

Game Screen
-Play: boolean[1] -input: string[1] -arraySigns: string[1] -numColumns -numRows -numWin -numPlayers -Player: int[1] -scanner:Scanner[1] -addToList: char[1] -gameCondition: char[1] -gb: Gameboard[1] -gb: GameBoardMem[1] Letter: char[1] Column:int[1] reset:char[1]
+main(void): int

Board Position
-row: int[1] -column: int[1]
-getRow(void): int -getColumn(void): int -==(BoardPosition,BoardPosition): bool

Game Board
-HEIGHT: int[1] -WIDTH: int[1] -turnNumber: int[1] -MAXTURNS: int[1] -numToWin: int[1] -Array: char[][]
+getNumRows(void): int +getNumColumns(void): int +getNumtoWin(void): int +empty(void): void + checkIfFree(int): bool + checkForWin(int,char): bool + checkTie(void): bool + placeToken(char, int): void + checkHorizWin(BoardPosition, char): bool + checkVertWin(BoardPosition, char): bool + checkDiagWin(BoardPosition, char): bool + whatsAtPos(BoardPosition): char + isPlayerAtPos(BoardPosition, char): bool + toString(void): String

Game Board Mem
-HEIGHT: int[1] -WIDTH: int[1] -turnNumber: int[1] -MAXTURNS: int[1] -numToWin: int[1] -Array: Map<Character,List<BoardPosition>>
+getNumRows(void): int +getNumColumns(void): int +getNumtoWin(void): int +empty(void): void + checkIfFree(int): bool + checkForWin(int,char): bool + checkTie(void): bool + placeToken(char, int): void + checkHorizWin(BoardPosition, char): bool + checkVertWin(BoardPosition, char): bool + checkDiagWin(BoardPosition, char): bool + whatsAtPos(BoardPosition): char + isPlayerAtPos(BoardPosition, char): bool + toString(void): String

User Stories

1. As a user, I should be able to use the command line.
2. As a user, I should be able to launch the code.
3. As a player, I should be able to start a game.
4. As a player, I should be able to place my symbol on my turn.
5. As a player, I should be able to replace a token if I choose an out-of-bounds column
6. As a player, I should be able to take a turn after my opponent
7. As a player, I can win by getting 5 in a row horizontally.
8. As a player, I can win by getting 5 in a row vertically.
9. As a player, I can win by getting 5 in a row diagonally.
10. As a player, I should be able to quit the game.
11. As a player, I should be able to choose to play again.
12. As a player, I should be able to see the board between turns.
13. As a player, I should be able to see who's turn it is.
14. As a player, I should be able to see who won at the end.
15. As a player, I should be told the instructions to play the game.
16. As a player, I should be able to enter the size of the array's height
17. As a player, I should be able to resize the array height if I choose an out-of-bounds value
18. As a player, I should be able to enter the size of the array's width
19. As a player, I should be able to resize the array width if I choose an out-of-bounds value
20. As a player, I should be able to enter the size of the array's number in a row to win
21. As a player, I should be able to reset the number to win height if I choose an out-of-bounds value
22. As a player, I should be able to enter the number of players in the game
23. As a player, I should be able to change the number of players if I choose an out-of-bounds value
24. As a player, I should be able to enter my token letter
25. As a player, I should be able to change my token if I choose an out-of-bounds value
26. As a player, I should be able to choose to play a fast game
27. As a player, I should be able to choose to play a memory efficient game
28. As a player, I should be able to choose a fast or memory efficient game again if I choose an out-of-bounds value
29. As a player, I should be able to resize the array height if I choose to play again
30. As a player, I should be able to resize the array width if I choose to play again
31. As a player, I should be able to change the numberToWin if I choose to play again
32. As a player, I should be able to change the number of players if I choose to play again
33. As a player, I should be able to change he player tokens I choose to play again
34. As a player, I should be able to choose a fast or memory efficient game if I choose to play again

