Programmation et robotique en classe. Séances 4 et 5

Inducteur

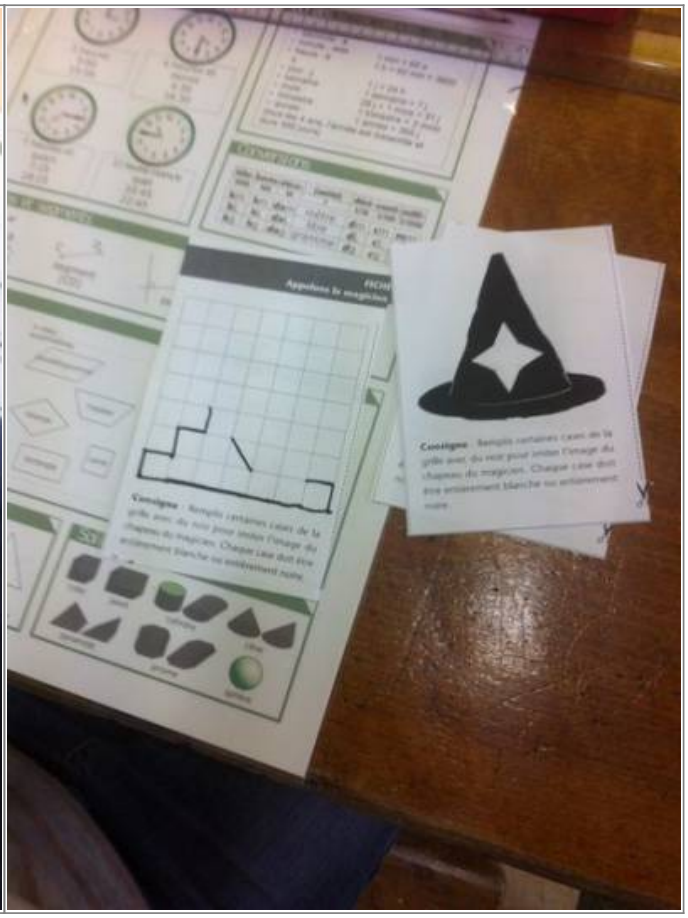
L'histoire de notre héros continue. Il a dès lors récupéré le trésor. Il l'ouvre et un message se situe à l'intérieur. Il contient une formule magique que seul un sorcier peut exécuter. L'île est trop grande pour être explorée, le héros a une idée : il va demander aux oiseaux de rechercher ce sorcier depuis leur vol dans le ciel. Pour se faire il doit dessiner le chapeau du sorcier pour leur faire comprendre qui rechercher. Aussi il va utiliser des galets soit noir soit blanc répartis sur un quadrillage de 7 cases par 7 cases.

Et hop ! L'inducteur est en place.

Pixelliser et coder une image NOIR ET BLANC

L'activité

ON distribue une image du chapeau et des quadrillages de 7 X 7.



Les cases sont appelées “**pixels**” et les élèves doivent reproduire ce chapeau. Il y a a donc **7 X 7 =**

49 cases / **pixels** à remplir ou non. Une case ne peut être que noire ou blanche. Pas de découpe possible à l'intérieur des cases.

Pas de soucis pour la réalisation de ce travail. Attention certains placent la base trop haute et donc les proportions ne sont pas respectées mais dans l'ensemble, tout se passe bien.

On récupère et analyse les productions. Unanimes : on reconnaît le chapeau mais ce n'est pas top top.

On cherche pourquoi et ce qui pourrait améliorer ce travail. Très vite, certains proposent l'idée d'utiliser des cases plus petites.

A partir de l'idée "*si les cases étaient 2 fois plus petites ce serait mieux*", on recommence l'activité. Donc le nombre de cases doublent (7 X 2) X (7 X 2) et on travaille donc avec **196 pixels** plutôt que 49. L'image obtenue est de ce fait beaucoup plus nette. On vient d'aborder la notion de **résolution**.

Pour bien montrer que chaque image d'un ordinateur fonctionne ainsi, il suffit d'en projeter une au tableau et de zoomer jusqu'à l'apparition des pixels.

L'encodage

Au tableau, on synthétise ce qui a été fait lors du premier test (7 X 7) Et on propose de coder cette image. Très vite ils réinvestissent ce qui a été vu par rapport au codage de l'alphabet et au code César et proposent

- 0 pour une couleur –> blanc
- 1 pour l'autre –> noir

Ils ont compris que l'image réalisée n'est qu'une succession de 0 et de 1. **1 couleur codée sur 1 bit**. Par contre pour reconstituer, l'image du tableau, on a besoin de connaître sa dimension. On réservera donc les 2 premières cases aux dimensions de l'image (nécessairement 2 cases car il se peut que l'image soit rectangulaire)

Ce qui donnerait pour notre image le codage suivant :

7 7 0001000 0011100 0011100 0110110 0100010 1110111 1111111

L'entête de ce fichier (les deux premières cases) indique qu'il faut repartir ces valeurs dans un tableau de 7 X 7

En groupe classe, on réinvestit le tout avec un exemple simple (ici damier) :

Enfin, trace murale : depuis une grille de 8 X 8 au tableau, on encode le chiffre 3 telle une image, ce qui donne :

Pixel Art

Enfin on finit la séance par une activité de réinvestissement plus ludique avec des images couleurs à pixeliser.

Cf section Ressources pour des utilitaires en ligne de **Pixel-Art** et **fiche Ressources 18** avec grilles de **la fiche 17**.

Pixelliser et coder une image NOIR ET BLANC (2)

Réinvestissement de la séance précédente. On distribue aux élèves les fiches 39 et 40 Et on leur demande à l'aide d'un calque de reproduire une figure sur une grille de leur choix.

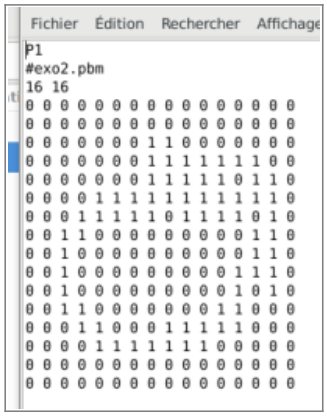
On mutualise les productions et réinvestit les notions de pixel et de résolution.

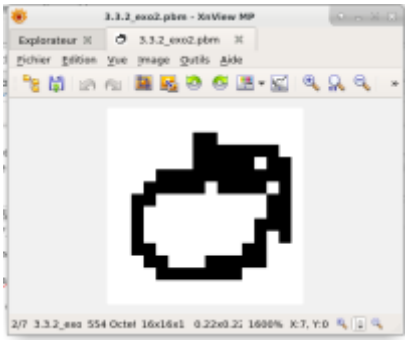
On rappelle les règles de l'encodage, c'est à dire 1 pour le noir et 0 pour le blanc.

Ensuite sur un éditeur de fichier le plus simple (mousepad, leafpad, xed… sous Linux ou Notepad sous Windows), ils encodent leur dessin en respectant

Première ligne : Nom du fichier. Exemple P1 Seconde ligne : Taille de ligne Longueur par Largeur Troisième : Encodage ligne par ligne

Ensuite on ouvre ce même fichier dans un visualisateur d'image.





|

Je commande xnView car il ne “lisse” pas les pixels et permet de garder un zoom avec des pixels réellement noirs ou blancs de forme carrée.

Objectif : une image n'est qu'un fichier texte contenant un code couleur pour chacun de ses pixels. Plus il y a de pixel plus l'image est nette (résolution)

Pixelliser et coder une image EN NIVEAUX DE GRIS (16)

Séance à venir

Pixelliser et coder une image COULEUR

Séance à venir

Ressources

- Les fiches 1 2 3 Codez
  - fiche_16.pdf
  - fiche_17.pdf
  - fiche_18.pdf
- Petit logiciel pour pixeliser une image
- http://castor-informatique.fr/questions/lamap/demo_guide_lamap.html
- Des logiciels de “pixel art” en ligne
- Make Pixel Art
- Pix Attack

Sous Linux : pour le codage RGB, utiliser en parallèle les utilitaires **gpick** ou **gcolor2**

```
aptitude search gpick gcolor2
```

Des équivalents existent pour **windows** et **Mac**