

CPSC 2150 Project Report

Jason Rodgers

Requirements Analysis

Functional Requirements:

1. As a player, I need to know if I am X's or O's, so that I know if I am going first or not.
2. As a player, I need to know where I can move on the board, so that I can make the best possible next move.
3. As a player, I need to know if it is my turn to make a move, so that I can so that I can place my token on the board.
4. As a player, I want to be able see the board, so that I can see how I am doing
5. As a player, I need to know what spots are taken on the board, so that I don't move there on the board.
6. As a player, I need to know if I won or loss, so that I know when the game is over.
7. As a player, I want to know if I can play again, so that I can restart the game.
8. As a player, I want to be able to see all the moves made so far on the board, so that I can see if I need to play defense or offense.
9. As a player, I need to know if there are no possible moves left on the board, so that I know the game ended in a tie.
10. As a player, I want to see the winning combination that won the game, so that I can see how I won or loss.
11. As a player, I want to know what the columns are listed as, so that I ensure I place my token in the right column.
12. As a player, I need to identify where my tokens are, so that I can spot out the many ways for me to win the game.
13. As a player, I need to the know different combinations in which I can win, so that I know how to play the game.
14. As a player, I want to know how many rows and columns there are, so that I can make a strategy of how I am going win.
15. As a player, I want to know if I can change the rows and columns of the game, so that I can change how many tokens I would need in a row to win the game.
16. As a player, I want to know if I can change my token position after I place it, so that I can change my mistake if I misplace my token.
17. As a player, I want to know If I can change the token names from X and O, so that I can have some customizability in the game.
18. As a player, I want to enter the number of rows and columns that the game board will have, so that I have some control over the game.

19. As a player, I want to be able to see why that I cannot place my token at a certain position, so that I can put it at another valid position.
20. I want to get a winning message when I win the game, so that I feel rewarded after winning.
21. As a player, I can place a marker in a column, so I can claim a space.
22. As a player, if I get 5 in a row horizontally, I will win the game so that I can win the game.
23. As a player, if I get 5 in a row vertically, I will win the game so that I can win the game.
24. As a player, if I get 5 in a row diagonally, I will win the game so that I can win the game.
25. As a player, I can choose to play again, so I can play again.

Non-Functional Requirements

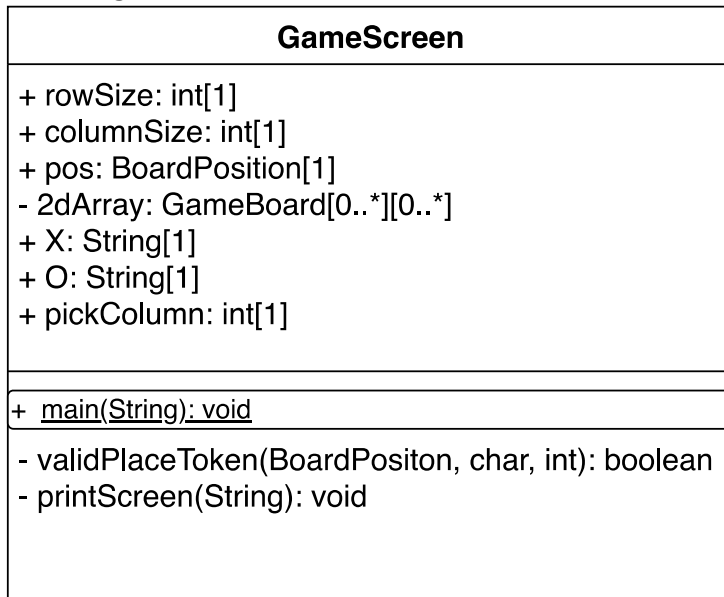
1. The system must be written in Java.
2. The system must be able to gather input from the user.
3. The system must be able to generate a 6 x 9 game board.
4. The system must be able to handle two players.
5. The system must utilize algorithms to decrease run time and increase performance and efficiency.
6. Position 0,0 is at the bottom left of the game board.

Deployment Instructions

Details in Projects 2-5.

