**第一章：静态绘制(三)**

**转置、缩放与混合通道**

这篇文章主要介绍图元旋转、缩放与混合通道变换。

[转置变换]

图元的转置变换极其简单

使用setAngle(float angle);或setTheta(float theta)即可设置角度变换。该变换满足的是左手定则，以Y轴负向为0点，进行顺时针角度设置。

Angle使用角度制、Theta使用弧度制。

例子：

UESI UES = new R();

Obj rect = UES.creatObj(UESI.BGIndex);

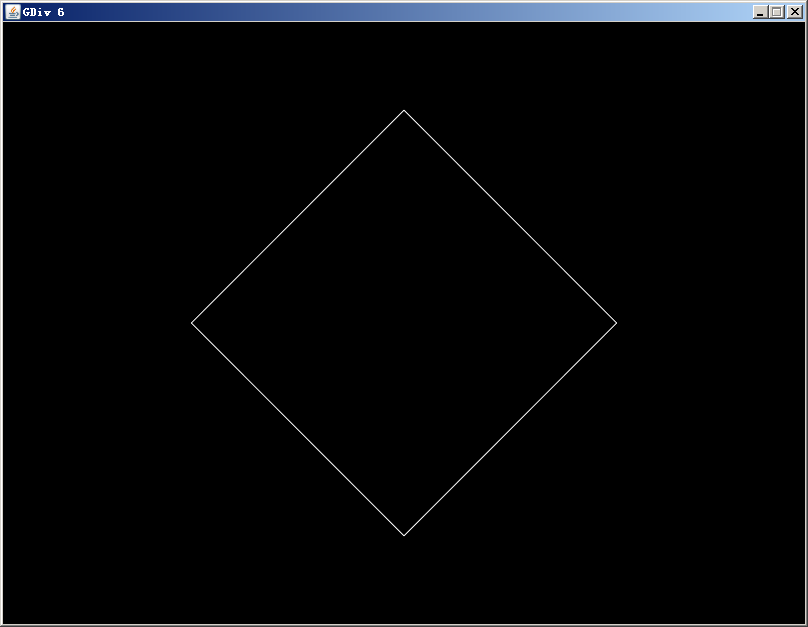
rect.addGLRect("FFFFFF",0,0,300f,300f);

rect.setPosition(CANExPos.POS\_CENTER);

rect.setAngle(45);//等同于rect.setTheta((float)Math.PI/4);

rect.show();

结果：



同样，使用getAngle、getTheta可以获得角度变换量，它们包含有图元索引重载。

[尺度变换（缩放）]

使用setWidth、setHeight可以进行尺度变换，get同上，包含有图元索引重载。

例子：

UESI UES = new R();

Obj rect = UES.creatObj(UESI.BGIndex);

rect.addGLRect("FFFFFF",0,0,300f,300f);

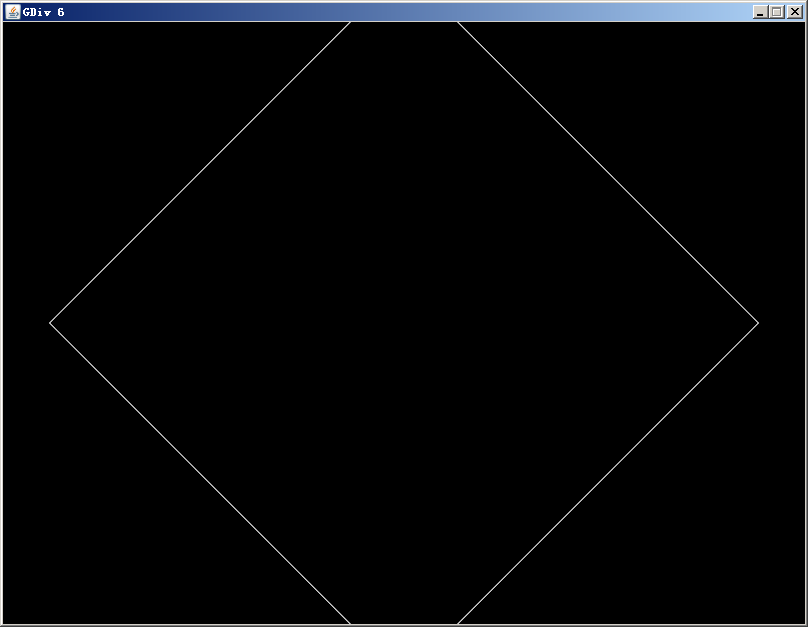
rect.setWidth(500);

rect.setHeight(500);

rect.show();

rect.setPosition(CANExPos.POS\_CENTER);

rect.setAngle(45);



[通道变换]

使用setAlph(float alpha);进行alph通道变换，alph为0到1的浮点数，越界时会自动截取。您暂时可以简单地将alph理解为透明度。其get方法同上，具备图元索引重载。

你也可以使用setAllAlph将整个图元的通道系数改为指定值。

例子：

UESI UES = new R();

Obj rect = UES.creatObj(UESI.BGIndex);

rect.addGLRect("FFFF00",0,0,300f,300f);

rect.setGLFill(true);

rect.setPosition(CANExPos.POS\_CENTER);

rect.setAngle(45);

rect.show();

Obj rect2 = UES.creatObj(UESI.UIIndex);

