Aline Ellul – Projet Info – 2019 – 2ème année

Projet Informatique

Langage C#

Prérequis Etudiants

Tous les concepts vus pendant les 3 premiers semestres sont à connaître pour la résolution de ce projet informatique. Votre solution devra faire l'objet d'une application de programmation objet.

Préambule

Vous avez 5 sessions de 3 heures et autant de travail personnel.

Seule la session 1 est organisée sous forme de TD, vous organisez les suivantes en fonction de l'état d'avancement de votre projet.

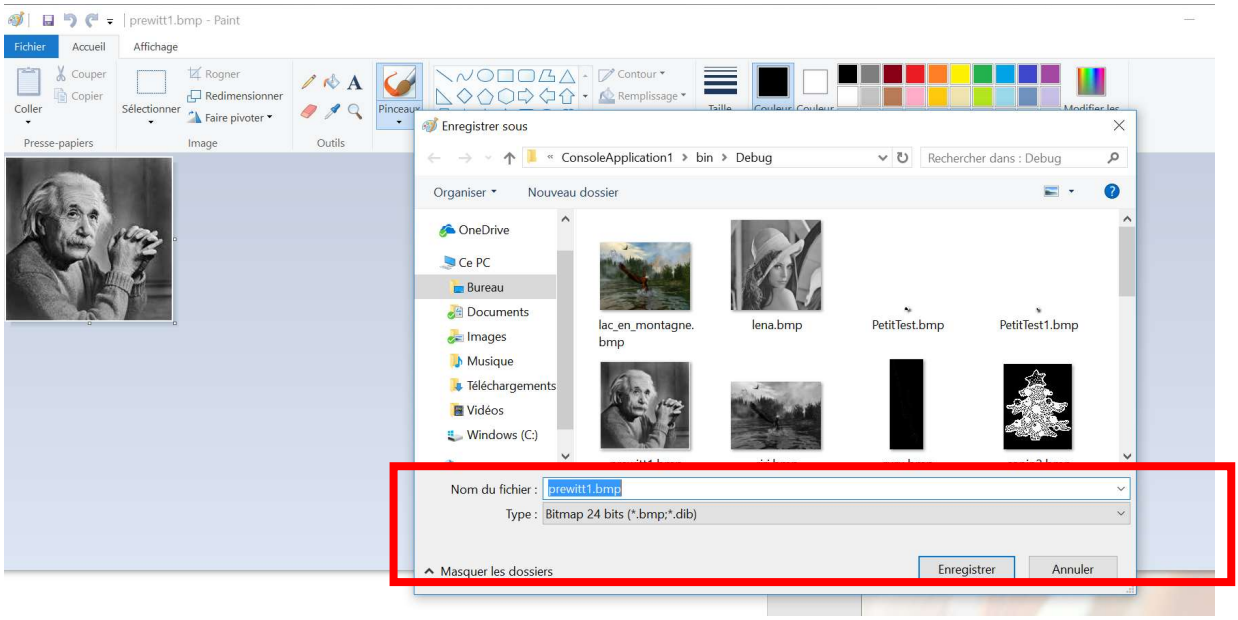
Le projet informatique a pour objet de mettre en application les concepts vus en C# de façon autonome sur le sujet du traitement des images.

Pour cela, vous avez besoin d'un outil de visualisation d'images

- Windows Paint est simple et suffisant
- GIMP est un logiciel libre plus complexe et plus complet

https://www.gimp.org/

Aline Ellul – Projet Info – 2019 – 2ème année



Vous n'utiliserez que des images en format Bitmap 24bits.

L'idée est de créer un outil qui lit une image dans un format donné (bitmap ou csv), traite cette image (agrandir, rétrécir …) et sauvegarde l'image dans un fichier de sortie différent de celui donné en entrée (toujours format Bitmap ou csv). On ne veut pas perdre les fichiers originaux.

Puisque les formats en entrée et sortie peuvent être différents, il est important de bien réfléchir au format pivot en mémoire qui sera utilisé.

- Lire et écrire une image (à partir d'un format .bmp ou .csv) TD1
- Traiter une image : TD2
  - Passage d'une photo couleur à une photo en nuances de gris et en noir et blanc
  - Agrandir/Rétrécir une image
  - Rotation (90,180 ou 270°)
  - Effet Miroir

- Appliquer un filtre (matrice de convolution) TD3
  - Détection de contour (https://docs.gimp.org/fr/plug-in-convmatrix.html)
  - Renforcement des bords (https://docs.gimp.org/fr/plug-in-convmatrix.html)
  - Flou (https://docs.gimp.org/fr/plug-in-convmatrix.html)
  - Repoussage (https://docs.gimp.org/fr/plug-in-convmatrix.html)

- Créer une image nouvelle (à partir de rien) TD4
  - Créer une fractale

  https://fr.wikipedia.org/wiki/Ensemble_de_Mandelbrot

- Créer un histogramme se rapportant à une image https://fr.wikipedia.org/wiki/Histogramme_(imagerie_num%C3%A9rique)

