



TITLE: MEMERY CARD MATCHING GAME

GROUP MEMBER:

- SANIA ARSHAD (098)
- EMAN WAHEED (027)
- MALAIKA AROOJ (057)
- MARYAM(061)

SEMESTER: 3(A)

DEPARTMENT: COMPUTER SCIENCE

SUBJECT: ARTIFICAL INTELLIGENCE

INSTRUCTOR: QURATULAIN MAJID



1. Introduction

1.1 Background

The **Memory Card Matching Game** (also known as **Concentration**) is a classic card-matching puzzle where the objective is to find pairs of identical cards. Traditionally, it is played with physical cards, and the challenge lies in remembering the position of the cards as they are flipped over. This project transforms the classic game into a **digital format**, offering modern interactive features like move tracking, a timer, and a restart button.

Using **Python** and **Tkinter**, the game is designed to enhance user experience with an interactive **graphical interface**, which is more engaging compared to the traditional paper-and-card setup. Players now get instant feedback through timers and move counters, making it more competitive.

1.2 Objective of the Project

The primary goals of the project are:

- **Modern Features:** Adding interactive features such as a **timer**, **move counter**, and **restart button** that make the game more dynamic. This enables players to track their progress and enhances replay value.
- **Engaging Interface:** The game will provide an intuitive, visually appealing interface suitable for a wide range of players.
- **Cognitive Development:** By introducing a **move limit** and a **timer**, the game challenges players to think strategically, improving cognitive skills like **memory**, **attention**, and **problem-solving**.
- **Future Scalability:** The game's design allows for the addition of more advanced features, such as multiplayer modes, AI players, and even **different difficulty levels** (e.g., larger grids for more challenge).

1.3 Importance of Cognitive Skills

The game specifically enhances key cognitive abilities:

- **Attention:** Players must remain focused to remember the card locations. The timer adds time pressure, further enhancing attention span.
- **Problem-solving:** Players need to plan their moves to minimize unnecessary flips. This reinforces planning and decision-making skills.
- **Memory:** By frequently playing the game, players exercise both **short-term** and **long-term memory**, which is critical for recalling card locations and learning from past moves.

2. Methodology

2.1 Planning and Design

2.1.1 Game Concept and Design

The concept involves recreating the classic **Memory Card Matching Game** using Python. The player will flip two cards at a time to find matching pairs. The game will include features like **move tracking**, **timing**, and a **restart button** to make the game more engaging and competitive.

2.1.2 Layout and Features

The game features a **4x4 grid**, but the grid size can be dynamically changed for different difficulty levels (e.g., 6x6 for advanced players).

- **Card Grid:** Initially, all cards are face-down. The cards are represented by numbers, where each number appears twice in the grid to form pairs.
- **Move Counter:** This counter will increment each time the player flips a card, keeping track of the player's moves. This adds a layer of strategy to the game.
- **Timer:** Displays how long the player takes to complete the game. The time adds a competitive edge and motivates faster play.
- **Restart Button:** If players want to start a new session, they can click the restart button, which reshuffles the cards, resets the timer, and resets the move counter.

2.1.3 Tools and Technologies

The game is developed using the following tools:

- **Python:** Chosen for its simplicity, readability, and rich libraries for quick development.
- **Tkinter:** The **standard Python library** for creating **graphical user interfaces** (GUIs). Tkinter allows the creation of **buttons**, **labels**, and **windows** necessary for the game interface.
- **Random Module:** Used for shuffling the cards so that each game is unique and provides an unpredictable challenge.

2.2 Game Design and Flow

1. Card Logic:

- **Card Generation:** Cards are represented by numbers (e.g., 1-8 for a 4x4 grid), and these numbers are duplicated to create pairs.
- **Shuffling:** Using the `random.shuffle()` method, the cards are shuffled at the start of each game to ensure that every session is different.
- **Matching:** The game checks if two flipped cards match. If they do, the cards stay face-up; otherwise, they flip back after a short delay, giving players a chance to remember their positions.

2. User Interface:

- **Game Window:** Created using Tkinter, the window houses all interactive components like the cards, timer, move counter, and restart button.
- **Card Display:** Initially, each card button is marked with a placeholder (e.g., "*"). Once clicked, the actual number is revealed.

- **Game Feedback:** The game continuously updates the move counter and timer, and once the game ends, feedback is provided (win or loss).

3. Game Flow:

- The player clicks on two cards per turn. If they match, the cards stay visible; if not, they are flipped back after a short delay.
- **Win Condition:** The player wins if all pairs are matched before reaching the move limit (e.g., 20 moves).
- **Loss Condition:** The player loses if the move limit is exceeded without matching all pairs, and a "Game Over" message is shown.

