

2. Stocker les images ou les sons

Bonjour à tous. Bienvenue dans cette deuxième séquence du codage binaire. Nous avons vu dans la séquence précédente comment représenter les informations sous forme de bits. Maintenant, nous allons regarder plus précisément comment utiliser cette technique pour stocker des images ou des sons.

Les images, de nos jours, sont utilisées énormément pour les jeux vidéo, la télévision, la réalité virtuelle. Comment est-ce qu'on fait pour les afficher concrètement? On va utiliser ce qu'on appelle des pixels. Les pixels, ce sont des assemblages de petits points de différentes couleurs, du vert, du bleu et du rouge, qu'on va allumer pour faire croire à l'œil qu'il s'agit d'une couleur particulière parce qu'ils sont tellement petits que l'œil ne peut pas distinguer les trois composantes, le rouge, le vert et le bleu. Donc l'œil va penser que c'est une seule couleur. Ces pixels vont servir pour l'affichage, mais vont servir pour énormément de choses, notamment pour faire de la retouche parce que l'information dans l'ordinateur va être stockée de la même façon, que ce soit pour l'affichage ou pour faire de la retouche, de l'impression, etc., on va utiliser cette technique qui est d'utiliser des points qu'on appelle les pixels pour représenter les images.

Pour coder les pixels, on va utiliser des suites de bits. Chacun des pixels de l'image, chacun des points de l'image, va être représenté par un ensemble de bits. Et là, ça va dépendre de la quantité de couleurs qu'on va vouloir afficher. Si on a une image qui est en noir ou en blanc, chacun des pixels va avoir un seul bit qui va dire "Est-ce qu'il est blanc" ou "Est-ce qu'il est noir". Si on veut avoir une image en niveaux de gris, on va pouvoir utiliser par exemple huit bits pour chacun des pixels. Huit bits, ça nous donne 256 valeurs possibles, 2 puissance 8. On va avoir le zéro qui va être du noir, 255 - le maximum - qui sera du blanc, et les valeurs intermédiaires qui seront des gris plus ou moins foncés. Enfin, on va pouvoir faire de la couleur en combinant du rouge, du vert et du bleu comme on l'a vu précédemment. Là on va utiliser huit bits par composante. Huit bits pour le rouge, donc si on est à zéro pour le rouge on ne va avoir aucun rouge dans la couleur. Si on est à 255 pour le rouge, la couleur sera plus rouge. Chacune des composantes va comme ça avoir une valeur entre zéro et 255 et on les combinant, on pourra choisir la couleur parmi énormément de couleurs différentes : 16 millions.

La question qui se pose ensuite, c'est "Si j'ai une suite de bits qui correspond à une image, comment savoir si 24 bits là-dedans correspondent à 24 pixels noirs et blancs, parce que chacun des pixels utilise un bit ou alors à un seul pixel en couleurs, parce que chacun de ces pixels utilise 24 bits". Il va falloir en fait ajouter de l'information pour préciser que mon image est stockée par des pixels et chacun des pixels est codé avec un certain nombre de bits, noir et blanc, gris ou couleur. Et de la même façon, il va falloir préciser la taille de l'image. Est-ce qu'elle est plutôt rectangulaire? Plutôt allongée? Plutôt verticale? Est-ce que c'est un carré? L'image, ça va être un ensemble de pixels décrits par un ensemble de bits, mais qui va en plus être décrite par des caractéristiques générales qui sont la taille, les dimensions, la hauteur, la largeur et également le nombre de bits utilisés pour chacun des pixels.

Ce qu'on vient de décrire, ce sont les images qu'on appelle bitmap, c'est en fait un ensemble de bits qui vont décrire chacun des pixels. En pratique, l'ordinateur n'a aucune idée du contenu de cette image. Ce qu'on a sur la droite de la présentation ici, c'est un disque qui a été

représenté par des pixels. En pratique, l'ordinateur ne sait pas que c'est un disque. S'il veut le savoir, il va falloir qu'on lui donne un algorithme pour faire de la reconnaissance de forme. Si on ne lui donne pas ça, il sait simplement qu'il y a un ensemble de pixels avec des couleurs plus ou moins noires et c'est tout.

En pratique, une image comme ce disque, on pourrait la coder de manière un peu plus pratique. Par exemple, ici au lieu de dire que c'est un ensemble de pixels qui sont blancs si on n'est pas dans le disque, noirs si on est dans le disque et un peu gris quand on est au bord du disque, on pourrait dire directement que c'est un disque qui est noir et de telle taille et à tel endroit dans l'image. Ça serait beaucoup plus précis parce qu'on n'aurait pas les problèmes en bordure avec les points qui sont légèrement grisés parce qu'ils sont à cheval sur le bord du disque et en plus, ça serait beaucoup plus concis parce qu'au lieu de décrire chacun des pixels, on décrirait simplement le disque, sa couleur, sa taille, sa position et c'est tout. Ça, c'est ce qu'on appelle une image vectorielle. C'est une image où au lieu de décrire chacun des pixels, on va décrire comment on construit l'image avec des formes mathématiques. Les formes mathématiques, ça va être des disques, des rectangles, des polygones, des courbes mathématiques plus compliquées. On va donner un ensemble de caractéristiques pour décrire cette courbe mathématique et également des informations sur comment la colorier, comment colorier l'intérieur éventuellement avec des dégradés et des choses comme ça, mais toujours sous forme mathématique, donc avec des nombres. Alors ça, ça va être bien pour faire du dessin technique parce que le dessin technique en général, c'est justement des formes très précises, des carrés, des rectangles, des ronds avec des couleurs assez précises. Par contre, si on veut faire des photos, ça va être plus compliqué parce que les photos en général on a énormément de couleurs différentes avec des formes qui ne sont pas forcément très précisément des formes thématiques comme on a l'habitude de les décrire avec des carrés et des rectangles. Ça ne va pas forcément être très adapté. Donc, pour les photos, on va plutôt faire des images bitmap avec des pixels comme on l'a vu avant.

