

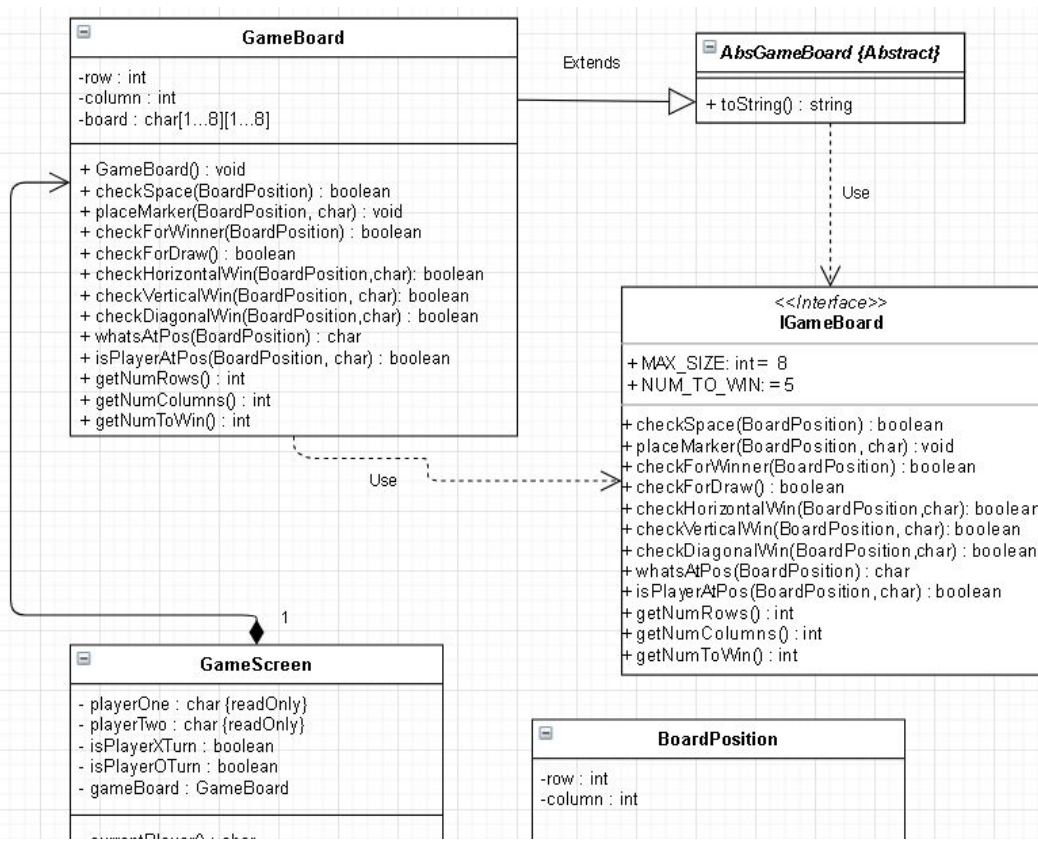
Requirement Analysis

1. As a user I can enter my desired row and column location so my mark can be placed on the board
2. As a user I can play with one more person on the board so I can play with friends
3. As a user I can win the game Vertically/Diagonally/Horizontally so I can maybe play again
4. As a user I can tie the game so I can maybe play again
5. As a user I end the game in a tie so that I can maybe restart
6. As a user I can end the game in a win or loss so that I can maybe restart
7. As a user I can lose the game so I can restart or play again
8. As a user after I either win lose or tie I can chose to play again or close the game so I can play or go do something else
9. As a user I can see the board printed out so I can determine what my next placement is going to be
10. As a user I can choose to either be X or O by choosing what order I want to go in so that I can be on the side I desire
11. As a user if I mis enter a position I can re enter my position so that I can continue playing
12. As a user if I chose a already chosen position I can re enter my position so that I can continue playing

