Paige Clemons (pclemon)
Lab Section 5
Homework 1
Due: 9/23/18

Requirement Analysis

Functional Requirements

- Player should be able to see the board in order to know where they can place their tokens.
- Players should be able to choose where to place their tokens in order to effectively play the game to beat their opponent.
- Players should be able to choose if they want to play again in order to have a rematch with an opponent.

Non-functional Requirements

- Usability
  - An updated game board should be printed to the screen after every move.
  - A prompt should display which players turn it is, and it should ask where the player wants to move next.
  - The player should also be asked if he would like to play again.
- Reliability
  - A prompt should also display stating that a column is full if a player tries to put a token in that column.
  - A prompt should also display if the player tries to input an invalid column number.
- Performance
  - The game should keep track of two players data.
  - The data should be handled in the most efficient way possible.
- Supportability
  - The game should be playable between two players.
  - The program can run on any computer with Java installed.
- Implementation
  - The program must be coded in Java.
  - The program must be able to run on Unix.
  - Comments should be added to better understand the code.
  - Javadoc comments should also be added to know the requirements of the classes and functions.

Testing

I started by writing the constructor for the GameBoard class then I wrote the toString. I created the printBoard function in the Connect4Game class to print information to the screen. So, I wanted to do I quick test to see if those three functions were functioning properly. However, I ran it and received a null pointer exception in the toString function when I expected an empty board to be displayed. I eventually figured out that I did not make board, my 2D array member variable of the class GameBoard, an instance variable in the constructor.

Next, I wrote the checkIfFree function. In order to test this, I changed the constructor to be filled with Xs and called the checkIfFree function in main and passed column number 0 into it. I then printed the returned value. In this case I expected the resulting boolean value to be false since column 0 was full, and I did receive the expected result. I changed the constructor back to filling the array with spaces and then ran the program. I expected the resulting boolean value to be true since column 0 was empty, and I did receive the expected result.

Next, I tested the placeToken function by inputting the column number 1. I expected for the board to only have one token, X, at the very bottom of the column 1. However, the entire column filled with tokens. I fixed this by starting from the bottom of the column instead of the top of the column.

I next tested the checkVertWin by filling column 1 with Xs in rows 2, 3, 4, and 5 in the constructor. I expected to get the true return from the function, since there is a win with 4 in a row. I did get true returned from the function. I tested that I would get false by changing row 3 column 1 to a O. I did get false returned from the function.

I next tested the checkHorizWin by filling row 5 columns 1-4 in the constructor with Xs, and I called checkHorizWin by passing the parameters 5, 3, X. I expected to get true. However,

I got false. I figured out that I need && instead of || in my while loop. I also tested this function by having row 5 columns 1-4 with Xs, row 5 column 5 with an O, and row 5 column 6 with an X, and I called checkHorizWin by passing the parameters 5, 3, X. I expected to get true returned and that is in fact what I got. I next tested with row 5 columns 0 – 3 with Xs and row 5 columns 4 and 5 with Os and row 5 column 6 with an X. I also passed the values 5, 0, X into checkHorizWin. I expected to get true returned, and I did in fact get true. I next tested with row 5 column 0 with an X, row 5 column 1 and 2 with an O, row 5 columns 3 – 6 with Xs. I passed the values 5, 6, X into checkHorizWin. I expected to get true returned, and I did get true. I next tested alternating X and O starting in row 5 column 1. I passed 5, 4 ,X into checkHorizWin. I expected to get false. However, I got true. After the last test, I had to completely rethink my logic for this function. After, I rewrote the function I retested with the same test as above and got all the right output. I also add two new test. The first one I added was row 5 columns 1 – 4 filled with Xs and row 5 column 5 with an O and lastly row 5 column 6 with a X. I then passed 5, 5, O into checkHorizWin. I did not get the expected result of false. I needed to add a condition that checked that both the current spot in the array and the next spot over where equal to the player token that was passed in. After I did this though I got false which is the expected output. The next test I added was filling row 5 columns 0 – 3 with Xs, row 4 columns 4 & 5 with Os, and filled row 5 column 6 with a X. I then passed 5, 4, O into checkHorizWin. I expected to get false which is what I got. I next tested with row 5 columns 0 – 3 filled with Xs, row 5 columns 4 & 5 with Os, and finally row 5 column 6 with a X. I then passed 5, 0, X into checkHorizWin. I expected to get true, however; I got an array out of bounds exception. In order to fix this, all I needed to do was fix the conditional statement that checked if the position was in the bounds of the array. After I did this, I got the expected answer of true.

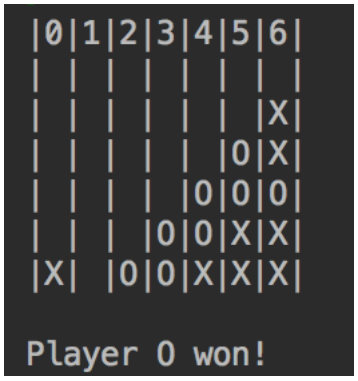
Next, I test the checkDiagWin. I filled row 0 column 5 with a X, row 1 column 4 with a X, row 2 column 3 with a X, and row 3 column 4 with a X. I used this board with my first four tests. I tested this board by first passing 2,3,X into checkDiagWin. I expected to get true and this is what I got. I next passed 0,5,X into checkDiagWin, I expected to get true and that is what I got. Next, I passed 1,4,X into checkDiagWin, I expected once again to get true and that is the answer I got. Finally, I test this board one last time using 3,2,X as the values passed into checkDiagWin. I expected to get true, and I did get true. My next test was filling row 0 column 5 with a X, row 1 column 4 with an O, row 2 column 3 with a X, and row 3 column 2 with a X. I then passed 0,5,X into checkDiagWin. I expected to get false, and that is in fact what I got. I next tested by filling row 0 column 1 with a X, row 1 column 2 with a X, row 2 column 3 with a X, and row 3 column 4 with a X. I then passed 1,2,X into checkDiagWin. I expected to get true, but I got false. I fixed this by changing the one of my conditions. Once I fixed the condition I got true. Using the same board, I passed 0,1,X into checkDiagWin. I expected to get true and that is what I got.

