## Kellen Haas CPSC 2150

Project 2 10/11/20

**Requirements Analysis**

*Functional Requirements*

1. As a user, I can choose which row I want to place my marker in.
2. As a user, I can choose which column I want to place my marker in.
3. As a user, I can view the game board before my turn and after my turn with my updated marker that I just placed.
4. As a user, after the first user takes their turn, I can then choose my row.
5. As a user, after the first user takes their turn, I can choose my column.
6. As a user, I can expect that the system will notify me and my opponent if someone has won horizontally.
7. As a user, I can expect that the system will notify me and my opponent if someone has won vertically.
8. As a user, I can expect that the system will notify me and my opponent if someone has won diagonally.
9. As a user, I want to be notified by the system if there is a draw.
10. As a user, if I choose a position where a marker already has been placed, the system will tell me that I cannot place a marker there.
11. As a user, if I choose a position that is out of the bounds of the board, the system will tell me that it is not a valid position.
12. As a user, I want to be able to view both mine and my opponents placed markers after every turn.
13. As a user, I expect the top of the board to be the index 0, 0.
14. As a user, I want to be asked after the game has ended if I want to play again.
15. As a user, if I choose to play again, then the program should start over from the beginning and clear the game board.

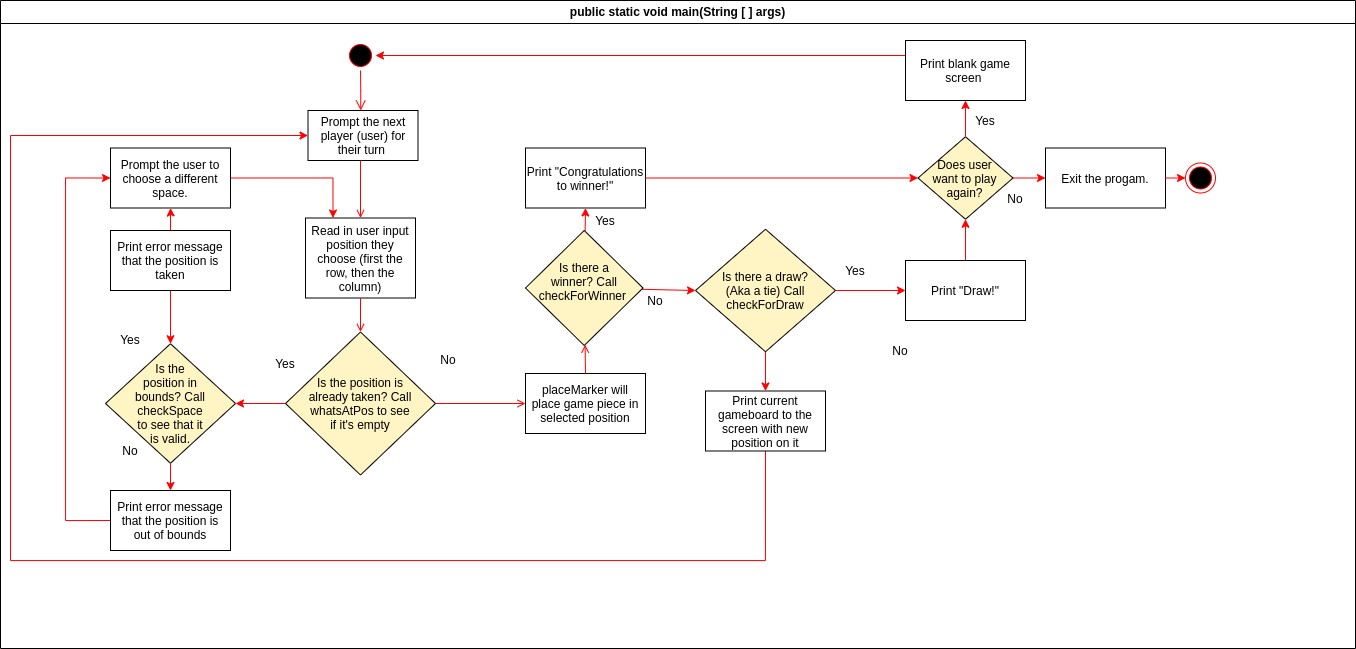
*Non-Functional Requirements*

## The system must be coded in Java programming language.

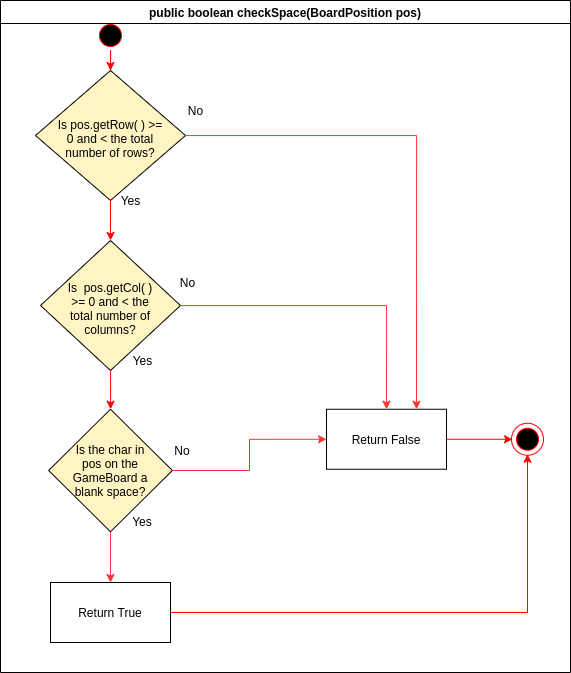
1. The system must be able to run on Unix/Linux.

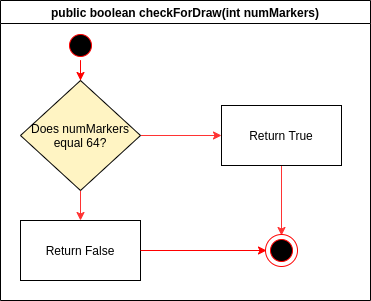
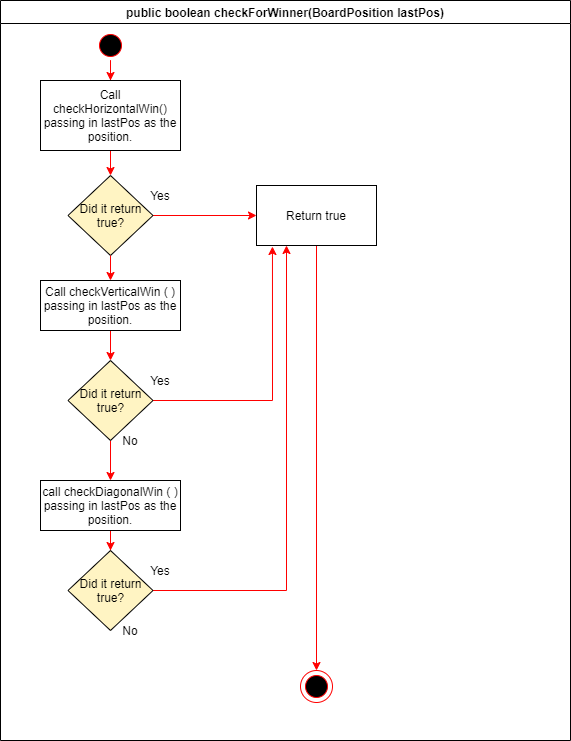
## Program must be able to compile and run quickly and efficiently.

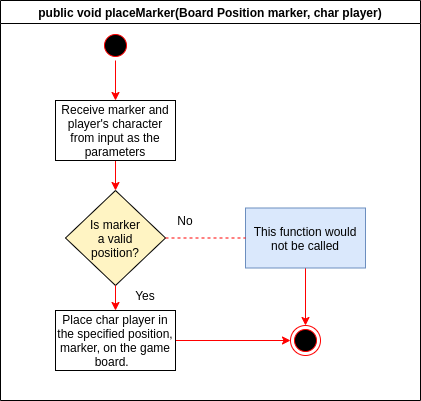
1. The system must be written in IntelliJ IDE for debugging purposes in the future.
2. The top left corner of the game board must be 0, 0.
3. The game board is currently 8x8.
4. Currently, player X goes first and then player O.

