



# A la base des effets spéciaux...



**L'incrustation** est une technique d'effets visuels dans le domaine du cinéma et de la photo. Cela consiste à intégrer dans une même image par logiciel, des objets filmés séparément ou des objets 3D par ordinateur.

Nous pouvons donc recréer des environnements irréels, ou placer un sujet dans une situation sans même bouger du studio.

Dans la grande majorité des cas, on utilise un fond vert pour permettre aux logiciels de détacher les contours de l'objet que l'on cherche à incruster.

## ➤ Pourquoi le vert ?

Avant la naissance de techniques numériques de pointe, c'était le bleu qui était utilisé, car il permettait d'avoir un minimum de grain et donc une meilleure lecture des détails.

- Le vert s'est substitué au bleu, qui continue tout de même d'être utilisé dans de rares cas.
- Le vert est la couleur la plus éloignée de celle de la peau humaine. Le vert permet donc à la peau des acteurs de bien se détacher du fond et donc à l'incrustation de pas déborder sur les visages des gens.
- Les capteurs vidéo ont une sensibilité pour le vert. Comme ils la repèrent mieux, l'incrustation est beaucoup plus précise mais aussi, grâce à cette grande sensibilité pour cette couleur, les fonds ont besoin de moins de lumière.



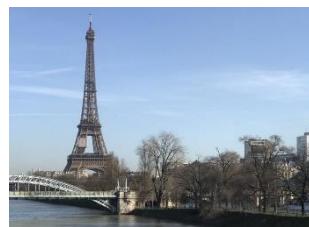


## ➤ Manipulation d'un fond vert avec Python

L'objectif de l'activité est de d'enlever le fond vert de l'image « **stardestroyer.png** » et de le remplacer par le fond « **Paris.jpg** ».



+



=



**ATTENTION : Pour réaliser cette activité, les images doivent avoir exactement la même taille ! Redimensionnez vos images avec Photofiltre si ce n'est pas le cas.**

Dans l'exemple ci-dessus, les deux images font 800 pixels x 600 pixels.

Téléchargez votre document réponse que vous enregistrerez dans le dossier « **Documents/SNT/THEME\_7** » sous le nom « **a732-VotreNomDeFamille.doc** ».

Prenez connaissance de la fiche de cours : [les formats d'images](#) (CTRL+CLIC)

Question : pourquoi utiliser ici une image au format « **.png** » plutôt que le format « **.jpg** »

### 1. Analyse de la première image

Les fichiers « **stardestroyer.png** » et « **Paris.jpg** » sont à télécharger depuis le site et à enregistrer dans le dossier « **Documents/SNT/THEME\_7** ».

Tous les fichiers de cette activité devront être stockés dans ce dossier.

Ouvrez l'IDE Pyzo à exécuter depuis « [Autres raccourcis/SNT/Pyzo\\_SNT](#) ».

Créez le programme « **a731-01-détourage.py** » qui contiendra les lignes suivantes :

```
from PIL import Image  
  
img1 = Image.open("stardestroyer.png")  
p = img1.getpixel((0,0))  
print(p)
```

1. Exécutez le script et observez le code couleur du pixel affiché dans le Shell. Le code affiché est de la forme (R,V,B,T) soit (Rouge, Vert, Bleu, Transparence) ; on ne gère pas la transparence (T=255). Ce résultat vous paraît-il correct ?
2. Modifiez votre code afin d'observer la valeur du pixel en bas à droite de l'image, puis essayer de trouver les coordonnées d'un pixel qui ne soit pas un pixel vert.
3. Rajoutez une ligne afin d'afficher le message « **Ce pixel est vert** » si la valeur du pixel observé correspond effectivement à celle d'un pixel vert.



## 2. Parcours de tous les pixels de l'image

**Activité 1 :** À l'aide d'une double boucle, parcourez tous les pixels de l'image. Lorsque vous rencontrez un pixel vert, affichez le message « **Ce pixel est vert** ».

**Aide :** Vous pouvez utiliser le code du programme « **a721-image\_test.py** » de l'activité 2 (en l'adaptant).

Vous pouvez interrompre votre code si son exécution totale prend trop de temps.

**Activité 2 :** Modifiez votre code précédent afin que tous les pixels verts deviennent bleus. Vous pouvez sauvegarder votre image en rajoutant `img1.save("stardestroyerbleu.png")` en fin de programme.

**Rappel :** L'expression `img1.putpixel((x,y),(255,0,0))` permet de colorier le pixel de coordonnées (x,y) en rouge.

Pensez à enlever la ligne `print("le pixel est vert")` de votre code, car elle ralentit le programme...

