Mike Brandin

Dr. Yu-Shan Sun

CPSC 2150 Section 002

March 21st, 2021

Project Report:

Project 3

Requirements Analysis

Functional Requirements

Gamescreen.java

- As a user, I can input a column number, so I can mark a spot on the game board and progress the game.
- As a user, I can input a Y or an N after the game has concluded, to start a new game or terminate the game session, respectively.
- As a user, I can place four markers touching horizontally, to conclude the match and receive a congratulatory message and a request to play again.
- As a user, I can place four markers touching vertically, to conclude the match and receive a congratulatory message and a request to play again.
- As a user, I can place four markers touching diagonally, to conclude the match and receive a congratulatory message and a request to play again.
- As a user, I can fill the board fully with markers and not win, to conclude the match with a tie and a request to play again.
- As a user, I can input a column value not within the confines of the gameboard and receive an error message followed by a prompt to re-enter the value.
- As a user, I can input an integer value for the number of players, so that I the number of players can be established.
- As a user, I can input a character value for each player, to represent the player's tokens on the gameboard.
- As a user, I can input a row length, to establish the length of the rows on the gameboard.
- As a user, I can input a column length, to establish the length of the columns on the gameboard.
- As a user, I can input a number to win, to establish the desired number needed to achieve in order to win the game.
- As a user, I can input an f/F or m/M, to decide whether or not to run the gameboard as a fast implementation or a more memory efficient implementation.

Gameboard.java

- As a user, I must be able to input a position value and receive character value, to know what player is in that position.
- As a user, I must be able to input a column number, to input a token into the highest available row in that column.

- As a user, I must be able to call a function to receive the value of the number of rows.
- As a user, I must be able to call a function to receive the value of the number of columns.
- As a user, I must be able to call a function to receive the value of the number needed to win.

GameboardMem.java

- As a user, I must be able to input a position value and receive character value, to know what player is in that position.
- As a user, I must be able to input a column number, to input a token into the highest available row in that column.
- As a user, I must be able to call a function to receive the value of the number of rows.
- As a user, I must be able to call a function to receive the value of the number of columns.
- As a user, I must be able to call a function to receive the value of the number needed to win.

AbsGameboard.java

- As a user, I must be able to request a fully formatted string representation of the gameboard, to visualize the current gameboard.

IGameboard.java

- As a user, I must be able to input a column number and receive a true or false value, to know whether or not a column is free for more tokens.
- As a user, I must be able to input a column number and receive a true or false value, to know whether or not the last placed token resulted in a win.
- As a user, I must be able to input a column number and receive a true or false value, to know whether or not the last placed token resulted in a tie (a full board).
- As a user, I must be able to input a character token and a position value and receive a true or false value, to know whether or not the last placed token resulted in a horizontal win.
- As a user, I must be able to input a character token and a position value and receive a true or false value, to know whether or not the last placed token resulted in a vertical win.
- As a user, I must be able to input a character token and a position value and receive a true or false value, to know whether or not the last placed token resulted in a diagonal win.
- As a user, I must be able to input a character representing a player and a position value and receive a true or false value, to know whether or not that player is in that position.

BoardPosition.java

- As a user, I must be able to receive a string output to display the row and column coordinates of the position.
- As a user, I must be able to request the Row variables value, to know the value of the board position's row.
- As a user, I must be able to request the Column variables value, to know the value of the board position's column.
- As a user, I must be able to compare two BoardPosition variables, to know whether their positional values are equivalent.

Nonfunctional Requirements

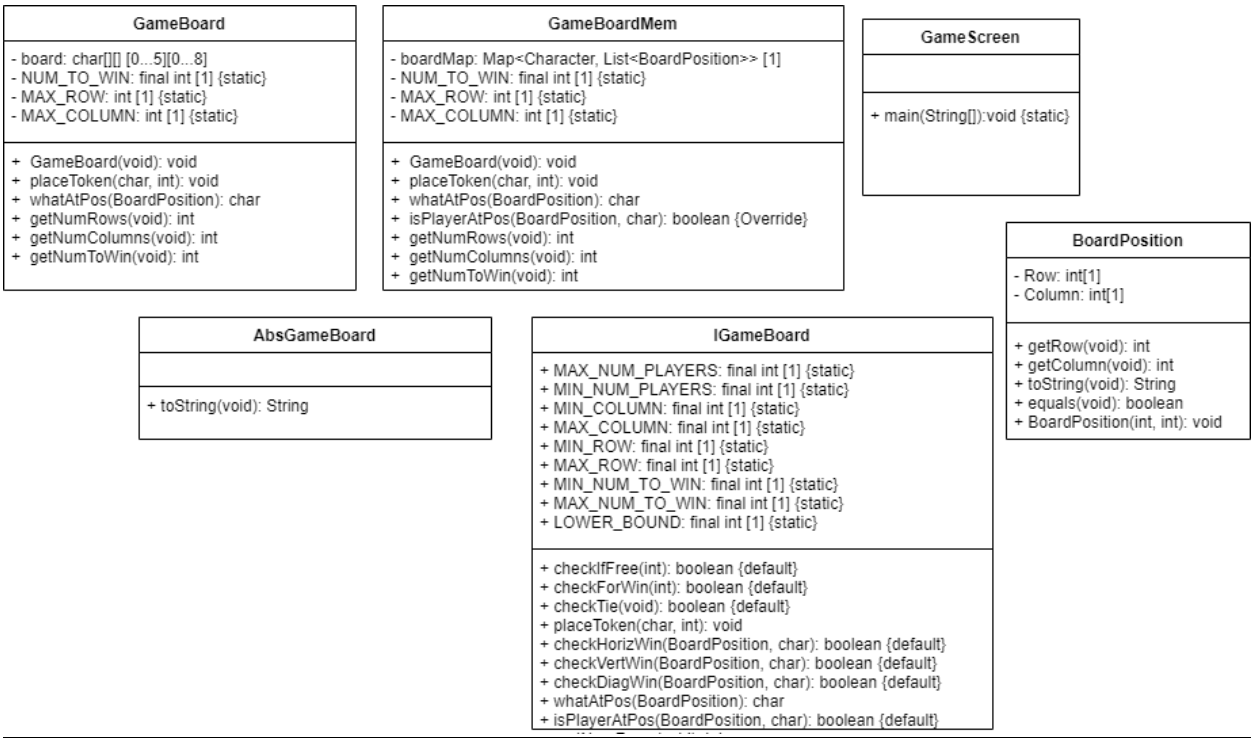
- Must have a device that supports Java.
- Must have a keyboard to play the game.
- Must run on the Schools of Computing's virtual machine.
- Must have adequate memory to allocate towards objects.
- Must handle all I/O in GameScreen.java
- Gameboard size must not exceed 100 columns nor 100 rows.
- Gameboard size must not be less than 3 columns nor 3 rows.
- Players must take turns in the order they selected their character at the start of the game.

Make File Instructions

To use the make file, open the terminal in the Project3 directory. Type **make** into the terminal followed by enter to compile the project files. Then, type **make run** into the terminal followed by enter to run the program. Once you are finished, type **make clean** into the terminal followed by enter to remove all compiled files from the extendedConnectX package directory.

