

Projet Scientifique Informatique

Table des matières

Projet Scientifique Informatique	2
Prérequis Etudiants	4
Organisation	4
Outils	4
Livrables.....	5
Le jour précédant votre dernier TD (TD8 d'1h30).....	5
Le jour du dernier TD.....	6
Programme.....	6
Préambule	7
Bibliographie Générale.....	7
1-Format Bitmap et Structure d'une image Bitmap.....	9
1-1 Utilisation de la classe Bitmap C#	9
1-2 Retour aux sources : une image est une succession de bits	10
TD1-2 Mise en place du format Pivot.....	12
TD3 -TD5 Traitement d'images.....	13
TD 6-7 Compression : QuadTree	14

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 en programmation objet.**

Organisation

2 CMOs sont programmés :

- 23/01 : Présentation du projet jusqu'au -> TD4
- 10/03 : Présentation Windows Form & TD5→TD8

Vous avez 8 sessions de TD de 1h30 heures et autant de travail personnel minimum. Vous pouvez réaliser votre projet en binôme ou en trinôme. Le travail en monôme n'est possible qu'exceptionnellement et à l'appréciation de votre chargé de TD.

Outils

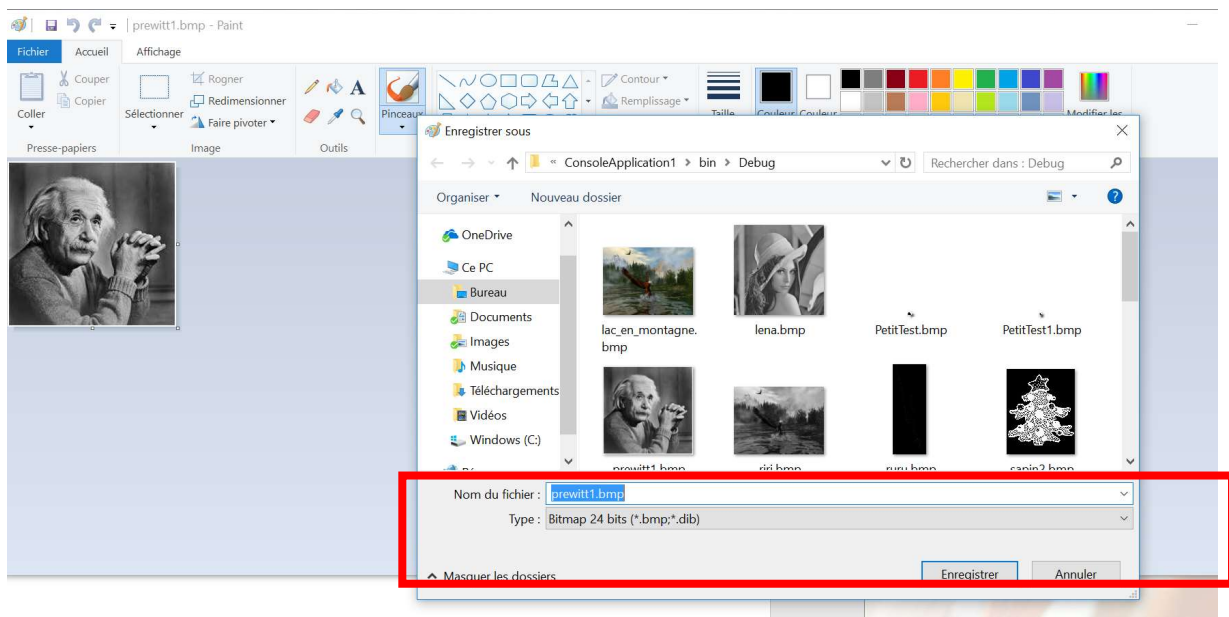
Pour cela, vous avez besoin d'un outil de visualisation d'images

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

<https://www.gimp.org/>

<https://docs.gimp.org/2.10/fr/> (documentation qui peut vous donner des idées)

Le projet repose sur des images de type **Bitmap 24 bits**. Vous aurez donc soin au moment de créer vos images de les sauvegarder sous ce format comme vous le montre l'image ci-dessous avec Windows Paint.



Livrables

Le chargé de TD est là pour répondre aux questions. Préparez vos séances pour qu'elles soient profitables.

Au début et à l'issue de chaque session, vous déposerez sur DVO votre dossier.

Votre chargé de TD évaluera votre travail en continu, cette note d'évaluation contribuera à hauteur de 20% de votre note finale.

