

Installation de la bibliothèque graphique libgfl{2|3}d

libgfl est une bibliothèque graphique, écrite en C, basée sur [OpenGL](#) et [freeglut](#).

- elle est conçue pour fonctionner sur systèmes Unix-like (donc tous les linux)
- elle peut éventuellement s'installer sur MacOS moyennant quelques adaptations du script d'installation
- elle peut s'installer sur Windows.10+, via un émulateur linux comme [WSL](#)

Archive d'installation

⇒ Récupérer et décompresser l'archive (de préférence dans votre répertoire racine ~/)

- via eLearning, sur la page du cours qui vous concerne
- via les machines de l'université : `$> cp /home/ens/~/incerti/KITLIBGFL.zip ./`

```
xterm
$> mv KITLIBGFL.zip ~/
$> unzip KITLIBGFL.zip
$> cd KITLIBGFL
```

attention : l'installation va créer ou modifier (sans en affecter le fonctionnement) le fichier de configuration `~/.bashrc` pour y inscrire les chemins vers les éléments de la libgfl.

🔊 Il faut prendre garde aux deux points suivants :

- les systèmes "Unix-like" détestent les espaces et accents dans les noms de fichier ou de répertoire.
⇒ **NE PAS faire l'install dans un répertoire comme 'Mes Documents' ou 'Téléchargements' ou 'Bureau' ça ne fonctionnerait pas.**
- des variables d'environnement sont créées pour la compilation. Dans le cas d'une **installation 'locale'**, ces variables (`$incGFL` et `$libGFL`) sont définies par rapport au répertoire courant d'installation (par défaut `~/KITLIBGFL/`)
⇒ en cas de **déplacement / renommage** du répertoire, il faut **recommencer l'install.**

Contenu

- deux documents .pdf.
- un script d'installation : `loc_install.sh` pour une installation 'locale' et partielle, sans besoin des droits d'administrateur,
- un répertoire `gfl/` contenant les fichiers utiles de la lib.
 - un répertoire `include/` : les fichiers d'en-tête de la lib. (`gfl_*.h`)
 - un répertoire `src/` : les fichiers source de la lib. (`gfl_*.c`)
 - un répertoire `bin/` (vide) : contiendra les différentes versions compilées (`libgfl*.a|so`)
 - un `Makefile`
- deux répertoires `2Ddemo/` et `3Ddemo/`
 - un répertoire `include/` (vide)
 - un répertoire `src/` contenant quelques codes d'exemple
 - un `Makefile`

Installation locale (sans droits admin)

→ par exemple sur les machines `linux` de l'Université

→ suppose que `freeglut3-dev` est déjà correctement installé sur le système, ainsi que les outils externes `Netpbm` (gestion d'image) et `mencoder` (video)

- dans le répertoire courant, lancer le script d'installation : `$> sh ./install_local.sh`
- vous devriez obtenir à la fin 16 fichiers binaires dans le répertoire `gfl/bin/` :

<code>libgfl{2 3}d.{a so}</code>	: version par défaut (C)
<code>libgfl{2 3}d.gdb.{a so}</code>	: même chose, avec option <code>-g</code>
<code>libgfl{2 3}d++.a so}</code>	: version compatible (C++)
<code>libgfl{2 3}d++.gdb.{a so}</code>	: même chose, avec option <code>-g</code>

La sélection de la version adéquate sera faite automatiquement par le `Makefile` (cf. `./demo/`) selon les options de compilation :

- `GDB=0/1` pour (dés-)activer le mode 'debug' (utilisable avec `gdb`),
- `CPP=0/1` pour lier la lib à des modules `C` ou `C++`

remarque : les compilateurs `g++` et/ou `clang++` vont produire beaucoup de `warnings` portant sur des chaînes de caractères. Ils peuvent être ignorés.

Exemples d'utilisation

Une fois les lib. installées, descendre dans le répertoire `{2|3}Ddemo/` et lancer la compilation des quelques exemples proposés `$> make`

Tous ces exemples sont d'un intérêt très limité mais sont là pour illustrer ce que l'on peut faire et comment le faire. Chacun illustre un type d'usage particulier. Ils peuvent (doivent!) servir de base pour d'autres développements.

- `src/gfl{2|3}_REFERENCE.c` : c'est le plus important, à bien conserver sans le modifier. Cet exemple regroupe la plupart des fonctionnalités utiles.
- `src/gfl{2|3}_squelette.c` : ne fait rien d'autre qu'ouvrir une fenêtre vide mais servira de point de départ à tous vos programmes.
- le fichier de compilation `Makefile` est préformaté pour utiliser la config. d'installation choisie.
Ne le modifiez pas, sauf si vous savez exactement ce que vous faites.
- la commande `$> make ?` affiche les paramètres de compilation

Il faut ensuite comprendre la philosophie de fonctionnement de cette lib. et des quelques règles d'usage à respecter 📖 cf document `tuto-libgfl.pdf` présent dans l'archive.