

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

TP rotation image

Christophe Viroulaud

Terminale - NSI

Algo 03

[Principe](#)[Algorithme de rotation](#)[Chargement de l'image](#)[Résoudre un petit problème](#)[Diviser](#)[Créer une bibliothèque](#)

La rotation d'une image est une fonctionnalité proposée par n'importe quel logiciel de retouche tel *Gimp*. L'opération n'est cependant pas triviale et peut demander une durée non négligeable.

Principe

Algorithme de
rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une
bibliothèque

Construire un algorithme de rotation d'une image en appliquant le principe de *diviser pour régner*.

1. Principe

2. Algorithme de rotation

3. Créer une bibliothèque

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Principe

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Diviser pour régner se décompose en trois parties :

- ▶ *diviser* : Le problème est partagé en plusieurs petits problèmes identiques.
- ▶ *traitement* : Chaque petit problème est résolu.
- ▶ *recombinaison* : Les petits problèmes résolus sont assemblés pour remonter au problème principal.

Principe

Algorithme de rotation

Chargement de l'image
Résoudre un petit problème
Diviser

Créer une bibliothèque

Activité 1 : Réflexion commune : Considérons une image aux dimensions connues. Quelles étapes pourrions-nous imaginer pour répondre à notre problématique ?

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Principe

Algorithme de rotation

- Chargement de l'image
- Résoudre un petit problème
- Diviser

Créer une bibliothèque



FIGURE 1 – 1 pixel : rien à faire

Principe

Algorithme de rotation

- Chargement de l'image
- Résoudre un petit problème
- Diviser

Créer une bibliothèque

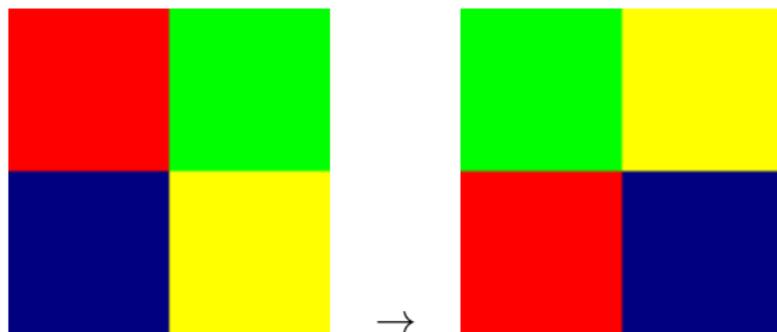


FIGURE 2 – Rotation

Principe

Algorithme de rotation

Chargement de l'image
Résoudre un petit problème
Diviser

Créer une bibliothèque



FIGURE 3 – Récursivité : on divise la taille des problèmes par 2.

Principe

Algorithme de rotation

Chargement de l'image
Résoudre un petit problème
Diviser

Créer une bibliothèque

- ▶ Si la taille t est égal à 1, ne rien faire.
- ▶ Sinon : **découper en sous problèmes**
 - ▶ diviser la taille t en 2,
 - ▶ effectuer récursivement la rotation des **quatre parties** de la portion carrée comprise entre (x,y) et $(x+t, y+t)$
- ▶ **résoudre les petits problèmes** : Effectuer la rotation des pixels.

1. Principe

2. Algorithme de rotation

2.1 Chargement de l'image

2.2 Résoudre un petit problème

2.3 Diviser

3. Créer une bibliothèque

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Chargement de l'image

PIL (Python Image Library) -anciennement *pillow*- est une bibliothèque de traitement d'image.

```
1 from PIL import Image
2
3 im = Image.open("image.png")
4 im.show()
```

Code 1 – Charger une image

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

```
1 largeur, hauteur = im.size
2 px = im.load()
```

Code 2 – Récupérer des informations

Information

La variable `px` contient une matrice représentative des pixels de l'image. La couleur du pixel de coordonnées (x,y) est donnée par l'instruction `px[x,y]`. Il est également possible d'affecter une nouvelle couleur `c` à un pixel : `px[x,y] = c`.

Principe

Algorithme de
rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une
bibliothèque

Activité 2 :