Le jour précédant votre dernier TD (TD8 d'1h30)

Vous déposerez votre solution C# sur DVO avec un **rapport** (2 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 unitaires (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

Le jour du dernier TD

Votre chargé de TD vous confiera la tâche de corriger 4 projets de vos pairs à raison de 15 à 20mn par projet selon un barème qui vous sera communiqué. Cette correction se doit d'être objective en effet, une sur-note évidente pourrait donner lieu à un malus pour l'équipe correctrice.

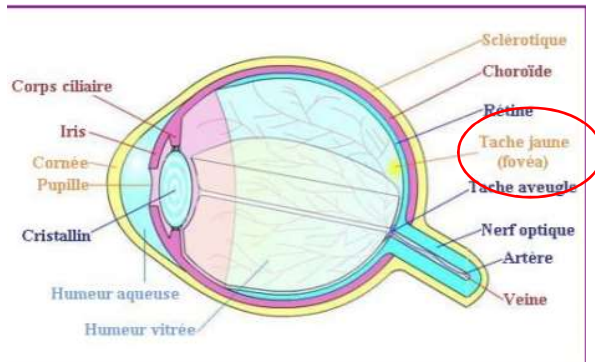
Programme

- **Lire et écrire une image (à partir d'un format .bmp)** TD1-2
 - lit une image dans un format donné (Bitmap),
 - traitement simple d'une image (Passage d'une photo couleur à une photo en nuances de gris et en noir et blanc)
 - sauvegarde l'image dans un fichier de sortie différent de celui donné en entrée (toujours format Bitmap)

On ne veut pas modifier les fichiers originaux.
Puisque les formats en entrée et sortie pourraient être différents, il est important de bien réfléchir au **format pivot** en mémoire qui sera utilisé.
- **Traiter une image :** TD3
 - Rotation (avec un angle quelconque)
 - Agrandissement (avec un coefficient quelconque)
- **Appliquer un filtre générique (matrice de convolution)** TD4
 - Détection de contour
 - Renforcement des bords
 - Flou
 - Repoussage
- **Créer ou extraire une image nouvelle** TD5
 - Créer une image décrivant une fractale
 - Coder et Décoder une image dans une image (stéganographie)
- **Implémentation des algorithmes de compression QuadTree ou Huffman** TD6-TD7
- **Innovation** Hors TD

Préambule

Perception par l'œil des images



La fovea est la région de l'œil où l'image est la plus précise. L'œil possède 2 types de photo-récepteurs :

- Les cônes responsables des dimensions chromatiques grâce à des pigments absorbant le **bleu, le rouge ou le vert**
- Les bâtonnets responsables de la vision nocturne

Nous allons travailler sur des images numériques avec une représentation matricielle. Chaque point de l'image est un pixel c'est-à-dire une combinaison des couleurs Rouge, Verte et Bleue. (Contrairement aux images vectorielles représentées par des fonctions mathématiques)

De nombreux formats d'images numériques ont vu le jour, nous allons nous focaliser sur le format Bitmap 24 bits, format propriétaire Microsoft

Bibliographie Générale

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

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

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

Sur les liens suivants, vous découvrirez la structure d'un fichier Bitmap

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

[Format BMP - wxFrantz's Concept \(free.fr\)](https://www.wxfrantz.com/fr/concept/)

