

# **Rapport de Projet d'Algorithmique**

## **Description des Classes:**

Program: Contient le Main et permet de lancer le programme

Pixel: Chaque pixel est composé de 3 bytes R, G et B pour les couleurs rouge, vert et bleu. On a donc pour chaque Pixel 3 valeurs situées entre 0 et 255 permettant de savoir de quelle couleur est celui-ci.

MyImage: Contient la plupart des méthodes liées au traitement d'image. Chaque image est définie comme étant une matrice de Pixel.

Point: Complexe utilisé pour la suite pour définir l'ensemble de mandelbrot de la fractale

## **Description des Méthodes:**

From Image To File: Permet de convertir une image en un fichier .bmp

Convertir Endian To Int: Permet de convertir les Endian(byte) en Int

Convertir Int To Endian: Permet de convertir les Int en Endian(byte)

Contours: Permet de mettre en valeur les contours des formes présentes sur l'image. Pour ce faire, chaque Pixel de l'image (sauf les bords) est égal à la multiplication de son voisinage (le pixel en question et les 8 qui l'entourent) et d'une matrice de convolution que l'on appelle un filtre. Ici la matrice est  $\begin{Bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{Bmatrix}$

Flou: Permet de flouter l'image. Pour ce faire, chaque Pixel de l'image (sauf les bords) est égal à la multiplication de son voisinage (le pixel en question et les 8 qui l'entourent) et d'une matrice de convolution que l'on appelle un filtre. Ici la matrice est  $\begin{Bmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{Bmatrix}$

Repoussage: Permet de renforcer les bords de l'image. Pour ce faire, chaque Pixel de l'image (sauf les bords) est égal à la multiplication de son voisinage (le pixel en question et les 8 qui l'entourent) et d'une matrice de convolution que l'on appelle un filtre. Ici, le résultat est l'addition de l'image de base et des contours, c'est pourquoi la matrice est  $\begin{Bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{Bmatrix}$

Noir et Blanc: Met les valeurs R,G et/ou B des pixels de l'image à 255 (blanc) si elle dépasse 128 et à 0 (Noir) si elle est en dessous ou égale à 128, permettant d'obtenir une image en noir et blanc.

Niveau de gris: Fait la moyenne des valeurs R,G,B des pixels de l'image afin d'obtenir une image avec des nuances de gris.

Agrandissement: Permet d'agrandir l'image selon un coefficient demandé. Ainsi, si on veut une image 2 fois plus grande, on créera une nouvelle image avec 2 fois plus de Pixel par ligne et par colonne: chaque Pixel de l'image de base en devient donc 4 dans celle agrandie 2 fois

Quart supérieur gauche: Permet d'afficher le quart supérieur gauche de l'image en créant une nouvelle image dont la hauteur et la largeur sont la moitié de celles de l'image d'origine.

Rotation: Permet de tourner l'image selon un angle que l'utilisateur choisit. La position de chaque pixels de l'image tournée est calculée grâce aux fonctions trigonométriques. Cependant, en faisant cela, certains pixels de l'image inclinée restent noirs, cela est dû aux arrondis dans le calcul de la position des pixels: certains pixels n'ont pas pu être placés. Pour pallier ce problème, on fait le chemin inverse: on regarde à quels pixels de l'image d'origine correspondent les pixels noirs de l'image inclinée puis on complète les trous en fonction de cela.

IMPORTANT: la hauteur et la largeur de l'image doivent être divisible par 4

Fractale: Donne le choix d'une forme pour effectuer une fractale.

Mandelbrot: Met en évidence les points sur l'image où la suite  $Z_{n+1}=Z_n^2+c$ ;  $Z_0=0$  avec  $Z$