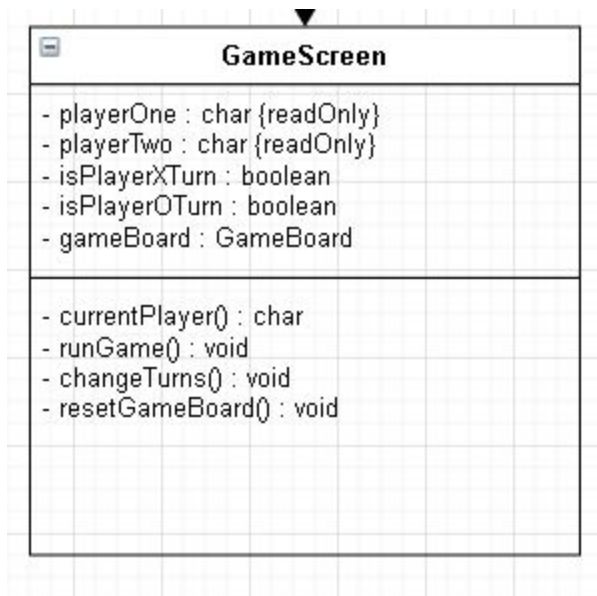
Nonfunctional Requirements

1. Must be written in java
2. Must run on Unix
3. Board is 8x8
4. X always go first
5. (0,0) is top left

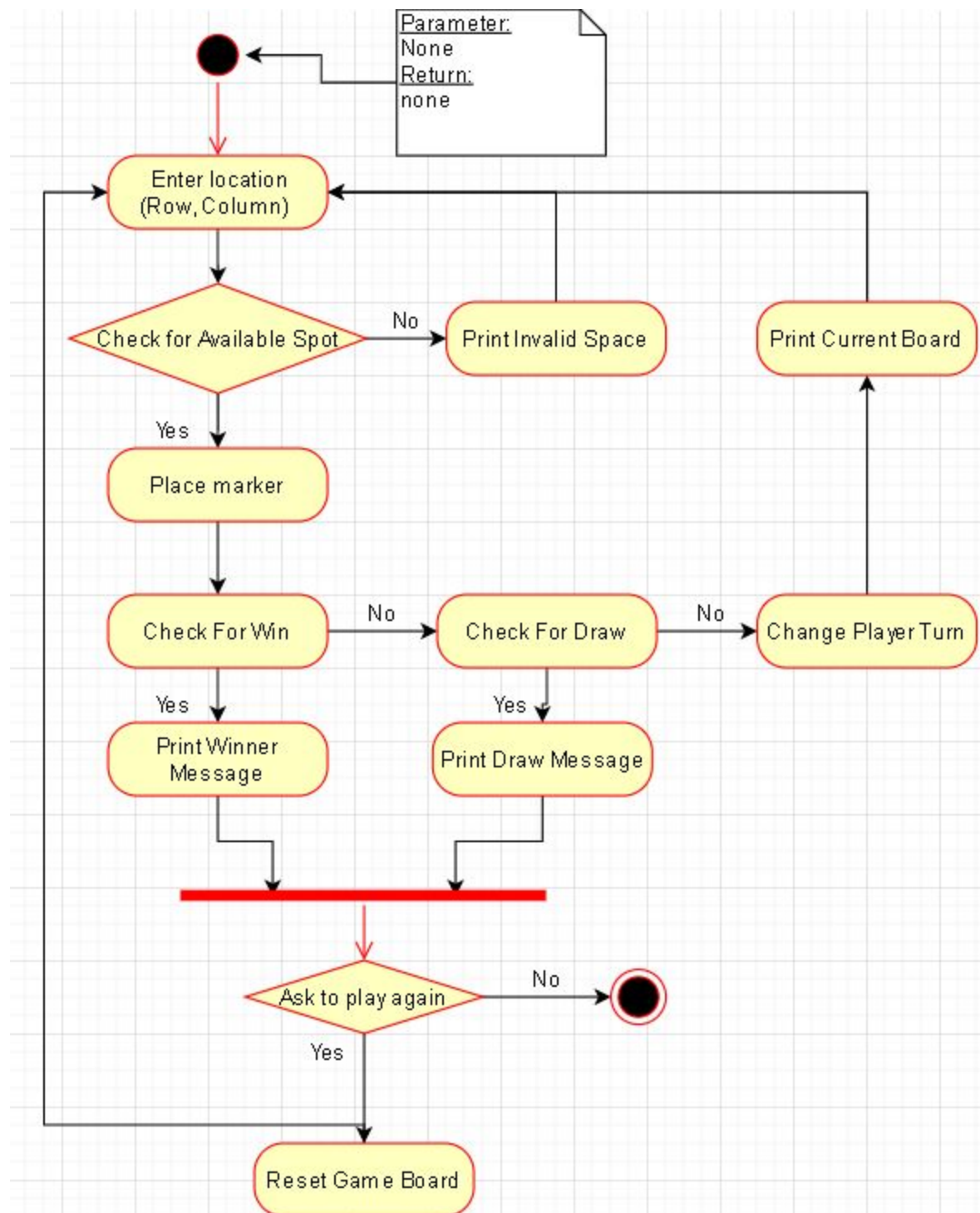
Design -



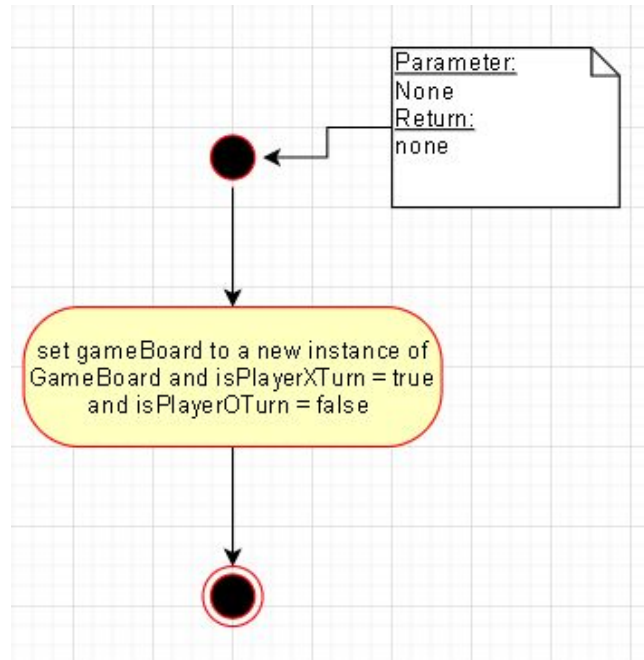
1. Class Diagram of GameScreen



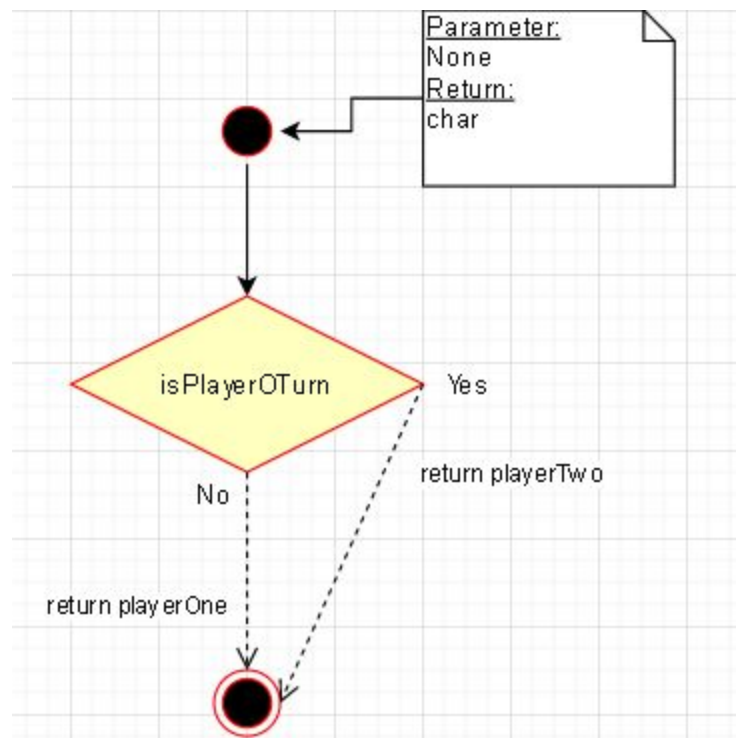
[GameScreen] runGame()



