TP #01 - MinilibX et evenements

majdi.toumi@epitech.eu
astek_resp@epitech.org

Abstract: L'objectif de ce TP et d'apprendre a manipuler les differentes fonctions de base de la minilibX

Table des matières

.1 Pre-requis

Pour mener a bien ce TP vous devez :

- Avoir vu le cours qui porte sur ce TP.
- Savoir faire un bon Makefile... a la norme.

`Le TP#0 de la journee detente de la piscine C est lui aussi plus que necessaire pour reussir ce TP.`

.2 Connexion et fenetre

Utilisons tout d'abord un exemple simple de connexion et d'ouverture de fenetere :

```
1  int         main()
2  {
3    void      *mlx_ptr;
4    void      *win_ptr;
5
6    mlx_ptr = mlx_init();
7    win_ptr = mlx_new_window(mlx_ptr, 500, 500, "My first ↩
       window");
8    return (0);
9  }
```

Reportez-vous aux man de la minilibX ou bien au cours pour le detail des fonctions mlx_init et mlx_new_window

Par exemple: "man -M /u/prof/astek/public/igraph/man mlx"

Voici les pages de man existantes: mlx_loop, mlx_new_image, mlx_new_window et mlx_pixel_put

Nommez ce programme **tp1_windows_empty** et compilez-le.

N'oubliez-pas les parametres de compilation vus durant le jour de detente en piscine!

Que se passe-t-il ?
La fenetre n'est pas permanente, elle disparait aussitot.
Rajoutez alors dans votre code (au bon endroit...) une boucle infinie pour que votre programme ne s'arrete pas.

La fenetre reste alors a l'ecran. Le serveur X est dresse pour faire le menage des fenetres quand un programme s'arrete.

Modifiez votre programme pour obtenir a l'ecran une fenetre plus grande.

.3 Les pixels

.3.1 Un peu...

Nous utilisons desormais la fonction mlx_pixel_put.

Creez un nouveau programme a partir du precedent et nommez-le **tp1_windows_color**.
Rajoutez la ligne suivante a ce nouveau programme :

```
1  mlx_pixel_put(mlx_ptr, win_ptr, 250, 250, 0xFFFFFF);
```

Mettez-la au bon endroit ;)
Compilez-le tout et executez le programme.
Une fenetre doit s'afficher a l'ecran, avec un pixel blanc au milieu.

.3.2 Beaucoup...

Modifiez votre programme afin d'obtenir la couleur dans toute la fenetre.

.3.3 Passionnement !

Creez un nouveau programme **tp1_print_rectangle** qui remplis un rectangle au milieu de la fenetre, avec une certaine couleur.

Choisissez n'importe quelle position, taille et couleur du rectangle dans un premier temps. Ensuite, faites en sorte que ces 5 parametres soient specifies en ligne de commande.

```
1  $> ./tp1_print_rectangle 100 100 50 50 255
```

Vous pouvez aussi avoir la couleur exprimee en hexadecimal

```
1  $> ./tp1_print_rectangle 100 100 50 50 FF
```

my_getnbr_base for the win :p

.4 Les evenements

.4.1 Affichage

Creez un programme **tp1_print_event** qui affiche l'arrivee d'un evennement, au moyen de my_putstr.

N'oubliez-pas le principe de base vu dans le cours :

- Initialisation des evenements :
  3 fonctions servent a dire a la minilibX : "Laquelle de mes fonctions doit etre executee par la minilibX, lorsqu'il y a un evenement ?"
  Cette operation n'est faite qu'une seule fois, pour chacun des trois types d'evenements, grace aux fonctions mlx_key_hook, mlx_mouse_hook et mlx_expose_hook.

- Boucle d'evenements :
  La fonction mlx_loop, qui est une boucle infinie (on ne sort donc jamais de cette fonction) a pour mission d'attendre les evennements et d'executer la fonction de votre choix.

Pour le moment, transmettez 0 comme 3eme parametre pour les fonctions hook.

Faites une premiere version de **tp1_print_event** qui fonctionne pour l'evenement expose. Des l'execution du programme, vous devez avoir quelque chose qui s'affiche autre que "segmentation fault" :)

Faites ensuite une seconde version qui affiche les 3 types d'evenements, et enfin, les 3 types d'evenements avec les details : le numero de touche du clavier, ou bien le bouton et la position de la souris.

.4.2 Action

Nous allons associer une action a un evenement :

Modifiez votre programme **tp1_print_event** de sorte qu'il s'arrete lorsque qu'une touche du clavier est appuyee.

Refaites le meme exercice mais avec un clique de souris.

Encore un fois, mais c'est le bouton du milieu de la souris qui permet de quitter le programme, pas les autres.

Modifiez une quatriemme fois votre programme pour faire en sorte que seule la touche ESC termine le programme (comment savoir quel numero correspond a la touche ESC, hein ? bonne question !).

.4.3 Dessin

Realisez un programme **tp1_click_pixel** qui affiche un pixel blanc a l'endroit de la souris lorsque l'on clique sur le bouton droit.

Pour cela, faites les modifications spontannees qui vous viennent a l'esprit, et tentez de compiler votre programme. Il est probable que cela ce passe mal :)

Regardez le type d'erreur que vous obtenez, cela devrait vous indiquer qu'il vous manque certaines informations dans la fonction apellee lors d'un evenement.

Il semblerait que le parametre mysterieux void *param ait un role a jouer.

Verifiez que ce qui est transmis comme void *param a une fonction hook, est bien recupere dans la fonction que vous associez a un evenement (par ex. mettez 42 a la place de 0 et voyez si vous obtenez bien 42 dans le parametre correspondant de la fonction appelee pour l'evenement).

Et si au lieu de mettre 0 (ou 42) dans ce parametre on mettait une information utile ?

```
Les globales sont hors norme!
```

.4.4 Expose

Un des evenements a ne pas negliger est l'expose !

Vous pouvez etre tres souvent ammene a devoir re-afficher votre fenetre.

Des maintenant prenez spontannement l'habitude de gerer l'expose dans tous vos programmes avec la minilibX.

Creez un programme **tp1_print_expose** qui affiche un rectangle dans la fenetre (ca vous l'avez deja fait normalement), et qui le re-affiche si un evenement expose arrive.

.5 Les couleurs

Nous allons faire un petit retour sur les couleurs.

Creez un programme **tp1_print_cmap** afin d'obtenir un affichage de la colormap.
La fenetre est remplie de pixels de couleurs, avec le rouge en haut a gauche, le vert en bas au milieu, et le bleu en haut a droite.
Il y a de jolis degrades de couleurs entre ces 3 positions principales.

Vous pouvez commencer dans un premier temps, par une version plus facile : creez une fenetre de 256x256 pixels.

Ensuite, chercher a faire votre colormap avec une taille quelconque de fenetre (par exemple 420x420, 100x100, ...)

Comment faire pour mettre la couleur qu'on veut ?

Commencez par relire le TP de la journee detente de la piscine ; Il y a tout un blabla sur ces probleme de couleur.