

CG Program系统使用说明书

张昊*

(南京大学 计算机科学与技术系, 南京 210093)

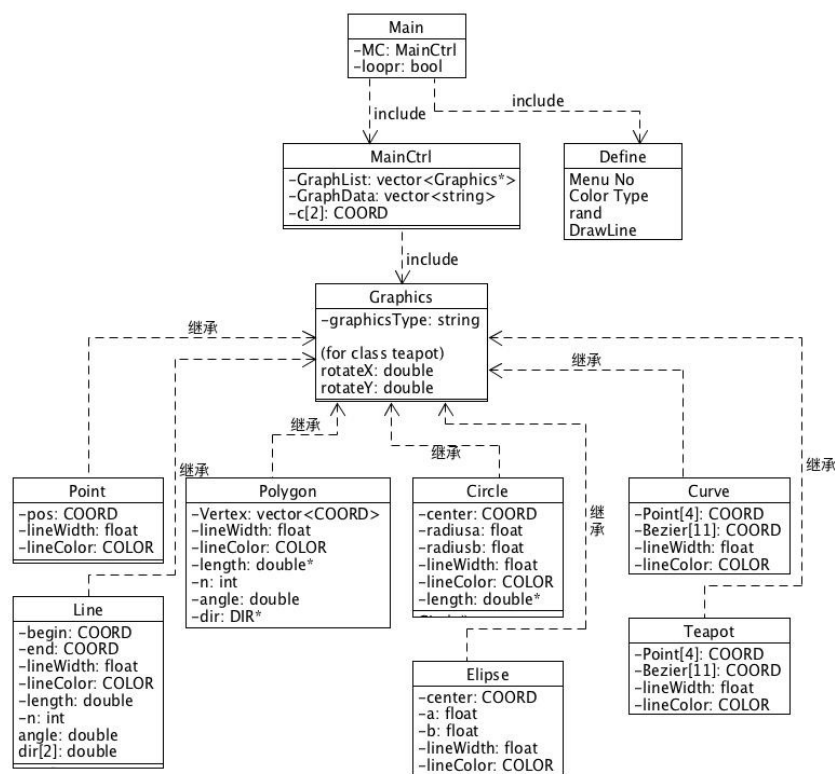
摘要： 如果不考虑一个架构完整的图形处理系统,单独实现某条直线或多边形的输入编辑输出功能是非常简易的,甚至可以说是无趣的,而提到设计一个完整的系统就不仅仅是解决某条直线某种算法的问题,这上升到考察你计划和架构整个系统框架的能力,这将是一个更大的挑战。《计算机图形学》的这份学期大作业无疑与此类似,检测学生对某种图形的生成和变换是基础,更加需要学生去掌握对这些图形显示和编辑功能的控制能力,从无到有,从有到完整,从设计这个系统的每一步体会、反思、提高自己的编程能力。

关键词： 计算机图形学、OpenGL、图形输入、图形变换、图形保存

一、引言

《计算机图形学》学期大作业要求实现一个功能完善的图形处理系统。图形种类主要包括点、直线、多边形、圆、椭圆、曲线、三维图形；系统的功能主要包括图形数据的输入、线宽、颜色设置、平移、缩放、旋转、裁剪、填充、保存等；历时将近三个月的编程和完善，该系统的要求已全部实现，并在与用户的交互上做了一些改进。

二、模块划分及联系：



* 作者简介：姓名张昊,学号141220141,电子邮箱为hao_silence@qq.com

三、模块解析：

1、Graphics模块：

(1) 模块简介：

Graphics模块为各种图形的基础，它充当各种图形的基类，所有的图形都从此处继承和派生。该类的内部成员函数基本上全部是纯虚函数，需要针对具体的图形种类具体实现。

(2) 模块示意图：

Graphics
-graphicsType: string
(for class teapot)
rotateX: double
rotateY: double
Graphics()
~Graphics()
float2str(float f): string
COLOR2str(COLOR c): string
virtual setLineWidth(float lw) = 0: void
virtual setLineColor(float r, float g, float b) = 0: void
virtual Draw() = 0: void
virtual Up() = 0: void
virtual Down() = 0: void
virtual Left() = 0: void
virtual Right() = 0: void
virtual Increase() = 0: void
virtual Decrease() = 0: void
virtual Rotate(char) = 0: void
virtual Fill(float r, float g, float b) = 0: void
virtual Disappear() = 0: void
virtual SaveFile() = 0: String

(3) 数据成员：

graphicsType为图形类型，取值有point、line、polygon、ellipse、circle、curve、teapot等。

rotateX和rotateY是针对于三维图形teapot旋转功能的角度变量，因为三维图形的旋转有对应的键盘响应事件，所以把这两个变量放在了这里，一定程度上违背了C++的设计原则，但也是不得已而为之。

(4) 成员函数：

Graphics()和~Graphics()为该类的构造函数和析构函数，在此没有具体的成员初始化。

virtual void setLineWidth(float lw) = 0为设置线宽的纯虚函数，需要在具体的派生类中实现。

virtual void setLineColor(float r, float g, float b) = 0为设置颜色的纯虚函数，此处不实现。

virtual void Draw() = 0为生成图形的函数，需在具体的图形类中具体实现

virtual void Up() = 0为向上移动图形的纯虚函数

virtual void Down() = 0为向下移动图形的纯虚函数

virtual void Left() = 0为向左移动图形的纯虚函数

virtual void Right() = 0为向右移动图形的纯虚函数

virtual void Increase() = 0为放大图形的纯虚函数
 virtual void Decrease() = 0为缩小图形的纯虚函数
 virtual void Rotate(char c) = 0为旋转图形的纯虚函数
 virtual void Fill(float r, float g, float b) = 0为填充图形的纯虚函数
 virtual void Disappear() = 0为清空图形的纯虚函数
 virtual string SaveFile() = 0为保存图形数据的纯虚函数
 string float2str(float f)是将float数据转换为string类型的函数，用于将线宽转换成string
 string COLOR2str(COLOR c)是将COLOR类型转换为string的函数，COLOR为自定义结构体

2、Point模块：

(1) 模块简介：Point是点的类定义，继承于Graphics类，包含点的坐标、线宽、颜色等属性，以及对点的线宽、颜色设置、绘制、移动、擦除、保存等功能。

(2) 模块示意图：

Point
-pos: COORD -lineWidth: float -lineColor: COLOR
Point() Point(COORD p) Point(float x, float y) ~Point() getPos() const: COORD setPos(COORD p): void getLineWidth() const: Float setLineWidth(float p): void getLineColor() const: COLOR setLineColor(float r, float g, float b): void Draw(): void Up(): void Down(): void Left(): void Right(): void Increase(): void Decrease(): void Rotate(char c): void Fill(float r, float g, float b): void Disappear(): void SaveFile(): string

(3) 数据成员：

pos为COORD类型，标示点的坐标。

lineWidth为float类型，标示点的size。

lineColor为COLOR类型，COLOR有r、g、b三个成员，标示点的颜色。

(4) 成员函数：

Point()和Point(COORD p)以及Point(float x, float y)为Point的三个构造函数

~Point()为Point的析构函数

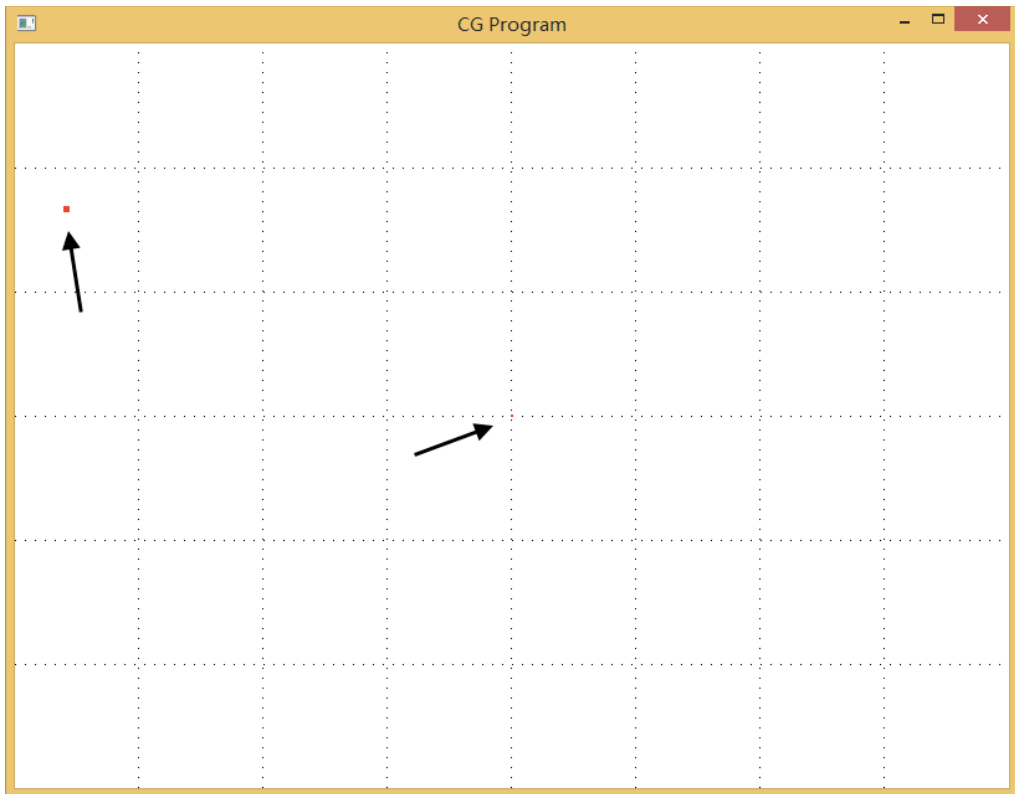
COORD getPos() const是获取点坐标的函数

void setPos(COORD p)是设置点坐标的函数

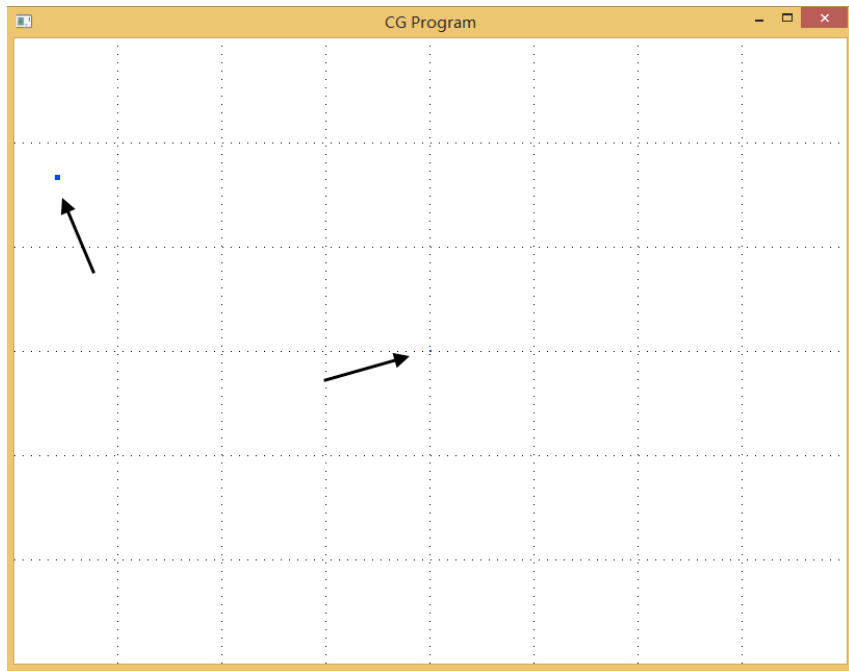
```
float getLineWidth() const;    获取线宽
void setLineWidth(float p);    设置线宽
COLOR getLineColor() const;    获取线条颜色
void setLineColor(float r, float g, float b);    设置线条颜色
void Draw();    绘制图形的函数
void Up();    上移图形的函数
void Down();    下移图形的函数
void Left();    左移图形的函数
void Right();    右移图形的函数
void Increase();    放大图形
void Decrease();    缩小图形
void Rotate(char c);    旋转图形
void Fill(float r, float g, float b);    填充图形
void Disappear();    擦除图形
string SaveFile();    保存图形数据
```

(5) 功能演示:

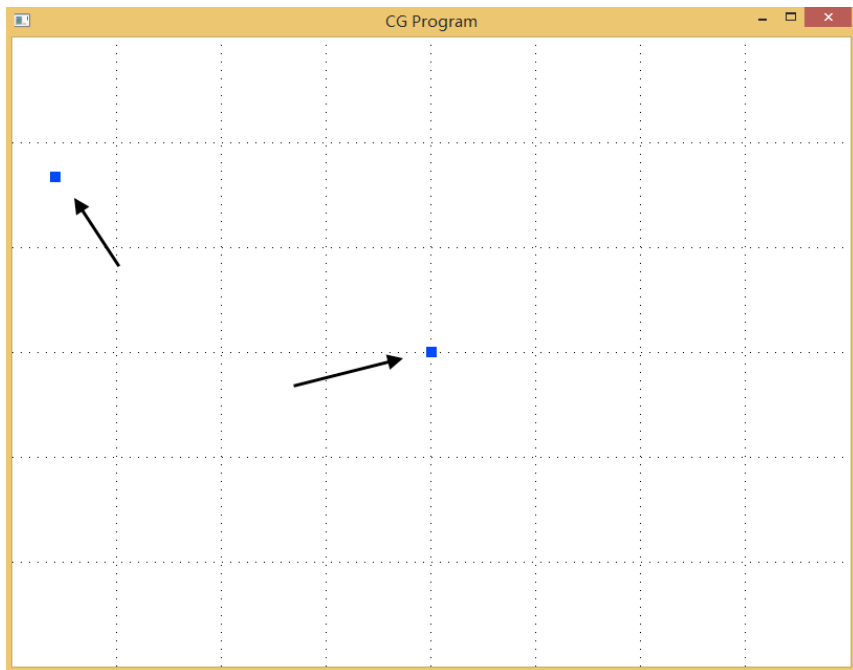
新建点:



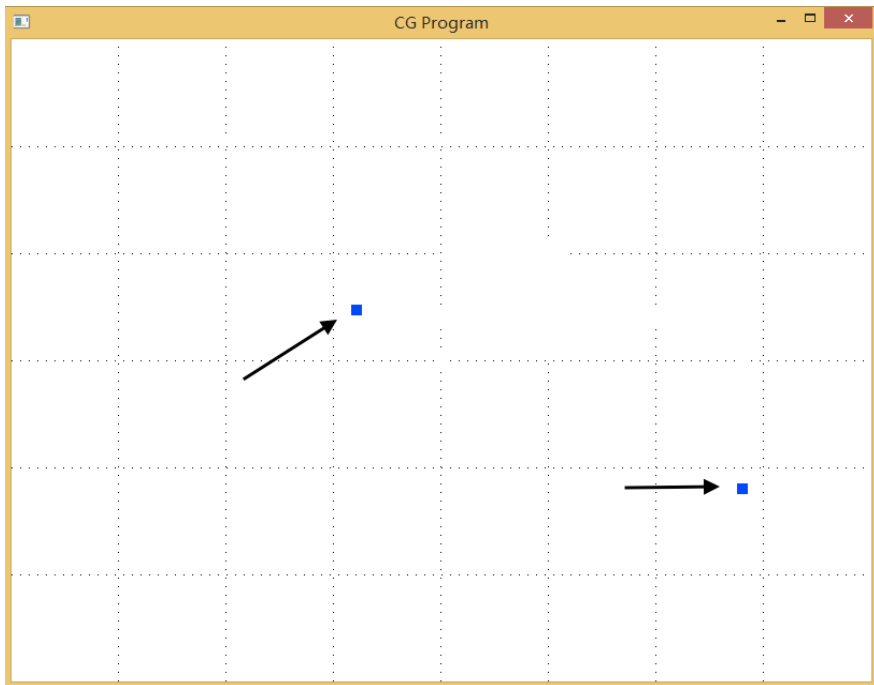
设置颜色：



设置线宽：



平移:



保存图形数据:

```
C:\Windows\system32\cmd.exe

'\\psf\Home\Documents\Visual Studio 2012\Projects\CGProgram\CGProgram'
用作为当前目录的以上路径启动了 CMD.EXE。
UNC 路径不受支持。默认值设为 Windows 目录。
2.84
#CGP# 请输入点的坐标, 用空格隔开 (-400<X<400, -300<Y<300): 0 0
119.115
#CGP# 请输入保存的文件路径: 1.txt
point -79 47 10 <0,0,1>
point 280 -120 10 <0,0,1>
#CGP# 图形数据保存成功! _
```

3、Line模块：

(1) 模块简介：Line是直线的类定义，继承于Graphics类，包含直线端点的坐标、线宽、颜色等属性，以及对直线的线宽、颜色设置、绘制、缩放、移动、旋转、保存等功能。

(2) 模块示意图：

Line
-begin: COORD -end: COORD -lineWidth: float -lineColor: COLOR -length: double -n: int angle: double dir[2]: double
Line() Line(COORD s, COORD e) Line(float sx, float sy, float ex, float ey) ~Line() getLinePos() const: SMALL_RECT setLinePos(COORD b, COORD e): void getLineWidth() const: Float setLineWidth(float p): void getLineColor() const: COLOR setLineColor(float r, float g, float b): void Draw(): void Up(): void Down(): void Left(): void Right(): void Increase(): void Decrease(): void Rotate(char c): void Fill(float r, float g, float b): void Disappear(): void SaveFile(): string

