## Activités sur la photo numérique

Un pixel est défini par ses coordonnées. L'origine du repère qui permet de définir ces coordonnées se trouve en haut à gauche de l'image. L'axe des abscisses est orienté de gauche à droite et l'axe des ordonnées est orienté de haut en bas.



Une image est formée de pixels colorés, représentés par trois nombres RVB (Rouge, Vert, Bleu).

Des algorithmes sont utilisés dans les appareils de photographie numérique lors de la prise de vue et lors du développement de l'image. Une image étant créée, on peut la traiter à l'aide de programmes informatiques pour la transformer en agissant sur les trois composantes de ses pixels.

Dans la suite, on propose de faire des activités qui utilisent des morceaux de codes en Python.

On utilisera la bibiothèque PIL d'Edupython.

Un pixel est défini par ses coordonnées. L'origine du repère qui permet de définir ces coordonnées se trouve en haut à gauche de l'image. L'axe des abscisses est orienté de gauche à droite et l'axe des ordonnées est orienté de haut en bas.**Fonctions utiles** 

Voici des exemples d'instructions et de morceaux de codes qui permettent de travailler sur une image.

from PIL import Image	#On importe le module Image de la bibliothèque PIL
<pre>img1=Image.open("Mon_Image.jpg")</pre>	# Ouverture du fichier Mon_Image.jpg et création de l'objet img1
<pre>img1.getpixel((i, j))</pre>	# On récupère la couleur du pixel (i,j) de l'image imgl sous la forme d'un tuple (r, g, b)
<pre>img.putpixel((i, j), (r,v,b))</pre>	# On attribue la couleur (r,v,b) au pixel (i,j)
<pre>img.save("img2.jpg")</pre>	# on enregistre l'image crée sous le sous nom img2.jpg
<pre>img2.show()</pre>	#On montre l'image img2

IMPORTANT : avant de commencer, il faut créer un dossier « photo numerique » sur le bureau ou sur votre clé usb, dossier dans lequel vous devez mettre :

- L'image que vous souhaitez modifier
- Le module python

## Activité 1 : passage au négatif

Ouvrir l'image « joconde.jpg » (mathssa.fr/joconde) et enregistrer cette image dans le dossier « photo numerique »

Ouvrir un nouveau module d'Edupython et l'enregistrer dans le dossier « photo numérique » sous le nom « negatif.py »

Le programme ci-dessous permet de récupérer pour chaque pixel le triplet (r,v,b) puis on modifie ce triplet afin d'obtenir l'image en « négatif » (le négatif d'une image s'obtient en faisant soustrayant 255 à chacun des nombres r,v,b)

Recopier et compléter le programme ci-dessous : (inutile de recopier les commentaires) from PIL import Image #Importe le module Image de la bibliothèque PIL

```
image0 = Image.open("joconde.jpg") # Ouverture du fichier image source et création de l'objet Image
dimx = image0.size[0] # largeur de l'image
dimy = image0.size[1] # hauteur de l'image
image1 = Image.new('RGB', (dimx, dimy), (0, 0, 0)) # Création d'une nouvelle image ayant la meme dimension
for i in range(dimy):
    for j in range(dimy):
        r, v, b = image0.getpixel((i, j)) # On récupère la couleur du pixel (i,j) sous la forme d'un triplet (r, v, b)
        image1.save("négatif.jpg") # on sauvegarde sous un autre nom
        image1.show()
```

Si vous avez une image, disponible vous pouvez obtenir le négatif de votre image , en utilisant le programme précédent (remplacer « joconde.jpg » et « négatif.jpg » - ne pas oublier d'enregistrer votre image)

## Activité 2 : permutation des couleurs

Ouvrir l'image « joconde.jpg » ou l'image de votre choix (sauvegardée dans le dossier « photo numerique »

Ouvrir un nouveau module d'Edupython et l'enregistrer dans le dossier « photo numérique » sous le nom « permutation.py »

Le programme ci-dessous permet d'effectuer pour chaque pixel une permutation du triplet (r,v,b) .

Recopier et compléter le programme ci-dessous : (inutile de recopier les commentaires) from PIL import Image #Importe le module Image de la bibliothèque PIL

## Activité 3 : mosaïque

Dans l'activité proposée ci-dessous, on se propose de dessiner une mosaïque inspirée par le portrait de Marylin Monroe réalisé par Andy Warhol.

