**Compte-rendu de TD2**

**Chapitre 1**

Nous avons passé beaucoup de temps sur la recherche documentaire et sur la manière avec laquelle nous allions aborder ce projet scientifique informatique. Ainsi, nous avons créé plusieurs classes :

* La classe MyImage qui va prendre un fichier image BMP ou un tableur csv et le convertir en une instance de classe
* La classe Pixel qui va nous servir à créer l’image

Nous avons beaucoup réfléchi sur comment est composée l’image (type, largeur, hauteur…) et sur comment faire le constructeur de notre classe MyImage. Nous avons finalement compris qu’il fallait lire le fichier puis convertir les valeurs des octets en longueur, largeur, type d’image …

Nous avons réussi à récupérer le type de fichier.

**Chapitre 2**

Le constructeur de la classe MyImage est finalisé : il donne la largeur, la hauteur, la taille, le nombre de bits et le type de notre image. De plus grâce à la classe Pixel, nous avons crée dans la classe MyImage une matrice de Pixel qui va définir notre image. A partir de l’octet 54, et tous les 3 octets, chaque octet est associé à une couleur (Rouge, Vert, Bleu) pour former un pixel que l’on insère dans la matrice.

Nous avons écrit sur feuille les méthodes pour faire le traitement d’image :

* Passer d’une photo en couleur à une photo en nuances de gris :
* Faire la moyenne des valeurs des couleurs rouge, vert et bleu (R+V+B)/3 pour chaque pixel de l’image
* Agrandir/Rétrécir l’image
* Faire une méthode qui prend en paramètres l’image et un pourcentage rentré par l’utilisateur
* Multiplier la largeur et la hauteur de l’image par ce pourcentage et définir la nouvelle image avec ces dimensions
* Définir les coordonnées du point de départ de l’agrandissement
* Rotation
* Une rotation à 180° est facile : les pixels vont prendre la place de leur symétrique par rapport au centre de l’image (les pixels en haut à gauche vont aller en bas à droite)
* Les rotations à 90° et 270° sont plus compliquées car comme les images sont rectangulaire, la hauteur (m) et la largeur (n) de l’image de départ sont inversées par rapport aux dimensions de l’image d’arrivée. Ensuite on prend le pixel en haut à droite et on le met en haut à gauche etc : image[0,0]=image2[n,0]