Julio Reyes

Professor Murali Sitaraman

CPSC 2150

February 7th 2021

Project Report

Requirements Analysis

Functional Requirements

1) As a user, I can view the current board, so that I know how the game is going.
2) As a user, I can enter a column position, so that I can place my piece in a specific position.
3) As a user, I can view the results of the game, so that I know who has won.
4) As a user, I can select how many rows I want my gameboard to have
5) As a user, I can select how many columns I want my gameboard to have
6) As a user, I can select how many players I want to play in the game
7) As a user, I can select my own unique player token
8) As a user, I can select whether I want a fast vs memory efficient game
9) As a user, I can be notified when I try to place my piece in an already filled column, so that I know an illegal move has been made.

10) As a user, I can have the option to play again after finishing a game, so that I can start a new game if I want to.
11) As a user, I can have the option to end the program after the game has finished, so that I can stop playing if I want to.
12) As a user, I can see when the game has ended in a tie, so that I know that no one has won.
13) As a user, I can see when a player has won due to placing five tokens in a row horizontally
14) As a user, I can see when a player has won due to placing five tokens in a row vertically
15) As a user, I can see when a player has won due to placing five tokens in a row diagonally
16) As a user, I can be given the option to place my token after my opponent's turn
17) As a user, I can be notified whenever a column I've chosen is already full.
18) As a user, I can be notified whenever I make a selection that is out of the bounds of the game board.
19) As a user, I can be notified if I choose a token that has already been chosen by another player

Non Functional Requirements

1) Must run on the Clemson School of Computing server.
2) Must be in Java.
3) Need to create UML class diagrams.
4) Need to create UML activity diagrams.

5) Need to create contracts for each method in my classes.
6) Create javadoc comments, specifying parameters, invariants, etc.
7) Game Board must be of user specified size
8) The bottom left of the board has coordinates [0, 0] and the top right of the board has coordinates [5, 8], depending on how many rows and columns the user wants to have (in this case it would be 6 rows and 9 columns.

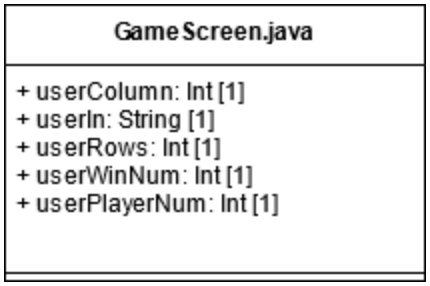
Deployment

Use “make” to compile all of the provided files. “make run” to run the actual program. Finally, “make clean” can be used after running to remove any compiled class files

