

3/26/20 S. Lanquetin – L3 Info, Synthèse d'images 9

# GLUT

---

## OpenGL

9

3/26/20 S. Lanquetin – L3 Info, Synthèse d'images 10

## Application GLUT

Session graphique contrôlée uniquement par des événements	Opérations de tracé uniquement dans la fonction d'affichage	Boucle d'événement = dernière action. Réactualise l'affichage, capte les interactions.
Le système déclenche la fonction d'affichage.	Exécution de la fonction d'affichage avec <code>glutPostRedisplay()</code>	Utilisation de variables globales d'état pour contrôler l'affichage en fonction des interactions.

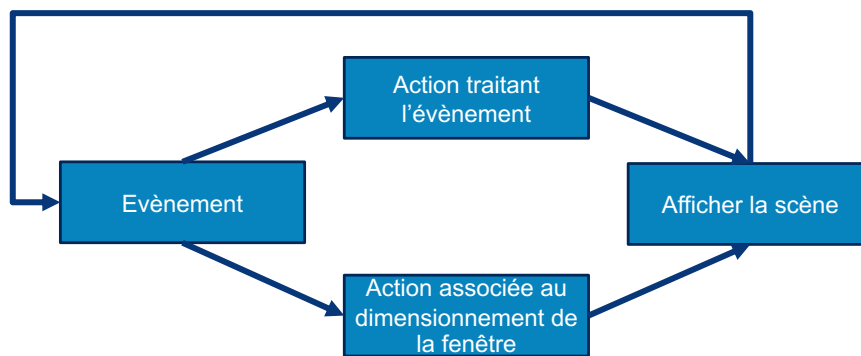
10

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

11

## Fonctionnement



11

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

12

## Structure d'une application GLUT

- Actions du main :
  - initialisation du fenêtrage puis création de la fenêtre
  - désignation de la fonction d'affichage dans la fenêtre courante,
  - désignation de la fonction déclenchée par un redimensionnant la fenêtre courante,
  - association d'une fonction pour chaque type d'interruption,
  - la boucle d'événements.

12

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

13

```

void main(int argc, char** argv)
{
    /*Initialisation d'une session GLUT*/
    glutInit(&argc,argv); //initialise la bibliothèque GLUT
    glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH | GLUT_DOUBLE);
    glutInitWindowSize(500,500); //Taille fenêtre
    glutInitWindowPosition(200,200); //Position fenêtre
    glutCreateWindow(argv[0]); //Création fenêtre
    /*Initialisation de paramètres éventuels*/
    Init();
    /*Enregistrement des fonctions de rappel*/
    /*Fonctions définissant la scène 3D et sa projection*/
    glutDisplayFunc(affichage);
    glutReshapeFunc(refenetrer);
    /*Fonctions liées aux interactions*/
    glutKeyboardFunc(clavier);
    glutMouseFunc(mouse);
    glutSpecialFunc(clavierSpecial);
    /*Boucle d'événements*/
    glutMainLoop(); //lancement de la boucle principale
    return (0);
}

```

13

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

14

## Structure d'une application GLUT

```

void affichage(){
    « effacer l'écran »
    positionner la caméra
    construction (tracé) de la scène
    glFlush();
    glutSwapBuffers(); //On échange les buffers
}

void refenetrer(int x,int y){
    redéfinition du cadre d'affichage après
    redimensionnement de la fenêtre
    définition de la projection 3D->2D
}

void Init(){
    initialisation éventuelle de paramètres propres à
    l'application (éclairage...)
}

```

14

## Structure d'une application GLUT

```
void clavier(unsigned char touche,int x,int y){
    modification du contexte d'affichage sur un événement
    clavier
    switch (touche)...
    glutPostRedisplay() ;
}

void mouse(int button, int state,int x,int y){
    modification du contexte d'affichage sur un événement
    souris
    glutPostRedisplay() ;
}

void clavierSpecial(unsigned char key, int x, int y){
    action déclenchée sur une touche F1, ..., F10, flèches
    glutPostRedisplay() ;
}
```

