

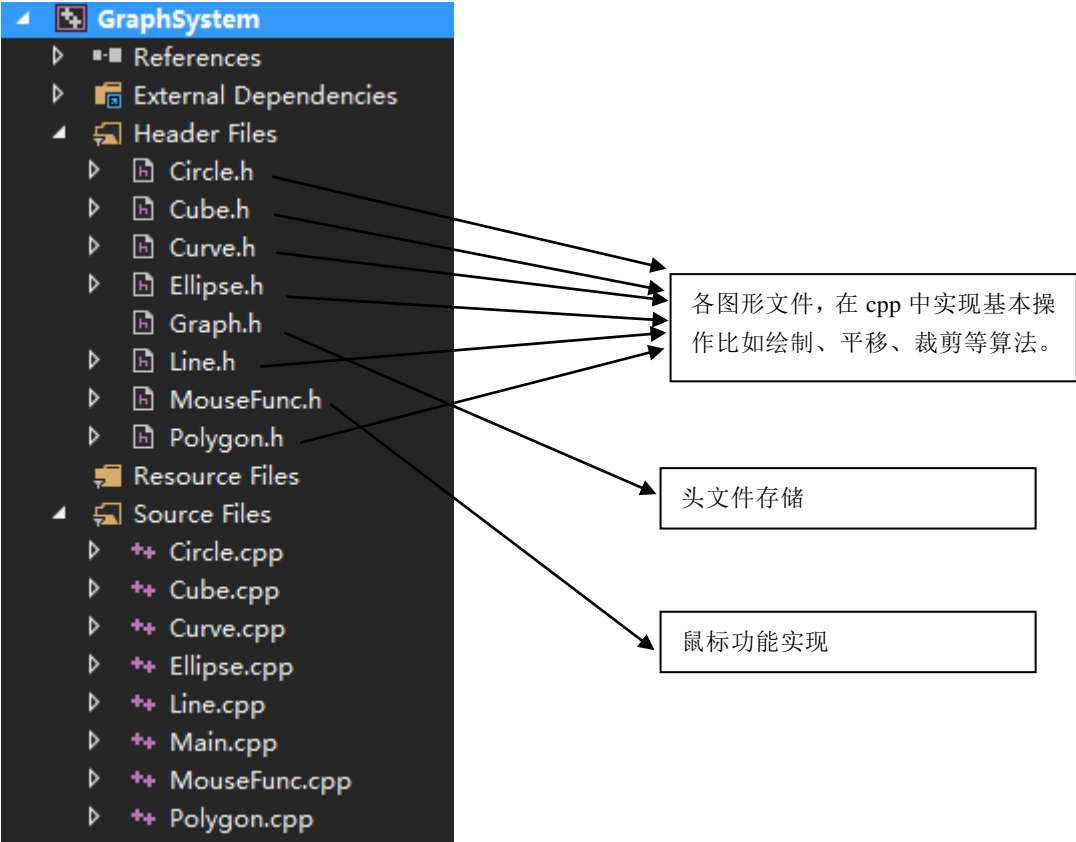
计算机图形学 系统使用说明书（功能+操作说明）

作者姓名 李东昊

联系方式: 961171432@qq.com 15905192178

(南京大学 计算机科学与技术系, 南京 210093)

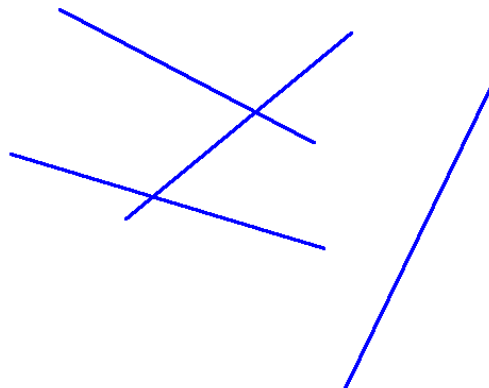
1 功能



1.1 图形绘制

1.1.1 直线

图形学图形编辑处理系统



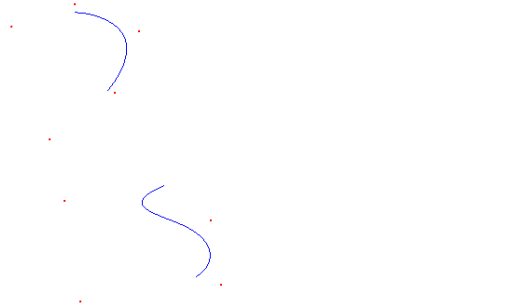
```
5 void drawLineDDA(int xa, int ya, int xb, int yb) {  
6     glColor3f(0.0, 0.0, 1.0);  
7     glPointSize(3.0f);  
8  
9     int dx = xb - xa;  
10    int dy = yb - ya;  
11    int steps, k;  
12  
13    float xIncrement, yIncrement, x = xa, y = ya;  
14  
15    if (abs(dx) > abs(dy))  
16        steps = abs(dx);  
17    else  
18        steps = abs(dy);  
19  
20    xIncrement = (float)(dx) / (float)(steps);  
21    yIncrement = (float)(dy) / (float)(steps);  
22  
23    for (k = 0; k < steps; k++) {  
24        glBegin(GL_POINTS);  
25        glVertex2i(round(x), round(y));  
26        glEnd();  
27        x += xIncrement;  
28        y += yIncrement;  
29    }  
30    glFlush();  
31 }
```

