

系统使用说明书

目录

系统使用说明书

目录

- 开发环境
- 窗口布局
- 文件菜单
 - 保存图片
 - 设置画笔
 - 设置画笔颜色
 - 设置画笔粗细
 - 重置画布
- 退出
- 编辑菜单
 - 选择
 - 复制
 - 粘贴
 - 平移
 - 旋转
 - 缩放
 - 裁剪
- 绘制菜单
 - 线段
 - 多边形
 - 椭圆
 - 曲线
 - Bezier算法绘制
 - B-spline算法绘制
- 视图
 - 显示网格
- 帮助
 - 系统文档

开发环境

(由于虚拟机出了些问题，所以我在Windows 10 + Anaconda下完成开发QAQ)

```
• conda==4.11.0
• python==3.7.4
• numpy==1.18.1
• pillow==7.0.0
• pyqt5==5.15.4
```

- 其他导入的python库：sys, math, os, warnings, webbrowser(基本都是常用的库)
- 还导入了 `MyDialog.py` (与 `cg_gui.py` 同目录下的自定义对话框模块)

```
4 import sys
5 import math
6 import os
7 import numpy as np
8 from PIL import Image
9 import warnings
10 import cg_algorithms as alg
11 from typing import Optional
12 from PyQt5.QtWidgets import *
13 from PyQt5.QtGui import *
14 from PyQt5.QtCore import *
15 from MyDialog import *
16 import webbrowser
17 warnings.filterwarnings('ignore')
18
```

窗口布局

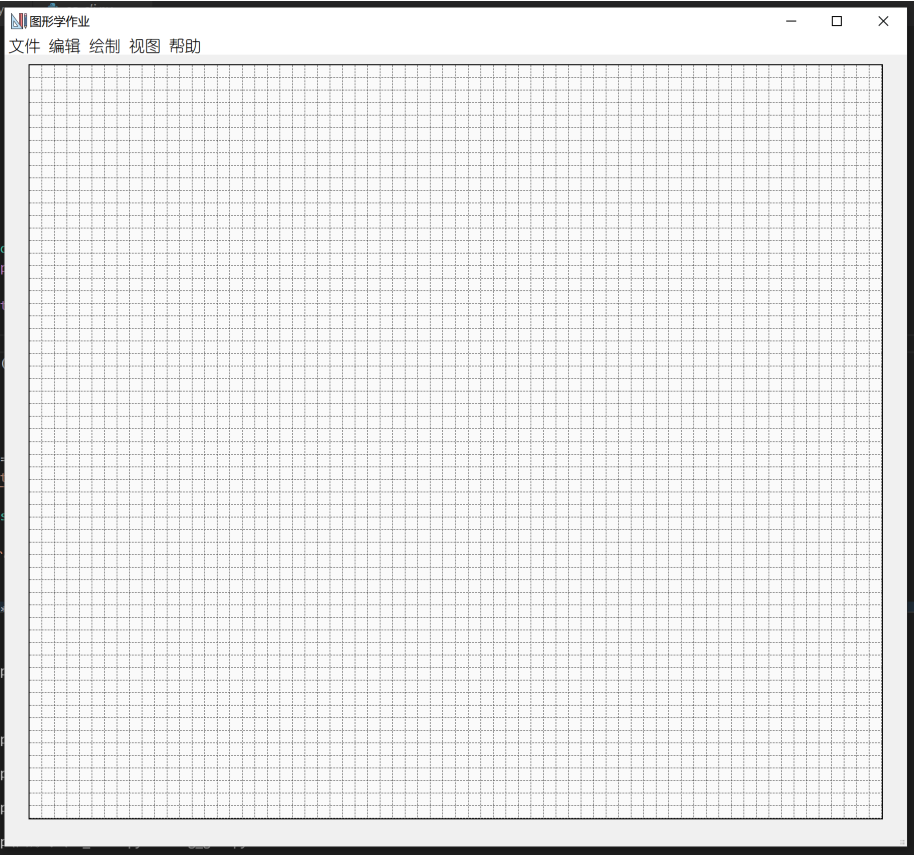
```
1 cd source          # 进入到source目录下
2 python cg_gui.py   # 运行用户程序界面
```

初始主窗体宽/高为1800/1600, 可在 `cg_gui.py` 中修改变量 `MAINWINDOW_W` 和 `MAINWINDOW_H` 来修改;

