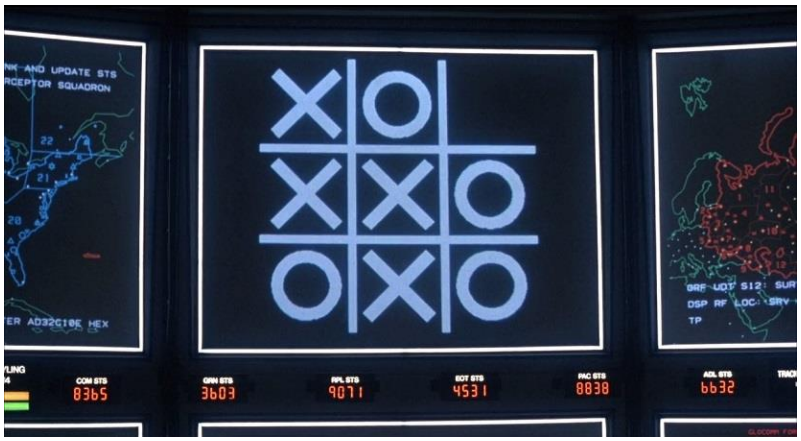
ASE Exercise 8 (Christmas 2021)

"Shall we play a game?"

In the famous 1983s movie "War Games" an artificial intelligence (called WOPR **W**ar **O**peration **P**lan **R**esponse), controlling the nuclear missiles of the U.S., learns, that a nuclear war is unwinnable by playing the game "tic-tac-toe" (also known as "noughts and crosses" or "Xs and Os").



The game tic-tac-toe is a turn-based game played on a 3x3 grid by two players. Each player fills one field of the grid with his symbol (either X or O) when it is his turn. One player wins the game, if he succeeds in filling one horizontal, vertical, or diagonal line on the gird.

Example of four rounds, Player X wins:

IMA - RWTH Aachen University
Institute of Information Management in Mechanical Engineering IMA

Task:

Write a Java program that enables the user to play the tic-tac-toe game against the computer. When the program is started, the user is asked, if he wants to play a game:

```
Shall we play a game?
```

The user might answer that question with `yes` or `no`. In case the user inputs `no`, the program ends. In case of `yes`, the user is asked for the difficulty level of the computer player:

```
Choose difficulty:
```

The user might input `easy` or `hard`. Depending on his choose, the computer uses a different strategy to play the game (more on that later). The game then starts by printing the empty grid to the screen and the player is asked to make his move:

```
   A B C
1  _|_|_
2  _|_|_
3   |  |

Player, make your move:
```

The player then makes his move by entering: `B2`
The players move gets saved (the player uses the Xs) and the computer immediately does his move (using the Os) to, for example, `C3`. Afterwards the new grid is printed to the screen.

```
   A B C
1  _|_|_
2  _|X|_
3   |  |O

Player, make your move:
```

And so on…

The game ends, as soon as one player wins or if no further moves are possible.

In case the computer wins, the program prints out:

```
Sorry, you lose!
```

In case the player wins, the program prints out:

```
Congratulations, you win!
```

In case of no more possible moves (draw), the program prints out:

```
A STRANGE GAME.
THE ONLY WINNING MOVE IS NOT TO PLAY.
```

In any case, the user is asked to play another round and the program starts from the beginning.

```
Shall we play a game?
```

Your task is to implement this tic-tac-toe application and two different computer opponents, each playing with a different strategy.

1. The first computer opponent, the *easy* one, will just make random, but valid moves on the grid.
2. The computer opponent of difficulty *hard* will try to win the game. Therefore, it tries to find the best possible valid move based on the current grid situation. That might be to defend a players move (e.g. the player already has two in a row, the computer has to set his move to the third field in that row) or to try to win himself (by achieving a row of three if a defense is not needed).
   Think of useful rules and measures of the current game status to accomplish that.

Additional information:
Of course, the human player and the computer opponent can only do valid moves, which means, that only an empty field can be chosen!
If the human player choses a field, that is already filled with an X or an O, he will be asked to make his move again.