Homework 2 – Program Report

Arnav Guneta

Functional Requirements

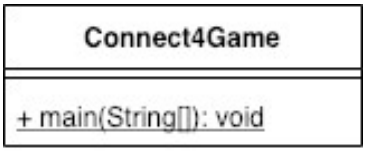
As a player of the connect 4 game, I get to pick the number of rows, columns, and number of markers in a row needed to win the game to customize the game for myself. As a player, I get to see the board, with the specified condition from before, to decide where to place my token. As a player, I will take turns dropping tokens into the grid in order to win. I need to get four tokens in a row either horizontally, vertically, or diagonally to win. As a player, I get asked to pick a column (between 0 and the column number I picked) to place my token into so I can make a move. As a player, I get to view the board after placing my token so I can see what the board looks like. The game alternates and asks the other player so there is competition. As a player, I can't add tokens to a full column because it wouldn't fit. As a player, I keep dropping tokens until either me or the opponent wins or draws. After that, I get to choose if I want to keep playing so I can quit if I wanted to. As a player of the game, if I choose to play again, I can re-specify my board dimensions and markers needed in a row in order to win so I can keep playing with a board that I want.

Nonfunctional Requirements

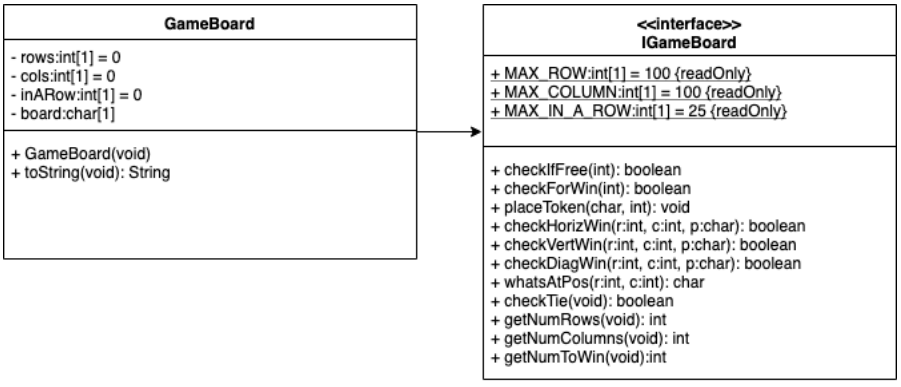
The program must be coded in Java. The program must run on Unix (console-based). The program must be robust and be compatible for updates/ add-ons.

