Lucas Summers (lsumme01@calpoly.edu)
Braeden Alonge (balonge@calpoly.edu)
Nathan Lim (nlim10@calpoly.edu)
Megan Fung (mfung06@calpoly.edu)

CSC 480 Team Project Proposal

Project Idea (Problem Statement/ Question)

- What is the best design of an intelligent 2048 agent? Specifically, which AI methods amongst search-based, heuristic, Monte-Carlo, and reinforcement-learning techniques tend to produce the highest average scores in ?

Abstract (Summary of Project Purpose + Approach)

- This project aims to develop an AI agent capable of achieving the highest possible score in the single player puzzle game 2048. We will experiment with a variety of AI techniques, including Expectimax search, heuristic evaluation functions, and reinforcement learning methods. Our bot will dynamically decide optimal moves at each game state by predicting future outcomes and maximizing reward. We aim to develop a model that is able to achieve an average game score above the 2048 tile in a 4x4 board and achieve a best case high score as close to tile 131,072 as possible—the theoretical limit of a 4x4 board. Through this project, we hope to explore how AI planning and decision-making algorithms perform under uncertain, partially observable environments.

Motivation (Why)

- Games like 2048 offer an ideal environment for experimenting with AI techniques, balancing strategy, planning, and randomness. Building an AI that can consistently achieve high scores in 2048 is a challenging way to apply search algorithms, heuristic designs, and reinforcement learning, all while revealing how each different AI algorithm compares with one another. Success in this project would demonstrate practical mastery of game playing AI techniques, and deepen our understanding of how artificial agents can make decisions with incomplete information.

Background (Related Works + Existing Solutions)

- Previous attempts to build 2048 AI bots have primarily used two approaches:
    - Reinforcement Learning: some models apply deep reinforcement learning, allowing agents to learn policies over millions of games.
        - OpenAI: uses reinforcement learning agents in simple games to learn complex strategies autonomously

    - Expectimax Search: unlike minimax, Expectimax accounts for probabilistic outcomes (random tile spawns in 2048).
        - Matt Overlan's AI for 2048: shows that using Expectimax with a strong evaluation function can consistently reach 4096+ tiles.
- Despite these successes, challenges remain: computational cost of deep searches, crafting an effective evaluation function, and balancing exploration vs exploitation in learned policies.

Relevance (Related Class Topics)

- This project directly ties into several AI topics we've covered in class:
    - Search algorithms– expectimax, heuristic based searches
    - Adversarial and Stochastic Planning– handling randomness in environments (non-determinism)
    - Reinforcement Learning– agent training through rewards over time

---

Bibliography

1. **Matt Overlan, "Building a 2048 AI with Expectimax,"** GitHub, 2014.
   https://github.com/nneonneo/2048-ai
   – Discusses applying Expectimax search to 2048 with heuristic evaluation functions.

2. **Guei, Hung. On Reinforcement Learning for the Game of 2048. Diss. National Yang Ming Chiao Tung University, 2023.**
   https://arxiv.org/abs/2212.11087
   – Explores the use of Reinforcement Learning as an approach to creating 2048 solving agents.

3. **P. Rodgers and J. Levine, "An investigation into 2048 AI strategies," 2014 IEEE Conference on Computational Intelligence and Games, Dortmund, Germany, 2014, pp. 1-2, doi: 10.1109/CIG.2014.6932920.**
   https://ieeexplore.ieee.org/document/6932920
   – Goes over various approaches to 2048 solving agents, including MCTS and ADLS (Averaged Depth Limited Search).

4. **Watanabe, Shota and Kiminori Matsuzaki. "Enhancement of CNN-based 2048 Player with Monte-Carlo Tree Search." 2022 International Conference on Technologies and Applications of Artificial Intelligence (TAAI) (2022): 48-53.**
   https://www.semanticscholar.org/paper/Enhancement-of-CNN-based-2048-Player-with-Tree-Watanabe-Matsuzaki/7ea90a71c41240386da55bed2a1530e13b330de4
   – Use of a deep learning CNN-based algorithm to solve 2048 while combining MCTS.

5. **Zhou, Yulin. "From AlphaGo Zero to 2048." (2019).**
   https://www.semanticscholar.org/paper/From-AlphaGo-Zero-to-2048-Zhou/b9cc9a861

531ede494a365e42b7ba788d638eaec
– Exploration of how the algorithm used in the successful AlphaGo Zero can be applied to solve 2048.

Timeline

Week 1:

- Set up Git repository
- Define a tech stack (likely Python, NumPy, PyTorch) and workspace to begin the coding portion of the project for all members
- Begin development of a custom 2048 grid to develop on
- Decide on size of grid (ie. 4x4, 5x5)

Week 2:

- Begin/continue research on how to implement Expectimax, and/or examine other possible algorithms to implement
- Finish development of custom 2048 grid
- Start final paper; begin with outline and move onto drafting afterwards
- Start code of the AI model using the finalized custom 2048 grid
- Revise proposal

Week 3:

- Continue development of code, experimenting with different algorithms and coming to conclusions on what the final algorithm used should be
- Continue adding to the final paper, completing at least half of it by the end of the week
- Attain a final paper rough draft

Week 4:

- Finish final touch ups of the "best" agent and run large-scale evaluation using at least 1000 games to get final stats of model
- Finish final paper
- Create final presentation slides and prepare to present

- Make collaboration documents