



ASTEK



## TP #01 - Intro et jour de detente de Piscine

[majdi.toumi@epitech.eu](mailto:majdi.toumi@epitech.eu)  
[astek\\_resp@epitech.org](mailto:astek_resp@epitech.org)

*Abstract: Nous considererons cette journee comme une journee de detente qui va nous permettre d'aborder une premiere fois le dessin en C et plus particulierement l'utilisation d'une librairie "maison" : la minilibx*



# Table des matières

.1	Graphique ? . . . . .	2
.2	La minilibX . . . . .	3
.2.1	mlx_init . . . . .	3
.2.2	mlx_new_window . . . . .	3
.2.3	mlx_pixel_put . . . . .	3
.2.4	mlx_clear_window . . . . .	4
.2.5	Les autres fonctions . . . . .	4
.3	On a dit RGB ? . . . . .	5
.4	Exercices . . . . .	6
.5	A propos des exercices . . . . .	7



## .1 Graphique ?

Jusqu'à présent, vous n'avez affiché que des informations sur un écran texte. Vous travaillez pourtant dans un environnement graphique avec des jolies couleurs chatoyantes et des petits dessins.

Comment faire de même dans vos programmes ?

Il suffit d'utiliser les fonctions graphiques que l'on trouve dans les bibliothèques graphiques. Le système X-Window (cet environnement graphique que vous utilisez) est composé de deux parties qui communiquent :

- Un **serveur X** qui gère clavier, souris, écran et qui reçoit des ordres d'affichage et de dessin
- Plein d'**applications X** qui demandent au serveur de tracer des lignes, des caractères, etc...

La commande **ps -auwx** vous permet de voir tous les programmes qui tournent sur votre machine. Vous remarquerez d'un côté le serveur (X/Xorg/XFree86) et de l'autre les applications que l'on voit à l'écran : emacs, dclock, google-chrome, et bien d'autres.

Toutes les applications utilisent les mêmes bibliothèques pour donner des ordres d'affichage au serveur et nous allons faire de même.

Afin d'éviter de se perdre dans les profondeurs des bibliothèques X, nous allons utiliser une petite bibliothèque, beaucoup plus simple à manier, qui permettra cependant de réaliser tous les projets graphiques de l'année.

Cette bibliothèque appelée minilibX est en fait une "surcouche" des bibliothèques X standard ; Elle utilise les bibliothèques X pour fonctionner.



Il s'agit d'une bibliothèque "faite-maison". Merci de nous signaler ce que vous pensez être un bug :)



## .2 La minilibX

Voici les principales fonctions de la librairie. Vous retrouverez la totalite des fonctions dans le fichier **mlx.h**

### .2.1 mlx\_init

```
1 void    *mlx_init()
```

Cette fonction qui doit etre appele avant tout autre fonction de la librairie initialise tout ce qu'il faut comme il faut :)

Elle renvoie un pointeur sur void, qui servira d'identifiant pour l'affichage (ce sur quoi ca pointe, n'est pas vraiment votre probleme).

### .2.2 mlx\_new\_window

```
1 void    *mlx_new_window(void *mlx_ptr, int size_x, int size_y, ↵  
    char *title)
```

Cette fonction cree une nouvelle fenetre a l'ecran :

- "mlx\_ptr" contient la valeur que l'on a recupere de la fonction mlx\_init
- "size\_x" et "size\_y" correspondent a la taille de la fenetre (en nombre de pixels)
- "title" est une chaine de caractere correspondant au titre de la fenetre.

Attention, la fenetre n'apparaitra probablement pas tant que rien n'aura ete affiche dedans.

La fonction renvoie un void\* qui servira d'identifiant pour cette nouvelle fenetre. Il sera utilise pour toute action relative a cette fenetre (il est possible d'ouvrir plusieurs fenetres...)

### .2.3 mlx\_pixel\_put

```
1 int      mlx_pixel_put(void *mlx_ptr, void *win_ptr, int x, int ↵  
    y, int color)
```

Cette fonction permet d'afficher un pixel dans une fenetre :



- "mlx\_ptr" et "win\_ptr" sont les 2 identifiants renvoyés par les deux fonctions précédentes.
- "x" et "y" sont les coordonnées en pixel à l'intérieur de la fenêtre et par rapport au coin supérieur gauche qui est en (0,0).
- "color" contiendra la couleur du pixel à afficher. Elle sera stockée dans un int sous la forme suivante : **0x00RRGGBB** (en hexadecimal). Le premier octet de l'int est à zéro, chacun des trois autres contient une valeur entre 0 et 255 pour chacune des trois composantes, Red, Green et Blue.



La coordonnée en "y" n'est pas négative!

## .2.4 mlx\_clear\_window

```
1 int      mlx_clear_window(void *mlx_ptr, void *win_ptr)
```

Cette fonction efface la fenêtre. Les 2 pointeurs sont toujours ceux relatifs à l'affichage et à la fenêtre.

## .2.5 Les autres fonctions

Il y a bien évidemment tout un tas d'autres fonctions intéressantes à découvrir mais pour cela il faut être minimum curieux(se) et lire le fichier mlx.h :)



N'oubliez pas que pour compiler vos programmes, il vous faut ajouter la minilibX ainsi que les lib X11 du système :

```
-L/usr/lib64 -lmlx_${HOSTTYPE} -L/usr/lib64/X11 -lXext -lX11
```



Pour les plus curieux(se), les sources de la minilibX sont disponibles à la fin du TP.



### .3 On a dit RGB ?

J'en vois qui ont un peu de mal avec les RGB dans l'int pour mlx\_pixel\_put ...

Au final, ce n'est pas tres complique : on considere notre int non pas comme un int, mais comme 4 octets. Trois de ces octets vont contenir les trois composantes RGB de la couleur desiree.

Procedons par etapes :

1. Une fois la couleur choisie, il faut trouver les composantes rouge, verte et bleue. Pour cela, le meilleur remede c'est d'essayer au hasard et de voir a l'ecran ce que l'on obtient.
2. Il ne reste plus qu'a mettre nos trois valeur dans le meme int. Pour cela, plusieurs solutions :

- Utilisation des decalages de bits :

*Retournez voir le cours de C avec les exemples de decalages de bits pour comprendre comment faire passer une valeur du dernier octet de l'int dans l'avant-dernier. Une fois vos trois composantes correctement decalees, il suffit de les rassembler dans un meme int, grace a un OU binaire.*

***Faites un dessin, c'est plus facile !!***

- Utilisation d'un autre type :

*Mettez l'adresse d'un int dans un pointeur sur char.*

*Vous pouvez alors acceder aux differents char qui composent l'int, et les remplir separement.*

*Attention : souvenez-vous de l'exo "is\_bigendian"; Ce type de code ne va pas forcement marcher partout.*

***L'ordre des octets dans l'int varie selon le type de machine.***

Voila, votre int est pret a etre utilise par mlx\_pixel\_put !



Il existe certains fichiers decrivant les principales couleur.  
Faites un "locate rgb.txt" sur votre systeme d'exploitation.



## .4 Exercices

Rien à rendre, on s'amuse aujourd'hui.



## .5 A propos des exercices

Du coup il y a rien ici non plus :)

Bon ok... amusez-vous avec la minilibx ou prenez de l'avance sur les prochains TPs :)

Au menu :

- `aff_colormap` : affiche de jolis dégradés de couleurs
- `aff_ligne` : affiche une ligne
- `aff_cercle` : affiche un cercle
- ...



Bien sur, c'est à vous de trouver les algos pour tracer les lignes et les cercles. Ils seront vu en cours d'infographie un peu plus tard.