(3) 数据成员：

begin和end为COORD类型，标示两个端点的坐标。

lineWidth为float类型，标示直线的线宽。

lineColor为COLOR类型，COLOR有r、g、b三个成员，标示点的颜色。

length是直线的长度，用begin和end两点的坐标利用距离公式求得。

n、angle是旋转时要用的变量，分别代表旋转计数和每次旋转的角度。

dir[] 数组是代表起点到终点指向得单位方向向量。

(4) 成员函数：

Line(); //Line的默认构造函数

Line(COORD s, COORD e); //用两个COORD型参数初始化直线的两 endpoint

Line(float sx, float sy, float ex, float ey); //用四个坐标值初始化直线的两 endpoint

~Line(); //直线的析构函数

void setLinePos(COORD b, COORD e); //更改直线的位置

SMALL_RECT getLinePos(); 获取直线的端点位置，返回类型是SMALL_RECT

float getLineWidth() const; 获取线宽

void setLineWidth(float p); 设置线宽

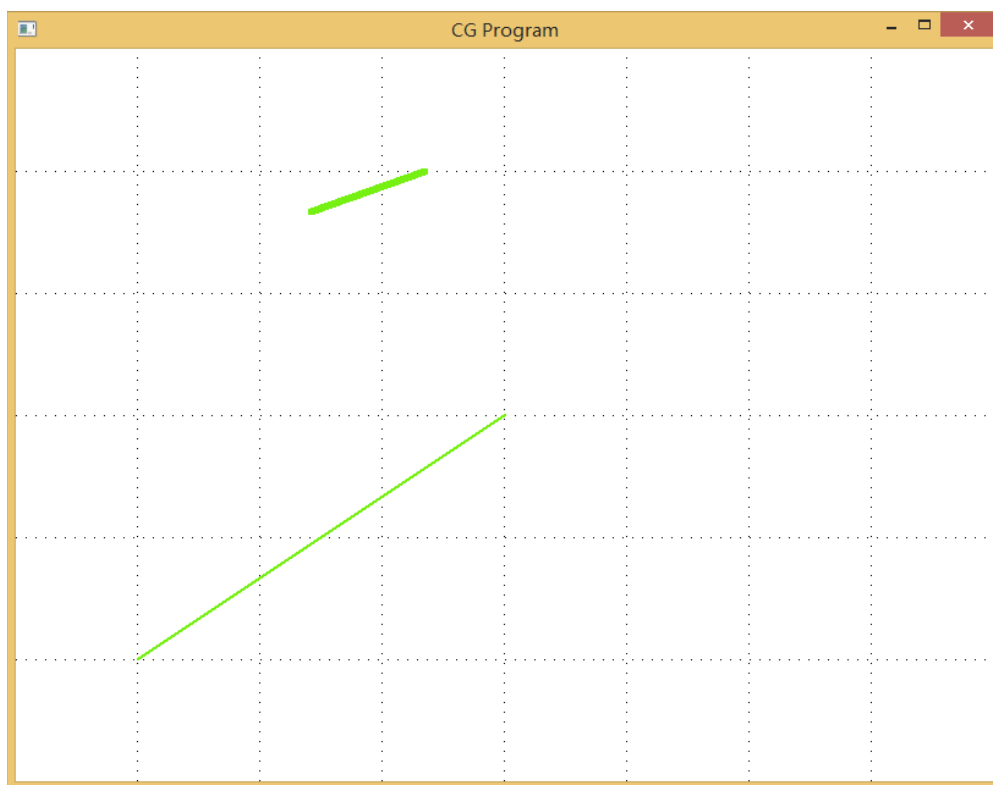
COLOR getLineColor() const; 获取线条颜色

void setLineColor(float r, float g, float b); 设置线条颜色

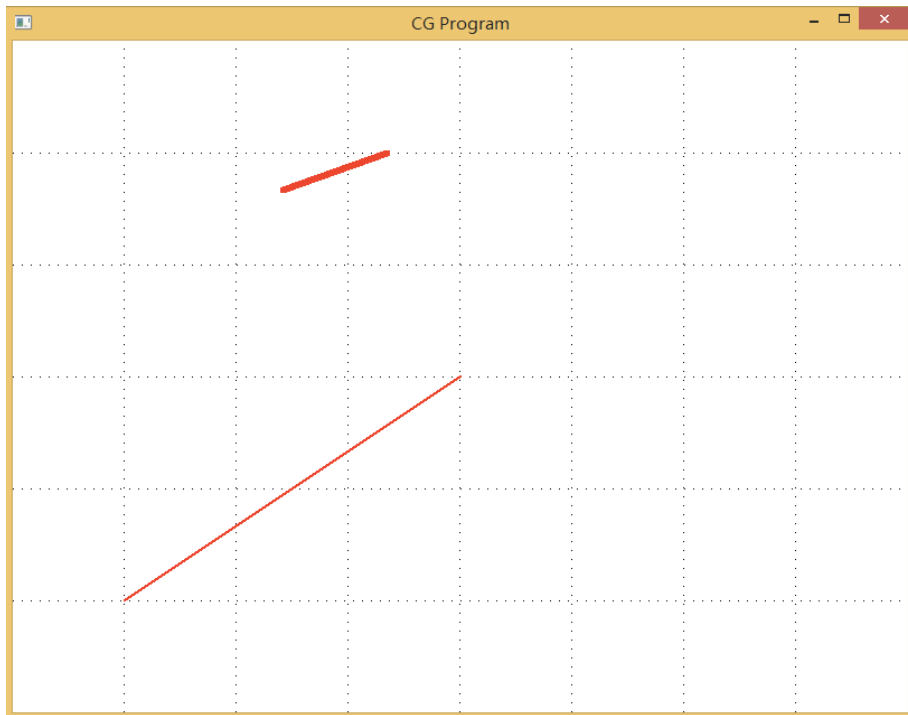
`void Draw();` 绘制图形的函数
`void Up();` 上移图形的函数
`void Down();` 下移图形的函数
`void Left();` 左移图形的函数
`void Right();` 右移图形的函数
`void Increase();` 放大图形
`void Decrease();` 缩小图形
`void Rotate(char c);` 旋转图形
`void Fill(float r, float g, float b);` 填充图形
`void Disappear();` 擦除图形
`string SaveFile();` 保存图形数据

(5) 功能演示:

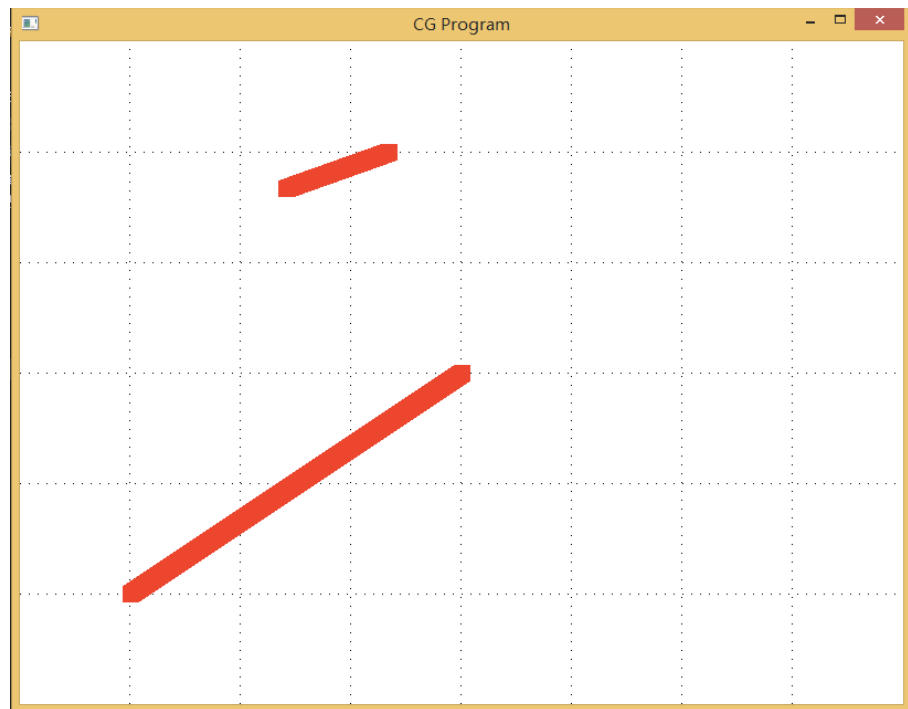
新建直线:



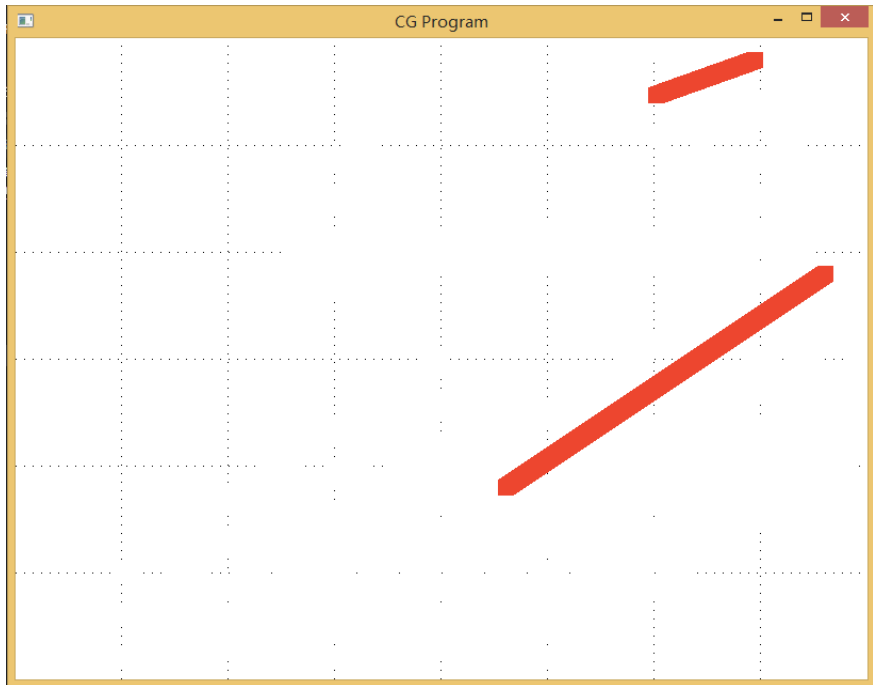
设置颜色:



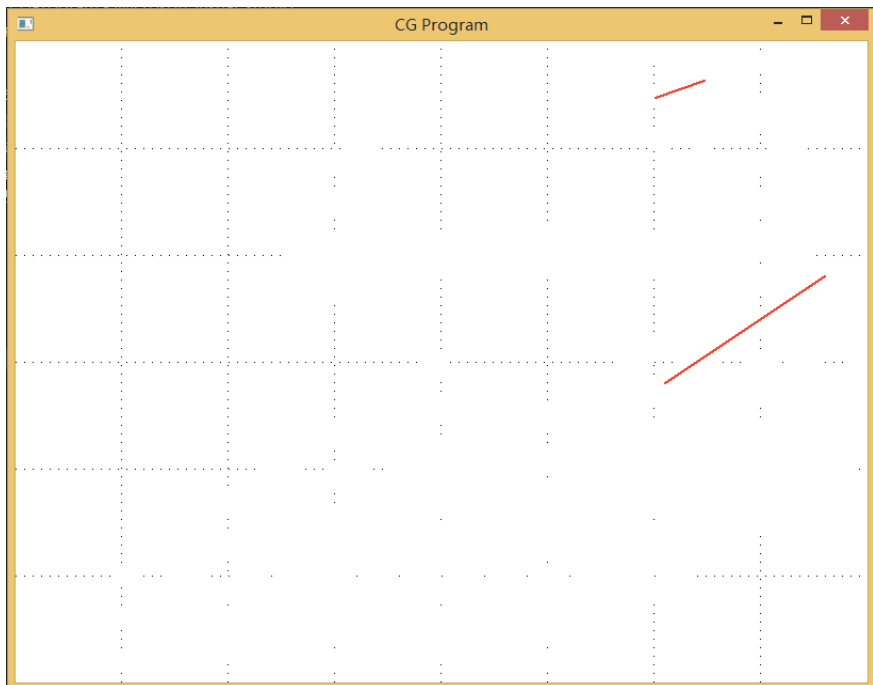
设置线宽:



平移:



缩放:



4、Polygon模块：

(1) 模块简介：Polygon是多边形的类定义，继承于Graphics类，包含多边形各顶点的坐标、线宽、颜色等属性，以及对多边形的线宽、颜色设置、绘制、缩放、移动、旋转、保存等功能。

(2) 模块示意图：

Polygon
~Vertex: vector<COORD> ~lineWidth: float ~lineColor: COLOR ~length: double* ~n: int ~angle: double ~dir: DIR*
Polygon(); Polygon(int n, COORD c[]); ~Polygon(); getVertex(int n): COORD getLineWidth() const: Float setLineWidth(float p): void getLineColor() const: COLOR setLineColor(float r, float g, float b): void Draw(): void Up(): void Down(): void Left(): void Right(): void Increase(): void Decrease(): void Rotate(char c): void Fill(float r, float g, float b): void Disappear(): void SaveFile(): string

(3) 数据成员：

Vertex为COORD类型的vector数组，存储多边形的顶点坐标。

lineWidth为float类型，标示直线的线宽。

lineColor为COLOR类型，COLOR有r、g、b三个成员，标示点的颜色。

length数组是其余各点到第一个点的长度，利用两点之间的距离公式求得。

n、angle是旋转时要用的变量，分别代表旋转计数和每次旋转的角度。

dir[] 数组是代表第一个顶点到其余各点指向得单位方向向量，其中DIR为一个struct，有double型的x和y两个成员。

(4) 成员函数：

Polygon(); 多边形的默认构造函数

Polygon(int n, COORD c[]); 表示顶点数为n，c[]为顶点坐标的数组

~Polygon(); 析构函数

COORD getVertex(int n); 获取指定下标的顶点坐标

float getLineWidth() const; 获取线宽

void setLineWidth(float p); 设置线宽

COLOR getLineColor() const; 获取线条颜色

void setLineColor(float r, float g, float b); 设置线条颜色

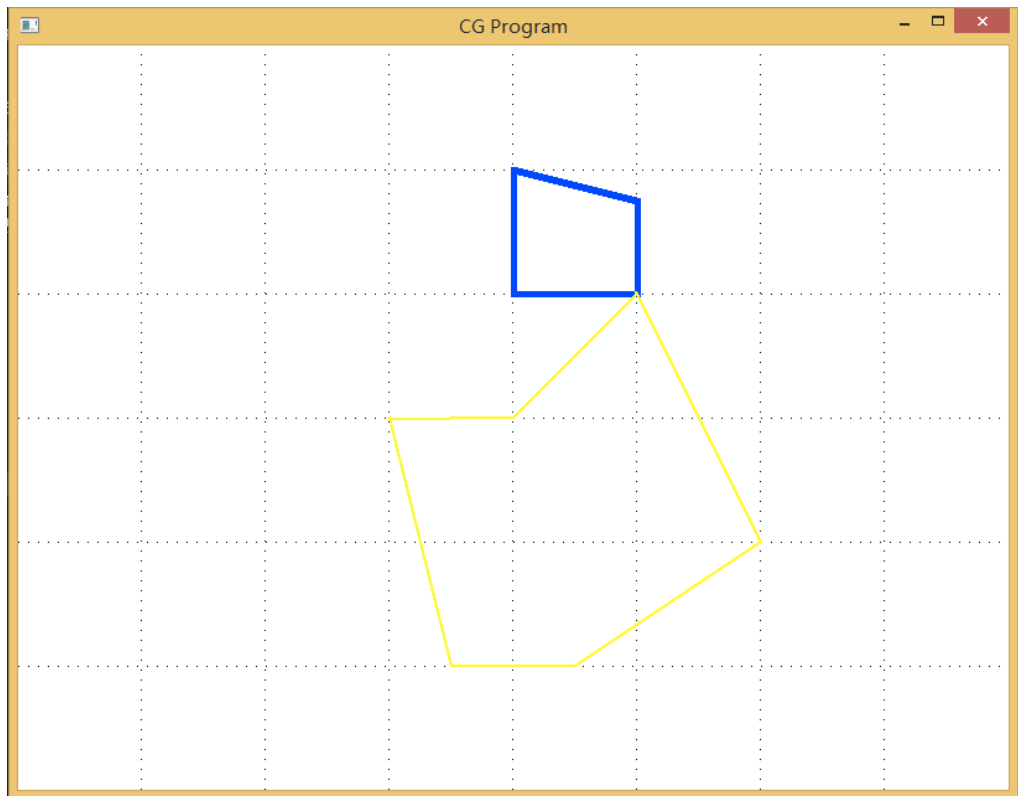
void Draw(); 绘制图形的函数

void Up(); 上移图形的函数

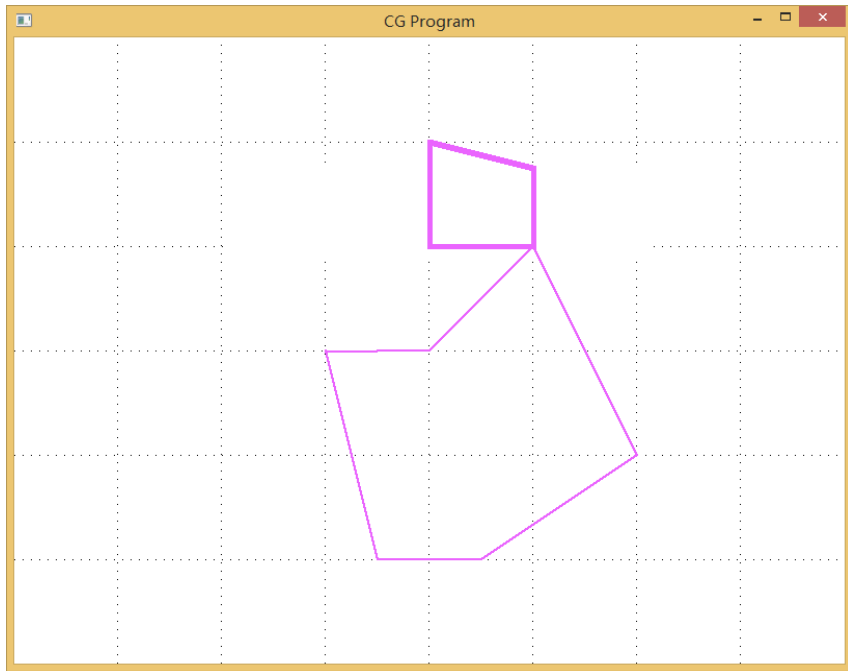
void Down(); 下移图形的函数

void Left(); 左移图形的函数
void Right(); 右移图形的函数
void Increase(); 放大图形
void Decrease(); 缩小图形
void Rotate(char c); 旋转图形
void Fill(float r, float g, float b); 填充图形
void Disappear(); 擦除图形
string SaveFile(); 保存图形数据