Nonfunctional Requirements

1. The program should run on UNIX.
2. The program should be built to run on clemson server computers.
3. The program should be able to be launched.
4. The program should not crash.
5. The program should not emit audio.
6. The program should allow the user(s) to play connect five on a 6x9 board.
7. The program should not allow the user to play games other than connect five.
8. The program should run in the terminal.
9. The program should be able to take command line input.
10. The program must function while the user is running it.
11. The code should be written in java.
12. The program should run quickly at all times.
13. The program should make player 1 go first
14. The Program's board should set 0,0 to the bottom left of the array
15. The program's board should set a board of board size (MaxCols,MaxRows)
16. The program should allow the number of columns to be between 3 and 100
17. The program should allow the number of rows to be between 3 and 100
18. The program should allow the number of players to be between 2 and 10
19. The program should allow the number needed to win to be between 3 and the shorter of number of columns and number of rows

Makefile Commands and documentation

Default: Compiles the .java files

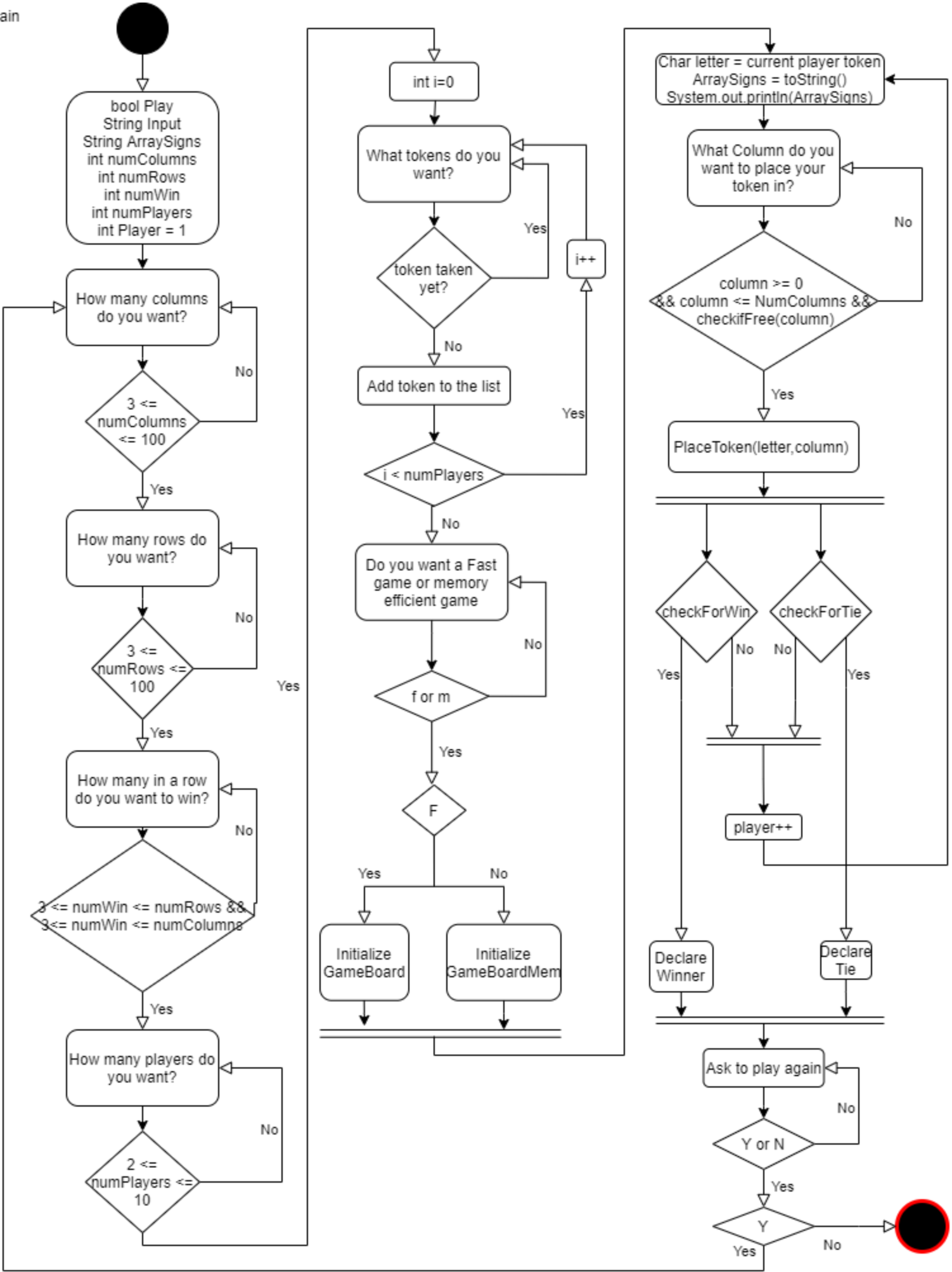
Run: Runs the compiles class files

Test: Compiles the test cases

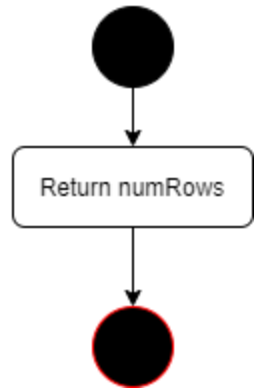
TestGB: runs the gameboard test cases

TestGBMem: runs the gameboardmem test cases

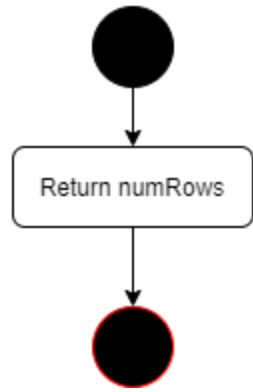
Clean: removes all class files