Class 1: GameScreen

Class Diagram:

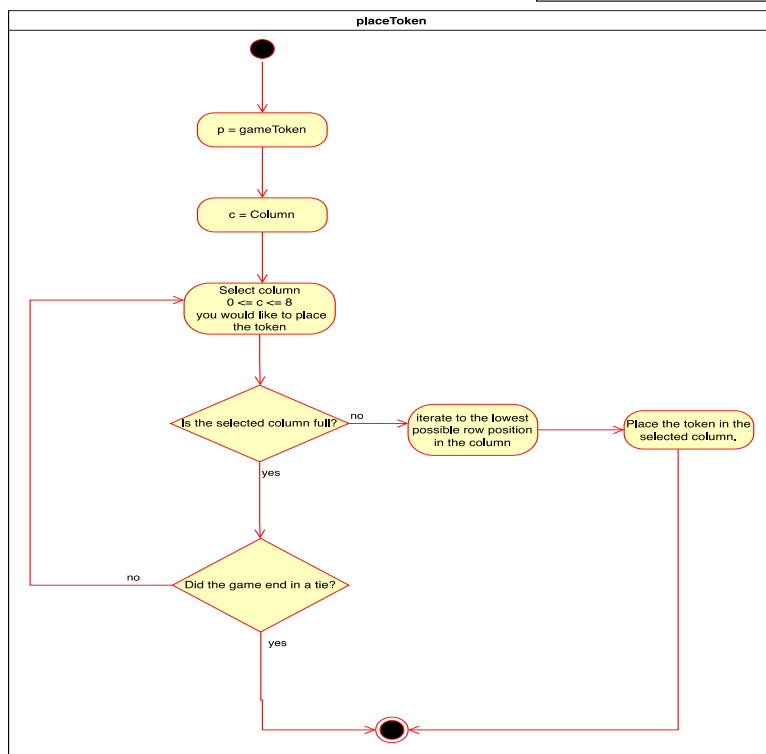
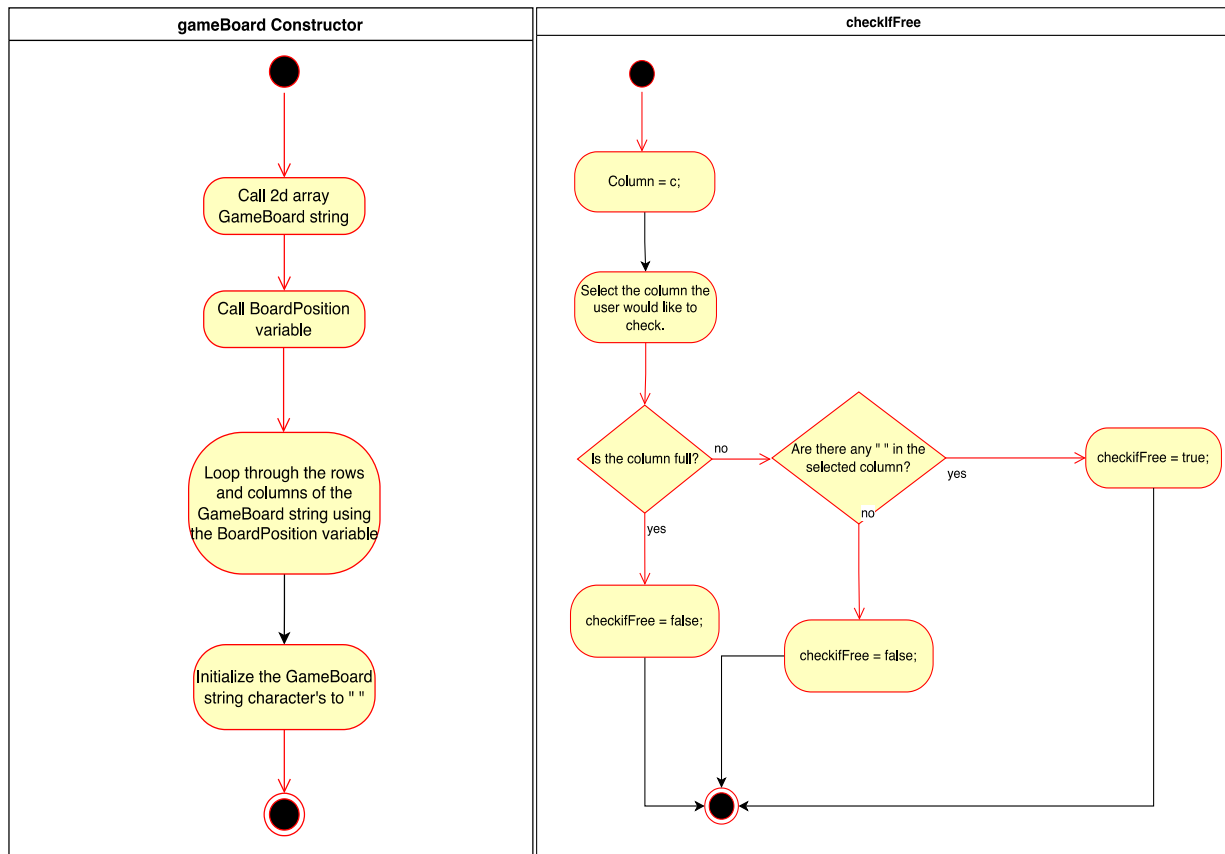


