Ex. No. : 1 Date: 06.11.2025

Register No.: 241701039 Name: NEHA S

Memory Match Game using Python Tkinter

AIM:

To design and develop a Memory Match Game using the Tkinter GUI toolkit in Python, where players flip cards to find matching pairs, enhancing their memory and concentration skills through an interactive interface.

Procedure:

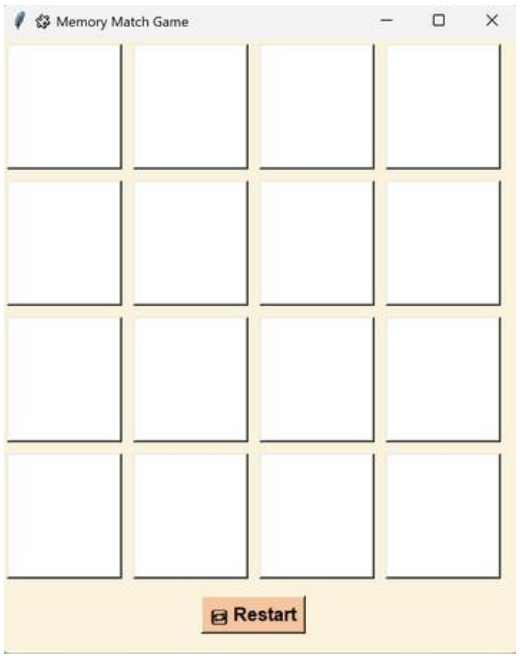
1. Import Required Modules
2. Initialize the GUI Window
3. Prepare Game Data
4. Create Button Grid
5. Implement Click Logic
6. Add Reset Functionality
7. Winning Condition 8. Run the Application

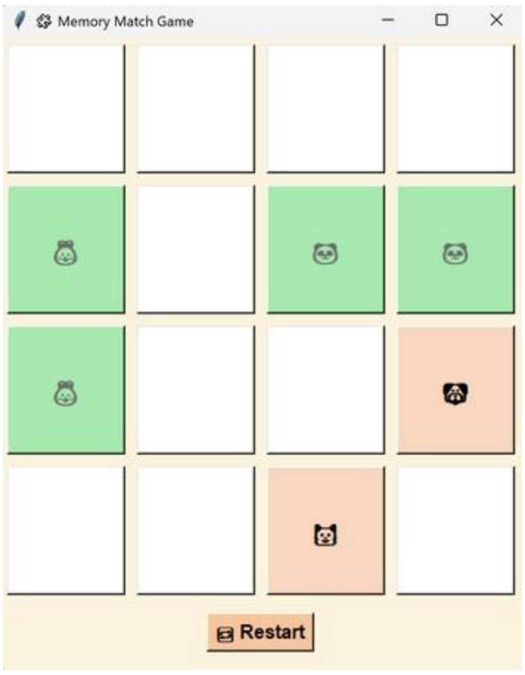
Program:

```
import tkinter as tk import
random
from functools import partial from
tkinter import messagebox
```

```
# Create main window
root = tk.Tk()
root.title("🧩 Memory Match Game")
root.geometry("400x400")
root.config(bg="#faf3dd")

# Emoji pairs (each appears twice)
emojis = ['🐶', '🐱', '🐰', '🐼', '🦊', '🐸', '🐻',
'🐭'] pairs = emojis * 2
random.shuffle(pairs)

# Track game state
first_click = None
second_click = None
buttons = [] matched
= set()

# Reset game function def
reset_game():
   global first_click, second_click, matched, pairs
random.shuffle(pairs)     first_click = second_click
= None     matched.clear()     for i, btn in
enumerate(buttons):        btn.config(text="",
state="normal", bg="#fff")

# Check if all matched def
check_win():     if len(matched)
== len(pairs):
      messagebox.showinfo("🎉 Congrats!", "You matched all pairs!")
reset_game()

# Handle click def on_click(i):
global first_click, second_click

   btn = buttons[i]     if i in matched or
btn["state"] == "disabled":        return
```

```
btn.config(text=pairs[i], bg="#f7d6bf")  if not
first_click:
 first_click = i  elif not second_click and i !=
first_click:
 second_click = i
 root.after(800, check_match) def
check_match():
 global first_click, second_click  if
pairs[first_click] == pairs[second_click]:
 buttons[first_click].config(bg="#a7e9af", state="disabled")
buttons[second_click].config(bg="#a7e9af", state="disabled")
matched.add(first_click)  matched.add(second_click)  else:
 buttons[first_click].config(text="", bg="#fff")
buttons[second_click].config(text="", bg="#fff")
first_click = None  second_click = None  check_win()

# Create 4x4 grid of buttons for i in
range(16):
 btn = tk.Button(root, text="", width=6, height=3, font=("Arial", 18, "bold"),
bg="#fff", command=partial(on_click, i))  btn.grid(row=i // 4, column=i % 4, padx=5,
pady=5)  buttons.append(btn)

# Restart button reset_btn = tk.Button(root, text="🔁 Restart", font=("Arial",
12, "bold"),  bg="#f7c59f", command=reset_game) reset_btn.grid(row=5,
column=0, columnspan=4, pady=10) root.mainloop()
```





Result:

The Memory Match Game was successfully developed using Python's Tkinter library.