UML Class Diagrams

Connect4Game UML class diagram

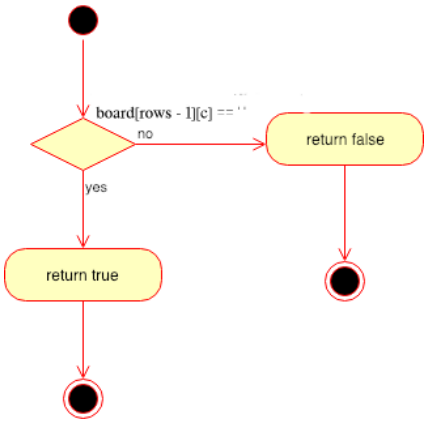


GameBoard UML class diagram

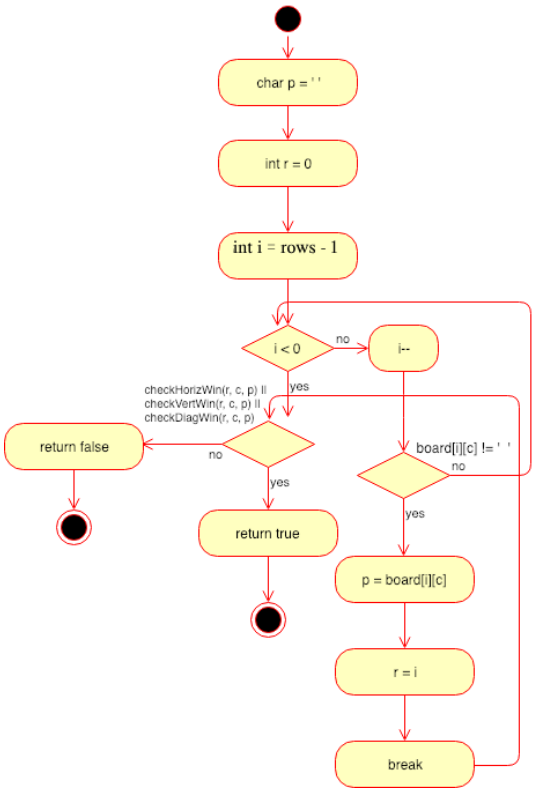


UML Activity Diagrams

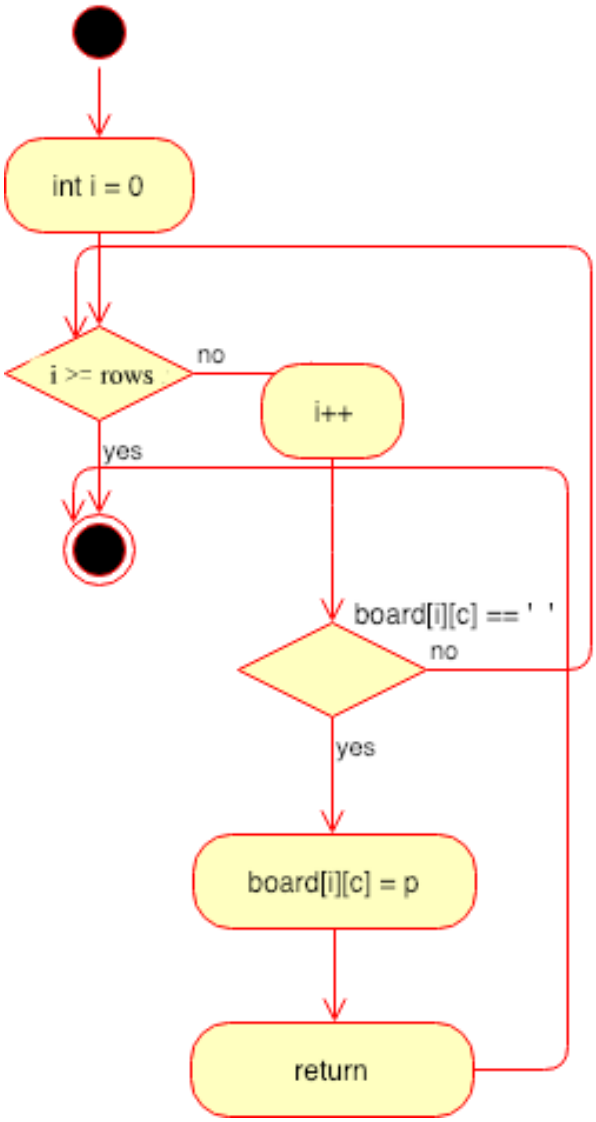
checkIfFree(int c)



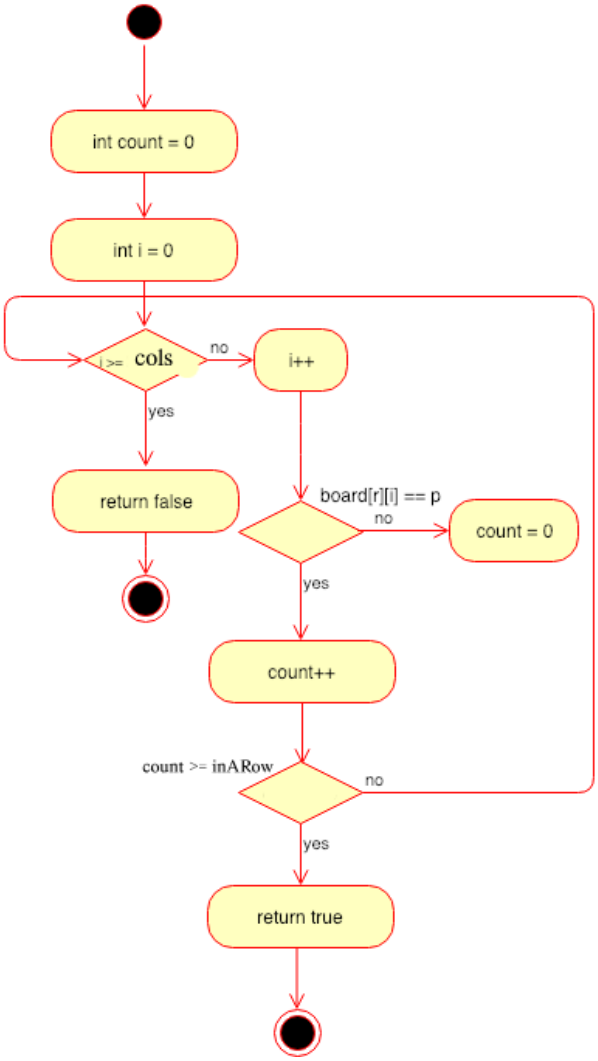
checkForWin(int c)



