**Memory Matching Game (Python + Pygame)**

**TEAM MEMBERS-**

**Gitika Mittal (24067)**

**Vanshika Gupta (24077)**

**Isha Garg (24079)**

**Moksha Dutt (24084)**

**Manan Bansal (24088**)

**1.Included Files**

| **File Name** | **Description** |
| --- | --- |
| memory\_game.py | Main game script containing the full implementation of the memory match game. |
| README.md (optional) | Describes the project purpose, how to run the game, features, and dependencies. |
| requirements.txt (optional) | Contains a list of required packages (e.g., pygame==2.5.2). |
| assets/ (optional) | Folder for game assets like icons, sound effects, or custom card designs. |

**2. Project Description & Setup Instructions**

**Project Objective:**

This is a Memory Matching Game coded in Python using the Pygame library. It is meant to enhance memory and pattern recognition skills by turning over and pairing two hidden cards. The level of difficulty increases in each level by growing the grid size and a little bit more time allowed to the player.  
It involves basic programming concepts combined with game design principles such as UI feedback, timer challenge, and increasing difficulty.  
  
**Setup Instructions:**

**Step 1: Install Library**  
Open terminal or command prompt and use:  
pip install pygame  
**Step 2: Run the Game**  
Move to the library with memory\_game.py and use:  
python memory\_game.py

**3. Key Features & Functionality**

**Game Mechanics**  
• The player clicks the two cards to flip them.  
• If they are a match, they stay revealed.  
• If not a match, they flip back after a short delay.

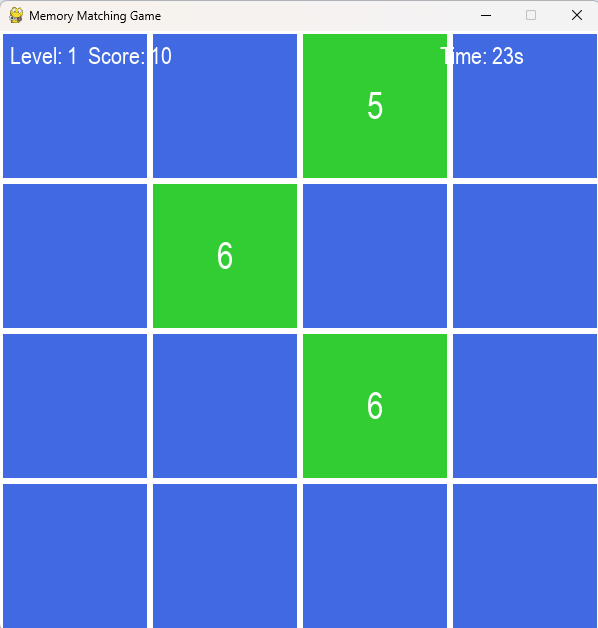
**Timer & Challenge**  
• Each level employs a countdown timer.  
• Timer starts at 35 seconds on Level 1, with the timer increasing by +5 seconds for every level.  
• Game is forfeit when time runs out before all pairs are matched.

**Scaling of Difficulty**  
• Grid size scales dynamically with levels (e.g., 4x4 → 6x6 → 8x8, etc.).  
• This creates additional pairs, increasing cognitive load and enhancing difficulty in gameplay.

**Score System**  
• Get +10 points for every correct pair.  
• Score is updated in real time and appears on the Win or Time's Up screens.

**User Interface**  
• Clean but efficient GUI with color-coded cards.  
• Blue: Hidden card  
• Green: Face-up card  
• White border for distinction

**Game States**  
• Start screen: Click to start game.  
• Win screen: Appears after completing all levels.  
• Timeout screen: Appears when countdown is 0.  
• All screens include an option to restart the game.



**4. Optional Enhancements (Future Scope)**

| **Feature** | **Benefit** |
| --- | --- |
| Sound effects | Flip sound, match chime, countdown warning |
| High Score Tracker | Saves top scores to a file or database |
| Save/Load feature | Resume from where player left off |
| Image Cards | Replace numbers with image tiles |
| Menu & Settings | Toggle sound, change difficulty, themes, etc. |

**5. Educational Value**

• Demonstrates core OOP and logic-building with Python.  
• Utilizes event-driven programming with Pygame.  
• Implements 2D coordinate math and array manipulation.  
• Encourages clean code structure and user-centered design