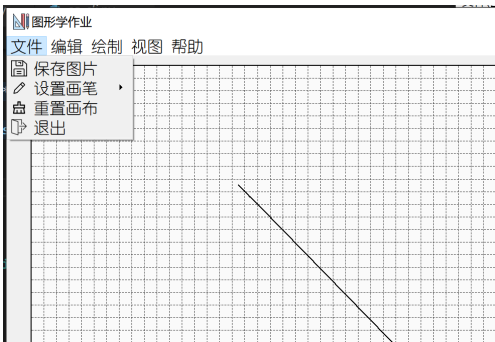
初始画布宽/高为1700/1500, 可在 `cg_gui.py` 的 `MainWindow` 类中修改变量 `DEFAULT_W` 和 `DEFAULT_H` 来修改;

```
# 主窗体的宽与高
MAINWINDOW_W = 1800
MAINWINDOW_H = 1600

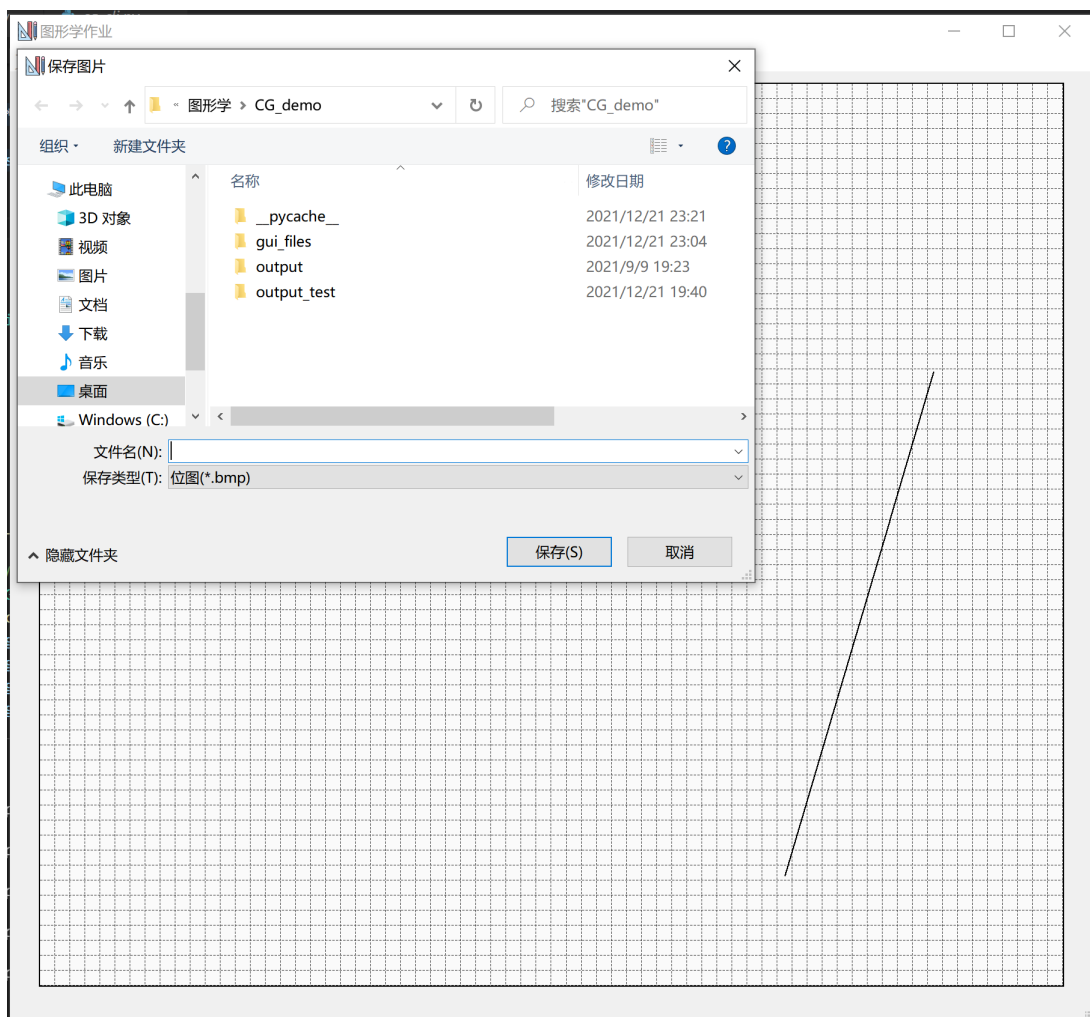
class MainWindow(QMainWindow):
    """
    主窗口类
    """
    # 默认画布的宽与高
    DEFAULT_W = 1700
    DEFAULT_H = 1500
```



