

Compte-Rendu

Tout d'abord, j'ai importé tkinter dans mon fichier python, ainsi que la commande askopenfilename puisque je savais qu'elle me serait utile ensuite.

```
import tkinter as tk
from tkinter.filedialog import askopenfilename
```

Ensuite, j'ai regardé dans le code qu'on a fait en classe pour retrouver comment créer une fenêtre, et j'ai repris cette partie du code en modifiant le titre de la fenêtre.

```
def main():
    window = tk.Tk()
    window.title("Mon Visionneur de Photos")
    window.columnconfigure(1, minsize=500)
    window.rowconfigure(0, minsize=400)
```

J'ai ajouté la ligne de code : `window.mainloop()` pour que la fenêtre reste ouverte.

J'ai ensuite repris la fonction ouvrir du code qu'on a fait en classe et j'ai demandé de l'aide à ChatGPT pour modifier cette fonction pour qu'elle ouvre une image plutôt qu'un fichier texte :



ChatGPT

D'accord, je comprends. Voici la fonction `ouvrir` modifiée pour ouvrir une image avec Tkinter uniquement :

python

Copy code

```
from tkinter.filedialog import askopenfilename
from tkinter import PhotoImage

def ouvrir(window, canvas):
    filepath = askopenfilename(filetypes=[("Fichiers image", "*.jpg;*.png;*.svg")])
    if not filepath:
        return
    image = PhotoImage(file=filepath)
    canvas.create_image(0, 0, anchor='nw', image=image)
    canvas.image = image # Garde une référence à l'image pour éviter la collecte des
    window.title(f"Mon Visionneur de Photos : {filepath}")
```

Je l'ai inclus dans mon fichier python après avoir demandé à ChatGPT ce qu'était « canvas » et comment l'utiliser dans la fonction main(), il m'a répondu avec ce bout de code :

```
# Création d'un canevas pour afficher l'image
canvas = Canvas(window)
canvas.grid(row=0, column=1, sticky="nsew") # Ajustement de la position du canevas

# Bouton pour ouvrir l'image
ouvrir_btn = Button(window, text="Ouvrir", command=lambda: ouvrir(window, canvas))
ouvrir_btn.grid(row=0, column=0, sticky="ew") # Ajustement de la position du bouton
```

J'ai ensuite ajouté ces codes en comparant avec celui qu'on a fait en classe et en le corrigeant pour qu'il corresponde bien au résultat souhaité, et en ajoutant les deux lignes de codes de la frame trouvés dans le code fait en classe :

```
frame = tk.Frame(window, relief = tk.RAISED, bd=2)
frame.grid(row=0, column=0, sticky="ns")

canvas = Canvas(window)
canvas.grid(row=0, column=1, sticky="nsew")

ouvrir_btn = Button(frame, text="Ouvrir", command=lambda: ouvrir(window, canvas))
ouvrir_btn.grid(row=0, column=0, padx=10, pady=5)
```

J'ai ensuite demandé à ChatGPT comment ajouter des boutons pour zoomer et dézoomer en lui écrivant l'entièreté de mon code, voici ce qu'il a expliqué et les deux bouts de codes qu'il m'a donné :



ChatGPT

Pour ajouter des boutons pour zoomer et dézoomer dans ton visionneur de photos, tu peux suivre ces étapes :

1. Ajouter deux nouveaux boutons dans le cadre `frame` pour le zoom et le dézoom.
2. Créer des fonctions pour gérer le zoom et le dézoom des images dans le canevas.
3. Lier ces fonctions aux nouveaux boutons.

```
def zoomer(canvas, scale=1.1):
    canvas.scale("all", 0, 0, scale, scale)

def dezoomer(canvas, scale=0.9):
    canvas.scale("all", 0, 0, scale, scale)
```

```
zoomer_btn = Button(frame, text="Zoomer", command=lambda: zoomer(canvas))
zoomer_btn.grid(row=1, column=0, padx=10, pady=5)

dezoomer_btn = Button(frame, text="Dézoomer", command=lambda: dezoomer(canvas))
dezoomer_btn.grid(row=2, column=0, padx=10, pady=5)
```

J'ai donc inclus ces différents codes dans le mien après avoir vérifié qu'ils fonctionnaient.

J'ai ensuite repris le bouton quit du code qu'on a fait en classe :

```
def quit(window):  
    window.quit()
```

```
quit_button = Button(frame, text="Quitter", command = lambda : quit(window))  
quit_button.grid(row=3, column=0, padx=10, pady=5)
```

Le code ne fonctionnait pas, j'ai donc demandé qu'est-ce qu'était l'erreur à ChatGPT, qui m'a répondu ceci :



ChatGPT

Il semble y avoir quelques problèmes dans votre code. Voici quelques suggestions pour le corriger :

1. Vous importez le module `tkinter`, mais vous n'importez pas correctement la classe `PhotoImage`. Vous devez spécifier le module, soit en ajoutant ``tk.`` devant ``PhotoImage``, soit en important spécifiquement la classe `from tkinter import PhotoImage`.

J'ai compris que j'avais fais la même erreur pour `Button` et `Canvas`, j'ai donc ajouté cette ligne de code :

```
from tkinter import Canvas, Button, PhotoImage
```

Après avoir vu que le code ne fonctionnait pas, ChatGPT m'a expliqué qu'il fallait utiliser une autre bibliothèque pour charger les images car les formats demandés ne marchaient pas sur Tkinter, j'ai donc ajouté cette ligne et j'ai fais les changements nécessaires :

```
from PIL import Image, ImageTk
```

J'ai ensuite continué de régler plusieurs bugs avec l'aide de ChatGPT, notamment pour les boutons zoom et dezoom. Voici ces fonctions terminées :

```
def zoomer(canvas, factor=1.1):  
    canvas.scale("all", canvas.winfo_width()/2, canvas.winfo_height()/2, factor, factor)  
  
def dezoomer(canvas, factor=0.9):  
    canvas.scale("all", canvas.winfo_width()/2, canvas.winfo_height()/2, factor, factor)
```

Le code est terminé.