Requirement Analysis:

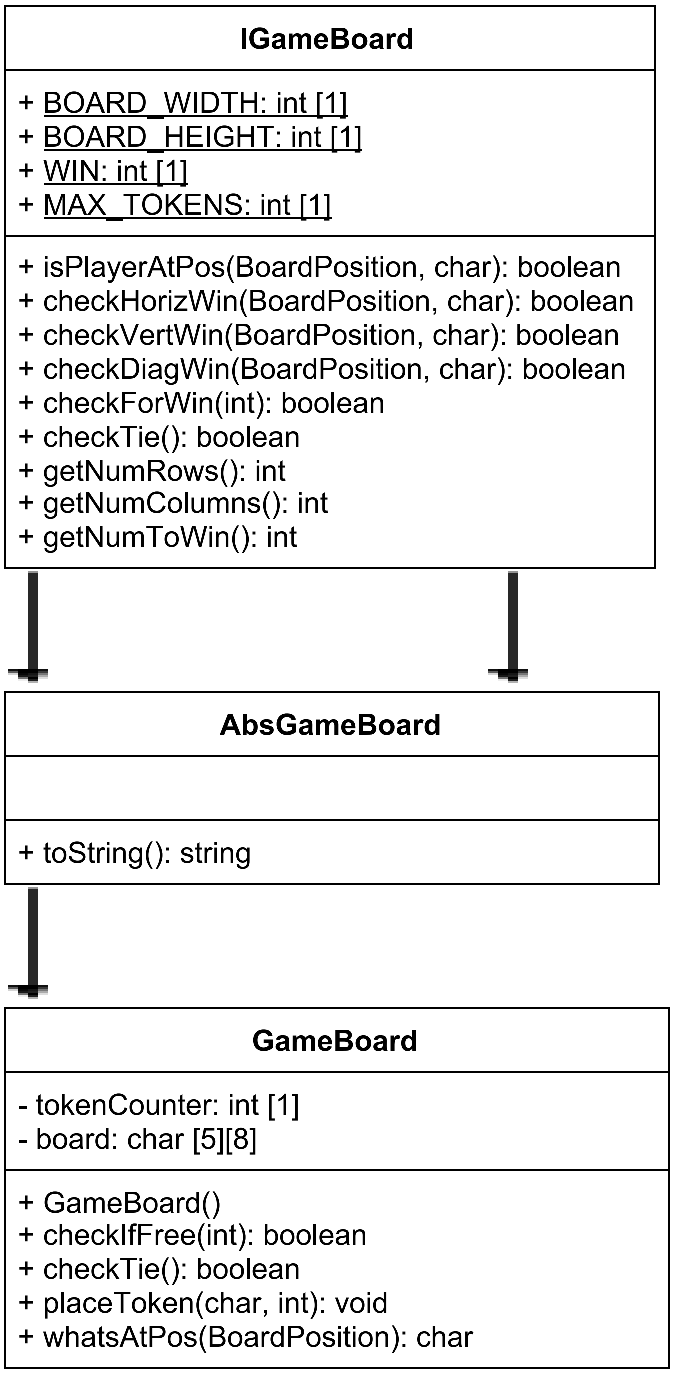
Functional Requirements:

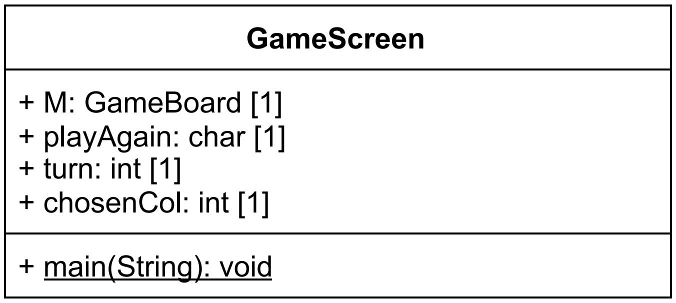
- As a player, I can be assigned player X or O to determine which tokens I will use and who gets the first move.
- As a player, I can place a token in a chosen column to try and connect 5 horizontally.
- As a player, I can place a token in a chosen column to try and connect 5 vertically.
- As a player, I can place a token in a chosen column to try and connect 5 diagonally.
- As a player, I can place enough tokens to reach the max count which will result in a tie condition.
- As a player, I can decide whether or not to play again at the end of a game by typing ‘y’ or ‘n’.
- As a player, I swap turns with another player, placing a token after the other player does so that the game is fair.
- As a player, I can select any column to place a token in, but if I am out of bounds when placing I will be prompted to place again, so that I do not waste a turn.
- As a player, I have access to seeing the game board to determine my next token placement.
- The game must accept column integer input from the user.
- The game must check to make sure that a column is not full when a player tries to place a token.
- The game must check to see if a player has won by connecting 5 tokens in a row horizontally, vertically, or diagonally.

