NOM :



2nde

**Chap 2 : IMAGES NUMERIQUES (2)**

Dans ce TP, nous allons créer et modifier une image numérique à l’aide du logiciel edupython.

1. **Coloriser des pixels**

Vous allez ouvrir le logiciel Edupython, et copier ce texte :

import numpy as np

import matplotlib.pyplot as plt

img = np.zeros((20, 20, 3), dtype=np.uint8)

…

plt.imshow(img)

plt.show()

Les deux premières instructions permettent d’importer des bibliothèques permettant de manipuler les images.

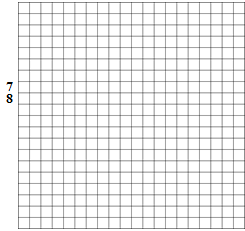
La troisième ligne permet de créer une image noire de 20 pixels sur 20 pixels que l’on appelle img.

Les 3 petits points représentent la partie qui sera complétée ou modifiée tout au long de ce TP.

Les 2 dernières lignes permettent d’afficher cette image (très agrandie pour que l’on voit bien les pixels).

Pour modifier un pixel, on utilise l’instruction « img[x, y] = (r, v, b) ». Dans cette instruction, [x,y] représentent les coordonnées du pixel à modifier, x représente le numéro de la ligne et y le numéro de la colonne.

Et (r, v, b) représentent les valeurs de l’intensité de rouge, de l’intensité de vert et de l’intensité de bleu pour ce pixel. Compléter les 3 petits points précédents avec les instructions :



**img[5, 10] = (0, 150, 0)**

**img[15, 5] = (150, 0, 0)**

Vous allez voir apparaitre un pixel vert et un pixel rouge dans votre dessin. Essayez de comprendre la numérotation des lignes et des colonnes.

Complétez le quadrillage ci-contre en continuant de numéroter correctement les lignes et les colonnes :

**Exercice 1 :**



Compléter le programme ci-dessous afin de faire apparaitre un pixel bleu à peu près au centre de l’écran et des pixels de différentes couleurs aux coins de l’écran.

**img[......, ……] = (……, ……, ……)**

**img[......, ……] = (……, ……, ……)**

**img[......, ……] = (……, ……, ……)**

**img[......, ……] = (……, ……, ……)**

**img[......, ……] = (……, ……, ……)**

Quelle est la définition de l’image créée? (voir le cours de la semaine précédente)

……………………………………………………………………………………………………………………………………………………………………….

Donnez approximativement la résolution de l’image ainsi créée.

……………………………………………………………………………………………………………………………………………………………………….

1. **Coloriser des lignes**

On repart du fichier initial (voir page précédente), mais cette fois-ci on va remplacer les 3 petits points par les instructions suivantes :



**for k in range(0,20):**

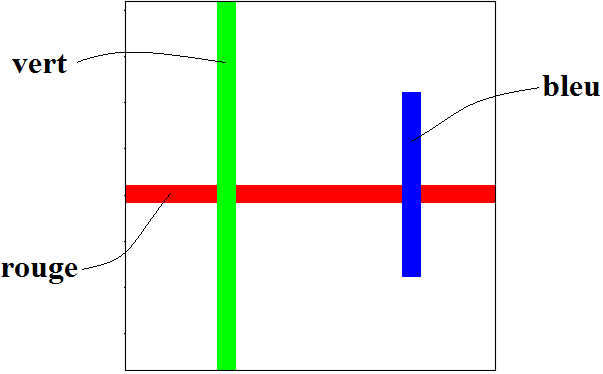
**img[10, k] = (0, 250, 150)**

On obtiendra alors une ligne bleue clair au centre de l’écran.



**Exercice 2 :**

Compléter le programme ci-dessous afin d’obtenir le dessin ci-dessous.



**for k in range(0,20):**

**img[ ……, …… ] = ( …… , …… , …… )**

**for k in range(0,20):**

**img[ ……, …… ] = ( ……, …… , …… )**

**for k in range( …… , …… ):**

**img[ ……, …… ] = ( ……, …… , …… )**

1. **Coloriser des formes**

On repart du fichier initial (voir page précédente), mais cette fois-ci on va remplacer les 3 petits points par les instructions suivantes :



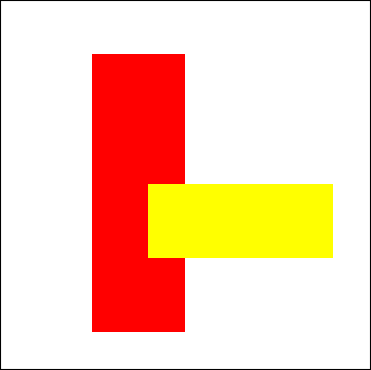
**for j in range(5,15):**

**for k in range(3,12):**

**img[j, k] = (150, 0, 50)**

On obtiendra alors un rectangle rose au centre de l’écran.

**Exercice 3 :**



Compléter le programme ci-dessous afin d’obtenir le dessin ci-contre.