绘制直线的 DDA 算法

1.1.2 曲线

图形学图形编辑处理系统

- □



B 样条绘制算法

```

void drawCurveBsp(int *pts_x, int *pts_y, int pts_num, int n) {
    glPointSize(1.2);
    glLineWidth(1.2);
    glColor3f(0.0f, 0.0f, 1.0f);
    float f1, f2, f3, f4;
    float deltaT = 1.0 / n;
    float T;
    int num_seg = pts_num - 3;

    glBegin(GL_LINE_STRIP);
    for (int num = 0; num < num_seg; num++) {
        for (int i = 0; i <= n; i++) {

            T = i * deltaT;

            f1 = (-T*T*T + 3 * T*T - 3 * T + 1) / 6.0;
            f2 = (3 * T*T*T - 6 * T*T + 4) / 6.0;
            f3 = (-3 * T*T*T + 3 * T*T + 3 * T + 1) / 6.0;
            f4 = (T*T*T) / 6.0;

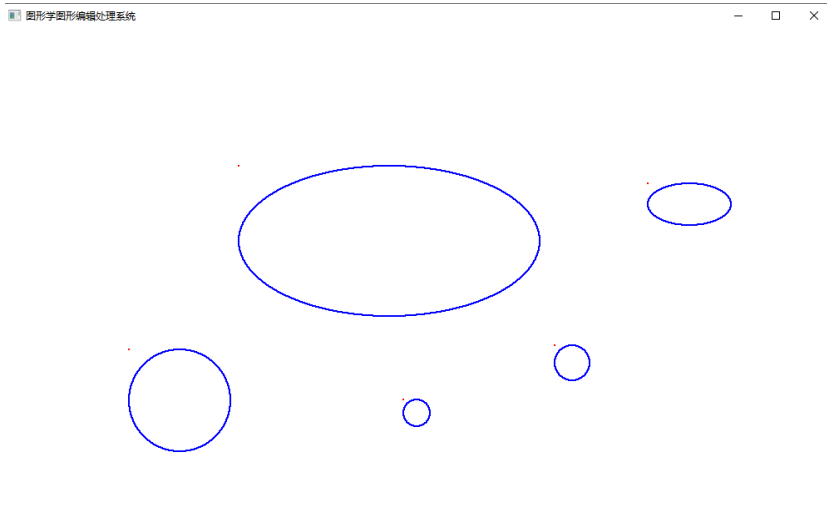
            glVertex2f(f1*pts_x[num] + f2*pts_x[num + 1] + f3*pts_x[num + 2] + f4*pts_x[num + 3],
                      f1*pts_y[num] + f2*pts_y[num + 1] + f3*pts_y[num + 2] + f4*pts_y[num+3]);

        }
    }

    glEnd();
    glFlush();
}

```

1.1.3 椭圆与圆



```
void drawEllipseMidPoint(int xCenter, int yCenter, int Rx, int Ry) {
    int Rx2 = Rx*Rx, Ry2 = Ry*Ry;
    int twoRx2 = 2 * Rx2, twoRy2 = 2 * Ry2;
    int p;
    int x = 0, y = Ry;
    int px = 0, py = twoRx2*y;

    ellipsePlotPoints(xCenter, yCenter, x, y);

    p = Round((Ry2 - (Rx2*Ry) + (0.25*Rx2)));
    while (px < py) {
        x++;
        px += twoRy2;
        if (p < 0)
            p += Ry2 + px;
        else {
            --y;
            py -= twoRx2;
            p += Ry2 + px - py;
        }
        ellipsePlotPoints(xCenter, yCenter, x, y);
    }

    p = Round((Ry2*(x + 0.5)*(x + 0.5) + Rx2*(y - 1)*(y - 1) - Rx2*Ry2));
    while (y > 0) {
        --y;
        py -= twoRx2;
        if (p > 0)
            p += Rx2 - py;
        else {
            ++x;
            px += twoRy2;
            p += Rx2 - py + px;
        }
        ellipsePlotPoints(xCenter, yCenter, x, y);
    }

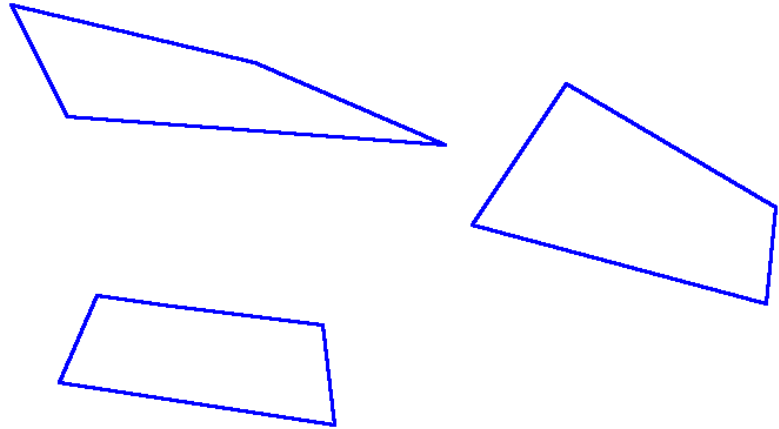
    glFlush();
}
```