Next, I tested the checkForWin function by first running three tests with the same board that board would be: row 5 columns 1 – 4 filled with Xs, row 5 column 5 with an O, and row 5 column 6. The first test I passed 5 into checkForWin, and I expected to get false. I did in fact get false. Next, I passed 6 into checkForWin, and I expected to get false again. I did once again get false. For my last test with this board, I passed 3 into checkForWin. I expected to get true from this test, but I got false. I then figured out I needed add 1 to the row when I called the checkHorizWin in the checkForWin function. Once I figured that out, I got true. I next tested with row 0 column 6 with a X, row 1 column 6 with a X, row 2 column 6 with a X, row 3 column 6 with an O, row 4 column 6 with a X, and row 5 column 6 with an O. Then I called

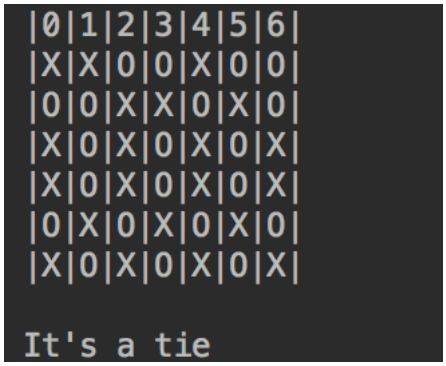
checkForWin with a 6 passed to it and expected to get false. I did get false. My next test rows 2 – 5 column 1 filled with Xs. I then passed 1 into the checkForWin function. I expected to get true from this test, but I got false. I fixed this by calling checkVertWin with row + 2 ,c , whatsAtPos(row + 2,c). After I did this, I got true. My final test row 0 column 5 with a X, row 1 column 4 with a X, row 2 & 3 column 4 with Os, row 4 column 4 with a X, row 5 column 4 with an O, row 2 column 3 with a X, and row 3 column 2 with a X. I then passed 4 to checkForWin. I expected to get true but got false. I fixed it by adding two to the row in the call to checkDiagWin in the checkForWin function and by calling whatsAtPos with 2 added to row in the call to checkDiagWin. Once I made that change I got the expected result of true.

To test the actually playing of the game, I used the example from the instructions. I started with the main and as soon as I started playing the game and typed in 3 like in the homework instructions it ended and said I won. It was supposed to ask the next player for a turn. I needed to add a condition in my checkForWin function and once I did that the game continued. Then when I entered a column that was full, I got stuck in an infinite loop. After looking at my code I discovered that I forgot to put the line of code to get the number from the user again. Once I added that though the program continued as expected. I next ran into a problem when I entered a move that was a winning move it did not recognize it as a win. I then discovered I had to change the condition I added earlier, but once I did that it worked as expected once again. I next had a problem when the user was asked to play again. When I enter 'y' I expected to start a new game, however; it asked me again like I entered an invalid letter. I fixed by resetting all the needed variable to start the game over.

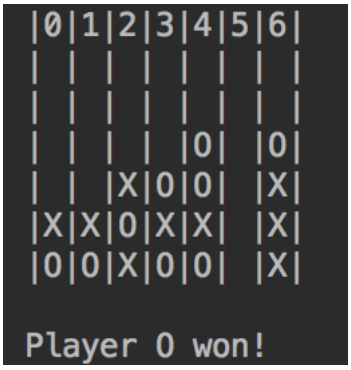
I next entered the following moves: 0, 3, 4, 4, 5, 2, 5, 5, 6, 4, 6, 6, 6, 3, 6, and 5. This is the output that I got was:



I next wanted to test the checkForTie. So, I made the following moves: 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, 4, 5, 6, 0 , 1, 2, 3, 4, 5, 6, 1, 0, 3, 2, 5, 4, 1, 6, 0, 2, 2, 0, 6, 5, 4, 1, 6, 3, 3, 4, and 5. This is the output I got:

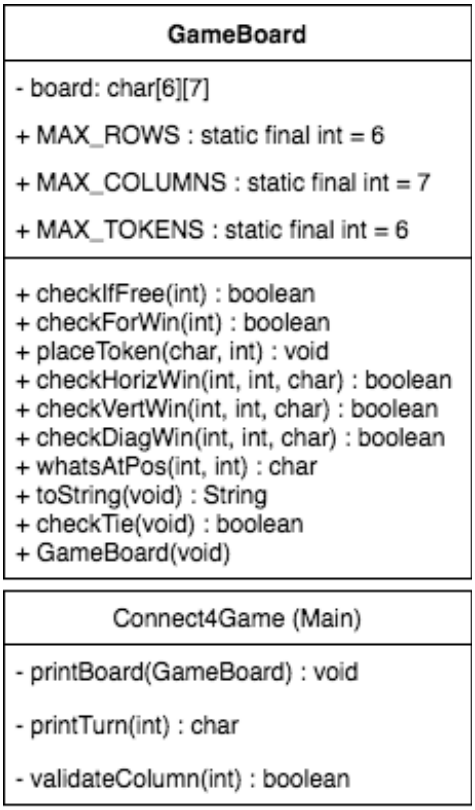


I next made the following moves: 6, 3, 3, 4, 2, 2, 6, 0, 0, 3, 6, 6, 4, 1, 1, 4, 2, and 4. This is the output I got:

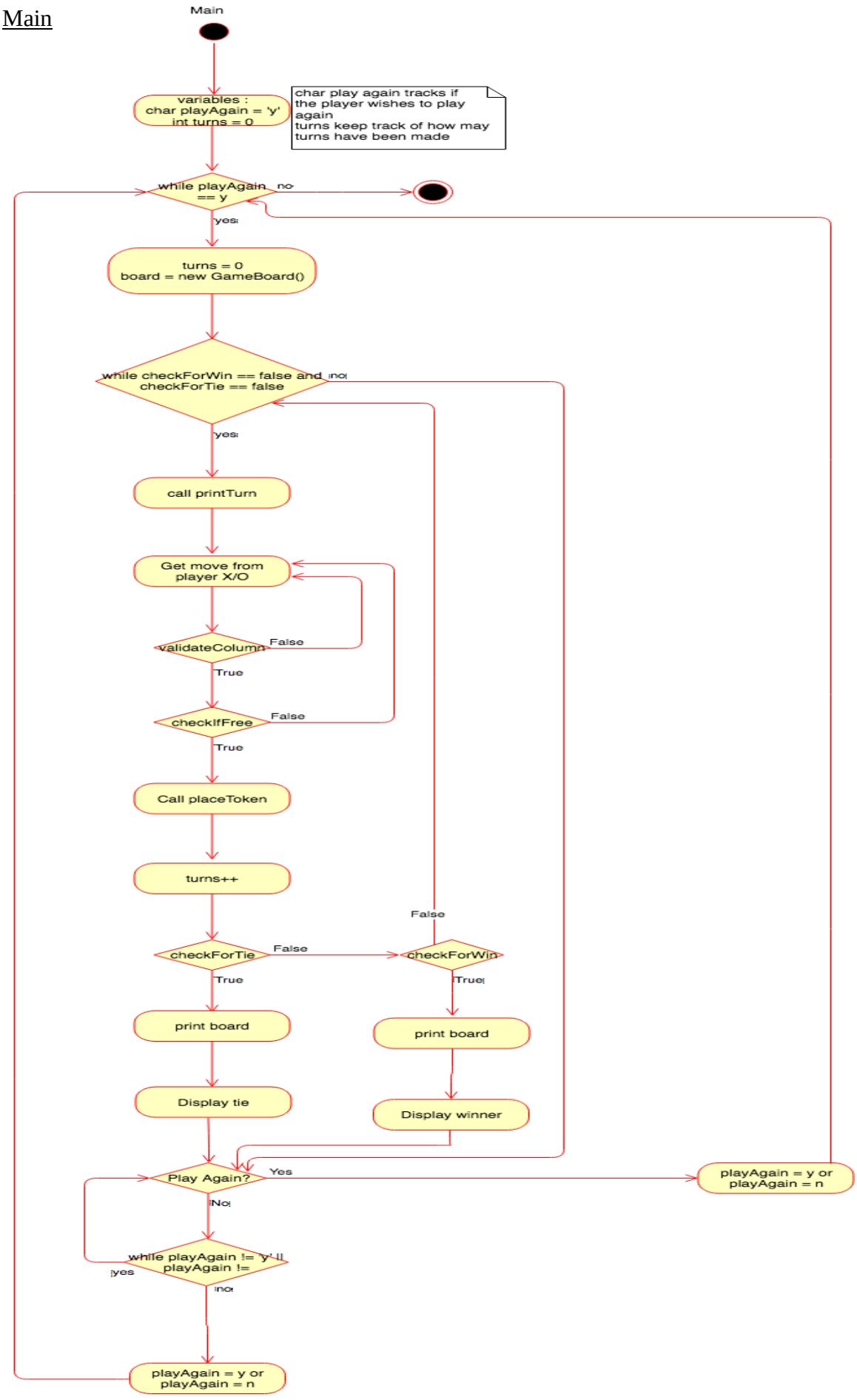


Diagrams

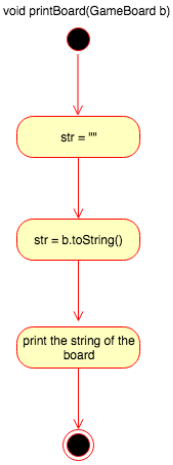
Class Diagrams

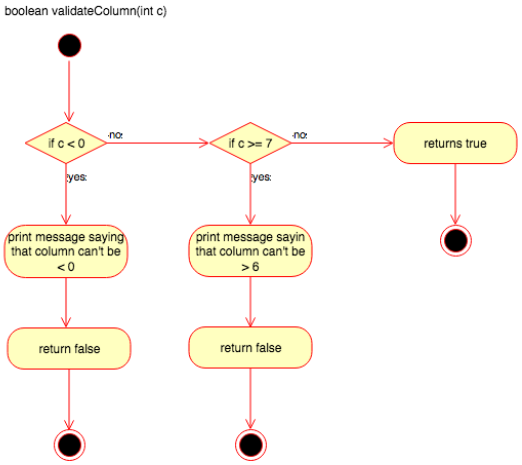


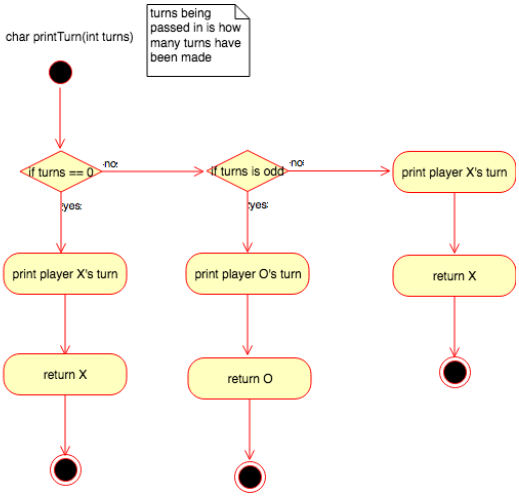
Main


Main
variables :
char playAgain = 'y'
int turns = 0
char play again tracks if the player wishes to play again
turns keep track of how may turns have been made
while playAgain == y
no
yes
turns = 0
board = new GameBoard()
while checkForWin == false and checkForTie == false
no
yes
call printTurn
Get move from player X/O
validateColumn
False
True
checkIfFree
False
True
Call placeToken
turns++
checkForTie
False
True
checkForWin
False
True
print board
Display tie
print board
Display winner
Play Again?
Yes
No
playAgain = y or playAgain = n
while playAgain != 'y' || playAgain !=
yes
no
playAgain = y or playAgain = n

