

## OpenGL 编程题

Collected and sorted out by Linst <root@sitao.org>

2.请使用 OpenGL 和 GLUT 编写一个简单的图形程序，用于显示一个填充的白色矩形。其中矩形规定为  $(-0.8, -0.8) \sim (0.8, 0.8)$ ，程序窗口的大小为  $(200, 200)$ ，标题为“白色矩形”。

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

void Paint()
{
    glClear(GL_COLOR_BUFFER_BIT); // 初始化颜色缓冲区
    glColor3f(1, 1, 1); // 设置颜色为白色
    glRectf(-0.8, -0.8, 0.8, 0.8);
    glFlush();
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(200, 200); // 程序窗口大小
    glutCreateWindow("白色矩形"); // 窗口标题
    glutDisplayFunc(Paint); // 执行场景绘制函数
    glutMainLoop(); // 开始循环执行 OpenGL 命令
}
```



3. 请使用 OpenGL 和 GLUT 编写一个简单的图形程序，用于显示一个填充的红色三角形。其中三角形的顶点分别是  $(-0.8, -0.8)$ 、 $(0.8, -0.8)$  和  $(0, 0.8)$ ，程序窗口大小为  $(200, 200)$ ，标题为“红色三角形”。

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

void Paint()
{
    glClear(GL_COLOR_BUFFER_BIT); // 初始化颜色缓冲区
    glColor3f(1, 0, 0);          // 设置颜色为红色 (R, G, B)
    glBegin(GL_TRIANGLES);      // 三角形
    glVertex2d(-0.8, -0.8);     // 顺时针方向指定三顶点坐标，连线
    glVertex2d(0.8, -0.8);
    glVertex2d(0, 0.8);
    glEnd(); // 三角形定义结束
    glFlush(); // 线段定义结束
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(200, 200); // 程序窗口大小
    glutCreateWindow("红色三角形"); // 窗口标题
}
```

```

glutDisplayFunc(Paint);    // 执行场景绘制函数
glutMainLoop();           // 开始循环执行 OpenGL 命令
}

```



4. 请使用 OpenGL 和 GLUT 编写一个简单的图形程序，用于显示一个填充的蓝色平行四边形。其中平行四边形的 4 个顶点分别是  $(-0.9, -0.4)$ 、 $(0.4, -0.4)$ 、 $(0.9, 0.4)$  和  $(-0.4, 0.4)$ ，程序窗口的大小为  $(300, 300)$ ，标题为“蓝色平行四边形”。

```

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

void Paint()
{
    glClear(GL_COLOR_BUFFER_BIT); // 初始化颜色缓冲区
    glColor3f(0, 0, 1);          // 设置颜色为蓝色 (R, G, B)
    glBegin(GL_QUADS);           // 四边形
    glVertex2d(-0.9, -0.4);      // 顺时针方向指定四顶点坐标，连线
    glVertex2d(0.4, -0.4);
    glVertex2d(0.9, 0.4);
    glVertex2d(-0.4, 0.4);
    glEnd(); // 定义结束
    glFlush();
}

```

```
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(300, 300);    // 程序窗口大小
    glutCreateWindow("蓝色平行四边形"); // 窗口标题
    glutDisplayFunc(Paint);          // 执行场景绘制函数
    glutMainLoop();                  // 开始循环执行 OpenGL 命令
}
```



5. 请使用 OpenGL 和 GLUT 编写一个简单的图形程序，用于显示一个填充的紫色梯形。其中梯形的 4 个顶点分别是  $(-0.9, -0.4)$ 、 $(0.4, -0.4)$ 、 $(0.4, 0.4)$  和  $(-0.4, 0.4)$ ，程序窗口的大小为  $(300, 300)$ ，标题为“紫色梯形”