中心椭圆算法

1.1.4 多边形

图形学图形编辑处理系统

— □ ×

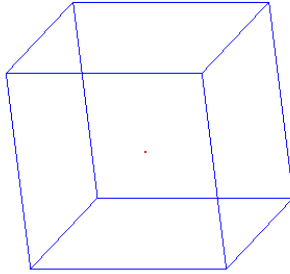


多边形绘制算法

```
void drawPolygon(float *pts_x, float *pts_y, int num) {  
    if (num < 3) return;  
    for (int i = 0; i < num - 1; i++) {  
        drawLineDDA(pts_x[i], pts_y[i], pts_x[i + 1], pts_y[i + 1]);  
    }  
    drawLineDDA(pts_x[num - 1], pts_y[num - 1], pts_x[0], pts_y[0]);  
}
```

1.1.5 六面体（未能实现消隐）

图形学图形编程处理系统



```
void drawCube(float centerX, float centerY, float centerZ) {
    glPushMatrix();
    glRotatef(20, 0, 1, 0);
    glRotatef(20, 1, 0, 0);

    glColor3f(0.0f, 0.0f, 1.0f);
    glLineWidth(1.2f);
    glBegin(GL_LINES);
    glVertex3f(centerX + HD, centerY + HD, centerZ + HD);
    glVertex3f(centerX - HD, centerY + HD, centerZ + HD);
    glVertex3f(centerX - HD, centerY + HD, centerZ + HD);
    glVertex3f(centerX - HD, centerY - HD, centerZ + HD);
    glVertex3f(centerX - HD, centerY - HD, centerZ + HD);
    glVertex3f(centerX + HD, centerY - HD, centerZ + HD);
    glVertex3f(centerX + HD, centerY - HD, centerZ + HD);
    glVertex3f(centerX + HD, centerY + HD, centerZ + HD);

    glVertex3f(centerX + HD, centerY + HD, centerZ - HD);
    glVertex3f(centerX - HD, centerY + HD, centerZ - HD);
    glVertex3f(centerX - HD, centerY + HD, centerZ - HD);
    glVertex3f(centerX - HD, centerY - HD, centerZ - HD);
    glVertex3f(centerX - HD, centerY - HD, centerZ - HD);
    glVertex3f(centerX + HD, centerY - HD, centerZ - HD);
    glVertex3f(centerX + HD, centerY - HD, centerZ - HD);
    glVertex3f(centerX + HD, centerY + HD, centerZ - HD);

    glVertex3f(centerX + HD, centerY + HD, centerZ + HD);
    glVertex3f(centerX + HD, centerY + HD, centerZ - HD);
    glVertex3f(centerX - HD, centerY + HD, centerZ + HD);
    glVertex3f(centerX - HD, centerY + HD, centerZ - HD);
    glVertex3f(centerX - HD, centerY - HD, centerZ + HD);
    glVertex3f(centerX - HD, centerY - HD, centerZ - HD);
    glVertex3f(centerX + HD, centerY - HD, centerZ + HD);
    glVertex3f(centerX + HD, centerY - HD, centerZ - HD);
    glEnd();
    glFlush();

    glRotatef(-20, 0, 1, 0);
    glRotatef(-20, 1, 0, 0);
    glPopMatrix();
}
```

六面体绘制

1.2 图形编辑

```
//平移
⊕void lineTransEvent() { ... }

⊕void polygonTransEvent() { ... }

⊕void ellipseTransEvent() { ... }

⊕void curveTransEvent() { ... }

//旋转
⊕void lineRotateEvent() { ... }

⊕void polygonRotateEvent() { ... }

⊕void ellipseRotateEvent() { ... }

⊕void curveRotateEvent() { ... }

//缩放
⊕void lineScalEvent() { ... }

⊕void polygonScalEvent() { ... }

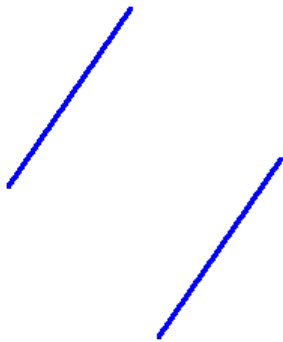
⊕void ellipsescalEvent() { ... }

⊕void curveScalEvent() { ... }

//裁剪
⊕void polygonClipEvent(int button, int state, int x, int y) { ... }
```

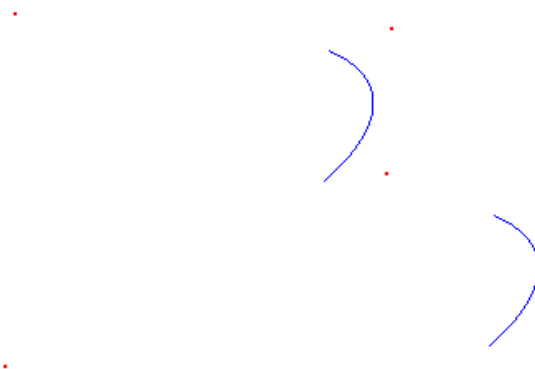
1.2.1 平移

直线平移:



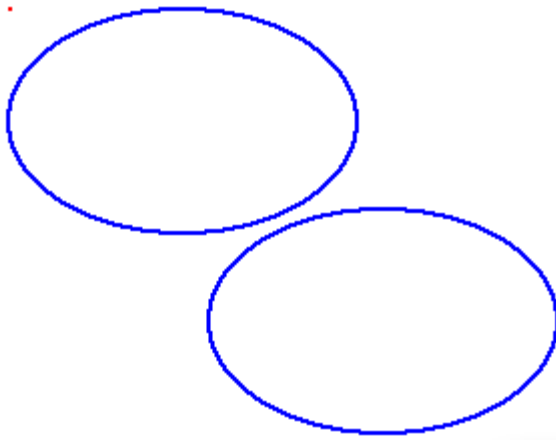
```
C:\WINDOWS\system32\cmd.exe
直线 0:(167,141) --> (85,260)
请输入要平移的直线编号
0
请输入x轴变化量
100
请输入y轴变化量
100
```

曲线平移:



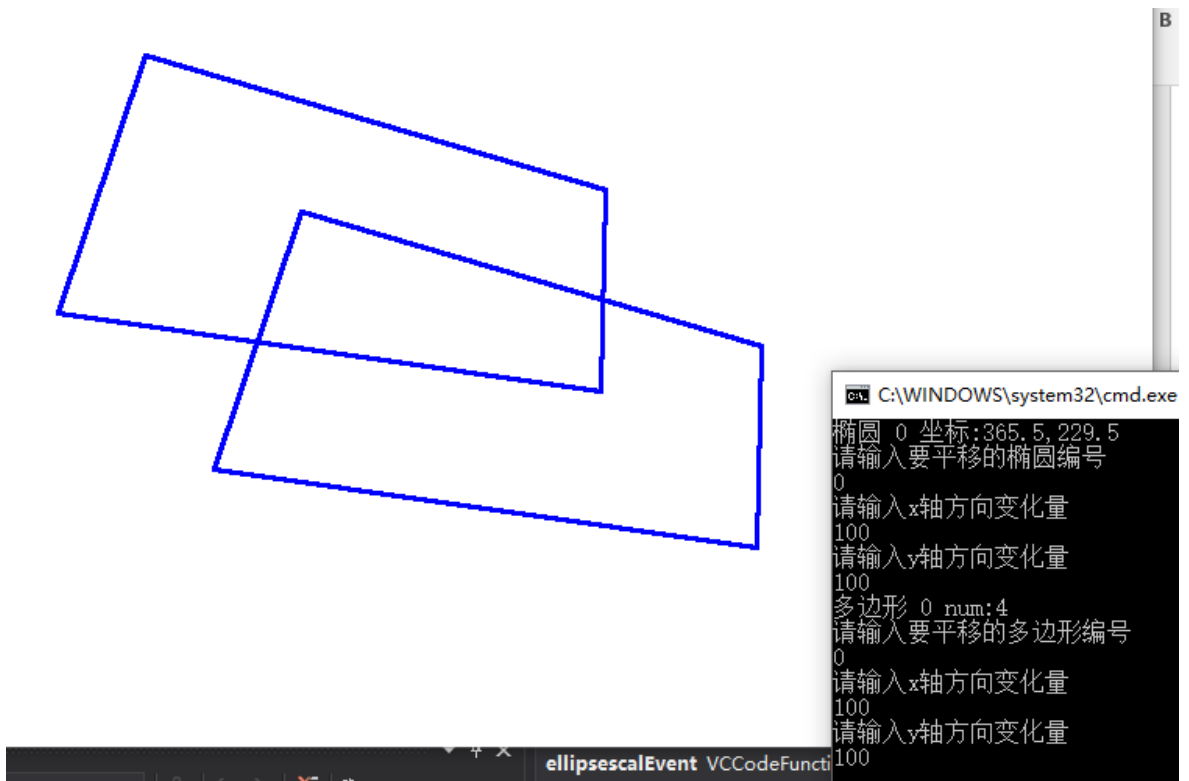
```
C:\WINDOWS\system32\cmd.exe
曲线 0
请输入要平移的曲线编号
0
请输入x方向变化量
100
请输入y轴方向变化量
100
```

椭圆平移:

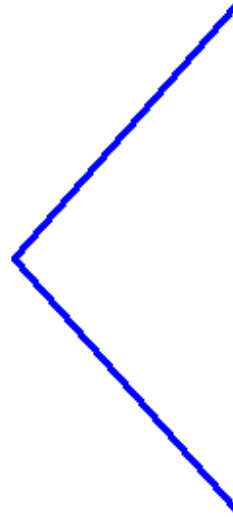


```
C:\WINDOWS\system32\cmd.exe
椭圆 0 坐标:365.5,229.5
请输入要平移的椭圆编号
0
请输入x轴方向变化量
100
请输入y轴方向变化量
100
```

多边形平移:



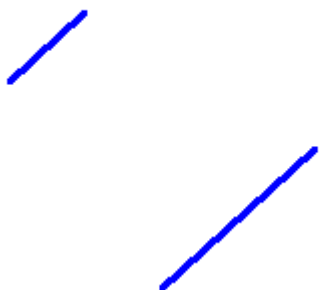
1.2.2 旋转

 图形学图形编辑处理系统

```
C:\WINDOWS\system32\cmd.exe
直线 0: (345, 176) --> (459, 305)
请输入要旋转的直线编号
0
请输入旋转角度
90
```

1.2.3 缩放

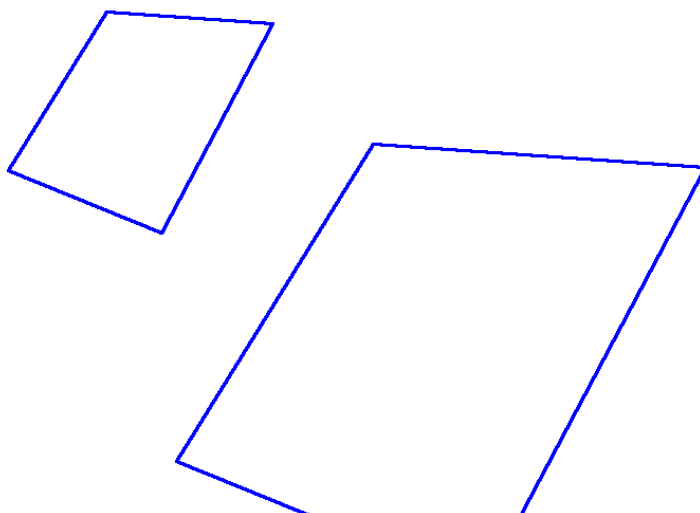
直线缩放



```
C:\WINDOWS\system32\cmd.exe
直线 0:(229,136) --> (152,206)
请输入要缩放的直线编号
0
请输入x轴方向比例
0.5
请输入y轴方向比例
0.5
0.5
```

多边形缩放:

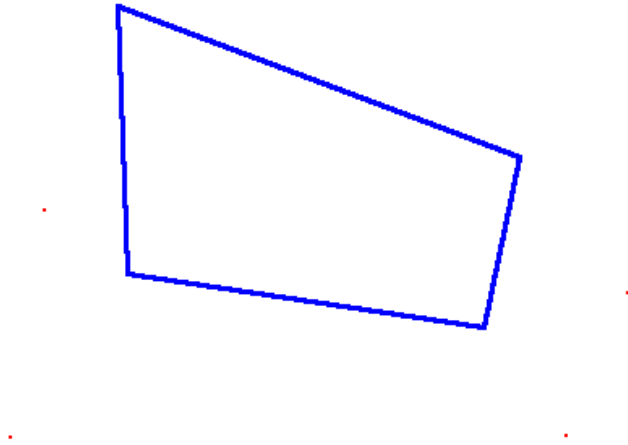
图形学图形编辑处理系统



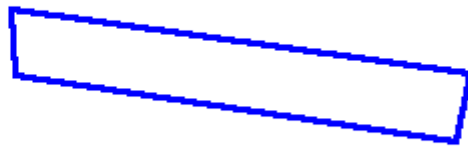
```
Main.cpp
开始 插入 设
Times New F 小五
B I U abc x
C:\WINDOWS\system32\cmd.exe
直线 0:(229,136) --> (152,206)
请输入要缩放的直线编号
0
请输入x轴方向比例
0.5
请输入y轴方向比例
0.5
多边形 0 num:4
请输入要缩放的多边形编号
0
请输入x轴方向比例
2
请输入y轴方向比例
2
```