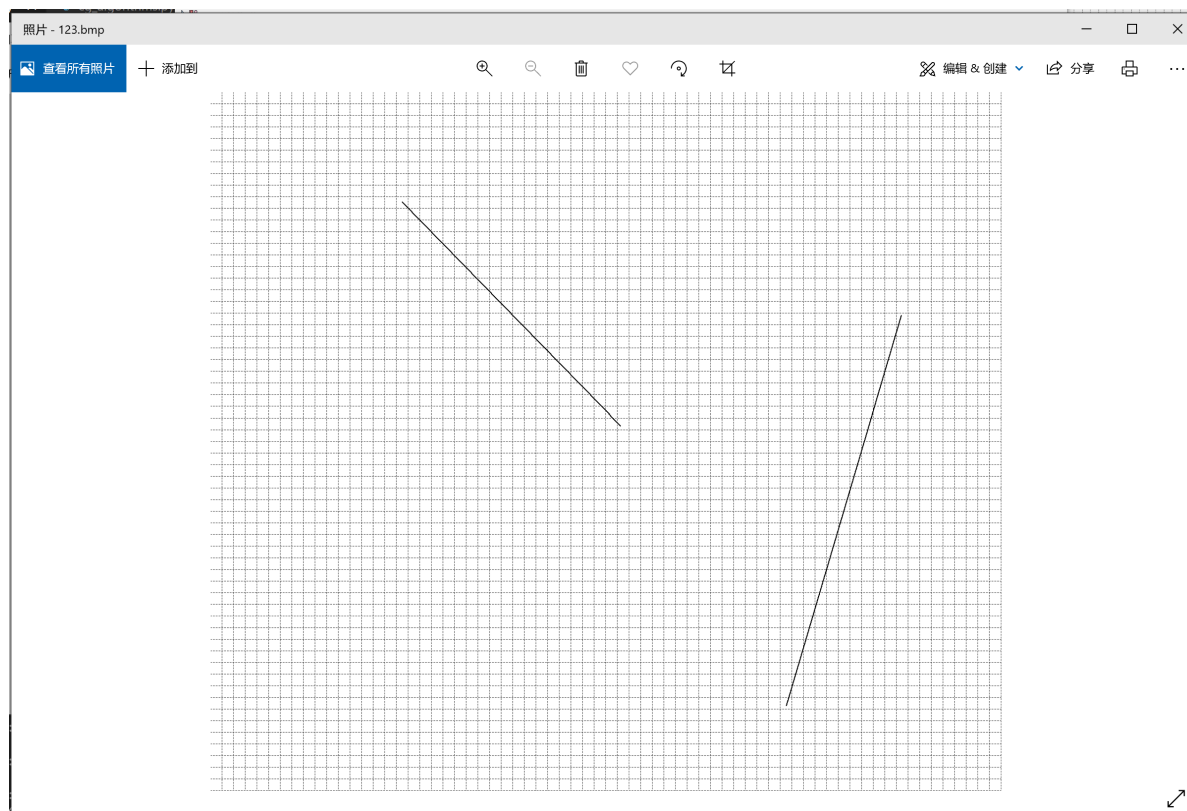
文件菜单



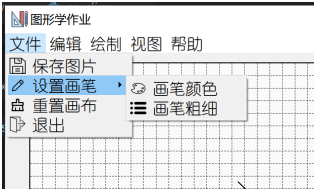
保存图片



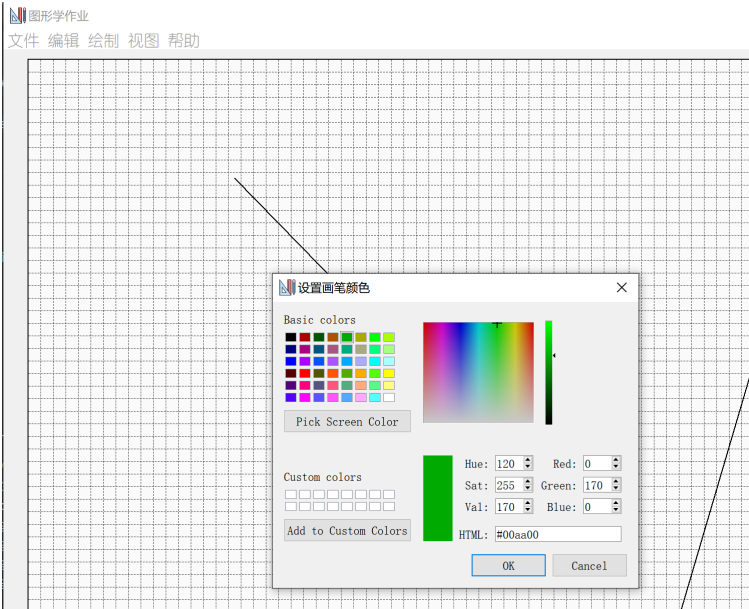
保存效果如下：



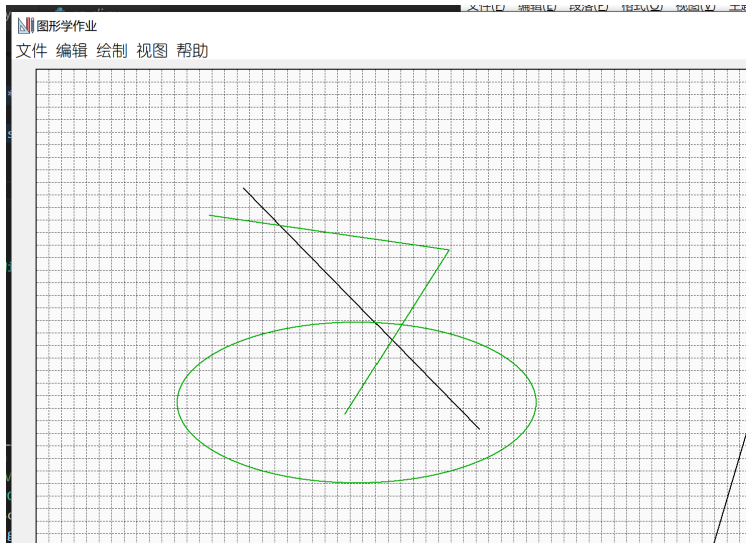
设置画笔



设置画笔颜色

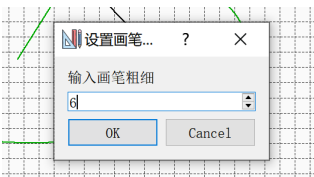


效果如下：

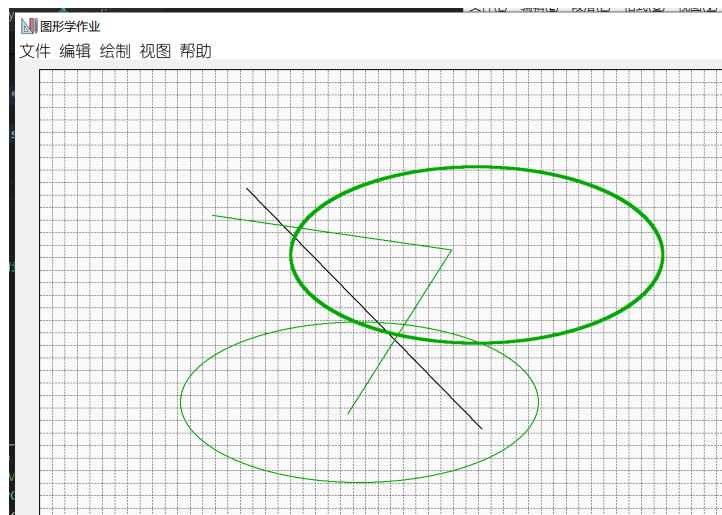


设置画笔粗细

大小范围[1, 10]，默认为2

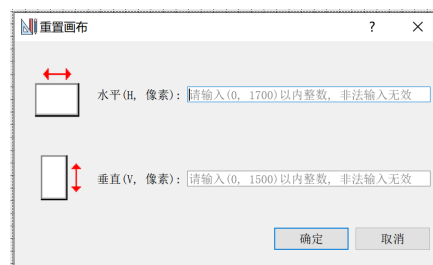


