CS 461 – ARTIFICIAL INTELLIGENCE

HOMEWORK #3 (5% OR 10 POINTS)

Assigned: Tue 24 Mar 2020

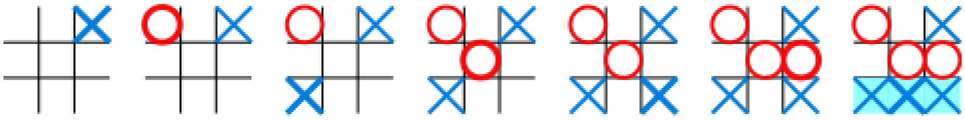
Due: Tue 7 Apr 2020 ** 2 pm **

You can do this homework in groups of 5 (or less). Your group for this homework should coincide with your term project group. In any case, do not forget to indicate clearly the students who are submitting this homework.

You must submit your entire work (including all the original code written) to our TAs. They'll tell you whether there is a need to submit hardcopy, etc.

Any programming language can be used as long as you have it available on a portable computer. Needless to say, group members should be prepared to give a homework demo (individually and using that portable computer) when requested to do so by our TAs.

The following excerpt is from Wikipedia (https://en.wikipedia.org/wiki/Tic-tac-toe):

> "**Tic-tac-toe** is a game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.
>
> The following example game is won by the first player, X:
>
> 
>
> Players soon discover that the best play from both parties leads to a draw."

In game theory, a game which permits a draw ("tie") when played properly by both players is known as a futile game. Thus, the 3x3 version of tic-tac-toe is a futile game. You wonder whether this last claim also holds for a bigger tic-tac-toe game, e.g., one which is 4x4 (instead of the classical one which is 3x3). You want to investigate the matter by way of **alpha-beta pruning** on a tic-tac-toe game tree. (But also see this mathematical approach: https://www2.stetson.edu/mathcs/wp-content/uploads/2018/01/levine-proposal.pdf)

You are free to use any existing alpha-beta pruning code, as long as you state where you've found it, whether you've made any modifications to it, etc. Needless to say, writing the code from scratch may be the best --- and clearly, the most instructive and eye-opening --- thing. Once again, it is completely up to you to reuse an existing program (with enrichments or amendations, if necessary) or write original code. If you are going to take the latter approach, you may first want to study a cleverly done algorithm animation in Dylan's AI page (http://web.mit.edu/dxh/www/adverse/index.html). You can get at most **3 points** from the code itself.

Your code is supposed to output:

- (**3 points**) Three arbitrary but complete games, as visited by your alpha-beta code. One of them should be a win for X, the other should be a win for O, and the third one should be a tie. The games' appearance (format) should be similar to the Wikipedia figure above. But don't forget: you are investigating a 4x4 tic-tac-toe game!
- (**3 points**) The virtual values of board situations immediately under the original board situation which is initially empty (the root). These values will be computed by the alpha-beta code, too.
- (**1 point**) Your verdict: Is the 4x4 version of the game also futile? Why (not)?

Without loss of generality, X is the maximizing player and O is the minimizing player. You should assume that a win for X is denoted with +1, a win for O is denoted with -1, and a draw is denoted with 0.

Your program should have a simple control for 'single stepping' (tracing your code) so that you and the TAs can inspect the intermediate stages of the problem-solving process in an incremental fashion.

GENERAL REMARKS (THESE ARE APPLICABLE TO ALL HOMEWORK ASSIGNMENTS)

- IF YOU ARE REQUESTED TO SUBMIT A HARDCOPY AT ANY TIME IN THIS COURSE, MAKE SURE THAT WHAT YOU SUBMIT IS CLEAN AND FULLY MACHINE-GENERATED. IF THERE IS A HANDWRITTEN ADDITION OR CORRECTION ON A PRINTOUT, YOU'LL DEFINITELY LOSE POINTS.
- Late submissions will first have 2 points deducted categorically. Then they'll have 2 points deducted for every late day. (A new day begins at 12:00 midnight.)