Celina Syafril Report: Meme Capsule MVP 24.02.2024

Introduction

In an era that is defined by digital connection and the ability to exchange information over the phone in an instant, internet memes have emerged as a cultural phenomenon that transcends linguistic barriers. The Meme capsule MVP's intentions are to remember the memories and associations we have to a certain time period, by providing users with a platform that lets them curate and preserve their favourite internet memes in a digital capsule. Memes continue to be a vital part of our internet life and with the increase in memes throughout the years, the need to archive these artefacts becomes more apparent. While the project aims to create a digital capsule for storing selected memes, this report serves as an exploration of the project by offering insights into the use of Python and libraries for creating a GUI, the design principles, and limitations of the MVP.

Methodology

The Meme Capsule MVP relied on the development of the GUI using two different libraries. Firstly, Tkinter was used and served as the foundation for building the main user interface components. Through Tkinter, I was able to construct the different labels and buttons that guide the user experience and facilitate a seamless navigation and interaction with the functions of the Meme Capsule.

Furthermore, the project incorporated the Python imagining library (PIL) or pillow in order to process and manipulate images as the background etc. The library allowed for the resizing, displaying and managing of the different images within the MVP. Therefore, ensuring the user's seamless selection, viewing and organizing of their favourite memes was done by using the libraries capabilities to its full extent. In terms of design, I broke down the application into separate pages dedicated to specific tasks. By doing this, I aimed to create a more organized experience for users and as a benefit made the code easier to manage.

Design

When discussing the Meme Capsule MVP's design, one is able to make out that the focus was on simplicity and usability. Like already mentioned, Tkinter was used for the interface, meaning that labels and buttons were the main interactive elements. The key to the design is the organization of the different functions into separate pages. Users are welcomed with the background image that tells them the name of the MVP "Rewind", and allows them to log in and access the other pages. Subsequent pages allow users to select the images and view their creation. The visual aesthetics were considered with special attention to background images and fitting colour schemes of labels.

Implementation

Moreover, a big part of the MVP is the actual implementation of the code. When launching the application, the user is faced with the welcome page, which prompts them to log in with their username. This feature is implemented using the *'handle_login ()'* function, which retrieved the entered username from the entry box. The username is then compared against a predefined set of allowed usernames, celina and sarah, using a conditional statement. Depending on the result, the 'result_label' is updated to display the corresponding message. This is able to viewed in lines 26 to 52 of the code and applies the knowledge learned from tech basics 1 on if-else statements. It is important to acknowledge that the login code was inspired from my Tech Basics 1 project, as I felt it made sense to use the same feature in this MVP.

Once the user has registered, the user can proceed to the main function of the MVP: creating the meme capsule and eventually locking it. This page is facilitated through the 'createpage()' function in line 75, which transfers the user interface to the capsule creation page('f2'). In the capsule creation page, one will see a grid of meme images that are already loaded into the page, retrieved from the "preloaded_images" partially inspired by the code from the holiday cards from Tech Basics 2 (line 56-69). These memes are displayed as clickable labels through 'img_label' and bound to the 'select_image()' function on lines 143-147, meaning that you can simply click on them to select them for your capsule. As a result, this function curates a capsule by selecting the memes and appending the path to the 'selected_images' list (Line 72). The code for the binding of the label and the function was taken from python tutorial.

Additionally, the create page includes buttons for information, saving and viewing the capsule. When users click the buttons to save the capsule or information, a message box is displayed to either confirm the successful saving of the capsule or inform the user on how to navigate the MVP. Similarly, clicking the "View Capsule" button prompts users with a confirmation message before navigating to the capsule viewing page. I applied the learned code from Tech basics 2 in order to display these message boxes which are pertinent for smooth interaction with the user, as well as the code for labels and button from Tech Basics 1.

Furthermore, the MVP allows the user to seamlessly navigate between the different pages through the 'createpage()' and 'view_capsule()' function, which handle the transition between the homepage, creation page and the capsule viewing page. These functions achieve this by destroying the current page (f1 or f2 on lines 75-78) and replacing it with the corresponding frame for the chosen page.

Finally, the part where users can pick and save memes to their capsule is crucial for the app's purpose. Whenever a user clicks on a meme, the select_image() function saves that

choice in a list called selected_images. When they're ready to save their capsule, hitting the save button triggers the save_capsule() function (Lines 72 &154).

Limitations and Transparency

With the version of the MVP now, there are certain limitations and constraints. First, it would be optimal if there were more advanced features like being able to make multiple capsules and store them in a different viewing page. This would allow the user to be able to see the different capsule they have created, with the possibility of theming the capsules differently. On top of that, the user is limited to staying on the create and viewing page after the welcome page. There is no button that allows the user to return to the previous page. This was a significant challenge for me that I did not find a solution for. I tried making a return button that would bring the user back but nothing worked, even after consulting ChatGPT and online websites I could not make a solution work within my code. But other challenges like, for example, importing the PIL library I did manage to tackle. One significant challenge I encountered was that PyCharm could not recognize my imported libraries from my virtual environment. Although, after deleting the outdated Pythons that I didn't need, this problem was solved. Another big issue was figuring out how I could preload the images and make them clickable, but after searching on the recommend website “Python tutorial”, I managed to bind the button with the function. Overall, when I started the project, I knew I wanted to use as much of the concepts covered in both of the tech basics courses as possible. But once I started coding, I realized that I could not directly use the codes for the concept, but had to modify them in order to fit my vision of the MVP. While my visions were big, I quickly realized that I was better off playing to my strengths and not overdoing it and end up confusing myself. Nonetheless, I found certain concepts like the message box to be a great tool for me to utilize in order to explain and facilitate an enhanced user experience. On top of that, did the recommended sites help me find innovative solutions that we hadn't yet covered in class, like the binding situation, and made me excited to keep going.

References:

Reference for binding functions and label:

Python Tutorial. “Understanding Tkinter Command Binding Clearly.” Python Tutorial - Master Python Programming for Beginners From Scratch, November 15, 2023. https://www.pythontutorial.net/tkinter/tkinter-command/.

```
def button_clicked():
    print('Button clicked')
button = ttk.Button(root, text='Click Me', command=button_clicked)
button.pack()
```

Reference for preloading images:

From holiday_card.py

```
# and if you want you can also select a random image
image1 = "images/img1.png"
image2 = "images/img2.png"
image3 = "images/img3.png"
image4 = "images/img4.png"
images = [image1, image2, image3, image4]
```

Reference for messageboxes:

From streamA_online_pet-gui_v1.py

```
 if username.get() in user_ids:
        tk.messagebox.showwarning("WARNING!", "This username already exists")
    # otherwise write the data to the .csv file and do more
```

Inspiration for images and grid:

Python, Real. “Python GUI Programming With Tkinter,” January 30, 2023. https://realpython.com/python-gui-tkinter/#controlling-layout-with-geometry-managers.

```
import tkinter as tk
```

```
window = tk.Tk()

for i in range(3):
    window.columnconfigure(i, weight=1, minsize=75)
    window.rowconfigure(i, weight=1, minsize=50)

    for j in range(0, 3):
        frame = tk.Frame(
            master=window,
            relief=tk.RAISED,
            borderwidth=1
        )
        frame.grid(row=i, column=j, padx=5, pady=5)
            label = tk.Label(master=frame, text=f"Row {i}\nColumn
{j}")
        label.pack(padx=5, pady=5)

window.mainloop()
```