

Une pseudo-duplication d'un tableau de maître

Dans l'image ci-contre, il vous semble voir deux fois le même tableau. Il n'en est rien car l'image de droite et l'image de gauche n'ont en effet rien de commun. C'est ce que nous allons vous faire découvrir dans cet article.



Nous avons tous un tableau préféré, un tableau pour lequel nous ne sommes pas dérangés à le voir deux fois. Ce tableau préféré pour l'auteur de cet article est La Laitière de Vermeer où cet artiste du XVIIe siècle nous fait partager sa maîtrise du baroque avec cette prouesse de représenter le monde qui nous entoure en cherchant à ce que les objets n'apparaissent pas comme des objets figés mais qu'au contraire nous ayons l'impression d'une scène bien réelle avec des objets qui semblent se mouvoir.

La Laitière
Tableau de Johannes Vermeer
Source : site Wikipédia 

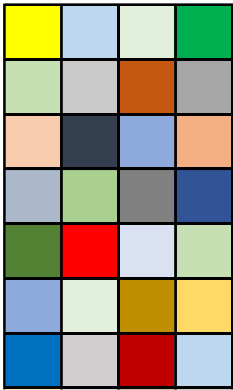
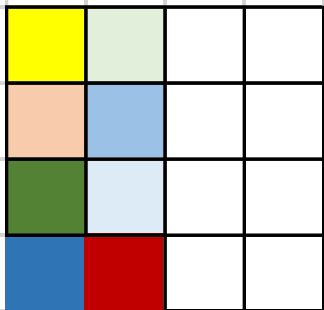
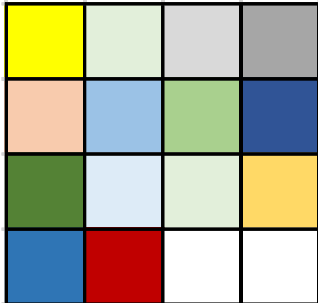
Notre intention étant ici de former au numérique, il va s'agir dans cet article de montrer comment le numérique peut procéder pour fournir une pseudo-duplication d'un tableau en construisant une image laissant penser que l'on est en présence de deux images strictement identiques alors qu'il n'en est rien : **les deux images (celle de droite et celle de gauche) n'ont en effet rien en commun**. C'est pour cela que nous parlons dans le titre de cet article d'une pseudo-duplication du tableau de Vermeer.

Présentation générale du processus de cette duplication

Cette duplication consiste à recopier les pixels de l'**image de départ** dans une nouvelle image de la façon suivante :

- On construit la **partie gauche** de la nouvelle image en recopiant uniquement les pixels placés sur des colonnes et lignes de rangs pairs ;
- On construit la **partie droite en recopiant** uniquement les pixels placés sur des colonnes et lignes de rangs impairs.

Nous pouvons expliciter en image ce processus :

Image de départ	Partie gauche	Partie droite
		

Présentation dans le détail du processus de cette duplication

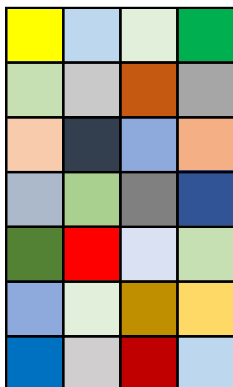
Pour construire cette pseudo-duplication, nous allons procéder en plusieurs étapes :

On construit une fonction que nous nommons double et qui a pour argument l'image au format JPEG que nous souhaitons dédoubler.

- Etape n°1 : on construit une image qui a le même nombre de colonnes que l'image de départ et un nombre de lignes sensiblement égal à la moitié du nombre de lignes de l'image de départ. Autrement dit, elle est de même largeur que l'image de départ et à peu près deux fois moins haute ;
- Etape n°2 : on construit tout d'abord la partie gauche de l'image qui est obtenue en ne prenant que les pixels situés sur les colonnes de rangs pairs et les lignes de rangs pairs ;
- Etape n°3 : on construit ensuite la partie droite de l'image qui est obtenue en ne prenant que les pixels situés sur les colonnes de rangs impairs et les lignes de rangs impairs ;

En fin de processus, on fait afficher l'image résultat qui contient les deux parties que nous venons de construire.

Présentation sur un exemple du processus



Afin de rendre plus explicite la construction de l'image de gauche puis celle de droite, nous allons effectuer cette construction sur l'image ci-contre. Cette image comporte 4 colonnes et 7 lignes que nous pouvons numéroté à partir de 0.

	0	1	2	3
0	Yellow	Light Blue	Light Green	Green
1	Light Green	Grey	Brown	Grey
2	Orange	Dark Blue	Blue	Orange
3	Light Blue	Light Green	Grey	Dark Blue
4	Dark Green	Red	Light Blue	Light Green
5	Light Blue	Light Green	Brown	Yellow
6	Blue	Grey	Red	Light Blue

1^{ère} étape : la première étape consiste à construire une image constituée du même nombre de colonnes. Pour obtenir le nombre de lignes :

- 1) On divise le nombre de lignes par 2 : on trouve ici 3,5 ;
- 2) On prend la partie entière de ce nombre de lignes : on trouve 3 ;
- 3) On ajoute 1 au résultat : on trouve 4.

	0	1	2	3
0				
1				
2				
3				

Au total, on a une image de 4 colonnes et 4 lignes.

2^{ème} étape : On construit l'image de gauche. Pour cela, on sélectionne uniquement les colonnes et lignes de rangs pairs de l'image de départ.

Puis l'on reporte pixel par pixel le code couleur sur l'image à construire.

