新建一个Cocoa Touch类然后继承自UIView，然后进行初始化

init(coder aDecoder: NSCoder ! ){

super.init(coder: aDecoder)

}

去掉 drawRect这个重写方法的注释

{

var context = UIGraphicsGetCurrentContext()

CGContextMoveToPoint(context, 100, 100) //移动画笔到点

CGContextAddLineToPoint(context, 100, 200)

CGContextStrokePath(context)

CGContextSetRGBStrokeColor(参数) //配置线条颜色

CGContextSetLineWidth(参数)//配置线条粗细

}

一些绘图API及功能说明:

var context = ….

1. 绘制矩形

CGContextAddRect(context,CGRect(参数))

CGContextFillPath(context)

用CGContextSetRGBFillColor()配置填充颜色

CGContextStrokeRect()添加边框

CGContextSetRGBStrokeColor()修改边框颜色

1. 绘制圆形,弧形,扇形

法一：

CGContextAddArc(context,参数) //添加一个弧,3.14是半圆

CGContextStrokePath(context)

法二：

CGContextAddEllipseInRect(context,CGRect(参数))

CGContextStrokePath(context)

1. 绘制图片

init(coder aDecoder: NSCoder ! ){

super.init(coder: aDecoder)

uiImage = UIImage(named: “图片名.jpg”).CGImage

}

var uiImage:CGImageRef?

drawRect方法中

CGContextSaveGState(context) //保存状态，否则会对后续图形有影响

CGContextTranslateCTM(context,10,400) //移动图片

CGContextScaleCTM(context, 1, -1)

CGContextDrawImage(context, CGRect(参数),uiImage)

//因为CGRect的坐标和CGImage的坐标是反着的，所以Y轴坐标变负即可

CGContextRestoreGState(context) //恢复原来状态

1. 创建简易画布

var path = CGPathCreateMutable()

override func touchBegan(…){ //点击时设定起始点

var p = touches.anyObject().locationInView(self)

//获取触摸点坐标

CGPathMoveToPoint(path, nil, p.x, p.y)

}

override func touchMoved(…){

var p = touches.anyObject().locationInView(self)

CGPathAddLineToPoint(path, nil, p.x, p.y)

setNeedsDisplay() //请求系统重绘

}

override func drawRect(…){

…

CGContextAddPath(context, path)

CGContextStrokePath(context)

}