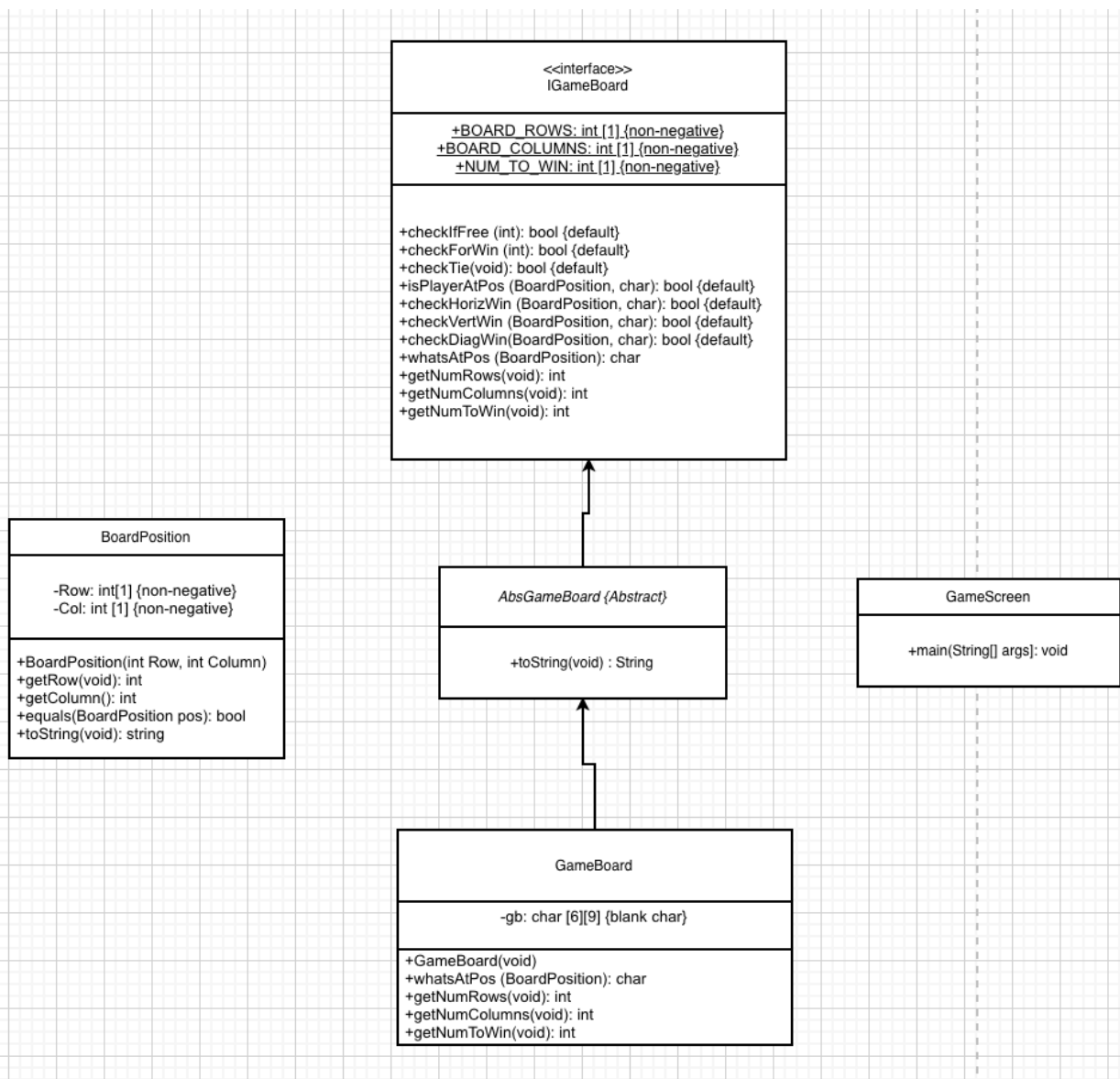


Extended ConnectX

Project 2 Report

Rob Hughes

UML Diagrams



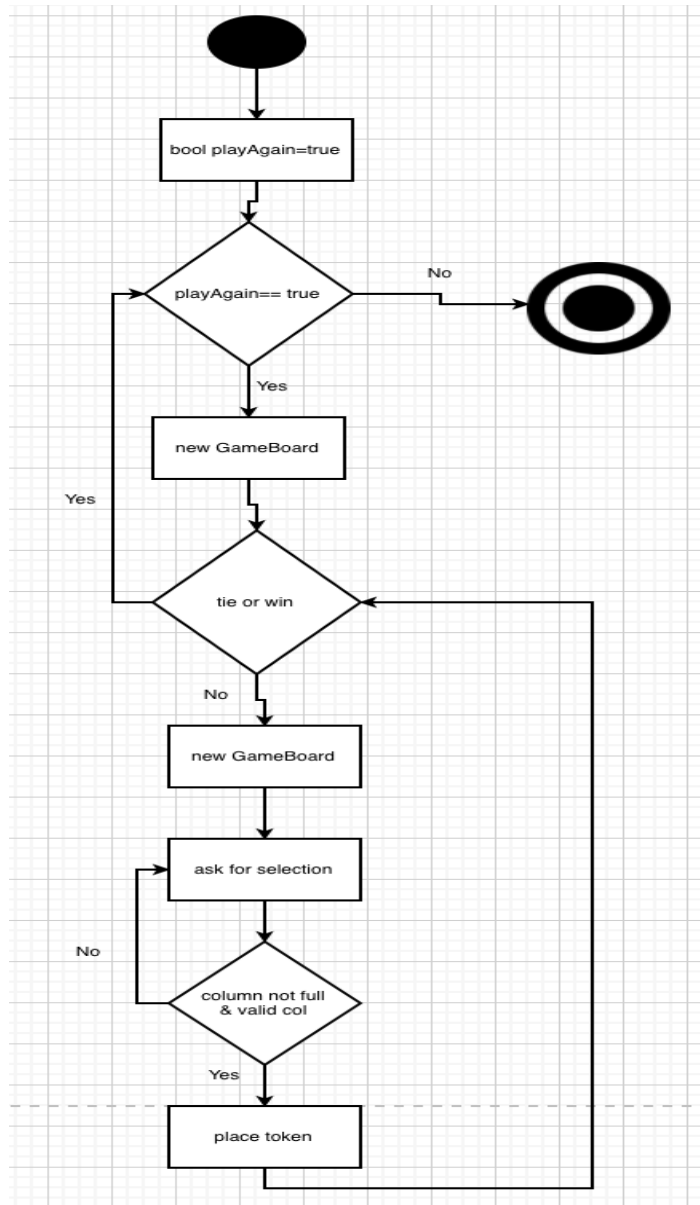
Requirements Analysis	
Functional Requirements (User Stories)	Non-Functional Requirements
<ul style="list-style-type: none"> • As a user, I must be able to pick my column, so I can place my token • As a user, I must be able to see the board so I know where I stand. • As a user, I must be prompted so that I know I incorrectly selected a column. • As a user, I must be prompted to see that I won. Due to out of bounds • As a user, I must be prompted to see that I lost. Horizontally. • As a user, I must be shown the columns so I what to choose from. • As a user, I must be prompted so that I know when to make my move. • As a user, I must be prompted so that I can play again. • As a user, I must be prompted so that I can quit playing. 	<ul style="list-style-type: none"> • 0,0 is bottom left of board • Project runs on Unix • Player X must go first. • The program must be written in Java. • The board must be 6x9. • Player needs 5 in a row to win. • There must only be two players. • There must be javadoc comments. • There must be contracts for each method

- As a user, I must be prompted so that I incorrectly selected a column due to the column being full.
- As a user, I must be prompted to see that I won diagonally.
- As a user, I must be prompted to see that I won vertically.