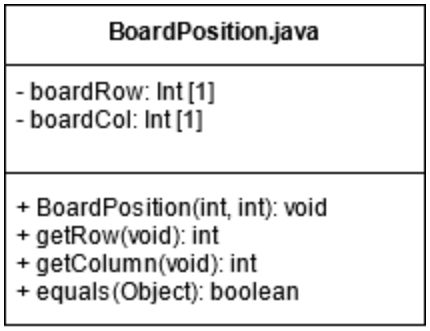
Design

UML Class Diagrams

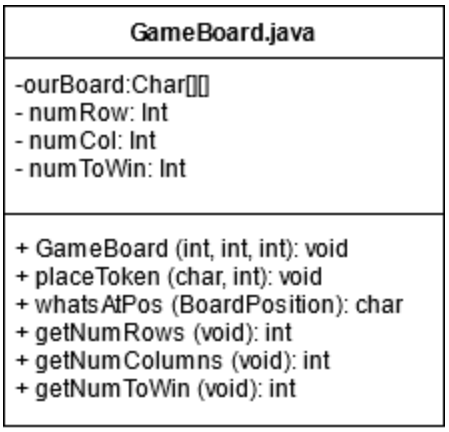
`GameScreen.java`



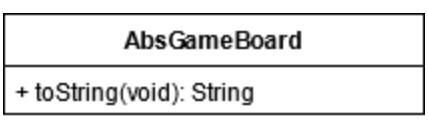
BoardPosition.java



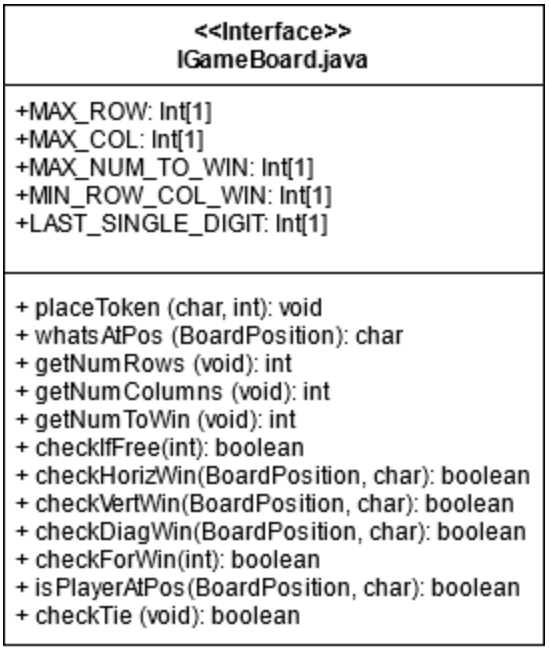
