

CG 说明书

胡俊豪 181240020@smail.nju.edu.cn

1 测试环境

Ubuntu 18.04, 64 位系统, Anaconda3, python 3.7.4, numpy 1.18.1, pillow 7.0.0, pyqt 5.9.2。

2 CLI 使用说明

在命令行界面启动程序: `python cg_cli.py input_path output_dir`。在 `input` 文件中输入一系列的指令, 来作图, 最后将做好的图片保存在 `output` 所指定的文件中。

在 `input` 文件中, 每行一条指令, 包括:

- 重置画布

```
resetCanvas width height
```

清空当前画布, 并重新设置宽高

width, height: int

$100 \leq \text{width}, \text{height} \leq 1000$

- 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

$0 \leq R, G, B \leq 255$

- 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”

我们在此提供一个示例 input 文件，和对应的示例 output 文件。其中 input 文件中的指令如下：

```

resetCanvas 600 600
setColor 0 255 0
drawLine line1 0 0 500 250 DDA
clip line1 50 50 400 200 Cohen-Sutherland
setColor 255 0 0
drawLine line2 500 250 250 500 Bresenham
translate line2 -50 -50
drawLine line3 100 100 100 500 Bresenham
clip line3 0 0 200 200 Liang-Barsky
drawLine line4 200 500 400 300 Bresenham
clip line4 250 200 350 400 Liang-Barsky
saveCanvas 1
resetCanvas 600 600
setColor 0 0 255
drawPolygon polygon1 100 100 500 500 100 500 DDA
saveCanvas 2
resetCanvas 600 600
drawPolygon polygon2 200 100 300 100 350 200 300 300 200 300 150 200 Bresenham
drawPolygon polygon3 200 100 300 100 350 200 300 300 200 300 150 200 Bresenham
scale polygon3 200 200 1.5
setColor 255 0 0
drawPolygon polygon4 200 100 300 100 350 200 300 300 200 300 150 200 Bresenham
rotate polygon4 300 200 30
saveCanvas 3
resetCanvas 600 600
setColor 0 255 0
drawEllipse ellipse1 100 100 500 400
drawEllipse ellipse2 100 100 500 400
scale ellipse2 50 50 0.7
setColor 255 0 0
drawEllipse ellipse3 130 181 339 100
drawEllipse ellipse4 204 332 377 403
saveCanvas 4
resetCanvas 600 600
drawCurve curve1 50 200 100 100 150 200 Bezier
drawCurve curve2 50 400 100 300 150 400 200 300 Bezier
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 5

```

output 文件如图 1。根据上面这个 input 文件的指令，我们可以得知，最后会得到五张保存的图片，分别是 1-5.bmp。

3 GUI 使用说明

在命令行界面启动程序：python cg_gui.py。

3.1 文件操作

3.1.1 选择画笔

在点击菜单中的“设置画笔”的时候，在窗口左下方显示出“设置画笔”的状态提示，并弹出调色盘供用户选择颜色。选定颜色后，之后所有的图元都是以这种颜色呈现，除非重新选择颜色。在默认情况下，所有的图元画出来都是黑色。

操作步骤及效果见图 2。

3.1.2 重置画布

在点击菜单中的“重置画布”的时候，在窗口左下方显示出“重置画布”的状态提示，画布回到最初的空白状态。

操作步骤及效果见图 3。

3.1.3 保存画布

在点击菜单中的“保存画布”的时候，在窗口左下方显示出“保存画布”的状态提示，然后弹出一个文件浏览的窗口，选择要保存的文件路径并输入文件名。需要注意的是，文件名必须以“.jpg”“.png”或者“.bmp”结尾，否则保存失败。

操作步骤及效果见图 4。

3.1.4 退出

在点击菜单中的“退出”的时候，在窗口左下方显示出“退出”的状态提示，然后会弹出一个询问窗口，询问是否要保存当前画布。如果选择“否”，那么直接退出。如果选择“是”，则弹出一个文件浏览的窗口，选择要保存的文件路径并输入文件名。需要注意的是，文件名必须以“.jpg”“.png”或者“.bmp”结尾，否则保存失败。保存后，系统退出。

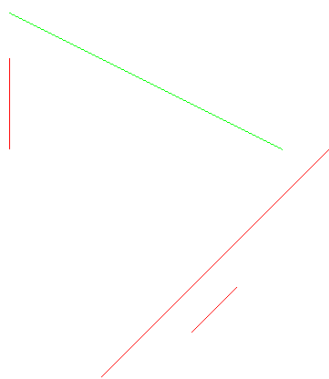
操作步骤及效果见图 5。

3.2 绘制

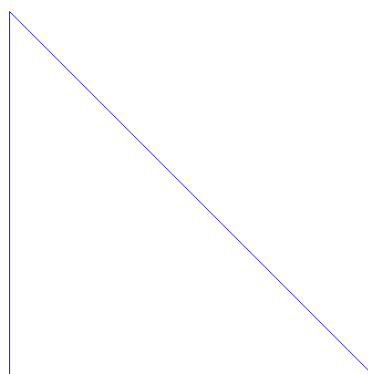
3.2.1 直线

在点击菜单中的“绘制-> 直线”的时候，我们可以任意选择一种算法（Naive, DDA, Bresenham），此时在窗口左下方显示出“xx 算法绘制直线”的状态提示。此时只需要拖动鼠标左键就可以完成直线的绘制。鼠标左键最初点击位置，以及鼠标左键最后松开位置决定了直线的方向与长短。

