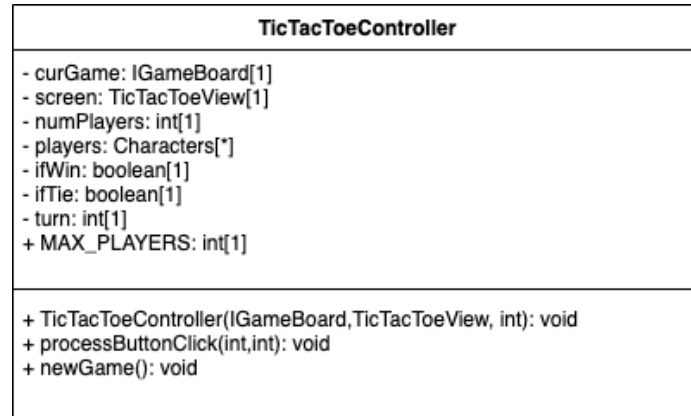
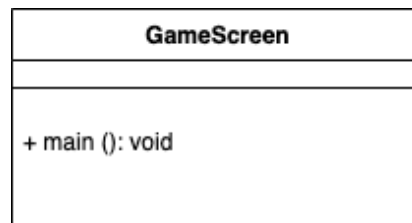


Sydney Trout  
Professor Van Scoy  
CPSC 2150  
Project 4  
11 November 2021

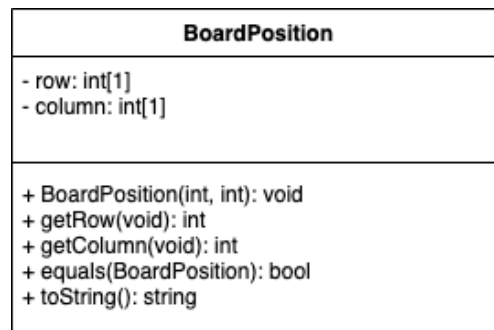
TicTacToeController Class Diagram:



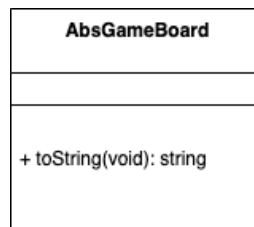
GameScreen Class Diagram:



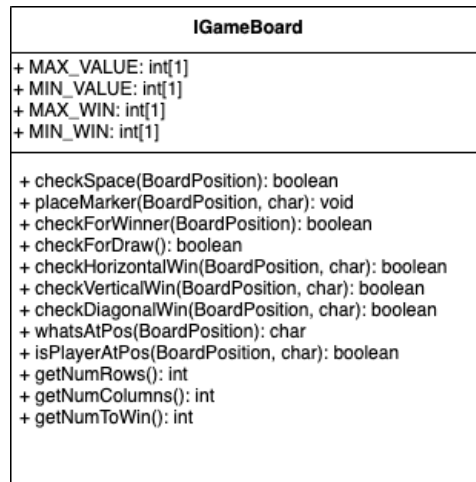
BoardPosition Class Diagram:



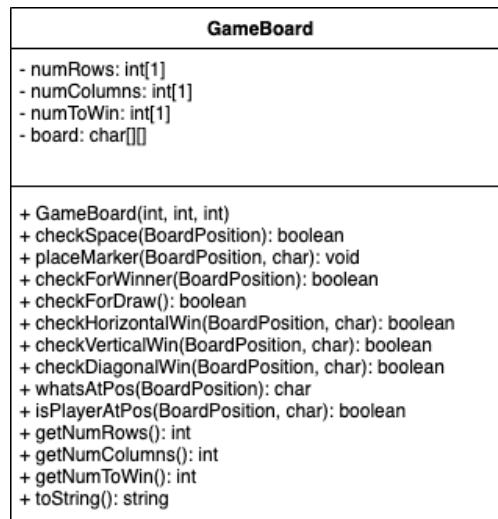
AbsGameBoard Class Diagram:



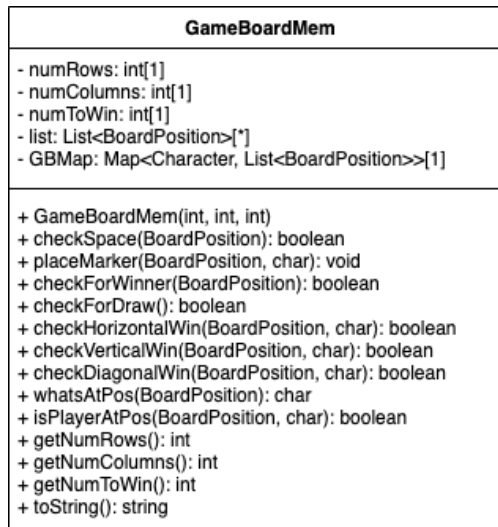
### IGameBoard Class Diagram:



### GameBoard Class Diagram:



### GameBoardMem Class Diagram:



## Requirement Analysis:

### User Stories

1. As a player, I can enter a row and column to place my marker there to try and win.
2. As a player, I can type 'Y' at the end of the game to play again.
3. As a player, I can type 'N' at the end of the game to quit.
4. As a player, I can enter the number of rows to make the board.
5. As a player, I can enter the number of columns to make the board.
6. As a player, I can enter the number to win to make the win requirements.
7. As a player, I can pick the implementation of the game to play fast or save memory.
8. As a player, I can place markers in a horizontal row to win the game.
9. As a player, I can place markers in a vertical column to win the game.
10. As a player, I can place markers in a diagonal line to win the game.
11. As a player, I can place a horizontal row, a vertical row, or a diagonal line to win the game.
12. As a player, I can pick the character for the markers to differentiate the players.
13. As a player, I can fill the entire board to tie the game.
14. As a player, I can make a move to switch to a different player.
15. As a player, I can view the game board to decide which play to make next.
16. As a player, I can pick again if my original pick was occupied.
17. As a player, I can pick again if my original pick was out of the boundaries.

### NonFunctional Requirements

1. The program must be written in Java.
2. The program must run on Windows Unix.
3. The program must run on MacOS.
4. The program must produce a GUI.

### Deployment Instructions

Compile the program by typing "make" in the terminal. Typing "make run" will run the program. To compile the test cases type "make test." Then "make testGB" or "make testGBmem" will run test cases on the program. "Make clean" will then remove the compiled .class files.

**GameBoard(int nr, int nc, int ntw) / GameBoardMem(int nr, int nc, int ntw)**

|   |   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |   |
|---|---|---|---|---|---|---|--|--|--|---|--|--|--|---|--|--|--|---|
| <b>Input:</b><br><br>State:<br>Nothing<br><br>Rows = 3<br>Columns = 3<br>NumToWin = 3 | <b>Output:</b> <table><tr><td></td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td></tr></table><br>gb.getRow() = 3<br>gb.getColumn() = 3<br>gb.getNumToWin() = 3 |   | 0 | 1 | 2 | 0 |  |  |  | 1 |  |  |  | 2 |  |  |  | <b>Reason:</b><br>This test case is unique and distinct because it calls the constructor method with the minimum number of rows, columns, and number to win.<br><br><b>Function Name:</b><br>min_size_constructor |
|   | 0   | 1 | 2 |   |   |   |  |  |  |   |  |  |  |   |  |  |  |   |
| 0   |   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |   |
| 1   |   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |   |
| 2   |   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |   |

|  |  |   |     |     |     |     |   |  |  |  |  |   |  |  |  |  |     |  |  |  |  |     |  |  |  |  |   |
|--|--|---|-----|-----|-----|-----|---|--|--|--|--|---|--|--|--|--|-----|--|--|--|--|-----|--|--|--|--|---|
| <b>Input:</b><br><br>State:<br>Nothing<br><br>Rows = 100<br>Columns = 100<br>NumToWin = 25 | <b>Output:</b> <table><tr><td></td><td>0</td><td>1</td><td>...</td><td>100</td></tr><tr><td>0</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td></tr><tr><td>...</td><td></td><td></td><td></td><td></td></tr><tr><td>100</td><td></td><td></td><td></td><td></td></tr></table><br>gb.getRow() = 100<br>gb.getColumn() = 100<br>gb.getNumToWin() = 25 |   | 0   | 1   | ... | 100 | 0 |  |  |  |  | 1 |  |  |  |  | ... |  |  |  |  | 100 |  |  |  |  | <b>Reason:</b><br>This test case is unique and distinct because it calls the constructor method with the maximum number of rows, columns, and number to win.<br><br><b>Function Name:</b><br>max_size_constructor |
|  | 0  | 1 | ... | 100 |     |     |   |  |  |  |  |   |  |  |  |  |     |  |  |  |  |     |  |  |  |  |   |
| 0  |  |   |     |     |     |     |   |  |  |  |  |   |  |  |  |  |     |  |  |  |  |     |  |  |  |  |   |
| 1  |  |   |     |     |     |     |   |  |  |  |  |   |  |  |  |  |     |  |  |  |  |     |  |  |  |  |   |
| ...  |  |   |     |     |     |     |   |  |  |  |  |   |  |  |  |  |     |  |  |  |  |     |  |  |  |  |   |
| 100  |  |   |     |     |     |     |   |  |  |  |  |   |  |  |  |  |     |  |  |  |  |     |  |  |  |  |   |

|   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
|---|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|
| <b>Input:</b><br><br>State:<br>Nothing<br><br>Rows = 5<br>Columns = 5<br>NumToWin = 5 | <b>Output:</b> <table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>gb.getRow() = 5<br>gb.getColumn() = 5<br>gb.getNumToWin() = 5 |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  |  |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Reason:</b><br>This test case is unique and distinct because it calls the constructor method with a normal number of rows, columns, and number to win.<br><br><b>Function Name:</b><br>normal_size_constructor |
|   | 0   | 1 | 2 | 3 | 4 |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 0   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 1   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 2   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 3   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 4   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |

