

CPSC 2150 Project Two Report

Jake Macdonald

Functional Requirements:

1. As a player I can enter the desired column and row location for my marker to progress the game.
2. As a player I need the marker type start with a value of X at the start of a game for consistent playthroughs.
3. As a player I need the marker type to switch value for every other move to make this a versus game.
4. As a player I need indication of inappropriate input so that I may correct the mistake.
5. As a player I need an updating display of the board so I may watch the game progress.
6. As a player I need to know when win conditions have been satisfied so the game may conclude.
7. As a player I need to know when the game has been drawn so I may be aware to restart the game.
8. As a player I need to be prompted after concluding to game to begin another, so I don't need to rerun the application.
9. As a player I need the ability to place and store a market at desired positions to play the game.
10. As a player I need the board to be designed as a 5 X 8 grid to play a specific variety of tic-tac-toe.
11. As a player I need the win condition to be a line of 5 adjacent similar markings so to play a specific variety of tic-tac-toe.
12. As a player I need the gameboard display to be expressed in a readable manner so I may bear greater witness.
13. As a player, the unmarked locations of the board should possess the default value, ' ' so it may be clear where available places positions are.
14. As a player, the placed marker value should be either 'X', or 'O', so player progress may be clearly seen.

Non-Functional Requirements

1. The application must be developed in Java.
2. The application must function in the Ubuntu v.20 environment.
3. The class, GameScreen will possess the only main function of the program.
4. The application must exclusively use three classes: GameBoard, GameScreen, and BoardPosition.
5. The class GameBoard must exclusively use methods prescribed within assignment documentation.
6. Attributes of the class, BoardPosition, must exclusively be accessible by getter methods.
7. All U/I interaction will be exclusively preformed within the GameScreen method.
8. BoardPosition will have only one constructor method.
9. BoardPosition attributes may only be set within the constructor.
10. BoardPosition must have a methods overriding the equals() and toString() methods.
11. All attributes of GameBoard must be private unless they are static and final.
12. Gameboard is of size 5 X 8
 - I think this is more of a functional requirement than non because this requirement is made explicitly apparent while in use. But I got points off last time for this not being here.
13. X always goes first
14. Five in a row is the win condition
15. GameBoard will extend AbsGameboard.
16. AbsGameBoard implements IGameBoard.
17. Board element (0,0) should be top position of the board
18. No dead code should be present in the project
19. Makefile should have targets : default, run, and clean

makefile instructions:

make: Compiles GameBoard, AbsGameBoard, IGameBoard, BoardPosition, and GameScreen

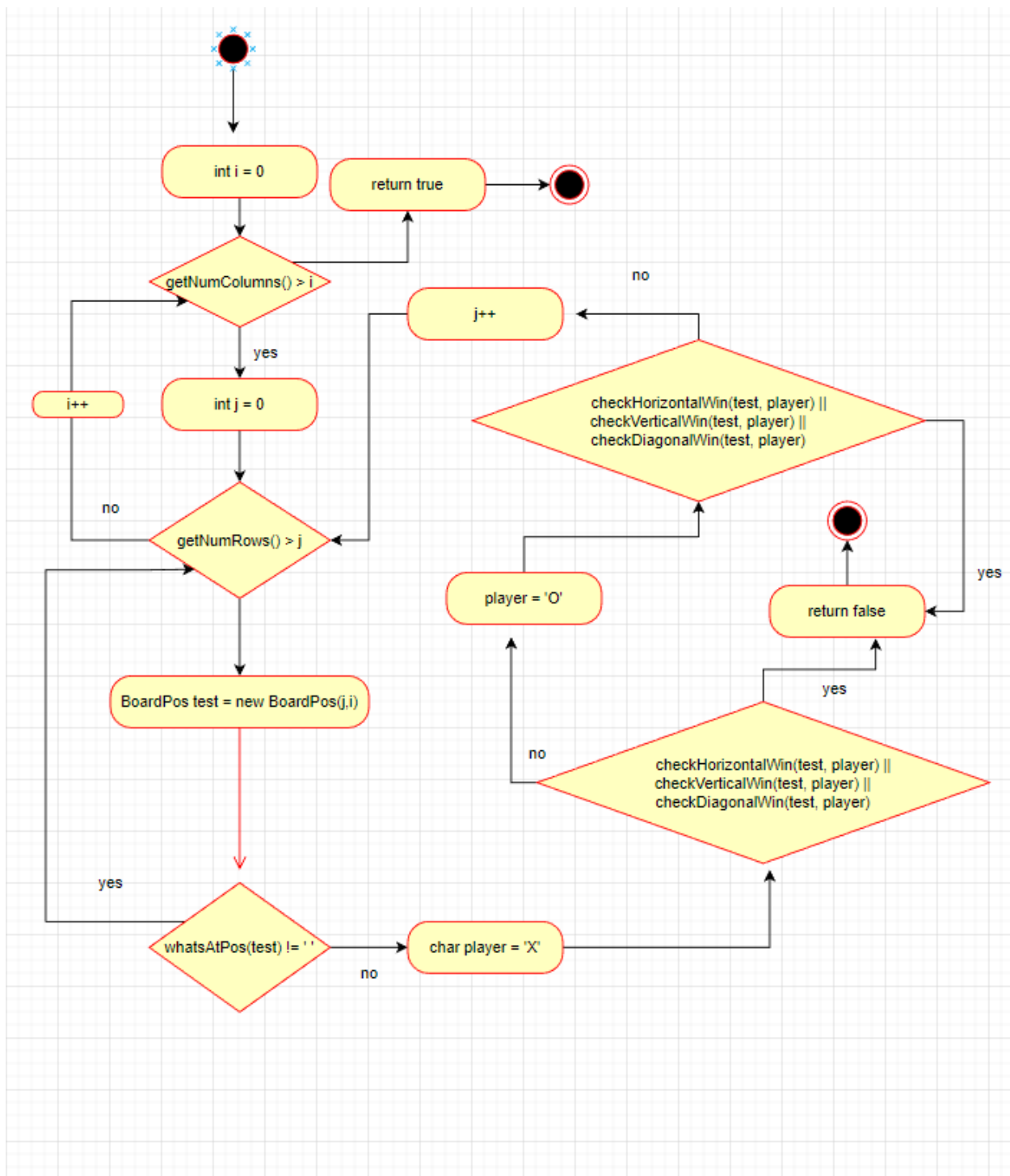
make run: Executes GameScreen.class

make clean: Deletes the GameScreen.class file, and deletes all class files in the models directory