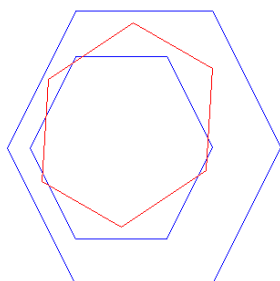
操作步骤及效果见图 6。



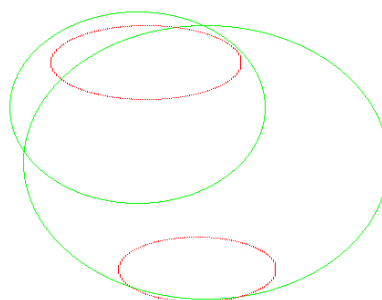
(a) 1.bmp



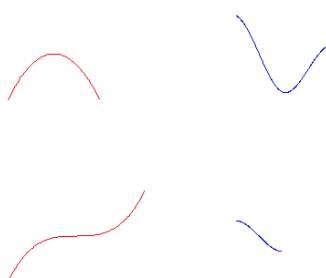
(b) 2.bmp



(c) 3.bmp

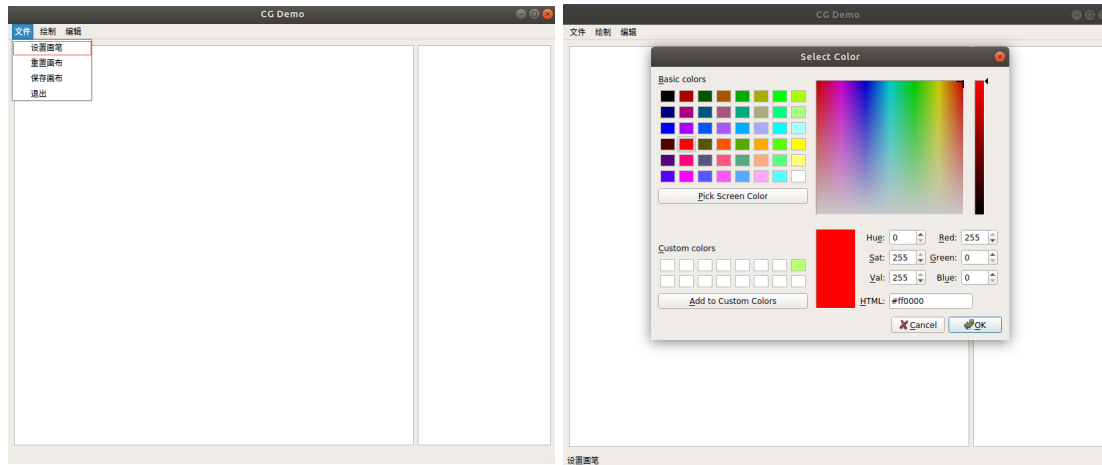


(d) 4.bmp



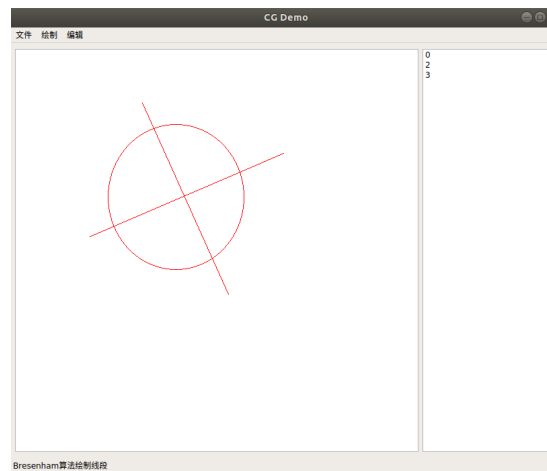
(e) 5.bmp

图 1: CLI 的 output 文件示例



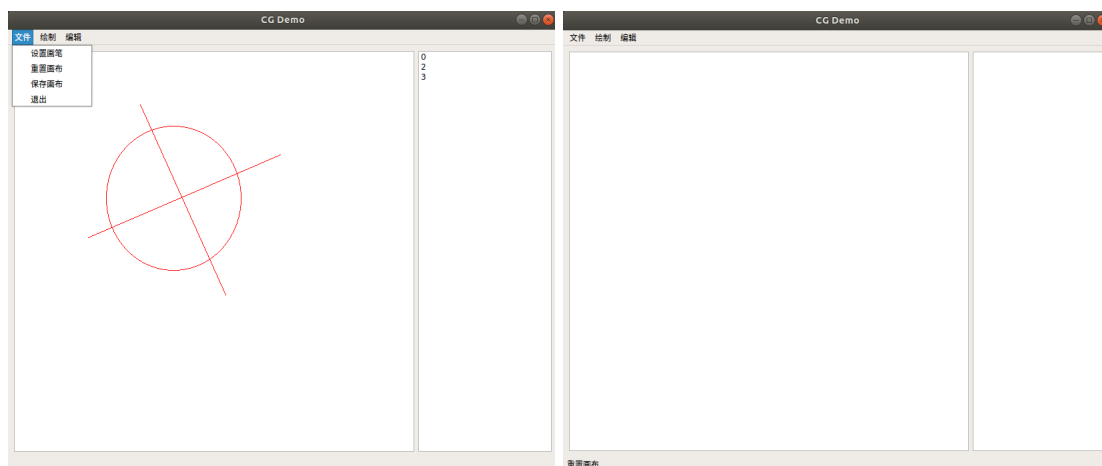
(a) 选择“设置画笔”

