Projet Informatique et Sciences du Numériques

LE DALTONISME

Julien Constant| ISN | Année 2018-2019

Sommaire

Pourquoi ce projet ? 2
Les différentes étapes du projet ? 2
    Les différentes parties du projet : 2
    Répartition des tâches : 2
Qu'est-ce que le daltonisme ? 3
Mon rôle dans le projet : 3
Bilan : 7
Annexes : 8
    Code de la fonction protanopie : 8
    Code de la fonction sauvegarder : 8
    Code de la création des menus déroulants avec appel des fonctions : 8

Pourquoi ce projet ?

Après de multiples réflexions, nous avons décidé de mener ce projet autour du daltonisme et ce afin d'aider à la compréhension de leurs perceptions du monde. Par la suite nous avons décidé d'élargir notre projet en nous appuyant sur la vue des mammifères qui nous entourent. Afin de répondre au mieux à cette thématique nous avons décidé d'utiliser le langage Python que nous maîtrisons.

Ce projet se constitue de moi-même, Julien Constant élève de TS1 et de Mylène Pozar élève de Ts2, étant des amis proches dans la vie courante, c'est de façon naturelle qu'elle et moi avons formé ce groupe.

Ce projet a pris forme durant les séances en cours mais tout particulièrement lors de notre temps libre.

Les différentes étapes du projet ?

Les différentes parties du projet :

- L'ouverture d'une image et son positionnement.
- Le traitement de l'image selon le choix de l'utilisateur.
- La sauvegarde de l'image à l'emplacement choisi par l'utilisateur.

Répartition des tâches :

Après avoir listé les différentes étapes à la réalisation de ce projet, la répartition des tâches s'est faite naturellement :

- Julien Constant : L'ouverture d'une image, le traitement de l'image selon le choix de l'utilisateur (partie daltonisme) et la sauvegarde de l'image après transformation.
- Mylène Pozar : Affichage du texte après traitement, le traitement de l'image (partie Mammifères) et le positionnement de l'image.

Il est important de préciser et de prendre en compte que ce projet et avant tout un travail de groupe, certaines tâches se sont donc faites-en commun.

Qu'est-ce que le daltonisme ?

Le daltonisme (ou dyschromatopsie) est une anomalie de la vision affectant la perception des couleurs. D'origine généralement génétique elle a alors pour cause une déficience d'un ou plusieurs des trois types de cônes de la rétine oculaire.
Selon Wikipédia.

De part cette anomalie génétique, on peut sortir trois sortes de daltonisme :

- Monochromate ou Achromate :

  Absence totale de la perception des couleurs, la personne touchée ne perçoit le monde qu'en nuances de gris, les cônes de la cornée sont dépourvus des trois pigments permettant la perception des couleurs. Elle ne touche qu'une personne sur quarante-mille.

- Dichromate :

  L'absence d'un gène, autrement dit d'un des trois pigments de l'œil, la personne touchée ne perçoit que deux couleurs primaires, il en existe trois sortes :

  - La protanopie : vert et bleu
  - La deutéranopie : rouge et bleu
  - La tritanopie : vert et rouge

- Trichromate :

  Dysfonctionnement d'un des trois cônes, la personne touchée perçoit les trois couleurs dont l'une d'intensité anormale, il en existe aussi trois sortes :

  - La protanomalie : lacune du rouge
  - La deutéranomalie : lacune du vert
  - La tritanomalie : lacune du bleu

Mon rôle dans le projet :

Mon rôle dans le projet a été de réaliser l'interface du programme, l'ouverture et la sauvegarde de l'image ainsi que son traitement (vue d'un daltonien uniquement).

J'ai donc commencé par réaliser une interface à l'aide du module Tkinter, en créant des menus déroulants pour le choix des options, mais aussi en créant deux « canevas », celui de gauche étant réservé pour la présentation de l'image d'origine, que l'utilisateur à décidé d'importer. Celui de droite quant à lui est réservé à l'image finale, c'est-à-dire l'image obtenue après traitement selon les choix de l'utilisateur.

De prime abord, il a fallu importer l'image à traiter, le problème étant qu'il existe plusieurs extensions d'images et surtout plusieurs formats d'images, notamment les extensions très connues tel que PNG, JPEG ou GIF qui n'étaient pas toutes supportées par Tkinter (seulement le format PNG à vrai dire), il a donc fallu reconnaître l'extension

et créer une image secondaire avec l'extension PNG avec le module OS, si cela était nécessaire. De plus, le format de l'image fut un autre problème, en effet les différents formats tels que le RGBA (images avec de la transparence) ou le P (images avec une palette unique de 256 couleurs) allait enrayer notre programme, là aussi il fallut convertir les images, mais cette fois-ci à l'aide du module Pillow (alias PIL) et de la recréer avec un format RGB.