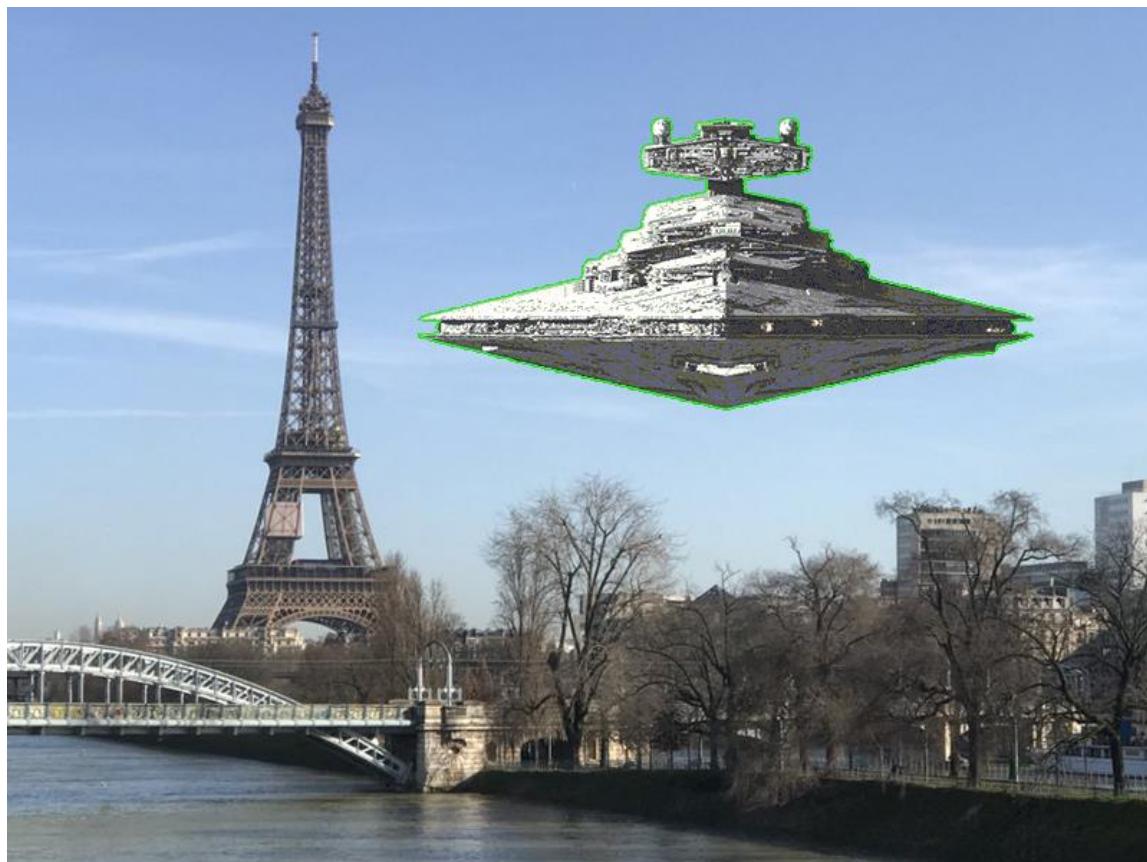
## 3. Fusion des deux images

Vous allez maintenant ouvrir la deuxième image « **Paris.jpg** » en ajoutant la ligne de code suivante :

```
img2 = Image.open("Paris.jpg")
```

En vous inspirant des parties précédentes, remplacez chaque pixel vert par le pixel situé au même endroit.

```
q = img2.getpixel((x,y)) #permet de lire un pixel sur l'image 2 (img2)
```