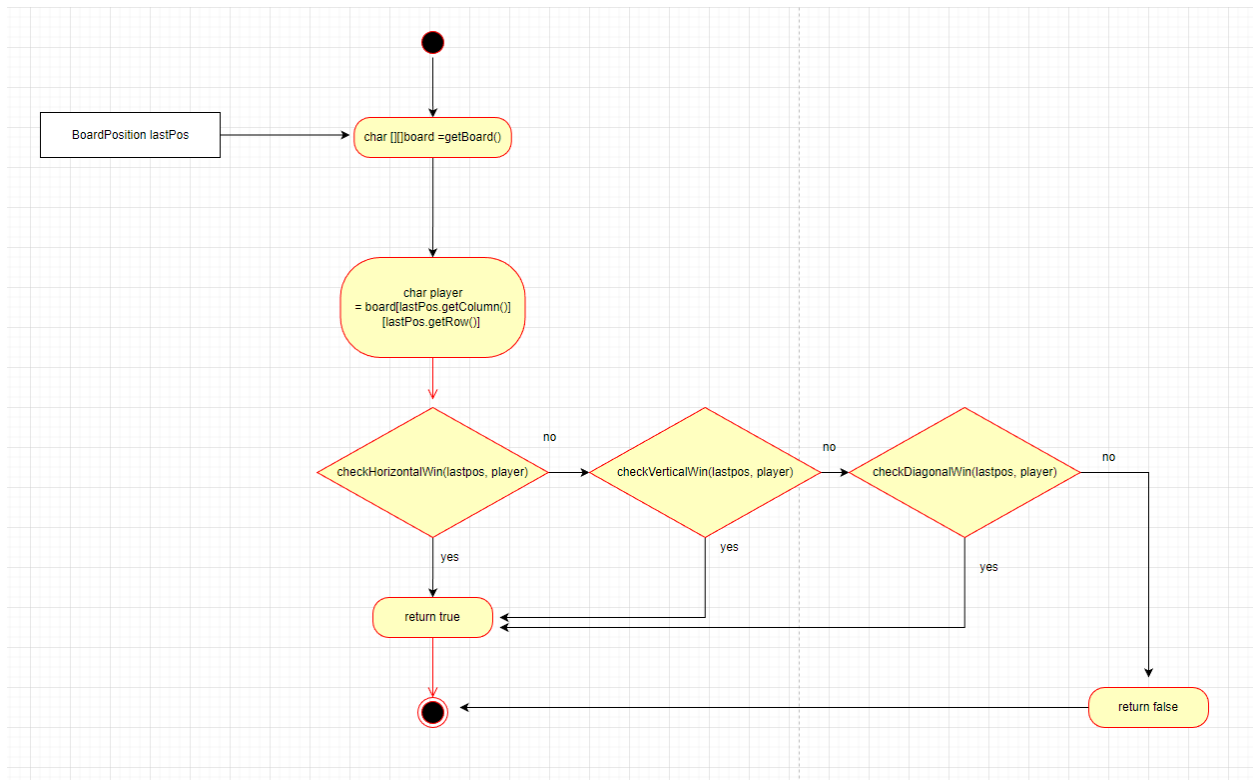
make zip: zips all files needed for project submission

UML activity Diagrams:

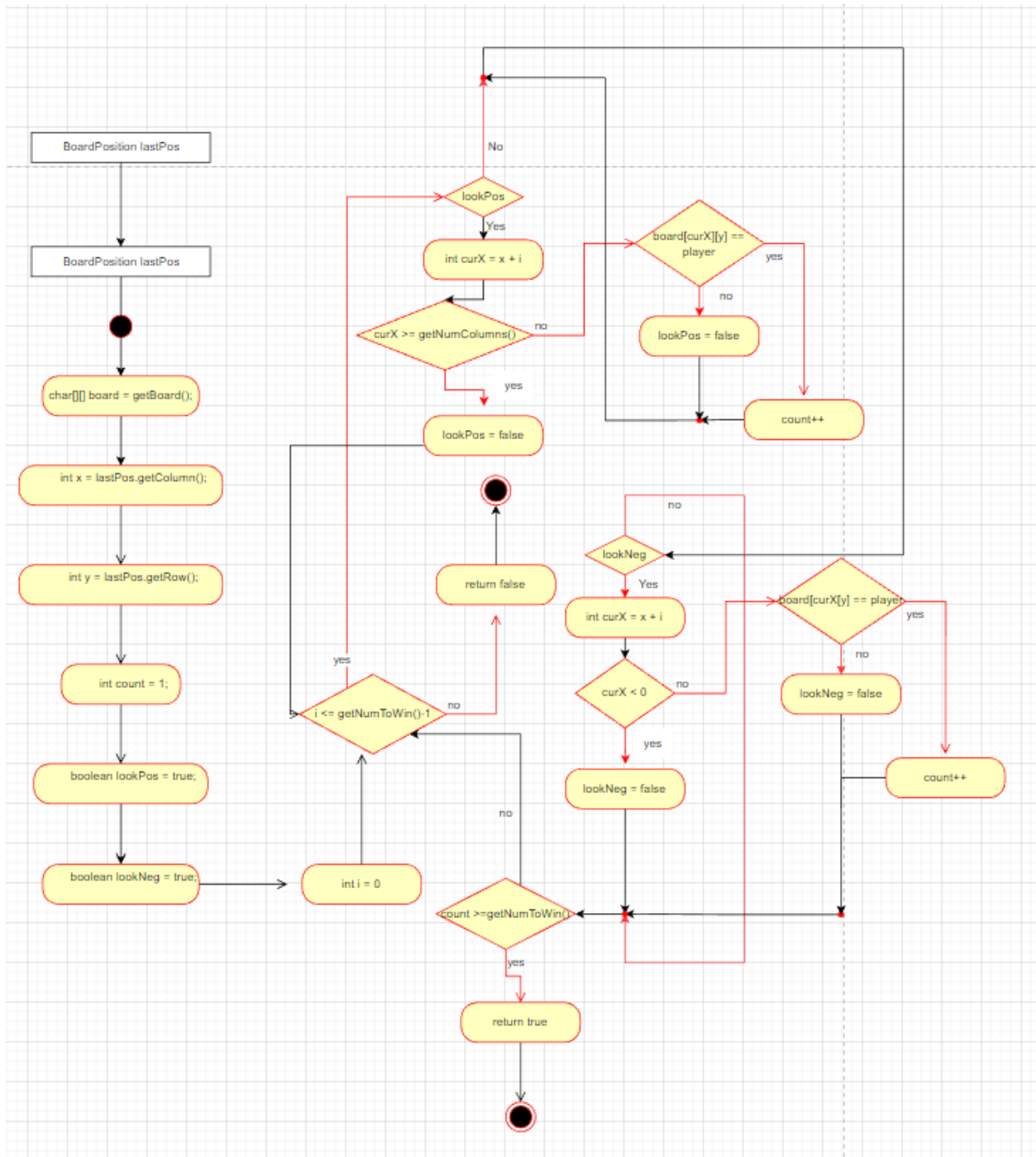
checkForDraw



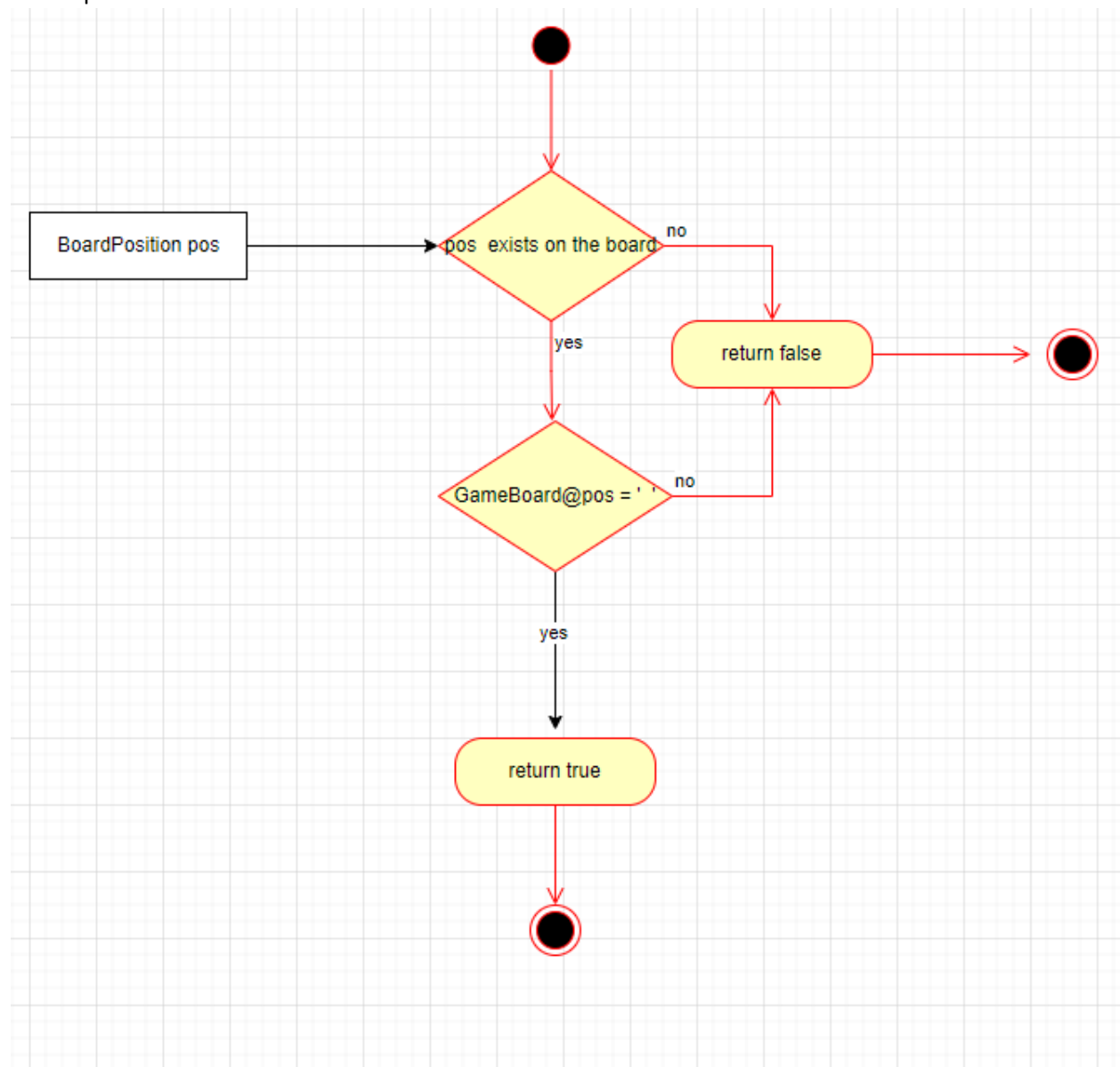
checkForWin



