CGproject

系统实现

使用Python3语言编程，实现一个绘图系统

1. 实现的内容

- 核心算法模块（各种图元的生成、编辑算法）：cg_algorithms.py
  - 只依赖math库
  - 见src/cg_algorithms.py
- 命令行界面（CLI）程序：cg_cli.py
  - 读取包含了图元绘制指令序列的文本文件，依据指令调用核心算法模块中的算法**绘制图形**以及**保存图像**
  - 依赖第三方库numpy和Pillow，用于将画布保存为图像
  - 程序接受两个外部参数：指令文件的路径和图像保存目录
  - 测试程序时的指令格式如下：

    ```
    python cg_cli.py input_path output_dir
    ```

  - 见src/cg_cli.py
- 用户交互界面（GUI）程序：cg_gui.py
  - 以鼠标交互的方式，通过鼠标事件获取所需参数并调用核心算法模块中的算法**将图元绘制到屏幕上**，或**对图元进行编辑**
  - 选择GUI库为PyQt5
  - 测试程序时的指令格式如下：

    ```
    python cg_gui.py
    ```

  - 见src/cg_gui.py

2. 指令文件格式

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 <= 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~~对角顶点坐标

- 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~~对角顶点坐标

  algorithm: string, 裁剪使用的算法，包括"Cohen-Sutherland"和"Liang-Barsky"