CSC 215-01 Artificial Intelligence (Spring 2023)

Project 4: Solving (Wild) Tic-Tac-Toe using Minimax Search

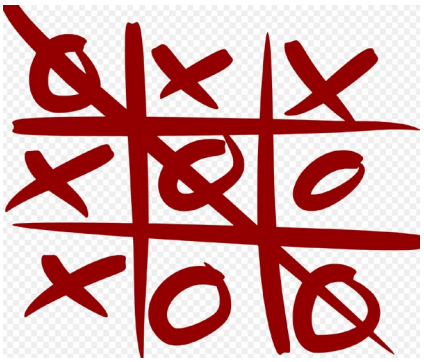
Due at 3:00 pm, Monday, April 24, 2023

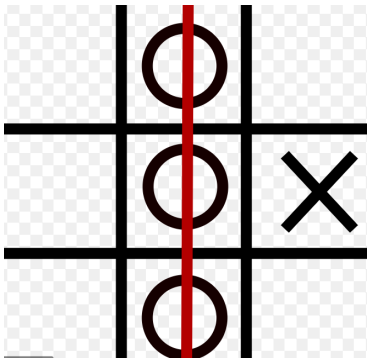
Demo: class time, Monday, April 24, 2023

1. Problem Formulation

Download the two notebooks on Canvas. This project is twofold:

1. Complete the minimax function for the Tic-Tac-Toe and Wild Tic-Tac-Toe.
2. Verify the following characteristics of each game by simulations. Run your code multiple times (e.g., 10 or 20 times) under each setting (i.e., optimal vs optimal; optimal vs random), observe the results (i.e., how many times each player wins or loses out of all the runs), and include your findings in your report.





1.1 Tic-Tac-Toe

Rules:

- The first player plays x and the second player plays o. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game. If all cells are used without someone getting a three in a row, the game is a draw.

Characteristics to verify by simulations:

1. If both players play optimally, the game will always be a draw.
2. If one player is optimal and the other is random, the optimal player will win most of the games and the random player will never win.

1.2 Wild Tic Tac Toe

Rules:

- On each turn a player can play either x or o. The player who gets the first three (x's or o's) in a row wins the game. If all cells are used without someone getting a three in a row, the game is a draw.

Characteristics to verify by simulations:

1. If both players play optimally, the first player can always win if the middle cell is chosen as his/her first move.
2. If both players play optimally, the game will be always a draw if the first player does NOT choose the middle cell as his/her first move.
3. If one player is optimal and the other is random, the optimal player will win all the games if she moves first and the optimal player will win most of the games if she moves second.

For each game, I wrote three verification functions for your simulations. They are in the provided notebooks on Canvas:

- optimal_vs_optimal: both players are optimal. Note that it is pure optimal, meaning that even the first move is optimal. Our minimax function always returns the first best score and plays that move.
- random_vs_optimal: the first player plays randomly, and the second player plays optimally.
- you_vs_optimal: user gets to play as the first player against an optimal player.

2. Output examples

```
In [10]: optimal_vs_optimal()
```

```
Player x and Player o Both play optimally.

Board :

  0  |  1  |  2  |
-----------------
  3  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------

x :

  x  |  1  |  2  |
-----------------
  3  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------

o :

  x  |  1  |  2  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------

x :

  x  |  x  |  2  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------

o :

  x  |  x  |  o  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------

x :

  x  |  x  |  o  |
-----------------
  3  |  o  |  5  |
-----------------
  x  |  7  |  8  |
-----------------

o :

  x  |  x  |  o  |
-----------------
  o  |  o  |  5  |
-----------------
  x  |  7  |  8  |
-----------------
```

```
x :

  x  |  x  |  o  |
-----------------
  o  |  o  |  x  |
-----------------
  x  |  7  |  8  |
-----------------

o :

  x  |  x  |  o  |
-----------------
  o  |  o  |  x  |
-----------------
  x  |  o  |  8  |
-----------------

x :

  x  |  x  |  o  |
-----------------
  o  |  o  |  x  |
-----------------
  x  |  o  |  x  |
-----------------

Draw!
Out[10]: 'Draw'
```

3. Grading Breakdown

You may feel this project is described with some certain degree of vagueness, which is left on purpose. In other words, **creativity is strongly encouraged**. Your grade for this project will be based on the soundness of your design, the novelty of your work, and the effort you put into the project.

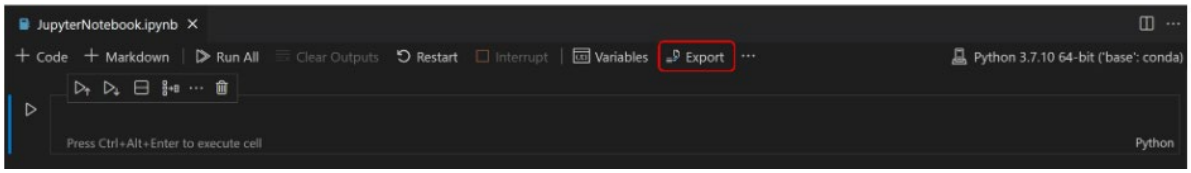
Use the evaluation form on Canvas as a checklist to make sure your work meet all the requirements.

4. Teaming

Students must work in teams of 2 people. Think clearly about who will do what on the project. Normally people in the same group will receive the same grade. However, the instructor reserve the right to assign different grades to team members depending on their contributions. So you should choose partner carefully!

5. Deliverables

(1) The HTML version of your notebook that includes all your source code. In VS Code, you can export a Jupyter Notebook as an HTML file. To export, select the Export action on the main toolbar. You'll then be presented with a dropdown of file format options.



(2) Your report in PDF format, with your name, your id, course title, assignment id, and due date on the first page. As for length, I would expect a report with more than one page. Your report should include the following sections (but not limited to):

- Problem Statement
- Methodology
- Experimental Results and Analysis
- Task Division and Project Reflection
- Additional Features

In the section "Task Division and Project Reflection", describe the following:

- who is responsible for which part,
- challenges your group encountered and how you solved them
- and what you have learned from the project as a team.

In the section "Additional Features", you describe and claim credit for additional features.

To submit your notebook and report, go to Canvas "Assignments" and use "Project 4".

All the deliverables must be submitted **by team leader** on Canvas before

3:00 pm, Monday, April 24, 2023

NO late submissions will be accepted.

6. Hint

My partial code will be provided on Canvas.

7. Code Efficiency

Your grade for this project will be only based on code correctness, not on code efficiency. In other words, although you are more than welcome to optimize your code to make it run faster (e.g., by breaking the loop earlier without exploring unnecessary tree nodes/states), you are not required to do that. The goal of the project is to verify the characteristics specified about the two games, irrespective of how long each simulation would take.

Btw, my wild Tic-tac-toe optimal vs optimal code took about 30 seconds to run. Try if you can beat me! Again, code optimization is not required.

8. **Extra features (each feature 10 pts)**

- In this project, each move in the two games is ***optimal*** (notice that here we search the entire game tree to get minimax values since Tic-Tac-Toe has a relatively small game tree). A more efficient AI can be implemented by using either (or both) of the following:
  - alpha-beta pruning.
  - limiting the search depth and using some heuristic function or machine learning model to evaluate the goodness/reward of the non-terminal states at the cutoff tier.

Can you improve your solution using any strategy above?

9. In-class Presentation.

On the due day, each team has 5 minutes to present your work in the class. Explain your solutions by referring to your notebook. You do not have to prepare the PowerPoint slides for your presentation.