Codage des informations : les images (plutôt pour les professeurs)

Contenus	Capacités attendues
Traitement d'image	Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels.

Activité n°1 : La numérisation, codage de l'image

1. La numérisation

On appelle **BIT** (BInary digiT) le plus petit élément d'information stockable par un ordinateur. Un bit ne peut prendre que deux valeurs (0 ou 1) correspondant à deux états possibles d'un élément de circuit électrique (tension présente ou absente aux bornes d'un dipôle). L'opération qui consiste à transformer (ou coder) une information en une suite de bits est appelée **numérisation**

La numération décimale (base 10)

Elle utilise 10 symboles ou CHIFFRES : 0, 1,2, 3,4, 5,6, 7, 8 et 9.
Exemple : $2459 = 2 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 9 \times 10^0 = 2000 + 400 + 50 + 9 = 2459$

La numération binaire (base 2)
Elle utilise 2 symboles : 0 et 1

La numération hexadécimale (base 16)
Très utilisé par les informaticiens, elle utilise 16 symboles :
0, 1,2, 3,4, 5,6, 7, 8, 9, A, B, C, D et F
Répondre aux questions suivantes en réinvestissant la « mécanique » de la numérisation décrite pour la base 10

- ***Ecrire le nombre binaire de 8 bits (1 octet) en décimal :***

1	0	0	1	1	0	1	1

- ***Quelle est le plus grand nombre décimal que l'on peut écrire en binaire avec 1 octet ?***
- ***Ecrire le nombre hexadécimal en décimal puis en binaire :***

B	5

- ***Quelle est la numérisation hexadécimale du nombre décimal 255 ?***

2. Le codage de l'image

Codage d'une image en noir et blanc :

Pour ce type de codage, chaque pixel est soit noir, soit blanc. Il faut un bit pour coder un pixel (0 pour noir, 1 pour blanc). Une image de 100000 pixels codés occupe donc au moins 100000 bits en mémoire.

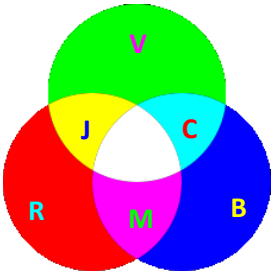
Codage d'une image en niveaux de gris :

Si on code chaque pixel sur 2 bits on aura 4 possibilités (noir, gris foncé, gris clair, blanc).
L'image codée sera très peu nuancée. En général on code chaque pixel sur 8 bits = 1 octet.
On a alors 256 possibilités (on dit 256 niveaux de gris).

- ***Quel est le nombre de bits minimal occupé en mémoire d'une image de 100 000 pixels ?***

Codage d'une image en couleurs 24 bits :

Notre cerveau réalise la **synthèse additive** des lumières colorées reçues par l'œil : il ne peut percevoir séparément les différentes lumières colorées qui lui parviennent simultanément, et il crée une **couleur unique** à partir des informations que lui communiquent les cônes.



Le rouge, le bleu et le vert sont appelés **couleurs primaires** car, par synthèse additive de ces trois couleurs, on peut obtenir toutes les autres couleurs de l'arc en ciel.

- ***Ouvrir l'animation suivante : ImageGetPixelLoupe.swf pour vous aider à comprendre.***

Pouvoir séparateur (ou de résolution) de l'oeil :

Notre œil n'est pas capable de discerner séparément 2 points proches l'un de l'autre si les rayons lumineux qui proviennent de ces points viennent exciter le même récepteur (cône ou bâtonnet) sur la rétine. L'écart angulaire minimum est d'environ 1/60ème de degré.

Ce pouvoir de résolution nous permet de différencier des zones de 100 km sur la surface de la lune, ou un détail d'environ 1 mm pour un objet ou une image située à 3 m de distance.

- *Observez cette mire de Foucault en vous éloignant progressivement de la feuille.*

- ***Que constatez-vous ?***
  Calculez votre pouvoir séparateur.

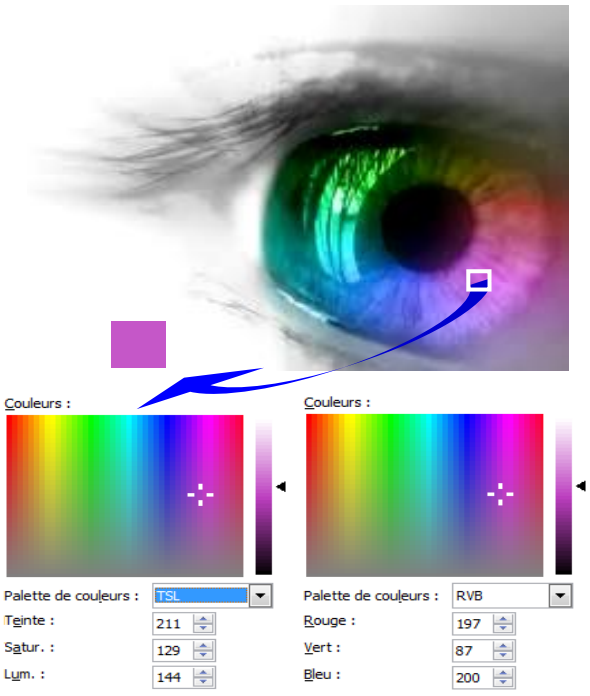
Ces "défauts" de l'œil permettent de reconstituer une image à partir de **pixels** suffisamment proches les uns des autres, et constitués chacun de 3 **luminophores** correspondants aux 3 couleurs primaires : bleu, vert et rouge...
Et notre cerveau semble percevoir une image uniformément répartie sur l'écran !

