Aufbau unseres Codes – Meme Master

1. **Installation von erforderlichen Bibliotheken**

* Das Programm verwendet **os, random, Tkinter, Pillow (PIL),** sowie **Emoji**, welche wir via **Terminal** und *pip install* heruntergeladen haben

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **os** | **random** | **tkinter** | **PIL** | **Emoji** |
| Interaktion mit dem Betriebssystem, in diesem Fall Zugriff auf die Dateien der Memes | Zufällige Ausgabe der Memes | Grafische Benutzeroberfläche (GUI) mit Fenstern, Buttons, Eingabefeldern und mehr | Bildbearbeitung zum Bilder laden, bearbeiten, konvertieren und speichern | Zugriff auf Emojis (Kurzcodes oder Unicode) |

* Wir nutzen jeweils noch spezielle Module von Tkinter und PIL, welche über „*from* \_\_\_ *import*“ installiert werden

***from tkinter import***

|  |  |  |  |
| --- | --- | --- | --- |
| **filedialog** | **colorchooser** | **font** | **messagebox** |
| Öffnen/Speichern-Dialoge | Farbauswahl-Dialog | Schriftarten verwalten | Pop-up-Nachrichtenfenster |

***from PIL import***

|  |  |  |
| --- | --- | --- |
| **Image** | **ImageTk** | **ImageOps** |
| Hauptmodul zum Laden, Bearbeiten und Speichern von Bildern | Ermöglicht die Verwendung von PIL.Image in tkinter als Tkinter-kompatibles Bildformat | Bietet verschiedene Bildbearbeitungsfunktionen wie Spiegeln, Zuschneiden und Farboperationen |

1. **Vorbereitung der Meme-Dateien:**

* Wir haben mehrere Ordner auf dem Computer, in denen Memes gespeichert sind. Diese Memes sind nach verschiedenen Kategorien unterteilt, z.B.:

**CATEGORIES**: Ein Dictionary

* Tiere (mit Unterkategorien wie Hunde, Katzen, Wildtiere)
* Emotionen (z.B. Glücklich, Traurig, Verwirrt)
* Gaming, IT, Kochen, Musik, usw.

* Wir haben die Ordner **organisiert** und sichergestellt, dass die **Dateien korrekt benannt** sind, damit sie später leichter gefunden werden können
* Im Code greifen wir auf die hier erstellten Ordner über ein **Dictionary** zu

**3. Erstellen der Benutzeroberfläche (GUI):**

- Wir haben eine grafische Benutzeroberfläche (GUI) mit **Tkinter** erstellt

* ein Hauptfenster, das die Anwendung darstellt.
* Im Hauptfenster ein Textfeld (für die **Suchfunktion**) und Buttons -> Die Buttons zeigen verschiedene **Kategorien** und **Unterkategorien** der Memes an
* **root**: öffnet das Hauptfenster der Tkinter-Anwendung
* **search\_var und search\_entry**: Ein Textfeld, in das der Benutzer eine Suchanfrage eingeben kann, um nach Memes zu suchen
* **search\_button**: Ein Button, der die Suchfunktion auslöst

**4. Funktionen im Detail:**

**show\_random\_meme:**

Diese Funktion zeigt ein zufällig ausgewähltes Meme aus einem gegebenen Ordner an

* Wenn eine **Unterkategorie** übergeben wird, wird der entsprechende Ordner aus dem Dictionary **CATEGORIES** ausgewählt
* **random.choice(memes)** wird verwendet, um zufällig ein Meme aus dem Ordner auszuwählen
* Das Bild wird geöffnet, auf 500x500 Pixel skaliert und im Tkinter-Canvas angezeigt

**show\_categories:**

Diese Funktion zeigt alle Hauptkategorien an (z.B. "Tiere", "Emotionen", "Gaming")

* Für jede Kategorie wird ein Button erzeugt. Wenn der Benutzer auf eine Kategorie klickt, wird die Funktion **show\_subcategories** aufgerufen (falls es Unterkategorien gibt) oder **show\_random\_meme** direkt ausgeführt.