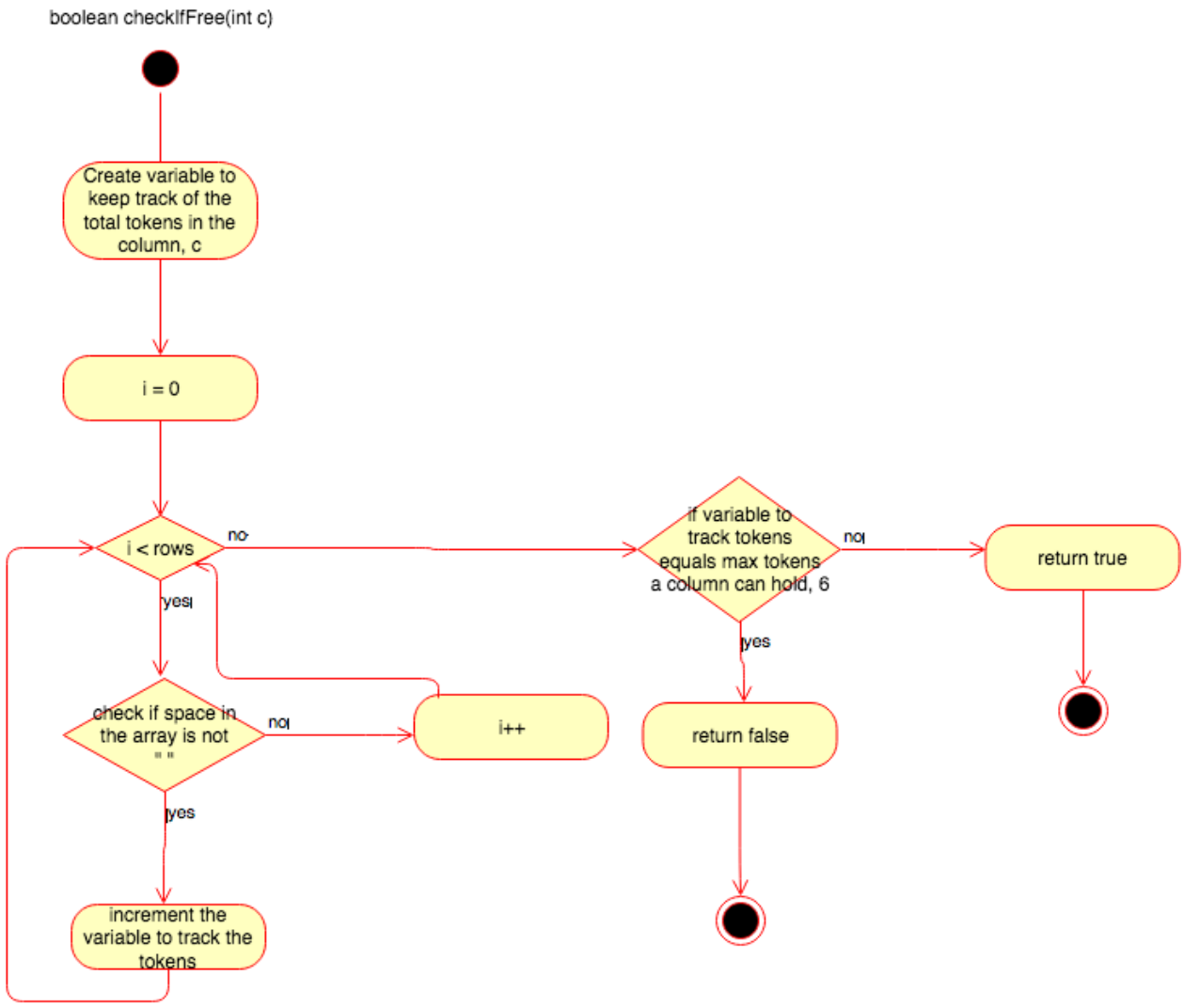
Private Helper Functions in Main



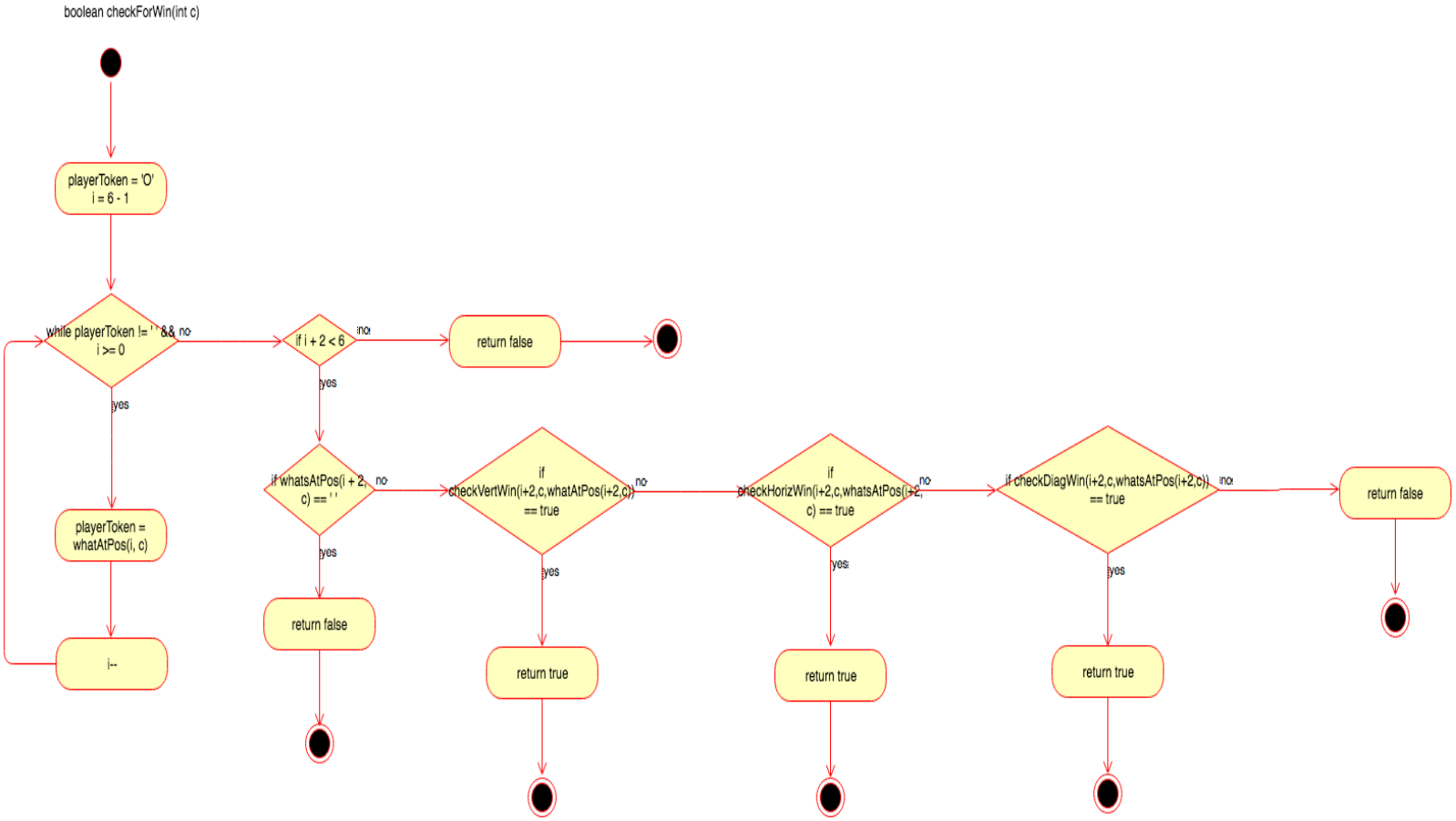




