《计算机图形学》系统使用说明书

2021 年 1 月 22 日

1 系统开发环境

- 虚拟机　Oracle VM VirtualBox 6.0.12
- Ubuntu　18.04.1
- IDE/编辑器　vscode+Python extension
- PyQt5　5.9.2
- pillow　7.0.0

2 通过cli绘图

cli有如下指令：

- 重置画布：resetCanvas width height
  清空当前画布，并重新设置宽高width, height: int
  $100 <= width, height <= 1000$
- 保存画布：saveCanvas name
  将当前画布保存为位图
  name.bmp name: string
- 设置画笔颜色：setColor R G B
  R, G, B: int
  $0 <= R, G, B <= 255$
- 绘制线段：drawLine id x0 y0 x1 y1 algorithm
  id: string, 图元编号，每个图元的编号是唯一的
  x0, y0, x1, y1: int, 起点、终点坐标
  algorithm: string, 绘制使用的算法，包括”DDA”和”Bresenham”
- 绘制多边形：drawPolygon id x0 y0 x1 y1 x2 y2 ... algorithm
  id: string, 图元编号，每个图元的编号是唯一的
  x0, y0, x1, y1, x2, y2 ... : int, 顶点坐标
  algorithm: string, 绘制使用的算法，包括”DDA”和”Bresenham”

- 绘制椭圆（中点圆生成算法）：drawEllipse id x0 y0 x1 x1
  id: string, 图元编号，每个图元的编号是唯一的
  x0, y0, x1, y1: int, 椭圆矩形包围框的左上角和右下角顶点坐标

- 绘制曲线：drawCurve id x0 y0 x1 y1 x2 y2 ... algorithm
  id: string, 图元编号，每个图元的编号是唯一的
  x0, y0, x1, y1, x2, y2 ... : int, 控制点坐标
  algorithm: string, 绘制使用的算法，包括"Bezier"和"B-spline"，其中"B-spline"要求为三次（四阶）均匀B样条曲线，曲线不必经过首末控制点

- 图元平移：translate id dx dy
  id: string, 要平移的图元编号
  dx, dy: int, 平移向量

- 图元旋转：rotate id x y r
  id: string, 要旋转的图元编号
  x, y: int, 旋转中心
  r: int, 顺时针旋转角度（°）

- 图元缩放：scale id x y s
  id: string, 要缩放的图元编号
  x, y: int, 缩放中心
  s: float, 缩放倍数

- 对线段裁剪：clip id x0 y0 x1 y1 algorithm
  id: string, 要裁剪的线段编号
  x0, y0, x1, y1: int, 裁剪窗口的左上角和右下角顶点坐标
  algorithm: string, 裁剪使用的算法，包括"Cohen-Sutherland"和"Liang-Barsky"

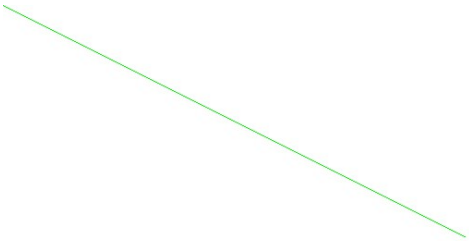
2.1 绘制直线

2.1.1 DDA算法绘制直线

使用指令

```
resetCanvas 600 600
setColor 0 255 0
drawLine Line1 0 0 500 250 DDA
saveCanvas DDA
```

绘制过点$(0,0),(500,250)$的直线，DDA.bmp文件如下图所示：

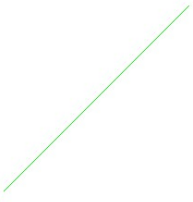


2.1.2 Bresenham算法绘制直线

使用指令

```
resetCanvas 600 600
setColor 0 255 0
drawLine Line2 200 500 400 300 Bresenham
saveCanvas Bresenham
```