1.2.4 裁剪

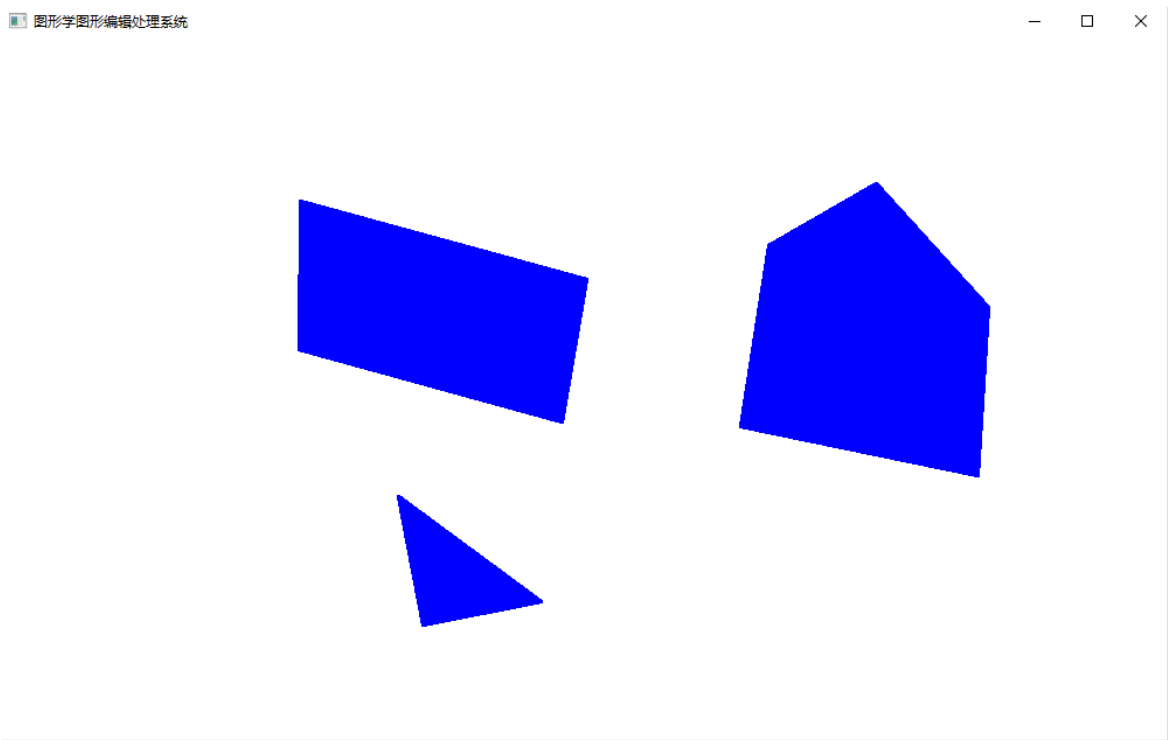
多边形裁剪:



裁剪后：



1.2.5 多边形填充



1.2.6 保存最终并未实现。

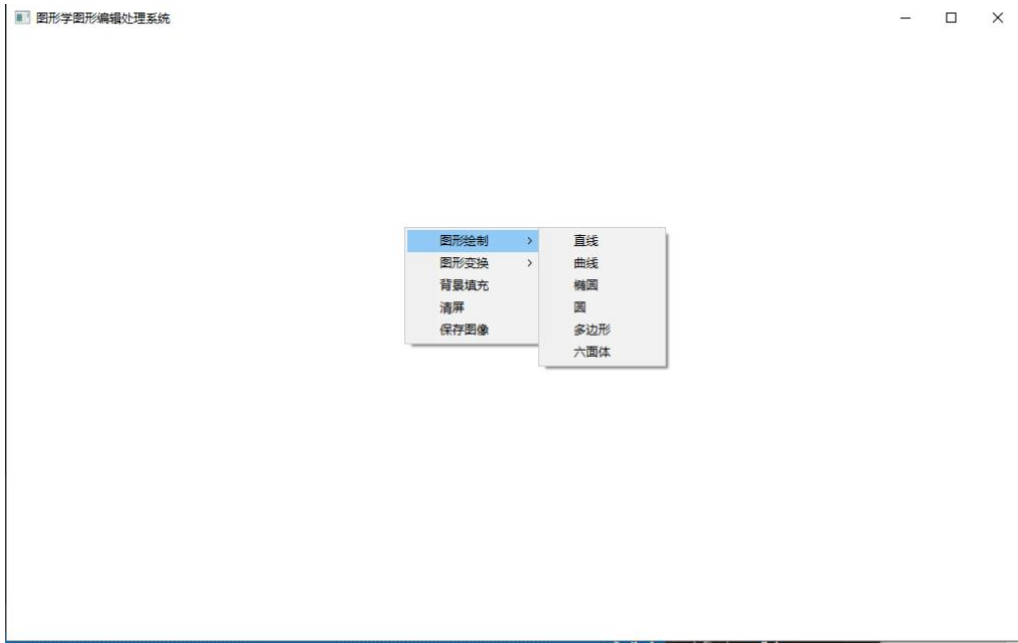
2 操作说明

```
//菜单子选项
+void processSubMenuDraw(int option){ ... }
+void processMenuEvents(int option){ ... }
+void processChangeMenuEvents(int option){ ... }
+void processClipmenuEvents(int option){ ... }
+void processRotateMenuEvents(int option){ ... }
+void processScalMenuEvents(int option){ ... }
+void processTransformMenuEvents(int option){ ... }

//菜单选项
+void createMenus(){ ... }
```