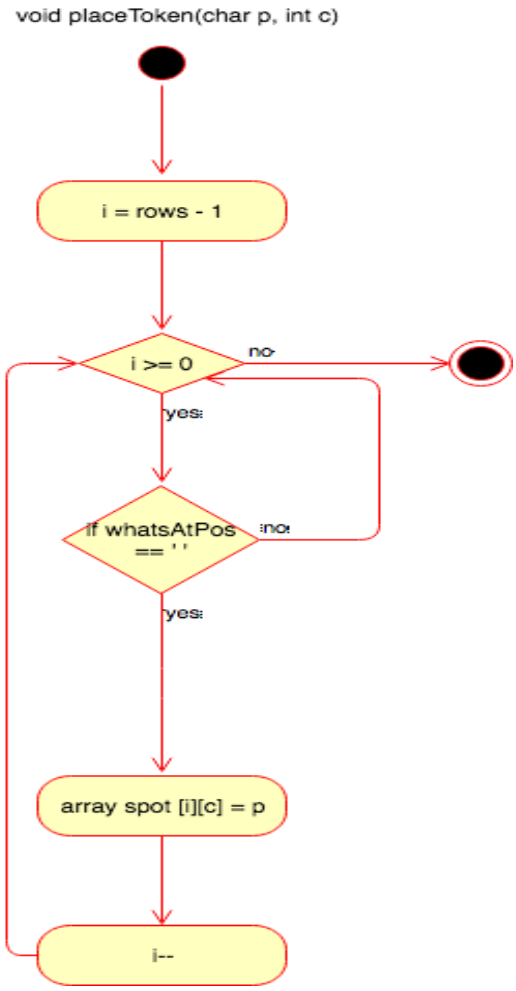
checkIfFree



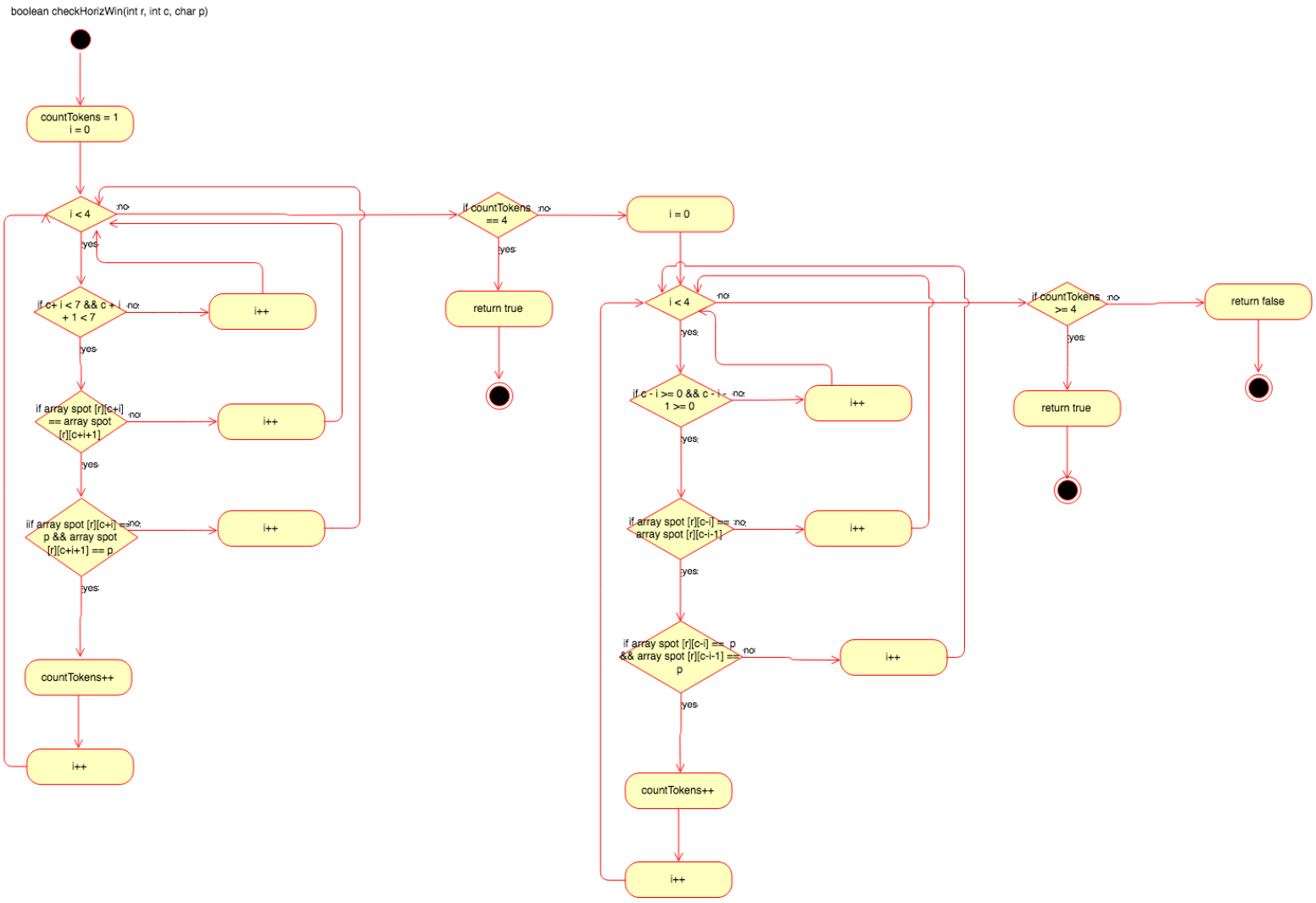
checkForWin



placeToken

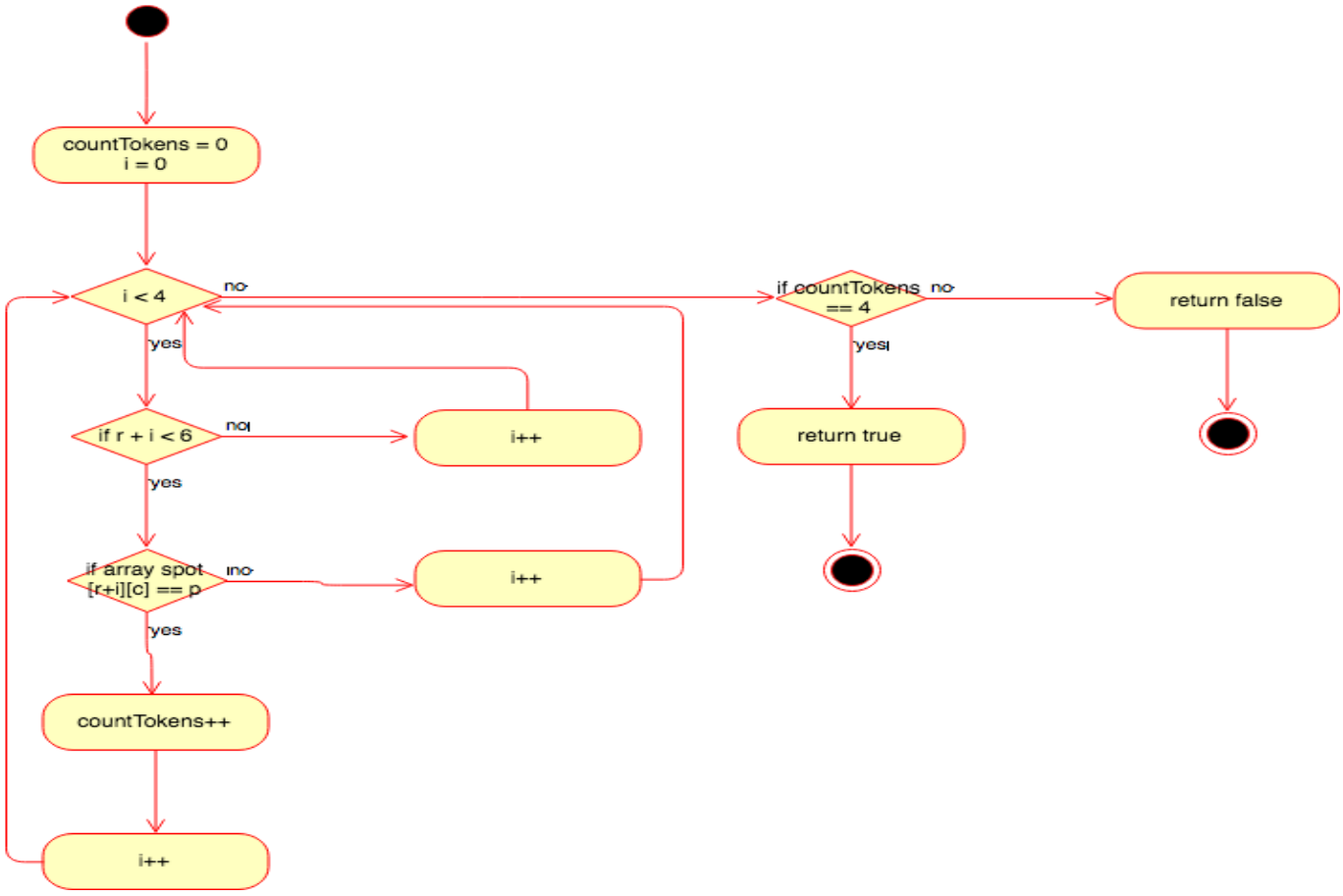

void placeToken(char p, int c)
i = rows - 1
i >= 0
no
yes
if whatsAtPos == ' '
no
yes
array spot [i][c] = p
i--

