Mike Brandin

Dr. Yu-Shan Sun

CPSC 2150 Section 002

February 27th, 2021

Project Report:

Project 2

Requirements Analysis

Functional Requirements

Gamescreen.java

- As a user, I can input a column number, so I can mark a spot on the game board and progress the game.
- As a user, I can input a Y or an N after the game has concluded, to start a new game or terminate the game session, respectively.
- As a user, I can place four markers touching horizontally, to conclude the match and receive a congratulatory message and a request to play again.
- As a user, I can place four markers touching vertically, to conclude the match and receive a congratulatory message and a request to play again.
- As a user, I can place four markers touching diagonally, to conclude the match and receive a congratulatory message and a request to play again.
- As a user, I can fill the board fully with markers and not win, to conclude the match with a tie and a request to play again.

Gameboard.java

- As a user, I must be able to input a position value and receive character value, to know what player is in that position.
- As a user, I must be able to input a column number, to input a token into the highest available row in that column.
- As a user, I must be able to call a function to receive the value of the number of rows.
- As a user, I must be able to call a function to receive the value of the number of columns.
- As a user, I must be able to call a function to receive the value of the number needed to win.

AbsGameboard.java

- As a user, I must be able to request a fully formatted string representation of the gameboard, to visualize the current gameboard.

IGameboard.java

- As a user, I must be able to input a column number and receive a true or false value, to know whether or not a column is free for more tokens.

- As a user, I must be able to input a column number and receive a true or false value, to know whether or not the last placed token resulted in a win.
- As a user, I must be able to input a column number and receive a true or false value, to know whether or not the last placed token resulted in a tie (a full board).
- As a user, I must be able to input a character token and a position value and receive a true or false value, to know whether or not the last placed token resulted in a horizontal win.
- As a user, I must be able to input a character token and a position value and receive a true or false value, to know whether or not the last placed token resulted in a vertical win.
- As a user, I must be able to input a character token and a position value and receive a true or false value, to know whether or not the last placed token resulted in a diagonal win.
- As a user, I must be able to input a character representing a player and a position value and receive a true or false value, to know whether or not that player is in that position.

BoardPosition.java

- As a user, I must be able to receive a string output to display the row and column coordinates of the position.
- As a user, I must be able to request the Row variables value, to know the value of the board position's row.
- As a user, I must be able to request the Column variables value, to know the value of the board position's column.
- As a user, I must be able to compare two BoardPosition variables, to know whether their positional values are equivalent.

Nonfunctional Requirements

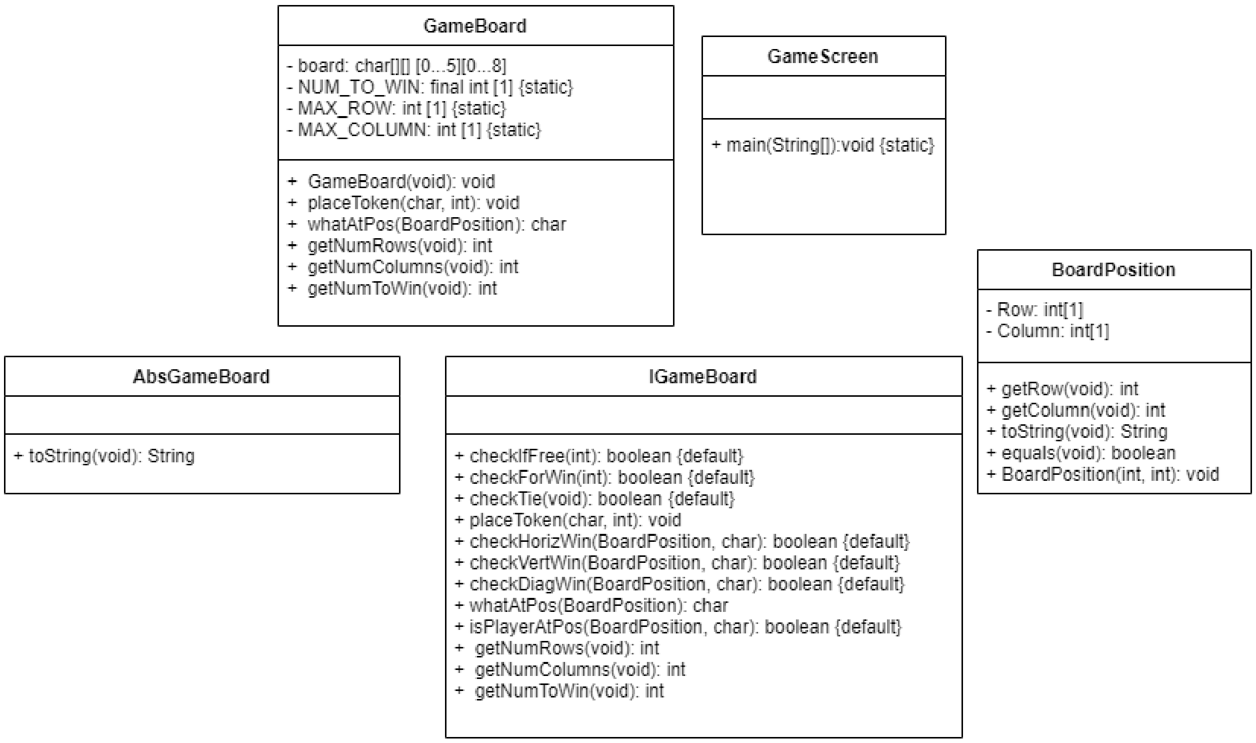
- Must have a device that supports Java.
- Must have a keyboard to play the game.
- Must run on the Schools of Computing's virtual machine.
- Must have adequate memory to allocate towards objects.
- Must handle all I/O in GameScreen.java
- Gameboard must be arranged in a 6 by 9 arrangement.
- Player X must start by taking his turn first then followed by Player O's turn.

Make File Instructions

To use the make file, open the terminal in the Project2 directory. Type **make** into the terminal followed by enter to compile the project files. Then, type **make run** into the terminal followed by enter to run the program. Once you are finished, type **make clean** into the terminal followed by enter to remove all compiled files from the extendedConnectX package directory.