Activity Diagrams

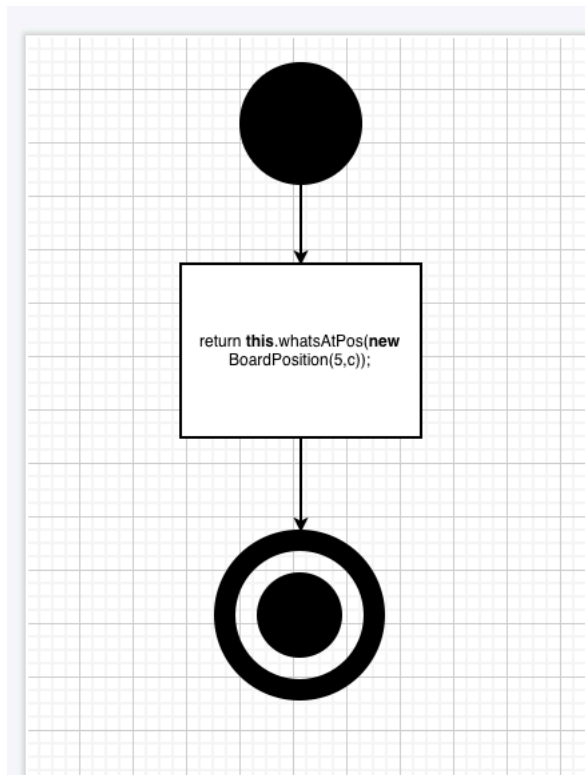
GameScreen.java

main

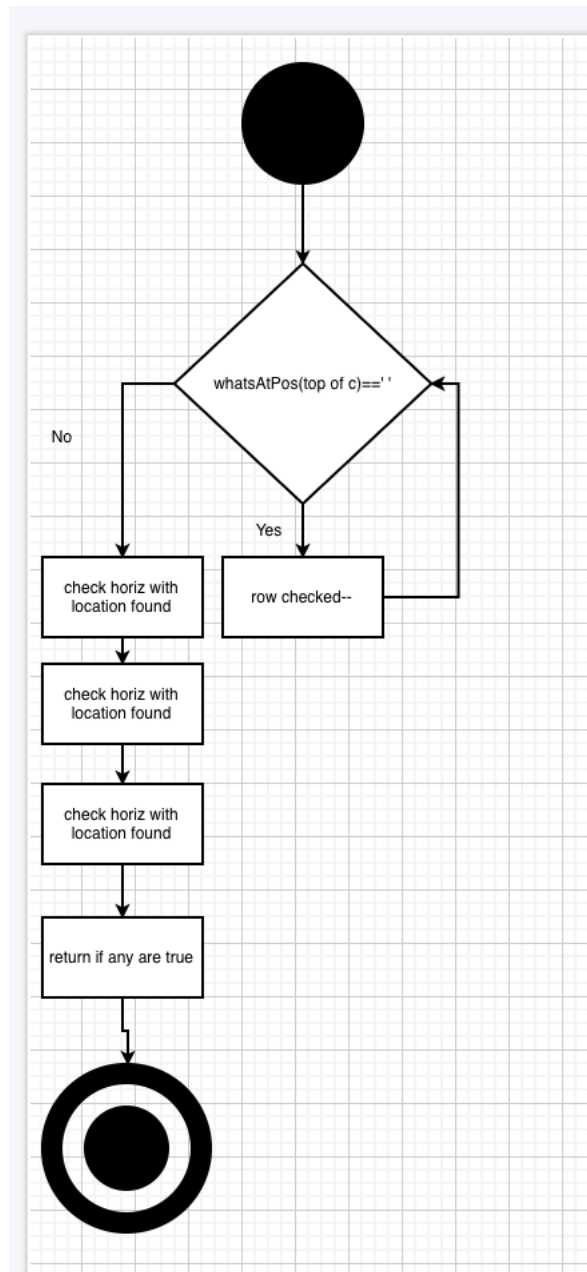


GameBoard.java

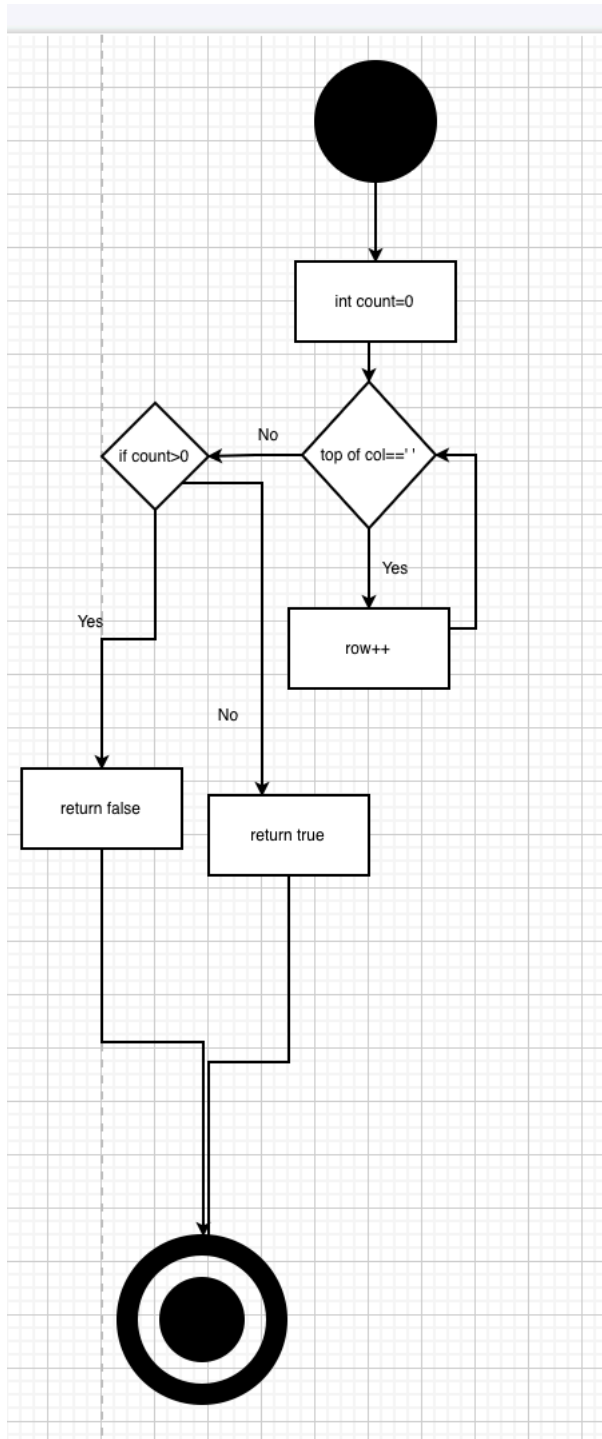
checkIfFree(int c)



