

III. La librairie GLUT

1. Fonctionnalités de GLUT (OpenGL Utility Toolkit)

Les fonctionnalités de GLUT :

- créer et gérer plusieurs fenêtres d'affichage,
- interruptions (click souris, touches clavier, ...),
- menus déroulants
- connaître la valeur d'un certain nombre de paramètres systèmes,
- quelques fonctions supplémentaires : objets 3D (cube, sphère, ...).

GLUT : fenêtrage basé sur la **programmation événementielle**

➔ structuration particulière de l'application.

Principe de la programmation Événementielle :

1. Définir autant de fcts (actions à exécuter) que de types d' interruption
 - fct qui traite un clic souris
 - fct qui traite une touche clavier
 - etc.
2. Désigner ces fcts au système en les associant à une interruption
(clic \Leftrightarrow fct clic , clavier \Leftrightarrow fct clavier, ...),
3. Donner ensuite la main au système pour scruter les interruptions

Lorsqu' il y a une interruption :

- c' est le système qui appelle la fct associée à cette interruption
- avec les paramètres prévus pour cette interruption
(coordonnées souris, ...)

Gestion des interruptions sous GLUT :

Boucle qui scrute six événements :

1. click souris : action1
2. touches alphanumériques : action2
3. touches spéciales (->, F1, ...) : action3
4. menus et sous menus : action4
5. redimensionner une fenêtre : action5
6. **glutPostRedisplay()** ; //demande d'affichage

L'affichage est géré comme une interruption :

- il y a une seule fonction principale d'affichage
- on ne doit pas l'appeler directement
- on demande au système d'appeler cette fonction avec `glutPostRedisplay()`

2. Structure d'une application GLUT

```
#include <stdio.h>
#include <GLUT/glut.h>    /* <GL/glut.h> */

Fonctions quelconques

Fonction associée à un click souris

Fonction associée aux touches alphanumériques

Fonction associée aux touches spéciales (flèches, F1, F2...)

Fonction menu

Fonction de fenêtrage (redimension de la fenêtre)

Fonction d'affichage (tracé de la scène)

void main()
{  initialisation d'une session graphique
   désignation de la fct d'affichage
   désignation de la fct de fenêtrage
   ...
   /* boucle de capture d'événements */
   glutMainLoop();
}
```

a) Fonction d'affichage

```
void afficheMaScene (void)
{ "effacer" l'écran
  positionner la caméra
  tracer la scène
  glutSwapBuffers(); /* glFlush() */
}

void main (int argc, char **argv)
{
  ...
  /* Associations interruption <-> fonction */
  glutDisplayFunc(afficheMaScene);
  ...
  /* boucle de capture d'évenements */
  glutMainLoop();
}
```

b) Modification de fenêtre

```
void monFenetrage(int largeur, int hauteur)
{ cadrage dans la fenêtre
  definition de la projection 3D->2D
  glutPostRedisplay() ; //demande de réaffichage de la scène
}
```

```
void main (int argc, char **argv)
{ ...
  /* Associations interruption <-> fonction */
  ...
  glutReshapeFunc (monFenetrage) ;
  ...
  /* boucle de capture d'evenements */
  glutMainLoop();
}
```

c) Autres interruptions

```
void clicSouris(int bouton, int etat, int x, int y)
{ action à déclencher suivant l'état (GLUT_DOWN, GLUT_UP)
  glutPostRedisplay() ; //si la scène doit être réaffichée
}

void mouvementSouris(int x, int y)
{ action liée à un déplacement souris bouton enfoncé
  glutPostRedisplay() ; //si la scène doit être réaffichée
}

void main (int argc, char **argv)
{ ...
  /* Associations interruption <-> fonction */
  ...
  glutMouseFunc(clicSouris) ;
  glutMotionFunc(mouvementSouris) ;
  /* boucle de capture d'évenements */
  glutMainLoop();
}
```



```
void clavier(unsigned char key, int x, int y)
{ action déclenchée sur une touche type caractère ASCII
  glutPostRedisplay() ; //si la scène doit être réaffichée
}

void touchesSpeciales(unsigned char key,int x,int y)
{ action déclenchée sur une touche F1, ..., F10, flèches
  glutPostRedisplay() ; //si la scène doit être réaffichée
}

void main (int argc, char **argv)
{ ...
  /* Associations interruption <-> fonction */
  ...
  glutKeyboardFunc(clavier);
  glutSpecialFunc(touchesSpeciales);
  ...
  /* boucle de capture d'évenements */
  glutMainLoop();
}
```

d) Structure du main

```
int main (int argc, char **argv)
{ /* initialisation d'une session GLUT */
  glutInit(argc, argv);
  glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB|GLUT_DEPTH);
  glutInitWindowSize(500, 500);
  glutInitWindowPosition(10, 10); /* coin sup gauche */
  glutCreateWindow("fenetre demo");
/* initialisation propre a l'application */
  initScene();
/* Associations interruption <-> fonction */
  glutDisplayFunc(afficheMaScene);
  glutReshapeFunc(monFenetrage);
  glutMouseFunc(clicSouris);
  glutMotionFunc(mouvementSouris);
  glutKeyboardFunc(clavier);
  glutSpecialFunc(touchesSpeciales);
/* boucle de capture d'evenements */
  glutMainLoop();
  return 0 ;
}
```