**boolean checkSpace(BoardPosition pos)**

|   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |  |  |
|---|---|---|---|---|---|--|--|--|---|--|--|--|---|--|--|--|--|--|
| <b>Input:</b><br>State:<br><table><tr><td></td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td></tr></table><br>badPos.getRow() = 1<br>badPos.getColumn() = 4 |   | 0 | 1 | 2 | 0 |  |  |  | 1 |  |  |  | 2 |  |  |  | <b>Output:</b><br><br>checkSpace = false<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check space on a BoardPosition that is out of bounds<br><br><b>Function Name:</b><br>out_of_bounds_checkSpace |
|   | 0 | 1 | 2 |   |   |  |  |  |   |  |  |  |   |  |  |  |  |  |
| 0   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |  |  |
| 1   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |  |  |
| 2   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |  |  |

|  |   |   |   |   |   |  |  |  |   |  |  |   |   |  |  |  |  |  |
|--|---|---|---|---|---|--|--|--|---|--|--|---|---|--|--|--|--|--|
| <b>Input:</b><br>State:<br><table><tr><td></td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td>x</td></tr><tr><td>2</td><td></td><td></td><td></td></tr></table><br>badPos.getRow() = 1<br>badPos.getColumn() = 2 |   | 0 | 1 | 2 | 0 |  |  |  | 1 |  |  | x | 2 |  |  |  | <b>Output:</b><br><br>checkSpace = false<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check space on a BoardPosition that is already full<br><br><b>Function Name:</b><br>already_full_checkSpace |
|  | 0 | 1 | 2 |   |   |  |  |  |   |  |  |   |   |  |  |  |  |  |
| 0  |   |   |   |   |   |  |  |  |   |  |  |   |   |  |  |  |  |  |
| 1  |   |   | x |   |   |  |  |  |   |  |  |   |   |  |  |  |  |  |
| 2  |   |   |   |   |   |  |  |  |   |  |  |   |   |  |  |  |  |  |

|   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |   |   |
|---|---|---|---|---|---|--|--|--|---|--|--|--|---|--|--|--|---|---|
| <b>Input:</b><br>State:<br><table><tr><td></td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td></tr></table><br>badPos.getRow() = 1<br>badPos.getColumn() = 2 |   | 0 | 1 | 2 | 0 |  |  |  | 1 |  |  |  | 2 |  |  |  | <b>Output:</b><br><br>checkSpace = true<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check space on a BoardPosition that is empty<br><br><b>Function Name:</b><br>normal_checkSpace |
|   | 0 | 1 | 2 |   |   |  |  |  |   |  |  |  |   |  |  |  |   |   |
| 0   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |   |   |
| 1   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |   |   |
| 2   |   |   |   |   |   |  |  |  |   |  |  |  |   |  |  |  |   |   |

**boolean checkHorizontalWin(BoardPosition lastPos, char player)**

|  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
|--|---|---|---|---|---|---|---|---|---|---|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 0<br>player = 'x'   |   | 0 | 1 | 2 | 3 | 4 | 0 | x |   |   |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkHorizontalWin = false<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check horizontal win on a position that does not create a win<br><br><b>Function Name:</b><br>not_a_win_checkHorizontalWin              |
|  | 0 | 1 | 2 | 3 | 4 |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 0  | x |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 1  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 2  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 3  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 4  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td>x</td><td>x</td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 2<br>player = 'x' |   | 0 | 1 | 2 | 3 | 4 | 0 | x | x | x |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkHorizontalWin = true<br>State of board unchanged  | <b>Reason:</b><br>This test case is unique and distinct because it calls check horizontal win on a position that is at the right end of the win<br><br><b>Function Name:</b><br>last_on_right_checkHorizontalWin |
|  | 0 | 1 | 2 | 3 | 4 |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 0  | x | x | x |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 1  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 2  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 3  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 4  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |

|  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
|--|---|---|---|---|---|---|---|---|---|---|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td>x</td><td>x</td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 0<br>player = 'x' |   | 0 | 1 | 2 | 3 | 4 | 0 | x | x | x |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkHorizontalWin = true<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check horizontal win on a position that is at the left end of the win<br><b>Function Name:</b><br>last_on_left_checkHorizontalWin |
|  | 0 | 1 | 2 | 3 | 4 |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 0  | x | x | x |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 1  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 2  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 3  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 4  |   |   |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |

|  |   |  |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
|--|---|--|---|---|---|---|---|---|---|---|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)  | <b>Output:</b><br><br>checkHorizontalWin = true<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check horizontal win on a position that is in the middle of the win |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| <table border="1"><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td>x</td><td>x</td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 1<br>player = 'x' |   | 0  | 1 | 2 | 3 | 4 | 0 | x | x | x |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  |  | <b>Function Name:</b><br>last_in_middle_checkHorizontalWin |
|  | 0   | 1  | 2 | 3 | 4 |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 0  | x   | x  | x |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 1  |   |  |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 2  |   |  |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 3  |   |  |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 4  |   |  |   |   |   |   |   |   |   |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |

**boolean checkVerticalWin(BoardPosition lastPos, char player)**

|  |   |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
|--|---|---|---|---|---|---|---|---|--|--|--|--|---|---|--|--|--|--|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 0<br>player = 'x'   |   | 0 | 1 | 2 | 3 | 4 | 0 | x |  |  |  |  | 1 |   |  |  |  |  | 2 |   |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkVerticalWin = false<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check vertical win on a position that does not create a win<br><br><b>Function Name:</b><br>not_a_win_checkVerticalWin          |
|  | 0 | 1 | 2 | 3 | 4 |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 0  | x |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 1  |   |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 2  |   |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 3  |   |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 4  |   |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 2<br>lastPos.getColumn() = 0<br>player = 'x' |   | 0 | 1 | 2 | 3 | 4 | 0 | x |  |  |  |  | 1 | x |  |  |  |  | 2 | x |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkVerticalWin = true<br>State of board unchanged  | <b>Reason:</b><br>This test case is unique and distinct because it calls check vertical win on a position that is at the bottom of a win<br><br><b>Function Name:</b><br>last_on_bottom_checkVerticalWin |
|  | 0 | 1 | 2 | 3 | 4 |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 0  | x |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 1  | x |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 2  | x |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 3  |   |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 4  |   |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |



|  |   |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
|--|---|---|---|---|---|---|---|---|--|--|--|--|---|---|--|--|--|--|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 0<br>player = 'x' |   | 0 | 1 | 2 | 3 | 4 | 0 | x |  |  |  |  | 1 | x |  |  |  |  | 2 | x |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkVerticalWin = true<br>State of board unchanged | <b>Reason:</b><br><br>This test case is unique and distinct because it calls check vertical win on a position that is at the top of a win<br><br><b>Function Name:</b><br>last_on_top_checkVerticalWin |
|  | 0 | 1 | 2 | 3 | 4 |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 0  | x |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 1  | x |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 2  | x |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 3  |   |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 4  |   |   |   |   |   |   |   |   |  |  |  |  |   |   |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |

|   |   |  |
|---|---|--|
| <div><div><div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div>lastPos.getRow() = 1</div><div>lastPos.getColumn() = 0</div><div>player = 'x'</div></div> | <div><div><div>checkVerticalWin = true</div><div>State of board unchanged</div></div></div> | <div><div><div><div>Reason:</div><div>This test case is unique and distinct because it calls check vertical win on a position that is in the middle of a win</div></div><div><div>Function Name:</div><div>last_in_middle_checkVerticalWin</div></div></div></div> |
|---|---|--|

**boolean checkDiagonalWin(BoardPosition lastPos, char player)**

|  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
|--|---|---|---|---|---|---|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|---|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 0<br>player = 'x' |   | 0 | 1 | 2 | 3 | 4 | 0 | x |  |  |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkDiagonalWin = false<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check diagonal win on a position that does not create a win<br><br><b>Function Name:</b><br>not_a_win_checkDiagonalWin |
|  | 0 | 1 | 2 | 3 | 4 |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| 0  | x |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| 1  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| 2  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| 3  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| 4  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |

|  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
|--|---|---|---|---|---|---|---|---|--|--|--|--|---|--|---|--|--|--|---|--|--|---|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td>x</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 2<br>lastPos.getColumn() = 2<br>player = 'x' |   | 0 | 1 | 2 | 3 | 4 | 0 | x |  |  |  |  | 1 |  | x |  |  |  | 2 |  |  | x |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkDiagonalWin = true<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check diagonal win on a position that is in the bottom of the right diagonal of the win<br><br><b>Function Name:</b><br>last_on_bottom_right_checkDiagonalWin |
|  | 0 | 1 | 2 | 3 | 4 |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 0  | x |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 1  |   | x |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 2  |   |   | x |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 3  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 4  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |

|  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
|--|---|---|---|---|---|---|---|---|--|--|--|--|---|--|---|--|--|--|---|--|--|---|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td>x</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 1<br>lastPos.getColumn() = 1<br>player = 'x' |   | 0 | 1 | 2 | 3 | 4 | 0 | x |  |  |  |  | 1 |  | x |  |  |  | 2 |  |  | x |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkDiagonalWin = true<br>State of board unchanged | <b>Reason:</b><br><br>This test case is unique and distinct because it calls check diagonal win on a position that is in the middle of the right diagonal of the win<br><br><b>Function Name:</b><br>last_on_middle_right_checkDiagonalWin |
|  | 0 | 1 | 2 | 3 | 4 |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 0  | x |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 1  |   | x |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 2  |   |   | x |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 3  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 4  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |

|  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
|--|---|---|---|---|---|---|---|---|--|--|--|--|---|--|---|--|--|--|---|--|--|---|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|---|
| <b>Input:</b><br>State: (number to win = 3)  | <b>Output:</b><br><br>checkDiagonalWin = true<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check diagonal win on a position that is in the top of the right diagonal of the win |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| <table border="1"><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td>x</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 0<br>player = 'x' |   | 0   | 1 | 2 | 3 | 4 | 0 | x |  |  |  |  | 1 |  | x |  |  |  | 2 |  |  | x |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  |  | <b>Function Name:</b><br>last_on_top_right_checkDiagonalWin |
|  | 0   | 1   | 2 | 3 | 4 |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| 0  | x   |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| 1  |   | x   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| 2  |   |   | x |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| 3  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |
| 4  |   |   |   |   |   |   |   |   |  |  |  |  |   |  |   |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |   |

|   |   |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
|---|---|---|---|---|---|---|---|--|--|---|--|--|---|--|---|--|--|--|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table border="1"><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td>1</td><td></td><td>x</td><td></td><td></td><td></td></tr><tr><td>2</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 2<br>lastPos.getColumn() = 0<br>player = 'x' |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  | x |  |  | 1 |  | x |  |  |  | 2 | x |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkDiagonalWin = true<br>State of board unchanged | <b>Reason:</b><br><br>This test case is unique and distinct because it calls check diagonal win on a position that is in the bottom of the left diagonal of the win<br><br><b>Function Name:</b><br>last_on_bottom_left_checkDiagonalWin |
|   | 0 | 1 | 2 | 3 | 4 |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 0   |   |   | x |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 1   |   | x |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 2   | x |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 3   |   |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 4   |   |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |

|  |   |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
|--|---|---|---|---|---|---|---|--|--|---|--|--|---|--|---|--|--|--|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td>1</td><td></td><td>x</td><td></td><td></td><td></td></tr><tr><td>2</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 1<br>lastPos.getColumn() = 1<br>player = 'x' |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  | x |  |  | 1 |  | x |  |  |  | 2 | x |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkDiagonalWin = true<br>State of board unchanged | <b>Reason:</b><br><br>This test case is unique and distinct because it calls check diagonal win on a position that is in the middle of the left diagonal of the win<br><br><b>Function Name:</b><br>last_on_middle_left_checkDiagonalWin |
|  | 0 | 1 | 2 | 3 | 4 |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 0  |   |   | x |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 1  |   | x |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 2  | x |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 3  |   |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 4  |   |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |

|  |   |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
|--|---|---|---|---|---|---|---|--|--|---|--|--|---|--|---|--|--|--|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td>1</td><td></td><td>x</td><td></td><td></td><td></td></tr><tr><td>2</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 2<br>player = 'x' |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  | x |  |  | 1 |  | x |  |  |  | 2 | x |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>checkDiagonalWin = true<br>State of board unchanged | <b>Reason:</b><br><br>This test case is unique and distinct because it calls check diagonal win on a position that is in the top of the left diagonal of the win<br><br><b>Function Name:</b><br>last_on_top_left_checkDiagonalWin |
|  | 0 | 1 | 2 | 3 | 4 |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 0  |   |   | x |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 1  |   | x |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 2  | x |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 3  |   |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 4  |   |   |   |   |   |   |   |  |  |   |  |  |   |  |   |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |