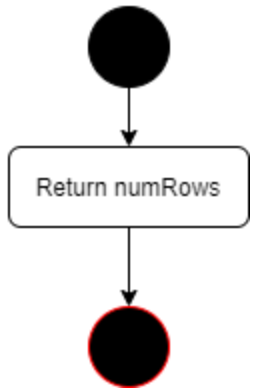
GetNumRows



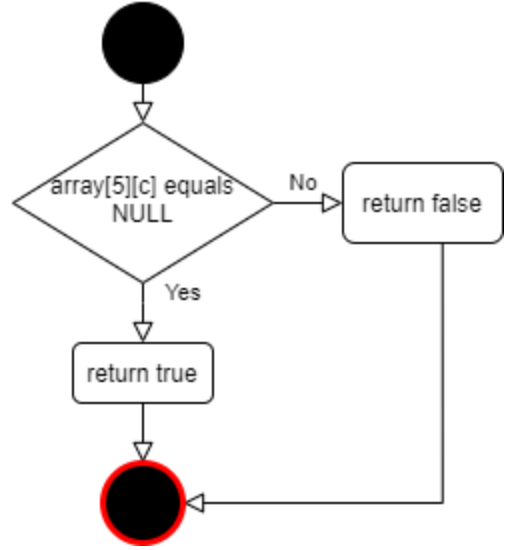
GetNumColumns



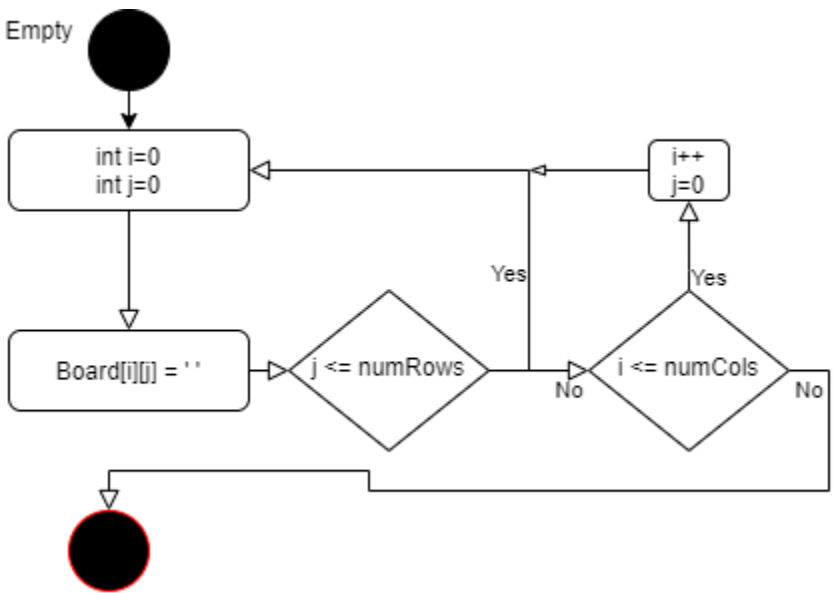
GetNumToWin



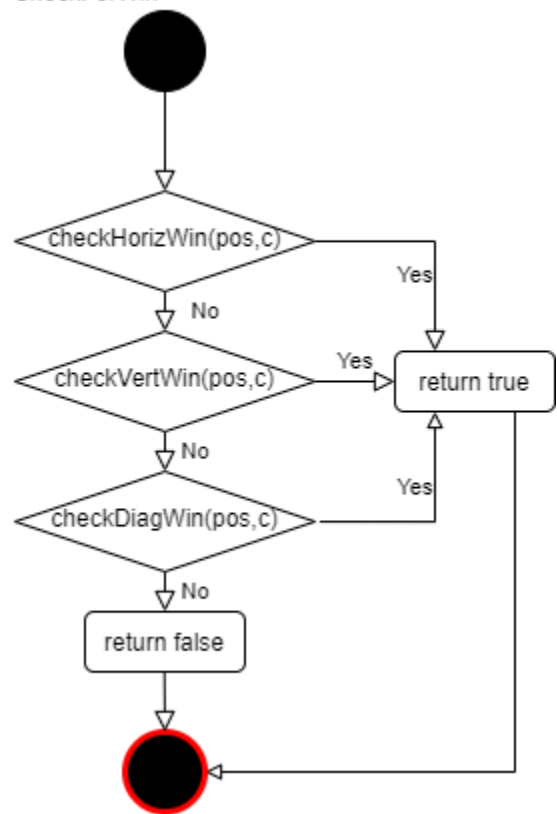
checkIfFree



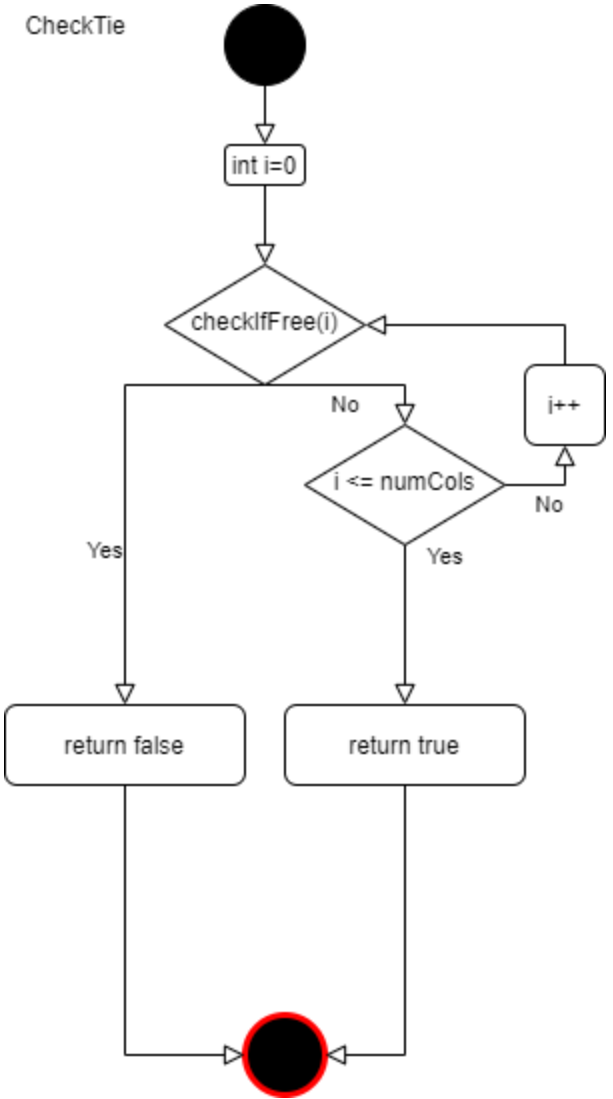
Empty



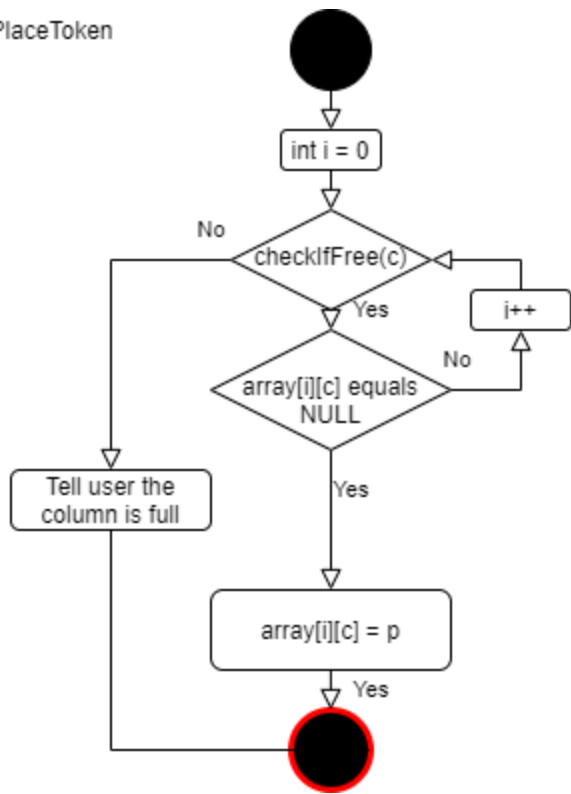