Pour cela, on utilisera une image en couleur de votre choix ou l'image dont le lien est mathssa.fr/femme.jpg et on copiera cette image dans un dossier « photo numerique » .

Le nouveau module python s'appellera mosaique.py.



On souhaite coder un programme qui reproduira l'image ci-dessous avec les traitements proposés sous l'image sur sa table de couleurs (R1,G1,B1).



Traitements

- Image n° 2 : les nouvelles couleurs (R2,G2,B2) des pixels sont permutées de telle manière que : R2=B1, G2=R1, B2=G1
- Image n° 3 : les nouvelles couleurs (R3,G3,B3) des pixels sont permutées de telle manière que : R3=G1, G3=B1, B3=R1

- Image n° 4 : les nouvelles couleurs (R4,G4,B4) des pixels sont inversées de telle manière que : R4=255-R1, G4=255-G1, B4=255-B1
- Image n° 5 : elle représente l'image en niveaux de gris en faisant la moyenne des composantes (R1,G1,B1) de l'image initiale
- Image n° 6 : les couleurs (R5,G5,B5) des pixels de l'image n° 5 sont inversées de telle manière que : R6=255-R5, G6=255-G5, B6=255-B5
- Image n° 7 : elle représente l'image initiale en supprimant la composante bleue.
- Image n° 8 : elle représente l'image initiale en supprimant la composante rouge.
- Image n° 9 : elle représente l'image initiale en supprimant la composante verte.
- Image n° 10 : elle représente l'image initiale en ne gardant que la composante rouge.
- Image n° 11 : elle représente l'image initiale en ne gardant que la composante verte.
- Image n° 12 : elle représente l'image initiale en ne gardant que la composante bleue.

Pour cela on complètera le code suivant :

```
from PIL import Image
                                         #Importe le module Image de la bibliothèque PIL
image0 = Image.open("Femme.jpg") # Ouverture du fichier image source et création de l'objet Image
dimx = image0.size[0] # largeur de l'image
dimy = image0.size[1] # hauteur de l'image
dimx = image0.size[0] # largeur de l'image
dimy = image0.size[1] # hauteur de l'image
image1 = Image.new('RGB', (3*dimx, 4*dimy), (0, 0, 0)) # Création d'une nouvelle image 3 fois plus large et 4 fois plus haute
for i in range(dimx):
                                   # parcours des deux dimensions pour modifier chaque pixel
      for j in range(dimy):
           , v, b = image0.getpixel((i, j))  # On récupère la couleur du pixel (i,j) sous la forme d'un tuple (r, g, b)
image1.putpixel((i, j), (r,v,b))  # On recopie cette image dans le coin en haut à gauche
            image1.putpixel((dimx+i, j), (...,...)) #Etape 2 : permutation de (r,v,b)
           image1.putpixel((2*dimx+i, j), (...,..)) #Etape 3 : permutation de (r,v,b)
image1.putpixel((i, dimy+j), (...,..)) #Etape 4 : passage au négatif de l'image
#Etape 5 : elle représente l'image en niveaux de gris en faisant la moyenne des composantes (r,v,b) de l'image initiale
           g=int((r+v+b)/3)
            image1.putpixel((...,..), (.,.,.))
           image1.putpixel((...,..), (.,.,.)) #Etape 6 : passage au négatif de l'image de l'étape 5
           image1.putpixel((...,..), (.,.,.)) # Etape 7 : image initiale en supprimant la composante bleue.
image1.putpixel((...,..), (.,.,.)) # Etape 8 :image initiale en supprimant la composante rouge
            image1.putpixe1((...,..), (.,.,.)) # Etape 9 :image initiale en supprimant la composante ver
           image1.putpixel((...,..), (...,.)) # Etape 10 :image initiale en supprimant la composante bleue et verte.
image1.putpixel((dimx+i, 3*dimy+j), (...,)) #Etape 11 :image initiale en supprimant la composante rouge et bleue
image1.putpixel((2*dimx+i, 3*dimy+j), (...,)) #Etape 12 :image initiale en supprimant la composante rouge et verte
image1.save("mosaique.jpg") # on sauvegarde sous un autre nom
image1.show()
```