Par la suite, il m'est devenu indispensable d'utiliser le module Pillow pour le traitement d'image. Ce module simplifie grandement leurs traitements et rend ainsi le code beaucoup moins lourd et beaucoup plus lisible. Cependant, il existe des fonctions du module Pillow que mes professeurs m'ont interdit d'utiliser, rendant le traitement d'images beaucoup trop simpliste, je m'y suis donc tenu.

Ainsi pour ne pas utiliser les fonctions, il a fallu se documenter pour connaître la façon dont l'image est traitée, en effet les images telles que nous les connaissons sont constituées pour la plupart d'un mélange de trois couleurs primaires (Red Green Blue ou Rouge Vert Bleu en français). Or lorsqu'on est daltonien, l'image que nous percevons et dépourvue d'un pigment (pour le cas le plus général) cependant au niveau informatique, le traitement de l'image ne se fait pas de la même manière. En effet si l'on retire une des trois compositions de la couleur à l'image on obtient un résultat qui n'est pas la réalité :

Vision normale :

L'image ci-dessus n'est pas modifiée, la matrice d'un de ces pixels se compose donc ainsi :

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Vision non réaliste de la protanopie :

Ici on a simplement supprimé le rouge de l'image, la matrice d'un pixel de l'image se compose donc ainsi :

$$\begin{bmatrix} 0 \\ G \\ B \end{bmatrix}$$

Vision réelle de la protanopie :

Or on sait que la protanopie ne rend pas la vision de la personne totalement verte, il a donc fallu pousser les recherches plus loin et se rendre compte que la protanopie n'était pas la suppression pure et simple de la couleur rouge, mais le remplacement de celle-ci (au niveau informatique) par un mélange des deux autres selon des coefficients bien précis.

Qui plus est, ce remplacement s'effectue sur tout type de daltonisme, comme le montre les recherches effectuées sur ce site (en anglais) :

https://ixora.io/projects/colorblindness/color-blindness-simulation-research/

Ainsi nous obtenons les coefficients suivants pour les cas des dichromates :

- Protanopie :

$$\begin{bmatrix} 1.05 \times G + -0.05 \times B \\ G \\ B \end{bmatrix}$$

- Deutéranopie :

$$\begin{bmatrix} R \\ 0.95 \times R + 0.05 \times B \\ B \end{bmatrix}$$

- Tritanopie :

$$\begin{bmatrix} R \\ G \\ -0.6 \times R + 2.3 \times G \end{bmatrix}$$

À noter que les coefficients ont étés arrondis dans la mesure ou les changements apportés par une plus grande précision de ceux-ci était minime.

Pour les cas des monochromates (ou achromates), qui ne perçoivent aucune couleur, il faut remplacer les trois couleurs par un mélange des trois selon cette formule :

$$\begin{bmatrix} R \times 0{,}299 + G \times 0{,}587 + B \times 0{,}114 \\ R \times 0{,}299 + G \times 0{,}587 + B \times 0{,}114 \\ R \times 0{,}299 + G \times 0{,}587 + B \times 0{,}114 \end{bmatrix}$$

Pour finir, les cas de tritanopie étant la déficience d'un des trois cônes de la rétine, il faut proposer à l'utilisateur de choisir un pourcentage de déficience. La suite du traitement de l'image est un bête produit en croix entre l'image originale et sa version deutéranopie :

- Exemple avec la protanomalie :

$$\begin{bmatrix} (R \times per) + ((1.05 \times G + -0.05 \times B) \times (1 - per)) \\ G \\ B \end{bmatrix}$$

Ou per est le pourcentage au préalable choisi par l'utilisateur.

Pour finir, la sauvegarde de l'image s'est avérée assez simple, l'utilisateur choisit l'emplacement et le logiciel enregistre l'image à l'aide du module OS.

Bilan :

J'ai particulièrement apprécié réaliser ce projet durant l'année. J'ai notamment acquis de nouvelles compétences en programmation sous Python et j'ai pu apporter mes compétences au sein du groupe pour aider et faire avancer le projet.

Je pense que le projet a atteint un stade ou il est plus nécessaire d'ajouter des fonctionnalités, mais il serait plus intéressant d'en faire une application exécutable en .exe permettant ainsi de se dégager de la contrainte de devoir installer Python et ces modules pour pouvoir ouvrir le programme. Mais pour cela il faudrait réécrire totalement le code, ce qui serait très long et fastidieux.

