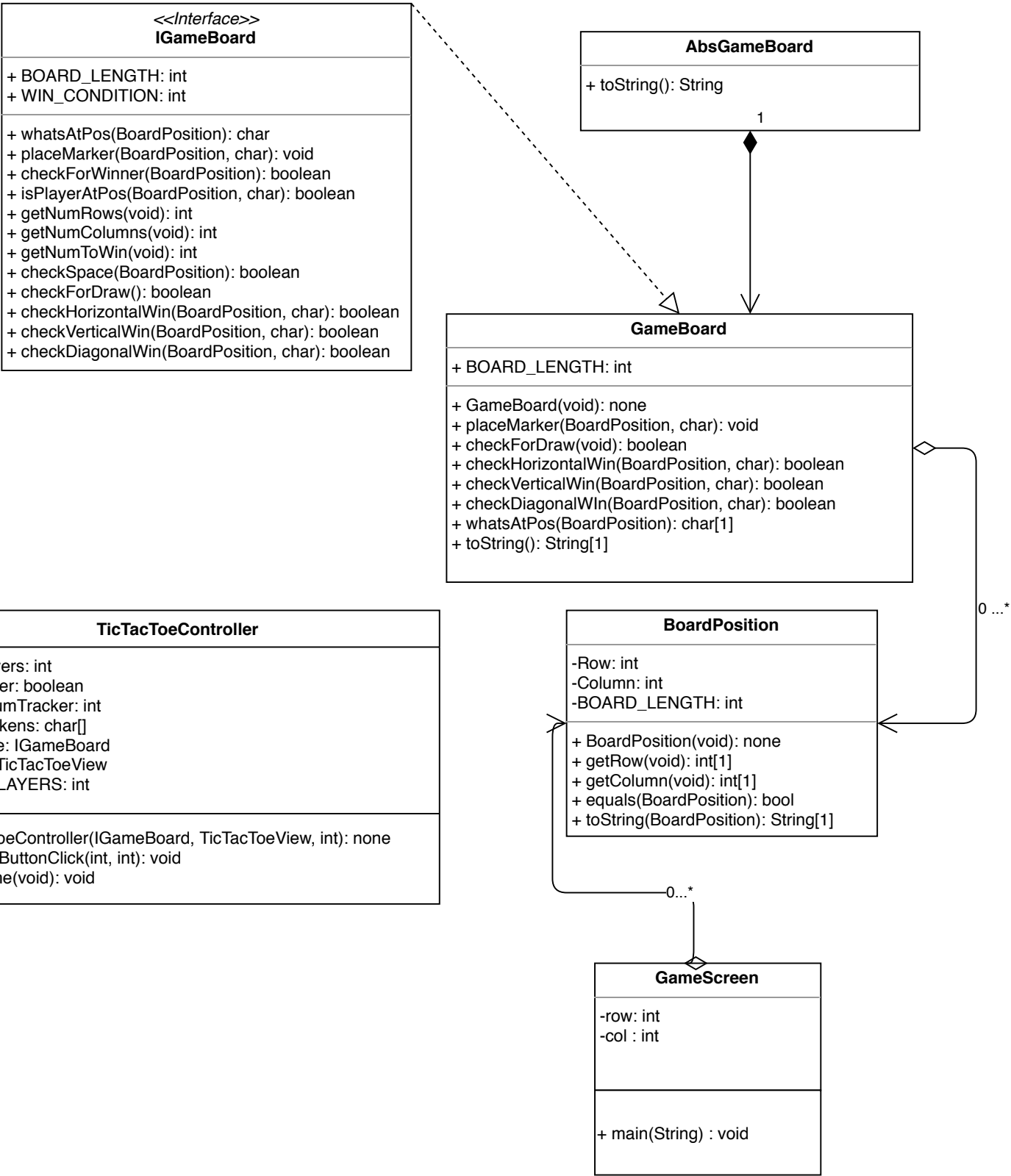
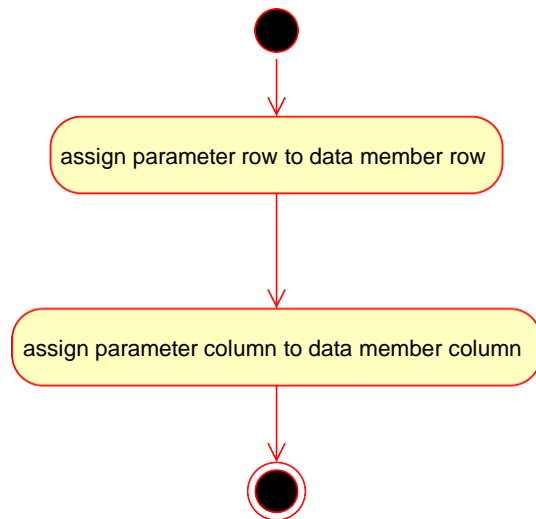


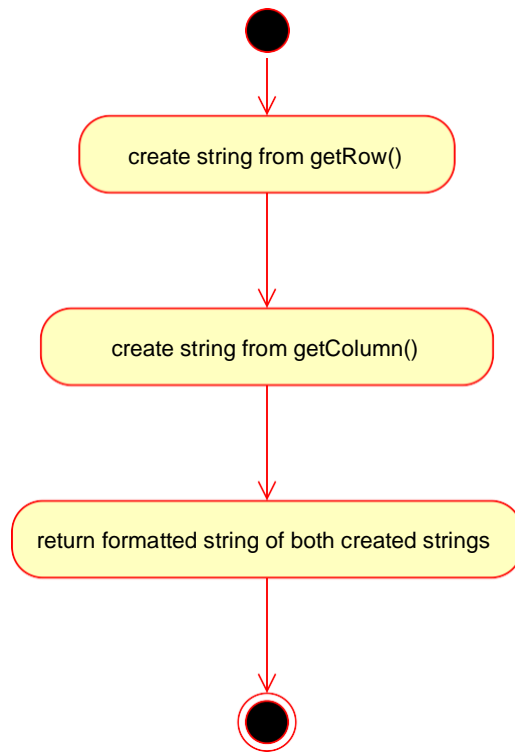
- Stage 1:
 - Functional Requirement:
 - As a player, I can view the game board to play the game
 - As a player, I can select which character I want to be my character tag
 - As a player, I can select locations on the board to place my character tag
 - As a player, I can choose whether or not to play another game to initiate another game
 - As a player, I can play an extended tic tac toe, to have fun
 - As a player, I Can win the game by aligning a selected number of my characters in a row, diagonally, horizontally, or vertically
 - As a player, I can tie my opponent by filling in all locations of the game board
 - The system will validate all user input
 - As the game progresses, the system will prompt all players sequentially
 - As a player I can specify the number of rows on the game board
 - As a player I can specify the number of columns on the game board
 - As a player I can specify the number of character tags placed in a row to win
 - As a player, I can specify whether I want a fast implementation or one that uses less memory
 - Non-Functional requirements:
 - The system has to be written in java
 - The system has to run on Unix
 - The system must be created according to the project direction files
 - The game must begin, letting Player 1 take a turn first, and all players taking turns sequentially afterwards
 - On the game board 0,0 must be the position on the top left
 - The game board must have a size specified by the user
 - The system must have two implementation types



BoardPosition

BoardPosition(int row, int column)

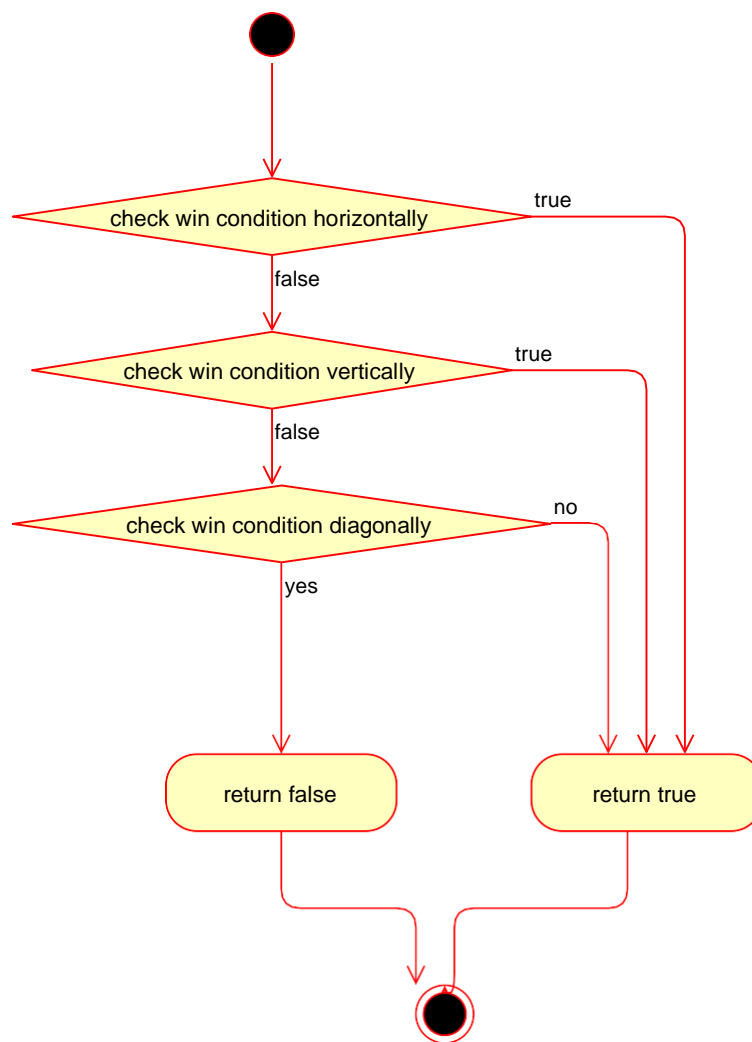




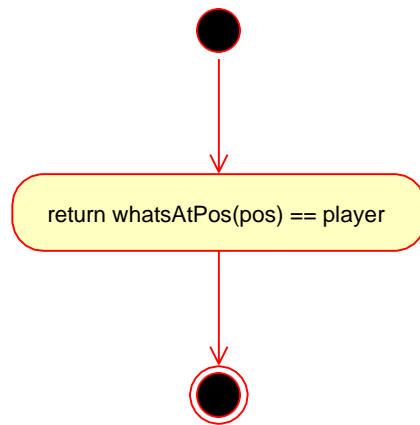
public String toString()

IGameBoard

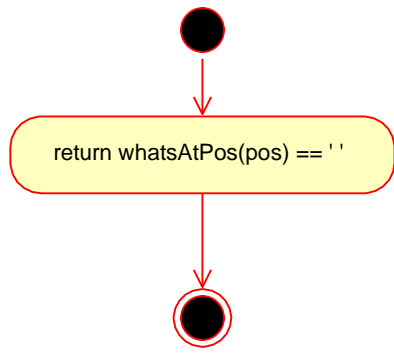
default boolean checkForWinner(BoardPosition lastPos)