假设设置为6，效果如下：

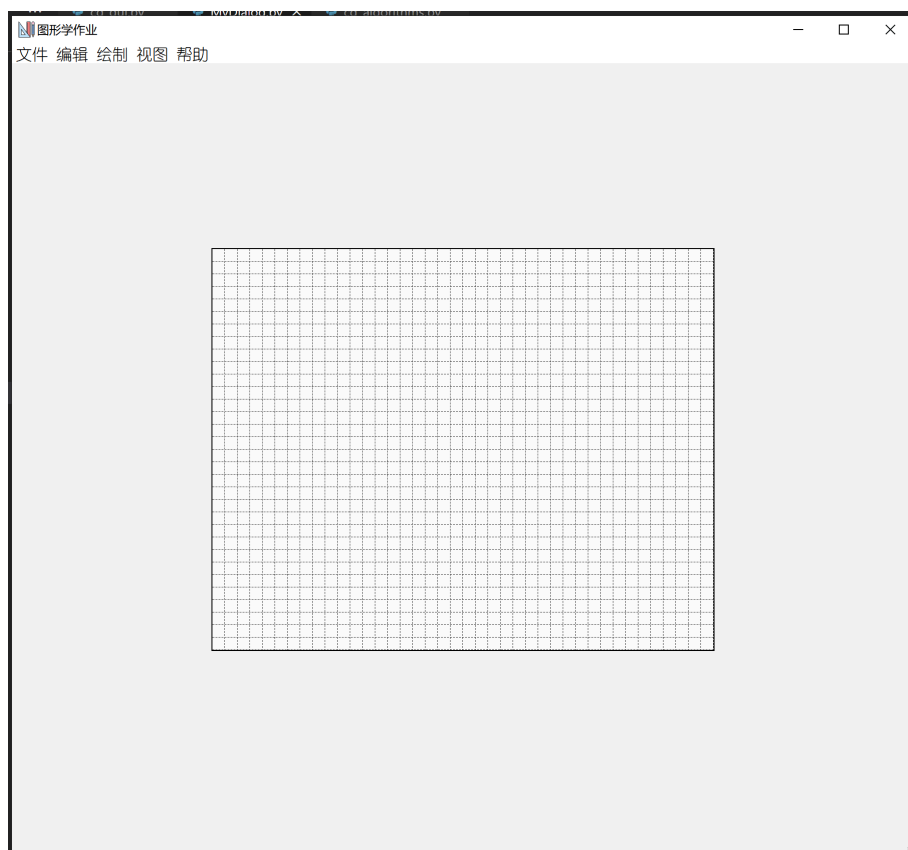


重置画布

按照对话框提示输入，即可



假设水平设为1000，垂直为800，效果如下：



退出

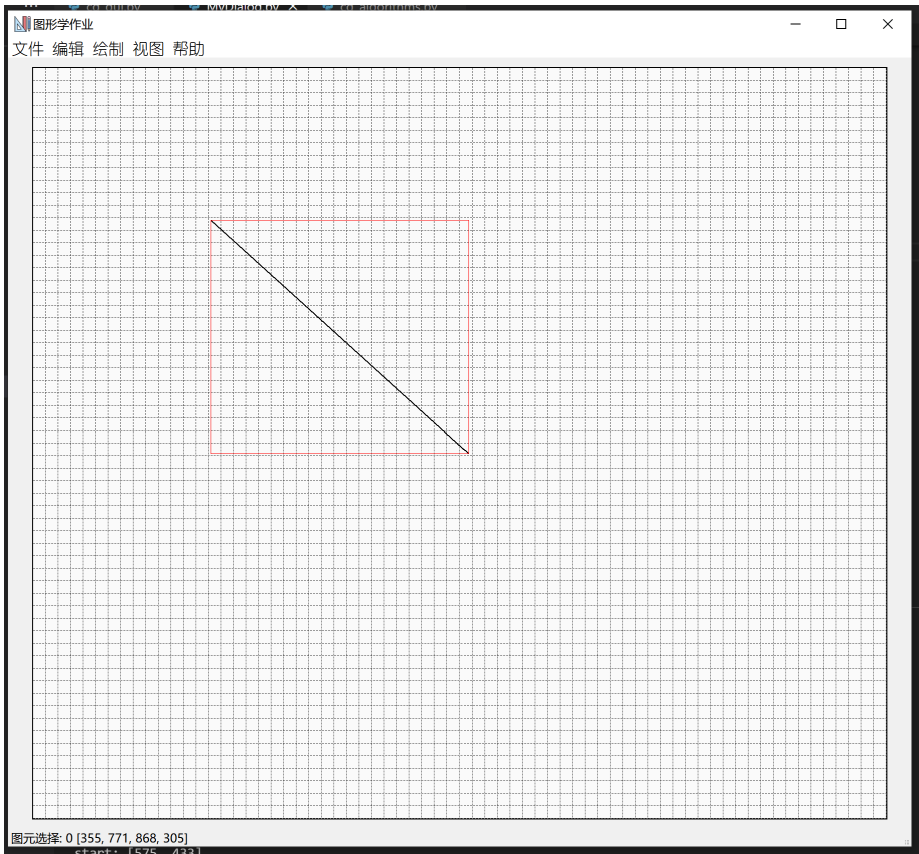
直接退出

编辑菜单



选择

选择图元：鼠标左键点击选择图元，点击空白处取消选择，多个图元叠加，优先选择图元编号小者



复制

如果直接点击复制，则会提示：

复制图元：无选中图元，先选择图元，再点击复制

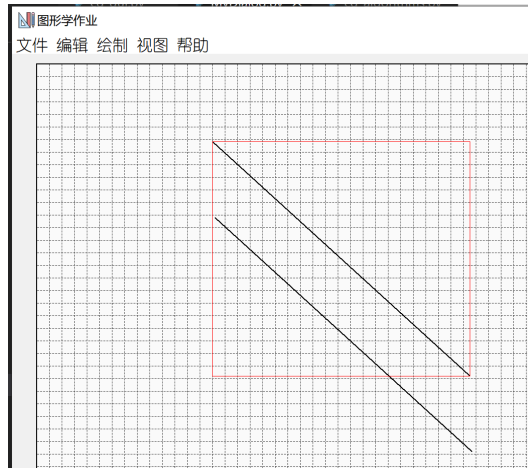
如果点击了图元，则会提示：

复制图元：复制成功!

