Patrick Woodrum

Project 3 Extended Tic Tac Toe

CpSc 2150 Section 001

Extended Tic Tac Toe Report

Requirements Analysis

User Stories (Functional):

As a user, I should be able to:

-decide how many players will be playing the game (min:2,max:10) so that the game will be set up properly

-decide what character/letter I want to be represented by so that I have a team marker

-pick my row so that my token will be placed on that row

-pick my column so that my token will be placed on that column

-place my marker in the positions previously chosen

-view the entire board after each turn so that I can plan for next turn

-pick a new board position if the position I choose is out of range

-win the game and ask to play again

-end the game in a tie and ask to play again

-lose the game and ask to play again

Non-Functional:

As a system, it should be able to:

-This systems code was written in Java and must be able to be compiled and ran on Unix.

-The system will construct a board that is the size of the player's choice

-The player will choose the amount of rows, columns, and number in a row to win

-The system will run until either a player wins, or there is a tie, then will be prompted to play again or not.

-The system will continue to run even if the user inputs an invalid integer.

-The system will ask the user for a new input if input was invalid.

-The system reads in the inputs from the players, and adequately assigns their move to the correct row and column.

-The system will update the gameboard after each turn to properly display where the tokens are placed.

-The system will check to see if there is a winner Vertically by having whatthe user inputted in a row and column.

-The system will check to see if there is a winner Horizontally by having what the user inputted in a row and column.

-The system will check to see if there is a winner Diagonally by having whatthe user inputted in a row and column.

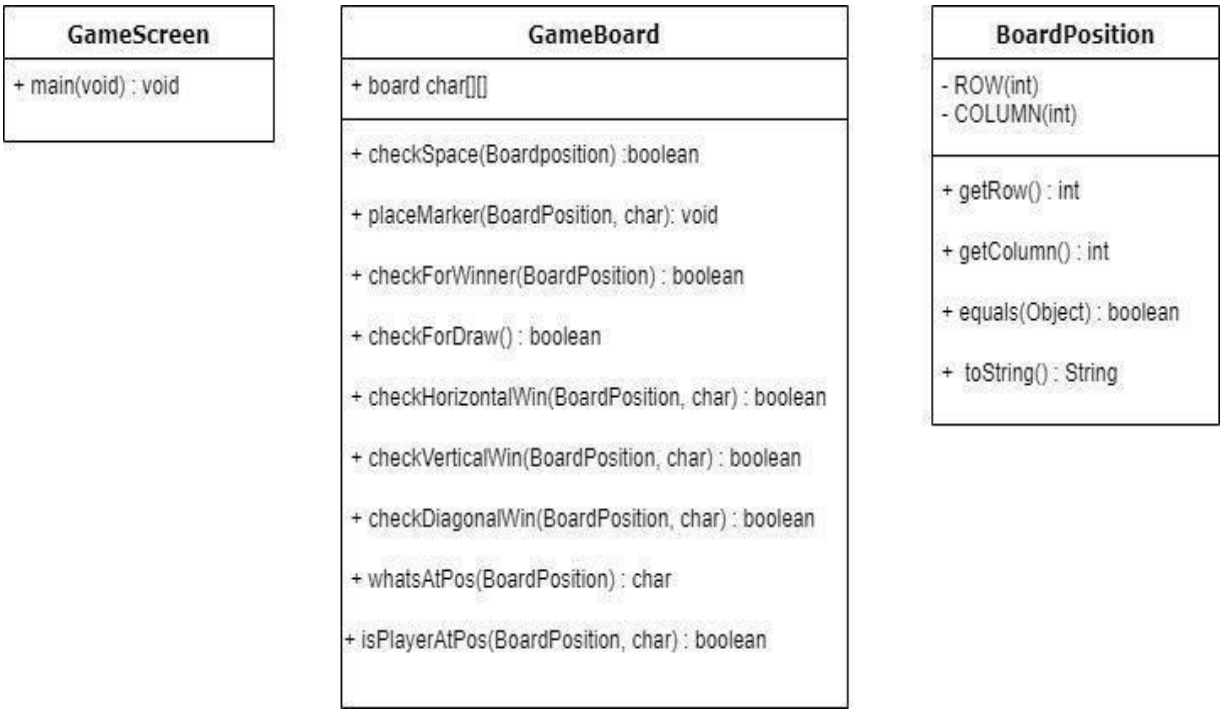
-The system will display a message saying who won, if there is a winner.

-The system will display a message if a tie occurs.

-The system will prompt the user to play again or not.

