

# Compte rendu projet « Image stylizer »

Au départ, mon objectif principal était de produire un programme avec Tkinter et PIL, qui permettrait de modifier une image importée par l'utilisateur, en lui appliquant un processus de « quantization », et en lui laissant ensuite modifier les couleurs à souhait.

Au final, j'ai réalisé ces objectifs, et j'ai en plus de cela ajouté une option d'exportation, qui permet d'avoir en plus de l'aperçu un rendu en haute qualité. Évidemment, le temps de rendu dépend entièrement de la puissance de l'ordinateur utilisé. J'ai également ajouté quelques boutons liés à des fonctions aléatoires, qui donnent à un utilisateur qui manquerait d'inspiration la possibilité de laisser l'algorithme tenter des modifications.

Je m'attendais à trouver le plus de difficultés dans la modification de l'image et dans les fonctions utilisant PIL pour changer les couleurs. Il se trouve que cette partie était au contraire moins difficile que l'implémentation dans une interface dynamique avec Tkinter, qui a été beaucoup plus fastidieuse et complexe.

## Les modules

En termes de modules, j'ai finalement utilisé :

- **PIL** (Image, ImageTk) : pour modifier des images et les afficher avec tkinter
- **Tkinter** : pour créer une interface, et Tkinter.filedialog pour ouvrir des fenêtres système pour ouvrir/sauvegarder des fichiers
- **Random** : pour générer des nombres aléatoires.
- **Colorsys** : pour gérer les calculs complexes de conversion de RGB à HSV et inversement.

Le projet étant séparé en 2 fichiers (un fichier d'exécution et un répertoire de fonctions diverses), j'importe dans "JeremiesImageStylizer.py" les fonctions du fichier "FonctionsFile.py" comme d'un module.

## L'UX (User Experience)

Pour lancer le programme, il faut exécuter le fichier "JeremiesImageStylizer.py" avec un éditeur python comme IDLE. Le dossier Images contient quelques images pour tester le programme.

L'expérience utilisateur se divise en 4 phases :

1. Choix d'une image, avec une fenêtre système
2. Choix du nombre de teintes et traitement de l'image par l'algorithme
3. Modification de l'image par l'utilisateur

4. Export de l'image grâce au bouton « Export », qui affiche une fenêtre système.

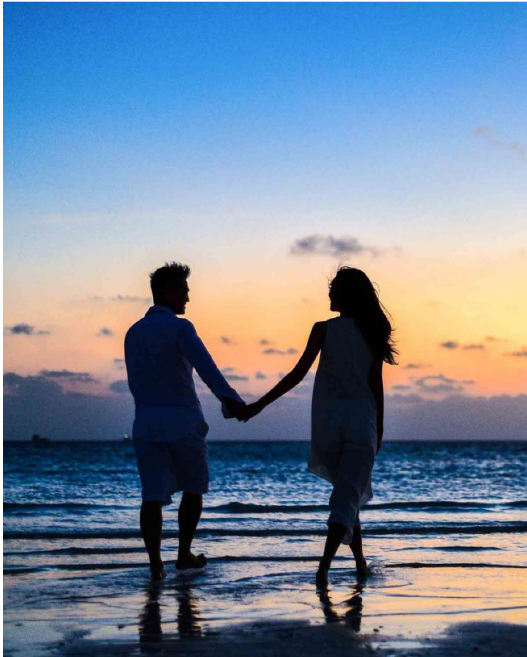
## L'interface

La phase 3, qui correspond à la modification de l'image, a une interface composée de plusieurs boutons/options.

- Le bouton « **Random** » change toutes les couleurs en des couleurs complètement aléatoires.
- Le bouton « **Random (Keep contrast)** » change les couleurs de cette façon : d'abord, il les convertit dans le système de coordonnées HSV, puis modifie aléatoirement l'angle H (Hue) de chaque teinte. Cela permet de conserver la saturation et la luminosité de chaque couleur.
- Le bouton « **Random (Keep coherence)** » change les couleurs d'une façon plus ou moins similaire : il les convertit en HSV, puis modifie aléatoirement l'angle H (Hue) de toutes les teintes, du même angle. Cela permet de conserver non seulement la saturation et la luminosité de chaque couleur, mais en plus de conserver une certaine cohérence colorimétrique : les couleurs restent en rapport entre elles. Par exemple, les couleurs complémentaires restent complémentaires.
- Le bouton « **Reset** » permet de réinitialiser l'image comme elle était juste après son traitement.
- Les **entrées** (qui sont au nombre des teintes de l'image une fois traitée) représentent donc cesdites teintes. Le fond de chaque entrée est de la couleur qu'elle représente, pour transformer les codes hexadécimaux en un langage visuel, compréhensible par l'utilisateur. Si la couleur est sombre, le texte sera alors affiché en blanc pour la lisibilité. L'édition directe du code hexadécimal est possible, et les transformations presque instantanées, de sorte à ce qu'il soit facile de reporter une charte de couleurs sur l'image.
- Les **boutons** à droite des entrées permettent de choisir la couleur plus aisément, à l'aide d'une fenêtre, le « colorChooser », qui dépend du système d'exploitation. Cette dernière est initialisée avec la valeur de la couleur actuelle, pour faciliter les petites modifications de couleur. L'expérience est meilleure sur Windows que sur Linux.
- Le bouton « **Export** » permet d'exporter l'image traitée, en appliquant les modifications faites sur une image de haute qualité (la qualité de l'export dans le code peut être modifiée avec la constante EXPORT\_QUALITY, elle est par défaut à 1920, ce qui signifie que la dimension - x ou y - la plus grande de l'image ne dépassera pas 1920 px). Enfin, si l'export est fait sans préciser l'extension, l'image sera automatiquement enregistrée au format png.

## Exemples d'images produites

Avant



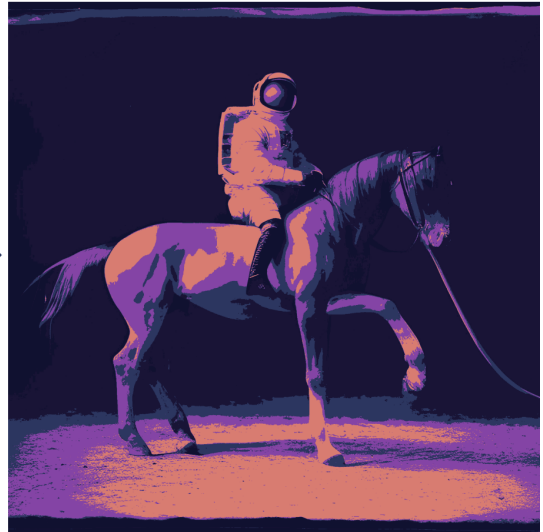
Après



Avant



Après



Avant



Après



Merci d'avoir lu ce compte rendu !