Convertisseurs Hexadecimal / RGB

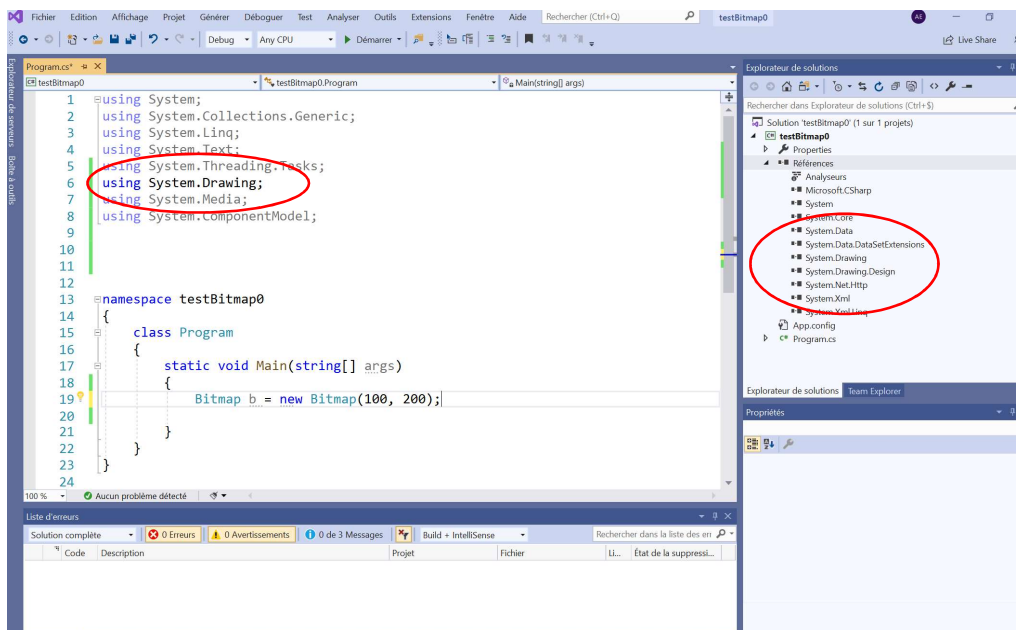
<http://www.proftnj.com/RGB3.htm>

1-Format Bitmap et Structure d'une image Bitmap

1-1 Utilisation de la classe Bitmap C#

La classe Bitmap existe en C#, vous pouvez l'utiliser dans un premier temps pour vous familiariser et comprendre ce que l'on attend de vous, puisque le but de ce projet est effectivement de ré-écrire cette classe

Après création de solution Console, vous ajoutez les références Drawing et Drawing Reference



Et ensuite vous allez utiliser la classe Bitmap et ses méthodes

```

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);
}
    
```

Problème Scientifique Informatique A2 - 2023

Vous faites quelques tests simples comme suit :

```
Bitmap b = new Bitmap("./Images/lena.bmp");
b.RotateFlip(RotateFlipType.Rotate180FlipX);
b.Save("./Images/lenasortie1.bmp");
```

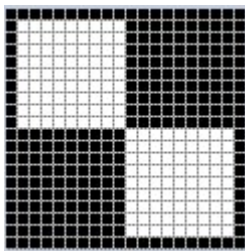
Lorsque vous allez créer votre image de sortie, vous pouvez utiliser l'instruction suivante pour faciliter la visualisation du résultat (si perroquet.bmp est une image stockée sous le répertoire bin/debug de votre solution)

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

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

1-2 Retour aux sources : une image est une succession de bits

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



Problème Scientifique Informatique A2 - 2023

Conformément aux informations

```
byte[] myfile = File.ReadAllBytes("./Images/Test.bmp");
//myfile est un vecteur composé d'octets représentant les métadonnées et les données de l'image

//Métadonnées du fichier
Console.WriteLine("\n Header \n");
for (int i = 0; i < 14; i++)
    Console.Write(myfile[i] + " ");
//Métadonnées de l'image
Console.WriteLine("\n HEADER INFO \n");
for (int i = 14; i < 54; i++)
    Console.Write(myfile[i] + " ");
//L'image elle-même
Console.WriteLine("\n IMAGE \n");
for (int i = 54; i < myfile.Length; i = i + 60)
{
    for (int j = i; j < i + 60; j++)
    {
        Console.Write(myfile[j] + " ");
    }
    Console.WriteLine();
}

File.WriteAllBytes("./Images/Sortie.bmp", myfile);
```

Le résultat de ce code affiche les informations suivantes :

[illegible]

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 dans un premier temps.

1-2-1 Explication

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

Problème Scientifique Informatique A2 - 2023

- 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 40 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 54^{ème} 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

2-TD1-2 Mise en place du format Pivot

L'objectif des TD1-TD2 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, veuillez à concevoir éventuellement d'autres classes qui simplifieront la lisibilité et la sémantique du code.

Problème Scientifique Informatique A2 - 2023

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

- public MyImage(string myfile) lit un fichier (.bmp) et le transforme en instance de la classe MyImage
- 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
- public int Convertir_Endian_To_Int(byte[] tab ...) convertit une séquence d'octets 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

3 - TD3 -TD5 Traitement d'images

Application des différentes opérations sur les images et création de nouvelles images

- **Traiter une image :** TD3
 - Rotation (avec un angle quelconque)
 - Agrandissement (avec un coefficient quelconque)
- **Appliquer un filtre générique (matrice de convolution)** TD4
 - Détection de contour
 - Renforcement des bords
 - Flou
 - Repoussage

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

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

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

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

- **Créer ou extraire une image nouvelle** TD5
 - Créer une image décrivant une fractale

<https://www.maths-et-tiques.fr/index.php/detentes/les-fractales>

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

- Coder et Décoder une image dans une image

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

4 - TD 6-7 Compression : QuadTree et/ou Huffman

Sera complété dans la version 1

