

TP 5

Utilisation de la bibliothèque graphique MLV

L'objet de ce TP est d'utiliser la librairie MLV pour dessiner et interagir avec votre programme à l'aide de la souris et du clavier.

Des programmes de démonstration sont disponibles dans le répertoire `/usr/share/doc/mlv/examples`. Recopier ces exemples par la commande `cp -r /usr/share/doc/mlv/examples .`

Les différents programmes proposés sont classés en trois catégories : `beginner` (débutant), `medium` (moyen) et `advanced` (avancé).

Pour ce TP, vous aurez uniquement besoin d'utiliser les fonctions proposées dans la catégorie `beginner`.

Pour compiler les programmes de la catégorie `beginner`, placez vous dans le répertoire de compilation en tapant la commande `cd examples/beginner`, puis tapez la commande: `make`.

Pour compiler votre programme `demo.c`, recopier le `Makefile` situé dans le répertoire `/usr/share/doc/mlv/examples/beginner/Makefile` dans le le répertoire où se trouve `demo.c`. Modifier le fichier `Makefile` ainsi copié en remplaçant la ligne `TARGET=...` par:

```
TARGET = demo
```

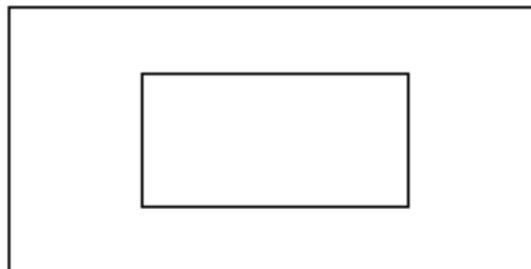
Les commentaires de chacune des démonstrations constituent une documentation des fonctions de la librairie MLV. Vous devez donc lire les différents codes sources présents dans le répertoire `beginner` en recherchant les fonctions utilisables.

Une documentation plus complète et détaillée peut être trouvée en tapant dans la commande : `man MLV_all.h`.

Si vous souhaitez installer cette librairie chez vous, vous pouvez vous reporter à la documentation présente dans le dossier `/usr/share/doc/mlv/index.html` ou vous rendre sur le site <http://www-igm.univ-mlv.fr/~boussica/mlv/index.html>.

1. Rectangle

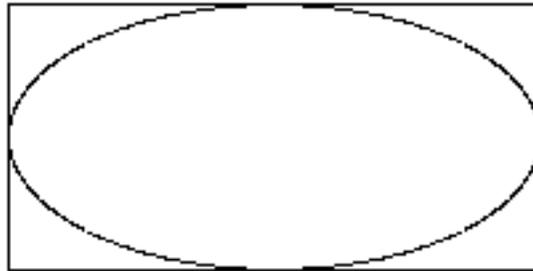
- Écrire un premier programme qui ouvre une fenêtre de largeur 400 pixels et de hauteur 300 pixels.
- Écrire le programme `Rectangle` dessinant un rectangle dans une fenêtre dont les dimensions ont été fournies par l'utilisateur. Les dimensions du rectangle seront exactement la moitié de celle de la fenêtre et le rectangle devra être centré dans la fenêtre. Par exemple, on aimerait avoir :



- Modifier le programme `Rectangle` afin d'afficher un rectangle rempli de couleur bleue.

2. Ellipse

- Écrire le programme `Ellipse` dessinant la plus grande ellipse inscrite dans une fenêtre dont les dimensions ont été fournies par l'utilisateur. Par exemple, on aimerait avoir :



- Écrire le programme `Cercle` dessinant le plus grand cercle inscrit dans une fenêtre dont les dimensions ont été fournies par l'utilisateur.

3. Cercle

Écrire un programme `Cercle_epais` permettant de dessiner un cercle dont le centre (x,y) , le rayon, la couleur et l'épaisseur du trait (exprimé en points) peuvent être donnés par l'utilisateur. Deux techniques sont envisageables: discuter les avantages et inconvénients de chacune d'elles.

4. Animation

Écrire le programme `Cercle_anim` affichant un cercle centré dans la fenêtre et dont le rayon croît selon un pas (en nombre de pixels) donné par l'utilisateur. L'affichage du cercle est stoppé lorsque le cercle atteint la plus grande dimension (largeur ou hauteur) de la fenêtre. Ces dimensions ont été préalablement fournies par l'utilisateur.

5. Saisie de points avec la souris

Écrire le programme `Polygone` permettant à l'utilisateur de dessiner un polygone dans une fenêtre en cliquant dans la fenêtre pour indiquer la position des n sommets. Le nombre n sera préalablement fourni par l'utilisateur et permettra de générer un polygone de $(n-1)$ arêtes.

6. Transformer les programmes précédents en fonction callable à partir d'un menu