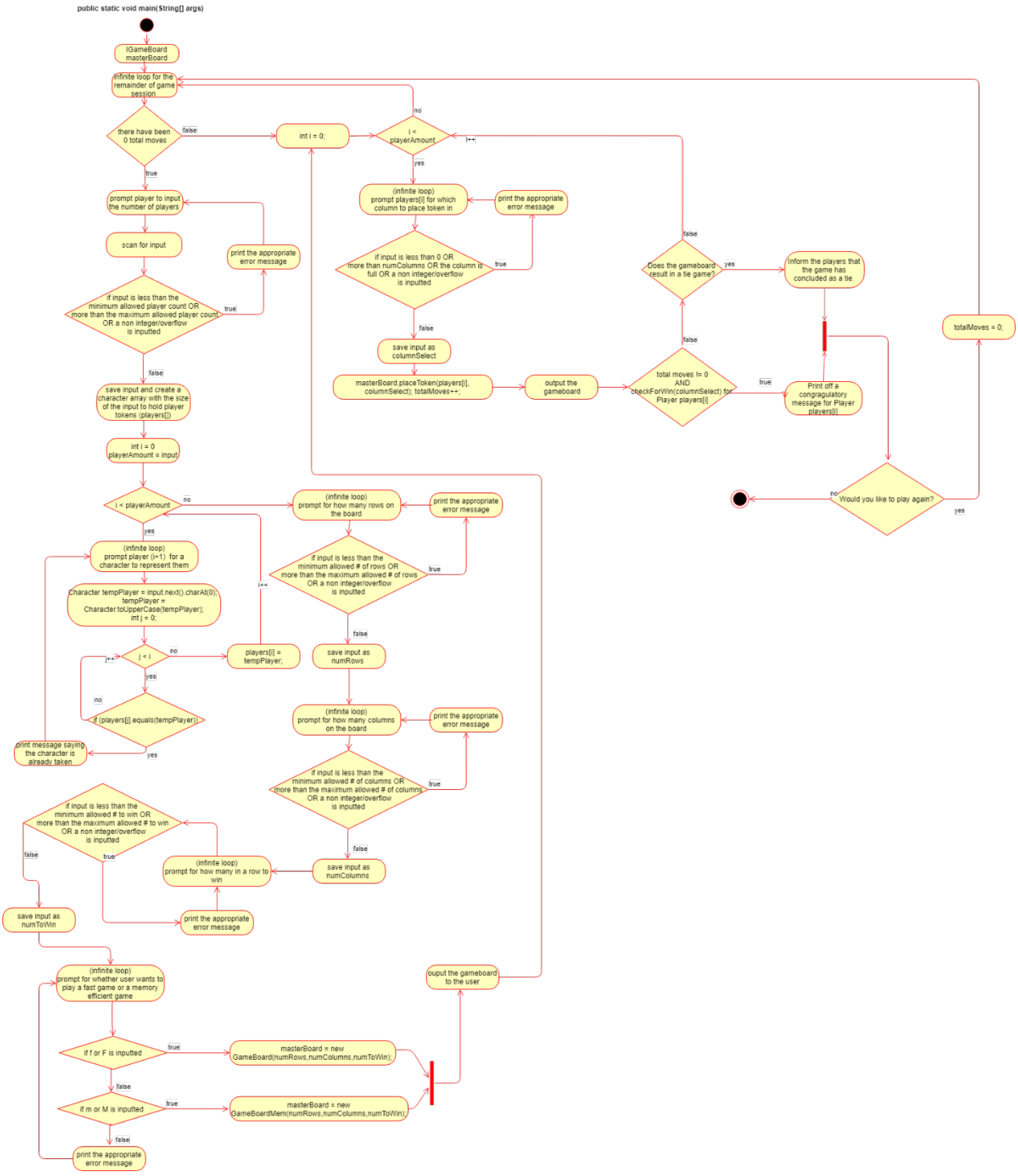
Design

UML Class Diagrams

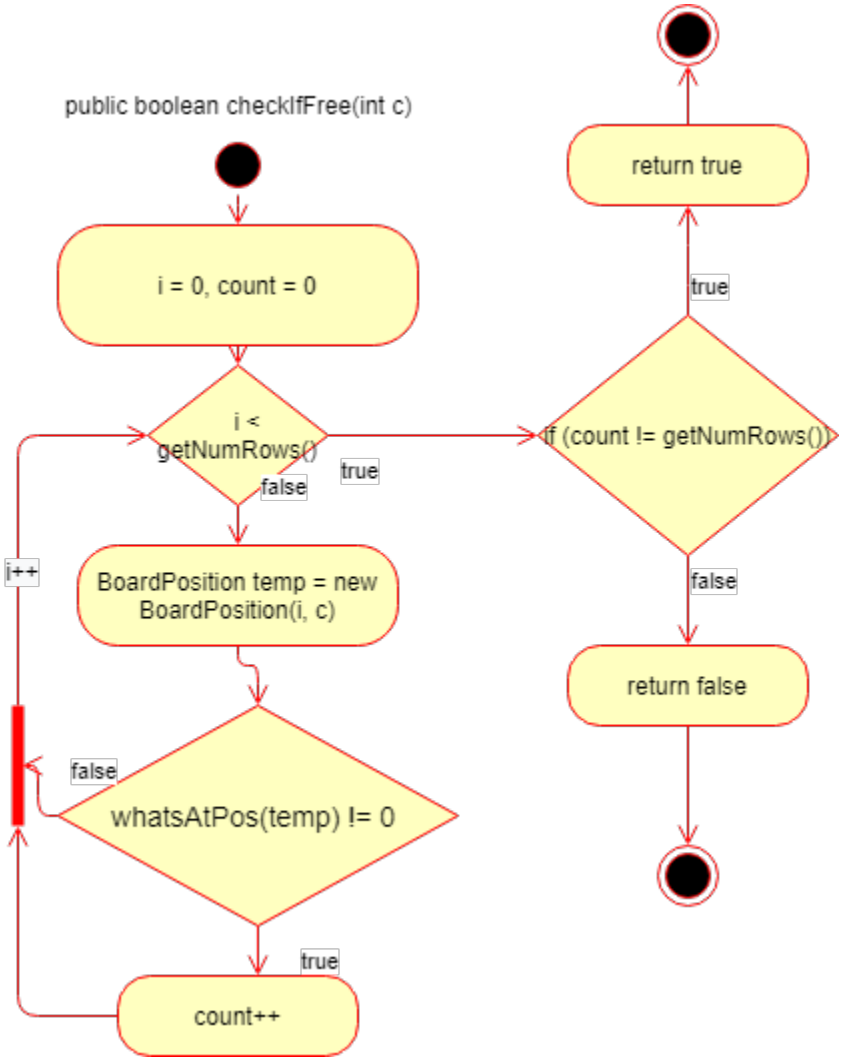


UML Activity Diagrams