Design

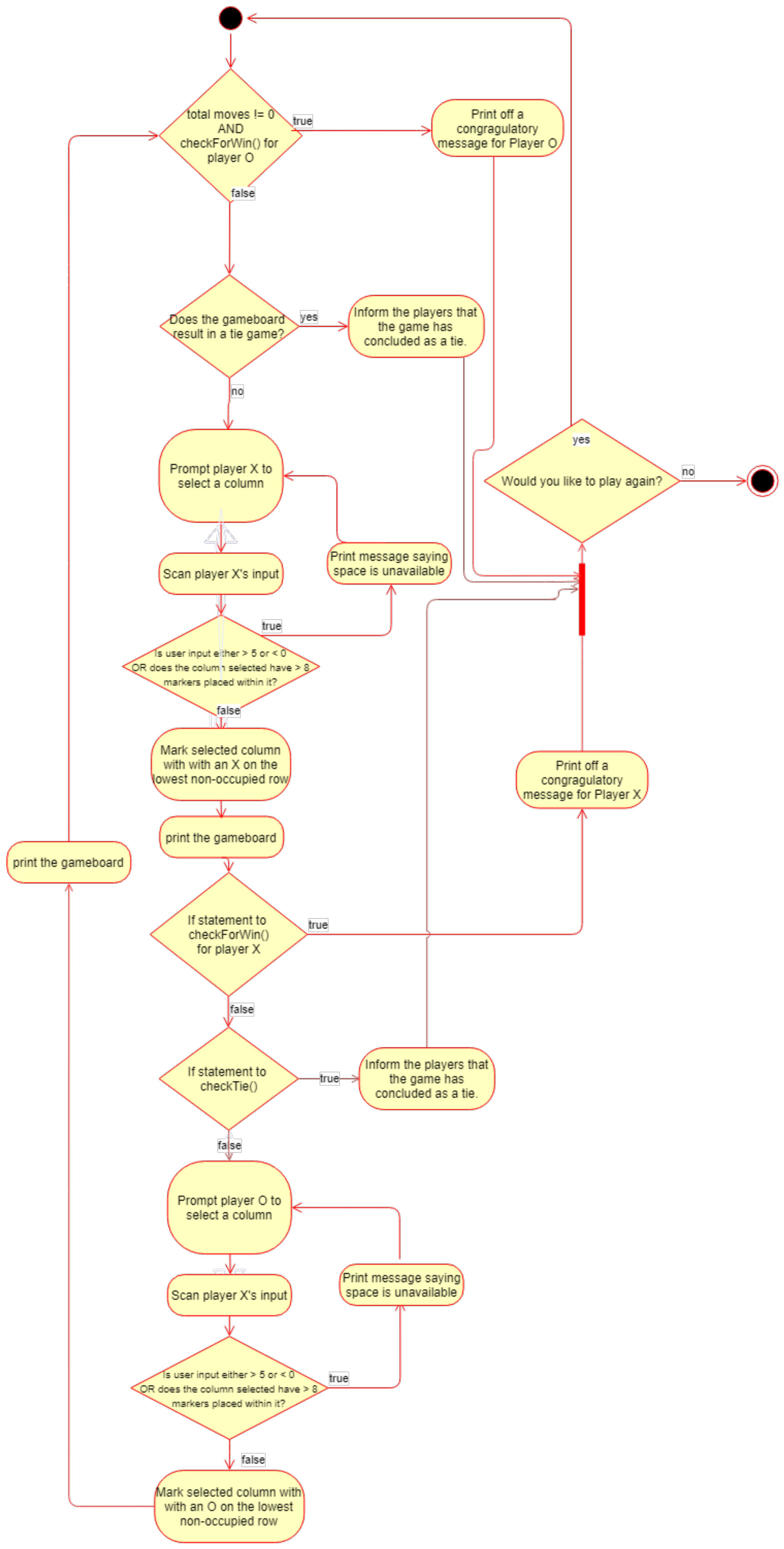
UML Class Diagrams


GameBoard
- board: char[][] [0...5][0...8]
- NUM_TO_WIN: final int [1] {static}
- MAX_ROW: int [1] {static}
- MAX_COLUMN: int [1] {static}
+ GameBoard(void): void
+ placeToken(char, int): void
+ whatAtPos(BoardPosition): char
+ getNumRows(void): int
+ getNumColumns(void): int
+ getNumToWin(void): int
GameScreen
+ main(String[]):void {static}
BoardPosition
- Row: int[1]
- Column: int[1]
+ getRow(void): int
+ getColumn(void): int
+ toString(void): String
+ equals(void): boolean
+ BoardPosition(int, int): void
AbsGameBoard
+ toString(void): String
IGameBoard
+ checkIfFree(int): boolean {default}
+ checkForWin(int): boolean {default}
+ checkTie(void): boolean {default}
+ placeToken(char, int): void
+ checkHorizWin(BoardPosition, char): boolean {default}
+ checkVertWin(BoardPosition, char): boolean {default}
+ checkDiagWin(BoardPosition, char): boolean {default}
+ whatAtPos(BoardPosition): char
+ isPlayerAtPos(BoardPosition, char): boolean {default}
+ getNumRows(void): int
+ getNumColumns(void): int
+ getNumToWin(void): int

