TIC TAC TOE 2.1

Abstract

If you are not acquainted with the game Tic Tac Toe, here you can find the rules for the classical game: Rules for classical Tic Tac Toe: https://en.wikipedia.org/wiki/Tic-tac-toe

We want to bring the pen-and-paper classical game to the digital age, but with a little twist: the size of the playfield should be configurable between 3x3 and 10x10. And we also want the symbols (usually O and X) to be configurable. Also, it should be for 3 players instead of just 2. A player can win the game by filling in a whole row, column or diagonal. If the playfield is 5x5 - then the player must fill all the 5 cells in a row, column or diagonal to win.

The three players play all together against each other. One of the players is an AI. Who is starting is random. In and output should be on the console. Input of the AI is automatic, no user action should be required. After each move, the new state of the playfield is displayed and the player can enter the next position of their character one after another. The next position should be provided in a format like 3,2. Invalid inputs are expected to be handled appropriately.

Requirements

- Use the programming language you feel most comfortable with.
- Size of the playground. Valid values are between 3 and 10. This must be configurable via configuration file.
- Player use a single character to mark their moves on the board. For example, **X**, **O** and **C**. This must be configurable via configuration file.

Guidelines

- Software design (Clean Code, SOLID) is more important than a highly developed AI
- You may use external libraries only for testing or building purposes e.g. JUnit, Gradle, Maven, Rspec, Rake, GulpJS, Mocha, etc.
- If you are coding in JavaScript - you can use node.js as your runtime environment, the same way you can use JRE, if you are coding in Java, etc. You can use the provided standard APIs within these platforms.
- You cannot use simple libraries like Apache StringUtils, or complex frameworks like Spring MVC, Spring Boot, Django, React, Angular, etc.
- Please provide an explanation how to run your code
- Please explain your design decisions and assumptions
- Don't include executables* in your submission.
- Please write your solution in a way, that you would feel comfortable **handing this over to a colleague** and deploy it into production.
- We especially look at design aspects (e.g. OOP) and if the code is well tested and understandable.

* excluding shell scripts for building or running your software. However, the software itself must be source code only.