**boolean checkForDraw()**

|   |   |   |   |   |   |  |   |
|---|---|---|---|---|---|--|---|
| <b>Input:</b><br>State: (number to win = 3) |   |   |   |   |   | <b>Output:</b><br><br>checkForDraw = false<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check draw on a board with only one input<br><br><b>Function Name:</b><br>one_input_checkForDraw |
|   | 0 | 1 | 2 | 3 | 4 |  |   |
| 0   | x |   |   |   |   |  |   |
| 1   |   |   |   |   |   |  |   |
| 2   |   |   |   |   |   |  |   |
| 3   |   |   |   |   |   |  |   |
| 4   |   |   |   |   |   |  |   |

|   |   |   |   |   |   |   |  |
|---|---|---|---|---|---|---|--|
| <b>Input:</b><br>State: (number to win = 3) |   |   |   |   |   | <b>Output:</b><br><br>checkForDraw = true<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls check draw on a full board that is a draw<br><br><b>Function Name:</b><br>full_board_checkForDraw |
|   | 0 | 1 | 2 | 3 | 4 |   |  |
| 0   | x | x | o | o | x |   |  |
| 1   | o | o | x | x | o |   |  |
| 2   | x | x | o | o | x |   |  |
| 3   | o | o | x | x | o |   |  |
| 4   | x | x | o | o | x |   |  |

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| <b>Input:</b><br>State: (number to win = 3) |   |   |   |   |   |
|   | 0 | 1 | 2 | 3 | 4 |
| 0   | x | x | o | o | x |
| 1   | o | o | x | x | o |
| 2   | x | x | o | o | x |
| 3   | o | o | x | x | o |
| 4   | x | x | o | o |   |

 **Output:**   checkForDraw = false  State of board unchanged | **Reason:**  This test case is unique and distinct because it calls check draw on an almost full board   **Function Name:**  almost\_board\_checkForDraw |

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| <b>Input:</b><br>State: (number to win = 3) |   |   |   |   |   |
|   | 0 | 1 | 2 | 3 | 4 |
| 0   | x | x | o | o | x |
| 1   | o | o | x | x | o |
| 2   |   |   |   |   |   |
| 3   |   |   |   |   |   |
| 4   |   |   |   |   |   |

 **Output:**   checkForDraw = false  State of board unchanged | **Reason:**  This test case is unique and distinct because it calls check draw on a normal board   **Function Name:**  normal\_board\_checkForDraw |

**char whatsAtPos(BoardPosition pos)**

|  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
|--|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 2<br>lastPos.getColumn() = 0 |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  |  |  |  | 1 |  |  |  |  |  | 2 | x |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>whatsAtPos = x<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls whatsAtPos at a position on the left wall of the board<br><br><b>Function Name:</b><br>left_wall_whatsAtPos |
|  | 0 | 1 | 2 | 3 | 4 |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 0  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 1  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 2  | x |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 3  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 4  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |

|  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
|--|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|---|---|--|--|--|--|--|---|--|--|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td>x</td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 2<br>lastPos.getColumn() = 4 |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  |  |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  | x | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>whatsAtPos = x<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls whatsAtPos at a position on the right wall of the board<br><br><b>Function Name:</b><br>right_wall_whatsAtPos |
|  | 0 | 1 | 2 | 3 | 4 |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 0  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 1  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 2  |   |   |   |   | x |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 3  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 4  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |  |  |



|   |   |   |   |   |   |   |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|--|--|---|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table border="1"><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 2 |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  | x |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>whatsAtPos = x<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls whatsAtPos at a position on the top wall of the board<br><br><b>Function Name:</b><br>top_wall_whatsAtPos |
|   | 0 | 1 | 2 | 3 | 4 |   |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 0   |   |   | x |   |   |   |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 1   |   |   |   |   |   |   |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 2   |   |   |   |   |   |   |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 3   |   |   |   |   |   |   |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 4   |   |   |   |   |   |   |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |

|   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |  |
|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|---|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table border="1"><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td>x</td><td></td><td></td></tr></table><br>lastPos.getRow() = 4<br>lastPos.getColumn() = 2 |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  |  |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  | x |  |  | <b>Output:</b><br><br>whatsAtPos = x<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls whatsAtPos at a position on the bottom wall of the board<br><br><b>Function Name:</b><br>bottom_wall_whatsAtPos |
|   | 0 | 1 | 2 | 3 | 4 |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |  |
| 0   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |  |
| 1   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |  |
| 2   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |  |
| 3   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |  |
| 4   |   |   | x |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |  |

|  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
|--|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|---|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 2<br>lastPos.getColumn() = 2 |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  |  |  |  | 1 |  |  |  |  |  | 2 |  |  | x |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Output:</b><br><br>whatsAtPos = x<br>State of board unchanged | <b>Reason:</b><br><br>This test case is unique and distinct because it calls whatsAtPos at a position in the middle of the board<br><br><b>Function Name:</b><br>center_whatsAtPos |
|  | 0 | 1 | 2 | 3 | 4 |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 0  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 1  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 2  |   |   | x |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 3  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |
| 4  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |

**boolean isPlayerAtPos(BoardPosition pos, char player)**

|   |   |   |   |   |   |   |  |
|---|---|---|---|---|---|---|--|
| <b>Input:</b><br>State: (number to win = 3)                   |   |   |   |   |   | <b>Output:</b><br><br>isPlayerAtPos = false<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls isPlayerAtPos on a position with a different character in it<br><br><b>Function Name:</b><br>player_not_there_isPlayerAtPos<br>os |
|   | 0 | 1 | 2 | 3 | 4 |   |  |
| 0   | x |   |   |   |   |   |  |
| 1   |   |   |   |   |   |   |  |
| 2   |   |   |   |   |   |   |  |
| 3   |   |   |   |   |   |   |  |
| 4   |   |   |   |   |   |   |  |
| lastPos.getRow() = 0<br>lastPos.getColumn() = 0<br>player = o |   |   |   |   |   |   |  |

|   |   |   |   |   |   |  |   |
|---|---|---|---|---|---|--|---|
| <b>Input:</b><br>State: (number to win = 3)                   |   |   |   |   |   | <b>Output:</b><br><br>isPlayerAtPos = true<br>State of board unchanged | <b>Reason:</b><br>This test case is unique and distinct because it calls isPlayerAtPos on a position on the left wall of the board<br><br><b>Function Name:</b><br>player_there_left_wall_isPlayerAtPos |
|   | 0 | 1 | 2 | 3 | 4 |  |   |
| 0   |   |   |   |   |   |  |   |
| 1   |   |   |   |   |   |  |   |
| 2   | x |   |   |   |   |  |   |
| 3   |   |   |   |   |   |  |   |
| 4   |   |   |   |   |   |  |   |
| lastPos.getRow() = 2<br>lastPos.getColumn() = 0<br>player = x |   |   |   |   |   |  |   |

|   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| <div><div><div><b>Input:</b></div><div>State: (number to win = 3)</div><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td>x</td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><div>lastPos.getRow() = 2<br/>lastPos.getColumn() = 4<br/>player = x</div></div></div> |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  |  |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  | x | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <div><div><div><b>Output:</b></div><div>isPlayerAtPos = true<br/>State of board unchanged</div></div></div> | <div><div><div><b>Reason:</b></div><div>This test case is unique and distinct because it calls isPlayerAtPos on a position on the right wall of the board</div></div></div> <div><div><div><b>Function Name:</b></div><div>player_there_right_wall_isPlayerAtPos</div></div></div> |
|   | 0 | 1 | 2 | 3 | 4 |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 0   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 1   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 2   |   |   |   |   | x |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 3   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |
| 4   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |

|   |
|---|
| <div><div><div><div><div><div></div><div>0</div><div>1</div><div>2</div><div>3</div><div>4</div></div><div><div>0</div><div></div><div></div><div>x</div><div></div><div></div></div><div><div>1</div><div></div><div></div><div></div><div></div><div></div></div><div><div>2</div><div></div><div></div><div></div><div></div><div></div></div><div><div>3</div><div></div><div></div><div></div><div></div><div></div></div><div><div>4</div><div></div><div></div><div></div><div></div><div></div></div></div></div></div><div><div>lastPos.getRow() = 0</div><div>lastPos.getColumn() = 2</div><div>player = x</div></div></div> <div><div><div>Output:</div><div><div>isPlayerAtPos = true</div><div>State of board unchanged</div></div></div></div> <div><div><div>Reason:</div><div><div>This test case is unique and distinct because it calls isPlayerAtPos on a position on the top wall of the board</div></div></div><div><div><div>Function Name:</div><div>player_there_top_wall_isPlayerAtPos</div></div></div></div> |
|---|

|  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |   |
|--|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|---|--|--|--|---|
| <b>Input:</b><br>State: (number to win = 3)<br><table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td>x</td><td></td><td></td></tr></table><br>lastPos.getRow() = 4<br>lastPos.getColumn() = 2<br>player = x |   | 0 | 1 | 2 | 3 | 4 | 0 |  |  |  |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  | x |  |  | <b>Output:</b><br><br>isPlayerAtPos = true<br>State of board unchanged | <b>Reason:</b><br><br>This test case is unique and distinct because it calls isPlayerAtPos on a position on the bottom wall of the board<br><br><b>Function Name:</b><br>player_there_bottom_wall_isPlayerAtPos |
|  | 0 | 1 | 2 | 3 | 4 |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |   |
| 0  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |   |
| 1  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |   |
| 2  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |   |
| 3  |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |   |
| 4  |   |   | x |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |   |  |  |  |   |

