4 MAI 2021

Rapport Projet Scientifique Informatique

Réalisé par :
Antoine RAULIN
Hortense DU MAROUSSEM

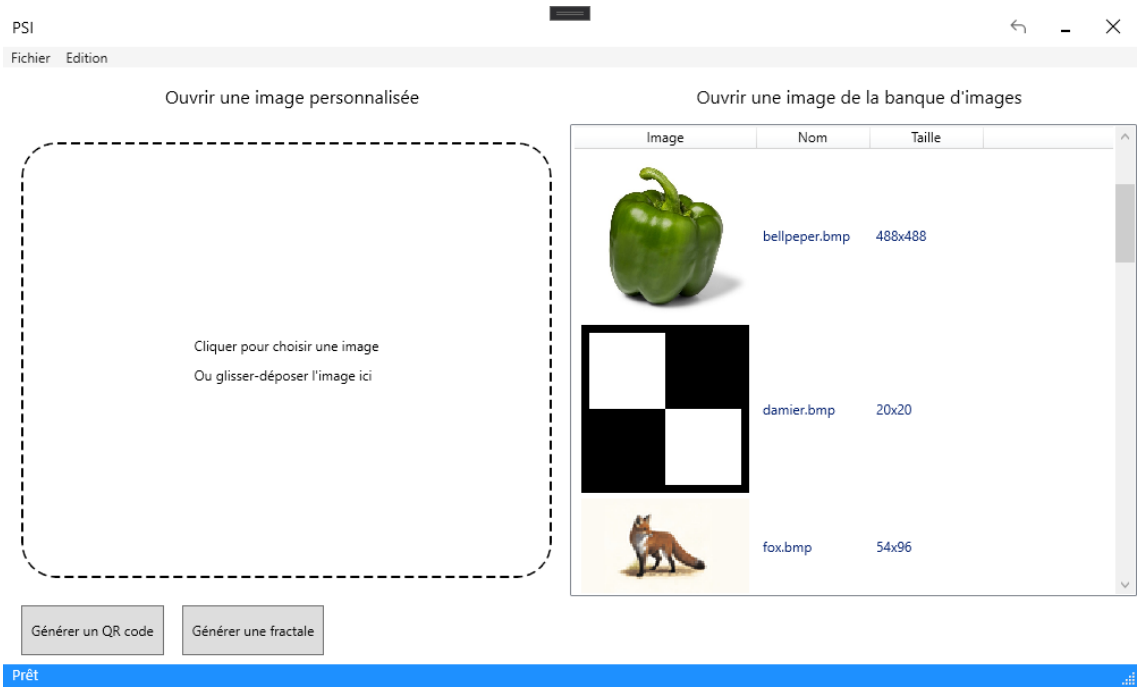

Table des matières

Explication des structures de données

Nous avons choisi de créer une classe MyImage. Cette classe, définie comme public, nous permet de créer un objet pour chaque image. Elle sera caractérisée par le nom de l'image, sa hauteur et sa largeur.

A chaque image sera ensuite assigné la taille (size), l'offset, la profondeur (depth), la taille réelle (realImageSize) ainsi qu'une matrice Bgr. Cette matrice est une matrice de pixels, dont chaque élément sera un objet de la classe BGR. Cette matrice sera la représentation de notre image.

Notre deuxième élément est le struct BGR. Elle est caractérisée par 3 int, b, g et r, qui correspondent aux valeurs du rouge, du vert et du bleu de notre pixel.Ils sont donc tous compris entre 0 et 255. Nous avons défini cette classe comme une public readonly struct car nous n'avons pas besoin d'utiliser de méthodes (de void par exemple) dans cette classe. De plus, une classe est passée par référence alors que le struct est passé par une copie, ce qui peut éviter certains problèmes. Notre struct s'appelle BGR et non pas RGB parce que en format BMP, les pixels sont codés de cette manière.

Nous avons également réalisé plusieurs tests unitaires afin de vérifier le bon fonctionnement de nos méthodes. Nous en avons réalisé un qui vérifie le bon fonctionnement de la fonction nuances de gris (Grayscale). On vérifie si l'image qui sort de cette fonction fait la même taille que l'image que l'on est supposé obtenir. On vérifie ensuite bit par bit si elle correspond, et on en conclut si notre image fonctionne ou non. Si un des bits ne correspond pas, c'est que notre fonction n'a pas fonctionné.

Par ailleurs nous avons la classe ReedSolomon, utilisée dans notre QR code et qui nous a été fournie par l'école.

