

# 计算机图形学

## 作业发布报告

2019-09-09

# 作业内容

## 系统功能要求：

- 要实现的内容：
  - ① 核心算法模块（各种图元的生成、变换算法）
  - ② 文件输入接口（读取包含了图元绘制指令序列的文本文件，依据指令调用①中的算法绘制图形以及保存图像）
  - ③ 用户交互接口（以鼠标交互的方式，通过鼠标事件获取所需参数并调用①中的算法将图元绘制到屏幕上）
- 编程语言不限（C++，Python，Java等）
- 开发平台不限（Windows，Mac OS，Linux等）
- GUI开发框架不限（Qt，Tkinter，Web等）
- **不能用框架中的函数直接绘制图元，要自己实现课程中讲到的算法**

# 要实现的功能/算法说明

命令行界面程序接受两个参数：

- 指令序列文件（每行一条指令）
- 图像保存目录

指令包括：

- 重置画布
- 保存画布
- 设置画笔颜色
- 绘制线段
- 绘制多边形
- 绘制椭圆
- 绘制曲线
- 对图元平移
- 对图元旋转
- 对图元缩放
- 对线段裁剪

# 要实现的功能/算法说明

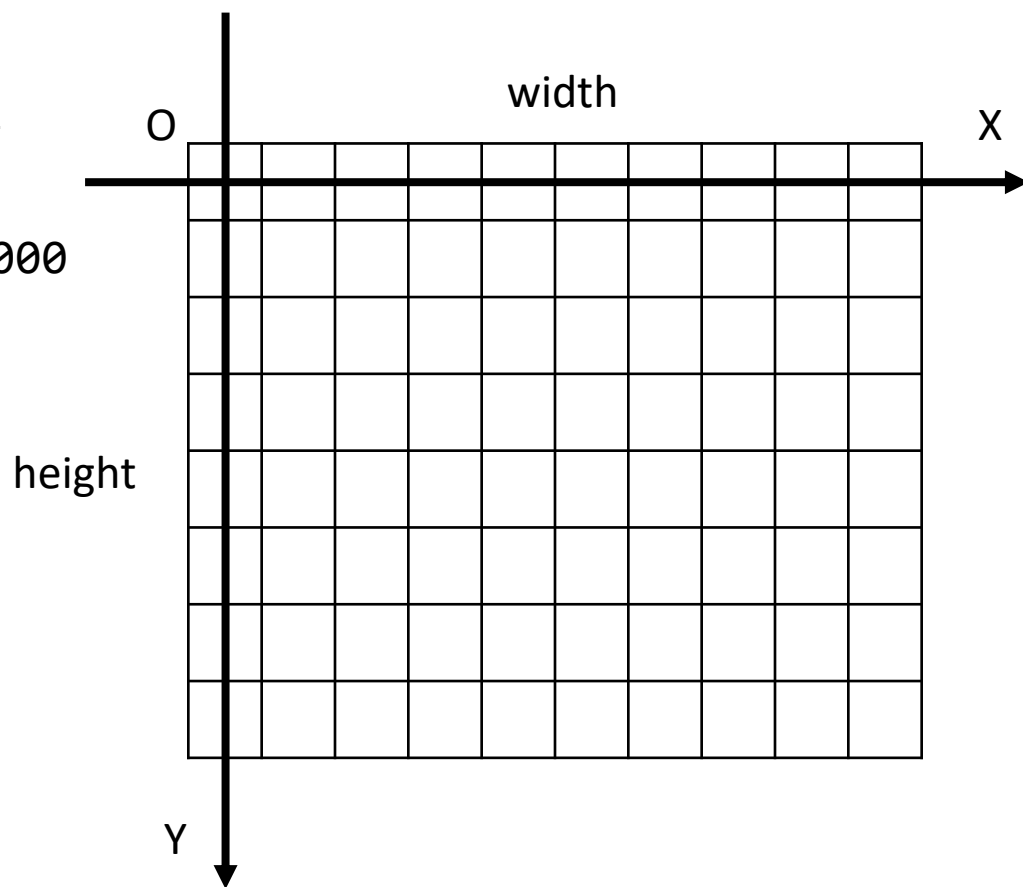
重置画布:

`resetCanvas width height`

清空当前画布，并重新设置宽高

`width, height: int`

`100 <= width, height <= 1000`



# 要实现的功能/算法说明

保存画布：

`saveCanvas name`

将当前画布保存为位图name.bmp

`name: string`

# 要实现的功能/算法说明

设置画笔颜色：

setColor R G B

R, G, B: int

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

# 要实现的功能/算法说明

绘制线段:

```
drawLine id x1 y1 x2 y2 algorithm
```

**id: int**

图元编号，每个图元的编号是唯一的

**x1, y1, x2, y2: int**

起点、终点坐标

**algorithm: string**

绘制使用的算法，包括“DDA”和“Bresenham”

# 要实现的功能/算法说明

绘制多边形:

```
drawPolygon id n algorithm
```

```
x1 y1 x2 y2 ... xn yn
```

`id: int`

图元编号，每个图元的编号是唯一的

`n: int`

顶点数

`x1, y1, x2, y2 ... : int`

顶点坐标

`algorithm: string`

绘制使用的算法，包括“DDA”和“Bresenham”



# 要实现的功能/算法说明

绘制椭圆（中点圆生成算法）：

`drawEllipse id x y rx ry`

`id: int`

图元编号，每个图元的编号是唯一的

`x, y: int`

圆心坐标

`rx, ry: int`

长短轴半径

# 要实现的功能/算法说明

绘制曲线：

```
drawCurve id n algorithm
```

```
x1 y1 x2 y2 ... xn yn
```

**id: int**

图元编号，每个图元的编号是唯一的

**n: int**

控制点数量

**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”

# 要实现的功能/算法说明

对图元平移:

`translate id dx dy`

`id: int`

要平移的图元编号

`dx, dy: int`

平移向量

# 要实现的功能/算法说明

对图元旋转：

`rotate id x y r`

`id: int`

要旋转的图元编号

`x, y: int`

旋转中心

`r: int`

顺时针旋转角度 (°)

# 要实现的功能/算法说明

对图元缩放:

`scale id x y s`

`id: int`

要缩放的图元编号

`x, y: int`

缩放中心

`s: float`

缩放倍数

# 要实现的功能/算法说明

对线段裁剪:

`clip id x1 y1 x2 y2 algorithm`

`id: int`

要裁剪的图元编号

`x1, y1, x2, y2: int`

裁剪窗口左下、右上角坐标

`algorithm: string`

裁剪使用的算法, 包括“Cohen-Sutherland”和“Liang-Barsky”

# 示例

input.txt

```
resetCanvas 100 100
setColor 255 0 0
drawLine 5 7 39 93 71 DDA
saveCanvas output_1
setColor 0 162 232
drawLine 233 96 35 15 58 Bresenham
saveCanvas output_2
clip 5 33 10 70 58 Cohen-Sutherland
saveCanvas output_3
rotate 5 33 49 90
saveCanvas output_4
drawEllipse 123 31 49 8 18
setColor 0 0 0
drawPolygon 666 6 DDA
44 22 73 30 74 77 59 56 32 63 25 49
saveCanvas output_5
rotate 666 59 56 -90
translate 666 -8 -18
saveCanvas output_6
resetCanvas 160 100
drawCurve 10 4 Bezier
28 34 9 86 61 4 129 42
setColor 0 255 0
drawCurve 11 4 Bezier
114 88 60 92 99 3 42 24
saveCanvas output_7
```



output\_1.bmp



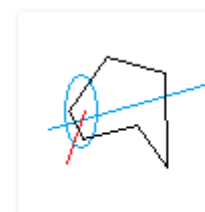
output\_2.bmp



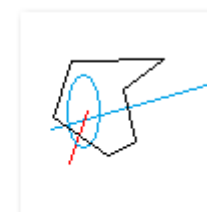
output\_3.bmp



output\_4.bmp



output\_5.bmp



output\_6.bmp



output\_7.bmp