- Coder et Décoder une image dans une image TD5

  http://www.bibmath.net/crypto/index.php?action=affiche&quoi=stegano/cacheimage

- Une création (autre que ce qui est proposé ci-dessus bien évidemment)

Avant le premier TD, consulter quelques liens pour vous familiariser avec le monde du traitement d'images et son vocabulaire : format, bitmap, pixel

https://en.wikipedia.org/wiki/Bitmap

https://en.wikipedia.org/wiki/Pixel

https://fr.wikipedia.org/wiki/Rouge_vert_bleu

Lorsque vous allez créer votre image de sortie, vous pouvez utiliser l'instruction suivante pour faciliter la visualisation du résultat (si perroquet.bmp est une image)

```
using System.IO;
using System.Diagnostics;

namespace LectureImage
{
    class Program
    {
        static void Main(string[] args)
        {
            Process.Start("perroquet.bmp");
            Console.ReadLine();
        }
    }
}
```

TD – 1

Vous allez travailler sur des images de type Bitmap non compressé, 24 bits (un octet par couleur RGB). L'idée de ce TD est de réécrire et comprendre en partie la classe Bitmap fournie par C#.

```
public sealed class Bitmap : Image
{
    public Bitmap(string filename);
    public Bitmap(Stream stream);
    public Bitmap(Image original);
    public Bitmap(string filename, bool useIcm);
    public Bitmap(Type type, string resource);
    public Bitmap(Stream stream, bool useIcm);
    public Bitmap(int width, int height);
    public Bitmap(Image original, Size newSize);
    public Bitmap(int width, int height, PixelFormat format);
    public Bitmap(int width, int height, Graphics g);
    public Bitmap(Image original, int width, int height);
    public Bitmap(int width, int height, int stride, PixelFormat format, IntPtr scan0);

    public static Bitmap FromHicon(IntPtr hicon);
    public static Bitmap FromResource(IntPtr hinstance, string bitmapName);
    public Bitmap Clone(RectangleF rect, PixelFormat format);
    public Bitmap Clone(Rectangle rect, PixelFormat format);
    public IntPtr GetHbitmap();
    public IntPtr GetHbitmap(Color background);
    public IntPtr GetHicon();
    public Color GetPixel(int x, int y);
    public BitmapData LockBits(Rectangle rect, ImageLockMode flags, PixelFormat format);
    public BitmapData LockBits(Rectangle rect, ImageLockMode flags, PixelFormat format, BitmapData bitm
    public void MakeTransparent();
    public void MakeTransparent(Color transparentColor);
    public void SetPixel(int x, int y, Color color);
    public void SetResolution(float xDpi, float yDpi);
    public void UnlockBits(BitmapData bitmapdata);
}
```

Sur le lien suivant, vous découvrirez la structure d'un fichier Bitmap

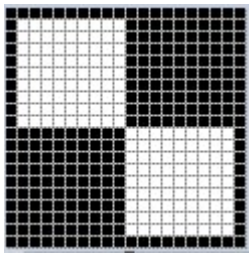
https://fr.wikipedia.org/wiki/Windows_bitmap

https://en.wikipedia.org/wiki/BMP_file_format

http://www.proftnj.com/RGB3.htm

1 - Un petit exemple

Soit l'image suivante Test.bmp (sur quelques pixels pour simplifier la compréhension), chaque carré est un pixel grossi au maximum. Il s'agit d'une image de 20 pixels sur 20 pixels



Conformément aux informations

```csharp
Console.WriteLine("HEADER"); Console.WriteLine(); Console.WriteLine();
for (int i = 0; i < 14; i++)
    Console.Write(myfile[i] + " ");
Console.WriteLine("\n\nHEADER INFO\n\n");
for (int i = 14; i < 54; i++)
    Console.Write(myfile[i] + " ");
Console.WriteLine("\n\nIMAGE\n");
for (int i = 54; i < myfile.Length; i++)
{
    Console.Write(myfile[i] + "\t");

}
```

Le résultat de ce code affiche les informations suivantes :

```
HEADER


66 77 230 4 0 0 0 0 0 0 54 0 0 0

HEADER INFO


40 0 0 0 20 0 0 0 20 0 0 0 1 0 24 0 0 0 0 0 176 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

IMAGE

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	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	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	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	0	0	0	0	0	0	0	0	0
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	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	0	0	0	0	0	0	0	0	0
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	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	0	0	0	0	0	0	0	0	0
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	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	0	0	0	0	0	0	0	0	0
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	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	0	0	0	0	0	0	0	0	0
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	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	0	0	0	0	0	0	0	0	0
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	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	0	0	0	0	0	0	0	0	0
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	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	0	0	0	0	0	0	0	0	0
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
```

Chaque ligne de l'image doit comporter un nombre total d'octets qui est un multiple de 4; si ce n'est pas le cas, la ligne doit être complétée par des 0 de telle manière à respecter ce critère. Vous ne traiterez que des images multiples de 4.

