Kellen Haas
CPSC 2150
Project 3
10/28/20

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

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

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

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

4. The system must be written in IntelliJ IDE for debugging purposes in the future.

5. The top left corner of the game board must be 0, 0.

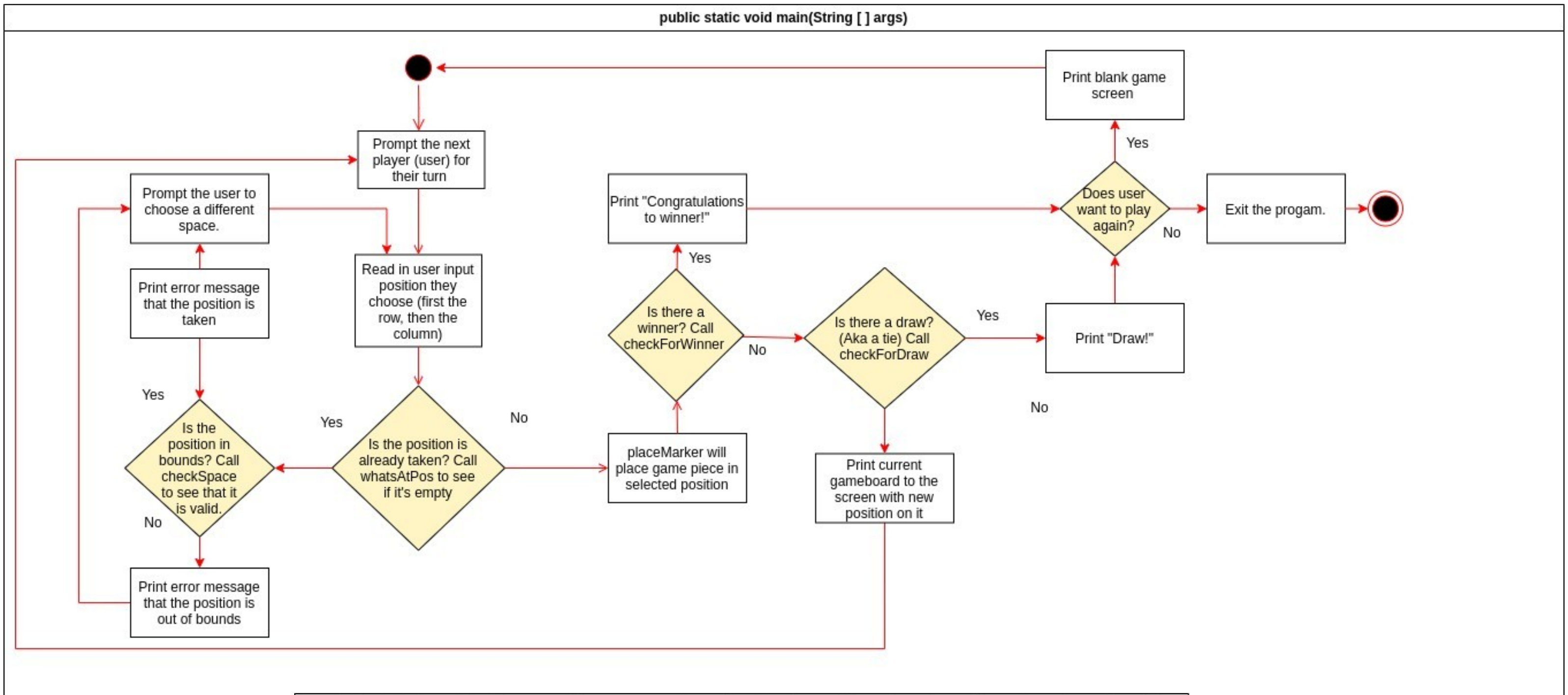
6. The game board is currently 8x8.

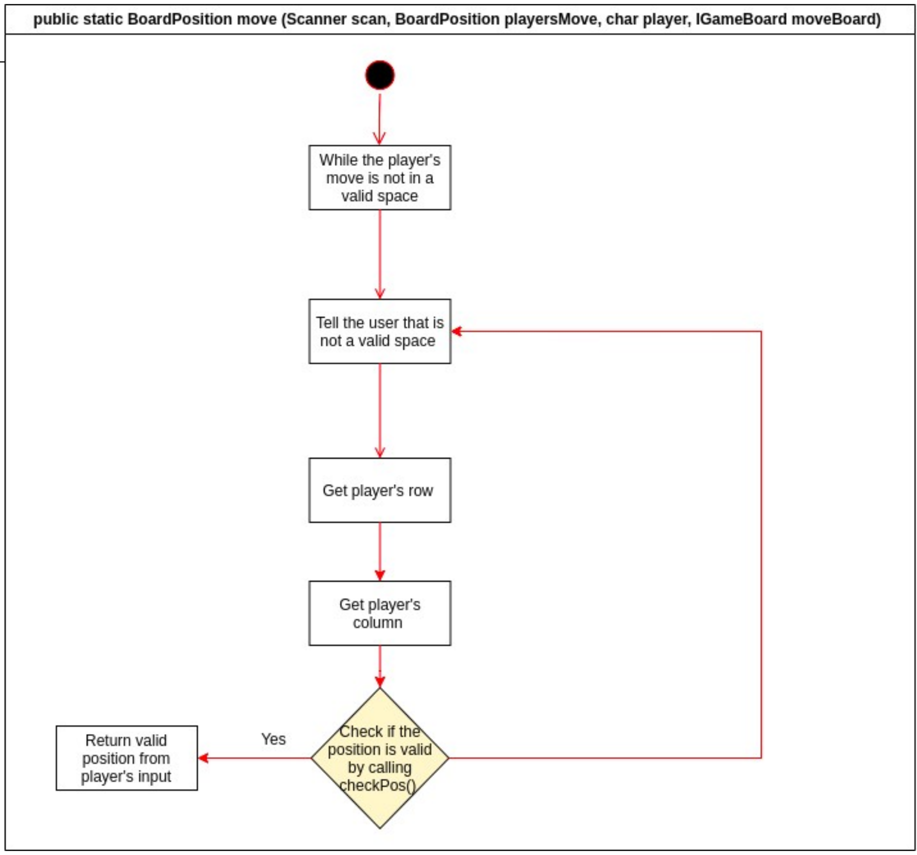
7. Currently, player X goes first and then player O.