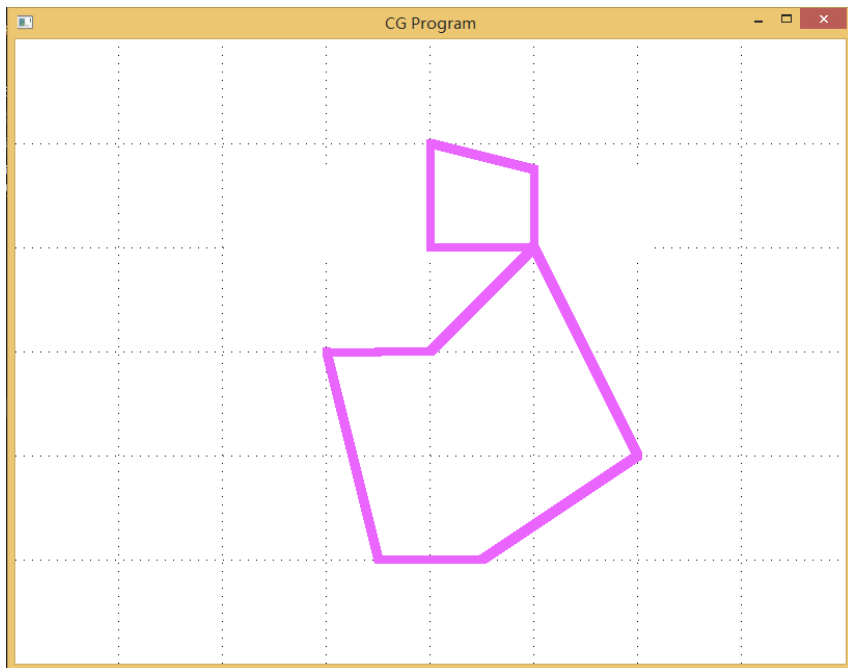
(5) 功能演示：
新建多边形：



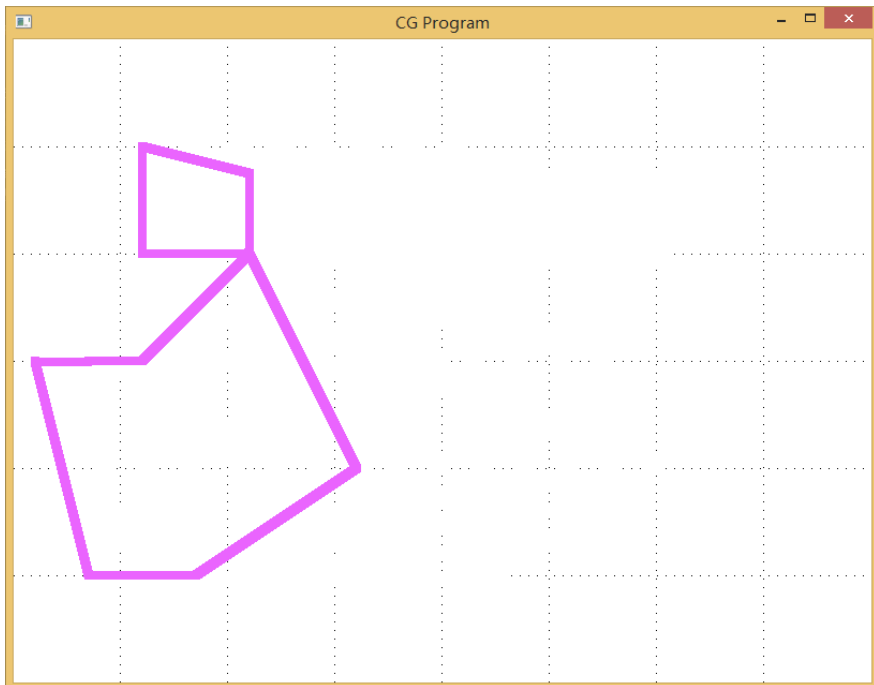
设置颜色:



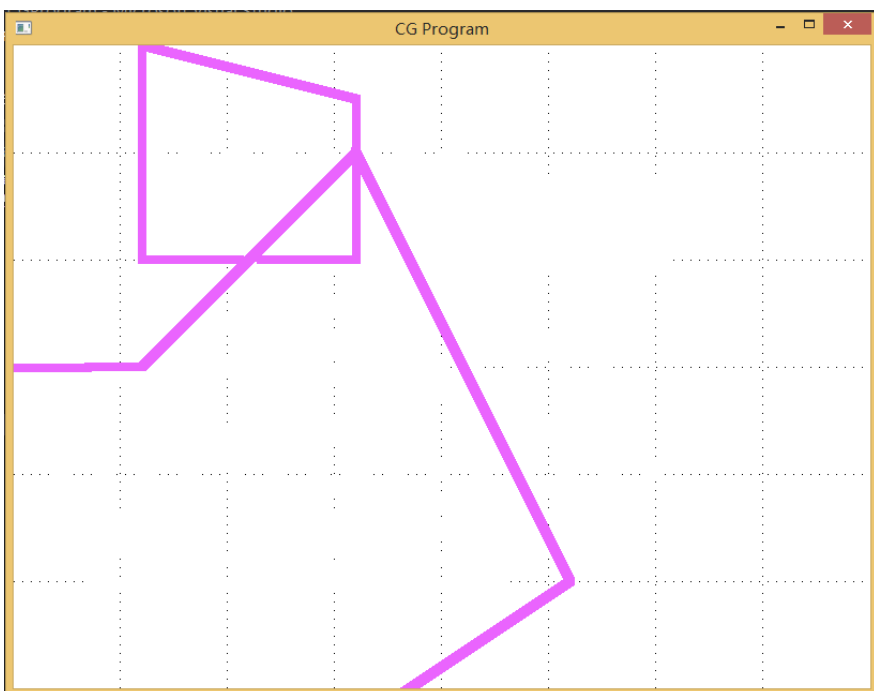
设置线宽:



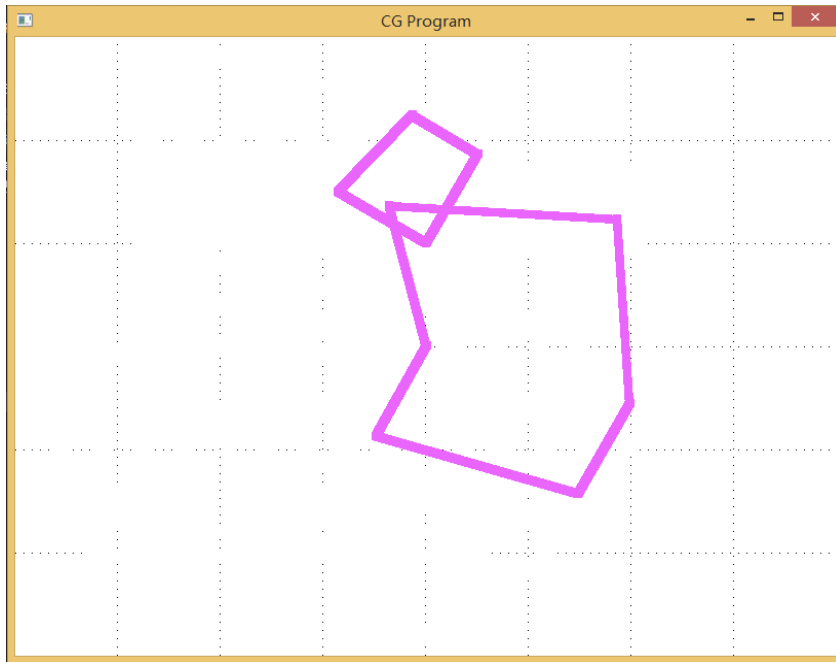
平移:



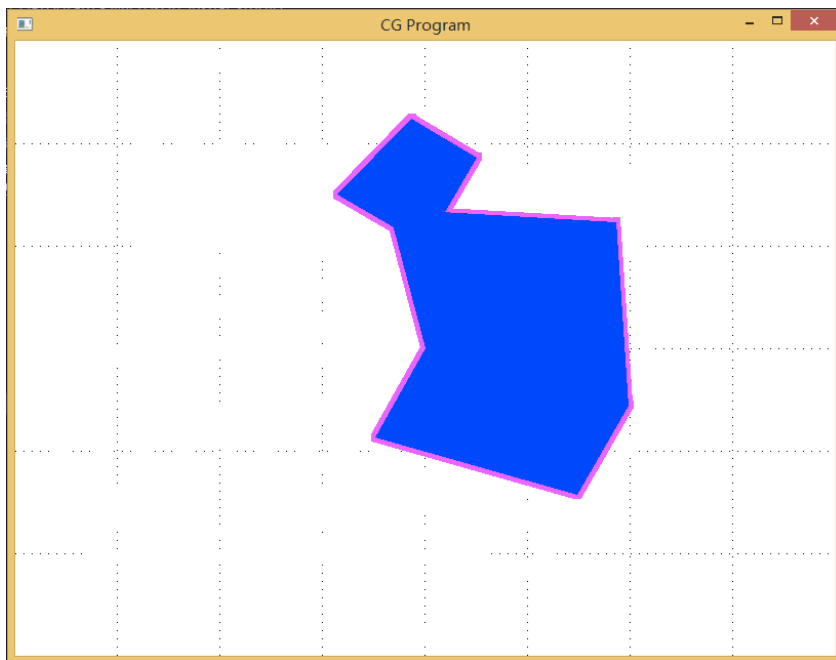
缩放:



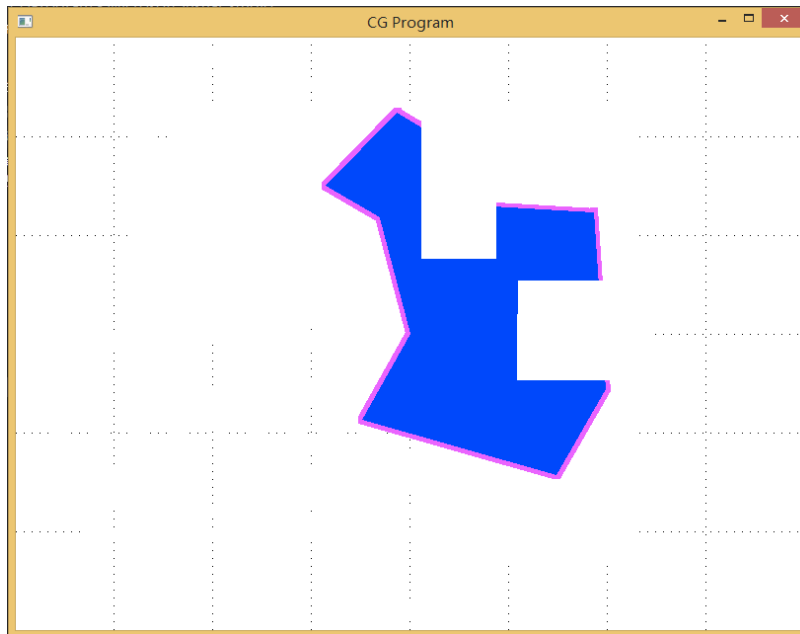
旋转:



填充:



裁剪:



保存数据:

```
C:\Windows\system32\cmd.exe

'\\psf\Home\Documents\Visual Studio 2012\Projects\CGProgram\CGProgram'
用作为当前目录的以上路径启动了 CMD.EXE。
UNC 路径不受支持。默认值设为 Windows 目录。
73.108
73.108
垂直x轴
k<1
垂直x轴
垂直y轴
#CGP# 请输入保存的文件路径: 1.txt
polygon 0 100 0 200 100 175 100 100 5 <0,0,1>
#CGP# 图形数据保存成功!
```

5、Circle模块：

(1) 模块简介：Circle是圆的类定义，继承于Graphics类，包含圆心的坐标、半径长度、线宽、颜色等属性，以及对圆的线宽、颜色设置、绘制、缩放、移动、保存等功能。

(2) 模块示意图：

Circle
-center: COORD -radiusa: float -radiusb: float -lineWidth: float -lineColor: COLOR -length: double*
Circle() Circle(COORD c, float r) ~Circle() getLineWidth() const: Float setLineWidth(float p): void getLineColor() const: COLOR setLineColor(float r, float g, float b): void Draw(): void Up(): void Down(): void Left(): void Right(): void Increase(): void Decrease(): void Rotate(char c): void Fill(float r, float g, float b): void Disappear(): void SaveFile(): string

(3) 数据成员：

center为COORD类型的变量，存储圆心坐标。

radius和radiusb相等，都表示圆的半径，只是为了迎合椭圆而设为两个变量当作长短轴。

lineWidth为float类型，标示直线的线宽。

lineColor为COLOR类型，COLOR有r、g、b三个成员，标示点的颜色。

(4) 成员函数：

Circle();

Circle(COORD c, float r);