绘制过点$(0,0),(500,250)$的直线，DDA.bmp文件如下图所示：

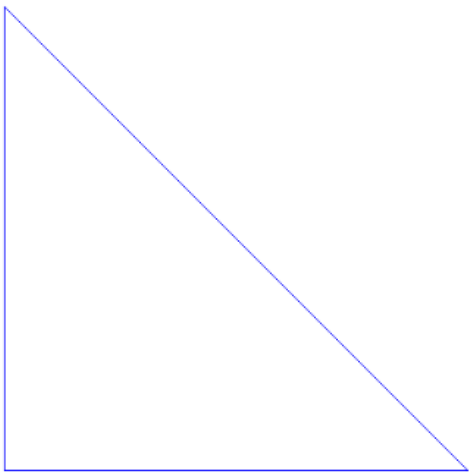


2.2 绘制多边形

2.2.1 DDA算法绘制多边形

使用指令

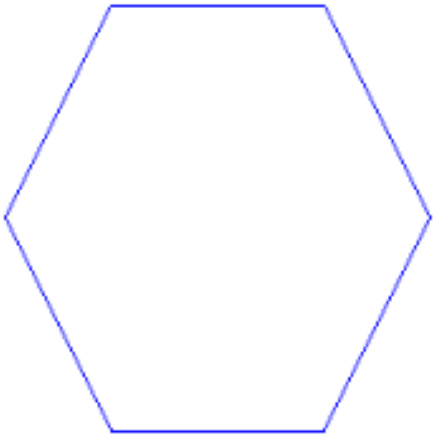
```
resetCanvas 600 600
setColor 0 0 255
drawPolygon polygon1 100 100 500 500 100 500 DDA
saveCanvas DDA_polygon
```



2.2.2 Bresenham算法绘制多边形

使用指令

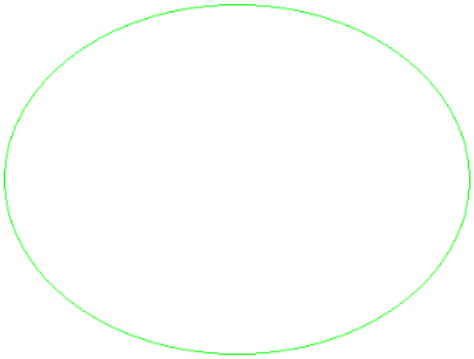
```
resetCanvas 600 600
setColor 0 0 255
drawPolygon polygon2 200 100 300 100 350 200 300 300 200 300 150
    200 Bresenham
saveCanvas Bresenham_polygon
```



2.3 绘制椭圆

使用指令

```
resetCanvas 600 600
setColor 0 255 0
drawEllipse ellipse1 100 100 500 400
saveCanvas ellipse
```



2.4 绘制曲线

2.4.1 Bezier算法绘制曲线

使用指令

```
resetCanvas 600 600
setColor 255 0 0
drawCurve curve1 50 200 100 100 150 200 Bezier
drawCurve curve2 50 400 100 300 150 400 200 300 Bezier
saveCanvas Bezier
```

2.4.2 B-spline算法绘制曲线

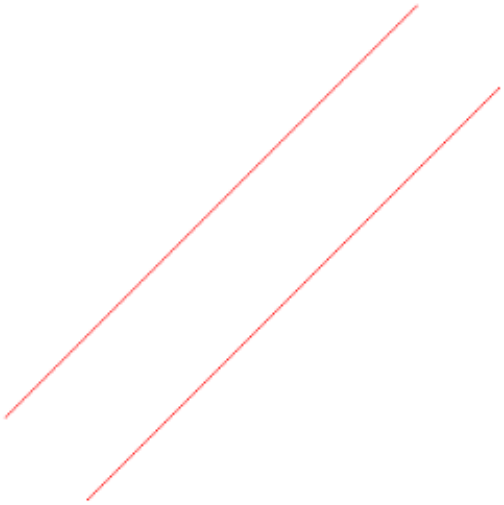
使用指令

```
resetCanvas 600 600
setColor 0 0 255
drawCurve curve3 250 400 300 300 350 400 400 300 B-spline
drawCurve curve4 250 200 300 50 350 250 400 100 450 200 B-spline
saveCanvas B-spline
```

2.5 图元平移

使用指令

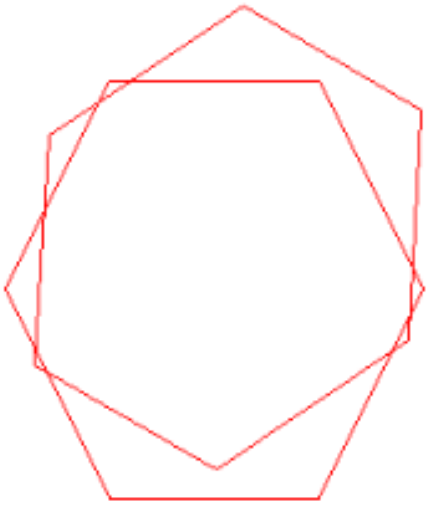
```
resetCanvas 600 600
setColor 0 255 0
drawLine line3 500 250 250 500 Bresenham
translate line3 -50 -50
drawLine line4 500 250 250 500 Bresenham
saveCanvas translate
```