Design

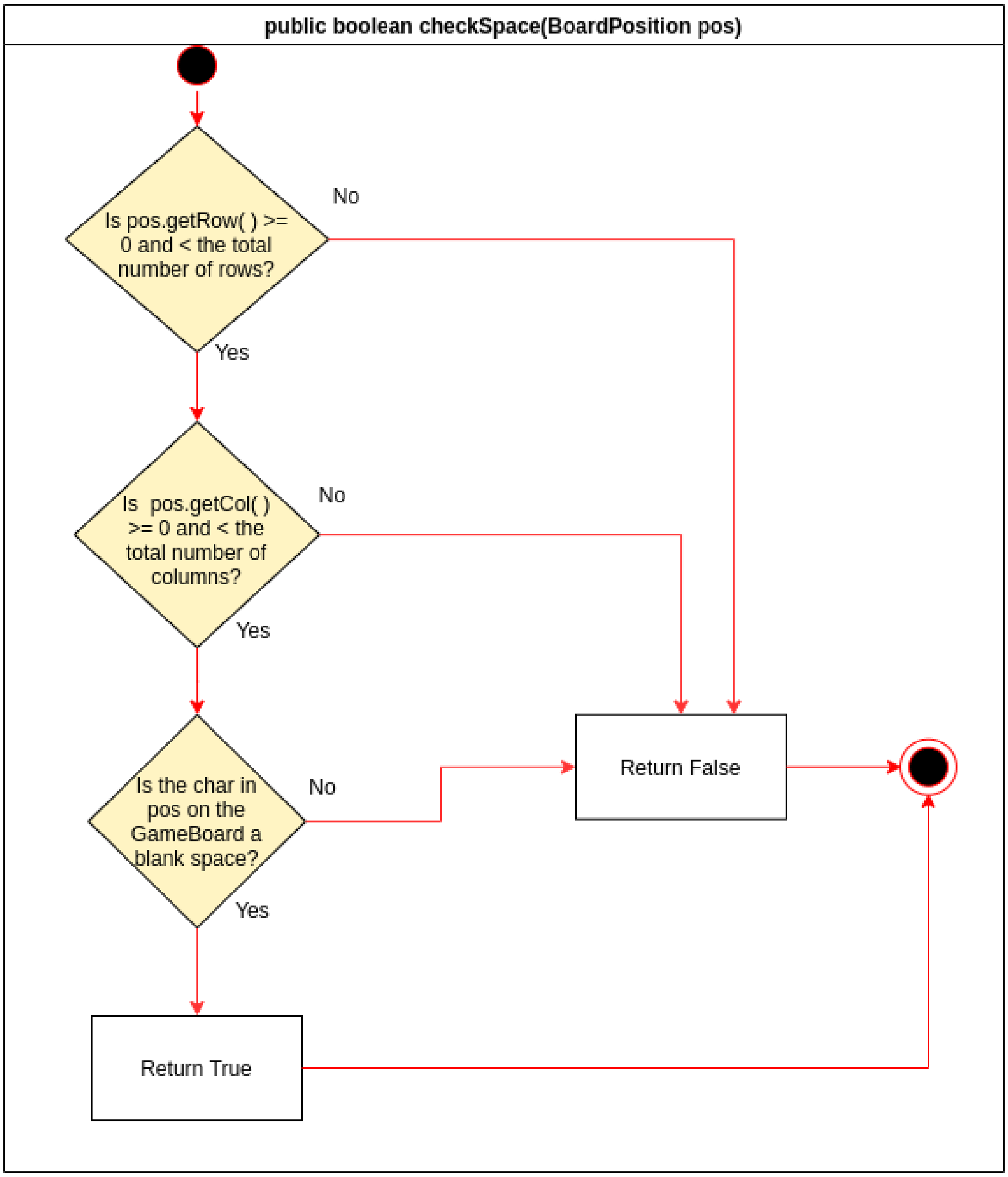
Activity Diagrams

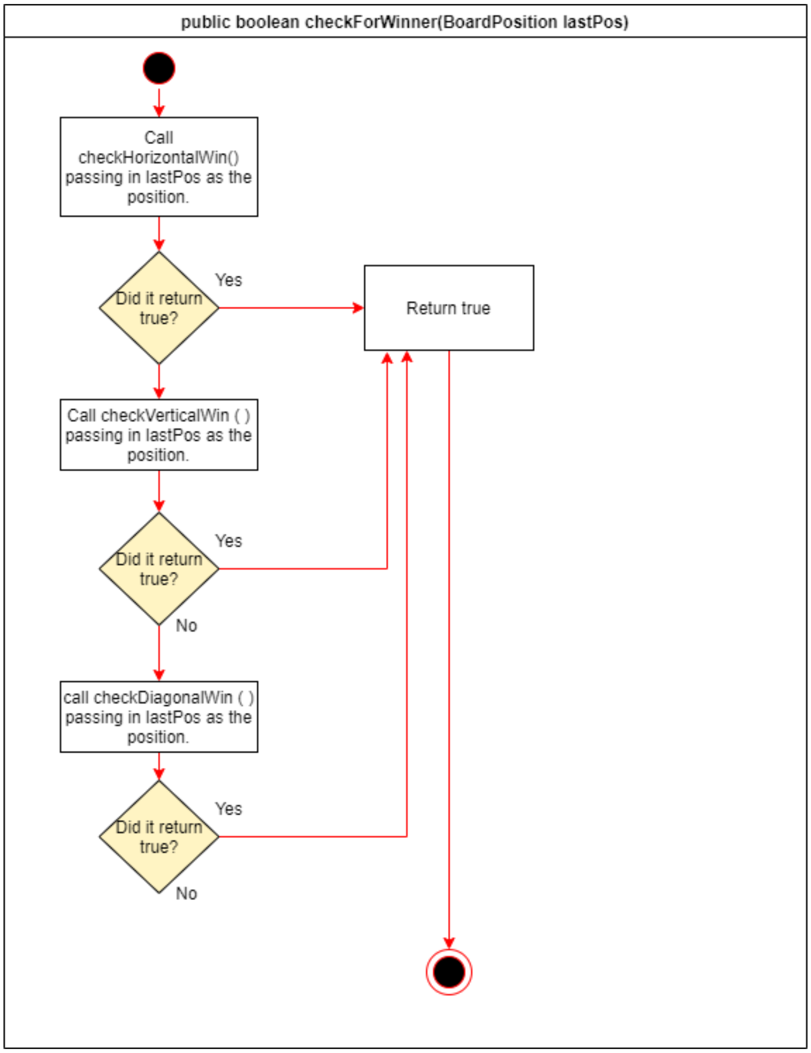
GameScreen.java

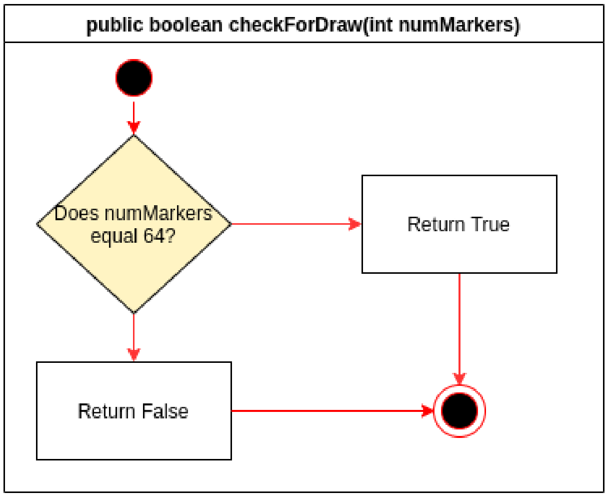

public static void main(String [ ] args)
Prompt the next player (user) for their turn
Read in user input position they choose (first the row, then the column)
Is the position is already taken? Call whatsAtPos to see if it's empty
Yes
No
Is the position in bounds? Call checkSpace to see that it is valid.
Yes
No
Print error message that the position is taken
Prompt the user to choose a different space.
Print error message that the position is out of bounds
placeMarker will place game piece in selected position
