CS 461 – ARTIFICIAL INTELLIGENCE

HOMEWORK #3 (5% OR 10 POINTS)

Assigned: Wed 6 Nov 2019

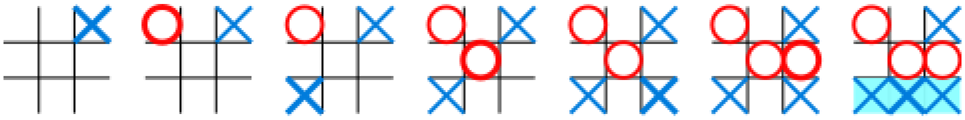
Due: Wed 20 Nov 2019 ** 2 pm **

You can do this homework in groups of 5 (or less). Your group for this homework should normally coincide with your term project group. In any case, do not forget to indicate clearly the students who are submitting this homework (i.e., write at most five names on the submission).

You must submit your entire homework (include all the original code written) to our TAs. Just a single submission per group! Our TAs will soon send you a note explaining the mechanics of submissions. They may also tell you whether there'll be 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, a group member 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[1].

Your job is to prove this last remark using **alpha-beta pruning** on a tic-tac-toe game tree. 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.

[1] A game which permits a draw ("tie") when played properly by both players is known as a futile game. Thus, tic-tac-toe is a futile game.

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).

CAVEAT: Since the complete game tree for tic-tac-toe is not ridiculously large, you may want to do this homework using ***minimaxing*** *rather than alpha-beta pruning. This is fine but then you'll lose 2 pts (1%) from whatever grade you receive from your submission. Thus, if you receive 9 pts from your submission (out of 10), your eventual grade will in fact be 9 - 2 = 7 pts. (And for minimaxing, again you may write your own code or use an existing program, provided that you give the necessary explanations.)*

Regardless of the approach you've employed, print:

- at least one complete game (computed by your program) which is indeed a tie. The game's appearance (format) should be similar to the Wikipedia figure above.
- the virtual values of board situations immediately under the original board situation which is initially empty (the root).

You can 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. Needless to say, this is also useful for debugging your program during the development stage.

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 12:01 midnight.)