Ecole IA Microsoft Brest by Simplon

Rapport de l'application Trombinoscope

Auteurs :
Nacira IKHENECHE
Gwenn Le Roch

Référent :
Stéphane Jamin-Normand

4 janvier 2022

Table des matières

1 Introduction

Un trombinoscope est une liste des photographies des membres d'une organisation, d'un groupe, d'une classe ou d'une entreprise.

Dans ce rapport, nous allons décrire une application d'un trombinoscope de la promotion deux, de l'école IA de Microsoft, nomée BREIZH-MEIZ (Intelligence bretonne en français).
Nous donnerons dans un premier temps le guide d'utilisation de l'application, nous présenterons le schéma de la base de données pour ensuite présenter l'application à savoir : l'organigramme et le code python.
Enfin, un bilan de l'apprentissage et des difficultés rencontrées sera dressé.

2 Guide d'utilisation

Cette section sert à présenter le guide d'utilisation de l'application trombinoscope.

2.1 L'interface de l'application

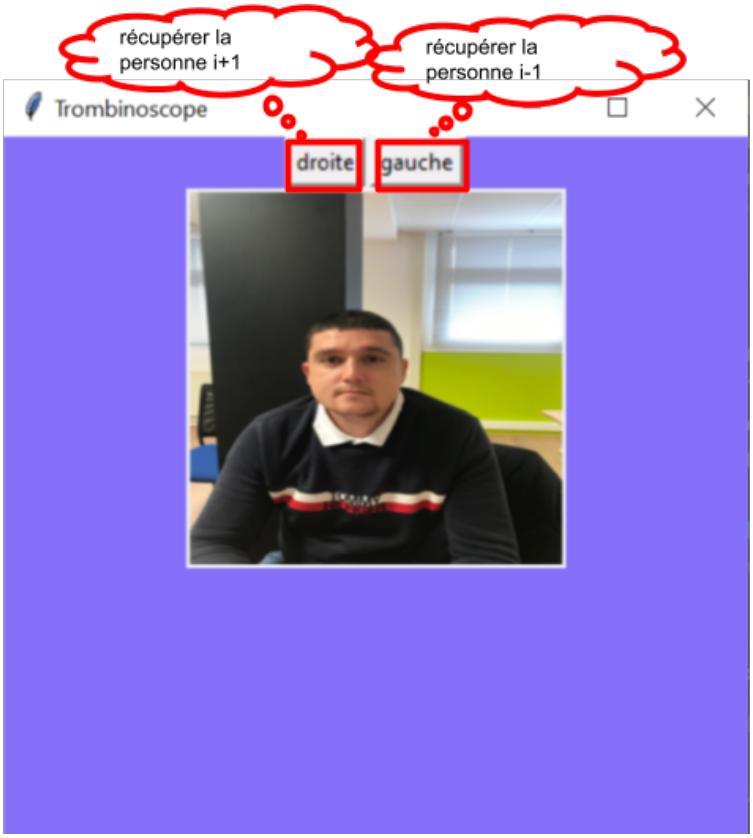

FIGURE 1 – Explication du rôle des boutons

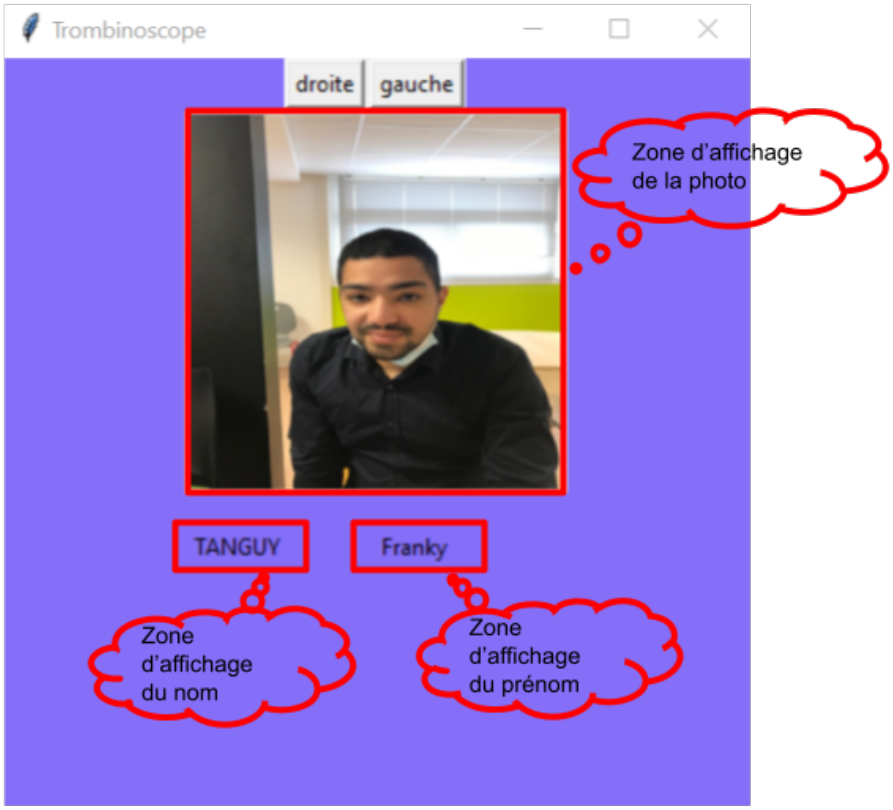

FIGURE 2 – Explication des zones d'affichage

2.2 Fonctionnement de l'application

Lorsque l'utilisateur lance l'application, l'interface dans la figure 1 s'affiche. Elle comporte :

- Le bouton 'gauche' sert à récupérer la personne $i-1$
- Le bouton 'droite' sert à récupérer la personne $i+1$.