15

## GLUT

- Initialisation de GLUT

**void glutInit(int \*arg, char \*\*argv)**

➡ initialise GLUT

- Boucle de traitement des événements

**void glutMainLoop(void)**

- ➡ dernière chose à faire
- ➡ toutes les fenêtres qui ont été créées sont maintenant affichées
- ➡ le rendu dans ces fenêtres effectué

16

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

17

## GLUT : Gestion des fenêtres

### **void glutInitDisplayMode(unsigned int mode)**

- ➡ Modèle chromatique RGBA
- Fenêtre en double tampon
- Tampon de profondeur


### **void glutInitWindowPosition (int x, int y)**

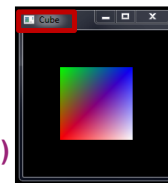
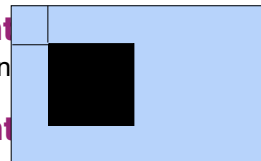
- ➡ coordonnées écran du coin haut gauche de la fenêtre

### **void glutInitWindowSize(int width, int height)**

- ➡ taille en pixels de la fenêtre

### **void glutCreateWindow(char \*string)**

- ➡ crée une fenêtre avec un nom de type string,
- ➡  Fenêtre affichée à l'appel de **glutMainLoop()**

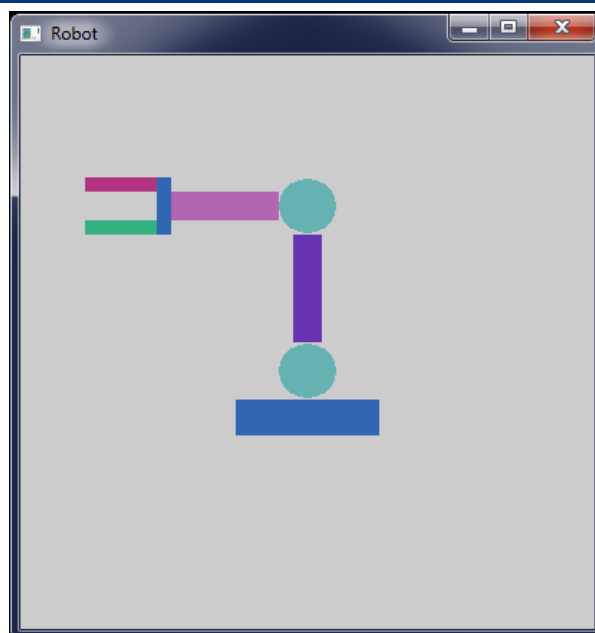


17

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

18



18

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

19

## GLUT : Gestion des fenêtres

### **void glutDestroyWindow (int win);**

- ➡ détruit la fenêtre identifiée par le paramètre win

### **void glutPostRedisplay(void)**

- ➡ Réaffichage de la fenêtre

### **void glutSwapBuffers(void)**

- ➡ Échange des tampons et appel de **glFlush()** quand mode = double tampon

### **void glutFullScreen (void)**

- ➡ Change la taille de la fenêtre en plein écran

19

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

20

## GLUT : Gestion du curseur

### **void glutSetCursor(int cursor);**

- ➡ Modifier/cacher le curseur du pointeur de la souris
- ➡ cursor : GLUT\_CURSOR\_NONE, GLUT\_CURSOR\_CYCLE, GLUT\_CURSOR\_SPRAY, GLUT\_CURSOR\_WAIT, GLUT\_CURSOR\_TEXT, ...

### **void glutWarpPointer ( int x , int y)**

- ➡ Positionner le pointeur de la souris en (x, y)

20

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

21

## GLUT : Gestion du curseur

### **void glutSetCursor(int cursor);**

- ➡ Modifier/cacher le curseur du pointeur de la souris
- ➡ cursor : GLUT\_CURSOR\_NONE, GLUT\_CURSOR\_CYCLE, GLUT\_CURSOR\_SPRAY, GLUT\_CURSOR\_WAIT, GLUT\_CURSOR\_TEXT, ...