Non-Functional Requirements:

- The game executable must run on Unix.
- The game must be a 6x9 board of characters
- The game must allow player X to always go first
- The game must allow <0>,<0> to be the bottom left board position on the board
- The game must allow <5>,<8> to be the top right board position on the board
- The game code must be able to compile in Java 11.
- The game code must be able to run in Java 11.

UML Class Diagrams:

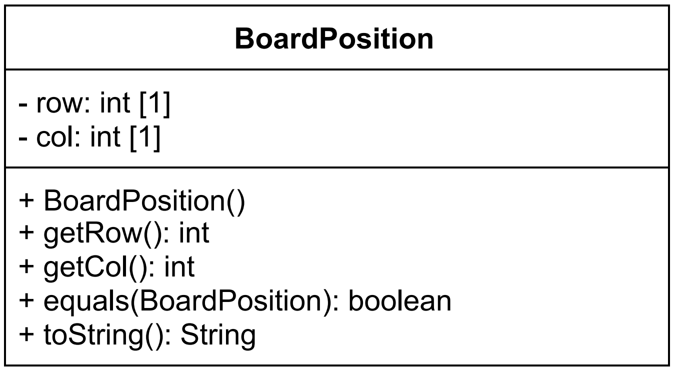

IGameBoard
+ BOARD_WIDTH: int [1]
+ BOARD_HEIGHT: int [1]
+ WIN: int [1]
+ MAX_TOKENS: int [1]
+ isPlayerAtPos(BoardPosition, char): boolean
+ checkHorizWin(BoardPosition, char): boolean
+ checkVertWin(BoardPosition, char): boolean
+ checkDiagWin(BoardPosition, char): boolean
+ checkForWin(int): boolean
+ checkTie(): boolean
+ getNumRows(): int
+ getNumColumns(): int
+ getNumToWin(): int
AbsGameBoard
+ toString(): string
GameBoard
- tokenCounter: int [1]
- board: char [5][8]
+ GameBoard()
+ checkIfFree(int): boolean
+ checkTie(): boolean
+ placeToken(char, int): void
+ whatsAtPos(BoardPosition): char



GameScreen

+ M: GameBoard [1]
+ playAgain: char [1]
+ turn: int [1]
+ chosenCol: int [1]

+ main(String): void



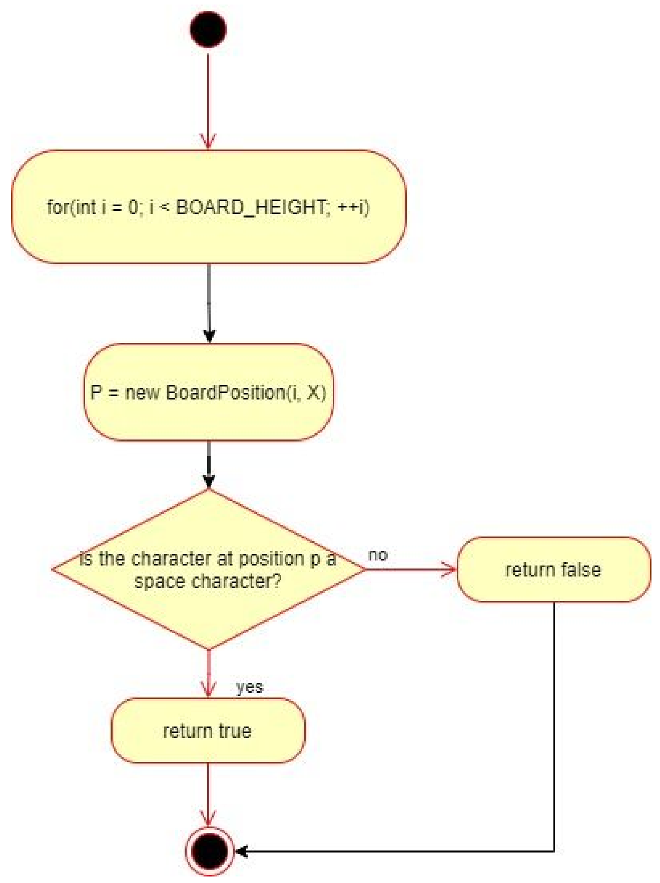
BoardPosition

- row: int [1]
- col: int [1]