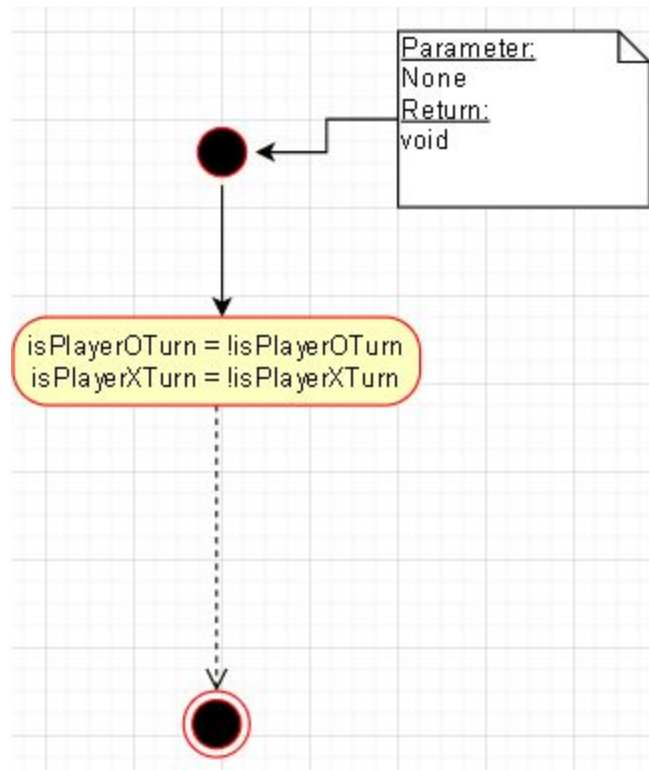
[GameScreen] resetGameBoard()



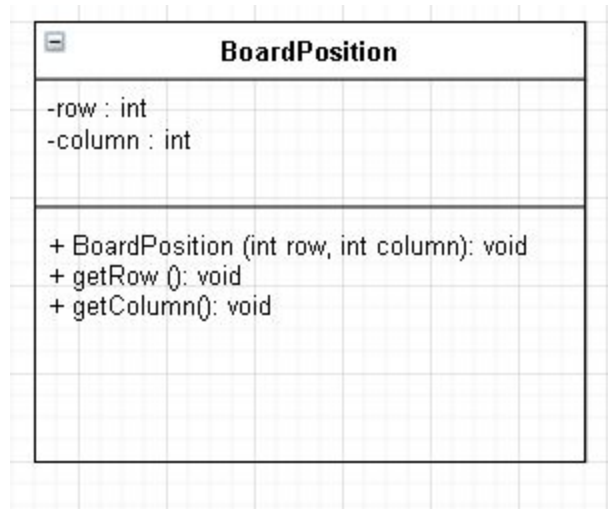
[GameScreen] currentPlayer()



