

Mike Brandin

Dr. Yu-Shan Sun

CPSC 2150 Section 002

April 20th, 2021

Project Report:

Project 5

Requirements Analysis

Functional Requirements

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.

GameboardMem.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

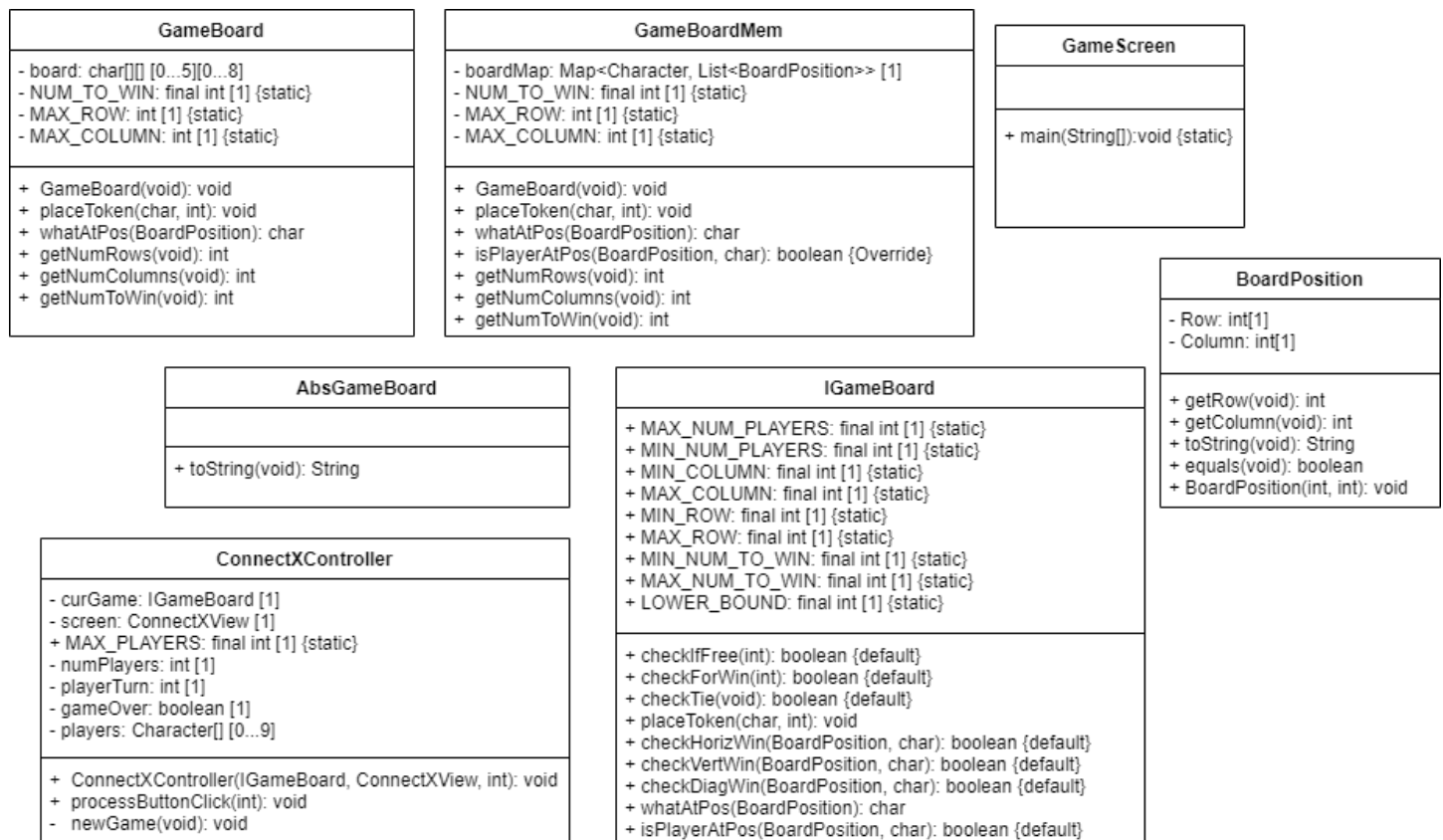
- As a user, I must be able to receive a string output to display the row and column coordinates of the position.
- As a user, I must be able to request the Row variables value, to know the value of the board position's row.
- As a user, I must be able to request the Column variables value, to know the 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 have a mouse to play the game.
- Must have adequate memory to allocate towards objects.
- Gameboard size must not exceed 20 columns nor 20 rows.
- Gameboard size must not be less than 3 columns nor 3 rows.
- Players must take turns in the order they selected their character at the start of the game.

