

# CPSC 2150 Project Report

Joel Miller

## Requirements Analysis

### Functional Requirements:

1. Enumerated list of functional requirements
2. **PLAYER**
  - As a game player, I want to be able to play against an opponent {O} and I will be {X}
  - As a game player, I want to be able to win if I have 5 {O} in a row[Diagonal,vertical,horizontal]
  - As a game player, I want to be able to have the choice where to play my {O or X}
  - As a game player, I want the game board to be a 9X7
  - As a game player, I want a column to be full once it has 9 tokens in it
  - As a game player, I want the game to inform the player who won or tied
  - As a game player, I want the game to ask if we want to play again at the end of the game

### Non-Functional Requirements

1. Enumerated list of Non-functional requirements
  - The game Code must be written in Java
  - The game must be compatible with a command line interface
  - The game must have three classes
    - GameScreen.java/BoardPosition.java/GameBoard.java
    - **Gamescreen.java**
      - Must contain the main method
      - This class will interface with the player{It will alternate the game between players, say whose turn it is, get the column they would like, and place their marker}
    - **BoardPosition.java**
      - Will keep track of the row position and column position
      - Only 1 constructor which takes in an int for row and an int for column
      - Must also have some getter functions getcolumn and getrow
    - **GameBoard.java**
      - All attributes must be private
      - Functions must be made public
      - Each position will have a blank character



## BoardPosition

-Row:int

-Column:int

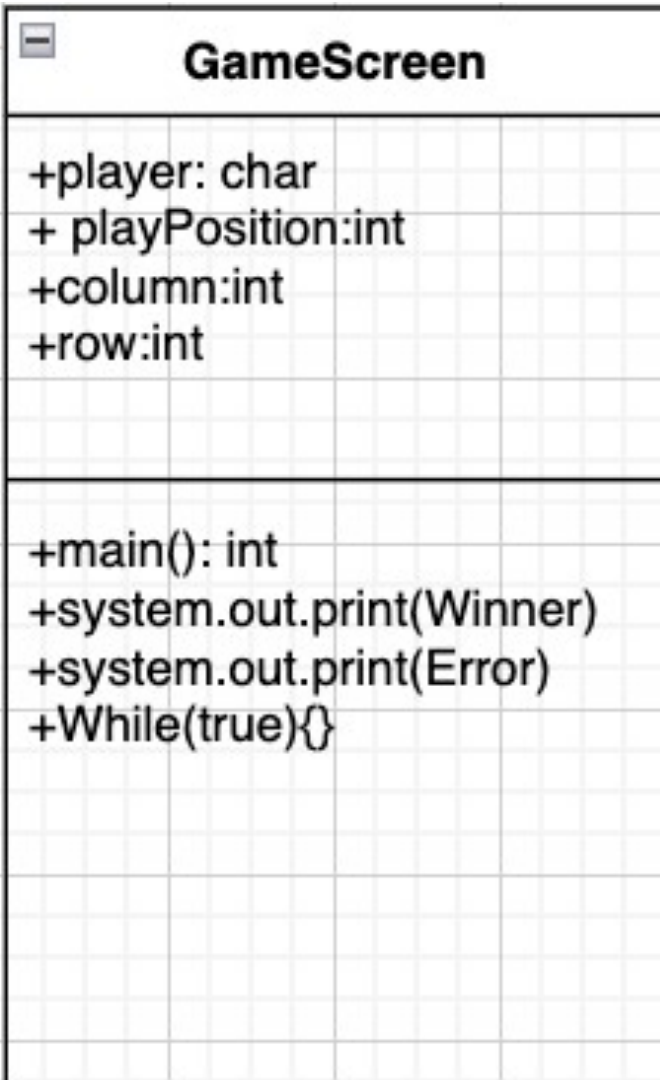
+BoardPosition(int ro, int colu)

