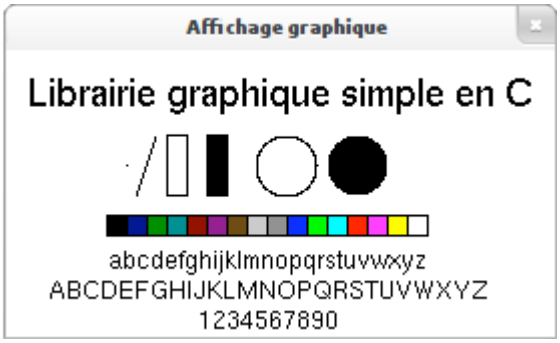
Université d'Aix-Marseille - Faculté des Sciences

I4 - Informatique

Librairie Graphique Simple en C

André Abrame



Ce document présente une librairie graphique simple basée sur les bibliothèques *openGL*[1] et *freeGLUT*[2]. Elle permet l'affichage de figures graphiques simples (pixel, ligne, rectangle et cercle) et de texte.

1 Installation et utilisation

1.0.1 Installation de freeglut

Pour que la librairie graphique puisse fonctionner correctement, il faut que *freeglut* soit installé sur la machine. Cela devrait normalement être le cas sur vos machine de TP. Si vous voulez utiliser la librairie graphique sur vos machines personnelles, vous pouvez installer *freeglut* comme suit :

- **sur Ubuntu via le gestionnaire de paquets**
  Dans le terminal, tapez :

1. `http://www.opengl.org/`
2. `http://freeglut.sourceforge.net/`

```
sudo apt-get install freeglut3 freeglut3-dev
```

- **sur un système GNU/Linux à partir des sources**
  Récupérez le code source de freeglut sur le site `http://freeglut.sourceforge.net/`, puis décompressez l'archive que vous avez téléchargée. Ouvrez un terminal dans le répertoire contenant le code source, puis tapez :

  ```
  ./configure
  make
  sudo make install
  ```

- **sur tout autre système ou si vous rencontrez des difficultés**
  Demandez de l'aide à vos enseignants !

1.0.2 utilisation de la librairie

Placez les trois fichiers libgraphique.h, libgraphique.c et libgraphique_fonts.h dans le même dossier que vos fichiers sources. Au début de chacun de vos fichiers sources utilisant les fonctions de la librairie graphique, placez la directive :

```
#include "libgraphique.h"
```

À la compilation, vous devez ajouter à la liste de vos fichiers sources à compiler le fichier libgraphique.a et passer au compilateur les options suivantes : -lglut -lGLU -lGL -lm. Ainsi, pour compiler un fichier exemple.c utilisant la librairie graphique, on écrira :

```
gcc -o exemple exemple.c libgraphique.c -lglut -lGLU -lGL -lm
```

2 Démarrage et arrêt du mode graphique

Avant de pouvoir utiliser les fonctions du mode graphique, celui-ci doit être démarré par la fonction :

```
void start_graphics();
```

Cette fonction ouvre une nouvelle fenêtre, composée d'une multitude de points appelés "pixels" (contraction de *picture element*). Une ligne contient 640 pixels et une colonne 480 pixels. L'écran est doté d'un repère cartésien.

Le mode graphique peut être arrêté par la fonction :

```
void stop_graphics();
```

Lorsque le mode graphique est actif, les fonctions d'affichage et de saisie de caractère (comme printf() et scanf()) fonctionnent mais n'ont d'impact que sur la fenêtre du terminal, et pas sur la fenêtre graphique. Nous verrons par la suite comment dessiner et récupérer les touches saisies par l'utilisateur dans la fenêtre graphique.

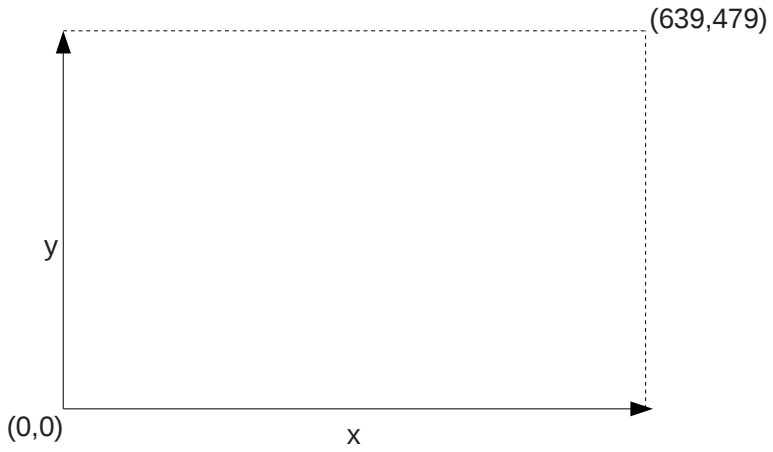


FIGURE 1 – L'écran et son repère en mode graphique.