2.6 图元旋转

使用指令

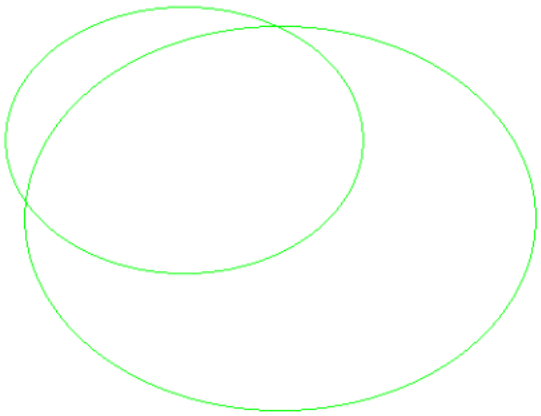
```
resetCanvas 600 600
setColor 255 0 0
drawPolygon polygon3 200 100 300 100 350 200 300 300 200 300 150
    200 Bresenham
rotate polygon3 300 200 30
drawPolygon polygon4 200 100 300 100 350 200 300 300 200 300 150
    200 Bresenham
saveCanvas rotate
```



2.7 图元缩放

使用指令

```
resetCanvas 600 600
setColor 0 255 0
drawEllipse ellipse3 100 100 500 400
scale ellipse3 50 50 0.7
drawEllipse ellipse4 100 100 500 400
saveCanvas scale
```



2.8 对线段裁剪

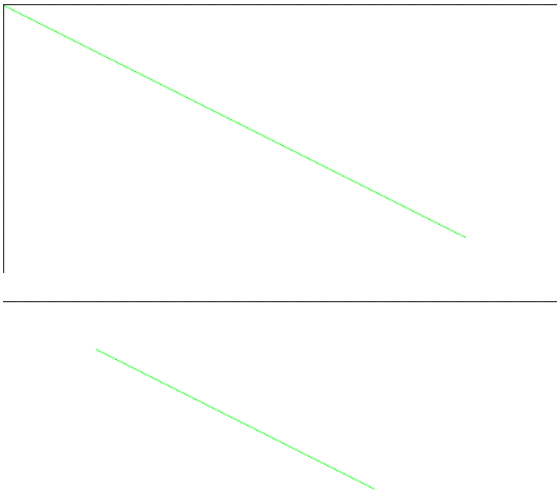
2.8.1 Cohen-Sutherland算法

使用指令

```
resetCanvas 600 600
setColor 0 255 0
```

```
drawLine line5 0 0 500 250 DDA
clip line5 50 50 400 200 Cohen-Sutherland
saveCanvas Cohen-Sutherland
```