+getRow():int

+getColumn():int

+equals() override

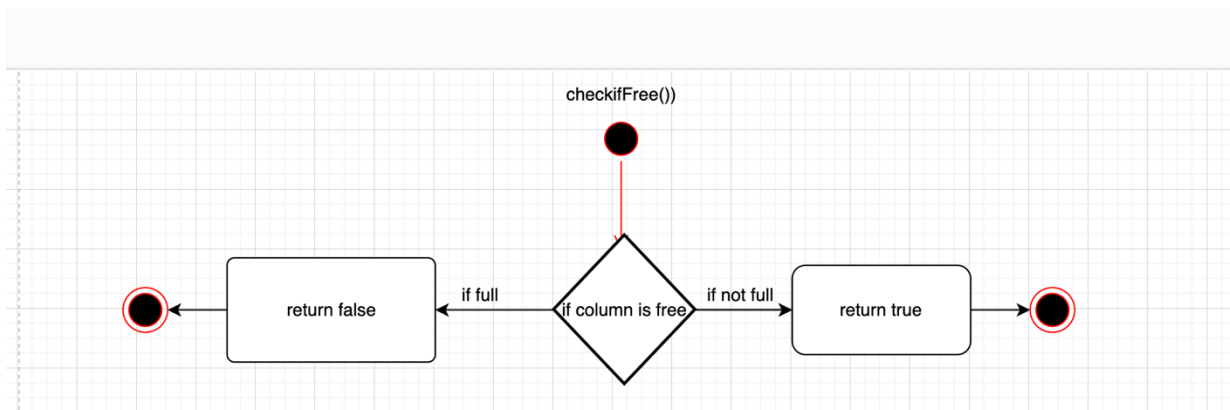
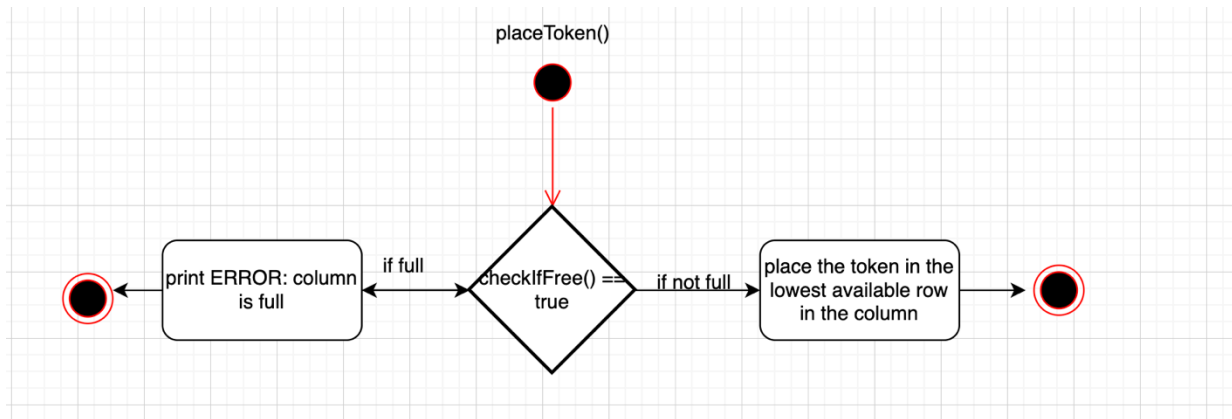
+toString() override