checkHorizWin

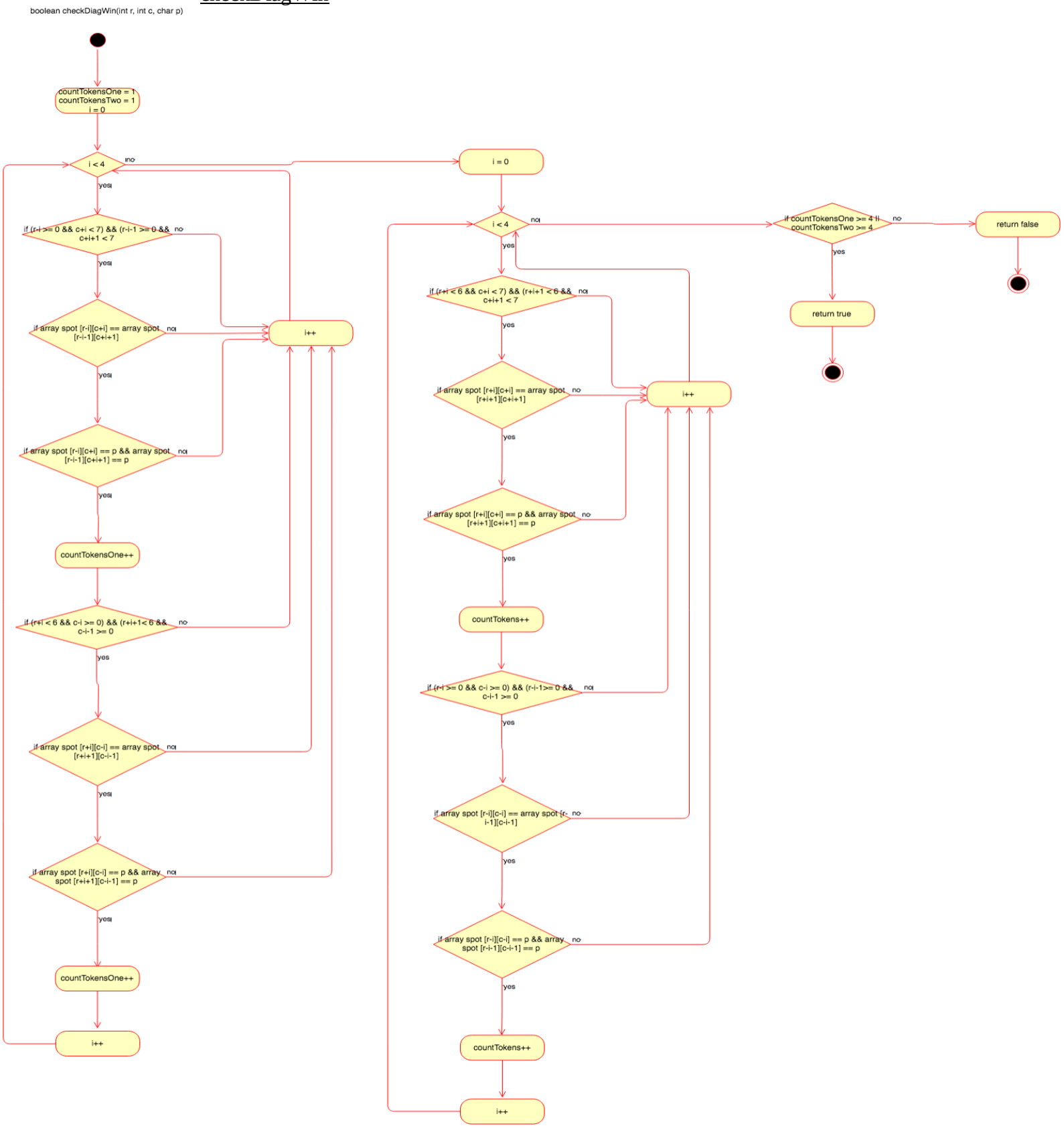


checkVertWin

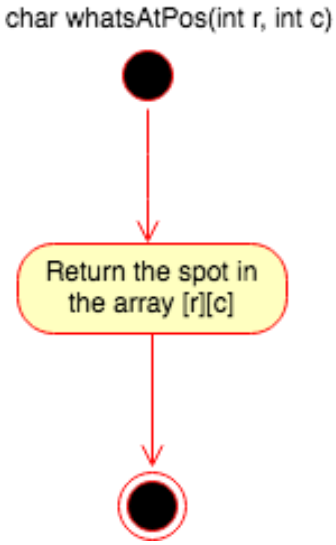
boolean checkVertWin(int r, int c, char p)



