**CS470/570
Artificial Intelligence
Spring 2018
Project #2
Due: Friday, March 9th**

Connect Four is played on a vertical board with 7 columns each 6 positions high . Players alternate dropping different colored pieces into one of the 7 columns. Once a column is filled (i.e. six pieces have been dropped into it) that column is no longer a legal move. The goal is to get four pieces in a row: vertically, horizontally, or diagonally. It is possible for the game to be a tie, if the board is filled without anyone connecting four pieces. This is a relatively easy game for computers because of the low branching factor. There are lots of on-line versions (e.g. <https://www.mathsisfun.com/games/connect4.html>) if you want to play.

**Project:** Write a program to play Connect Four against a human opponent.

**Requirements:** The program must use a minmax search algorithm (it can be coded as negamax). The user must be able to determine which side goes first. The program must display the board after each move. (This can be a simple text based display.) The program must not take more than 10 seconds to make a move.

**Algorithms:** Your program must use, at least, minmax search. You will need to write an evaluation heuristic because searching the entire game tree is not feasible. Your heuristic should weigh wins and losses so that the program always blocks a potential win by the opponent (if there are three opponent pieces in a row the computer places a piece to block the win) and always makes a winning move if one is available. Note: if there are two possible paths to a sure victory the program may take the longer one.

**Scoring:** The project will be scored out of 100. Just a minmax search algorithm is worth up to 80 points. Alpha-beta pruning is worth 15 points. Additional features (selective evaluation, selective ordering of the moves, shortest path to win or longest path to lose) is worth 5 points.

**Hand-In:**

You need to hand in a typed write-up containing the following:

1. An **abstract** (1 paragraph) summarizing what you did and what the results were.
2. An **algorithm section** (no more than 3 pages with screenshots), including the following,
	1. A brief description of your minmax algorithm and how well it worked
	2. The depth of your search
	3. A list of extra features, if any, that you added, alpha-beta pruning, selective evaluation, move ordering, etc.
	4. A description of the evaluation function.
	5. Whether the program always takes a win when possible and always blocks a loss when possible.
	6. A few screen shots showing game play
3. A **conclusions section** (1-2 paragraphs), including,
	1. A discussion of the strengths and weaknesses of the program (if any).
	2. A brief description of how well you felt the program played.
	3. Improvements you would like to make.
4. Your code.