粘贴

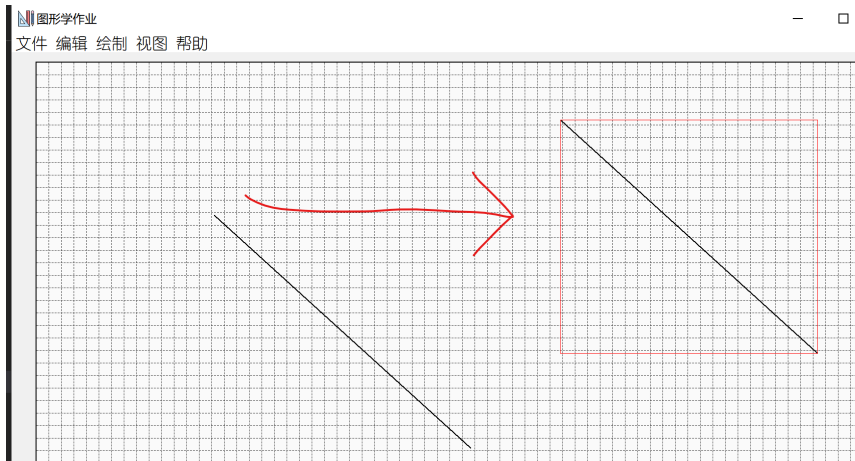
粘贴图元：鼠标左键点击即可粘贴图元，图元尽可粘贴一次

点击后效果如下：



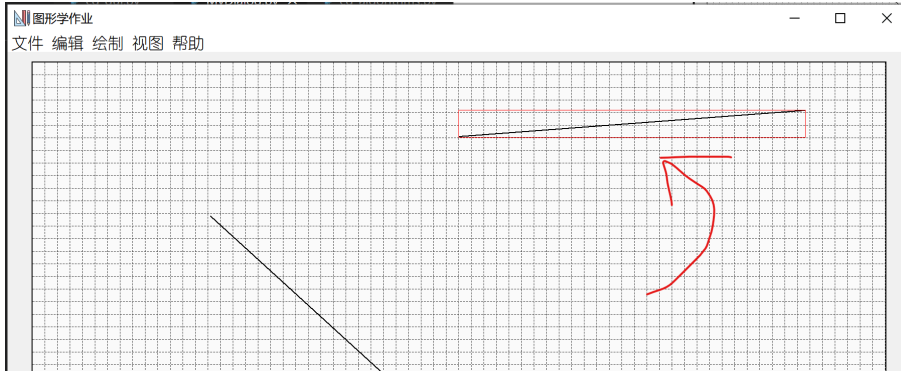
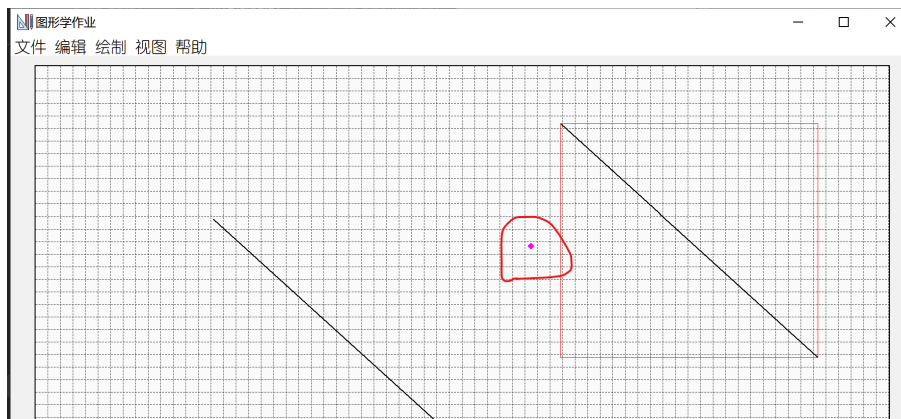
平移

平移图元：按住鼠标左键平移图元，松开结束



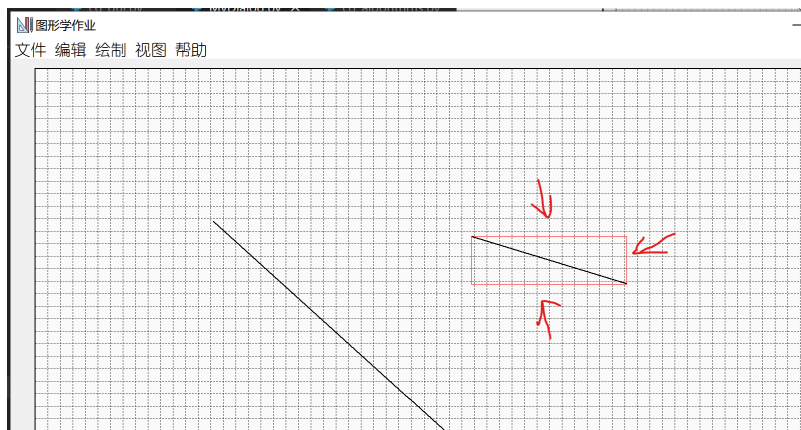
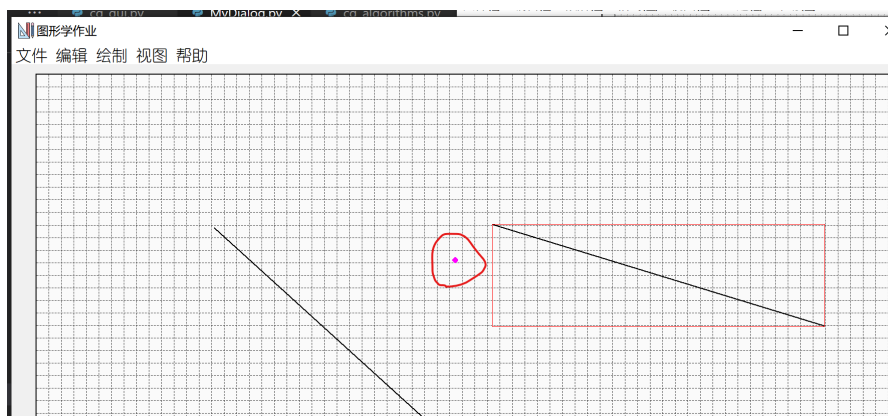
旋转