**void placeMarker(BoardPosition marker, char player)**

|  |                |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
|--|----------------|--|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|---|---|---|---|---|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)  | <b>Output:</b> | <b>Reason:</b><br>This test case is unique and distinct because it places a marker on an empty board |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| <table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 0<br>lastPos.getColumn() = 0<br>player = x |                | 0  | 1 | 2 | 3 | 4 | 0 |  |  |  |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table> |  | 0 | 1 | 2 | 3 | 4 | 0 | x |  |  |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Function Name:</b><br>empty_board_placeMarker |
|  | 0              | 1  | 2 | 3 | 4 |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 0  |                |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 1  |                |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 2  |                |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 3  |                |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 4  |                |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
|  | 0              | 1  | 2 | 3 | 4 |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 0  | x              |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 1  |                |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 2  |                |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 3  |                |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 4  |                |  |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |  |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |

|  |                |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
|--|----------------|---|---|---|---|---|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|---|---|---|---|---|---|---|---|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)  | <b>Output:</b> | <b>Reason:</b><br>This test case is unique and distinct because it places a marker representing a character that has not been used before |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| <table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table> |                | 0   | 1 | 2 | 3 | 4 | 0 | x |  |  |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td>o</td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table> |  | 0 | 1 | 2 | 3 | 4 | 0 | x | o |  |  |  | 1 |  |  |  |  |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Function Name:</b><br>new_character_placeMarker |
|  | 0              | 1   | 2 | 3 | 4 |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 0  | x              |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 1  |                |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 2  |                |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 3  |                |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 4  |                |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
|  | 0              | 1   | 2 | 3 | 4 |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 0  | x              | o   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 1  |                |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 2  |                |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 3  |                |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| 4  |                |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |
| lastPos.getRow() = 0<br>lastPos.getColumn() = 1<br>player = o  |                |   |   |   |   |   |   |   |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |  |

|   |   |   |   |   |   |                |   |   |   |   |   |  |  |  |  |  |  |
|---|---|---|---|---|---|----------------|---|---|---|---|---|--|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)                   |   |   |   |   |   | <b>Output:</b> |   |   |   |   |   | <b>Reason:</b><br>This test case is unique and distinct because it places a marker representing a character that has been used before<br><br><b>Function Name:</b><br>same_character_placeMarker |  |  |  |  |  |
|   | 0 | 1 | 2 | 3 | 4 |                | 0 | 1 | 2 | 3 | 4 |  |  |  |  |  |  |
| 0   | x | o |   |   |   | 0              | x | o | o |   |   |  |  |  |  |  |  |
| 1   |   |   |   |   |   | 1              |   |   |   |   |   |  |  |  |  |  |  |
| 2   |   |   |   |   |   | 2              |   |   |   |   |   |  |  |  |  |  |  |
| 3   |   |   |   |   |   | 3              |   |   |   |   |   |  |  |  |  |  |  |
| 4   |   |   |   |   |   | 4              |   |   |   |   |   |  |  |  |  |  |  |
| lastPos.getRow() = 0<br>lastPos.getColumn() = 2<br>player = o |   |   |   |   |   |                |   |   |   |   |   |  |  |  |  |  |  |

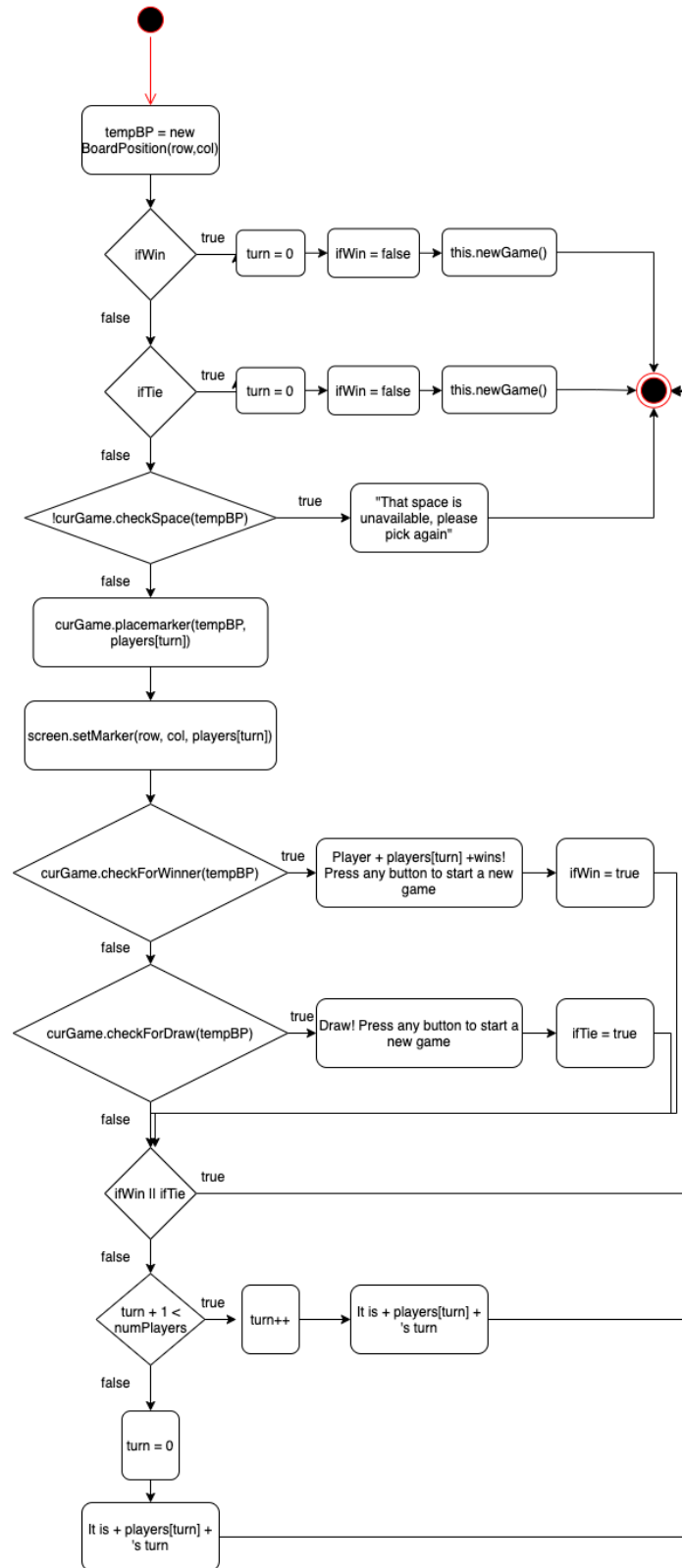
|   |   |   |   |   |   |                |   |   |   |   |   |   |  |  |  |  |  |
|---|---|---|---|---|---|----------------|---|---|---|---|---|---|--|--|--|--|--|
| <b>Input:</b><br>State: (number to win = 3)                   |   |   |   |   |   | <b>Output:</b> |   |   |   |   |   | <b>Reason:</b><br>This test case is unique and distinct because it places a marker in the last spot on the board<br><br><b>Function Name:</b><br>full_board_placeMarker |  |  |  |  |  |
|   | 0 | 1 | 2 | 3 | 4 |                | 0 | 1 | 2 | 3 | 4 |   |  |  |  |  |  |
| 0   | x | x | o | o | x | 0              | x | x | o | o | x |   |  |  |  |  |  |
| 1   | o | o | x | x | o | 1              | o | o | x | x | o |   |  |  |  |  |  |
| 2   | x | x | o | o | x | 2              | x | x | o | o | x |   |  |  |  |  |  |
| 3   | o | o | x | x | o | 3              | o | o | x | x | o |   |  |  |  |  |  |
| 4   | x | x | o | o |   | 4              | x | x | o | o | x |   |  |  |  |  |  |
| lastPos.getRow() = 4<br>lastPos.getColumn() = 4<br>player = x |   |   |   |   |   |                |   |   |   |   |   |   |  |  |  |  |  |

|   |                |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
|---|----------------|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|
| <b>Input:</b><br>State: (number to win = 3)   | <b>Output:</b> | <b>Reason:</b><br>This test case is unique and distinct because it places a marker on a normal board |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| <table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td>x</td><td>o</td><td>o</td><td>x</td></tr><tr><td>1</td><td>o</td><td>o</td><td>x</td><td>x</td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table><br>lastPos.getRow() = 1<br>lastPos.getColumn() = 4<br>player = o |                | 0  | 1 | 2 | 3 | 4 | 0 | x | x | o | o | x | 1 | o | o | x | x |  | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>x</td><td>x</td><td>o</td><td>o</td><td>x</td></tr><tr><td>1</td><td>o</td><td>o</td><td>x</td><td>x</td><td>o</td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table> |  | 0 | 1 | 2 | 3 | 4 | 0 | x | x | o | o | x | 1 | o | o | x | x | o | 2 |  |  |  |  |  | 3 |  |  |  |  |  | 4 |  |  |  |  |  | <b>Function Name:</b><br>normal_board_placeMarker |
|   | 0              | 1  | 2 | 3 | 4 |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 0   | x              | x  | o | o | x |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 1   | o              | o  | x | x |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 2   |                |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 3   |                |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 4   |                |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
|   | 0              | 1  | 2 | 3 | 4 |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 0   | x              | x  | o | o | x |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 1   | o              | o  | x | x | o |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 2   |                |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 3   |                |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |
| 4   |                |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |  |  |  |  |  |   |