- La zone d'affichage de la photo (figure 2)
- La zone d'affichage du nom (figure 2)
- La zone d'affichage du prénom (figure 2)

3 Présentation du schéma de la base de données

Pour réaliser le trombinoscope, il a fallut créer une base de données constituée des trois tables suivantes et reprise dans le schéma de la figure 3 :

- La table personnes, où un enregistrement de cette table est caractérisé par un identifiant unique, un nom, un prénom , une photo, un genre et un statut. Ces deux derniers attributs représentent des clés étrangères vers les tables décrites ci-dessous.
- La table statuts où chaque statut est caractérisé par un identifiant et un label.
- La table genres, de la même façon l'enregistrement genre est caractérisé par un identifiant et un label.

Les tables personnes et genres sont reliées par le lien : id_genre de la table personnes à id_genre de la table genres. De la même façon les tables personnes et statuts sont reliées par le lien : id_statut de la table personnes à id_statut de la table statuts.

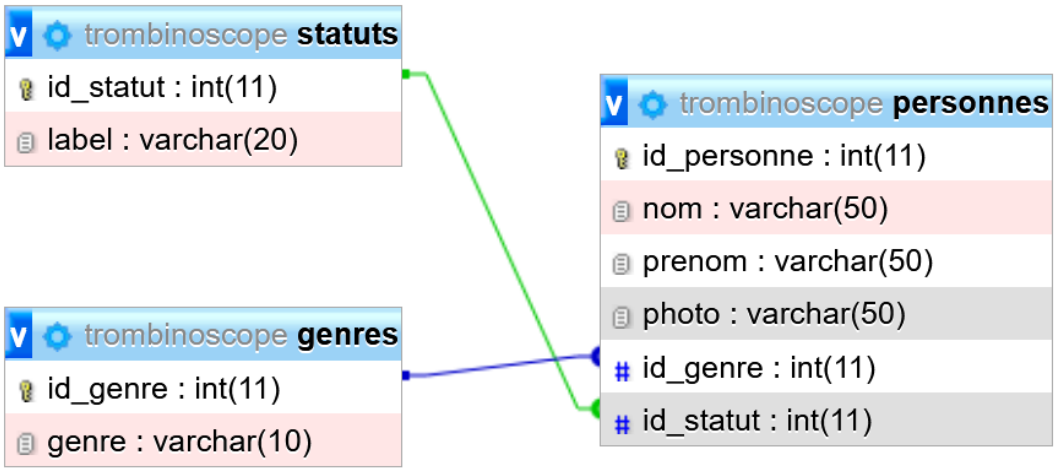

FIGURE 3 – Schéma de base de donnée

4 Présentation de l'application

4.1 Conception : l'organigramme

Nous résumons ici la méthode de l'application dans l'organigramme ci-dessous. Notons que les rectangle représente une action et le losange représente une alternative.