GameBoard.java



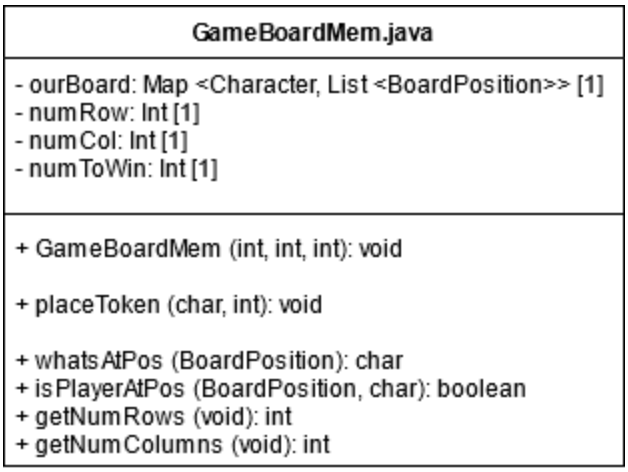
AbsGameBoard.java



IGameBoard.java



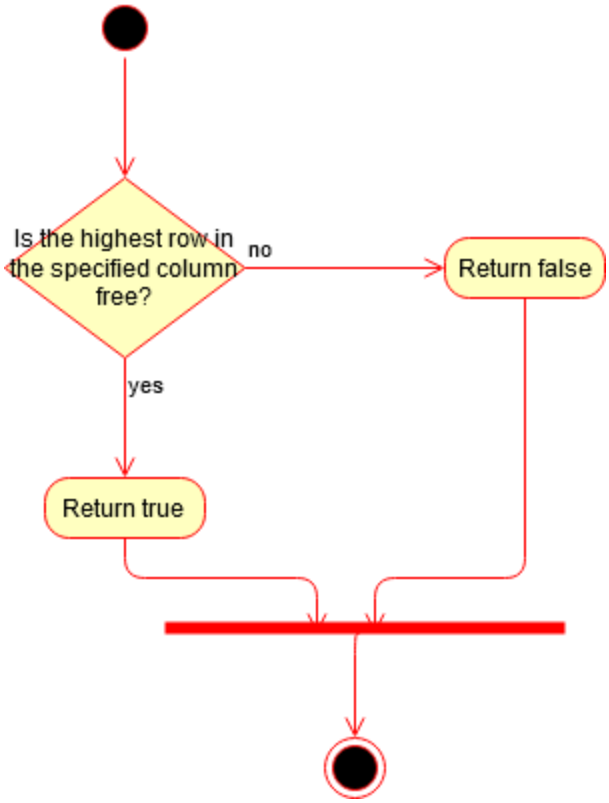
GameBoardMem.java



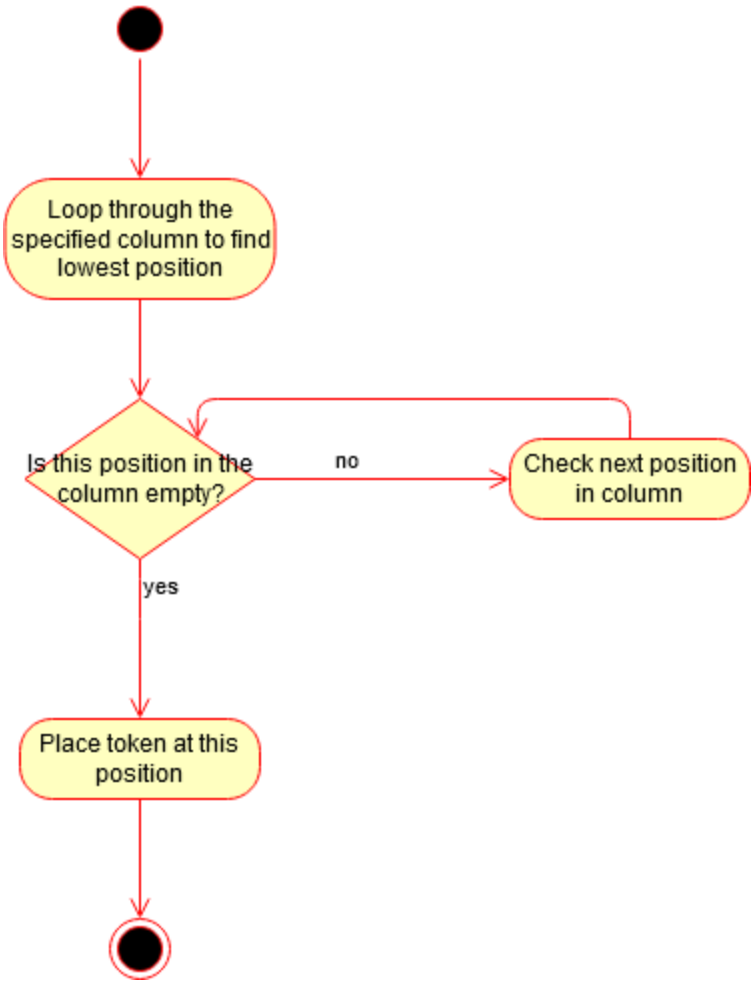
UML Activity Diagrams

