Artificial Intelligence
Machine Problem 2 – Alpha/Beta Search for Generalized Tic-Tac-Toe

Introduction
For this assignment, you will implement the minimax algorithm with alpha-beta pruning in order to find the optimal move for a game of generalized tic-tac-toe. In this version of the game, the players can choose different board sizes, for example 4x4 or 5x5, instead of the normal 3x3. The game proceeds with the usual rules for tic-tac-toe (see https://en.wikipedia.org/wiki/Tic-tac-toe).

Requirements
You are to modify the mp2basecode program to implement the alpha-beta search for making the computer's move. This will require implementing additional methods for testing for terminal states and finding the utility of states, among others. You can follow the textbook's pseudocode for the algorithm.

Additional Requirements
1. The name of your source code file should be `mp2.py`. All your code should be within a single file.
2. You can only import numpy, random, and math packages.
3. Your code should follow good coding practices, including good use of whitespace and use of both inline and block comments.
4. You need to use meaningful identifier names that conform to standard naming conventions.
5. At the top of each file, you need to put in a block comment with the following information: your name, date, course name, semester, and assignment name.

What to Turn In
You will turn in the single mp2.py file using BlackBoard.

HINTS
- It's easiest to use the backtracking method. That is, instead of generating hypothetical states, just apply the moves to the game board, compute the utility, and then backtrack the move. This requires that you save the current board configuration before trying each move, so you can backtrack it later

Grading Rubric

Category	Unsatisfactory (0-1 points)	Satisfactory (2-3 point)	Distinguished (4-5 points)
Program Correctness	• Program does not execute due to errors • Incorrect results for most or all input	• Program works and completes most tasks appropriately • Program fails to work for special cases	• Program runs and completes all required tasks • Handles any required special cases • Executes without errors
Programming Style	• No name, date, or assignment title included • Poor use of white space • Disorganized and messy • No or few comments in the source code • Poor use of variables (improper scope/visibility, ambiguous naming).	• Includes name, date, and assignment title. • White space makes program fairly easy to read. • Well organized code. • Some comments missing in the source code or too many comments • Good use of variables (few issues with scope/visibility or unambiguous naming).	• Includes name, date, and assignment title. • Excellent use of white space. • Perfectly organized code. • Source code is commented throughout when needed • Excellent use of variables (no issues with scope/visibility or unambiguous naming).
Following Specifications	• Incorrect filenames • Incorrect specified identifier names • Source code organization different from requirements • Additional requirements not satisfied	• Correct filenames and class names • Few issues with other specified identifier names • Source code organization close to requirements • Some additional requirements not satisfied	• Correct filenames and specified identifier names • Source code organization satisfies all requirements • All additional requirements satisfied

Sample Program Output

```
CLASS: Artificial Intelligence, Lewis University
NAME: [put your name here]

Please enter the size of the board n (e.g. n=3,4,5,...): 3
  1 2 3
 -------
1| | | |
 -------
2| | | |
 -------
3| | | |
 -------
Player's Move

Choose your move (row, column): 2,2
  1 2 3
 -------
1| | | |
 -------
2| |X| |
 -------
3| | | |
 -------
  1 2 3
 -------
1|O| | |
 -------
2| |X| |
 -------
3| | | |
 -------
Player's Move

Choose your move (row, column): 1,3
  1 2 3
 -------
1|O| |X|
 -------
2| |X| |
 -------
3| | | |
 -------
  1 2 3
 -------
1|O| |X|
 -------
2| |X| |
 -------
3|O| | |
 -------
Player's Move

Choose your move (row, column): 2,3
  1 2 3
 -------
```

```
1|O| |X|
 -------
2| |X|X|
 -------
3|O| | |
 -------
  1 2 3
 -------
1|O| |X|
 -------
2|O|X|X|
 -------
3|O| | |
 -------
GAME OVER
You Lost!
```