Design

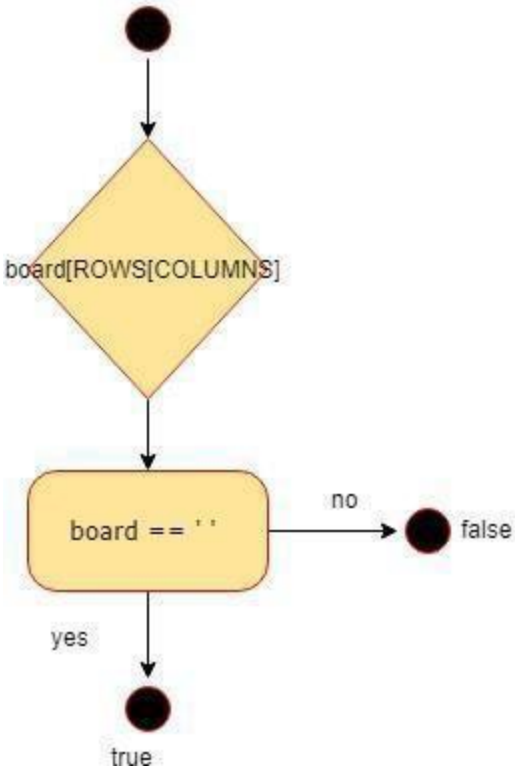
UML Class Diagrams:



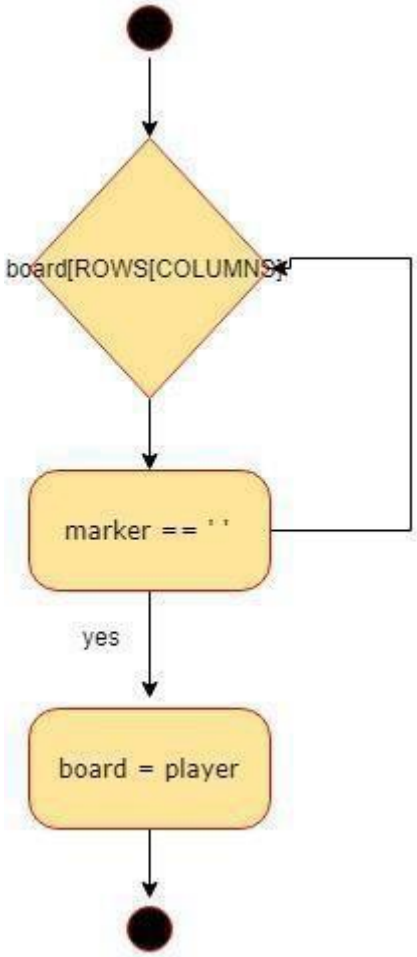
UML Activity Diagrams:

GameBoard.java

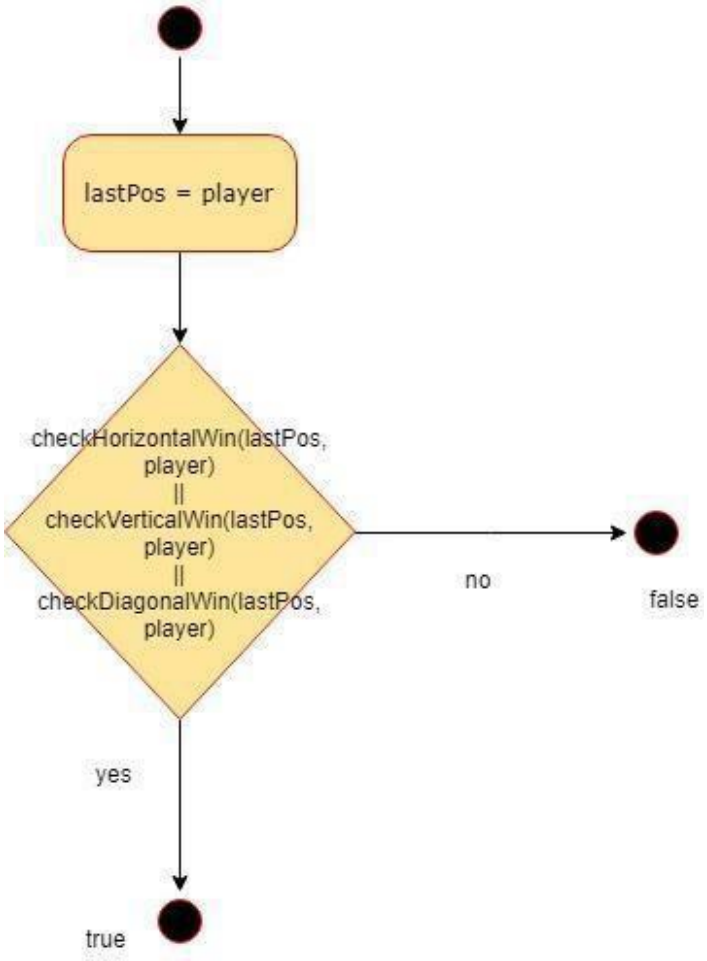
Public boolean checkSpace(BoardPosition pos)