(b) 选择颜色



(c) 画图

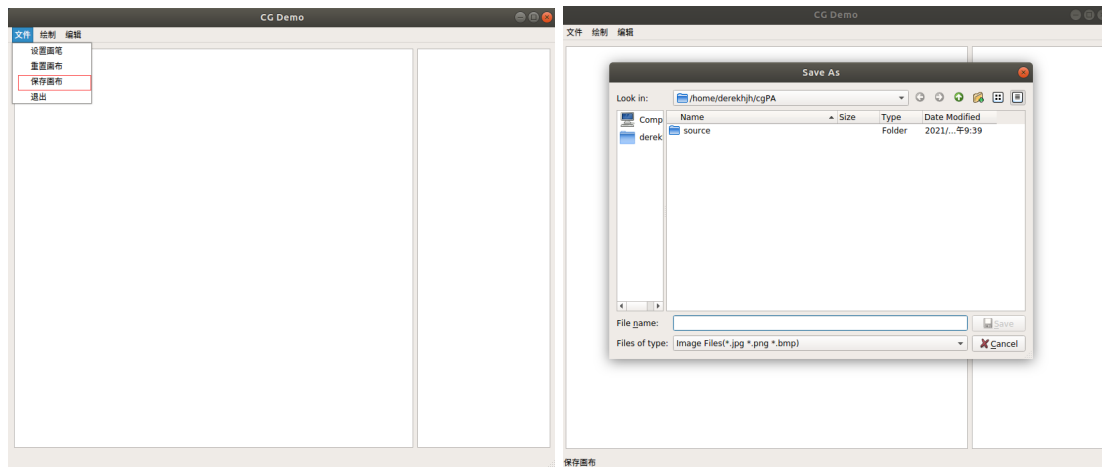
图 2: “设置画笔” 操作步骤



(a) 选择“重置画布”

(b) 画布空白

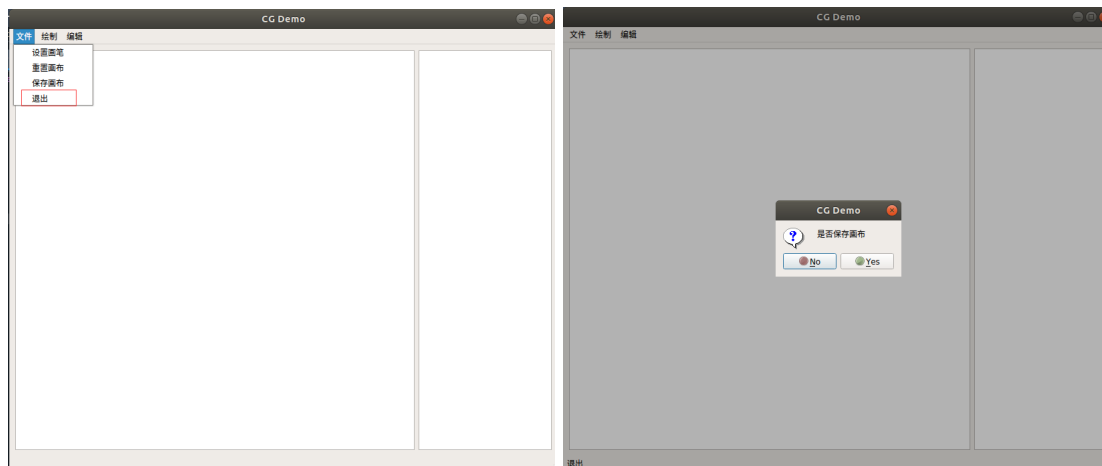
图 3: “重置画笔” 操作步骤



(a) 选择“保存画布”

(b) 画布保存界面

图 4: “画布保存” 操作步骤



(a) 选择“退出”