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

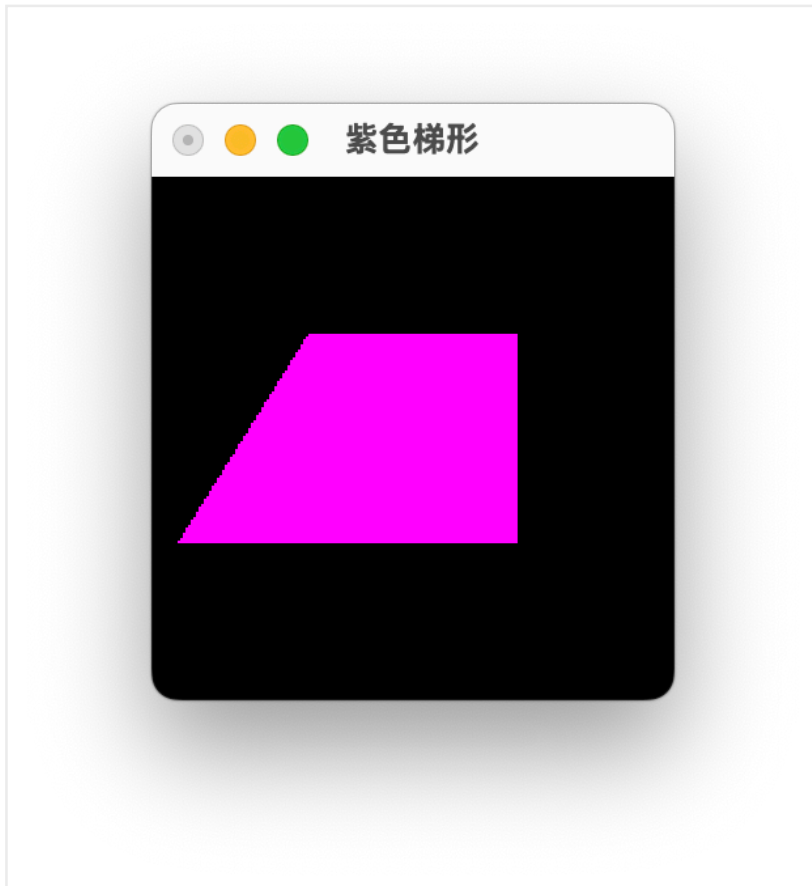
```
void Paint()
```

```

{
    glClearColor(GL_COLOR_BUFFER_BIT); // 初始化颜色缓冲区
    glColor3f(1, 0, 1); // 设置颜色为紫色 (R, G, B)
    glBegin(GL_QUADS); // 四边形
    glVertex2d(-0.9, -0.4); // 顺时针方向指定四顶点坐标, 连线
    glVertex2d(0.4, -0.4);
    glVertex2d(0.4, 0.4);
    glVertex2d(-0.4, 0.4);
    glEnd(); // 定义结束
    glFlush();
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(300, 300); // 程序窗口大小
    glutCreateWindow("紫色梯形"); // 窗口标题
    glutDisplayFunc(Paint); // 执行场景绘制函数
    glutMainLoop(); // 开始循环执行 OpenGL 命令
}

```



28. 请使用 OpenGL、GLU 和 GLUT 编写一个显示线框立方体的程序。其中立方体的半径为 1.5 单位，并首先绕  $(0, 0, 0) \sim (1, 1, 0)$  旋转 30 度，然后远移 6.5 单位；观察体规定为：视场角=30 度，宽高比=1，近=1，远=100；程序窗口的大小为 (200, 200)，标题为“线框立方体”。

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

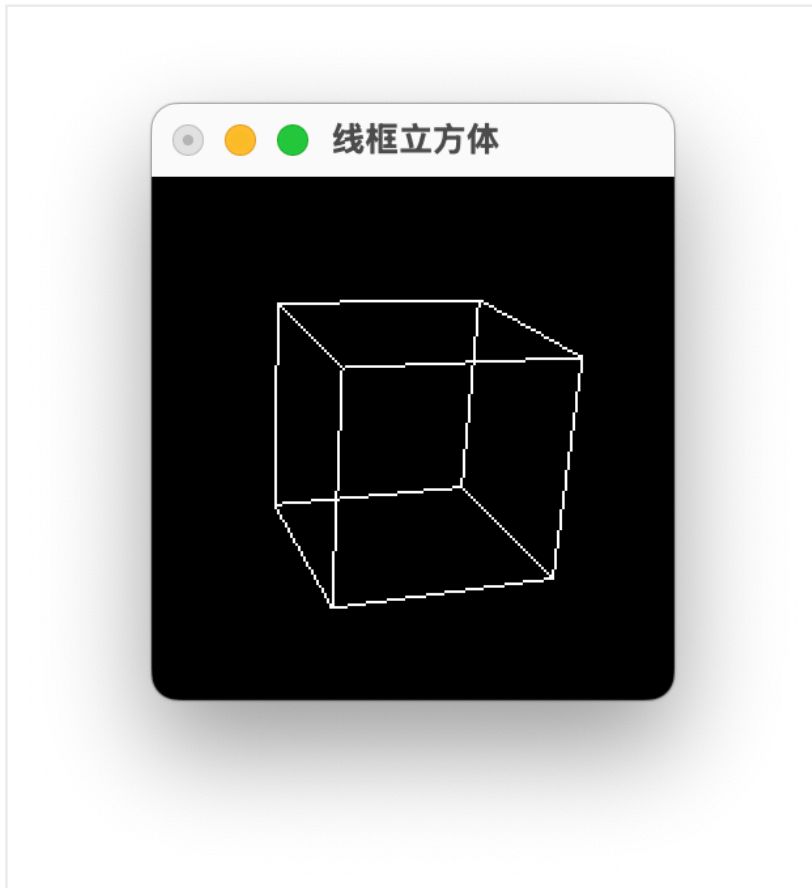
void Paint()
{
    glClear(GL_COLOR_BUFFER_BIT); // 初始化颜色缓冲区