菜单栏

主页面如下：



2.1 图形绘制

右键弹出菜单栏，选择六种图形进行绘制

2.1.1 直线

鼠标点击为起点，拖动至任意位置，即终点，绘制即可完成。

2.1.2 曲线

鼠标点击取一系列点后，按下鼠标滑轮中键，即可进行绘制。

2.1.3 椭圆/圆

同直线，鼠标拖动，即可绘制。

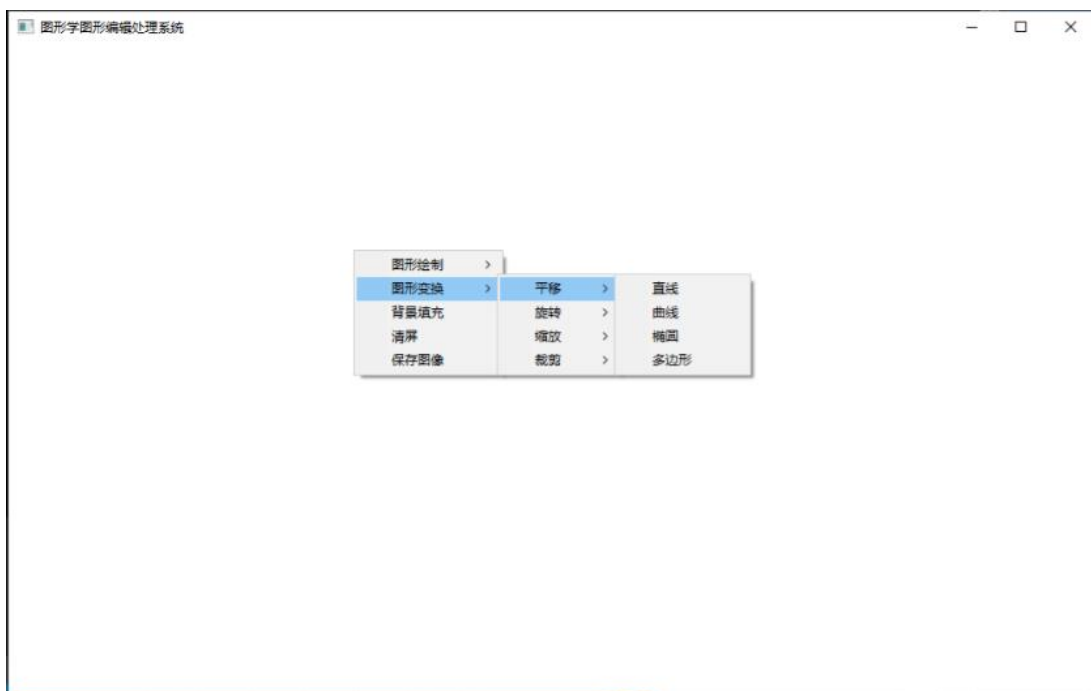
2.1.4 多边形

鼠标点击取一系列点后，按下滑轮中键，完成绘制。

2.1.5 六面体

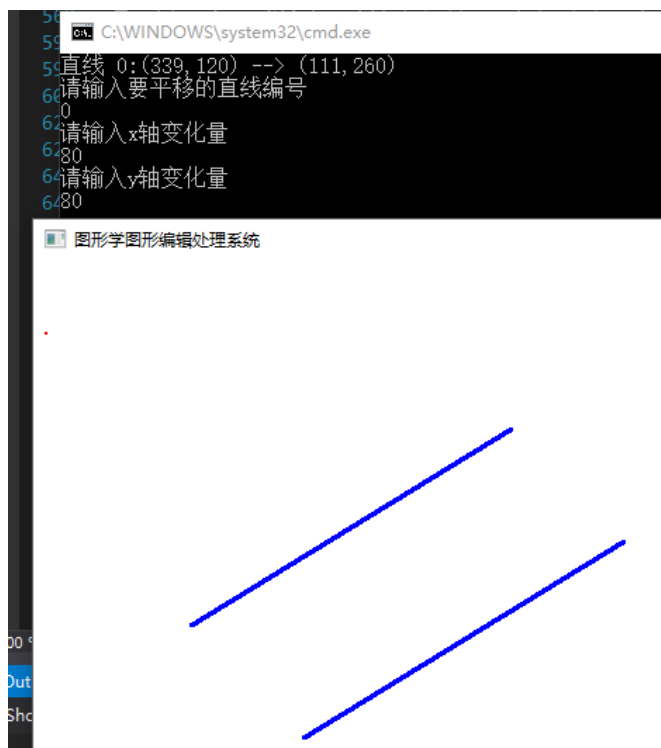
鼠标点击，即可绘制系统规定大小的六面体。

2.2 图形编辑



2.2.1 平移

鼠标选择平移对象，控制台会提示输入编号以及 xy 轴变化量



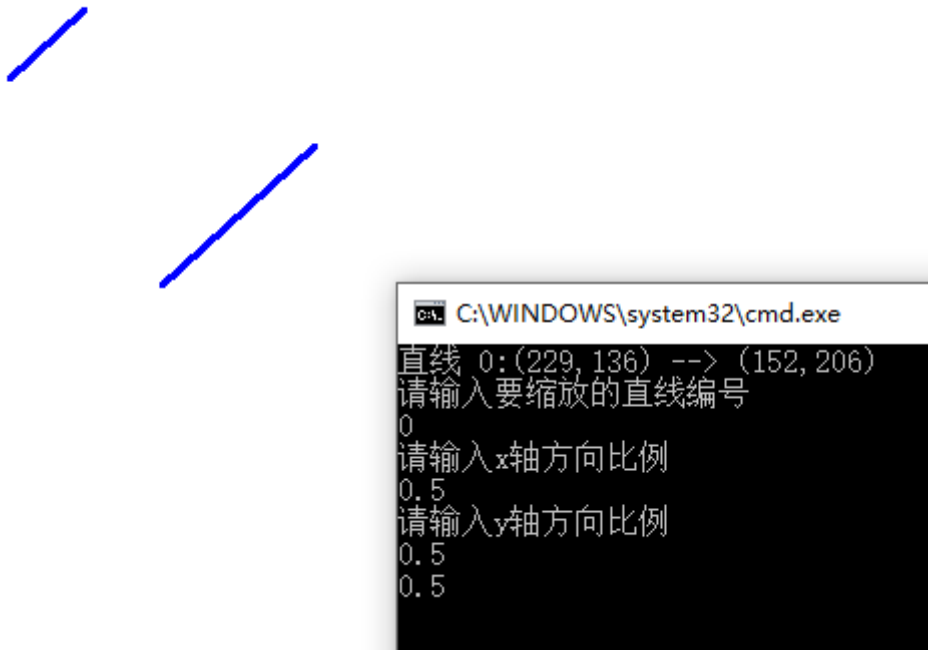