3. Initialisation d'une session GLUT

glutInit(argc, argv)

initialise la bibliothèque *GLUT*

négoce une session avec le système de fenêtrage.

glutInitWindowSize(largeur, hauteur)

spécifie la taille de la fenêtre.

glutInitWindowPosition(x0, y0)

positionne la fenêtre sur l'écran par rapport au coin supérieur gauche

glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)

spécifie le mode d'affichage dans la fenêtre. Ici :

- en double buffer
- en mode RGB
- avec gestion de la profondeur (parties cachées)

4. Gestion des fenêtres

```
int glutCreateWindow( char * name )
```

crée une fenêtre et retourne un entier positif identifiant la fenêtre

```
void glutDestroyWindow( int numFenetre )
```

détruit une fenêtre

```
void glutSetWindow ( int numFenetre )
```

établit quelle est la fenêtre courante

```
int glutGetWindow ( void )
```

retourne le numéro de la fenêtre courante

5. Gestion du dessin d' une scène

Appel de la fonction d' affichage : **glutPostRedisplay()** ;

- c' est le signal pour dire au système que la fenêtre doit être rafraîchie
- le système relancera dès que possible la fonction d' affichage qui a été désignée par `glutDisplayFunc`

L' image à afficher est construite dans un buffer (zone tampon) :

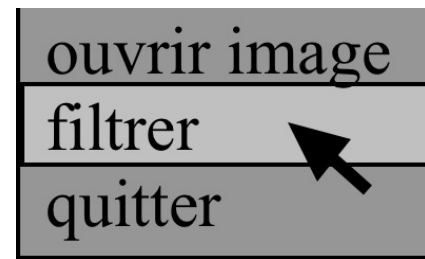
void glutSwapBuffers(void)

recopie le buffer dans la fenêtre courante si le mode est *double buffer*

void glFlush(void)

répercute le tampon dans la fenêtre courante si le mode est *simple buffer*

6. Gestion des menus



Création d'un menu :

- dans une fonction, définir les actions associées à chaque gâchette


```
void monMenu(int n) /* n° de gachette transmis par le système */
{ switch(n)
  {case 0 : lire(image) ; break ;
   case 1 : filtrer(image) ; break ;
   case 2 : exit(0) ;
  }
  glutPostRedisplay() ;
}
```

- attacher cette fonction au click souris avec un menu

```
glutCreateMenu(monMenu) ;
  glutAddMenuEntry("ouvrir image", 0) ;
  glutAddMenuEntry("filtrer", 1) ;
  glutAddMenuEntry("quitter", 2) ;
glutAttachMenu(GLUT_RIGHT_BUTTON) ; // clôture du menu
```

Création d'un sous-menu :

ouvrir image	
filtrer	passe bas
quitter	passe haut



```
int numSousMenu ;
```

```
...
```

```
numSousMenu = glutCreateMenu(monSousMenu) ;
```

```
    /* gachettes du sous-menu */
```

```
    glutAddMenuEntry("passe bas", 0) ;
```

```
    glutAddMenuEntry("passe haut", 1) ;
```

```
glutCreateMenu(monMenu) ;
```

```
    /* gachettes du menu */
```


```
    glutAddMenuEntry("ouvrir image", 0) ;
```

```
    glutAddSubMenu("filtrer", numSousMenu) ;
```

```
    glutAddMenuEntry("quitter", 2) ;
```

```
glutAttachMenu(GLUT_RIGHT_BUTTON) ;
```

ouvrir image	
filtrer	passe bas
quitter	passe haut



```
void monSousMenu(int n)
{ switch(n)
  { case 0 : passeBas(image) ; break ;
    case 1 : passeHaut(image) ; break ;
  }
  glutPostRedisplay() ;
}
```

```
void monMenu(int n)
{ switch(n)
  {case 0 : lire(image) ; break ;
   case 2 : exit(0) ;
  }
}
```



```
int glutCreateMenu(void (* fct) (int value))
```

crée un nouveau menu déroulant

La fonction **fct** est appelée lorsqu'un élément du menu est sélectionné

La valeur passée à la fonction correspond au numéro de la sélection

```
int glutAddSubMenu(char *name, int value)
```

Ajoute un sous-menu au bas du menu courant. Le 1^{er} paramètre définit le titre de la gachette. Le deuxième paramètre est le numéro du sous-menu ajouté.

```
void glutAddMenuEntry(char * name, int numeroAction )
```

ajoute une gachette au bas du menu courant

```
void glutAttachMenu( int boutonSouris )
```

```
void glutDetachMenu( int boutonSouris )
```

attache ou détache le menu courant à un des boutons de la souris

7. Piloter à la souris

Variables globales pour la position et le déplacement :

```
int Xmouse, Ymouse, dX, dY ; // coordonnées et déplacement
```

Détection d'un clic :

```
void clicSouris(int bouton, int state, int x, int y)
{ switch (bouton)
  { case GLUT_LEFT_BUTTON: Xmouse=x;  Ymouse=y; break;
    ...
  }
}
```

Déplacement bouton enfoncé :

```
void mouvementSouris(int x, int y)
{ dX = x-Xmouse;    dY = Ymouse-y; // !!!repère window indirect
  Xmouse=x;  Ymouse=y; // mémorisation de la nouvelle position
  ...
  glutPostRedisplay(); // si action sur la scène
}
```