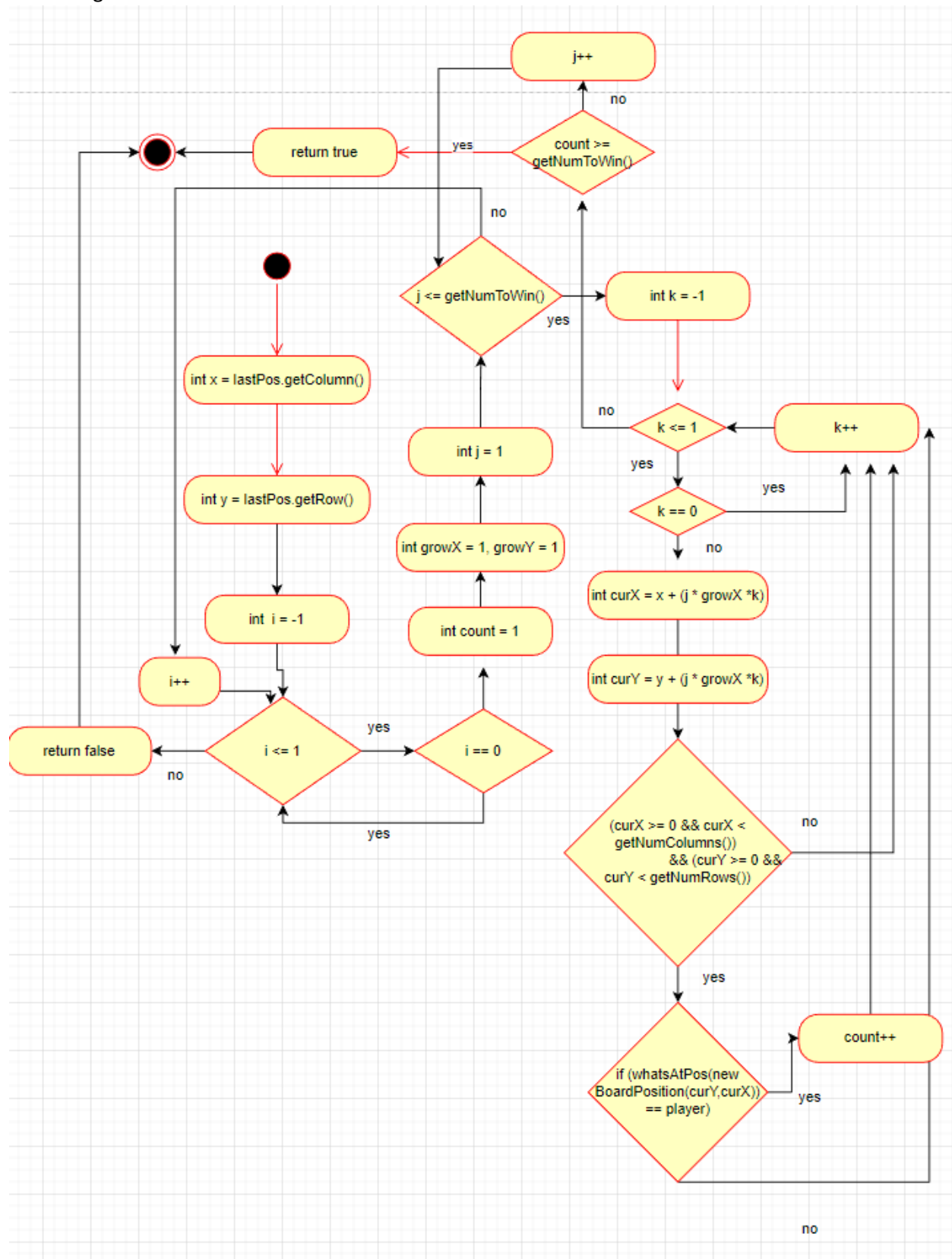
CheckHorizontalWin



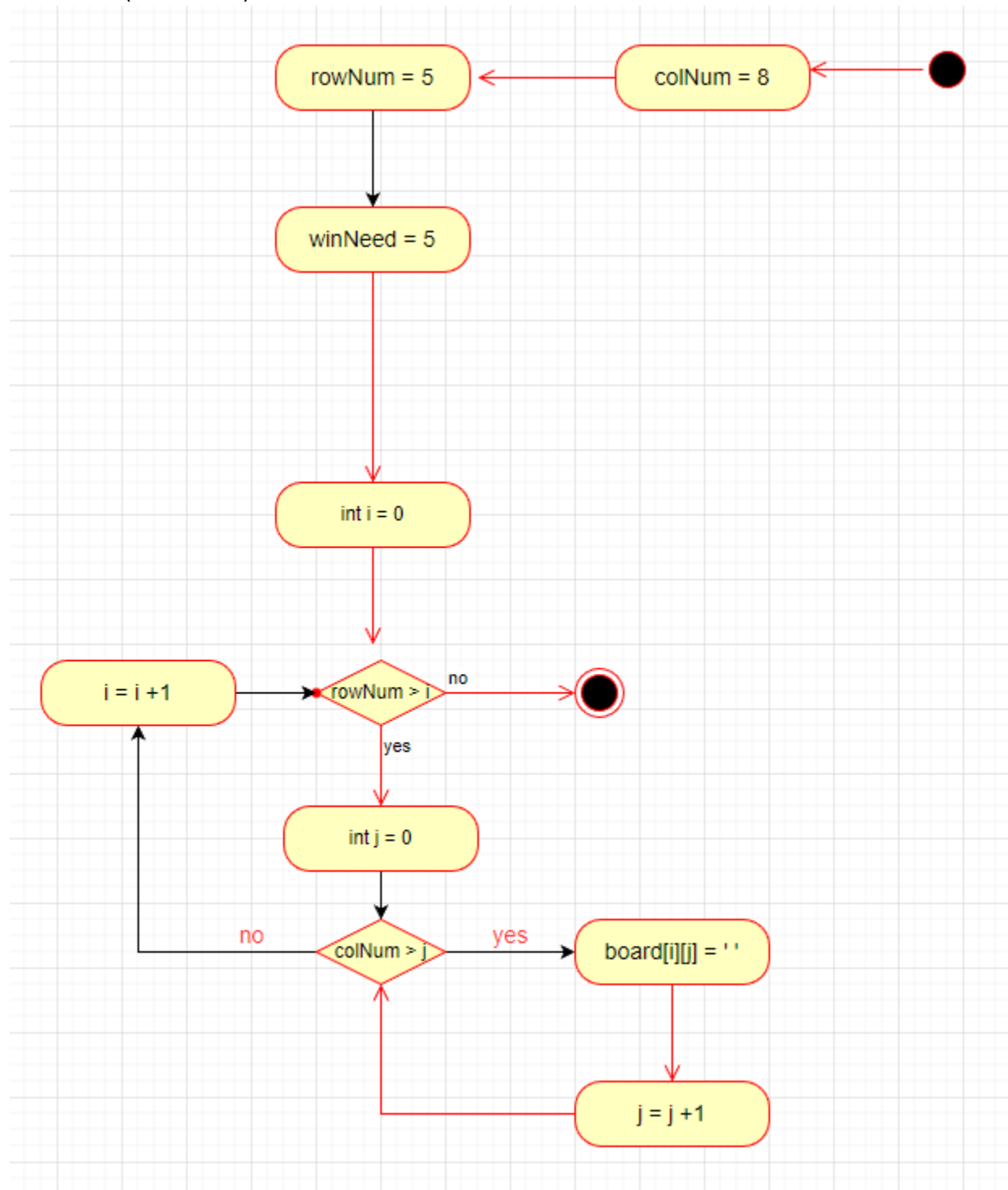
checkSpace

