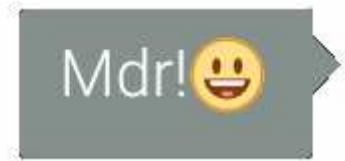


DIC	OTSCIS	MSOST	IP
Séance 2 Seq16	Comment un message informatique est-il codé et envoyé ?		Technologie Cycle 4 - 4ème
IP 1.2 IP 4.2 IP 5.5	Organiser, structurer et stocker des ressources numériques.		

Je viens de répondre ce message à un ami :

Nous allons tenter de savoir comment ce message est codé et envoyé par le téléphone (→ Ce SMS est composé d'un texte, d'un Smiley et il faut l'envoyer à un destinataire précis !)



II) Comment une image est-elle codée par un ordinateur ?

Nous savons qu'avec **un octet** (ensemble de 8 bits), on pouvait coder **256 possibilités**. Pour du texte, c'est suffisant, mais qu'en est-il pour une image ?

Si vous n'avez pas photofiltre sur ton ordinateur, il est téléchargeable ici :

https://www.01net.com/telecharger/windows/Multimedia/photo_numerique/fiches/telecharger-21756.html

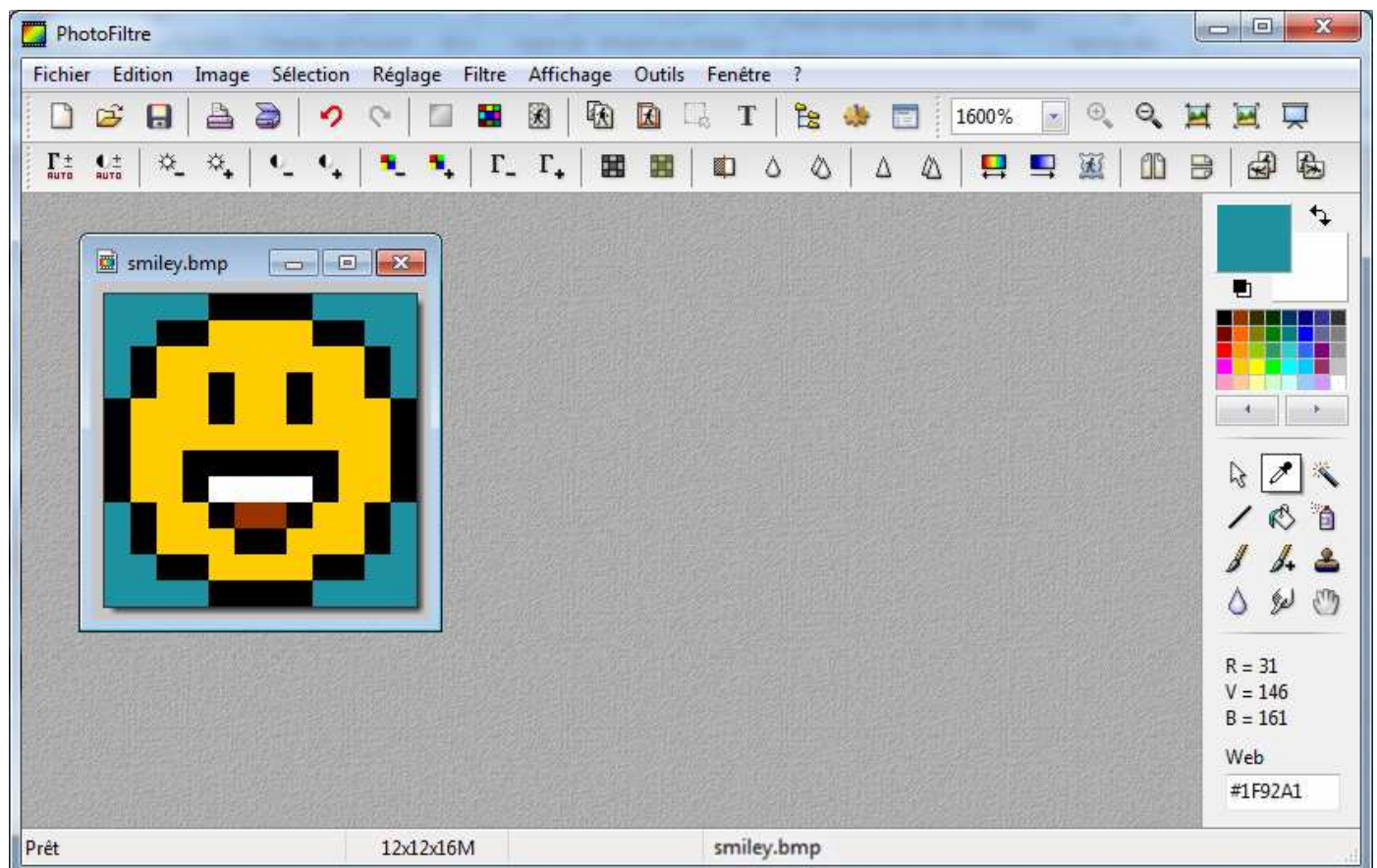
Sur ordinateur, ouvrez avec Photofiltre le fichier « smiley.bmp » disponible en fichier joint.