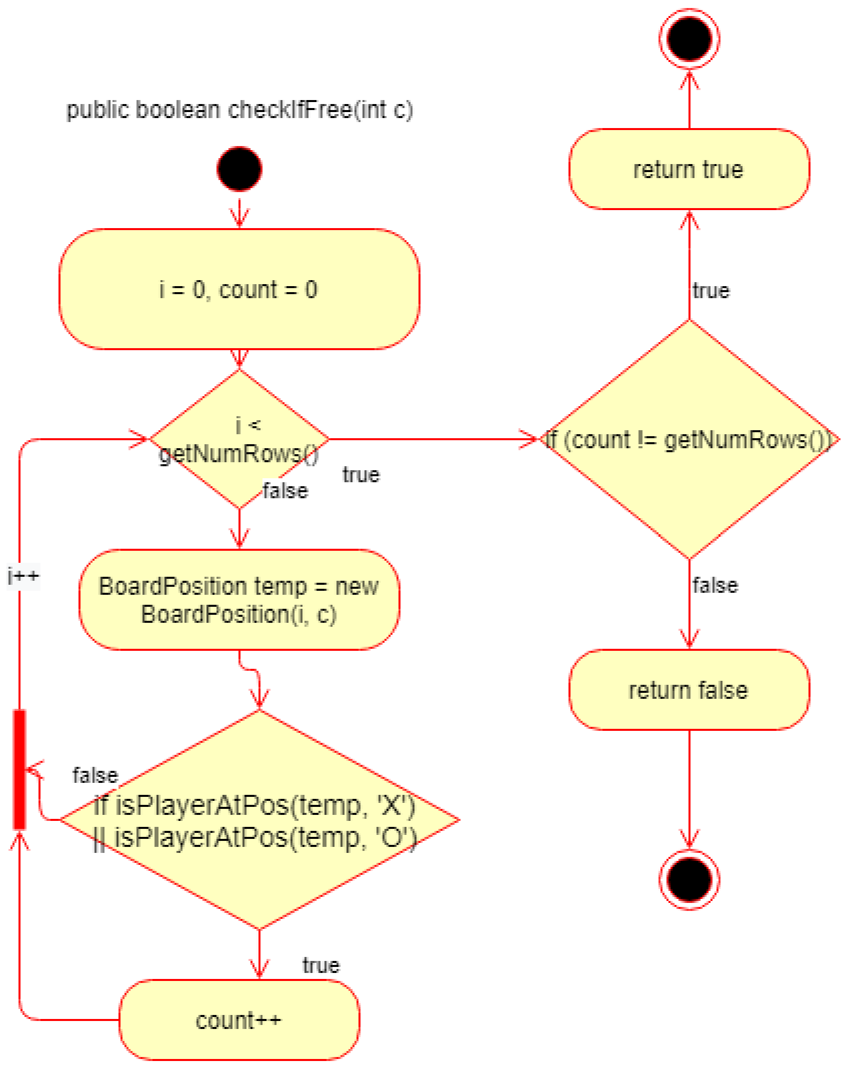
UML Activity Diagrams

- GameScreen.java:
    - main function

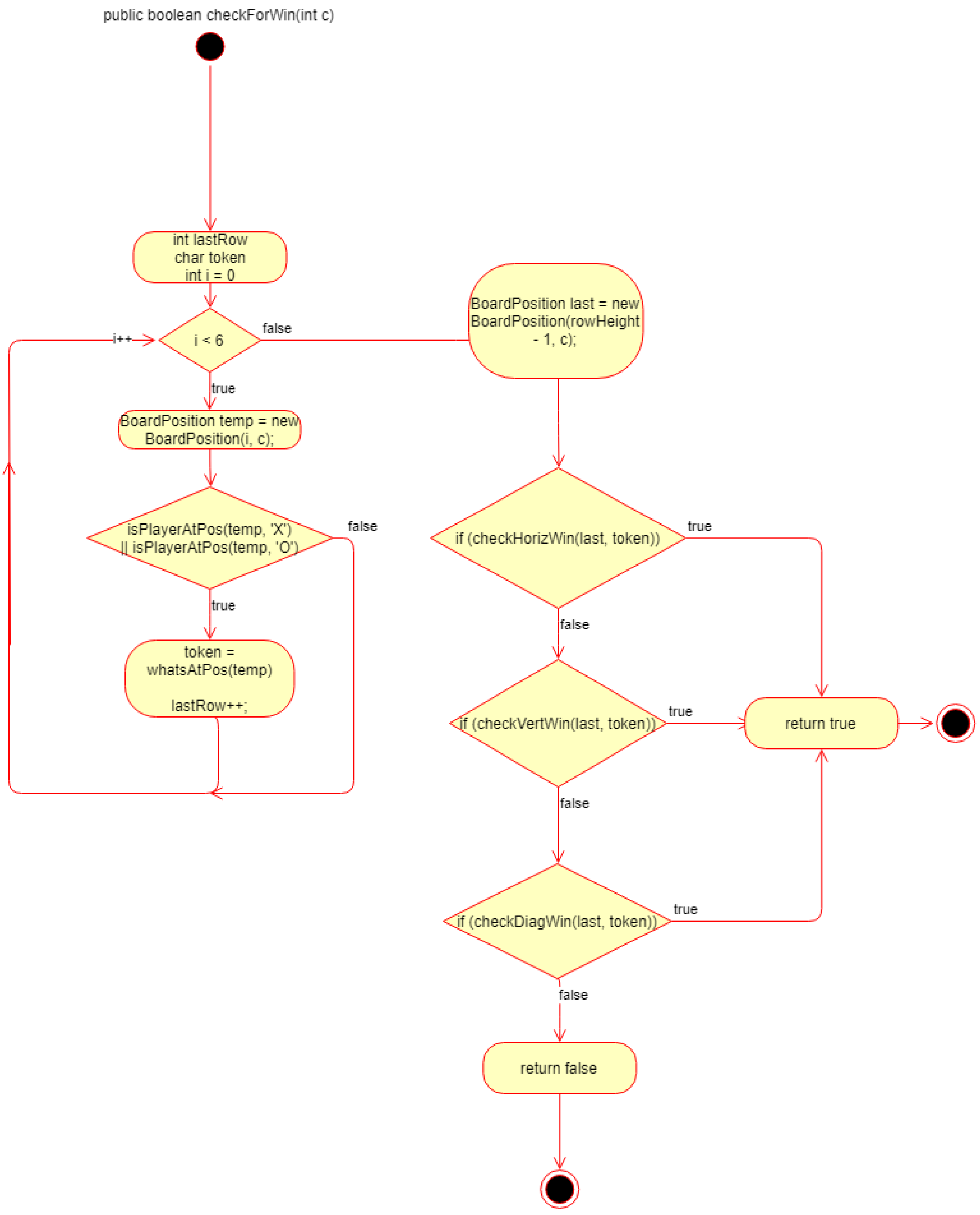
public static void main(String[] args)


total moves != 0 AND checkForWin() for player O
true
Print off a congragulatory message for Player O
false
Does the gameboard result in a tie game?
yes
Inform the players that the game has concluded as a tie.
no
Prompt player X to select a column
Scan player X's input
Print message saying space is unavailable
true
Is user input either > 5 or < 0 OR does the column selected have > 8 markers placed within it?
false
Mark selected column with with an X on the lowest non-occupied row
print the gameboard
If statement to checkForWin() for player X
true
Print off a congragulatory message for Player X
false
If statement to checkTie()
true
Inform the players that the game has concluded as a tie.
false
Prompt player O to select a column
Scan player X's input
Print message saying space is unavailable
Is user input either > 5 or < 0 OR does the column selected have > 8 markers placed within it?
true
false
Mark selected column with with an O on the lowest non-occupied row
print the gameboard
yes
Would you like to play again?
no

