**第四章：图元详解（二）**

**实用方法**

这篇文章主要对图元中的实用方法进行一系列介绍。

[实用方法]

在图元的方法当中，除去变换图形的视觉样式之外，还提供了一系列实用方法，它们会在您的开发过程中提供额外支持。

[面积函数]

getSquare方法可以返回一个封闭多边形的面积。

例子：

public static void main(String[] args) {

UESI UES = new R();

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

rect.addGLRect("FFFFFF",0,0,30,50);

System.out.println(rect.getSquare());

}

控制台打印结果：1500.0

[碰撞函数]

PolygonCollide函数可以判断两个多边形是否发生了碰撞。

例子：

UESI UES = new R();

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

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

rect.setAngle(45);

rect.setPosition(CANExPos.POS\_CENTER);

rect.setDx(rect.getDx() + 200);

rect.show();

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

rect2.addGLRect("FFFF00",0, 0,300,300);

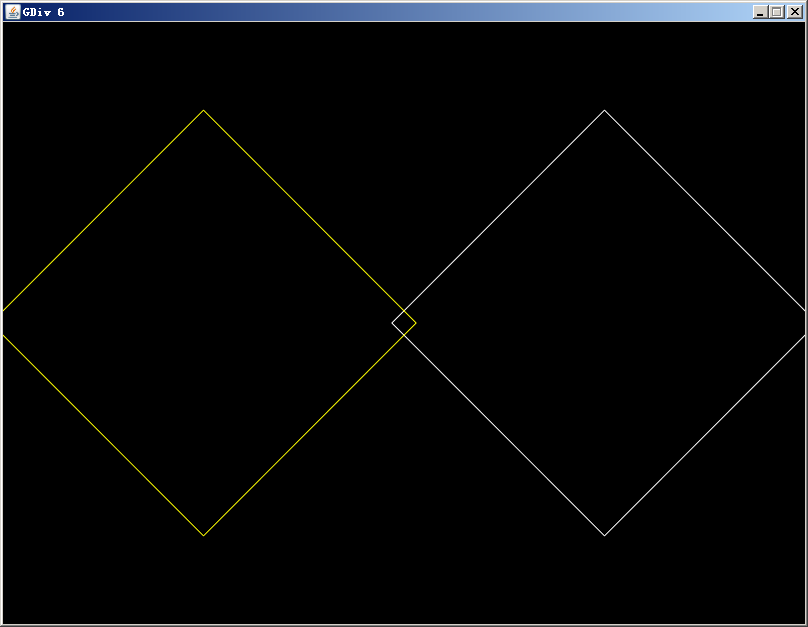
rect2.setAngle(45);

rect2.setPosition(CANExPos.POS\_CENTER);

rect2.setDx(rect2.getDx() - 200);

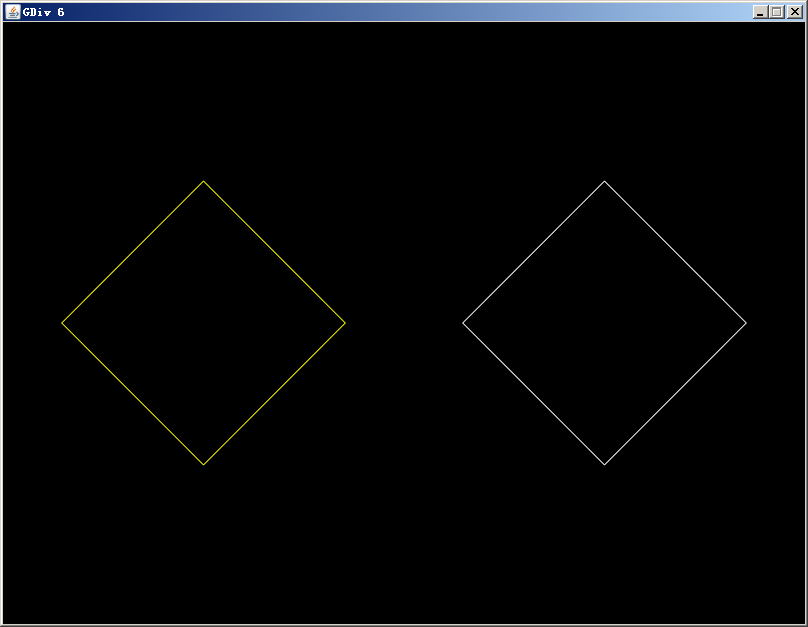
rect2.show();

System.out.println(rect.PolygonCollide(rect2));



打印结果：true

将上例中两个矩形长宽改为200：



打印结果：false

[距离相关性]

getRefDistance函数可返回两个图元的中心距离，将以false为结果的上例，加入如下语句：

rect.getRefDistance(rect2);

打印结果：

false

400.0

[角度相关性]

rect.getRefTheta函数可以返回另一个图元中心相对于本图元中心的弧度值，该值以Y负方向为0点，顺时针为正。

UESI UES = new R();

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

rect.addGLRect("FFFFFF",0,0,100,100);

rect.setPosition(CANExPos.POS\_CENTER);

rect.show();

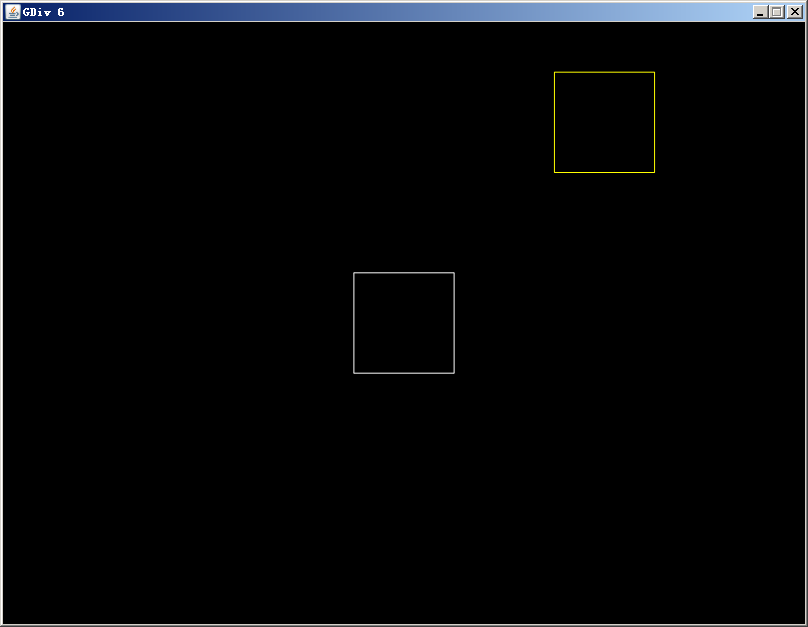
Obj rect2 = UES.creatObj(UESI.BGIndex);

rect2.addGLRect("FFFF00",0, 0,100,100);

rect2.setPosition(CANRefPos.RPO\_CORNER\_RIGHTTOP,rect,100f);

rect2.show();

System.out.println(rect.getRefTheta(rect2));



打印结果：0.7853982（约为PI/4）

[简单动画]

在某种场合下，我们仅需要进行一些简单的移动实现，为了这个简单的动画书写Serial似乎很不值得，因此有一组移动图元的方法：

moveCTo(Dx, Dy, allms)：将图元的中心移动到给定位置，最后一个参数allms代表经历的时间。

moveDTo(Dx, Dy, allms)：原理如上，是把图元的位置（不是中心）移动到指定值。

在实际的开发过程中，这两个方法用到的并不多，所以大部分情况下，实现动画效果还是老老实实地写Serial吧。