

Projet individuel NSI

Projet Photoposh



Membres du groupe :

- Maxime GUG
- Mattéo Metz
- Mattéo Houbre

Lycée Polyvalent Jean-Baptiste Schwilgué, 67600 Sélestat

Année 2021-2022

Sommaire :

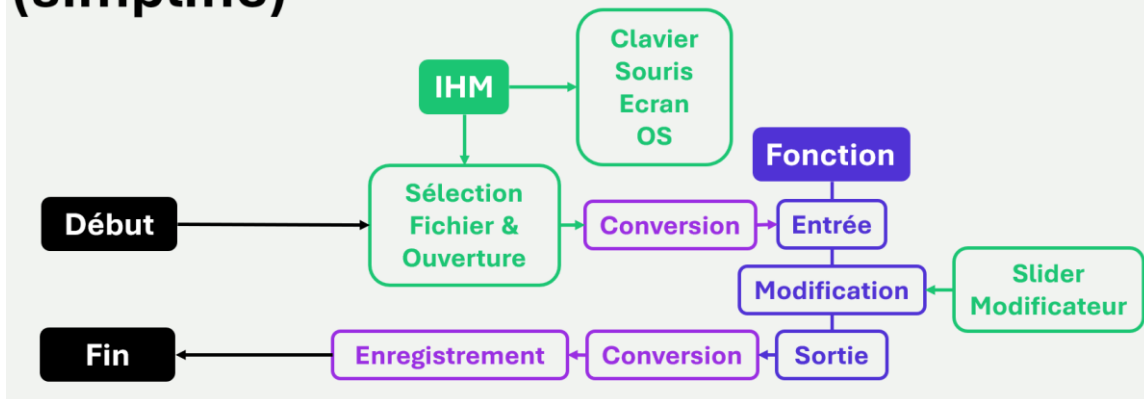
- 1. Présentation :**
 - Problématique
 - Cas d'utilisation synthétique
 - Cahier des charges
- 2. Fonctions :**
 - Interface utilisateur
 - Gestion Image
 - Rendu d'image
 - Luminosité
 - Contraste
 - Inversion
 - Nuances de gris
- 3. Problèmes survenus dans l'équipe**

1. Présentation :

- **Problématique :**

Logiciel d'édition d'image interactif avec une liste de fonctions prédéfinie comme par exemple, une gestion de la saturation, ou bien encore de la luminosité de l'image.

Cas d'utilisation (simplifié)



- **Cahier des charges :**

- Le programme se doit d'être un fichier exécutable (Windows au minimum) permettant au programme de tourner une fois compilé.
- L'interface se doit d'être adaptative à l'écran l'affichant
- L'interface se doit d'être adaptative au contenu affiché
- L'interface se doit d'être simple et fiable
- Les fonctions se doivent d'être fiable et commentées

2. Fonctions :

Interface utilisateur :

Options disponibles :

- Kivy
- Tkinter
- Pygame
- Django

Solutions retenues : Kivy & Tkinter

Kivy propose un large éventail de possibilités, ainsi que de ressources. De plus, sa syntaxe est plus aisée que les autres. Tkinter est le seul module à proposer des liens avec l'explorateur de fichier par défaut de l'OS de la manière dont on l'utilise. C'est donc pour ces raisons que ces modules ont été choisis.

Gestion Image :

Options disponibles :

- Pil
- ImageMagick
- Wand

Solution retenue : Pil

Pil permet une gestion simple, autant sur la syntaxe que sur la complexité algorithmique des fonctions utilisées. De plus, il existe beaucoup de ressources sur ce module.

Rendu d'image :

Options disponibles :

- Kivy
- Pil
- OpenCV
- Pygames

Solution retenue : Kivy

Kivy permet d'avoir un affichage intégré dans la même fenêtre et donc éviter d'avoir deux applications.

Luminosité :

Algorithme Choisi :

Procedure Truncate(value)

If value < 0 then value = 0

If value > 255 then value = 255

Return value

EndProcedure

L'algorithme suivant propose une grande simplicité et fiabilité. C'est donc pour cette raison qu'il a été choisi.