Is there a winner? Call checkForWinner
Yes
No
Print "Congratulations to winner!"
Is there a draw? (Aka a tie) Call checkForDraw
Yes
No
Print "Draw!"
Print current gameboard to the screen with new position on it
Does user want to play again?
Yes
No
Print blank game screen
Exit the progam.

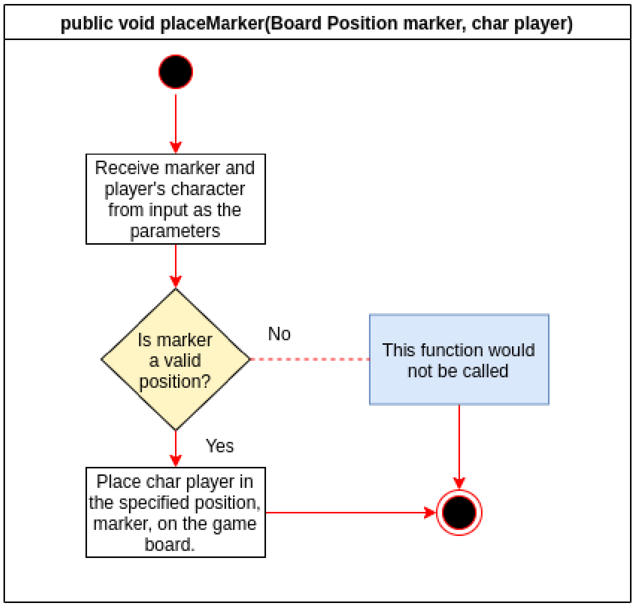

public static BoardPosition move (Scanner scan, BoardPosition playersMove, char player, IGameBoard moveBoard)
While the player's move is not in a valid space
Tell the user that is not a valid space
Get player's row
Get player's column
Check if the position is valid by calling checkPos()
Yes
Return valid position from player's input