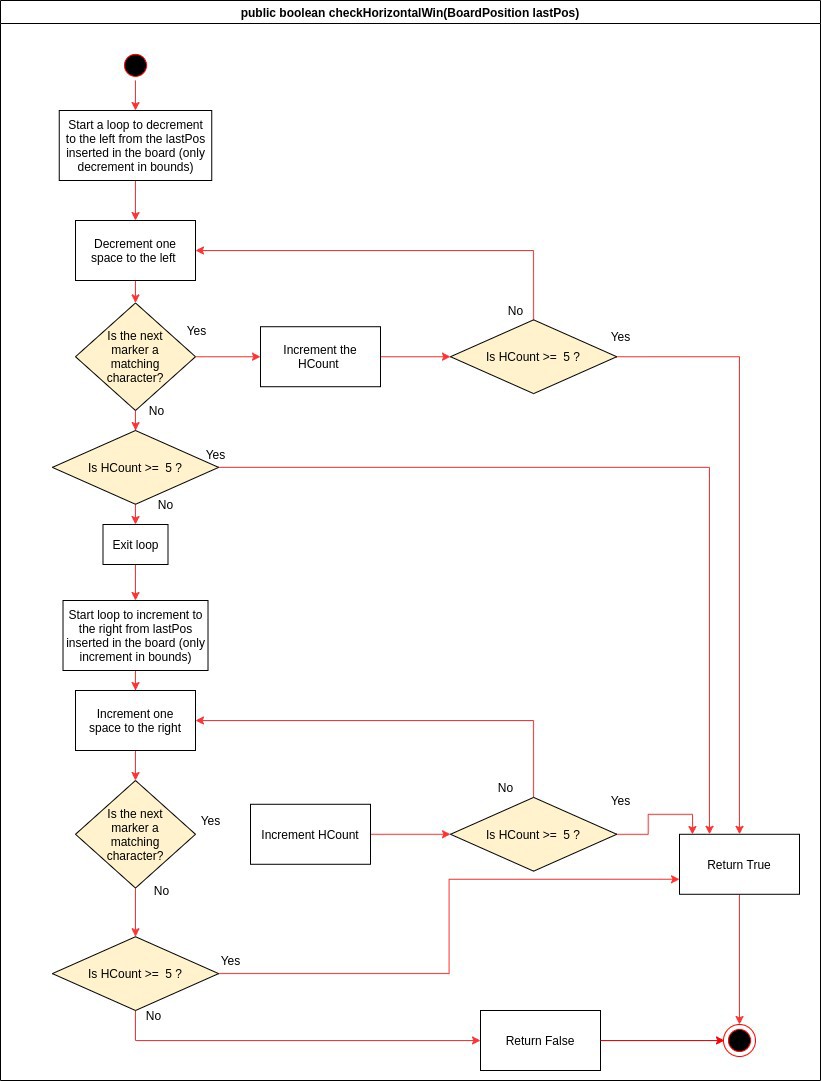
**Design**  **Activity Diagrams***GameScreen.java*

