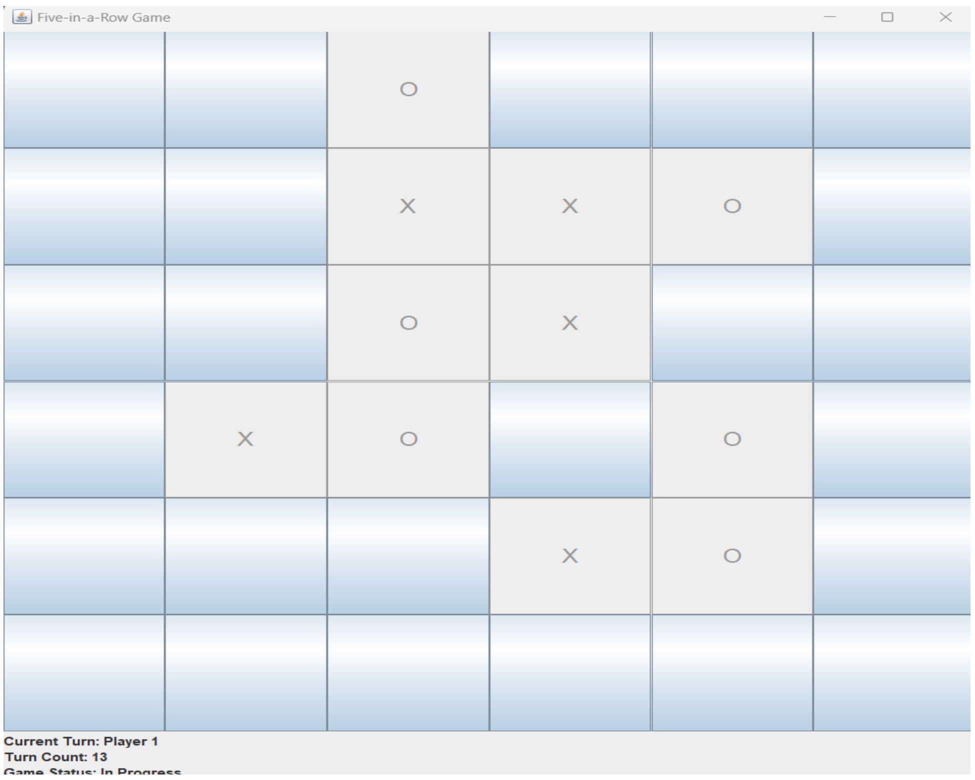


```
Current Turn: Player 2
Turn Count: 8
Game Status: In Progress
```

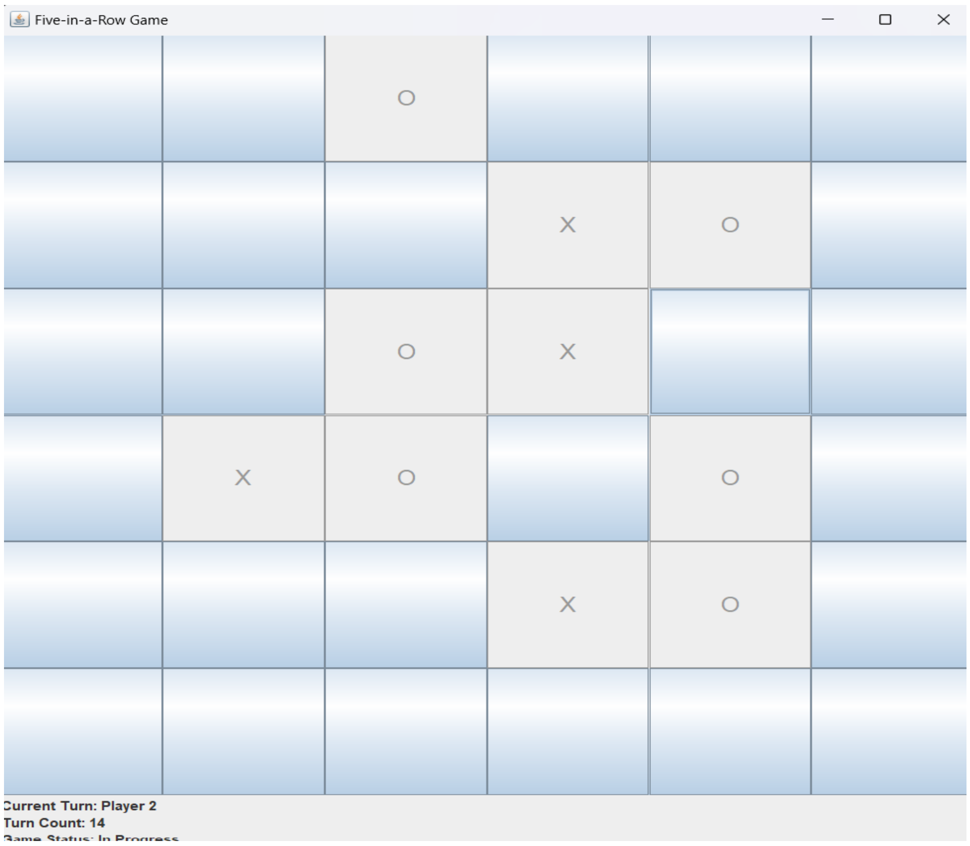
7. Symbol Removal on Four in Row:

AS a player
I WANT TO lose two of my symbols if I create exactly four consecutive symbols
GIVEN I have placed my symbol on the board creating exactly four consecutive symbols
WHEN my turns ends
THEN two of my symbols in that sequence should be removed randomly, and the game should proceed

After placing X for the third time, the number of X on the screen should have been 6 but instead they are 4 which shows that the two symbol removal is working.



```
Current Turn: Player 1
Turn Count: 13
Game Status: In Progress
```

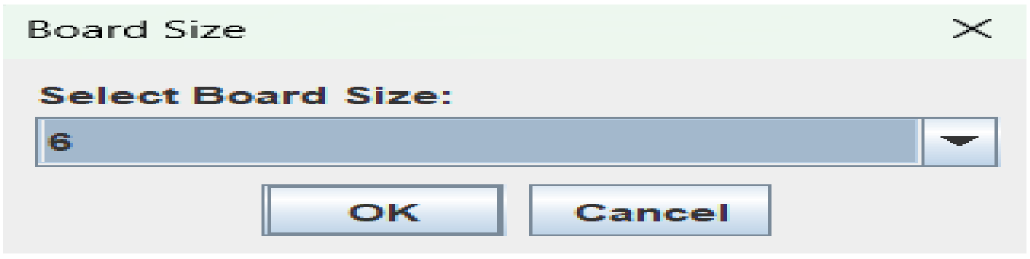


Current Turn: Player 2
Turn Count: 14
Game Status: In Progress

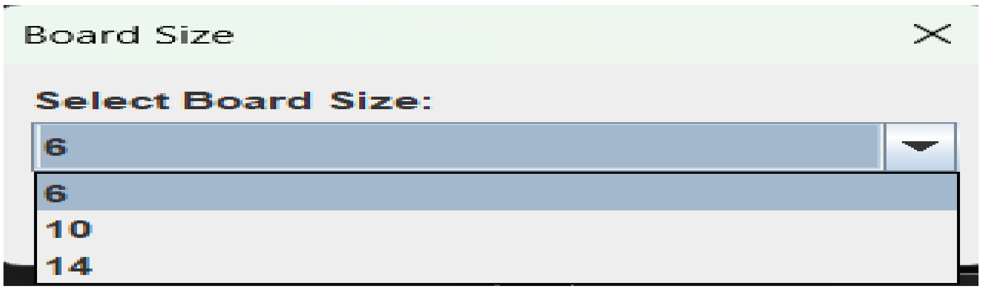
8. Board Selection:

AS a player
I WANT TO chose a board size when starting the game
GIVEN I am prompted to choose a board size at the start
WHEN I select to cancel the prompt
THEN the game should exit

Clicking on the arrow, I am getting a dropdown to select the board and clicking on cancel exits the program.



Select Board Size:
6

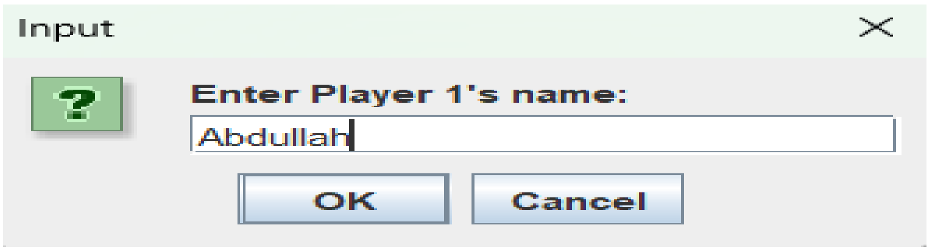

Board Size

Select Board Size:

6

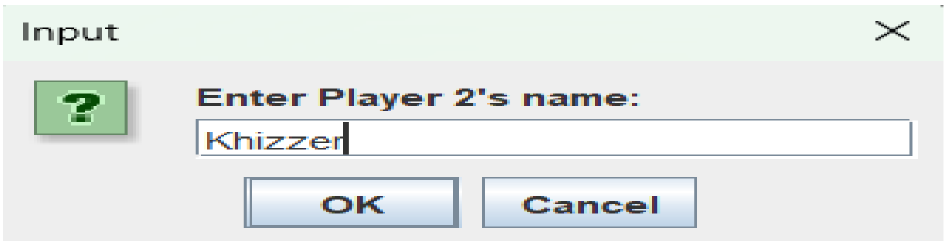
9. Naming the Players:

AS a player
I WANT TO write the names of the players
GIVEN I am prompted to enter the names of the players
WHEN I enter the names of the players
THEN the name of the players are displayed


Input

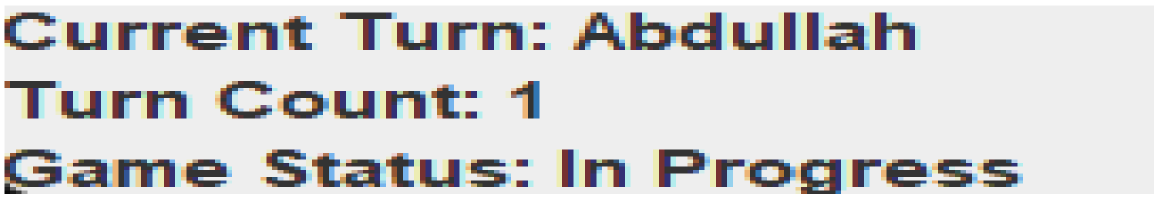
Enter Player 1's name:

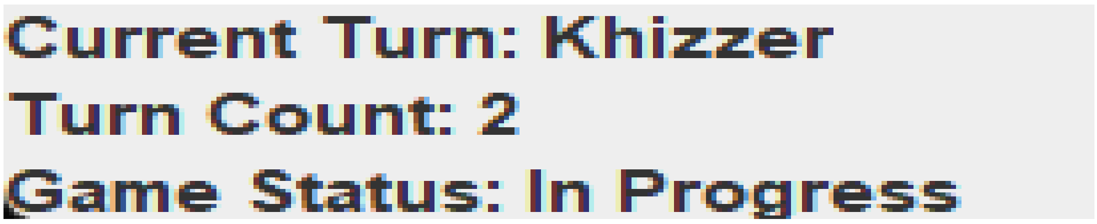
Abdullah


Input

Enter Player 2's name:

Khizzer


Current Turn: Abdullah
Turn Count: 1
Game Status: In Progress


Current Turn: Khizzer
Turn Count: 2
Game Status: In Progress

Black Box Testing:

1. Adjacent Count:
   AS a developer
   I WANT TO check the adjacent of the symbol placed on the board
   GIVEN the ***countAdjacent()*** method is called with a row and column and a symbol
   WHEN the symbol is placed on the board
   THEN the method should return the adjacent count of the symbol by checking all its directions.

2. Removing Random Signs:
   AS a developer
   I WANT TO check the whether random signs are removed from the board
   GIVEN the ***removeRandomSigns()*** method is called with a number of signs to remove and a symbol
   WHEN the condition of removing the signs in fulfilled
   THEN the method should remove the signs from the board according to the number given to it.

3. Full Board Check:

AS a developer
I WANT TO check whether the whole board is filled with symbols
GIVEN the ***isFull()*** method is called
WHEN the board is filled with symbols
THEN the method should return true that the board is full

4. Win Condition Check:

AS a developer
I WANT TO check whether the five in a row condition is fulfilled
GIVEN the ***checkFiveInARow()*** method is called with the symbol
WHEN the symbol is placed on the board
THEN the method should return true and the game won dialog should be shown

5. Start New Game:

AS a developer
I WANT TO check whether there is an option to start a new game
GIVEN the ***showEndDialog()*** method is called
WHEN the game is won by a player
THEN the method should show a dialog box asking the user to quit or start a new game

6. Move to the Next Player:

AS a developer
I WANT TO move to the next player after the turn is made
GIVEN the ***makeMove()*** method is called
WHEN the move is made by one player

THEN the method should change the move and pass it to the next player