旋转图元：先在画布上单击一处抛掷锚点，再按住鼠标左键围绕锚点旋转图元，松开结束



缩放

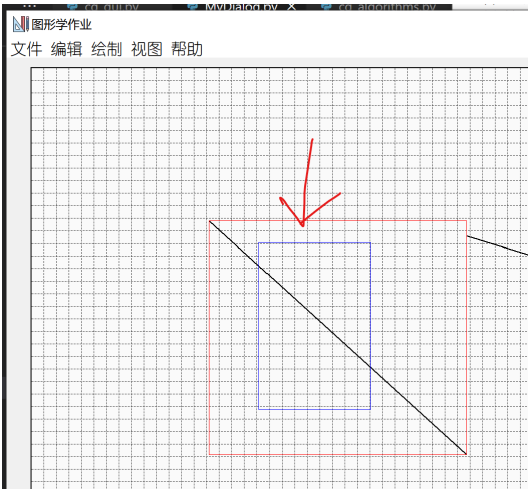
缩放图元：先在画布上单击一处抛掷锚点，再按住鼠标左键围绕锚点缩放图元，松开结束



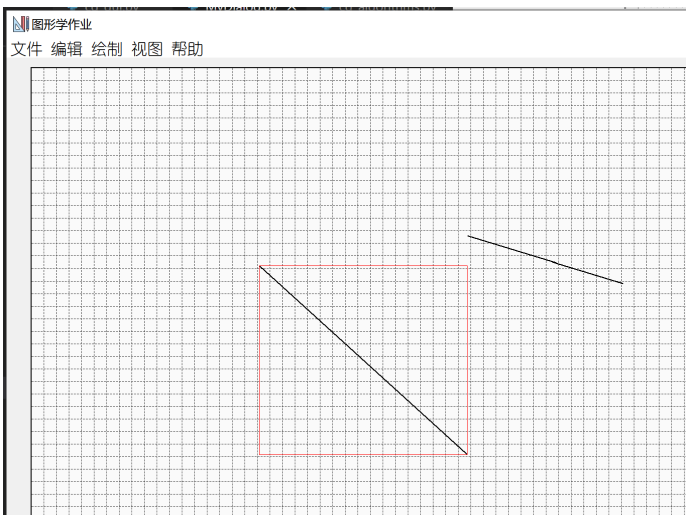
裁剪

Cohen-Sutherland算法裁剪线段：按住鼠标左键并移动，产生裁剪框，松开左键完成裁剪，注意非线段不裁剪

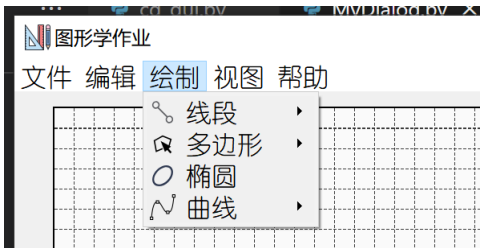
蓝色框为裁剪框，



裁剪结果为，



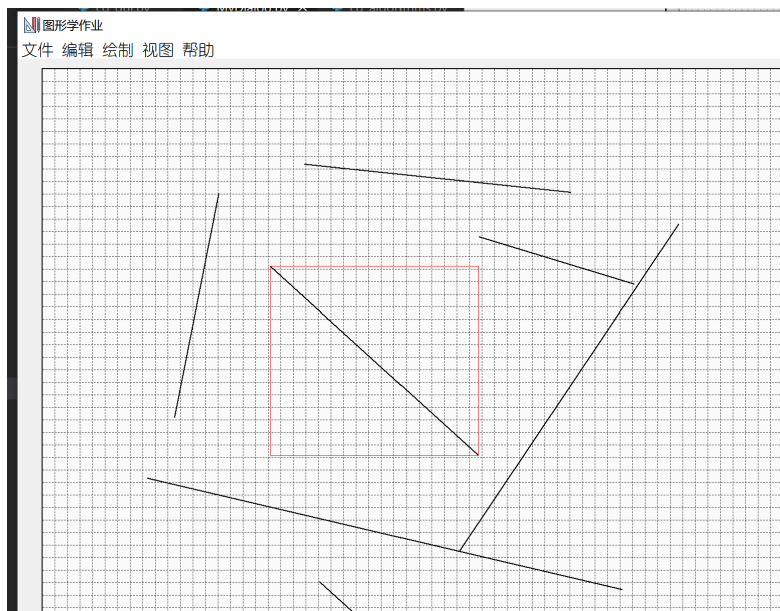
绘制菜单



线段

DDA算法绘制线段：按住鼠标左键并移动，产生线段，松开左键完成绘制
start: [455, 349]

