

PROJET INFORMATIQUE ET SCIENCES DU NUMÉRIQUE

Le daltonisme
Julien Constant





QU'EST-CE QUE LE DALTONISME ?

*Les différents spectres
qu'ils perçoivent*

MONOCHROMATE (ACHROMATE) :

Spectres :

Vision normale :



Vision du daltonien :



DICHROMATE :

Spectres :

Vision normale :



Vision du daltonien :



Protanopie



Deutéranopie



Tritanopie

TRICHROMATIE :

Spectres :

Vision normale :



Vision du daltonien :



Protanomalie à 50%



Deutéranomalie à 50%



Tritanomalie à 50%



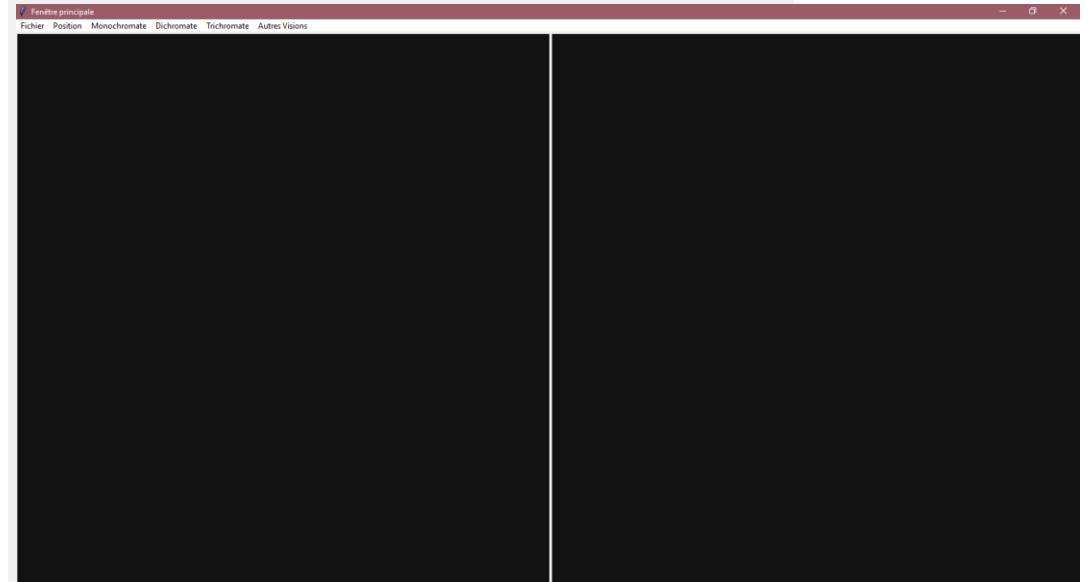
MON RÔLE DANS LE PROJET

Le code de A à Z

INTERFACE DU PROGRAMME :

Doit répondre à certaines attentes :

- Simple de compréhension
- Menus explicites
- Division de l'interface en 2 parties :
 - 1 importée
 - 1 résultat