**for j in range( …… , …… ):**

**for k in range( …… , …… ):**

**img[j, k] = (150, 0, 50)**

**for j in range( …… , …… ):**

**for k in range( …… , …… ):**

**img[j, k] = (50, 150, 0)**

On repart du fichier initial (voir première page), mais cette fois-ci on va remplacer les 3 petits points par les instructions suivantes :



**u = 10**

**for j in range( 2 , 12 ):**

**for k in range( 15-u , 15 ):**

**img[j, k] = ( 155, 255, 0 )**

**u = u - 1**

On obtiendra alors une sorte de triangle pixélisé.

On peut remarquer que les nombres « 2 » et « 12 » font référence aux numéros des lignes, et que « 15 » est le numéro de la colonne la plus à droite.

On commence par colorier 10 pixels sur la ligne « 5 », puis sur la ligne « 6 », on colorie seulement 9 pixels, puis sur la ligne « 7 », on colorie 8 pixels et ainsi de suite…

On peut aussi imaginer dessiner le triangle dans l’autre sens :

**u = 10**



**for j in range( 2 , 12 ):**

**for k in range( 5 , 5 + u ):**

**img[j, k] = ( 155, 255, 0 )**

**u = u - 1**

On peut remarquer que les nombres « 2 » et « 12 » font référence aux numéros des lignes, et que « 5 » est le numéro de la colonne la plus à gauche.

On commence par colorier 10 pixels sur la ligne « 5 », puis sur la ligne « 6 », on colorie seulement 9 pixels, puis sur la ligne « 7 », on en colorie 8 et ainsi de suite…

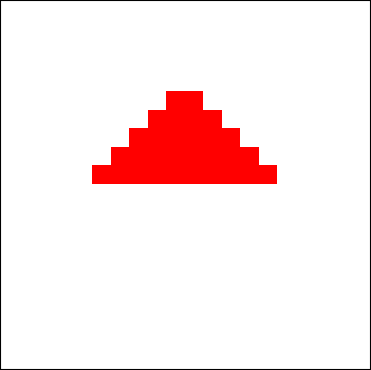
**Exercice 4 :**



Compléter le programme ci-dessous afin d’obtenir le dessin ci-contre.



**u = 1**



**for j in range( …… , …… ):**

**for k in range( …… , …… ):**

**img[j, k] = (255, 0, 0)**

**u = u + 1**

**u = 1**

**for j in range( ……. , …… ):**

**for k in range( …… , …… ):**

**img[j, k] = (255, 0, 0)**

**u = u+ 1**

1. **Faire des dégradés**

On repart du fichier initial (voir première page), mais cette fois-ci on va remplacer les 3 petits points par les instructions suivantes (attention à l’indentation de la dernière ligne) :



**u = 0**

**for j in range(0,20):**

**for k in range(0,20):**

**img[j, k] = (u, 0, 0)**

**u = u + 12**

On obtiendra alors un dégradé de rouge.



**Exercice 5 :**

Compléter le programme ci-dessous afin d’obtenir un dégradé de bleu :



**u = 0**

**for j in range(0,20):**

**for k in range(0,20):**

**img[j, k] = (…… , …… , …… )**

**u = u + 12**

Compléter le programme ci-dessous afin d’obtenir un dégradé allant du bleu au vert :



**u = 0**

**for j in range(0,20):**

**for k in range(0,20):**

**img[j, k] = (…… , …… , …… )**

**u = u + 12**

Essayez d’obtenir un dégradé qui passe successivement par 3 couleurs différentes et montrez le résultat à votre professeur.

**Correction:**

import matplotlib.image as mpimg

import matplotlib.pyplot as plt

img = np.zeros((20, 20, 3), dtype=np.uint8)

----------------

**Exercice 1:**

img[5, 10] = (0, 150, 0)

img[15, 5] = (0, 0, 0)

-------------------------------

**Exercice 2:**

for k in range(0,20):

img[10, k] = (255, 0, 0)

for k in range(0,20):

img[k, 5] = (0, 255, 0)

for k in range(5,15):

img[k, 15] = (0, 0, 255)

-----------------------------------

**Exercice 3:**

for j in range(3,18):

for k in range(5,10):

img[j, k] = (150, 0, 0)

for j in range(10,14):

for k in range(8,18):

img[j, k] = (50, 150, 0)

---------------------------------------

**Exercice 4:**

u=1

for j in range(5,10):

for k in range(10-u,10):

img[j, k] = (155, 0, 0)

u=u+1

u=1

for j in range(5,10):

for k in range(10,10+u):

img[j, k] = (155, 0, 0)

u=u+1

----------------------------------

**Exercice 5:**

u = 0

for j in range(0,20):

for k in range(0,20):

img[j, k] = (0, 0, u)

u = u + 12

u = 0

for j in range(0,20):

for k in range(0,20):

img[j, k] = (0, u, 255-u)

u = u + 12

u = 0

for j in range(0,10):

for k in range(0,20):

img[j, k] = (0, u, 255-u)

u = u + 24

for j in range(10,20):

for k in range(0,20):

img[j, k] = (255-u, u, 0)

u = u -24

----------------------------------

plt.imshow(img)

plt.show()