CS 461
Spring 2019
Program 2
Monte Carlo Tree Search

Read and ask questions about the handout on Monte Carlo Tree Search before beginning this project.

The game of Connect Four is played between two players (Red and Yellow) using a vertical board with 7 columns, with 6 slots in each column. Red always plays first, with players alternating turns. Each drops a marker with their color into any column with an empty slot. Gravity then causes that token to drop to the lowest unoccupied slot. Once a piece is played, it cannot be moved or altered. Each player tries to get 4 markers of their color in a straight line, either horizontally, vertically, or diagonally. Play continues until one player wins or there are no more legal moves (which is a draw, scored as a half-win for both players). It is not possible to 'pass' or skip a turn; a player must make a play if there is any legal play to make.

Write a program that plays Connect Four using a Monte Carlo tree search. For each turn, print the move selected, estimated wins, estimated probability this is the best move, and print the board position. To keep search times manageable, keep the number of trials per position fairly low—maybe 50 or so during development. Use a lightweight playout. To show the advantages of further search, give one player a 50% bonus in the number of searches when it's their turn (e.g. 75 for one player, 50 for the other). You should find that the player with more searches can win fairly consistently, and that both players will improve their play if the total number of searches increases.

Write up a short document describing your program, its algorithm, and relevant data structures. No, "the source code is self-documenting" isn't adequate.

Extra credit! For up to 10% (1 letter grade) extra credit, give your program the ability to learn from its play by storing some or all of the positions that come up during search and their win/loss information. In future games, if the position arises during a search, its information can be put to use so the results of previous searches need not be lost. This allows the program to search deeper, and more effectively. (Indeed, this was the primary learning method for Alpha Zero.)