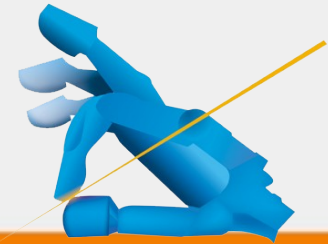


Infographie-CAO-RV



IMERIR

Ecole d'ingénierie informatique et robotique

Exemple d'animation /OpenGL



A.Rharmaoui-2012

ANIMATION 1/8

Animation d'un cube par les touches + et -

Le programme opengl-anim1.c fait une animation d'un cube en rotation en appuyant sur les touches + et -.

Principe

Une animation est obtenue en modifiant progressivement les paramètres d'un affichage et en retraçant la figure après chaque modification.

En raison de la persistance rétinienne, l'observateur voit une scène en mouvement continu.

Dans le programme ci-dessus, l'angle de rotation du cube est modifié par l'appui sur les touches + et -.

ANIMATION 2/8

Instructions pour l'animation par double buffer

L'animation est gérée par deux buffers :

- un buffer de tracé non visible (le buffer arrière)
- un buffer visible (le buffer avant).

Une fois le nouveau tracé effectué, on permute les deux buffers au moyen de la commande **glutSwapBuffers()** (bibliothèque GLUT) après **glFlush()**.

Allocation du double buffer

À placer dans l'initialisation d'OpenGL.

```
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEPTH);
```

Échange des deux buffers

À placer après chaque dessin de la scène.

```
glutSwapBuffers();
```

Modification des valeurs du tracé

Après chaque modification des valeurs du tracé, il faut redemander un affichage

ANIMATION 3/8

Animation automatique d'un cube par une fonction de call-back d'attente

Le programme opengl-anim3.c fait une animation d'un cube en rotation automatique.

Définition d'une fonction de call-back d'attente

On définit une fonction de call-back d'attente qui est appelée dès que le système ne traite plus un autre événement ou un autre call-back.

glutIdleFunc(&window_idle);

Cette fonction incrémente l'angle de rotation et demande le réaffichage.

```
GLvoid window_idle()
{
    angle++;
    glutPostRedisplay();
}
```

ANIMATION 4/8

Animation automatique (suite)

Selon les machines, une même animation automatique s'exécutera plus ou moins vite si l'on ne contrôle pas le rythme de mise à jour.

Définition d'une vitesse maximale d'animation

On définit un intervalle minimal de rafraîchissement de 40ms.

```
void anime(int pas){
    /* action à effectuer en fonction du numéro d'image */
    if(pas == 2000) {
        return;
    }

    /* redemande l'affichage */
    glutPostRedisplay();

    pas++;
    /* rappellera la fonction d'animation anime dans 40 millisecondes
    au plus tôt */
    glutTimerFunc( 40, anime, pas );
}
```

ANIMATION 5/8

Animation automatique (suite)

Attention, la fonction de call-back définie par `glutTimerFunc` n'est appelée qu'une seule fois.

Le programme `opengl-anim4.c` propose une animation contrôlée par un temps de latence de 40ms.

ANIMATION 6/8

Animation automatique (fin)

Quelques principes pour réussir une animation:

temporiser les animations avec la fonction glutTimerFunc

avoir une variable globale unique t pour le temps,

lire le temps machine entre deux frames pour voir si le rythme de mise à jour est suffisant (au minimum 30 fps),

prévoir des procédures d'optimisation en cas de charge trop forte (niveaux de détail, maillage adaptatif, remplacement d'objets par des imposteurs...).

ANIMATION 7/8

Animation d'un cube par déplacement de la souris

Le programme src/opengl-anim6.c fait une animation d'un cube par déplacement de la souris avec le bouton gauche enfoncé.

Définition d'une fonction de call-back de gestion des clicks souris (avant un déplacement)

```
glutMouseFunc(&window_mouseFunc);
```

On note le lieu où a lieu un click souris de sorte à gérer les mouvements relatifs de la souris lors de son déplacement.

```
GLvoid window_mouseFunc(int button, int state, int x, int y)
{
    if (state == GLUT_DOWN && button == GLUT_LEFT_BUTTON) {
        mouse_pos_x = x;
        mouse_pos_y = y;
    }
}
```


ANIMATION 8/8

Animation par déplacement de la souris (suite et fin)

Définition d'une fonction de call-back de gestion des déplacements de la souris

```
glutMotionFunc(&window_motionFunc);
```

Cette fonction modifie l'angle de rotation en fonction des déplacements relatifs de la souris et demande le réaffichage.

```
GLvoid window_motionFunc(int x, int y)
{
    angle_x += y - mouse_pos_y;
    angle_y += x - mouse_pos_x;

    mouse_pos_x = x;
    mouse_pos_y = y;

    glutPostRedisplay();
}
```