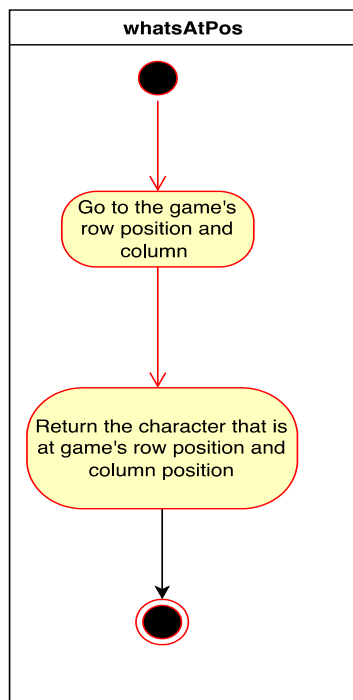
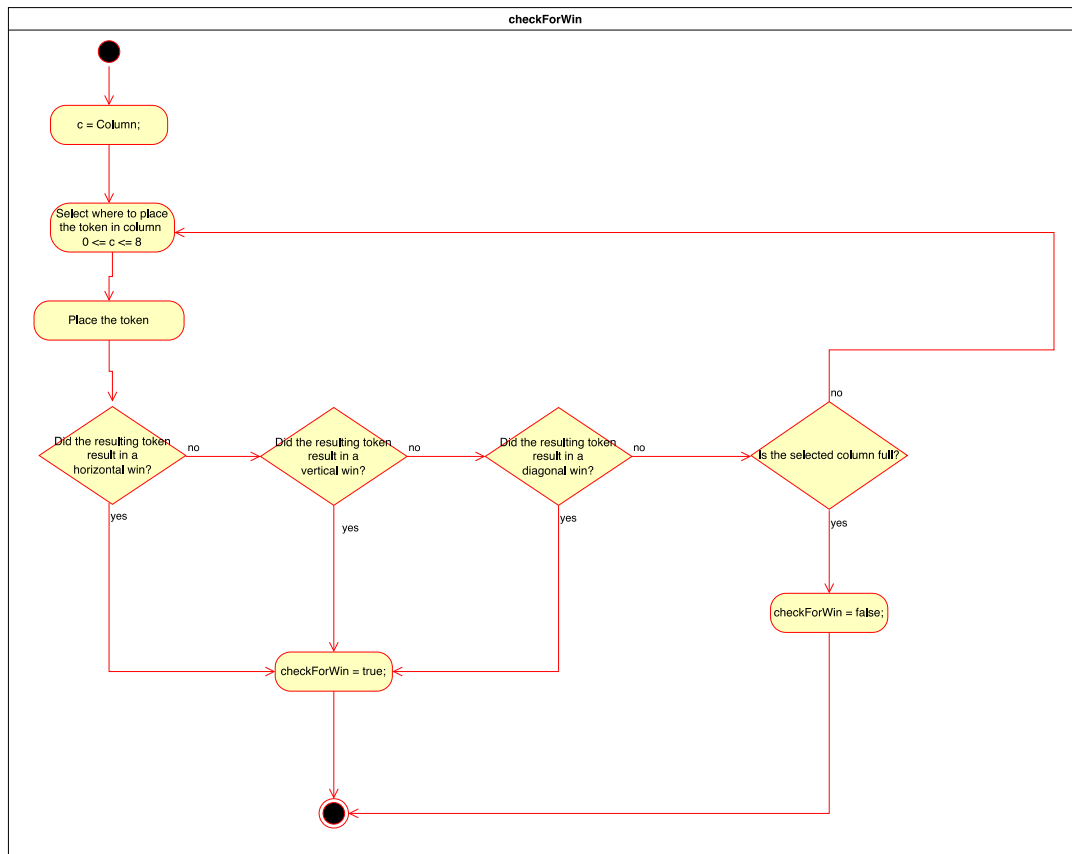
Activity diagrams (UML Diagram)