`GameBoard.java - GameBoard()`

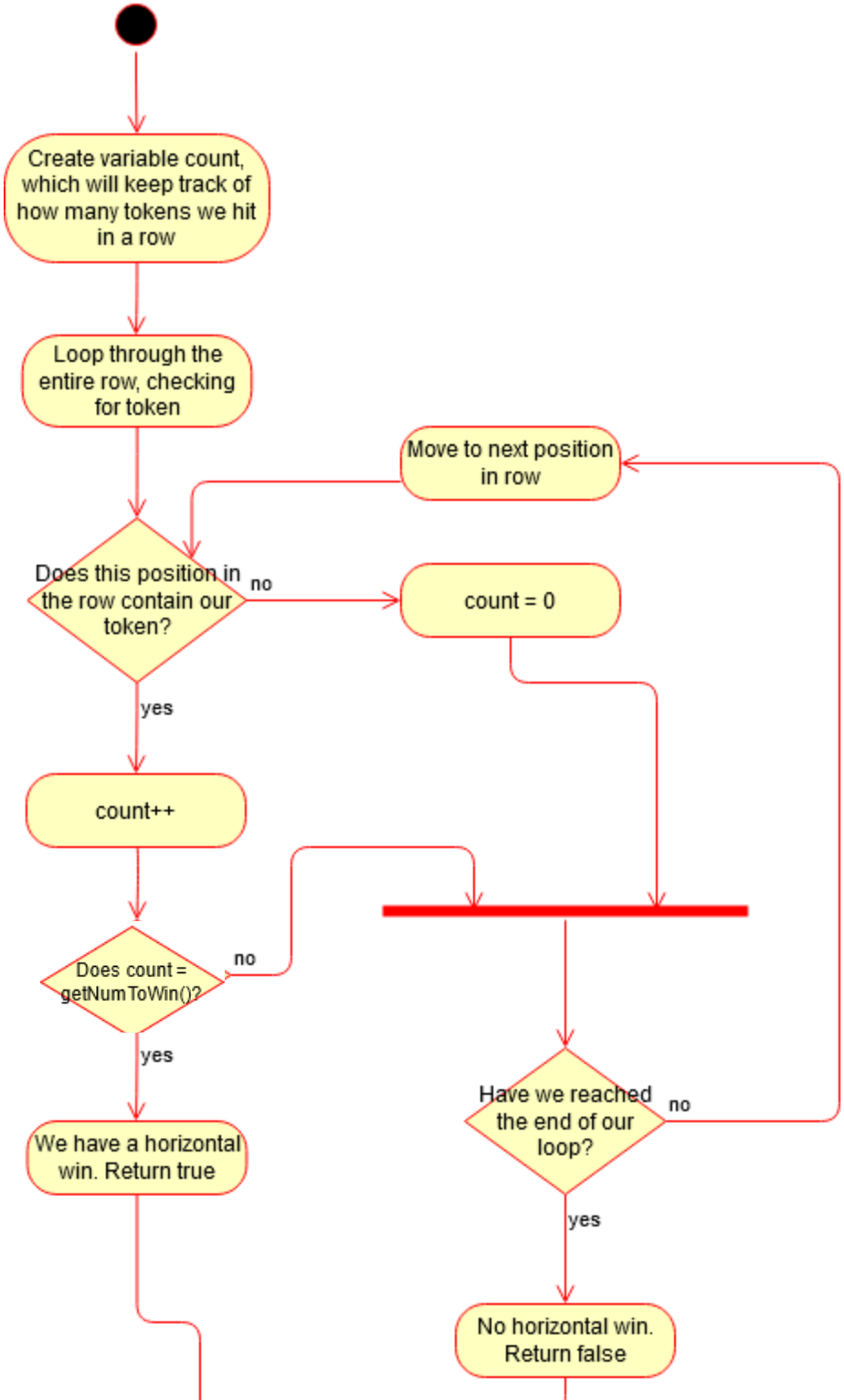
`IGameBoard - checkIfFree()`



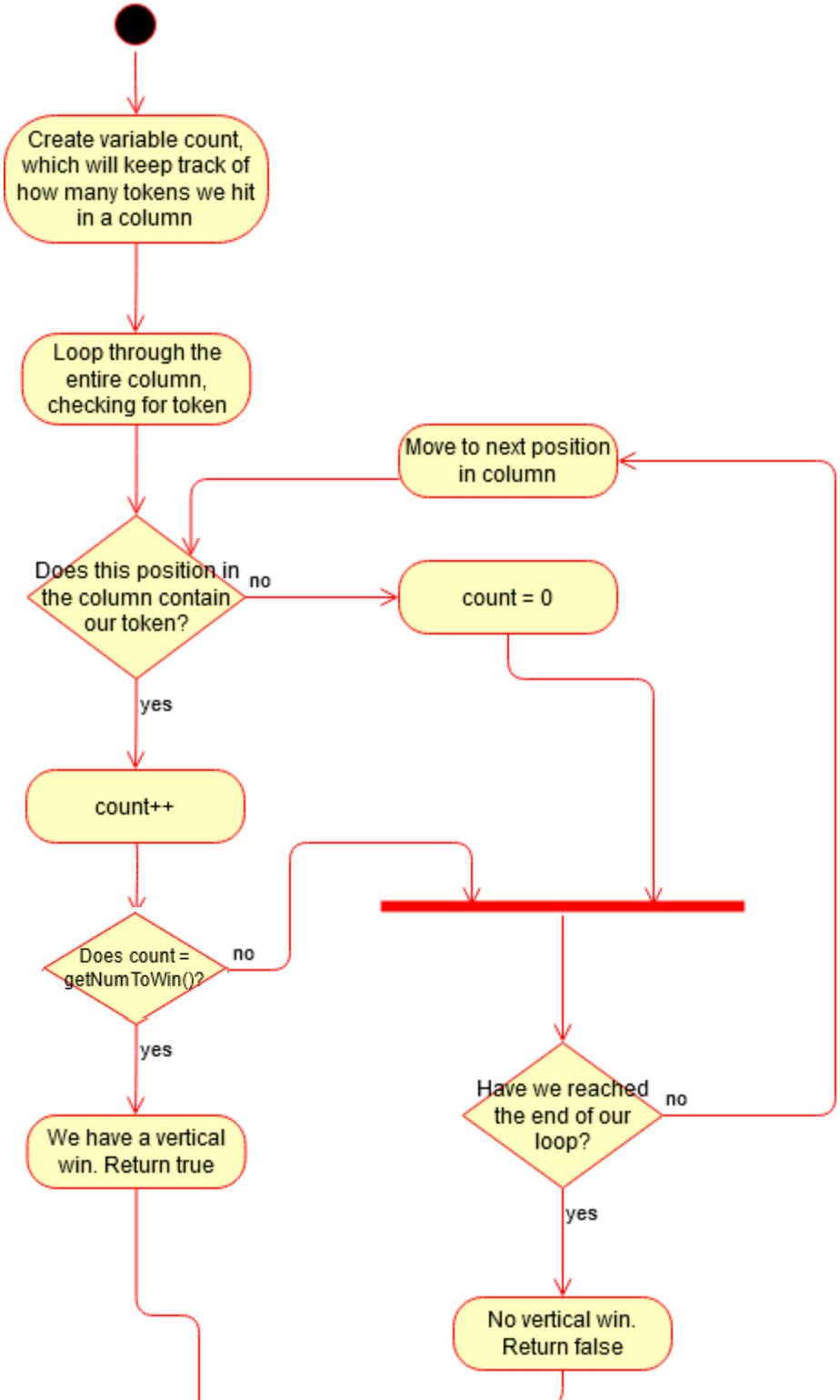
`GameBoard - placeToken()`



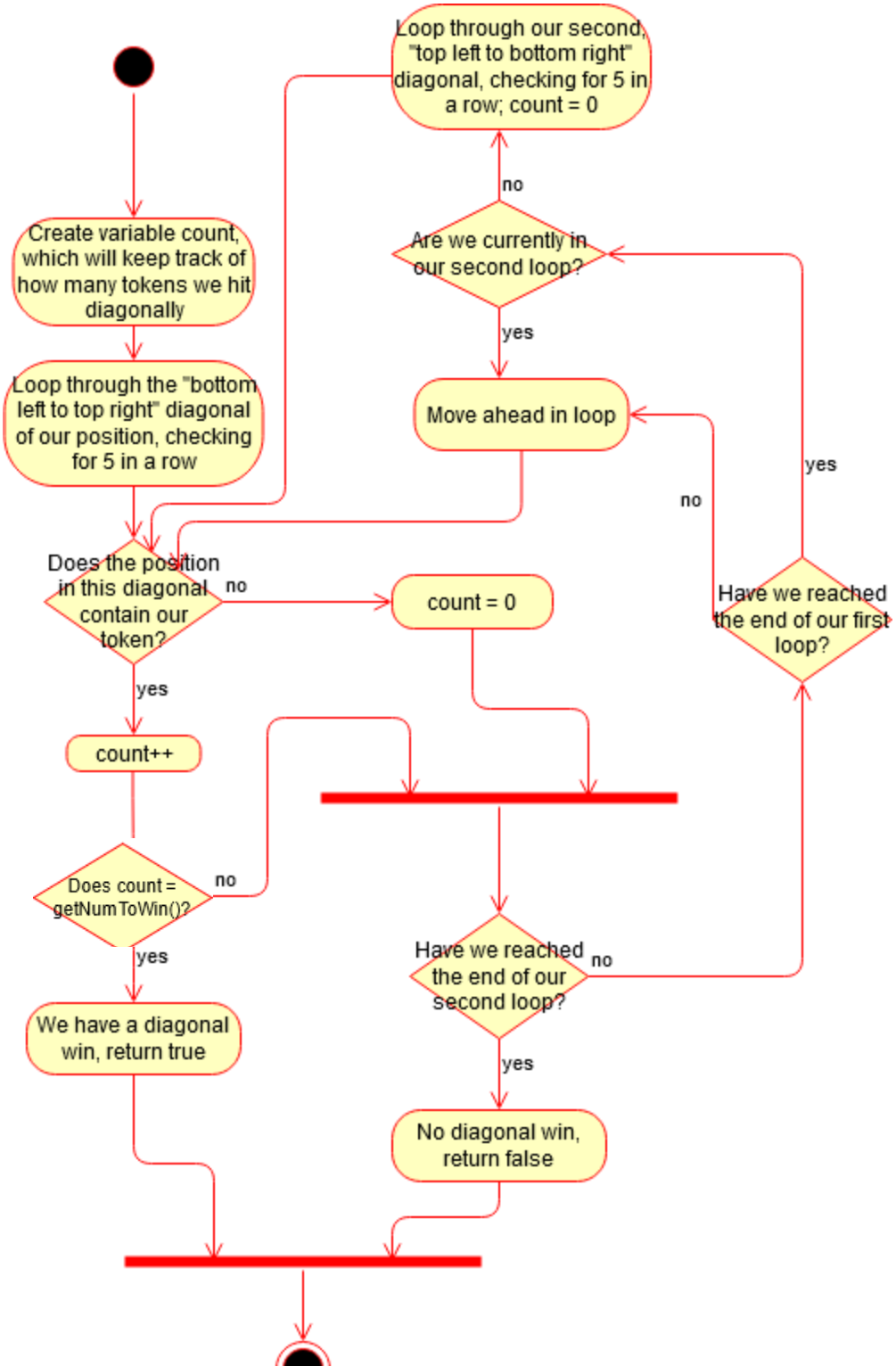
IGameBoard - checkHorizWin()