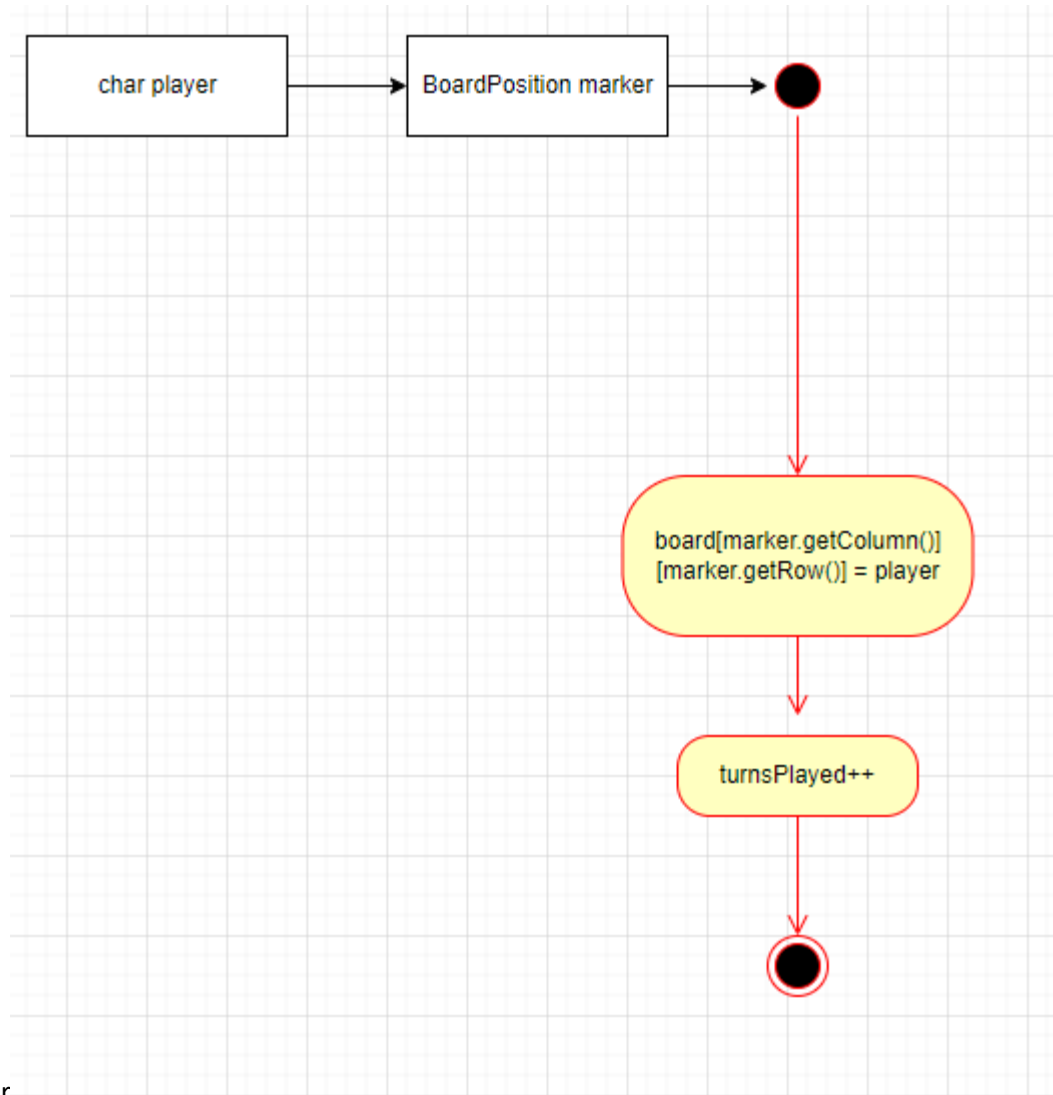


CheckDiagonalWin

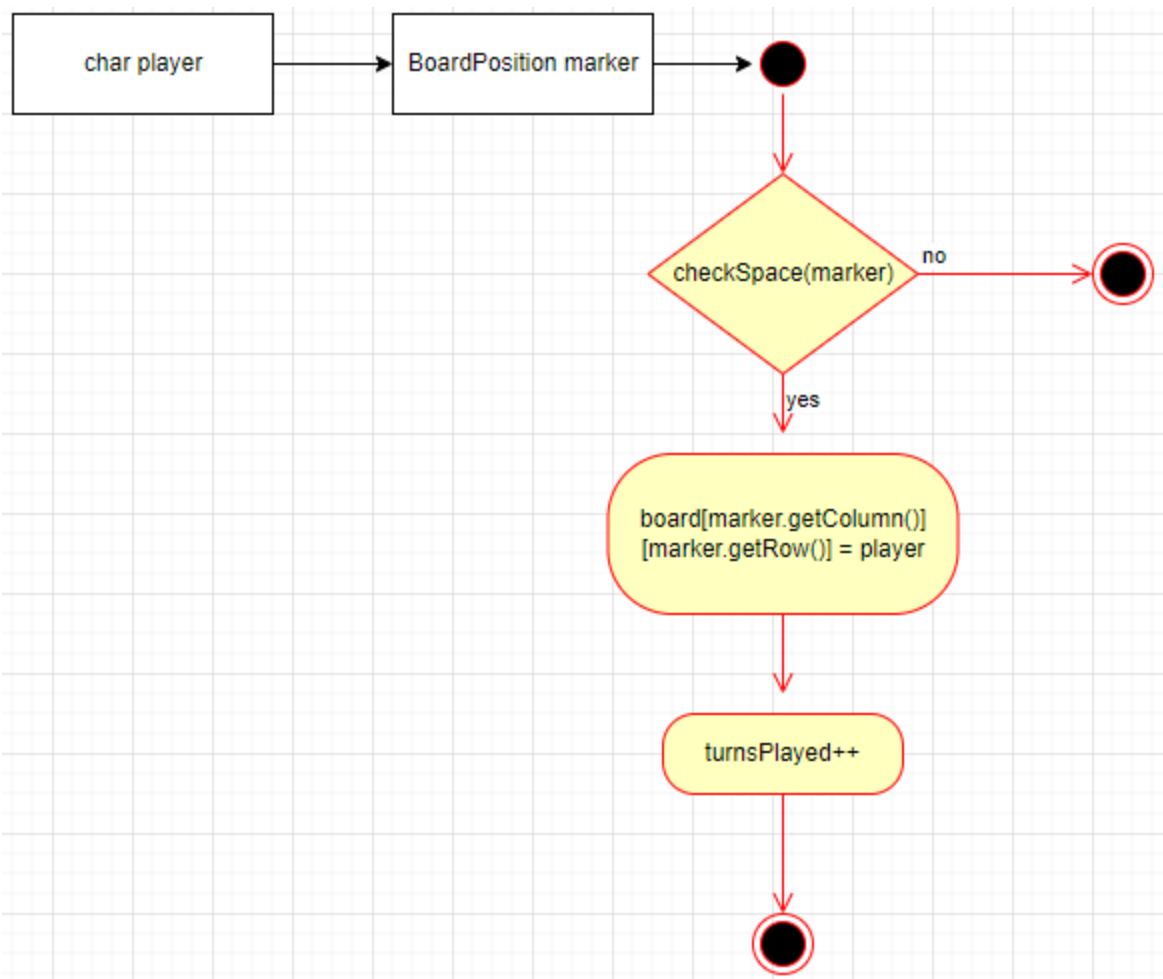


Gameboard(constructor)

