Semester Project Proposal

Project Title: AI Solver for 2048

1. Introduction

The objective of this project is to develop an AI solver for the game 2048, employing heuristic algorithms to enhance decision-making capabilities. The game 2048 is a single-player sliding block puzzle played on a 4x4 grid, where the goal is to combine tiles with the same number to create a tile with the number 2048. Players can move all tiles on the board in one of four directions—up, down, left, or right. After each move, a new tile (with a value of 2 or 4) randomly appears on an empty spot on the board. The game ends when no valid moves are possible.

Heuristic and Rules:

- **Heuristic Function:** The AI will utilize a combination of heuristic strategies, including:
  - **Snake Pattern Scoring:** Prioritizing configurations where higher-value tiles are arranged in a snake-like pattern to facilitate merging.
  - **Adjacent Tiles Scoring:** Encouraging moves that position similar tiles adjacent to each other, increasing merging opportunities.
  - **Empty Tiles Count:** Favoring moves that result in a higher number of empty tiles, providing more flexibility for future moves.
- **Rules and Constraints:**
  - The AI must operate within the standard rules of 2048.
  - It should handle the stochastic nature of new tile placements effectively.
  - The AI should aim to achieve the highest possible tile value, ideally reaching or exceeding the 2048 tile.

2. Implementation Strategy (might change as we start coding)

Algorithms:

- **Expectimax Algorithm:** To account for the randomness in new tile placements, the AI will implement the expectimax algorithm, which models both the player's moves and the probabilistic nature of tile generation.
- **Heuristic Evaluation Function:** A composite scoring function will be developed, integrating to evaluate the desirability of game states.

Implementation Steps:

1. **Game Simulation:**
    a. Develop a Python-based simulation of the 2048 game using the numpy library for efficient array manipulations.
    b. Implement a “Grid” class to manage the game state, including tile movements and merging logic.
2. **AI Development:**
    a. Implement the expectimax algorithm to evaluate potential moves and predict outcomes based on the composite heuristic function.
    b. Incorporate depth-limited search to balance computational efficiency with decision quality.
3. **Testing and Optimization:**
    a. Conduct extensive testing against various game scenarios to assess the AI's performance.
    b. Optimize the heuristic weights and search depth based on empirical results to enhance the AI's effectiveness.

3. Deliverables

- **Codebase:** A fully functional Python implementation of the 2048 game and the AI solver.
- **Documentation:** Comprehensive documentation detailing the code structure, algorithmic decisions, and usage instructions.
- **Performance Analysis Report:** An analytical report presenting the AI's performance metrics, including average scores, highest tile achieved, and success rates in reaching the 2048 tile or higher.
- **Presentation:** A demonstration of the AI's capabilities, showcasing its decision-making process and effectiveness in playing 2048.

Team Members:

- **Muhammad Haris (22K-0505)**
- **Huzaifa (22K-0502)**
- **Daniyal (22K-4294)**