    glLoadIdentity();           // 初始化矩阵
    gluPerspective(30, 1, 1, 100); // 视角

    glTranslated(0, 0, -6.5); // 远移 6.5 单位
    glRotated(30, 1, 1, 0); // 绕 (0, 0, 0)~(1, 1, 0) 旋转 30 度

    glutWireCube(3.0); // 线框体，半径为 1.5 单位
    glFlush();
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(200, 200); // 程序窗口大小
    glutCreateWindow("线框立方体"); // 窗口标题
    glutDisplayFunc(Paint); // 执行场景绘制函数
    glutMainLoop(); // 开始循环执行 OpenGL 命令
}
```



29. 请使用 OpenGL、GLU 和 GLUT 编写一个三维犹他茶壶程序。其中茶壶的半径为 1 单位，并远移 6.5 单位；观察体规定为：视场角=30 度，宽高比=1，近=1，远=100；程序窗口的大小为 (200, 200)，标题为“尤他茶壶”。

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

void Paint()
{
    glClear(GL_COLOR_BUFFER_BIT); // 初始化颜色缓冲区
    glLoadIdentity();           // 初始化矩阵
    gluPerspective(30, 1, 1, 100); // 视角
    glTranslated(0, 0, -6.5);    // 远移 6.5 单位
    // glColor3f(0, 1, 0);
    glutSolidTeapot(1); // 尤他茶壶，半径为 1 单位
    glFlush();
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(200, 200); // 程序窗口大小
    glutCreateWindow("尤他茶壶"); // 窗口标题
    glutDisplayFunc(Paint);      // 执行场景绘制函数
}
```

```
    glutMainLoop();          // 开始循环执行 OpenGL 命令
}
```



30. 请使用 OpenGL 和 GLUT 编写一个显示线框球体的简单图形程序。其中球体的半径为 0.8，经线数为 24，纬线数为 12，并绕 x 轴旋转 30 度，程序窗口的大小为 (200, 200)，标题为“线框球”。

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

void Paint()
{
    glClear(GL_COLOR_BUFFER_BIT); // 初始化颜色缓冲区
    glLoadIdentity();           // 初始化矩阵
    glRotated(30, 1, 0, 0);     // 旋转
    glutWireSphere(0.8, 24, 12); // 线框球 (半径、经线、纬线)
    glFlush();
}

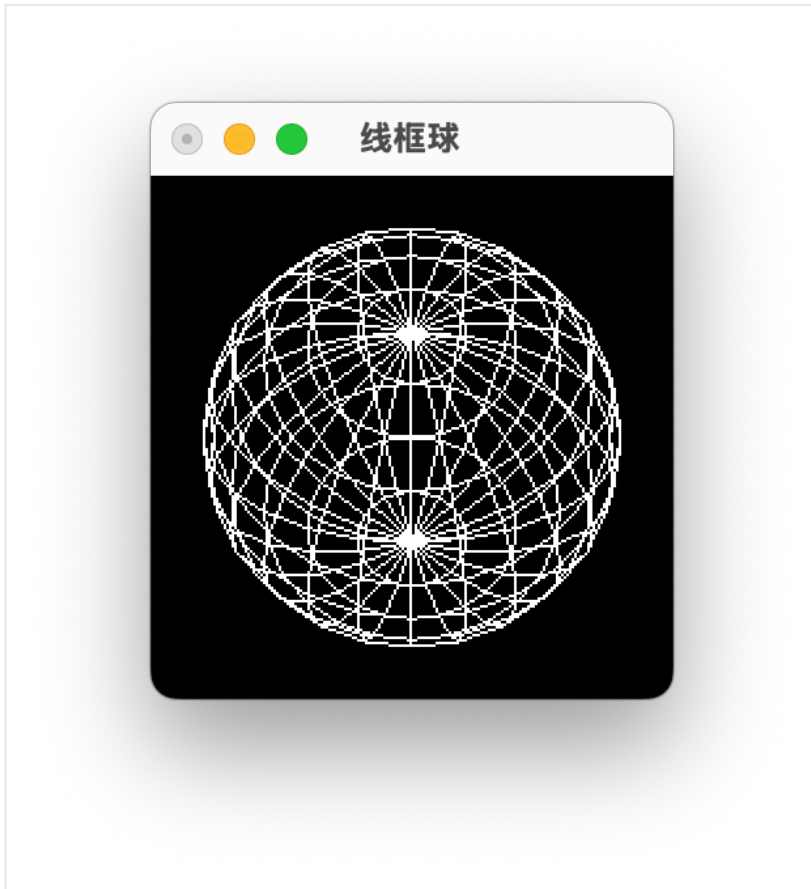
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(200, 200); // 程序窗口大小
    glutCreateWindow("线框球"); // 窗口标题
}
```