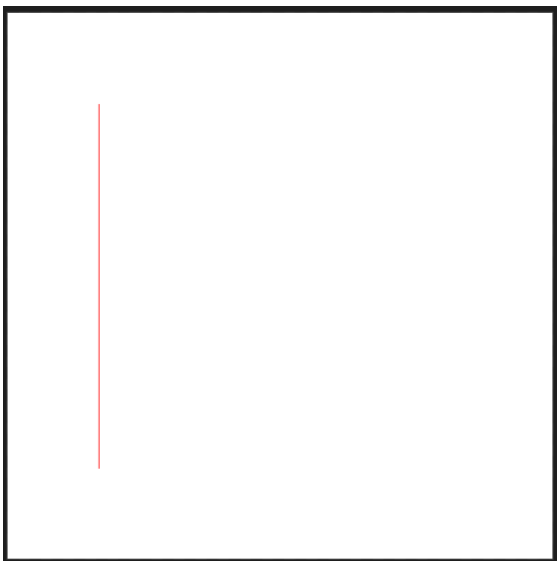
第一张图为裁剪前，第二张图图为裁剪后

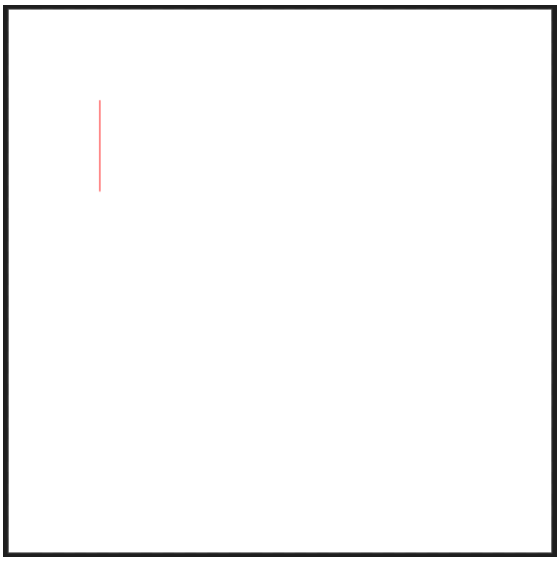


2.8.2 Liang-Barsky算法

使用指令

```
resetCanvas 600 600
setColor 0 255 0
drawLine line6 100 100 100 500 Bresenham
clip line6 0 0 200 200 Liang-Barsky
saveCanvas Liang-Barsky
```