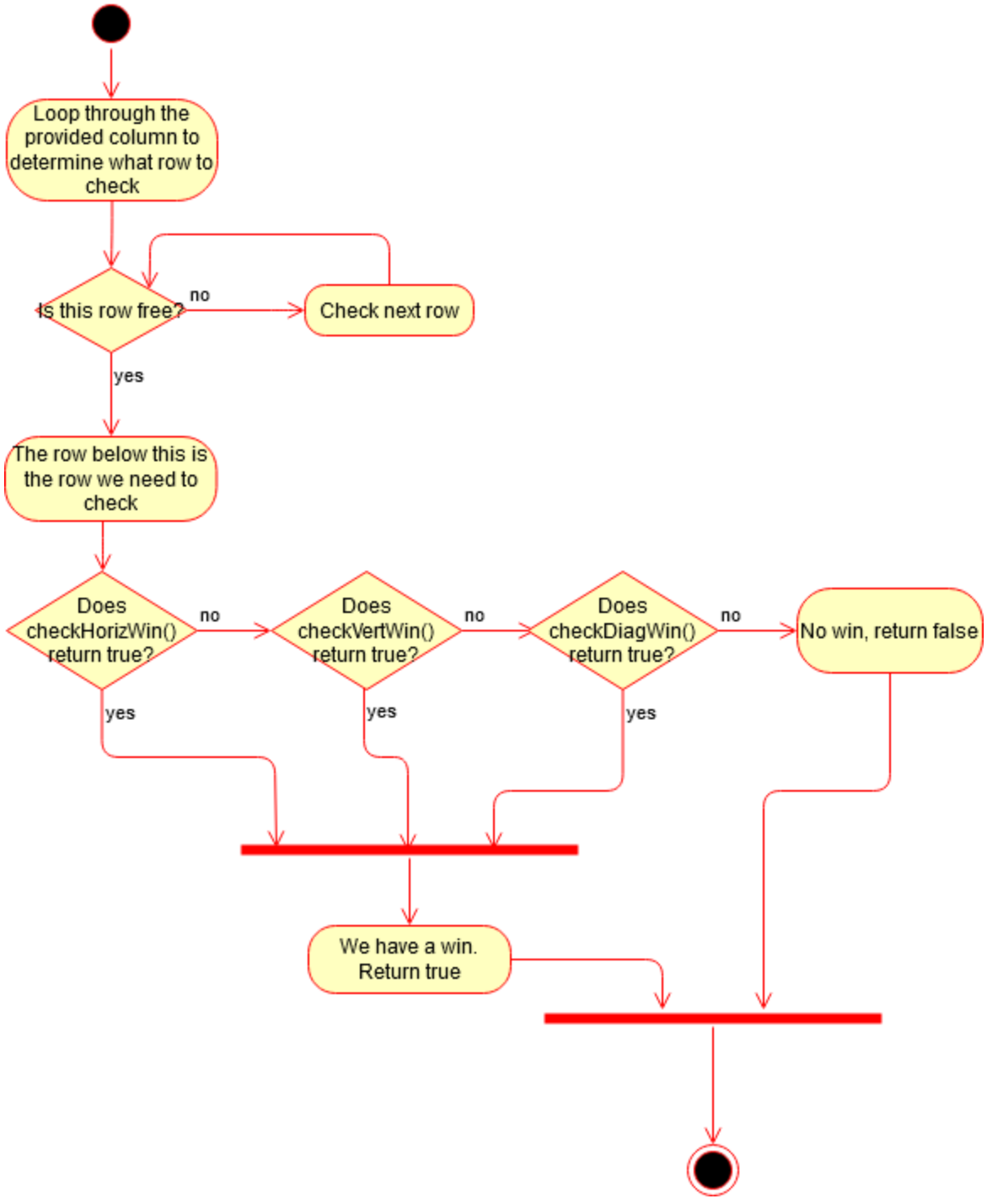
IGameBoard - checkVertWin()



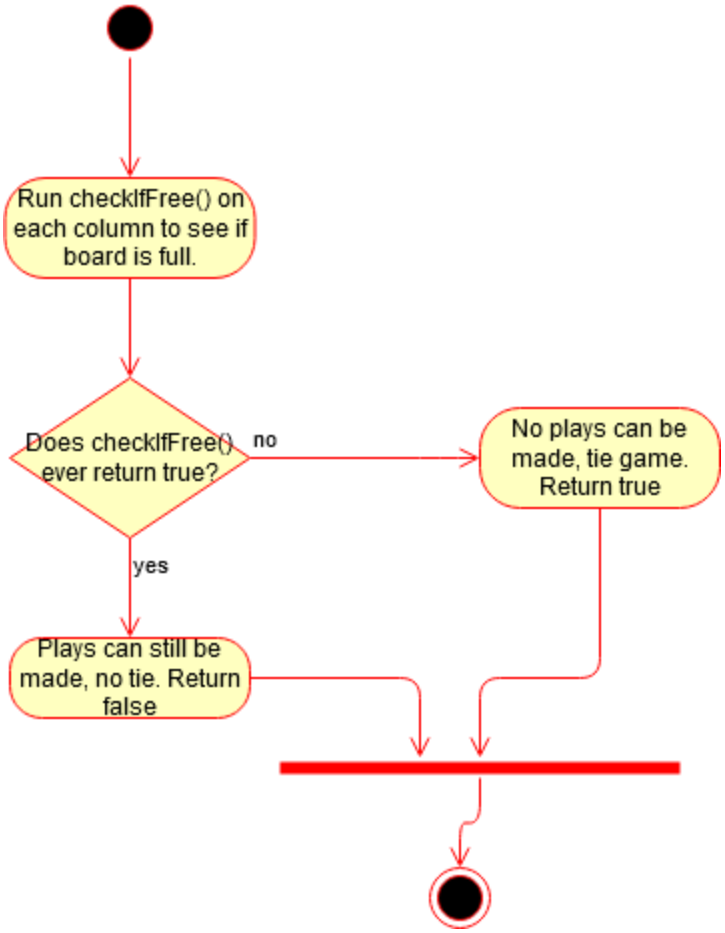
IGameBoard - checkDiagWin()