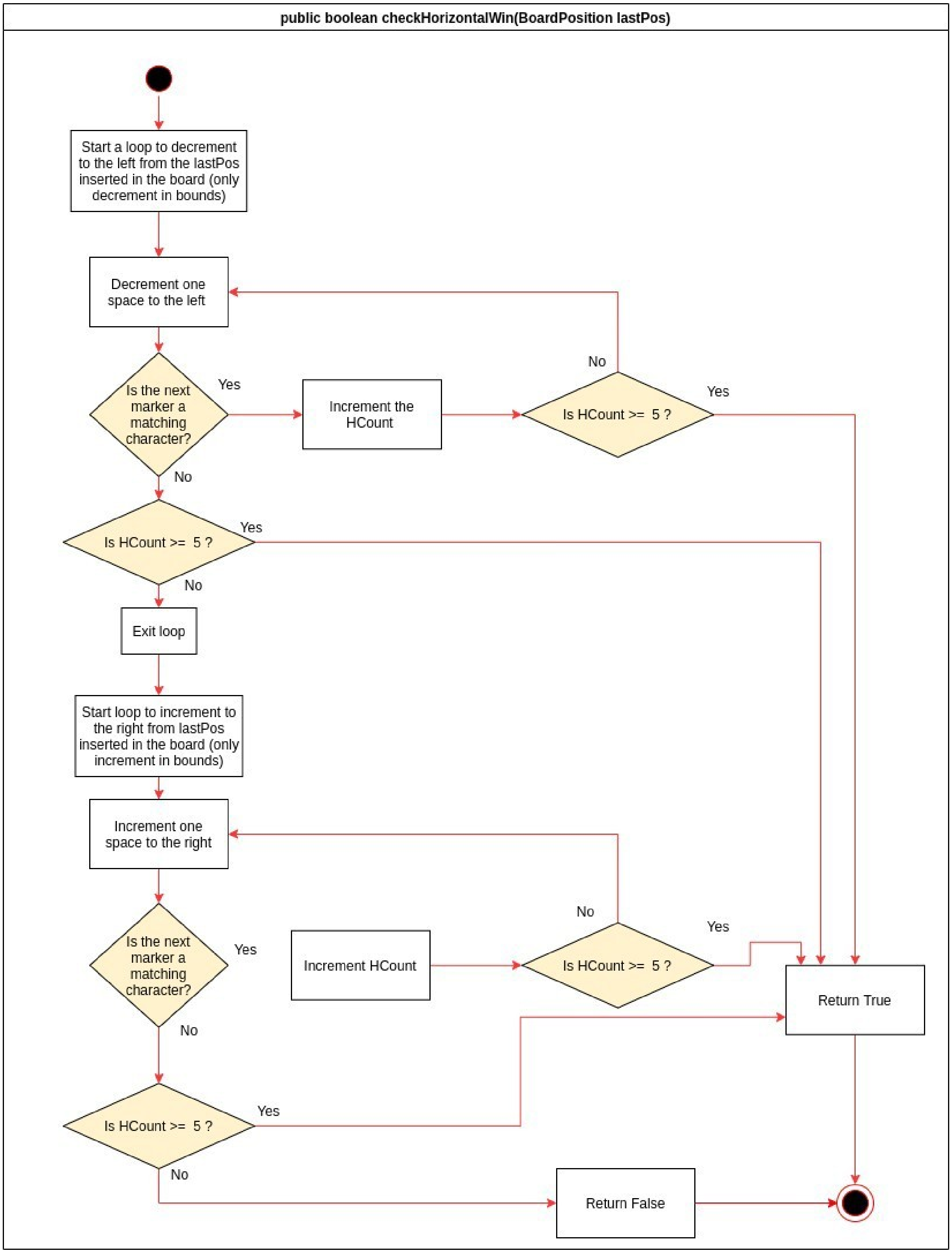
GameBoard.java

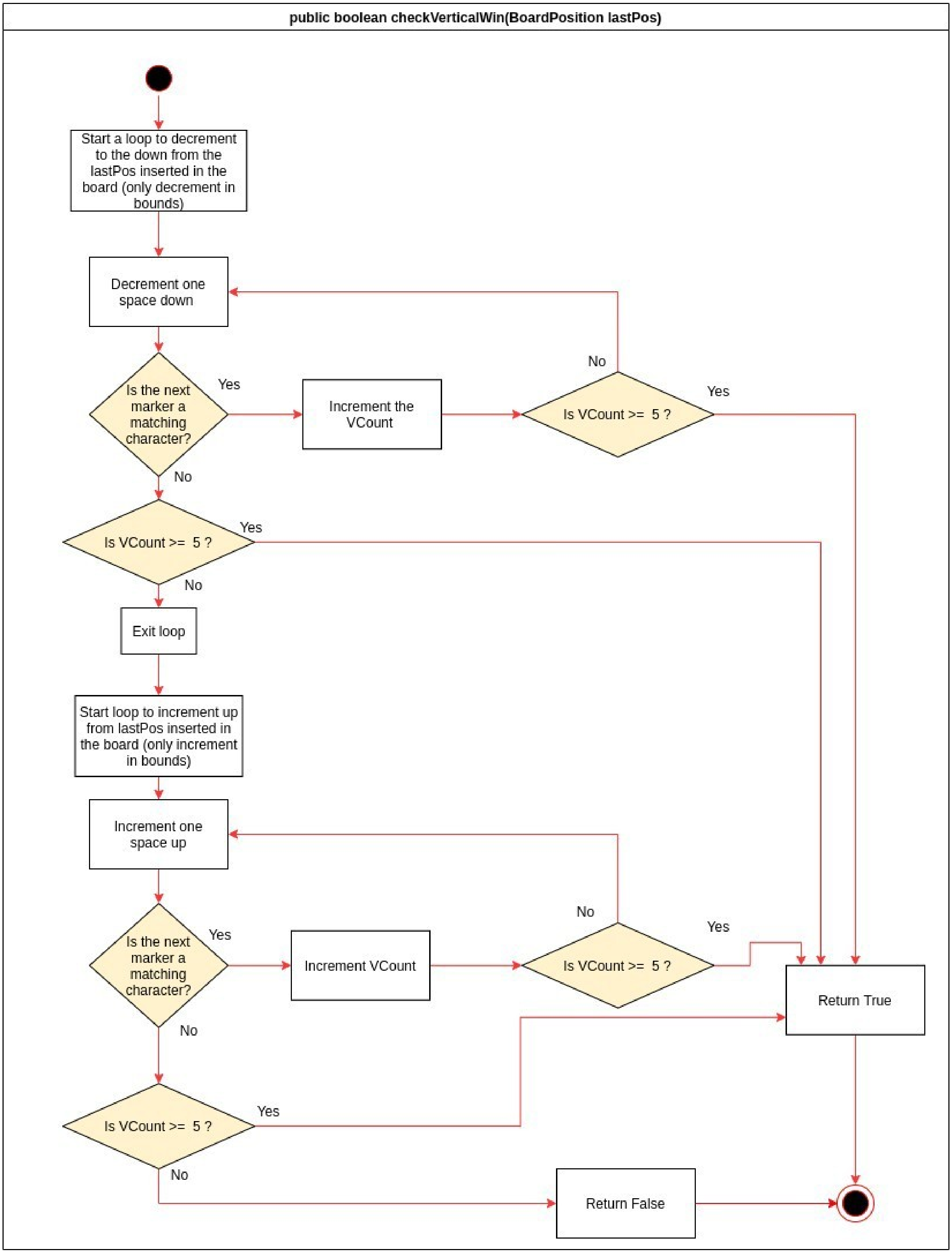

public boolean checkSpace(BoardPosition pos)
Is pos.getRow( ) >= 0 and < the total number of rows?
No
Yes
Is pos.getCol( ) >= 0 and < the total number of columns?
No
Yes
Is the char in pos on the GameBoard a blank space?
No
Yes
Return False
Return True


public boolean checkForWinner(BoardPosition lastPos)
Call checkHorizontalWin() passing in lastPos as the position.
Did it return true?
Yes
Return true
Call checkVerticalWin ( ) passing in lastPos as the position.
Did it return true?
Yes
No
call checkDiagonalWin ( ) passing in lastPos as the position.
Did it return true?
Yes
No