**show\_subcategories:**

Diese Funktion zeigt die Unterkategorien für eine ausgewählte Hauptkategorie an

* Für jede Unterkategorie wird ein Button erzeugt, der wiederum ein zufälliges Meme aus der gewählten Unterkategorie anzeigt

**search\_memes:**

Die Funktion wird ausgeführt, wenn der Benutzer im Suchfeld nach einem Meme sucht

* Die Eingabe wird in Kleinbuchstaben konvertiert, um die Suche zu erleichtern
* Der Code durchläuft alle Kategorien und Unterkategorien, um nach Dateinamen zu suchen, die den Suchbegriff enthalten
* Wenn Ergebnisse gefunden werden, werden sie als Buttons angezeigt, die das entsprechende Meme öffnen

**show\_meme:**

Diese Funktion zeigt das Meme an, das der Benutzer durch die Suchergebnisse ausgewählt hat

* Es prüft, ob die Datei existiert, und öffnet dann das Bild im Tkinter-kompatiblen Format

**5. Meme zufällig auswählen:**

Wenn der Benutzer auf eine Kategorie klickt (z.B. **Tiere**), zeigt das Programm zufällig ein Meme aus dieser Kategorie an. Das passiert folgendermaßen:

* Das Programm schaut in den entsprechenden Ordner (z.B. "Tiere\Hunde").
* Es wählt zufällig eine Datei (ein Meme) aus diesem Ordner aus.
* Das Meme wird dann im Hauptfenster angezeigt

**6.** **Zugriff auf Dateien:**

* Der Code verwendet **os.listdir()**, um die Dateien aus den Ordnern zu laden und nach Memes zu suchen.
* **os.path.join()** wird verwendet, um den vollständigen Pfad zu einem Meme zu erstellen

**7. Suchfunktion hinzufügen:**

- wir haben eine **Suchfunktion** eingebaut

- damit kann der Benutzer nach einem bestimmten Meme suchen, indem er ein **Suchwort** in ein Textfeld eingibt (z.B. "Hund").

* Das Programm durchsucht dann alle Meme-Dateien, ob der Dateiname des Memes das Suchwort enthält.
* Wenn es passende Ergebnisse gibt, werden diese als **Buttons** angezeigt, und der Benutzer kann darauf klicken, um das Meme zu sehen.

**8. Meme-Anzeige:**

- wenn der Benutzer ein Meme auswählt, wird dieses Meme in einem speziellen Bereich der Benutzeroberfläche angezeigt

- wir haben **PIL (Pillow)** verwendet, um das Bild im richtigen Format anzuzeigen und es auf die passende Größe zu skalieren

* **frame**: Ein Container, in dem die Buttons für die Kategorien und Suchergebnisse angezeigt werden.
* **meme\_label**: Ein Label, auf dem das ausgewählte Meme angezeigt wird.

**9. Zurück-Button:**

- der Benutzer kann jederzeit zum vorherigen Menü zurückkehren, indem er auf einen **Zurück-Button** klickt

- Dieser Button bringt den Benutzer entweder zu den Unterkategorien oder zurück zu den Hauptkategorien

**10. Fehlerbehandlung:**

- das Programm gibt dem Benutzer eine klare Fehlermeldung, falls etwas schiefgeht:

* Wenn keine Memes in einem Ordner gefunden werden.
* Wenn keine Suchergebnisse zu einem bestimmten Suchbegriff vorhanden sind.
* Wenn ein Meme nicht mehr existiert oder gelöscht wurde.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**11. Zusammenfassung des Arbeitsablaufs:**

* Der Benutzer öffnet die Anwendung und sieht ein Hauptmenü mit Kategorien
* Der Benutzer klickt auf eine Kategorie oder Unterkategorie, um ein zufälliges Meme anzusehen
* Der Benutzer kann auch nach einem Meme suchen, indem er ein Stichwort eingibt
* Gefundene Memes werden als Buttons angezeigt. Wenn der Benutzer auf einen Button klickt, wird das Meme angezeigt
* Der Benutzer kann jederzeit zum vorherigen Bildschirm zurückkehren, um eine andere Kategorie oder Unterkategorie auszuwählen