



# Initiation à la Programmation C

## Travaux Pratiques - L2

### Utilisation de tableaux



---

Dans cette séance de travaux pratique, nous abordons les points suivants :

- l'utilisation de la librairie 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.

#### ► Exercice 0. Prise en main de la librairie MLV

Des programmes de démonstration de la librairie MLV sont disponibles dans le répertoire

```
/usr/share/doc/libmlv3-dev/doc/mlv/examples
```

1. Recopier ces exemples par la commande :

```
cp -r /usr/share/doc/libmlv3-dev/doc/mlv/examples .
```

Le `.` final de la commande précédente (**à ne pas oublier !**) désigne votre répertoire; la commande copie donc le répertoire `/usr/share/doc/mlv/examples` et tous ses sous répertoires (en raison du `-r`, `r` pour récursivement) dans le répertoire d'où vous avez lancé la commande.

2. Les différents programmes proposés sont classés en trois catégories :

- **beginner** (débutant)
- **medium** (moyen)
- **advanced** (avancé).

Pour ce TP, vous n'aurez besoin que d'utiliser les fonctions proposées dans la catégorie **beginner**.

Pour compiler l'ensemble des programmes de la catégorie **beginner**, placez vous dans le répertoire de compilation en tapant la commande :

```
cd beginner
```

puis tapez la commande

```
make
```

3. Pour compiler un seul programme, `demo.c`, utilisez la commande de compilation :

```
clang -std=c17 -Wall demo.c -o demo -lMLV
```

4. 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`.

**Remarque 1 :** Si vous travaillez sous **Linux** et que vous souhaitez installer cette librairie chez vous, vous pouvez vous reporter à la documentation présente dans le dossier `/usr/share/doc/index.html` ou vous rendre sur le site <http://www-igm.univ-mlv.fr/~boussica/mlv>.

**Remarque 2 :** De nombreuses personnes très douées ont déjà essayé ces dernières années d'installer la librairie sous Mac ou sous Windows...

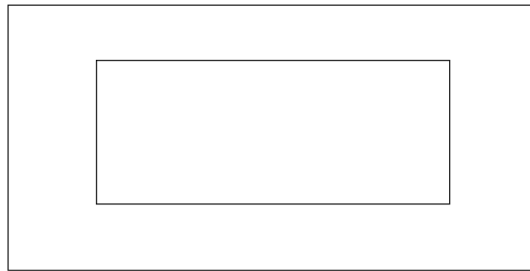
**Sans succès !**

S'il vous plaît, ne perdez pas votre temps à retenter cela ; utilisez la machine virtuelle disponible sur le **elearning**.

### ► Exercice 1. Rectangle

1. Écrire un premier programme qui ouvre une fenêtre de largeur 400 pixels et de hauteur 300 pixels.
2. 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 :

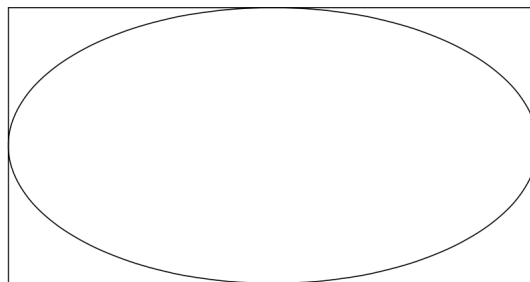


3. Modifier le programme **Rectangle** afin d'afficher un rectangle rempli de couleur bleue.

### ► Exercice 2. Ellipse

1. 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 :



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

► **Exercice 3. Cercle epais**

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.

► **Exercice 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.

► **Exercice 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  $(n)$  arêtes.

► **Exercice 6. Menu**

Transformer les programmes précédents en fonctions et écrire un menu permettant de les choisir.