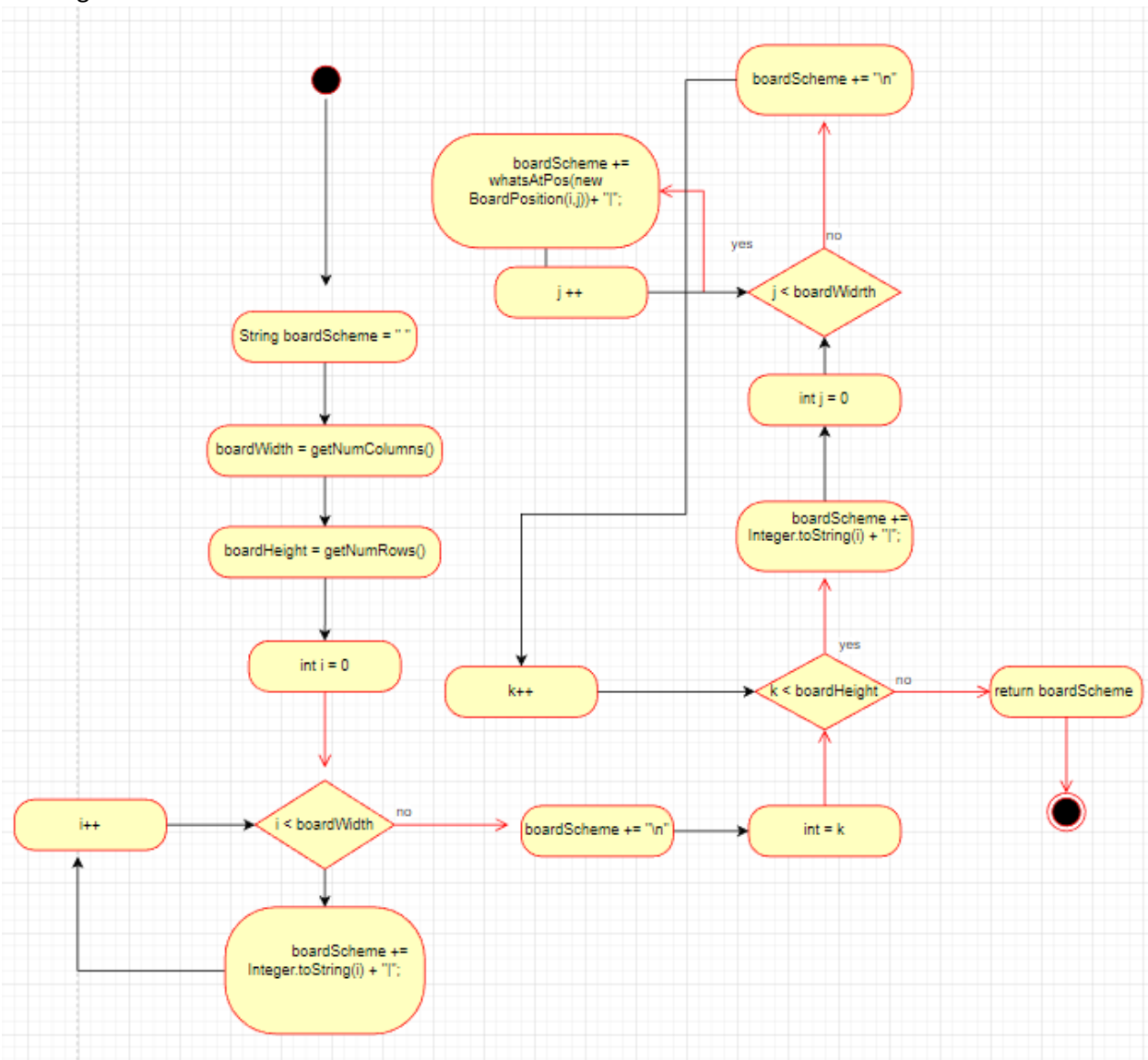




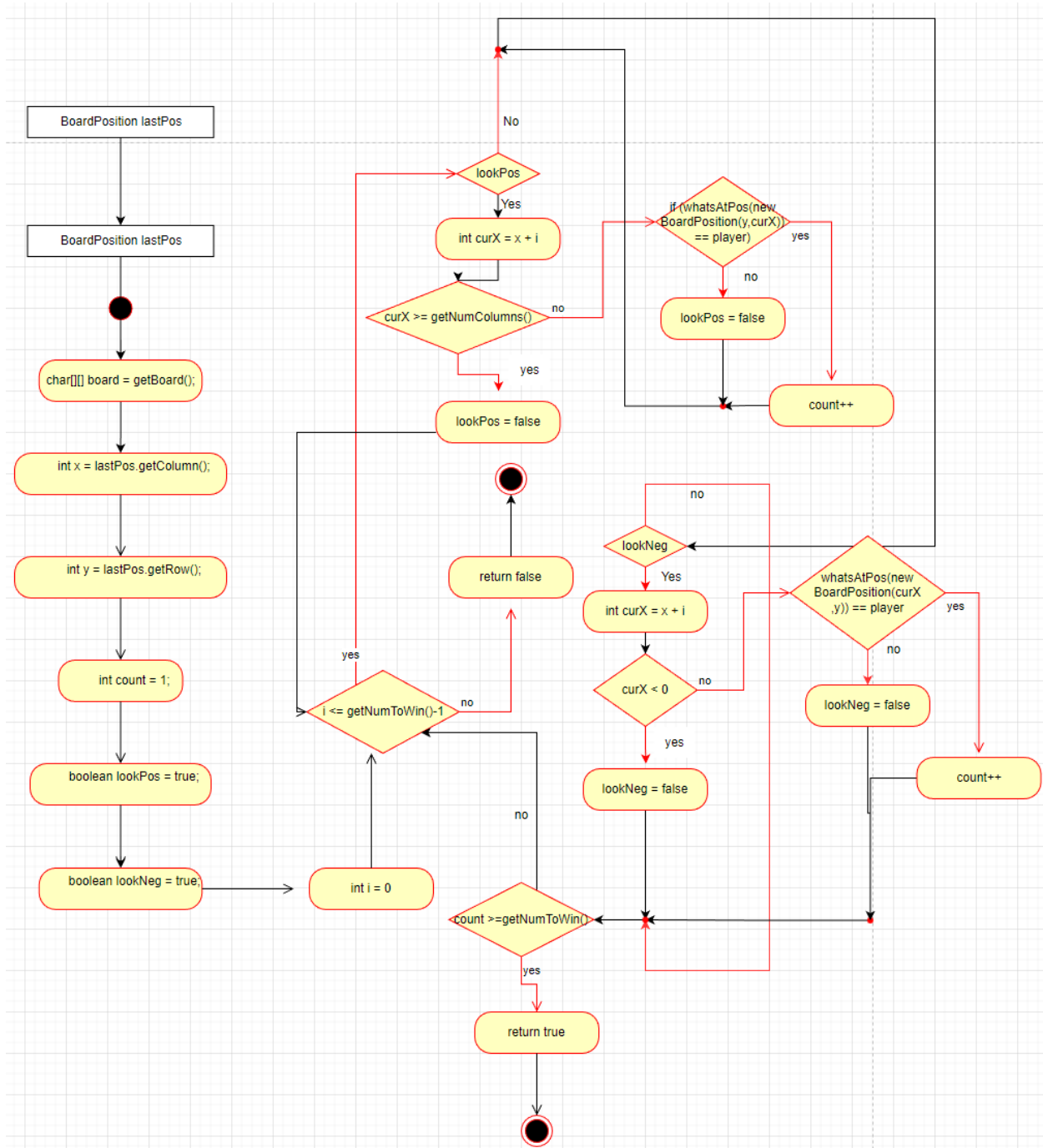
placeMarker

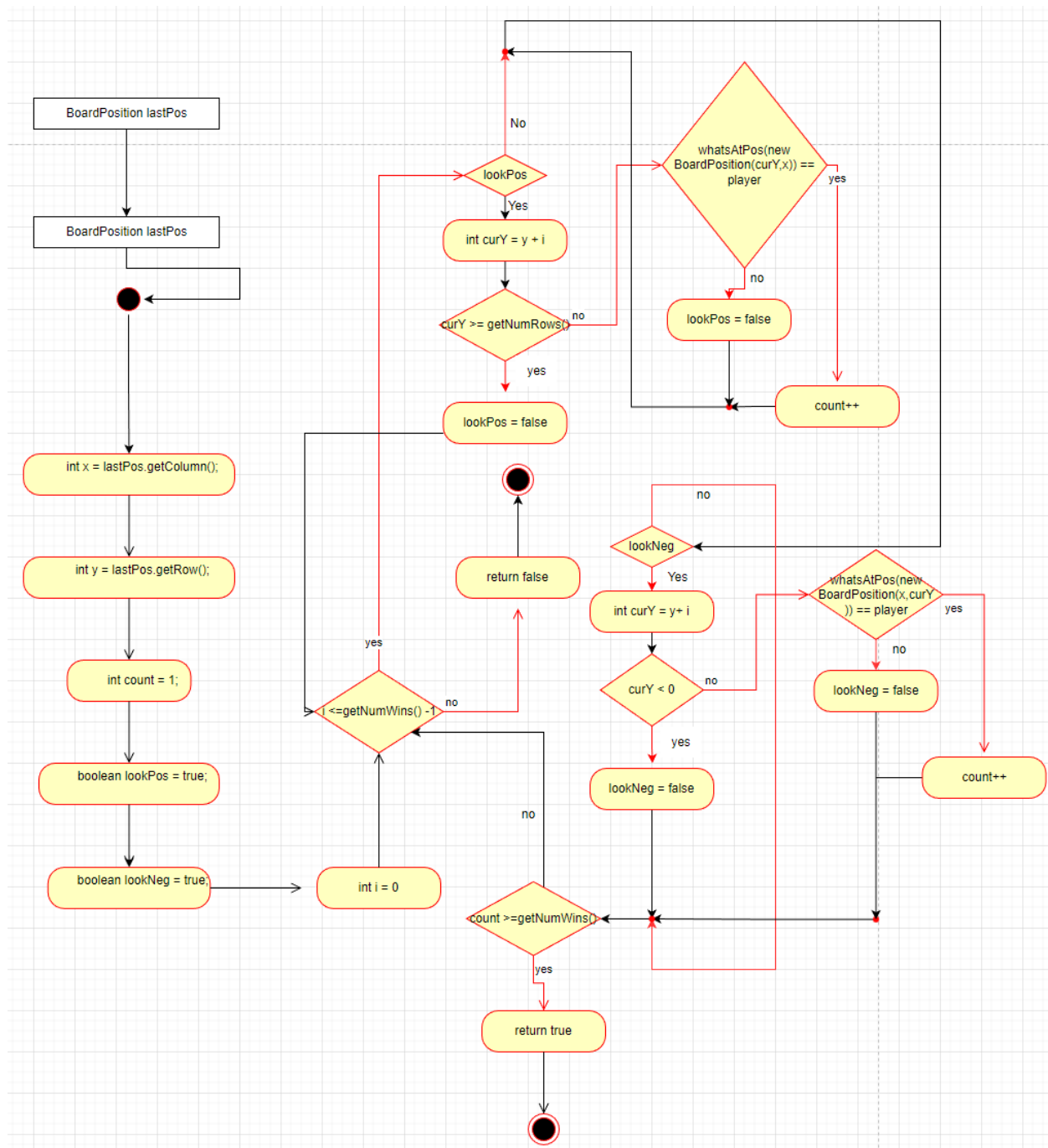


toString

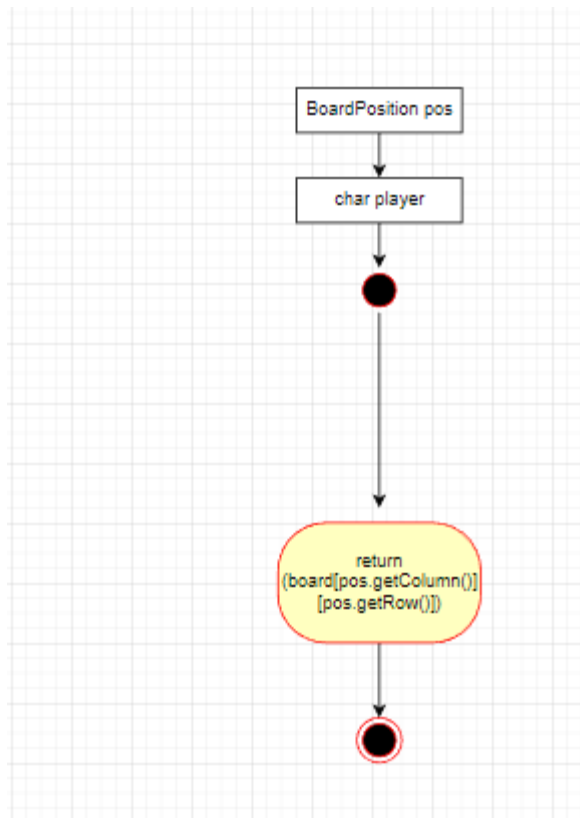