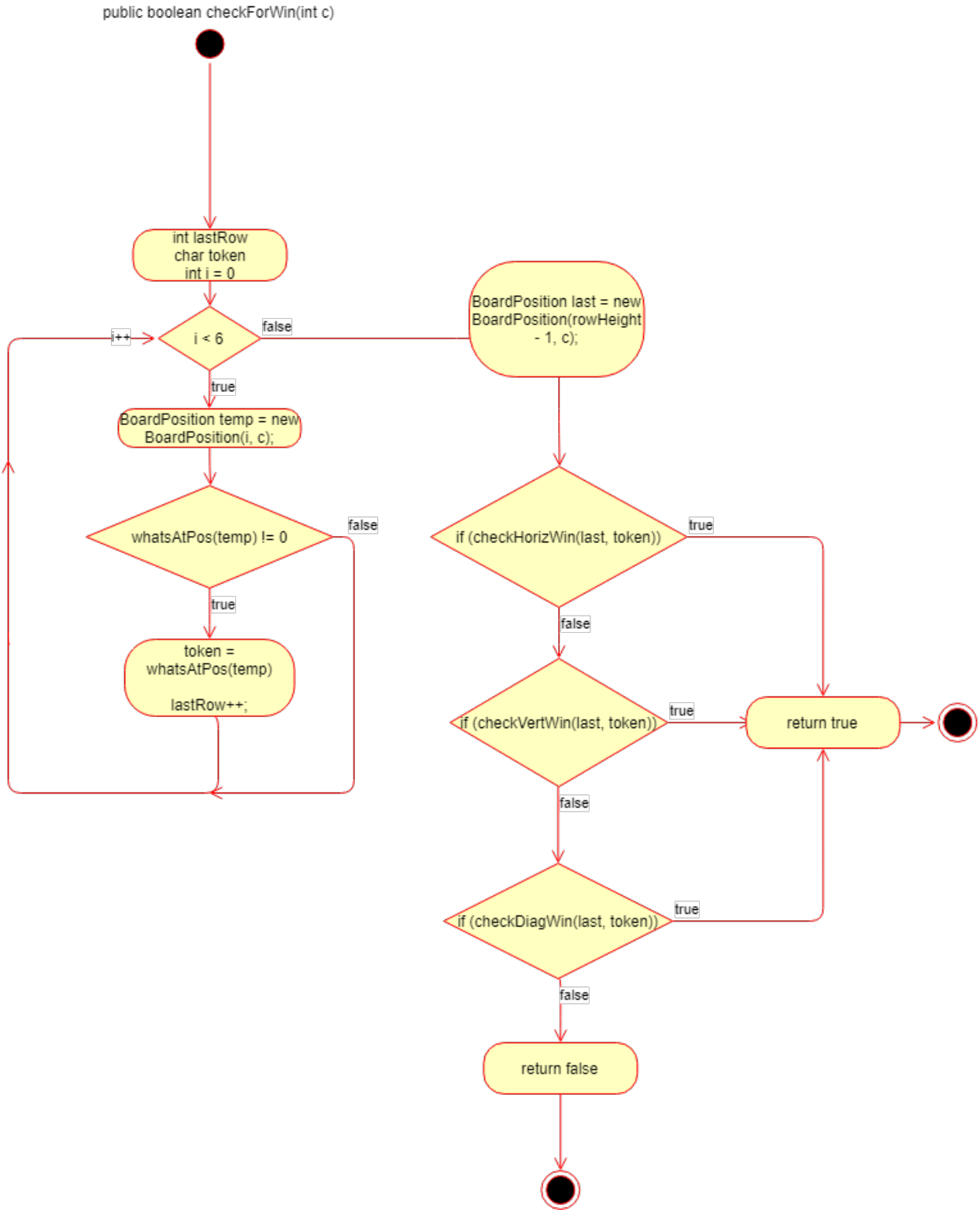
- GameScreen.java:
  - main function



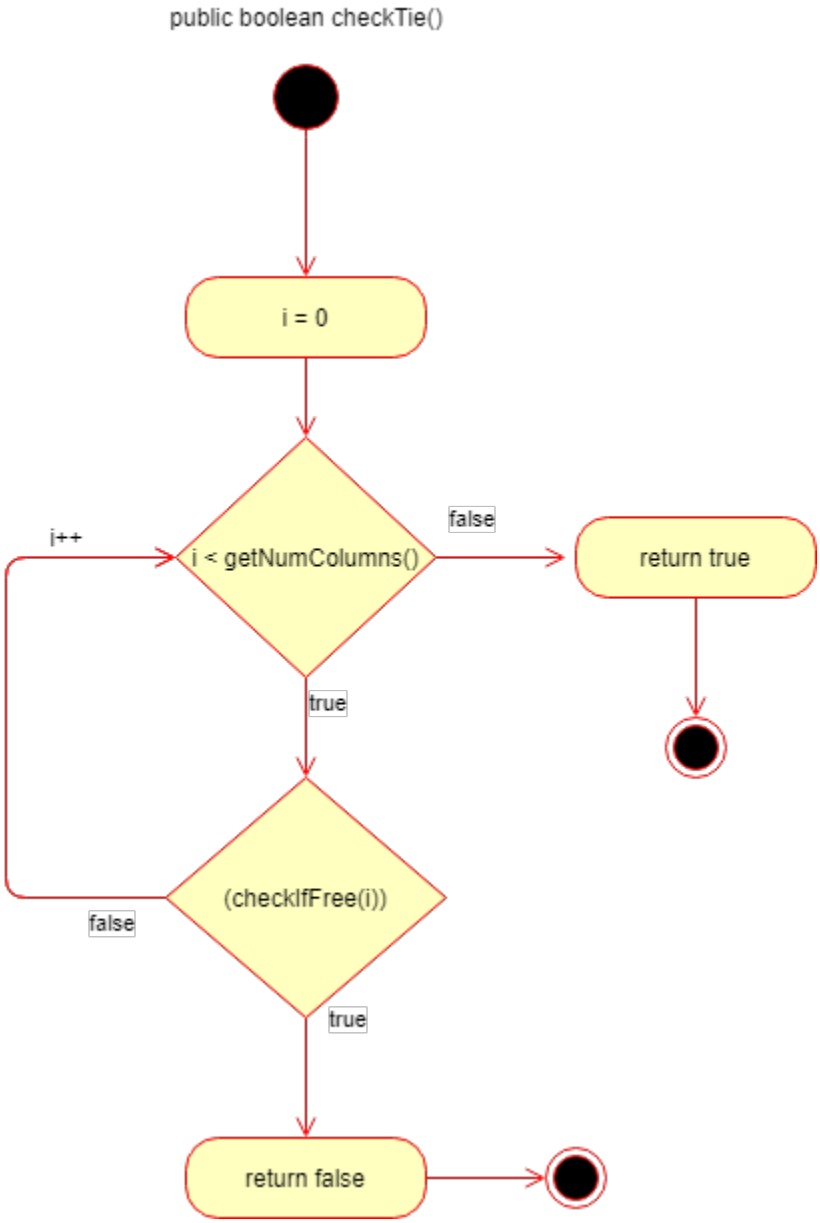
- IGameBoard.java
    - checkIfFree method