(b) 询问是否保存

图 5: “退出” 操作步骤

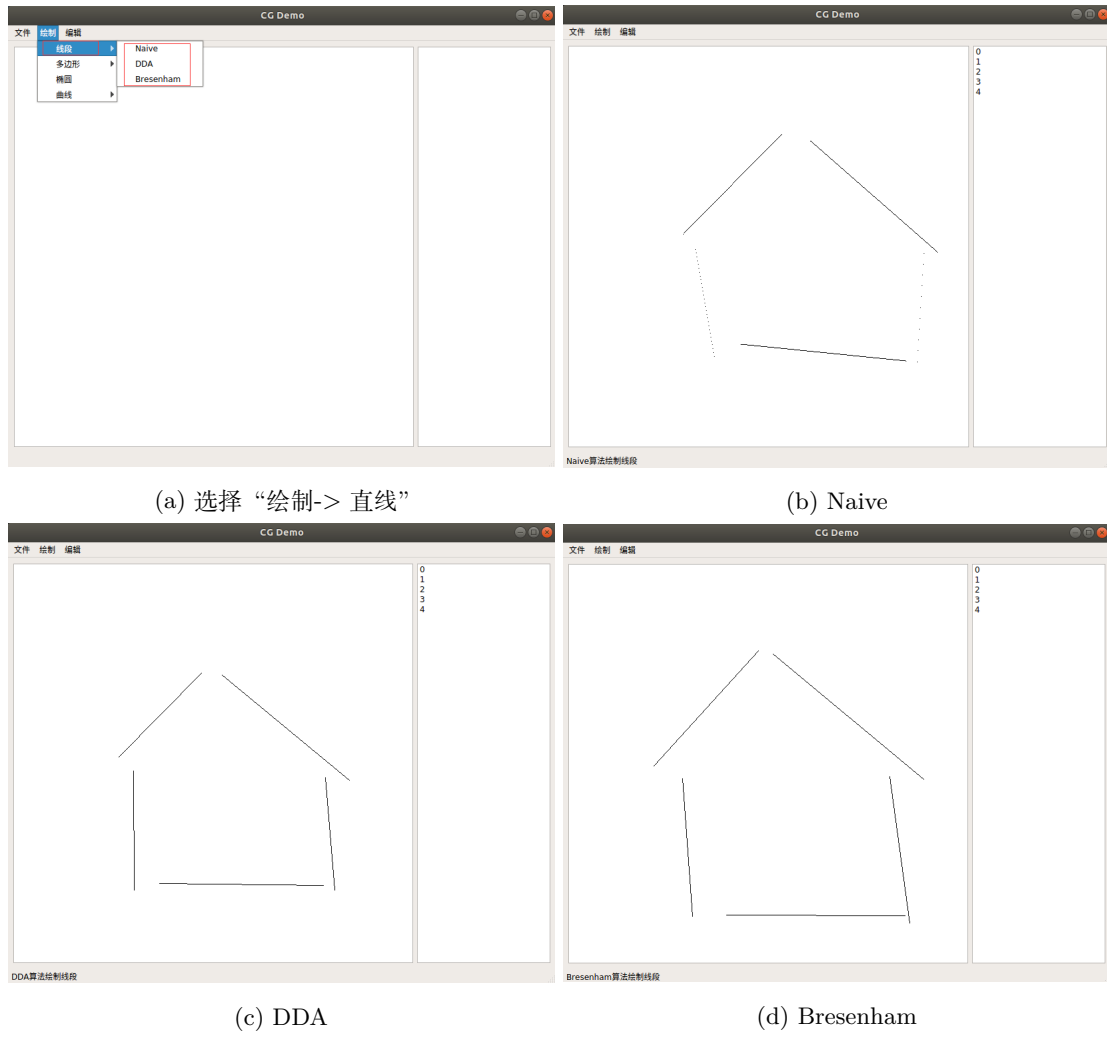
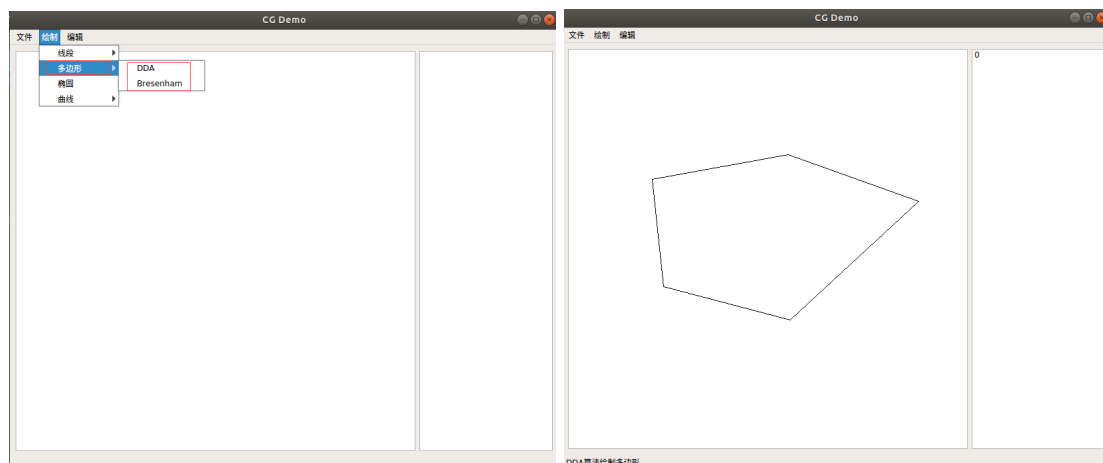
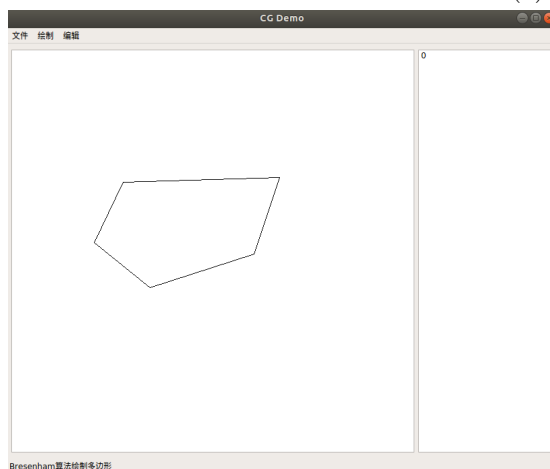


图 6: “绘制-> 直线” 操作步骤



(a) 选择“绘制->多边形”