2.2.2 旋转

同平移一样，选择编号后输入旋转角度



2.2.3 缩放

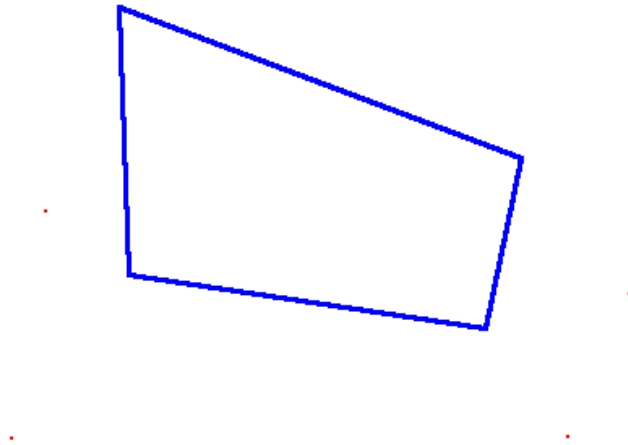
依旧是选择缩放图形对象，以及 xy 轴上的变化比例



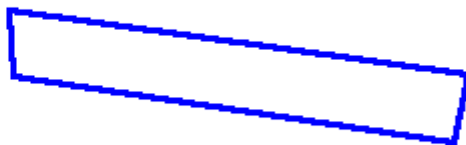
2.2.4 裁剪

裁剪只实现了多边形裁剪：

首先绘制多边形，然后选择功能“裁剪-多边形”，在多边形上利用鼠标点击规定裁剪区域，按鼠标中键进行裁剪

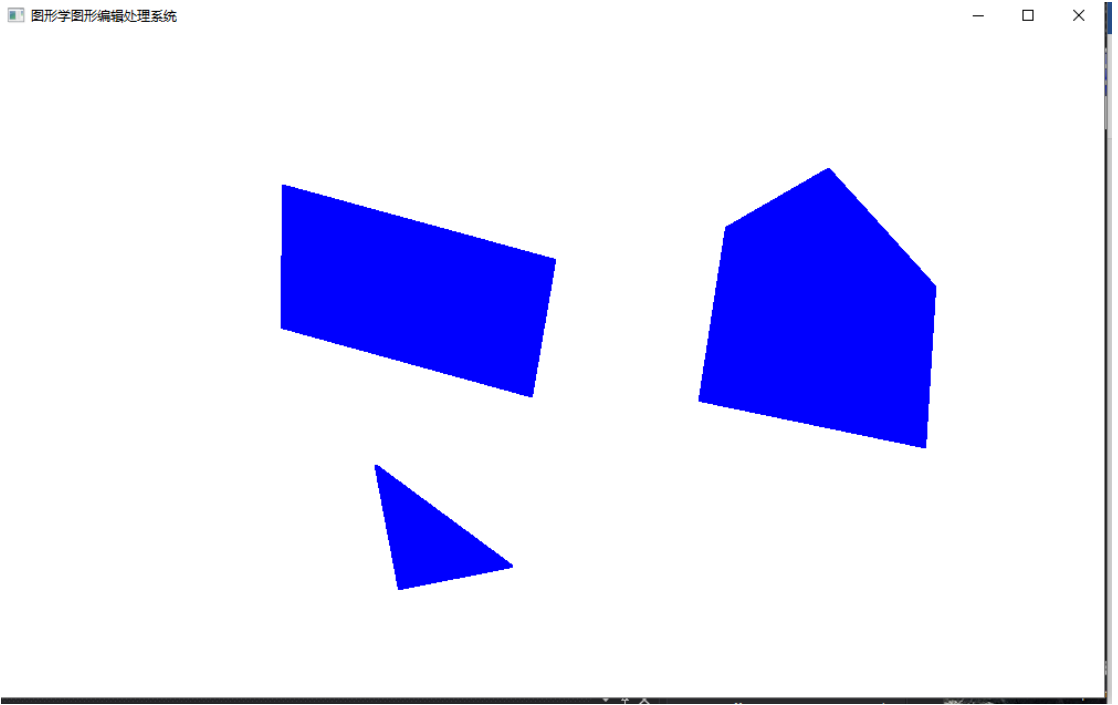


裁剪后：



2.2.5 多变形填充

绘制多边形图形后，选择填充功能，按鼠标左键，进行填充



2.2.6 保存

保存并未实现，实在抱歉。

致谢 在此,向对本文的工作给予支持和建议的同学表示感谢。