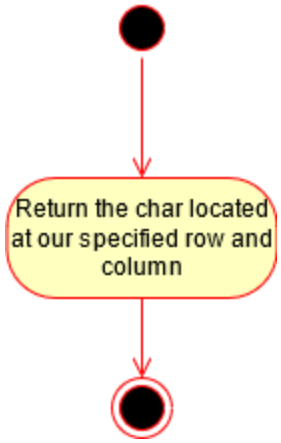
IGameBoard - checkForWin()



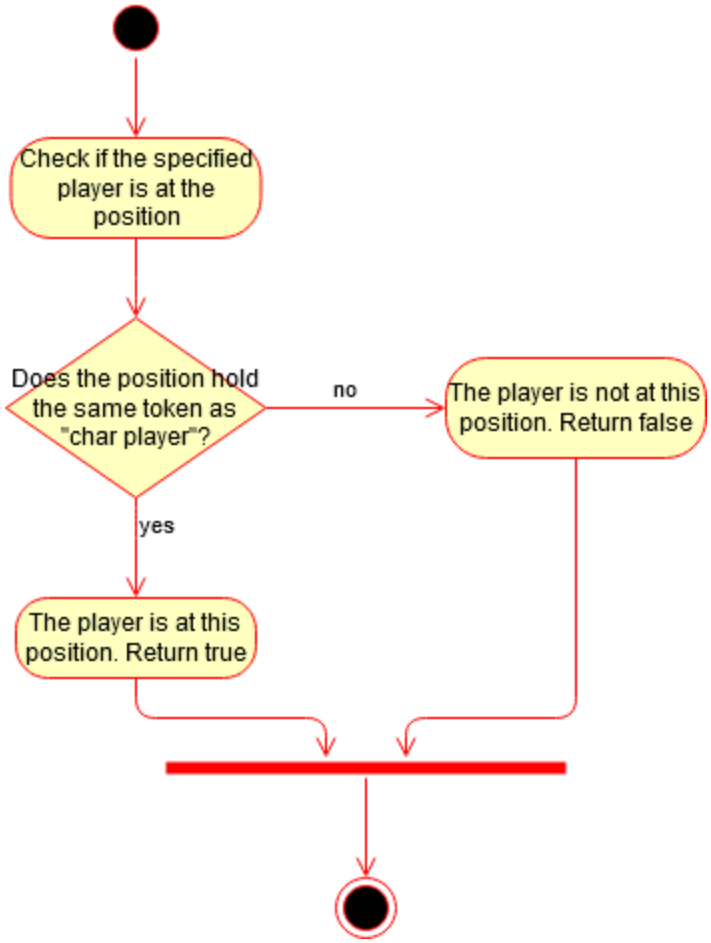
`IGameboard - checkTie()`



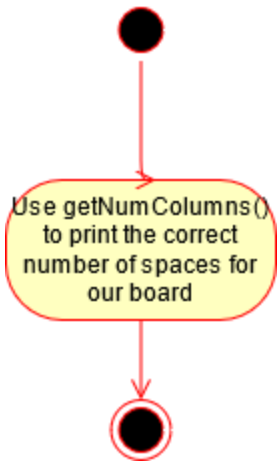
`Gameboard - whatsAtPos()`



IGameBoard - isPlayerAtPos()

