

1, 2, 3, codez ! - Activités cycle 3 - Séance 3.2: Comment encoder une image en noir et blanc ?

Résumé	Les élèves appliquent leurs découvertes de la séance précédente à l'encodage numérique d'images en noir et blanc. Ils visualisent tout d'abord un même fichier avec un logiciel éditeur de texte et avec un logiciel éditeur d'images, pour comprendre comment est effectué l'encodage. Puis ils encodent eux-mêmes un petit damier et le visualisent.
Notions	« Information » <ul style="list-style-type: none"> • Une image peut être représentée sous la forme d'une grille de carrés appelés pixels. • Un ordinateur représente n'importe quelle information par un code qui n'utilise que 2 symboles, 0 et 1, appelés bits : c'est le code binaire. • En noir et blanc, chaque pixel peut être représenté par un unique bit.
Matériel	Par binôme <ul style="list-style-type: none"> • Ordinateur avec logiciel éditeur de texte basique (Bloc-notes par exemple) et un logiciel éditeur d'image basique et dossier accessible aux élèves contenant les fichiers III-3.2_recherche_NB et III-3.2_defi_NB_vierge. • Loupe • vidéoprojecteur Pour la classe <ul style="list-style-type: none"> • Fiche 41 projetée au tableau
Lexique	Image, pixel, encodage
Durée :	1 heure

Note pédagogique

- Nous recommandons fortement d'utiliser l'éditeur d'images XnView dans sa version minimale, car il cumule les avantages suivants :
 - Il est gratuit pour un usage éducatif ;
 - Il lit correctement les fichiers de formats pbm (et pgm ou ppm pour la [Séance III-3.3](#)) ;
 - Son interface est simple ;
 - Il permet de zoomer sans saisie de texte à l'aide des loupes « + » et « - » ;
 - Il permet de réactualiser une image déjà ouverte (commande « Réouvrir » du menu « Fichier » ou raccourci clavier « Ctrl R »), ce qui facilite grandement les essais/erreurs des élèves.
- XnView peut être téléchargé ici : <http://www.xnview.com/fr/xnview>

Situation déclenchante

L'enseignant reprend au tableau l'image de la pomme ([Fiche 39](#), image A) pixellisée à la séance précédente sur 256 pixels (grille 2) : «*nos explorateurs veulent transmettre cette image à la base, sans se déplacer. Comment peuvent-ils faire ?*».

Le débat qui s'ouvre permet d'énoncer des moyens, des concepts très variés. L'enseignant note au tableau les différentes idées. En particulier, plusieurs pistes méritent d'être creusées : «*il y a des pixels blancs et des pixels noirs*», «*on peut décrire chaque pixel comme en morse avec une lampe-torche*».

Les propositions permettront de rappeler la notion de codage binaire (0 pour les pixels noirs, 1 pour les pixels blanc) si cette notion a été abordée à la [séquence III-1](#). «*Imaginez que les explorateurs ne puissent envoyer qu'un texte simple (un SMS) pour décrire cette image.*» Petit à petit émerge l'hypothèse qu'un texte composé de 0 et de 1 doit suffire à décrire l'image.

Observation : comprendre l'encodage d'une image en noir et blanc (collectivement puis par binômes)

Afin de vérifier si un texte rempli de 0 et de 1 peut suffire à représenter une image pixellisée en noir et blanc, l'enseignant montre aux élèves comment ouvrir le fichier texte [III-3.2_recherche_NB.pbm](#) avec un éditeur de texte minimaliste (Bloc-notes, sous Windows, par exemple). Le contenu du fichier est le suivant :

```
P1
16 16
0000000000000000
0000000000000000
0000000110000000
0000000111111100
0000000111110110
0001111111111110
0001111101111010
0011000000000110
0010000000000110
0010000000001110
0010000000001010
0011000000001100
0001100011111000
0000111111100000
0000000000000000
0000000000000000
```

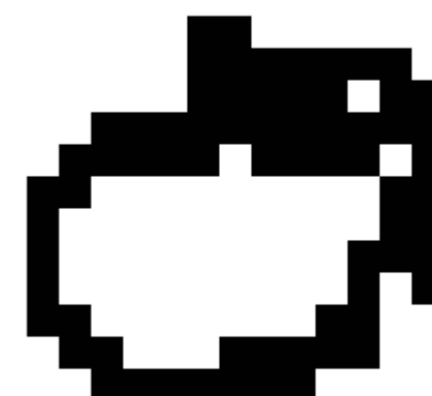
Il explique que ce fichier est une version numérique de l'image de la pomme pixellisée avec 16x16 pixels. Il demande si la grille de 0 et de 1 semble effectivement dessiner la pomme : c'est le cas, on reconnaît un peu la feuille et le contour, dessinés par des « 1 ». Si les élèves ne sont pas convaincus, on peut remplacer automatiquement tous les « 0 » par des « . » dans le bloc-notes (par « Ctrl H »), la pomme apparaît alors très clairement (annuler aussitôt cette modification par une commande « Ctrl Z »).