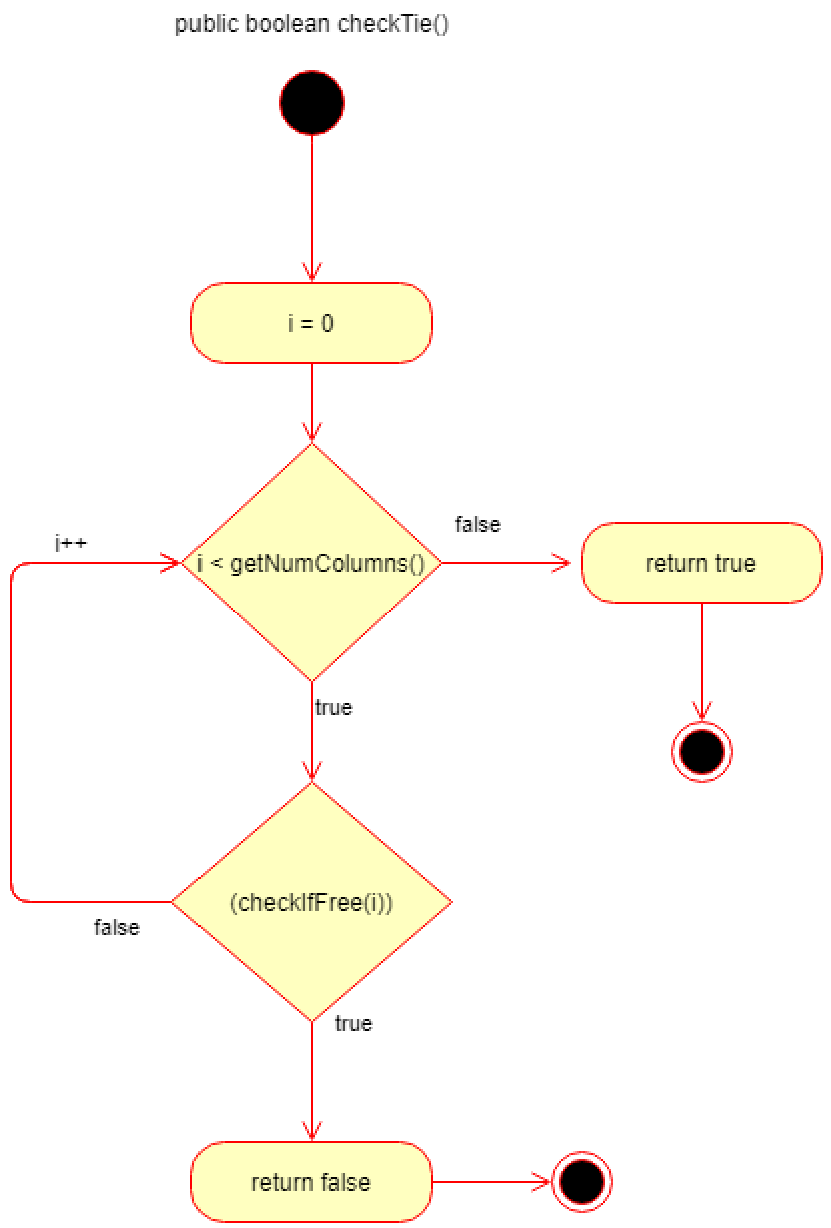
- IGameBoard.java
    - checkIfFree method


public boolean checkIfFree(int c)
i = 0, count = 0
i < getNumRows()
true
false
BoardPosition temp = new BoardPosition(i, c)
if isPlayerAtPos(temp, 'X') || isPlayerAtPos(temp, 'O')
false
true
count++
i++
if (count != getNumRows())
true
return true
false
return false

