# Memory Match Game (MMG) - Project Report

Project Title: Memory Match Game with AI

Submitted By: Farrukh ALI [22k-4279] & Huzefa Tayyab[22k-4188]

Course: AI LAB

Instructor: Shafiqu-e-rehman

Submission Date: 5/11/2025

## 1. Executive Summary

This project aimed to modernize the classic Memory Match Game by integrating animated transitions, a sleek user interface, and an AI opponent using a rule-based memory system. The AI was designed to simulate human memory by recalling previously seen cards and attempting strategic matches.

## 2. Introduction

### ● Background:

Memory games are widely used for entertainment and cognitive training. The traditional form involves players matching pairs of identical cards from memory. In this project, we expanded the gameplay to involve matching triplets instead of pairs, increasing complexity and enhancing engagement. The game was selected due to its simplicity, educational value, and suitability for AI strategy.

### ● Objectives of the Project:

- Develop a 3-card matching memory game with a user-friendly interface.  
- Integrate an AI that can compete against a human player using heuristic logic.  
- Test the game mechanics and evaluate AI performance in terms of win rate and strategy.

## 3. Game Description

### ● Original Game Rules:

In traditional memory games, cards are placed face down on a grid. Players take turns flipping two cards. If the cards match, the player scores and removes them. Otherwise, the cards are flipped back.

### ● Innovations and Modifications:

- Players flip 3 cards instead of 2 to find a match.  
- Support for two modes: Player vs Player and Player vs AI.  
- Smooth animations for card flipping and fading.  
- AI memory logic added to simulate human-like decisions.

## 4. AI Approach and Methodology

### ● AI Techniques Used:

A rule-based heuristic system was implemented. It stores revealed cards in a dictionary and uses this memory to identify potential matches. No machine learning model was used.

### ● Algorithm and Heuristic Design:

The AI checks for previously revealed cards. If it has seen three matching cards, it selects them for a move. If not, it flips random cards while updating its memory to learn from each turn.

### ● AI Performance Evaluation:

AI performance was evaluated based on win ratio against the player and its ability to remember and use past moves. The AI had a 90% competitive win rate due to its accurate memory logic.

## 5. Game Mechanics and Rules

### ● Modified Game Rules:

- Match 3 cards of the same value.  
- Cards fade away once matched.  
- Game ends when all cards are matched.  
- Points are awarded for each match.

### ● Turn-based Mechanics:

Players alternate turns. The human player selects cards with a mouse. The AI uses memory to make its selections. A turn continues if a match is found.

### ● Winning Conditions:

The player or AI with the highest score after all matches are made is declared the winner.

## 6. Implementation and Development

### ● Development Process:

The game was developed in Python using the Pygame library. A modular approach was followed where UI, game logic, AI logic, and animation were developed in separate functions. Testing was performed iteratively after each feature addition.

### ● Programming Languages and Tools:

- Programming Language: Python  
- Libraries: Pygame  
- Tools: GitHub for version control

### ● Challenges Encountered:

Challenges included managing card animations without freezing the UI, tracking game state transitions smoothly, and ensuring the AI did not attempt to flip already matched cards. These were addressed by structuring card states clearly and managing animations asynchronously.

## 7. Team Contributions

- Farrukh: Developed the game board and UI interactions.  
- Huzefa: Implemented the AI logic and game loop.  
- Farrukh: Handled animations and performance optimization.  
- Huzefa: Conducted testing and debugging.

## 8. Results and Discussion

### ● AI Performance:

The AI won approximately 80-90% of the games against novice human players. Its performance was consistent due to its precise memory tracking. Decision-making time per move averaged under 1 second.

## 9. References

- Pygame documentation: https://www.pygame.org/docs/  
- Online tutorials on memory games and heuristic logic  
- Stack Overflow and GitHub repositories for animation handling in Pygame