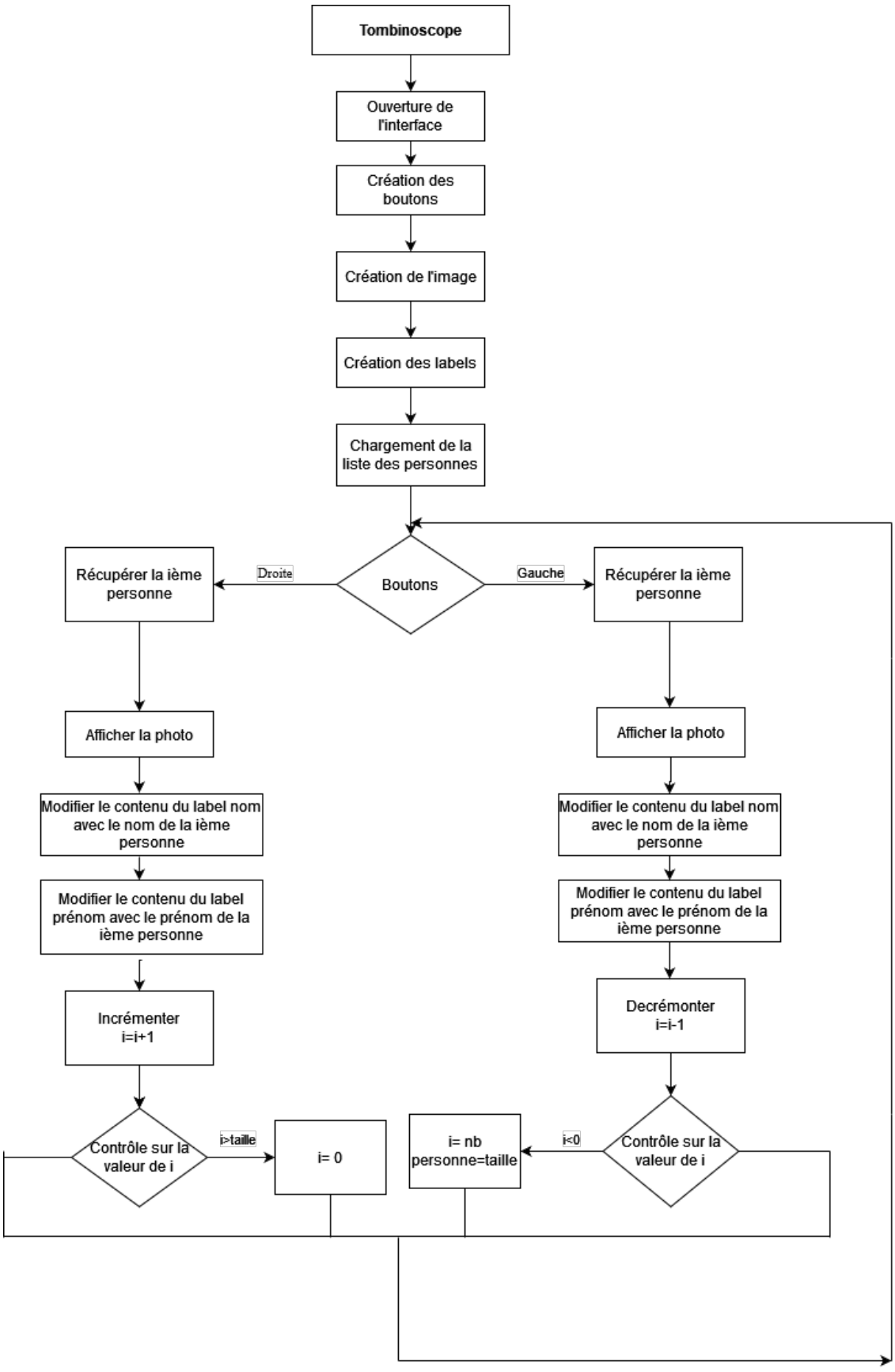

FIGURE 4 – l'organigramme de la méthode

4.2 Réalisation : le code python

Le code est constitué de deux fichiers python :

- Un premier fichier nommé Tromb.py qui est le back-end de l'application dont l'objectif est de créer une liste à partir d'une table de personnes grâce à une fonction qu'on a définit par : **def recupere_liste ()**, qui nous permet la connexion à la base de données, la création du curseur, la création de la requête et l'exécution de celle-ci voir le code (figure 5).

```
import mysql.connector as msc # on a importé un connecteur

def recupere_liste ():#connetion a la base de données
        bdd = msc.connect(user='ISEN', password='ISEN',
                          host='127.0.0.1',port='8081',
                          database='trombinoscope')
        cursor = bdd.cursor()# on a crée un curseur
#qui = 1
#création de la requete
        query = "SELECT nom, prenom, photo FROM personnes;"
        cursor.execute(query)# exécution de l requete

        liste_personnes=[]
# création de liste à partir du résultat d'exéction de la requete
        for enregistrement in cursor :
            liste_personnes.append(enregistrement)
        cursor.close ()
        bdd.close()
        return liste_personnes
```

FIGURE 5 – Tromb.py

- Un deuxième fichier nommé fenetre.py qui est le front-end de l'application dont l'objectif est de créer l'interface graphique de cette dernière grâce à Tkinter.
  Une première partie du code consiste en l'importation des fichiers nécessaires, la récupération de la liste et la création de la fenêtre (voir figure 6).

```
1   import tkinter as tk
2   import Trombi as tb
3   import tromb as tr #importation du fichier tromb,
4   #recuperation de la liste personnes
5   from PIL import Image, ImageTk
6   personnes=tr.recupere_liste()# la recupération de la liste des personnes
7   fenetre =tk.Tk()# creation de la fenetre
8   fenetre.title("Trombinoscope") # titre
9   fenetre.geometry ("400x400")  # la taille de la fenetre
10  fenetre.configure(bg='#856ff8') # mettre une couleur
```