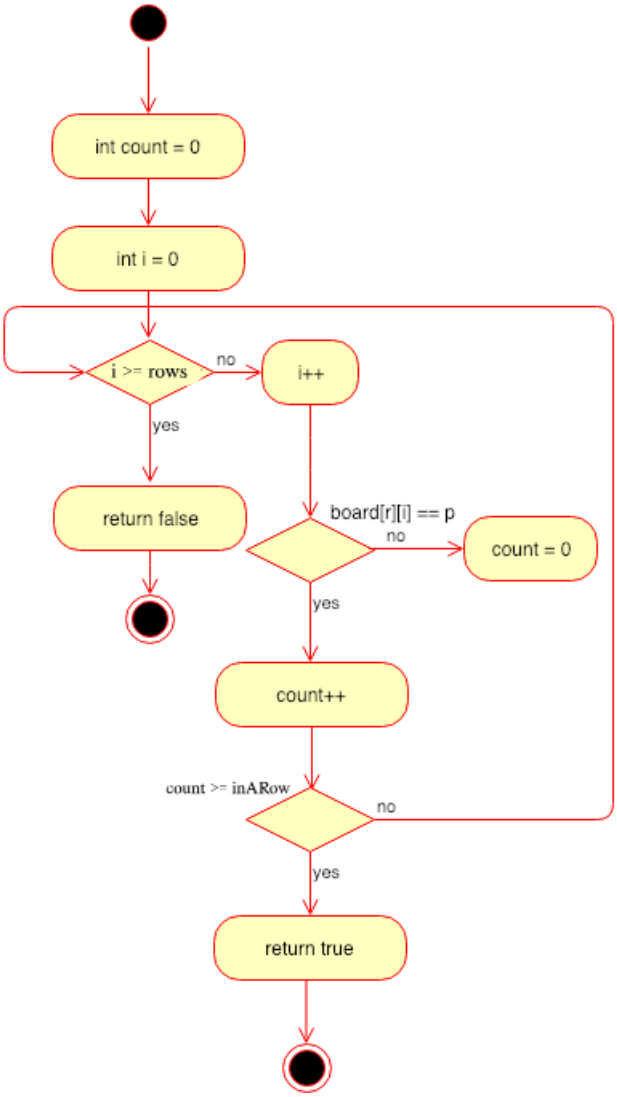
placeToken(char p, int c)