[GameScreen] changeTurns



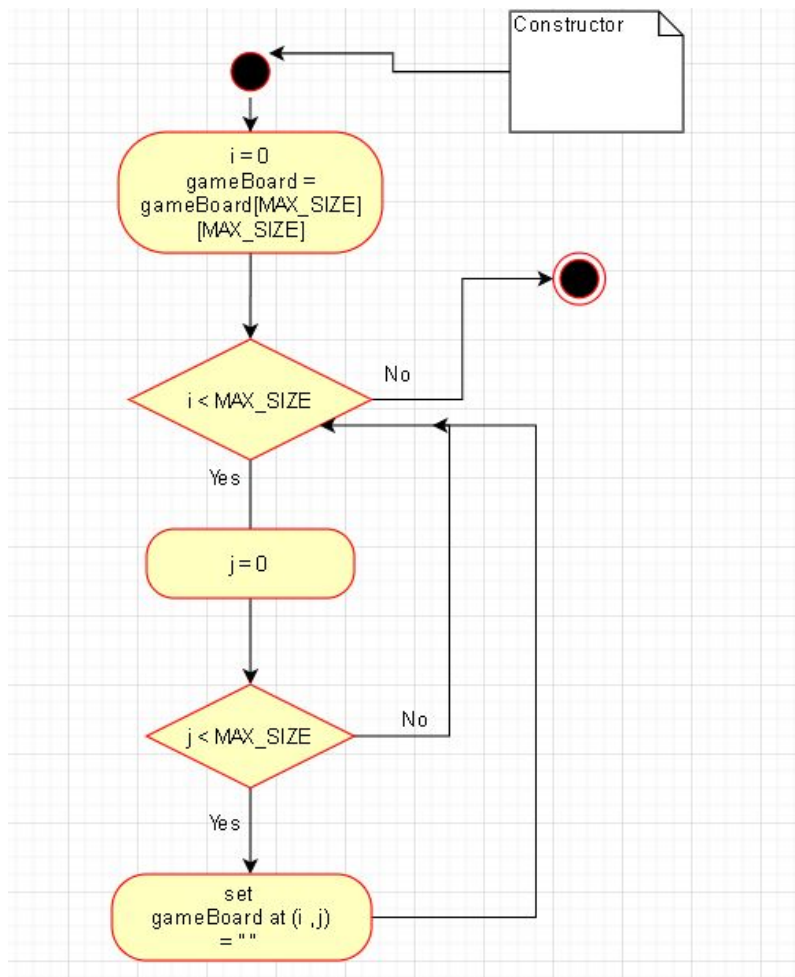
2. Class Diagram of BoardPosition



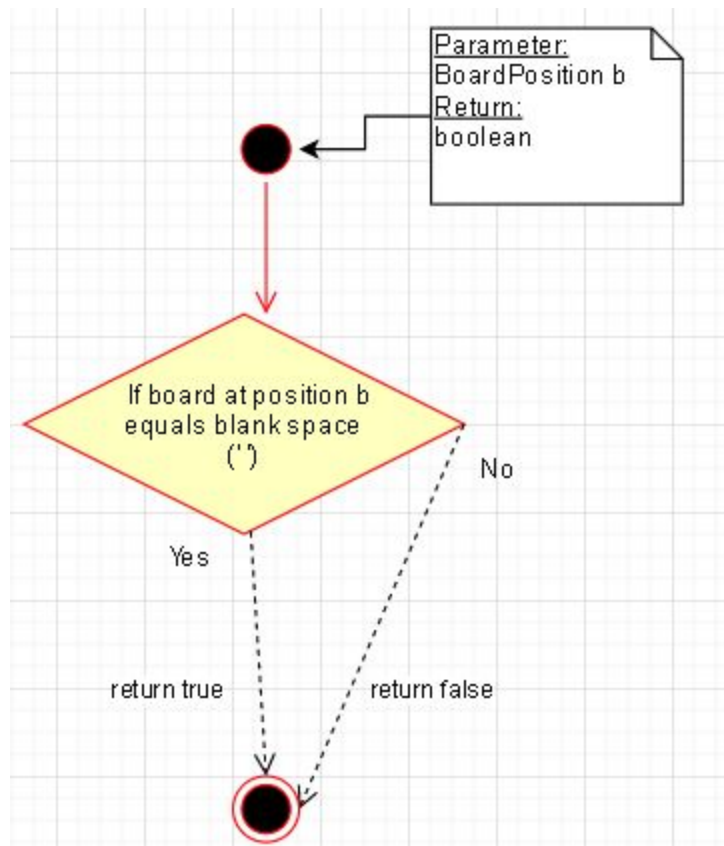
3. Class Diagram of GameBoard



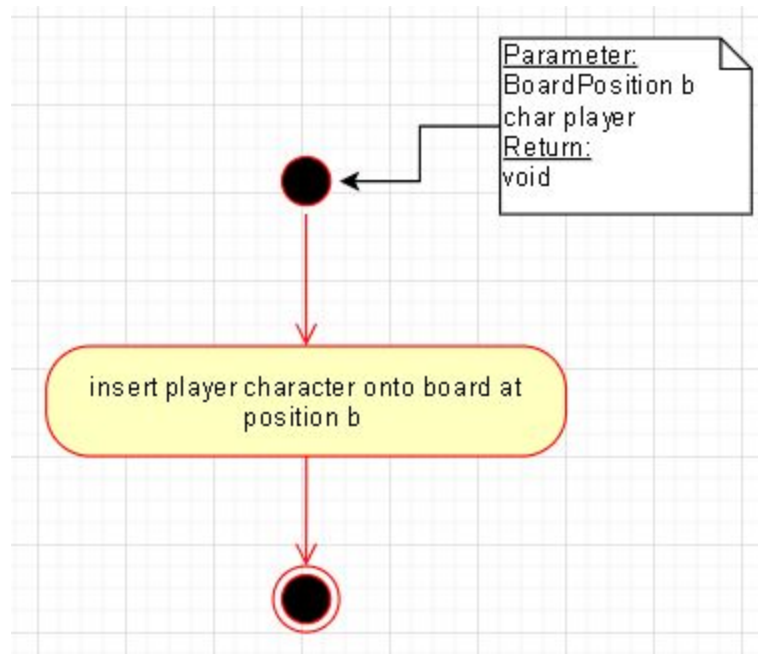
[GameBoard] GameBoard()



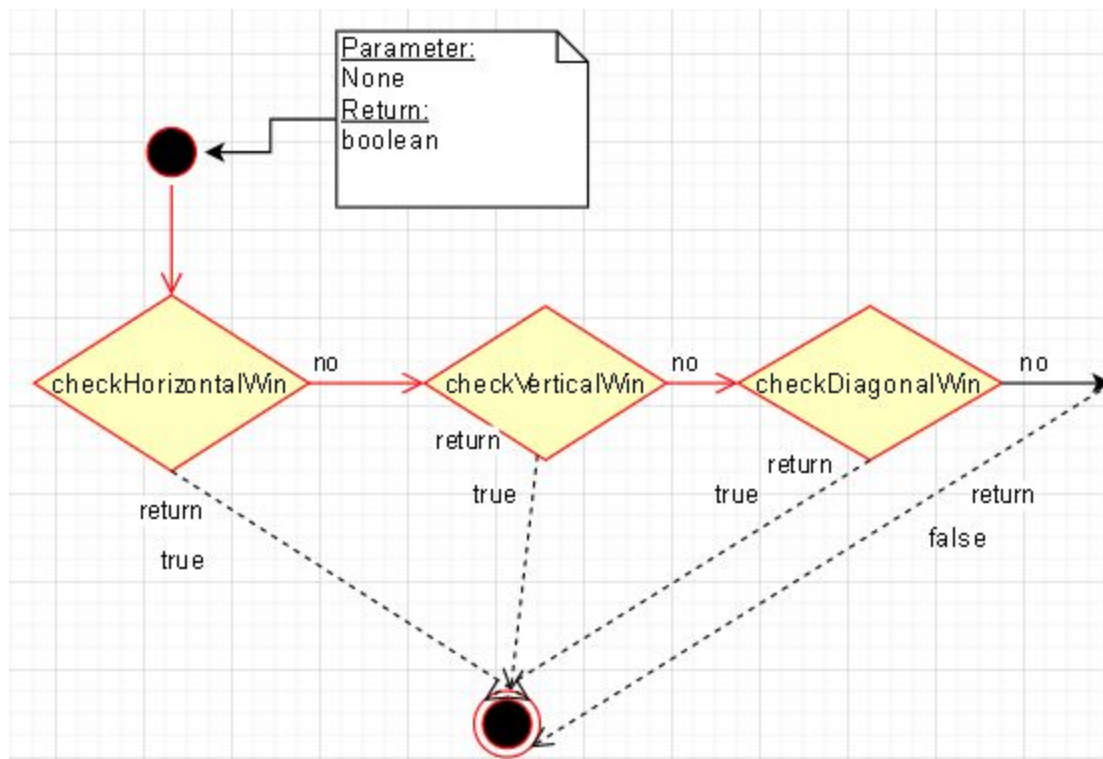
[GameBoard] checkSpace(BoardPosition b)