FIGURE 6 – fenetre.py

Quant à la deuxième partie du code, elle consiste en l'initialisation des composantes graphiques : boutons, labels, ainsi qu'à la description de la fonctionnalité de l'application, lorsque les boutons sont cliqués. On retrouve des commentaires explicatifs de chaque instruction dans le code voir les figures 7,8,9 et l'organigramme de la méthode en général.

```
i=0 # initialiser la variable globale
def changer_droite():
    global i
    personne=personnes[i] # recuperer la ieme personne
    #constitution du chemin vers la photo
    chemin="Trombi\\" +personne[2] + ".jpg"
    img = Image.open(chemin) # ouverture de la photo qui se trouve ici
    newsize=(200,200) # redimensionner l'image
    img=img.resize(newsize)
    photo = ImageTk.PhotoImage(img) #affichage de la photo
    #modifier le contenu du label nom(prenom) avec le nom(prenom) de la ieme personne
    string_nom.set(personne[0]) #modifier le contenu du label
    string_prenom.set(personne[1])
    i=i+1 # incrementation
    print(len(personnes))
    # control sur la valeur de i
    if i==len(personnes):
        i=0
    #configure() change l'état d'un composant
    photo_affich.configure(image=photo)
    fenetre.mainloop()
```

FIGURE 7 – fenetre.py

```
def changer_gauche():
    global i
    personne=personnes[i]# personnes c'est toute la liste des personnes
    chemin="Trombi\\" +personne[2] + ".jpg"
    img = Image.open(chemin)
    newsize=(200,200)
    img=img.resize(newsize)
    photo = ImageTk.PhotoImage(img)
    #modifiction de string nom et prenom, set=modifier=affecter
    string_nom.set(personne[0])
    string_prenom.set(personne[1])
    i=i-1
    if i == -1 :
        i=len(personnes)-1
    #configure() change l'état d'un composant
    photo_affich.configure(image=photo)
    fenetre.mainloop()
#bouton_gauche.pack() #ajouter  le bouton
p = tk.PanedWindow(fenetre, orient=tk.HORIZONTAL)
bouton_droite=tk.Button(p,text="droite",command=changer_droite) # créer un bouton
bouton_gauche=tk.Button(p,text="gauche",command=changer_gauche) # créer un bouton
p.add(bouton_droite)
p.add(bouton_gauche)
#*****************************************************************************
```

FIGURE 8 – fenetre.py

```
img = Image.open("Trombi\Loic_Coroller.jpg")
newsize=(200,200)
img=img.resize(newsize)
photo = ImageTk.PhotoImage(img)
photo_affich = tk.Label(fenetre,image=photo)
#Variable de type chaine de charac qu on initiaise au vide
string_nom=tk.StringVar(value="")
string_prenom=tk.StringVar(value="")
#creation de label dans la fentre, couleur bleu, le texte dedans=sting nom, et sa postion
nom=tk.Label(fenetre,bg='#856ff8', textvariable=string_nom).place(x=100,y=250)
prenom=tk.Label(fenetre,bg='#856ff8', textvariable=string_prenom).place(x=200,y=250)
p.pack()
photo_affich.pack()
fenetre.mainloop()#ouverture de la fenetre
```

FIGURE 9 – fenetre.py

5 Bilan

5.1 Conclusion

Dans ce travail, nous avons développé une application trombinoscope de la promotion deux de l'école IA microsoft. Pour ce faire, nous avons suivi les étapes suivantes :

1. créer une base de données des apprenants avec le SGBD PHPMyAdmin.
2. concevoir l'interface graphique de l'application en utilisant la bibliothèque Tkinter (front-end)
3. implémenter la fonctionnalité de l'application (back-end) qui consiste à récupérer les informations de la promotion en utilisant le langage de requêtage SQL à partir d'une fonction implémentée sous python.

Les difficultés rencontrées sont liées à l'apprentissage des nouvelles notions et technologies à savoir : les bases de données, SQL, SGBD, Python, Tkinter.

5.2 Perspectives

Dans le but d'améliorer la qualité et l'optimalité du code, nous suggérons de rendre le programme modulaire en créant une fonction qui regroupe les traitements communs aux deux boutons. Par ailleurs, nous avons pensé à une deuxième solution où la récupération de la liste des personnes se fait personne par personne au lieu de la récupérer entièrement au lancement de l'application. L'avantage d'une telle solution est qu'elle est économique en mémoire, en revanche, elle nécessite un requêtage fréquent de la base de données qui pourrait saturer le réseau si la base de données est distante.