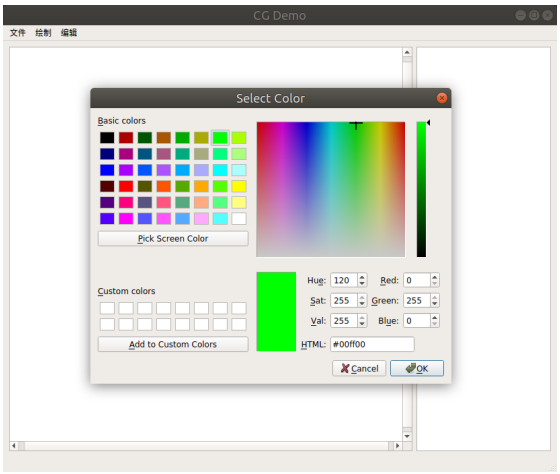


3 通过gui绘图

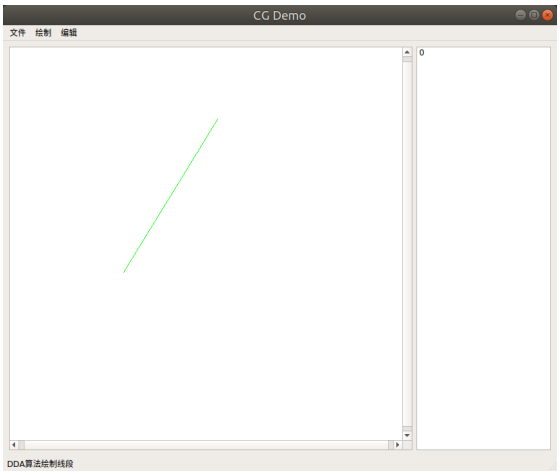
通过python cg_gui.py使用图形系统

3.1 gui设置画笔

通过鼠标点选图形界面工具栏，文件– >设置画笔，选择颜色

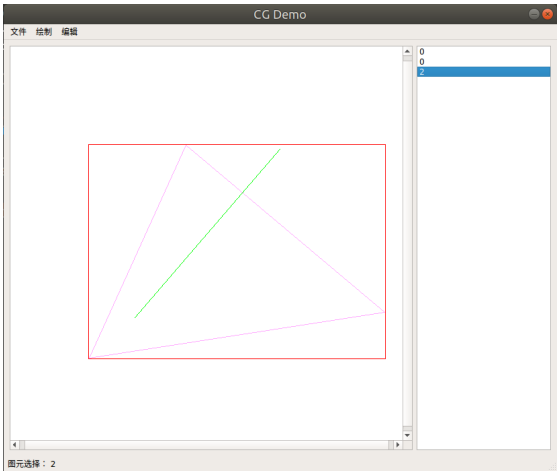


再选用绘制功能即可绘制出选择好的颜色

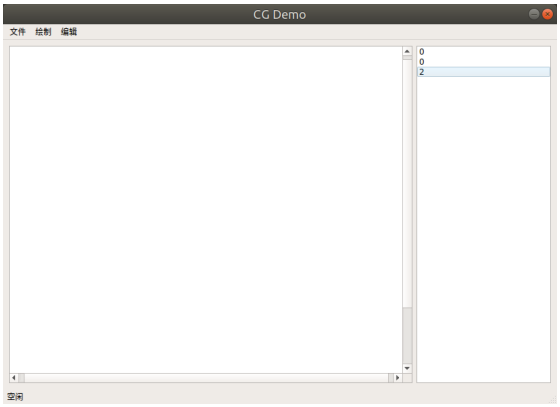


3.2 gui重置画布

先绘制出一些图像，如下所示



再通过鼠标点选界面工具栏，文件– >重置画布，设置好新的高和宽（范围：100-1000）,示例中高为500，宽为600

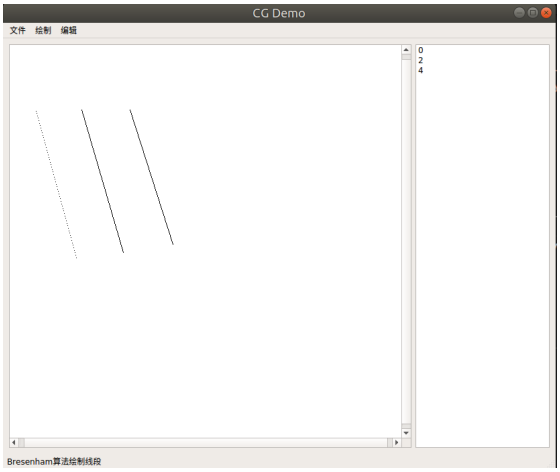


3.3 gui绘制直线

通过鼠标点选图形界面工具栏，绘制– >线段选择算法绘制直线

操作方法：在画布上按住鼠标即可绘制直线，鼠标第一次点下所在位置为第一个点，鼠标左键放开所在位置是第二个点

下图从左到右分别使用了Naive、DDA、Bresenham算法绘制直线

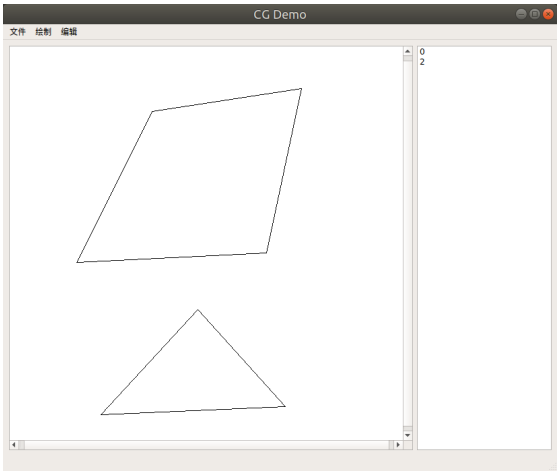