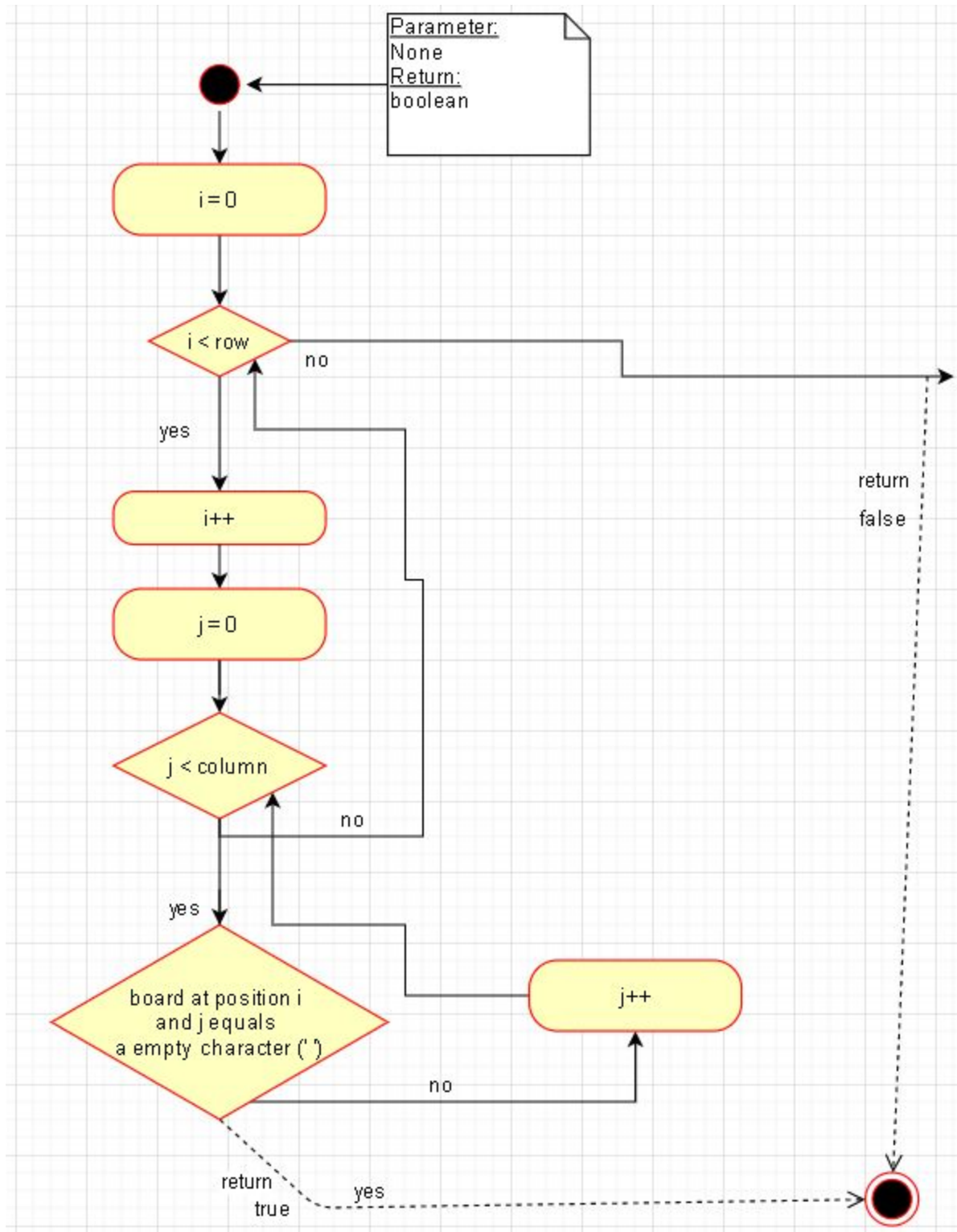
[GameBoard] placeMarker(BoardPosition b, char player)



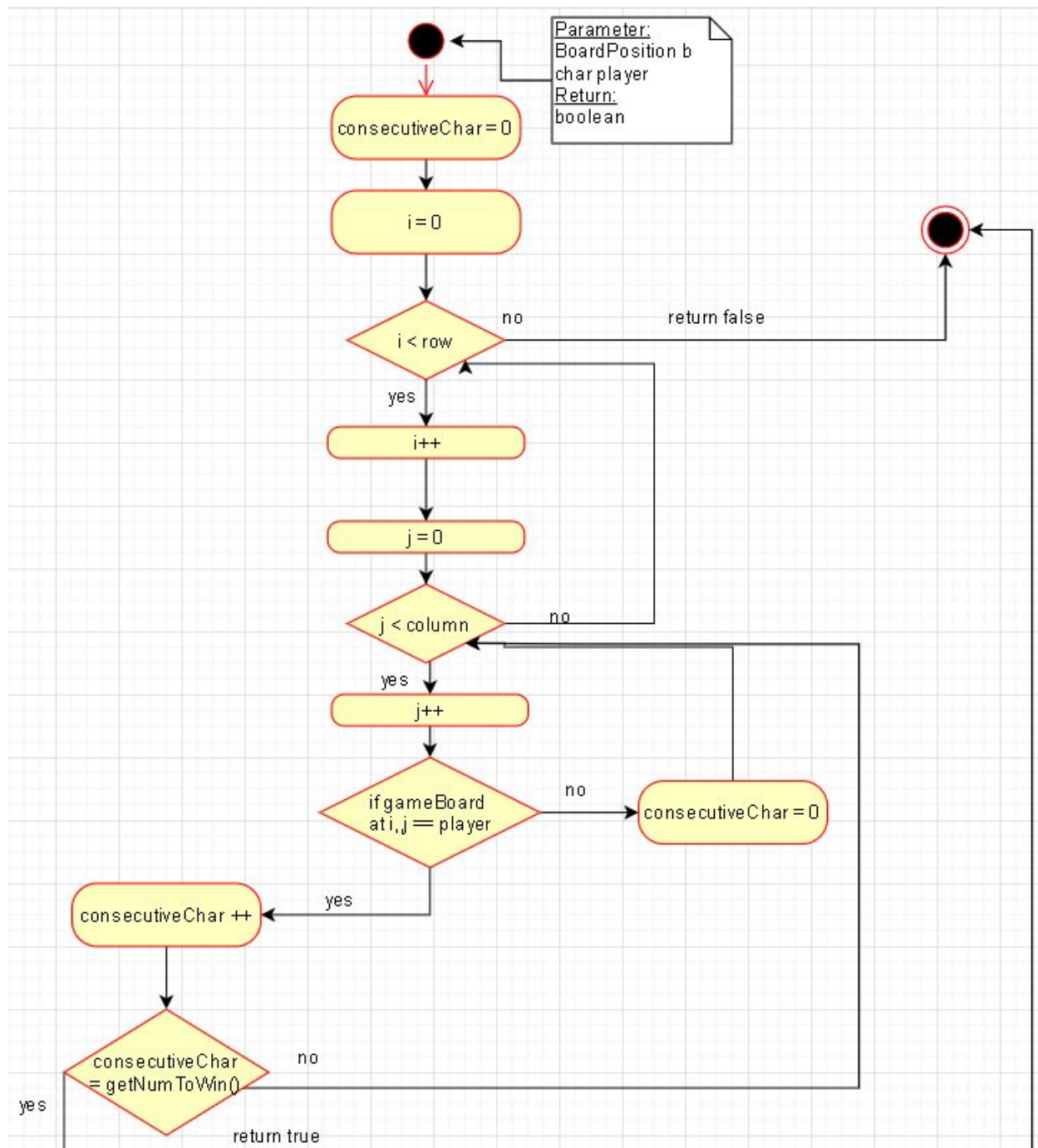
[GameBoard] checkForWinner()



