

Final Project Milestone 1: Proposal

Sawyer Clemmons, Bryant Kelley, Allen Kinzalow, and Luke Welton

Problem Formulation and Task Definition

The task we are looking at is comparing the performance of algorithms and advanced techniques on the popular video game Threes. Specifically, we are looking at the performance of A* versus advanced techniques such as Monte Carlo Tree Search and Expectimax.

An absolute baseline would be to choose a random move every turn while the more meaningful baseline is A* for attempting the game. After observing the performance using A*, we will observe the performance of Monte Carlo Tree Search and Expectimax.

Data Source

General Data

The data source will be generated as the systems play through the game. The information returned from the game engine are the values of tiles on the board and the overall score. The implementation of Threes itself will be based on the work of Angela Li on ThreesJS (which is in turn based on the work of Asher Vollmer), but we will be using Python instead of JavaScript. We will also be improving our version to be more similar to the original in its generation of the next tile.

Input and Output

Input into our functions will be the values and positions of tiles on the board and the next tile look-ahead. The output will be the score which will be used for evaluating the functionality of the systems.

Literature Review

There are a number of attempts to solve the game 2048 using various algorithms. “it implements a highly-optimized brute-force search over the game tree...”⁴ However, there have not been any attempts to use reinforcement learning to teach an AI to play and beat Threes such as an AI that learned to beat international master chess players in less than 72 hours.

We have chosen to utilize Monte Carlo Tree Search and Expectimax based on the report by Philip Rodgers and John Levine, who implemented the former and approximated the latter for a comparison of AI methods to solve 2048, a similar game to Threes.

References

1. <http://asherv.com/threes/>
2. <https://www.technologyreview.com/s/541276/deep-learning-machine-teaches-itself-chess-in-72-hours-plays-at-international-master/>
3. <https://github.com/angelali/threesjs>
4. <https://ieeexplore.ieee.org/abstract/document/6932920>