Requirement Analysis:

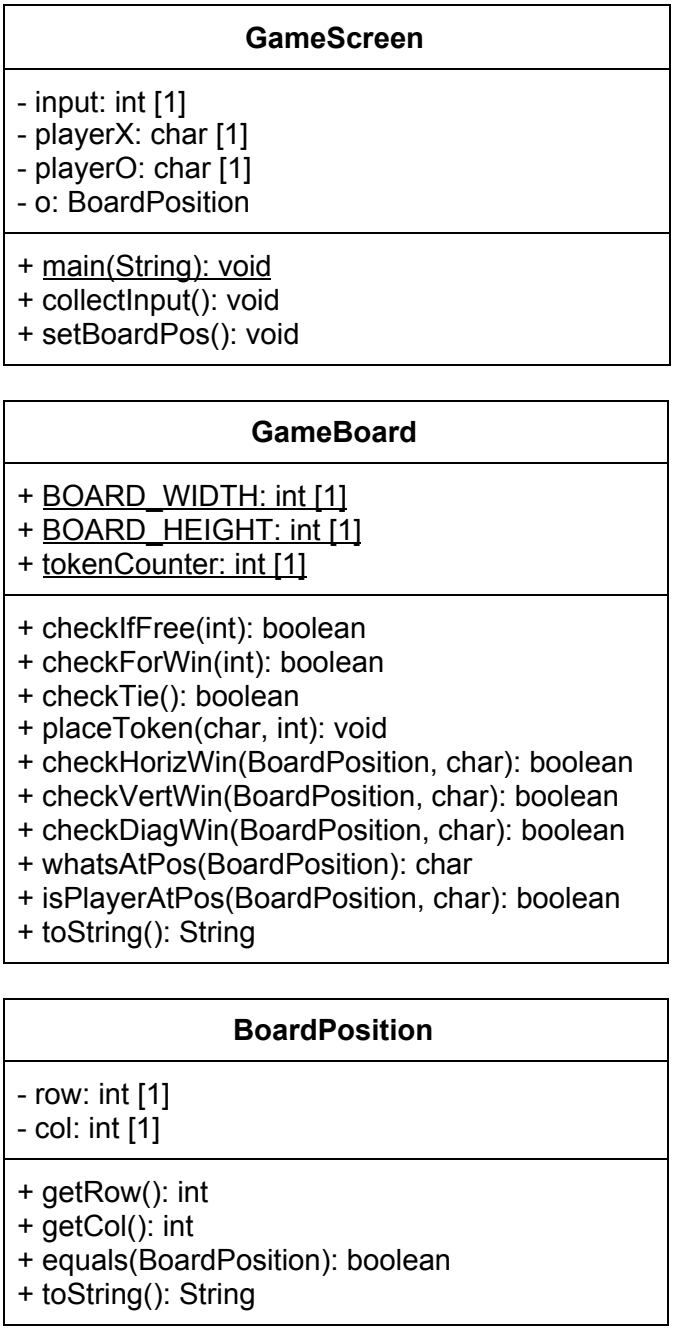
Functional Requirements:

- As a player, I can be assigned player X or O to determine which tokens I will use and who gets the first move.
- As a player, I can place a token in a chosen column to try and connect 5 horizontally, vertically, or diagonally.
- As a player, I can decide whether or not to play again at the end of a game by typing ‘y’ or ‘n’.
- As a player, I have access to seeing the game board to determine my next token placement.
- The game must accept column integer input from the user.
- The game must check to make sure that a column is not full when a player tries to place a token.
- The game must check to see if a player has won by connecting 5 tokens in a row horizontally, vertically, or diagonally.

