Project Title: AI Solver for 2048
Submitted By: Khair Muhammad (22K – 4290)
Course: AI
Instructor: Sir Abdullah Yaqoob
Submission Date: 10 – 3 - 2023

---

1. Project Overview

- **Project Topic:**
  *This project focuses on developing an **AI solver for the game 2048** using heuristic-based search strategies. The goal is to explore different AI techniques and optimize tile movement decisions to achieve the highest possible score in 2048.*

- **Objective:**

  The primary goal of this project is to implement a strategic AI that can efficiently play 2048 using:

  - Heuristic-based evaluations to determine optimal moves.
  - The **Expectimax algorithm**, which considers the probabilistic nature of tile generation.
  - Alternative techniques such as **Minimax with Alpha-Beta Pruning** for deterministic move selection.

2. Game Description

- **Original Game Background:**
  *2048 is a sliding tile puzzle game played on a **4×4 grid**. Players merge numbered tiles by shifting them in one of four directions (**up, down, left, right**) with the goal of reaching the tile **2048** or higher. New tiles (with values 2 or 4) appear randomly after every move, making the game progressively challenging.*
- **Innovations Introduced:**
  - **AI-powered gameplay:** Unlike human players, the AI will use heuristics and search algorithms to maximize the score and avoid losing conditions.
  - **Multiple heuristic strategies:** The AI will analyze tile placement, empty spaces, and tile clustering to make **optimal moves**.
  - **Expectimax Algorithm Implementation:** Instead of purely deterministic decisions, Expectimax will account for **random tile placements**, leading to **more intelligent moves**.
  - **Enhanced tile positioning strategy:** The AI will prioritize forming high-value tiles in specific patterns (**e.g., snake pattern**) to improve game efficiency.

3. AI Approach and Methodology

- **AI Techniques to be Used:**
    - **Expectimax Algorithm:** Used to predict possible future board states, considering both player moves and random tile spawns.
    - **Minimax Algorithm with Alpha-Beta Pruning:** An alternative approach that evaluates moves based on a minimization-maximization strategy, potentially optimizing tile placement decisions.

    - **Heuristic-Based Scoring**: The AI will evaluate the game grid using custom heuristics.

- **Heuristic Design:**

  The AI will use multiple heuristics to score each board state:

  - **Snake Pattern Scoring:** Encourages high-value tiles to be placed in an ordered fashion (e.g., in one corner).
  - **Tile Merging Potential:** Prioritizes moves that lead to larger tile combinations.
  - **Empty Tile Count:** Rewards moves that maintain more available spaces on the board.
  - **Tile Smoothing:** Minimizes large gaps between adjacent tile values to make merging easier.

- **Complexity Analysis:**

  - The **Expectimax algorithm** operates with $O(b^d)$ **complexity**, where b is the branching factor (typically 4 for 2048) and d is the depth of search.
  - Using **heuristic evaluations** will help reduce computational overhead compared to exhaustive search methods.

4. Game Rules and Mechanics

- **Modified Rules:**
    - The AI will play automatically based on its decision-making algorithm, rather than a human controlling the game.
    - **The AI will** predict multiple moves ahead **rather than making a single move at a time.**

- **Winning Conditions:**

    • The AI aims to **achieve the highest score possible** or **reach the 2048 tile** (or beyond).
    The game ends when **no valid moves remain**.

- **Turn Sequence:**
    - AI evaluates the **current grid state**.
    - It **simulates all possible moves** and assigns scores.
    - AI selects the **best move** based on the heuristic evaluations.
    - The move is executed, and a **new tile (2 or 4) appears randomly**.
    - Steps 1-4 repeat until the game ends.

5. Implementation Plan

- **Programming Language:** *Python*
- **Libraries and Tools:**
    - **Pygame** (For visualization, if needed)
    - **NumPy** (Efficient matrix operations for board representation
    - **Random module** (To simulate new tile appearance)
- **Milestones and Timeline:**
    - **Week 1-2:** Research and finalize AI strategy (Expectimax, heuristics).
    - **Week 3-4:** Implement core game logic (tile merging, movement, new tile spawning).
    - **Week 5-6:** Develop and test AI move prediction.
    - **Week 7:** Optimize heuristic functions for better performance.
    - **Week 8:** Final testing, report writing, and project submission.

6. References

Quora Reference

AI - 2048 Solver