~Circle();

float getLineWidth() const; 获取线宽

void setLineWidth(float p); 设置线宽

COLOR getLineColor() const; 获取线条颜色

void setLineColor(float r, float g, float b); 设置线条颜色

void Draw(); 绘制图形的函数

void Up(); 上移图形的函数

void Down(); 下移图形的函数

void Left(); 左移图形的函数

void Right(); 右移图形的函数

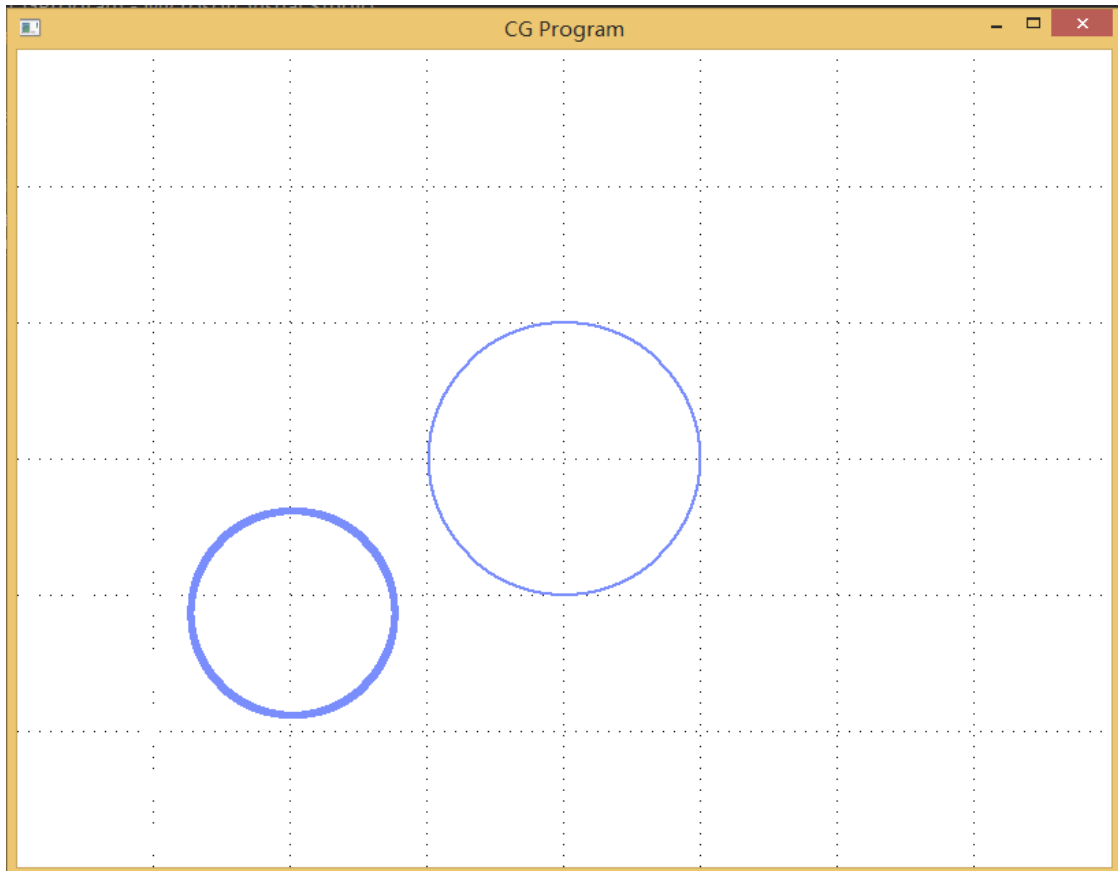
void Increase(); 放大图形

void Decrease(); 缩小图形

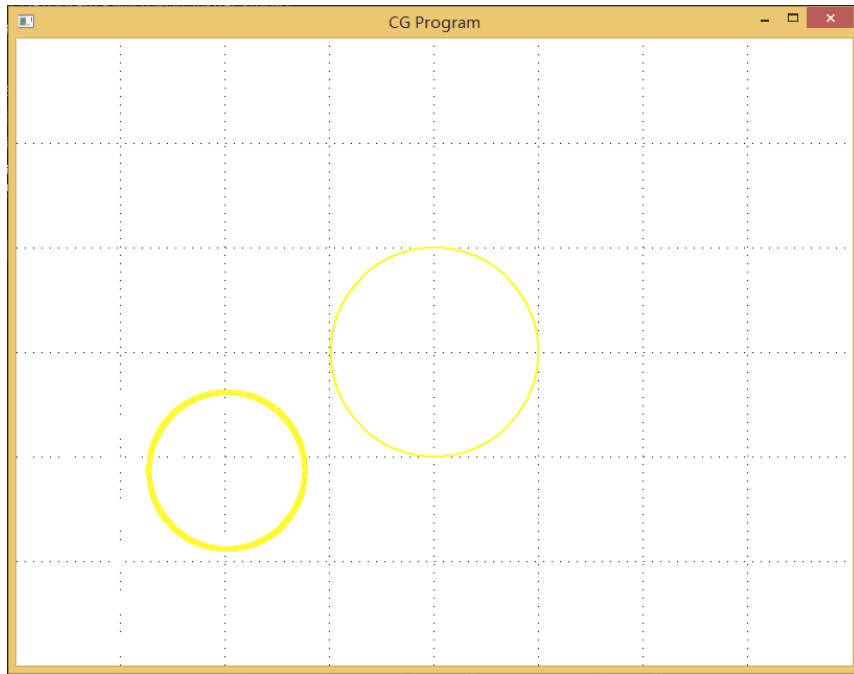
`void Rotate(char c);` 旋转图形
`void Fill(float r, float g, float b);` 填充图形
`void Disappear();` 擦除图形
`string SaveFile();` 保存图形数据

(5) 功能演示:

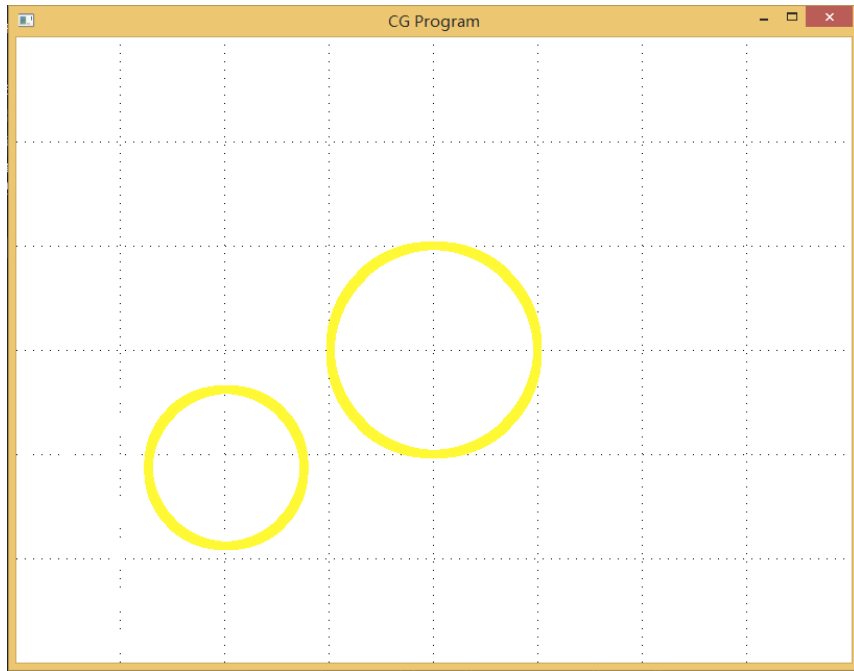
新建圆:



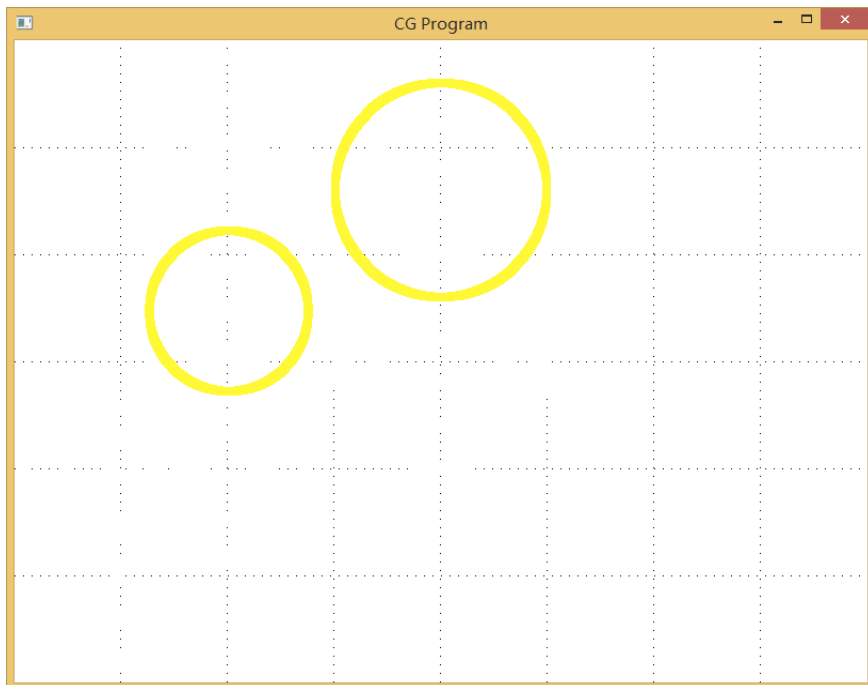
设置颜色：



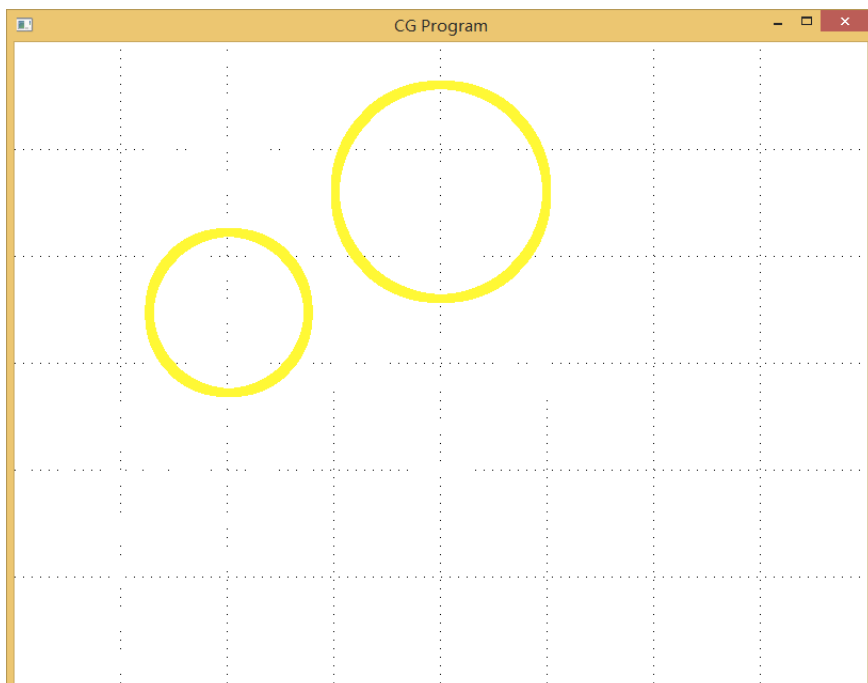
设置线宽：



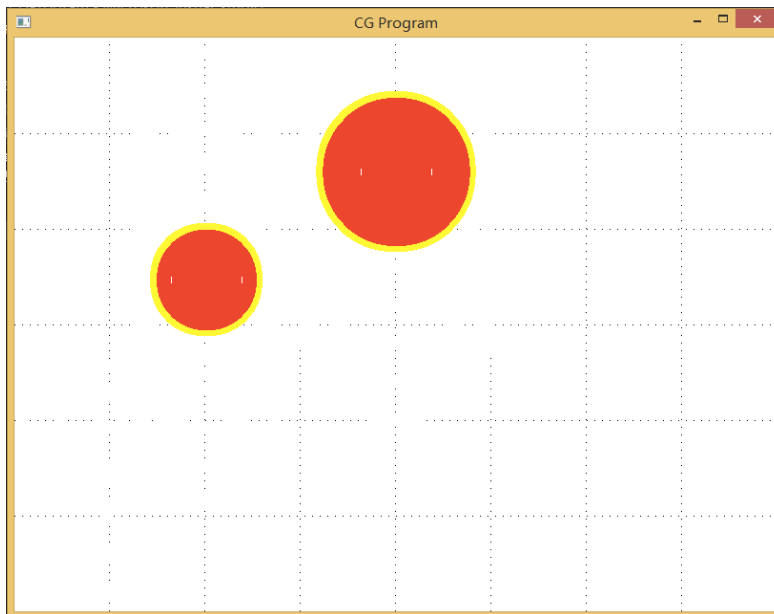
平移:



缩放:



填充:



保存:

```
C:\Windows\system32\cmd.exe
用作为当前目录的以上路径启动了 CMD.EXE。
UNC 路径不受支持。默认值设为 Windows 目录。
-114,108
-114,108
#CGP# 请输入圆的圆心坐标和半径, 空格隔开 (-400<X<400, -300<Y<300): 0 0 100
85,145
85,145
#CGP# 请输入保存的文件路径: 1.txt
circle -199 47 55 8 <1,1,0>
circle 0 160 80 8 <1,1,0>
#CGP# 图形数据保存成功!
```

6、Ellipse模块：

(1) 模块简介：Ellipse是椭圆的类定义，继承于Graphics类，包含椭圆中心的坐标、长短周长、线宽、颜色等属性，以及线宽、颜色设置、绘制、缩放、移动、保存等功能。

(2) 模块示意图：

Ellipse
-center: COORD -a: float -b: float -lineWidth: float -lineColor: COLOR
Ellipse() Ellipse(COORD c, float a, float b) ~Ellipse() getLineWidth() const: Float setLineWidth(float p): void getLineColor() const: COLOR setLineColor(float r, float g, float b): void Ellipsepot(int x0, int y0, int x, int y): void setPixel(int x, int y): void Draw(): void Up(): void Down(): void Left(): void Right(): void Increase(): void Decrease(): void Rotate(char c): void Fill(float r, float g, float b): void Disappear(): void SaveFile(): string

(3) 数据成员：

center为COORD类型的变量，存储中心坐标。

a是椭圆的长轴，b是椭圆的短轴。

lineWidth为float类型，标示直线的线宽。

lineColor为COLOR类型，COLOR有r、g、b三个成员，标示点的颜色。

(4) 成员函数：

Ellipse(); 默认无参数的构造函数

Ellipse(COORD c, float a, float b); 有参的构造函数，c为中心坐标，a、b分别为长短轴

~Ellipse(); 析构函数

float getLineWidth() const; 获取线宽

void setLineWidth(float p); 设置线宽

COLOR getLineColor() const; 获取线条颜色

void setLineColor(float r, float g, float b); 设置线条颜色

void Draw(); 绘制图形的函数

void Up(); 上移图形的函数

void Down(); 下移图形的函数

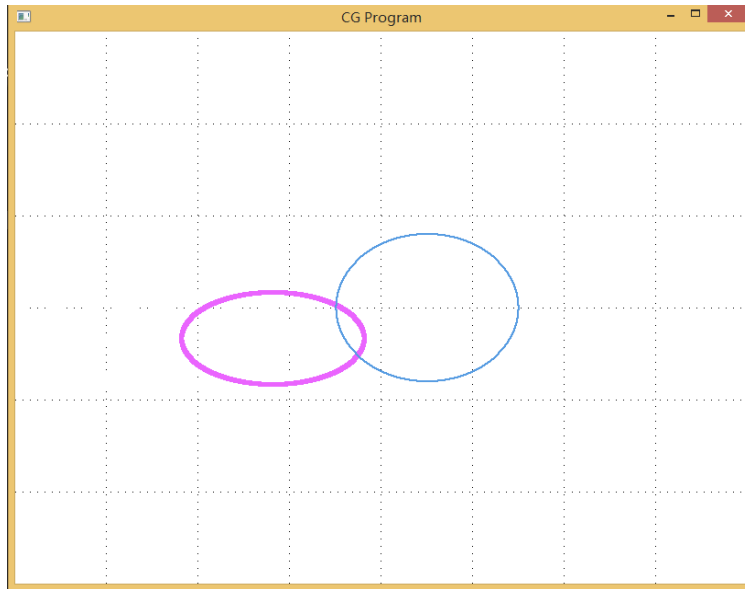
void Left(); 左移图形的函数

void Right(); 右移图形的函数

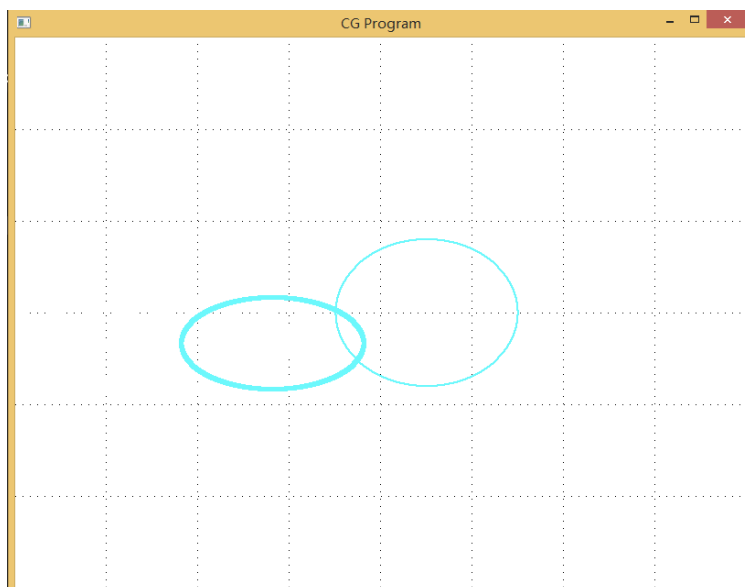
```
void Increase();    放大图形  
void Decrease();   缩小图形  
void Rotate(char c);  旋转图形  
void Fill(float r, float g, float b);  填充图形  
void Disappear();   擦除图形  
string SaveFile();  保存图形数据
```

(5) 功能演示:

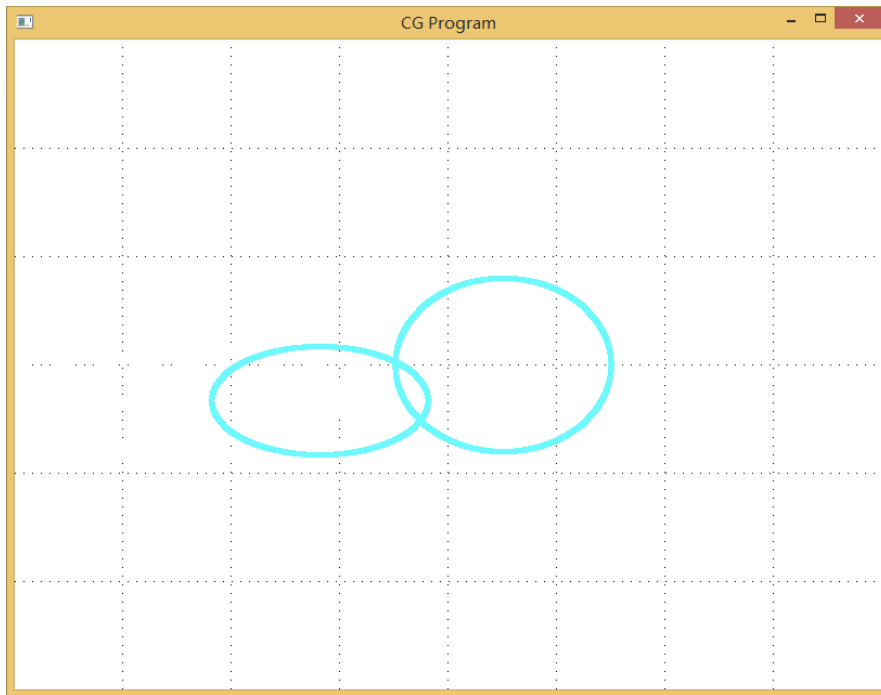
新建椭圆:



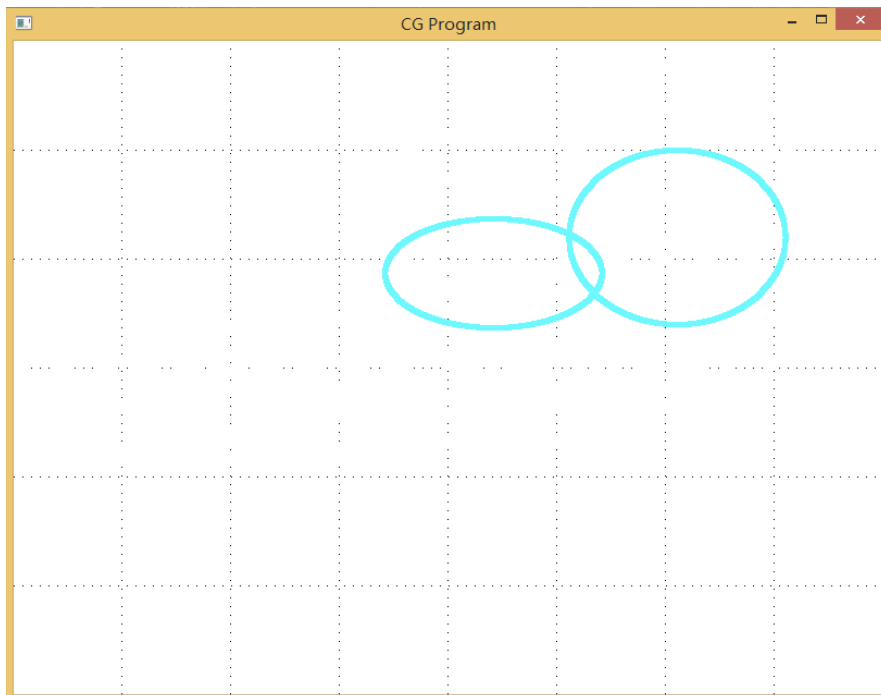
设置颜色:



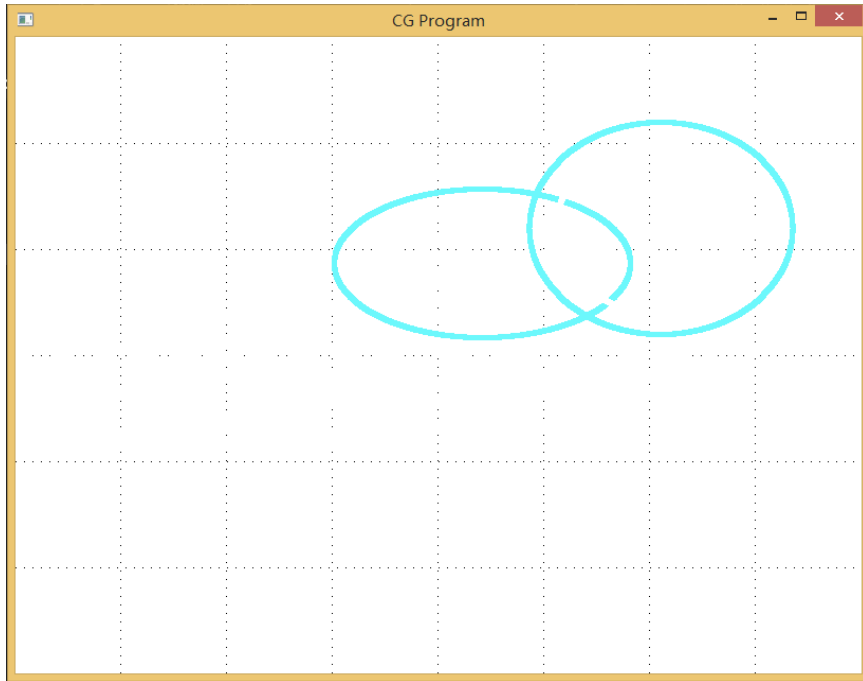
设置线宽：



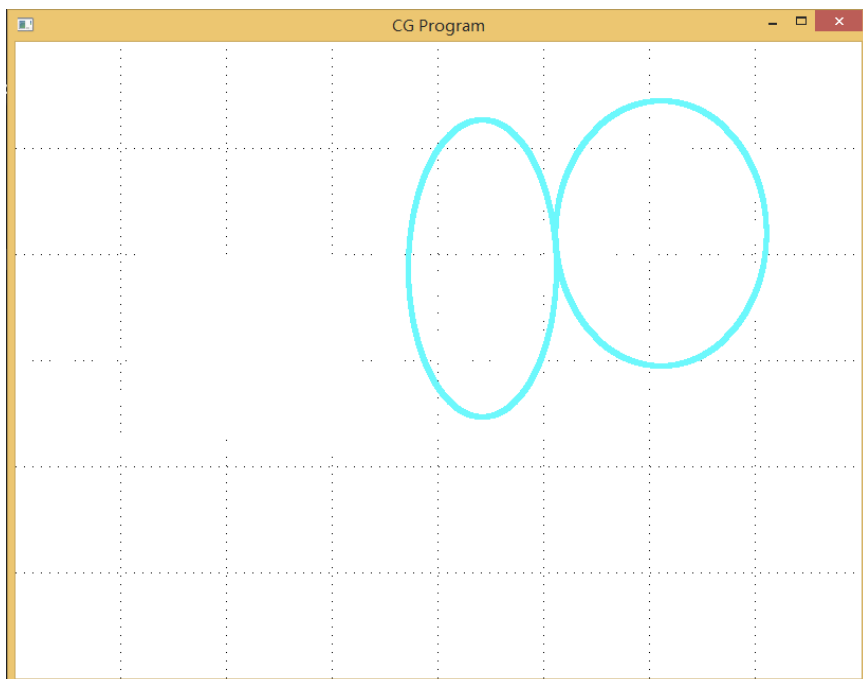
移动椭圆：



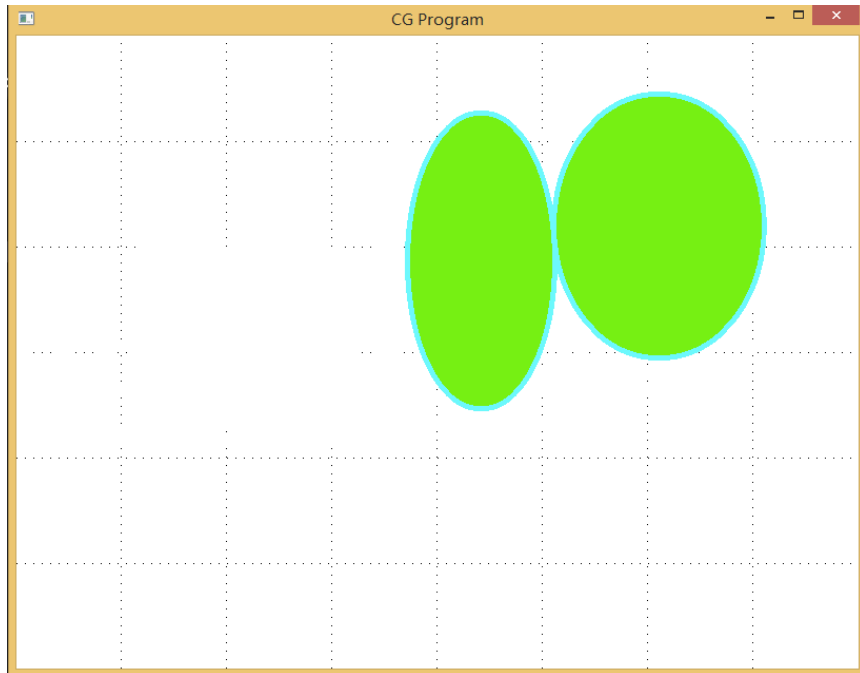
缩放椭圆：



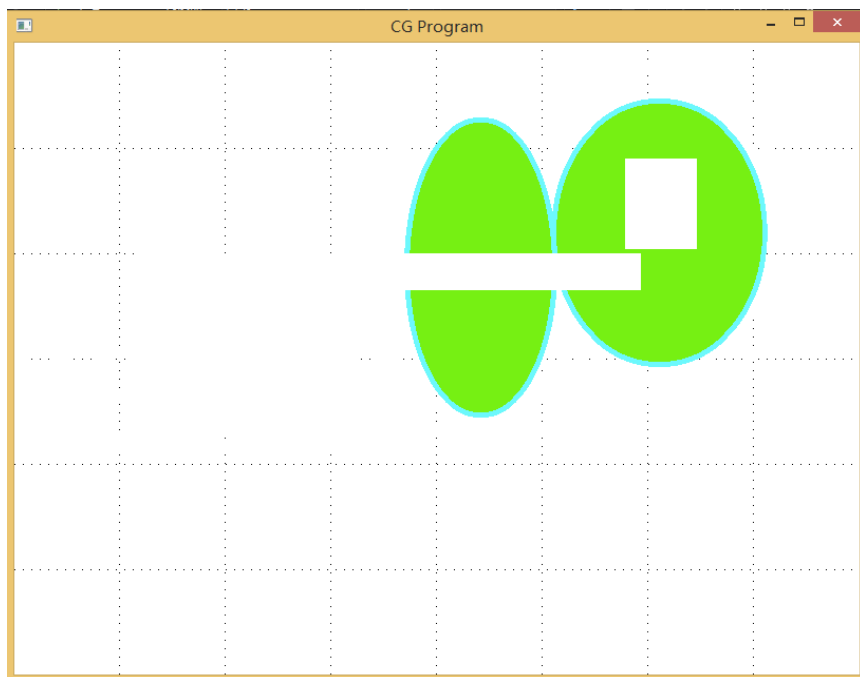
旋转椭圆：



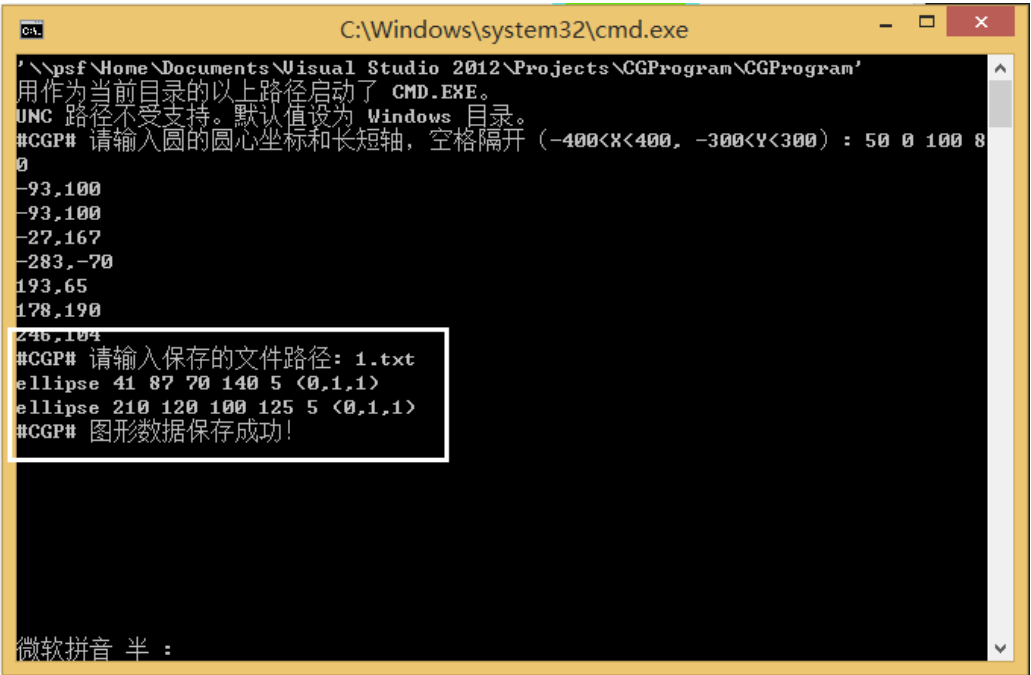
填充:



裁剪:



保存数据：



7、Curve模块：

(1) 模块简介：Curve是曲线的类定义，继承于Graphics类，包含控制顶点、曲线的折点、线宽、颜色等属性，以及线宽、颜色设置、绘制、缩放、移动、旋转、保存等功能。

(2) 模块示意图：

Curve
-Point[4]: COORD
-Bezier[11]: COORD
-lineWidth: float
-lineColor: COLOR
Curve()
Curve(int n, COORD c[])
~Curve()
getLineWidth() const: Float
setLineWidth(float p): void
getLineColor() const: COLOR
setLineColor(float r, float g, float b): void
calBezier(): void
DrawPoint(int n, COORD c[]): void
DrawLine(int n, COORD c[]): void
Draw(): void
Up(): void
Down(): void
Left(): void
Right(): void
Increase(): void
Decrease(): void
Rotate(char c): void
Fill(float r, float g, float b): void
Disappear(): void
SaveFile(): string

(3) 数据成员:

point是控制顶点的数组，目前只支持4个控制顶点的曲线。

Bezier数组保存曲线的折点，因为采用了Bezier曲线生成算法故取此名。

lineWidth为float类型，标示直线的线宽。

lineColor为COLOR类型，COLOR有r、g、b三个成员，标示点的颜色。

(4) 成员函数:

Curve()和Curve(int n, COORD c[])是曲线的构造函数，若无参数，则进行默认初始化，若有参数则根据c[]中的顶点坐标初始控制顶点。

~Curve(); 析构函数

float getLineWidth() const; 获取线宽

void setLineWidth(float p); 设置线宽

COLOR getLineColor() const; 获取线条颜色

void setLineColor(float r, float g, float b); 设置线条颜色

void Draw(); 绘制图形的函数

void Up(); 上移图形的函数

void Down(); 下移图形的函数

void Left(); 左移图形的函数

void Right(); 右移图形的函数

void Increase(); 放大图形

void Decrease(); 缩小图形

void Rotate(char c); 旋转图形

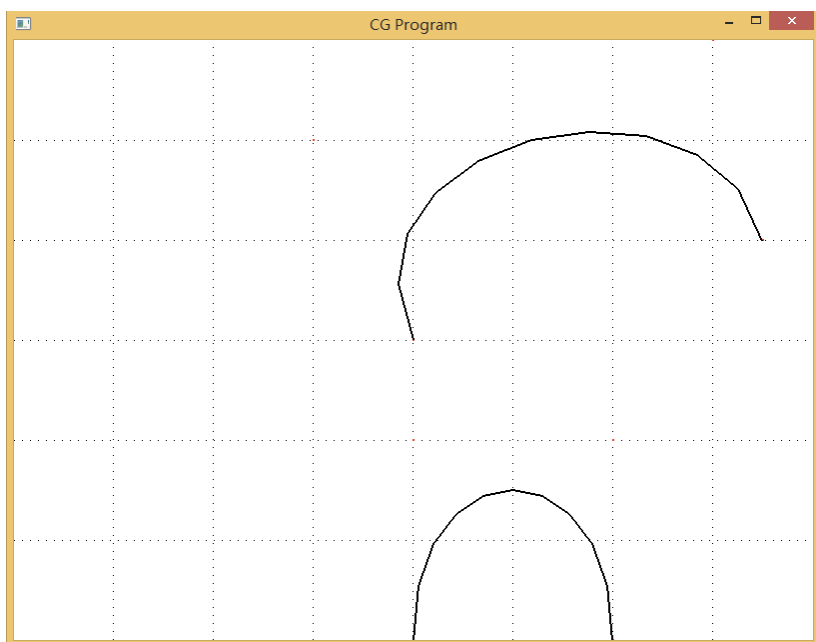
void Fill(float r, float g, float b); 填充图形

void Disappear(); 擦除图形

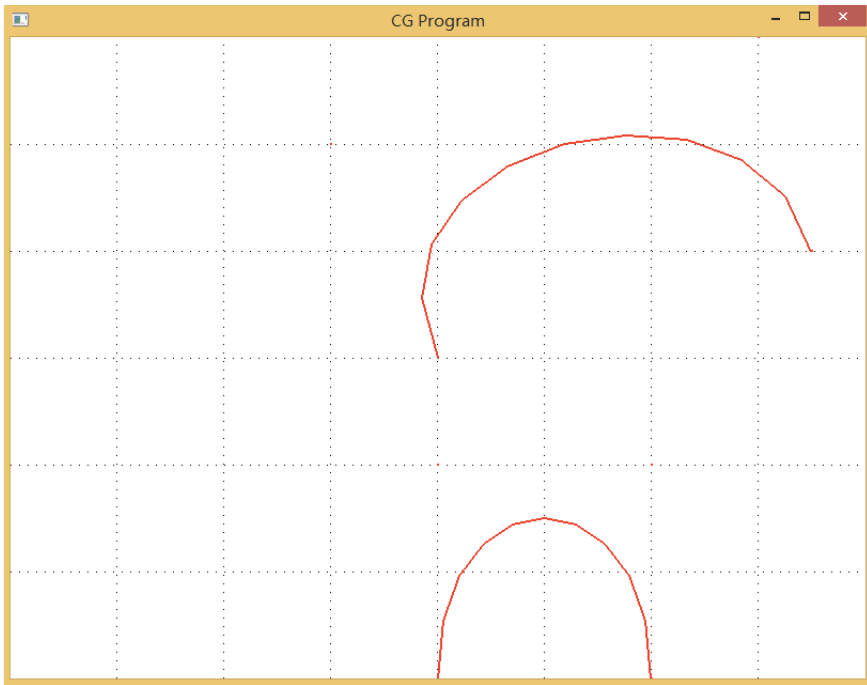
string SaveFile(); 保存图形数据

(5) 功能演示:

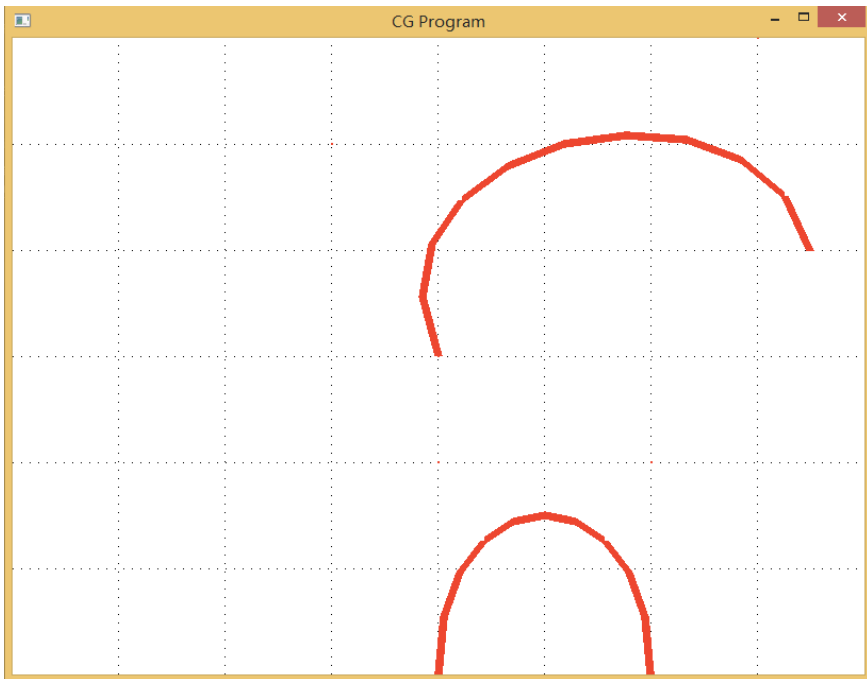
新建曲线:



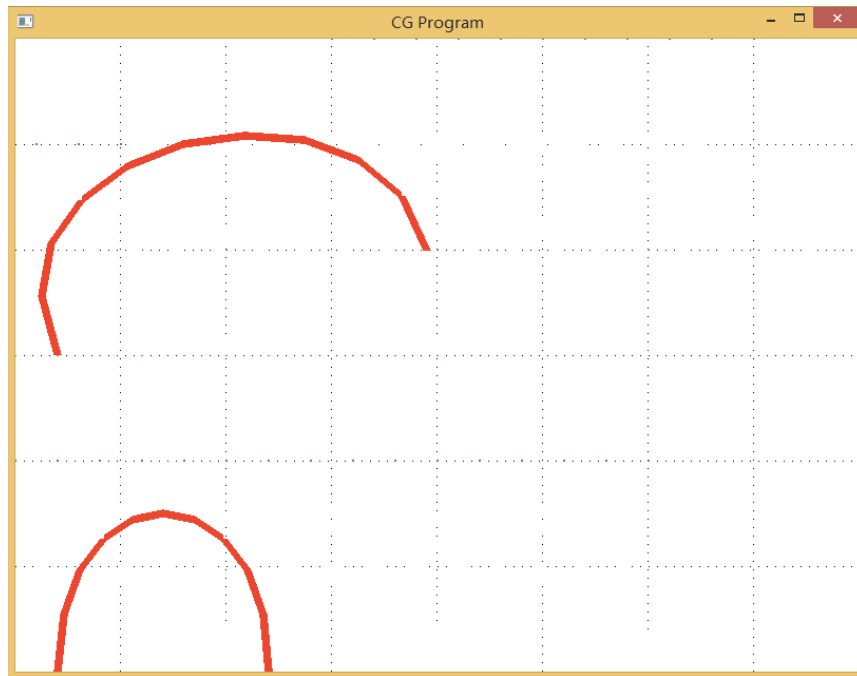
设置颜色:



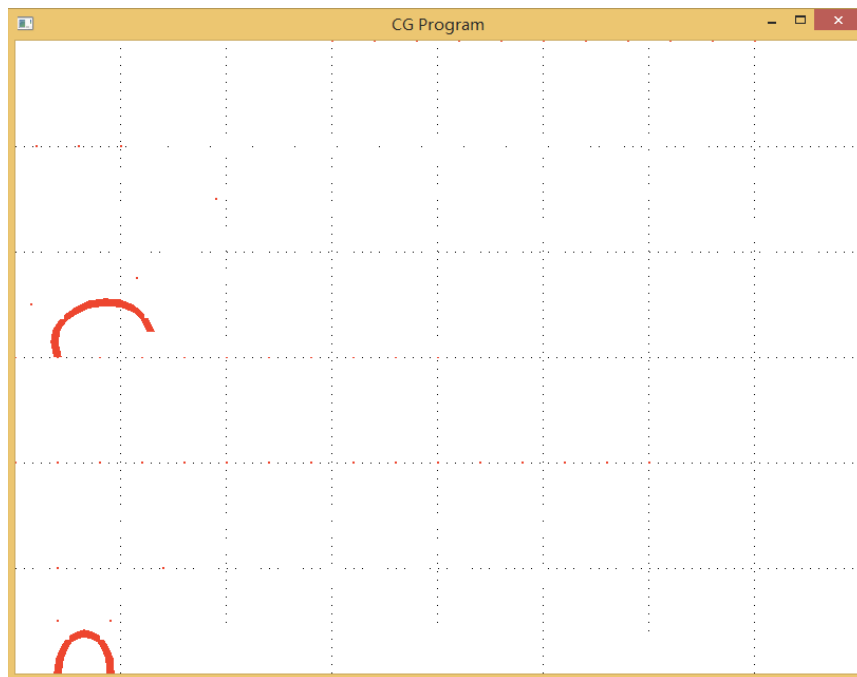
设置线宽:



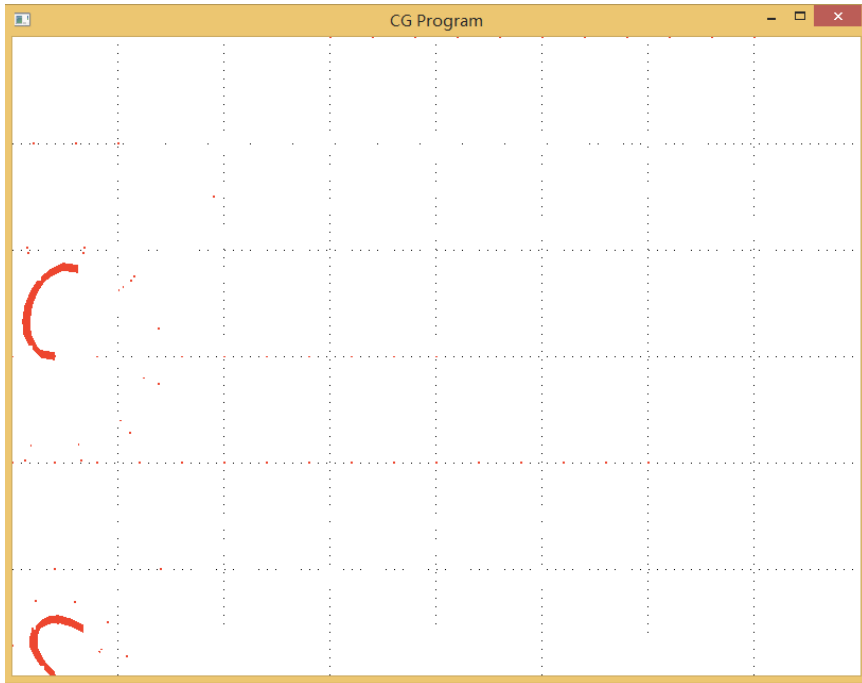
平移:



缩放:



旋转：



保存：

```
C:\Windows\system32\cmd.exe

'\\psf\Home\Documents\Visual Studio 2012\Projects\CGProgram\CGProgram'
用作为当前目录的以上路径启动了 CMD.EXE。
UNC 路径不受支持。默认值设为 Windows 目录。
-99,-107
-99,-107
-96,269
-96,269
#CGP# 请输入曲线的四个点坐标，空格隔开（-400<X<400，-300<Y<300）： 0 0 -100 200 3
00 300 350 100
169,140
169,140
-76,-78
-76,-78
-25,-64
-25,-64
52,92
52,92
-5,68
-5,68
#CGP# 请输入保存的文件路径： 1.txt
curve -360 -300 -400 -272 -378 -230 -334 -256 15 <1,0,0>
curve -360 0 -411 1 -385 97 -339 83 15 <1,0,0>
#CGP# 图形数据保存成功！

微软拼音 半：
```


8、Teapot模块：

(1) 模块简介：Curve是曲线的类定义，继承于Graphics类，包含控制顶点、曲线的折点、线宽、颜色等属性，以及线宽、颜色设置、绘制、缩放、移动、旋转、保存等功能。

(2) 模块示意图：

Teapot
~size: double ~lineColor: COLOR ~lineWidth: float ~posX: double ~posY: double ~posZ: double ~filled: bool
Teapot() Teapot(double s) ~Teapot() setLineWidth(float lw): void getLineWidth() const: float setLineColor(float r, float g, float b): void getLineColor() const: COLOR setPos(double x, double y, double z): void getPosX() const: double getPosY() const: double getPosZ() const: double setAngle(double x, double y): void setFilled(bool e): void getFilled() const: bool Draw(): void Up(): void Down(): void Left(): void Right(): void Increase(): void Decrease(): void Rotate(char c): void Fill(float r, float g, float b): void Disappear(): void SaveFile(): string

(3) 数据成员：

size是指茶壶的规格大小，在绘制时需要指明这个参数

lineWidth为float类型，标示直线的线宽。

lineColor为COLOR类型，COLOR有r、g、b三个成员，标示点的颜色。

posX、posY、posZ是茶壶的位置参数。

filled是bool类型的变量，用来指示茶壶是否需要被填充

(4) 成员函数：

Teapot()和Teapot(double s)是Teapot的两个构造函数，参数s用来初始化茶壶的规格大小
~Teapot(); 析构函数

void setPos(double x, double y, double z); 设置茶壶的位置

double getPosX() const; 获取茶壶位置的X方向的分量值

double getPosY() const;

double getPosZ() const;

void setAngle(double x, double y);

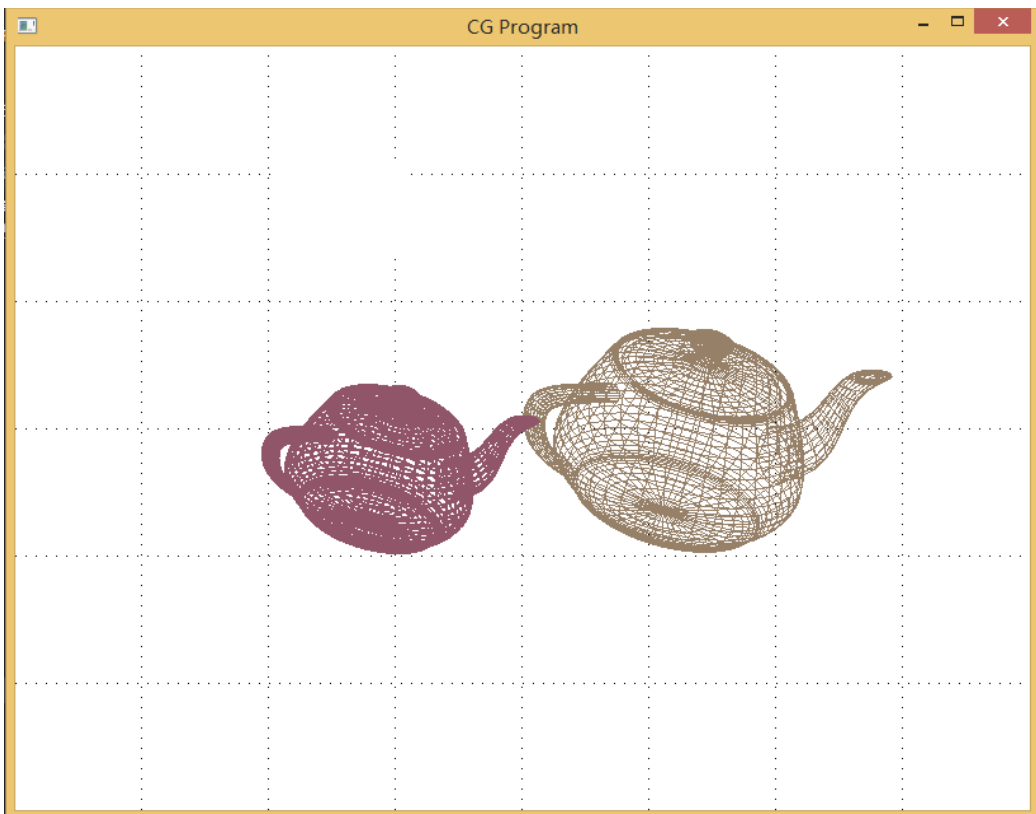
void setFilled(bool e); 对茶壶是否需要填充进行设置

bool getFilled() const;

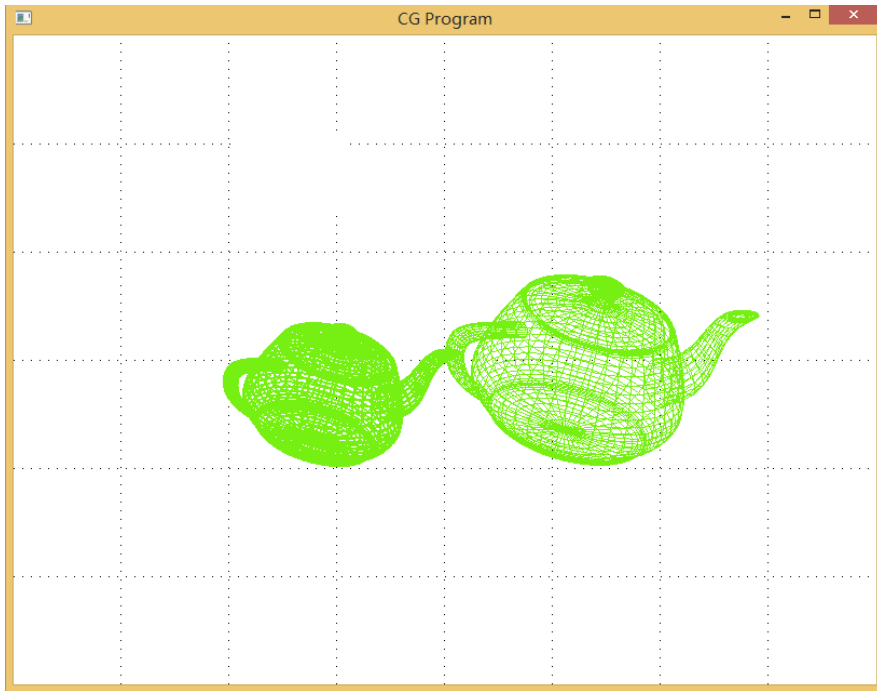
```
float getLineWidth() const;    获取线宽
void setLineWidth(float p);    设置线宽
COLOR getLineColor() const;    获取线条颜色
void setLineColor(float r, float g, float b);    设置线条颜色
void Draw();    绘制图形的函数
void Up();    上移图形的函数
void Down();    下移图形的函数
void Left();    左移图形的函数
void Right();    右移图形的函数
void Increase();    放大图形
void Decrease();    缩小图形
void Rotate(char c);    旋转图形
void Fill(float r, float g, float b);    填充图形
void Disappear();    擦除图形
string SaveFile();    保存图形数据
```

(5) 功能演示:

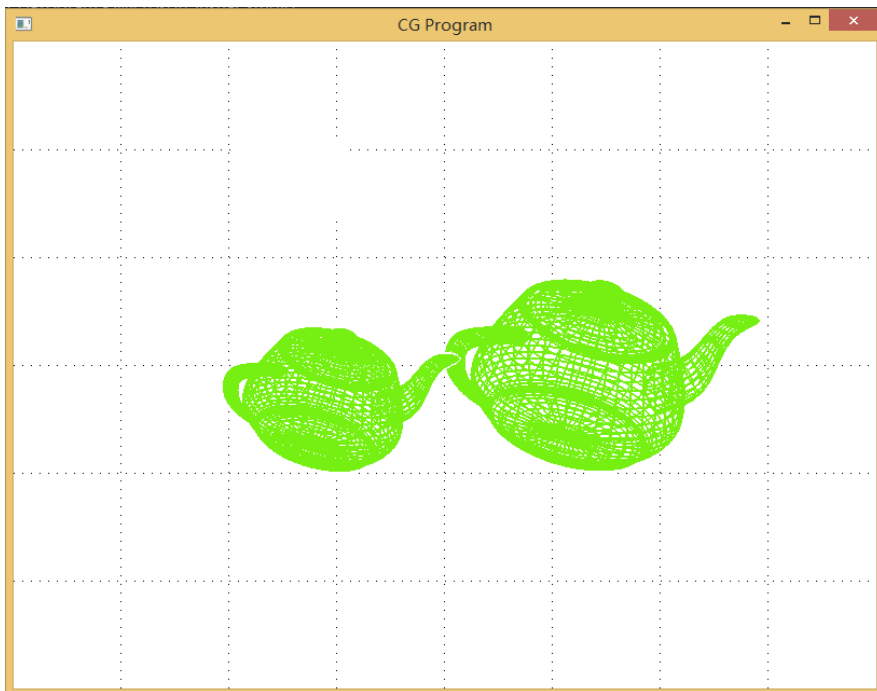
新建茶壶:



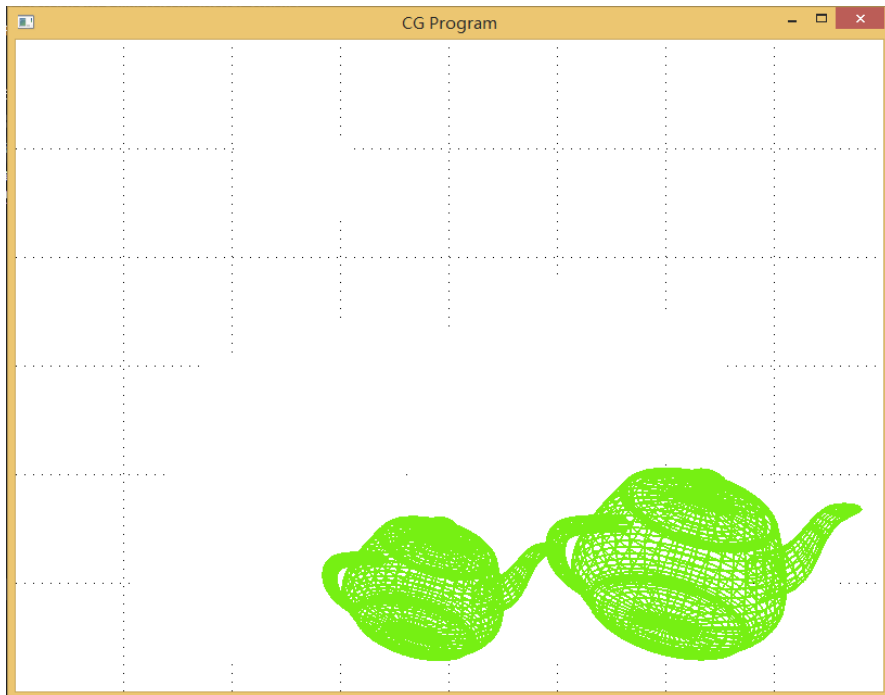
设置颜色：



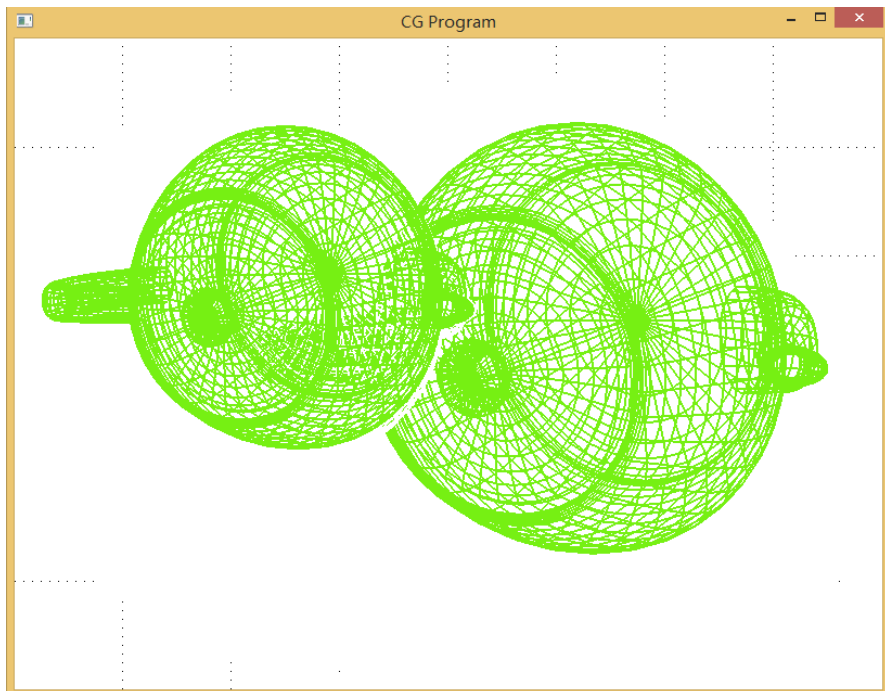
设置线宽：



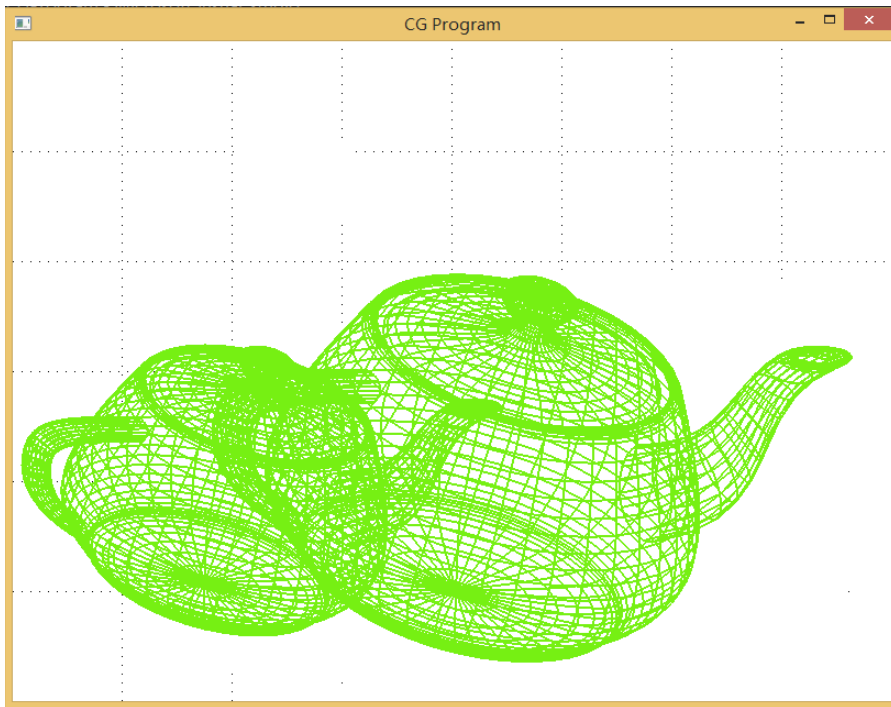
平移:



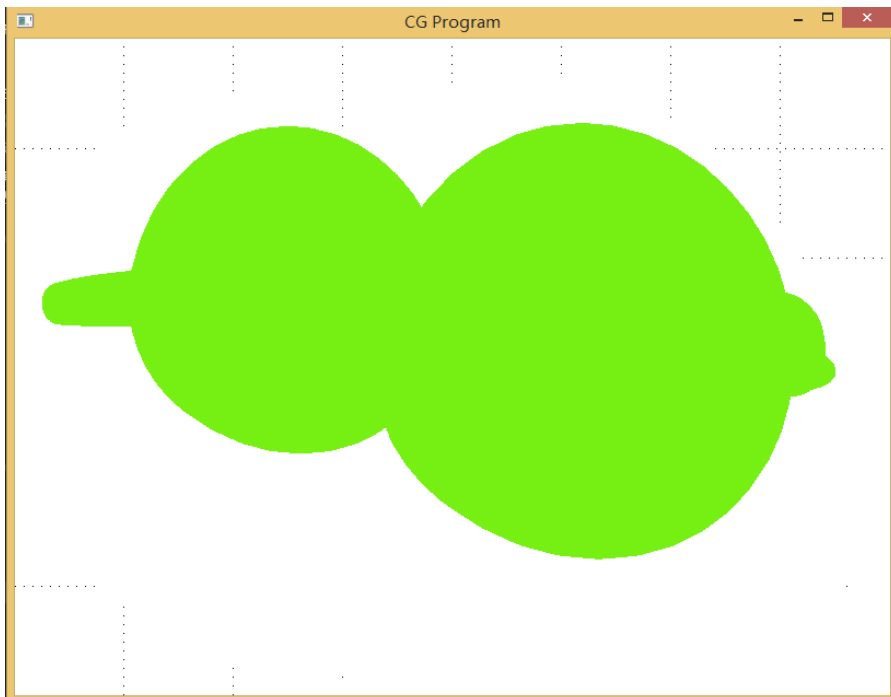
缩放:



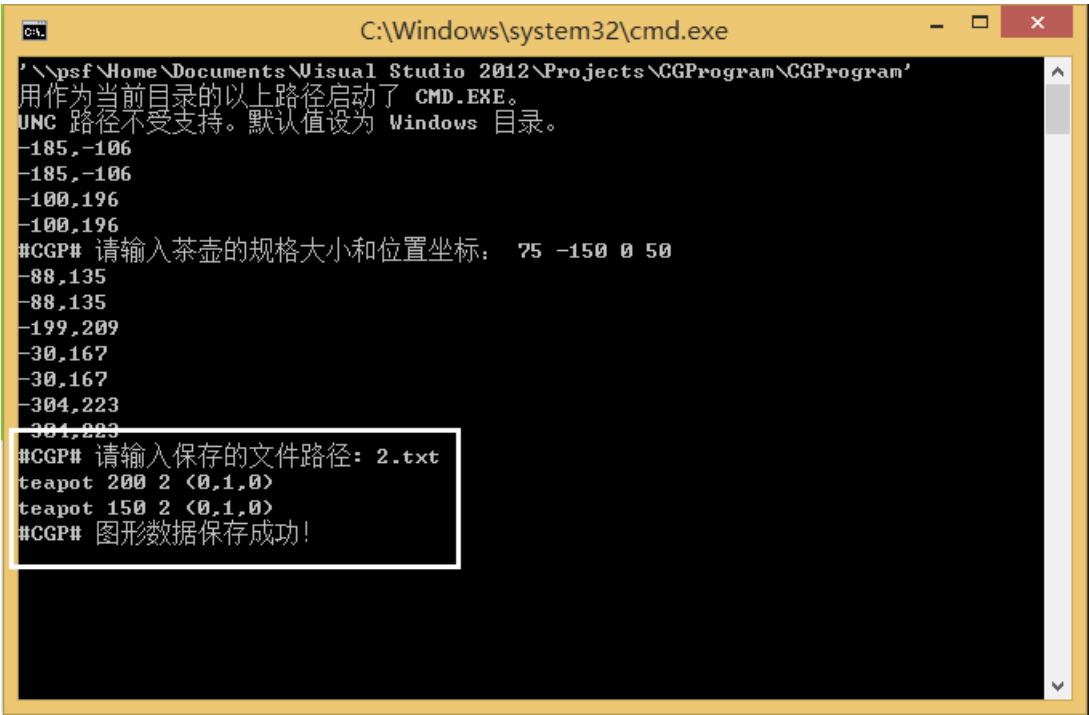
旋转：



填充：



保存数据:



9、管理模块:

(1) 模块简介: 管理模块有Define和MainCtrl两部分, 一部分是整个系统的宏定义部分, 另一部分是对图形数据的控制管理部分。

(2) 模块示意图:

Define
Menu No
Color Type
rand
DrawLine

MainCtrl
-GraphList: vector<Graphics*> -GraphData: vector<string> -c[2]: COORD
MainCtrl(); MainCtrl(string filename): void ReadFromFile(string filename); ~MainCtrl(); SetLineColor(float r, float g, float b): void SetLineWidth(float lw): void DrawGraph(): void UpGraph(): void DownGraph(): void LeftGraph(): void RightGraph(): void Increase(): void Decrease(): void Rotate(char c): void Fill(float r, float g, float b): void CutGraph(): void Disappear(): void SaveToFile(string filename): void str2float(string s): float str2COLOR(string s): COLOR splitStr(string line, string flag): vector<string>

(3) 数据成员:

(Define部分)

MenuNo是管理弹出菜单的宏定义，对每个菜单定义变量号，从而方便processMenuEvent()中对相应菜单事件的处理

ColorType是COLOR结构体的定义，为了方便对图形颜色的设置而定义的。

rand是一个随机函数，用来随机生成数字构造图形的位置坐标。

DrawLine是一个画直线的宏定义，将直线的两端点坐标传入方便用户理解。

(MainCtrl部分)

GraphList是一个存储图形数据的链表，新建的图形均保存在这个数据结构中。

GraphData是保存读入的图形数据字符串，方便程度对其切割解析。

c[]数组上面提到过，用于存储裁剪矩形的起点和终点坐标。

(4) 成员函数:

MainCtrl()和 MainCtrl(string filename)是构造函数，若无参数则用已有的图形数据初始化GraphList，若有参数则参数为filename，从文件中读入图形数据初始化。

void ReadFromFile(string filename); 具体的文件读取图形数据函数

~MainCtrl(); 析构函数

void SetLineColor(float r, float g, float b); 管理链表种图形的颜色

void SetLineWidth(float lw); 管理链表种图形的线宽

void DrawGraph(); 绘制链表中的图形

void UpGraph(); 上移图形

void DownGraph(); 下移图形

void LeftGraph(); 左移

void RightGraph(); 右移

<code>void Increase();</code>	放大链表中的所有图形
<code>void Decrease()</code>	缩小链表中的图形
<code>void Rotate(char c);</code>	对图形进行旋转
<code>void Fill(float r, float g, float b);</code>	对链表中的图形填充特定的颜色
<code>void CutGraph();</code>	裁剪图形
<code>void Disappear();</code>	清空画板上已经绘制的所有图形
<code>void SaveToFile(string filename);</code>	保存图形数据到文件
<code>float str2float(string s);</code>	读入图形数据后要对数据字符串解析
<code>COLOR str2COLOR(string s);</code>	解析数据字符串时用
<code>vector<string> splitStr(string line, string flag);</code>	用来切割从文件中读入的字符串

10、交互模块：

(1) 模块简介：交互模块有鼠标交互和键盘交互两部分。通过鼠标的右键弹出命令菜单，通过左键的按下和拖动动作画裁剪矩形进行图形的裁剪；键盘的交互主要有两处，通过上下左右键移动图形；通过asdw旋转图形；另外通过r、e两个键单独旋转茶壶方便观察效果（e为停止）。

