TIC TAC TOE 2.0

We want to bring the pen-and-paper game Tic-tac-toe 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 playfiled is 5x5 - then the player must fill all the 5 cells in a row, column or diagonal to win.

General Rules: https://en.wikipedia.org/wiki/Tic-tac-toe

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:

- Do not use external frameworks and libraries, i.e. Spring, Django, React JS, Vue JS, Angular JS, etc.
- You can use libraries only for testing or building purposes: e.g. JUnit, Gradle, Rspec, Rake, GulpJS, etc.
- Software design is more important than a highly developed AI

- Use the programming language you feel most comfortable with.

- Configuration:

  - Size of the playground - always a square. Valid values are between 3 and 10.

  - Play character 1, 2 and 3: A single character for the first human player, A single character for the second human player - A single character for the computer.

  - These configurations should come from a file

Rules:

- 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.

** this includes: asp, bat, class, cmd, com, cpl, dll, exe, fon, hta, ini, ins, iw, jar, jsp, js, jse, pif, scr, shs, sh, vb, vbe, vbs, ws, wsc, wsf, wsh & msi*