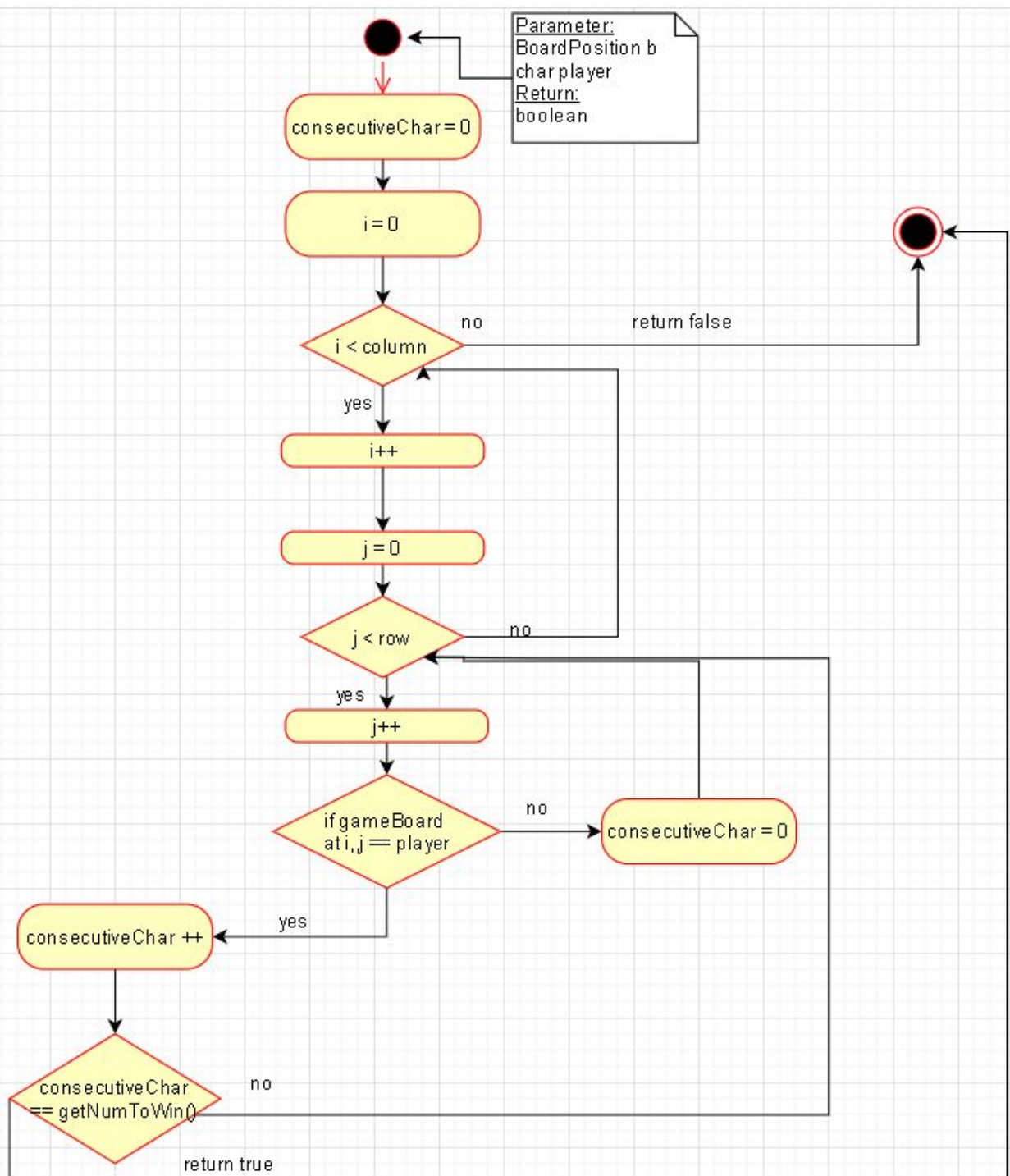
[GameBoard] checkForDraw()



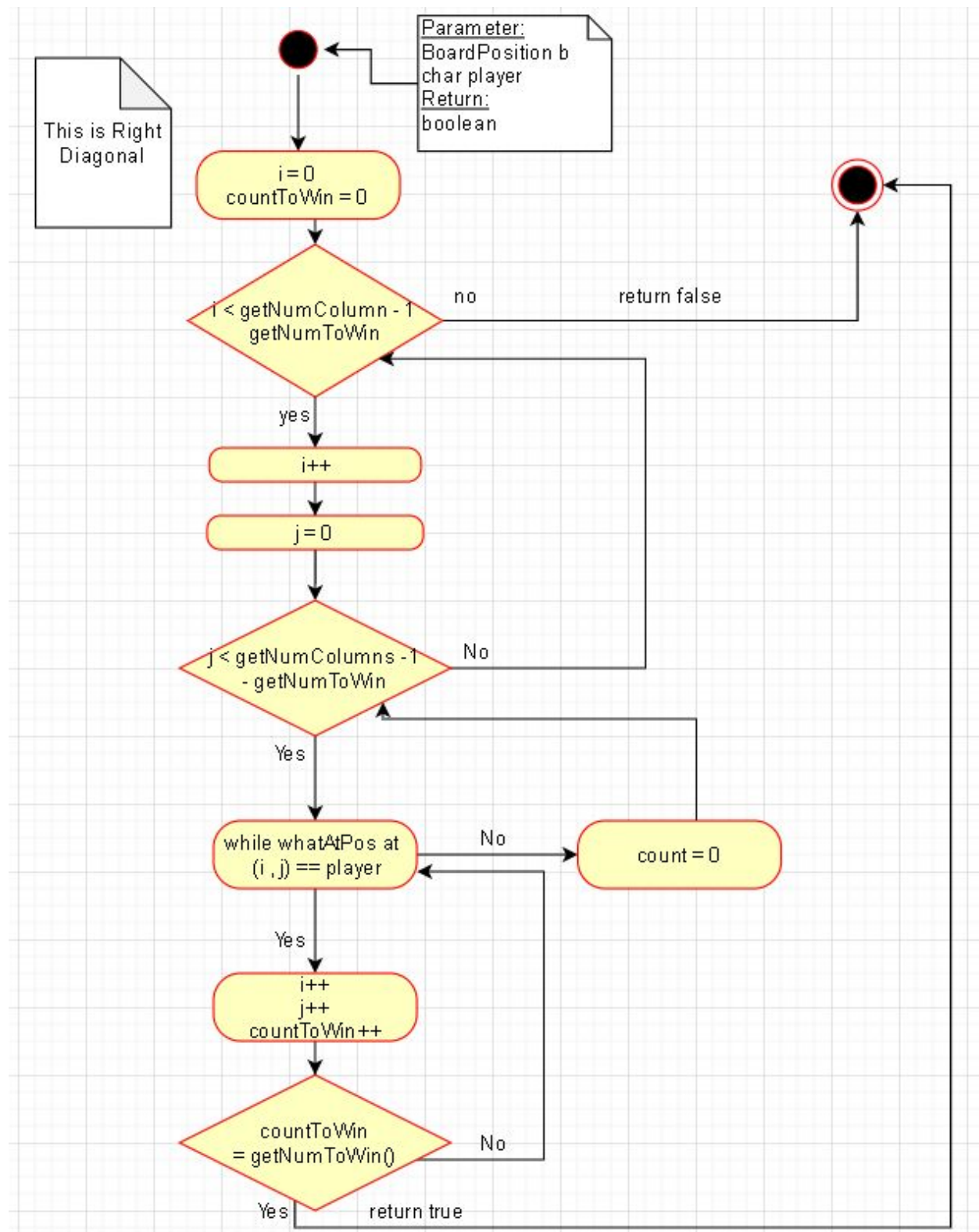
[GameBoard] checkForHorizontalWin(BoardPosition b , char player)

