

J'ai décider pour commencer de reprendre le travail que on a fait en classe j'ai recopier et adapté a mes besoins en ajoutant de plusieurs bouton zoom+ et zoom- ensuite j'ai retiré le bouton save et la fonction qui va avec car elle ne me servait pas.

J'ai ensuite modifier la fonction open fichier dans le but de pouvoir ouvrir des image et non des textes pour qu'elle puisse ouvrir des fichier png et jpg

-j'ai du rechercher sur internet de l'aide pour pouvoir ouvrir un image et j'ai du mettre a jour pip et installer une nouvelle bibliothèque pillow

-j'ai réussi a placer et afficher une image a l'emplacement ligne 0 et colonne 1 cependant si on recharge une nouvelle image elle vont se superposer

pour pallier a ce problème il faut d'abord effacer le contenu précédant de cette case

alors j'ai du aller chercher sur internet et j'ai trouver ma réponse sur le site

<https://stackoverflow.com/questions/23189610/how-to-remove-widgets-from-grid-in-tkinter>

grâce a la fonction `grid_forget`

version 2 :

inconvenient du widget label et qu'il ne peut pas être redimensionner .

Donc j'ai décider de faire une deuxième version de mon code.

Il faut chercher un autre type de widget pour y placer l'image. Dans l'éditeur de texte en classe on utilisait une zone de texte « textarea »

après je suis allée sur internet pour rechercher le bon widget il s'agirait du canvas

source : <https://python-course.eu/tkinter/canvas-widgets-in-tkinter.php>

ça m'a permis de faire la déclaration du canvas

```
mon_canvas = tk.Canvas(window,width=800, height = 600, bg = "#FCFF00")
mon_canvas.grid(row=0, column = 1, sticky="nsew")
```

voilà mon code

j'ai ajouter une couleur

bg veut dire couleur

j'ai choisi le jaune car c'est ma couleur préférée

j'ai ouvert une image et je l'ai inséré dans le canvas a l'aide la fonction `canvas.create_image`

puis j'ai dû coder des calculs pour pouvoir que le zoom marche et j'ai ajouter les command

zoomplus et zoommoins au bouton a l'aide de la fonction `resize` de la bibliothèque pil grace au lien ci

<https://www.geeksforgeeks.org/python-pil-image-resize-method/>

plutôt que d'écrire deux fonction quasiment identique zoomplus et zoommoins ce qui n'est pas optimale j'ai préféré optimiser et réunir les deux fonction en une seule fonction zoom en ajoutant un paramètre précisant si il s'agit d'un zoom en avant ou d'un zoom en arrière car c'est plus pratique voici

. la partie qui change est gérée par un test

```
def zoom(canvas, plus):
    #test permettant de fixer le zoom True -> +10% si False -> -10%
    if config['image_en_cours']:
        if plus:
            config['zoom']=config['zoom']*1.1
        else:
            config['zoom']=config['zoom']*0.9
```

