

实验 7 图形交互技术 参考资料

1 . 实验目的 :

理解掌握一个 OpenGL 程序的常见交互方法。

2 . 实验内容 :

(1) 运行示范实验代码 1, 掌握程序鼠标交互方法, 尝试为其添加键盘与菜单控制, 实现同样功能 ;

(2) 运行示范实验代码 2, 掌握程序鼠标坐标获取与绘图方法, 尝试为其添加绘制直线功能 ;

实验扩展:

(3) 结合上述两步, 能否实现通过鼠标右键菜单切换实现一个简单的绘图程序, 可以绘制直线、三角形、正方形等常见图形 ?

3 . 实验原理 :

要想在 OpenGL 中处理鼠标事件非常的方便, GLUT 已经为我们的注册好了函数, 只要我们提供一个方法。使用函数 `glutMouseFunc`, 就可以帮我们注册我们的函数, 这样当发生鼠标事件时就会自动调用我们的方法。

函数的原型是 :

```
void glutMouseFunc(void(*func)(int button,int state,int x,int y));
```

参数 : `func` : 处理鼠标 click 事件的函数的函数名。

从上面可以看到, 处理鼠标单击事件的函数, 一定有 4 个参数。第一个参数表明哪个鼠标键被按下或松开, 这个变量可以是下面的三个值中的一个 :

`GLUT_LEFT_BUTTON`

`GLUT_MIDDLE_BUTTON`

`GLUT_RIGHT_BUTTON`

第二个参数表明，函数被调用发生时，鼠标的状态，也就是是被按下，或松开，可能取值如下：

GLUT_DOWN

GLUT_UP

当函数被调用时，state 的值是 GLUT_DOWN，那么程序可能会假定将会有个 GLUT_UP 事件，甚至鼠标移动到窗口外面，也如此。然而，如果程序调用 glutMouseFunc 传递 NULL 作为参数，那么 GLUT 将不会改变鼠标的状态。剩下的两个参数 (x,y) 提供了鼠标当前的窗口坐标（以左上角为原点）。

键盘相关知识可参考：http://blog.csdn.net/xie_zi/article/details/1911891

菜单相关知识可参考：http://blog.csdn.net/xie_zi/article/details/1963383

4 . 示范代码：

代码 1: 鼠标控制旋转的正方形

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

#include <math.h>

#include <stdlib.h>

#define DEGREES_TO_RADIANS 3.14159/180.0

static GLfloat spin = 0.0;

GLfloat x = 0.0;

GLfloat y = 0.0;

GLfloat size = 50.0;

GLfloat angle = 2.0;

GLsizei wh = 500, ww = 500;

void square()
{
    glBegin(GL_QUADS);
    glVertex2f(x,y);
    glVertex2f(-y,x);
    glVertex2f(-x,-y);
    glVertex2f(y,-x);
}
```

```

    glEnd();
}

void spinDisplay(void)
{
    spin = spin + 2.0;
    if (spin > 360.0) spin = spin - 360.0;
    x=125.0 * cos(DEGREES_TO_RADIANS*spin);
    y=125.0 * sin(DEGREES_TO_RADIANS*spin);
    glutPostRedisplay();
}

void mydisplay()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);
    square();
    glutSwapBuffers();
}

void init()
{
    glClearColor (0.0, 0.0, 0.0, 1.0);
}

void myreshape(GLint w, GLint h) {
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-w/2, w/2, -h/2, h/2, -1.0, 1.0);
    glMatrixMode(GL_MODELVIEW);

    ww = w;
    wh = h;
}

```

```

void mymouse(GLint button, GLint state, GLint wx, GLint wy)
{
    if(button ==GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
        exit(0);
    if(button ==GLUT_LEFT_BUTTON && state == GLUT_DOWN)
        glutIdleFunc(spinDisplay);
    if(button== GLUT_MIDDLE_BUTTON && state == GLUT_DOWN)
        glutIdleFunc(NULL);

}

void main(int argc, char** argv)
{
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(500,500);
    glutInitWindowPosition(0,0);
    glutCreateWindow("double");
    init();
    glutDisplayFunc(mydisplay);
    glutReshapeFunc(myreshape);
    glutMouseFunc(mymouse);
    glutIdleFunc(spinDisplay);
    glutMainLoop();
}

```

代码 2: 鼠标当前位置绘制方框

```

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

GLfloat x = 0.0;

```

```

GLfloat y = 0.0;

GLfloat size = 50.0;

GLsizei wh = 500, ww = 500;

void drawSquare(GLint x, GLint y) {
    y = wh-y;
    glBegin(GL_POLYGON);
    glVertex3f(x + size, y + size, 0);
    glVertex3f(x - size, y + size, 0);
    glVertex3f(x - size, y - size, 0);
    glVertex3f(x + size, y - size, 0);
    glEnd();
}

void mydisplay()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);
    drawSquare(x, y);
    glutSwapBuffers();
    glutPostRedisplay();
}

void init()
{
    glClearColor (0.0, 0.0, 0.0, 1.0);
}

void myreshape(GLint w, GLint h) {
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0, w, 0, h, -1.0, 1.0);
    glMatrixMode(GL_MODELVIEW);
}

```

```

    ww = w;

    wh = h;
}

void mymouse(GLint button, GLint state, GLint wx, GLint wy)
{ if(button ==GLUT_RIGHT_BUTTON && state == GLUT_DOWN)

    exit(0);

    if(button ==GLUT_LEFT_BUTTON && state == GLUT_DOWN)

    { x = wx;y = wy; }

}

void main(int argc, char** argv)
{
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(500,500);
    glutInitWindowPosition(0,0);
    glutCreateWindow("double");
    init();
    glutDisplayFunc(mydisplay);
    glutReshapeFunc(myreshape);
    glutMouseFunc(mymouse);
    glutMainLoop();
}

```