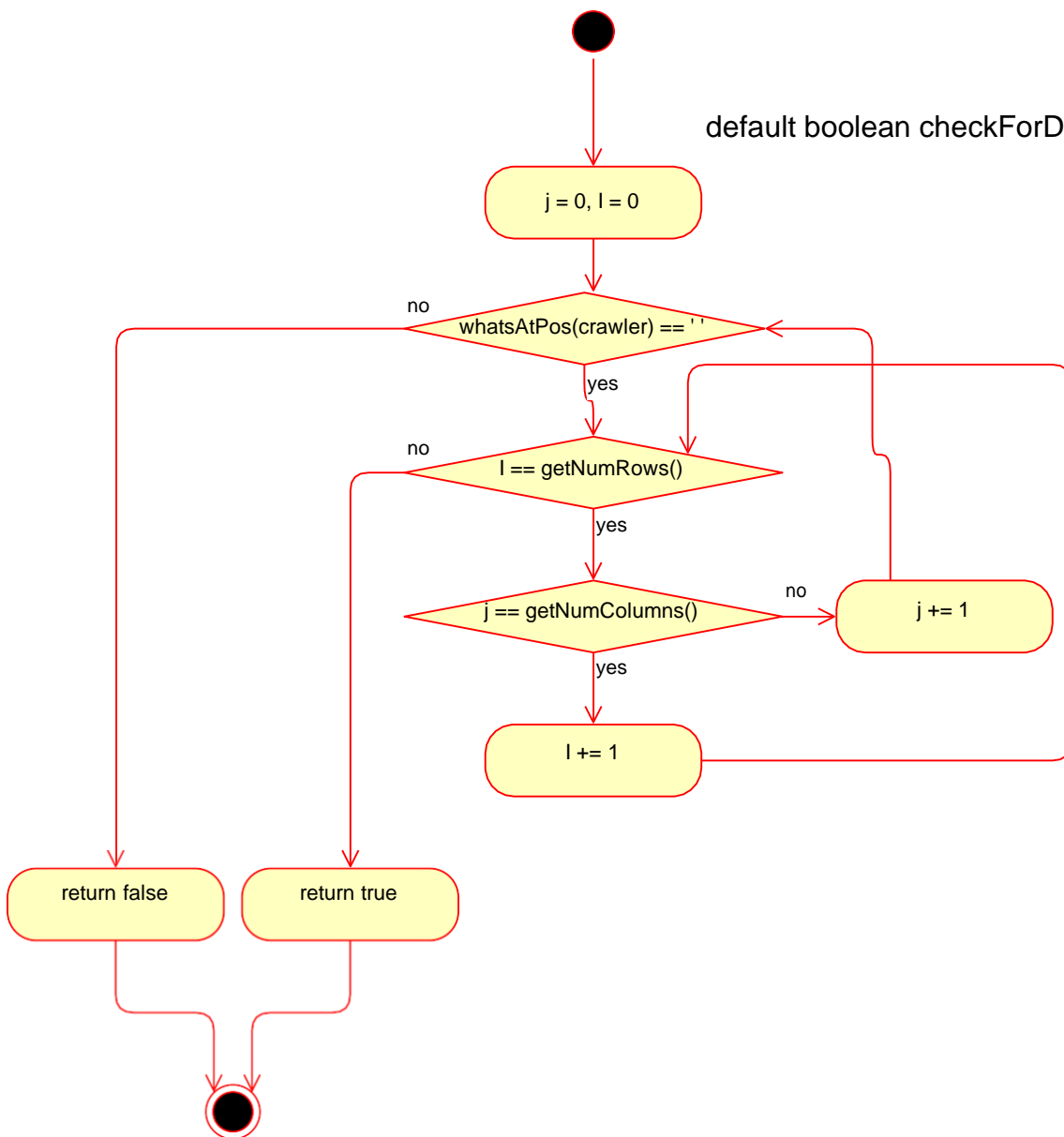
default boolean isPlayerAtPos(BoardPosition pos, char player)



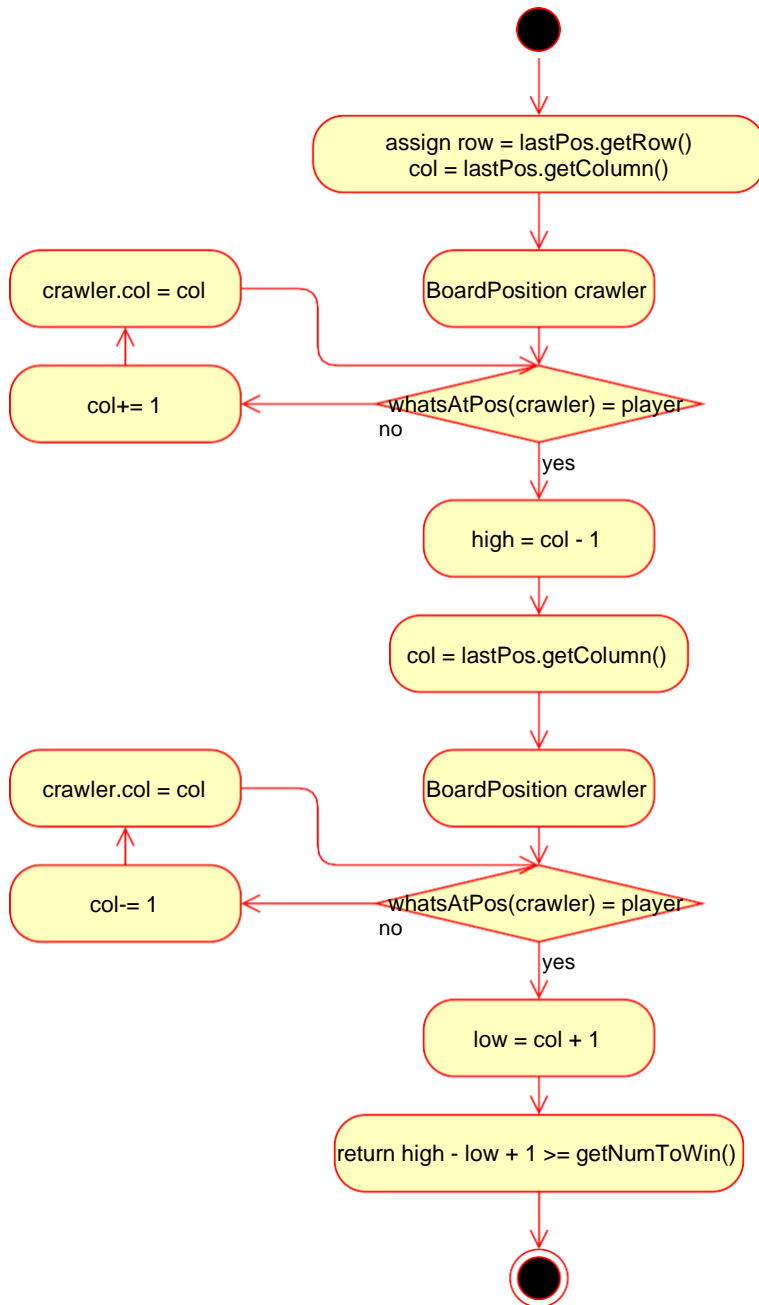
default boolean checkSpace(BoardPosition pos)



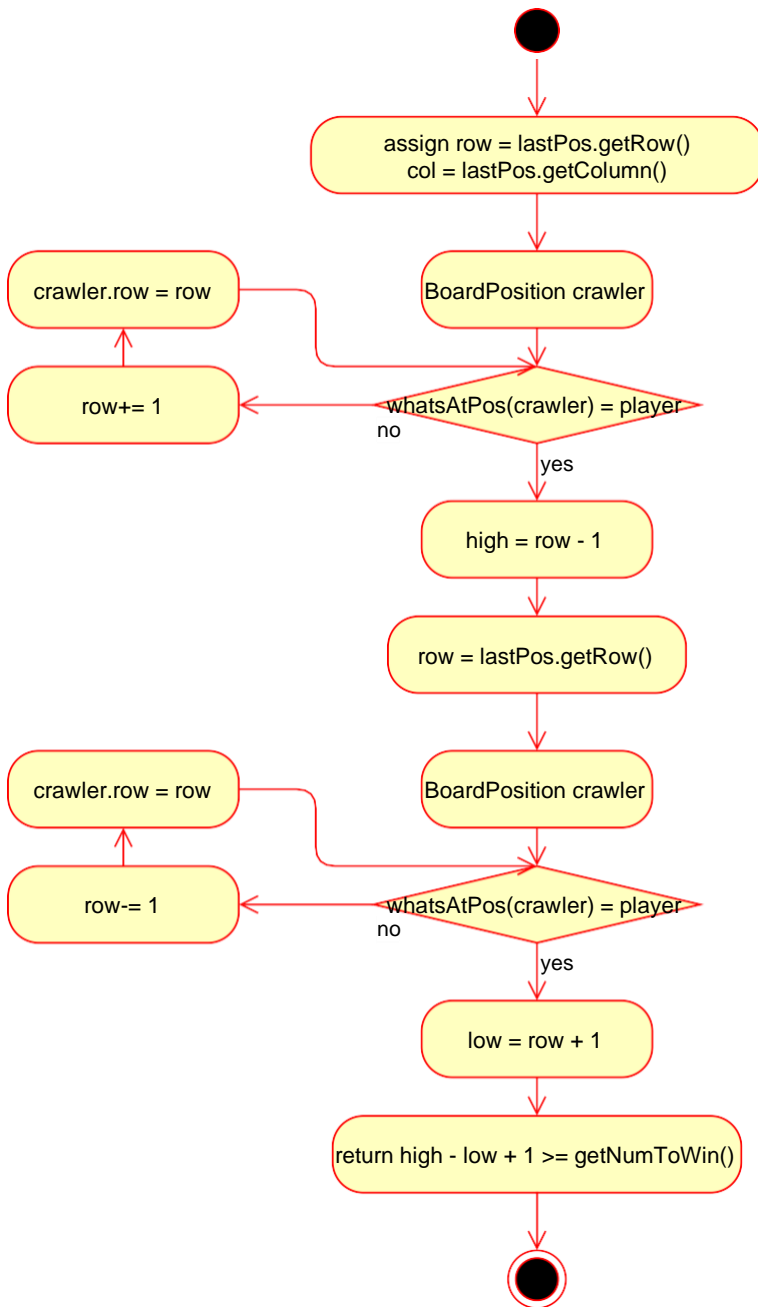
default boolean checkForDraw()

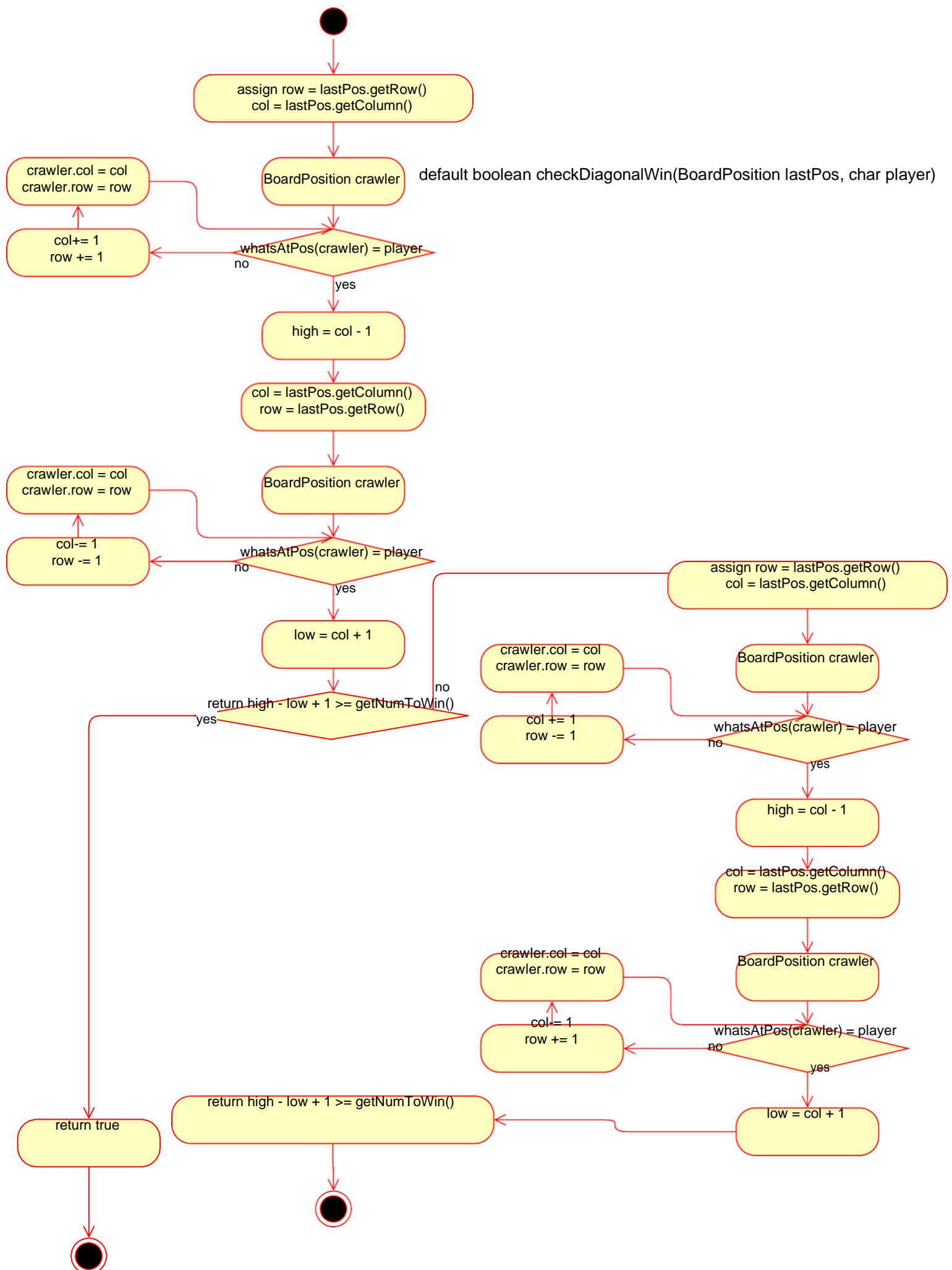


default boolean checkHorizontalWin(BoardPosition lastPos, char player)