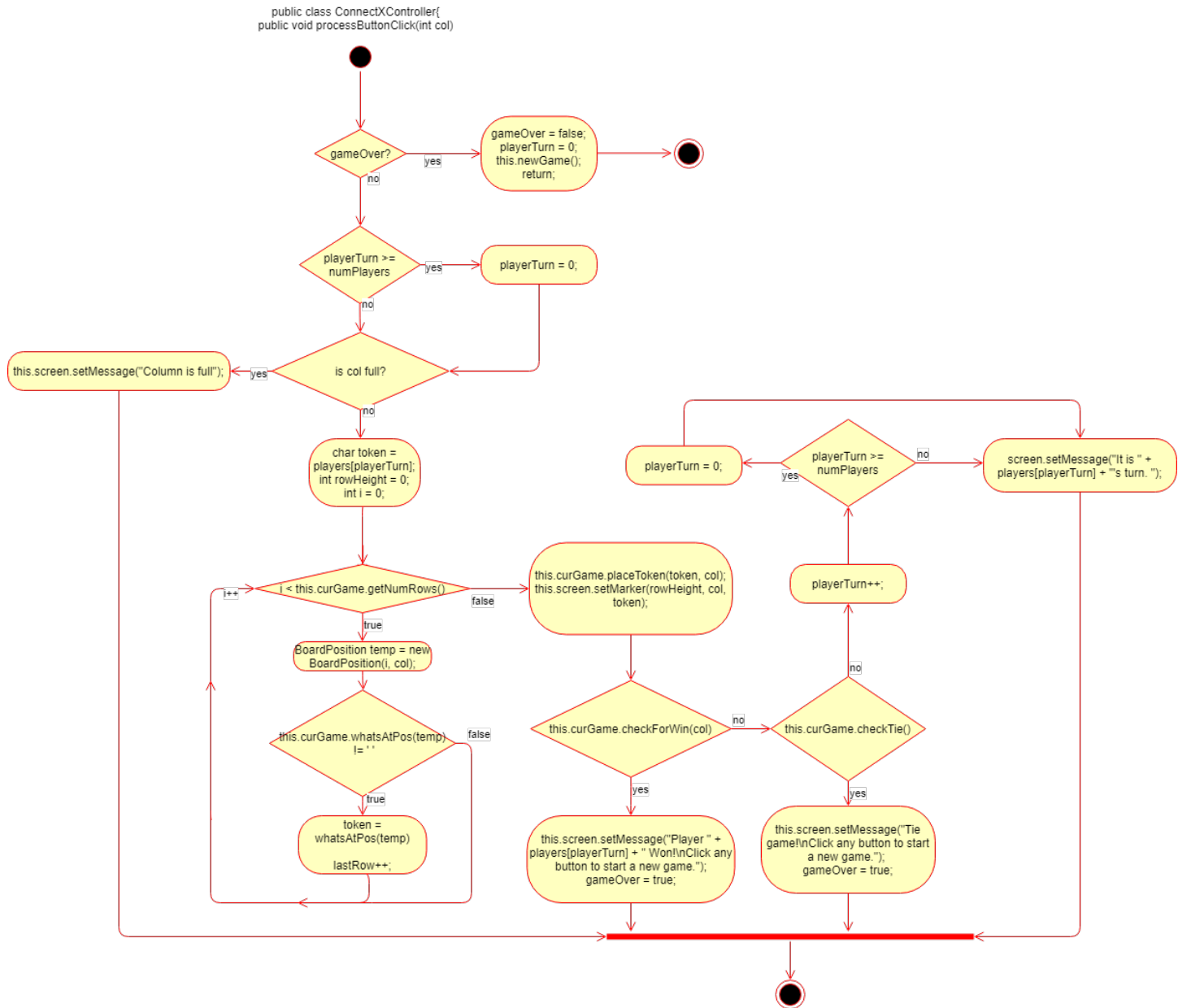
Design

UML Class Diagrams

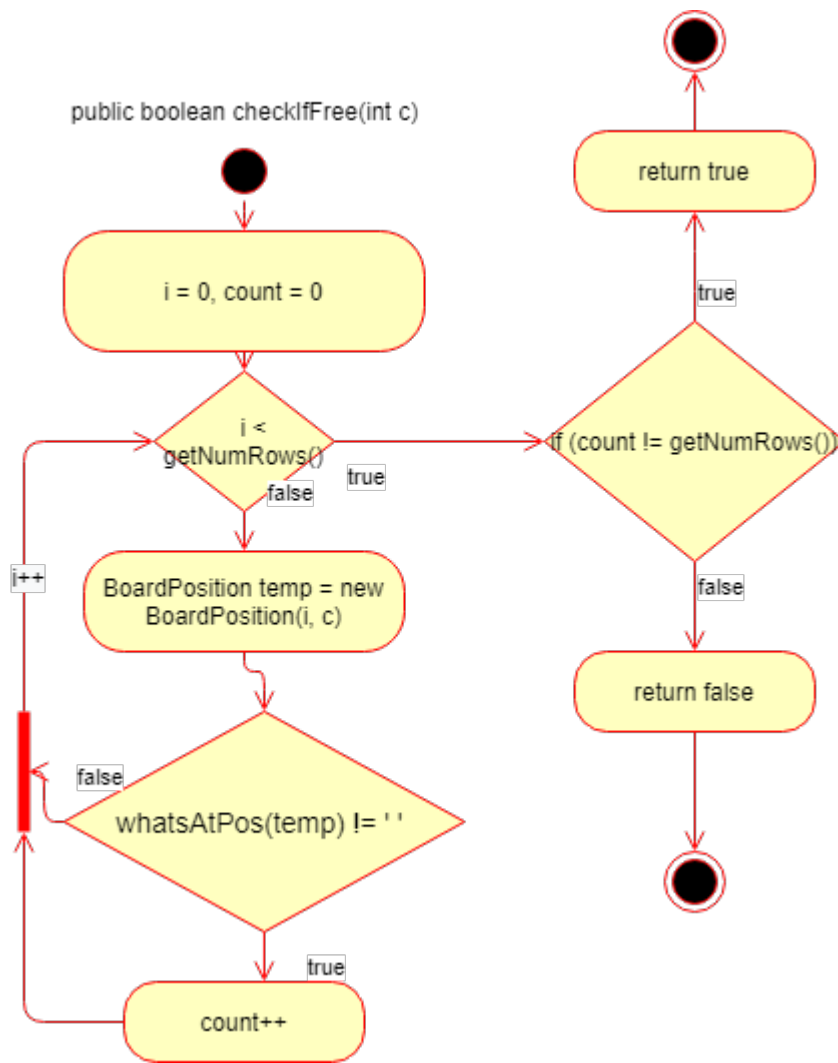


UML Activity Diagrams

- ConnectXController.java:
 - o processButtonClick method

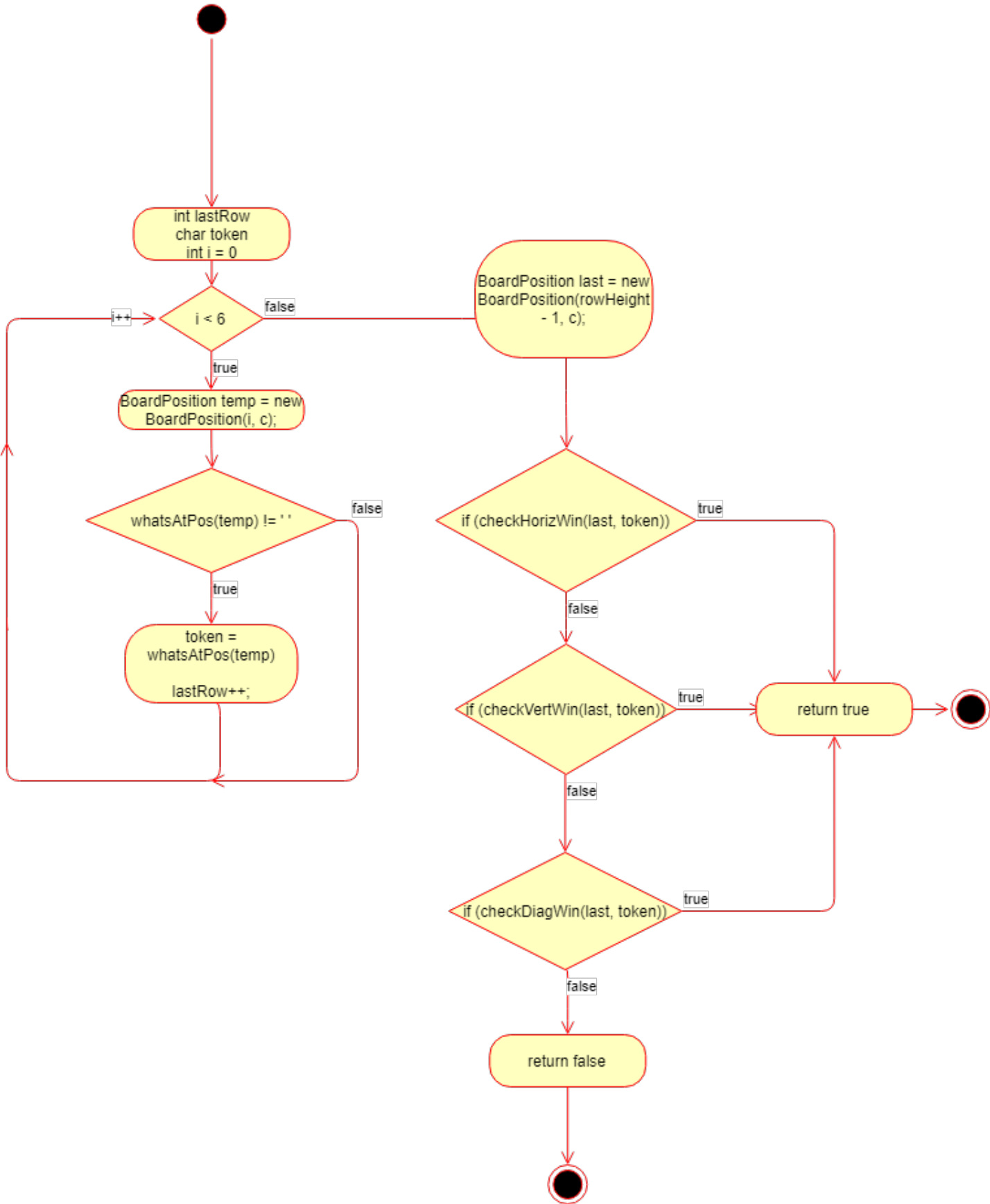


- IGameBoard.java
 - o checkIfFree method



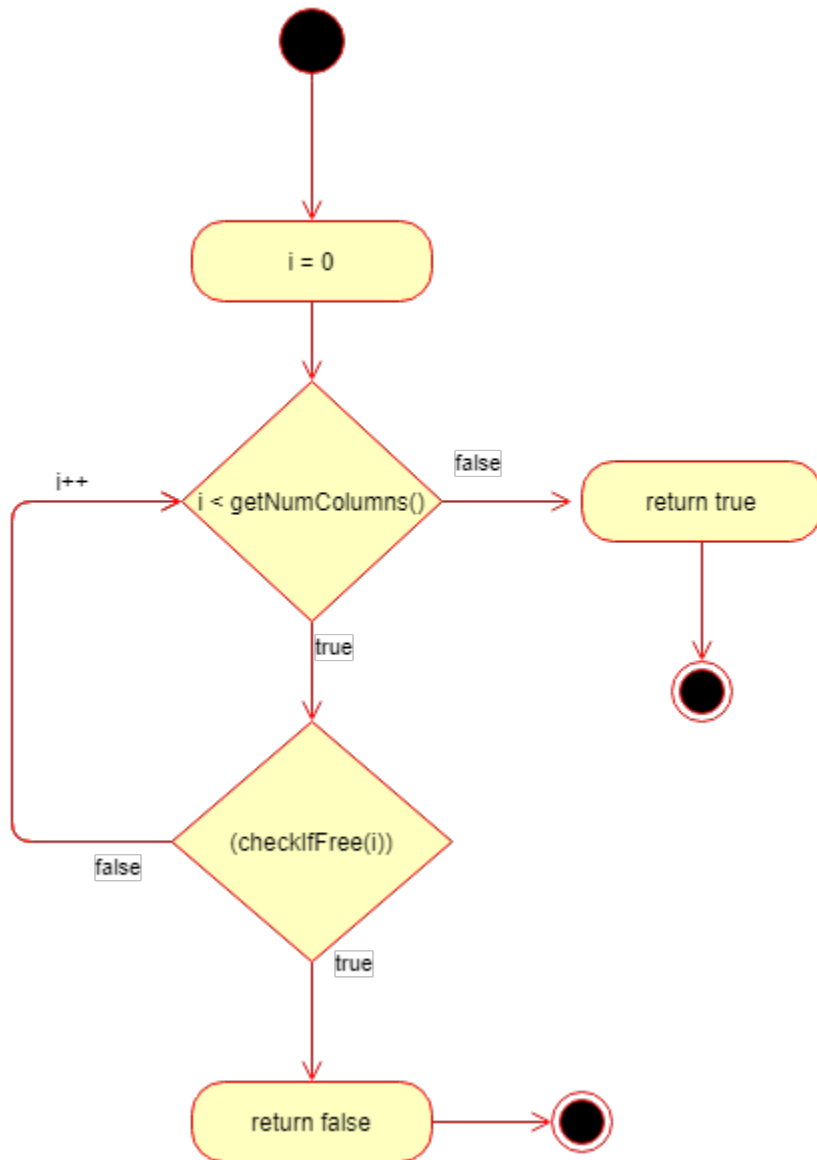
- `checkForWin` method

public boolean checkForWin(int c)



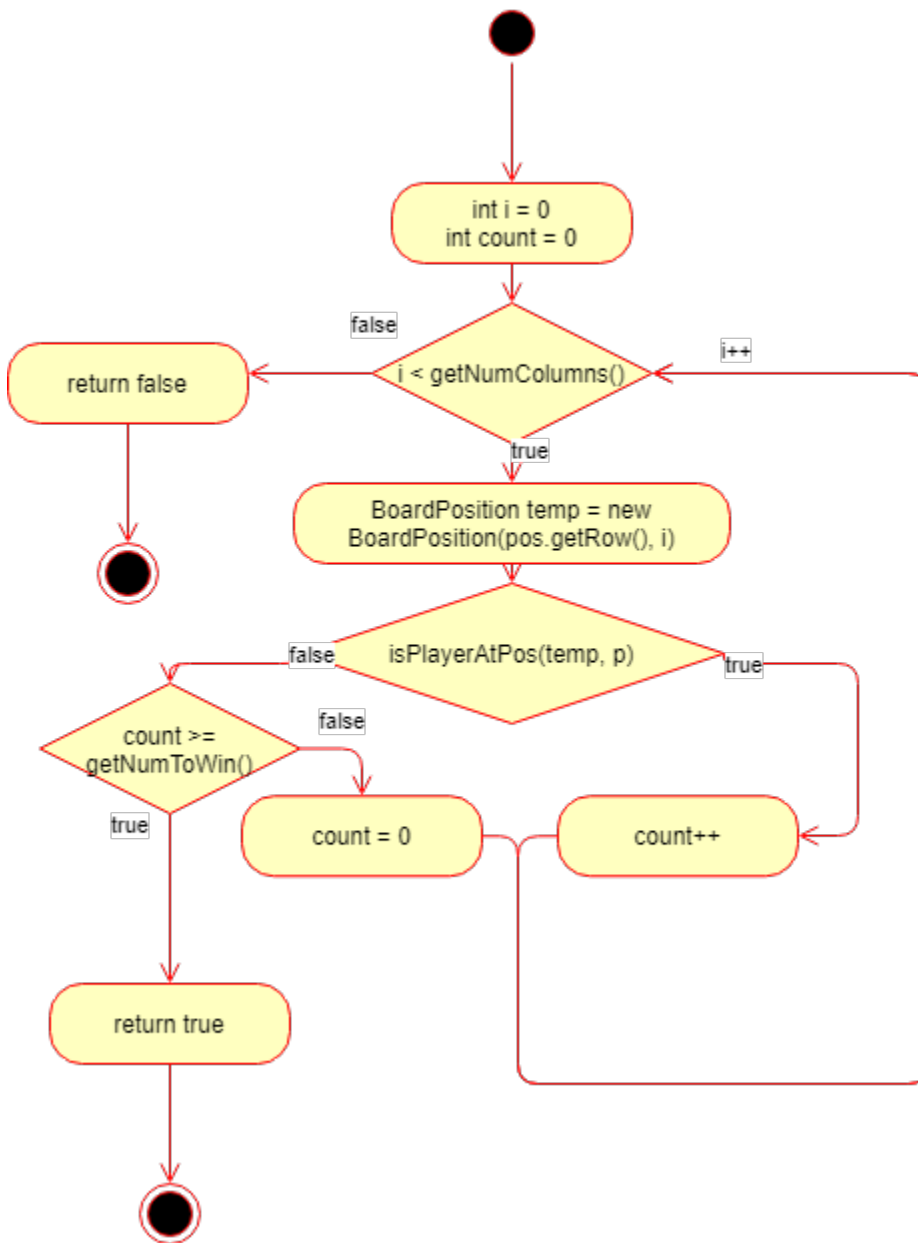
- checkTie method

public boolean checkTie()



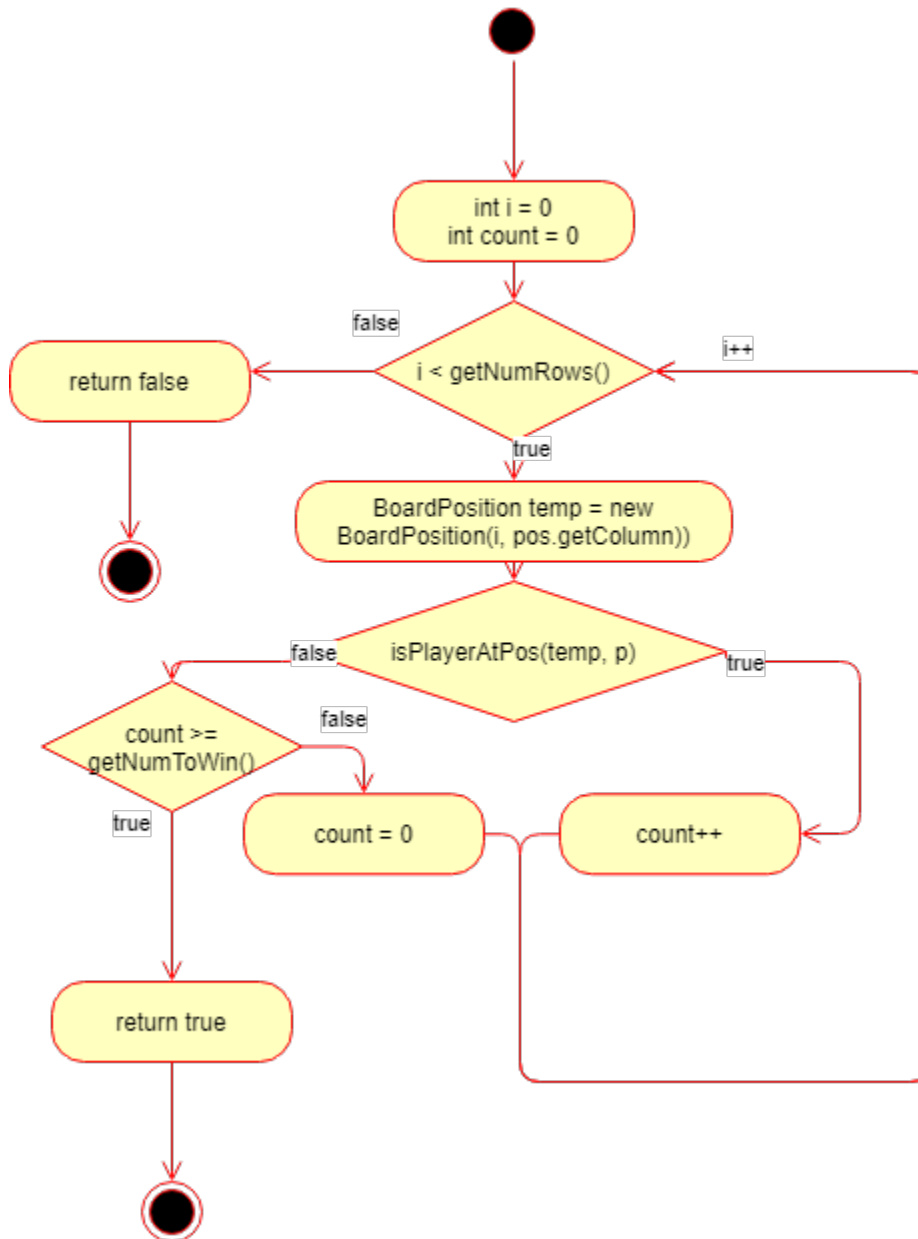
- checkHorizWin method

```
public boolean checkHorizWin(Boardposition pos, char p)
```



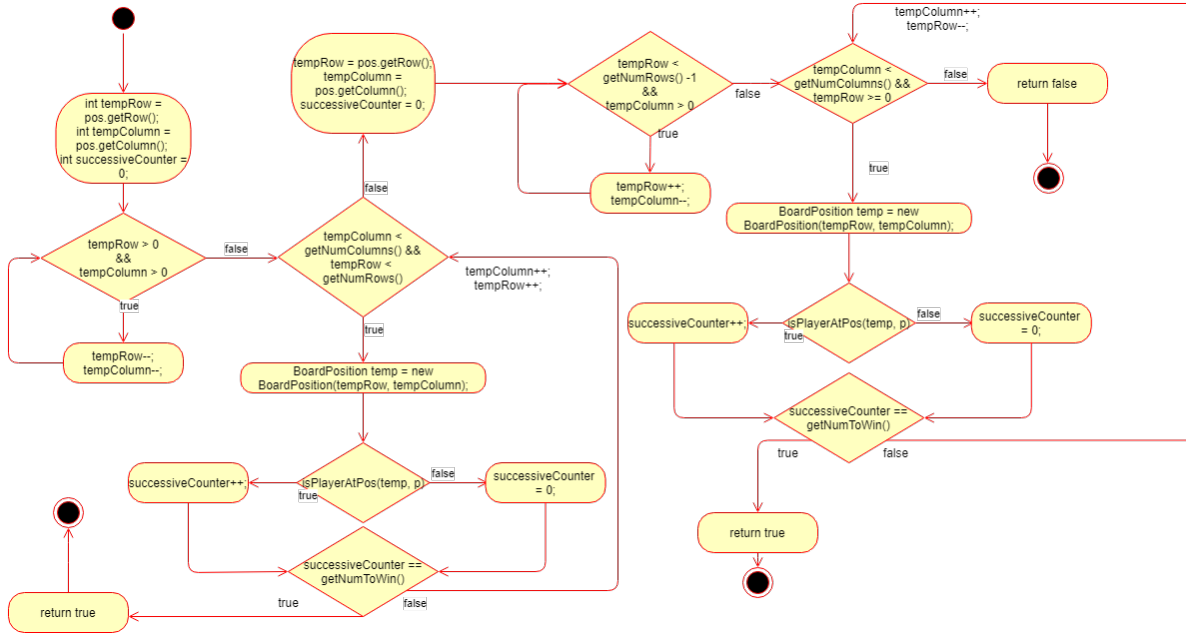
- `checkVertWin` method

```
public boolean checkVertWin(Boardposition pos, char p)
```



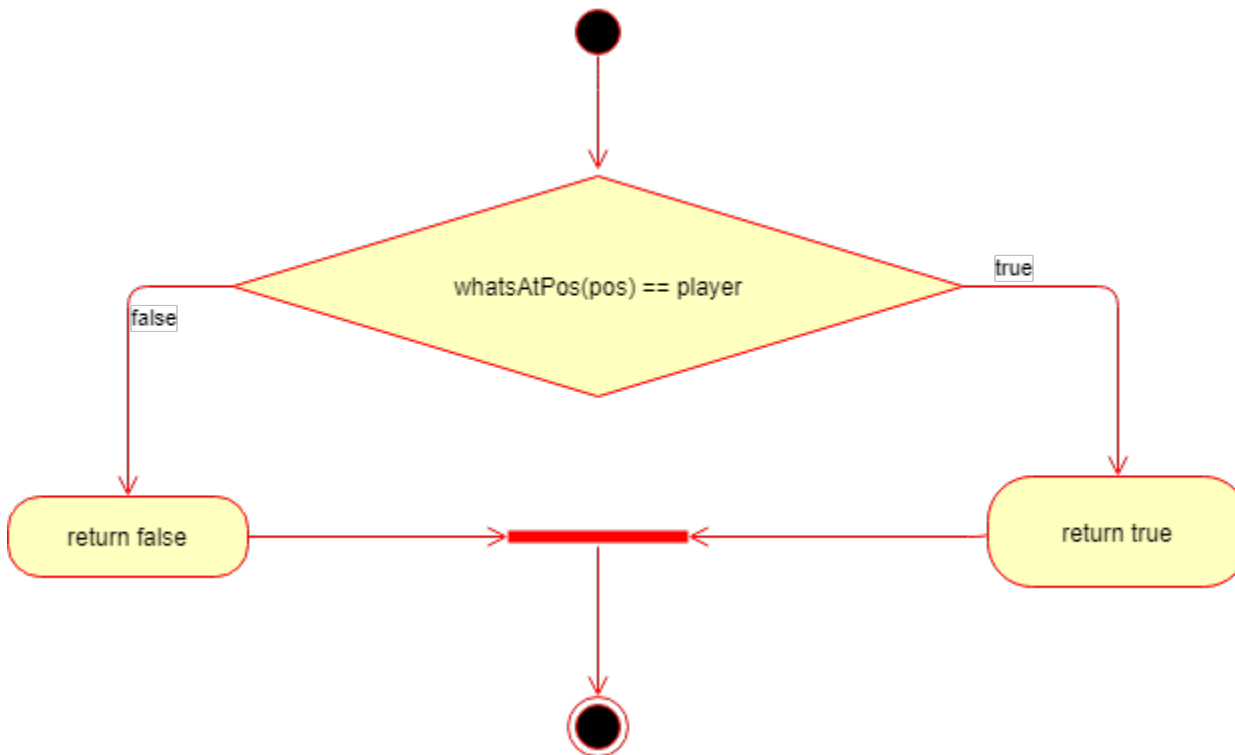
- checkDiagWin method

public boolean checkDiagWin(BoardPosition pos, char p)