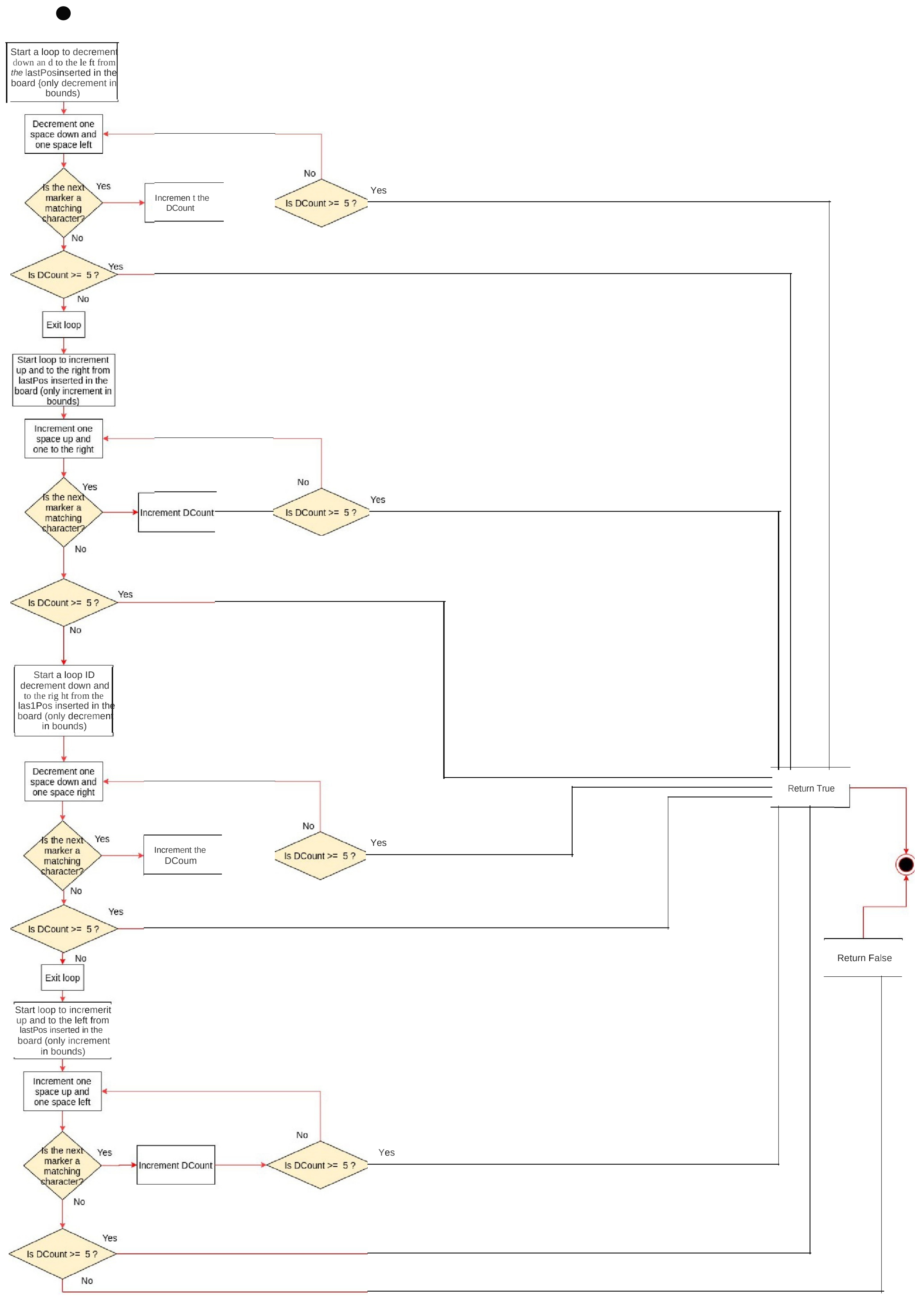

public boolean checkForDraw(int numMarkers)
Does numMarkers equal 64?
Return True
Return False

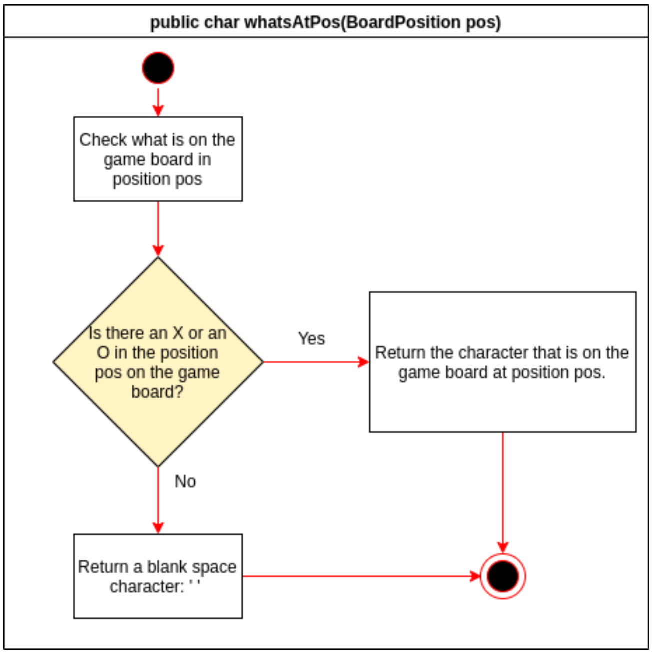

public void placeMarker(Board Position marker, char player)
Receive marker and player's character from input as the parameters
Is marker a valid position?
No
This function would not be called
Yes
Place char player in the specified position, marker, on the game board.

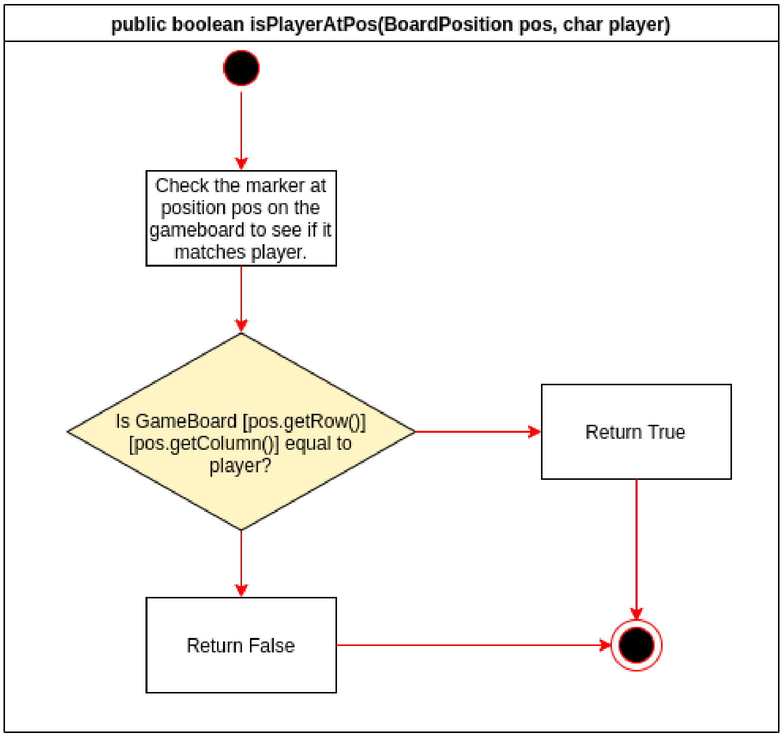

public boolean checkHorizontalWin(BoardPosition lastPos)
Start a loop to decrement to the left from the lastPos inserted in the board (only decrement in bounds)
Decrement one space to the left
Is the next marker a matching character?
Yes
Increment the HCount
Is HCount >= 5 ?
No
Yes
No
Is HCount >= 5 ?
Yes
No
Exit loop
Start loop to increment to the right from lastPos inserted in the board (only increment in bounds)
Increment one space to the right
Is the next marker a matching character?
Yes
Increment HCount
Is HCount >= 5 ?
No
Yes
Return True
No
Is HCount >= 5 ?
Yes
No
Return False