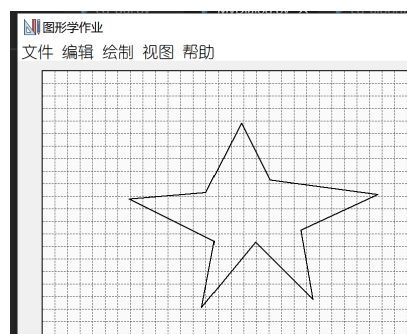
绘制效果：



多边形

Bresenham算法绘制多边形：按住左键移动，然后松开完成多边形一条边的绘制，点击右键完成绘制，多边形首末点会自动连接一条线

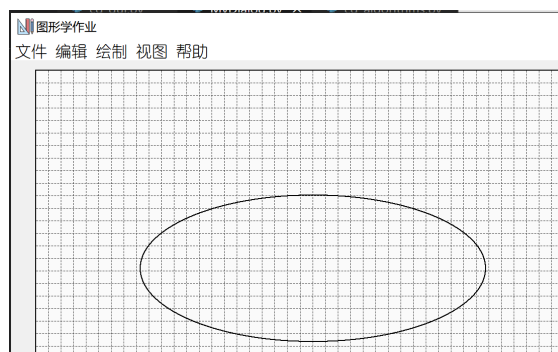
效果如下：



椭圆

绘制椭圆：按住左键移动，然后松开完成椭圆绘制

效果如下：



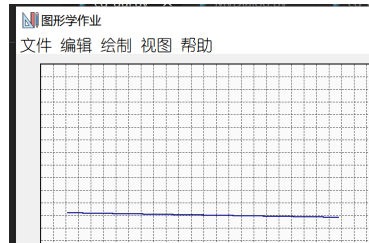
曲线

曲线最多可按6个点或者按右键结束，控制点数可在 `cg_gui.py` 中的 `MAX_NUM_CONTROL_POINTS` 全局变量中修改个数。

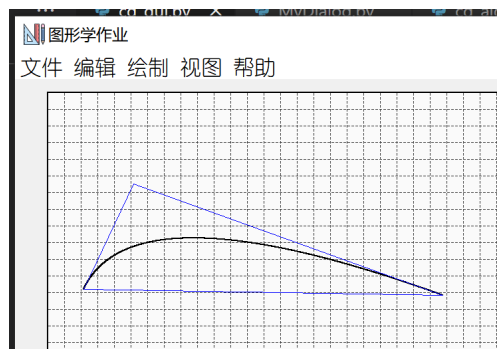
Bezier算法绘制

Bezier算法绘制曲线：首先按住左键绘制一条线段，然后松开，之后按照多边形绘制方法绘制控制多边形，同时生成曲线，最多绘制6个控制点或按右键提前结束绘制

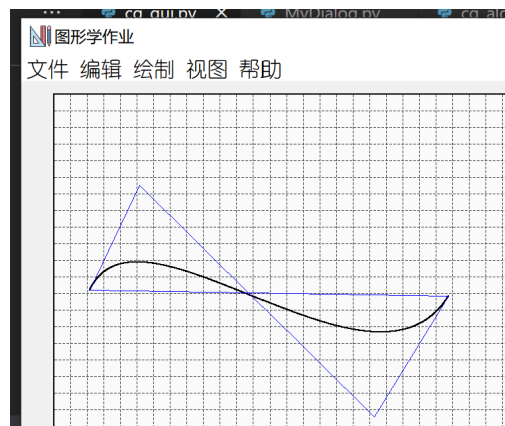
1.先画出一条直线



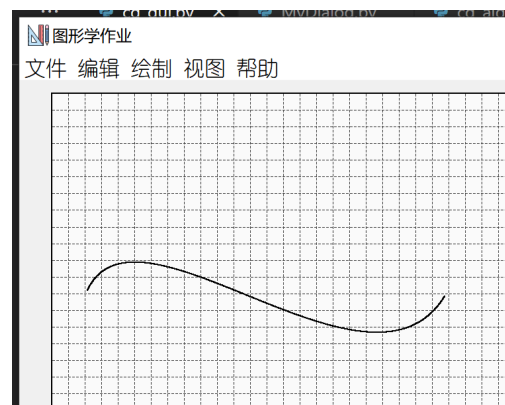
2.再按住鼠标左键移动，绘制出控制点多边形，松开鼠标左键，控制点生成



3.再绘制一个控制点



4.按右键结束绘制，控制多边形也随之消失

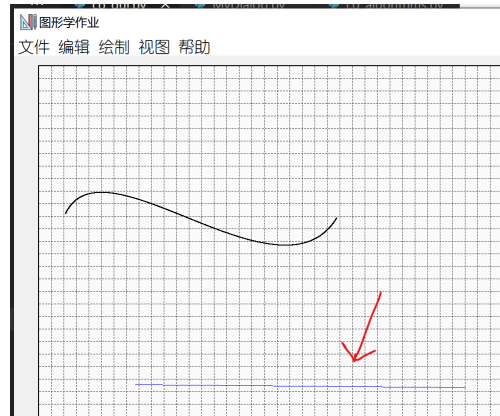


B-spline算法绘制

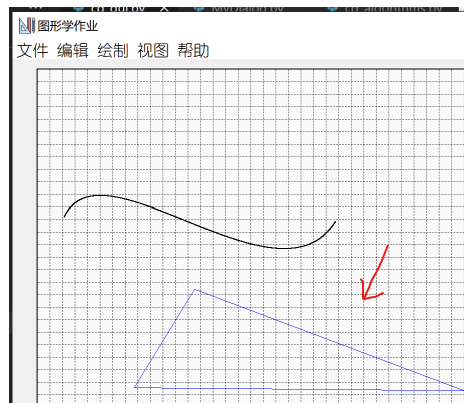
由于是3次B样条，故最少要4个控制点才能绘制出曲线

B-spline算法绘制曲线：首先按住左键绘制一条线段，然后松开，之后按照多边形绘制方法绘制控制多边形，同时生成曲线，最多绘制6个控制点或按右键提前结束绘制
start: [456, 371]