CheckForWin



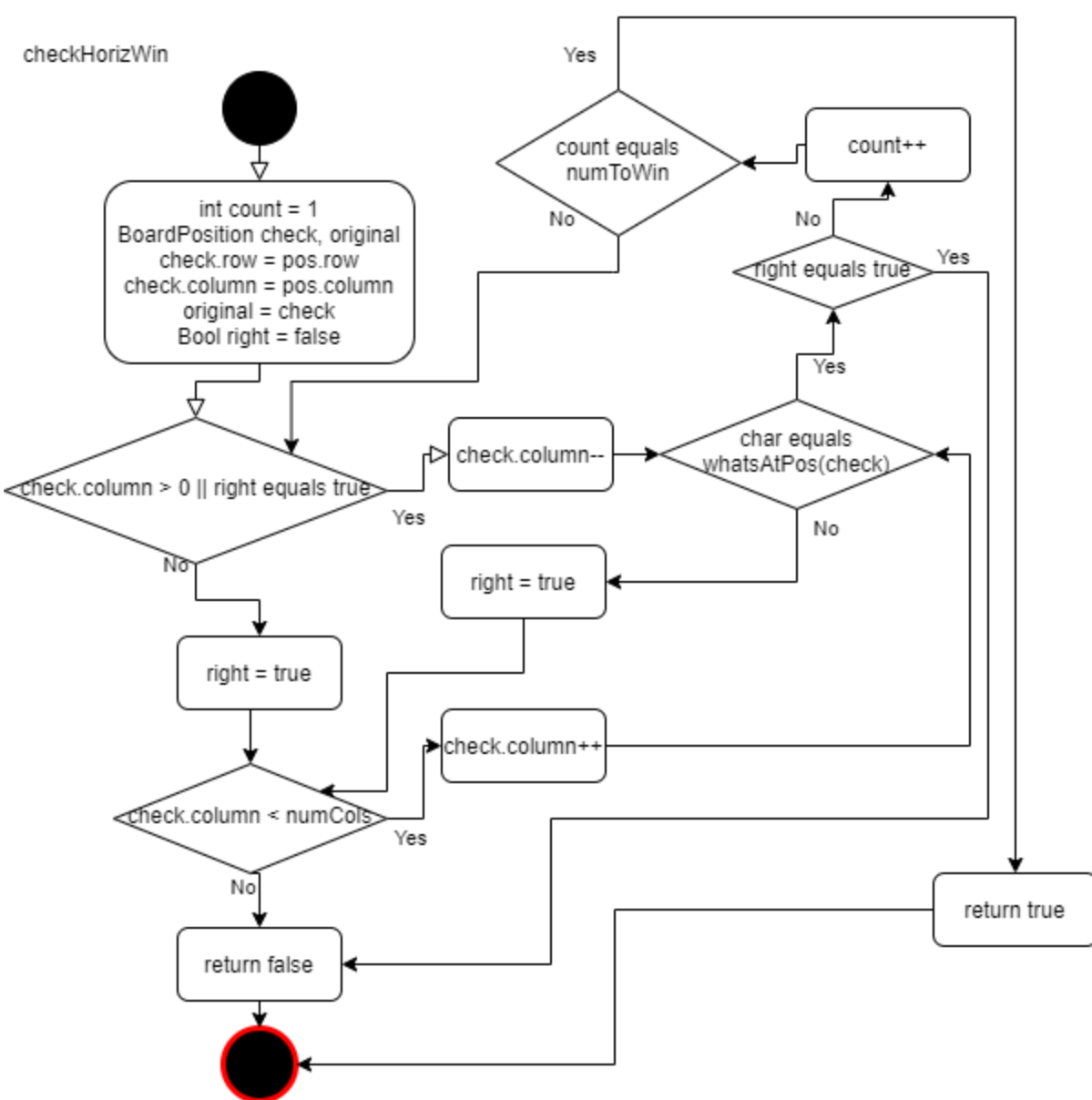
CheckTie



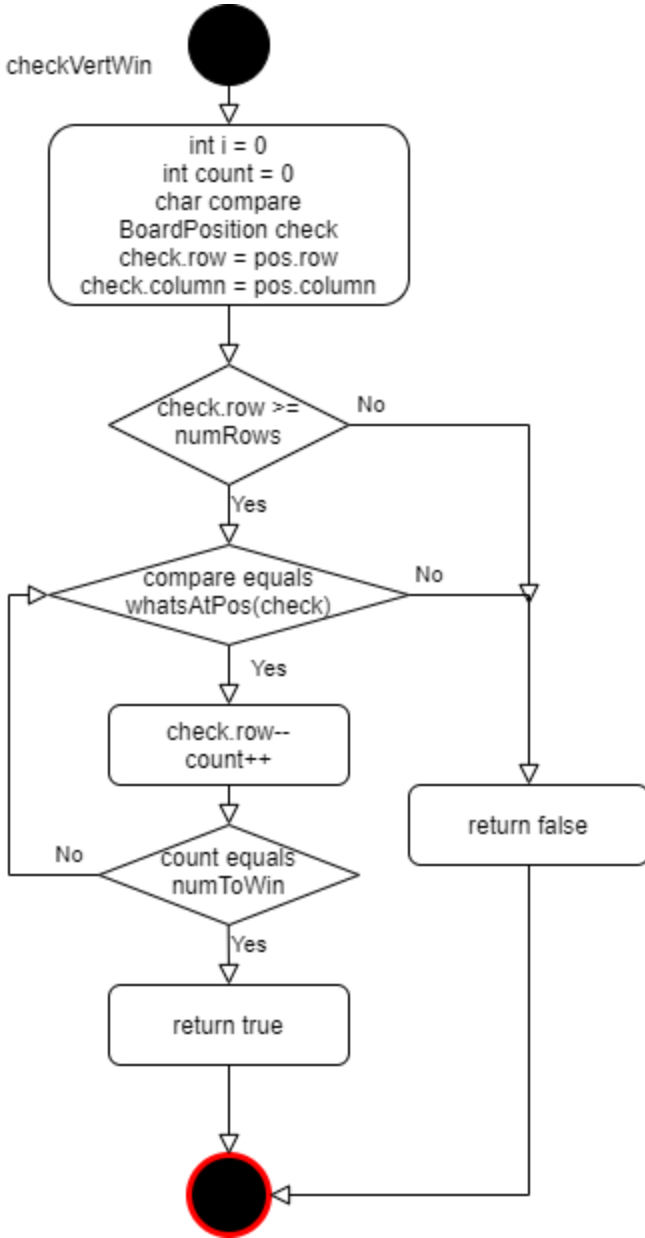
PlaceToken

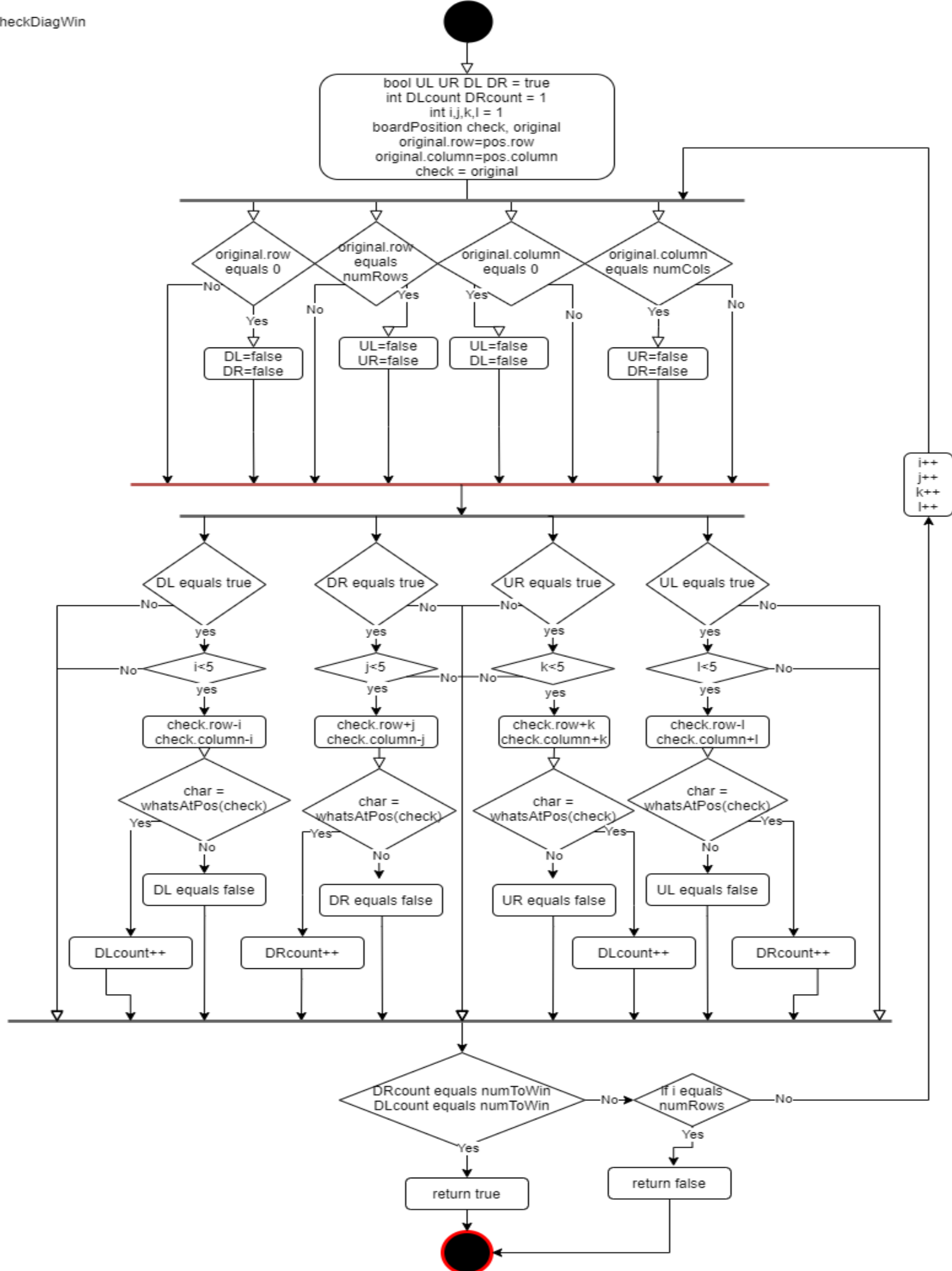


checkHorizWin

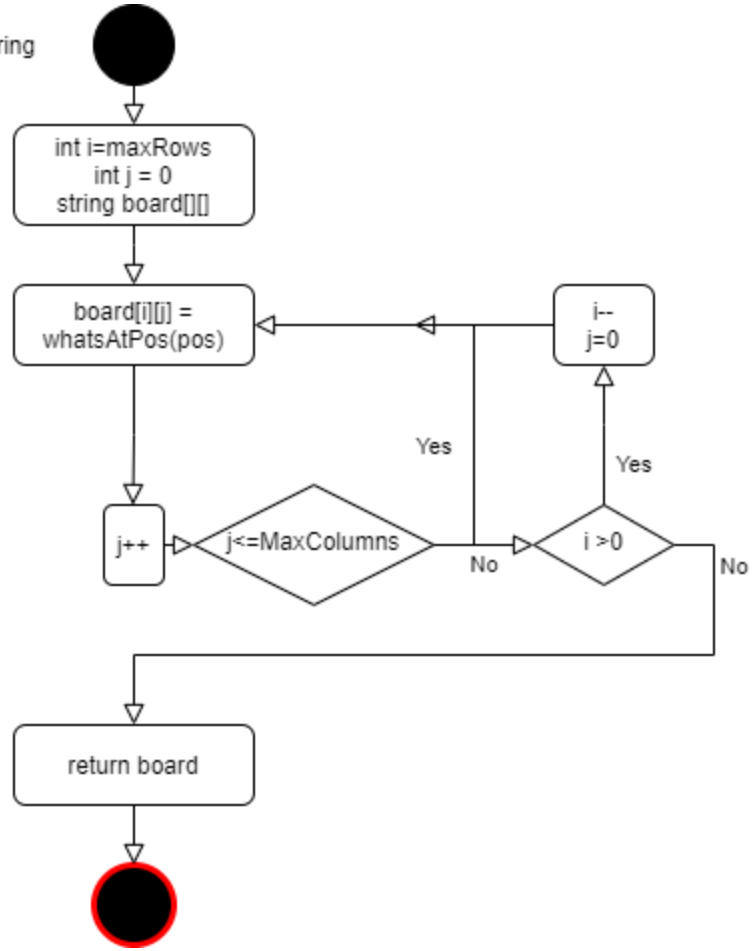


checkVertWin

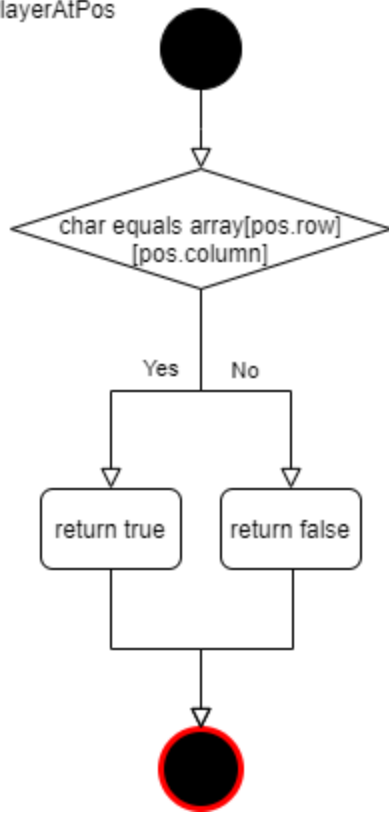




toString



isPlayerAtPos



WhatsAtPos