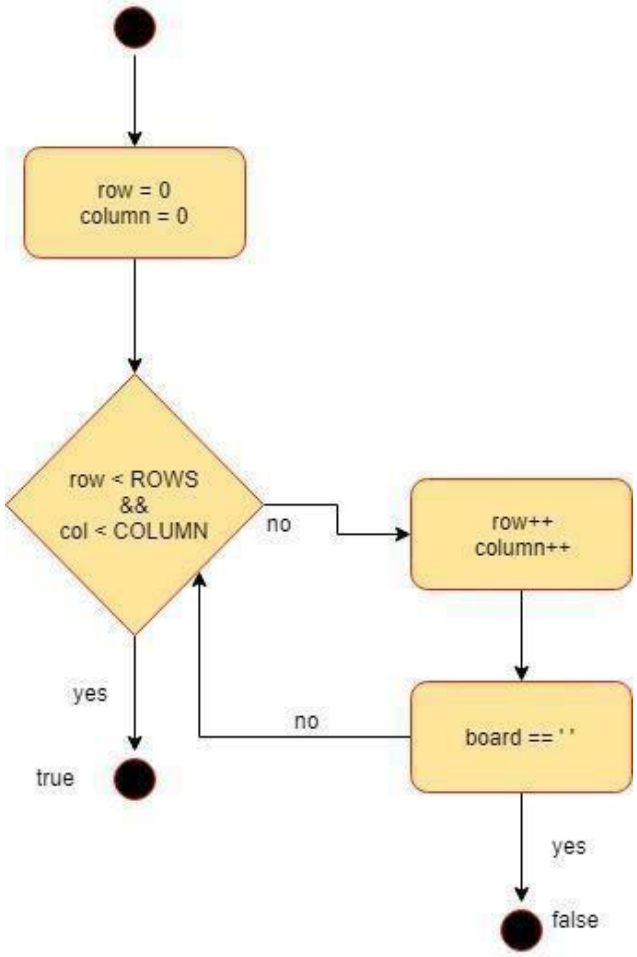
Public void placeMarker(BoardPosition marker, char player)



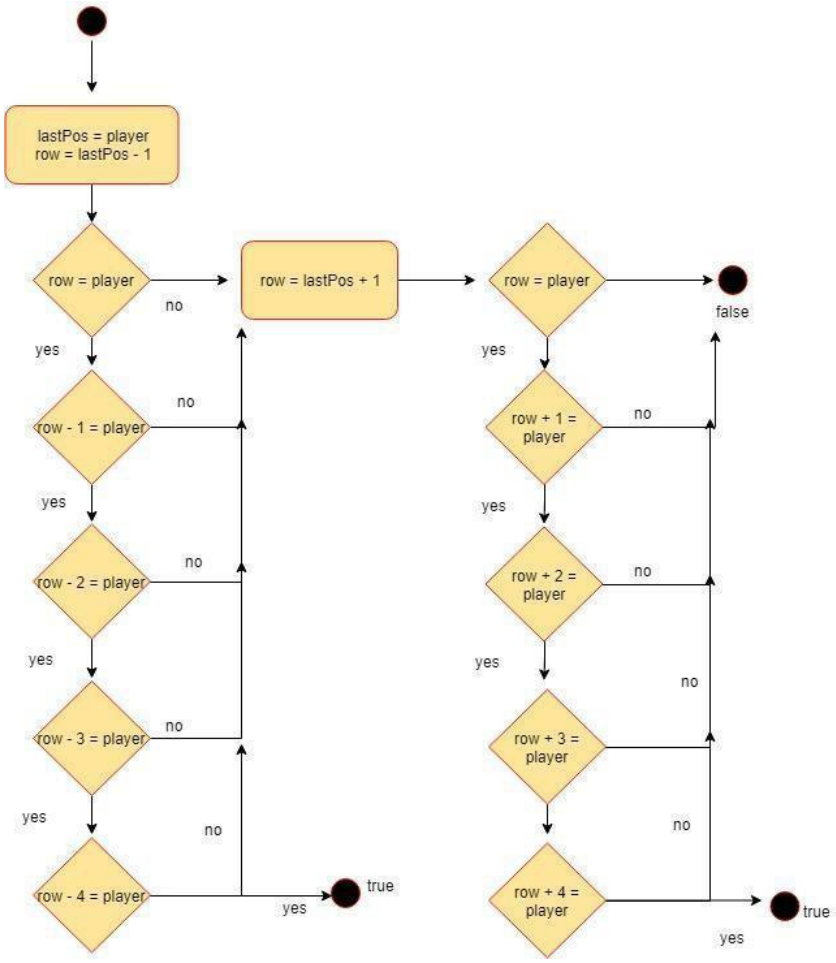
default boolean checkForWinner(BoardPosition lastPos)