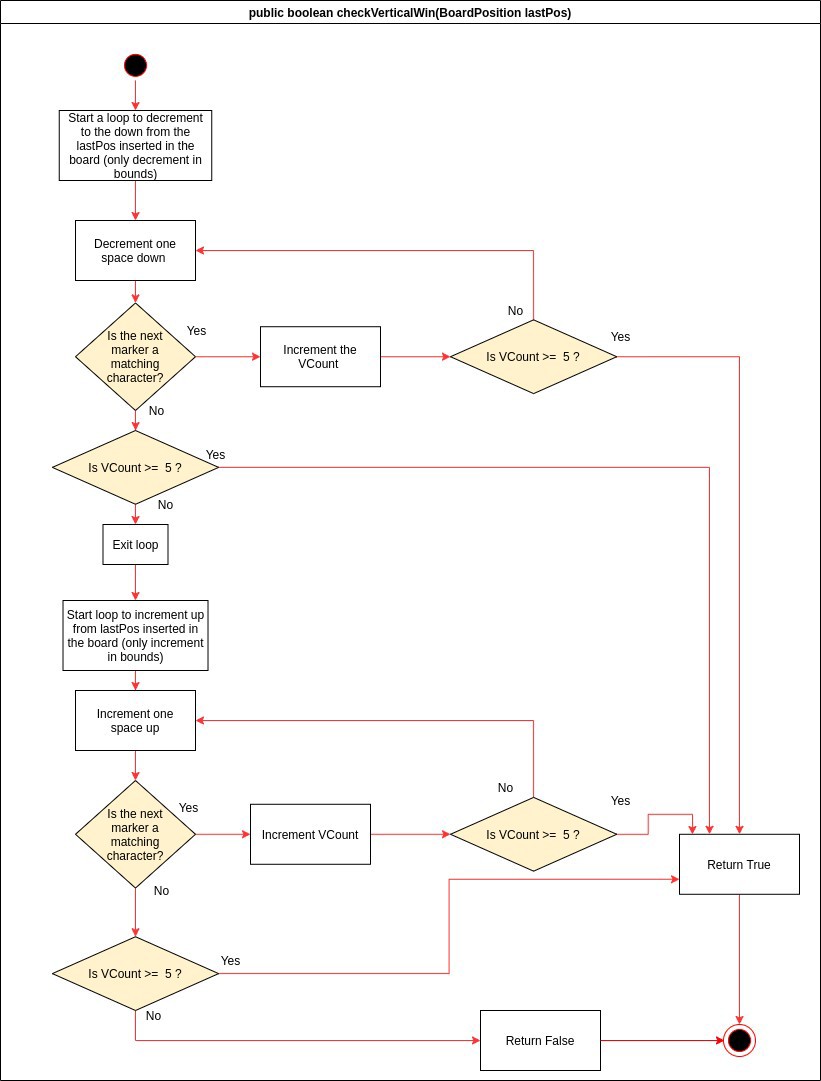
# GameBoard.java











•

Start a loop to decrement down an d to the le ft from *the* lastPosinserted in the board {only decrement in bounds)

