

OpenGL实例：几何变换

作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/>

更多请查看：[计算机图形学](#)

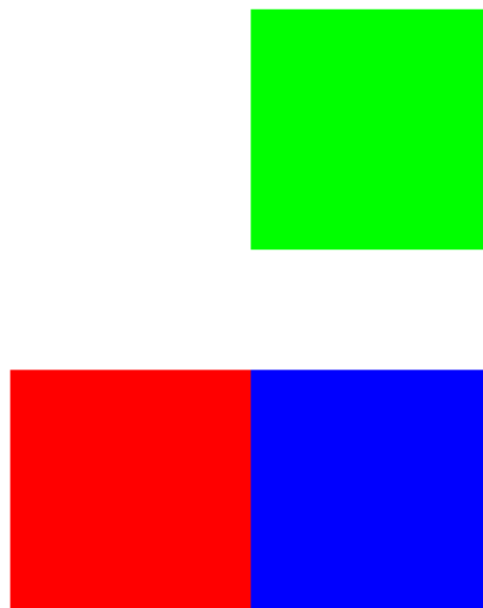
1. 平移

```
#include <GL/glut.h>
void init(void)
{
    glClearColor(1.0, 1.0, 1.0, 0.0); //指定清除颜色
    //初始化视景体
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-5.0, 5.0, -5.0, 5.0); //设置显示的范围是X:-5.0~5.0, Y:-5.0~5.0
    glMatrixMode(GL_MODELVIEW);
}
//绘制中心在原点，边长为2的正方形
void drawSquare(void)
{
    glBegin(GL_POLYGON); //顶点指定需要按逆时针方向
    glVertex2f(-1.0f, -1.0f); //左下点
    glVertex2f(1.0f, -1.0f); //右下点
    glVertex2f(1.0f, 1.0f); //右上点
    glVertex2f(-1.0f, 1.0f); //左上点
    glEnd();
}
void myDraw1(void)
{
    glClear(GL_COLOR_BUFFER_BIT); //清空
    glLoadIdentity(); //将当前矩阵设为单位矩阵
    glColor3f(1.0, 0.0, 0.0);
    drawSquare(); //在原点处绘制边长为2红色正方形
    glTranslatef(2.0, 3.0, 0.0); //向右移动2单位，向上移动3单位
    glColor3f(0.0, 1.0, 0.0);
    drawSquare(); //绘制边长为2绿色正方形
    glTranslatef(0.0, -3.0, 0.0); //再向下移动3单位
    glColor3f(0.0, 0.0, 1.0);
    drawSquare(); //绘制边长为2蓝色正方形
    glFlush();
}
void myDraw2(void)
```

```
{
    glClear(GL_COLOR_BUFFER_BIT); //清空
    glLoadIdentity(); //将当前矩阵设为单位矩阵
    glColor3f(1.0, 0.0, 0.0);
    drawSquare(); //在坐标原点处绘制边长为2红色正方形
    glPushMatrix();
    glTranslatef(2.0, 3.0, 0.0); //向右移动2单位，向上移动3单位
    glColor3f(0.0, 1.0, 0.0);
    drawSquare(); //绘制边长为2绿色正方形
    glPopMatrix();
    glTranslatef(2.0, 0.0, 0.0); //再向右移动2单位
    glColor3f(0.0, 0.0, 1.0);
    drawSquare(); //绘制边长为2蓝色正方形
    glFlush();
}

void main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(0, 0);
    glutInitWindowSize(600, 600);
    glutCreateWindow("Translate函数示例");
    init();
    glutDisplayFunc(myDraw1);
    glutMainLoop();
}
```

图形:



2. 旋转

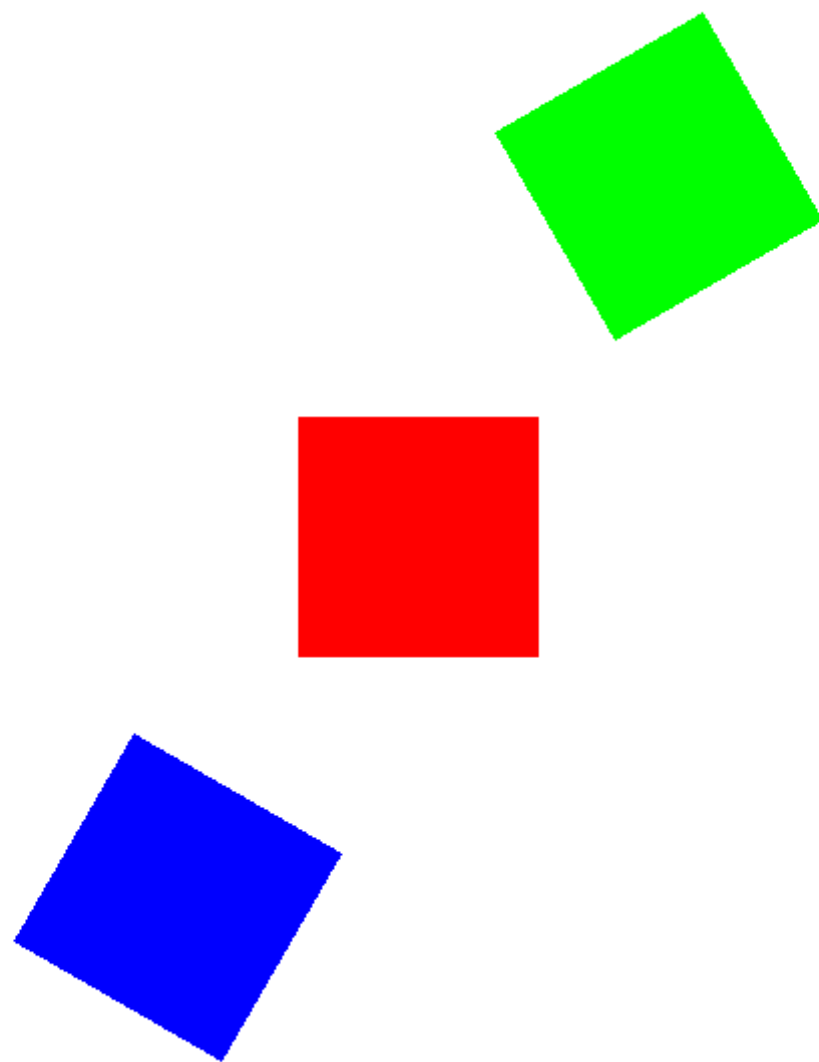
```

#include <GL/glut.h>
void init(void)
{
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-5.0, 5.0, -5.0, 5.0); //设置显示的范围是X:-5.0~5.0, Y:-5.0~5.0
    glMatrixMode(GL_MODELVIEW);
}
void drawSquare(void) //绘制中心在原点，边长为2的正方形
{
    glBegin(GL_POLYGON); //顶点指定需要按逆时针方向
    glVertex2f(-1.0f, -1.0f); //左下点
    glVertex2f(1.0f, -1.0f); //右下点
    glVertex2f(1.0f, 1.0f); //右上点
    glVertex2f(-1.0f, 1.0f); //左上点
    glEnd();
}
void myDraw1(void)
{
    glClear(GL_COLOR_BUFFER_BIT); //清空
    glLoadIdentity(); //将当前矩阵设为单位矩阵
    glColor3f(1.0, 0.0, 0.0);
    drawSquare(); //在原点处绘制边长为2红色正方形
    glTranslatef(2.0, 3.0, 0.0); //向右移动2单位，向上移动3单位
    glRotatef(30, 0.0, 0.0, 1.0); //顺时针旋转30角度
    glColor3f(0.0, 1.0, 0.0);
    drawSquare(); //绘制边长为2绿色正方形
    glLoadIdentity(); //将当前矩阵设为单位矩阵
    glTranslatef(-2.0, -3.0, 0.0); //向左移动2单位，向下移动3单位
    glRotatef(-30, 0.0, 0.0, 1.0); //逆时针旋转30角度
    glColor3f(0.0, 0.0, 1.0);
    drawSquare(); //绘制边长为2蓝色正方形
    glFlush();
}
void myDraw2(void)
{
    glClear(GL_COLOR_BUFFER_BIT); //清空
    glLoadIdentity(); //将当前矩阵设为单位矩阵
    glColor3f(1.0, 0.0, 0.0);
    drawSquare(); //在原点处绘制边长为2红色正方形
    glPushMatrix(); //把当前矩阵压入堆栈
    glTranslatef(2.0, 3.0, 0.0); //向右移动2单位，向上移动3单位
    glRotatef(30, 0.0, 0.0, 1.0); //顺时针旋转30角度
    glColor3f(0.0, 1.0, 0.0);
    drawSquare(); //绘制边长为2绿色正方形
    glPopMatrix(); //从堆栈栈顶弹出一个矩阵为当前矩阵
    glTranslatef(-2.0, -3.0, 0.0); //向左移动2单位，向下移动3单位
}

