Description

Let's write the "hard" level difficulty.

Compared to the "medium" level difficulty, this level not just go one move ahead to see an immediate win or prevent an immediate loss. This level can see two moves ahead, three moves ahead and so on. Basically, it can see all possible outcomes till the end of the game and choose the best of them considering his opponent also would play perfectly. So, it doesn't rely on the blunders of the opponent, it plays perfectly regardless of the opponent's skill.

The algorithm that implements this is called Minimax. This is the brute force algorithm that maximizes the value of the own position and minimizes the value of the opponent's position. It's not only an algorithm for Tic-Tac-Toe, but for every game with two players with alternate move order, for example, chess. You need to implement it as the "hard" difficulty level. This link will help to understand details.

You also should add "hard" parameter to be able to play against this level.

Example

The example below shows how your program should work.

```
Input command: start hard user
Making move level "hard"
---------
|       |
| X     |
|       |
---------
Enter the coordinates: 2 2
---------
|       |
| X O   |
|       |
---------
Making move level "hard"
---------
|   X   |
| X O   |
|       |
---------
Enter the coordinates: 2 1
---------
|   X   |
| X O   |
|   O   |
---------
Making move level "hard"
---------
| X X   |
| X O   |
|   O   |
---------
Enter the coordinates: 1 1
---------
| X X   |
| X O   |
| O O   |
---------
Making move level "hard"
---------
| X X X |
| X O   |
| O O   |
---------
X wins

Input command: exit
```