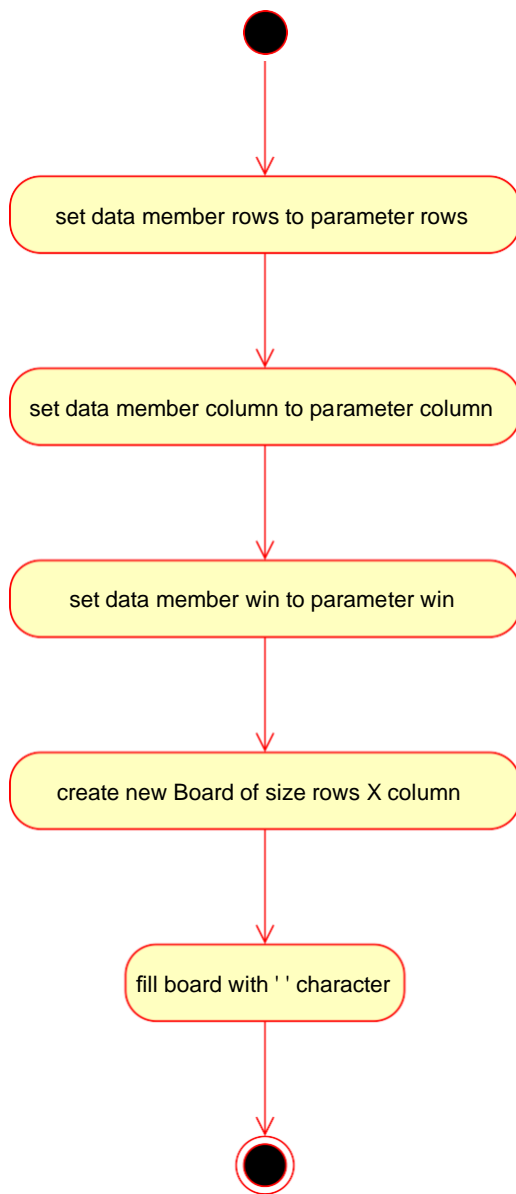
default boolean checkVerticalWin(BoardPosition lastPos, char player)



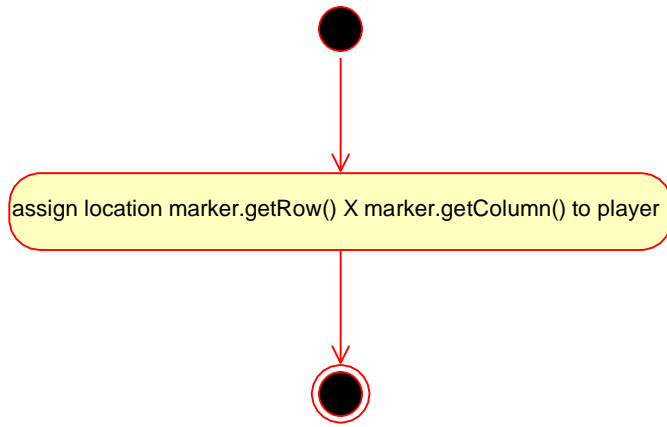


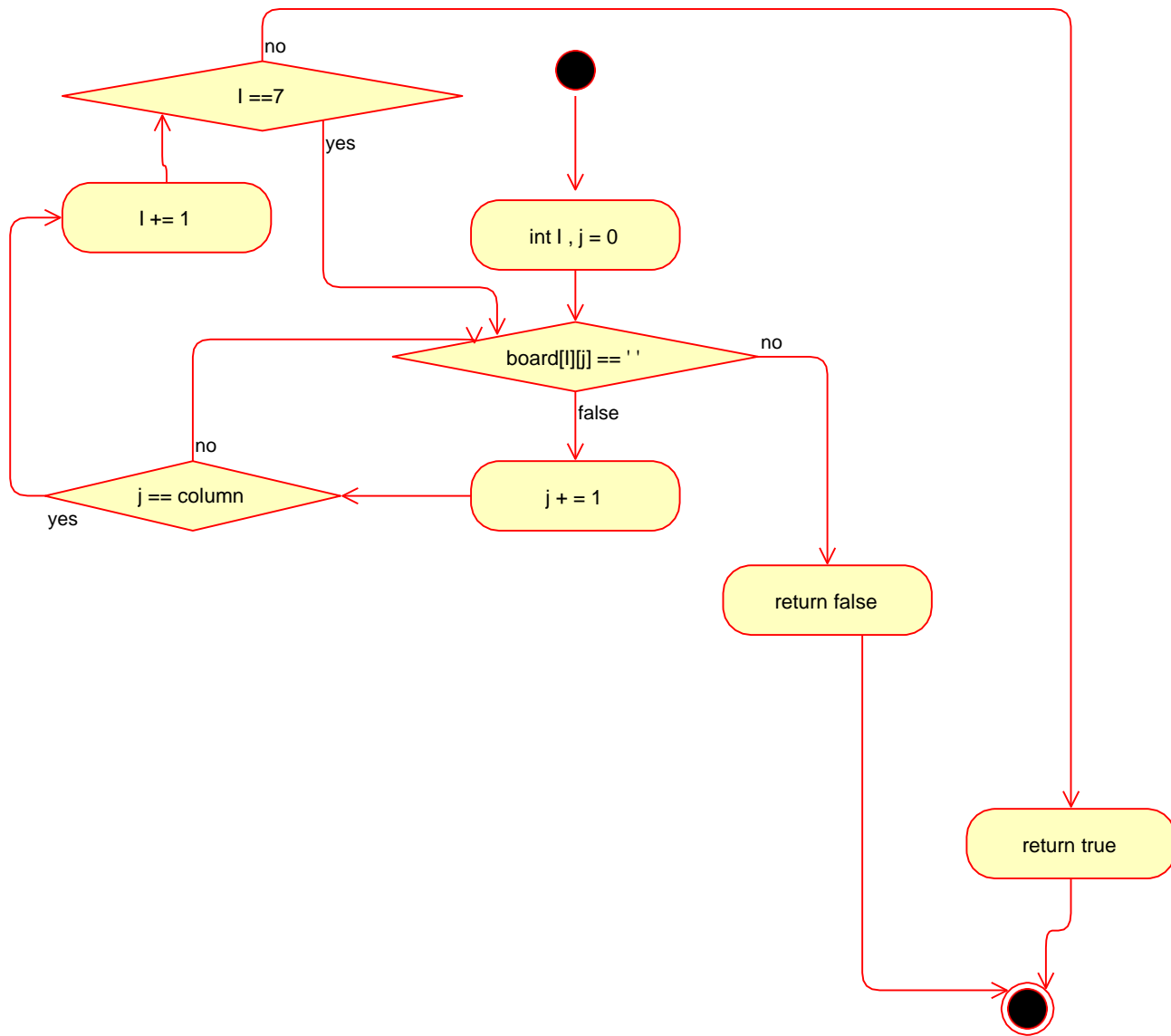
GameBoard

GameBoard(int rows, int column, int win)



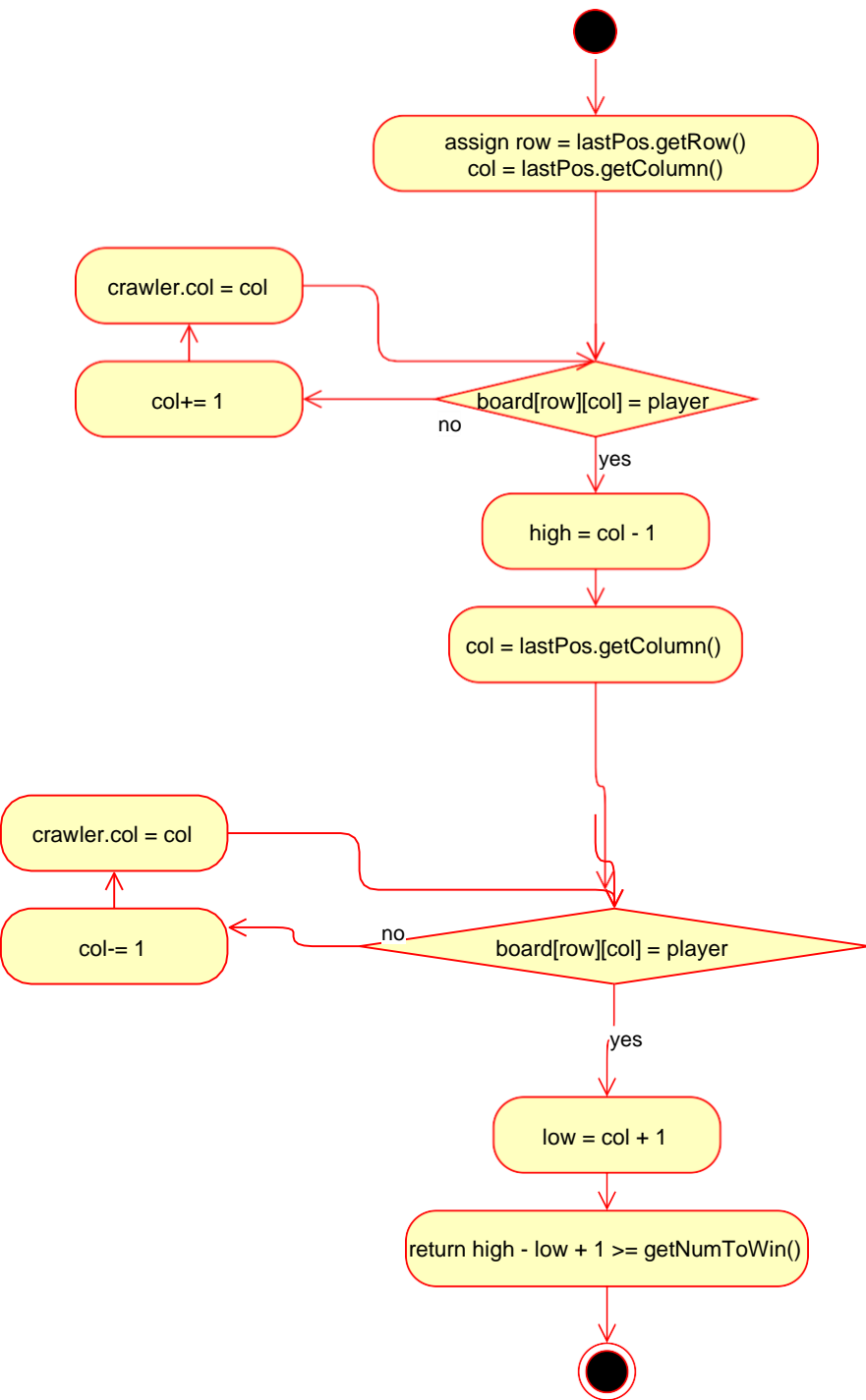
```
public void placeMarker(BoardPosition marker, char player)
```