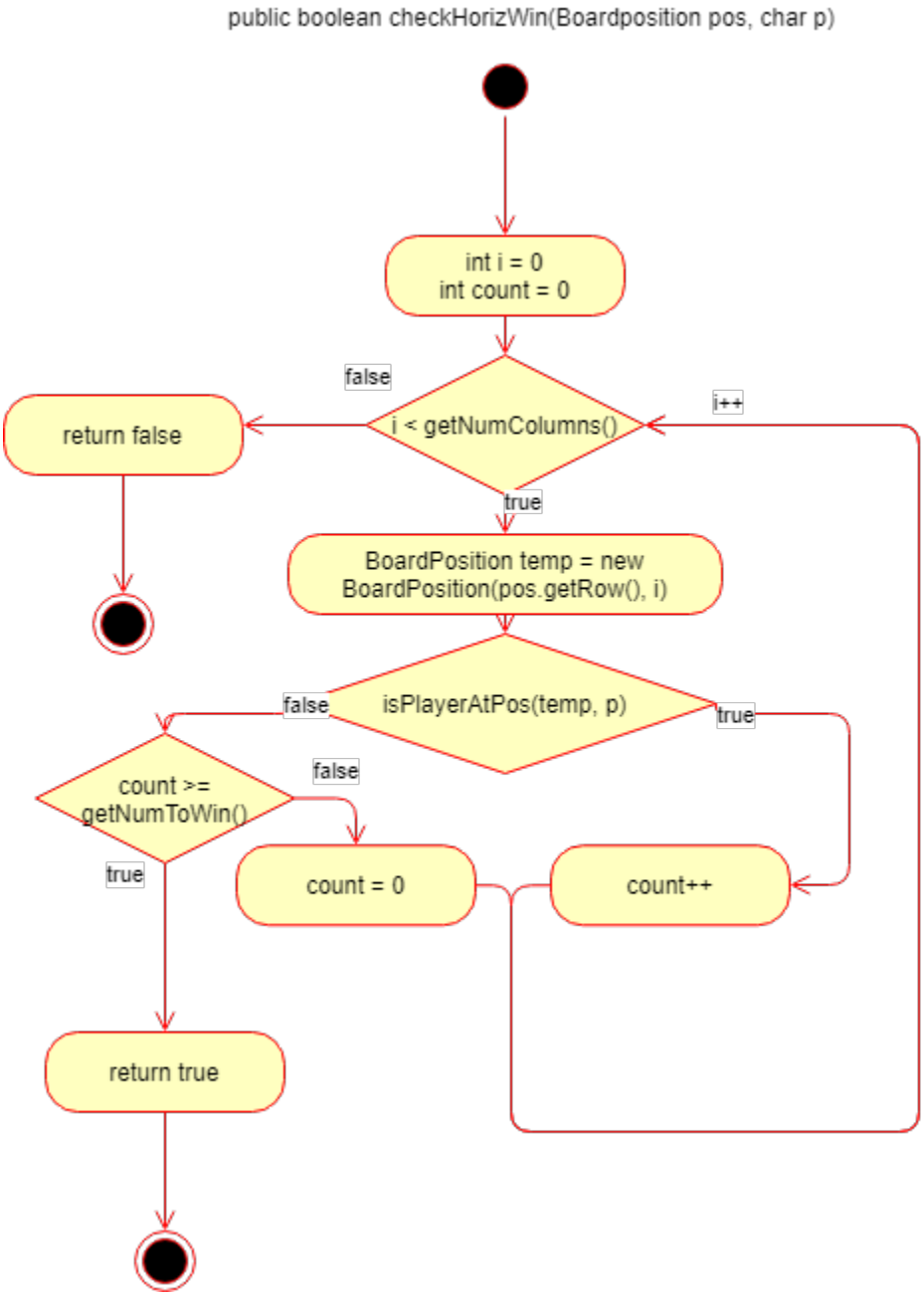
- checkForWin method



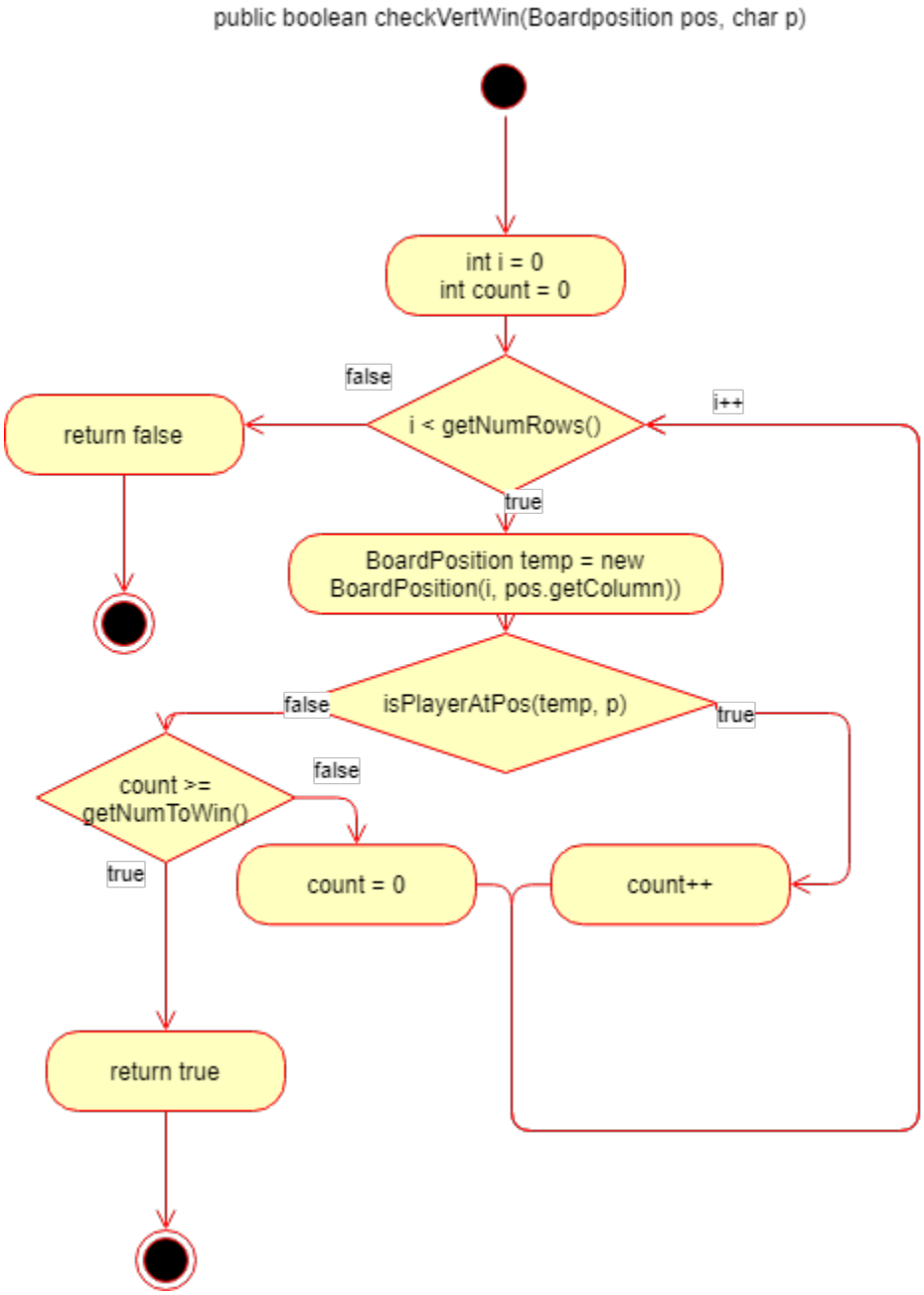
- checkTie method



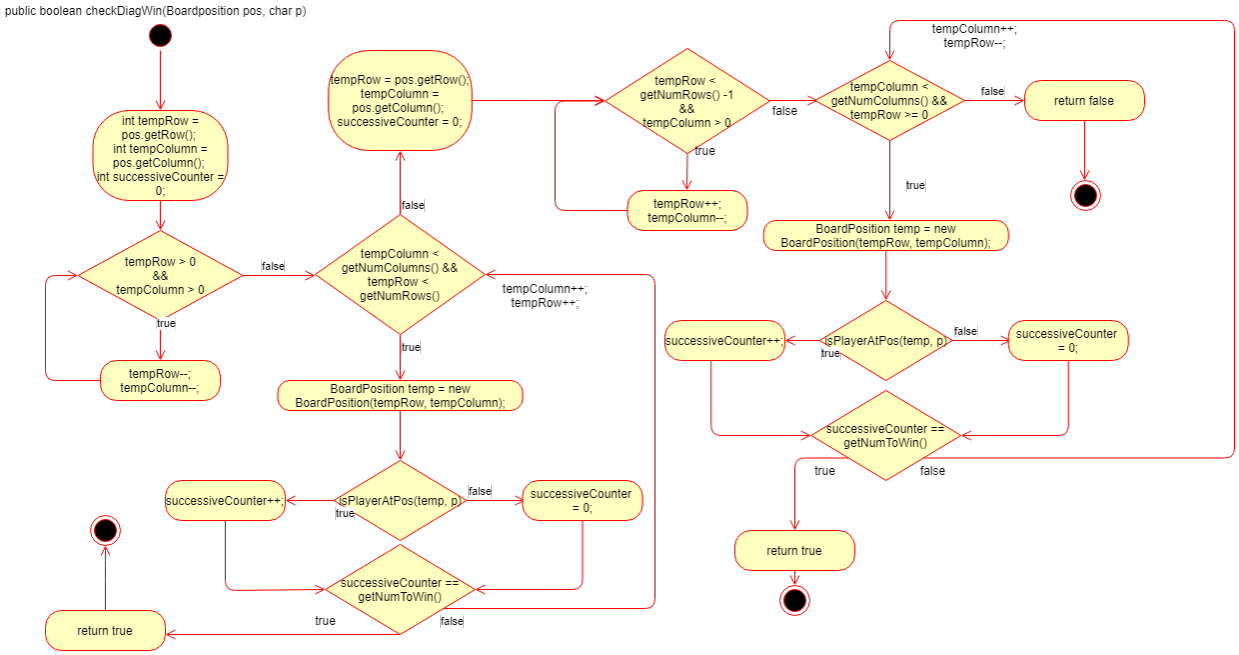
- checkHorizWin method