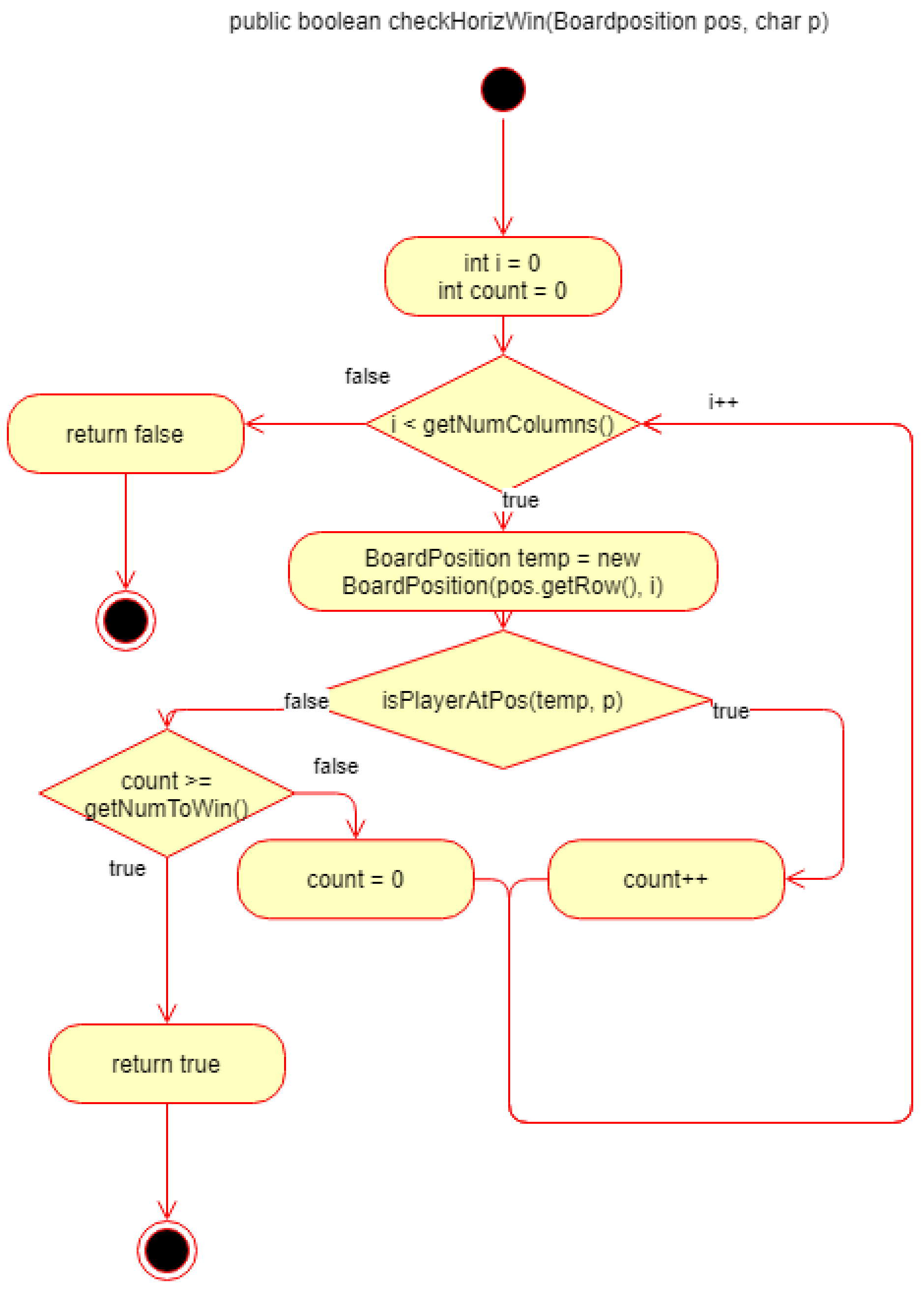
- checkForWin method


public boolean checkForWin(int c)
int lastRow
char token
int i = 0
i++
i < 6
false
true
BoardPosition temp = new BoardPosition(i, c);
isPlayerAtPos(temp, 'X') || isPlayerAtPos(temp, 'O')
false
true
token = whatsAtPos(temp)
lastRow++;
BoardPosition last = new BoardPosition(rowHeight - 1, c);
if (checkHorizWin(last, token))
true
false
if (checkVertWin(last, token))
true
return true
false
if (checkDiagWin(last, token))
true
false
return false

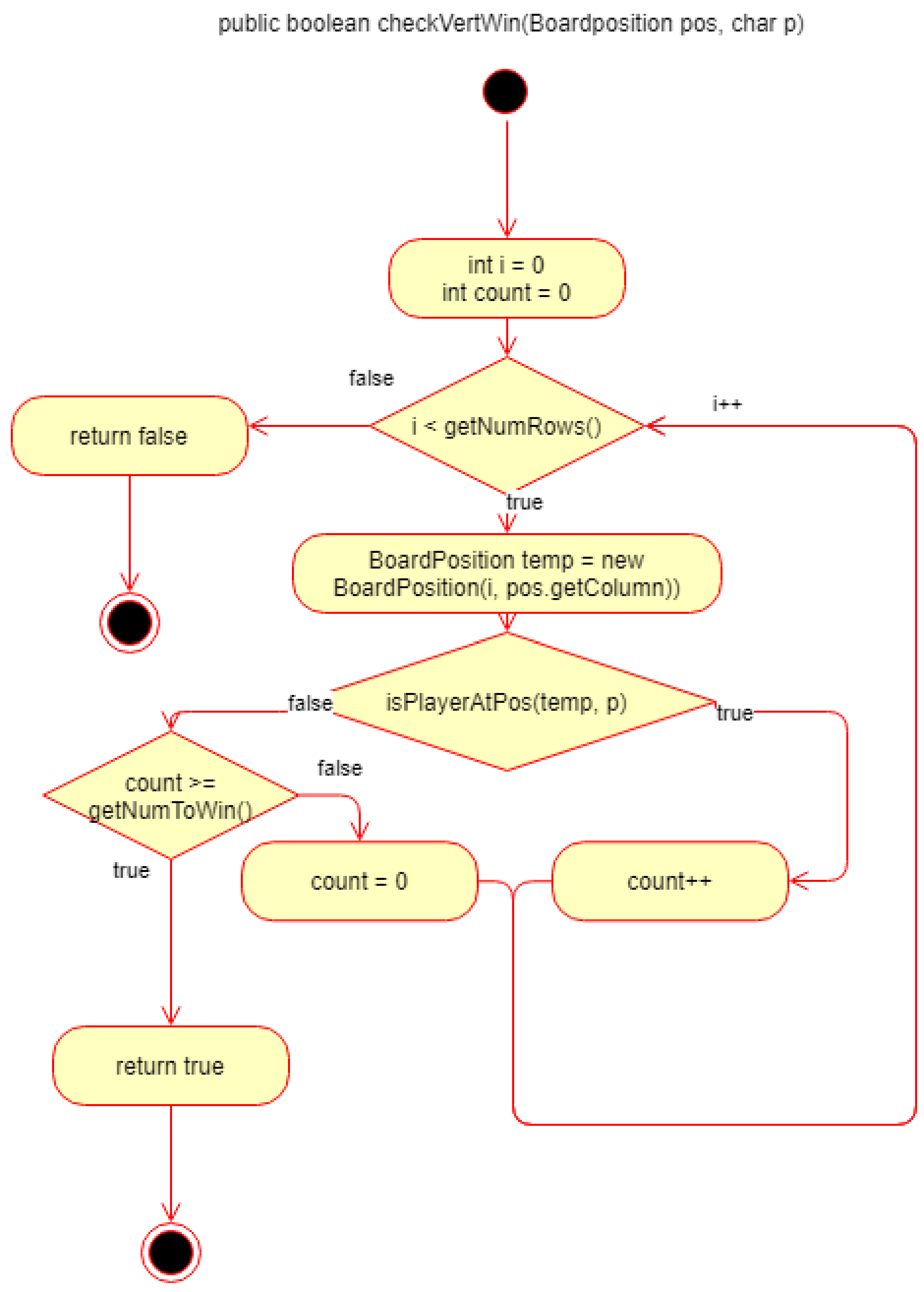
- checkTie method



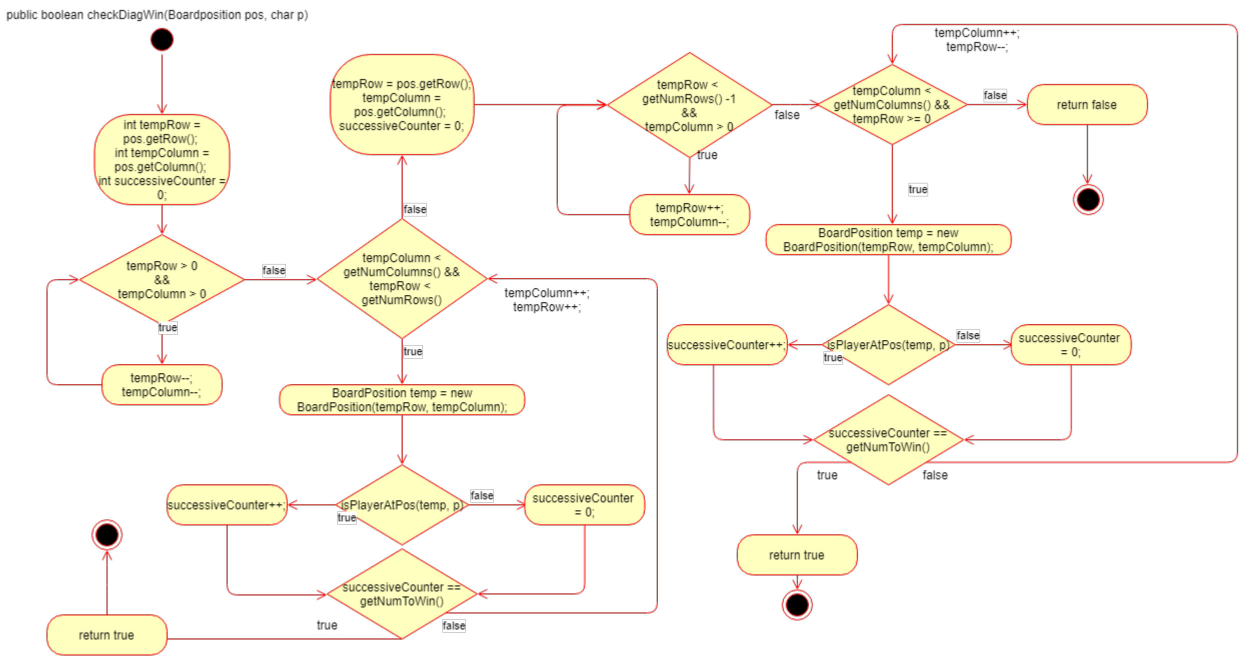
- checkHorizWin method



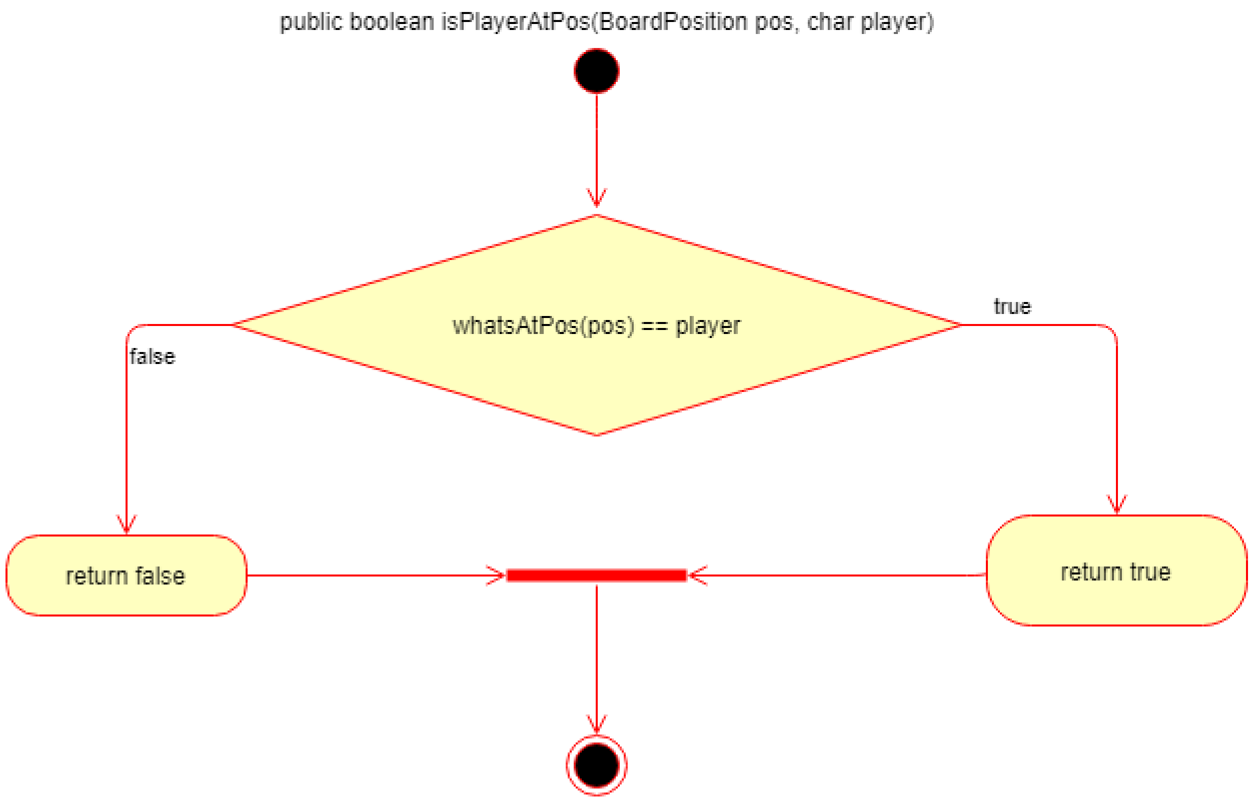
- checkVertWin method



- checkDiagWin method

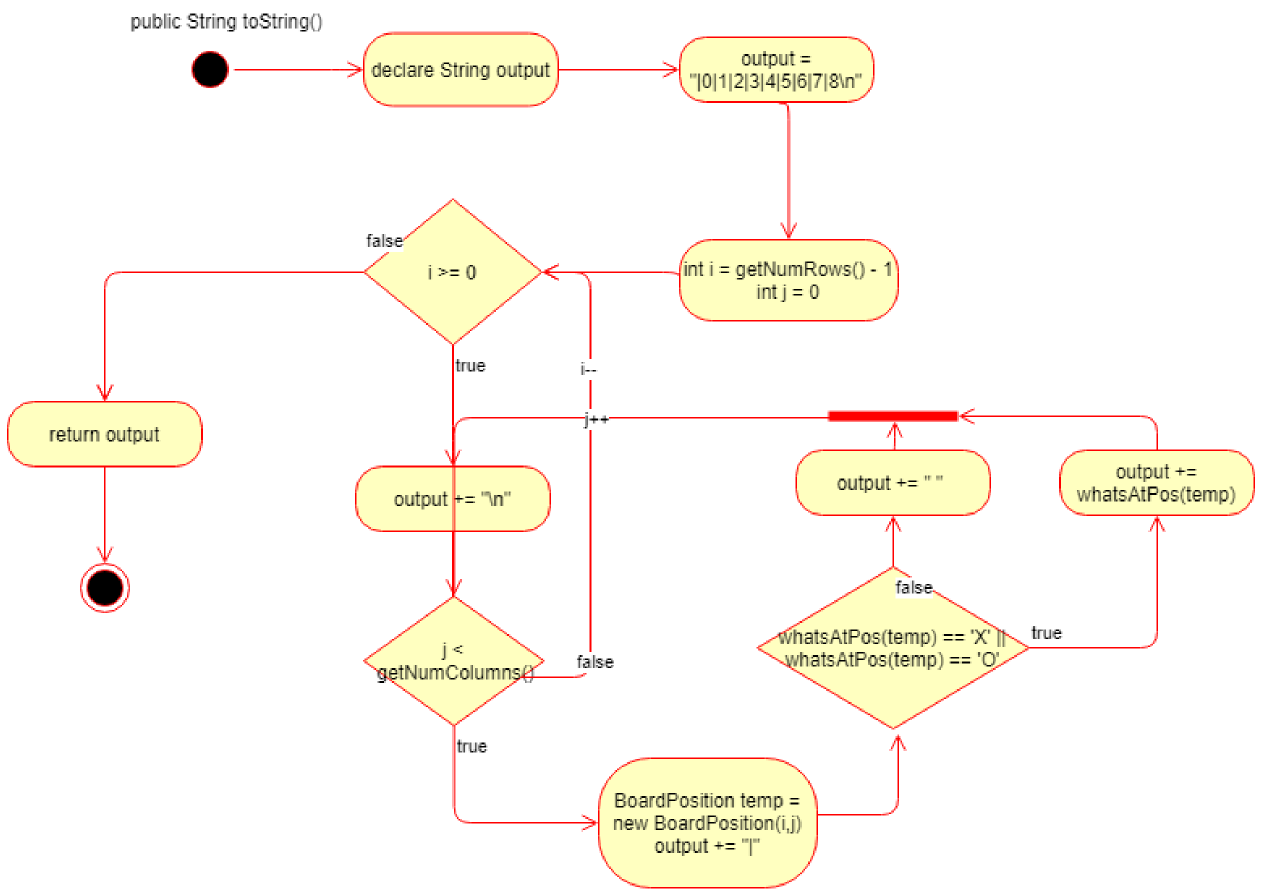

public boolean checkDiagWin(Boardposition pos, char p)
int tempRow = pos.getRow(); int tempColumn = pos.getColumn(); int successiveCounter = 0;
tempRow > 0 && tempColumn > 0
true
false
tempRow--; tempColumn--;
tempColumn < getNumColumns() && tempRow < getNumRows()
true
false
tempColumn++; tempRow++;
BoardPosition temp = new BoardPosition(tempRow, tempColumn);
isPlayerAtPos(temp, p)
true
false
successiveCounter++;
successiveCounter = 0;