public boolean checkVerticalWin(BoardPosition lastPos)
Start a loop to decrement to the down from the lastPos inserted in the board (only decrement in bounds)
Decrement one space down
Is the next marker a matching character?
Yes
Increment the VCount
No
Is VCount >= 5 ?
Yes
No
Is VCount >= 5 ?
Yes
No
Exit loop
Start loop to increment up from lastPos inserted in the board (only increment in bounds)
Increment one space up
Is the next marker a matching character?
Yes
Increment VCount
No
Is VCount >= 5 ?
Yes
Return True
No
Is VCount >= 5 ?
Yes
No
Return False

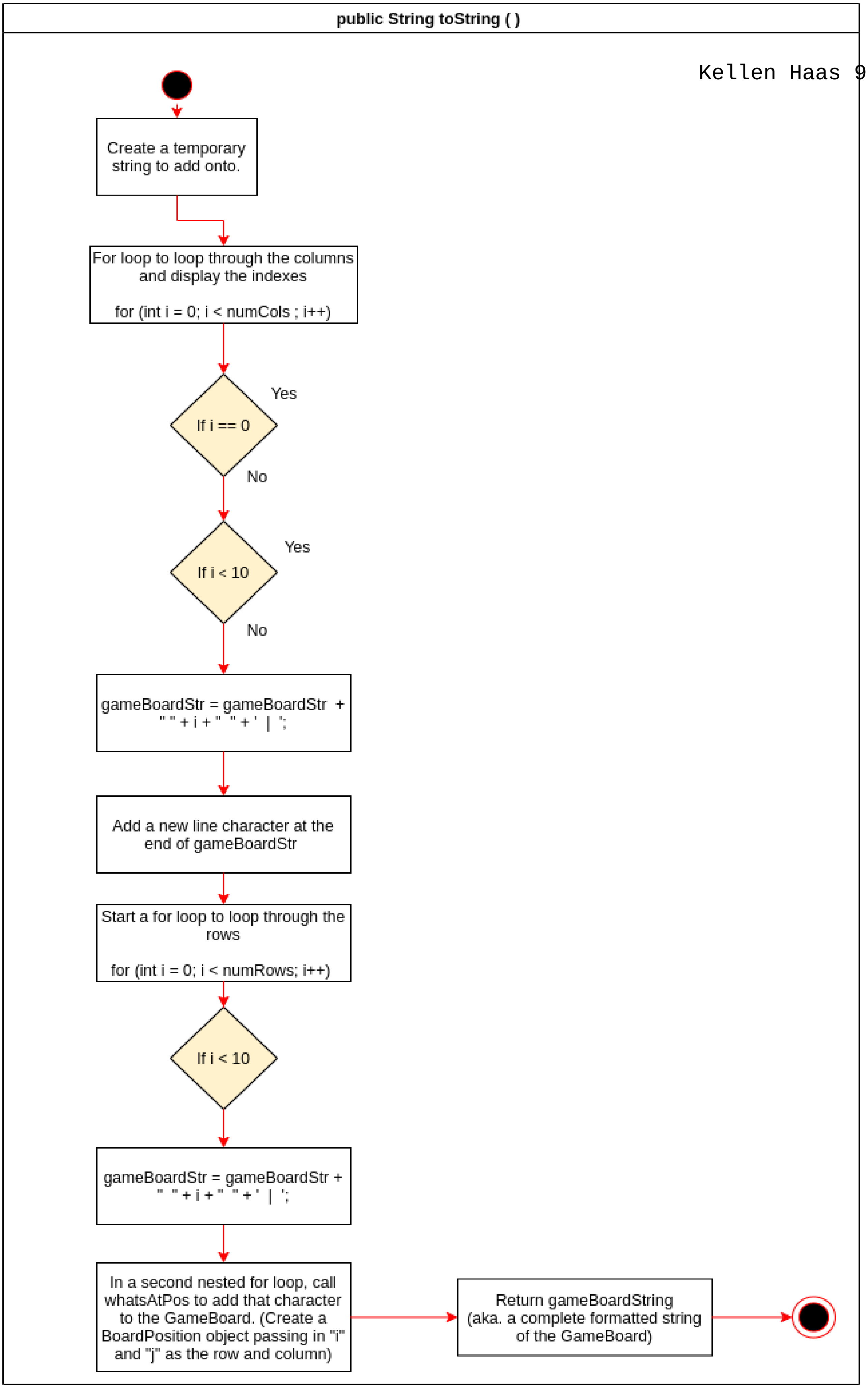
publi c boolean checkDiagonalWin(BoardPosition lastPos)


Start a loop to decrement down an d to the le ft from the lastPosinserted in the board {only decrement in bounds)
Decrement one space down and one space left
Is the next marker a matching character?
Yes
Incremen t the DCount
No
Is DCount >= 5 ?
Yes
No
Is DCount >= 5 ?
Yes
No
Exit loop
Start loop to increment up and to the right from lastPos inserted in the board (only increment in bounds)
Increment one space up and one to the right
Is the next marker a matching character?
Yes
Increment DCount
No
Is DCount >= 5 ?
Yes
No
Is DCount >= 5 ?
Yes
No
Start a loop ID decrement down and to the rig ht from the las1Pos inserted in the board (only decrement in bounds)
Decrement one space down and one space right
Is the next marker a matching character?
Yes
Increment the DCoum
No
Is DCount >= 5 ?
Yes
No
Is DCount >= 5 ?
Yes
No
Exit loop
Start loop to incremerit up and to the left from lastPos inserted in the board (only increment in bounds)
Increment one space up and one space left
Is the next marker a matching character?
Yes
Increment DCount
No
Is DCount >= 5 ?
Yes
No
Is DCount >= 5 ?
Yes
No
Return True
Return False

