

Le but de ce TP est d'écrire une première version du projet utilisant les éléments étudiés précédemment (interface graphique, gestion des fichiers). Il faut organiser le code en différents modules de façon rigoureuse. Placez vous dans votre répertoire correspondant au projet de programmation 1 et créez un répertoire `projet`. Un module de lecture/écriture de fichiers PGM binaire vous est fourni. Téléchargez le à l'aide de la commande : `wget http://dept-info.labri.fr/ENSEIGNEMENT/projetprog1/src/fichiers-3.tar.gz`

Modules pgm et tableaux2D

La structure de données représentant une image en niveau de gris sera un tableau à deux dimensions d'entiers. C'est cette structure qui est utilisée dans le module `pgm.py`.

Étape 1 – Ecrire et tester le module `tableaux2D.py` contenant les fonctions :

- `creer_tableau (h, l, val)` qui retourne une matrice de hauteur `h` et de largeur `l` dont toutes les valeurs sont initialisées à `val`.
- `copier_tableau (t1, t2)` qui recopie les valeurs du tableau `t1` dans le tableau `t2`. Ces deux tableaux sont des tableaux à deux dimensions de même taille.

Étape 2 – Tester le module `pgm.py`.

Programme principal

Étape 3 – Ecrire le programme principal. Ce sera un script python de nom `main` réalisant la création des composants graphiques de l'interface présents au lancement de l'application. Ce script ne contiendra aucune définition de fonction.

La fenêtre principale contiendra une barre de menu avec deux menus déroulants :

- un menu *Fichier* avec les items *Ouvrir...* (choix d'un nom de fichier image à ouvrir), *Sauver sous...* (choix d'un nom de fichier destiné à stocker l'image courante) et *Quitter* (quitter l'application) ;
- un menu *Traitement d'image* dont chaque item correspondra à un algorithme de traitement d'image. Dans un premier temps, ce menu ne contiendra qu'un item qui permettra d'appliquer un premier traitement à l'image courante (une symétrie gauche/droite, voir Etape 5).

La fenêtre principale contiendra également un canevas où seront affichées les images et une zone de texte positionnée sous le canevas destinée à afficher des informations comme le niveau de gris d'un pixel.

Module interface

Étape 4 – Ecrire dans un fichier `interface.py` toutes les fonctions appelées lors d'un événement (les callbacks).

On rappelle que l'utilisateur peut :

- Cliquer sur l'image courante et obtenir ainsi les coordonnées de la souris et le niveau de gris de l'image sous le curseur. Cette fonctionnalité n'a de sens que si une image est ouverte.
- Ouvrir un fichier image : l'image est alors affichée dans le canevas dont la taille s'adapte à la taille de l'image.
- Appliquer un traitement à l'image courante. L'image traitée remplace l'image initiale dans le canevas. Rien ne se passe s'il n'y a pas d'image ouverte. Les algorithmes de traitement d'image ne sont pas implémentés dans ce module mais dans le module `traitement` (voir question suivante).
- Sauver l'image courante dans un fichier. Cette fonctionnalité n'a de sens que si une image est ouverte.

Module `traitement`

Ce module (fichier `traitement.py`) regroupera toutes les fonctions de traitement d'image. Dans un premier temps, toutes les fonctions de traitement d'image auront un unique paramètre : le tableau contenant les valeurs des pixels. Ce tableau est modifié par la fonction de traitement.

Étape 5 — Ecrire une fonction qui réalise une symétrie autour d'une droite verticale centrée dans l'image (échange entre gauche et droite).

Étape 6 — S'il vous reste du temps :

Ecrire une fonction qui ajoute des bandes noires verticales sur l'image traitée.