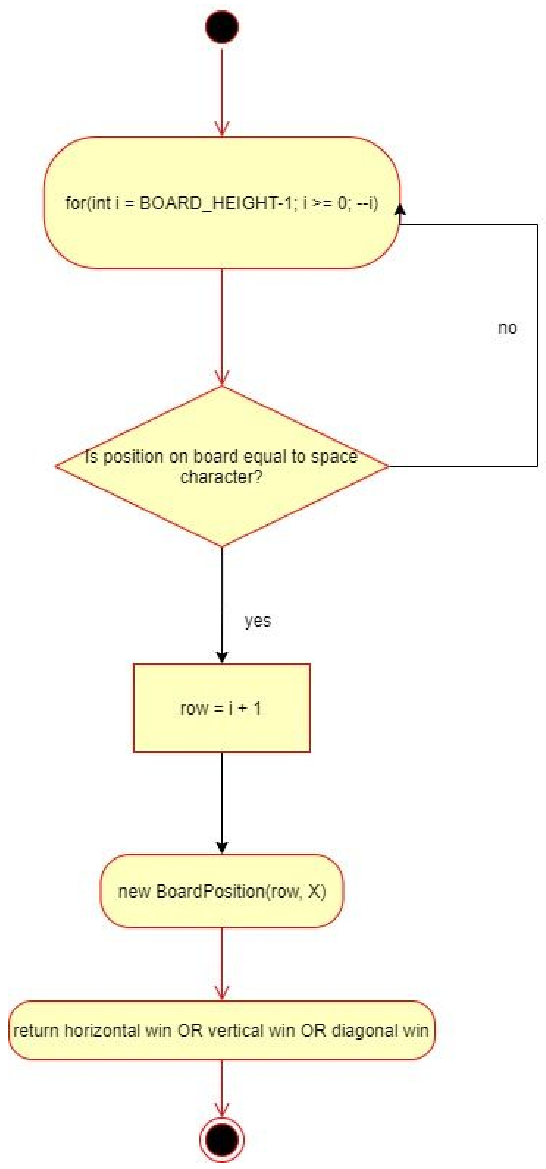
+ BoardPosition()
+ getRow(): int
+ getCol(): int
+ equals(BoardPosition): boolean
+ toString(): String

UML Activity Diagrams:

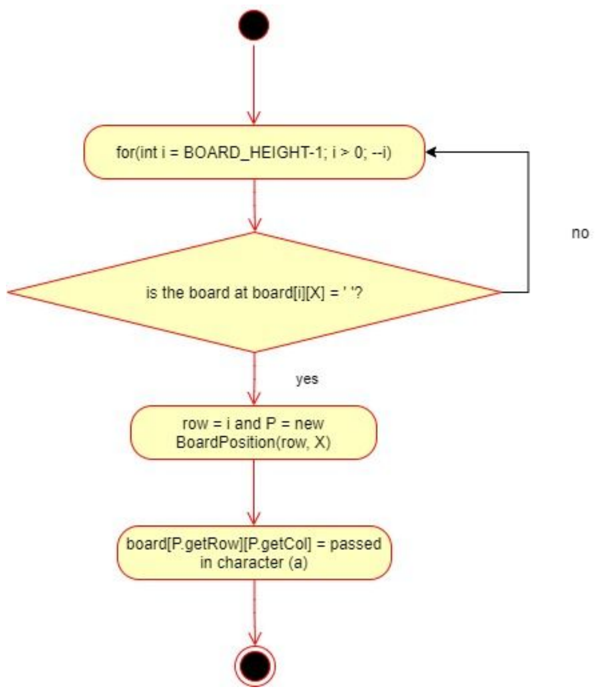
checkIfFree(int): boolean


for(int i = 0; i < BOARD_HEIGHT; ++i)
P = new BoardPosition(i, X)
is the character at position p a space character?
no
return false
yes
return true