1. Récupérer une image carrée sur <https://www.freepng.fr/>.
2. Charger et afficher cette image.

1. Principe

2. Algorithme de rotation

2.1 Chargement de l'image

2.2 Résoudre un petit problème

2.3 Diviser

3. Créer une bibliothèque

Principe

Algorithme de rotation

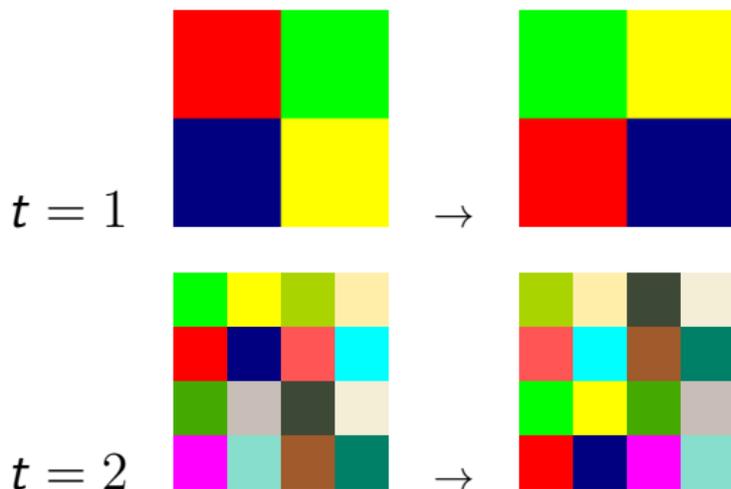
Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Résoudre un petit problème



Activité 3 : Écrire la fonction `tourner(px: object, x: int, y: int, t: int) → None` qui effectue une rotation anti-horaire pour les pixels compris dans l'intervalle de colonnes $[x; x + t]$ et l'intervalle de lignes $[y; y + t]$.

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

```
1 def tourner(px: object, x: int, y: int, t: int) ->
  None:
2   for l in range(y, y+t):
3     for c in range(x, x+t):
4       px[l, c+t], px[l+t, c+t], \
5       px[l+t, c], px[l, c] = \
6       px[l, c], px[l, c+t], \
7       px[l+t, c+t], px[l+t, c]
```

Code 3 – Chaque *bloc* tourne d'un cran dans le sens anti-horaire

1. Principe

2. Algorithme de rotation

2.1 Chargement de l'image

2.2 Résoudre un petit problème

2.3 Diviser

3. Créer une bibliothèque

Principe

Algorithme de rotation

Chargement de l'image

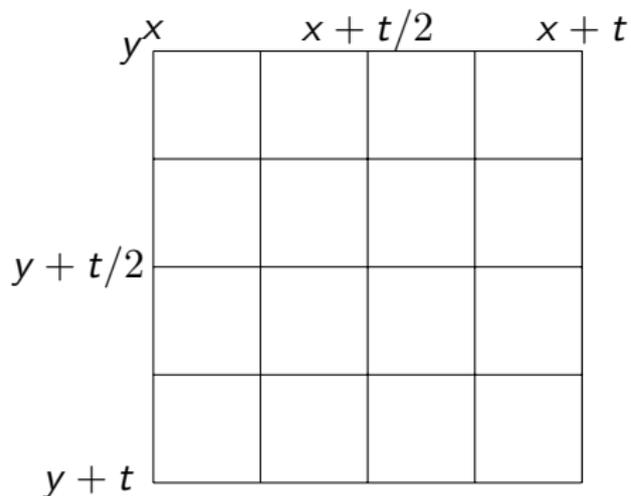
Résoudre un petit problème

Diviser

Créer une bibliothèque

Diviser

Partant d'une taille d'image t il faut diviser le problème en quatre problèmes plus petits.



Principe

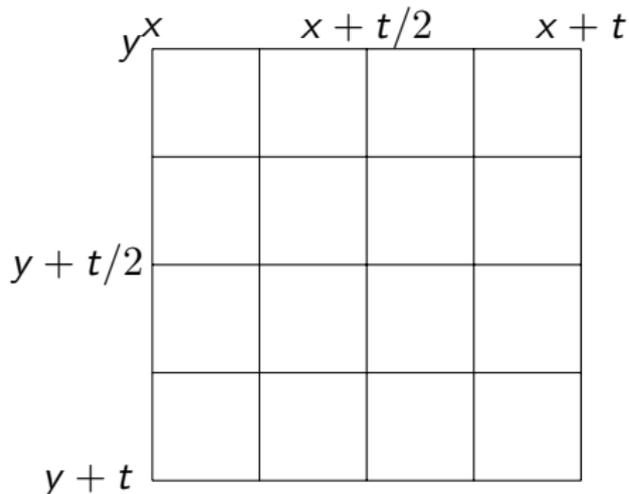
Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque



Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Activité 4 : Écrire la fonction récursive `rotation(px: object, x: int, y: int, t: int) → None` qui divise récursivement le problème en quatre, puis effectue la rotation à l'aide de la fonction `tourner`.

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Correction

```
1 def rotation(px: object, x: int, y: int, t:
   int) -> None:
2     if t > 1:
3         t //= 2
4         rotation(px, x, y, t)
5         rotation(px, x+t, y, t)
6         rotation(px, x, y+t, t)
7         rotation(px, x+t, y+t, t)
8
9     tourner(px, x, y, t)
```

Code 4 – Le cas limite est atteint quand on a 1 seul pixel.

```
1 rotation(px, 0, 0, largeur)
```

Code 5 – Appel principal

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

1. Principe

2. Algorithme de rotation

3. Créer une bibliothèque

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Créer une bibliothèque

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Pour un utilisateur, le passage des différents paramètres (\mathbf{px} , \mathbf{x} , \mathbf{y} , \mathbf{t}) peut paraître fastidieux. Typiquement il ne devrait avoir à fournir qu'une information : le chemin du fichier image.