**publi c boolean checkDiagonalWin(BoardPosition lastPos)**

Incremen t the DCount

Yes

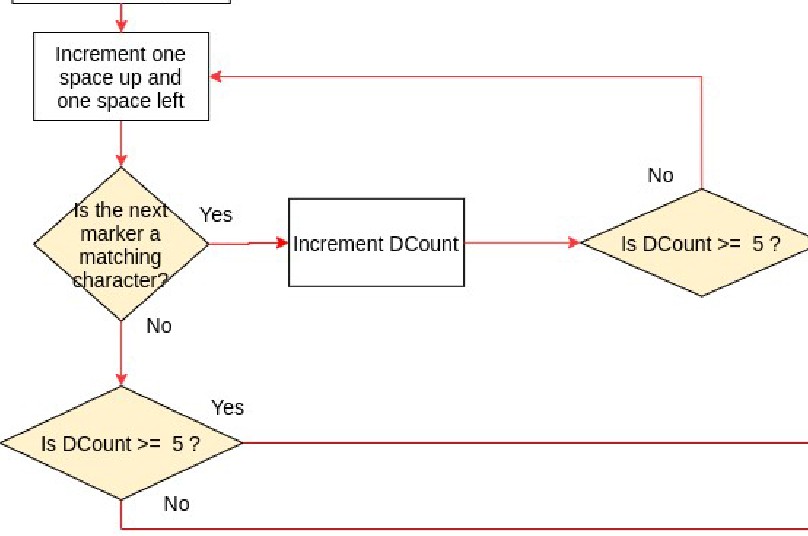
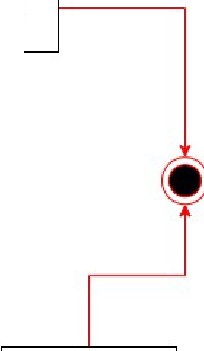
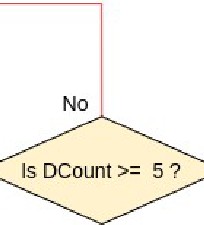
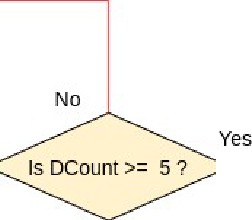
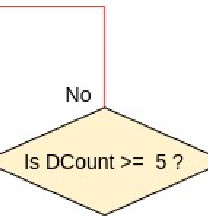
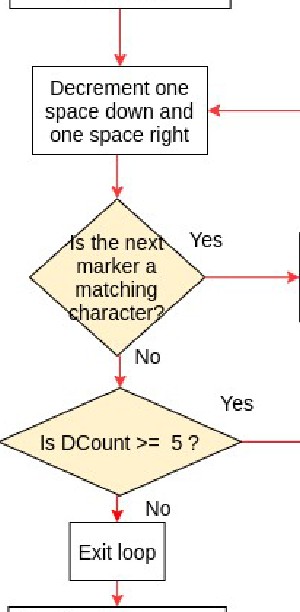
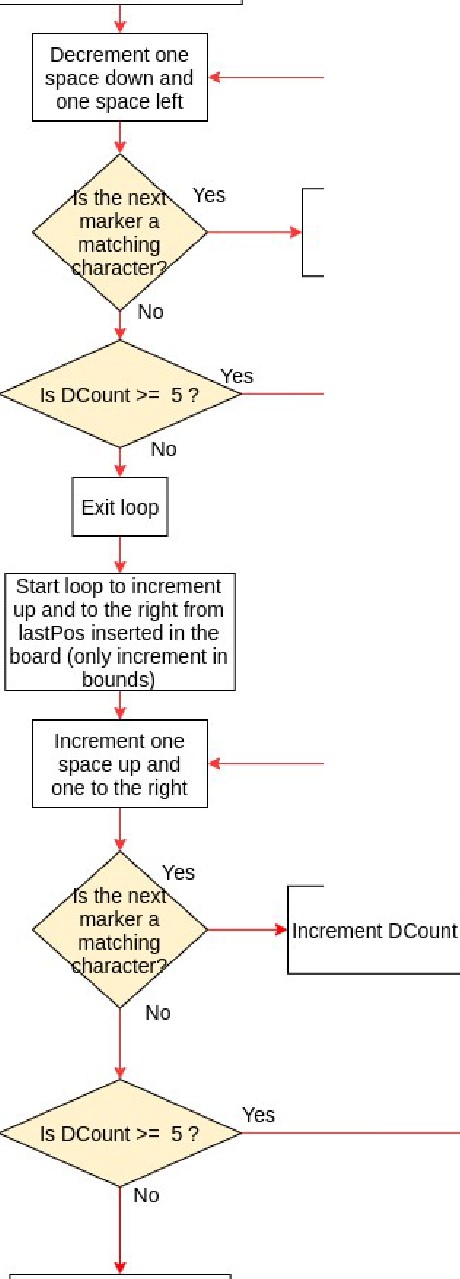
Start a loop ID decrement down and to the rig ht from the las1Pos inserted in the board (only decrement in bounds)

