Connect 4

Akhil Mitta, Riya Gunda, Lavan Aditya, Siddhant Joshi, Tia Burmi, Carmella Holloway

Introduction

There is an 8x8 vertical grid on which players take turns placing their pieces. The goal of the game is to have four of your game pieces connected in a row, column, or diagonally. When a player places a piece in a column, the piece will land in the bottom-most available row.

As the players play the game, they should be updated with how many pieces they have placed and the max number of connected game pieces they currently have (this will be less than four until the game is won).

Requirements

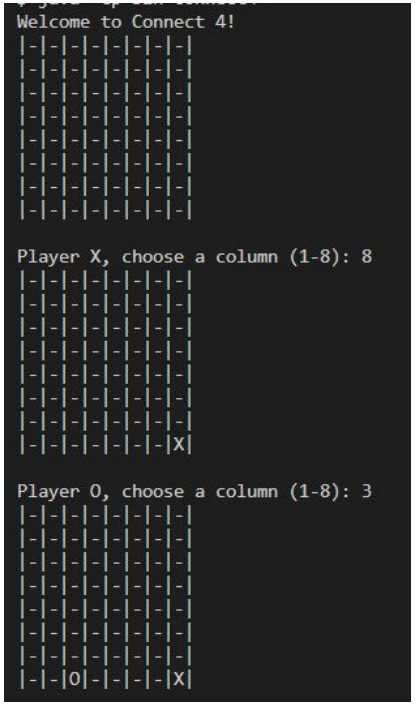
- The game shall be a two player game.
- The game shall provide clear instructions on how to play the game.
- The game shall have an 8x8 grid.
- The game shall display the current status of the board after each move.
- The game shall prompt each player to take turns entering their moves, and shall validate that the entered move is a valid column number on the board.
- The game shall detect when a player has won by connecting four pieces in a row, column, or diagonal.
- The game shall detect when a draw has occurred,
- The game shall allow the players to start a new game after a game has ended.

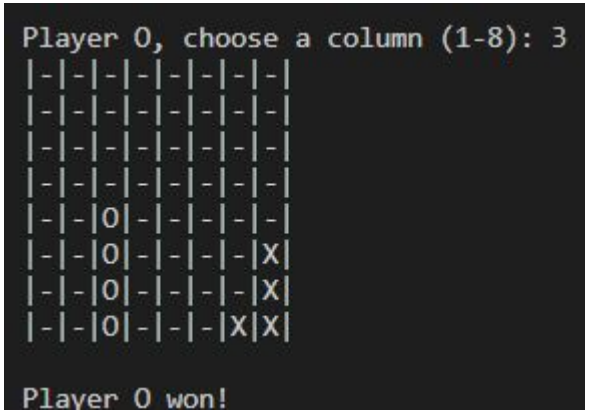
User Interface Examples

- 8 x 8 vertical square displayed
- Prompts player to place their piece at a column 1 - 8
  - Player 1 will have X pieces, Player 2 will have O pieces
- The player's piece is displayed in lowest available space
- After each placement, the board is updated with the placed piece and the other player's
- Upon victory, the winning player (Player 1 or Player 2) is displayed

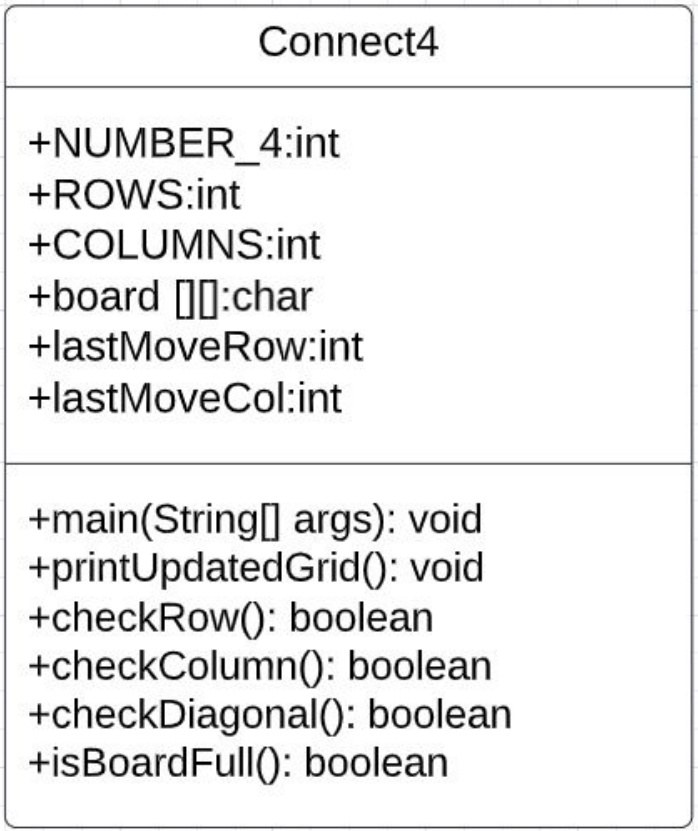
Grid display is shown with the following characters - _ , |

User Interface Examples continued…





UML Diagram


Connect4
+NUMBER_4:int
+ROWS:int
+COLUMNS:int
+board [][]:char
+lastMoveRow:int
+lastMoveCol:int
+main(String[] args): void
+printUpdatedGrid(): void
+checkRow(): boolean
+checkColumn(): boolean
+checkDiagonal(): boolean
+isBoardFull(): boolean

Constants

- private static final int ROWS = 8
- private static final int COLUMNS = 8;
- private static char[][] board = new char[ROWS][COLUMNS];
- private static int lastMoveRow = -1;
- private static int lastMoveCol = -1;

Design

- public static String printUpdatedGrid() → prints new board after every move
- public static boolean checkRow() → checks the row in which the most recent move was made to see if the game was won and returns a boolean.
- public static boolean checkColumn() → checks the column in which the most recent move was made to see if the game was won and returns a boolean.
- public static boolean checkDiagonal() → checks both the diagonals of the move that was recently made to see if the game was won and returns a boolean.
- public static boolean isBoardFull() method → This method checks if the board is full or has any more indices left to place game pieces on.

What we learned

- How to design a project using proper syntax and word choice
- How to collaborate with a team to create a cohesive project
- Implementation methods of a board printed out to the console
- How to problem-solve and utilize creative thinking within bounds that we created