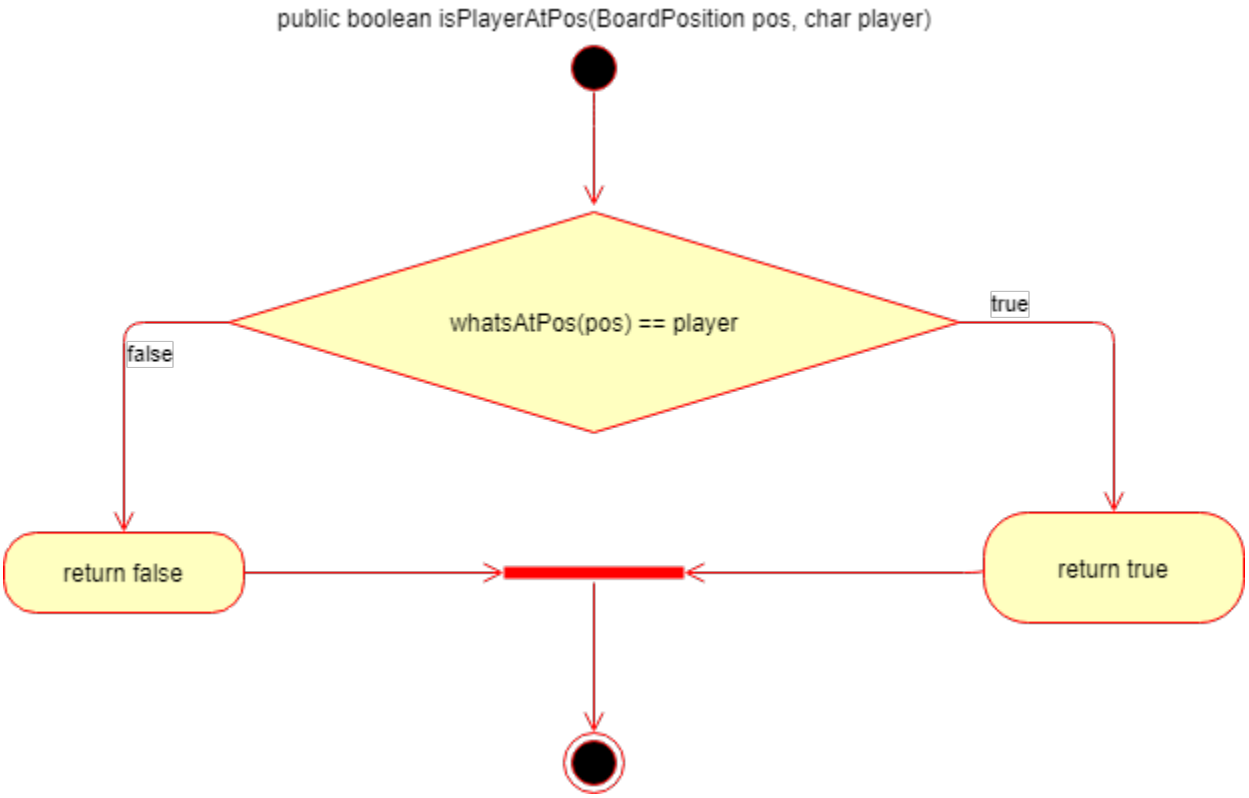
- checkVertWin method



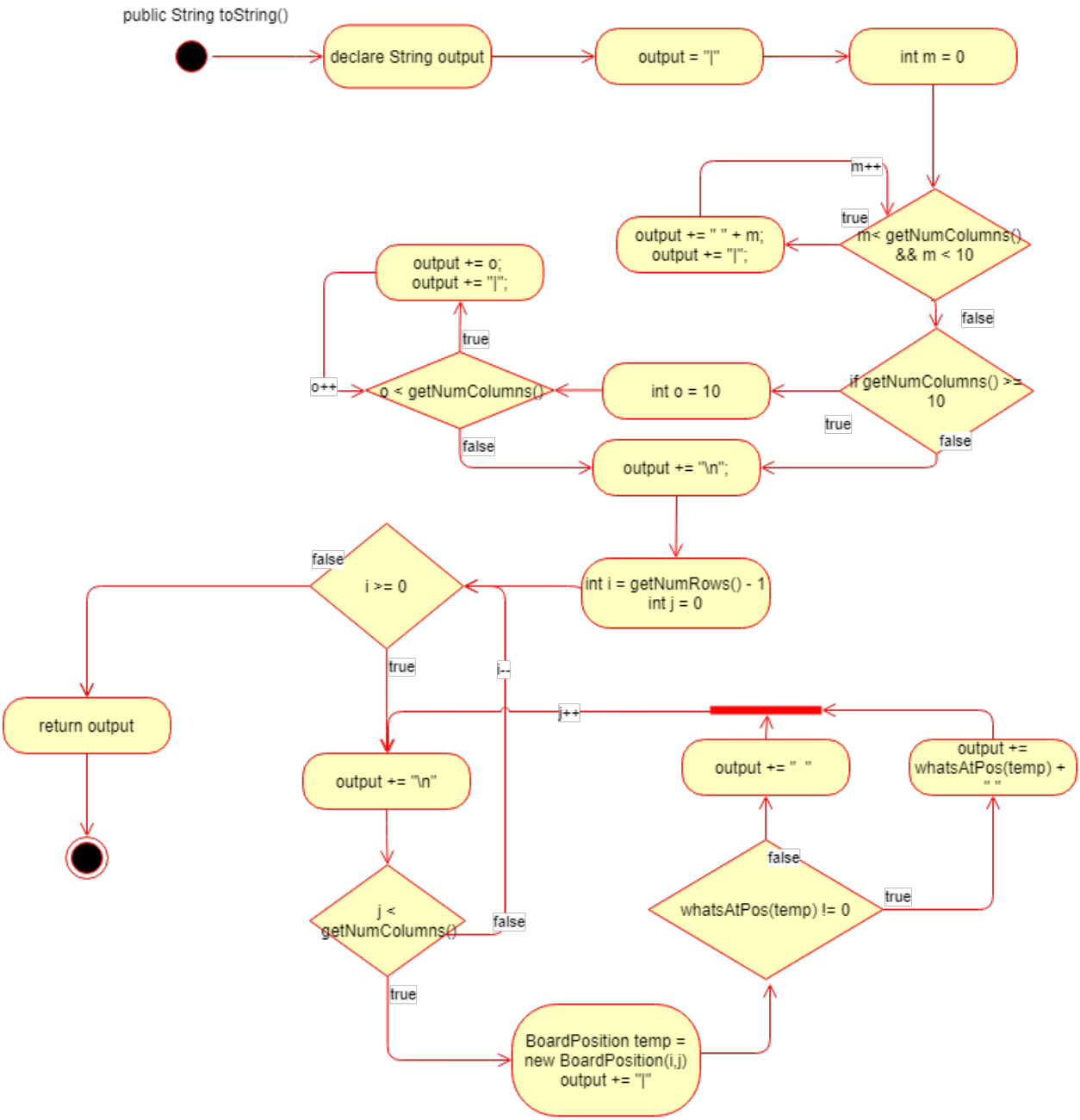
- checkDiagWin method



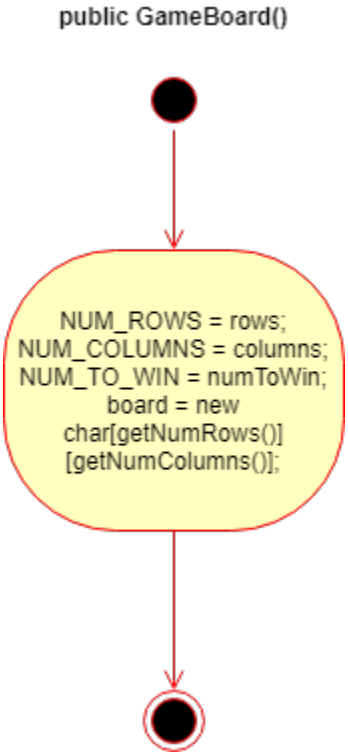
- isPlayerAtPos method



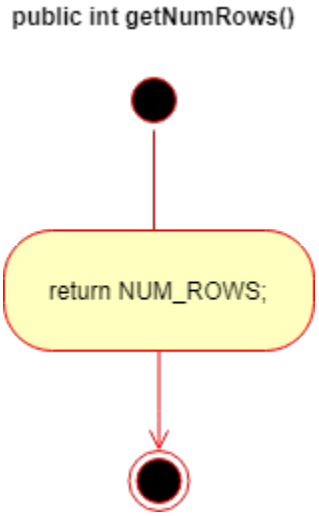
- AbsGameBoard.java
    - toString method



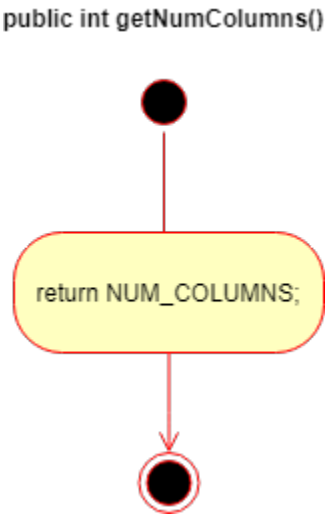
- GameBoard.java
    - GameBoard constructor