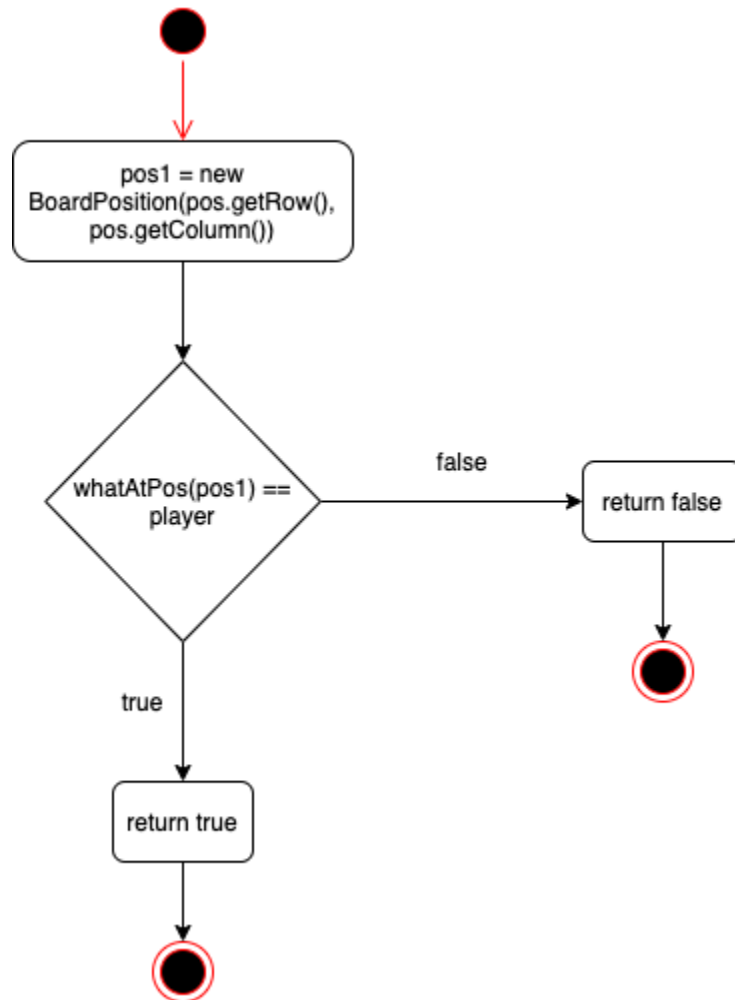
# TicTacToeController Activity Diagram

public void processButtonClick(int row, int col)

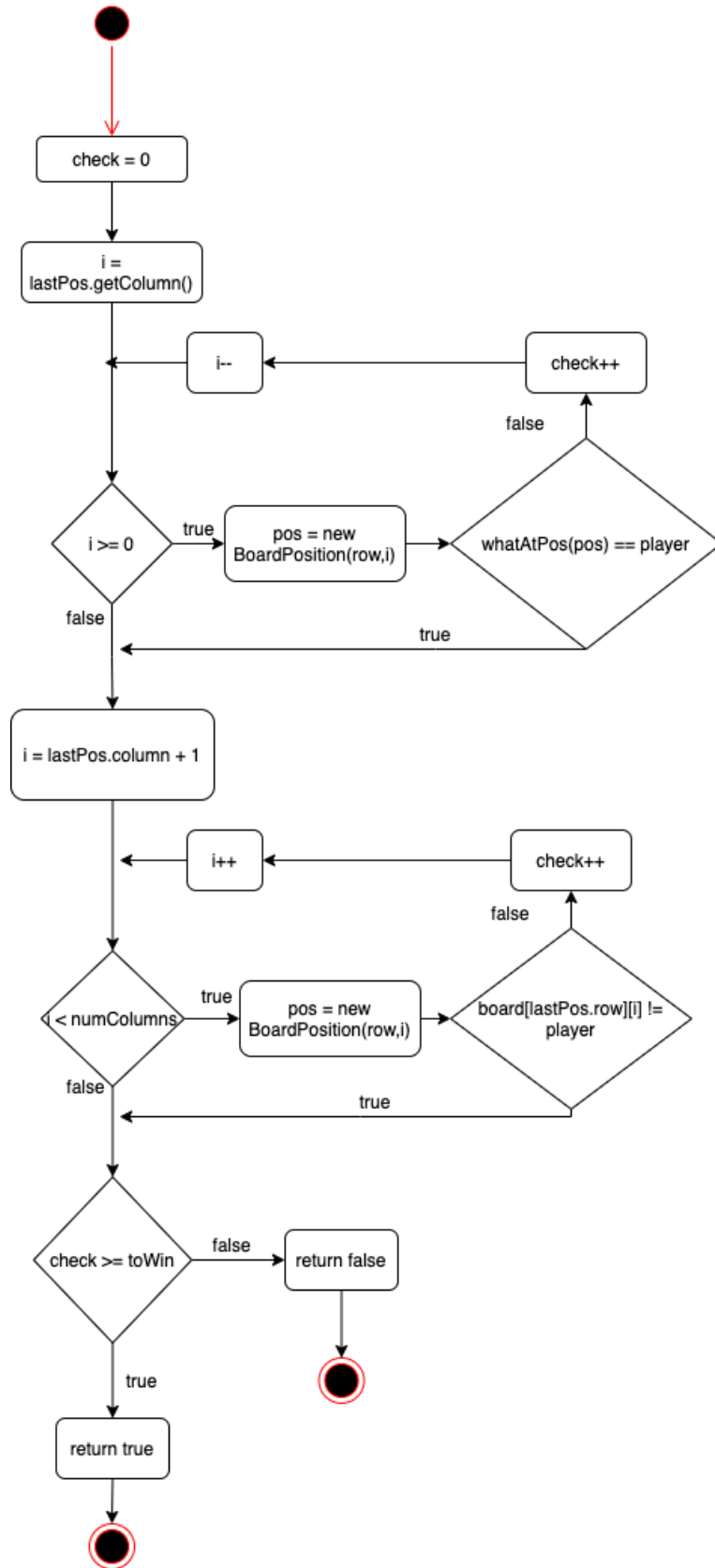


### IGameBoard Default Activity Diagrams

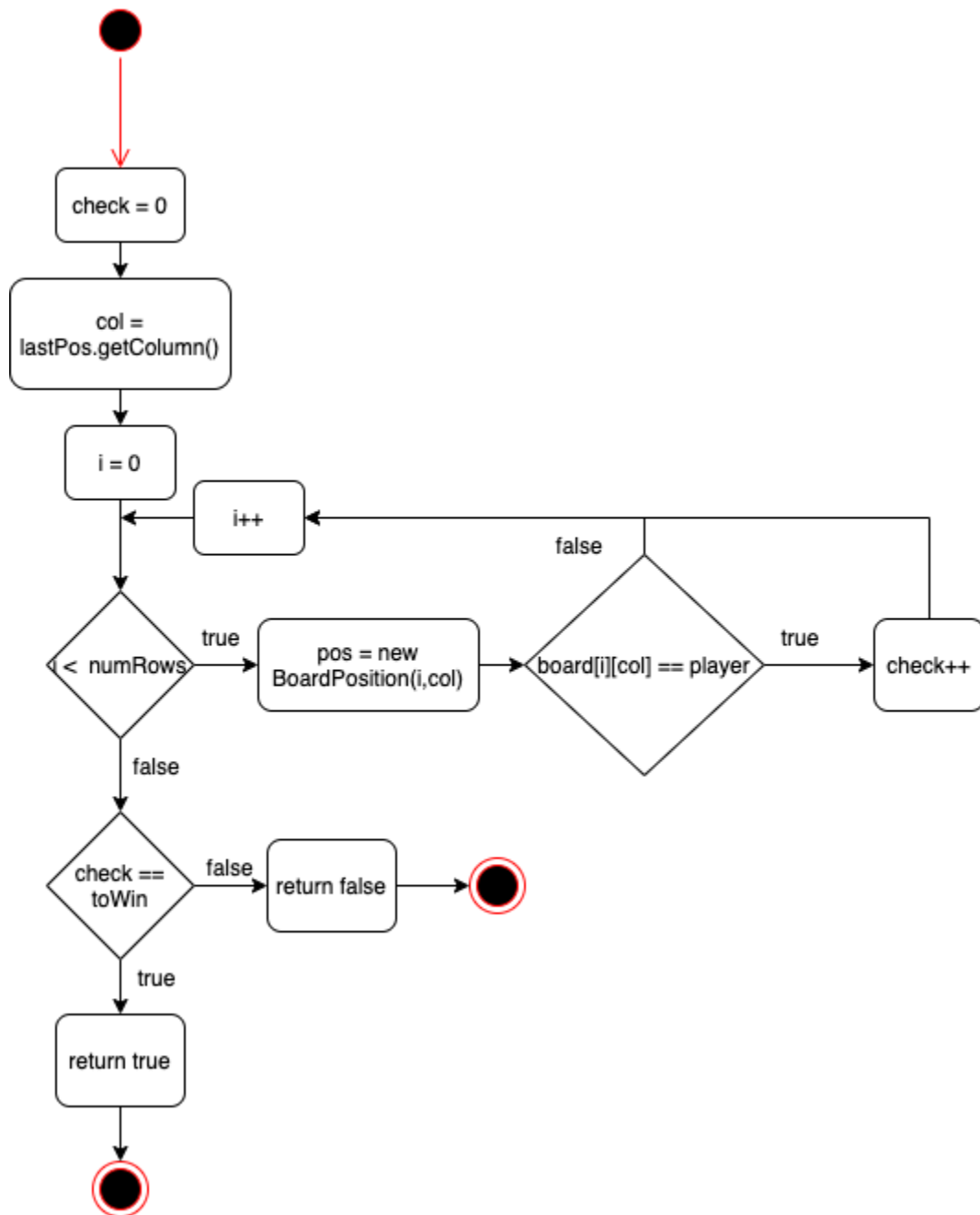
public boolean isPlayerAtPos(BoardPosition pos, char player)