(b) DDA



(c) Bresenham

图 7: “绘制->多边形” 操作步骤

3.2.2 多边形

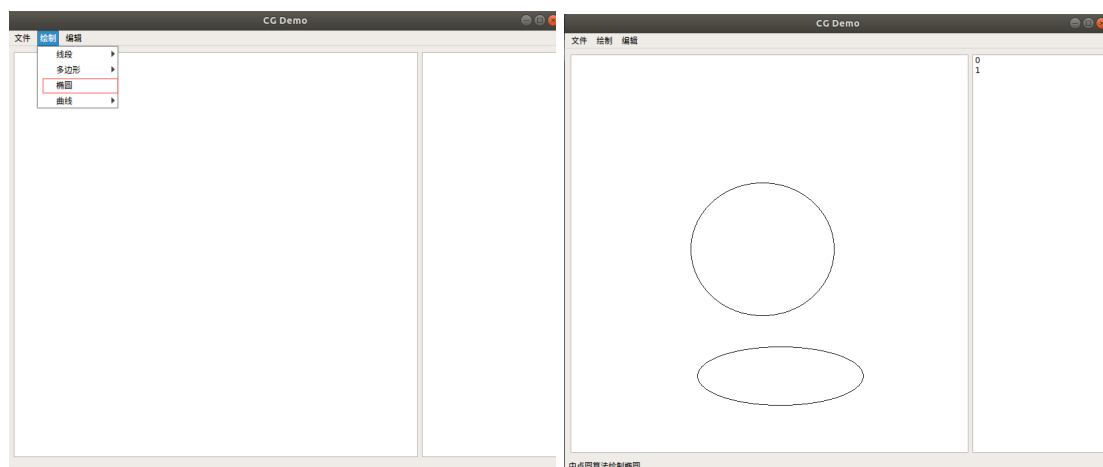
在点击菜单中的“绘制->多边形”的时候，我们可以任意选择一种算法（DDA，Bresenham），此时在窗口左下方显示出“xx 算法绘制多边形”的状态提示。此时只需要多次点击鼠标左键就能够选择要绘制的多边形的顶点，选择顶点的过程中，系统会动态绘制根据已有定点确定的多边形，直到点击鼠标右键，停止该多边形的顶点选择，结束此多边形的绘制。

操作步骤及效果见图 7。

3.2.3 椭圆

在点击菜单中的“绘制->椭圆”的时候，在窗口左下方显示出“中点圆算法绘制椭圆”的状态提示。此时只需要拖动鼠标左键就能够完成椭圆的绘制，鼠标左键最初点击位置，以及鼠标左键最后松开位置决定了椭圆的大小与形状。

操作步骤及效果见图 8。



(a) 选择“绘制->椭圆”

(b) 画出椭圆

图 8: “绘制->椭圆”操作步骤

3.2.4 曲线

在点击菜单中的“绘制->曲线”的时候，我们可以任意选择一种算法（Bezier，B-spline），此时在窗口左下方显示出“xx 算法绘制曲线”的状态提示。此时只需要多次点击鼠标左键就能够选择要绘制的曲线的顶点，选择顶点的过程中，系统会动态绘制根据已有顶点确定的曲线，直到点击鼠标右键，停止该曲线的顶点选择，结束此曲线的绘制。

操作步骤及效果见图 9。

3.3 编辑

在进行编辑操作的时候，我们要做的第一件事情是先选中某一个图元，如果没有选中某一个图元，直接点击编辑的相关选项，则会弹出提示框，提示用户先选择一个图元作为编辑的对象。如图 10

选中图元后，图元会被一个红色的矩形线框包围，以提示用户。需要注意的是，无论是平移还是旋转还是缩放操作，鼠标都不是必须要点击在红色线框内部才能操纵相应的图元。只要在编辑状态，点击任何地方（包括不在红色线框内部的地方）都会对图元产生影响，具体可一件下面几个小节更详细的说明。

3.3.1 平移

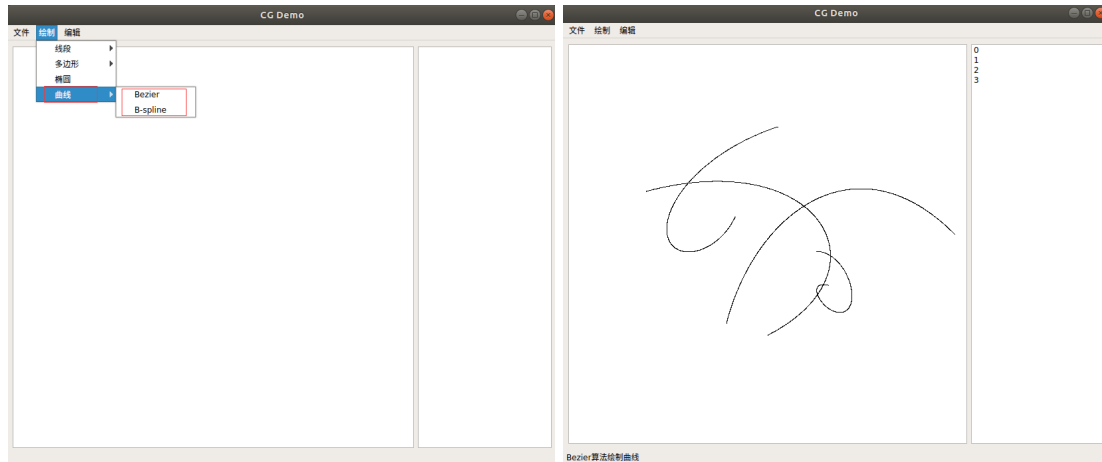
选中图元后，点击“编辑->平移”，左下角的状态栏就会显示现在处于“平移”状态。点击鼠标左键并按住不动，向各个方向平移，会导致选中的图元，按照鼠标移动方向相同的方向，平移相同的距离。