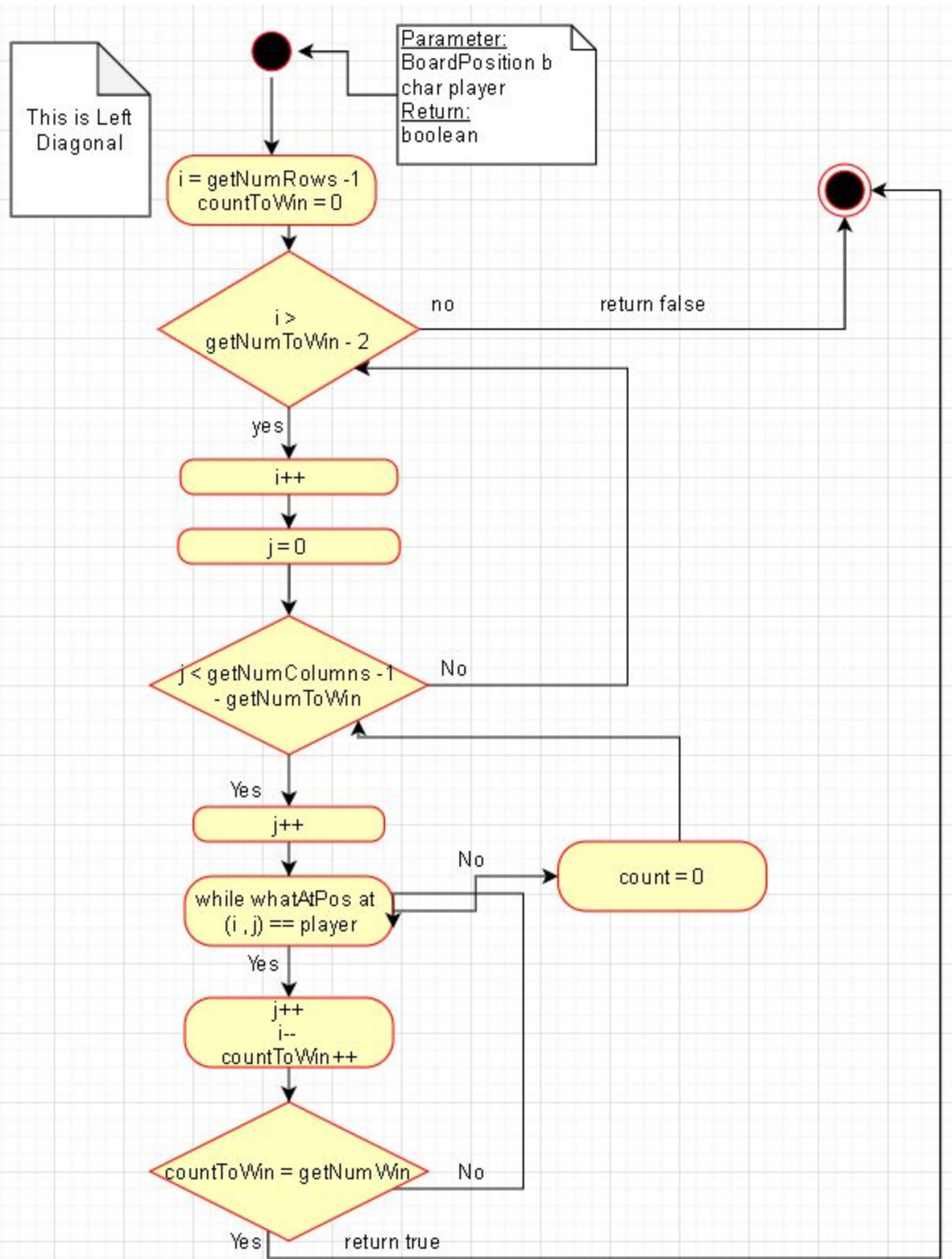


[GameBoard] checkForVerticalWin()

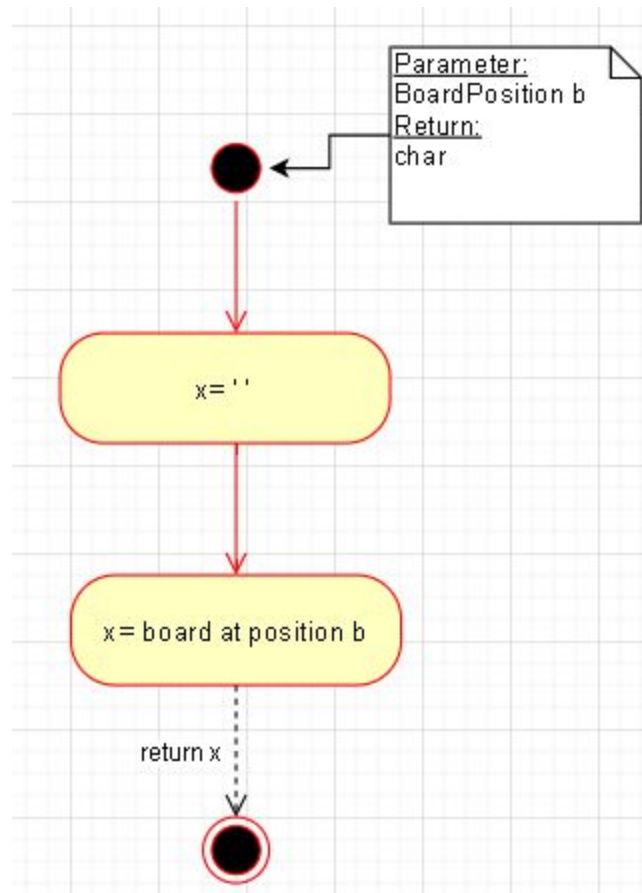


[GameBoard] checkDiagonalWin()