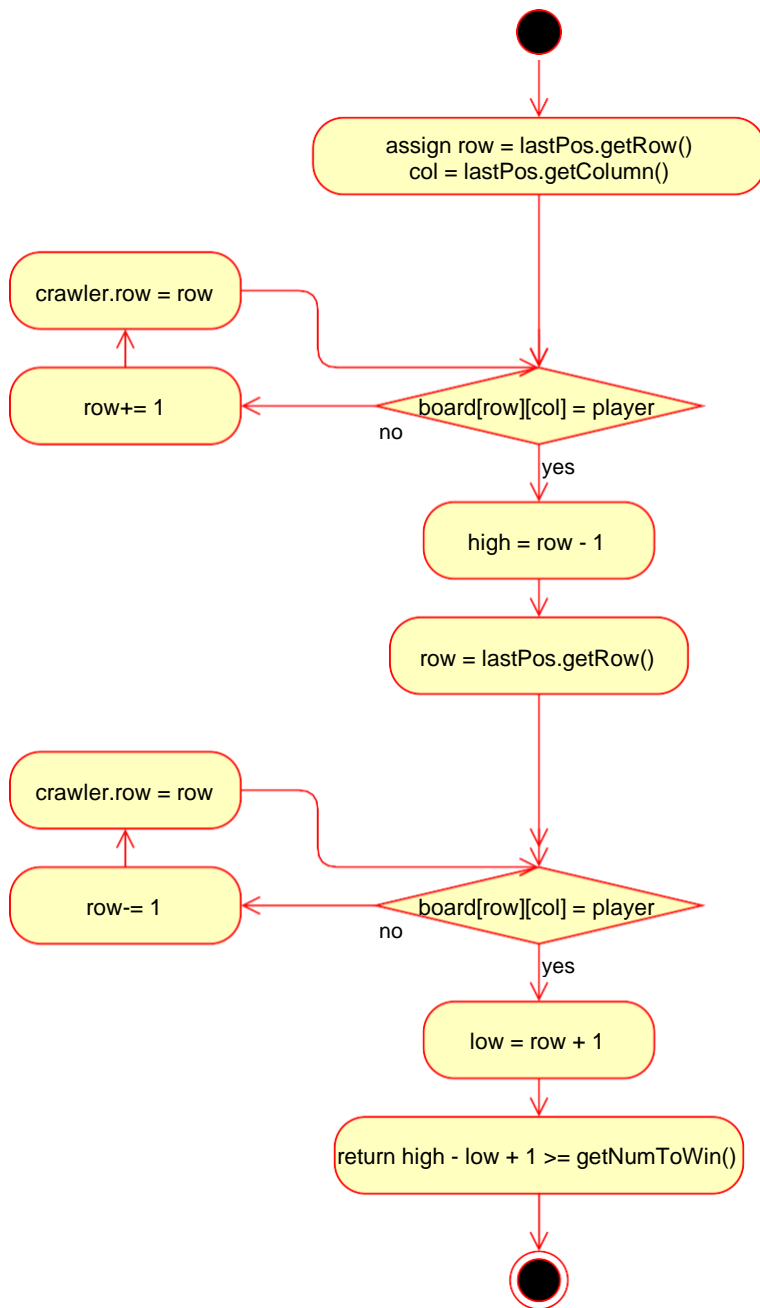


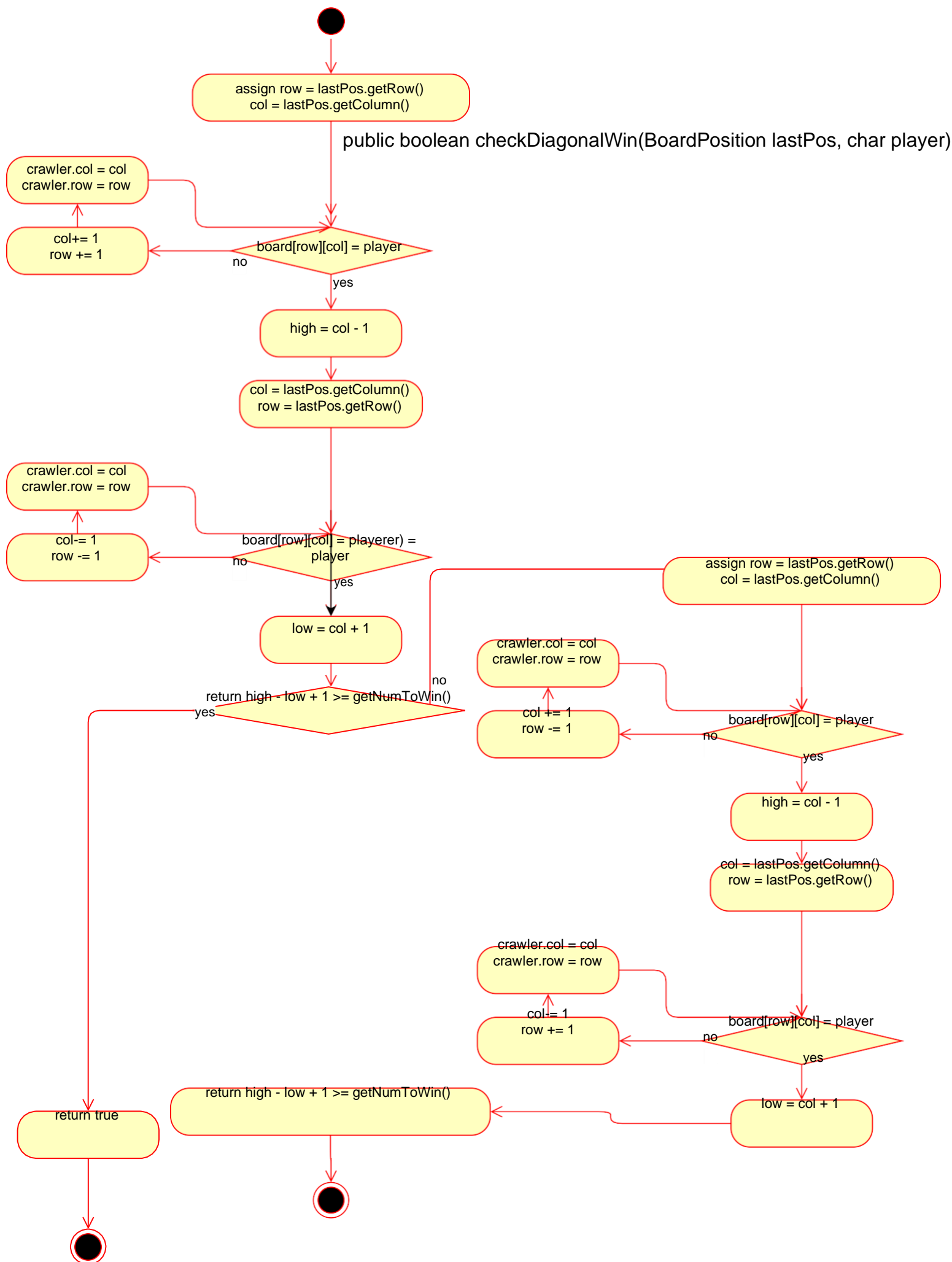
```
public bool checkForDraw()
```

public boolean checkHorizontalWin(BoardPosition lastPos, char player)

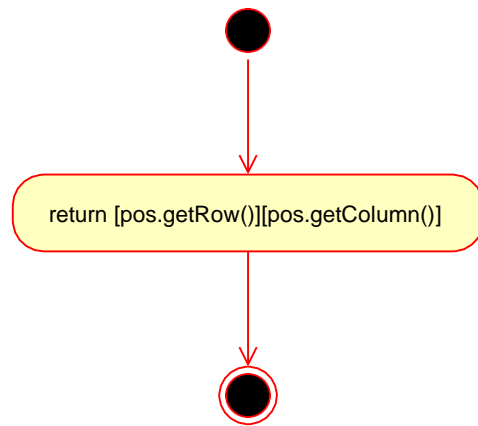


public boolean checkVerticalWin(BoardPosition lastPos, char player)



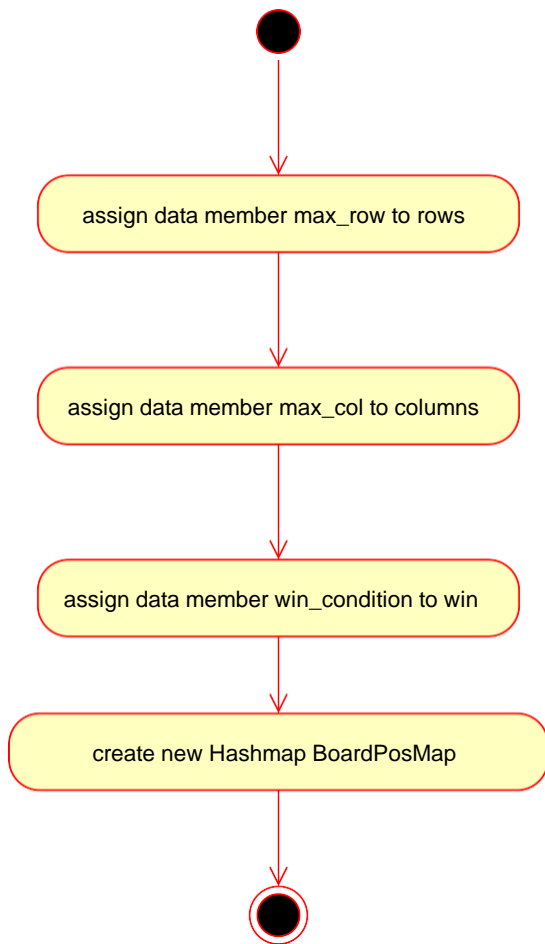


```
public char whatsAtPos(BoardPosition pos)
```

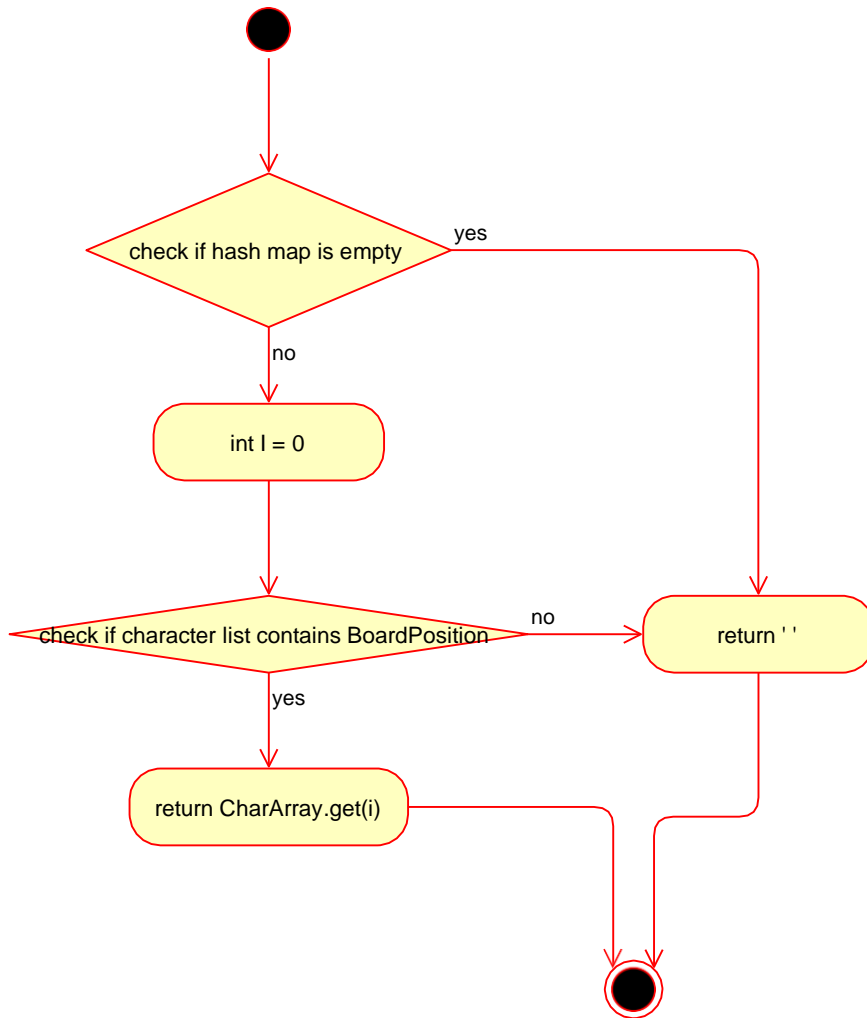


GameBoardMem

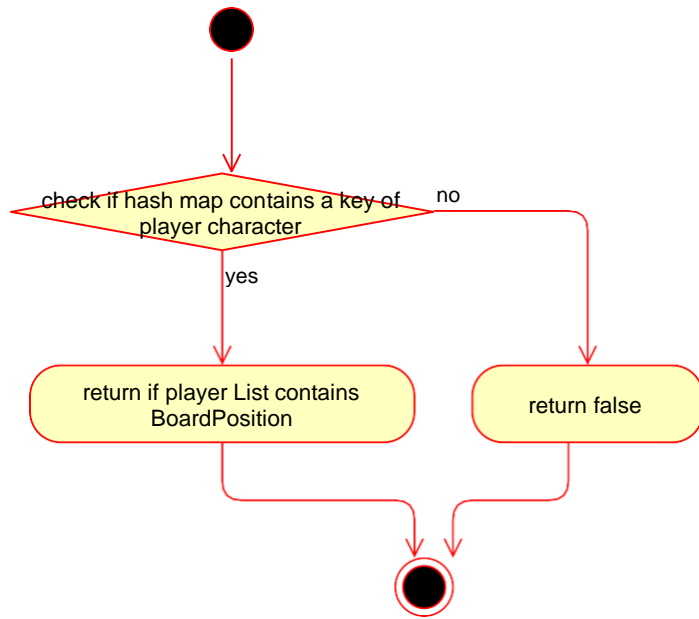
GameBoardMem(int rows, int columns, int win)