```

```
    glRotatef(-30, 0.0, 0.0, 1.0); //逆时针旋转30角度
    glColor3f(0.0, 0.0, 1.0);
    drawSquare(); //绘制边长为2蓝色正方形
    glFlush();
}
void main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(0, 0);
    glutInitWindowSize(600, 600);
    glutCreateWindow("Rotate函数示例");
    init();
    glutDisplayFunc(myDraw1);
    glutMainLoop();
}
```

图形:



3. 缩放

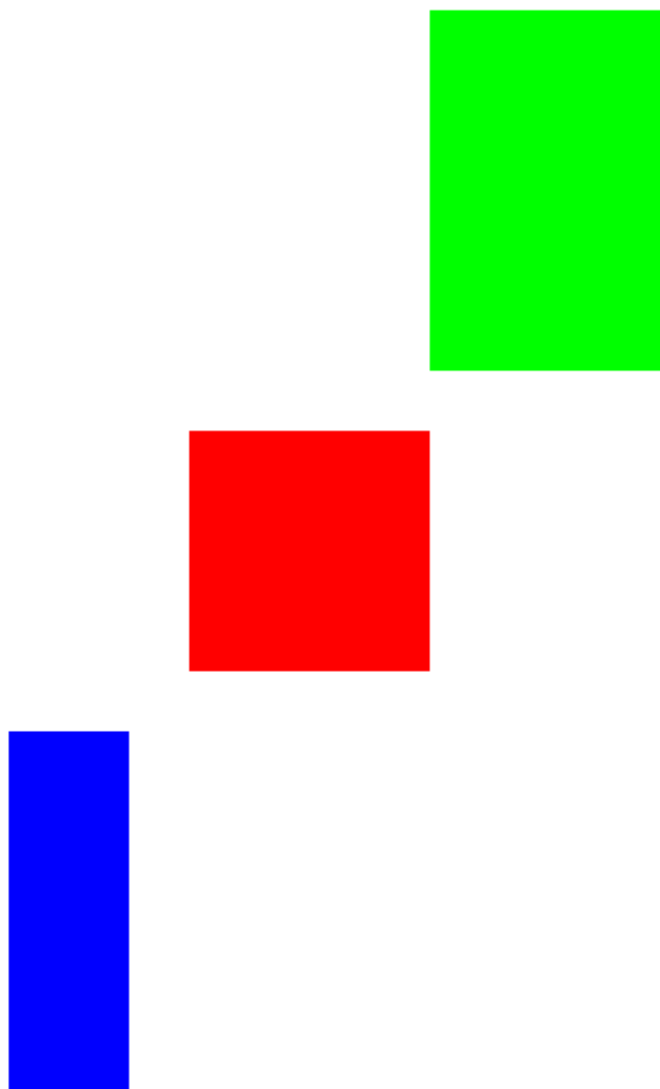
```

#include <GL/glut.h>
void init(void)
{
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-5.0, 5.0, -5.0, 5.0); //设置显示的范围是X:-5.0~5.0, Y:-5.0~5.0
    glMatrixMode(GL_MODELVIEW);
}
void drawSquare(void) //绘制中心在原点，边长为2的正方形
{
    glBegin(GL_POLYGON); //顶点指定需要按逆时针方向
    glVertex2f(-1.0f, -1.0f); //左下点
    glVertex2f(1.0f, -1.0f); //右下点
    glVertex2f(1.0f, 1.0f); //右上点
    glVertex2f(-1.0f, 1.0f); //左上点
    glEnd();
}
void myDraw1(void)
{
    glClear(GL_COLOR_BUFFER_BIT); //清空
    glLoadIdentity(); //将当前矩阵设为单位矩阵
    glColor3f(1.0, 0.0, 0.0);
    drawSquare(); //在原点处绘制边长为2红色正方形
    glTranslatef(2.0, 3.0, 0.0); //向右移动2单位，向上移动3单位
    glScalef(1.0, 1.5, 1.0); //X和Z方向保持不变，Y方向放大为原来的1.5倍
    glColor3f(0.0, 1.0, 0.0);
    drawSquare(); //绘制边长为2绿色正方形
    glLoadIdentity(); //将当前矩阵设为单位矩阵
    glTranslatef(-2.0, -3.0, 0.0); //向左移动2单位，向下移动3单位
    glScalef(0.5, 1.5, 1.0); //Z方向保持不变，X方向缩小为原来的0.5倍，Y方向放大为原来的1.5倍
    glColor3f(0.0, 0.0, 1.0);
    drawSquare(); //绘制边长为2蓝色正方形
    glFlush();
}
void myDraw2(void)
{
    glClear(GL_COLOR_BUFFER_BIT); //清空
    glLoadIdentity(); //将当前矩阵设为单位矩阵
    glColor3f(1.0, 0.0, 0.0);
    drawSquare(); //在原点处绘制边长为2红色正方形
    glPushMatrix(); //把当前矩阵压入堆栈
    glTranslatef(2.0, 3.0, 0.0); //向右移动2单位，向上移动3单位
    glScalef(1.0, 1.5, 1.0); //X和Z方向保持不变，Y方向放大为原来的1.5倍
    glColor3f(0.0, 1.0, 0.0);
    drawSquare(); //绘制边长为2绿色正方形
    glPopMatrix(); //从堆栈栈顶弹出一个矩阵为当前矩阵
    glTranslatef(-2.0, -3.0, 0.0); //向左移动2单位，向下移动3单位
}