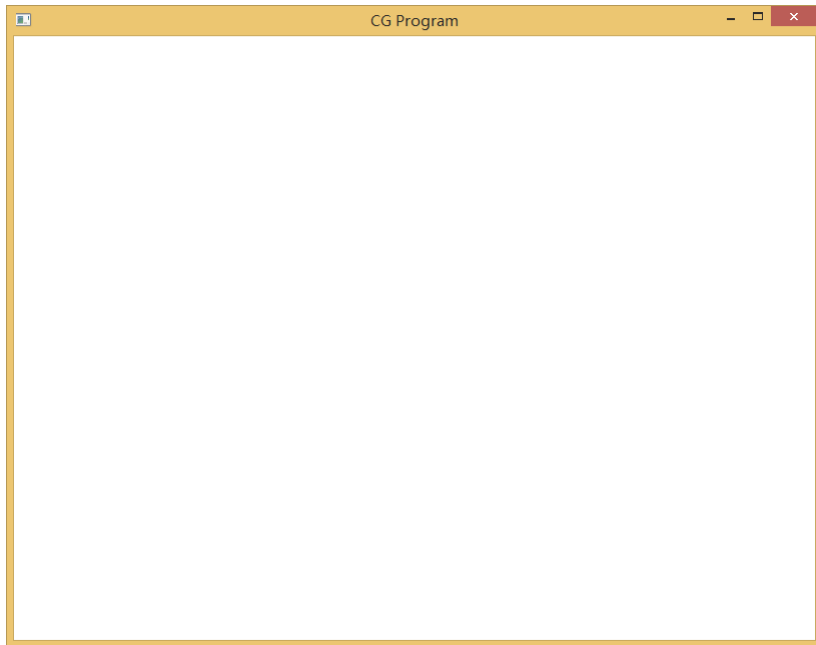
(2) 功能展示：

鼠标点击右键的效果：

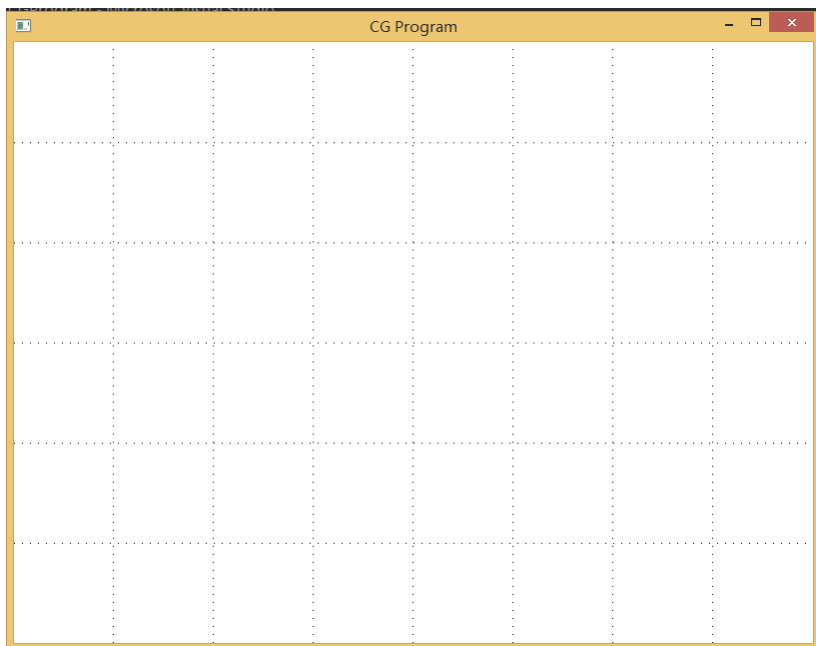


四、系统操作说明：

- 1、运行程序，会弹出CG Program的画板，如下图所示：



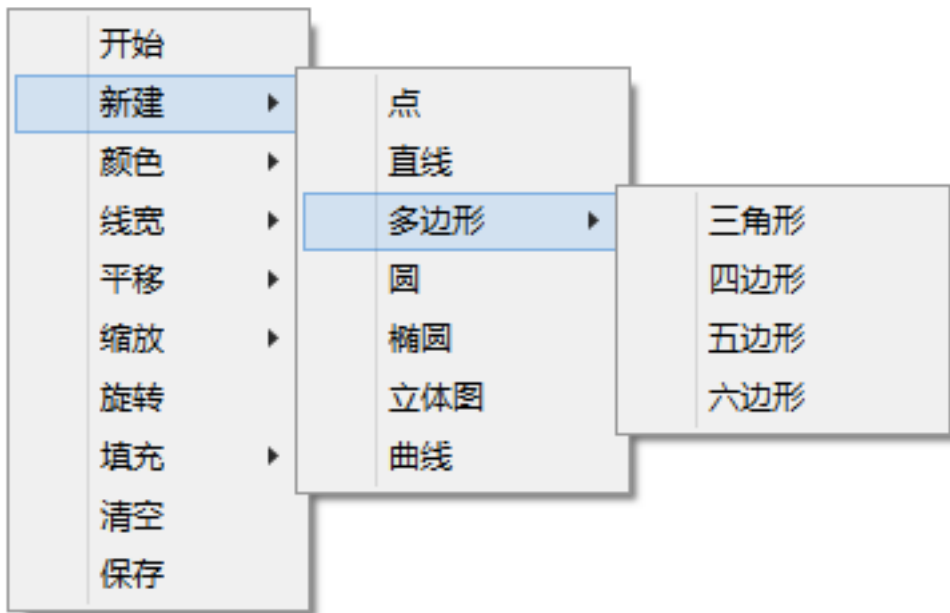
- 2、点击鼠标左键，面板上显示网格线，方便用户观察坐标。



3、点击鼠标右键，会弹出命令菜单，如交互模块的功能展示所示的菜单。

4、菜单操作——开始：绘制已经初始化的图形；

5、菜单的新建操作：新建图形；新建是一个子菜单，在下属菜单中选择具体的图形种类，如下图。



(1) 新建点：点击“新建——点”，然后在控制台窗口根据文字提示输入相应的图形数据，要求的格式为“点的X坐标（空格）点的Y坐标”，例如：100 200。（注意，坐标需满足 $-400 < X < 400$ ， $-300 < Y < 300$ 以方便观察）

(2) 新建直线：点击“新建——直线”，然后在控制台窗口根据文字提示输入相应的图形数据，要求的格式为“起点的X坐标（空格）起点的Y坐标（空格）终点的X坐标（空格）终点的Y坐标”，例如：100 200 200 200。（注意，坐标需满足 $-400 \leq X \leq 400$ ， $-300 \leq Y \leq 300$ 以方便观察）

(3) 新建多边形：点击“新建——多边形——具体的多边形种类”，然后在控制台窗口根据文字提示输入各点的横纵坐标，要求的格式为“点的X坐标（空格）点的Y坐标……”，例如选择新建四边形，就应该输入：100 200 200 200 100 100 100。（注意，坐标需满足 $-400 \leq X \leq 400$ ， $-300 \leq Y \leq 300$ 以方便观察）

(4) 新建圆：点击“新建——圆”，然后在控制台窗口根据文字提示输入圆的圆心坐标和半径，要求的格式为“圆心的X坐标（空格）圆心的Y坐标（空格）半径长度”，例如输入：100 200 20。（注意，坐标需满足 $-400 \leq X \leq 400$ ， $-300 \leq Y \leq 300$ 以方便观察）

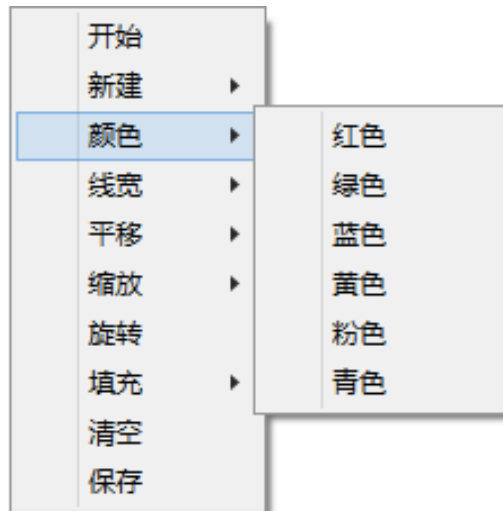
(5) 新建椭圆：点击“新建——椭圆”，然后在控制台窗口根据文字提示输入中心坐标和长短轴，要求的格式为“中心的X坐标（空格）中心的Y坐标（空格）长轴（空格）短轴”，例如：100 100 50 30。（注意，坐标需满足 $-400 \leq X \leq 400$ ， $-300 \leq Y \leq 300$ 以方便观察）

(6) 新建三维图形：点击“新建——立体图”，然后在控制台窗口根据文字提示输入响应图形信

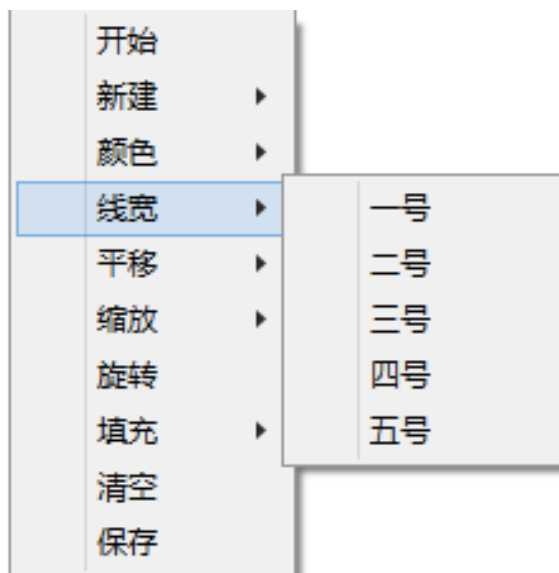
息，要求的格式为“图形的规格大小（空格）位置X分量（空格）位置Y分量（空格）位置Z分量”，例如：100 150 0 0。（注意，坐标需满足 $-400 \leq X \leq 400$ ， $-300 \leq Y \leq 300$ 以方便观察）

（7）新建曲线：点击“新建——曲线”，然后在控制台窗口根据文字提示输入曲线的四个控制顶点坐标，要求的格式为“点的X坐标（空格）点的Y坐标……”，目前只支持四个控制顶点。例如输入：100 200 200 300 350 250 300 100。（注意，坐标需满足 $-400 \leq X \leq 400$ ， $-300 \leq Y \leq 300$ ）

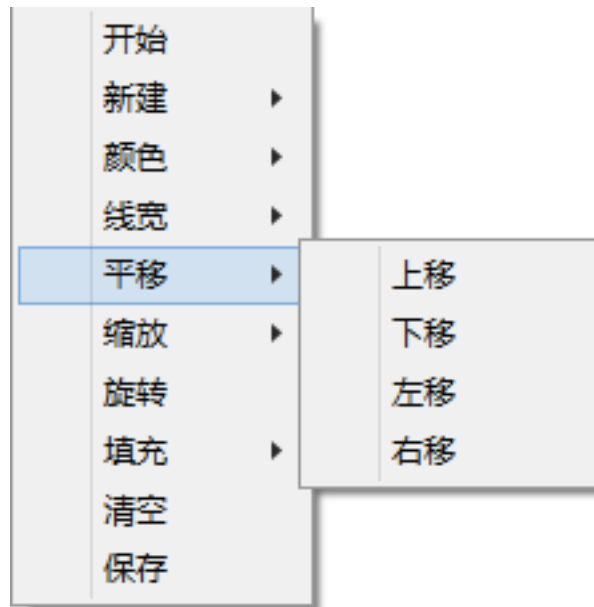
6、菜单的颜色操作：对已经初始化的图形进行颜色设置，点击“颜色——具体的颜色”即可设置，颜色的选项见下图：



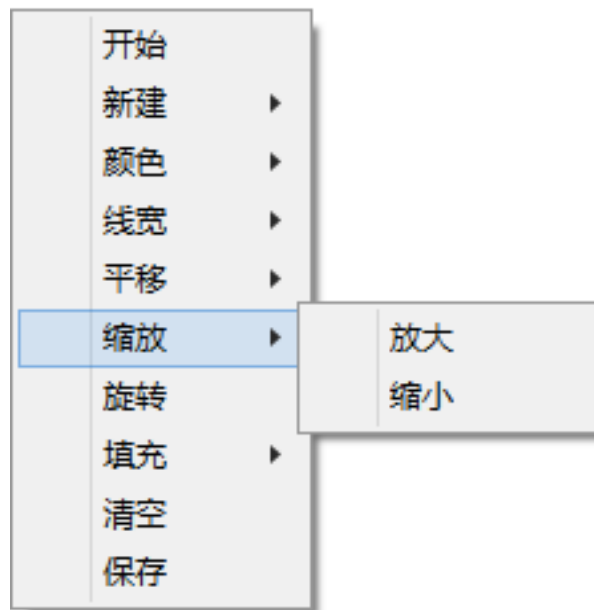
7、菜单的线宽操作：对画板上的图形进行线条宽度的设置，点击“线宽——具体的线宽型号”即可设置，线宽的型号选项见下图：



8、菜单的平移操作：对面板上的图形进行平移操作，可以朝上下左右四个方向平移，菜单的截图如下：

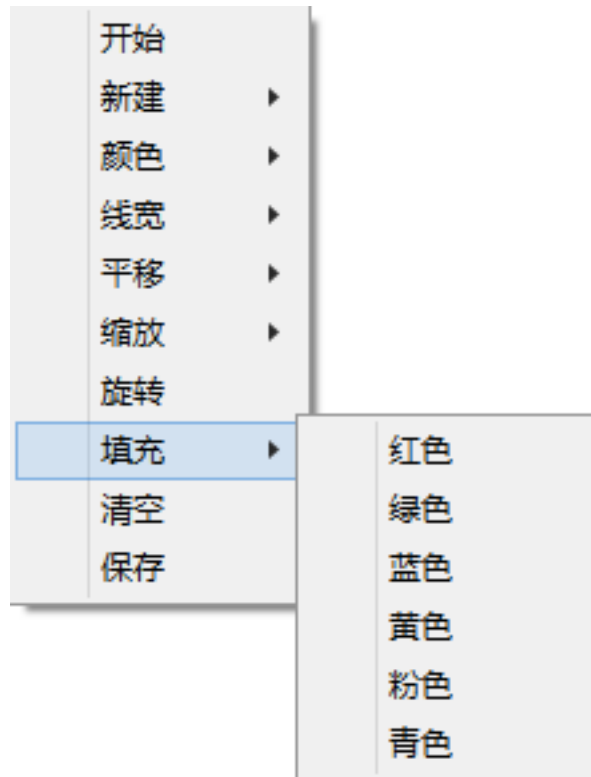


9、菜单的缩放操作：对面板上的图形进行缩放，可选择放大还是缩小，点击“缩放——放大or缩小”即可，菜单的截图如下：



10、菜单的旋转操作：对面板上的图形旋转（旋转只针对于直线、多边形、椭圆、曲线、三维图）。点击“旋转”即实现上述图形的一次旋转。

11、菜单的填充操作：对面板上的多边形、圆、椭圆、三维图填充具体的颜色，点击“填充——具体的颜色”即可。颜色选项见下图：

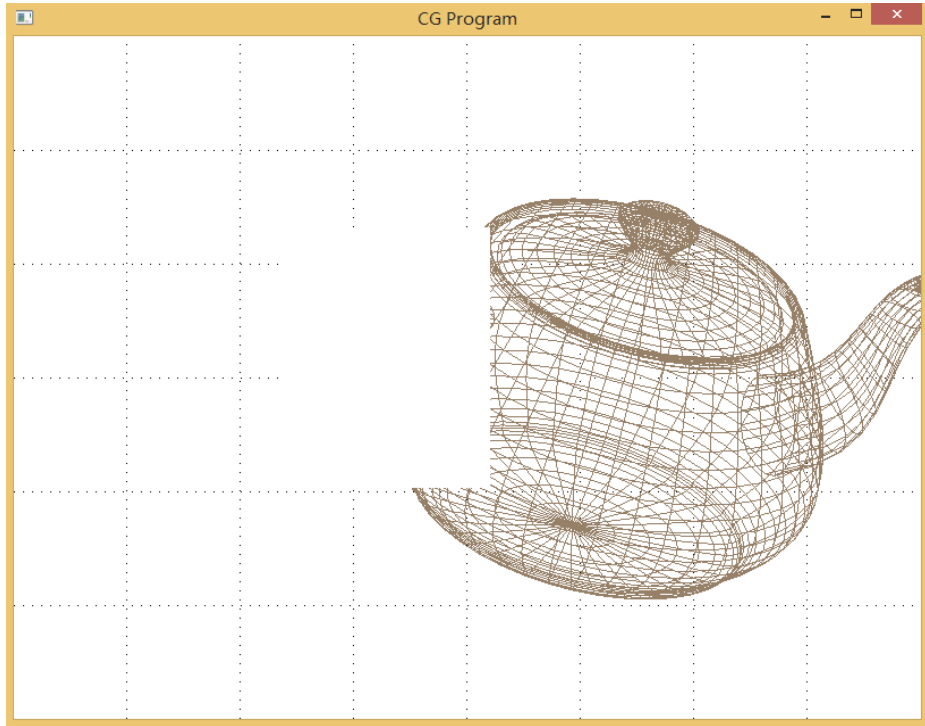


12、菜单的清空操作：清楚画板上绘制的所有图形，并重新绘制网格线，还原最初的状态。

13、菜单的保存操作：保存GraphList的所有图形。点击“保存”选项后，在控制台窗口根据文字提示输入保存的文件路径，例如：1.txt。

```
#CGP# 请输入保存的文件路径: 1.txt
teapot 100 150 0 0 1 <0.6,0.5,0.4>
#CGP# 图形数据保存成功!
```

14、鼠标左键按下并拖动：用于实现图形的裁剪功能。鼠标左键按下到拖动所形成的矩形区域是裁剪矩形，在矩形区域内部的图形部分会被擦除。



15、键盘的上下左右方向键：分别实现图形的上下左右移动，与右键菜单中的移动字菜单功能相同。

16、键盘的w、a、s、d键：控制图形沿不同的方向旋转，尤其以三维图形的旋转效果最为明显。

17、键盘的r键：按下该键后，面板上的三维图形会保持旋转状态一直旋转。

18、键盘的e键：停止r键实现的旋转动作。

结束语

本文以真实形象的语言解释了CG Program的使用方法，并配以大量的操作截图和效果展示方便用户的理解和使用。加快了助教或者其他编程爱好者对此项目的认知和理解。自此，图形学大作业就要告一段落，这学期的这份大作业让我受益匪浅，既理解了计算机图形学领域中的基础知识，也让我有机会以C++练手，充分掌握C++的面向对象程序设计，增强自己的编程能力。若系统使用说明书有疏漏和不完善之处，敬请包涵；另对本项目说明书一级整个项目有任何建议和疑问，欢迎联系笔者，电子邮箱为hao_silence@qq.com。

致谢 在此,笔者向对《计算机图形学》课程教学的孙教授表示衷心感谢,对负责此项作业的两位助教表示感谢,以及对像这次大作业提供帮助和支持的同学、博主表示感谢。

References

无