L'enseignant demande s'il y a d'autres choses que cette grille de 0 et de 1 dans le fichier. Les élèves s'interrogent sur les 2 premières lignes du fichier et peuvent faire l'hypothèse que 16 et 16 correspondent aux dimensions de la grille de pixels. L'enseignant complète en expliquant que ces premières lignes définissent le format des données écrites dans ce fichier : « P1 » désigne le codage binaire (des 0 et des 1 exclusivement), et la ligne « 16 16 » annonce qu'un tableau de 16 colonnes et 16 lignes de données va suivre. Effectivement, il y a 16x16=256 bits (des 0 ou des 1) inscrits dans le corps du fichier (description de la grille de pixels proprement dite).

L'enseignant montre ensuite comment visualiser ce même fichier, non plus avec un éditeur de texte, mais avec un éditeur d'images, comme XnView (cf note pédagogique en début de séance). La pomme apparaît en tout petit, il faut zoomer fortement pour la voir effectivement. La classe constate, par comparaison entre la grille de 0 et de 1 et l'image de la pomme, que les pixels noirs sont encodés par « 1 » et les pixels blancs par « 0 ».

Les enfants appliquent immédiatement ce qui a été montré par l'enseignant : ouverture du fichier d'une part avec l'éditeur de texte et d'autre part avec l'éditeur d'image (voir partie haute de la [Fiche 41](#)). Ils observent la pomme miniature à la loupe sur l'écran avant d'avoir zoomé puis zooment autant que nécessaire.

Une synthèse collective est effectuée concernant le format du fichier : P1 sur la première ligne, le nombre de colonnes puis le nombre de lignes sur la deuxième, puis une grille de 0 (pour le blanc) et de 1 (pour le noir) de dimension adéquate sur les lignes suivantes.

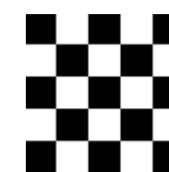


Notes scientifiques

- L'extension pbm du fichier signifie *portable bit map* » ou « carte transportable binaire ».
- Si, dans la grille de pixels, on indique des valeurs différentes de 0 et 1, elles sont interprétées modulo 2 : 0 donne du blanc, 1 donne du noir, 2 (puis toutes les valeurs paires) donne de nouveau du blanc, 3 (puis toutes les valeurs impaires) donne du noir, etc.

Défi : encodage numérique d'un damier (par binômes)

L'enseignant lance aux élèves le défi suivant : ils doivent créer avec l'éditeur de texte un fichier représentant un petit damier noir et blanc de 5 lignes et 5 colonnes, avec des cases noires aux angles. Le fichier, vide pour le moment, s'appelle [III-3.2_defi_NB_vierge.pbm](#). Puis ils doivent sauvegarder le fichier et l'ouvrir avec l'éditeur d'image, sans fermer l'éditeur de texte, et appeler l'enseignant lorsqu'ils ont, côte à côte, d'une part le fichier ouvert avec l'éditeur de texte, et d'autre part le damier visible avec l'éditeur d'image.



L'encodage que les élèves doivent proposer est le suivant :

```
P1
5 5
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
```

Conclusion et trace écrite

La classe synthétise collectivement ce qui a été appris au cours de cette séance :

- On peut représenter les pixels d'une image par des nombres.
- Chaque pixel d'une image en noir et blanc est représenté soit par un 0 (pixel blanc), soit par un 1 (pixel noir).
- Pour transmettre une image en noir et blanc, on l'encode puis on transmet la version encodée, que l'on décode à l'arrivée pour reconstituer l'image.

Les élèves notent ces conclusions dans leur cahier de sciences. L'enseignant, quant-à-lui, met à jour la portion « Information » de l'affiche intitulée « Qu'est-ce que l'informatique ? ».

<< [Séance III-3.1](#)

[Séquence III-3](#)

[Séance III-3.3](#) >>

Source URL: <https://www.fondation-lamap.org/fr/page/34530/1-2-3-codez-activites-cycle-3-seance-32-comment-encoder-une-image-en-noir-et-blanc>