### **void glutWarpPointer ( int x , int y)**

- ➡ Positionner le pointeur de la souris en (x, y)

21

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

22

## GLUT : Touches modificatrices

### **void glutGetModifiers()**

- ➡ Permet de connaître l'état d'une touche modificatrice
- ➡ GLUT\_ACTIVE\_SHIFT, GLUT\_ACTIVE\_CTRL, GLUT\_ACTIVE\_ALT

22

## GLUT : Fonctions de rappel

### **glutDisplayFunc(affichage);**

- ➔ événement le plus important, actualisation d'affichage
- ➔ Appel de la fonction d'affichage

### **void affichage(){...}**

- ➔ Contient toutes les routines nécessaires à la reconstitution de la scène

### **glutReshapeFunc (refenetrer);**

- ➔ action à exécuter quand la fenêtre est redimensionnée

### **void refenetrer (int w, int h){...}**

- ➔ w et h sont les dimensions de la nouvelle fenêtre

23

## GLUT : Fonctions de rappel clavier

### **glutKeyboardFunc(clavier);**

- ➔ associer une routine à une touche ASCII du clavier

### **void clavier(unsigned char key, int x, int y){...}**

- ➔ key correspond au code ASCII de la touche
- ➔ x et y : coordonnées de la souris lors de l'appui sur la touche
- ➔ [Escape], [Backspace] et [delete] génèrent des caractères ASCII

### **glutSpecialFunc(clavierSpecial);**

- ➔ associer une routine à une touche non ASCII du clavier

### **void clavierSpecial(int key, int x, int y){...}**

- ➔ GLUT\_KEY\_F\*
- ➔ GLUT\_KEY\_UP, GLUT\_KEY\_DOWN, GLUT\_KEY\_LEFT, GLUT\_KEY\_RIGHT
- ➔ GLUT\_KEY\_PAGE\_UP, GLUT\_KEY\_PAGE\_DOWN
- ➔ GLUT\_KEY\_HOME, GLUT\_KEY\_END, GLUT\_KEY\_INSERT

*freeglut\_std.h*

24



## GLUT : Fonctions de rappel souris

### **glutMouseFunc(souris);**

➔ associer une routine à un bouton (pression ou relâchement)

**void souris(int button, int state, int x, int y){...}**

➔ button : GLUT\_LEFT\_BUTTON, GLUT\_RIGHT\_BUTTON, GLUT\_MIDDLE\_BUTTON

➔ state : GLUT\_DOWN, GLUT\_UP

### **glutMotionFunc(sourisBouton) (int x, int y );**

➔ associer une routine quand la souris se déplace avec un bouton pressé

**void sourisBouton(int x, int y){...}**

### **glutPassiveMotionFunc (int x, int y);**

➔ associer une routine qd la souris se déplace avec bouton non pressé

**void sourisNonBouton(int x, int y){...}**

25

## GLUT : Fonctions de rappel

### **glutIdleFunc (anim);**

➔ fonction à exécuter quand aucun évènement n'est en cours

**void anim(){...}**

26

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

27

## GLUT : objets géométriques 3D

### Mode fil de fer

- Wire `glutWire* (...)`

### Mode surfacique

- Solid `glutSolid* (...)`

[http://openglut.sourceforge.net/group\\_\\_geometry.html](http://openglut.sourceforge.net/group__geometry.html)

27

3/26/20

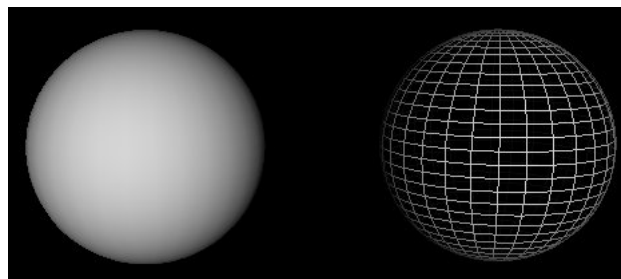
S. Lanquetin – L3 Info, Synthèse d'images

