

Langage C++

- CIR2 -



TP4 : Manipulation des images PGM

Ayoub KARINE (ayoub.karine@isen-ouest.yncrea.fr)

Compétences travaillées dans ce TP :

- Encapsulation
- Constructeurs
- Destructeurs
- Traitement d'images
- Fichiers

Consignes :

- Tester dans le main, à chaque fois, chaque méthode développée
 - Utiliser le débogage
-

Principe des images PGM

Le format PGM (Portable GreyMAp) est un format de fichier permettant de stocker des données d'image. Ce type d'image contient :

- Ligne 1 : P2 (magic number indiquant le format d'image)
- Ligne 2 : largeur hauteur (2 valeurs entières indiquant la taille de l'image)

- Ligne 3 : valmax (1 valeur entière indiquant le niveau de gris maximal, souvent 255)
- Ligne 4 et suivantes : les valeurs entières sous forme de matrice représentant l'intensité en niveau de gris de chaque pixel de l'image variant entre 0 (noir) et 255 (blanc). Chaque ligne de l'image est représentée par une ligne d'entiers (séparés par des espaces) représentant l'intensité en niveaux de gris de chaque pixel de la ligne. En traitement d'image, le coin haut gauche est le point de coordonnées (0,0)

L'extension de ce type d'images est ".pgm"

Exemple des images PGM

Les données ci-dessous

```
P2
24 7
255
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 255 0 0 50 50 50 50 0 0 11 11 11 11 0 0 200 0 0 0 200 0 0
0 0 255 0 0 50 0 0 0 0 0 11 0 0 0 0 0 200 200 0 0 200 0 0
0 0 255 0 0 50 50 50 50 0 0 11 11 11 0 0 0 200 0 200 0 200 0 0
0 0 255 0 0 0 0 0 0 50 0 0 11 0 0 0 0 200 0 0 200 200 0 0
0 0 255 0 0 50 50 50 50 0 0 11 11 11 11 0 0 200 0 0 0 200 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

génère l'affichage suivant :



Pour plus d'information sur ce type d'image, visiter le lien suivant :

https://fr.wikipedia.org/wiki/Portable_pixmap#PGM

Création des images PGM

- 1) Créer une classe PGM comportant les attributs privés suivants :
 - a) width : un entier représentant la largeur de l'image
 - b) height : un entier représentant la hauteur de l'image
 - c) maxvalue : un entier représentant la valeur maximale de l'intensité en niveau de gris
 - d) data : un tableau 2d d'entiers qui contient juste les pixels de l'image sans les trois premières lignes de l'exemple ci-dessus
- 2) Ecrire les 4 setters de la classe PGM
- 3) Ecrire la méthode `int** initImage()` permettant d'allouer en mémoire un tableau d'entiers 2d dynamique. height et width sont respectivement la hauteur et la largeur de l'image
- 4) Ecrire la méthode `void deleteImage()` permettant de libérer l'espace alloué par le tableau dynamique `data`
- 5) Ecrire la méthode `int** createImage(int minpix, int maxpix)` permettant de remplir un tableau 2d par des valeurs entières aléatoires dans l'intervalle [minpix, maxpix]. Si ces deux valeurs ne sont pas spécifiées par l'utilisateur, elles sont initialisées à 0 et 255 respectivement. Pour générer les valeurs aléatoires, utiliser la fonction `rand()`. Exemple d'utilisation : <http://www.cplusplus.com/reference/cstdlib/rand/>
N'oublier pas d'utiliser la fonction `srand()` (voir tp2) pour ne pas avoir des valeurs aléatoires identiques
- 6) Ecrire un constructeur par défaut

- 7) Ecrire un deuxième constructeur qui prend comme arguments : hauteur, largeur, valeur minimum des pixel, valeur maximal des pixel. Pour ce faire, appeler que les setters
- 8) Ecrire un destructeur. Pour ce faire, appeler juste une méthode déjà programmée
- 9) Ecrire un constructeur de copie
- 10) Ecrire une méthode **void afficherImage()** permettant d'afficher la matrice des pixels (**data**)
- 11) Ecrire une méthode **void writeFile(char* filename)** permettant de stocker (voir exemple ci-dessus)

- a) P2 dans la ligne 1
- b) La valeur de la largeur suivi d'un espace suivi de la valeur de la hauteur dans la ligne 2
- c) La valeur maximale de niveau de gris dans la ligne 3
- d) La matrice data dans le reste des lignes avec des espaces entre les valeurs d'intensité

dans le fichier "filename". Pour manipuler les fichiers en c++, utiliser la classe ofstream : <http://www.cplusplus.com/doc/tutorial/files/>

Ce fichier doit avoir l'extension ".pgm".

Une fois que le fichier est créé, ouvrir le dans le logiciel IrfanView (à télécharger depuis le lien suivant : <https://www.irfanview.com/>) pour visualiser le résultat.

- 12) Ecrire une méthode permettant de savoir le nombre d'images PGM créées par l'utilisateur.
 - a) Utiliser le débogage pour savoir le nombres d'images PGM créées à la fin du programme

- b) Proposer une solution pour que le nombre de ces images PGM soit exactement 0 à la fin du programme