checkHorizontalWin

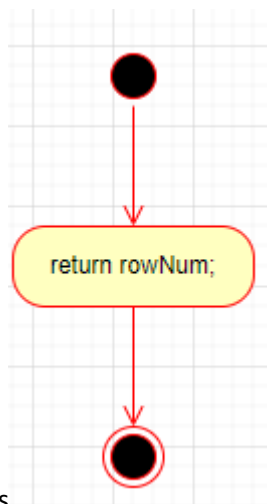




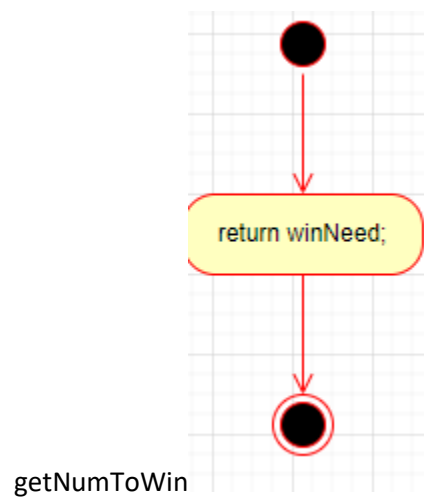
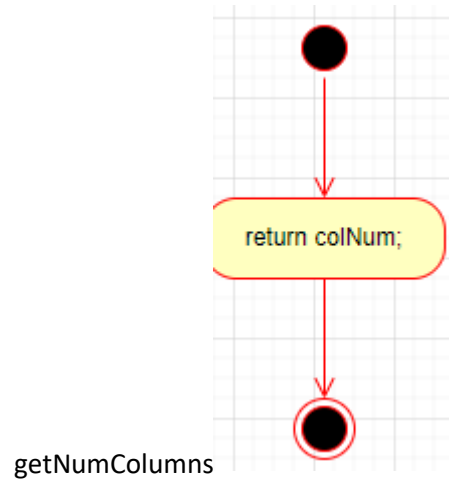
checkVerticalWin



