

**University of Sulaimani**

**College of Science**

**Computer Science**



# Memory Puzzle Game – Project

**Prepared by:**

**Lawand Rebwar Othman**

**Mohammed Abubakr Abdulla**

**Warida Pshtewan**

**Dene Ayub**

**Supervised By:**

**Ramyar Abdulrahman Teimoor Ismael**

# Contents

- **Introduction**
- **Game Overview: Match the Fruits**
  - **Key Features**
  - **Objective and Rules**
  - **Game Mechanics**
- **AI Algorithm Used: Rule-Based Logic**
  - **Game Flow Description**
- **Programming Language and Tools**
  - **Conclusion**



# Introduction

The Memory Puzzle Game is a single-player card-matching puzzle developed using Python. It challenges players to match pairs of hidden fruit cards within a time limit, enhancing memory, concentration, and logical thinking skills.

This report details the game's development, core features, AI algorithms, and overall design, demonstrating practical application of programming and algorithmic concepts.



## Game Overview: Match the Fruits!

The game presents a grid of face-down cards, each concealing a unique fruit image. Every fruit appears exactly twice. Players' objective is to flip two cards at a time to find matching pairs.

The game features multiple levels, with difficulty progressively increasing by adding more card pairs, ensuring a continuous challenge for players.

# Key Features

## Multiple Difficulty Levels

Progressive challenges with increasing card pairs.

## Countdown Timer

Time limits for each level add urgency and excitement.

## Hint System

Assistance for players when they get stuck.

## Pause & Resume

Flexibility to control gameplay at any moment.

## Sound & Animations

Engaging audio-visual feedback for an immersive experience.

# Objective and Rules

## Objective

The primary goal is to match all card pairs before the level's timer expires.

## Rules

- Players can flip two cards per turn.
- Matching cards remain face-up.
- Non-matching cards automatically flip back down.





**Game Over**

**Restart**

# Game Mechanics

## 1 Match All Pairs

All card pairs must be matched within the given time limit to clear the level.

## 2 Increasing Difficulty

Each subsequent level introduces more cards, increasing the challenge.

## 3 Limited Hints

A finite number of hints are available to assist players.

## 4 Time-Out Penalty

If the timer runs out, the game restarts from level one.





# AI Algorithm Used: Rule-Based Logic

The game employs rule-based artificial intelligence logic, not machine learning, to govern game mechanics. This logic dictates card behavior, matching decisions, hint functionality, and level progression.

## Algorithm Concepts

- Random card shuffling for varied gameplay.
- State tracking to monitor card positions and status.
- Conditional match checking for pair verification.
- Hint detection logic to guide players.
- Timer-based control for game progression and limits.



# Game Flow Description

1

Start Game

Initiates the game sequence.

2

Initialize Level & Timer

Sets up the current level and countdown.

3

Shuffle Cards

Randomizes card positions.

4

Display Cards

Presents the face-down card grid.

5

Player Selects Cards

User interaction to flip two cards.

6

Match Verification

Checks if selected cards form a pair.

7

Update Game State

Reflects changes like matched cards or timer progress.

8

Level Completion/Restart

Determines next action based on game state.

9

End Game Summary

Displays final score and performance.

# Programming Language and Tools

The Memory Puzzle Game is developed entirely in Python, leveraging a suite of powerful libraries and tools to create a robust and interactive experience.

## Python

The core programming language for game logic.

## Tkinter

Used for building the graphical user interface (GUI).

## PIL (Pillow)

Handles all image processing and display within the game.