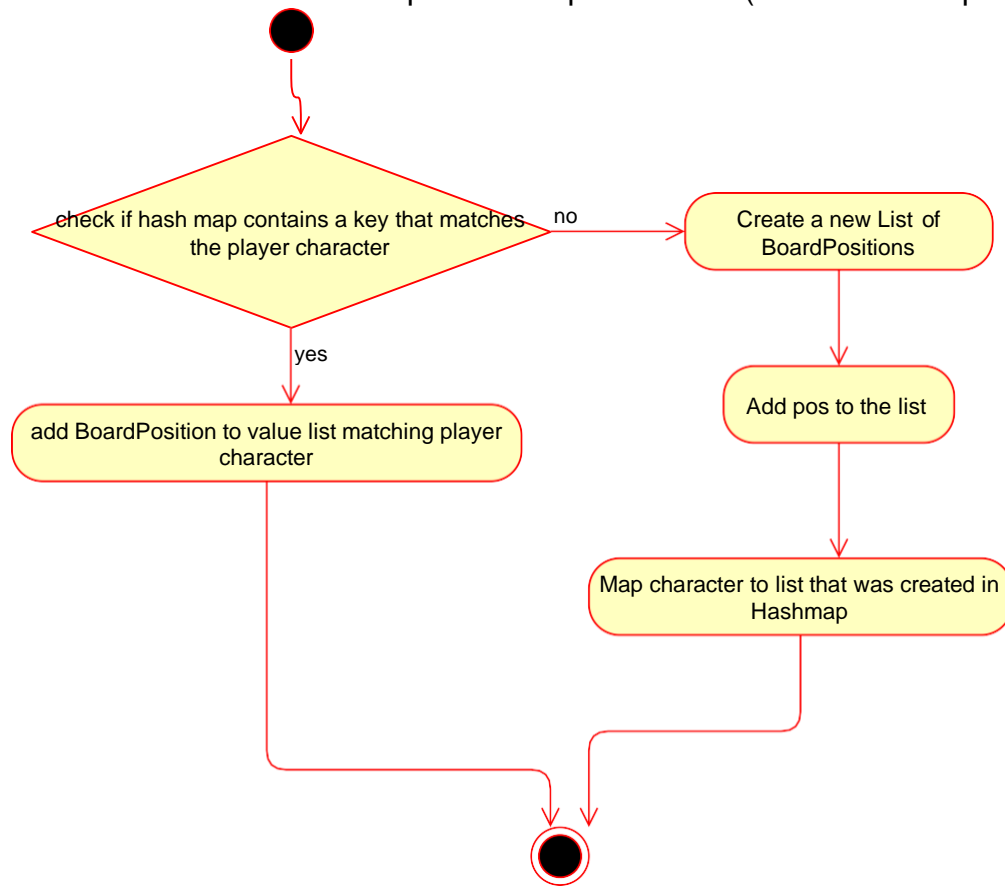
public char whatsAtPos(BoardPosition pos)



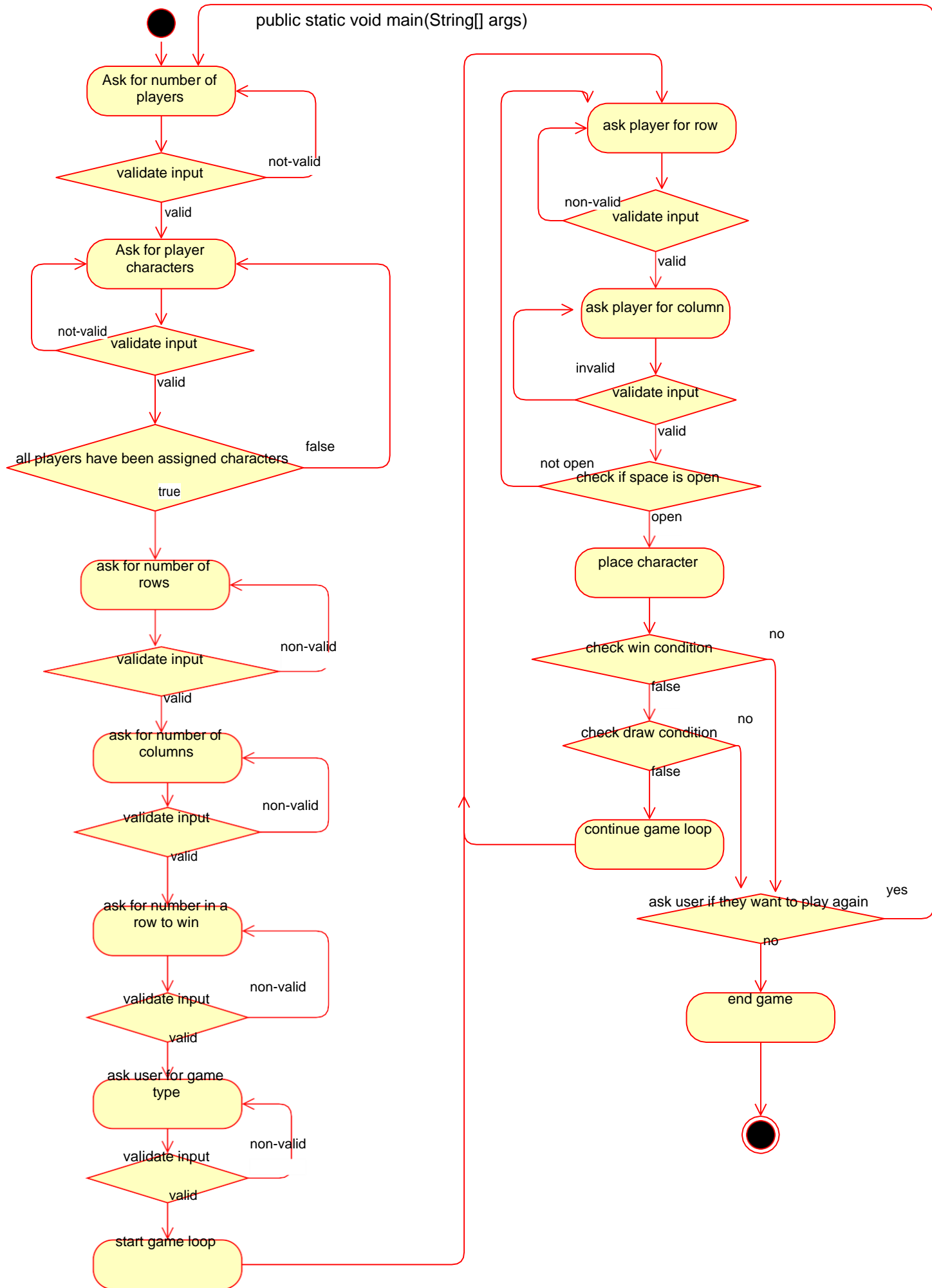

```
public boolean isPlayerAtPos(BoardPosition pos, char player)
```



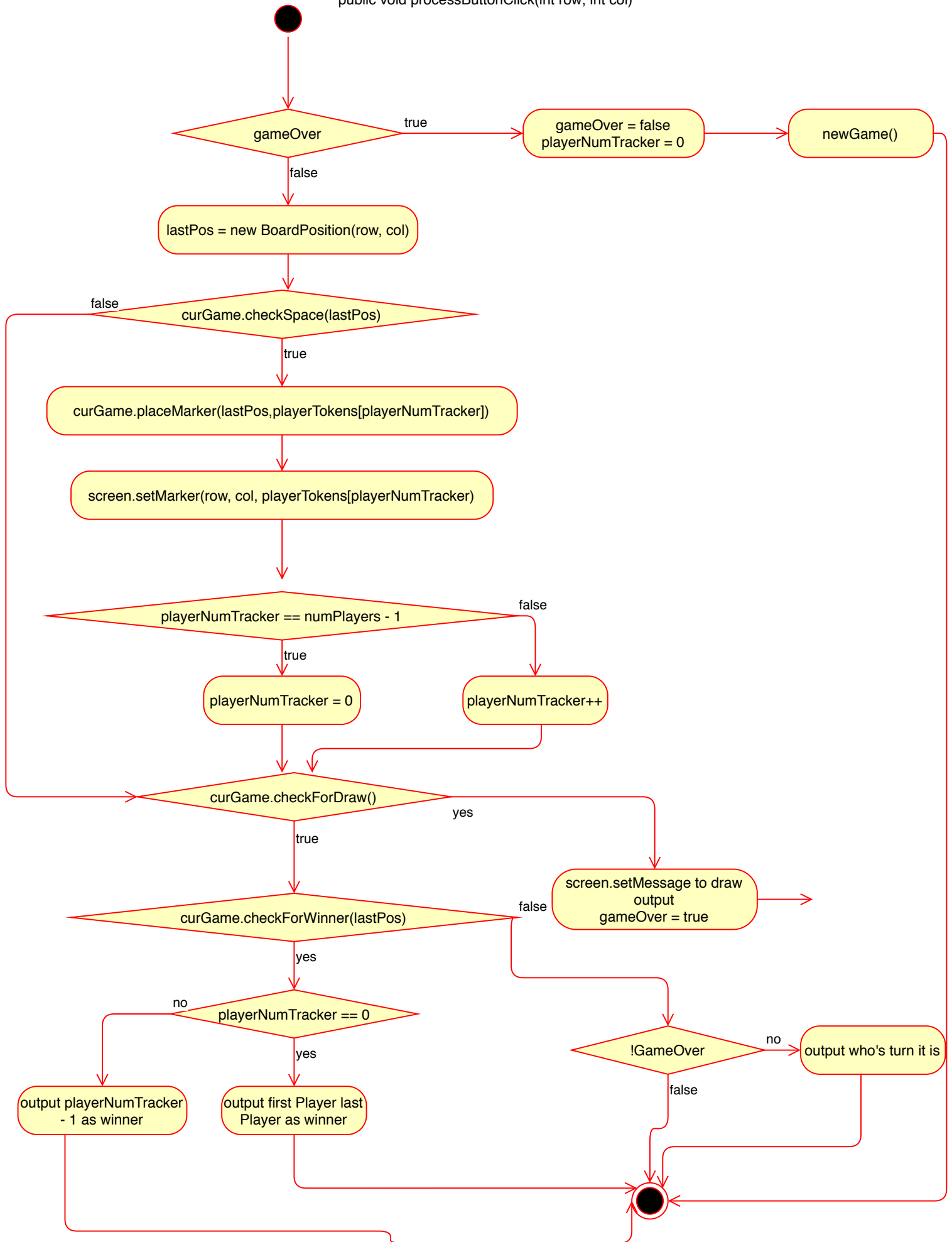
public void placeMarker(BoardPosition pos, char player)



GameScreen



public void processButtonClick(int row, int col)



Test Cases:

Test Cases are labeled as the method that is used by
Gameboard/GameBoardMem

Example: GameBoard(int rows, int columns, int win/
GameBoardMem(int rows, int columns, int win)

Examples are listed in this way, because testing is conducted in the same way for both GameBoard and GameBoardMem classes. In this way we can condense the length of this test cases section in half.

