2048 Agent

Seth Carroll and Jenny LaPierre

Problem description: What problem are you solving? Describe the problem from a computational perspective. What are the inputs and outputs (exactly)? Why is it interesting?

For our project, we would like to write an agent to solve the game 2048 (https://gabrielecirulli.github.io/2048/) using an expectimax adversarial search. The game state is represented by the numbers in the grid, which we will parse from the HTML. Actions for the agent in the game are arrow key presses, which can be simulated using JavaScript. For each turn, the action of the "opponent" (the game itself) is a tile added to the game, which has a value of either 2 or 4 and can have up to four possible locations. We plan to implement a JavaScript front end (perhaps as a browser extension) that will interface with the game in-browser, as well as a Python backend that will handle all the computations. We are interested in this problem because we want to investigate how effective an expectimax agent will be in solving the game that uses a state-based, ma.

Algorithms: What algorithms do you use? Why are these algorithms appropriate? How are these algorithms typically used, and how are you using them?

We will be applying an Expectimax algorithm to solve the game and achieve a win state. In each state of Expectimax, a player can make one of four moves (Up, Down, Left, Right). Using Expectimax we can have the agent maximize over all possible moves, max nodes being these moves. Then, the chance nodes represent where new tiles will generate and which tiles will spawn on the board after each player move. These algorithms are typically used to maximize outcomes in the face of uncertain decision (from a limited set of known options) on the part of another agent in a search problem. This is typically used in circumstances where we assume a player wishes to play optimally but we do not know that of the other player (Russell and Norvig 2010). In this case, we will make our player moves aiming to maximize scores, quite literally in the numerical values present in 2048. We will use the random placement of new tiles as uncertain chance steps.

Results: What results do you expect to show?

We expect to show that our agent can win the game more than 50% of the time (hopefully closer to 100%). For the purposes of this project, we will consider a win to be a value of 2048 for any tile on the board, even though the game offers players an option to go beyond that.

Division of Work: Who is responsible for each part of the project?

For the report, Seth will write the Introduction, Background, and Project Description sections, while Jenny will write the Abstract, Related Work, Experiments, and Conclusion sections. For the code, we anticipate pair programming on as much of the project as possible. Collaborative work will enable a better understanding of the interface we're working with and provide maximum opportunities for learning on both our parts.

References

Russell, S. and Norvig, P. 2010. Artificial Intelligence A Modern Approach. ed. Michael Hirsch, 178-179. New Jersey: Pearson Educations, Inc.