

# Artificial Intelligence: Homework #4

Assigned on 5/4/2021 (Tuesday); Due at 12:00 am on 6/8/2021 (Tuesday)

## Description:

Tic-Tac-Toe is a simple two-player game that, if played “perfectly” by both players, will always result in a tie. In other words, it is possible to implement an AI program that will never lose to a Tic-Tac-Toe game. The purpose of this assignment is to leverage the skills you have learned so far this semester to make a terminal-based Tic-Tac-Toe game. The following website will give you a feel for the game:

<https://freetictactoe.com/>

To emulate our version, select “One Player” with DIFFICULTY on the “Expert” level.

## The Rules of the Game:

1. The game will be one real player (X), with the program AI (O) taking over the second player.
2. The grid is  $3 \times 3$  consists of numbers 1 to 9 unrepeated in all cells like the following setup.

1	2	3
4	5	6
7	8	9

3. The real player (X) should pick a spot by entering the numbers 1 to 9 on the corresponding coordinates of the above  $3 \times 3$  table.
4. The real player can enter 0 at any time to exit the game.
5. The user should be prompted until the coordinates entered correspond to an empty cell. The game board should be printed out after every move made by either the player or the AI.
6. Your program should behave as follows:
  - i. The real player (X) will always go first.
  - ii. Then the AI (O) and the real player (X) alternate until one of them gets three in a row vertically, diagonally, or horizontally.
7. When it is the AI’s turn, you can randomly (easy), systematically (medium), or strategically (hard) pick an empty cell in which to place the AI’s move (O).
8. You will only receive basic scores (70%) if you choose to let AI plays randomly.

## Example Program Run:

- The input for the real player is in red. The output of your program should try to follow the format in the example. Nevertheless, you have the freedom to define your own winning, losing, and tie messages.

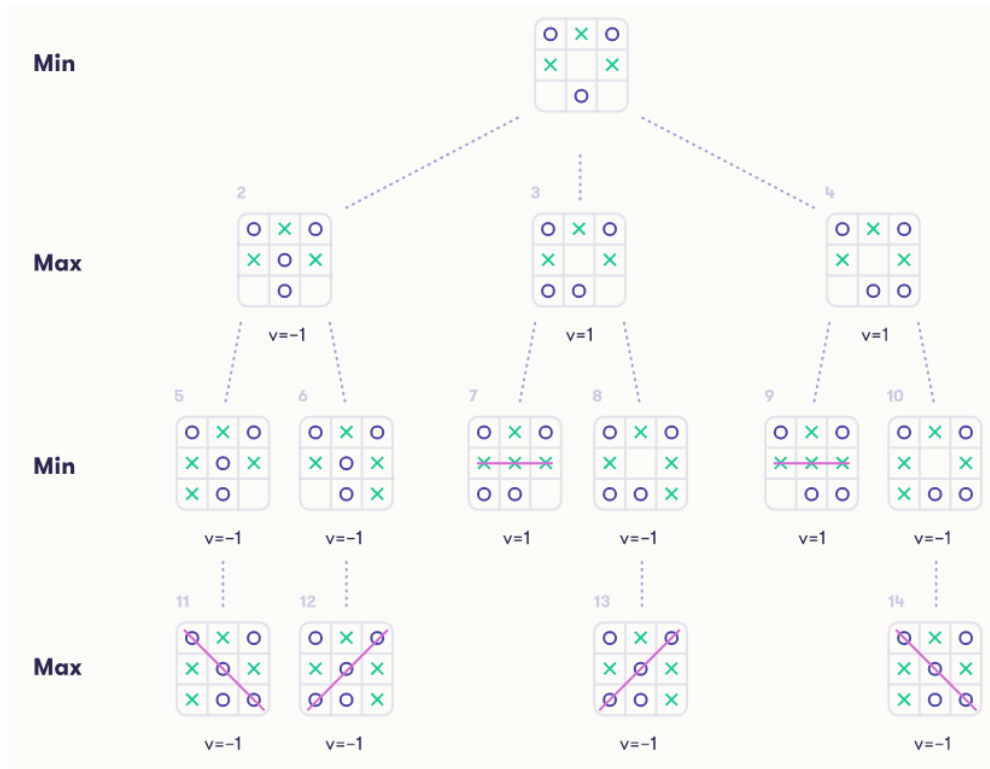
Round 0:  1 2 3   4 5 6   7 8 9  Your move: <b>5</b>	Round 4:  O 2 O   4 X 6   X 8 9  Your move: <b>2</b>	Round 8:  O X O   O X X   X O 9  Your move: <b>9</b>
Round 1:  1 2 3   4 X 6   7 8 9  AI's move: 1	Round 5:  O X O   4 X 6   X 8 9  AI's move: 8	Round 9:  O X O   O X X   X O X  The game is a tie!
Round 2:  O 2 3   4 X 6   7 8 9  Your move: <b>7</b>	Round 6:  O X O   4 X 6   X O 9  Your move: <b>6</b>	
Round 3:  O 2 3   4 X 6   X 8 9  AI's move: 3	Round 7:  O X O   4 X X   X O 9  AI's move: 4	

## Some Tips for the Implementations:

For a systematically approach, you could perhaps use some of the following heuristics:

- Make the AI player pick a random spot UNLESS they can win in that move.
- Make the AI player pick a random spot only if they cannot win and they cannot block the other player from winning.
- Add additional rules for the AI player so that they are more likely to win.

For a strategically approach, the [minimax](#) algorithm applies a backtracking strategy to help the AI player find the optimal move in each round, assuming that your opponent also plays optimally. In this algorithm, the two players are called maximizer and minimizer. The maximizer tries to get the highest score possible while the minimizer tries to do the opposite and get the lowest score possible. A game tree will be generated first and the goal states on the leaf nodes will be evaluated based on the final results of the game: 1 (win), 0 (tie), or  $-1$  (lose). A subtree of the whole game tree is shown as follows:



You are also encouraged to implement the algorithm with  $\alpha$ - $\beta$  pruning to reduce the search of the game tree. For more information, please check out this [website](#).

### Submission Requirements:

- You can use any programming language to implement this homework.
- Please zipped your source code along with a simple report (in pdf) to explain how you implement your AI player.
- Please named your zipped file in the format “{Student ID}\_{Name}.zip”, and named your report in the format “{Student ID}\_{Name}.pdf”.
- Honor code: This is an individual homework. DO NOT copy others’ work. We will use the Turnitin system to check for plagiarism. Both the plagiarist and the victim will get a 0 in this homework.