## Random Module

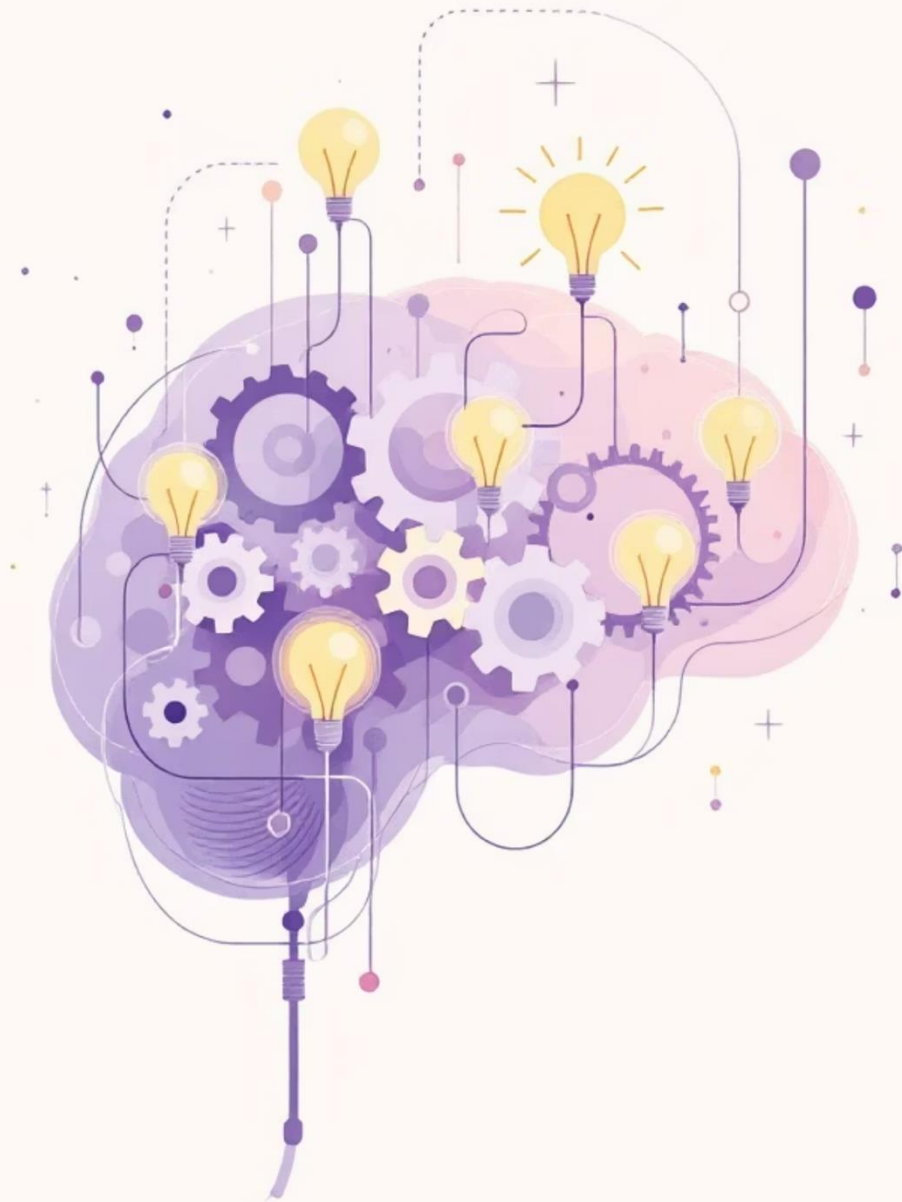
Ensures fair and unpredictable card shuffling.

## Time & After()

Manages the in-game timer and event scheduling.

## Winsound

Integrates sound effects for an enhanced user experience.



## Conclusion

This project effectively demonstrates Python's capabilities in game development, showcasing rule-based AI logic and event-driven programming. It highlights modular design principles, logical thinking, and the practical application of algorithms in an interactive environment.

The Memory Puzzle Game serves as a testament to how fundamental programming concepts can be combined to create engaging and educational software.

# Thank You!

We hope you enjoyed this report on the Memory Puzzle Game. For any questions or further details, please feel free to reach out.

This project was a valuable exercise in bringing theoretical concepts to life through practical application.