2.3 Implementation

- **Card Shuffling and Display:** Cards are represented by numbers stored in a shuffled list, and each card is displayed as a button in the grid. When clicked, the card's value is revealed, and if two cards match, they remain visible; otherwise, they are flipped back.
- **Move Counter and Timer:** Every time a card is flipped, the move counter increases, encouraging players to think strategically. The timer starts at the beginning of the game, showing how long it takes to complete, adding a sense of urgency.
- **Win and Loss Conditions:** If the player matches all pairs within the allowed number of moves (e.g., 20), they win. If they exceed the move limit without matching all pairs, the game triggers a "Game Over" message.

3. Testing

3.1 Testing Plan

1. Functionality Testing:

- **Card Shuffling:** Ensure that cards are shuffled properly, and no two cards appear in the same position in different games.
- **Card Flipping:** Verify that each card shows the correct value when clicked and flips back if not matched.
- **Move Tracking:** Confirm that the move counter increments correctly after every flip.
- **Game Outcome:** Verify that win and loss conditions trigger accurately after the game ends.

2. Performance Testing:

- Test the game on various devices, ranging from low-end to high-end, to ensure it runs smoothly without lag, even with multiple cards.

3. Usability Testing:

- Collect user feedback on interface design, game clarity, and ease of use. Conduct user trials to ensure that players can easily understand and enjoy the game.

4. Edge Case Testing:

- Test edge cases such as rapid card clicking, flipping cards after the game has ended, or restarting the game during an active session. The game should be able to handle these without crashing.

3.2 Results

Functionality:

- **Card Shuffling:** The randomization of cards was thoroughly tested. The `random.shuffle()` function was used to ensure that each game started with a different layout of cards. This ensures that no two games are alike and that the player cannot memorize card positions across multiple sessions.
- **Card Flipping:** Each card flip was checked to confirm that the correct card value is revealed when clicked. Additionally, if the player flips two cards and they don't match, the cards are flipped back automatically after a brief delay. This functionality was tested for consistency, ensuring that cards display and hide correctly.
- **Move Tracking:** The move counter was tested to verify that each flip of a card increments the move counter properly. The game tracks every player action accurately and provides real-time feedback on how many moves the player has made, contributing to the strategic aspect of the game.
- **Timer Updates:** The timer was confirmed to start as soon as the game begins and accurately tracks the time spent by the player. The timer updates every second, which adds a layer of challenge and urgency to the game.

Interface:

- The game's **user interface** (UI) was designed to be intuitive and straightforward. The feedback from users who tested the game indicated that they were able to understand and start playing the game without needing additional instructions. This suggests that the interface is user-friendly.
- **Ease of Use:** Users commented positively about the layout of the game window, the visibility of the cards, and the accessibility of the restart button and move counter. The buttons for each card were responsive, and there were no issues with layout or navigation.

Performance:

- The game was tested on a variety of devices, ranging from **low-end** smartphones and computers to **high-end** devices. It was confirmed that the game runs smoothly across all platforms without lag, even when playing with larger grid sizes (e.g., 6x6 or 8x8 grids).
- **Device Compatibility:** The game's performance was consistent on both low and high-end devices, ensuring that the game is accessible to a wide range of users. It was found that the

game does not require high processing power, making it suitable for casual gaming on devices with limited resources.

Summary of Results:

- **Core functionalities** such as card shuffling, card flipping, move tracking, and timer updates were confirmed to work as expected.
- The **interface** was tested for usability and was found to be easy to navigate, even for first-time players.
- **Performance** was optimized for a variety of devices, and the game ran smoothly without lag, making it accessible for a broad audience.

4. Conclusion

4.1 Achievements

- The game was successfully implemented with all core features: card shuffling, move tracking, timer, and win/loss conditions.
- Positive feedback highlighted the game's smooth performance, intuitive interface, and engaging mechanics.
- The game is suitable for players of all ages, making it both entertaining and beneficial for cognitive development.

4.2 Future Enhancements

1. **Multiplayer Mode:** Introduce a multiplayer feature where two or more players can compete against each other in real time. Players can either take turns matching pairs, and the winner is the one with the highest number of matches, or work cooperatively to clear the grid within a set time or move limit.
2. **Dynamic Grid Sizes:** Expand the game's difficulty options by providing grid sizes like 6x6, 8x8, or even custom layouts. Larger grids can challenge the player's memory and make the game more engaging for advanced players.
3. **Leaderboard and Achievements:** Create a leaderboard to rank players based on their performance, such as fewest moves, fastest time, or highest scores. Introduce achievements like "Perfect Match" for no incorrect flips or "Speedster" for finishing under a specific time.
4. **AI Opponent:** Add an AI player that can simulate a challenging opponent in single-player mode. The AI could use memory-based logic to find matches, allowing players to test their skills against an intelligent system.
5. **Themed Card Designs:** Incorporate customizable themes for cards, such as animals, fruits, or holidays, to keep the game visually fresh and appealing. Seasonal updates or user-created themes can be introduced for added engagement.