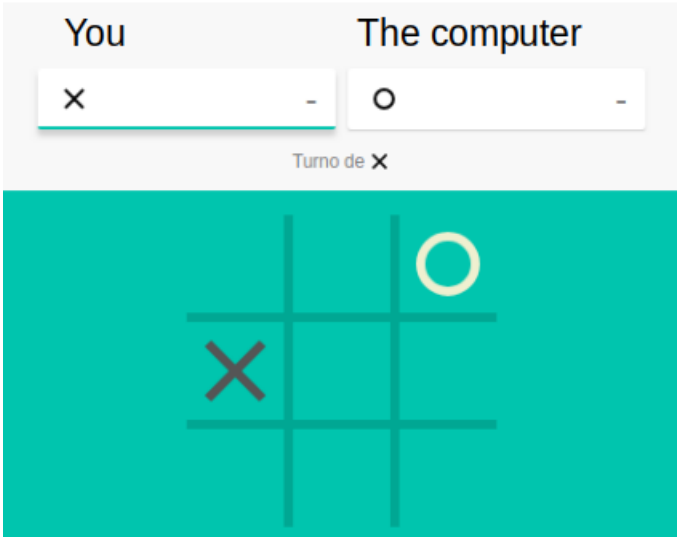
Assignment 4: Adversarial Search

Submission: Saturday 29/09/2018 23h moodle

Groups of maximum 2 students

Carolina Higuera
Introduction to Artificial Intelligence, 2018-2
/9 points

The file `tictactoe.py` is a simple implementation of the famous zero-sum game TicTacToe, knows as *triqui* in Spanish. Your task is to build an intelligent agent, the computer, to behave optimal against a smart opponent, you.



To do so, you have to implement some **depth-limited strategies** that allows the computer, the MAX player, to find a policy in the adversarial game:

1. (3 points) MINIMAX: the part for the max player is already implemented. You can use it to build the min player and the other algorithms.
2. (3 points) EXPECTIMAX: max player as in minimax. The other player behaves as a chance node, with uniform probabilities for its actions.
3. (3 points) $\alpha - \beta$ pruning.

Take into account the following useful methods:

- `board.makeMove(move, player)`: generates a successor state

- `board.availableMoves()`: legal actions in the current state

The utilities for leaf nodes in the depth-limited tree are defined by:

- `utility=0` if you, the X player wins
- `utility=100` if the computer, the O player wins
- `utility=50` other states

Submission

The python file `tictactoe.py` with the methods minimax, expectimax and alphaBetaPruning implemented.

Take into account:

- The implementation of the three strategies must be of your own, that is mandatory.
- You have to implement the strategies that are require in this assignment, no others.