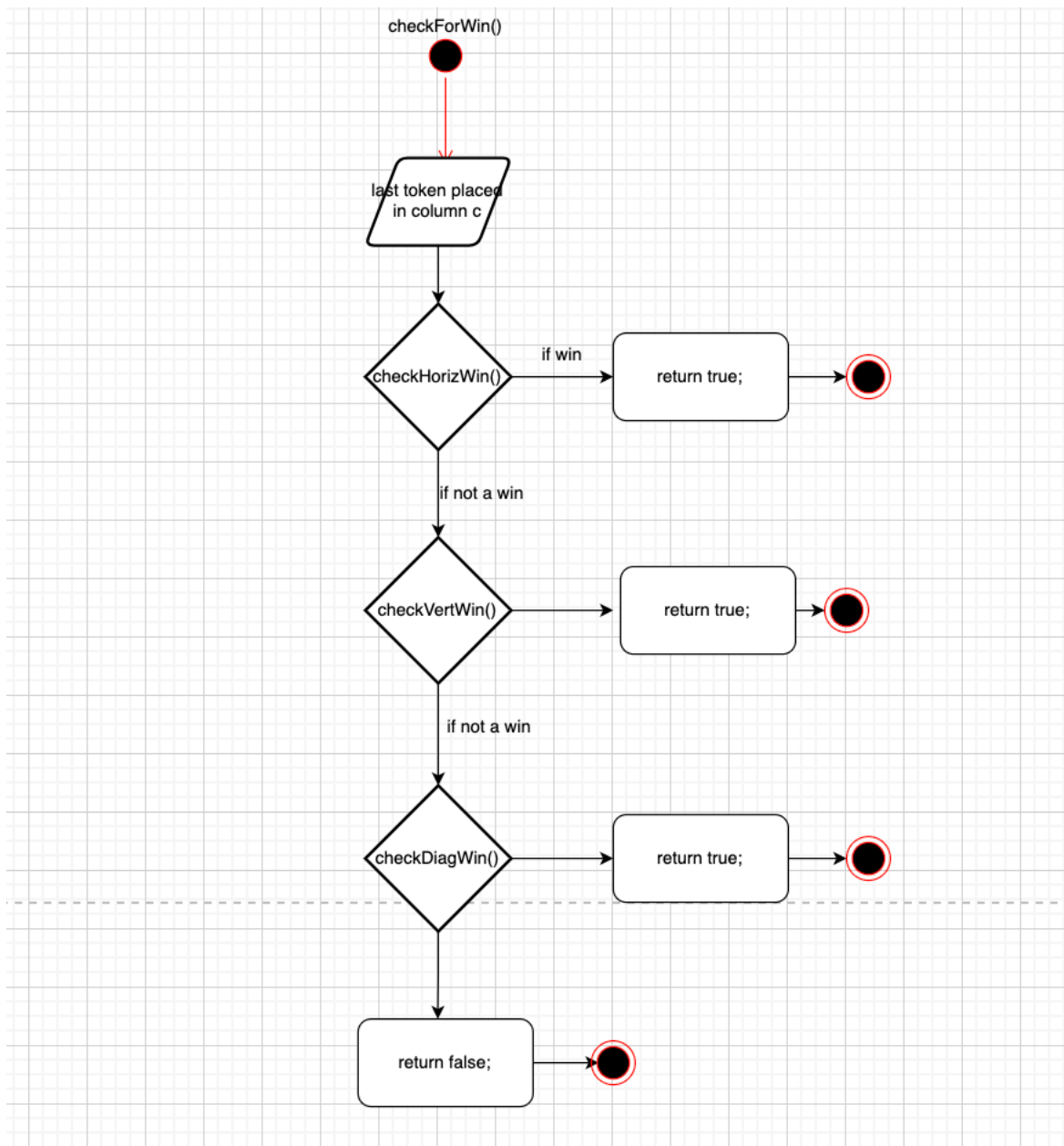


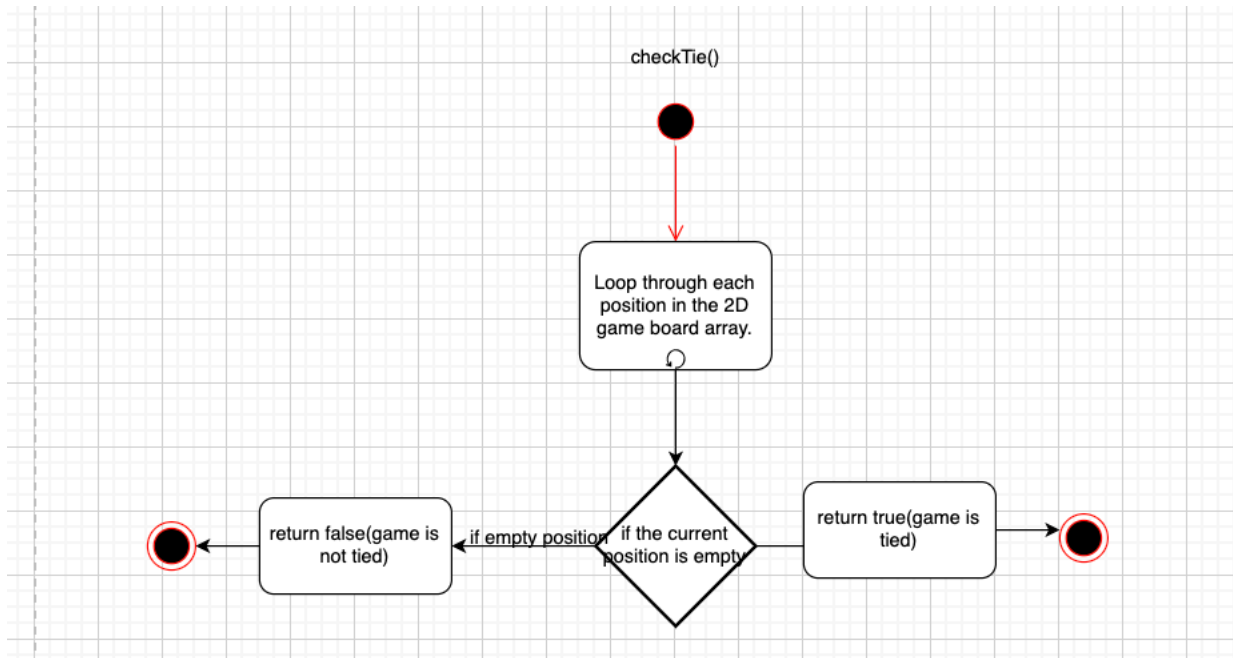
## GameBoard

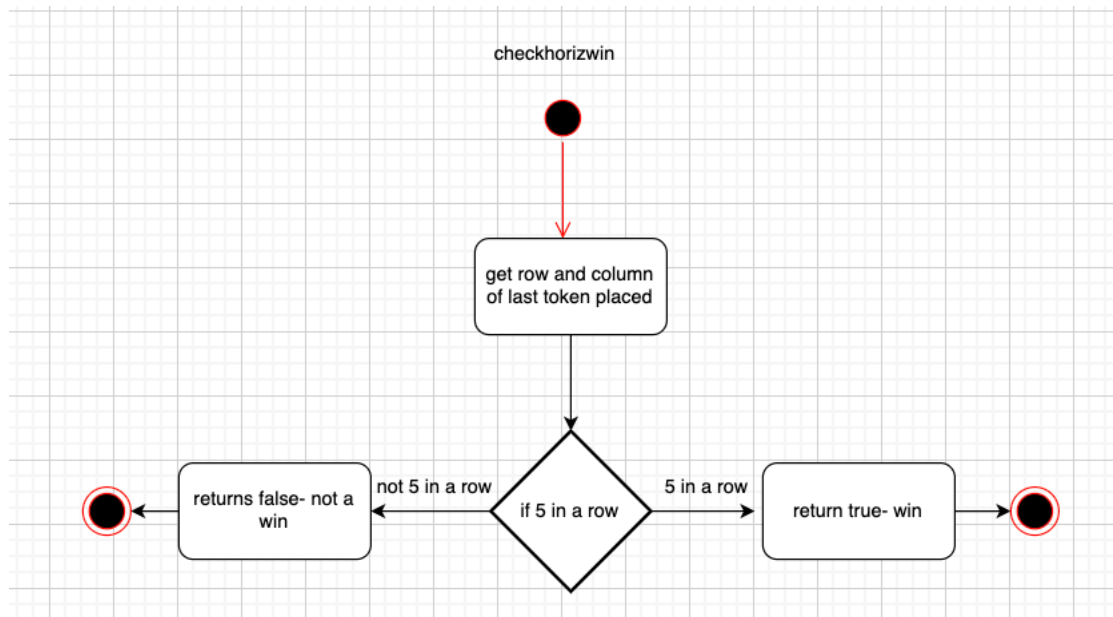
```
-BoardPosition pos;  
-p:char  
-player:char
```

```
+GameBoard()  
+checkIfFree(int c):bool  
+placeToken(char p,int c):void  
+checkForWin(int c): bool  
+checkTie():bool  
+checkHorizWin(BoardPosition pos, char p): bool  
+checkVertWin(BoardPosiition pos, char p):bool  
+checkDiagWin(BoardPosition pos,char p):bool  
+whatsAtPost(BoardPosition pos): char  
+isPlayerAtPos(BoardPosition pos, char player):bool  
+toString();
```