In the test cases, if the names of the methods for both GameBoard and GameBoardMem classes are the same, that means that both classes have their own definition for the Method.

Default methods come from IGameBoard Interface

```
public GameBoard(int rows, int columns, int win)/
public GameBoardMem(int rows, int columns, int win)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|
| Input: rows = 5 columns = 5 win = 3 | Output: Board is Created State: <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> | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Reason: Test Case is unique because it has the minimum number for win, but not the minimum numbers for rows and columns. Function Name: testConstructor_5x5_win_3() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public GameBoard(int rows, int columns, int win)/
public GameBoardMem(int rows, int columns, int win)
```

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|--|--|--|---|--|--|--|---|--|--|--|---|
| Input: rows = 3 columns = 3 win = 3 | Output: Board is Created State: <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> | | 0 | 1 | 2 | 0 | | | | 1 | | | | 2 | | | | Reason: Test Case is unique because it has the minimum number for both rows and columns, testing the limits of the constructor. Function Name: testConstructor_3x3_win_3() |
| | 0 | 1 | 2 | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | |

```
public GameBoard(int rows, int columns, int win)/  
public GameBoardMem(int rows, int columns, int win)
```

| | | |
|--|---|--|
| Input: rows = 100 columns = 100 win = 25 | Output: Board is Created Board is not pictured, because it is too big to fit. | Reason: TestCase is unique because it has the maximum number for both rows and columns, testing the limits of the constructor. Function Name: testConstructor_100x100_win_25() |
|--|---|--|

GameBoard and GameBoardMem both use the same default method

default Boolean checkSpace(BoardPosition pos)

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|
| Input: State: (number to win = 5) <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> pos.getRow() = 0 pos.getColumn() = 0 | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: Board is unchanged checkSpace = false | Reason: Test Case is unique because it checks the minimum values of pos.getRow() and pos.getColumn() with one character marker, and it is the only Test case to be false Function Name: testCheckSpace_5x5_win_5_check_0_0_after_1_turn() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

default Boolean checkSpace(BoardPosition pos)

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|---|
| Input: State: (number to win = 5) <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> pos.getRow() = 0 pos.getColumn() = 0 | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: Board is unchanged checkSpace.= true | Reason: Test Case is unique because it checks the minimum values of pos.getRow() and pos.getColumn() with an empty board Function Name: testCheckSpace_5x5_win_5_check_0_0_empty |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

default Boolean checkSpace(BoardPosition pos)

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| Input: State: (number to win = 5) <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> pos.getRow() = 4 pos.getColumn() = 4 | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: Board is unchanged checkSpace = true | Reason: Test Case is unique because it checks the maximum values of pos.getRow() and pos.getColumn() with an empty board Function Name: testCheckSpace_5x5_win_5_check_outer_edge() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkHorizontalWin(BoardPosition pos, char player)/
default Boolean checkHorizontalWin(BoardPosition lastPos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|--|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|
| Input: State: (number to win = 5) <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>a</td><td>a</td><td></td><td>a</td><td>a</td></tr><tr><td>1</td><td>b</td><td>b</td><td></td><td>b</td><td>b</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> pos.getRow() = 1 pos.getColumn() = 4 player = 'b' | | 0 | 1 | 2 | 3 | 4 | 0 | a | a | | a | a | 1 | b | b | | b | b | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: Board is unchanged checkHorizontalWin = false | Reason: Test Case is unique because it checks a case that causes errors, there is a line of 4 correct markers, with one missing in the middle. Function Name: testCheckHorizontalWin_5x5_win_5_check_split_pair() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | a | | a | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | b | b | | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkHorizontalWin(BoardPosition pos, char player)/
default Boolean checkHorizontalWin(BoardPosition lastPos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|---|
| Input: State: (number to win = 5) <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>a</td><td>a</td><td>b</td><td>a</td><td>a</td></tr><tr><td>1</td><td>b</td><td>b</td><td>b</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> pos.getRow() = 0 pos.getColumn() = 4 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | a | a | b | a | a | 1 | b | b | b | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: Board is unchanged checkHorizontalWin = false | Reason: Test Case is unique because it checks a case that causes errors, there is a line of 4 correct markers, with an opponent marker in the middle. Function Name: testCheckHorizontalWin_5x5_win_5_check_split_pair_with_opponent_in_middle() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | a | b | a | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | b | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

public Boolean checkHorizontalWin(BoardPosition pos, char player)/
 default Boolean checkHorizontalWin(BoardPosition lastPos, char player)

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| Input: State: (number to win = 5) <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>a</td><td>a</td><td>a</td><td>a</td><td>a</td></tr><tr><td>1</td><td></td><td>b</td><td>b</td><td>b</td><td>b</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> pos.getRow() = 0 pos.getColumn() = 0 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | a | a | a | a | a | 1 | | b | b | b | b | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: Board is unchanged checkHorizontalWin = true | Reason: Test Case is unique because it checks the methods ability to check for a win when the line of markers in question is on the edge of theboard. Function Name: testCheckHorizontalWin_5x5_win_5_check_winning_condition_on_board_edge() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | a | a | a | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | b | b | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

public Boolean checkHorizontalWin(BoardPosition pos, char player)/
 default Boolean checkHorizontalWin(BoardPosition lastPos, char player)

