Université Gustave Eiffel

INSTITUT D'ÉLECTRONIQUE ET D'INFORMATIQUE GASPARD-MONGE

Installation de la bibliothèque graphique `libg2x` (version 6e-23)

`libg2x` *est une bibliothèque graphique, écrite en* `C` *, basée sur* `OpenGl` *et* `freeglut` [a].

- *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

[a] (alternative à `GLU (GL-Utility)` et `GLUT (GL-Utility-Toolkit)`)

Archive d'installation

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

```
----------------------------
$> cd
$> unzip KITLIBG2X.zip
$> cd KITLIBG2X/
$> ls -l
----------------------------
```

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 `libg2x` .

☞ Il faut prendre garde aux deux points suivants :

- les systèmes `"Unix-like"` détestent les espaces dans les noms de fichier ou de répertoire.
  ⇒ **NE PAS** faire l'install dans un répertoire comme `'Mes Documents'` ça ne fonctionnerait pas.
- des variables d'environnement sont crées pour la compilation. Dans le cas d'une **installation 'locale'**, ces variables (`$incG2X` et `$libG2X`) sont definies par rapport au répertoire courant d'installation (par défaut `~/KITLIBG2X/`)
  ⇒ en cas de **déplacement** / **renommage** du répertoire, il faut **recommencer** l'install.

Contenu

- un fichier `README` (analogue à ce document).
- deux scripts d'installation : `install-local.sh` et `install-standard.sh`
  - `install-local.sh` pour une installation 'locale' et partielle, sans besoin des droits d'administrateur,
  - `install-standard.sh` pour une installation 'standard' dans `/usr/` .
    → droits administrateur (`sudo`) nécessaires.
    → cette version installe également la suite `Netpbm` (image) et `mencoder` (video).
- un répertoire `include/` contenant les fichiers d'en-tête de la lib.
- un répertoire `src/` contenant les fichiers source de la lib.
- un `Makefile`
- un répertoire `work/`
  - un répertoire `include/` (vide)
  - un répertoire `src/` 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 devez obtenir à la fin 8 fichiers binaires :
  - `libg2x.6e-23.a / libg2x.6e-23.so`
  - `libg2x.6e-23.dgb.a / libg2x.6e-23.dbg.so`
  - `libg2x.6e-23++.a / libg2x.6e-23++.so`
  - `libg2x.6e-23++.dbg.a / libg2x.6e-23++.dbg.so`

  La sélection de la version adéquate sera faite automatiquement par le `Makefile` (cf. `./work/`) selon les options de compilation :
  - `DEBUG=0/1` pour (dés-)activer le mode 'debug' (`gdb`),
  - `CPP=0/1` pour lier la lib à des modules `C` ou `C++`

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

Installation Standard (/usr/lib)

→ par exemple sur votre `PC-linux` personnel, comme une bibliothèque standard
→ **nécessite de disposer des droits admin (`sudo`)**

- dans le répertoire courant, lancer le script d'installation : `$> sh ./install_standart.sh`
  - `sudo apt-get install freeglut3-dev` : installe la lib `freeglut3-dev`, si elle est absente

    attention : le script d'installation présuppose que les fichiers de `freeglut3` (`libGL.so`, `libGLU.so`, `libglut.so`) seront installés dans le répertoire `/usr/lib/x86_64-linux-gnu/`

    → pour vérifier : `$>whereis libGL.so libGLU.so libglut.so`

    Si ce n'est pas le cas, il faudra ajuster le fichier `./install_standard.sh` en conséquence.
  - `sudo apt-get install netpbm` : installe (si absents) les outils de la suite `Netpnm`
  - `sudo apt-get install mencoder` : installe (si absente) la suite `mencoder`
  - lignes suivantes : création des répertoires, compilation(s) et export des fichiers dans `/usr/`
    - copie les fichiers `include/g2x*.h` dans `/usr/include/g2x/`
    - crée ou ajoute dans le fichier de configuration `~/.bashrc` les chemins vers les répertoires `/usr/lib/g2x/` et `/usr/include/g2x/` et crée les variables d'environnement `$incG2X` et `$libG2X` utilisées dans les fichiers `Makefile` (cf. `./work/Makefile`)
    - lance la compilation de la lib et produit 8 fichiers binaires :
      `libg2x.6e-23.a / libg2x.6e-23.so`
      `libg2x.6e-23.dbg.a / libg2x.6e-23.dbg.so`
      `libg2x.6e-23++.a / libg2x.6e-23++.so`
      `libg2x.6e-23++.dbg.a / libg2x.6e-23++.dgg.so`

      La sélection de la version adéquate sera faite automatiquement par le `Makefile` (cf. `./work/`) selon les options de compilation :
      - `DEBUG=0/1` pour (dés-)activer le mode 'debug' (`gdb`),
      - `CPP=0/1` pour lier la lib à des modules `C` ou `C++`

      remarque : les compilateurs `g++` et `clang++` vont produire beaucoup de `warnings` portant sur des chaînes de caractères. Ils peuvent être ignorés
    - déplace les binaires `libg2x.6e.*.[so/a]` dans le répertoire `/usr/lib/g2x/`

Exemples d'utilisation

Une fois les lib. installées, descendre dans le répertoire `work/` et lancer la compilation des exemples

```
  ---------------------------
$> make
$> ls g2x*
   g2x_00_squelette  g2x_03_baballe       g2x_06_hsvcolor
   g2x_01_control    g2x_04_drawfunction  g2x_REFERENCE
   g2x_01_cercle     g2x_05_hsvflower
  ---------------------------
```

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/g2x_REFERENCE.c` : c'est le plus important, à bien conserver sans le modifier. Cet exemple regroupe la plupart des fonctionalités utiles.
- `src/g2x_00_squelette.c` : ne fait rien d'autre qu'ouvrir une fenêtre vide mais servira de point de départ à tous vos programmes.
- fichier de compilation `Makefile` 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.