Lab 2: Minimax

Published Edit ⋮

Note: You may, if you wish, submit your lab early to have it graded before the midterm. If you elect for this, please submit by June 18, then send me an e-mail letting me know. I will grade and return your lab by June 19.

This assignment requires coding in Javascript. Use the template files provided in Lab2_template.zip (https://osu.instructure.com/courses/102347/files/32373889/download?download_frd=1) to get started.

You will be creating code to play tic-tac-toe:

---

1) Implement terminal check and utility functions for tic-tac-toe. See the `is_terminal` and `utility` functions in `tictactoe.js` for more detailed instructions. (2 pts)

---

2) Create a function to play tic-tac-toe, using the Minimax algorithm. Modify the template function `tictactoe_minimax`, provided in `tictactoe.js`, to accomplish this. (Note: You have already been provided a working recursive depth-first search implementation; you need only add Minimax to it.) (3 pts)

---

3) Use the provided `tictactoe.htm` file to play a few games against your code from (2). Does it play the way you expect? Can you beat it?

Also, notice that the evaluated/expanded state counts are reported while you are playing a game. Record these values as you play. How do they change over the course of the game? Is this what you expected? Discuss. (1 pt)

---

4) Modify your algorithm to use alpha-beta pruning. (Copy your code from (2) into the provided `tictactoe_minimax_alphabeta` function. Modify this.) (3 pts)

---

5) Repeat (3) using your new function. (1pt)

---

6) Configure the game so the human player goes second (and make sure the initial board is empty). Set for standard MiniMax and start the game. Report how many nodes are expanded to calculate the

first move. Repeat for MiniMax w/ Alpha-Beta Pruning.

How do the two algorithms compare? Are the results what you expected? Discuss.

Now, look again in `tictactoe.js` and notice the variable `move_expand_order`, which controls the order in which different possible moves are examined. An alternate ordering has been provided but commented out. Uncomment this and repeat the above experiment. How do the results change? Try to explain why you get this result. (1pt)

Extra Credit) Try improving on the ordering from (6). Can you explain why your result works better?

Scoring:

- 0.5pts: Any improvement over the results from (6).
- 1pt: Expands fewer than 50,000 nodes and can explain why the changes work.
- 2pts: Expands fewer nodes than any other student submission (and meets the requirements for getting 1pt)
- 2pts: Alternately, find the worst possible ordering (must expand more than 115,000 nodes).

(You may try for both 2pt options, but 2pts total is the max extra credit you may receive. Also, note that the node counts above assume Alpha > Beta.)

Include your extra credit submission in your report. Also include the number of nodes expanded by your submission in your report. And, of course, the version of `move_expand_order` for you submission in your code.

Create a report including answers to the asked questions (please use doc or pdf format, alternately txt format if you absolutely cannot do the other two).

Then, create a ZIP archive of your code files and report and submit it to the Lab 2 dropbox on Carmen.

(Please include the entire template in the zip, our grader should be able to download and unzip your submission and immediately run it, without copying anything in.)

Tips:

Hints and further directions have been provided as comments in the code, please read them carefully.

If you need to print out debug statements, you may use the `console.log()` function to print out to the browser’s debug console. To access this log, use Ctrl-Shift-J in Chrome or Shift-F5 (Console tab)

in Firefox.
Safari Option-Cmd-C

Alternately, you can use the `helper_log_write()` function (from `tictactoe_helper.js`) to output to the log region on the web page.

A debug function (and a button to run it) has been provided in `tictactoe.js` to help you if you need to run any code for testing purposes.