Groups and Advanced Timing

之前学习了在Layer层添加倍数、独立的动画等，如果想要同步执行呢？那么就用到了Animation groups, 这个API简化了代码、确保可以同步执行动画。

删去

viewWillAppear():

loginButton.center.y += 30.0

loginButton.alpha = 0.0

删去

viewDidAppear():

UIView.animateWithDuration(0.5, delay: 0.5, usingSpringWithDamping: 0.5, initialSpringVelocity: 0, options: nil, animations: {

self.loginButton.center.y -= 30.0

self.loginButton.alpha = 1.0 }, completion: nil}

添加一个动画群组

let groupAnimation = CAAnimationGroup() groupAnimation.beginTime = CACurrentMediaTime() + 0.5 groupAnimation.duration = 0.5

groupAnimation.fillMode = kCAFillModeBackwards

CAAnimationGroup继承自CAAnimation，所以你可以使用之前熟悉的属性，比如beginTime/fillMode/delegate/removedOnCompletion;

在这里给login按钮添加缩放、旋转、淡入淡出的效果，

缩放

let scaleDown = CABasicAnimation(keyPath: "transform.scale")

scaleDown.fromValue = 3.5

scaleDown.toValue = 1.0

旋转

let rotate = CABasicAnimation(keyPath: "transform.rotation")

rotate.fromValue = CGFloat(M\_PI\_4)

rotate.toValue = 0.0

淡入淡出

let fade = CABasicAnimation(keyPath: "opacity") fade.fromValue = 0.0

fade.toValue = 1.0

然后放入群组中，最后添加到login.Layer层

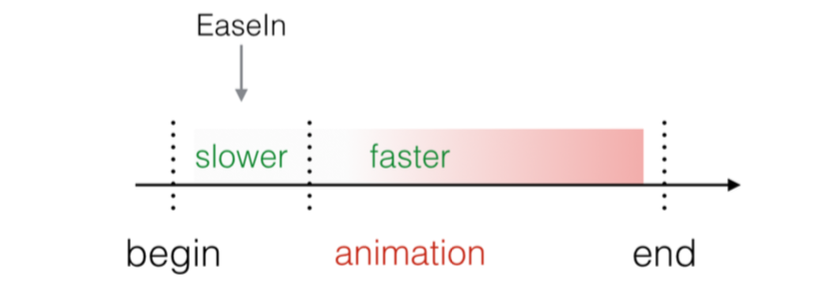
groupAnimation.animations = [scaleDown, rotate, fade] loginButton.layer.addAnimation(groupAnimation, forKey: nil)

处理UIKit动画是简单的，layer层动画在概念上是相同的,只有语法是不同的。

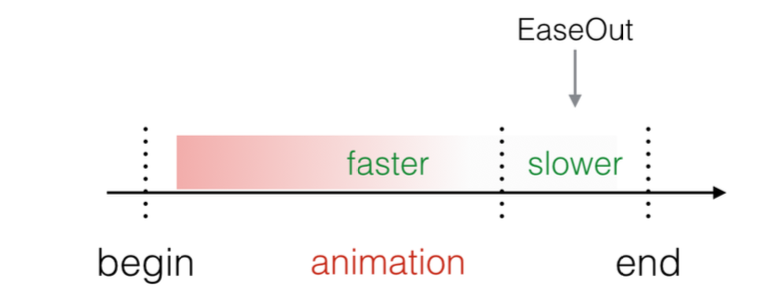
CAMediaTimingFunction 有一些默认的缓冲函数：

kCAMediaTimingFunctionLinear ：匀速执行动画；

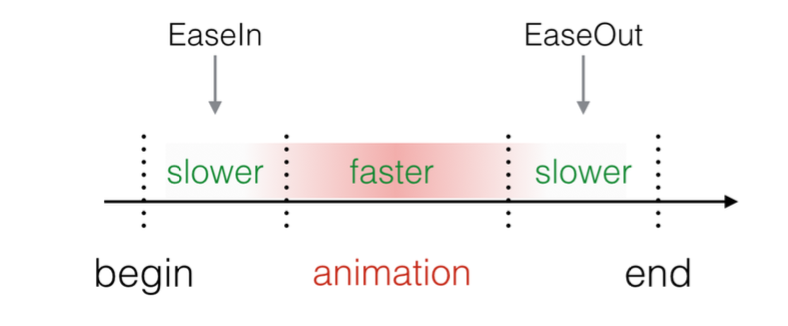
kCAMediaTimingFunctionEaseIn ：开始缓慢、结束时加速；



kCAMediaTimingFunctionEaseOut ：开始加速、结束时缓慢；



kCAMediaTimingFunctionEaseInEaseOut ：开始结束缓慢，中间加速：



重复动画

添加infolab从右至左到达指定位置的重复次数，

flyLeft.repeatCount = 4

如果你想设置总重复时间以秒为单位,而不是设置重复的数量,使用repeatDuration代替repeatCount属性。

这样的动画有些难受，没有流动感，可以这样做：

flyLeft.autoreverses = true;

流动感的动画实现了，但是最后一次是直接显示跳转到屏幕中心的，原因是当动画执行完毕后，最后的循环将label放置在屏幕之外，因为它是来回运动，那么将循环次数改为n.5，最后将恰好停止在指定位置。

flyLeft.repeatCount = 2.5

动画速度

尽管已经完成，但是速度有点慢，当然可以单独的控制某个视图的速度，通过设置speed，flyLeft.speed = 2.0

即使动画的持续时间为5秒，因为加一倍的速度，也就2.5秒执行完毕，可以这是动画本身的速度，当然也可以设置Layer层的速度；

info.layer.speed = 2.0

运行后发现，速度very faster,因为是2 \* 2 倍速度；

当然可以设置view.layer.speed = n;

玩动画速度,扭转和重复很有趣,但你的用户可能会不欣赏UI元素又开始在屏幕上!:]

综合运用：

1、详见BahamaAir-Completed