

# Logigraphe : ajouts

## 1. Ajouts de fonctionnalités

Ajouter la possibilité, en pressant une touche, de cocher/décocher une ligne/colonne de la grille

## 2. Compilation

Modifier votre `makefile` afin de le rendre plus générique (Utilisation de variables).

Créer différents répertoires permettant de stocker les différents types de fichier : `include/` pour les fichiers d'entête; `src/` pour les fichiers source; `obj/` pour les fichiers objets et `bin/` pour les exécutables. Modifier le `makefile` en conséquence.

Modifiez votre fichier `makefile` pour qu'il puisse satisfaire les conditions suivantes :

- Utiliser des variables pour définir le nom de l'exécutable, le compilateur utilisé, la liste des fichiers sources, ...
- La commande `make clean` devra effacer tous les fichiers produits (objets, exécutables, bibliothèques).
- La commande `make archive` devra créer, une archive (format `.tar.gz`) contenant l'arborescence du projet. Vous utiliserez pour cela la commande `tar`.

## 3. Manipulation d'image PPM

Ecrire une application permettant de transformer un fichier image au format `PPM` en un fichier contenant un logigraphe.

Les images au format `PPM` ont l'avantage d'être facilement lisibles par un programme. En pratique, le fait qu'elles ne soient pas compressées limite leur usage.

La couleur des points devra être définie par 1 composante dont la valeur est comprise entre 0 et 255. Dans le fichier `PPM`, la couleur des points sera écrit en texte (`ASCII`) et non en binaire (amélioration possible : lire les fichiers binaires).

Un fichier au format `PPM` se composera donc comme suit :

- Sur une ligne, un nombre décrivant le format utilisé (`P2` pour les images en niveau de gris et `P3` pour les images couleurs `ASCII`),
- Sur la ligne suivante, deux nombres indiquant respectivement la largeur et la longueur de l'image (en nombre de pixels)
- Sur la ligne suivante, la valeur maximale pour les composantes des couleurs de l'image (généralement 255).
- Puis sur les lignes suivantes, séparés par des caractères d'espacement (espace, tabulation, nouvelle ligne), les valeurs de la couleur des différents pixels de l'image écrit en texte (`ASCII`)

Remarque : Aucune ligne ne doit dépasser 70 caractères.

Exemple de fichier PPM (image.ppm).

L'image contient 304 lignes de 497 pixels. Le niveau de gris de chaque pixel est défini par une valeur comprise entre 0 et 255. Ici, seuls les 10 premiers points de l'image sont présentés ici (pixel en haut à gauche de l'image).

```
P3
497 304
255
253 155 0 253 155
253 155 0 253 160
```

Ecrire une application qui prend une image PPM de taille  $N \times M$  et en niveau de gris (Amélioration : une image couleur) en entrée et qui renvoie un logigramme, binaire, de taille  $P \times K$  (avec  $P \leq N$  et  $K \leq M$ ).

Remarque : le niveau de gris d'un pixel défini par trois composantes  $R$ ,  $V$  et  $B$  peut être obtenu par le calcul suivant :  $g = 0.299 * R + 0.587 * V + 0.114 * B$