Non-Functional Requirements:

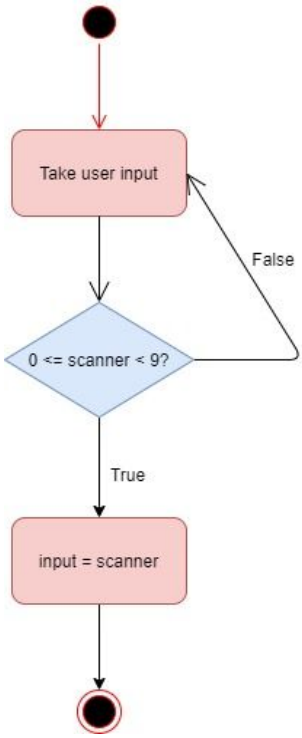
- The game executable must run on Unix.
- The game code must be able to compile in Java 11.
- The game code must be able to run in Java 11.

UML Class Diagrams:

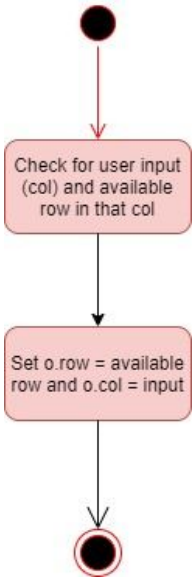


UML Activity Diagrams:

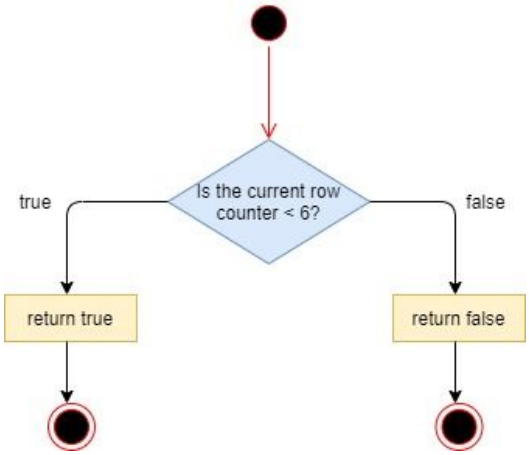
collectInput(): void



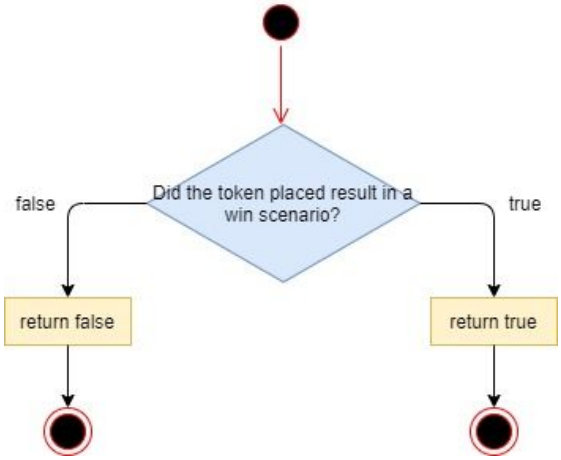
setBoardPos(): void



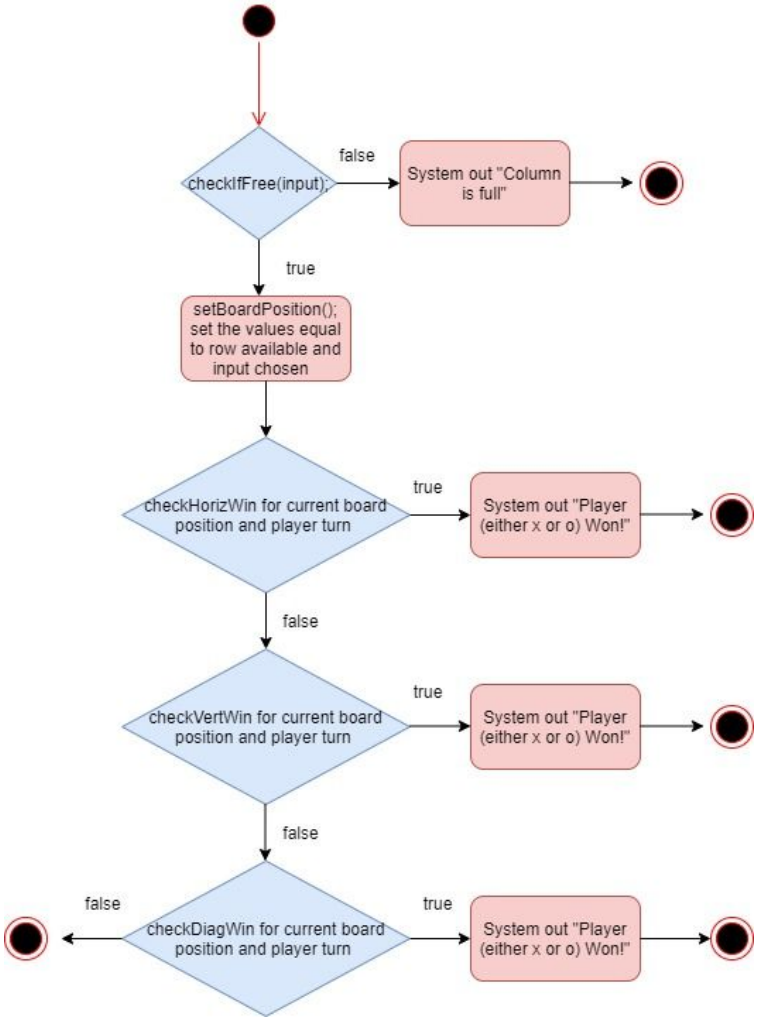
checkIfFree(int): boolean



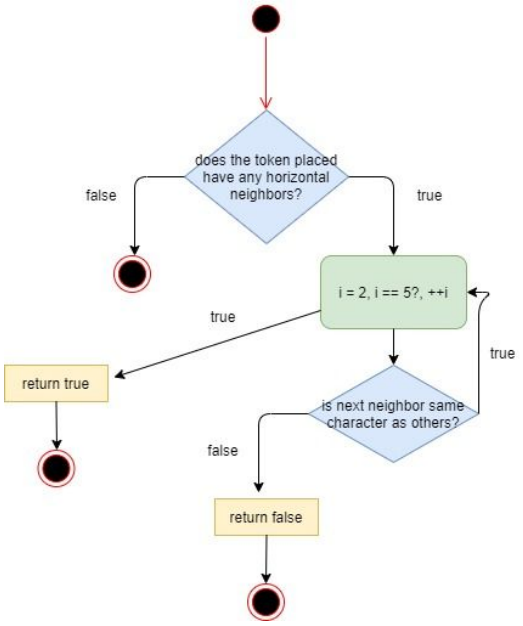
checkForWin(int): boolean



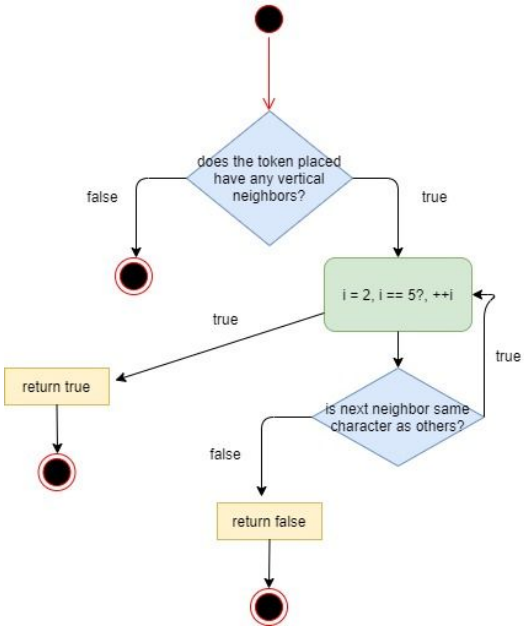
placeToken(char, int): void



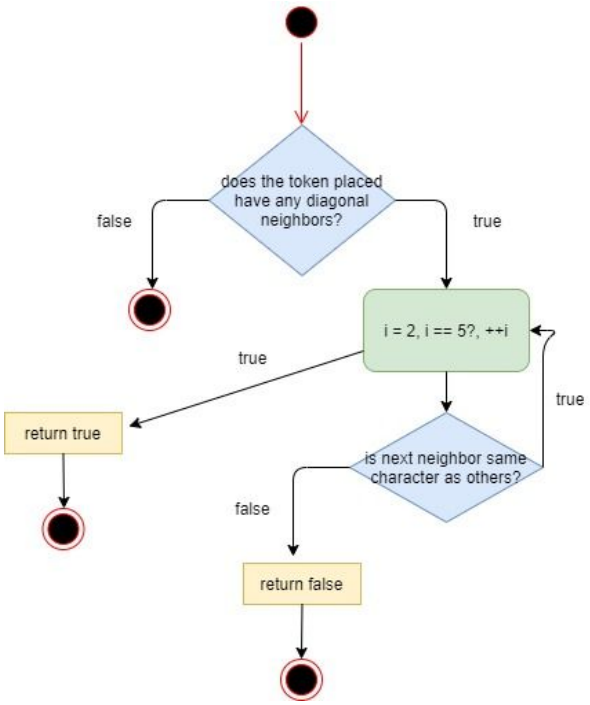