操作步骤及效果见图 11。

3.3.2 旋转

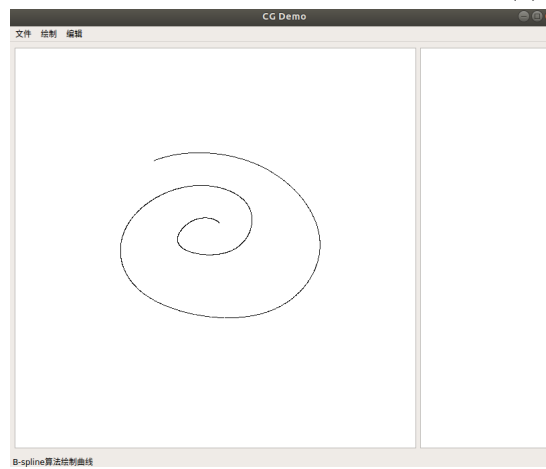
选中图元后，点击“编辑->旋转”，左下角的状态栏就会显示现在处于“旋转”状态。点击鼠标左键确定旋转中心，并按住不动，向某一个方向拖动一下确定旋转起始向量，按住鼠标左键，相对于起始向量向哪个地方转动，图元就以刚才确定的旋转中心向哪边转动。

操作步骤及效果见图 12。



(a) 选择“绘制->曲线”

(b) Bezier



(c) B-spline

图 9: “绘制->曲线” 操作步骤

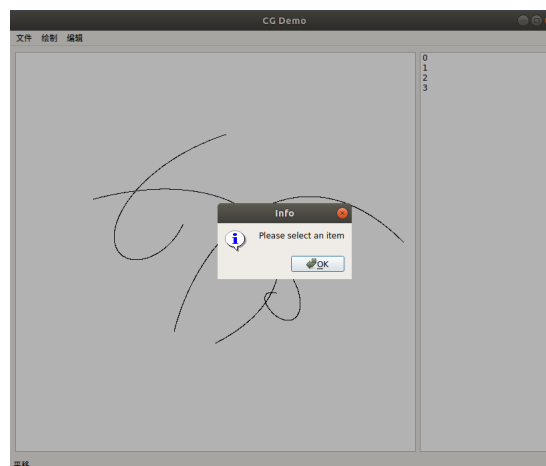
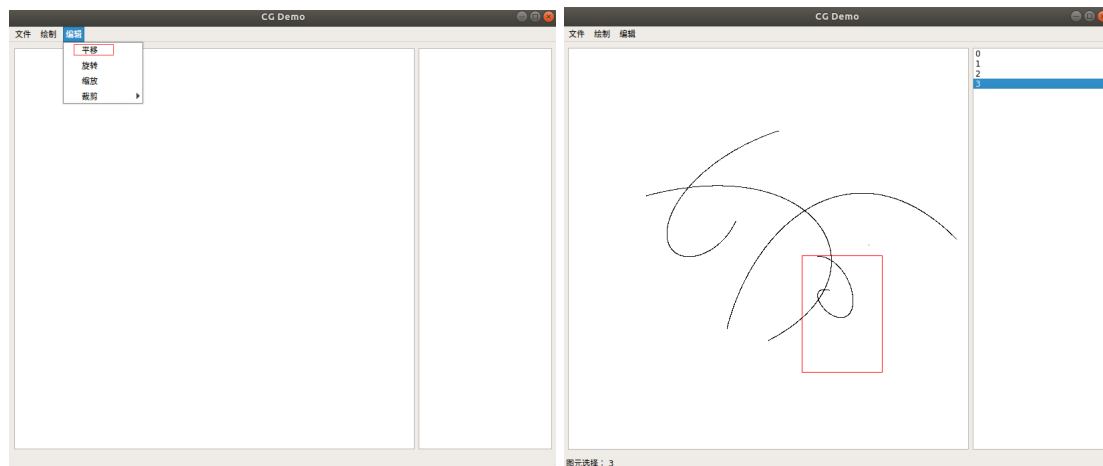
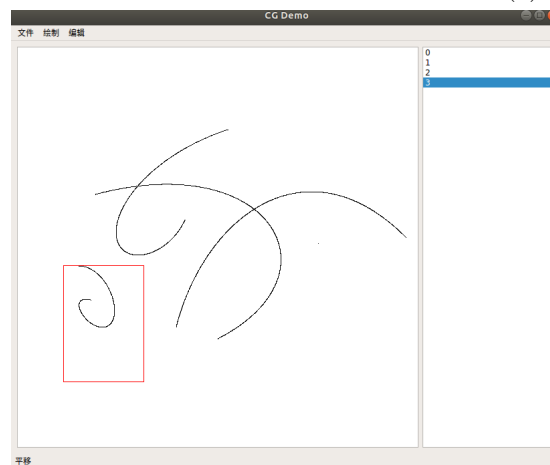