3 Les commandes d'affichage

Le mode graphique permet d'afficher des figures simples (pixel, ligne, rectangle et cercle), d'afficher des chaînes de caractères et de colorier des surfaces. Les fonctions de dessin ne sont pas directement appliquées à l'écran. Il faut, pour les faires apparaîtres, utiliser la fonction :

```
void update_graphics();
```

3.1 Primitives de dessin

Pour dessiner un point (un pixel) aux coordonnées (x,y) :

```
void draw_pixel(int x, int y);
```

Pour dessiner une ligne entre les points de coordonnées (x1,y1) et (x2,y2) :

```
void draw_line(int x1, int y1, int x2, int y2);
```

Pour dessiner un rectangle dont le coin supérieur gauche (resp. inférieur droit) a pour coordonnées (x1,y1) (resp. (x2,y2)) :

```
void draw_rectangle(int x1, int y1, int x2, int y2);
```

Et pour un rectangle plein :

```
void draw_rectangle_full(int x1, int y1, int x2, int y2);
```

Pour dessiner un cercle de centre (x, y) et de rayon r :

```
void draw_circle(int x, int y, int r);
```

Et pour un cercle plein :

```
void draw_circle_full(int x, int y, int r);
```

Pour colorier une surface telle que le point (x, y) appartient à cette surface (et pas à son coutour) et qui est délimitée par des pixels de couleur c :

```
void fill_surface(int x, int y, couleurs c);
```

Pour effacer l'écran :

```
void clear_screen();
```

3.2 Primitives d'affichage de texte

Pour afficher une chaîne de caractères simple s du texte sur la fenêtre graphique aux coordonnées (x, y), on utilisera la fonction :

```
void draw_string(int x, int y, char *s);
```

Pour afficher une chaîne de caractères utilisant les codes de remplacement de la famille de fonctions printf() aux coordonnées (x, y), on utilisera la fonction :

```
void draw_printf(int x, int y, const char *format, ...);
```

Attention, ces deux fonctions ne peuvent afficher que des caractères ASCII (c'est-à-dire pas de caractère accentué).

La police d'affichage et la taille des caractères peuvent être définies par la fonction suivante :

```
void set_font(polices p);
```

Les valeurs possibles de p sont disponibles dans la table 1.

Valeurs
font_HELVETICA_10
font_HELVETICA_12
font_HELVETICA_18

TABLE 1 – Valeurs possibles du type *polices*

3.3 Couleurs

Pour choisir la couleur du prochain tracé (figure ou texte) :

```
void set_drawing_color(couleurs c);
```

Pour choisir la couleur du fond de l'écran :

```
void set_background_color(couleurs c);
```

Pour choisir la couleur de remplissage des surfaces :

```
void set_fill_color(couleurs c);
```

Les valeurs possibles des arguments de ces trois fonctions sont présentées dans la Table 2.

Couleurs	**Valeurs**	**Couleurs**	**Valeurs**
Noir	color_BLACK	Gris foncé	color_DARKGRAY
Bleu	color_BLUE	Bleu clair	color_LIGHTBLUE
Vert	color_GREEN	Vert clair	color_LIGHTGREEN
Cyan	color_CYAN	Cyan clair	color_LIGHTCYAN
Rouge	color_RED	Rouge clair	color_LIGHTRED
Magenta	color_MAGENTA	Magenta clair	color_LIGHTMAGENTA
Marron	color_BROWN	Jaune	color_YELLOW
Gris clair	color_LIGHTGRAY	Blanc	color_WHITE

TABLE 2 – Valeurs possibles du type *couleurs*.

4 La gestion du clavier

Lorsque la fenêtre graphique est active, la fonction suivante permet de récupérer les touches saisies par l'utilisateur :

```
int get_key();
```

Cette fonction renvoie le code du caractère associé à la touche (par exemple 'a' pour la touche *a*, 'A' pour *Majuscule* + *a* et ainsi de suite). Pour les touches spéciales (*Echap*, *Entrée* ...), les codes renvoyés sont présentés dans la table 3. Cette fonction est bloquante, c'est-à-dire que si aucune touche n'a été saisie depuis le dernier appel de la fonction celle-ci attend une saisie avant de renvoyer une valeur.

key_F1	key_F2	key_F3
key_F4	key_F5	key_F6
key_F7	key_F8	key_F9
key_F10	key_F11	key_F12
key_PAGE_UP	key_PAGE_DOWN	key_HOME
key_END	key_INSERT	key_LEFT
key_RIGHT	key_UP	key_DOWN
key_ESCAPE		

TABLE 3 – Identifiants des touches spéciales.