1 Problématique

Artiste américain mort en 1987, Andy Warhol était un des principaux représentants du *pop art*. Le *dyptique de Marylin Monroe* réalisé en 1962 est une de ses œuvres célèbres. Il contient cinquante fois la même image de l'actrice mais avec un travail des couleurs différent.

FIGURE 1 – Dyptique de Marylin Monroe

Peut-on réaliser un programme qui reproduit le concept de cette œuvre ?

2 Manipuler une image

2.1 Présentation de la bibliothèque

La bibliothèque *Pillow* est un outil pour faciliter la manipulation des images.

Activité 1 : Expliquer le rôle de chaque ligne du code 1.

```
1 from PIL import Image
2
3 originale = Image.open("../ressources/joconde.jpg")
4 ligne, colonne = originale.size
```

Code 1 – La bibliothèque Pillow

2.2 Couleurs d'une image

Une image peut être vue comme un tableau de pixels de l lignes et c colonnes. Chacun de ces points de l'image code une couleur.
Pour obtenir une grande gamme de couleurs, le principe de la synthèse additive est utilisé. En combinant une quantité de Rouge, Vert, Bleu (RGB en anglais) il est possible de créer une palette importante.

Chaque pixel peut ainsi être vu comme un tuple (Rouge, Vert, Bleu), chaque composante étant codé sur *8 bits*.

Activité 2 :

1. Combien de valeurs peut prendre chaque composante ?
2. Calculer alors le nombre de couleurs qu'il est possible de coder.

Christophe Viroulaud

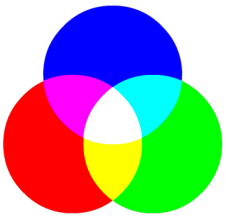

FIGURE 2 – Synthèse additive

2.3 Principe de la modification

Pour modifier l'image originale nous allons appliquer le protocole ci-après.
Pour chaque pixel :

— récupérer le tuple des trois composantes,

— créer un nouveau pixel en appliquant la transformation désirée,

— poser le nouveau pixel à la même position dans l'image.

Activité 3 :

1. À l'aide de la documentation de la bibliothèque (https://tinyurl.com/y3g95hj3) récupérer et afficher le tuple du pixel de coordonnées (10,10).
2. Créer un nouveau pixel qui ne conserve que la composante rouge de ce pixel.

3 Première œuvre d'art

Activité 4 :

1. En s'aidant de la documentation, *créer une copie* de l'image originale.
2. Écrire un programme qui charge l'image originale, la transforme en ne conservant que la composante rouge et affiche cette nouvelle version.

Le code 2 crée une nouvelle image vide deux fois plus grande et haute que l'originale.

```
1 img = Image.new('RGB', (ligne*2,colonne*2))
```

Code 2 – Une image vide

3. En s'aidant de la documentation, *coller* l'image rouge en haut à droite de *img*.
4. Créer trois variations de l'image originale et les placer aux trois autres positions de *img*.

4 Améliorer le programme

Si nous voulions créer une image avec cinquante variations de couleurs, il deviendrait très long. Pourtant on constate que de grandes portions de codes se ressemblent.

Christophe Viroulaud

4.1 Les fonctions mathématiques

En première approche, on peut voir une fonction mathématique comme une boîte réutilisable à qui on donne une valeur, qui effectue les opérations pour laquelle elle a été créée et qui nous renvoie le résultat.

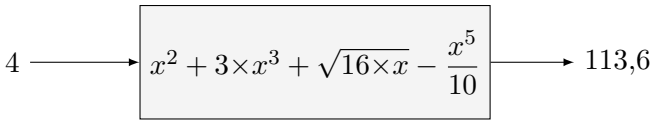

FIGURE 3 – La fonction $f(x)$ calcule et renvoie la valeur

4.2 Les fonctions en programmation

Le même principe est appliqué en programmation. En Python, la syntaxe est la suivante :

```
1 def ma_fonction(x):
2       f = x**2 + 3*x**3 + sqrt(16*x) - x**5/10
3       return f
```

Code 3 – Définir une fonction

Les mots clefs à retenir :

— **def** annonce qu'on définit une fonction qui a pour nom *ma_fonction* dans l'exemple.

— **return** précise qu'on sort de la fonction ici et qu'on *renvoie* la valeur de la variable f ici.

La fonction est crée, il faut maintenant l'utiliser dans le programme principal.

```
1 print(ma_fonction(4))
```

Dans le code 3 on remarque que la fonction possède un **paramètre** noté x. Dans le programme principal, on *passe* l'**argument** 4 à la fonction. Cette dernière remplacera chaque occurrence de x par 4.

Activité 5 :

1. Tester ma_fonction avec plusieurs valeurs de x.
2. Écrire une fonction **maxi(x, y)** qui demande deux paramètres x et y et qui renvoie la plus grande des deux valeurs.

4.3 Application à l'œuvre d'art

Les intérêts de créer une fonction sont multiples :

— créer des outils réutilisables

— rendre le code plus lisible

Le programme pour réaliser l'œuvre d'art présente des portions de code similaires.

Christophe Viroulaud

Activité 6 :

1. Créer une fonction **img_filtree(image, R, V, B)** qui prend une *image* comme argument ainsi que trois nombres, *R, V, B* compris entre 0 et 1. Ces valeurs représentent les proportions de Rouge, Vert et Bleu. La fonction renverra l'image filtrée.
2. Utiliser alors cette fonction et écrire un nouveau programme pour créer l'œuvre d'art.

Christophe Viroulaud