Manning Graham
Project 1
CPSC 2150
9-22-2021
Requirements Analysis
Functional Requirements:
1) Player
a) Place Token: As a player, I can decide where I wish to place my token during the game so
I can play and try to win.
b) Exit Game: As a player, after a game is completed I have the option of whether or not to
continue playing in a new game or exit the game.
c) Win game: As a player after I place a piece I am able to win 3 different ways. By having
five pieces in a row horizontally, vertically, or diagonally.
d) Tie game: As a player, I can tie the game if there are no options of winning left for me or
my opponent.
e) Out of bounds: As a player, if I choose a placement for a token out of bounds I am able to
choose another placement to continue playing without error.
f) Playing again: As a player, after a game is completed I have the option of playing again.
g) Moving after opponent: As a player after my opponent has made their move, It is then my
turn and I am able to make my move.

#### Non-Functional Requirements:

- 1) The program should know whether the player's column input location is valid or not.
- 2) The program should know whether the column input is even on the board.
- 3) The program should know whether the player has won after a move, either vertically, horizontally, or diagonally.
- 4) The program should know if there are any more spaces left to be played or if it is a tie.
- 5) The program should know who the winner is and display that they are the winner immediately after the winning move.
- 6) The program should know to display a blank game board after a win if the users wish to play again.
- 7) The program should be able to tell what player is taking up space in a certain position.
- 8) The program should be able to tell what token (X or O) is in what position
- 9) The program should be able to display the current game board after each move.
- 10) The program should establish a 5x8 game board.
- 11) The program should know 0,0 is the top left of the board.
- 12) The program should know that X goes first.
- 13) The program should be written in java.
- 14) The program should run on Unix.

## **Class Diagrams**

#### GameScreen:

## GameScreen

Board : GameBoard [1..\*]

- Row : int

- Column : int

- GameCounter: int

+ main(): void

## GameBoard:

#### GameBoard

## GameBoard: BoardPosition[][]

- + GameBoard();
- + checkSpace(BoardPosition pos) : boolean
- + placeMarker(BoardPosition marker, char player) : void
- + checkForWinner(BoardPosition lastPos) : boolean
- + checkForDraw(): void
- + checkHorizontalWin(BoardPosition lastPos, char player) : boolean
- + checkVerticalWin(BoardPosition lastPos, char player) : boolean
- + checkDiagonalWin(BoardPosition lastPos, char player) : boolean
- + whatsAtPos (BoardPosition) : char
- + isPlayerAtPos(BoardPosition pos, char player) : boolean
- + toString(): String

# BoardPosition:

# BoardPosition

- rowPos: int

- colPos : int

+ BoardPosition( int, int);

+ equals(BoardPosition): bool

+ toString(): String

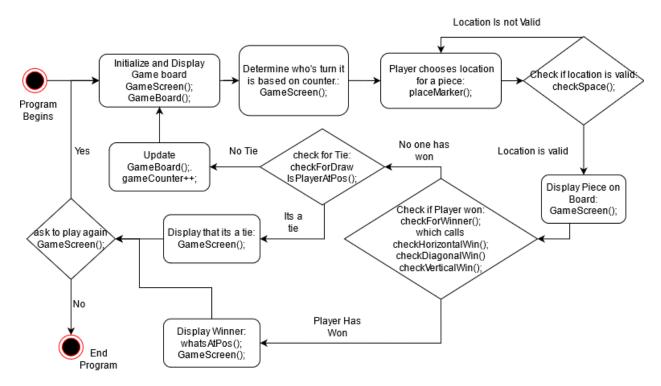
+ getCol() : int

+ getRow(): int

# **Activity Diagrams**

# GameScreen Activity Diagram:

## Main:



# GameBoard Activity Diagrams:

**End Program** 

Program Begins

Is Space In bounds?