```

glutDisplayFunc(Paint);    // 执行场景绘制函数
glutMainLoop();           // 开始循环执行 OpenGL 命令
}

```



31. 请使用 OpenGL 和 GLUT 编写一个显示线框椭球体的简单图形程序。其中椭球体的两极方向为上下方向，左右方向的半径为 0.98，上下方向的半径为 0.49，前后方向的半径为 0.6，经线数为 48，纬线数为 24，使用正投影，裁剪窗口为  $(-1, -0.5) \sim (1, 0.5)$ ，程序窗口的大小为  $(400, 200)$ ，标题为“线框椭球”。

```

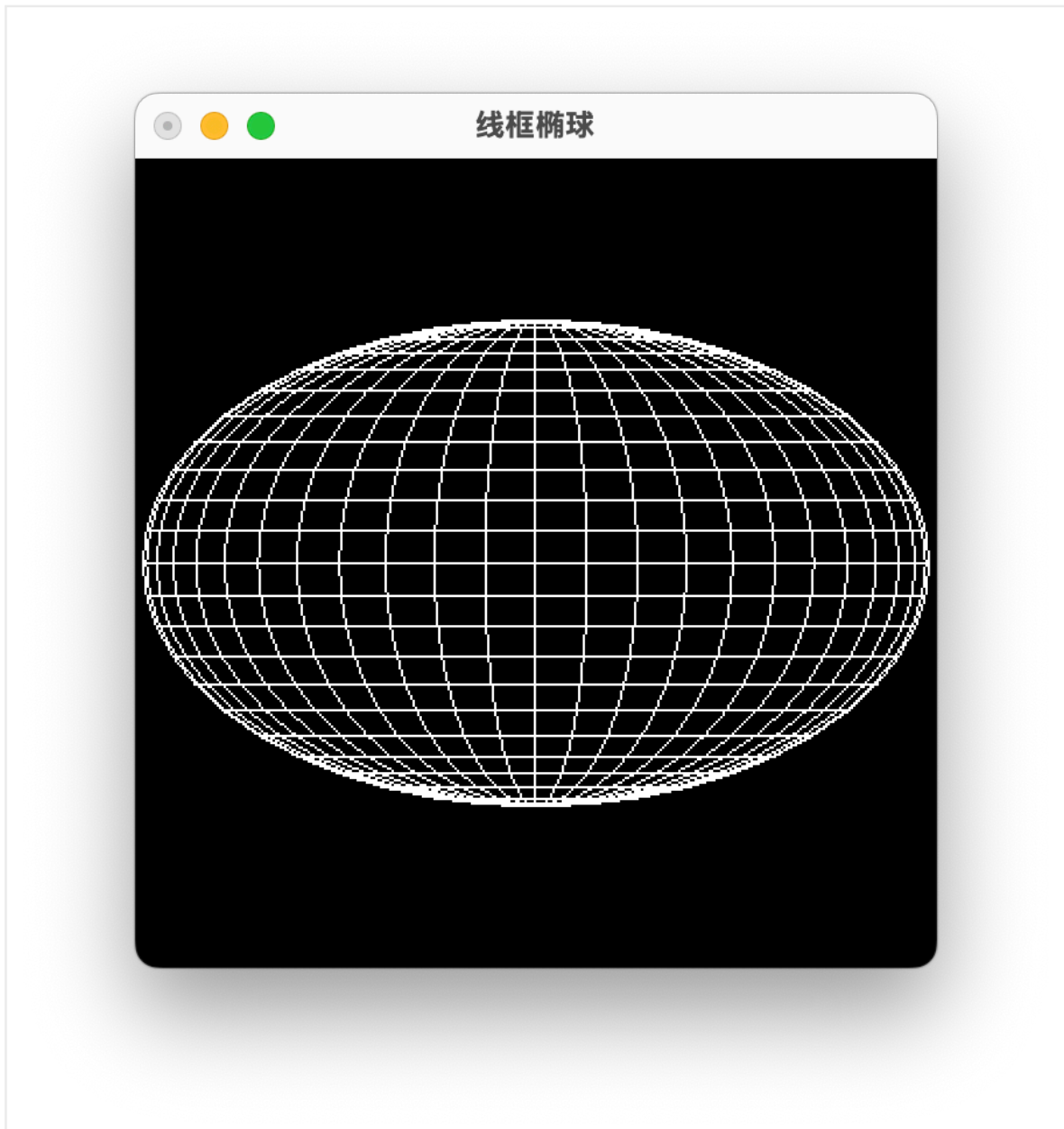
#include <GL/glu.h>
#include <GL/glut.h>