Return True

Increment the

DCoum

Yes

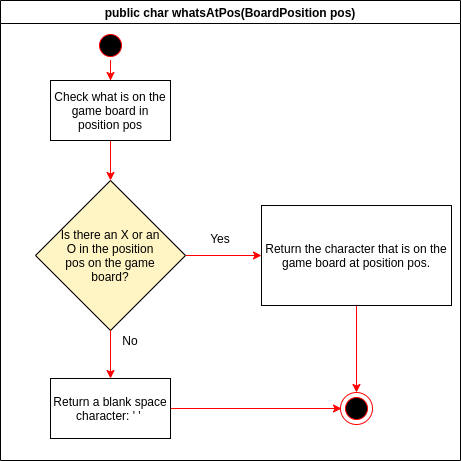


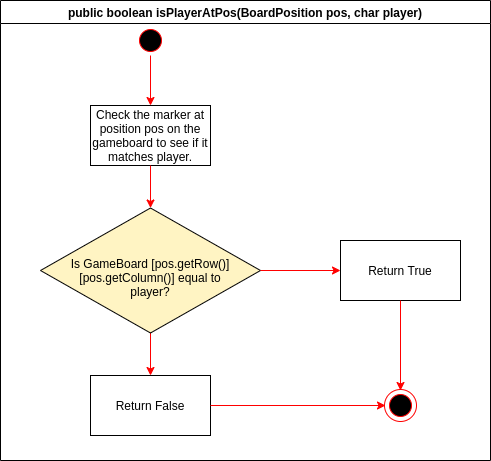
Return False

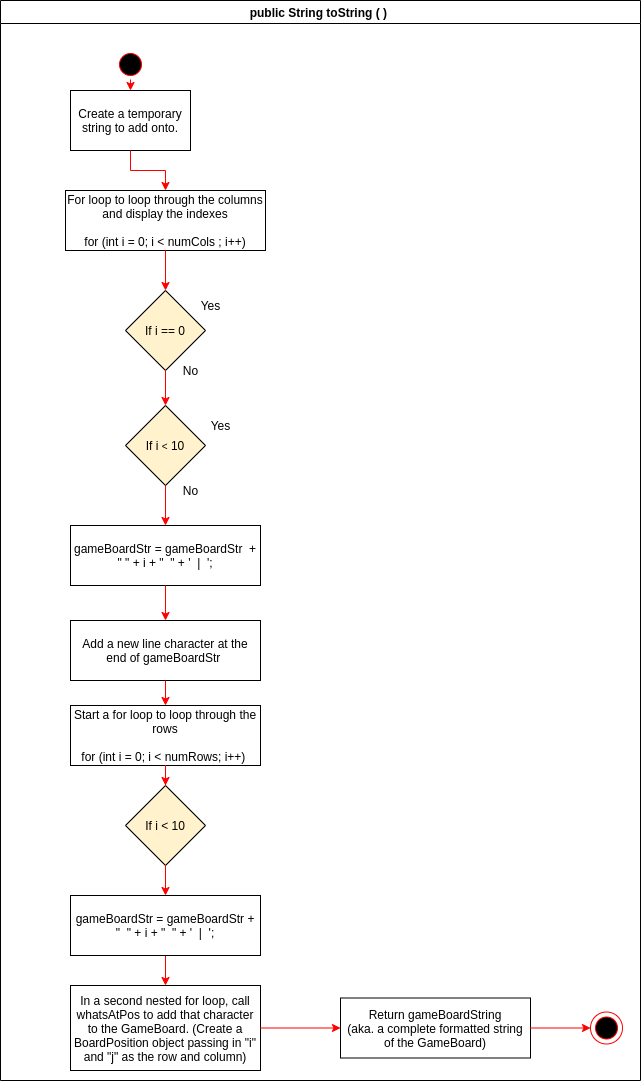
Start loop to incremerit up and to the left from lastPos inserted in the board (only increment in bounds)

Yes

## Kellen Haas 8

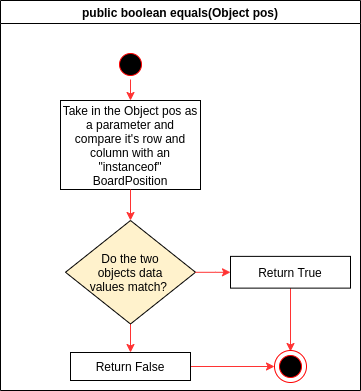
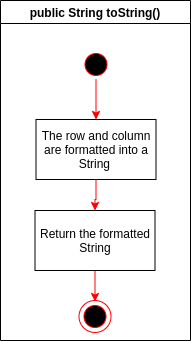


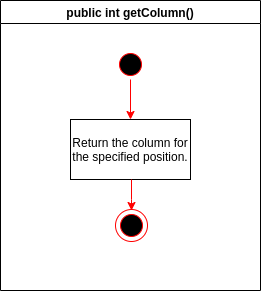
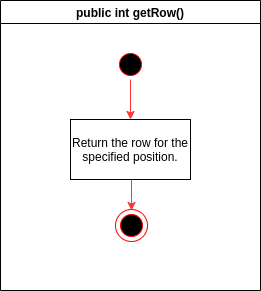




Kellen Haas 9

# BoardPosition.java





**UML Class Diagrams**