2 – Explication

Les 14 premiers octets décrivent le header de votre fichier (entête)

- 66 et 77 correspondent au code ascii des lettres B et M pour Bitmap
- 230 4 0 0 indique la taille du fichier en octets sur 4 octets à traduire en base décimale
- etc …

Vous noterez que toutes les tailles sont données dans le format « little endian »

https://fr.wikipedia.org/wiki/Endianness

Il faudra donc convertir ces tailles en entier en tenant compte de ce mode de formatage.

Les 50 autres octets suivants décrivent des informations liées à l'image.

- taille du header d'info (40 0 0 0) sur 4 octets
- largeur en pixels sur 4 octets (20 0 0 0)
- hauteur en pixel sur 4 octets (20 0 0 0)
- …
- Nombre d'octets par couleur (24 0) sur 2 octets
- ….

L'image démarre au 54ème octet. Le code RGB pour un pixel noir est égal à 0 0 0, le code RGB pour un pixel blanc est égal à 255 255 255

L'objectif du premier TD est donc de lire une image et de convertir cette image (fichier binaire) en une instance de classe MyImage que vous devez définir. Cette classe doit contenir les informations générales sur l'image et l'image par elle-même. Vous ferez en sorte que les données ne soient jamais dupliquées (il est hors de question d'avoir comme attributs une matrice de bytes et une matrice de pixels, c'est-à-dire la même donnée sous 2 formats différents)

Nous retiendrons les informations suivantes :

- type d'image (BM par exemple),
- taille du fichier (int), taille Offset (int),
- largeur et hauteur de l'image (int)
- nombre de bits par couleur(int)

- l'image par elle-même sur laquelle vous ferez les traitements proposés ensuite. (matrice de RGB)

Par ailleurs, veillez à concevoir éventuellement d'autres classes qui simplifieront la lisibilité et la sémantique du code.

Pour vous aider, vous créerez au moins pour la classe MyImage les constructeurs et méthodes suivantes :

- public MyImage(string myfile) lit un fichier (.csv ou .bmp) et le transforme en instance de la classe Image
- public void From_Image_To_File(string file) prend une instance de MyImage et la transforme en fichier binaire respectant la structure du fichier .bmp ou .csv
- public int Convertir_Endian_To_Int(byte[] tab …) convertit une séquence d'octet au format little endian en entier
- public byte[] Convertir_Int_To_Endian(int val …) convertit un entier en séquence d'octets au format little endian

Attention

Si toutefois, la lecture ou l'écriture binaire posait un problème au-delà de la session 2, on vous propose d'utiliser la classe Bitmap fournie par C#.

Vous serez alors noter en conséquence mais vous ne serez pas bloqués pour la suite du projet.

Pour cela, vous n'oublierez pas d'insérer `using System.Drawing;` dans votre fichier .cs et ajouter la référence Drawing comme ci-dessous

```
Bitmap bmpImage = new Bitmap("test3.bmp");
Console.WriteLine("width: " + bmpImage.Width + " height: " + bmpImage.Height);
for (int ii = 0; ii < bmpImage.Height; ii++)
    for (int j = 0; j < bmpImage.Width; j++)
    {
        Color c = bmpImage.GetPixel(j, ii);
        #region
        bmpImage.SetPixel(j, ii, Color.FromArgb(c.A, val, val, val));

    }
bmpImage.Save("testbitmap.bmp");
```

TD – 2 …. TD-5

Pour les séances suivantes, c'est à vous de gérer votre temps.

Aline Ellul – Projet Info – 2019 – 2ème année

Le chargé de TD est là pour répondre aux questions. Préparez vos séances pour qu'elles soient profitables, pour cela vous définirez vos objectifs d'une session à l'autre.

Pour vous aider à gérer votre projet dans le temps, nous vous demandons de venir à chaque séance avec un ensemble de questions. A l'issue de chaque session, vous déposerez sur Moodle votre dossier. Ainsi le chargé de TD sera capable de juger votre travail en continu jusqu'au 17 avril.

Vous déposerez votre solution C# sur Devinci-Online avec un rapport (4 pages maximum) dans lequel vous expliquerez vos structures de données et donnerez des détails de votre partie innovation.

Vous joindrez également un fichier de tests (C#) qui nous prouvera que chaque fonction de base a bien été développée et testée avant d'être associée à d'autres fonctions.

Enfin votre solution sera commentée de telle manière à générer le fichier XML résultant.

(/// avant chaque nom de fonction)

Encore une fois, les noms des fonctions et des variables doivent être lisibles pour faciliter la lecture du code.

Enfin un code qui ne compile pas ou qui « plante » immédiatement mérite 0/20

Les liens utiles :

Matrice de convolution :

http://xphilipp.developpez.com/articles/filtres

https://docs.gimp.org/fr/plug-in-convmatrix.html

https://en.wikipedia.org/wiki/Kernel_(image_processing)

http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig_ImageConvolution.pdf

http://lodev.org/cgtutor/filtering.html

http://micro.magnet.fsu.edu/primer/java/digitalimaging/processing/kernelmaskoperation/

http://xmcvs.free.fr/astroart/Chapitre4.pdf