Fonction :

```
def Brightness(i, v):
    for y in range(i.size[1]):
        for x in range(i.size[0]):
            P = list(i.getpixel((x, y)))
            for w in range(3):
                P[w] += int(v)
                if P[w] < 0:
                    P[w] = 0
                elif P[w] > 255:
                    P[w] = 255
            i.putpixel((x, y), tuple(P))
```

La fonction prends en entrée un objet Image du module PIL, ainsi qu'une valeur Float comprise entre -128 et 128 donné par le slider de Kivy. La fonction modifie l'objet directement, pixel par pixel, couleur par couleur.

Problèmes :

les valeurs doivent être comprises entre -128 et 128 et non -255 et 255 afin d'éviter d'avoir une image exclusivement noire ou blanche.

Contraste :

Algorithme Choisi :

factor = (259 * (contrast + 255)) / (255 * (259 - contrast))

colour = GetPixelColour(x, y)

newRed = Truncate(factor * (Red(colour) - 128) + 128)

newGreen = Truncate(factor * (Green(colour) - 128) + 128)

newBlue = Truncate(factor * (Blue(colour) - 128) + 128)

PutPixelColour(x, y) = RGB(newRed, newGreen, newBlue)

Fonction :

```
def Contrast(i, v):
    F = (259*(255+v))/(255*(259-v))
    for y in range(i.size[1]):
        for x in range(i.size[0]):
            P = list(i.getpixel((x, y)))
            for w in range(3):
                P[w] = int((F * (P[w] - 128) + 128))
            i.putpixel((x, y), tuple(P))
```

La fonction prends en entrée un objet Image du module PIL, ainsi qu'une valeur Float comprise entre -128 et 128 donné par le slider de Kivy. La fonction modifie l'objet directement, pixel par pixel, couleur par couleur.

Inversion :

Algorithme Choisi :

```
newR = a - r
newG = a - g
newB = a - b
```

Fonction :

```
def Inversion(i):
    for y in range(i.size[1]):
        for x in range(i.size[0]):
            P = list(i.getpixel((x, y)))
            for w in range(3):
                P[w] = int(P[3] - P[w])
            i.putpixel((x, y), tuple(P))
```

La fonction prends en entrée un objet Image du module PIL, La fonction modifie l'objet directement, pixel par pixel, couleur par couleur.

Problèmes :

La valeur doit prendre l'alpha et non 255

Nuances de gris :

Algorithme Choisi :

The Weighted Method

The weighted method, also called the luminosity method, weighs red, green, and blue according to their wavelengths. The improved formula is as follows:

$$\text{Grayscale} = 0.299R + 0.587G + 0.114B$$

Fonction :

```
def Grayscale(i):
    for y in range(i.size[1]):
        for x in range(i.size[0]):
            P = list(i.getpixel((x, y)))
            P[0] = int(P[0] * 0.299 + P[1] * 0.587 + P[2] * 0.114)
            P[1] = int(P[0] * 0.299 + P[1] * 0.587 + P[2] * 0.114)
            P[2] = int(P[0] * 0.299 + P[1] * 0.587 + P[2] * 0.114)
            i.putpixel((x, y), tuple(P))
```

La fonction prends en entrée un objet Image du module PIL, La fonction modifie l'objet directement, pixel par pixel, couleur par couleur.

Problèmes :

Il est arrivé que du bleu persiste sur l'image.

3. Problèmes survenus dans l'équipe :

Malheureusement, étant donné le désintérêt de mes compères pour le projet, je me suis retrouvé seul pour sa réalisation. Il n'atteint donc pas l'intégralité des objectifs fixés. Le projet a pu être mené à bien tant bien que mal mais la répartition des tâches finale ressemble à celle présentée ci-dessous.

Répartition des tâches prévue :

- **Maxime G.**

- Sélection de fichiers
- Lecture des informations du fichier
- Rendu d'image (Dans l'interface utilisateur)
- Rogner
- Détection de contours
- Mise à jour de taille

- **Mattéo H.**

- Sélection de format de sortie
- Luminosité
- Noir et blanc
- Nuances de gris
- Inversion

- **Mattéo M.**

- Outils d'interaction avec les fonctions (boutons, sliders)
- Contraste
- Saturation
- Flou & Netteté

Répartition des tâches finales :

- **Maxime G.**

- Sélection de fichiers
- Lecture des informations du fichier
- Rendu d'image (Dans l'interface utilisateur)
- Outils d'interaction avec les fonctions (boutons, sliders)
- Contraste
- Nuances de gris
- Inversion
- Luminosité

- **Mattéo H.**

- **Mattéo M.**