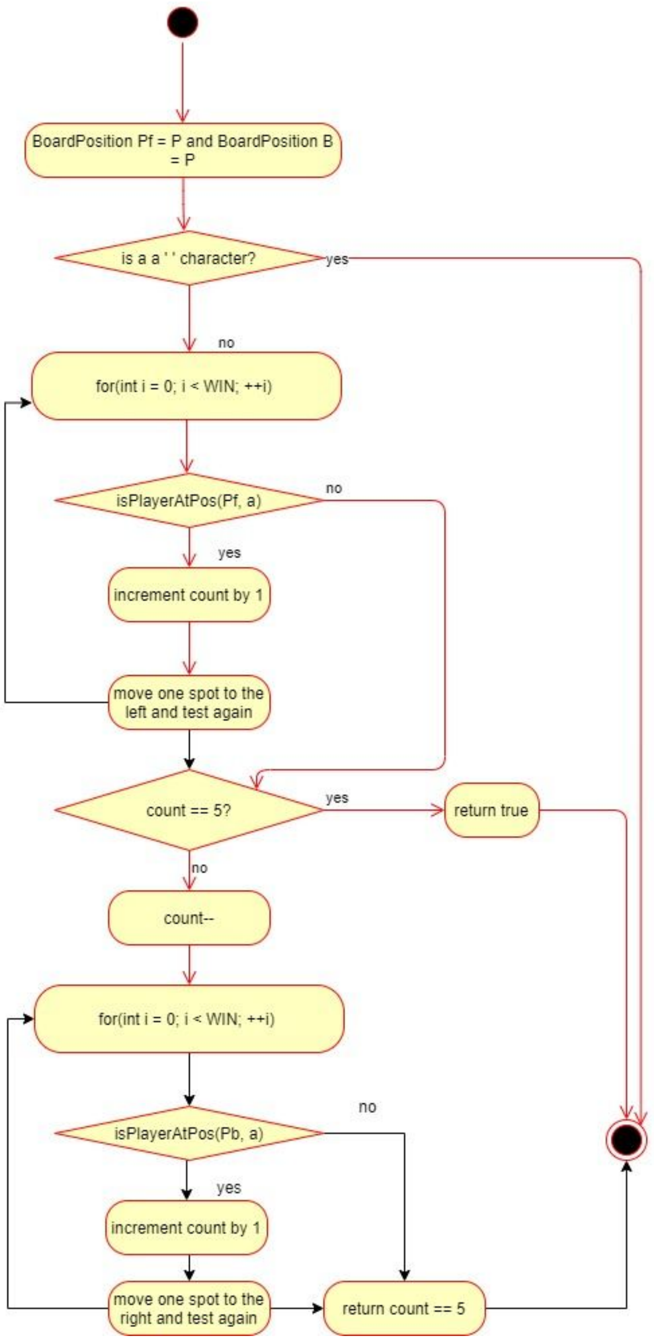
checkForWin(int): boolean


for(int i = BOARD_HEIGHT-1; i >= 0; --i)
no
Is position on board equal to space character?
yes
row = i + 1
new BoardPosition(row, X)
return horizontal win OR vertical win OR diagonal win