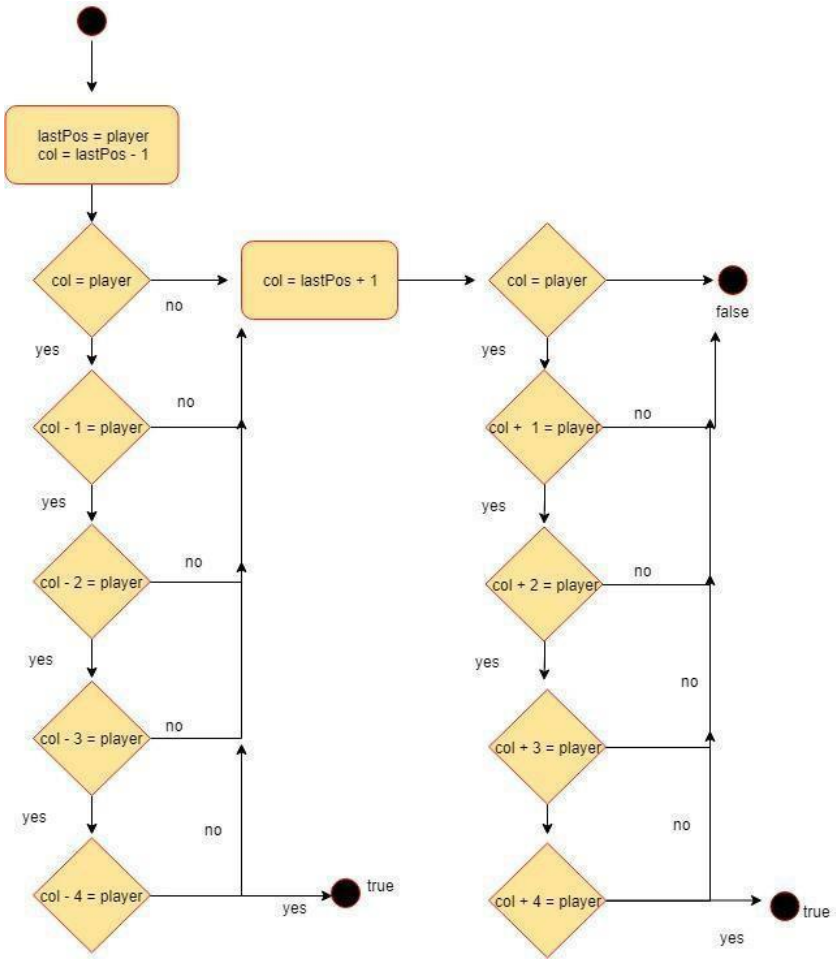
default boolean checkForDraw()



