

# Projet de programmation C avancée 2015/16

---

## Objectif

---

Le projet consiste à réaliser un outils de traitement d'images en ligne de commande à la façon de l'outils convert du projet Image Magick .

## Interface

---

Votre application, disons `app.exe`, pourra être lancée avec ou sans option. Elle travaillera alors sur une image courante et effectuera les différentes opérations requises par l'utilisateur par le menu ou à travers le système d'options. Lorsque aucune image courante n'est chargée (ce qui est le cas initialement), tout appel à une fonctionnalité entraînera l'affichage d'un avertissement comme quoi l'application ne peut pas appliquer l'opération associée ; l'application poursuivra alors son traitement en ignorant la demande.

### Lancement sans option

Lorsque votre application est lancée sans option par :

`./app.exe`

le menu suivant est alors lancé :

- Fichier
  - Ouvrir : demande à l'utilisateur un nom de fichier contenant une image et charge l'image en mémoire ; si une image courante est déjà chargée, la sauvegarde de celle-ci sera proposée automatiquement ; l'ouverture d'une nouvelle image entraînera la libération mémoire de l'image courante précédente
  - Sauvegarder : sauvegarde l'image courante dans un fichier dont le nom devra être spécifié par l'utilisateur (le nom du fichier d'origine sera proposé par défaut)
    - PBM binaire : l'image sera convertie automatiquement en noir et blanc avec un seuil de 128 (voir c-dessous)
    - PBM ASCII : l'image sera convertie automatiquement en noir et blanc avec un seuil de 128 (voir ci-dessous)
    - PGM binaire : l'image sera convertie automatiquement en niveaux de gris
    - PGM ASCII : l'image sera convertie automatiquement en niveaux de gris
    - PPM binaire
    - PPM ASCII
    - PNG
    - BMP
    - ...
  - Information : affiche les informations concernant l'image courante (nom du fichier d'origine, hauteur et largeur en pixels) puis affiche le canal G de l'image en ASCII Art dans un format respectant les dimensions de l'image mais tenant sur un écran 80×60 caractères
  - Quitter : quitte l'application ; une sauvegarde de l'image courante sera proposée
- Couleur
  - Négatif : calcule le négatif de l'image courante (chaque composante autre que le canal **a** passe de  $v$  à  $255 - v$ )
  - Niveau de gris : convertit l'image en niveau de gris
  - Noir et blanc
    - Seuil : convertit l'image en niveau de gris dans un premier temps puis transforme chaque pixel soit en noir soit en blanc suivant qu'il est au dessous ou au dessus d'un seuil donné par l'utilisateur (un seuil par défaut de 127 sera proposé : de 0 à 127, le pixel sera noir, de 128 à 255, il sera blanc)
    - Moyenne : convertit l'image en noir et blanc comme décrit ci-dessus avec comme seuil la moyenne des niveaux de gris
    - Médian : convertit l'image en noir et blanc comme décrit ci-dessus avec comme seuil la valeur médiane des niveaux de gris
- Transformation :

- Médian : convertit l'image en noir et blanc comme décrit ci-dessus avec comme seuil la valeur médiane des niveaux de gris
- Transformation :
  - Pivoter à gauche de  $90^\circ$  : pivote l'image courante de  $90^\circ$  vers la gauche (sens anti-horaire)
  - Pivoter à droite de  $90^\circ$  : pivote l'image courante de  $90^\circ$  vers la droite (sens horaire)
  - Retourner horizontalement : retourne l'image courante horizontalement
  - Retourner verticalement : retourne l'image courante verticalement
  - Redimensionner : redimensionne l'image courante suivant les nouvelles dimensions demandées par l'utilisateur (on se limitera dans un premier temps à des dimensions multiples ou diviseurs des dimensions de l'image - dans le cas contraire, on affichera à l'utilisateur un avertissement et l'image ne sera pas modifiée)
  - Découper : découpe l'image courante à partir d'une coordonnée et d'une nouvelle largeur et d'une nouvelle hauteur (devant rester dans les limites imposées par les dimensions de l'image courante - dans le cas contraire, on affichera à l'utilisateur un avertissement et l'image ne sera pas modifiée)

## Lancement avec options

Lorsque votre application est lancée avec options par :

```
./app.exe -i toto.ppm -RL -RR -o tata.png
```

les options seront exécutées les une à la suivante des autres. Les options disponibles sont les suivantes :

