
Programmation événementielle avec OpenGL

Frank Singhoff

Bureau C-203

Université de Brest, France

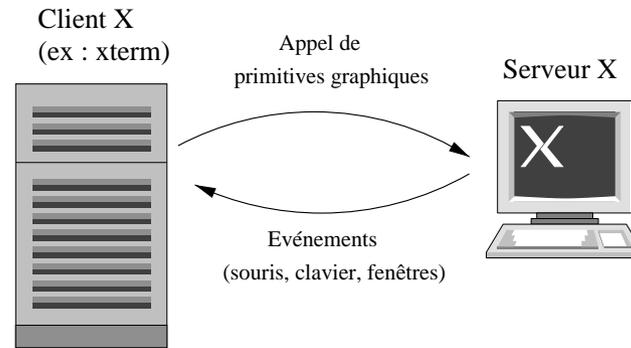
LISyC/EA 3883

singhoff@univ-brest.fr

La bibliothèque GLUT

- La GLUT offre des services annexes à l'animation 3D ... mais des services nécessaires :
 - Interaction avec le système de multi-fenêtrage (X11).
 - Interaction avec les périphériques d'entrées (souris/claviers).
 - Boucle d'événements et outils d'animation.
 - Gestion de menus simplifiés.
 - Quelques objets 3D simplifiés.

Interaction avec le serveur X11



- Programmation événementielle : *callbacks* + boucle d'attente des événements.
- Principales primitives OpenGL associées à X11 :
 - *glutCreateWindow* : création de fenêtre.
 - *glutInitWindowSize*, *glutInitWindowPosition* : positionnement de variables d'état OpenGL.
 - Callbacks de traitements des événements X11 (clavier, souris, événements *Map* et *Resize*).

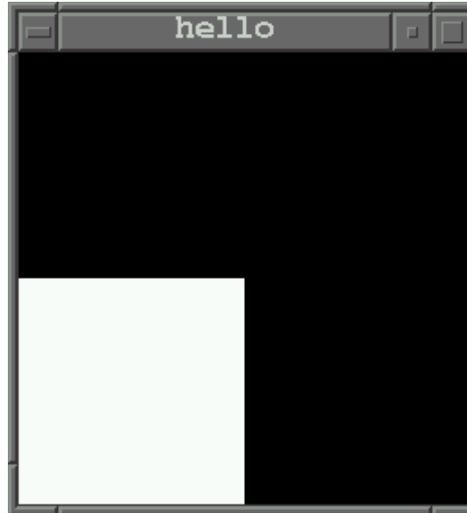
Boucle d'événements et d'animation (1)

- **Services nécessaires pour tous types d'animations :**
 - *glutMainLoop* : boucle d'attente des événements.
 - *gluInitDisplayMode()* : définit le mode de fonctionnement de la bibliothèque (simple/double buffering, codage des couleurs, ...).
 - *glutDisplayFunc* : associe un callback aux demandes d'affichage.
 - *glutPostRedisplay()* : provoque une restitution (et donc un ré-affichage).
- **Une animation peut être implantée de diverses manières :**
 1. Animation avec tampon simple. *glFlush* pour forcer la terminaison de la restitution.
 2. Animation avec tampon double. *glutSwapBuffers* pour forcer la terminaison de la restitution et échanger les tampons.

Boucle d'événements et d'animation (2)

- Généralement, les restitutions sont déclenchées sur occurrences d'événements et exécution de leur callback respectif. Pour ce faire, les *callbacks* invoquent *glutPostRedisplay* pour provoquer la restitution.
- Exemple de callbacks :
 1. Restitution sur clavier/souris : *glutMouseFunc* ou *glutKeyboardFunc* enregistrent les callbacks qui vont traiter les événements produits par le périphérique.
 2. Restitution sur timer : *glutTimerFunc* enregistre le callback associé au timer. Le callback est invoqué lorsque le compteur du timer atteint zéro.
 3. Restitution par tache de fond : *glutIdleFunc*.

Boucle d'événements et d'animation (3)



- Dessiner cette scène, c'est :
 1. Vider l'écran, c-a-d les tampons.
 2. Choisir une couleur (le blanc).
 3. Dessiner le carré (c'est un polygone).

Boucle d'événements et d'animation (4)

```
#include <stdlib.h>
#include <GL/glut.h>

void display(void) {
    /* Vider les tampons */
    glClear(GL_COLOR_BUFFER_BIT);
    glClear(GL_DEPTH_BUFFER_BIT);

    /* Dessiner le polygone */
    glColor3f (1.0, 1.0, 1.0);
    glBegin(GL_POLYGON);
        glVertex3f (0.0, 0.0, 0.0);
        glVertex3f (0.5, 0.0, 0.0);
        glVertex3f (0.5, 0.5, 0.0);
        glVertex3f (0.0, 0.5, 0.0);
    glEnd();

    /* Forcer le rendu maintenant ;
       pour du double buffering*/
    glutSwapBuffers ();
}
```

Boucle d'événements et d'animation (5)

```
#include <stdlib.h>
#include <GL/glut.h>

/*
   Initialisation de la scène (variables d'état OpenGL)
*/
void init (void)
{
    /* Choisir la couleur d'effacement */
    glClearColor (0.0, 0.0, 0.0, 0.0);

    /* Sélectionner la matrice de projection */
    glMatrixMode(GL_PROJECTION);

    /* Initialialise la matrice de projection */
    glLoadIdentity();

    /* Clipping de cadrage */
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
}
```

Boucle d'événements et d'animation (6)

- **Restitution déclenchée par le clavier :**

```
/* Callback clavier */
void key(unsigned char k, int x, int y) {
    ...
    glutPostRedisplay();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv); /* Initialisation de la GLUT */

    /* GLUT_SINGLE, GLUT_DOUBLE pour simple/double buffering */
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);

    /* Creation fenetre et initialisation de la scene */
    glutInitWindowPosition (100, 100);
    glutCreateWindow ("hello");

    /* Connexion callbacks puis boucle d'evenements */
    glutDisplayFunc(display); glutKeyboardFunc(key);
    glutMainLoop();
}
```

Boucle d'événements et d'animation (7)

- **Restitution déclenchée par timer :**

```
/* Callback timer */
void my_timer(int x) {
    ...
    glutTimerFunc(40, my_timer, 1);
    glutPostRedisplay();
}

int main(int argc, char** argv) {
/* Initialisation de la GLUT et choix du mode d'affichage */
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);

/* Creation fenetre et initialisation de la scene */
    glutInitWindowPosition (100, 100);
    glutCreateWindow ("hello");

/* Connexion callbacks puis boucle d'evenements */
    glutDisplayFunc(display); glutTimerFunc(40, my_timer, 1);
    glutMainLoop();
}
```

Boucle d'événements et d'animation (8)

- Il ne doit pas y avoir d'instruction de restitution hors du callback d'affichage. Appel à *glutPostRedisplay* pour forcer la restitution.
- Le callback d'affichage doit être idempotent (ne doit pas modifier l'état d'OpenGL).
- Ne pas activer de callback inutilement (performances).
- Attention aux calculs redondants (performances).