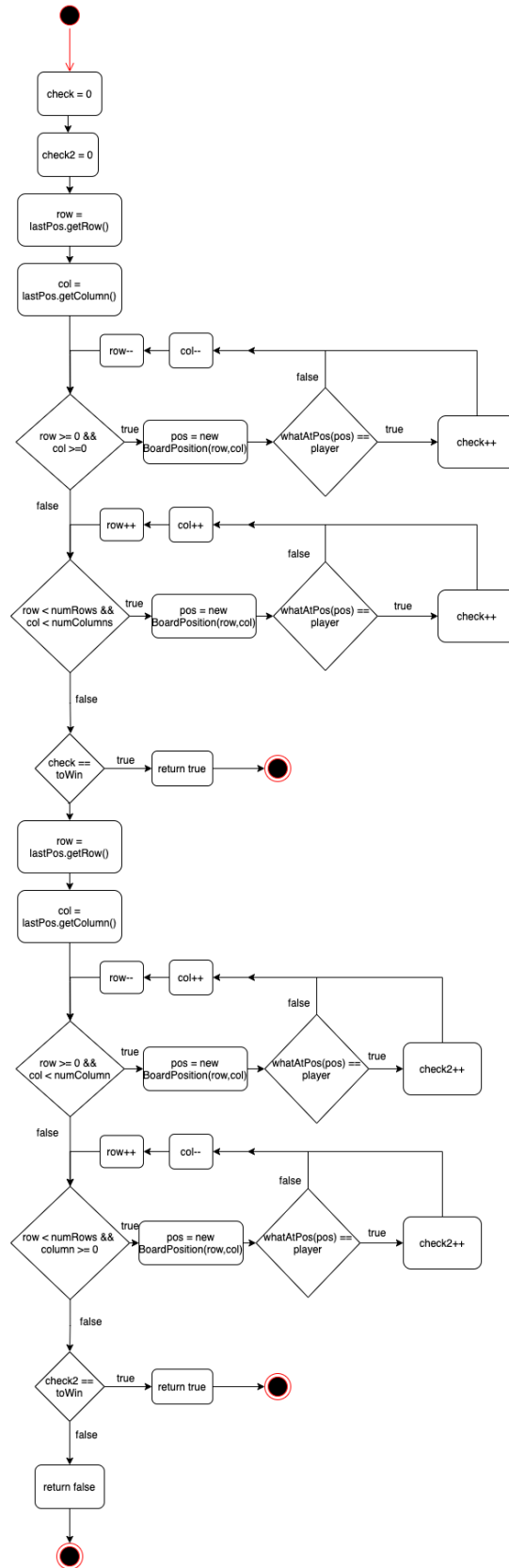
```
public boolean checkHorizontalWin(BoardPosition lastPos, char player)
```



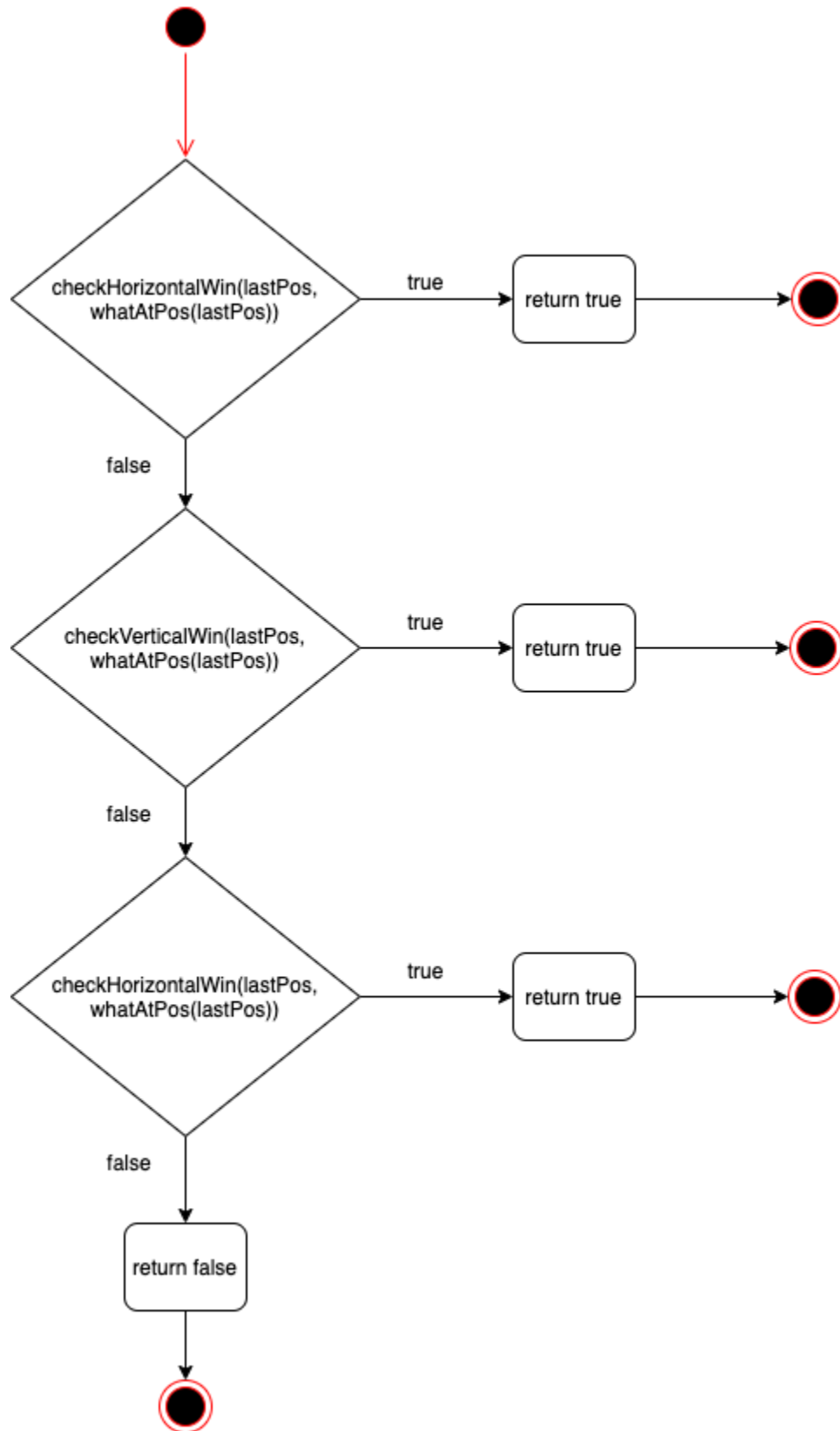
public boolean checkVerticalWin(BoardPosition lastPos, char player)



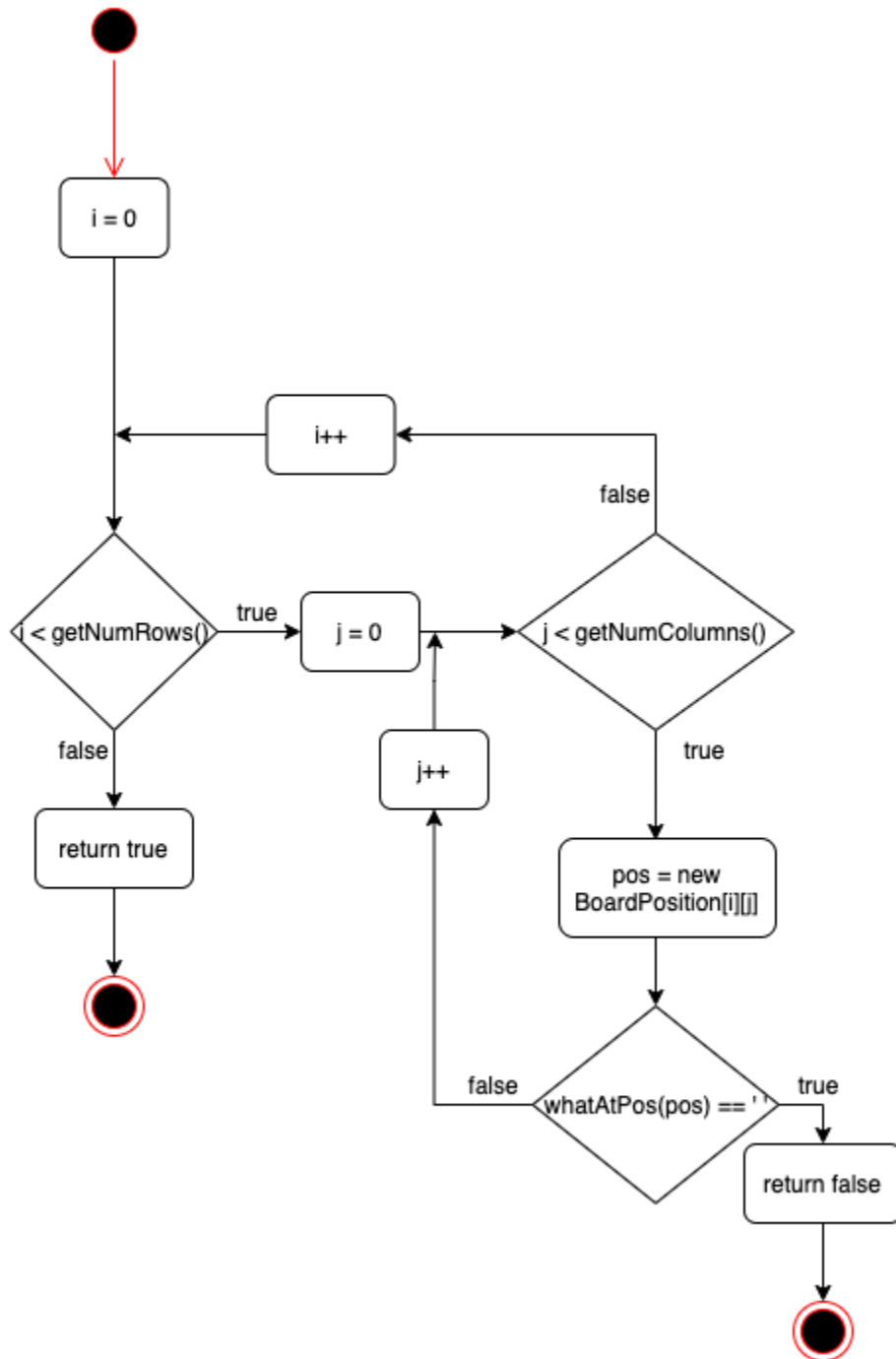
```
public boolean checkDiagonalWin(BoardPosition lastPos, char player)
```