Explication de l’innovation

Notre innovation consiste à avoir utilisé le WPF pour avoir une interface graphique plus jolie et intuitive que la console. Nous avons donc mis en place des fonctions comme le simple clic de la souris, le drag and drop etc. Toutes les fonctions que nous avons réalisées (Mirror, fractales, Noir et blanc etc) sont disponibles sur le menu généré par un simple clic.

Nous avons également mis en place la possibilité d’un "ctrl z", qui nous permet d’annuler une modification effectuée afin de retrouver l’image précédente.

Enfin, notre dernière innovation porte sur la fonction de redimensionnement sur laquelle nous avons ajouté la possibilité d’activer l’antialiasing qui détermine la couleur des pixels intermédiaire lors d’un agrandissement.

Problèmes rencontrés

Nous avons rencontré quelques problèmes avec la fonction "renforcement des bords". En effet, elle nous fait apparaitre quelques pixels de couleur complètement inadaptée à des endroits randoms de l'image, sans que nous n'ayons réussi à comprendre pourquoi.

Par ailleurs nous avons eu plus de difficultés concernant la partie du QR codes, où nous avons du chercher beaucoup de documents pour compléter ceux fournis dans le cours. Le décodage des QR codes, en particulier, nous a paru très compliqué.

Auto critique

Nous aurions pu améliorer notre système de ctrl z. En effet celui-ci n’a "que" un historique de 5 modifications.
Un autre facteur que nous pourrions ajouter serait la possibilité de zoomer sur notre WPF afin de mieux observer des images telles que les fractales.

Etude de compression d'images

Le but de la compression d'image est de réduire la redondance des données de l'image afin que elle soit moins lourde, tout en essayant de perdre le moins de qualité possible.. Cela lui permet d'être téléchargée ou envoyée plus rapidement.

Il existe plusieurs manières de compresser une image sans pertes :

- **codage de répétitions** : Cet algorithme est surtout utilisé pour les documents en noir et blanc. Il consiste à "compter" le nombre d'octets noirs ou blancs qui se suivent.
  Une ligne de code BBBBBWWWWWWBBBWB sera donc codée 5B6W3B1W1B. Ce type de compression est notamment utilisé pour les fichier BMP en 1,4 ou 8 bits/pixels.
- **codage entropique** : Ce mode de compression utilise les codes de Huffman ou le codage arithmétique. Il consiste à assigner à une partie de l'image un code à longueur variable. On va par exemple assigner un code très court aux mots les plus fréquents. Il utilise pour cela les statistiques. L'information à coder sera donc représentée par une variable aléatoire précise.
- **algorithmes à dictionnaires adaptables** : Un des algorithmes les plus connus de ce mode de compression est LZW. Cette technique associe aux éléments du texte une table de traduction des chaînes de caractères. Le compresseur va lire le texte, et ajouter toutes les chaînes de plus de 2 caractères à sa table de traduction comme une concaténation de chaînes déjà rencontrées, avec le code correspondant. Chaque fois qu'une chaîne déjà rencontrée est lue, la chaîne la plus longue déjà rencontrée est déterminée, et le code correspondant à cette chaîne avec le caractère concaténé est enregistré dans la table

Les compressions avec pertes, elles, sont :

- **Réduction de l'espace des couleurs** : Cette méthode va donc représenter chaque couleur dans un système colorimétrique. Chaque couleur sera donc caractérisée par un point dans un espace à trois dimensions.
- **sous échantillonage de la chrominance** :Cette méthode réduit le volume des images numériques en diminuant le nombre d'échantillons à traiter. L'oeil humain est moins sensible aux couleurs que à la luminosité, c'est pour cela que cette méthode va donc traiter moins d'informations, au détriment de la chrominance.
- **codage par transformation** : Une des plus connues est la transformée en cosinus discrète.

- **compression fractale** : Elle repose sur la détection de la récurrence des motifs, et tend à éliminer la redondance d'informations dans l'image. Il y a donc une grosse perte des données suite à la compression.

Sources

https://fr.wikipedia.org/wiki/Compression_d%27image
http://user.engineering.uiowa.edu/~dip/lecture/DataCompression.html
https://fr.wikipedia.org/wiki/Run-length_encoding
https://fr.wikipedia.org/wiki/Codage_entropique
https://fr.wikipedia.org/wiki/LZW
https://fr.wikipedia.org/wiki/Espace_colorim
https://fr.wikipedia.org/wiki/Sous-%C3%A9chantillonnage_de_la_chrominance https://fr.ryte.com/magazine/compression-images-reduction-temps-chargement