

# Memory Puzzle Game – Project

## 1. Introduction

The Memory Puzzle Game is a single-player card-matching puzzle developed using Python. The game challenges players to match pairs of hidden fruit cards within a time limit. It is designed to enhance memory, concentration, and logical thinking skills while demonstrating core programming and algorithmic concepts.

## 2. Game Overview

The game displays a grid of face-down cards, each containing a fruit image. Every fruit appears exactly twice. Players flip two cards at a time to find matching pairs. The game includes multiple levels, each increasing in difficulty by adding more card pairs.

### Key Features:

- Multiple difficulty levels
- Countdown timer for each level
- Hint system
- Pause and resume functionality
- Sound effects and animations

## 3. Objective and Rules

### Objective:

The primary objective is to match all card pairs before the timer expires.

### Rules:

1. The player can flip two cards per turn.
2. Matching cards remain face-up.
3. Non-matching cards flip back.
4. All pairs must be matched before time runs out.
5. Each level increases difficulty.
6. Limited hints are available.
7. If time expires, the game restarts from level one.

#### 4. AI Algorithm Used

The game uses rule-based artificial intelligence logic rather than machine learning. This logic controls card behavior, matching decisions, hints, and level progression.

##### **Algorithm Concepts:**

Random card shuffling

State tracking of cards

Conditional match checking

Hint detection logic

Timer-based game control

#### 5. Game Flow Description

**The game follows a structured algorithmic flow:**

1. Start the game
2. Initialize level and timer
3. Shuffle cards
4. Display cards
5. Player selects two cards
6. Match verification
7. Update game state
8. Level completion or restart
9. End game summary

## 6. Programming Language and Tools

**The game is developed entirely in Python using the following tools:**

Tkinter for GUI

PIL (Pillow) for image handling

Random module for shuffling

Time and after() for timer management

Winsound for sound effects

## 7. Conclusion

This project demonstrates effective use of Python for game development, rule-based AI logic, and event-driven programming. It highlights modular design, logical thinking, and practical application of algorithms in an interactive game environment.