Annexes :

Code de la fonction protanopie :

```
def Protanopie():  # Otto -> Protanopie : perception du vert et du bleu seulement
    global info_texte, pic_d_check, pic_d_pil, pic_d_tk, pic_f_tk, pic_f_pil, result_canvas, pic_f_check, pic_f_canvas
    print("Action : Protanopie")

    if pic_d_check is True:

        texte_aff("Vue d'un protanope:")


        width = pic_d_tk.width()
        height = pic_d_tk.height()

        pic_f_pil = Image.new('RGB', pic_d_pil.size)

        print("taille:", width, height)

        # Conversion de l'image avec PIL : A tester
        for y in range(height):
            for x in range(width):
                pixel_d = (x, y)
                R, G, B = pic_d_pil.getpixel(pixel_d)
                L = int(1.05 * G + -0.05 * B)
                pic_f_pil.putpixel((pixel_d), (L, G, B))
        pic_f_pil.save("temp_protanopie.png")

        # Image avec tkinter
        pic_f_tk = PhotoImage(file="temp_protanopie.png")
        real_width = pic_f_tk.width() + 20

        # positionnement sur le canvas de l'image ouverte (largeur, hauteur)
        pic_f_canvas = result_canvas.create_image(real_width // 2, 360, image=pic_f_tk)

        # supression du fichier temporaire
        os.remove("temp_protanopie.png")

        pic_f_check = True
    else:
        showinfo("Message à caractère informatif", "Aucun fichier à modifier")
```

Code de la fonction sauvegarder :

```
def save_as():
    global pic_f_pil, pic_f_check
    print("Action : save_as")

    # on regarde d'abord s'il existe un fichier à sauvegarder
    if pic_f_check is True:
        print("Il existe un fichier à sauvegarder")
        filename = filedialog.asksaveasfilename(title="Enregistrement de l'image", initialdir="fic", filetypes=(("fichier PNG", "*.png"), ("tous les fichiers", "*.*")))
        print("nom du fichier", filename)

        file_path, file_ext = os.path.splitext(filename)
        pic_f_pil.save(file_path + ".png")
    else:
        showerror("Message à caractère informatif", "Aucun fichier à sauvegarder")
```

Code de la création des menus déroulants avec appel des fonctions :

```
# /-/ pannel deroulant :

# / creation de la barre des menus :
menu_list = Menu(main_window)


# / creation du menu fichier
menu_file = Menu(menu_list, tearoff=0)
for lab, com in[('Ouvrir un fichier', open_file), ('Sauvegarder sous', save_as)]:
    menu_file.add_command(label=lab, command=com)

# / creation du menu autres vues
menu_vue = Menu(menu_list, tearoff=0)
for lab, com in[('Chat', cat_view), ('Cheval', horse_view), ('Chien', dog_view), ('À propos', autre_info)]:
    menu_vue.add_command(label=lab, command=com)

# / creation du menu position
menu_pos = Menu(menu_list, tearoff=0)
for lab, com in[('Image importée', pos_pic_d), ('Image obtenue', pos_pic_f)]:
    menu_pos.add_command(label=lab, command=com)

# / creation du menu monochromate
menu_monochromate = Menu(menu_list, tearoff=0)
for lab, com in[('Monochromatie', Monochromate), ('À propos', mono_info)]:
    menu_monochromate.add_command(label=lab, command=com)

# / creation du menu dichromate
menu_dichromate = Menu(menu_list, tearoff=0)
for lab, com in[('Protanopie', Protanopie), ('Deuteranopie', Deuteranopie), ('Tritanopie', Tritanopie), ('À propos', dicho_info)]:
    menu_dichromate.add_command(label=lab, command=com)

# / creation du menu trichromate
menu_trichromate = Menu(menu_list, tearoff=0)
for lab, com in[('Protanomalie', pro), ('Deuteranomalie', deu), ('Tritanomalie', tri), ('À propos', tricho_info)]:
    menu_trichromate.add_command(label=lab, command=com)

# / ajout des menu a la barre de menu
for lab, men in[('Fichier', menu_file), ('Position', menu_pos), ('Monochromate', menu_monochromate), ('Dichromate', menu_dichromate), ('Trichromate', menu_trichromate), ('Autres Visions', menu_vue)]:
    menu_list.add_cascade(label=lab, menu=men)

main_window.config(menu=menu_list)

# /-/ Met la fenêtre en attente

main_window.mainloop()
```