I. Préambule

Ce TP utilise la bibliothèque Pillow

Avant de commencer, je vous suggère d'aller piocher une image de test pas trop grosse (disons au maximum 1024 pixels dans chaque dimension). À défaut en voilà deux, obtenues à partir d'une photo mise à disposition par Hans Stieglitz sur les Wikimedia commons, et soumise à la licence CC-BY-SA 3.0 :

- rond-bac-SIN.jpg

II. Filtres d'image

Cette partie du TP concerne l'algorithmique de l'image. Plus précisément, on manipulera des images matricielles, c'est-à-dire représentées par des tableaux de pixels.

On utilise Pillow pour s'affranchir de la question des formats de fichiers.

II. 1. Ouverture et enregistrement de fichiers d'image avec Pillow

Le bout de code suivant convertit le fichier rond-bac-SIN.jpg (au format JPEG) en rond-bac-SIN.png (au format PNG) :

```
import PIL.Image as Image
im = Image.open(r' rond-bac-SIN.jpg')
im.save(r' rond-bac-SIN.png')
```

II. 2. Informations sur une image

Dans un interpréteur Python, essayez :

```
import PIL.Image as Image
im = Image.open(r'rond-bac-SIN.jpg')
```

Puis essayez d'utiliser la documentation interne de python (par exemple via la fonction dir de Python ou la commande help de l'interpréteur) pour explorer les informations fournies par l'objet image.

- Essayez par exemple d'obtenir sa taille.

II. 3. Représentation d'une image en mémoire

Si im est une image chargée avec PIL.Image.open, on accède à ses pixels via la fonction im.load() qui renvoie un tableau indexé par des couples d'entiers (et non pas une matrice au sens python du terme).

Par exemple,

```
pixels = im.load()
print(pixels[0,0])
```

Renvoie la valeur du pixel en haut à gauche de l'image.

- Affichez des pixels de l'image en couleurs, puis de l'image en noir et blanc. Que remarquez-vous ?

II. 4. Modifier une image

Pour modifier un pixel, on change sa valeur dans le tableau des pixels. Essayez :

```
pixels[0,0] = 0
im.save(r'rond-bac-SIN_mod.png')
```

- Est-ce que ça fonctionne avec l'image en noir et blanc ?
- Avec celle en couleurs ?
- Quel est l'effet produit ?

II. 5. Premiers filtres

Vous êtes parés pour écrire votre premier filtre : écrivez une fonction rev(im) qui remplace tous les pixels de l'image im par leur valeur en négatif. Traitez d'abord le cas en noir et blanc, puis celui en couleurs. Essayez de la modifier pour que ça fonctionne dans tous les cas.

II. 6. Couleurs

Autre exercice basique : écrivez les lignes qui prend en argument une image im en couleurs (mode "RGB") et renvoie le triplet d'images en nivaux de gris (mode "L") donnant les valeurs de chaque canal (rouge, vert, bleu) et qui transforme une image en couleurs vers une image en niveaux de gris.

- Est-ce convaincant ?
- Comparez avec l'image en niveaux de gris proposée plus haut : comment expliquez-vous la différence ?