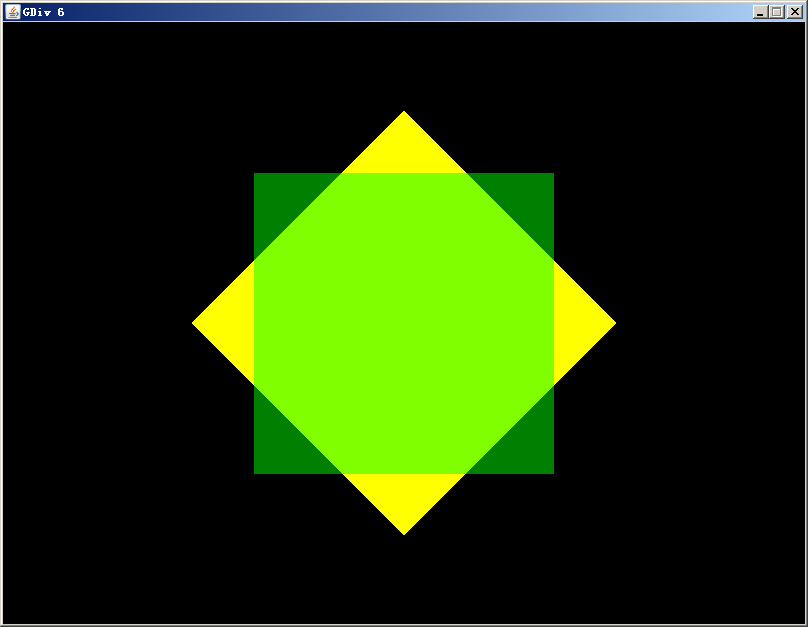
rect2.addGLRect("00FF00",0,0,300f,300f);

rect2.setGLFill(true);

rect2.setAlph(0.5f);//这里是绿色矩形，其“透明度”为0.5f，也就是“半”透明

rect2.setPosition(CANExPos.POS\_CENTER);

rect2.show();



[混合模式]

-关于alph的进阶理解：

这里是图形学中的概念而非引擎或OPENGL中的知识。这里笔者只能进行粗陋的讲解，具体内容可以搜索“混合模式”。

混合模式，通俗的讲就是两个颜色混合时得出结果颜色的算法。

这个算法的参数包括：两种颜色值、混合因子

其中，混合因子就是alph。

OPENGL常用的混合算法介绍:

1. GL.GL\_SRC\_ALPHA, GL.GL\_ONE\_MINUS\_SRC\_ALPHA:源色乘以alph系数，覆盖色乘1-原色的alph，之后加和得到目的色。也是这个引擎中除去渲染层（XR）采用的默认混色模式,它看起来就像是透明的效果。
2. GL.GL\_SRC\_ALPHA, GL.GL\_ONE：原色乘以alph，加上覆盖色得到目标色，覆盖层越多越接近1，因此，渲染层默认混色模式采用此种方式，在模拟火焰、光效、爆炸等特效时尤为有效。
3. GL.GL\_ONE\_MINUS\_DST\_COLOR, GL.GL\_ZERO：原色使用覆盖色的补色，覆盖色为0，此时，颜色为覆盖色反色，他可以实现类似底片的效果。

例子：上例稍加改动；

UESI UES = new R();

Obj rect = UES.creatObj(UESI.BGIndex);

rect.addGLRect("FFFF00",0,0,300f,300f);

rect.setGLFill(true);

rect.setPosition(CANExPos.POS\_CENTER);

rect.setAngle(45);

rect.show();

Obj rect2 = UES.creatObj(UESI.UIIndex);

rect2.addGLRect("FFFFFF",0,0,300f,300f);//将颜色改成了白色

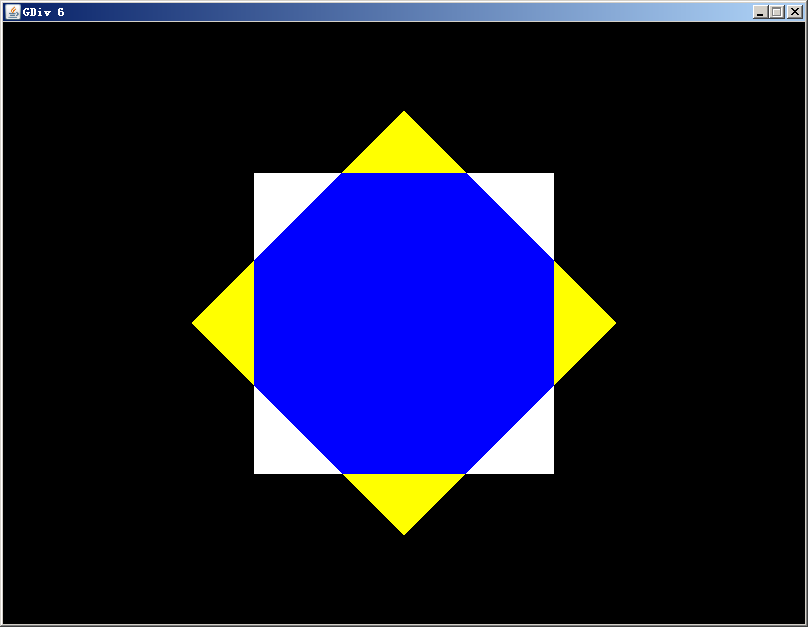
rect2.setGLFill(true);

rect2.setInverseColor(true);//注意这里！

rect2.setPosition(CANExPos.POS\_CENTER);

rect2.show();

结果：



与黄色重叠的部分变为了黄色的反色。

setInverseColor只是将混合模式设置成了GL.GL\_ONE\_MINUS\_DST\_COLOR, GL.GL\_ZERO；

你也可用通过：

setCustomMixMode(arg0, arg1)来设置自定义混合模式，并通过clearCustomMixModel()方法清除掉自定义混色模式。

[总结]

本节介绍了旋转、尺度变换、混合通道的设置、获取方法。我们能绘制的图形趋向于多样化了。