Shapes 与 Masks

本章本节:不仅你要开始使用一个不同的示例项目,你也会处理多图层样式,创建图层动画，彼此互动和形状之间的变形动画。

Shapes将会用到CALayer的子类CAShapeLayer ，可以在屏幕上绘制不同的形状。

主要的属性：

• strokeColor：笔画颜色。

• strokeStart：笔画开始位置。

• strokeEnd：笔画结束位置。

• fillColor：图形填充颜色。

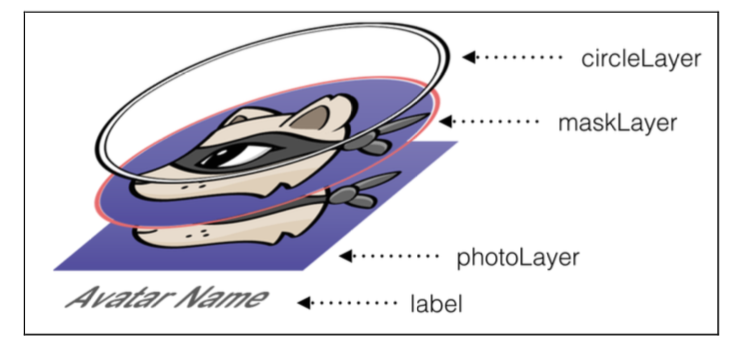
• lineWidth：笔画宽度，即笔画的粗细程度。

• lineDashPattern：虚线模式。

• path：图形的路径。

• lineCap：笔画未闭合位置的形状。

打开AvatarView。Swift,找到didMoveToWindow ，对比其中的属性



/// 实时渲染

///可以将自定义的代码实时渲染到Interface Builder中。而它们之间的桥梁就是通过两个指令来完成，即@IBDesignable和@IBInspectable。

///通过@IBDesignable告诉Interface Builder这个类可以实时渲染到界面中，但是这个类必须是UIView或者NSView的子类。

///通过@IBInspectable可以定义动态属性，即可在attribute inspector面板中可视化修改属性值。

计算图片动画位置以及形状，并在头像类中实现，运动方法，以供外界调用：

func searchForOpponent() { let avatarSize = myAvatar.frame.size let bounceXOffset: CGFloat = avatarSize.width/1.9 let morphSize = CGSize(

width: avatarSize.width \* 0.85,

height: avatarSize.height \* 1.1)

let rightBouncePoint = CGPoint( x: view.frame.size.width/2.0 + bounceXOffset, y: myAvatar.center.y)

let leftBouncePoint = CGPoint( x: view.frame.size.width/2.0 - bounceXOffset, y: myAvatar.center.y)

myAvatar.bounceOffPoint(rightBouncePoint, morphSize: morphSize) opponentAvatar.bounceOffPoint(leftBouncePoint, morphSize: morphSize)

}

添加形变动画

let morphAnimation = CABasicAnimation(keyPath: "path") morphAnimation.duration = animationDuration morphAnimation.toValue = UIBezierPath(

ovalInRect: morphedFrame).CGPath morphAnimation.timingFunction = CAMediaTimingFunction(

name: kCAMediaTimingFunctionEaseOut)

创建一个CABasicAnimation 并设置path属性，这个属性不同于Layer层的其他属性，你只可以设置toValue,Core Animation将会给你呈现中间动画。

设置持续时间，使用UIBezierPath创建一个椭圆型的曲线效果；UIBezierPath有便利的初始化方法，包括上边使用的UIBezierPath(

ovalInRect:).CGPath，使用CGRect 创建一个椭圆的曲线来适应原始的区域。

挑战

链接几秒后加载图片，

func foundOpponent() { status.text = "Connecting..."

opponentAvatar.image = UIImage(named: "avatar-2")

opponentAvatar.name = "Ray" }

最后变形为方块