28

## GLUT : sphère

`glutSolidSphere (GLdouble r, GLint m, GLint p);`

`glutWireSphere (GLdouble r, GLint m, GLint p);`



`glutSolidSphere (2.0, 30, 20);`

Centrée sur l'origine

28

3/26/20

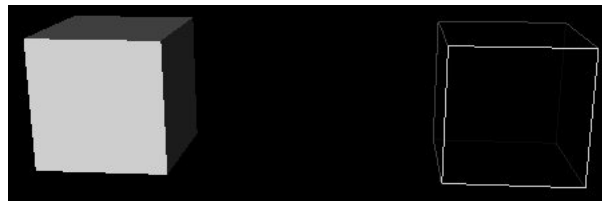
S. Lanquetin – L3 Info, Synthèse d'images

29

## GLUT : cube

```
glutSolidCube (GLdouble taille);
```

```
glutWireCube (GLdouble taille);
```



```
glutSolidCube (1.5);
```

Centré sur l'origine

29

3/26/20

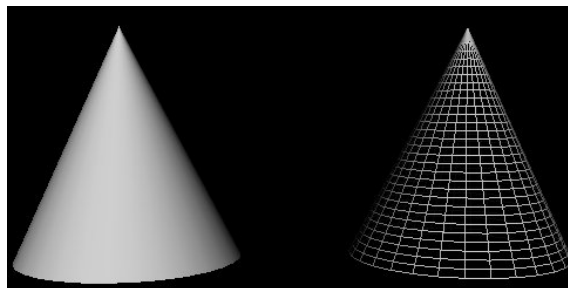
S. Lanquetin – L3 Info, Synthèse d'images

30

## GLUT : cone

```
glutSolidCone (GLdouble r, GLdouble h, GLint m, GLint p);
```

```
glutWireCone (GLdouble r, GLdouble h, GLint m, GLint p);
```



```
glutSolidCone (1.5, 2.0, 20, 15);
```

Centre de la base à l'origine, r ds xy et h dir z

30

3/26/20

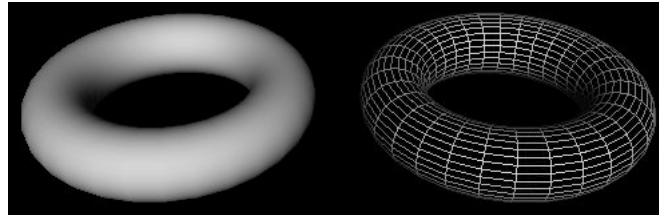
S. Lanquetin – L3 Info, Synthèse d'images

31

## GLUT : tore

```
glutSolidTorus (GLdouble rIn, GLdouble rOut, GLint m, GLint p);
```

```
glutWireTorus (GLdouble rIn, GLdouble rOut, GLint m, GLint p);
```



```
glutSolidTorus (0.2, 1.0, 20, 15);
```

Centré sur l'origine

31

3/26/20

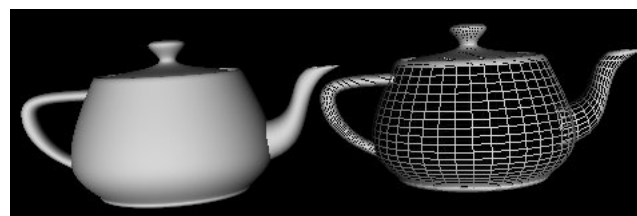
S. Lanquetin – L3 Info, Synthèse d'images

32

## GLUT : théière

```
glutSolidTeapot (GLdouble taille);
```

```
glutWireTeapot (GLdouble taille);
```



```
glutSolidTeapot (2.0);
```

32

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

33

## freeGLUT : cylindre

```
void glutSolidCylinder (GLdouble r, GLdouble h, GLint m, GLint p)
```

```
glutSolidCylinder (0.5, 1.0, 10, 5)
```

Centre de la base à l'origine et axes  
parallèles à l'axe z

33

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

34

## GLUT : objets 3D

```
glutSolidDodecahedron ( );  
glutWireDodecahedron ( );
```

```
glutSolidOctahedron ( );  
glutWireOctahedron ( );
```

```
glutSolidTetraahedron ( );  
glutWireTetraahedron ( );
```

[http://openglut.sourceforge.net/group\\_\\_geometry.html](http://openglut.sourceforge.net/group__geometry.html)

34