CS 461 – ARTIFICIAL INTELLIGENCE
Term Project Proposal

Group Members
Berk Çakar - 22003021
Ceren Akyar - 22003158
Deniz Mert Dilaverler - 22003530
Elifsena Öz - 22002245
İpek Öztaş - 22003250

Objective

This project aims to implement and test three algorithms for a tile-matching game called 2048: Min-Max Tree Search, Double Deep Q-Network (DDQN), and Monte Carlo Tree Search [1]. Our algorithm evaluation will consider achieving goals (reaching number levels 512, 1024, and 2048) and the final game score on 100 tries. The project deliverables will include the environment code, Monte Carlo Tree Search, Double Deep Q-Network, and the Min-Max Tree Search with our trial results and our analysis of the algorithm results.

Related Work

Several studies in the literature use AI approaches to play 2048. The most prominent one comes from a doctoral thesis [2]. There, the author tries techniques such as Monte Carlo Tree Search, Min-Max Tree Search Temporal Difference Learning, Expectimax Search, and Deep Reinforcement Learning. However, a clear comparison between these techniques is not provided. Similarly, the replication package of this study is not available. Another work related to our project is a paper that utilizes variants of Deep Q-Learning to play 2048, as well as Minesweeper and Sudoku. Similar to the previous one, no replication package is available, and the emphasis on 2048 is less than the other two games [3]. Another study also used Deep Reinforcement Learning to play 2048, but they only achieved the 2048 tile 7% of the time [4]. Hence, our project's significance will serve as an extension to the literature by providing a clear comparison between different AI techniques and a replication package that other researchers can use. Likewise, our initial intention is to improve the 7% success rate.

Technical Outline

Our principal goal is to analyze different approaches for solving the game of 2048. The Monte-Carlo Tree Search, Double Deep Q-Network, and Min-Max Tree Search algorithms will be evaluated and compared according to their max and average final scores and their percentage to reach 512, 1024, and 2048 after 100 tries. We will use an extension of the Gym library for the 2048 environment. PyTorch will be used for implementing the DDQN algorithm.

Timeline

We plan on implementing Min-Max Tree Search, Monte Carlo Tree Search and making significant progress on Double Deep Q-Network (DDQN) before the progress update report (i.e., before 26 November 2023). After the progress update, we aim to compare the algorithms and enhance the 7% success rate while making final touches until the final presentation, which is in the second week of December 2023.

References

[1] L. H. Chan, “Playing 2048 with Deep Q-Learning (with pytorch implementation),” Medium, Oct. 20, 2023. [Online]. Available: https://medium.com/@qwert12500/playing-2048-with-deep-q-learning-with-pytorch-implementation-4313291efe61. [Accessed: Oct. 20, 2023].

[2] H. Guei, *On Reinforcement Learning for the Game of 2048*, Jan. 2023. doi: https://doi.org/10.48550/arXiv.2212.11087

[3] A. Mehta, “Reinforcement Learning For Constraint Satisfaction Game Agents (15-Puzzle, Minesweeper, 2048, and Sudoku).” arXiv, Feb. 09, 2021. Accessed: Oct. 20, 2023. [Online]. Available: http://arxiv.org/abs/2102.06019

[4] A. Goga, *Reinforcement learning in 2048 game*, Oct. 2017. Accessed: Oct. 20, 2023. [Online]. Available: http://cogsci.fmph.uniba.sk/~farkas/theses/adrian.goga.bak18.pdf