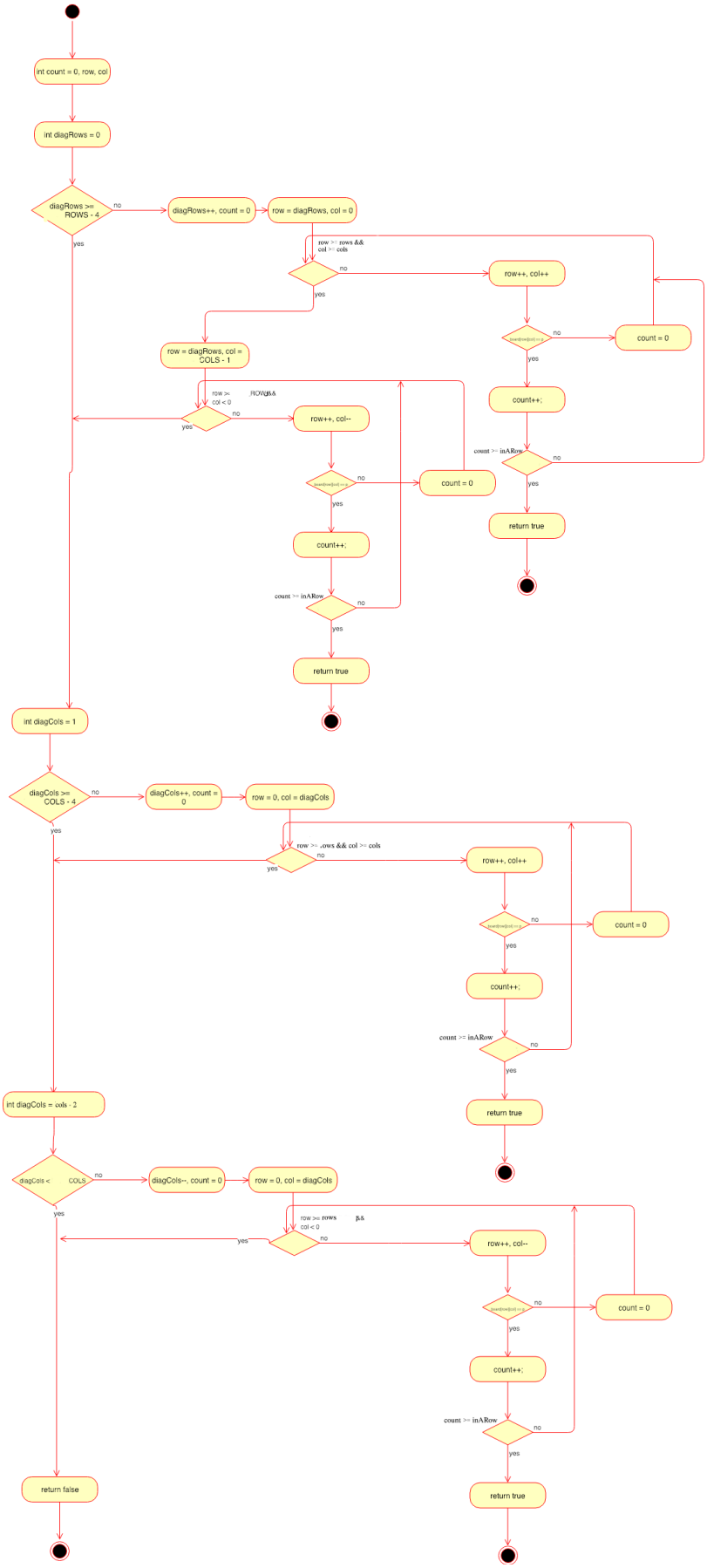
checkHorizWin(int r, int c, char p)



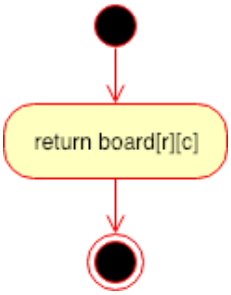
checkVertWin(int r, int c, char p)