3.4 gui绘制多边形

通过鼠标点选图形界面工具栏，绘制– >线段选择算法绘制多边形

具体操作：在画布上点选画图。注意：需要首尾相连才能完成一个多边形的绘制（首尾点坐标之差的绝对值小于十个像素即完成一个多边形）

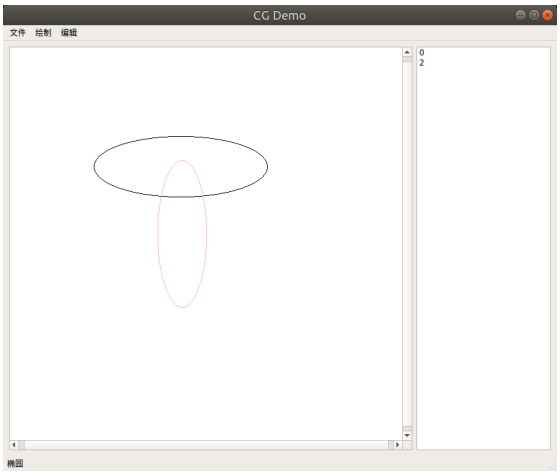
上半部分是DDA算法绘制的四边形；下半部分是Bresenham算法绘制的三角形



3.5 gui绘制椭圆

通过鼠标点选图形界面工具栏，绘制– >椭圆绘制椭圆

操作方式与画直线的方法相同

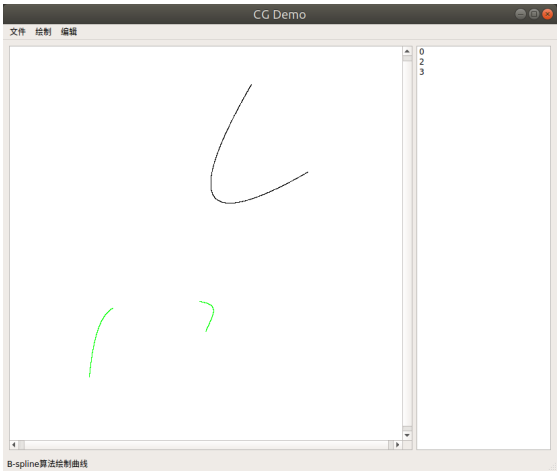


3.6 gui绘制曲线

通过鼠标点选图形界面工具栏，绘制– >曲线选择算法绘制椭圆

操作方式：在画布上点击选定控制顶点即可

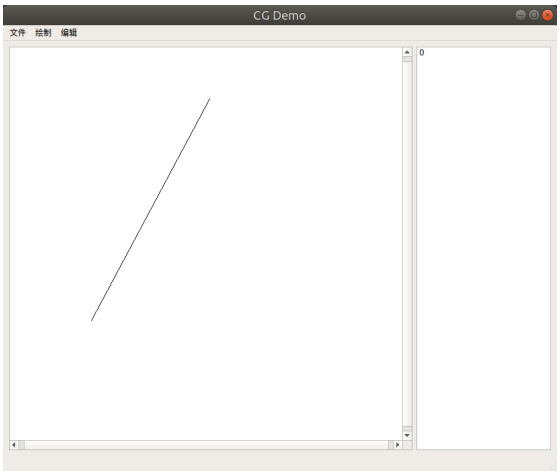
黑色为Bezier算法绘制结果，绿色为B-spline算法绘制结果



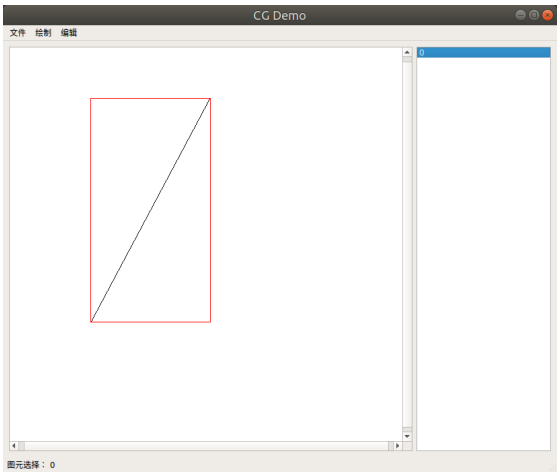
3.7 gui平移

操作方式：先点选窗口右边的编号，选择图形；再通过鼠标点选图形界面工具栏，编辑– >平移在画布上拖拽来平移图元。如下图所示：