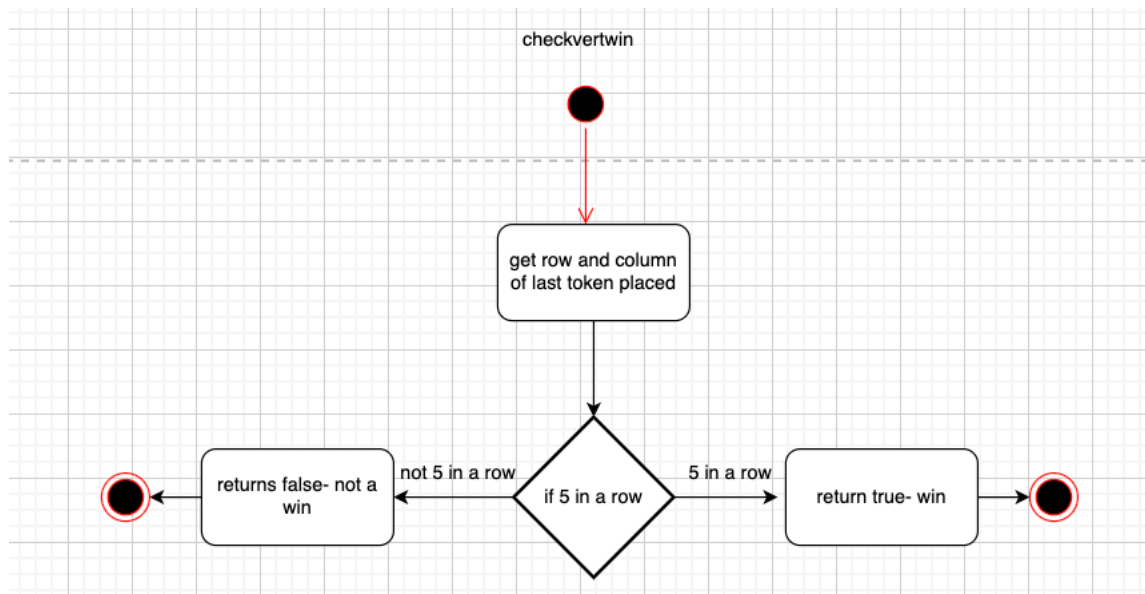


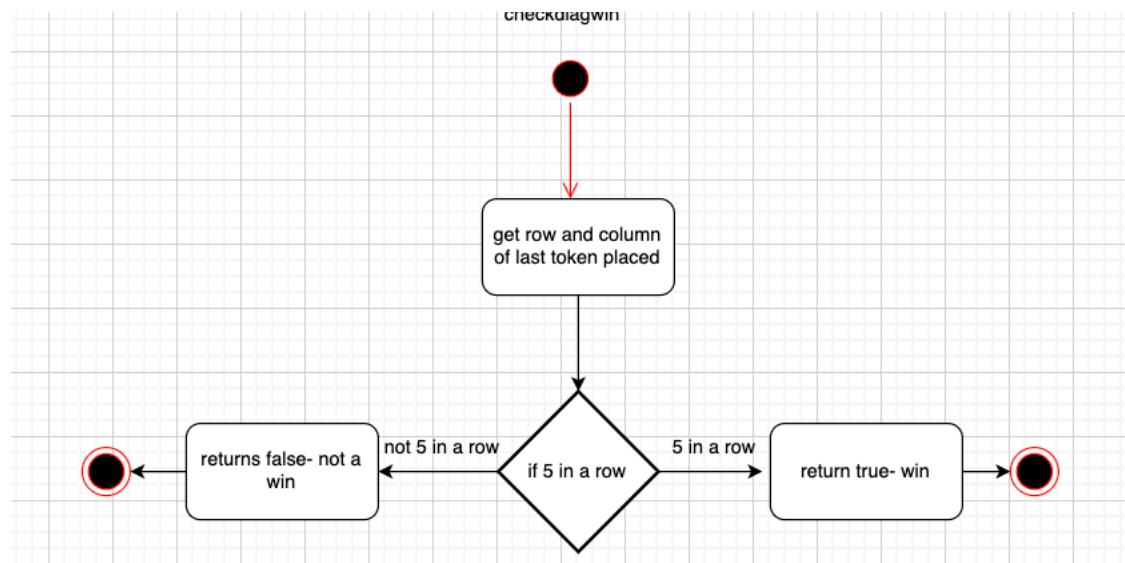


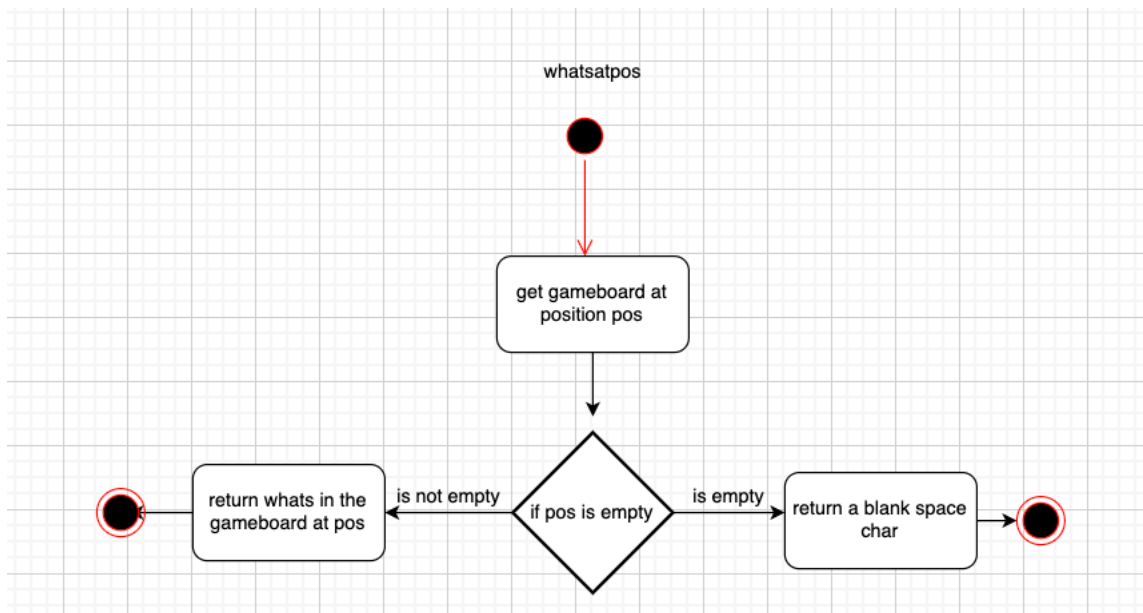


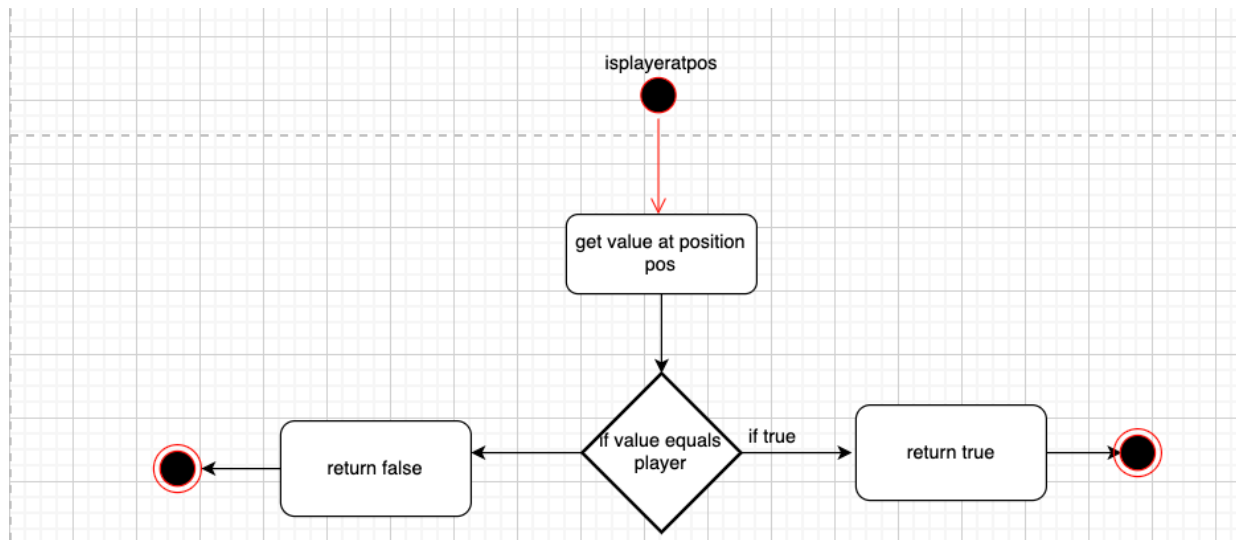