- `-i` (ou `-open` ou `-load`) + *nom\_de\_fichier* : charge en image courante l'image contenue dans le fichier de nom *nom\_de\_fichier*
- `-o` (ou `-save`) + *nom\_de\_fichier* : sauvegarde l'image courante dans le fichier de nom *nom\_de\_fichier* (le format sera déterminé par l'extension du nom de fichier - les sauvegarde des fichiers PPM, PGM et PBM seront en binaire)
- `-p` (ou `-print`) : affiche les informations concernant l'image courante (nom du fichier d'origine, hauteur et largeur en pixels) puis affiche le canal G de l'image en ASCII Art dans un format respectant les dimensions de l'image mais tenant sur un écran 80 x 60 caractères
- `-n` (ou `-negative`) : calcule le négatif de l'image courante (chaque composante autre que le canal **R** passe de *v* à **255 - v**)
- `-g` (ou `-grayscale`) : convertit l'image en niveau de gris
- `-bw` (ou `-black_white`) + *entier* : convertit l'image courante en noir et blanc avec *entier* pour seuil
- `-bwAvg` : convertit l'image courante en noir et blanc avec comme seuil la moyenne des niveaux de gris
- `-bwMed` : convertit l'image courante en noir et blanc avec comme seuil la valeur médiane des niveaux de gris
- `-RL` : pivote l'image courante de  $90^\circ$  vers la gauche (sens anti-horaire)
- `-RR` : pivote l'image courante de  $90^\circ$  vers la droite (sens horaire)
- `-RH` : retourne l'image courante horizontalement
- `-RV` : retourne l'image courante verticalement
- `-r` (ou `-resize`) + *entier1* + *entier2* : redimensionne l'image courante suivant les nouvelles dimensions *entier1* x *entier2*
- `-c` (ou `-crop`) + *entier1* + *entier2* + *entier3* + *entier4* : découpe l'image courante à partir du pixel de coordonnées (*entier1*,*entier2*) avec comme dimensions *entier3* x *entier4*

## Extensions

---

Vous pouvez apporter toute amélioration supplémentaire à votre projet. Les suggestions de nouvelles options et fonctionnalités sont les bienvenues.

Un regard particulier sera porté sur les projets proposant des entrées/sorties de formats supplémentaires (BMP et PNG sont à envisager). De plus amples informations vous seront remises ultérieurement.

## Organisation du code

---

On vous demande de programmer votre outil en respectant les règles de la compilation séparée vues en cours.

## Organisation du code

---

On vous demande de programmer votre outils en respectant les règles de la compilation séparée vues en cours, le tout piloté par un `Makefile`. Votre projet devra donc comporter **au moins** les fichiers suivants :

- `image.h`, `image.c` : bibliothèque de manipulation d'images décrite [TP 11](#)
- `menu.h`, `menu.c` : bibliothèque de création de menu pour la gestion d'interfaces textuelles décrite [TP 8](#)
- `args.h`, `args.c` : bibliothèque de manipulation des arguments entrés en ligne de commande décrite [TP 12](#)
- `main.c` : programmation de l'interface de l'application (menu et ligne de commande) et lancement des fonctionnalités du module `image` associées

Les fichiers supplémentaires concerneront les extensions que vous trouverez bon d'apporter. Il est cependant important que vous programmez les 3 modules demandés en respectant les consignes afin qu'ils puissent être testés séparément et automatiquement.

## Rendu

---

Le travail est à réaliser en **binôme**.

Votre projet sera à remettre sur EPREL dans le module *Programmation C* pour le **31/12/2015**. Il devra être rendu sous la forme d'une archive (`zip` ou `tar.gz`) contenant uniquement vos sources, votre `Makefile` et un fichier supplémentaire `README` vous permettant de préciser les noms et numéros étudiant de votre binôme et toutes informations supplémentaires que vous souhaiteriez porter à la connaissance du correcteur. Les projets constitués d'un simple fichier ne seront pas regardés.

Après décompression, un simple `make` en ligne de commande permettra de compiler votre outils. Votre projet **doit** évidemment compiler sans erreur ni *warning* et se montrer robuste aux tests. Une série d'exemples pouvant servir de tests vous seront proposés ultérieurement.

**Le non-respect des consignes entraînera la non-évaluation du projet.**

**Toute suspicion de copie ou de partage entre les binômes fera chuter votre note drastiquement.**