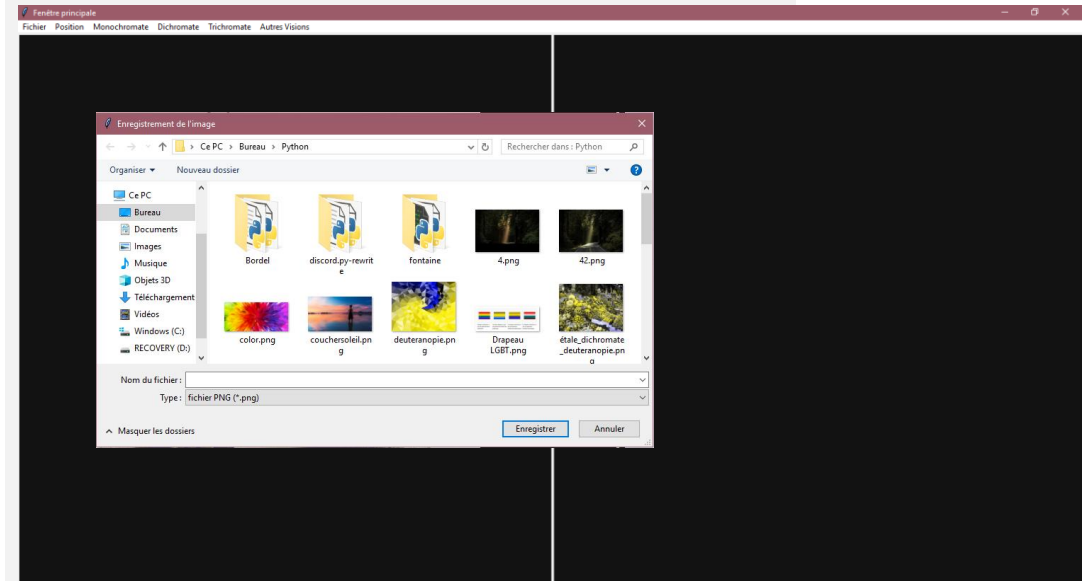
Code Python de l'interface :

```
914 # -*-/ fenetre principale :
915
916
917 main_window = Tk()
918 main_window.title("Fenêtre principale")
919 #main_window.resizable(width=False, height=False)
920
921 default_canvas = Canvas(main_window, relief=GROOVE, height=900, width=800, bg='gray7')
922 default_canvas.grid(row=0, column=0)
923 result_canvas = Canvas(main_window, relief=GROOVE, height=900, width=800, bg='gray7')
924 result_canvas.grid(row=0, column=1)
925 info_texte = result_canvas.create_text(100, 25 , text='', fill="white")
926
927
928 # /-/ pannel deroulant :
929
930 # / creation de la barre des menus :
931 menu_list = Menu(main_window)
932
933
934 # / creation du menu fichier
935 menu_file = Menu(menu_list, tearoff=0)
936 for lab, com in [('Ouvrir un fichier', open_file), ('Sauvegarder sous', save_as)]:
937     menu_file.add_command(label=lab, command=com)
938
939 # / creation du menu autres vues
940 menu_vue = Menu(menu_list, tearoff=0)
941 for lab, com in [('Chat', cat_view), ('Cheval', horse_view), ('Chien', dog_view), ('À propos', autre_info)]:
942     menu_vue.add_command(label=lab, command=com)
943
944 # / creation du menu position
945 menu_pos = Menu(menu_list, tearoff=0)
946 for lab, com in [('Image importée', pos_pic_d), ('Image obtenue', pos_pic_f)]:
947     menu_pos.add_command(label=lab, command=com)
948
949 # / creation du menu monochrome
950 menu_monochrome = Menu(menu_list, tearoff=0)
951 for lab, com in [('Monochromatie', Monochrome), ('À propos', mono_info)]:
952     menu_monochrome.add_command(label=lab, command=com)
953
954 # / creation du menu dichromate
955 menu_dichromate = Menu(menu_list, tearoff=0)
956 for lab, com in [('Protanopie', Protanopie), ('Deutanopie', Deutanopie), ('Tritanopie', Tritanopie), ('À propos', dicho_info)]:
957     menu_dichromate.add_command(label=lab, command=com)
958
959 # / creation du menu trichromate
960 menu_trichromate = Menu(menu_list, tearoff=0)
961 for lab, com in [('Protanomalie', pro), ('Deutanomalie', deu), ('Tritanomalie', tri), ('À propos', tricho_info)]:
962     menu_trichromate.add_command(label=lab, command=com)
963
964 # / ajout des menu a la barre de menu
965 for lab, men in [('Fichier', menu_file), ('Position', menu_pos), ('Monochrome', menu_monochrome), ('Dichromate', menu_dichromate), ('Trichromate', menu_trichromate), ('Autres Visions', menu_vue)]:
966     menu_list.add_cascade(label=lab, menu=men)
967
968 main_window.config(menu=menu_list)
969
970 # /-/ Met la fenêtre en attente
971
972 main_window.mainloop()
```


OUVERTURE ET SAUVEGARDE DE L'IMAGE :

Doit répondre à certaines attentes :

- Ouvrir tout type de fichier
- Sauvegarder l'image obtenue après le traitement



Code Python d'ouverture :

```
74 def open_file(): # pic_d... = pic_default
75     global pic_d_pil, pic_d_tk, default_canvas, pic_d_check, pic_d_canvas
76     print("Action : open_file")
77
78     pic_d_path = filedialog.askopenfilename(title="Ouverture d'une image", initialdir="fic", multiple=False, filetypes=(("fichier PNG", "*.png"), ("fichier JPEG", "*.jpg"), ("fichier GIF", ".gif"), ("tout les fichiers", "*.*")))
79     print(pic_d_path)
80
81     # Conversion en PNG (si JPEG)
82     path, ext = os.path.splitext(pic_d_path)
83     if ext == ".jpg" or ".gif": # si l'extension est jpg ou gif
84
85     elif ext == ".png": # si l'extension est un png
86         print("pas de conversion en png")
87         pic_d_pil = Image.open(pic_d_path)
88     else:
89         showinfo("Message à caractère informatif", "Le fichier n'est pas supporté (extension)")
90
91     # Test RGBA ou P ou RGB :
92     pal = pic_d_pil.getbands()
93     print(pal)
94
95     if pal == ('R', 'G', 'B', 'A'):
96
97     elif pal == ('P',): # palette de 256 couleurs uniques
98
99     elif pal == ('R', 'G', 'B'):
100         print("RGB")
101
102         # Image avec tkinter
103         pic_d_tk = PhotoImage(file=pic_d_path)
104         real_width = pic_d_tk.width() + 20
105
106         # positionnement sur le canvas de l'image ouverte (largeur, hauteur)
107         pic_d_canvas = default_canvas.create_image(real_width // 2, 360, image=pic_d_tk)
108         pic_d_check = True
109
110         showinfo("Message à caractère informatif", "Le fichier a été correctement importé\n( RGB)")
111
112     else:
113         showerror("Message à caractère informatif", "Le fichier n'est pas supporté (propriétés)")
```

