

Rapport du projet 2 : ULBMP

LI Min-Tchun

12/05/2024

Contents

1	Introduction	2
2	Initialisation	3
3	ULBMP 2.0	4
4	ULBMP 3.0	5
5	ULBMP 4.0	6
6	Conclusion	7

Chapter 1

Introduction

Pour le deuxième projet du cours INFO-F-106, nous avons dû implémenter un compresseur d'image appelé ULBMP, basé sur le format BMP, développé par Microsoft dans les années 80.

Afin de pouvoir venir à bout de ce projet, il est demandé d'importer le module *Pyside6* afin de pouvoir créer une interface graphique pour l'utilisateur. Il est donc important de lire la documentation de ce module afin de pouvoir l'utiliser correctement.

De plus, il est fondamental de maîtriser la manipulation de fichier, de bits (notamment le shifting et le masking), ainsi que la compréhension de la représentation binaire et hexadécimale.

En effet, comme nous devons utiliser des pixels, il est important de comprendre que ces derniers sont composés de 3 valeurs : Rouge, Vert, Bleu (RVB ou RGB en anglais). Chacune de ces valeurs est codée sur 8 bits, ce qui signifie que chaque valeur peut prendre $2^8 = 256$ valeurs différentes.

Il était donc utile d'utiliser le programme **hexdump**, à exécuter dans le terminal afin de pouvoir visualiser les données hexadécimales.

Cependant, une question se pose : comment peut-on représenter une image en format numérique ou bien comment peut-on compresser une image ? Afin de résoudre ce problème, on va procéder comme suit, on va créer une classe *Decoder* et *Encoder*.

Le premier permettra, à partir d'un chemin d'accès du fichier, de lire chaque byte et de retourner une *image* (on définira plus tard plus précisément ce que c'est).

Le deuxième permettra, à partir d'une *image*, d'écrire dans un fichier les bytes correspondants aux valeurs RGB de chaque pixel.

Il est important de noter qu'il y a plusieurs versions du compresseur d'image. Cela signifie que le header (ce qui se trouve à chaque début de fichier pour reconnaître le format du fichier) doit être adapté à chaque version.

Chapter 2

Initialisation

Comme cité dans l'introduction, les classes *Decoder* et *Encoder* nécessitent une image, définie par la classe *Image*. Cette classe contient les attributs suivants : la largeur, la hauteur de l'image et les pixels qui la constitue. Ces derniers sont également définis à partir d'une classe, *Pixel* qui contient les attributs suivants : la valeur de rouge, de vert et de bleu.

Bien évidemment, certaines fonctions spécifiques à la construction de classes doivent être créées pour faciliter la manipulation des pixels et des images. Par exemple, la fonction `getitemp` permet de récupérer un pixel à partir des coordonnées (x, y) .

Chapter 3

ULBMP 2.0

Ceci est le chapitre 2.

Chapter 4

ULBMP 3.0

Ceci est le chapitre 3.

Chapter 5

ULBMP 4.0

Ceci est le chapitre 4.

Chapter 6

Conclusion

Ceci est une conclusion.