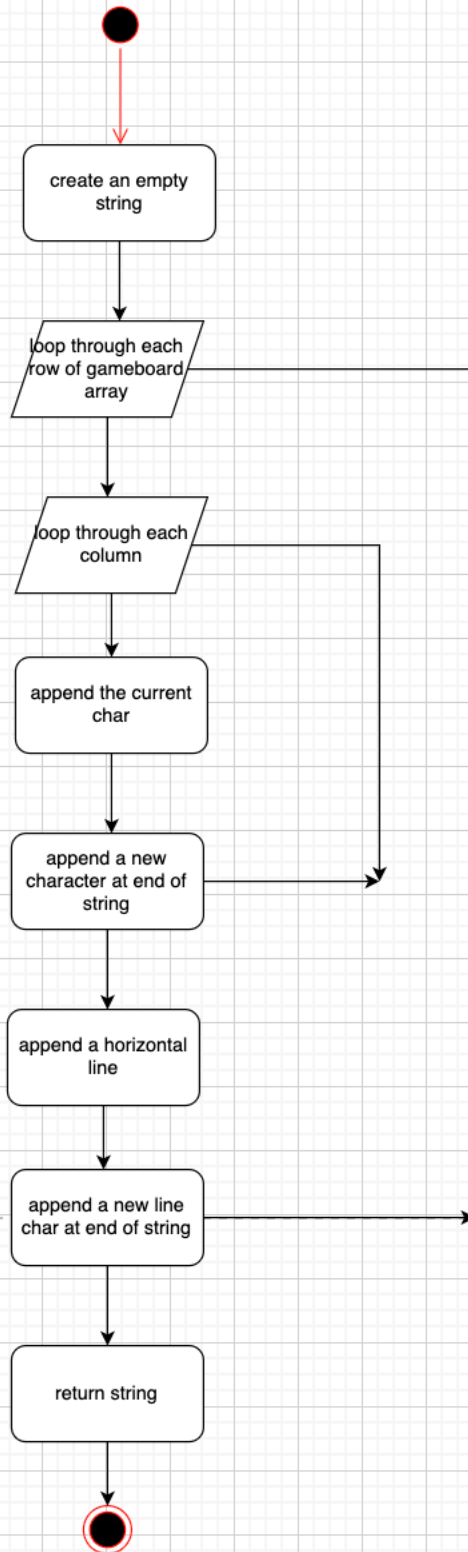


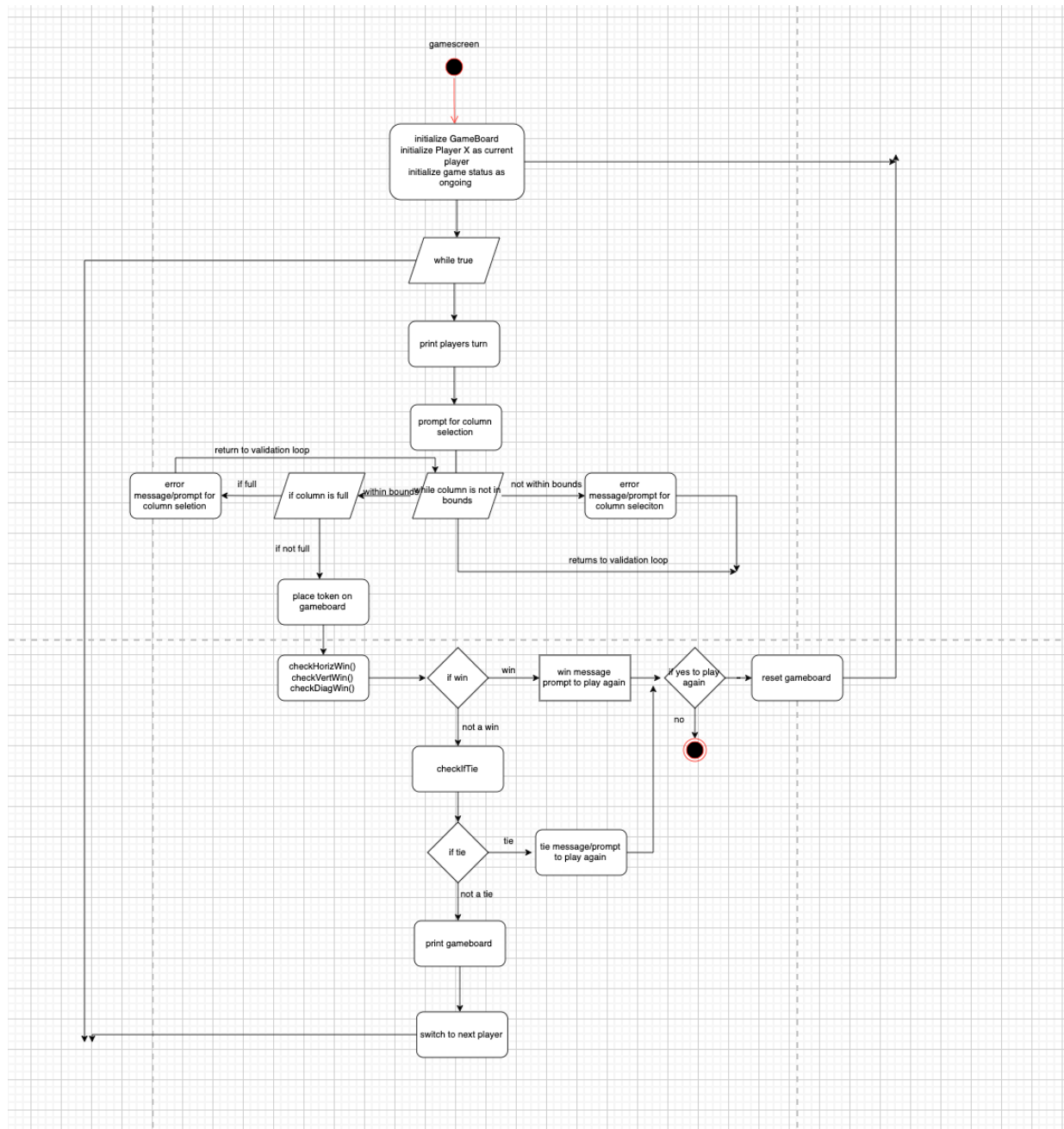






AbsGameBoard







## IGameBoard

- + checkIfFree(c:int):bool
- + placeToken(p:char, c:int)
- + checkForWin(c:int):bool
- + checkTie():bool
- + checkHorizWin(pos:BoardPosition, p:char):bool
- + checkVertWin(pos:BoardPosition, p:char):bool
- + isPlayerAtPos(pos:BoardPosition, player:char):bool
- + getRow():int
- + getColumn():int
- + equals(obj:Object):bool
- + getNumRows():int
- + getNumColumns():int
- + getNumToWin():int