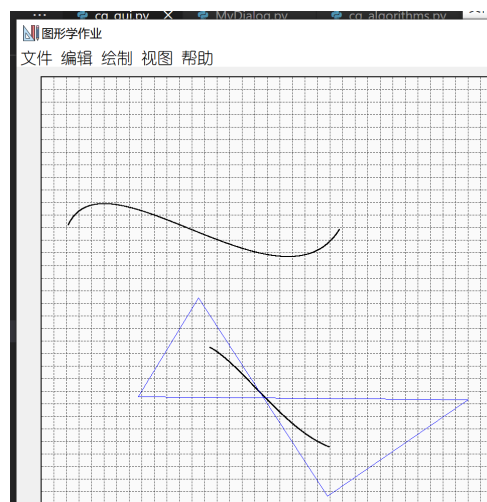
1.先画出一条直线



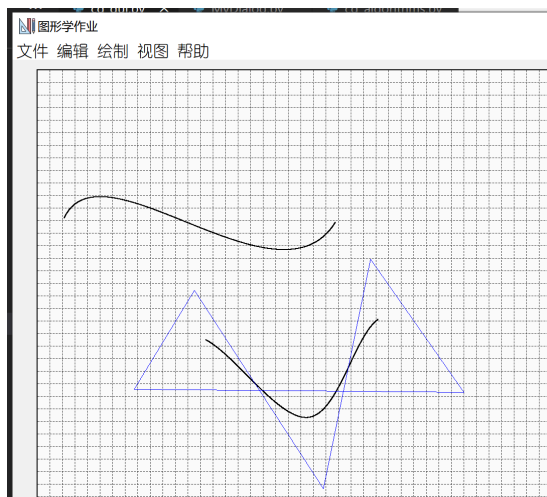
2.再按住鼠标左键移动，绘制出控制点多边形，松开鼠标左键，控制点生成



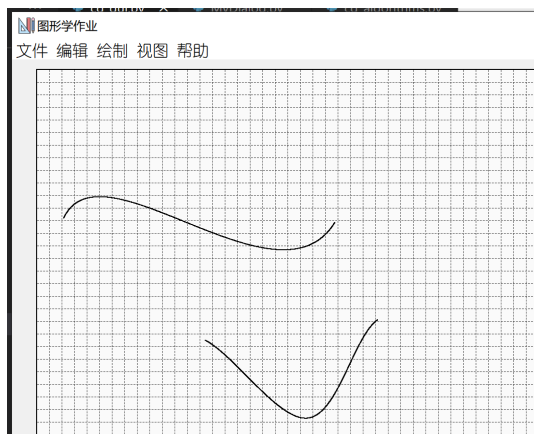
3.再绘制一个控制点（类似样例的形状）



4.再绘制一个控制点（类似样例的形状）



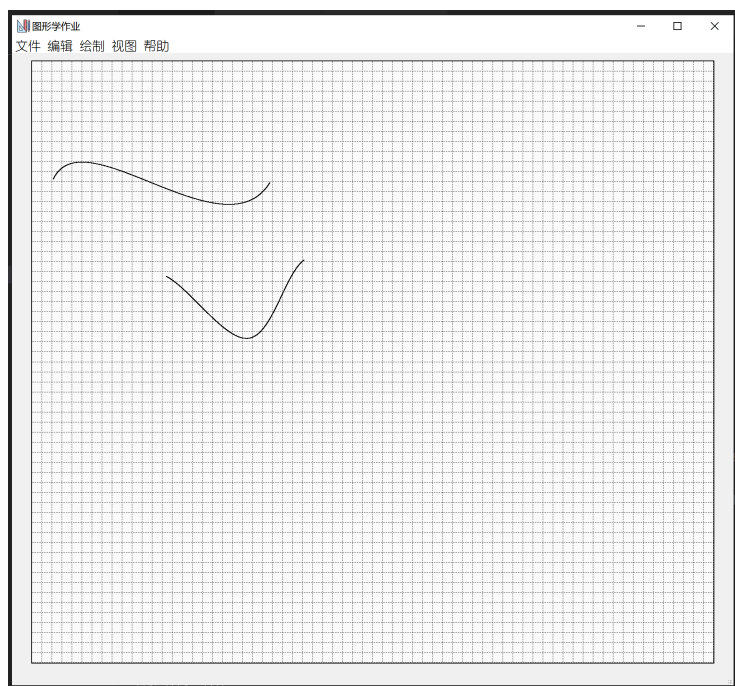
5.按右键结束绘制，控制多边形也随之消失



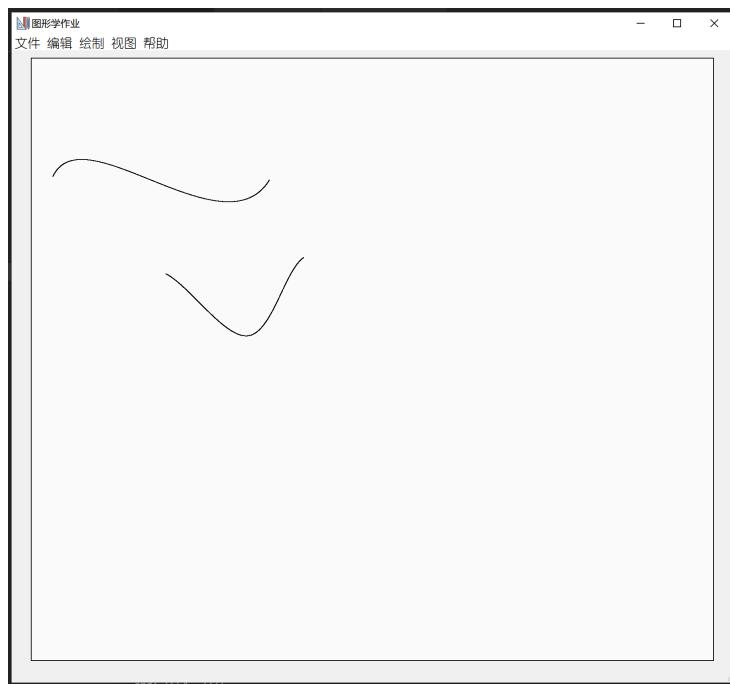
视图

显示网格

“显示网格”按下前：



“显示网格”按下后：



帮助

系统文档

点击“系统文档”将会跳转至 [网页版系统文档说明](#)。