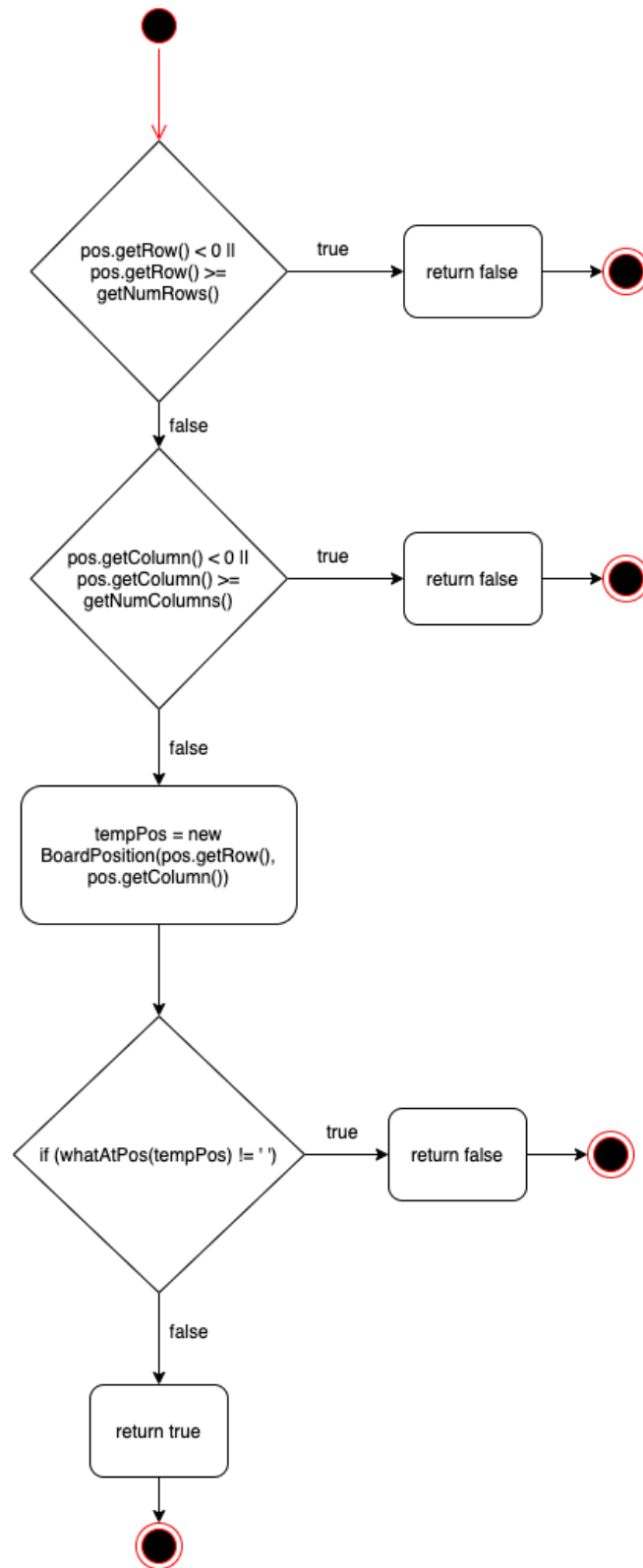
public boolean checkForWinner(BoardPosition lastPos)



public boolean checkForDraw()

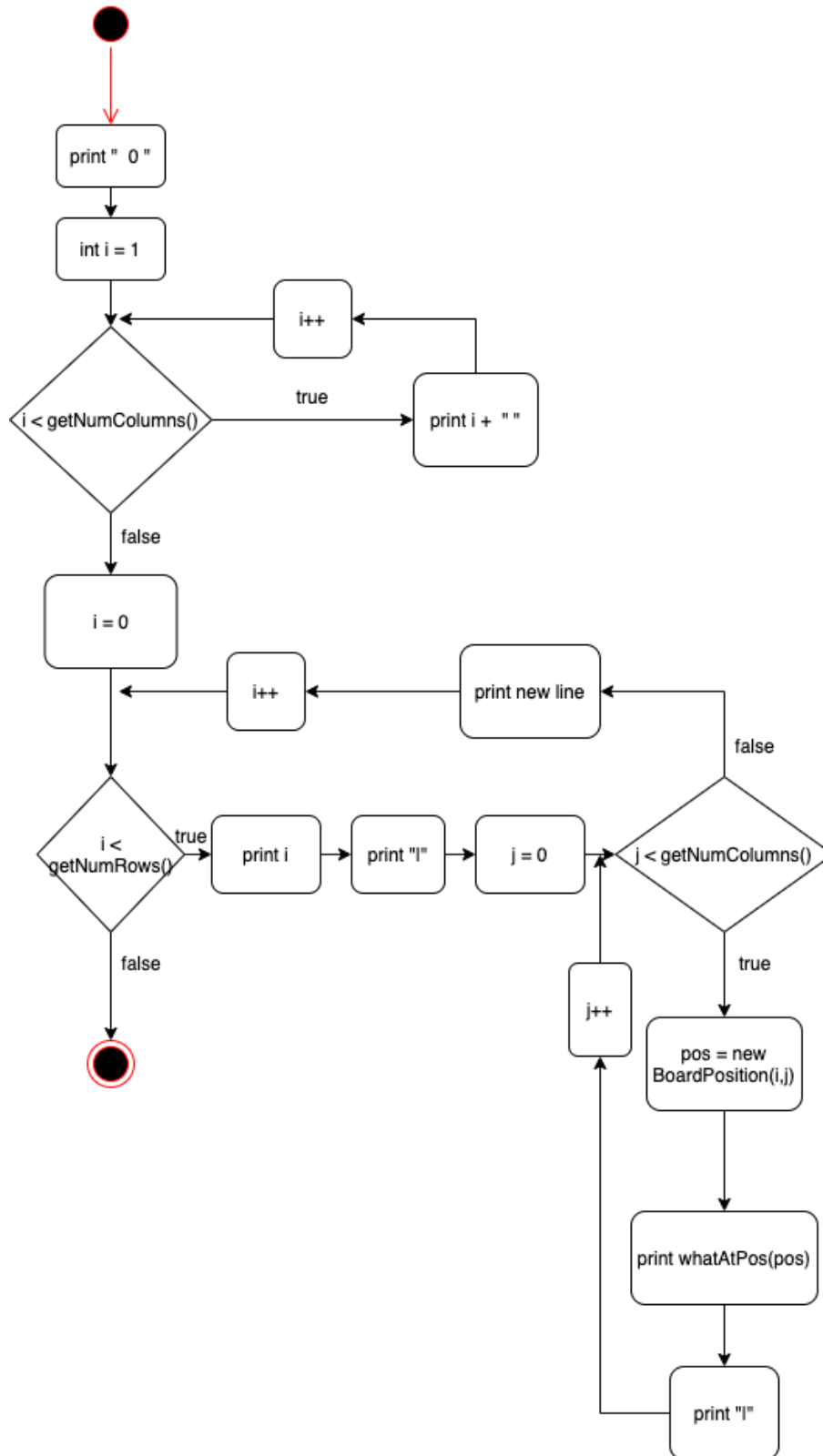


public boolean checkSpace(BoardPosition pos)



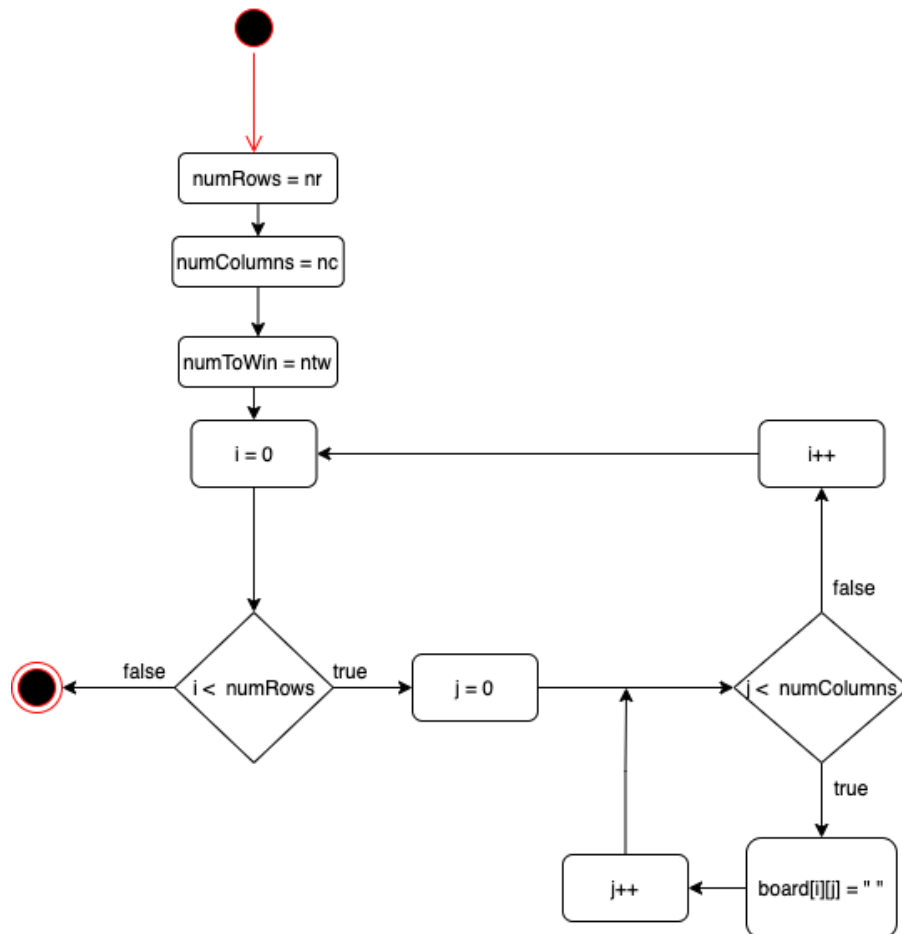


AbsGameBoard Activity Diagram  
public String toString()

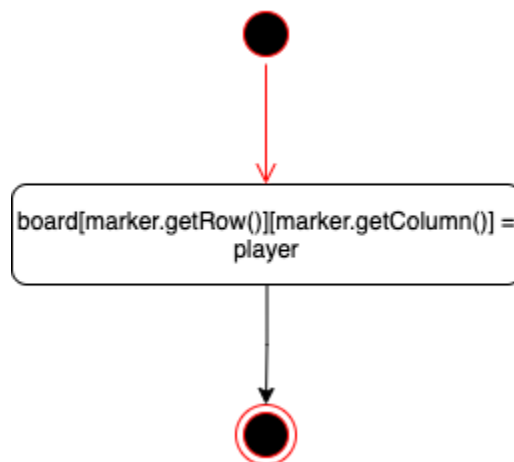


GameBoard(extends AbsGameBoard and implements IGameBoard) Activity Diagrams

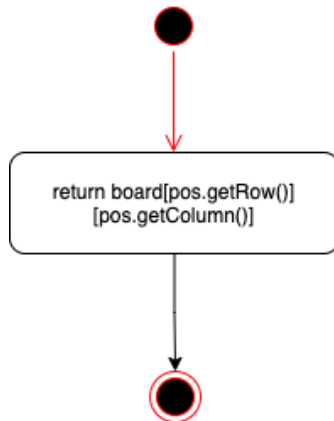
public GameBoard(int nr, int nc, int ntw)



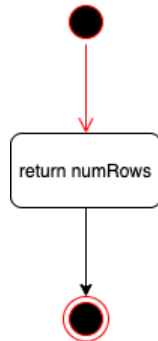
public void placeMarker(BoardPosition marker, char player)