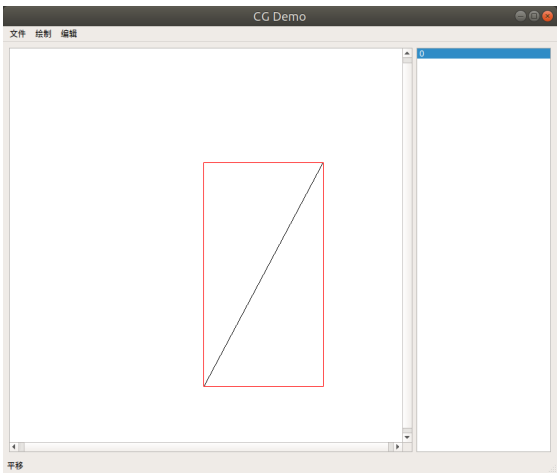
先绘制一个直线



再点选右侧编号选择图元



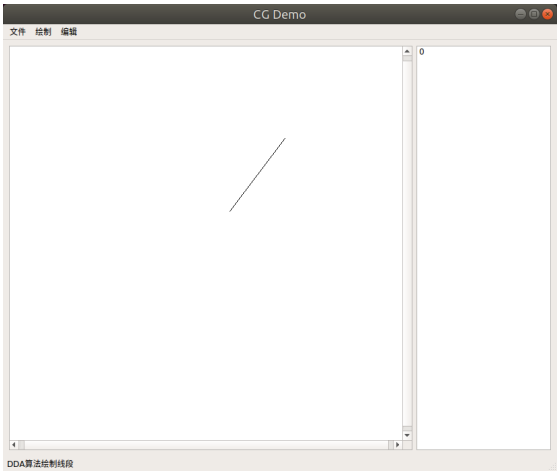
拖拽图元进行平移



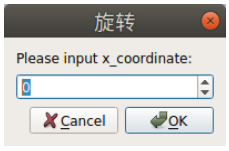
3.8 gui旋转

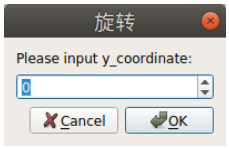
操作方式：先点选窗口右边的编号，选择图形；再通过鼠标点选图形界面工具栏，编辑– >旋转，出现输入框输入旋转中心，旋转角度，填入数据即可。如下图所示：

先绘制一个直线

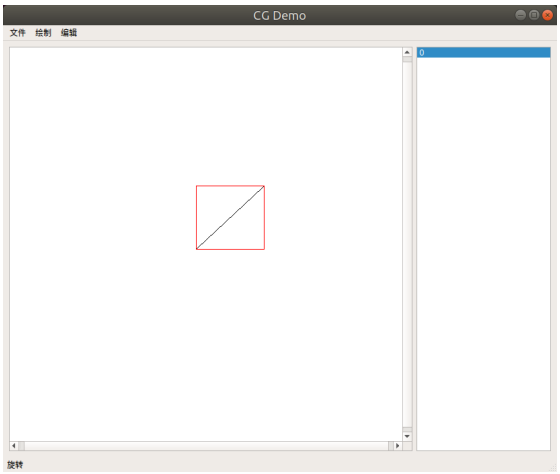


再点选右侧编号，工具栏点选旋转，填写数据





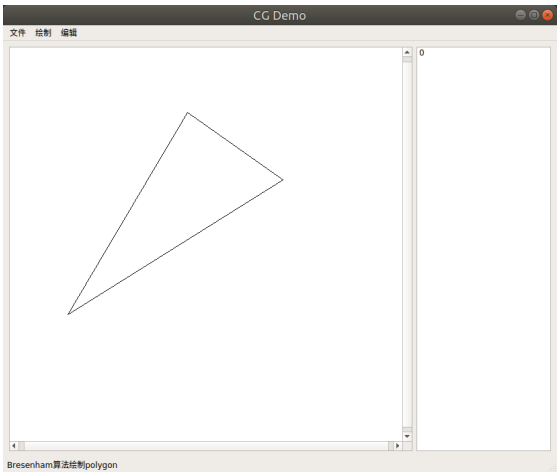
旋转结果如下



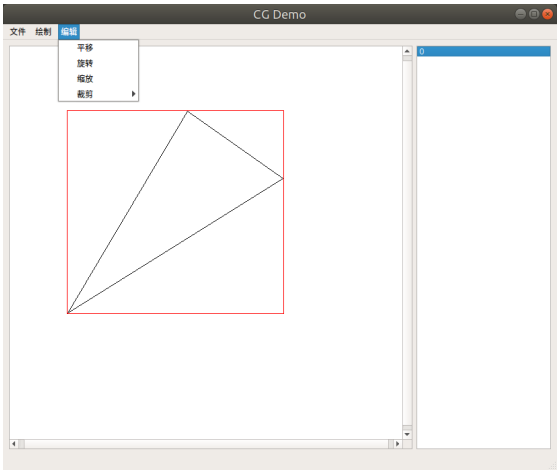
3.9 gui缩放

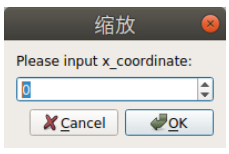
操作方式：先点选窗口右边的编号，选择图形；再通过鼠标点选图形界面工具栏，编辑– >缩放，出现输入框输入缩放中心，缩放倍数，填入数据即可。如下图所示：