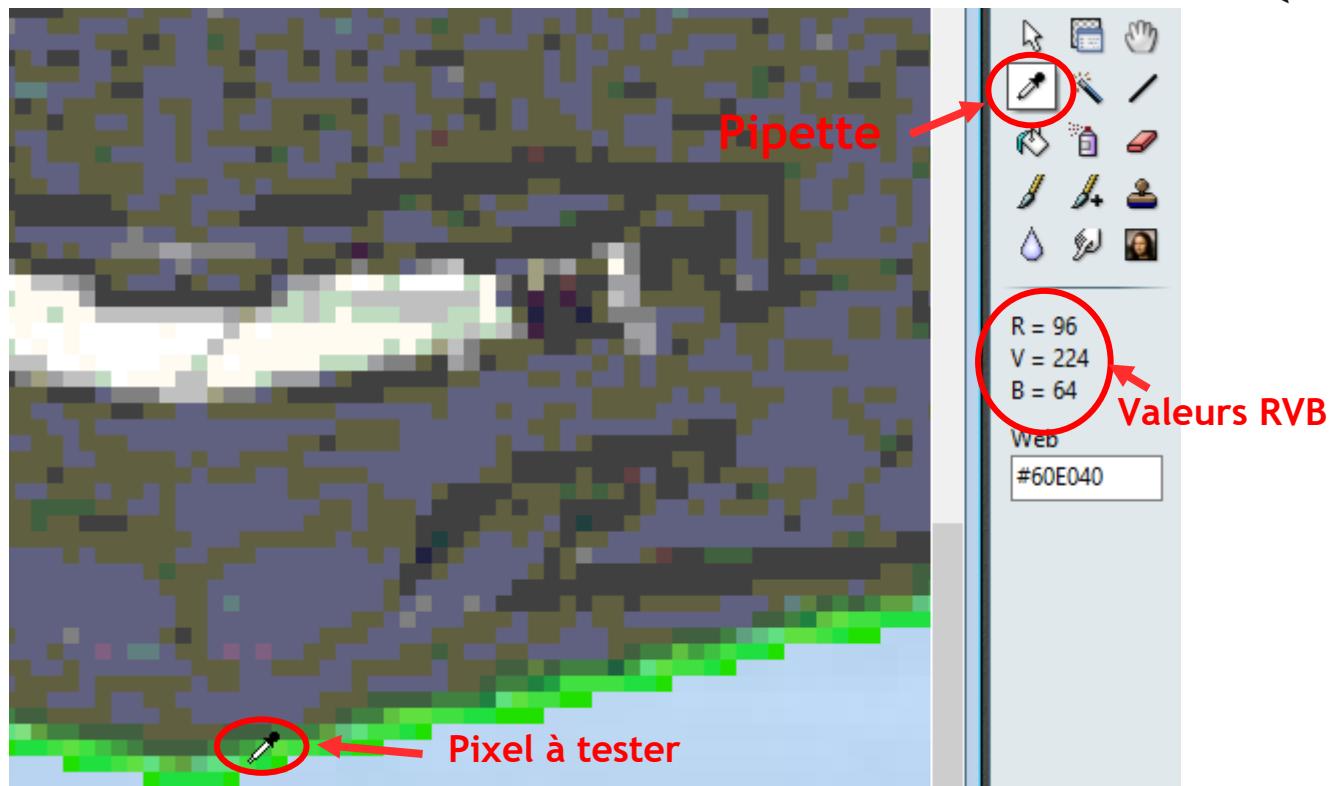




## 4. Amélioration du programme

Le détourage réalisé est imparfait, des pixels verts bordent encore la surface du vaisseau.

**Activité 3 :** ouvrez l'image réalisée avec **Photofiltre**. Avec la pipette, récupérez la valeur de la couleur des pixels qui bordent les contours du vaisseau.



`p = img1.getpixel((x,y))` permet de sauvegarder la valeur d'un pixel dans un **tuple** (une sorte de dictionnaire), où chaque valeur est séparée par une virgule.

En réalité p contient 4 valeurs :

- `p[0]` = Valeur de rouge (R)
- `p[1]` = Valeur de vert (V)
- `p[2]` = Valeur de bleu (B)
- `p[3]` = Transparence du pixel (ici on le laisse toujours à 255, c'est-à-dire opaque à 100%) (A)

`p == (p[0], p[1], p[2], p[3])`

On affectera une variable à chaque tuple :

```
R = p[0]
V = p[1]
B = p[2]
A = p[3]
```



## LA PHOTOGRAPHIE NUMÉRIQUE

De ce fait dans le test de la condition if, nous augmenterons la plage de mesure :

Par exemple, au lieu de tester : `if p == (0,255,0,255) :`

Nous pouvons augmenter le choix des teintes de vert :

```
if R <= 100 and V >= 200 and B <= 100 and A == 255:
```

Ce code signifie que le pixel sera remplacé si :

- La valeur de rouge est inférieure à 100
- et que la valeur de vert est supérieure à 200
- et que la valeur de bleu est inférieure à 100
- et que la valeur de transparence reste à 255

**Activité 4 :** testez votre programme avec ces nouvelles conditions. Qu'observez-vous ?

Essayez d'améliorer le programme jusqu'à un résultat convenable.

## ➤ L'intelligence artificielle au service des effets spéciaux.

De nouvelles techniques de détourage d'arrière-plan basées sur le deep-learning sont maintenant disponibles. Basé sur l'apprentissage grâce à des millions d'exemples d'images détournées, l'algorithme du site [remove.bg](http://remove.bg) (par exemple) produit des résultats impressionnantes.

Il est intéressant de le tester avec des photos d'animaux : le détourage échoue car la base de données d'entraînement ne contenait que des êtres humains.



**Activité 6:** envoyez votre photo sur le site remove.bg puis faites une importation du fichier.png correspondant sur un paysage dans le logiciel GIMP. Commentez le résultat...

Comment partager vos fichiers en vidéo



**CTR + Clic**