Maintenant, on va s'intéresser au codage du son. Le son, c'est des ondes sonores qui nous arrivent dans nos oreilles qui sont en fait à chaque instant une hauteur et un volume représentant le bruit qu'on entend. En pratique, on a une infinité de valeurs pour chacun des temps successifs. On ne va bien sûr pas pouvoir toutes les coder, sinon ça ferait une infinité de données à stocker dans l'ordinateur. Ce qu'on va faire, c'est ce qu'on appelle de l'échantillonnage. L'échantillonnage, ça consiste à diviser le temps en petites périodes et à dire par exemple "Pendant la milliseconde qui vient de s'écouler, le volume et la hauteur de la note jouée c'était en gros celle-ci". On va en fait moyennner, approximer pendant des petites périodes de temps pour éviter de stocker trop de valeurs. Évidemment, il va falloir faire ça de manière adaptée. Il faut que l'oreille humaine ne voit pas la différence, c'est-à-dire utiliser une fréquence d'échantillonnage qui soit suffisamment élevée pour que l'oreille humaine ne voit pas la différence entre le fait d'avoir moyennné sur une milliseconde et le signal réel. Et il faut aussi ne pas faire trop d'échantillonnage, c'est-à-dire de ne pas avoir une fréquence trop élevée parce qu'on aurait trop de données à stocker. En pratique, on va faire de l'échantillonnage avec quelque chose qui correspond à ce dont l'oreille humaine est capable, c'est-à-dire entendre des fréquences entre 20 hertz et 20 kilohertz.

En pratique, on va avoir un ensemble de valeurs successives pour chacune de ces petites périodes d'échantillonnage. Pour chacune de ces valeurs, on va avoir un paquet de bits qui va décrire quelle est la hauteur de la note, le volume, etc. Et ça, ça va être le son, cette suite de

bits, et comme pour les images on va voir une en-tête qui va nous préciser quelle est la fréquence de l'échantillonnage. Pour préciser, je vais te donner un certain nombre de valeurs et en fait chaque valeur correspond à par exemple une milliseconde du son. Ça, c'était le codage du son habituel.

Comme pour les images où on pouvait faire des images bitmap par pixels ou alors des images vectorielles avec des formules mathématiques, on va aussi pouvoir faire quelque chose de différent pour le son. Le son, quand on a un morceau de musique, ça va être une partition qui va être jouée par des instruments. Une partition, c'est quoi ? C'est en fait des notes successives qui sont caractérisées par une hauteur et également par une durée. Et les instruments qui vont la jouer, ils ont des caractéristiques aussi qu'on peut décrire, par exemple, le type d'instrument, le timbre, etc. Le morceau de musique, on va pouvoir le récréer en utilisant les informations sur la partition et les instruments qui la jouent. Ça, ça va être un peu comme pour une image vectorielle où on décrit l'image par la façon de la créer en utilisant des formes thématiques, ici on va décrire la musique en expliquant comment on l'a créée à partir de la partition et de l'instrument. Et ça, ça permet en fait de coder le son de manière différente, pas forcément aussi adaptée pour tous les cas, mais pour des morceaux de musique par exemple ça peut être une approche intéressante, complémentaire à l'approche qu'on avait vue précédemment.

Qu'est-ce qu'il faut retenir de cette séquence ? Alors, tout d'abord on peut coder les images et les sons par des suites de nombre de bits, comme on l'avait fait pour les nombres précédemment ou pour les lettres d'un texte. Après, on va avoir des programmes qui vont servir à l'ordinateur pour interpréter ce codage, c'est-à-dire lire les informations de l'image, quelle est la taille, la dimension, le nombre de bits utilisés pour les pixels et qui vont permettre à l'ordinateur d'afficher cette image à l'écran, de l'imprimer, de faire de la retouche, etc. De la même façon, l'ordinateur va avoir un programme pour lire les informations du son, pour les jouer sur un haut-parleur. À l'opposé, quand on va vouloir faire rentrer une image ou un son dans l'ordinateur, on va avoir un programme qui va recevoir le signal et le traduire en codages avec la stratégie exactement inverse. Entre les deux, les données sont codées avec les bits qu'on a vus précédemment. Et que ce soit pour le stockage sur le disque dur, en mémoire, un transfert sur Internet entre deux ordinateurs, de la retouche audio ou photo, les données vont rester codées. C'est uniquement quand les données vont rentrer dans l'ordinateur pour faire de l'enregistrement ou être lues ou affichées par l'ordinateur qu'on va les décoder pour les donner à l'être humain.