

The last homework assignment will be to create an AI agent that uses MCTS learning.

The goal of this assignment is to beat your best agent—either the heuristic or MC agent. The key is the storage / exploration. Storage can be done offline or online. Offline obviously doesn't impact the speed of online performance but doesn't necessary help. What would you do??? We'll discuss!

There are three ways to test your agent:

1. Play against a randomly playing agent—that uses **MC learning**
2. Play against your previous agent
3. Play against the random agent (not really relevant anymore)

As was the case in the previous homework assignments, you will be grading based on the logic, clarity and documentation of your solution. Some issues to consider:

- 1) Do you use the same tree for representing when the agent plays both sides – player one or two?
- 2) How did you set c in the MCTS—see: https://en.wikipedia.org/wiki/Monte_Carlo_tree_search

$$\frac{w_i}{n_i} + c \sqrt{\frac{\ln N_i}{n_i}}$$

In this formula:

- w_i stands for the number of wins for the node considered after the i -th move
- n_i stands for the number of simulations for the node considered after the i -th move
- N_i stands for the total number of simulations after the i -th move run by the parent node of the one considered
- c is the exploration parameter—theoretically equal to $\sqrt{2}$; in practice usually chosen empirically

- 3) How did you store the tree? (Dictionary - <https://gist.github.com/qpwo/c538c6f73727e254fdc7fab81024f6e1> or pickle - https://github.com/giladariel/Connect4/blob/master/agent_tree.py)

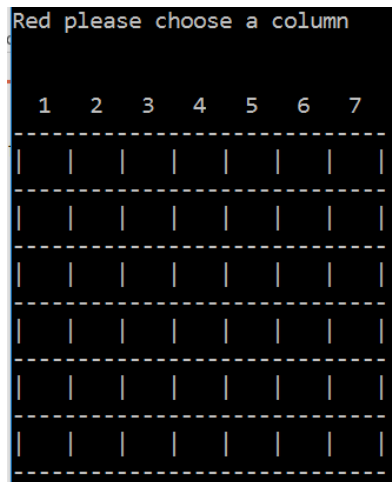
The grade breakdown will be:

60% -- creating a solution that beats the better one of your two previous agents. Contact me if you tried but failed to beat your best agent because you think it played optimally or near optimally.

20% -- documentation about the logic of your solution

20% -- the creativity of your solution – the amount you beat the previous agent by

The interface need not be anything fancy. This is what I had imagined and previously implemented myself:



The human player (black) can then chose a column 1-7.

To get the full 20% (20 points) for documentation, there should be both comments in your Python solution and in a separate document (up to one page) of general explanation of your code.

To get the last 20% (20 points) for creativity you will need to always wins when you play against yourself with a perfect game.

Submission will be via Github as is normally done. **Please send me a writeup of the results, not the full tree!!!! However, please **do** send me the Git link for the code that was used.**