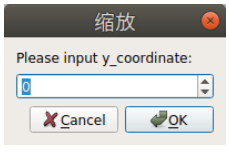
先绘制一个多边形

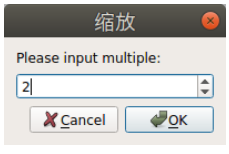


再点选右侧编号，工具栏点选缩放，填写数据

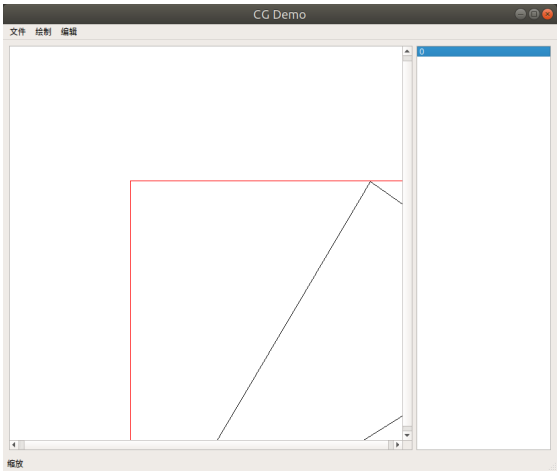








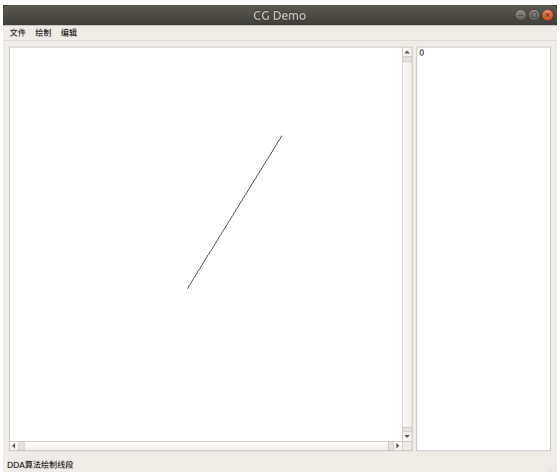
缩放结果如下



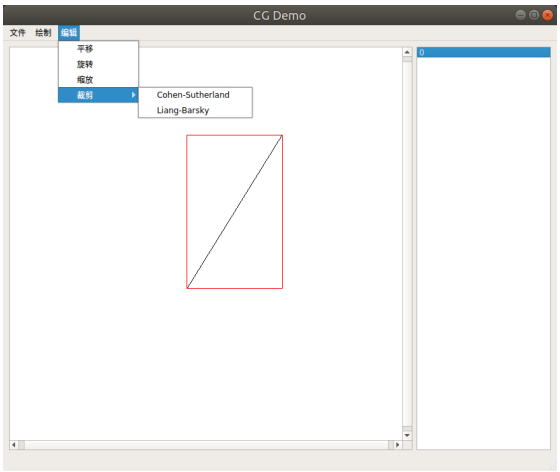
3.10 gui裁剪

操作方式：先点选窗口右边的编号，选择图形；再通过鼠标点选图形界面工具栏，编辑– >裁剪，选择算法。在画布上鼠标点下起始点为裁剪顶点，鼠标松开为另外一个裁剪顶点。如下图所示：

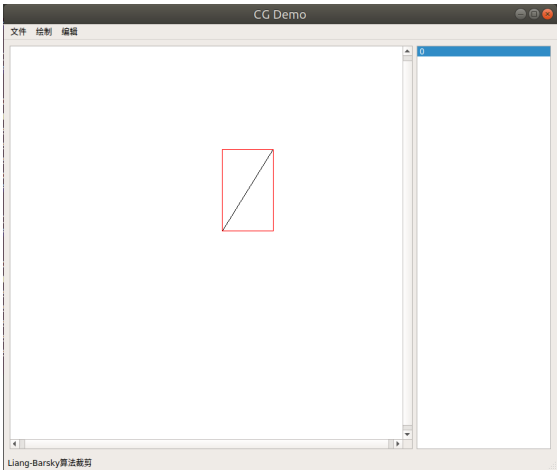
先绘制一个直线



再点选右侧编号，工具栏点选裁剪，选择算法，这里选择Liang-Barsky算法


CG Demo
文件
绘制
编辑
平移
旋转
缩放
裁剪
Cohen-Sutherland
Liang-Barsky

裁剪结果如下


CG Demo
文件
绘制
编辑
Liang-Barsky算法裁剪