图 10: 请选择一个图元进行编辑



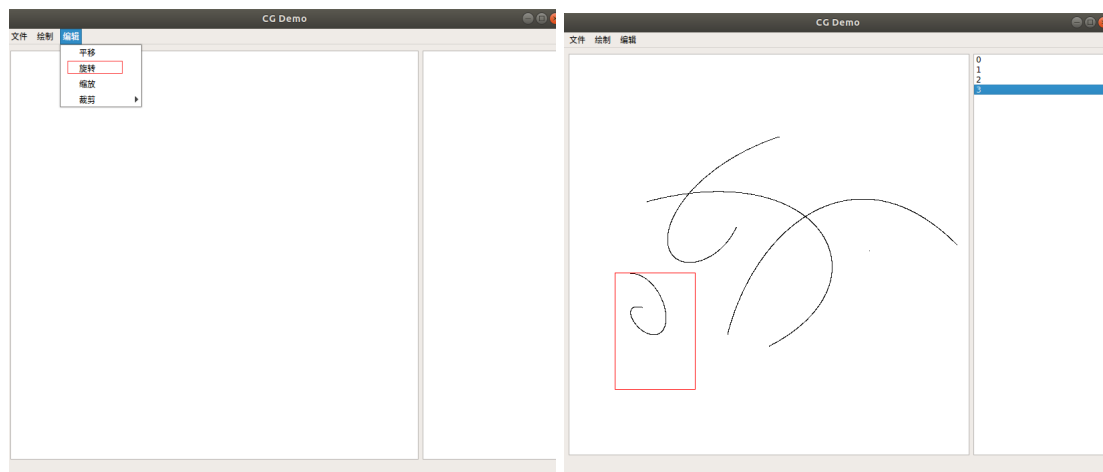
(a) 选择“编辑-> 平移”

(b) 平移前



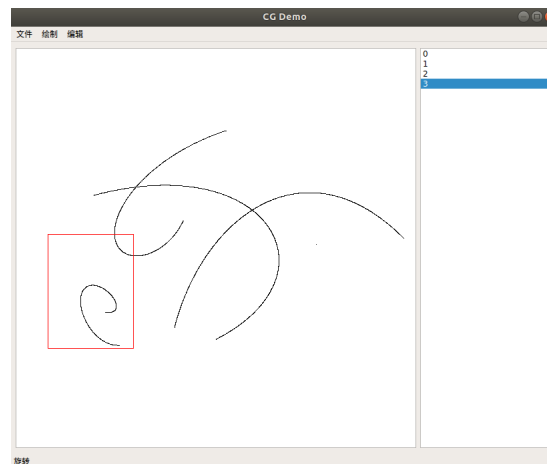
(c) 平移后

图 11: “编辑-> 平移”操作步骤



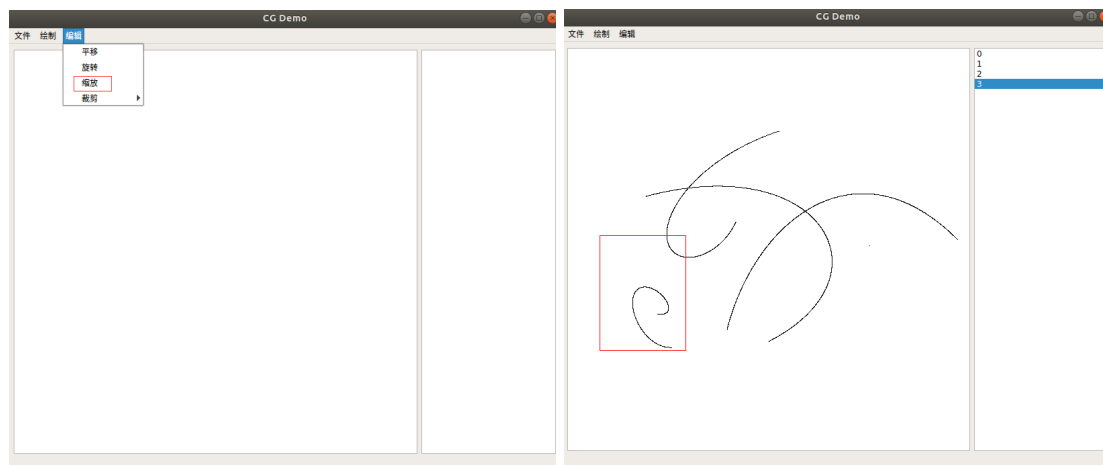
(a) 选择“编辑-> 旋转”

(b) 旋转前



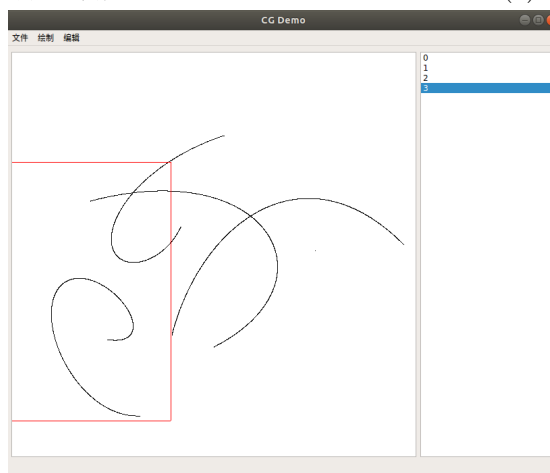
(c) 旋转后

图 12: “编辑-> 旋转”操作步骤



(a) 选择“编辑-> 缩放”

