

Séquence 3.2 : Comment sont codées les images en langage informatique ?

4ème

Compétences travaillées :

CT1.2 Mesurer des grandeurs de manière directe ou indirecte.

CT 4.2 Appliquer les principes élémentaires de l'algorithme et du codage à la résolution d'un problème simple.

CT 5.3 Lire, utiliser et produire des représentations numériques d'objets.

CS 1.7 Interpréter des résultats expérimentaux, en tirer une conclusion et la communiquer en argumentant.

A) Le code binaire

Vers la fin des années 30, Claude Shannon démontra qu'à l'aide d'interrupteurs fermés pour «vrai» et ouverts pour «faux» il était possible d'effectuer des opérations logiques en associant le nombre 1 pour «vrai» et 0 pour «faux». Ce codage de l'information est nommé base binaire. C'est avec ce codage que fonctionnent les ordinateurs. Il consiste à utiliser deux états (représentés par les chiffres 0 et 1) pour coder les informations.

A retenir : Le terme **bit** (b avec une minuscule) provient de «binary digit» (chiffre binaire en anglais), c'est-à-dire 0 ou 1 en numérotation binaire. Il s'agit de la plus petite unité d'information manipulable par une machine numérique (un ordinateur). Une suite de 8 bits est appelée un **octet**. L'**octet** est l'unité de **capacité numérique**. Cette unité permet de déterminer une quantité d'information numérique.

Avec un bit il est ainsi possible d'obtenir deux états : soit 1, soit 0.

Mais, grâce à 2 bits, il est possible d'obtenir quatre états (4 combinaisons) différents (2^2): 0 0, 0 1, 1 0, 1 1

Avec 3 bits, il est possible d'obtenir huit états différents (2^3) : 0 0 0, 0 0 1, 0 1 0, 0 1 1, 1 0 0, 1 0 1, 1 1 0, 1 1 1

Calcule combien d'états est possible d'obtenir avec 4 bits :

Avec 4 bits il est possible d'obtenir états différents (.... * * *)

Conversion nombre binaire vers nombre décimal

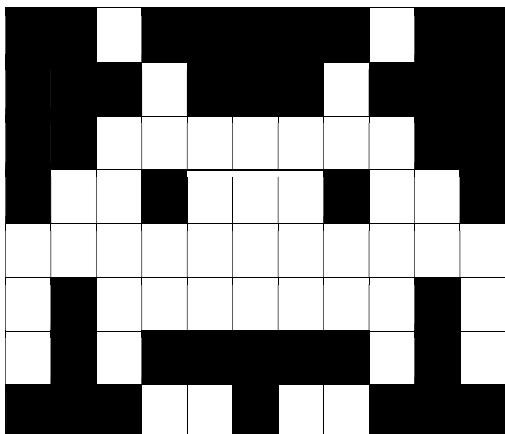
Pour convertir un nombre binaire en décimal, on peut utiliser le tableau suivant (on peut ajouter des cases à gauche si nécessaire et ajouter 2 fois le nombre précédent). A toi de finir le tableau :

NOMBRE BINAIRE	32 (16*2)	16 (8*2)	8 (4*2)	4 (2*2)	2 (2*1)	1	NOMBRE DECIMAL
1						1	1
10					1	0	2
111				1	1	1	$4+2+1 = 7$
11010		1	1	0	1	0	$16+8+2 = 26$
010011							
100100							
101							
00							
01010							

B) Le codage d'une image en noir et blanc

En 1976, au Japon, un jeune prodige, Tomohiro Nishikado crée le célèbrissime jeu «Space Invaders». Nous allons maintenant découvrir un personnage de ce jeu, et comment est codé son image.

Pour ce personnage, l'image sera définie sur 11 pixels en largeur et 8 pixels en hauteur. (on peut les compter très facilement sur l'image ci-dessus!)



Pour ce type de codage, chaque pixel est soit noir, soit blanc.

Il faut donc un bit pour coder un pixel : (0 pour noir, 1 pour blanc).

