**Project Title: AI Solver for 2048**

**Submitted By:**

* **Yousuf Raza (22K5079)**
* **Abdullah Arif (22K5071)**
* **Saad Zafar (22K5039)**

**Course: AI**

**Instructor: Alishba Subhani**

**Submission Date: 3/10/2025**

**1. Project Overview**

**● Project Topic:**

The project focuses on developing an AI solver for the game 2048 using heuristic-based decision-making techniques. The AI will predict optimal moves to achieve higher tile values efficiently.

**● Objective:**

The primary goal is to implement an intelligent AI solver for 2048 by leveraging heuristic scoring methods and AI algorithms. The AI will be capable of evaluating game states and making optimal moves based on well-defined heuristics. Future enhancements may include Expectimax or reinforcement learning for improved decision-making.

**2. Game Description**

**● Original Game Background:**

2048 is a sliding tile puzzle game played on a 4×4 grid, where players merge tiles of the same value by shifting them in four possible directions (up, down, left, right). The game introduces new tiles after each move, and the objective is to combine tiles strategically to form the 2048 tile (or higher). The game ends when no valid moves remain.

**● Innovations Introduced:**

* AI-Driven Gameplay: Instead of a human player, an AI agent will play the game autonomously.
* Heuristic-Based Move Selection: The AI will rank moves based on:
  + Snake Pattern Scoring: Encourages arranging tiles in a structured manner.
  + Adjacent Tile Scoring: Prioritizes moves that merge similar tiles efficiently.
  + Empty Tiles Count: Prefers moves that keep the grid open for better flexibility.
* Expectimax Algorithm (Potential Addition): A more advanced AI approach that incorporates randomness from new tile spawns.
* Custom Grid Class: Implements optimized game state handling for efficient computation.

These innovations improve strategic decision-making and increase the AI’s ability to reach higher tile values consistently.

**3. AI Approach and Methodology**

**● AI Techniques to be Used:**

* Heuristic-Based Decision Making: The AI evaluates different move possibilities based on pre-defined heuristics.
* Minimax Algorithm (Future Consideration): Can be used to explore deeper game states with adversarial thinking.
* Expectimax Algorithm: If implemented, it will consider both deterministic moves and probabilistic tile spawns.
* Reinforcement Learning (Optional Future Work): The AI could be trained via self-play to optimize move strategies.

**● Heuristic Design:**

* Snake Pattern Scoring: Rewards game states where higher-value tiles are arranged in a snake-like pattern.
* Adjacent Tile Clustering: Encourages merging of similar tiles to optimize space.
* Empty Tile Count: Prefers game states with more empty spaces to allow future moves.

**● Complexity Analysis:**

* The brute-force evaluation of all possible moves can be computationally expensive.
* Using heuristics significantly reduces computation time by prioritizing more promising moves.
* If Expectimax is implemented, complexity increases as multiple future states are simulated at each decision step.

**4. Game Rules and Mechanics**

**● Modified Rules:**

* The game is played autonomously by an AI agent instead of a human.
* The AI selects moves using heuristic evaluations.
* Additional Expectimax or learning-based decision-making may be introduced.

**● Winning Conditions:**

* The AI aims to reach the highest possible tile value before the board fills up.
* Performance will be measured based on maximum tile achieved and average game duration.

**● Turn Sequence:**

* The AI evaluates the current board state.
* It determines the best possible move based on heuristics.
* The move is executed, and a new tile appears randomly on the board.
* The process repeats until no valid moves remain.

**5. Implementation Plan**

**● Programming Language:**

* **Python**

**● Libraries and Tools:**

* NumPy (for grid operations and AI computations)
* Matplotlib (optional) (for visualizing AI performance)
* Pygame (optional) (for rendering a playable interface)

**● Milestones and Timeline:**

| **Week** | **Task** |
| --- | --- |
| 1-2 | Game design and heuristic selection |
| 3-4 | AI algorithm development (heuristics, move selection) |
| 5-6 | Coding and testing the game mechanics |
| 7 | AI integration, debugging, and performance tuning |
| 8 | Final testing and report preparation |

**6. References**

* Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach (3rd Edition)
* Online resources on heuristic algorithms and 2048 AI strategies
* Research papers on Expectimax and game AI

1. Multistage Temporal Difference Learning for 2048-Like Game

<https://ieeexplore.ieee.org/document/7518633>

1. Optimistic Temporal Difference Learning for 2048

<https://arxiv.org/pdf/2111.11090>