void Paint()
{
    glClear(GL_COLOR_BUFFER_BIT); // 初始化颜色缓冲区
    glLoadIdentity();           // 初始化矩阵
    glRotated(-90, 1, 0, 0);     // 两极为上下方向
    glScaled(0.98, 0.49, 0.6);   // 缩放 (左右、上下、前后)
    gluOrtho2D(-1, 1, -0.5, 0.5); // 裁剪窗口

    glutWireSphere(1, 48, 24); // 线框球 (半径、经线、纬线)
    glFlush();
}

```

```
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(400, 200); // 程序窗口大小
    glutCreateWindow("线框椭球"); // 窗口标题
    glutDisplayFunc(Paint); // 执行场景绘制函数
    glutMainLoop(); // 开始循环执行 OpenGL 命令
}
```



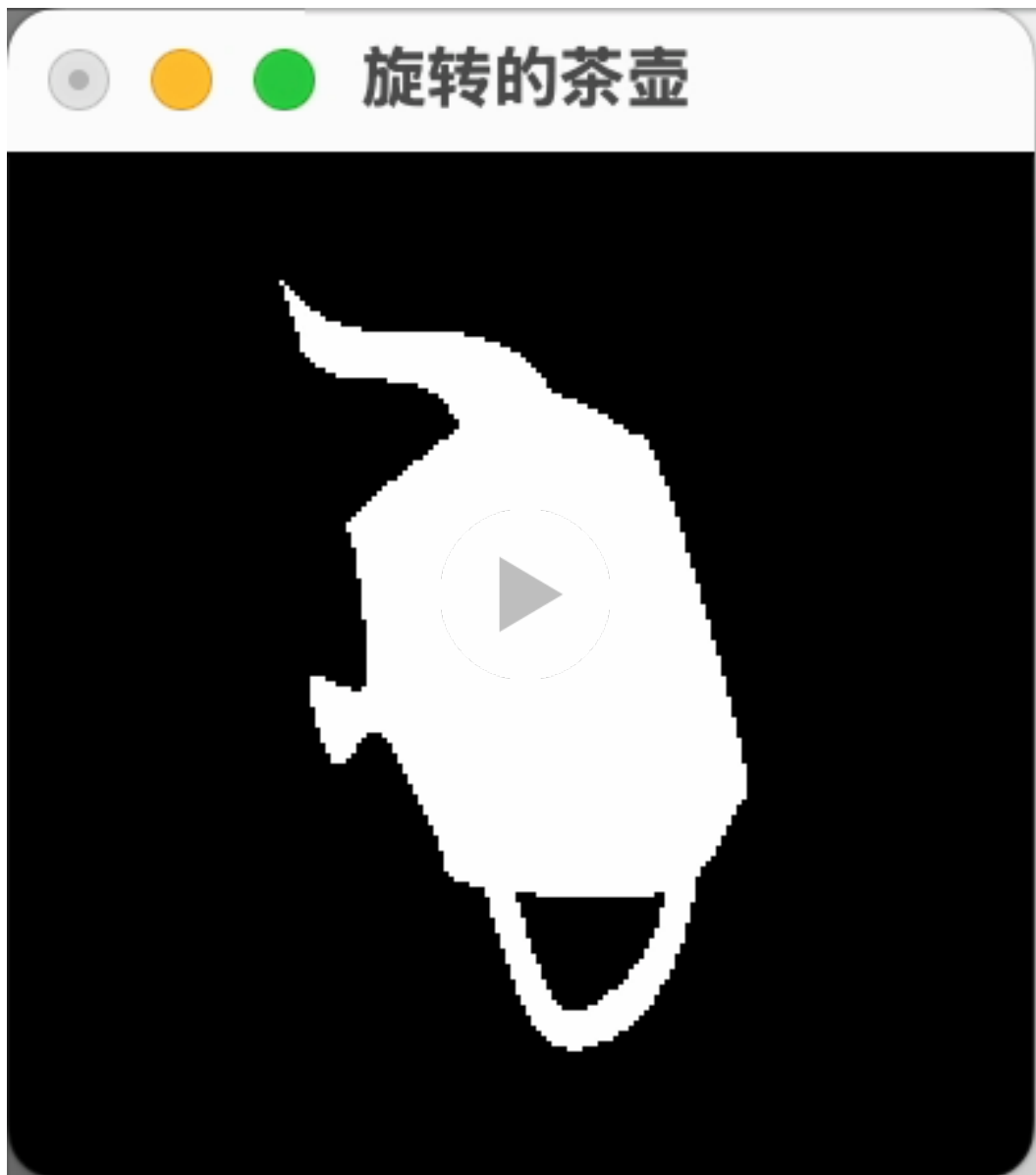
32. 请使用 OpenGL、GLU 和 GLUT 编写一个三维犹他茶壶程序。其中茶壶的半径为 1 单位，并远移 6.5 单位；观察体规定为：视场角=30 度，宽高比=1，近=1，远=100；程序窗口的大小为 (200, 200)，标题为“旋转的尤他茶壶”。茶壶绕 z 轴不断旋转，旋转的时间间隔为 25 毫秒，角度间隔为 2 度。注意旋转角度必须限定在 0 ~ 360 度以内。

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

int angle = 0;
void timer(int millis)
{
    angle = (angle + 2) % 360;
    glutPostRedisplay();
    glutTimerFunc(millis, timer, millis);
}

void Paint()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
    gluPerspective(30, 1, 1, 100);
    glTranslatef(0, 0, -6.5);
    glRotated(angle, 0, 0, 1);
    glutSolidTeapot(1);
    glutSwapBuffers();
}

int main(int argc, char *argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE);