```

```
    glScalef(0.5, 1.5, 1.0); //Z方向保持不变，X方向缩小为原来的0.5倍，Y方向放大为原来的1.5倍
    glColor3f(0.0, 0.0, 1.0);
    drawSquare(); //绘制边长为2蓝色正方形
    glFlush();
}
void main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(0, 0);
    glutInitWindowSize(600, 600);
    glutCreateWindow("Scale函数示例");
    init();
    glutDisplayFunc(myDraw1);
    glutMainLoop();
}
```

图形:



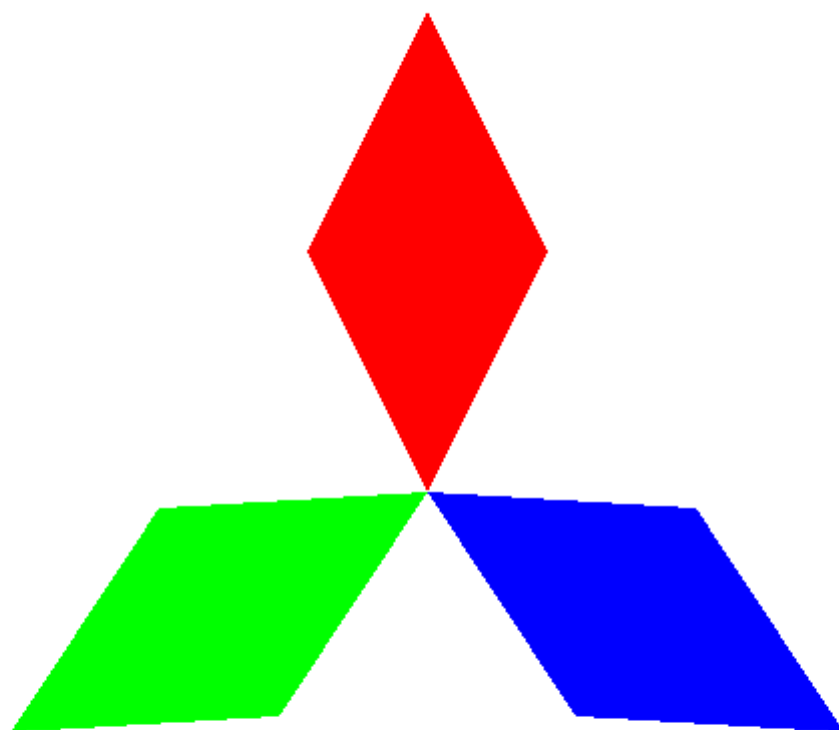
4. 菱形旋转变换

```

#include <GL/glut.h>
void init(void)
{
    glClearColor(1.0, 1.0, 1.0, 0.0); //指定清除颜色
    //初始化视景体
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-5.0, 5.0, -5.0, 5.0); //设置显示的范围是X:-5.0~5.0, Y:-5.0~5.0
    glMatrixMode(GL_MODELVIEW);
}
//绘制中心在原点的菱形
void drawDiamond(void)
{
    glBegin(GL_POLYGON); //顶点指定需要按逆时针方向
    glVertex2f(2.0f, -1.0f); //下点
    glVertex2f(4.0f, 0.0f); //右点
    glVertex2f(2.0f, 1.0f); //上点
    glVertex2f(-0.0f, 0.0f); //左点
    glEnd();
}
void wrrDraw(void)
{
    glClear(GL_COLOR_BUFFER_BIT); //清空
    glLoadIdentity(); //将当前菱形设为单位菱形
    glRotatef(90, 0.0, 0.0, 1.0); //逆时针旋转90角度
    glColor3f(1.0, 0.0, 0.0);
    drawDiamond(); //绘制红色菱形
    glRotatef(-120, 0.0, 0.0, 1.0); //顺时针旋转120角度
    glColor3f(0.0, 0.0, 1.0);
    drawDiamond(); //绘制蓝色菱形
    glRotatef(-120, 0.0, 0.0, 1.0); //顺时针旋转120角度
    glColor3f(0.0, 1.0, 0.0);
    drawDiamond(); //绘制绿色菱形
    glFlush();
}
void main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(0, 0);
    glutInitWindowSize(600, 600);
    glutCreateWindow("菱形函数的绘制");
    init();
    glutDisplayFunc(wrrDraw);
    glutMainLoop();
}