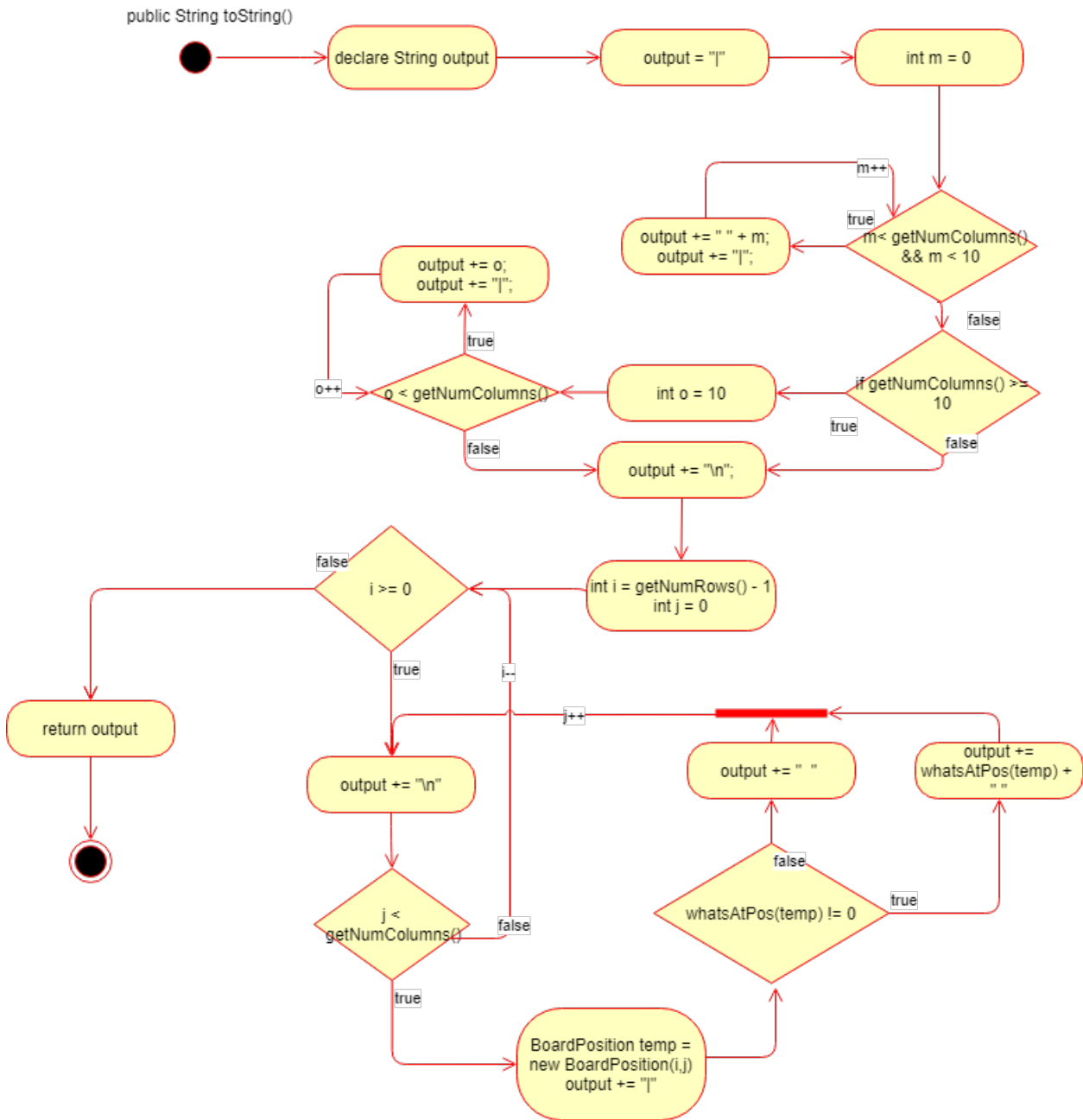


- isPlayerAtPos method

public boolean isPlayerAtPos(BoardPosition pos, char player)

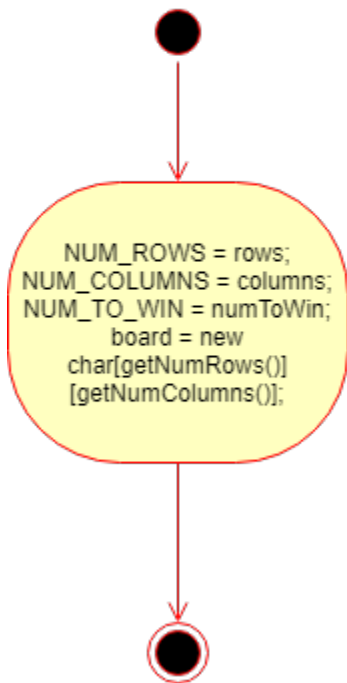


- AbsGameBoard.java
 - toString method



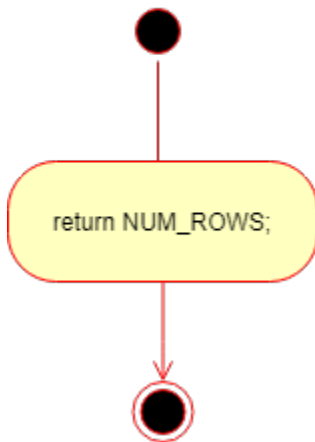
- GameBoard.java
 - o GameBoard constructor

public GameBoard()



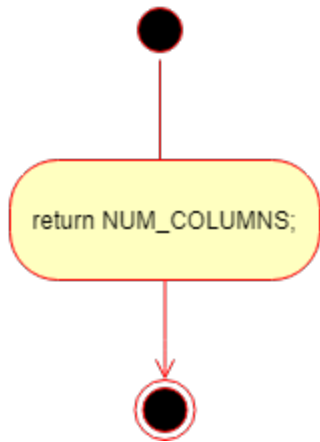
- getNumRows method

public int getNumRows()