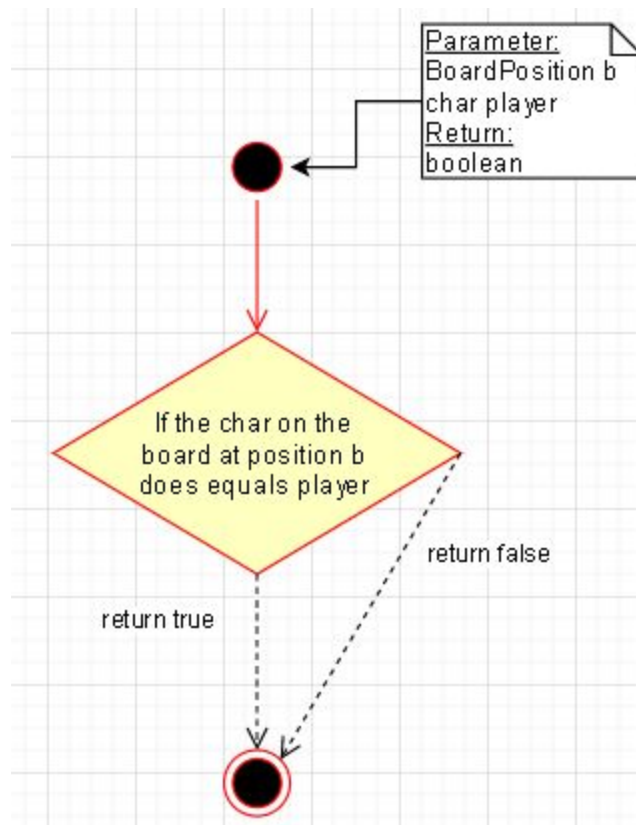




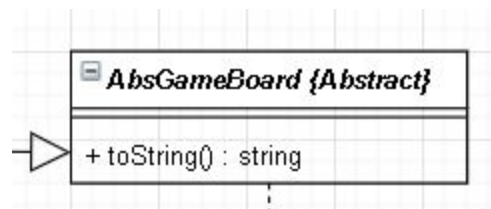
[GameBoard] whatsAtPos()



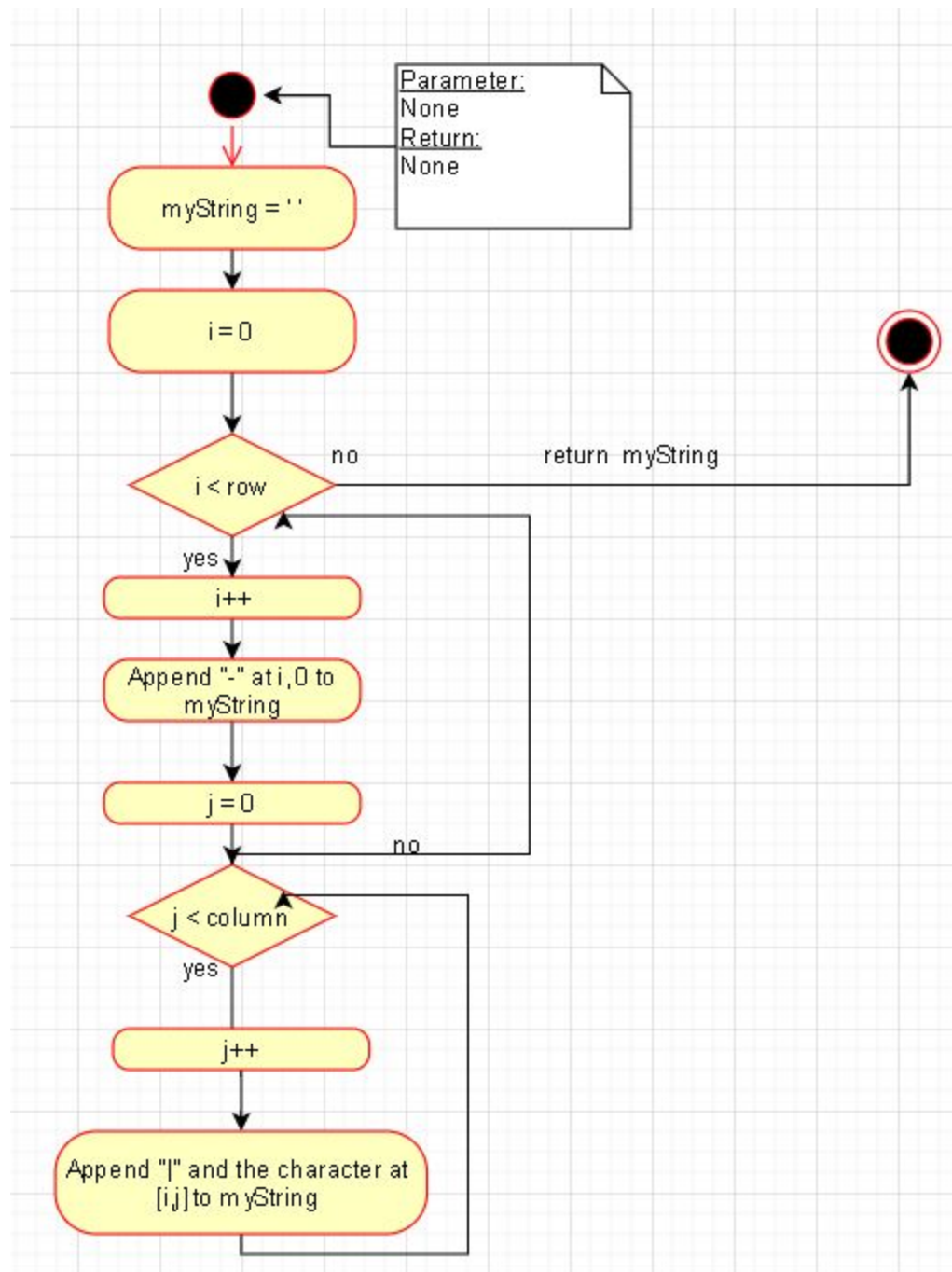
[GameBoard] isPlayerAtPos()



4. AbsGameBoard (Abstract Class)



[AbsGameBoard] toString()



5. IGameBoard (Interface)