0	0	1	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	1	0	0	0
0	0	1	1	1	1	1	1	1	0	0
0	1	1	0	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	0	1
1	0	1	0	0	0	0	0	1	0	1
0	0	0	1	1	0	1	1	0	0	0

Combien de bits composent cette image ? Comme nous venons d'apprendre, il faudra un bit pour coder chaque pixel, soit : $11 \times 8 = 88$. Cette image est composée par 88 bits.

Sachant que 1 octet est composé de 8 bits, quelle est la capacité numérique en octets de cette image ?

A toi de coder !



Sur la grille à carreaux ci-dessous, **réalise un tableau** (dont la trame est la définition de l'image) et **code chaque pixel** de l'image à gauche (comme dans l'exemple de la page précédente)

Quelle est la définition de l'image à gauche?

..... pixels en largeur x pixels en hauteur

Entre les deux personnages de Space Invaders, lequel a la capacité numérique la plus importante ? Justifie.

.....
.....
.....
.....
.....

C) Le codage d'une image en couleur

Codage d'une image en couleurs sur 4 bits :



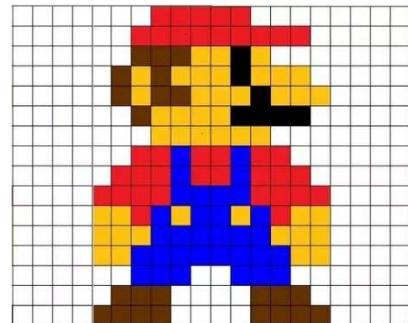
Le codage des images en couleur a évolué rapidement offrant au départ une palette de 4 niveaux de gris. Il fallait déjà 2 bits pour coder un pixel (au lieu de 1 seul bit pour le blanc et noir)

Plus tard, la palette de couleurs a évolué. Chaque pixel pouvait avoir un d'entre 16 couleurs différents. Donc, comme on a vu au début de la séquence, pour avoir 16 combinaisons différents il fallait coder chaque pixel sur 4 bits (un nombre binaire à 4 chiffres), soit 1/2 octet.

Quelle est la définition de l'image ci-contre ?

..... pixels en largeur x pixels en hauteur.

Quelle est la capacité numérique de cette image ?



Codage d'une image en couleurs sur 8 bits :

Dans ce cas on attache une palette de 256 couleurs à l'image. Chaque code (de 0000 0000 à 1111 1111) désigne une couleur choisie. Ainsi chaque pixel est codé sur bits (soit un octet).

Codage d'une image en couleurs sur 24 bits :

Dans ce cas, la couleur du pixel est codée sur une palette de 16 millions de couleurs

Chaque pixel est codé par une nuance de rouge, une nuance de vert et une nuance de bleu.

- 1 octet définit la nuance de rouge
- 1 octet définit la nuance de vert
- 1 octet définit la nuance de bleu



Chaque pixel est codé sur octets soit 24 bits.

D) Quelles sont les principales avantages et inconvénients de tous les codages d'image ?

Dans S:/Groupes/Travail/Classe/Technologie/Sequence 3.2, tu trouves 5 fichiers « image » au format .PNG

Ces 5 fichiers représentent la même image numérique. Comme le nom du fichier l'indique, ce doit être un joystick. Cette image a été défini sur de 1920 pixels de largeur × 1277 pixels de hauteur

Les noms de fichier (créé par le concepteur), nous donne un autre renseignement sur les images. Lequel?

Vous allez maintenant ouvrir chaque fichier numérique d'image, et bien les observer. Pensez à utiliser le zoom du logiciel qui les affichera.

A partir de tes observations d'images et des renseignements découverts précédemment, complète le tableau ci-dessous :

	Nombre de bit(s) par pixel utilisé(s) pour le codage de l'image	Capacité du fichier numérique	Qualité / lisibilité de l'image
Image 1			
Image 2			
Image 3			
Image 4			
Image 5			

Quelle conclusion peut-on tirer de ce tableau ?