(b) 缩放前



(c) 缩放后

图 13: “编辑-> 缩放”操作步骤

3.3.3 缩放

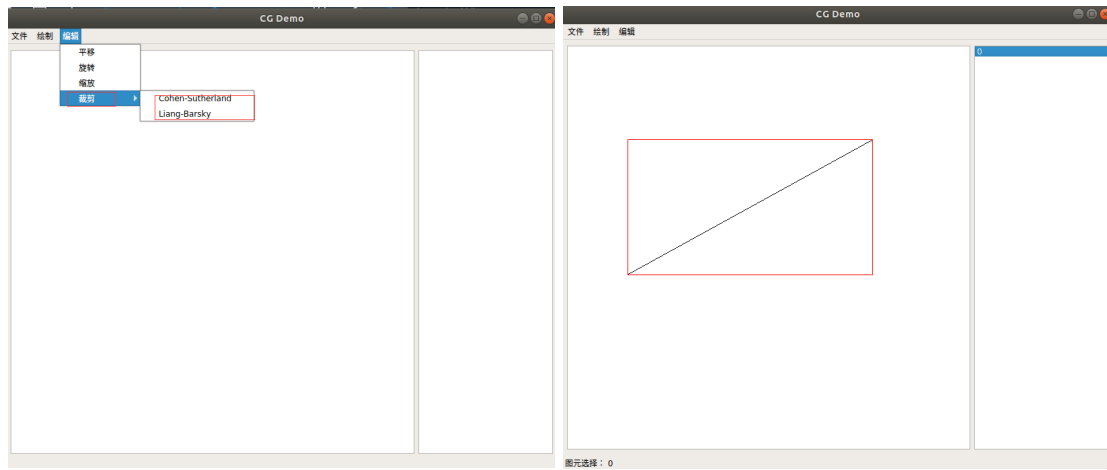
选中图元后，点击“编辑-> 缩放”，左下角的状态栏就会显示现在处于“缩放”状态。点击鼠标左键并按住不动，向右平移，会导致选中的图元，以鼠标点击位置为缩放中心，进行放大操作；点击鼠标左键并按住不动，向左平移，会导致选中的图元，以鼠标点击位置为缩放中心，进行缩小操作。

操作步骤及效果见图 13。

3.3.4 裁剪

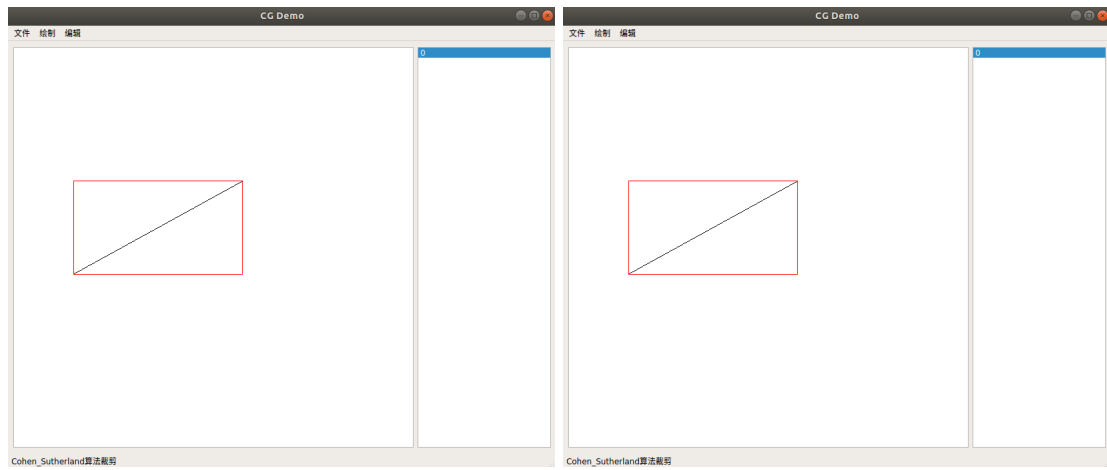
选中图元后，点击“编辑-> 裁剪”，左下角的状态栏就会显示现在处于“用 xx 算法裁剪”状态。点击鼠标左键并按住不动，向右下平移，会画出一个矩形线框。松开鼠标之后，矩形框框住的部分为裁剪后留下的部分。裁剪可以采用不同的算法（Cohen-Sutherland, Liang-Barsky）。

操作步骤及效果见图 14。



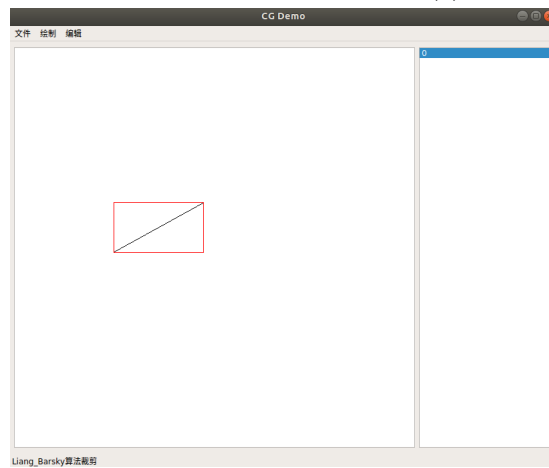
(a) 选择“编辑-> 裁剪”

(b) Cohen-Sutherland 裁剪前



(c) Cohen-Sutherland 裁剪后

(d) Liang-Barsky 裁剪前



(e) Liang-Barsky 裁剪后

图 14: “编辑-> 裁剪”操作步骤