Take Home Problem

The TicTacToe game

A Command Line Game to Play a game of TicTacToe with the computer.

Overview

1. As a user I want to play a game of tic tac toe (naughts and crosses) with the computer
2. Each time I run the program I start a new game of tic tac toe and show a cursor for user input. For example

   tictactoe>
3. At the beginning of the game the computer randomly chooses whether the user is X or O. If the user is X he can start otherwise the computer will start.
4. After every move (user's or computer's) the board status should be printed like below
   a. **| O | | X |**
      | X | X | O |
      | O | | X |
5. To make a move the user should type "move" followed by the coordinates of the square on which he wishes to place his move separated by ','. 0,0 is the top left corner and 2,2, is the button right corner and so on. For example

   tictactoe> move 0,0
6. If its an illegal move, details should be printed and the user's input should be requested again. Example:

   tictactoe> move 0,0
   Invalid move, place already taken
   tictactoe>
7. The computer should choose its move randomly. **No need to make the moves intelligent in any way.**
8. After a move is made, if user or computer wins the final board should be printed and the name of the winner should be revealed as so
   a. **| O | | O |**
      | X | X | X |
      | O | | X |
      User wins
   b. **| O | | O |**
      | X | X | X |
      | O | | X |
      Computer wins
9. If the game is a draw (i.e no valid move can be made by any user) The output should be "**The Game is a Draw**"
10. If user types 'quit' the game should end. In this case the computer is said to have won.
11. **Bonus Feature**

a. If the user types ‘scoreboard’ the game should show the scoreboard, how many times user has won and how many times computer has won. For example

```
tictactoe> scoreboard
| Computer | User |
| 3        | 4    |
```

Constraints

1. Do not use a database or any external infrastructure. The code should run on any linux laptop.
2. The submission should be uploaded to a public github repository and the link should be provided to us.
3. The code must be written from scratch in the repo. We are looking for multiple commits. As you add each piece of code, commit and push it to the repository.
4. We expect clean maintainable code, divided into components and having small functions.
5. DO NOT WRITE THE WHOLE CODE IN ONE FUNCTION.
6. Tests are an added bonus.
7. Please keep in mind that you will be asked to extend the code in the interview.
8. You can use any language to write the application.
9. Please add a README which explains how to setup.