- L'image est très petite, zoomez au maximum.

① Sur cette image : (pas besoin de compter, c'est écrit quelque part dans la fenêtre !)

- Combien y a-t-il de pixels en largeur ?
- Combien y a-t-il de pixels en hauteur ?
- Combien y a-t-il de pixels en tout ?
- Où trouve-t-on ces informations ?

- Cliquez sur l'outil « pipette ». Cliquez sur le premier Pixel en haut à gauche de l'image.



② D'après-vous, que signifient les 3 valeurs 31, 146 et 161 de R, V et B en dessous des icônes d'outils ?

③ À l'aide des documents des séances précédentes, donnez le codage en binaire de la couleur qui entoure le smiley

④ Donnez les valeurs en hexadécimal des quatre autres couleurs qui composent le smiley :

Noir, Blanc, Jaune, Marron,

⑤ Remplacez la couleur verte du fond par la couleur suivante :

R = 158 V = 158 B = 158

⑥ Quelle est cette couleur ? Pourquoi ?

⑦ Mettez le vert à 0, quelles couleurs obtient-on et pourquoi ?

⑧ Combien faut-il d'octet pour coder la couleur d'un pixel ?

⑨ A l'aide de la réponse à la question ①, indiquez combien faut-il d'octets au minimum pour coder l'ensemble des

pixels de l'image (précisez le calcul)

Voici à quoi ressemble le SMS envoyé à notre ami :

M d r !

01001101 01100100 01110010 00100001

10100001 10010010 00100001 10100001 10010010 00100001 10100001 10010010 00100001 10100001 10010010 00100001

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

10100001 10010010 00100001 10100001 10010010 00100001 10100001 10010010 00100001 10100001 10010010 00100001

10100001 10010010 00100001 10100001 10010010 00100001 00000000 00000000 00000000 00000000 00000000 00000000

0000 10100001 10010010 00100001 10100001 10010010 11111111 00000000 11001100 11111111 00000000 11001100 11111111

0000 00000000 00000000 00000000 00000000 00000000 10100001 10010010 00100001 10100001 10010010 00100001

1010 00000000 00000000 00000000 00000000 00000000 11001100 11111111 00000000 11001100 11111111

00000000 11001100 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11001100 11111111

00000000 11001100 11111111 00000000 11001100 11111111 00000000 00000000 00000000 00000000 10100001 10010010 00100001

10100001 10010010 00100001 00000000 00000000 00000000 00000000 11001100 11111111 00000000 11001100 11111111

00000000 00000000 00000000 00000000 00110011 10011001 00000000 00110011 10011001 00000000 00000000 00000000

00000000 11001100 11111111 00000000 11001100 11111111 00000000 00000000 00000000 00000000 10100001 10010010 00100001

00000000 00000000 00000000 00000000 11001100 11111111 00000000 11001100 11111111 00000000 00000000 00000000

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

00000000 00000000 00000000 00000000 1001100 11111111 00000000 11001100 11111111 00000000 00000000 00000000

00000000 00000000 00000000 00000000 11001100 11111111 00000000 11001100 11111111 00000000 00000000 00000000

00000000 00000000 00000000 00000000 11001100 11111111 00000000 11001100 11111111 00000000 00000000 00000000

00000000 00000000 00000000 00000000 11001100 11111111 00000000 11001100 11111111 00000000 11001100 11111111

00000000 11001100 11111111 00000000 11001100 11111111 00000000 11001100 11111111 00000000 11001100 11111111

00000000 11001100 11111111 00000000 11001100 11111111 00000000 11001100 11111111 00000000 11001100 11111111

00000000 00000000 00000000 00000000 00000000 00000000 00000000 10100001 10010010 00100001 10100001 10010010 00100001

10100001 10010010 00100001 00000000 00000000 00000000 00000000 11001100 11111111 00000000 00000000 00000000

10100001 10010010 00100001 00000000 00000000 00000000 00000000 11001100 11111111 00000000 11001100 11111111

00000000 11001100 11111111 00000000 11001100 11111111 00000000 11001100 11111111 00000000 11001100 11111111

00000000 11001100 11111111 00000000 11001100 11111111 00000000 00000000 00000000 00000000 10100001 10010010 00100001

10100001 10010010 00100001 10100001 10010010 00100001 00000000 00000000 00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

10100001 10010010 00100001 10100001 10010010 00100001 10100001 10010010 00100001 10100001 10010010 00100001

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

10100001 10010010 00100001 10100001 10010010 00100001 10100001 10010010 00100001 10100001 10010010 00100001

On remarque que le texte « Mdr! » n'est codé que sur les 4 premiers octets alors que l'image est codé sur 432 octets ! Les images prennent donc beaucoup plus de place qu'un texte.

Dans la réalité, avant le codage de pixels, on a 54 octets supplémentaires pour les informations sur l'image (type, taille, etc...). L'image fait donc 432 + 54 = 486 octets

smiley.bmp

Type d'élément : Fichier BMP

Dimensions : 12 x 12

Taille : 486 octet(s)