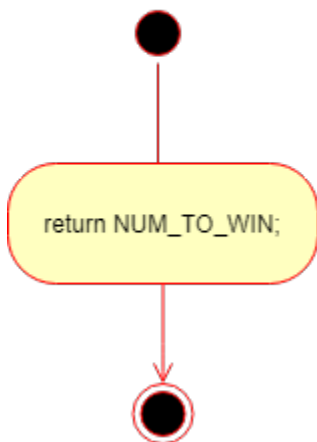
- getNumColumns method

`public int getNumColumns()`

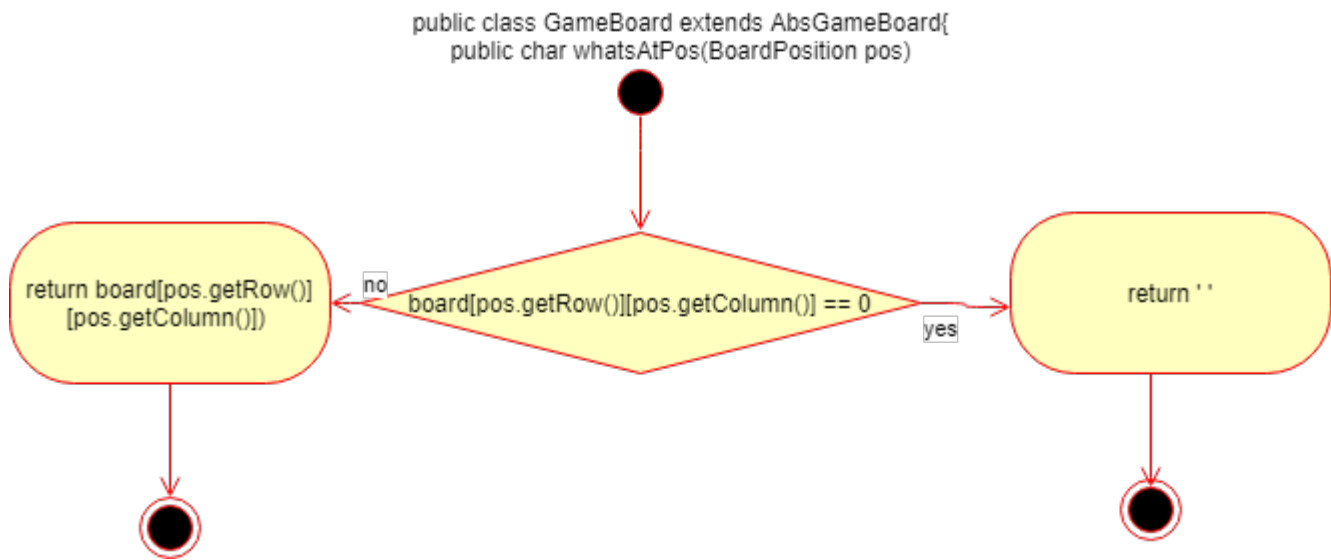


- `getNumToWin` method

`public int getNumToWin()`

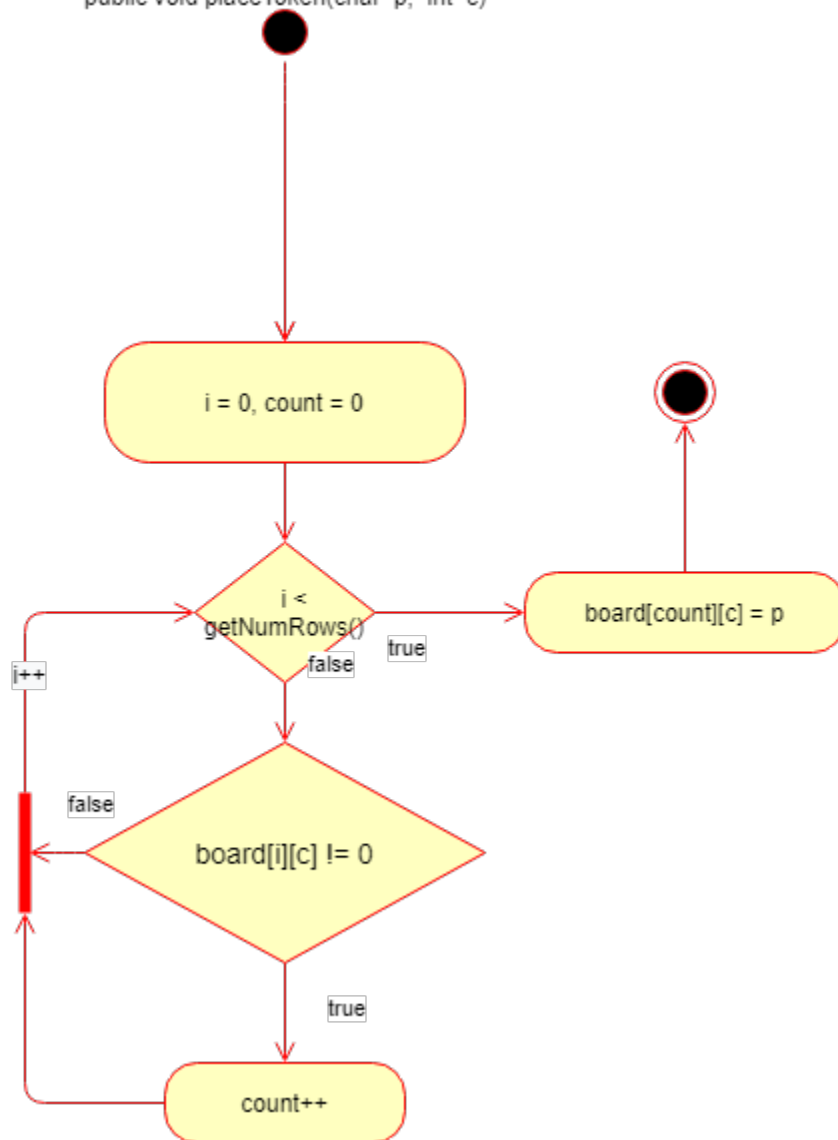


- `whatsAtPos` method



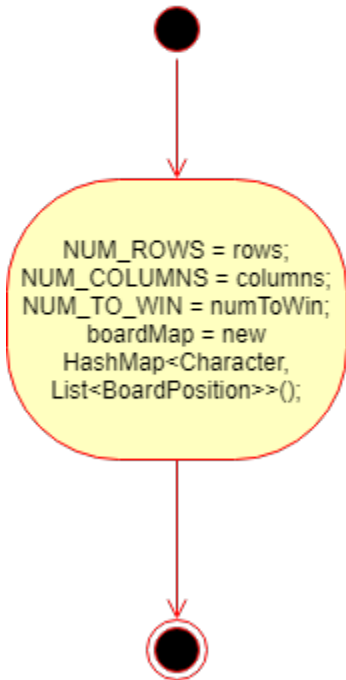
- placeToken method

```
public class GameBoard extends AbsGameBoard{
    public void placeToken(char p, int c)
```



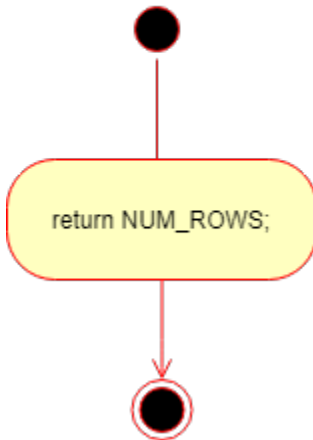
- GameBoardMem.java
 - o GameBoardMem constructor

public GameBoardMem()



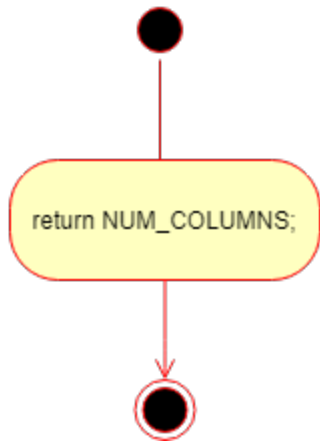
- getNumRows method

public int getNumRows()