Le codage RVB sur 24 bits :

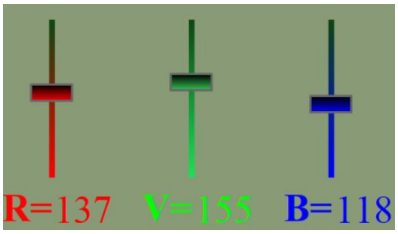
- ***En codant l'intensité lumineuse de chaque luminophore sur 8 bits, nous obtenons..... niveaux différents pour chaque composante colorée, soit au total..... x ..... x ..... = ... couleurs différentes!*** Le codage de la couleur correspondant au pixel agrandi ci-contre est donc, en respectant l'ordre RVB (Rouge Vert Bleu) : 100010101010111111001000

Ce qui s'écrit plus simplement en hexadécimal: 0xC55BC8

Noter le préfixe 0x en héxadécimal



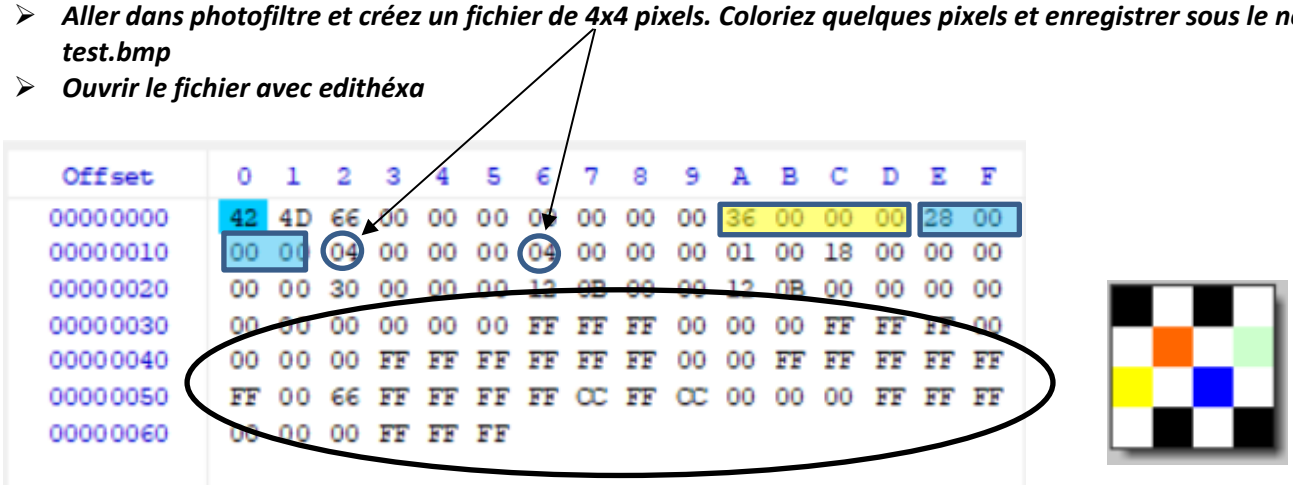
- ***À l'aide de l'animation imageRVBtoHSL.swf, faites varier les canaux rouge, vert et bleu afin d'obtenir différentes couleurs.***



- ***Comment obtenir du rouge ?***
- ***Comment obtenir du blanc ?***
- ***Comment obtenir du noir ?***
- ***Comment obtenir du jaune ?***
- ***Que se passe-t-il quand les trois canaux ont la même valeur (par exemple 125,125,125) ?***

Le BMP est un format d'image sans pertes, pour Windows uniquement. Sans compression, chaque pixel doit être codé à part. C'est pour ça qu'il est l'un des plus simples à coder. En contrepartie il prend beaucoup de place (2.86Mo pour une image de 1000*1000 en 24bit!) Mais dans des versions du BMP plus récentes il peut aussi être compressé avec la compression RLE. Mais celle-ci est très rarement utilisée.

