

Séquence 3: Traitement d'images avec Python

Source : diverses

Thème : La photographie numérique

Environnement envisagé : en classe avec PC équipé Python et module PIL (souvent fourni avec GUI)

Prérequis :

- notion de couleur en niveau de gris et RVB
- notion de fonction, boucle pour, instruction if en python

Durée :

- séance 2h00,

Objectifs :

- manipulation informatique d'image numérique
- travail sur les différents niveaux de couleurs (noir et blanc, niveaux de gris, RVB)
- travail sur l'algorithmique (2 boucles pour imbriqués)

Le but n'est pas de fournir des séquences clé en main, mais une base à retravailler et à adapter.

Autres sources sur magistère :

- activité débranchée : Nicolas BUYLE-BODIN & Brice PORTIER, négatif d'une image, Lyon
- activité débranchée : recherche du contour d'une image, Julien Launay, Lyon

Pistes pour les suites possibles :

- Étudier d'autres manipulations d'images mettant en œuvre d'autres structures algorithmiques (voir annexe pour symétrie, stéganographie, floutage, réduction de bruit, photomaton, ...)

Séquence : Traitement d'images avec Python

1. Création d'une image

On donne ci-dessous un script en langage python3.

```
from PIL import Image # import du module

def CreerImageRouge(nom, largeur, hauteur):
    im = Image.new ('RGB' , (largeur, hauteur))
    for i in range(largeur):
        for j in range(hauteur):
            im.putpixel((i,j),(i,0,0))
    im.save(nom)
    return "image sauvegardée dans rep travail"
```

1. Recopier et enregistrer le script dans un fichier.
2. Saisir en console *CreerImageRouge('exemple1.bmp', 200,200)*.
3. Ouvrir le fichier image qui a été créé.
 - a) Comment expliquer la couleur rouge dominante ?
 - b) Comment expliquer la couleur quasiment noir du bord gauche ?
 - c) Comment expliquer la couleur quasiment rouge du bord droit ?
4.
 - a) En mode console, saisir *CreerImageRouge('exemple1.jpg',200,200)*
 - b) Visuellement, les images sont-elles identiques ?
 - c) Comparer la taille des fichiers obtenus et donner une explication.
5. Créer une fonction *CreerImageVerte()* qui crée une image RGB en dégradé de vert. Les arguments seront le nom du fichier, la largeur et la hauteur de l'image.
6. Ajouter au fichier la fonction suivante puis exécuter le fichier




```
def lirePixel(nom,i,j):
    im = Image.open(nom)
    return im.getpixel((i,j))
```

- a) En mode console, saisir *lirePixel('exemple1.bmp',80,80)* et noter le retour obtenu.
- b) En mode console, saisir *lirePixel('exemple1.jpg',80,80)* et noter le retour obtenu.
- c) Le retour de la question b) n'est pas conforme à ce que l'on attendait. Donner une explication.

2. Manipulation d'une image en niveau de gris

On dispose d'une image au format bmp en niveau de gris. On souhaite écrire un programme pour effectuer 2 manipulations:

- faire le négatif de l'image
- transformer l'image en noir et blanc

image originale	négatif	noir et blanc
		

7. Dans une image en niveau de gris, les pixels ont une valeur allant de 0 (noir) à 255 (blanc). Pour créer le négatif de l'image, il faut « inverser » la valeur des pixels :
- 0 devient 255
 - 255 devient 0
 - 10 devient 245 (car éloigné de 10 unités du «bord»)
 - plus généralement, la valeur x devient $255-x$

On donne ci-dessous le code d'une fonction permettant de créer le négatif d'une photo dont le nom est passé en paramètre.

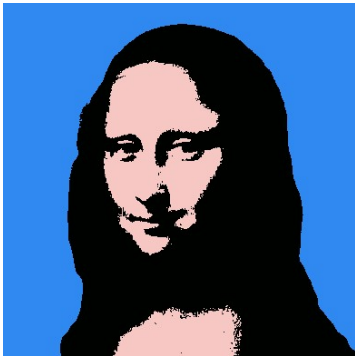

```
def negatif(nom):  
    im = Image.open(nom)  
    largeur,hauteur = im.size  
  
    im2 = im.copy()  
    for i in range(largeur):  
        for j in range(hauteur):  
            pixel = im.getpixel((i,j))  
            newPixel = 255-pixel  
            im2.putpixel((i,j),newPixel)  
    im2.save("negatif.bmp")  
    im2.show()  
    return "négatif créé"
```

Recopier et tester la fonction en mode console.

8. Pour créer une photo en noir et blanc à partir d'une photo en niveau de gris, il faut transformer les pixels « presque noir » en noir et ceux « presque blanc » en blanc. Ainsi :
- un pixel de valeur inférieure à 127 prendra pour valeur 0
 - un pixel de valeur supérieure à 127 prendra pour valeur 255
- Créer une fonction *noirEtBlanc(nom)* qui crée la version noir et blanc de la photo dont le nom est passé en paramètre.

3. Manipulation d'une image en RVB

L'image joconde.jpg est constituée uniquement de trois couleurs. Le but de cet exercice est de créer, à partir de cette image, une image à la façon d'Andy Warhol :

image originale	Wahrol
	

Pour effectuer cette modification, la composante (r, v, b) du pixel a été modifiée en (v, b, r).

9. Écrire une fonction `warhol(nom)` qui crée la version wahrol de la photo dont le nom est passé en paramètre.
On utilisera l'instruction `im2.save("warhol.jpeg")`
10. Écrire d'autres fonctions `warhol` en testant d'autres manipulations sur les pixels. (on pourra tester le « négatif » de tout ou partie des composantes RVB)