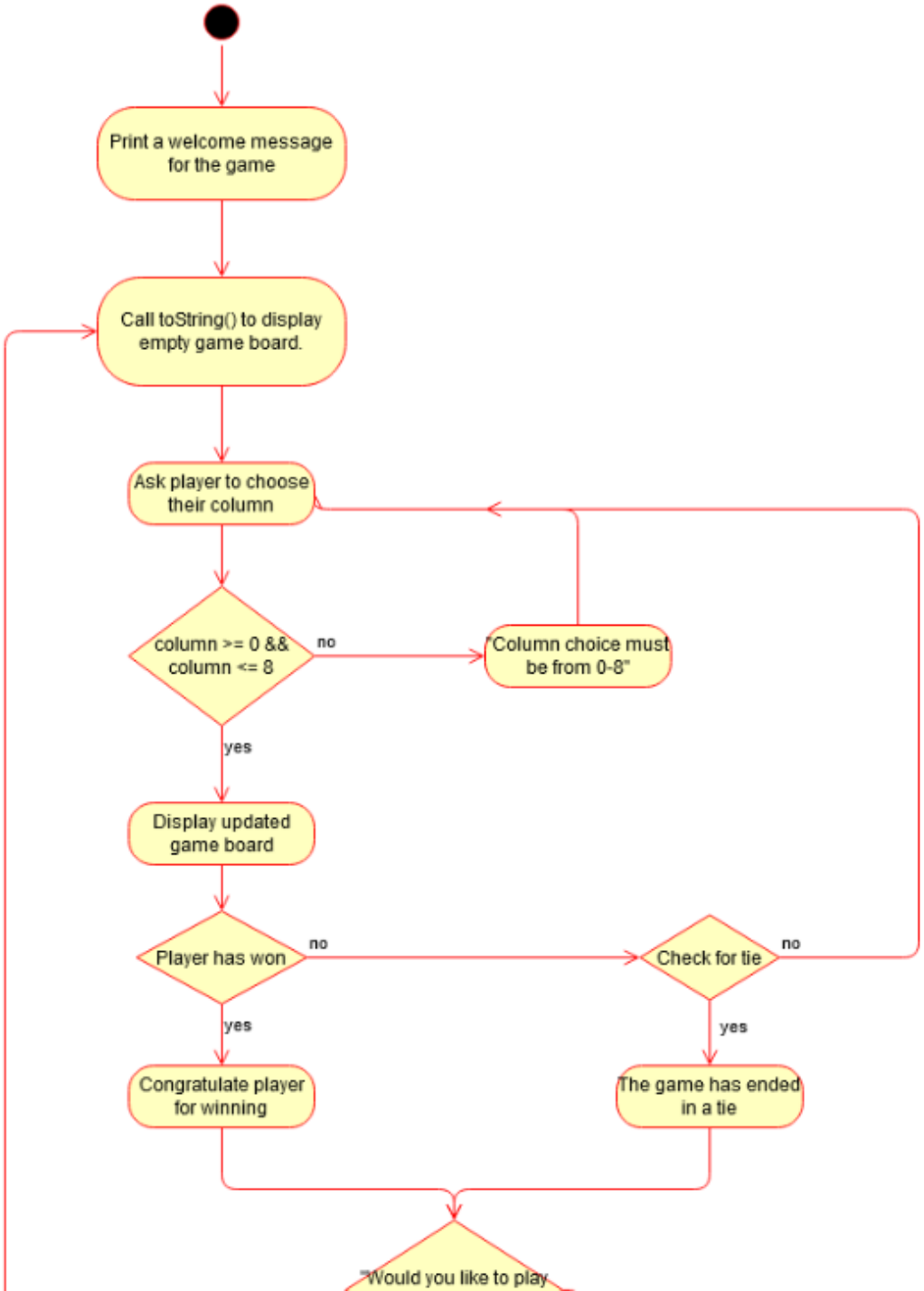


AbsGameBoard - toString()

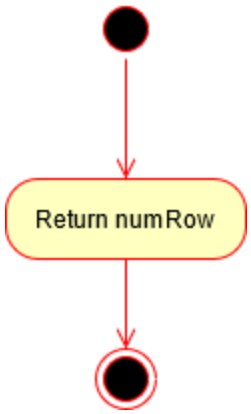


GameScreen.java - main()

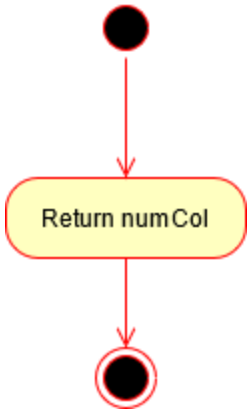
Ran out of time, please go easy on me :(



GameBoard.java AND GameBoardMem.java - getNumRows()

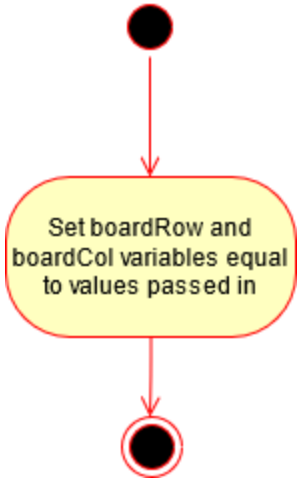


GameBoard.java AND GameBoardMem.java - getNumColumns()

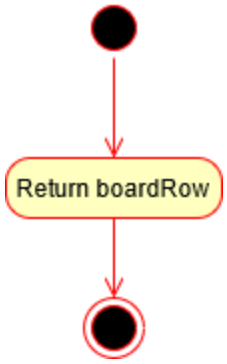


`GameBoard.java - getNumToWin()`

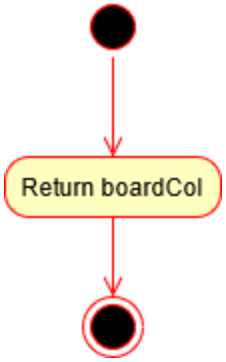
BoardPosition.java - BoardPosition()