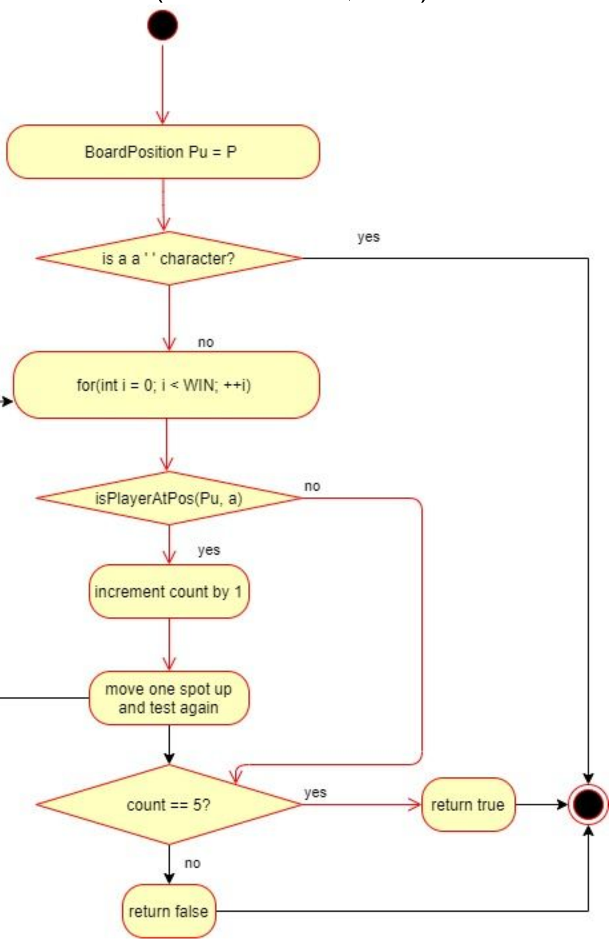
placeToken(char, int): void


for(int i = BOARD_HEIGHT-1; i > 0; --i)
no
is the board at board[i][X] = ' '?
yes
row = i and P = new BoardPosition(row, X)
board[P.getRow][P.getCol] = passed in character (a)

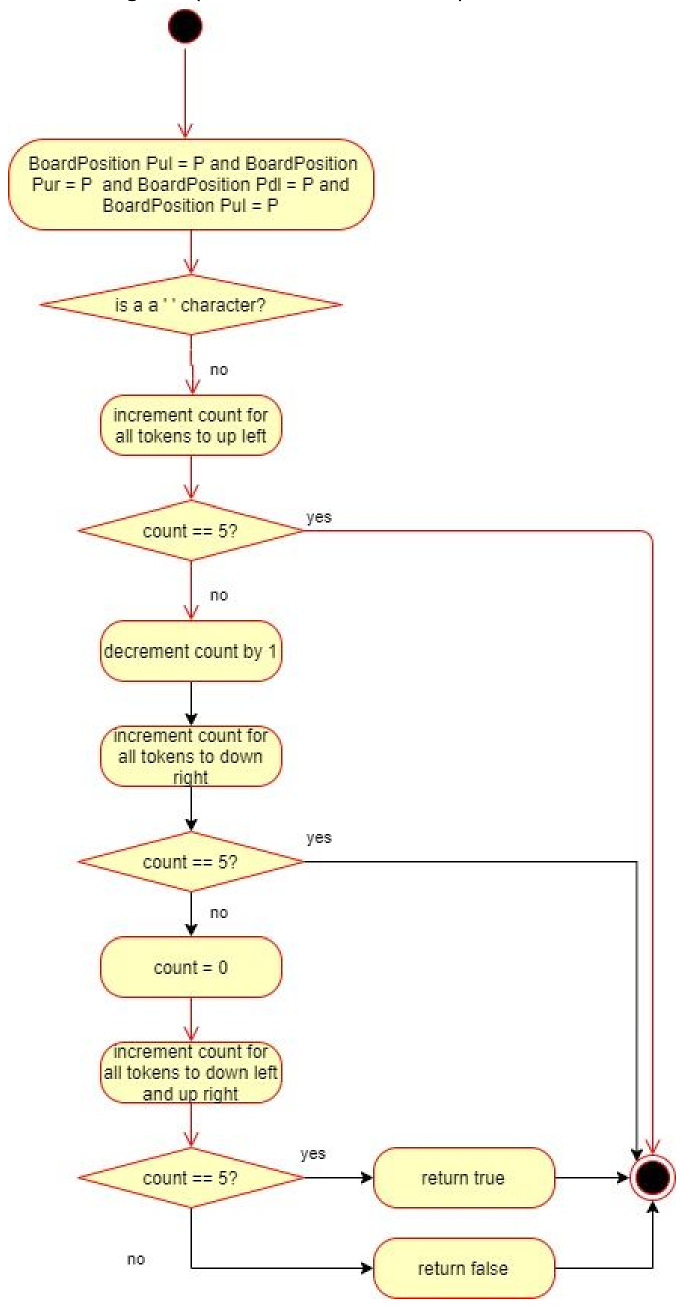
checkHorizWin(BoardPosition, char): boolean