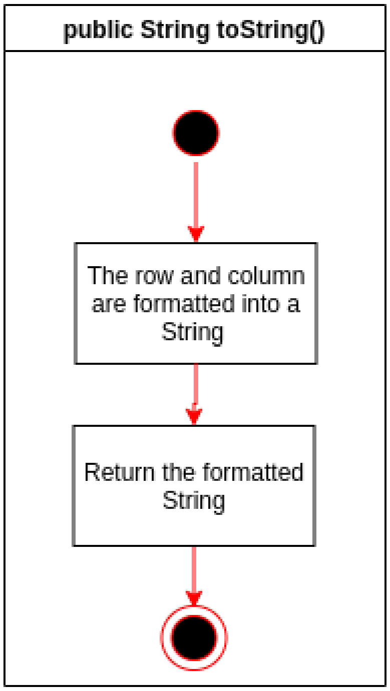

public char whatsAtPos(BoardPosition pos)
Check what is on the game board in position pos
Is there an X or an O in the position pos on the game board?
Yes
Return the character that is on the game board at position pos.
No
Return a blank space character: ' '

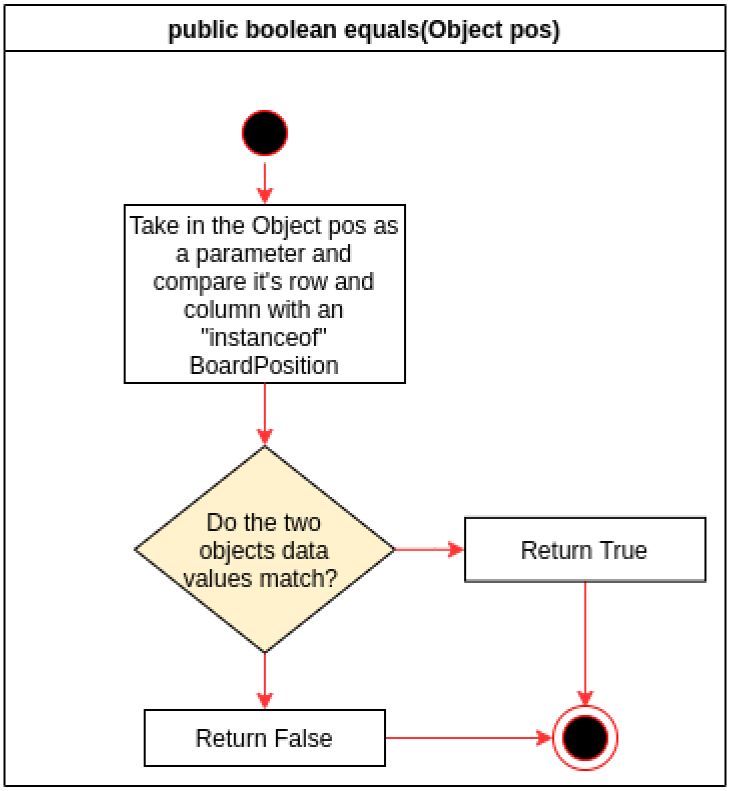

public boolean isPlayerAtPos(BoardPosition pos, char player)
Check the marker at position pos on the gameboard to see if it matches player.
Is GameBoard [pos.getRow()] [pos.getColumn()] equal to player?
Return True
Return False

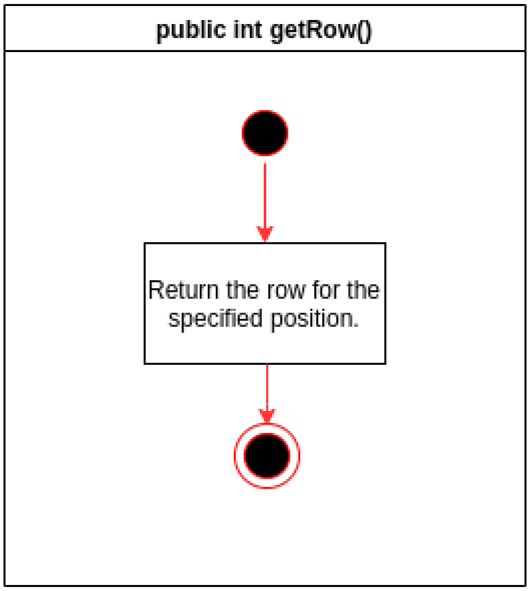
Kellen Haas 9

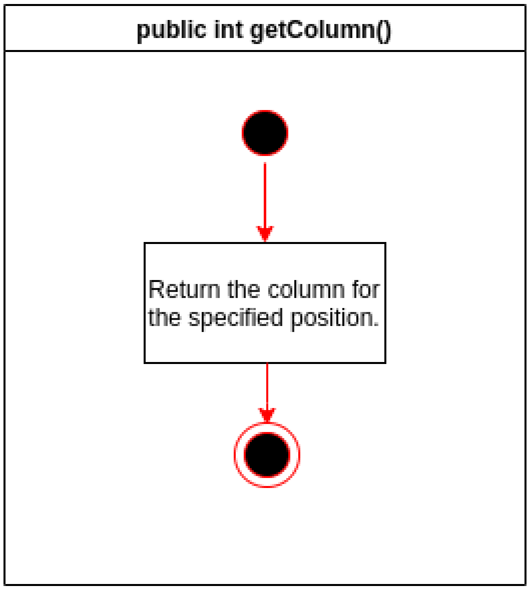

public String toString ( )
Create a temporary string to add onto.
For loop to loop through the columns and display the indexes
for (int i = 0; i < numCols ; i++)
If i == 0
Yes
No
If i < 10
Yes
No
gameBoardStr = gameBoardStr + " " + i + " " + ' | ';
Add a new line character at the end of gameBoardStr
Start a for loop to loop through the rows
for (int i = 0; i < numRows; i++)
If i < 10
gameBoardStr = gameBoardStr + " " + i + " " + ' | ';
In a second nested for loop, call whatsAtPos to add that character to the GameBoard. (Create a BoardPosition object passing in "i" and "j" as the row and column)
Return gameBoardString (aka. a complete formatted string of the GameBoard)

BoardPosition.java


public String toString()
The row and column are formatted into a String
Return the formatted String