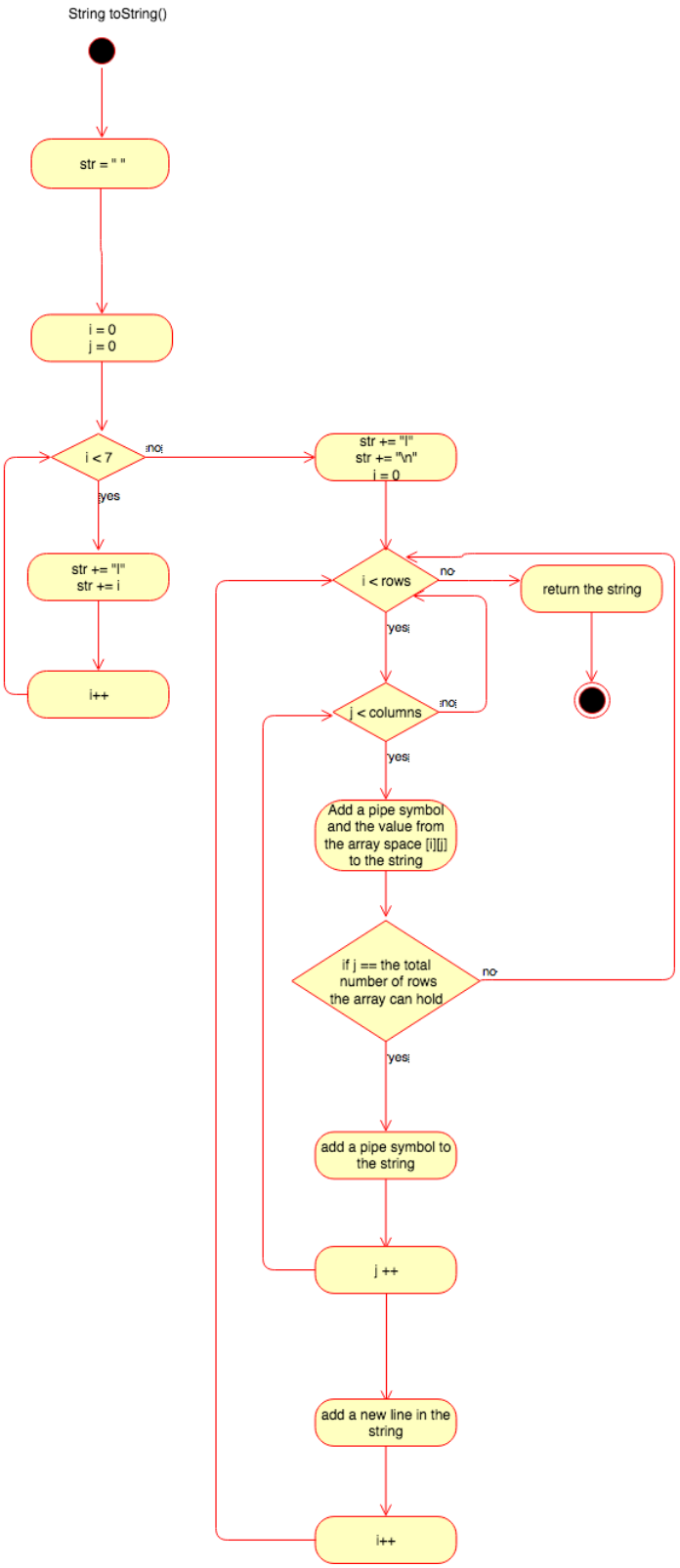
checkDiagWin



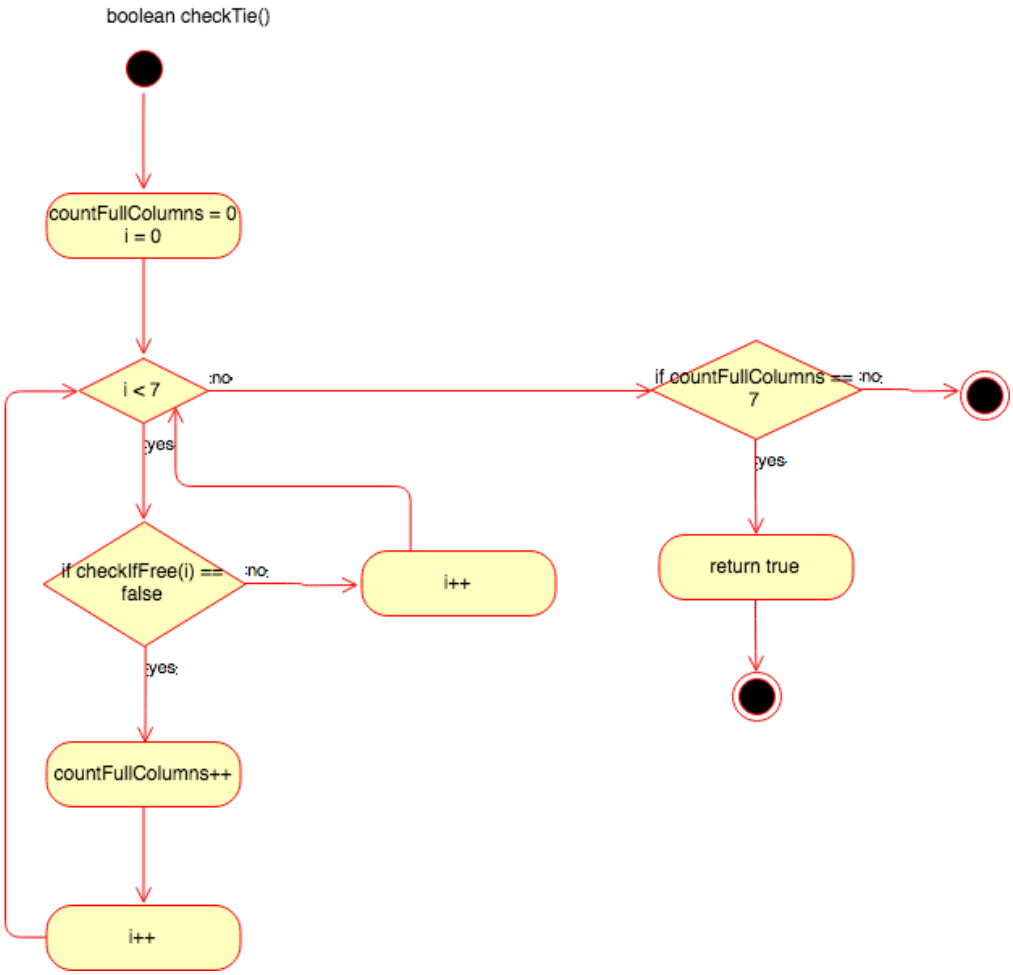
whatsAtPos



toString


String toString()
str = " "
i = 0
j = 0
i < 7
no
yes
str += "|"
str += i
i++
str += "|"
str += "\n"
i = 0
i < rows
no
return the string
yes
j < columns
no
yes
Add a pipe symbol and the value from the array space [i][j] to the string
if j == the total number of rows the array can hold
no
yes
add a pipe symbol to the string
j ++
add a new line in the string
i++

checkTie


boolean checkTie()
countFullColumns = 0
i = 0
i < 7
:no
:yes
if checkIfFree(i) == false
:no
:yes
i++
countFullColumns++
i++
if countFullColumns == 7
:no
:yes
return true

Deployment

In order to run my program, you first have to unzip the PaigeClemonsHW1 folder. Then navigate to that folder in your terminal. Once you in the PaigeClemonsHW1 folder, there should only be a cpsc2150 folder, the makefile, and my report in the folder. After you check that stay in that folder, you can compile my program by typing "make" into the terminal. Then to run the program all you have to do is type "make run" then you can play the Connect Four game as usual. If you wish to delete the .class files then all you have to do is type "make clean" into the terminal.