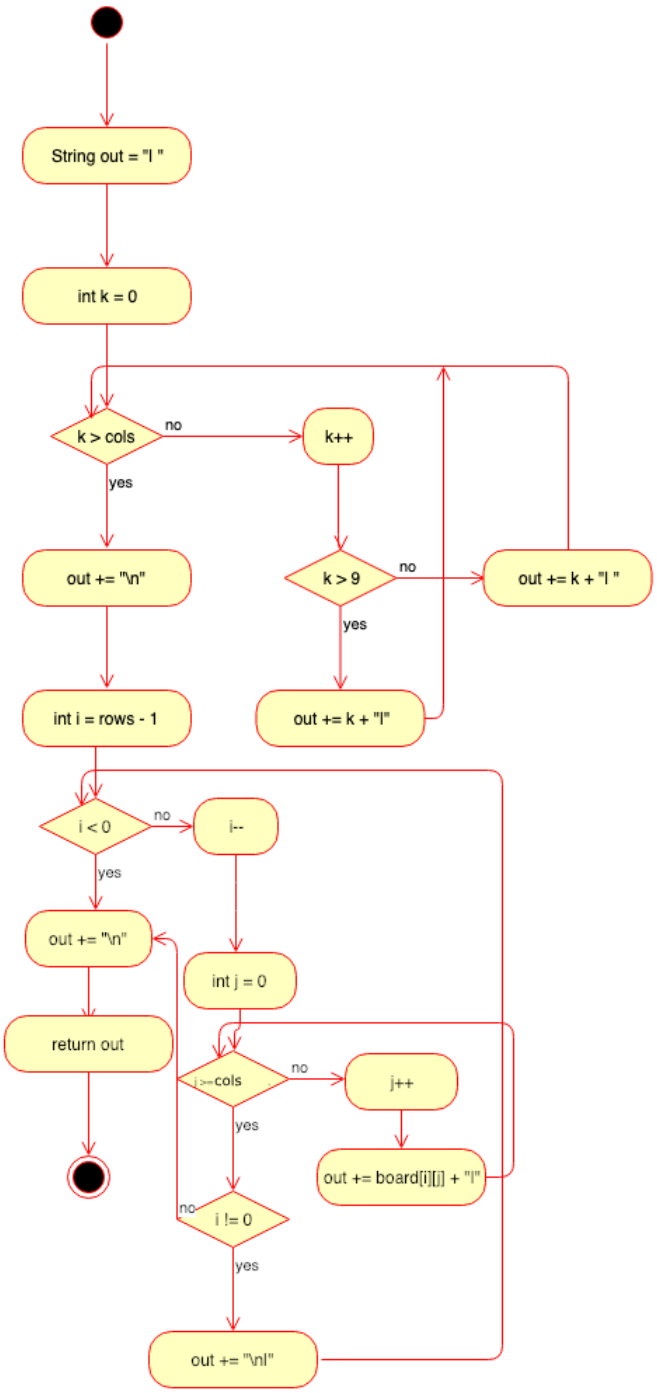
checkDiagWin(int r, int c, char p)


int count = 0, row, col
int diagRows = 0
diagRows >= ROWS - 4
no
yes
diagRows++, count = 0
row = diagRows, col = 0
row >= rows && col >= cols
no
yes
row++, col++
count = 0
yes
no
count++;
count >= inARow
no
yes
return true
row = diagRows, col = COLS - 1
row >= col < 0
yes
no
row++, col--
count = 0
yes
no
count++;
count >= inARow
no
yes
return true
int diagCols = 1
diagCols >= COLS - 4
no
yes
diagCols++, count = 0
row = 0, col = diagCols
row >= ,ows && col >= cols
yes
no
row++, col++
count = 0
yes
no
count++;
count >= inARow
no
yes
return true
int diagCols = cols - 2
diagCols < COLS
no
yes
diagCols--, count = 0
row = 0, col = diagCols
row >= rows && col < 0
yes
no
row++, col--
count = 0
yes
no
count++;
count >= inARow
no
yes
return true
return false

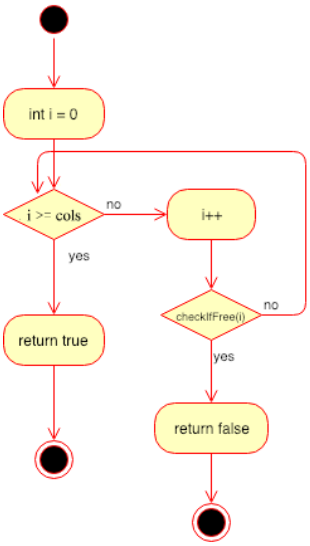
whatsAtPos(int r, int c)



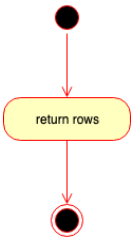
toString ()



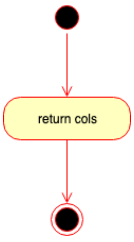
checkIfTie()



getNumRows()