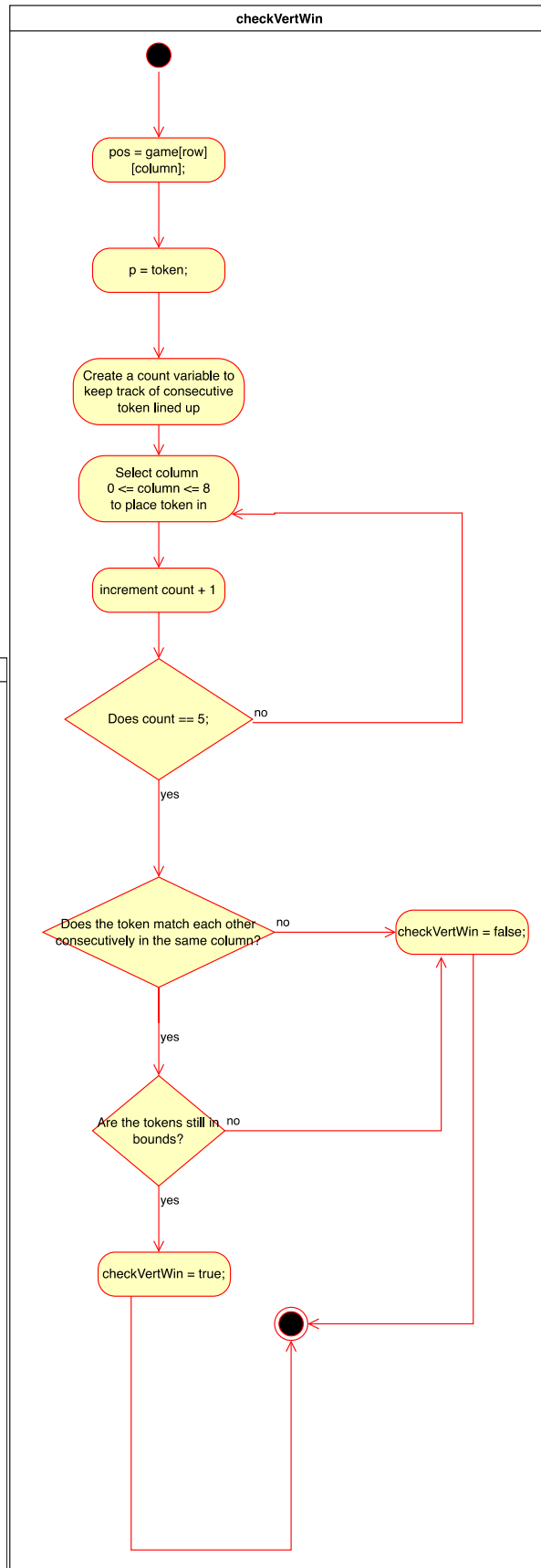
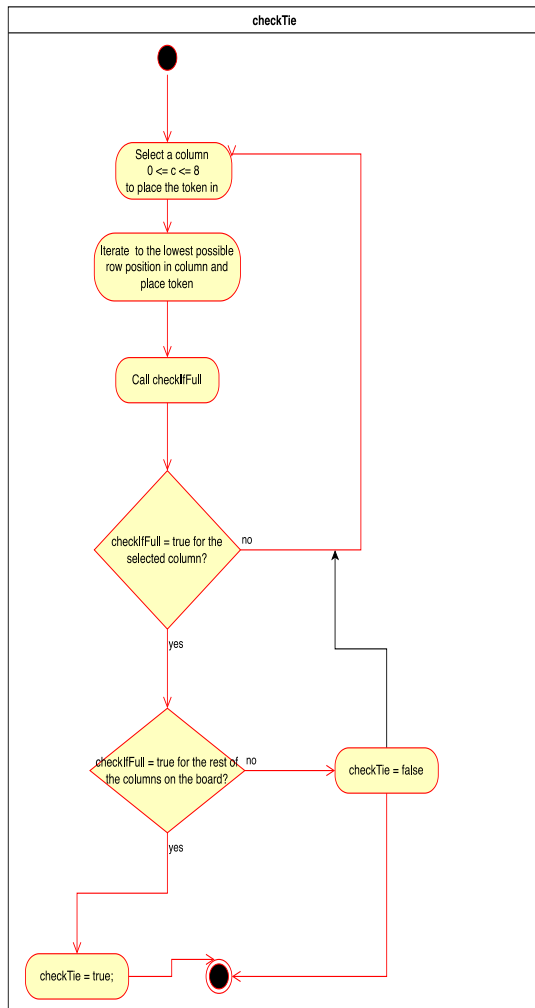
Class 2: GameBoard