- ***Aller dans photofiltre et créez un fichier de 4x4 pixels. Coloriez quelques pixels et enregistrer sous le nom test.bmp***
- ***Ouvrir le fichier avec edithéxa***



- ***Vérifiez les entêtes du fichier et de l'image, puis modifiez certaines valeurs qui se trouvent dans l'ovale ci-dessus enregistrez le fichier et le rouvrir avec photofiltre ou gimp. Pouvez-vous expliquez les modifications ?***

Entête du fichier :

o.	Valeurs en Hexa	Utilité
2	42 4D	Correspond à 'B' et 'M' (ASCI) pour indiquer le fichier BMP (d'autres valeurs sont possibles : BA, CI, CP, IC, PT)
4	XX XX XX XX	Taille de tout le fichier en octets.
2	XX XX	Signature du programme qui à produit le fichier. Sinon = 0
2	XX XX	Signature du programme qui à produit le fichier. Sinon = 0
4	36 04 00 00	Adresse de départ de l'image (du 1er pixel)

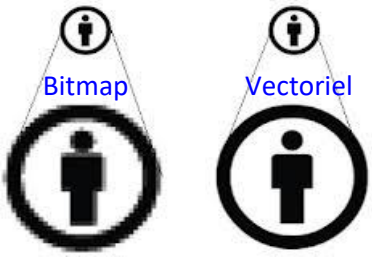
Entête de l'image :

o.	Valeurs en Hexa	Utilité
4	28 00 00 00	Taille de l'entête de l'image en octets : 40o
4	XX XX XX XX	Largeur de l'image en pixels
4	XX XX XX XX	Hauteur de l'image en pixels
2	01 00	Nb. de plans utilisés (1 = une image non animée)
2	XX XX	Nb. de bits par pixel (1, 4, 8, 16 ou 24)
4	XX XX XX XX	Méthode de compression (0 = sans compression)
4	XX XX XX XX	Taille de l'image en octets = H*L*[Nb. d'octets par pixel] + H*[Nb. de 0 à rajouter par ligne])

Attention : Les octets constituant les valeurs sont rangés de gauche à droite en commençant par l'octet de poids faible, et sont arrangés à gauche. Exemple : 7000 qui fait 1B58 en Hexadécimale serra écrit : 58 1B 00 00 sur 4 octets.

Pour aller plus loin:

On distingue 2 grandes catégories de codage d'images :
* Le codage Bitmap ou matriciel : l'image est codée comme un tableau de points
* Le codage vectoriel : l'image est codée par un ensemble de formules mathématiques



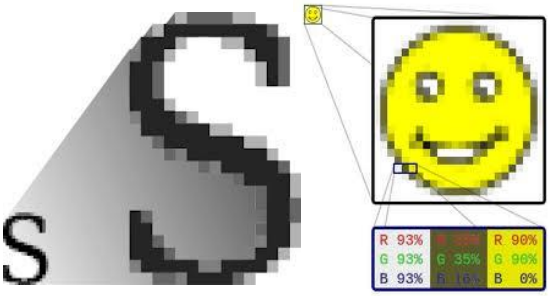
Le **codage vectoriel** convient bien pour des formes géométriques simples. Il nécessite peu d'espace mémoire et permet d'agrandir l'image d'origine sans déformation ni pixellisation apparente.

Le **codage Bitmap** découpe l'image en petits carrés (pixels) et affecte une couleur moyenne à chacun d'eux. C'est une méthode approximative.

Il nécessite beaucoup d'espace mémoire (24 bits soit 3 octets par pixel) et, lors d'un agrandissement, la taille des pixels augmente et devient visible, créant des dents de scies sur les contours.

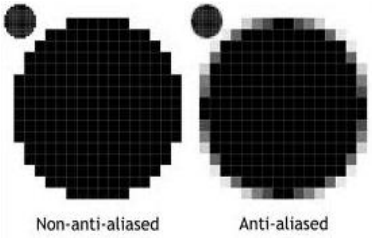
Un fichier image au format **BMP** se compose comme suit :

* les caractères P et 1 suivis d'un espace
* la largeur de l'image (en pixels) suivie d'un espace
* la hauteur de l'image suivie d'un espace
* la liste des pixels, ligne par ligne, de haut en bas et de gauche à droite, sans espaces



Le format **GIF** est un format qui utilise une compression sans perte de qualité. Il supporte la t**ransparence** et permet également de créer des animations : les GIFs animés.

Mais les images au format GIF peuvent contenir un maximum de **256 couleurs**, ce qui est insuffisant pour les photographies mais donne d'excellents résultats pour les logos, formes géométriques, boutons ...



Le format **JPEG** est un format de compression très efficace mais **avec perte de qualité**, qui ne gère pas la transparence.

Conclusion : affichage de page web.

Plus l'image est compressée, plus sa qualité visuelle diminue. Il convient donc de trouver un compromis permettant un chargement rapide tout en gardant une qualité acceptable.

5466 octets

2506 octets

1147 octets

Le codage standard donne une image qui va s'afficher petit à petit dans une page HTML, du haut vers le bas au fur et a mesure du chargement. D'autre part, le codage est sensé permettre un affichage progressif de l'image, celle-ci étant flou au début du chargement et devenant de plus en plus précise et nette.