OPTIONAL PROJECT: "4-IN-A-ROW"

- The game is played on an 8x8 board - one player plays the "X" pieces and the other plays the "O" pieces.
- Players alternate moves, placing one piece on any open square on each turn.
- The first player to get four pieces in a row, side-by-side or vertical, is the winner. (diagonals don't count, only horizontal or vertical)
- If the board becomes full, and nobody has 4 in a row, then the game is a draw.
- The following are examples of winning positions for "X":

```
--------       -------
--------       -------
-X--------   --------
-0--------   --------
OX-0------   -00-00--
-X-------   ---XXX--
-O-------   --------
--------       -------
```

- Work individually, or on a team.
- Your program is one player, and will attempt to defeat the human operator.
- It first asks the human whether he/she wants to move first or second.
- The current position is displayed on the screen before each move, with the axes labeled as follows:

```
  1 2 3 4 5 6 7 8
A - - - - - - -
B - - - - - - -
C - X X - - - -
D - - O - - - -
E - - - - - - -
F - - - - - - -
G - - - - - - -
H - - - - - - -
```

- The human enters moves using the notation above, for example by typing "D3".
- The program needs to detect when the game has ended, when a player has won/lost, or if the game is a draw.
- The program must never play an illegal move. It must only play on open square within the 8x8 boundary.
- The program must move within 5 seconds or less.
- It should be sufficiently bug-free to play through an entire game without ever crashing. Memory-leaks can cause these kinds of programs to fail late in the game, so test it on some long games.
- You MUST use minimax search, at least 3 plies deep. Alpha-beta pruning and other techniques will help.
- You can use any programming language you want. Compiled languages such as C, C++ are good choices.
- Give your program a “name”. The instructor’s program is named “Dark Cloud”.

*We will play the programs against each other for fun – the best one wins the “Great Tiger”!*