successiveCounter == getNumToWin()
true
false
return true
tempRow = pos.getRow(); tempColumn = pos.getColumn(); successiveCounter = 0;
tempRow < getNumRows() -1 && tempColumn > 0
true
false
tempRow++; tempColumn--;
tempColumn < getNumColumns() && tempRow >= 0
true
false
tempColumn++; tempRow--;
return false
BoardPosition temp = new BoardPosition(tempRow, tempColumn);
isPlayerAtPos(temp, p)
true
false
successiveCounter++;
successiveCounter = 0;
successiveCounter == getNumToWin()
true
false
return true

- isPlayerAtPos method

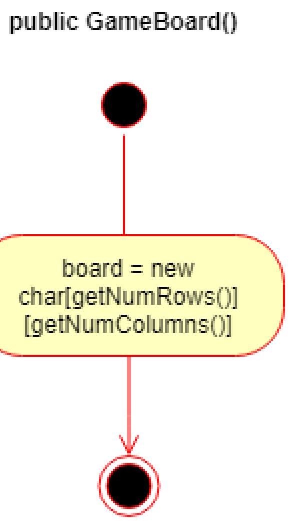

public boolean isPlayerAtPos(BoardPosition pos, char player)
whatsAtPos(pos) == player
false
true
return false
return true