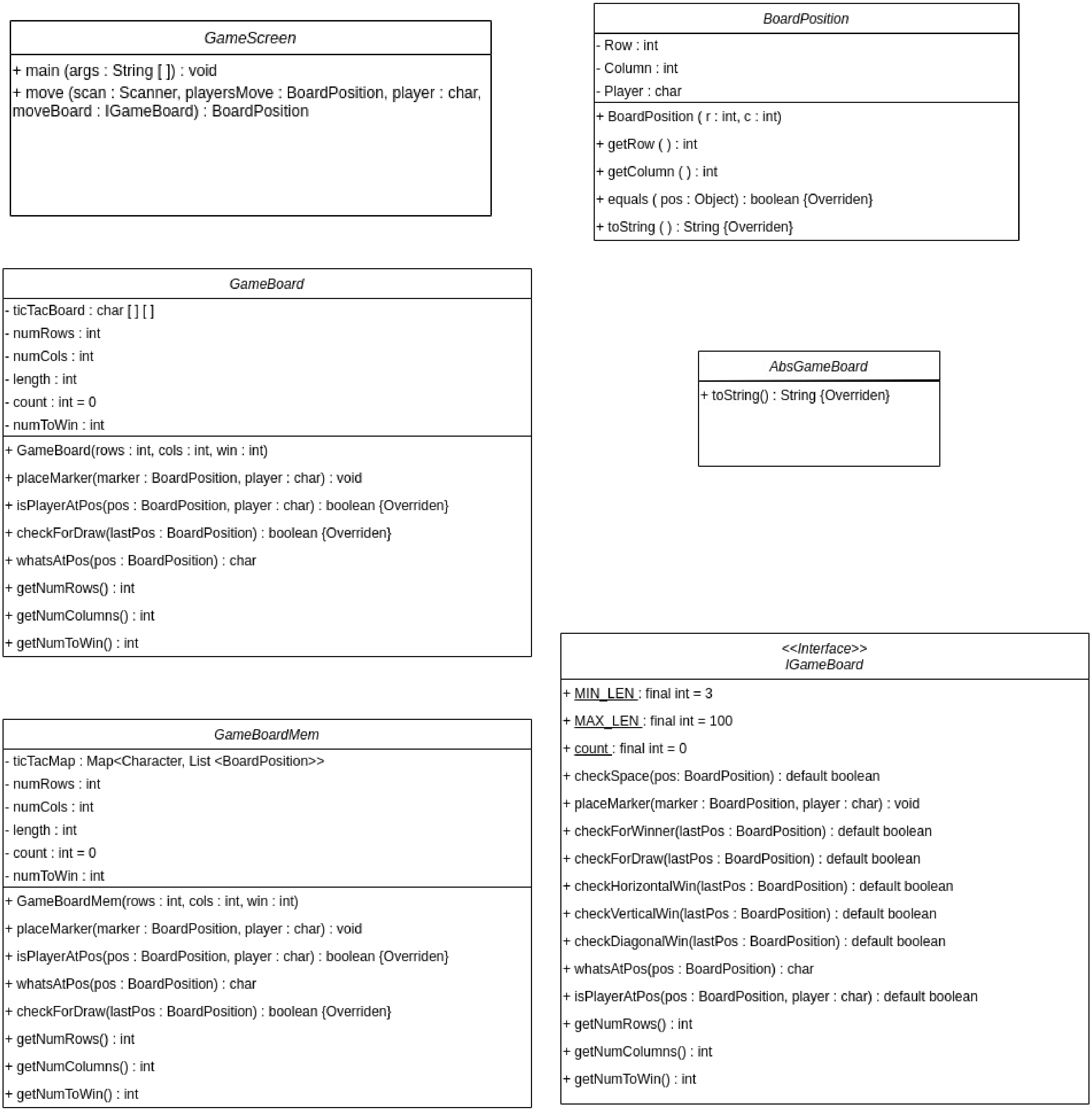

public boolean equals(Object pos)
Take in the Object pos as a parameter and compare it's row and column with an "instanceof" BoardPosition
Do the two objects data values match?
Return True
Return False





Kellen Haas 11

UML Class Diagrams



GameScreen

+ main (args : String [ ]) : void
+ move (scan : Scanner, playersMove : BoardPosition, player : char, moveBoard : IGameBoard) : BoardPosition

BoardPosition

- Row : int
- Column : int
- Player : char

+ BoardPosition (r : int, c : int)
+ getRow ( ) : int
+ getColumn ( ) : int
+ equals (pos : Object) : boolean {Overriden}
+ toString ( ) : String {Overriden}

GameBoard

- ticTacBoard : char [ ] [ ]
- numRows : int
- numCols : int
- length : int
- count : int = 0
- numToWin : int

+ GameBoard(rows : int, cols : int, win : int)
+ placeMarker(marker : BoardPosition, player : char) : void
+ isPlayerAtPos(pos : BoardPosition, player : char) : boolean {Overriden}
+ checkForDraw(lastPos : BoardPosition) : boolean {Overriden}
+ whatsAtPos(pos : BoardPosition) : char
+ getNumRows() : int
+ getNumColumns() : int
+ getNumToWin() : int

AbsGameBoard

+ toString() : String {Overriden}

GameBoardMem

- ticTacMap : Map<Character, List <BoardPosition>>
- numRows : int
- numCols : int
- length : int
- count : int = 0
- numToWin : int

+ GameBoardMem(rows : int, cols : int, win : int)
+ placeMarker(marker : BoardPosition, player : char) : void
+ isPlayerAtPos(pos : BoardPosition, player : char) : boolean {Overriden}
+ whatsAtPos(pos : BoardPosition) : char
+ checkForDraw(lastPos : BoardPosition) : boolean {Overriden}
+ getNumRows() : int
+ getNumColumns() : int
+ getNumToWin() : int

<<Interface>>
IGameBoard

+ MIN_LEN : final int = 3
+ MAX_LEN : final int = 100
+ count : final int = 0
+ checkSpace(pos: BoardPosition) : default boolean
+ placeMarker(marker : BoardPosition, player : char) : void
+ checkForWinner(lastPos : BoardPosition) : default boolean
+ checkForDraw(lastPos : BoardPosition) : default boolean
+ checkHorizontalWin(lastPos : BoardPosition) : default boolean
+ checkVerticalWin(lastPos : BoardPosition) : default boolean
+ checkDiagonalWin(lastPos : BoardPosition) : default boolean
+ whatsAtPos(pos : BoardPosition) : char
+ isPlayerAtPos(pos : BoardPosition, player : char) : default boolean
+ getNumRows() : int
+ getNumColumns() : int
+ getNumToWin() : int