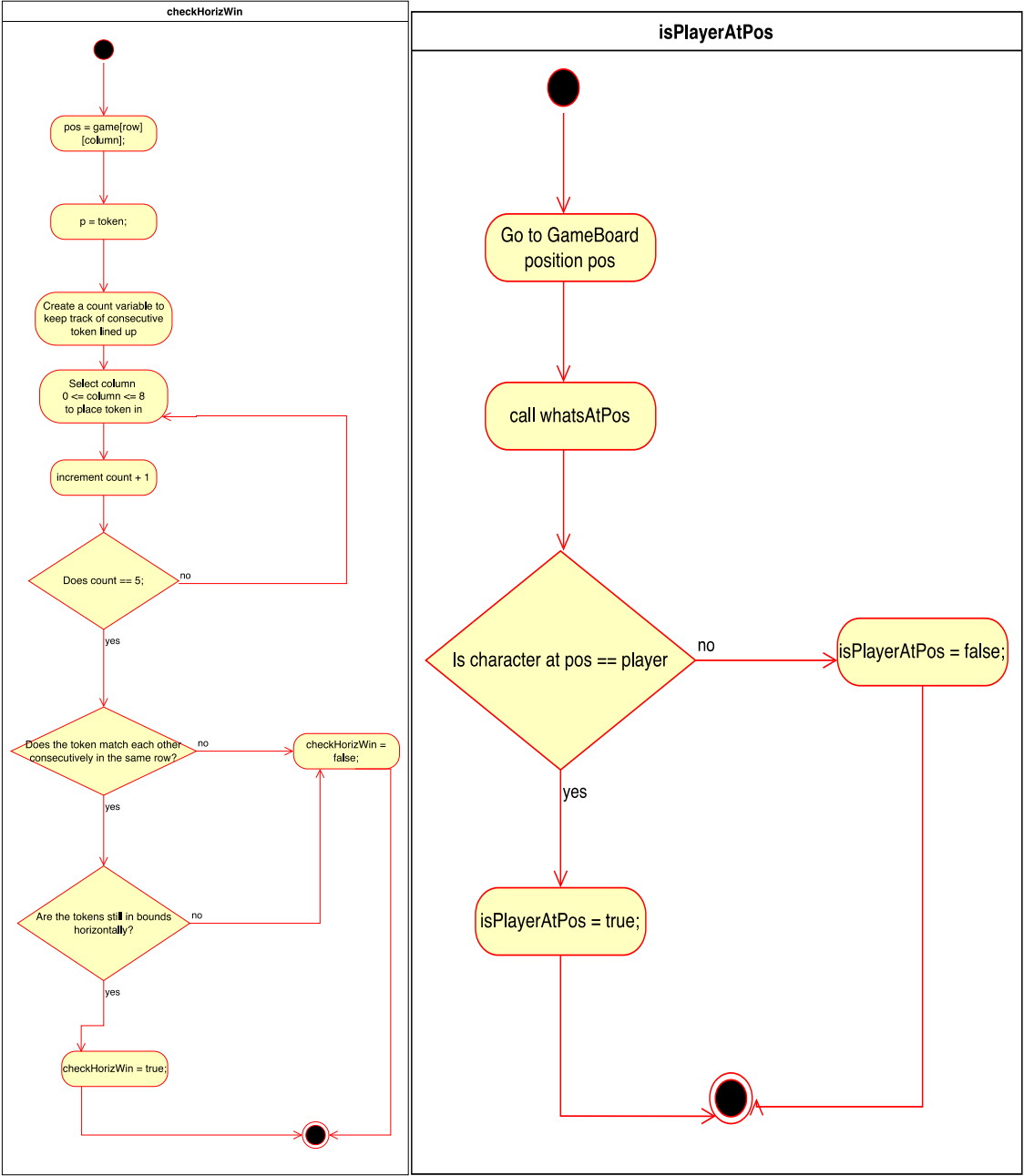
Class diagram:

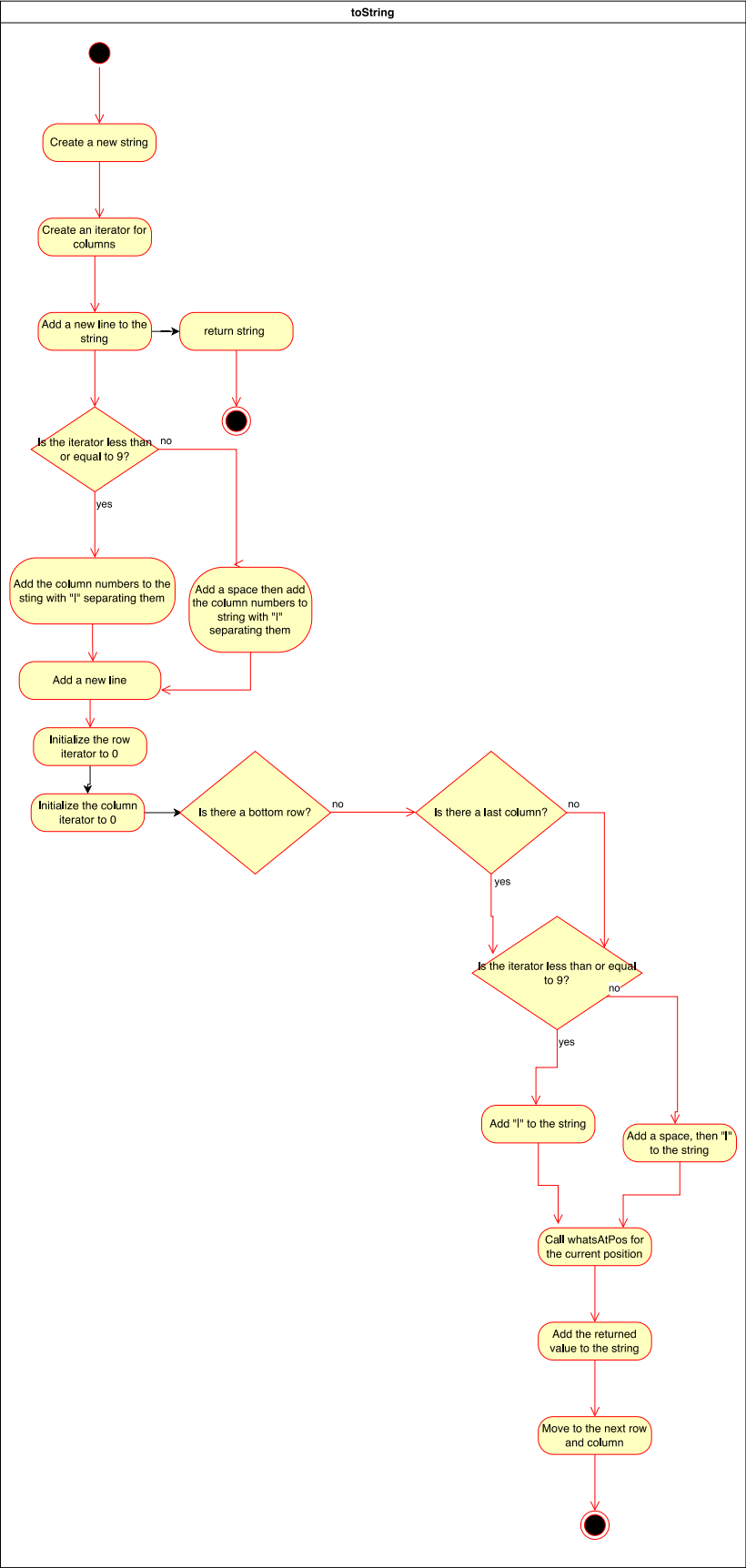
GameBoard
- gameBoard: char[][]
+ gameboard(void): void + checkIfFree(int): boolean + placeToken(char, int): void + checkForWin(int): boolean + checkTie(void): boolean + checkHorizWin(BoardPosition, char): boolean + checkDiagWin(BoardPosition, char): boolean + whatsAtPos(BoardPosition): char + isPlayerAtPos(BoardPosition, char): boolean + toString(void): String