Nb : certaines parties sont réduites

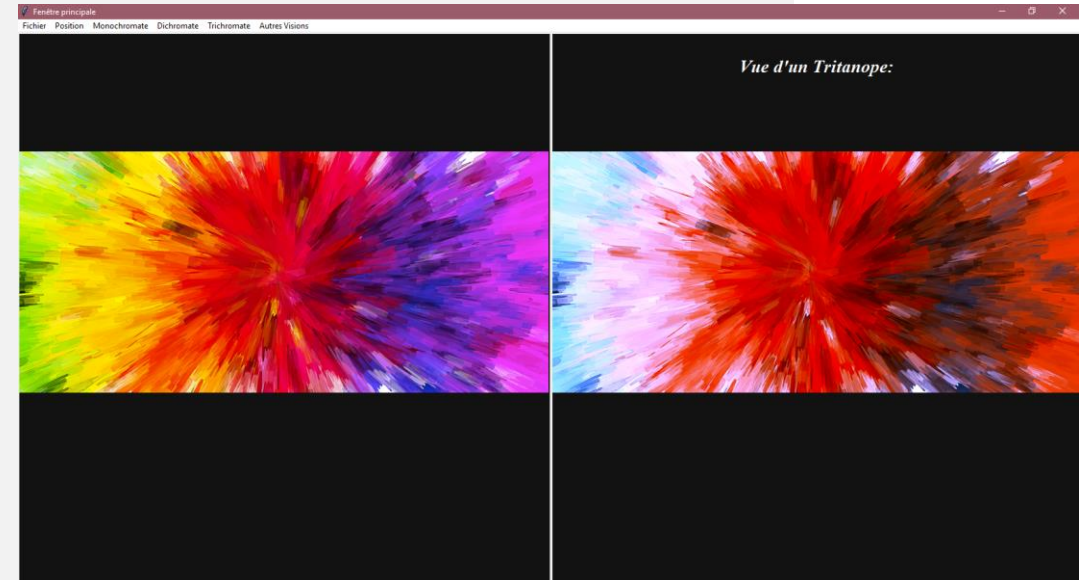
Code Python de sauvegarde :

```
165 def save_as():
166     global pic_f_pil, pic_f_check
167     print("Action : save_as")
168
169     # on regarde d'abord s'il existe un fichier à sauvegarder
170     if pic_f_check is True:
171         print("Il existe un fichier à sauvegarder")
172         filename = filedialog.asksaveasfilename(title="Enregistrement de l'image", initialdir="fic", filetypes=(("fichier PNG", "*.png"), ("tous les fichiers", "*.*")))
173         print("nom du fichier", filename)
174
175         file_path, file_ext = os.path.splitext(filename)
176         pic_f_pil.save(file_path + ".png")
177     else:
178         showerror("Message à caractère informatif", "Aucun fichier à sauvegarder")
```


TRAITEMENT DE L'IMAGE :

Doit répondre à certaines attentes :

- Retranscrire la vision du daltonien le plus fidèlement possible.
- L'image ne doit pas perdre en qualité.



TRAITEMENT DE L'IMAGE :

Problème rencontré : exemple sur la protanopie

Composition d'une image normale :

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

Composition d'une image idéaliste :

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

Composition d'une image réaliste :

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



Code Python du traitement de l'image (Monochromate) :

```
373 def Protanopie(): # Otto -> Protanopie : perception du vert et du bleu seulement
374     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
375     print("Action : Protanopie")
376
377     if pic_d_check is True:
378
379         texte_aff("Vue d'un protanope:")
380
381
382         width = pic_d_tk.width()
383         height = pic_d_tk.height()
384
385         pic_f_pil = Image.new('RGB', pic_d_pil.size)
386
387         print("taille:", width, height)
388
389         # Conversion de l'image avec PIL : A tester
390         for y in range(height):
391             for x in range(width):
392                 pixel_d = (x, y)
393                 R, G, B = pic_d_pil.getpixel(pixel_d)
394                 L = int(1.05 * G + -0.05 * B)
395                 pic_f_pil.putpixel((pixel_d), (L, G, B))
396         pic_f_pil.save("temp_protanopie.png")
397
398         # Image avec tkinter
399         pic_f_tk = PhotoImage(file="temp_protanopie.png")
400         real_width = pic_f_tk.width() + 20
401
402         # positionnement sur le canvas de l'image ouverte (largeur, hauteur)
403         pic_f_canvas = result_canvas.create_image(real_width // 2, 360, image=pic_f_tk)
404
405         # suppression du fichier temporaire
406         os.remove("temp_protanopie.png")
407
408         pic_f_check = True
409     else:
410         showinfo("Message à caractère informatif", "Aucun fichier à modifier")
```




BILAN

*Quae sunt Caesaris,
Caesari*

MERCI DE VOTRE ATTENTION

Julien Constant 📞 +33 7 86 35 13 39

✉ *julienconstant190@gmail.com*