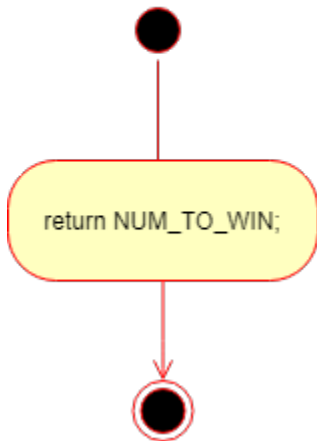
- getNumColumns method

`public int getNumColumns()`



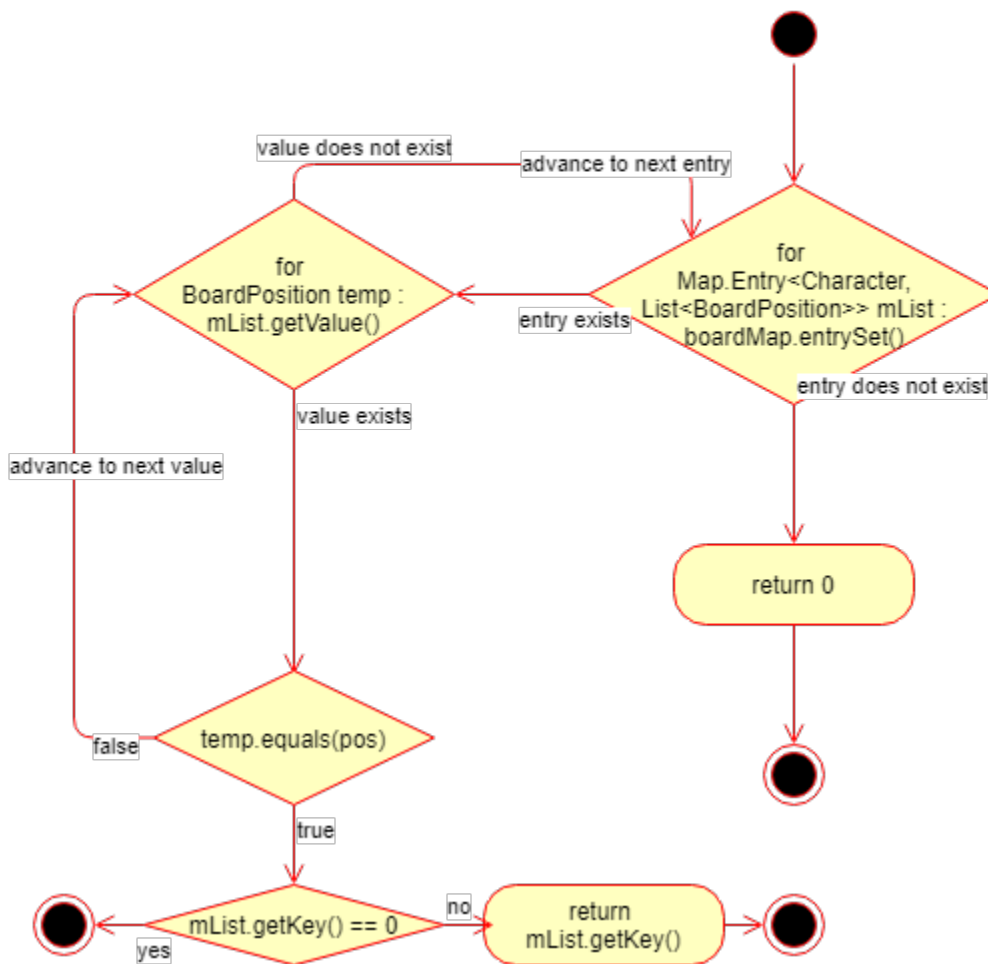
- `getNumToWin` method

`public int getNumToWin()`



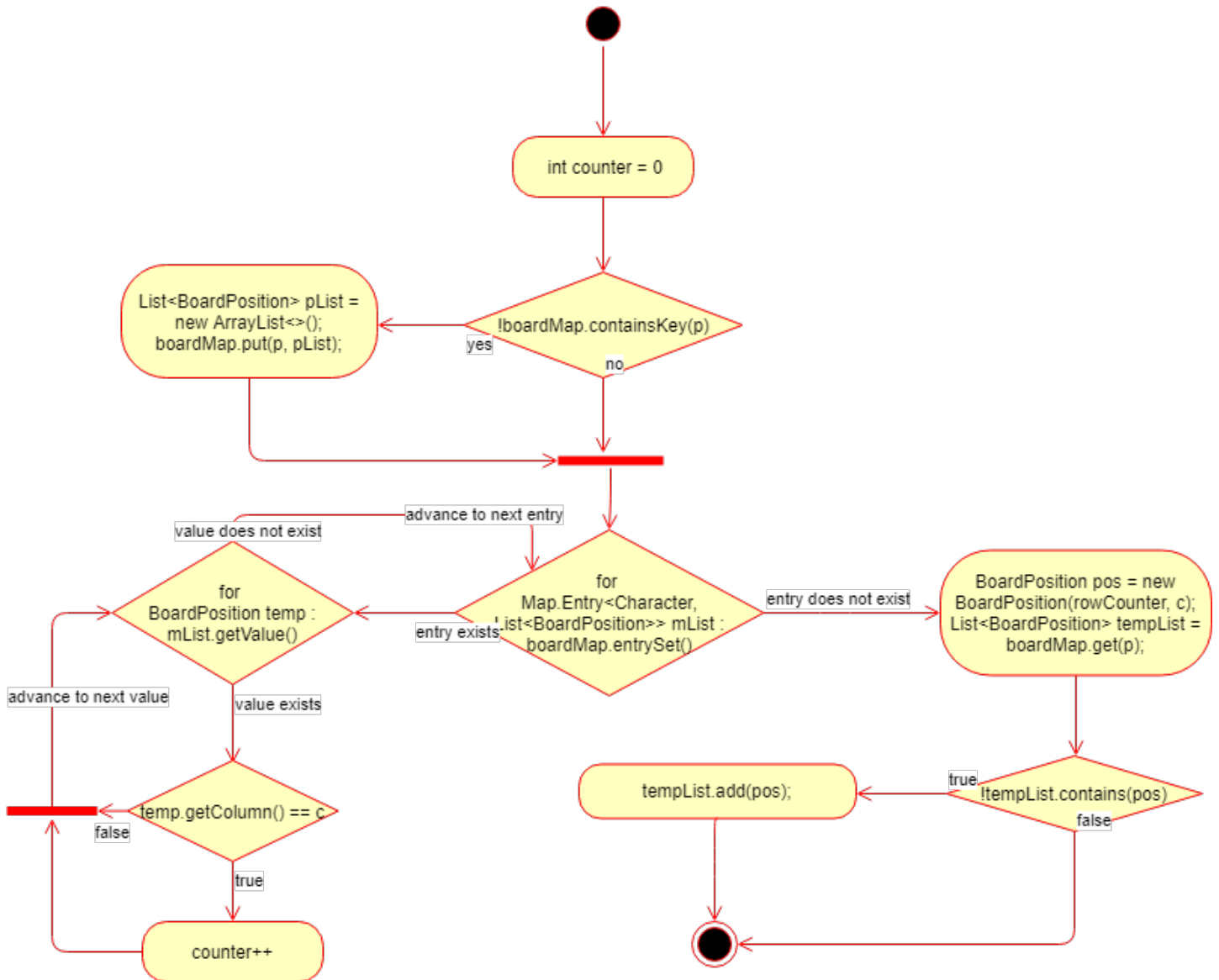
- `whatsAtPos` method


```
public class GameBoardMem extends AbsGameBoard{
    public char whatsAtPos(BoardPosition pos)
```



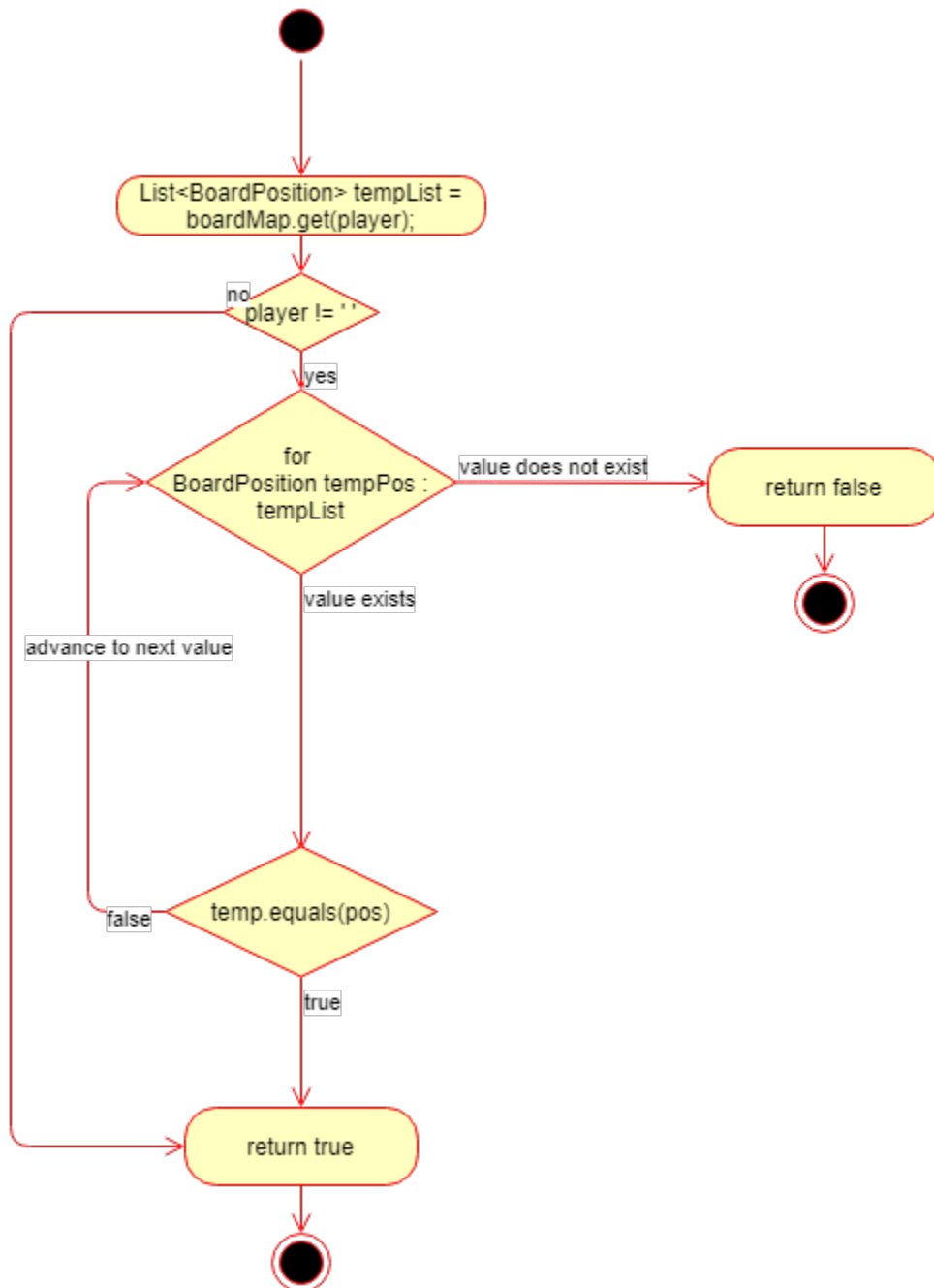
- placeToken method

```
public class GameBoardMem extends AbsGameBoard{
    public void placeToken(char p, int c)
```



- isPlayerAtPos method

```
public class GameBoardMem extends AbsGameBoard{  
public boolean isPlayerAtPos(BoardPosition pos, char player)
```



Testing

GameBoard(int rows, int columns, int numToWin)

Input: rows = 3 columns = 3 numToWin = 3	Output: State: <table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>										Reason: This test case is unique and distinct because I am testing the smallest possible row, columns, and numToWin inputs. Function Name: testConstructor_rows3_columns3_numToWin3																					
Input: rows = 100 columns = 100 numToWin = 25	Output: State: <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table> (actual column and row size is 20 times larger)																										Reason: This test case is unique and distinct because I am testing the largest possible row, columns, and numToWin inputs. Function Name: testConstructor_rows100_columns100_numToWin25					
Input: rows = 3 columns = 100 numToWin = 3	Output: State: <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> (actual column size is 10 times larger)																															Reason: This test case is unique and distinct because I am testing the largest possible column input with the smallest possible row and numToWin inputs. Function Name: testConstructor_rows3_columns100_numToWin3

--	--	--

boolean checkIfFree(int c)

<div><div><div>Input:</div><div>State:</div><div><table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table></div><div><div>c = 5</div></div></div></div>																																																																																																					<div><div>Output:</div><div>checkIfFree = true</div><div>state of the board is unchanged</div></div>	<div><div>Reason:</div><div>This test case is unique and distinct because I am testing an empty board and thus testing an empty no boundary column.</div><div>Function Name:</div><div>testCheckIfFree_EmptyBoard_Col5</div></div>

Input: State: <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table> c = 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Output: checkIfFree = false state of the board is unchanged	Reason: This test case is unique and distinct because I am testing a full board and testing the lower boundary input to make sure it returns false. Function Name: testCheckIfFree_FilledBoard_Col0
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													

Input: State: <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table> c = 9	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Output: checkIfFree = true state of the board is unchanged	Reason: This test case is unique and distinct because a partially full board in the one column (that is also a boundary input) that has a free place to ensure that checkIfFree returns true. Function Name: testCheckIfFree_PartiallyFilledBoard_EmptyCol9Row9
X	X	X	X	X	X	X	X	X																																																																																				
X	X	X	X	X	X	X	X	X	X																																																																																			
X	X	X	X	X	X	X	X	X	X																																																																																			
X	X	X	X	X	X	X	X	X	X																																																																																			
X	X	X	X	X	X	X	X	X	X																																																																																			
X	X	X	X	X	X	X	X	X	X																																																																																			
X	X	X	X	X	X	X	X	X	X																																																																																			
X	X	X	X	X	X	X	X	X	X																																																																																			
X	X	X	X	X	X	X	X	X	X																																																																																			

boolean checkHorizWin(BoardPosition pos, char p)

Input: State: (number to win = 5) <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td><td></td><td></td></tr></table> pos.getRow = 0 pos.getColumn = 4 p = 'X'																																																																																																							X	X	X	X	X				Output: checkHorizWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position is the middle of the tokens and it involves no boundary cases. Function Name: testCheckHorizWin_NonBoundary_LastPlacedMiddle
		X	X	X	X	X																																																																																																										

Input: State: (number to win = 5) <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr></table> pos.getRow = rows(10) - 1 pos.getColumn = 2 p = 'X'	X	X	X	X	X						O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	Output: checkHorizWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position is the middle of the tokens the top left boundary place. Function Name: testCheckHorizWin_TopLeft_LastPlacedMiddle
X	X	X	X	X																																																																																																		
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													

<p>Input:</p> <p>State: (number to win = 5)</p> <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr> </table> <p>pos.getRow = 0 pos.getColumn = columns(10) - 1</p> <p>p = 'X'</p>																																																																																																X	X	X	X	X	<p>Output:</p> <p>checkHorizWin = true</p> <p>state of the board is unchanged</p>	<p>Reason:</p> <p>This test case is unique and distinct because the last placed position is on the right and also on the lower bottom boundary.</p> <p>Function Name: testCheckHorizWin_BottomRight_LastPlacedRight</p>
					X	X	X	X	X																																																																																													