- AbsGameBoard.java

- toString method

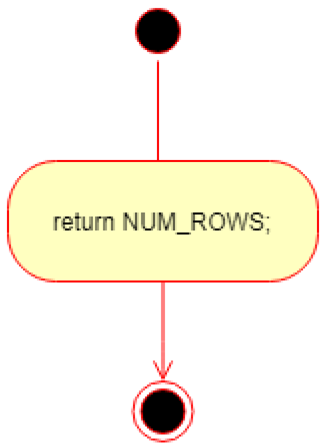


- GameBoard.java
    - GameBoard constructor



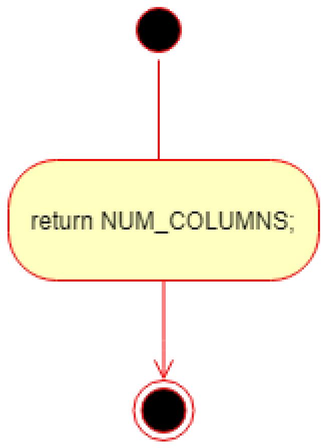
- getNumRows method

public int getNumRows()

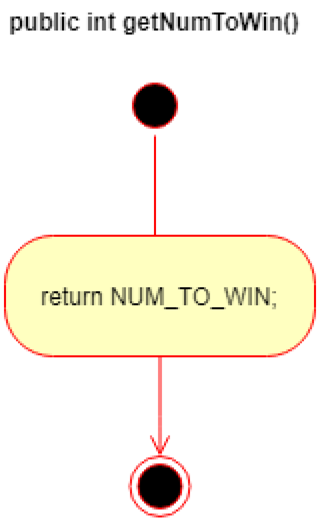


- getNumColumns method

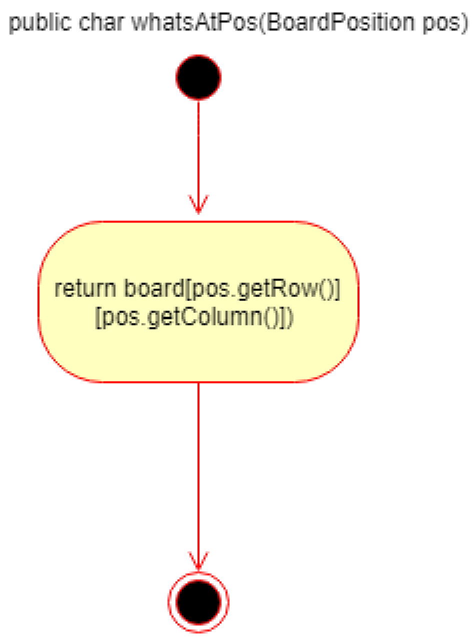
public int getNumColumns()



- getNumToWin method

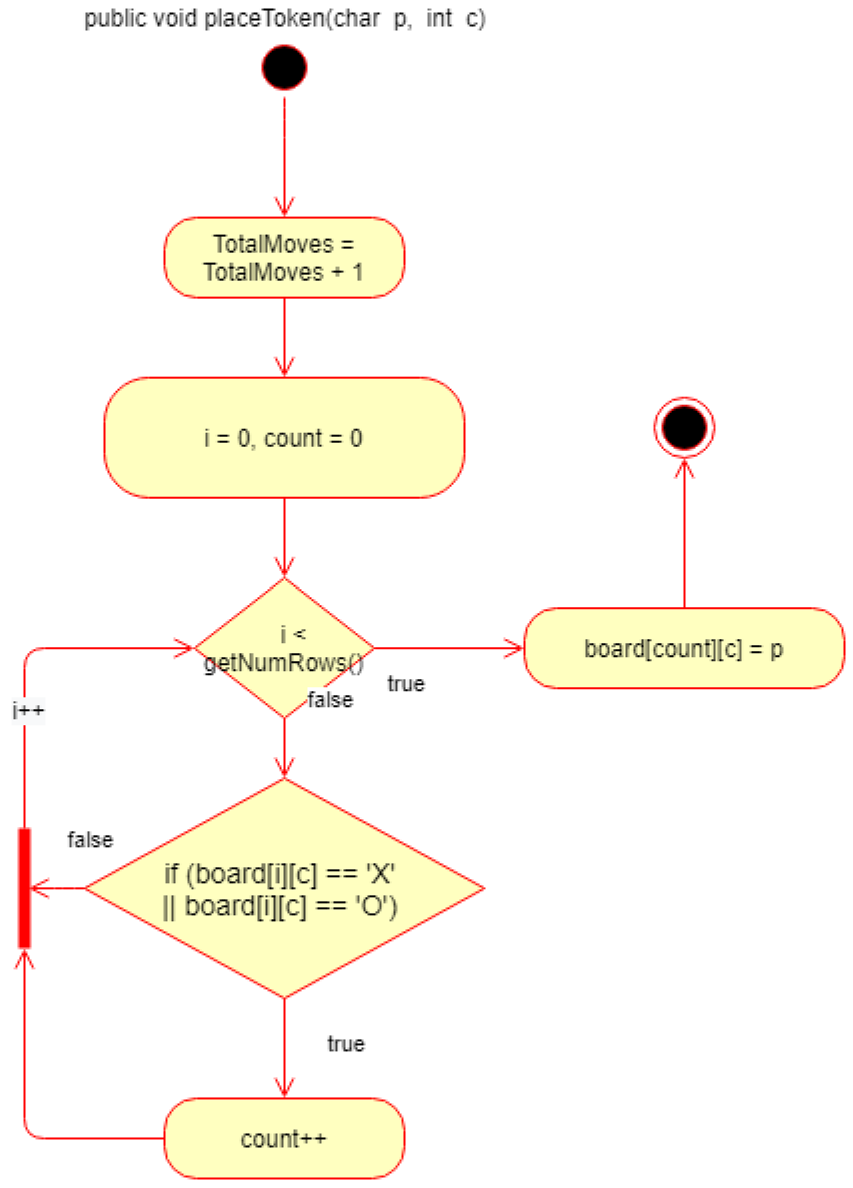


- whatsAtPos method



public char whatsAtPos(BoardPosition pos)
return board[pos.getRow()][pos.getColumn()])

- placeToken method


public void placeToken(char p, int c)
TotalMoves = TotalMoves + 1
i = 0, count = 0
i < getNumRows()
true
false
board[count][c] = p
i++
false
if (board[i][c] == 'X' || board[i][c] == 'O')
true
count++