Activité 5 :

1. Créer une classe `Image_perso` et son constructeur qui admet un paramètre : le chemin de l'image. Ce constructeur construira alors un objet `Image` de la bibliothèque `PIL`.
2. Écrire la méthode `montrer`, sans paramètre, qui affiche l'image.
3. Écrire la méthode `fait_tourner` sans paramètre, qui effectue une rotation anti-horaire de l'image. Il faudra adapter les fonctions construites précédemment pour en faire des méthodes internes à la classe.
4. **Pour les plus avancés** : Adapter la méthode `fait_tourner` qui acceptera un paramètre de type booléen. Si l'argument passé est `True` la rotation sera dans le sens horaire.

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

Principe

Algorithme de rotation

Chargement de l'image
Résoudre un petit problème
Diviser

Créer une bibliothèque

```
1 class Image_lib:
2
3     def __init__(self, fichier: str) -> None:
4         self.image = Image.open(fichier)
5         self.largeur, self.hauteur = self.
image.size
6         self.px = self.image.load()
```

Code 6 – Constructeur

Principe

Algorithme de
rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une
bibliothèque

```
1 def montrer(self):  
2     """  
3     affiche l'image  
4     """  
5     self.image.show()
```

Code 7 – Afficher l'image

```
1 def fait_tourner(self) -> None:
2     """
3     tourne l'image de 90° dans le sens anti-
4     horaire sinon
5     """
6     self.rotation(0, 0, self.largeur)
```

Code 8 – Il faut adapter la fonction `rotation`.

Remarque

Le paramètre `px` n'est plus nécessaire dans la méthode `rotation`. En effet, il est accessible depuis l'attribut créé dans le constructeur.

Principe

Algorithme de rotation

Chargement de l'image
Résoudre un petit problème
Diviser

Créer une bibliothèque

```
1 def rotation(self, x: int, y: int, t: int,  
2     horaire: bool) -> None:  
3     if t > 1:  
4         t //= 2  
5         self.rotation(x, y, t, horaire)  
6         self.rotation(x+t, y, t, horaire)  
7         self.rotation(x, y+t, t, horaire)  
8         self.rotation(x+t, y+t, t, horaire)  
9     self.tourner(x, y, t)
```

Principe

Algorithme de rotation

Chargement de l'image

Résoudre un petit problème

Diviser

Créer une bibliothèque

```
1 def tourner(self, x: int, y: int, t: int) -> None:
2     for l in range(y, y+t):
3         for c in range(x, x+t):
4             self.px[l, c+t], self.px[l+t, c+t], \
5             self.px[l+t, c], self.px[l, c] = \
6                 self.px[l, c], self.px[l, c + t], \
7                 self.px[l+t, c+t], self.px[l+t, c]
```

```
1 def fait_tourner(self, horaire: bool = True)
  -> None:
2     """
3     tourne l'image de 90°
4
5     Paramètres
6     -----
7     horaire: booléen; défaut: True
8         tourne de 90° dans le sens horaire si
9         True,
10        dans le sens anti-horaire sinon
11    """
    self.rotation(0, 0, self.largeur, horaire
    )
```

Code 9 – Avec choix de la rotation

Principe

Algorithme de
rotationChargement de l'image
Résoudre un petit problème
DiviserCréer une
bibliothèque

Principe

Algorithme de rotation

Chargement de l'image
Résoudre un petit problème
Diviser

Créer une bibliothèque

```
1 def rotation(self, x: int, y: int, t: int,
2     horaire: bool) -> None:
3     if t > 1:
4         t //= 2
5         self.rotation(x, y, t, horaire)
6         self.rotation(x+t, y, t, horaire)
7         self.rotation(x, y+t, t, horaire)
8         self.rotation(x+t, y+t, t, horaire)
9
10    if horaire:
11        self.tourner_horaire(x, y, t)
12    else:
13        self.tourner_antihoraire(x, y, t)
```

Code 10 – adaptation de rotation

```
1 def tourner_antihoraire(self, x: int, y: int, t: int)
  -> None:
2     for l in range(y, y+t):
3         for c in range(x, x+t):
4             self.px[l, c+t], self.px[l+t, c+t], \
5                 self.px[l+t, c], self.px[l, c] = \
6                 self.px[l, c], self.px[l, c + t], \
7                 self.px[l+t, c+t], self.px[l+t, c]
8
9 def tourner_horaire(self, x: int, y: int, t: int) ->
  None:
0     for l in range(y, y+t):
1         for c in range(x, x+t):
2             self.px[l, c+t], self.px[l+t, c+t], \
3                 self.px[l+t, c], self.px[l, c] = \
4                 self.px[l+t, c+t], self.px[l+t, c], \
5                 self.px[l, c], self.px[l, c+t]
```