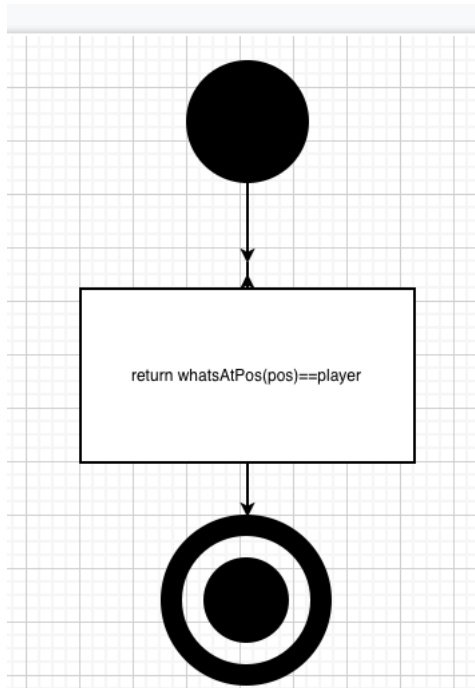
checkForWin(int c)



checkTie

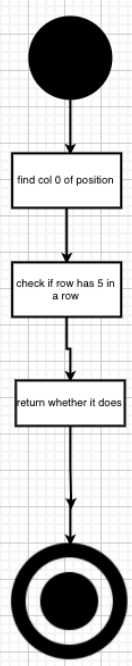


isPlayerAtPos(Boardposition pos, char player_

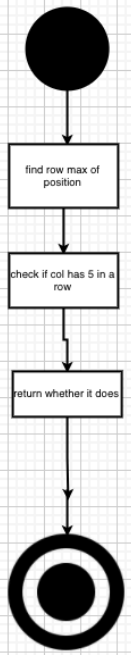


1. checkHorizWin
2. checkVertWin
3. checkDiagonalWin
4. whatsAtPos
5. placeToken
6. findNumRows
7. findNumColumns
8. findNumToWin

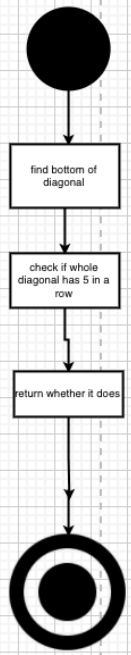
1.



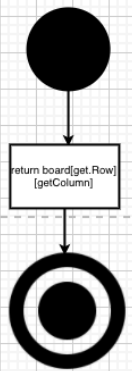
2.



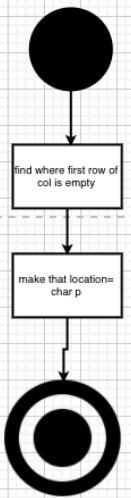
3.



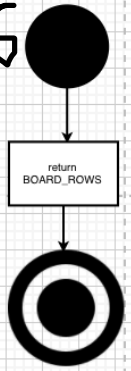
4.



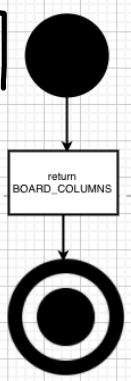
5.



6.



7.



8.

