WRITING ASSIGNMENT 4

Yahya Alhinai
alhin010@umn.edu
CSCI4511W

November 19, 2019

Team members

Yahya Alhinai

A Brief Description of the Problem

The purpose of this project is to build up a rational agent for a Tic-Tac-Toe game playing against the user. This project will account for the approaches/algorithm the agent uses and accounting for how fast it reaches a solution and how optimal the reached solution as well as accounting for memory usage.

Tic-Tac-Toe is a game that is played by two players each of them take turn to mark on an empty spot of a 3x3 grid. Each player has their own mark and the first player who get to complete 3 adjacent spot (vertically, horizontally, or diagonally) the game stops and the player wins the game. Otherwise, if non of the players could complete this requirement and the game ran out of empty spot then it is a tie.

I'm intending to me make the agent rational meaning the agent takes the best option possible every single time.

The approach you are going to use to solve it

An ideal plan would be searching all the algorithm that can solve a Tic-Tac-Toe game and not only the generic algorithms, such as MCTS, A*, MiniMax, and Hill-Climbing, but also

newly published algorithm. After identifying the algorithms that work, test each one of them to see with one works the best in terms of speed, optimal solution, and memory consumption.

After identifying the right algorithms, implementing it in Tic-Tac-Toe game would be the next step. This is will lead to filter out further until we end up with the best algorithm. Server test will be conduced for the game working at this algorithm to tune the performance and enhance the process of state elimination.

The last to would be to make the game aesthetically appealing by including a graphics representation of the game and interpolating movement between states.

The Software Needed

- **Python 3.6:** To run the code.
- **Processing:** For graphics making.
- **Generic Intended Algorithms:** MCTS, A*, MiniMax, Hill-Climbing

How to Run and Analyze Results

Conducting tests will be done by running each algorithm alone and test the possibility of combining two algorithms to see if they work better than running then individually. Identify the best algorithm or the combination of algorithms to be used to solve the Tic-Tac-Toe game.

After this, run several tests with random inputs for this the algorithm chosen and then compute the running time for ever single test and the winning rate for all tests. reporting the results of speed, accuracy, and memory usage for each test.

An Estimated Timeline

1. **November 20:** Finding all proper algorithm that can solve Tic-Tac-Toe game, analyzed each and every one of them, and choose the best one or the best two algorithm (in terms of speed, accuracy, and memory usage).
2. **November 27:** Ruining a simple Tic-Tac-Toe game (might be run graphically) in python and start implementing the chosen algorithms

3. **December 4:** Make sure the code is all working. Report and comapre results. Start the project report.

4. **December 11:** Test for improvements or last minute adjustment. Analyzing and interpreting The obtained results. Finishing up the project report.

5. **December 18:** Submitting the project.

References

[1] A. Bhatt, P. Varshney, and K. Deb. In search of no-loss strategies for the game of tic-tac-toe using a customized genetic algorithm. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 889–896. ACM, 2008.

[2] R. Bjarnason, A. Fern, and P. Tadepalli. Lower bounding klondike solitaire with monte-carlo planning. In *Nineteenth International Conference on Automated Planning and Scheduling*, 2009.

[3] J. B. Odili and M. N. Mohmad Kahar. Solving the traveling salesman's problem using the african buffalo optimization. *Computational intelligence and neuroscience*, 2016:3, 2016.

[]