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|--|--|--|--|--|---|--|---|---|---|--|---|--|--|---|---|--|---|--|--|--|--|--|---|--|--|--|--|--|
| <div><div><div>Input:</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>b</td><td>b</td><td>b</td><td></td></tr><tr><td>2</td><td></td><td></td><td>a</td><td>a</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><div>pos.getRow() = 1 pos.getColumn() = 1 player = 'b'</div></div></div> <div><div><div>Output:</div><div>Board is unchanged</div><div>checkHorizontalWin = true</div></div></div> <div><div><div>Reason:</div><div>Test Case is unique because it checks the methods ability to check for a win when the line of markers in question is in the middle of the board.</div><div>Function Name: testCheckHorizontalWin_5x5_win_3_check_winning_condition()</div></div></div> | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | b | b | b | | 2 | | | a | a | | 3 | | | | | | 4 | | | | | |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | b | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | a | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkVerticalWin(BoardPosition pos, char player)/
default Boolean checkVerticalWin(BoardPosition pos, char player
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|--|--|--|---|---|---|--|--|--|---|--|--|--|--|--|---|---|---|--|--|--|---|---|---|--|--|--|--|---|
| Input: State: (number to win = 5) <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>a</td><td>b</td><td></td><td></td><td></td></tr><tr><td>1</td><td>a</td><td>b</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>a</td><td>b</td><td></td><td></td><td></td></tr><tr><td>4</td><td>a</td><td>b</td><td></td><td></td><td></td></tr></table> pos.getRow() = 4 pos.getColumn() = 1 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | a | b | | | | 1 | a | b | | | | 2 | | | | | | 3 | a | b | | | | 4 | a | b | | | | Output: Board is unchanged checkVerticalWin = false | Reason: Test Case is unique because it checks the methods ability to check for a win when there are two pairs of markers on either side of an empty space Function Name: testCheckVerticalWin_5x5_win_5_check_split_pair() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkVerticalWin(BoardPosition pos, char player)/
default Boolean checkVerticalWin(BoardPosition pos, char player
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|--|--|--|---|---|---|--|--|--|---|---|---|--|--|--|---|---|--|--|--|--|---|---|--|--|--|--|--|--|
| Input: State: (number to win = 5) <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>a</td><td>b</td><td></td><td></td><td></td></tr><tr><td>1</td><td>a</td><td>b</td><td></td><td></td><td></td></tr><tr><td>2</td><td>b</td><td>b</td><td></td><td></td><td></td></tr><tr><td>3</td><td>a</td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>a</td><td></td><td></td><td></td><td></td></tr></table> pos.getRow() = 4 pos.getColumn() = 0 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | a | b | | | | 1 | a | b | | | | 2 | b | b | | | | 3 | a | | | | | 4 | a | | | | | Output: Board is unchanged checkVerticalWin = false | Reason: Test Case is unique because it checks the methods ability to check for a win when there are two pairs of markers on either side of an opponent marker. Function Name: testCheckVerticalWin_5x5_win_5_check_split_pair_with_opponent_in_middle() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkVerticalWin(BoardPosition pos, char player)/
default Boolean checkVerticalWin(BoardPosition pos, char player
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|--|--|--|--|--|---|--|---|--|--|--|---|--|---|---|--|--|---|--|---|---|--|--|---|--|--|--|--|--|---|---|
| Input: State: (number to win = 3) <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>a</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td>a</td><td>b</td><td></td><td></td></tr><tr><td>3</td><td></td><td>a</td><td>b</td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table> pos.getRow() = 1 pos.getColumn() = 1 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | a | | | | 2 | | a | b | | | 3 | | a | b | | | 4 | | | | | | Output: Board is unchanged checkVerticalWin = true | Reason: Test Case is unique because it checks the methods ability to check for a win the line of markers in question is in the center of the board Function Name: testCheckVerticalWin_5x5_win_5_check_winning_condition() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkVerticalWin(BoardPosition pos, char player)/
default Boolean checkVerticalWin(BoardPosition pos, char player
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|--|--|--|--|---|---|---|--|--|--|---|---|---|--|--|--|---|---|---|--|--|--|---|---|---|--|--|--|---|--|
| Input: State: (number to win = 5) <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>b</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>b</td><td>a</td><td></td><td></td><td></td></tr><tr><td>2</td><td>b</td><td>a</td><td></td><td></td><td></td></tr><tr><td>3</td><td>b</td><td>a</td><td></td><td></td><td></td></tr><tr><td>4</td><td>b</td><td>a</td><td></td><td></td><td></td></tr></table> pos.getRow() = 0 pos.getColumn() = 0 player = 'b' | | 0 | 1 | 2 | 3 | 4 | 0 | b | | | | | 1 | b | a | | | | 2 | b | a | | | | 3 | b | a | | | | 4 | b | a | | | | Output: Board is unchanged checkVerticalWin = true | Reason: TestCase is unique because it checks the methods ability to check for a win the line of markers in question is on the edge of the board Function Name: testCheckVerticalWin_5x5_win_3_check_winning_condition_on_board_edge() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkDiagonalWin(BoardPosition pos, char player)/
default Boolean checkDiagonalWin(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|--|--|--|---|--|--|--|--|--|---|--|--|--|---|--|---|--|--|--|--|---|--|--|
| Input: State: (number to win = 5) <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>a</td><td>b</td><td>b</td><td>b</td><td>b</td></tr><tr><td>1</td><td></td><td>a</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>a</td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td>a</td></tr></table> pos.getRow() = 3 pos.getColumn() = 3 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | a | b | b | b | b | 1 | | a | | | | 2 | | | | | | 3 | | | | a | | 4 | | | | | a | Output: Board is unchanged checkVerticalWin = false | Reason: TestCase is unique because it checks the methods ability to check for a win where the markers are arranged in two pairs around an empty space Function Name: testCheckDiagonalWin_5x5_win_5_pair_left_is_high_check_split() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | b | b | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkDiagonalWin(BoardPosition pos, char player)/
default Boolean checkDiagonalWin(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|--|--|--|---|--|--|---|--|--|---|--|--|--|---|--|---|--|--|--|--|---|--|---|
| Input: State: (number to win = 5) <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>a</td><td>b</td><td>b</td><td>b</td><td>b</td></tr><tr><td>1</td><td></td><td>a</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td>b</td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td>a</td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td>a</td></tr></table> pos.getRow() = 3 pos.getColumn() = 3 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | a | b | b | b | b | 1 | | a | | | | 2 | | | b | | | 3 | | | | a | | 4 | | | | | a | Output: Board is unchanged checkVerticalWin = false | Reason: Test Case is unique because it checks the methods ability to check for a win where the markers are arranged in two pairs around an opponent marker Function Name: testCheckDiagonalWin_5x5_win_5_check_split_pair_left_is_high_with_opponent_in_middle() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | b | b | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkDiagonalWin(BoardPosition pos, char player)/
default Boolean checkDiagonalWin(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|--|--|--|---|--|--|---|--|--|---|--|--|--|---|--|---|--|--|--|--|---|---|---|
| Input: State: (number to win = 5) <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>a</td><td>b</td><td>b</td><td>b</td><td>b</td></tr><tr><td>1</td><td></td><td>a</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td>a</td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td>a</td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td>a</td></tr></table> pos.getRow() = 3 pos.getColumn() = 3 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | a | b | b | b | b | 1 | | a | | | | 2 | | | a | | | 3 | | | | a | | 4 | | | | | a | Output: Board is unchanged checkVerticalWin = true | Reason: Test Case is unique because it checks the methods ability to check for a win where the markers are arranged with the left side higher on the board edge Function Name: testcheckDiagonalWin_5x5 _win_5_left_is_high_check _winning_condition_on _board_edge() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | b | b | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkDiagonalWin(BoardPosition pos, char player)/
default Boolean checkDiagonalWin(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|--|--|--|---|--|---|--|--|--|---|--|--|---|--|--|---|--|--|--|---|--|---|--|--|--|--|--|---|--|
| Input: State: (number to win = 3) <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>b</td><td>b</td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td>a</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td>a</td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td>a</td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table> pos.getRow() = 3 pos.getColumn() = 3 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | b | b | | | | 1 | | a | | | | 2 | | | a | | | 3 | | | | a | | 4 | | | | | | Output: Board is unchanged checkVerticalWin = true | Reason: Test Case is unique because it checks the methods ability to check for a win where the markers are arranged with the left side higher in the center of the board Function Name: testcheckDiagonalWin_5x5 _win_3_left_is_high_check _winning_condition_center _of_board() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |


```
public Boolean checkDiagonalWin(BoardPosition pos, char player)/
default Boolean checkDiagonalWin(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|---|--|---|--|--|--|--|--|---|--|---|--|--|--|---|---|--|--|--|--|--|---|
| Input: State: (number to win = 5) <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>b</td><td>b</td><td>b</td><td>b</td><td>a</td></tr><tr><td>1</td><td></td><td></td><td></td><td>a</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>a</td><td></td><td></td><td></td></tr><tr><td>4</td><td>a</td><td></td><td></td><td></td><td></td></tr></table> pos.getRow() = 4 pos.getColumn() = 0 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | b | b | b | b | a | 1 | | | | a | | 2 | | | | | | 3 | | a | | | | 4 | a | | | | | Output: Board is unchanged checkVerticalWin = false | Reason: Test Case is unique because it checks the methods ability to check for a win where the markers are arranged with the right side is higher and they are arranged in two pairs diagonally Function Name: testCheckDiagonalWin_5x5_win_5_pair_right_is_high_check_split() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | b | b | b | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkDiagonalWin(BoardPosition pos, char player)/
default Boolean checkDiagonalWin(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|--|---|---|--|--|--|---|--|---|--|--|---|--|--|---|--|---|--|--|--|---|---|--|--|--|--|--|---|
| Input: State: (number to win = 5) <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>b</td><td>b</td><td>b</td><td></td><td>a</td></tr><tr><td>1</td><td></td><td></td><td></td><td>a</td><td></td></tr><tr><td>2</td><td></td><td></td><td>b</td><td></td><td></td></tr><tr><td>3</td><td></td><td>a</td><td></td><td></td><td></td></tr><tr><td>4</td><td>a</td><td></td><td></td><td></td><td></td></tr></table> pos.getRow() = 4 pos.getColumn() = 0 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | b | b | b | | a | 1 | | | | a | | 2 | | | b | | | 3 | | a | | | | 4 | a | | | | | Output: Board is unchanged checkVerticalWin = false | Reason: Test Case is unique because it checks the methods ability to check for a win where the markers are arranged with the right side is higher and they are arranged in two pairs diagonally with an opponent character in the center. Function Name: testCheckDiagonalWin_5x5 _win_5_check_split_pair_right _is_high_with_opponent_in_middle() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | b | b | b | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkDiagonalWin(BoardPosition pos, char player)/
default Boolean checkDiagonalWin(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|---|--|---|--|--|---|--|--|---|--|---|--|--|--|---|---|--|--|--|--|---|---|
| Input: State: (number to win = 5) <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>b</td><td>b</td><td>b</td><td>b</td><td>a</td></tr><tr><td>1</td><td></td><td></td><td></td><td>a</td><td></td></tr><tr><td>2</td><td></td><td></td><td>a</td><td></td><td></td></tr><tr><td>3</td><td></td><td>a</td><td></td><td></td><td></td></tr><tr><td>4</td><td>a</td><td></td><td></td><td></td><td></td></tr></table> pos.getRow() = 2 pos.getColumn() = 2 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | b | b | b | b | a | 1 | | | | a | | 2 | | | a | | | 3 | | a | | | | 4 | a | | | | | Output: Board is unchanged checkVerticalWin = true | Reason: TestCase is unique because it checks the methods ability to check for a win where the markers are arranged with the right side is higher and they are arranged in a row long enough for a win on the boards edge Function Name: testcheckDiagonalWin_5x5_win_5_right_is_high_check_winning_condition_on_board_edge() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | b | b | b | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean checkDiagonalWin(BoardPosition pos, char player)/
default Boolean checkDiagonalWin(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|--|--|--|---|--|--|--|---|--|---|--|--|---|--|--|---|--|---|--|--|--|---|--|--|--|--|--|---|--|
| Input: State: (number to win = 3) <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>b</td><td>b</td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td>a</td><td></td></tr><tr><td>2</td><td></td><td></td><td>a</td><td></td><td></td></tr><tr><td>3</td><td></td><td>a</td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr></table> pos.getRow() = 1 pos.getColumn() = 3 player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | b | b | | | | 1 | | | | a | | 2 | | | a | | | 3 | | a | | | | 4 | | | | | | Output: Board is unchanged checkVerticalWin = true | Reason: Test Case is unique because it checks the methods ability to check for a win where the markers are arranged with the right side is higher and they are arranged in a row long enough for a win in the middle of the board Function Name: testcheckDiagonalWin_5x5_win_3_right_is_high_check_winning_condition_center_of_board() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

public Boolean checkForDraw()/ default Boolean checkForDraw()

| Input: | Output: | Reason: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---------|---------|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|---|
| State: (number to win = 5) <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> | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Board is unchanged checkForDraw = false | Test Case is unique because it checks the methods ability to check for a draw on the empty board Function Name: testCheckForDraw_5x5 _win_5_check_draw_on _empty_board() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

public Boolean checkForDraw()/ default Boolean checkForDraw()

| Input: | Output: | Reason: | | | | | | | | | | | | | | | | |
|---|---------|---------|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|---|
| State: (number to win = 3) <table><tr><td></td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td>a</td><td>a</td><td>a</td></tr><tr><td>1</td><td>a</td><td>a</td><td>a</td></tr><tr><td>2</td><td>a</td><td>a</td><td></td></tr></table> | | 0 | 1 | 2 | 0 | a | a | a | 1 | a | a | a | 2 | a | a | | Board is unchanged checkForDraw = false | TestCase is unique because it checks the methods ability to check for a draw on a board that is full except for a point on the edge of the board Function Name: testCheckForDraw_3x3 _win_3_check_draw_empty _space_on_edge_one_player _character() |
| | 0 | 1 | 2 | | | | | | | | | | | | | | | |
| 0 | a | a | a | | | | | | | | | | | | | | | |
| 1 | a | a | a | | | | | | | | | | | | | | | |
| 2 | a | a | | | | | | | | | | | | | | | | |

public Boolean checkForDraw()/ default Boolean checkForDraw()

| | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|--|---|
| Input: State: (number to win = 3) <table><tr><td></td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td>a</td><td>a</td><td>a</td></tr><tr><td>1</td><td>a</td><td></td><td>a</td></tr><tr><td>2</td><td>a</td><td>a</td><td>a</td></tr></table> | | 0 | 1 | 2 | 0 | a | a | a | 1 | a | | a | 2 | a | a | a | Output: Board is unchanged checkForDraw = false | Reason: TestCase is unique because it checks the methods ability to check for a draw on a board that is full except for a point in the middle of the board Function Name: testCheckForDraw_3x3 _win_3_check_draw_empty _space_in_center_one_ player_character() |
| | 0 | 1 | 2 | | | | | | | | | | | | | | | |
| 0 | a | a | a | | | | | | | | | | | | | | | |
| 1 | a | | a | | | | | | | | | | | | | | | |
| 2 | a | a | a | | | | | | | | | | | | | | | |

public Boolean checkForDraw()/ default Boolean checkForDraw()

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input: State: (number to win = 3) <table><tr><td></td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td>a</td><td>a</td><td>a</td></tr><tr><td>1</td><td>a</td><td>a</td><td>a</td></tr><tr><td>2</td><td>a</td><td>a</td><td>a</td></tr></table> | | 0 | 1 | 2 | 0 | a | a | a | 1 | a | a | a | 2 | a | a | a | Output: Board is unchanged checkForDraw = true | Reason: Test Case is unique because it checks the methods ability to check for a draw on a board that is full Function Name: testCheckForDraw_3x3 _win_3_check_win_one _player_character() |
| | 0 | 1 | 2 | | | | | | | | | | | | | | | |
| 0 | a | a | a | | | | | | | | | | | | | | | |
| 1 | a | a | a | | | | | | | | | | | | | | | |
| 2 | a | a | a | | | | | | | | | | | | | | | |

```
public char whatsAtPos(BoardPosition pos)/
public char whatsAtPos(BoardPosition pos)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| Input: State: (number to win = 5) <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> Pos.getRow() = 0 Pos.getcolumn() = 0 | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: Board is unchanged checkForDraw = true | Reason: Test Case is unique because it checks the methods ability to check what is at the position pos for an empty board Function Name: testwhatsAtPos_5x5 _win_5_check_0_0_ empty_space_empty_board() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public char whatsAtPos(BoardPosition pos)/
public char whatsAtPos(BoardPosition pos)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|
| <div><div><div>Input:</div><div>State: (number to win = 5)</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>a</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><div>Pos.getRow() = 0 Pos.getcolumn() = 0</div></div></div> <div><div><div>Output:</div><div>Board is unchanged</div><div>Returns 'a'</div></div></div> <div><div><div>Reason:</div><div>TestCase is unique because it checks the methods ability to check that a board position that holds a char does not evaluate to a space</div><div>Function Name: testWhatsAtPos_5x5 _win_5_check_not_equal_to_space()</div></div></div> | | 0 | 1 | 2 | 3 | 4 | 0 | a | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public char whatsAtPos(BoardPosition pos)/
public char whatsAtPos(BoardPosition pos)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|---|
| <div><div>Input:</div><div>State: (number to win = 5)</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>a</td><td>b</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><div>Pos.getRow() = 0 Pos.getcolumn() = 0</div></div> | | 0 | 1 | 2 | 3 | 4 | 0 | a | b | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | <div><div>Output:</div><div>Board is unchanged</div><div>Returns 'a'</div></div> | <div><div>Reason:</div><div>Test Case is unique because it checks the methods ability to check multiple positions on one board</div><div>Function Name: testWhatsAtPos_5x5 _win_5_check_multiple _player_locations_different _player_tokens()</div></div> |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | |
|---|--|--|
| <p>Input:</p> <p>Test runs method multiple times, this is the second instance</p> <p>Input is the same as above</p> <p>Pos.getRow() =0 Pos.getcolumn() = 1</p> | <p>Output:</p> <p>Board is unchanged</p> <p>Returns 'b'</p> | <p>Reason:</p> <p>Same Reason as above</p> <p>Function Name: testWhatsAtPos_5x5 _win_5_check_multiple _player_locations_different _player_tokens()</p> |
|---|--|--|

```
public char whatsAtPos(BoardPosition pos)/
public char whatsAtPos(BoardPosition pos)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|
| Input: State: (number to win = 5) <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>a</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> Pos.getRow() = 0 Pos.getcolumn() = 0 | | 0 | 1 | 2 | 3 | 4 | 0 | a | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: Board is unchanged Returns 'a' | Reason: TestCase is unique because it checks the methods ability to check multiple positions on one board and make sure that it has the ability to distinguish between character markers Function Name: testWhatsAtPos_5x5 _win_5_check_multiple _player_locations_to_make _sure_method_will_not _confuse_player_tokens() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | |
|--|--|--|
| <p>Input:</p> <p>Test runs method multiple times, this is the second instance</p> <p>Input is the same as above</p> <p>Pos.getRow() = 0 Pos.getcolumn() = 1</p> | <p>Output:</p> <p>Board is unchanged</p> <p>Returns 'b'</p> | <p>Reason:</p> <p>Same Reason as above</p> <p>Function Name: testWhatsAtPos_5x5 _win_5_check_multiple _player_locations_to_make _sure_method_will_not _confuse_player_tokens()</p> |
|--|--|--|