	0	1	2	3
0	Yellow		Light Green	
1				
2	Orange		Blue	
3				
4	Dark Green		Light Blue	
5				
6	Blue		Red	

	0	1	2	3
0	Yellow	Light Green		
1	Orange	Blue		
2	Dark Green	Light Blue		
3	Blue	Red		

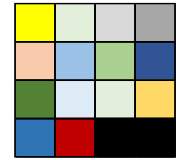
3^{ème} étape : On construit l'image de droite. Pour cela, on sélectionne uniquement les colonnes et lignes de rangs impairs de l'image de départ.

Puis l'on reporte pixel par pixel le code couleur sur l'image à construire.

	0	1	2	3
0				
1		Grey		Grey
2				
3		Light Green		Dark Blue
4				
5		Light Green		Yellow
6				

	0	1	2	3
0	Yellow	Light Green	Grey	Grey
1	Orange	Blue	Light Green	Dark Blue
2	Dark Green	Light Blue	Light Green	Yellow
3	Blue	Red		

On demande enfin au logiciel de faire apparaître l'image qui en l'occurrence sera celle de la figure traitée avec deux cases laissées noires car à sa construction l'image construite est constituée uniquement de cases noires.



Utilitaire pour la construction du programme

Le logiciel EduPython

Ce logiciel est téléchargeable à l'adresse suivante : <https://edupython.tuxfamily.org/>. Ce logiciel offre un environnement éducatif pour se former à l'utilisation du logiciel Python.



L'enregistrement de la photographie

Nous vous proposons de travailler pour l'instant à partir de l'image numérique de la photographie du tableau de La Laitière dont vous disposez sur la première page, il vous suffit :

- 1) De cliquer droit sur l'image numérique ci-contre du tableau que l'on souhaite dupliquer et de copier l'image ;
- 2) De coller ensuite cette image dans un éditeur de texte ;
- 3) Puis en cliquant droit à nouveau sur l'image de l'enregistrer sous le format jpeg dans le dossier dans lequel le logiciel EduPython est enregistré sur votre disque C:. On convient de donner à ce fichier le nom suivant : La Laitière Vermeer.



Remarque : Il sera utile par la suite de fournir au logiciel l'adresse de l'image à traiter, pour ce faire, il suffira de cliquer droit sur le fichier contenant l'image et de prendre l'adresse dudit fichier dans les propriétés de l'image : C:\EduPython\La Laitière Vermeer.jpg.

Le programme en Python

Le code Python	Explications
<pre>from PIL import Image from random import randint</pre>	On active un package de programmes qui nous serviront par la suite
<pre>def double(photographie):</pre>	On définit la fonction double qui attend comme argument le nom d'une image numérique.
<pre>#Etape n°1 Redimenssionnement de l'image citation (c,l)=photographie.size imageres=Image.new('RGB',(c,l//2+1))</pre>	Une ligne de texte On récupère les dimensions de la photographie On définit l'image résultat qui est au format JPEG et qui a les dimensions souhaitées.
<pre># Etape n°2 Construction de l'image de gauche for i in range(0,c,2): for j in range(0,l,2): p=photographie.getpixel((i,j)) imageres.putpixel((i//2,j//2),p)</pre>	Une ligne de texte On parcourt les colonnes de rangs pairs de l'image fournie en entrée et pour chaque colonne, on ne s'intéresse qu'aux lignes de rangs pairs. On relève le code couleur de chaque pixel ainsi identifié et l'on affecte ce code à un pixel de l'image résultat.
<pre># Etape n°3 Construction de l'image de droite for i in range(1,c,2): for j in range(1,l,2): p=photographie.getpixel((i,j)) imageres.putpixel((c//2+i//2,j//2),p)</pre>	Une ligne de texte On parcourt les colonnes de rangs impairs de l'image fournie en entrée et pour chaque colonne, on ne s'intéresse qu'aux lignes de rangs impairs. On relève le code couleur de chaque pixel ainsi identifié et l'on affecte ce code à un pixel de l'image résultat.
<pre># Renvoi de l'image obtenue imageres.show()</pre>	Une ligne de texte On fait afficher l'image résultat.

Nous fournissons à présent le programme in extenso afin que vous puissiez par Copier-coller l'obtenir directement dans la fenêtre d'édition EduPython.

```
from PIL import Image
from PIL import *
def double(photographie):
    #Etape n°1 Redimensionnement de l'image citation
    (c,l)=photographie.size
    imageres=Image.new('RGB',(c,l//2+1))
    # Etape n°2 Construction de l'image de gauche
    for i in range(0,c,2):
        for j in range(0,l,2):
            p=photographie.getpixel((i,j))
            imageres.putpixel((i//2,j//2),p)
    # Etape n°3 Construction de l'image de droite
    for i in range(1,c,2):
        for j in range(1,l,2):
            p=photographie.getpixel((i,j))
            imageres.putpixel((c//2+i//2,j//2),p)
    # Renvoi de l'image obtenue
    imageres.show()
```

Exécution du programme

Avant de lancer le programme, il nous faut activer l'image numérique à traiter. Pour ce faire, on fournit le lien vers ladite image puis l'on prend l'option open qui permet d'ouvrir ce lien.

On nomme im1 l'image ainsi obtenue. Le lancement du programme s'effectue en prenant la fonction double et en donnant comme argument im1.

```
a="C:\EduPython\La Laitière Vermeer.jpg"
im1=Image.open(a)
double(im1)
```

Conclusion

On retrouve dans ce traitement d'une image numérique les capacités du numérique à mettre en œuvre sa rapidité pour effectuer une tâche répétitive. Il est intéressant de noter au passage que notre perception du réel est entièrement soumise au programmeur ; il en va ainsi dans l'image ci-contre de deux images identiques alors que nous savons à présent qu'il n'en est rien.



Jean-Alain Roddier
IA-IPR de mathématiques