checkHorizWin(BoardPosition, char): boolean



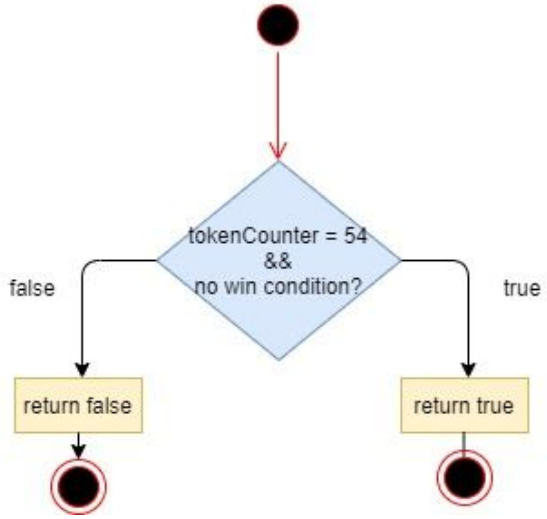
checkVertWin(BoardPosition, char): boolean



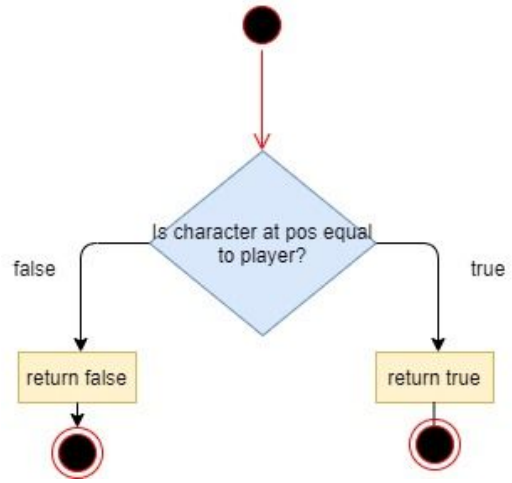
checkDiagWin(BoardPosition, char): boolean



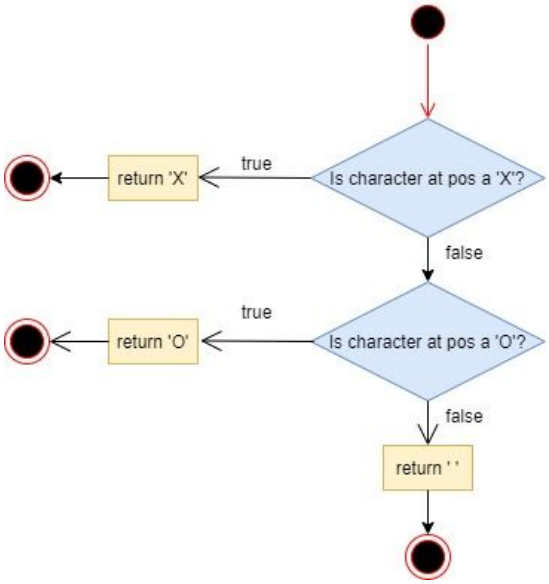
checkTie(): boolean



isPlayerAtPos(BoardPosition, char): boolean



whatsAtPos(BoardPosition): char



toString(): String