```
public char whatsAtPos(BoardPosition pos)/
publicchar whatsAtPos(BoardPosition pos)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|
| <div><div>Input:</div><div>State: (number to win = 5)</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>a</td><td>a</td><td>a</td><td>a</td><td>a</td></tr><tr><td>1</td><td>a</td><td>a</td><td>a</td><td>a</td><td>a</td></tr><tr><td>2</td><td>a</td><td>a</td><td>a</td><td>a</td><td>a</td></tr><tr><td>3</td><td>a</td><td>a</td><td>a</td><td>a</td><td>a</td></tr><tr><td>4</td><td>a</td><td>a</td><td>a</td><td>a</td><td>a</td></tr></table><div>Pos.getRow() = 0 Pos.getcolumn() = 0</div></div> | | 0 | 1 | 2 | 3 | 4 | 0 | a | a | a | a | a | 1 | a | a | a | a | a | 2 | a | a | a | a | a | 3 | a | a | a | a | a | 4 | a | a | a | a | a | <div><div>Output:</div><div>Board is unchanged</div><div>Returns 'a' 25 times as it loops through the board</div></div> | <div><div>Reason:</div><div>Test Case is unique because it checks the methods ability to check all positions on the board</div><div>Function Name: testWhatsAtPos_5x5 _win_5_check_all_locations()</div></div> |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | a | a | a | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | a | a | a | a | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | a | a | a | a | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | a | a | a | a | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | a | a | a | a | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |


```
public Boolean isPlayerAtPos(BoardPosition pos, char player)/
public Boolean isPlayerAtPos(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|---|
| Input: State: (number to win = 5) <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> Pos.getRow() = 0 Pos.getcolumn() = 0 Player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: Board is unchanged isPlayerAtPos = false | Reason: Test Case is unique because it checks the methods ability to check if player is at the position on an empty board Function Name: testisPlayerAtPos_5x5 _win_5_check_empty _space_empty_board() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean isPlayerAtPos(BoardPosition pos, char player)/
public Boolean isPlayerAtPos(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|---|
| Input: State: (number to win = 5) <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>a</td><td>b</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> Pos.getRow() = 0 Pos.getcolumn() = 0 Player = 'b' | | 0 | 1 | 2 | 3 | 4 | 0 | a | b | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: Board is unchanged isPlayerAtPos = false | Reason: Test Case is unique because it checks the methods ability to deny that a player is at location, when there is an opponents marker at the position Function Name: testisPlayerAtPos_5x5 _win_5_check_space _of_opponent() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean isPlayerAtPos(BoardPosition pos, char player)/
public Boolean isPlayerAtPos(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|---|--|---|
| Input: State: (number to win = 5) <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>a</td></tr></table> Pos.getRow() = 4 Pos.getcolumn() = 4 Player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | a | Output: Board is unchanged isPlayerAtPos = true | Reason: TestCase is unique because it checks the methods ability to confirm players location at maximum values of pos.getRow() and pos.getColumn() Function Name: testisPlayerAtPos_5x5 _win_5_check_space _max_val_rows_and _columns() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public Boolean isPlayerAtPos(BoardPosition pos, char player)/
public Boolean isPlayerAtPos(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|--|--|---|--|--|--|---|--|--|--|--|--|
| Input: State: (number to win = 3) <table><tr><td></td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td>a</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> Pos.getRow() = 0 Pos.getcolumn() = 0 Player = 'a' | | 0 | 1 | 2 | 0 | a | | | 1 | | | | 2 | | | | Output: Board is unchanged isPlayerAtPos = true | Reason: TestCase is unique because it checks the methods ability to confirm players location at minimum values of pos.getRow() and pos.getColumn() on a small board Function Name: testisPlayerAtPos_3x3 _win_3_small_board _test_correct() |
| | 0 | 1 | 2 | | | | | | | | | | | | | | | |
| 0 | a | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | |

```
public Boolean isPlayerAtPos(BoardPosition pos, char player)/
public Boolean isPlayerAtPos(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|
| Input: State: (number to win = 3) <table><tr><td></td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td>a</td><td>b</td><td>a</td></tr><tr><td>1</td><td>b</td><td>a</td><td>b</td></tr><tr><td>2</td><td>a</td><td>b</td><td>a</td></tr></table> (pos.getRow(), pos.getColumn()) = (0,0), (0,2), (1,1), (2,0), (2,2) Player = 'a' | | 0 | 1 | 2 | 0 | a | b | a | 1 | b | a | b | 2 | a | b | a | Output: Board is unchanged isPlayerAtPos = true | Reason: TestCase is unique because it checks the methods ability to confirm players location at minimum values of pos.getRow() and pos.getColumn() on a small board Function Name: testisPlayerAtPos_3x3 _win_3_small_board_test _all_location_full_board _test_correct() |
| | 0 | 1 | 2 | | | | | | | | | | | | | | | |
| 0 | a | b | a | | | | | | | | | | | | | | | |
| 1 | b | a | b | | | | | | | | | | | | | | | |
| 2 | a | b | a | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | |
|--|--------------------|----------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----------------------|-----------------------|
| Input: | Output: | Reason: | | | | | | | | | | | | | | | | |
| State: (number to win = 3) | Board is unchanged | Same as above | | | | | | | | | | | | | | | | |
| <table><tr><td></td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td>a</td><td>b</td><td>a</td></tr><tr><td>1</td><td>b</td><td>a</td><td>b</td></tr><tr><td>2</td><td>a</td><td>b</td><td>a</td></tr></table> | | 0 | 1 | 2 | 0 | a | b | a | 1 | b | a | b | 2 | a | b | a | isPlayerAtPos = true | Function Name: |
| | 0 | 1 | 2 | | | | | | | | | | | | | | | |
| 0 | a | b | a | | | | | | | | | | | | | | | |
| 1 | b | a | b | | | | | | | | | | | | | | | |
| 2 | a | b | a | | | | | | | | | | | | | | | |
| (pos.getRow(), pos.getColumn()) = (0,1), (1,0), (1,2), (2,1) Player = 'b' | | Same as above | | | | | | | | | | | | | | | | |

```
public void placeMarker(BoardPosition pos, char player)/
public void placeMarker(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|---|---|---|---|---|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|
| Input: State: (number to win = 5) <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> Pos.getRow() = 0 Pos.getColumn() = 0 Player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: State: (number to win = 5) <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>a</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 | a | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Reason: TestCase is unique because it checks the methods ability to place a player marker at the position 0,0 on the board Function Name: testPlaceMarker_5x5 _win_5_5x5_win_5 _after_1_turn()) |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public void placeMarker(BoardPosition pos, char player)/
public void placeMarker(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|
| Input: State: (number to win = 5) <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> (pos.getRow(), pos.getColumn()) = (0,0), (0,1), (0,2), (0,3), (0,4) Player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: State: (number to win = 5) <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>a</td><td>a</td><td>a</td><td>a</td><td>a</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 | a | a | a | a | a | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Reason: Test Case is unique because it checks the methods ability to place a player marker at multiple positions along the top row of the board Function Name: testPlaceMarker_5x5 _win_5_top_row _complete() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | a | a | a | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public void placeMarker(BoardPosition pos, char player)/
public void placeMarker(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|---|---|---|---|---|---|---|--|--|--|--|---|---|--|--|--|--|---|---|--|--|--|--|---|---|--|--|--|--|---|---|--|--|--|--|--|
| Input: State: (number to win = 5) <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> (pos.getRow(), pos.getColumn(), player) = (0,0), (1,0), (2,0), (3,0), (4,0) Player = 'a' | | 0 | 1 | 2 | 3 | 4 | 0 | | | | | | 1 | | | | | | 2 | | | | | | 3 | | | | | | 4 | | | | | | Output: State: (number to win = 5) <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>a</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>a</td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>a</td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td>a</td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>a</td><td></td><td></td><td></td><td></td></tr></table> | | 0 | 1 | 2 | 3 | 4 | 0 | a | | | | | 1 | a | | | | | 2 | a | | | | | 3 | a | | | | | 4 | a | | | | | Reason: Test Case is unique because it checks the methods ability to place a player marker at multiple positions along the first column of the board Function Name: testPlaceMarker_5x5 _win_5_first_column _complete() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public void placeMarker(BoardPosition pos, char player)/
public void placeMarker(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|----------------------------|--|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|
| Input: | Output: | Reason: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| State: (number to win = 5) | State: (number to win = 5) | Test Case is unique because it checks the methods ability to place a player marker at multiple positions with multiple charactertags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <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> | | 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>a</td><td>b</td><td>c</td><td>d</td><td>e</td></tr><tr><td>1</td><td>f</td><td>g</td><td>h</td><td>i</td><td>j</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 | a | b | c | d | e | 1 | f | g | h | i | j | 2 | | | | | | 3 | | | | | | 4 | | | | | | Function Name: testPlaceMarker_5x5 _win_5_several _character_tags() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | b | c | d | e | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | f | g | h | i | j | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
public void placeMarker(BoardPosition pos, char player)/
public void placeMarker(BoardPosition pos, char player)
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|----------------------------|--|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input: | Output: | Reason: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| State: (number to win = 5) | State: (number to win = 5) | Test Case is unique because it checks the methods ability to place a player marker at all positions with multiple character tags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <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> | | 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>a</td><td>b</td><td>a</td><td>b</td><td>a</td></tr><tr><td>1</td><td>b</td><td>a</td><td>b</td><td>a</td><td>b</td></tr><tr><td>2</td><td>a</td><td>b</td><td>a</td><td>b</td><td>a</td></tr><tr><td>3</td><td>b</td><td>a</td><td>b</td><td>a</td><td>b</td></tr><tr><td>4</td><td>a</td><td>b</td><td>a</td><td>b</td><td>a</td></tr></table> | | 0 | 1 | 2 | 3 | 4 | 0 | a | b | a | b | a | 1 | b | a | b | a | b | 2 | a | b | a | b | a | 3 | b | a | b | a | b | 4 | a | b | a | b | a | Function Name: testPlaceMarker_5x5 _win_5_full_board() |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | b | a | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | b | a | b | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | a | b | a | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | b | a | b | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | a | b | a | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (pos.getRow(), pos.getColumn()) = (0,0), (0,2), (0,4), (1,1), (1,3), (2,0), (2,2), (2,4), (3,1), (3,3), (4,0), (4,2), (4,4) Player = 'a' | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------------------------|----------------|---|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|
| Input: | Output: | Reason: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| State: (number to win = 5) | State: (number to win=5) | Same as above | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <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> | | 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>a</td><td>b</td><td>a</td><td>b</td><td>a</td></tr><tr><td>1</td><td>b</td><td>a</td><td>b</td><td>a</td><td>b</td></tr><tr><td>2</td><td>a</td><td>b</td><td>a</td><td>b</td><td>a</td></tr><tr><td>3</td><td>b</td><td>a</td><td>b</td><td>a</td><td>b</td></tr><tr><td>4</td><td>a</td><td>b</td><td>a</td><td>b</td><td>a</td></tr></table> | | 0 | 1 | 2 | 3 | 4 | 0 | a | b | a | b | a | 1 | b | a | b | a | b | 2 | a | b | a | b | a | 3 | b | a | b | a | b | 4 | a | b | a | b | a | Function Name: Same as above |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | a | b | a | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | b | a | b | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | a | b | a | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | b | a | b | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | a | b | a | b | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>(pos.getRow(), pos.getColumn()) = (0,1), (0,3), (1,0), (1,2), (1,4), (2,1), (2,3), (3,0), (3,2), (3,4), (4,1), (4,3) Player = 'b'</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Deployment:

Makefile directions * does not apply for project 5 with no test cases

make or make default

- Compiles all .java files not including the test classes

make run:

- Runs program

make test:

- Compiles all .java files including test classes

make testGB:

- Runs test program for Gameboard class

make testGBmem

- Runs test program for GameboardMem class

make clean

- Removes all .class files