Input: State: (number to win = 5) <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr></table> pos.getRow = 2 pos.getColumn = columns(10) - 1 p = 'X'																																																																							X	X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	Output: checkHorizWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position is on the right it tests both the left and right boundaries by taking up the entire row. Also, the number of consecutive X's exceeds the numToWin to test. Function Name: testCheckHorizWin_R2_EntireRow_LastPlacedLeft
X	X	X	X	X	X	X	X	X	X																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													

boolean checkVertWin(BoardPosition pos, char p)

Input: State: (number to win = 5) <table><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr></table> pos.getRow = rows(10) - 1 pos.getColumn = 0 p = 'X'	X										X										X										X										X										O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	Output: checkVertWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position is on the upper left boundary of the game board in a partially filled board. Function Name: testCheckVertWin_TopLeft_C0
X																																																																																																						
X																																																																																																						
X																																																																																																						
X																																																																																																						
X																																																																																																						
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													

Input: State: (number to win = 5) <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr></table> pos.getRow = rows(10) - 1 pos.getColumn = column(10) - 1 p = 'X'										X										X										X										X										X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	Output: checkVertWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position is on the upper right boundary of the game board in a partially filled board. Function Name: testCheckVertWin_TopRight_C9
									X																																																																																													
									X																																																																																													
									X																																																																																													
									X																																																																																													
									X																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													

Input: State: (number to win = 5) <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr></table> pos.getRow = numToWin(5) - 1 pos.getColumn = column(10) - 1 p = 'X'																																																												X										X										X										X										X	Output: checkVertWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position right boundary and it involves the lower right boundary. Function Name: testCheckVertWin_BottomRight_C9
									X																																																																																													
									X																																																																																													
									X																																																																																													
									X																																																																																													
									X																																																																																													

Input: State: (number to win = 5) <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr></table> pos.getRow = 7 pos.getColumn = 4 p = 'X'																									X										X										X										X										X						O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	Output: checkVertWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position involves no boundary cases and is approximately in the center. Function Name: testCheckVertWin_MiddleCenter_C4
				X																																																																																																		
				X																																																																																																		
				X																																																																																																		
				X																																																																																																		
				X																																																																																																		
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													

boolean checkDiagWin(BoardPosition pos, char p)

Input: State: (number to win = 5) <table><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td></td><td></td><td></td><td></td><td></td></tr></table> pos.getRow = rows(10) - 1 pos.getColumn = 0 p = 'X'	X										O	X									O	O	X								O	O	O	X							O	O	O	O	X						O	O	O	O	O						O	O	O	O	O						O	O	O	O	O						O	O	O	O	O						O	O	O	O	O						Output: checkDiagWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position is the top left boundary position, and the function must check in the NW/SE direction. Function Name: testCheckDiagWin_LastPlacedTopLeft_C0_NWSE
X																																																																																																						
O	X																																																																																																					
O	O	X																																																																																																				
O	O	O	X																																																																																																			
O	O	O	O	X																																																																																																		
O	O	O	O	O																																																																																																		
O	O	O	O	O																																																																																																		
O	O	O	O	O																																																																																																		
O	O	O	O	O																																																																																																		
O	O	O	O	O																																																																																																		

Input: State: (number to win = 5) <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr></table> pos.getRow = rows(10)-1 pos.getColumn = columns(10) -1 p = 'X'										X									X	O								X	O	O							X	O	O	O						X	O	O	O	O						O	O	O	O	O						O	O	O	O	O						O	O	O	O	O						O	O	O	O	O						O	O	O	O	O	Output: checkDiagWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position is the top right boundary position, and the function must check in the NE/SW direction. Function Name: testCheckDiagWin_LastPlacedTopRight_C9_NESW
									X																																																																																													
								X	O																																																																																													
							X	O	O																																																																																													
						X	O	O	O																																																																																													
					X	O	O	O	O																																																																																													
					O	O	O	O	O																																																																																													
					O	O	O	O	O																																																																																													
					O	O	O	O	O																																																																																													
					O	O	O	O	O																																																																																													
					O	O	O	O	O																																																																																													

Input: State: (number to win = 5) <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td>X</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td>O</td><td>X</td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td>O</td><td>O</td><td>X</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td></tr></table> pos.getRow = rows(10)-1 pos.getColumn = 0 p = 'X'																																																								X										O	X									O	O	X								O	O	O	X							O	O	O	O	X	Output: checkDiagWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position is the bottom right boundary position, and the function must check in the NW/SE direction. Function Name: testCheckDiagWin_LastPlacedBottomRight_C9_NWSE
					X																																																																																																	
					O	X																																																																																																
					O	O	X																																																																																															
					O	O	O	X																																																																																														
					O	O	O	O	X																																																																																													

<p>Input:</p> <p>State: (number to win = 5)</p> <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td>X</td><td>O</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td>X</td><td>O</td><td>O</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>X</td><td>O</td><td>O</td><td>O</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td><td></td><td></td><td></td><td></td><td></td></tr> </table> <p>pos.getRow = 0</p> <p>pos.getColumn = 0</p> <p>p = 'X'</p>																																																							X									X	O								X	O	O							X	O	O	O						X	O	O	O	O						<p>Output:</p> <p>checkDiagWin = true</p> <p>state of the board is unchanged</p>	<p>Reason:</p> <p>This test case is unique and distinct because the last placed position is the bottom left boundary position, and the function must check in the NE/SW direction.</p> <p>Function Name:</p> <p>testCheckDiagWin_LastPlacedBottomLeft_C0_NESW</p>
				X																																																																																																		
			X	O																																																																																																		
		X	O	O																																																																																																		
	X	O	O	O																																																																																																		
X	O	O	O	O																																																																																																		

Input: State: (number to win = 5) <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td></td><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td></td><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr></table> pos.getRow = 0 pos.getColumn = 0 p = 'X'										X									X	O								X	O	O							X	O	O	O						X	O	O	O	O					X	O	O	O	O	O				X	O	O	O	O	O	O			X	O	O	O	O	O	O	O		X	O	O	O	O	O	O	O	O	X	O	O	O	O	O	O	O	O	O	Output: checkDiagWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position is the bottom left boundary position, and the top right boundary is also involved, and the function must check in the NE/SW direction. Function Name: testCheckDiagWin_LastPlacedLeftMost_C0_NESW_TopRight
									X																																																																																													
								X	O																																																																																													
							X	O	O																																																																																													
						X	O	O	O																																																																																													
					X	O	O	O	O																																																																																													
				X	O	O	O	O	O																																																																																													
			X	O	O	O	O	O	O																																																																																													
		X	O	O	O	O	O	O	O																																																																																													
	X	O	O	O	O	O	O	O	O																																																																																													
X	O	O	O	O	O	O	O	O	O																																																																																													

<p>Input:</p> <p>State: (number to win = 5)</p> <table border="1"> <tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>O</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td><td></td></tr> <tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td></tr> <tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td></tr> <tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td><td></td></tr> <tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td></tr> </table> <p>pos.getRow = 0</p> <p>pos.getColumn = columns(10)-1</p> <p>p = 'X'</p>	X										O	X									O	O	X								O	O	O	X							O	O	O	O	X						O	O	O	O	O	X					O	O	O	O	O	O	X				O	O	O	O	O	O	O	X			O	O	O	O	O	O	O	O	X		O	O	O	O	O	O	O	O	O	X	<p>Output:</p> <p>checkDiagWin = true</p> <p>state of the board is unchanged</p>	<p>Reason:</p> <p>This test case is unique and distinct because the last placed position is the bottom right boundary position, and the top left boundary position is also involved, and the function must check in the NW/SE direction.</p> <p>Function Name: testCheckDiagWin_LastPlacedRightMost_C9_NWSE_To pLeft</p>
X																																																																																																						
O	X																																																																																																					
O	O	X																																																																																																				
O	O	O	X																																																																																																			
O	O	O	O	X																																																																																																		
O	O	O	O	O	X																																																																																																	
O	O	O	O	O	O	X																																																																																																
O	O	O	O	O	O	O	X																																																																																															
O	O	O	O	O	O	O	O	X																																																																																														
O	O	O	O	O	O	O	O	O	X																																																																																													

Input: State: (number to win = 5) <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>X</td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr></table> pos.getRow = 6 pos.getColumn = 5 p = 'X'														X							O	O	O	O	X						O	O	O	O	O	X					O	O	O	O	O	O	X				O	O	O	O	O	O	O	X			O	O	O	O	O	O	O	O			O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	Output: checkDiagWin = true state of the board is unchanged	Reason: This test case is unique and distinct because the last placed position is a nonboundary in the middle of a diagonal of X's, and none of the X's touch a boundary, and the function must check in the NW/SE direction. Function Name: testCheckDiagWin_LastPlacedRightMost_C9_NWSE_To pLeft
			X																																																																																																			
O	O	O	O	X																																																																																																		
O	O	O	O	O	X																																																																																																	
O	O	O	O	O	O	X																																																																																																
O	O	O	O	O	O	O	X																																																																																															
O	O	O	O	O	O	O	O																																																																																															
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													
O	O	O	O	O	O	O	O	O	O																																																																																													

boolean checkTie()

Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																																																																																																					Output: checkTie = false state of the board is unchanged	Reason: This test case is unique and distinct we are testing an empty board for a tie. Function Name: testCheckTie_EmptyBoard

Input:	Output:	Reason:																																																																																																				
State:	checkTie = true	This test case is unique and distinct we are testing a full board for a tie.																																																																																																				
<table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	state of the board is unchanged	Function Name: testCheckTie_FullBoard
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													

Input:	Output:	Reason:																																																																																																				
State:	checkTie = false	This test case is unique and distinct we are testing a nearly full board for a tie and this one empty spot is a bottom left boundary case.																																																																																																				
<table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	state of the board is unchanged	Function Name: testCheckTie_PartiallyFullBoard_EmptyPlaceTopRight
X	X	X	X	X	X	X	X	X																																																																																														
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													

Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																																																																																											X										Output: checkTie = false state of the board is unchanged	Reason: This test case is unique and distinct we are testing a nearly empty board for a tie and this one occupied spot is a bottom left boundary case. Function Name: testCheckTie_PartiallyFullBoard_OccupiedSpace_BottomLeft
X																																																																																																						

char whatsAtPos(BoardPosition pos)

Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> pos.getRow = rows(10)-1 pos.getColumn = columns(10)-1																																																																																																					Output: whatsAtPos = ‘ ‘ state of the board is unchanged	Reason: This test case is unique and distinct because it tests the top right boundary case on an empty board. Function Name: testWhatsAtPos_EmptyBoar d_TopRight

Input: State: <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table> pos.getRow = rows(10)-1 pos.getColumn = columns(10)-1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Output: whatsAtPos = 'X' state of the board is unchanged	Reason: This test case is unique and distinct because it tests the top right boundary case on a full board. Function Name: testWhatsAtPos_FilledBoard_TopRight
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													

Input: State: <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table> pos.getRow = 0 pos.getColumn = 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Output: whatsAtPos = 'X' state of the board is unchanged	Reason: This test case is unique and distinct because it tests the bottom left boundary case on a full board. Function Name: testWhatsAtPos_FilledBoard_BottomLeft
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													

Input: State: <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table> pos.getRow = rows(10)-1 pos.getColumn = columns(10)-1	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Output: whatsAtPos = ‘ ‘ state of the board is unchanged	Reason: This test case is unique and distinct because it tests the top right boundary case on a partially full board. Function Name: testWhatsAtPos_PartiallyFill edBoard_EmptyTopRight
X	X	X	X	X	X	X	X	X																																																																																														
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																													

<div><div>Input:</div><div>State:</div><div><table><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table></div><div><div>pos.getRow = rows(10)-1</div><div>pos.getColumn = 0</div></div></div>	O										O										O										O										O										O										O										O										O										O										<div><div>Output:</div><div>whatsAtPos = ‘O‘</div><div>state of the board is unchanged</div></div>	<div><div>Reason:</div><div>This test case is unique and distinct because it tests the top left boundary case on a partially filled board.</div></div> <div><div>Function Name:</div><div>testWhatsAtPos_PartiallyFilledBoard_OccupiedTopLeft</div></div>
O																																																																																																						
O																																																																																																						
O																																																																																																						
O																																																																																																						
O																																																																																																						
O																																																																																																						
O																																																																																																						
O																																																																																																						
O																																																																																																						
O																																																																																																						

boolean isPlayerAtPos(BoardPosition pos, char player)

<div><div><div>Input:</div><div>State:</div><div><table><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table></div><div><div>pos.getRow = rows(10)-1</div><div>pos.getColumn = 0</div><div>player = 'O'</div></div></div></div>	O									X									O									X									O									X									O									X									O									X									<div><div><div>Output:</div><div>isPlayerAtPos = true</div><div>state of the board is unchanged</div></div></div>	<div><div><div>Reason:</div><div>This test case is unique and distinct because it tests the top left boundary for an occupied O when the column has alternating X's and O's on a partially full board.</div></div><div><div>Function Name:</div><div>testIsPlayerAtPos_PartiallyFilledBoard_OccupiedTopLeft</div></div></div>
O																																																																																												
X																																																																																												
O																																																																																												
X																																																																																												
O																																																																																												
X																																																																																												
O																																																																																												
X																																																																																												
O																																																																																												
X																																																																																												

Input:	Output:	Reason:																																																																																																			
State:	isPlayerAtPos = true	This test case is unique and distinct because it tests the top right boundary for an empty space when the rest of the gameboard is full of alternating X's and O's.																																																																																																			
<table><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	O		O	O	O	O	O	O	O	O		X	X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O	X	X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O	X	X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O	X	X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O	X	X	X	X	X	X	X	X	X	X
O	O	O	O	O	O	O	O	O																																																																																													
X	X	X	X	X	X	X	X	X	X																																																																																												
O	O	O	O	O	O	O	O	O	O																																																																																												
X	X	X	X	X	X	X	X	X	X																																																																																												
O	O	O	O	O	O	O	O	O	O																																																																																												
X	X	X	X	X	X	X	X	X	X																																																																																												
O	O	O	O	O	O	O	O	O	O																																																																																												
X	X	X	X	X	X	X	X	X	X																																																																																												
O	O	O	O	O	O	O	O	O	O																																																																																												
X	X	X	X	X	X	X	X	X	X																																																																																												
pos.getRow = rows(10)-1		Function Name: testIsPlayerAtPos_PartiallyFilledBoard_EmptyTopRight																																																																																																			
pos.getColumn = column(10)-1																																																																																																					
player = ' '																																																																																																					

Input:	Output:	Reason:																																																																																																			
State:	isPlayerAtPos = true	This test case is unique and distinct because it tests the top left boundary for an occupied X when the entire board is full of alternating X's and O's.																																																																																																			
<table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr></table>	X		X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O	X	X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O	X	X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O	X	X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O	X	X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O
X	X	X	X	X	X	X	X	X	X																																																																																												
O	O	O	O	O	O	O	O	O	O																																																																																												
X	X	X	X	X	X	X	X	X	X																																																																																												
O	O	O	O	O	O	O	O	O	O																																																																																												
X	X	X	X	X	X	X	X	X	X																																																																																												
O	O	O	O	O	O	O	O	O	O																																																																																												
X	X	X	X	X	X	X	X	X	X																																																																																												
O	O	O	O	O	O	O	O	O	O																																																																																												
X	X	X	X	X	X	X	X	X	X																																																																																												
O	O	O	O	O	O	O	O	O	O																																																																																												
pos.getRow = rows(10)-1 pos.getColumn = 0 player = 'X'		Function Name: testIsPlayerAtPos_FilledBoard_TopLeft																																																																																																			

<div><div><div>Input:</div><div>State:</div><div><table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table></div><div><div>pos.getRow = 0</div><div>pos.getColumn = columns(10) - 1</div><div>player = ' '</div></div></div></div>																																																																																																					<div><div>Output:</div><div>isPlayerAtPos = true</div><div>state of the board is unchanged</div></div>	<div><div>Reason:</div><div>This test case is unique and distinct because it tests the top left boundary for an empty space on an empty board.</div><div>Function Name:</div><div>testIsPlayerAtPos_EmptyBoard_TopLeft</div></div>

void placeToken(char p, int c)

<div><div>Input:</div><div>State:</div><div><table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table></div><div><div>c = 0</div><div>p = 'X'</div></div></div>																																																																																																					<div><div>Output:</div><div>State:</div><div><table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table></div></div>																																																																																																					X										<div><div>Reason:</div><div>This test case is unique and distinct because it tests the bottom left boundary on an empty board of a nonboundary touching gameboard.</div><div><div>Function Name:</div><div>testPlaceToken_BottomLeft_rows10_columns10</div></div></div>
X																																																																																																																																																																																																																				

Input: State: <table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table> c = 2 p = 'X'										Output: State: <table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td>x</td></tr></table>									x	Reason: This test case is unique and distinct because it tests the bottom right boundary on an empty board of minimum size. Function Name: testPlaceTo ken_Bottom Righ_rows3 _columns3
		x																		

Input:	Output:	Reason:																																																																																																																																																																																																								
State:	State:	This test case is unique and distinct because it tests the top right boundary on a partially full board of a max sized gameboard.																																																																																																																																																																																																								
<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td></tr></table> <p>(actual size of gameboard is 10x larger)</p> <p>c = columns(10) - 1</p> <p>p = 'X'</p>																				O										X										O										X										O										X										O										X										O	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td></tr></table> <p>(actual size of gameboard is 10x larger)</p>										X										O										X										O										X										O										X										O										X										O	Function Name: testPlaceToken_TopRight _rows100_columns100
									O																																																																																																																																																																																																	
									X																																																																																																																																																																																																	
									O																																																																																																																																																																																																	
									X																																																																																																																																																																																																	
									O																																																																																																																																																																																																	
									X																																																																																																																																																																																																	
									O																																																																																																																																																																																																	
									X																																																																																																																																																																																																	
									O																																																																																																																																																																																																	
									X																																																																																																																																																																																																	
									O																																																																																																																																																																																																	
									X																																																																																																																																																																																																	
									O																																																																																																																																																																																																	
									X																																																																																																																																																																																																	
									O																																																																																																																																																																																																	
									X																																																																																																																																																																																																	
									O																																																																																																																																																																																																	
									X																																																																																																																																																																																																	
									O																																																																																																																																																																																																	

<p>Input:</p> <p>State:</p> <table border="1" data-bbox="110 241 699 625"> <tr><td></td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td></tr> <tr><td>X</td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td></tr> <tr><td>X</td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td></tr> <tr><td>X</td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td></tr> <tr><td>X</td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td></tr> </table> <p>c = 0</p> <p>p = 'X'</p>				O			X			O			X			O			X			O			X			O			<p>Output:</p> <p>State:</p> <table border="1" data-bbox="732 241 1321 625"> <tr><td>X</td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td></tr> <tr><td>X</td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td></tr> <tr><td>X</td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td></tr> <tr><td>X</td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td></tr> <tr><td>X</td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td></tr> </table>	X			O			X			O			X			O			X			O			X			O			<p>Reason: This test case is unique and distinct because it tests the top left boundary on a nearly full column where the gameboard is has the minimum size number of columns.</p> <p>Function Name: testPlaceToken_TopLeft_rows10_columns_3</p>
O																																																														
X																																																														
O																																																														
X																																																														
O																																																														
X																																																														
O																																																														
X																																																														
O																																																														
X																																																														
O																																																														
X																																																														
O																																																														
X																																																														
O																																																														
X																																																														
O																																																														
X																																																														
O																																																														

Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>O</td><td></td><td></td><td></td><td></td></tr></table> c = columns(10)/2 = 5 p = 'O'																																																																		X										O										X										O					Output: State: <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td><td></td><td></td><td></td></tr></table>																																																									O										X										O										X										O				Reason: This test case is unique and distinct because it tests the center row and center column on a partially full board of a nonboundary touching gameboard size. Function Name: testPlaceToken_Row5_Column5
					X																																																																																																																																																																																																					
					O																																																																																																																																																																																																					
					X																																																																																																																																																																																																					
					O																																																																																																																																																																																																					
						O																																																																																																																																																																																																				
						X																																																																																																																																																																																																				
						O																																																																																																																																																																																																				
						X																																																																																																																																																																																																				
						O																																																																																																																																																																																																				