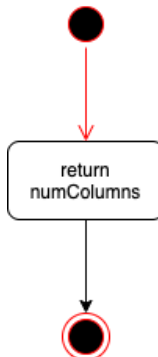
public char whatsAtPos(BoardPosition pos)



public getNumRows()



public getNumColumns()

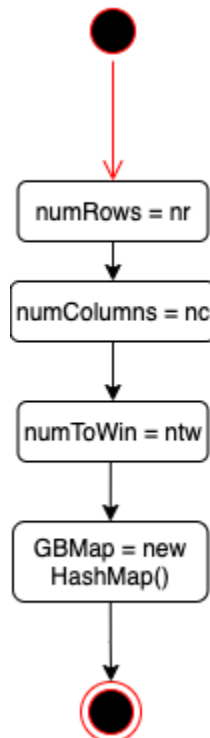


public getNumToWin()

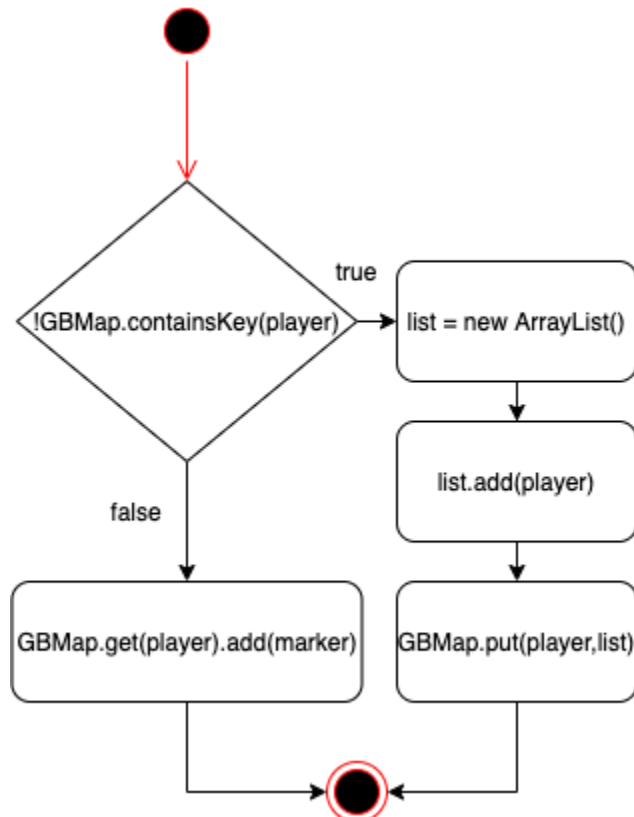


GameBoardMem(extends AbsGameBoard and implements IGameBoard) Activity Diagrams

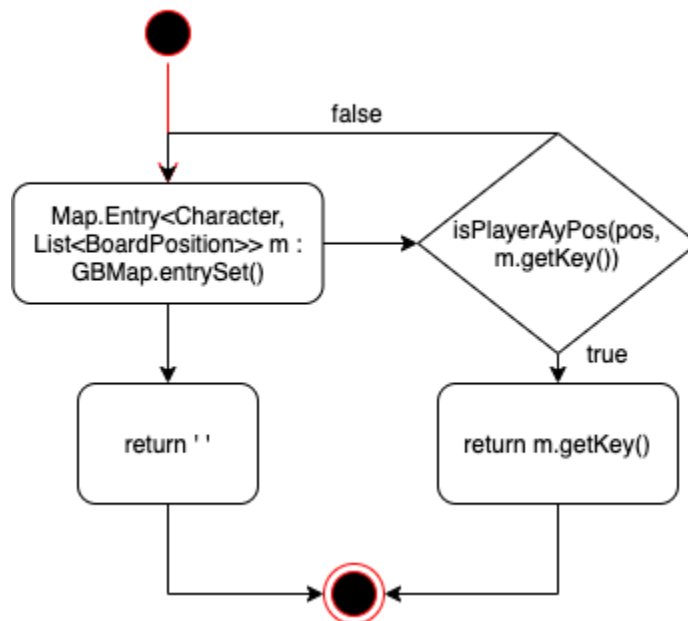
public GameBoardMem(int nr, int nc, int ntw)



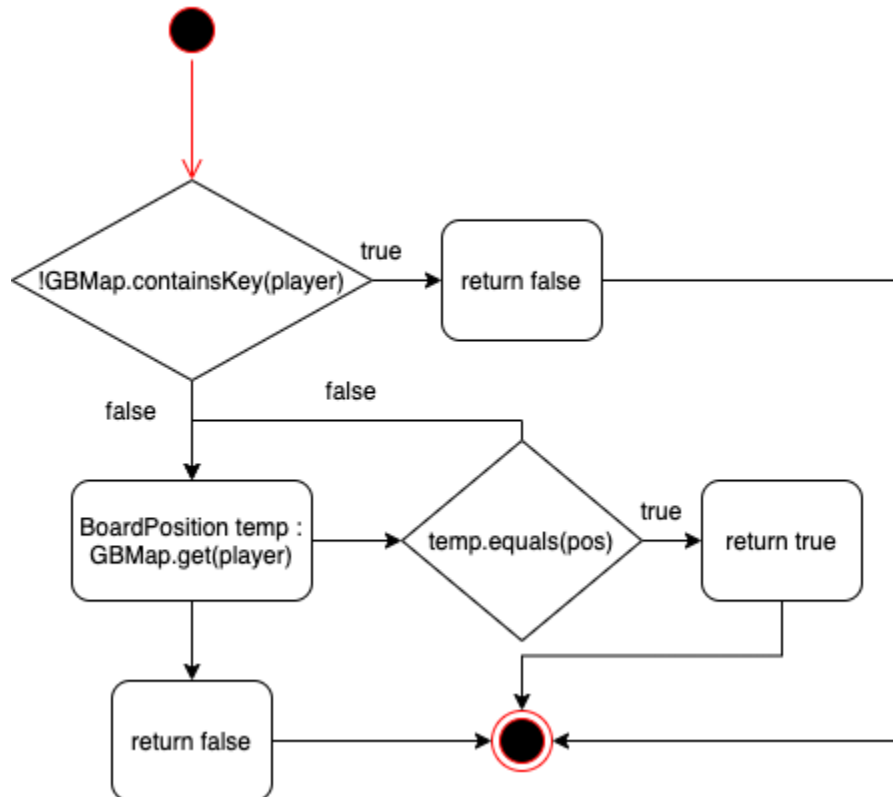
public void placeMarker(BoardPosition marker, char player)



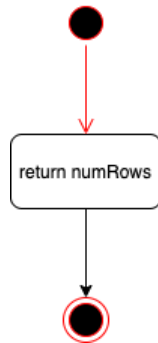
public char whatsAtPos(BoardPosition pos)



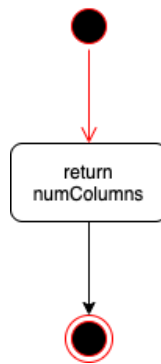
public boolean isPlayerAtPos(BoardPosition pos, char player)



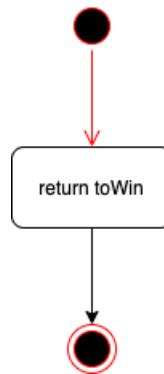
public getNumRows()



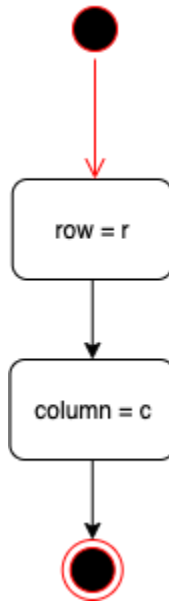
public getNumColumns()



public getNumToWin()



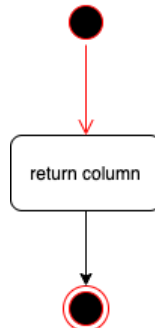
BoardPosition Activity Diagrams  
public BoardPosition(int r, int c)



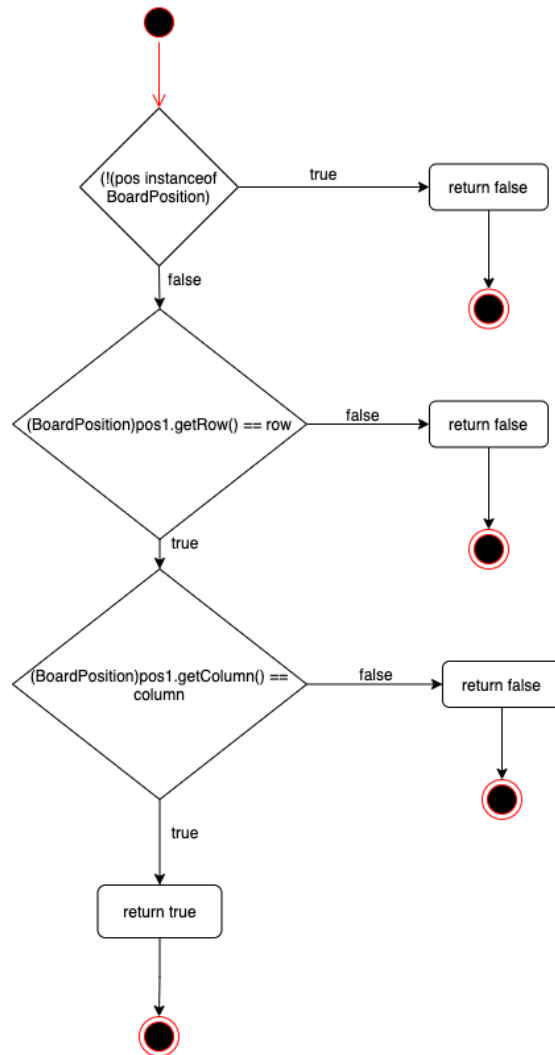
public int getRow()



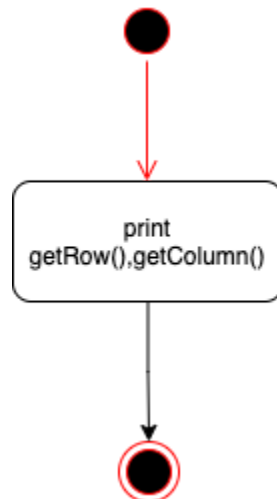
public int getColumn()



public boolean equals (BoardPosition pos1)



public String toString()





## GameScreen Activity Diagrams (sorry, please zoom in)