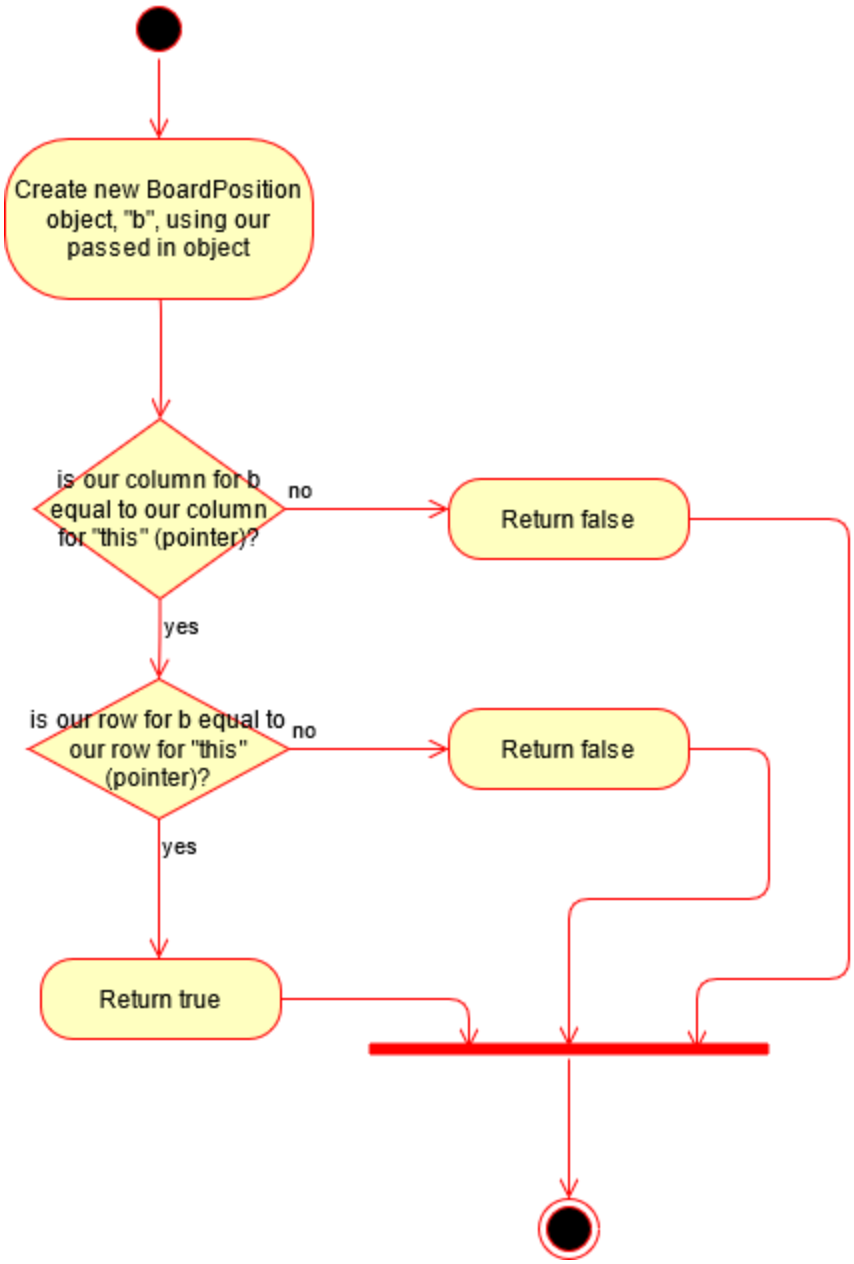
BoardPosition.java - getRow()



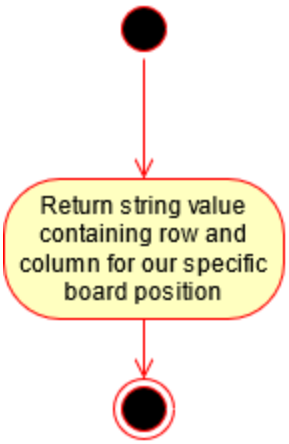
BoardPosition.java - getColumn()



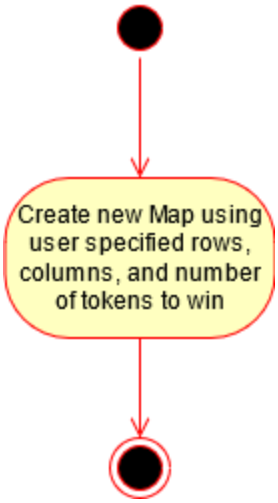
BoardPosition.java - equals()



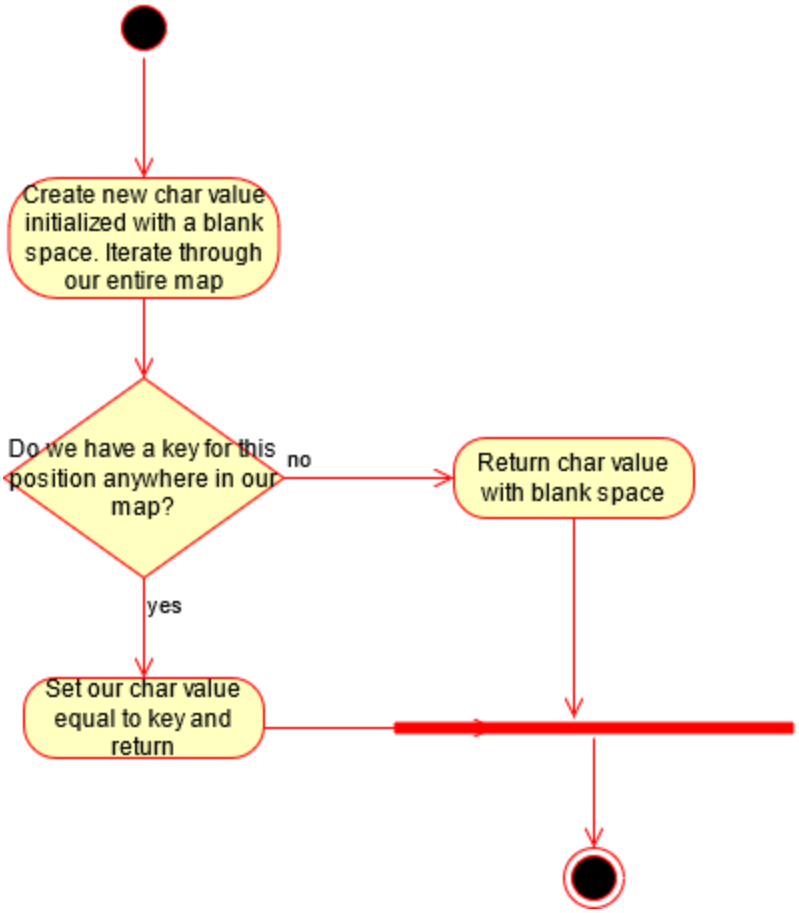
BoardPosition.java - toString



`GameBoardMem.java - GameBoardMem()`



GameBoardMem.java - whatsAtPos()



GameBoardMem - isPlayerAtPos()


Does our map contain the specified key?
no
Return false
yes
Iterate over map using player key
Does our map have the player at the specified position?
no
Return false
yes
Return true