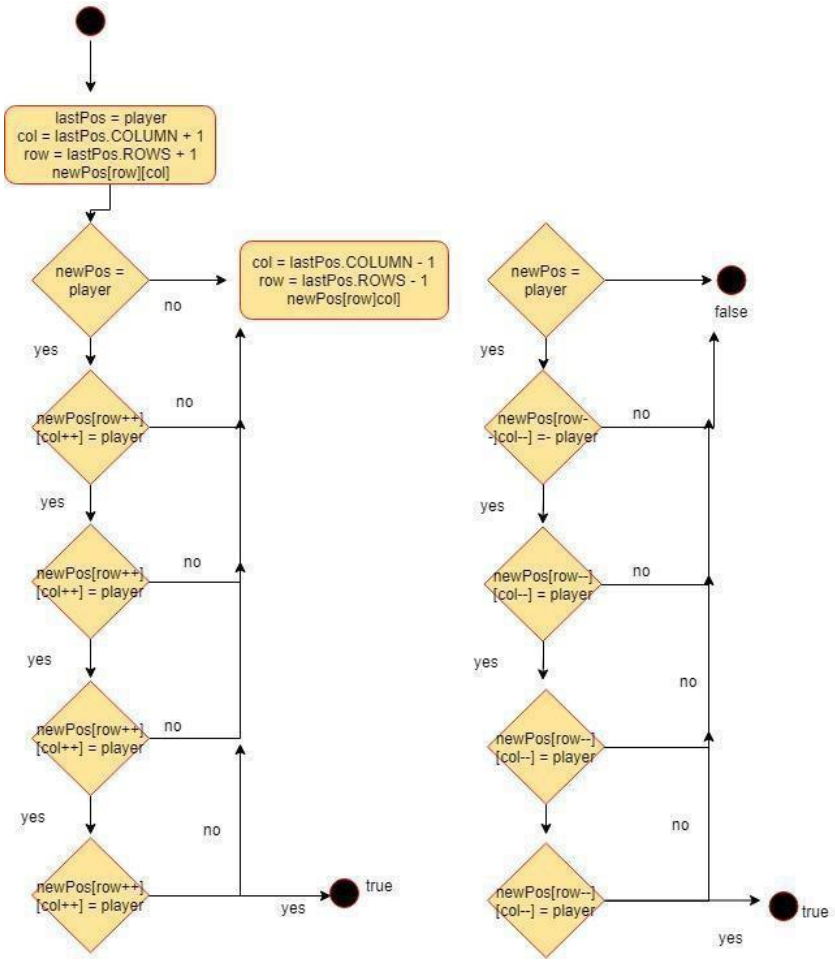
default boolean checkHorizontalWin(BoardPosition lastPos, char player)



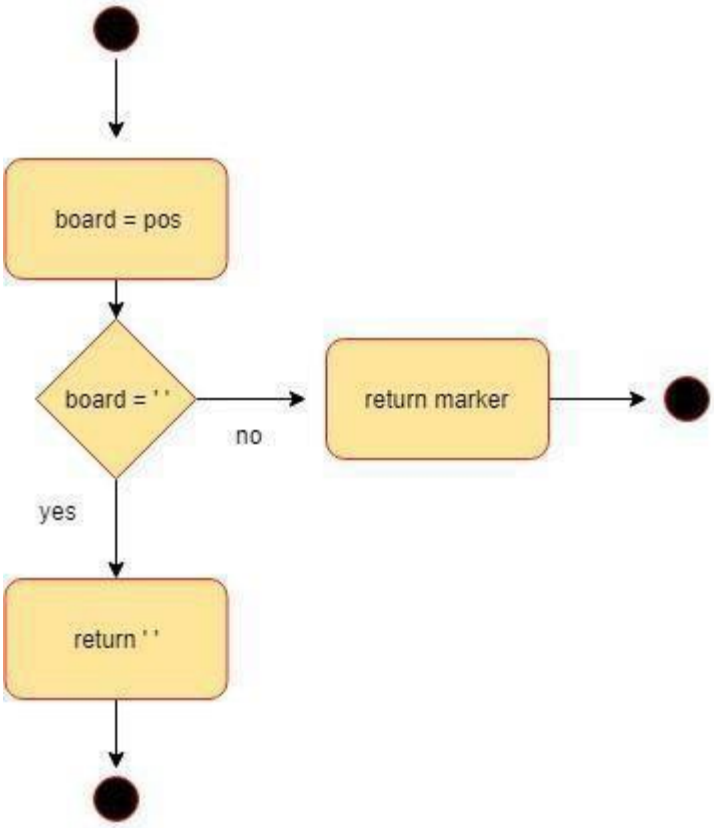
default boolean checkVerticalWin(BoardPosition lastPos, char player)



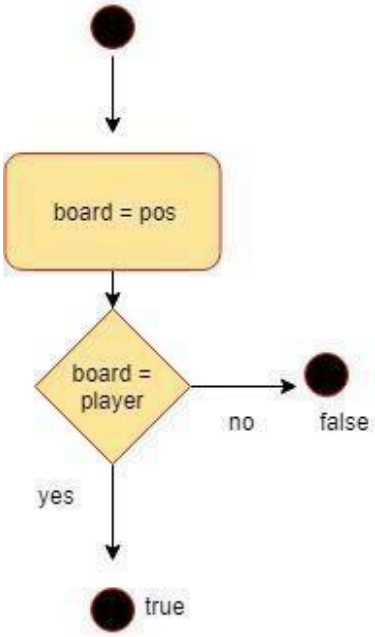
default boolean checkDiagonalWin(BoardPosition lastPos, char player)



Public char whatsAtPos(BoardPosition pos)



Public bool isPlayerAtPos(BoardPosition pos, char player)



GAMESCREEN: public static void main(String [] args)