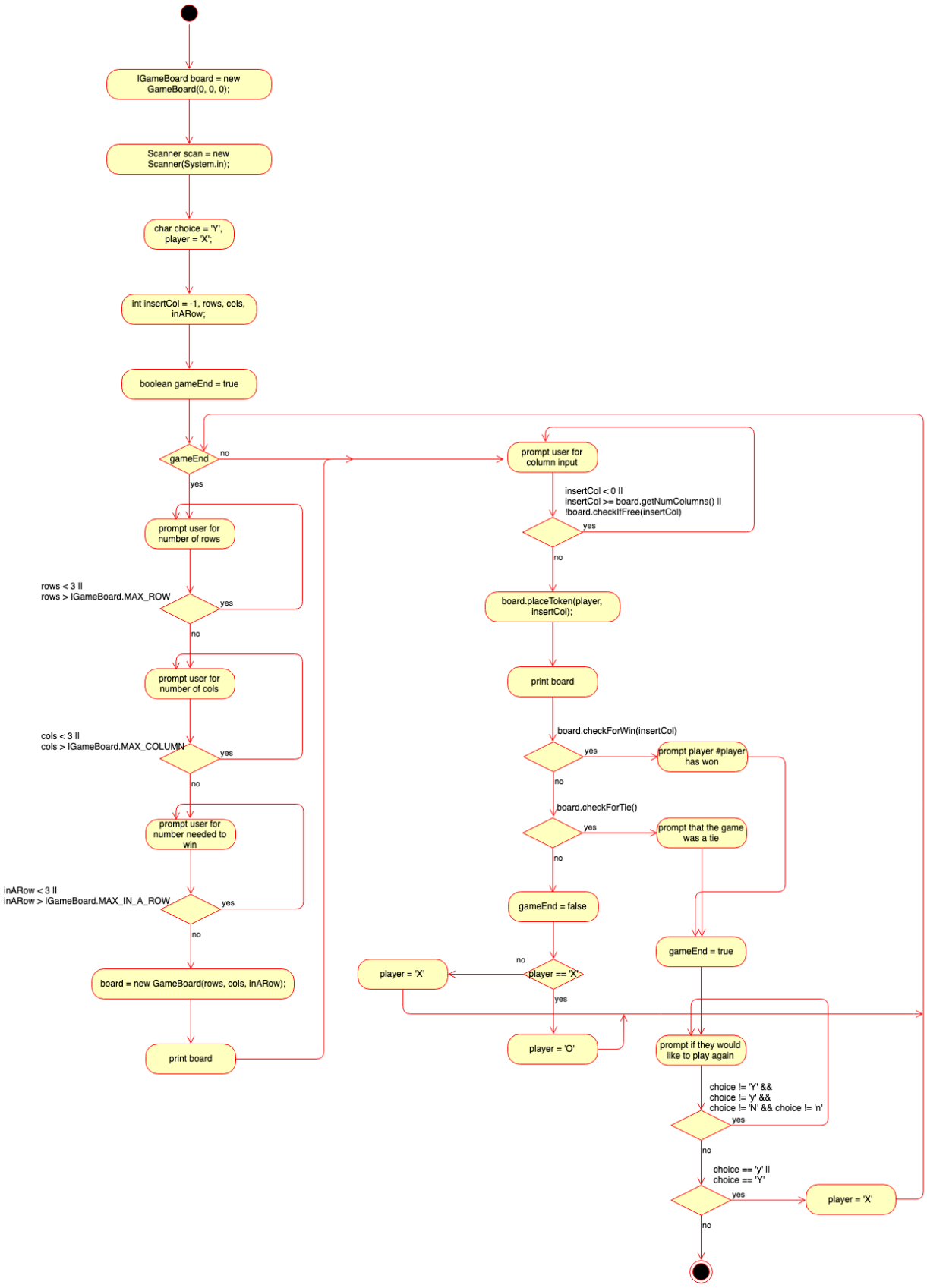


getNumColumns()



getNumToWin()

main(String[] args)



Deployment

1. Navigate to the project directory on your command-line terminal
2. Enter the command “make”
3. Enter the command “make run”