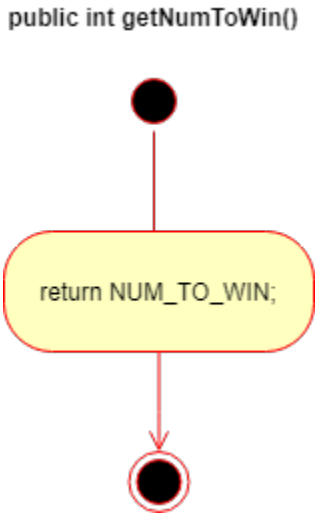
    - getNumRows method



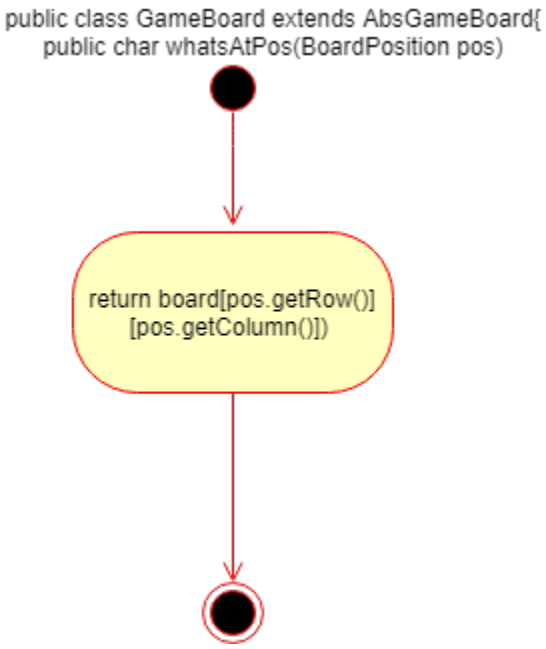
- getNumColumns method



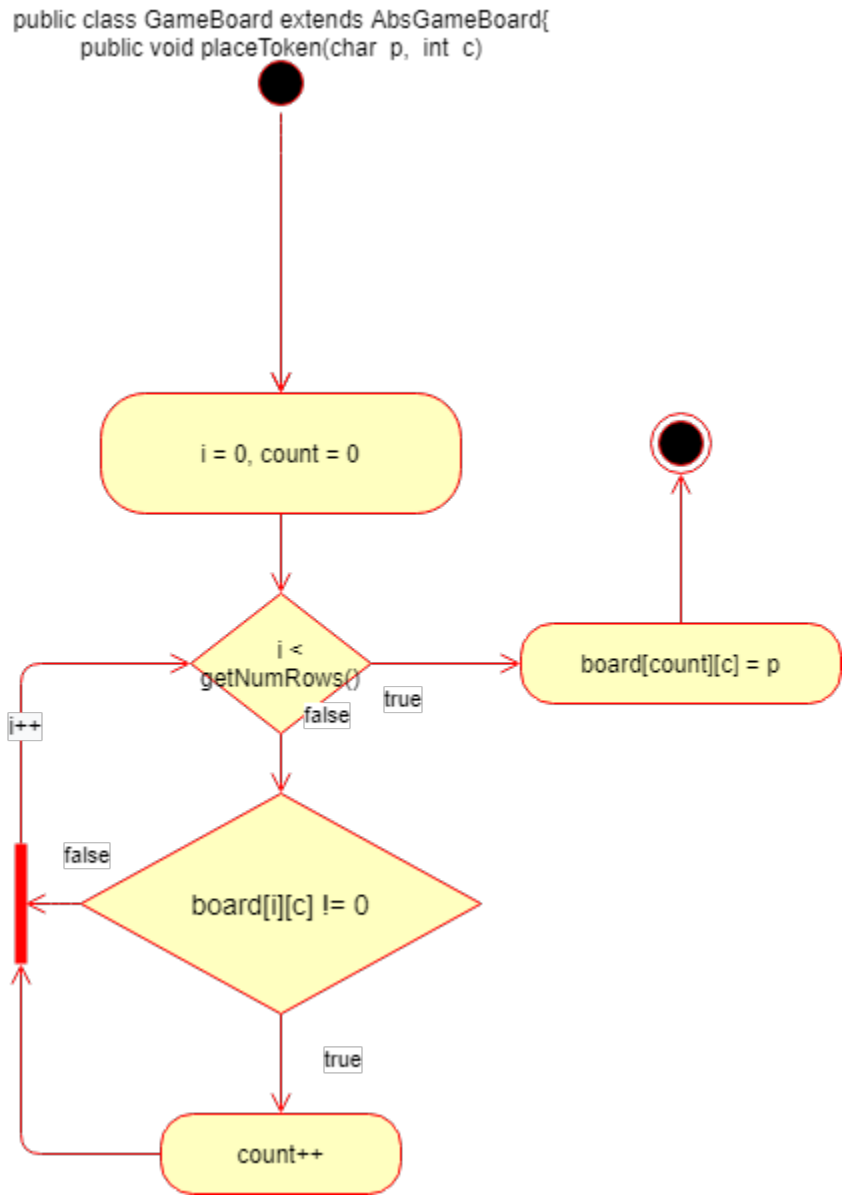
- getNumToWin method



- whatsAtPos method

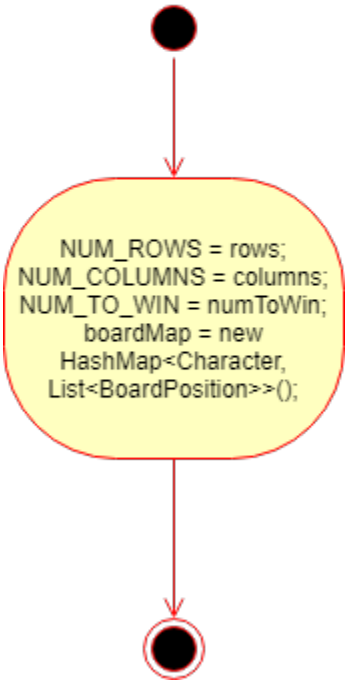


- placeToken method



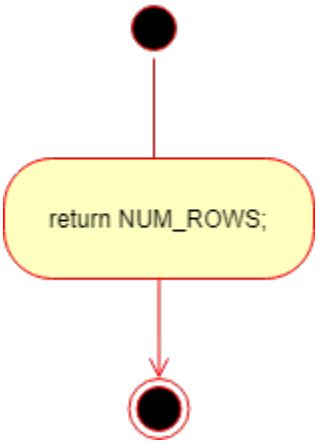
- GameBoardMem.java
    - GameBoardMem constructor

public GameBoardMem()

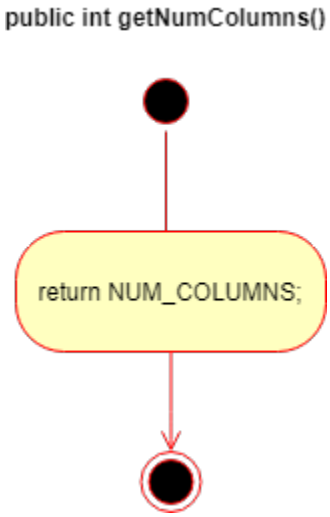


    - getNumRows method

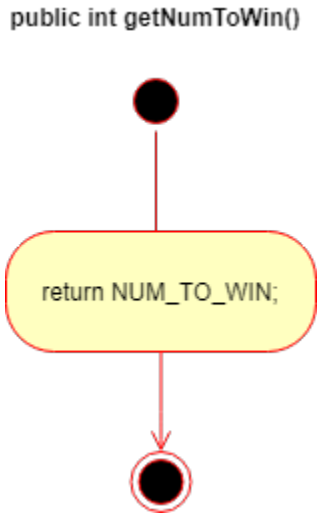
public int getNumRows()