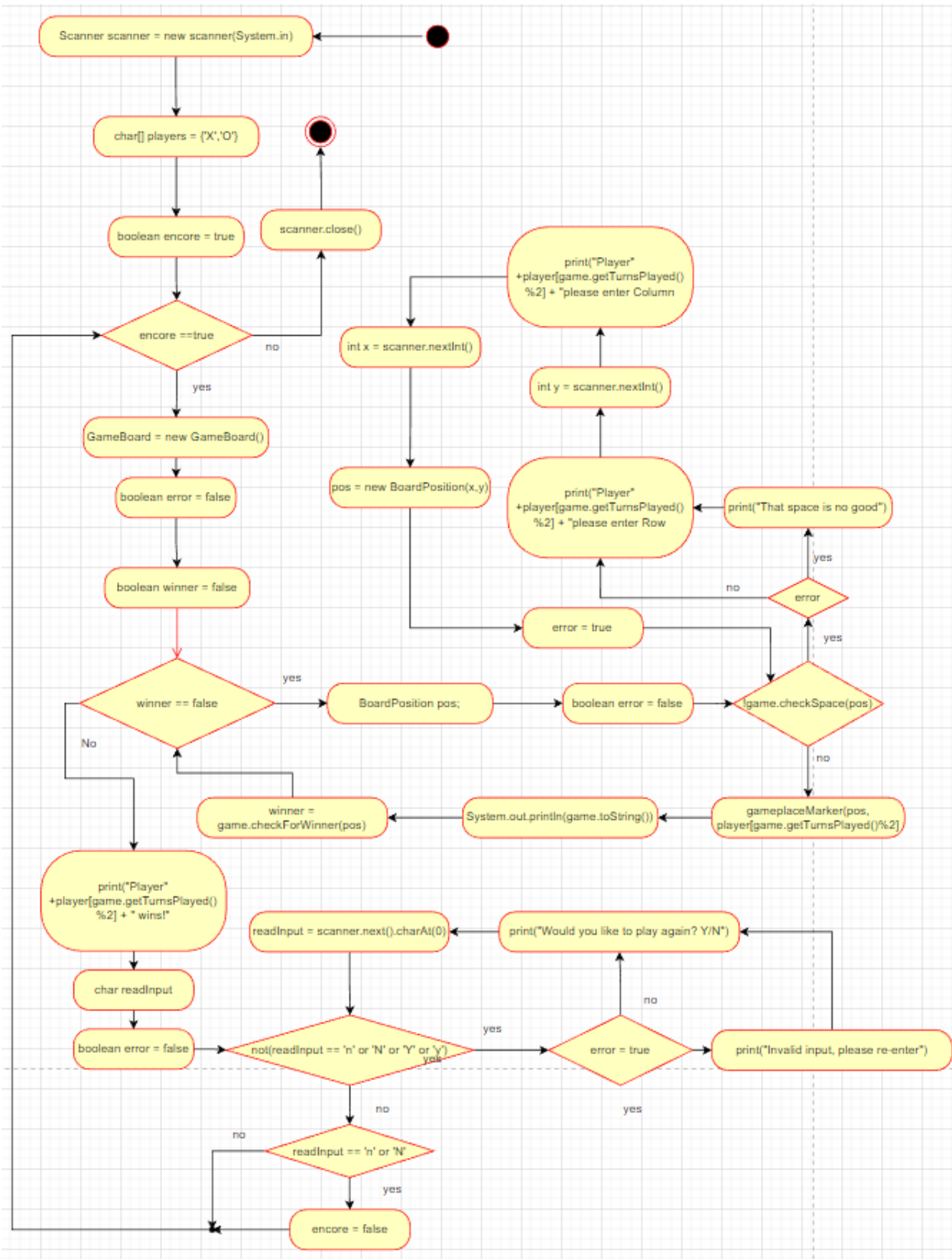
whatsAtPos

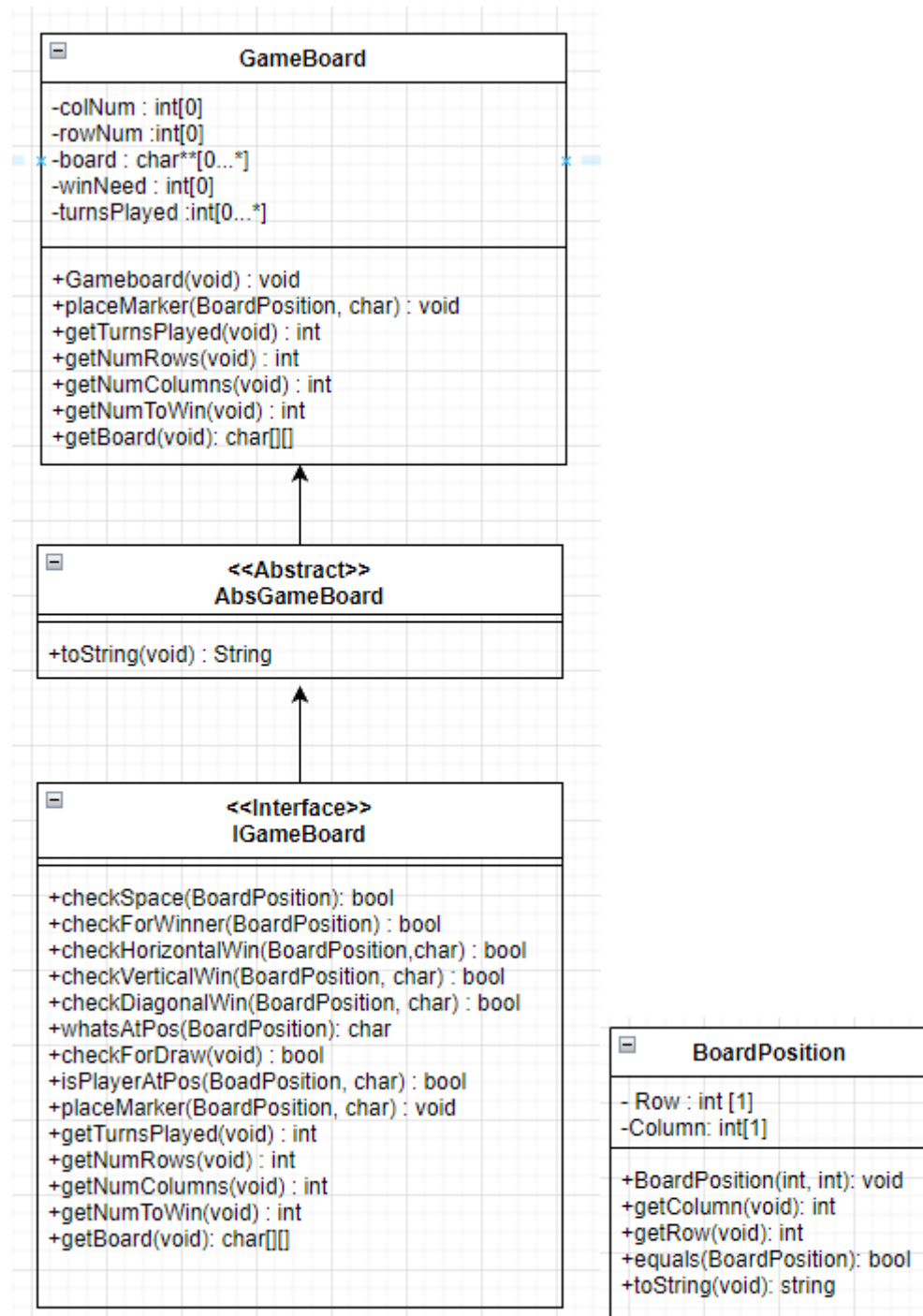


getNumRows



main





UML Class Diagrams