BoardPosition Pf = P and BoardPosition B = P
is a a ' ' character?
yes
no
for(int i = 0; i < WIN; ++i)
isPlayerAtPos(Pf, a)
no
yes
increment count by 1
move one spot to the left and test again
count == 5?
yes
return true
no
count--
for(int i = 0; i < WIN; ++i)
isPlayerAtPos(Pb, a)
no
yes
increment count by 1
move one spot to the right and test again
return count == 5

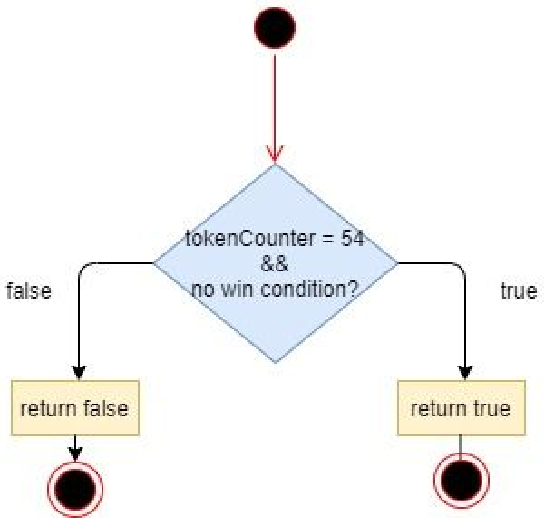
checkVertWin(BoardPosition, char): boolean



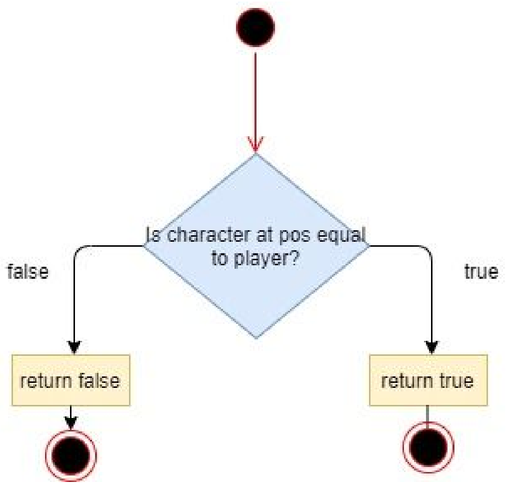
checkDiagWin(BoardPosition, char): boolean


BoardPosition Pul = P and BoardPosition Pur = P and BoardPosition Pdl = P and BoardPosition Pul = P
is a a ' ' character?
no
increment count for all tokens to up left
count == 5?
yes
no
decrement count by 1
increment count for all tokens to down right
count == 5?
yes
no
count = 0
increment count for all tokens to down left and up right
count == 5?
yes
return true
no
return false

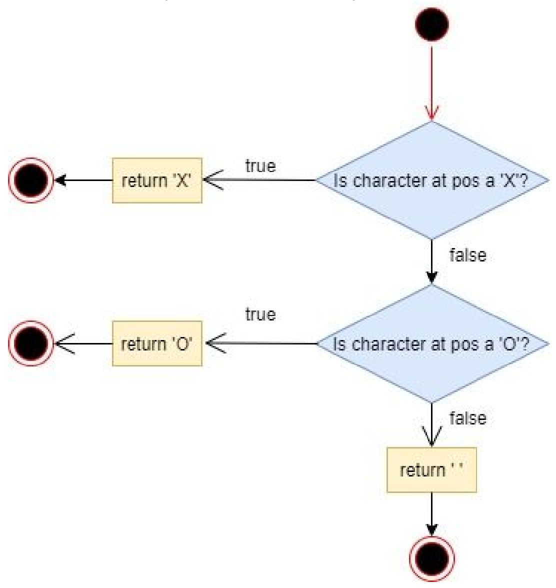
checkTie(): boolean



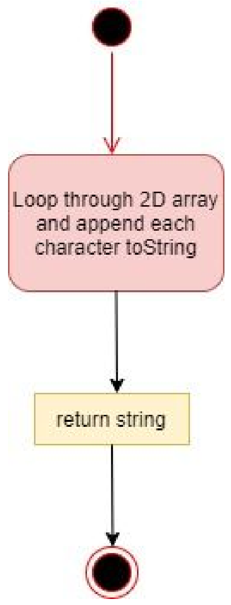
isPlayerAtPos(BoardPosition, char): boolean



whatsAtPos(BoardPosition): char



toString(): String



Loop through 2D array and append each character toString
return string