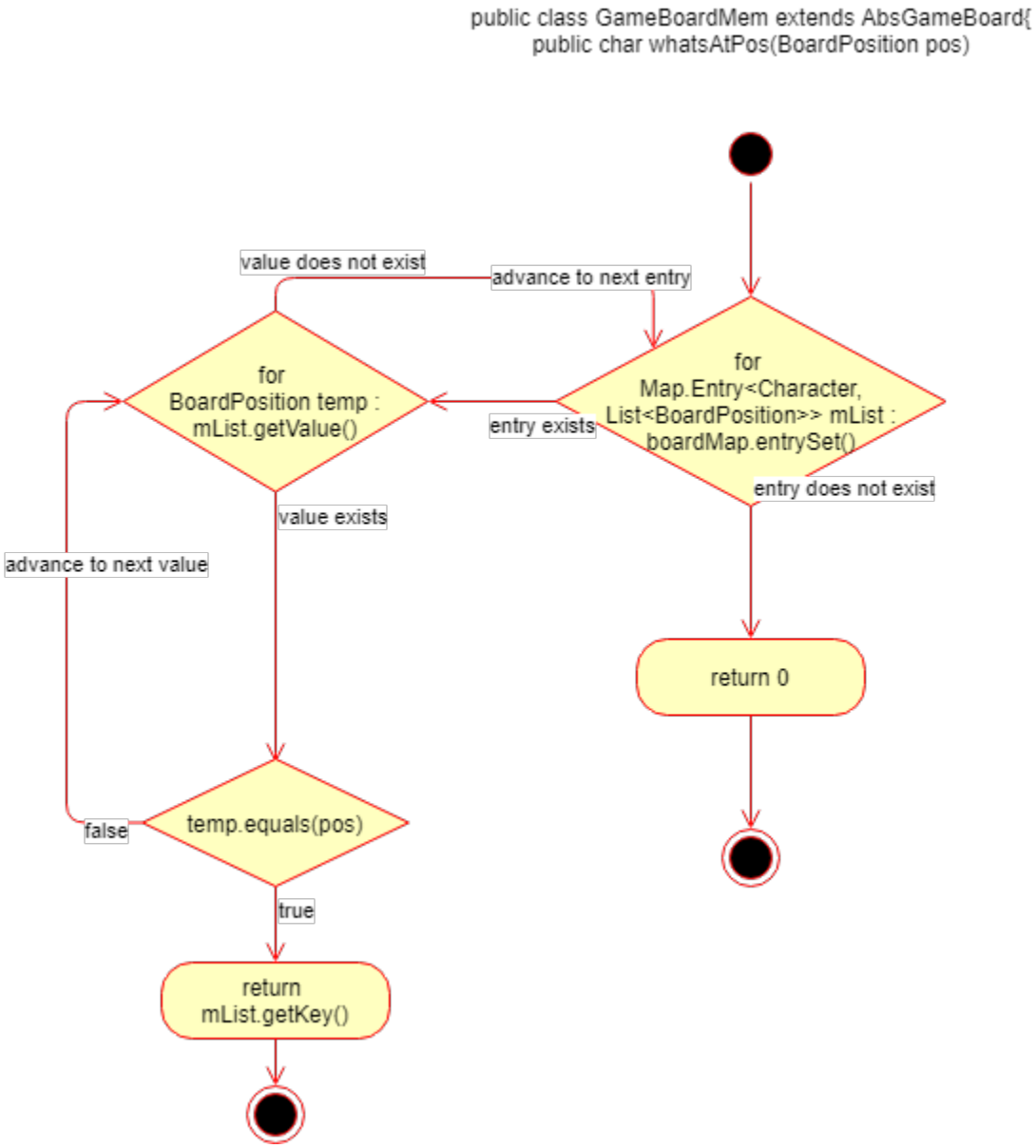
- getNumColumns method



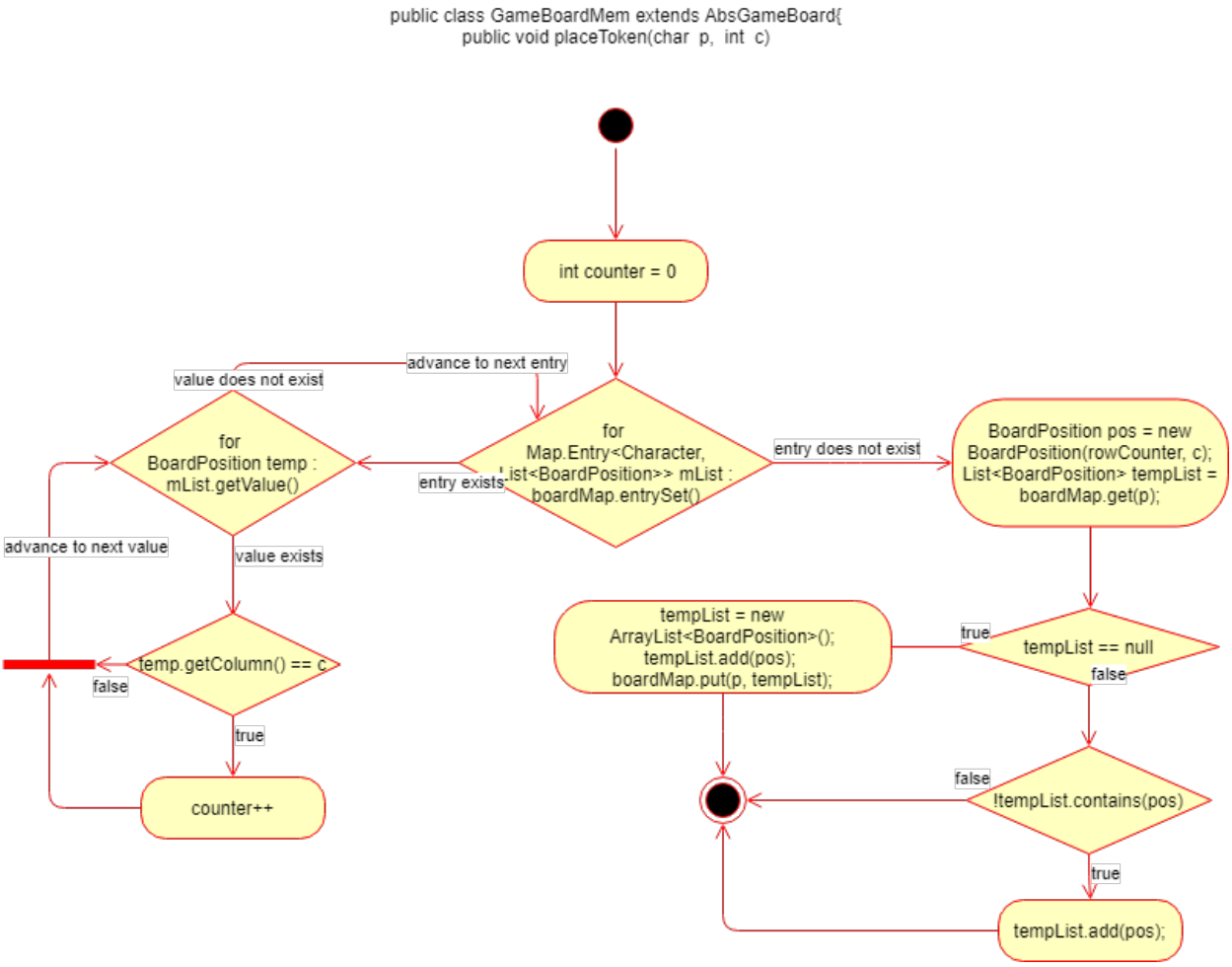
- getNumToWin method



- whatsAtPos method



- placeToken method



- isPlayerAtPos method