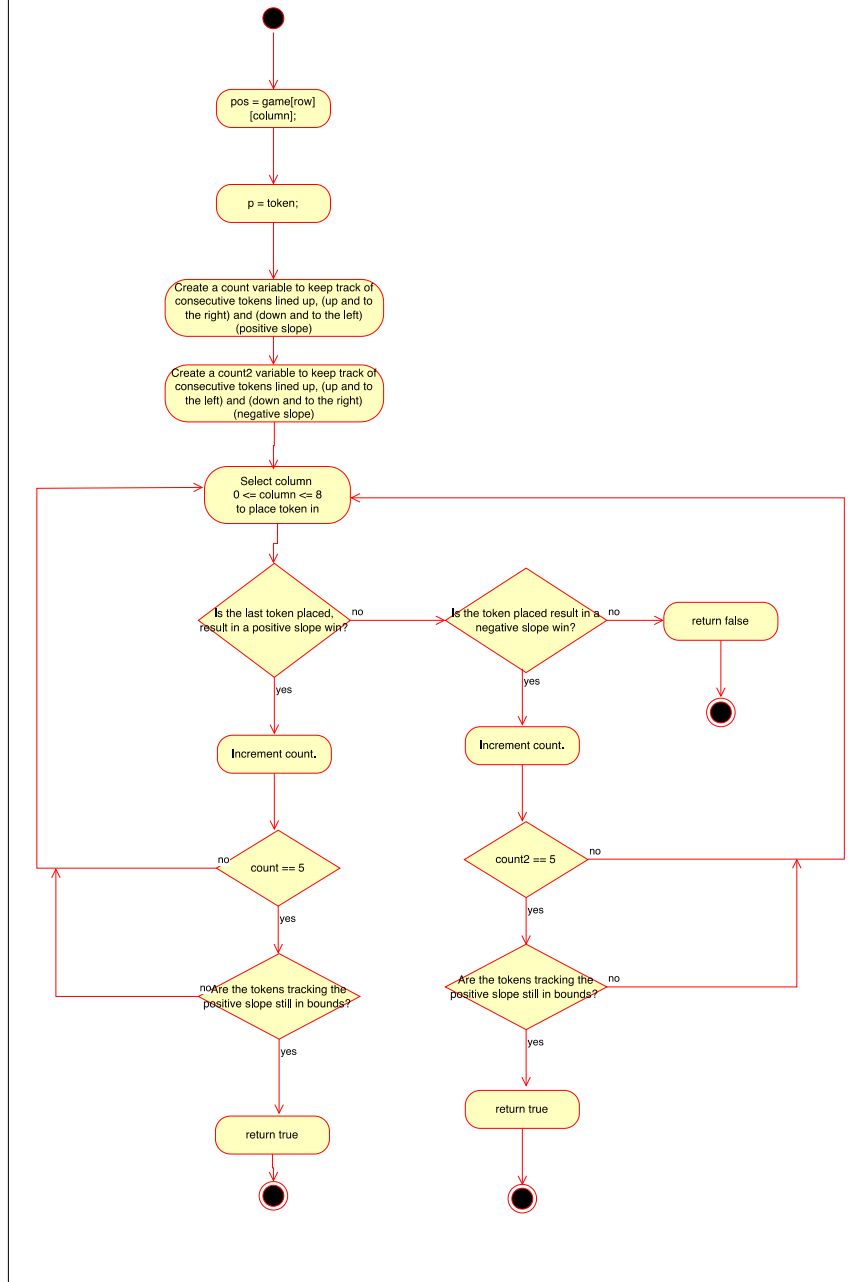








checkDiagWin



Class 3: BoardPosition

Class diagram:

BoardPosition
<u>+ Row: int[1]</u> <u>+ Column: int[1]</u>
+ BoardPosition(int, int): void + getRow(void): int + getColumn(void): int + equals(Object): boolean + toString(void): String

Test Cases

Details in Project 4.