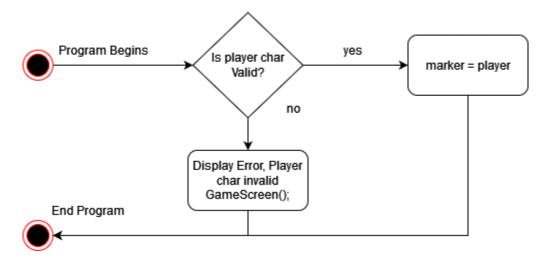
Check If Player is at pos. IsPlayerAtPos();

No player at pos

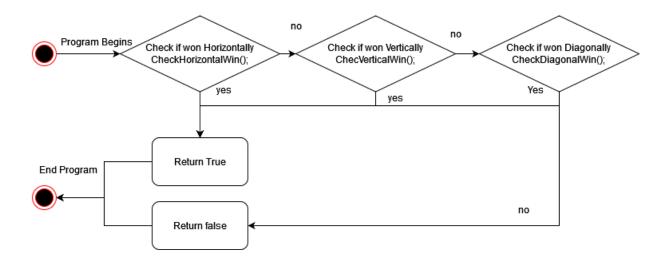
Return False

Return True

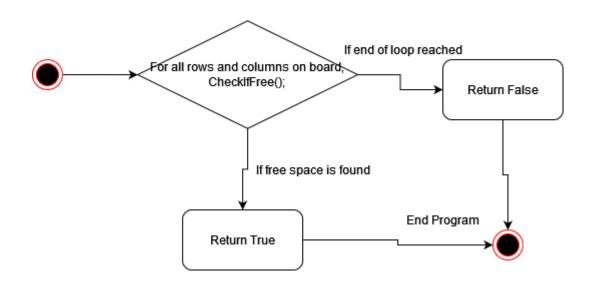
void placeMarker(BoardPosition marker, char player)



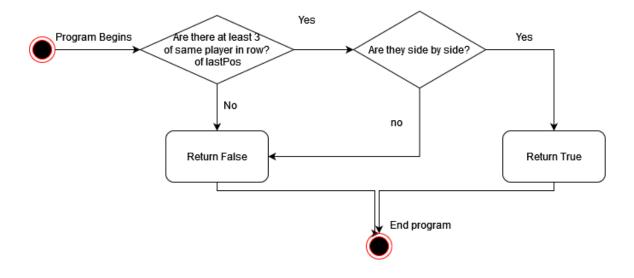
public boolean checkForWinner(BoardPosition lastPos)



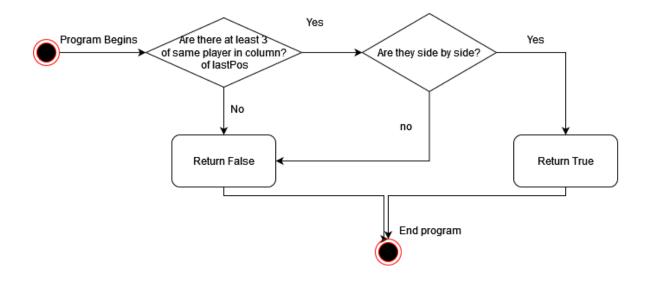
public boolean checkForDraw()



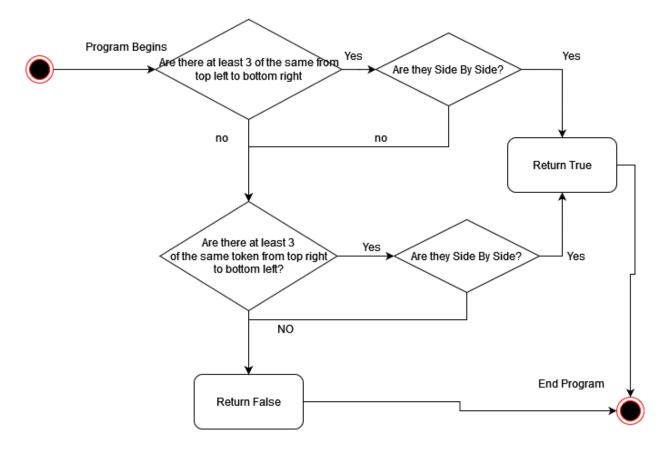
publicboolean checkHorizontalWin(BoardPosition lastPos, char player)

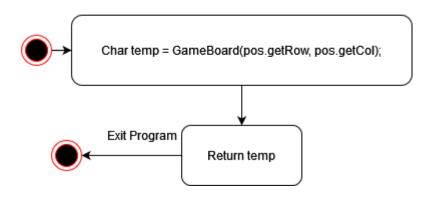


publicboolean checkVerticalWin(BoardPosition lastPos, char player)



publicboolean checkDiagonalWin(BoardPosition lastPos, char player)





boolean isPlayerAtPos(BoardPosition pos, char player)