    glutInitWindowSize(200, 200);
    glutCreateWindow("旋转的尤他茶壶");
    glutTimerFunc(25, timer, 25);
    glutDisplayFunc(Paint);
    glutMainLoop();
}
```



33. 请使用 OpenGL、GLU 和 GLUT 编写一个简单的多视口演示程序。要求：在屏幕窗口左下角的 1/4 部分显示一个红色的填充正三角形；在屏幕窗口右上角的 1/4 部分显示一个绿色的填充正方形；三角形和正方形的左下角顶点坐标值均为  $(0, 0)$ ，右下角顶点坐标值均为  $(1, 0)$ ；裁剪窗口均为  $(-0.1, -0.1) \sim (1.1, 1.1)$ ；程序窗口的大小为  $(200, 200)$ ，标题为“多视口演示”。

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

void Viewport(int x, int y, int w, int h)
{
    glViewport(x, y, w, h);
    glLoadIdentity();
    gluOrtho2D(-0.1, 1.1, -0.1, 1.1);
}

void Triangle()
```

```

{
    glBegin(GL_TRIANGLES);
    glVertex2d(0, 0);
    glVertex2d(1, 0);
    glVertex2d(0.5, 0.8660);
    glEnd();
}

void Paint()
{
    int w = glutGet(GLUT_WINDOW_WIDTH) / 2;
    int h = glutGet(GLUT_WINDOW_HEIGHT) / 2;
    glClear(GL_COLOR_BUFFER_BIT);
    Viewport(0, 0, w, h);
    glColor3f(1, 0, 0);
    Triangle();
    Viewport(w, h, w, h);
    glColor3f(0, 1, 0);
    glRectd(0, 0, 1, 1);
    glFlush();
}

int main(int argc, char *argv[])
{
    glutInit(&argc, argv);
    glutInitWindowSize(200, 200);
    glutCreateWindow("多视口演示");
    glutDisplayFunc(Paint);
    glutMainLoop();
}

```