```

图形:



5. 综合实验

```

#include <GL/glut.h>
void init(void)
{
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-5.0, 5.0, -5.0, 5.0); //设置显示的范围是X:-5.0~5.0, Y:-5.0~5.0
    glMatrixMode(GL_MODELVIEW);
}
void drawSquare(void) //绘制中心在原点，边长为2的正方形
{
    glBegin(GL_POLYGON); //顶点指定需要按逆时针方向
    glVertex2f(-1.0f, -1.0f); //左下点
    glVertex2f(1.0f, -1.0f); //右下点
    glVertex2f(1.0f, 1.0f); //右上点
    glVertex2f(-1.0f, 1.0f); //左上点
    glEnd();
}
void myDraw(void)
{
    glClear(GL_COLOR_BUFFER_BIT); //清空
    glLoadIdentity(); //将当前矩阵设为单位矩阵
    glPushMatrix();
    glTranslatef(0.0f, 2.0f, 0.0f);
    glScalef(3.0, 0.5, 1.0);
    glColor3f(1.0, 0.0, 0.0);
    drawSquare(); //上面红色矩形
    glPopMatrix();
    glPushMatrix();
    glTranslatef(-3.0, 0.0, 0.0);
    glPushMatrix();
    glRotatef(45.0, 0.0, 0.0, 1.0);
    glColor3f(0.0, 1.0, 0.0);
    drawSquare(); //中间左菱形
    glPopMatrix();
    glTranslatef(3.0, 0.0, 0.0);
    glPushMatrix();
    glRotatef(45.0, 0.0, 0.0, 1.0);
    glColor3f(0.0, 0.7, 0.0);
    drawSquare(); //中间中菱形
    glPopMatrix();
    glTranslatef(3.0, 0.0, 0.0);
    glPushMatrix();
    glRotatef(45.0, 0.0, 0.0, 1.0);
    glColor3f(0.0, 0.4, 0.0);
    drawSquare(); //中间右菱形
    glPopMatrix();
    glPopMatrix();
}

```

```
    glTranslatef(0.0, -3.0, 0.0);
    glScalef(4.0, 1.5, 1.0);
    glColor3f(0.0, 0.0, 1.0);
    drawSquare(); //下面蓝色矩形
    glFlush();
}
void main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(0, 0);
    glutInitWindowSize(600, 600);
    glutCreateWindow("几何变换函数综合示例");
    init();
    glutDisplayFunc(myDraw);
    glutMainLoop();
}
```

图形:

