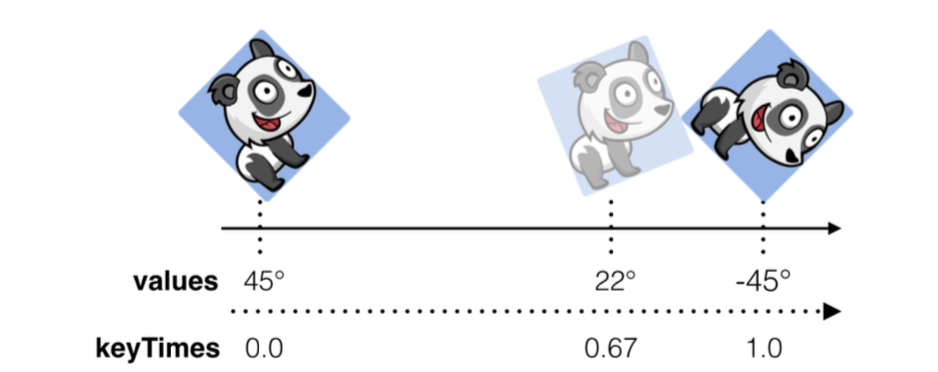
关键帧动画与结构体属性

Layer层的关键帧动画比View层的关键帧动画难一些；View层的关键帧动画是组合简单的动画在一起。CAKeyframeAnimation让你动画一个属性在给定层。可以定义不同的动画的关键点,但你不能有缺口或重叠在你的动画。尽管这听起来限制,但可以创建一些非常引人注目的与CAKeyframeAnimation效果。

通过fromValue 、 toValue ，Core Animation在规定的时间里逐步的更改Layer层的开始和结束的值；当你需要旋转Layer层从45°到-45°，你字需要更改这两个值，Layer自动渲染中的过程；

CAKeyframeAnimation 使用一个数组values;代替fromValue 、 toValue，元素的值代表动画的时间表，



1、创建一个Layer层的关键帧动画

let wobble = CAKeyframeAnimation(keyPath: "transform.rotation")

wobble.duration = 0.25

 wobble.repeatCount = 4

wobble.values = [0.0, -M\_PI\_4/4, 0.0, M\_PI\_4/4, 0.0] wobble.keyTimes = [0.0, 0.25, 0.5, 0.75, 1.0] heading.layer.addAnimation(wobble, forKey: nil)

类似于CABasicAnimation 的创建方式，其中

values数组是旋转Layer层从0°-- -45°--- 0°--- 45°；动画开始和结束是相同的数值；

keyTimes是确保开始到结束的时从0.0-1.0分别设置，避免其中的动画产生跳转。

2、设置动画结构体

Core Animation 是OC基于C的Framework，意味着结构体是有些不同的，OC中处理对象的API还是需要特殊的处理；这就是为什么执行Layer的属性相对简单，比如color/number；但是使用结构体是相对复杂的，比如CGPoint。

CALayer 层有好多属性都使用结构体，包括position类型的CGPoint，transform类型的CATransorm3，bound类型的CGRect，为了管理这些属性，Cocoa包含一个NSValue类，作为对象存取结构体的数值，就是NSValue.“boxes in” or “wraps” a struct value as an object, namely, NSValue.

NSValue提供了一系列的摆便利的初始化方法，

init!(CGPoint point: CGPoint)

init!(CGSize size: CGSize)

init!(CGRect rect: CGRect)

init!(CATransform3D t: CATransform3D)

举个例子：

let move = CABasicAnimation(keyPath: "position")

move.duration = 1.0

 move.fromValue = NSValue(CGPoint: CGPoint(x: 100.0, y: 100.0))

move.toValue = NSValue(CGPoint: CGPoint(x: 200.0, y: 200.0))

接下来就是添加关键帧和时间

flight.values = [

CGPoint(x: -50.0, y: 0.0),

 CGPoint(x: view.frame.width + 50.0, y: 160.0),

CGPoint(x: -50.0, y: loginButton.center.y)

].map { NSValue(CGPoint: $0) }

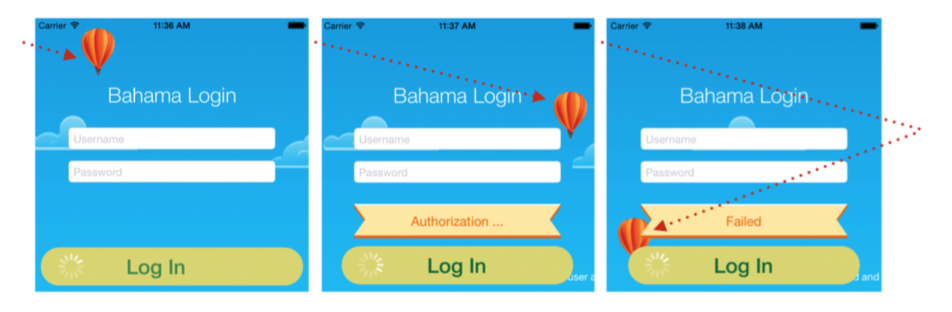
flight.keyTimes = [0.0, 0.5, 1.0]

balloon.addAnimation(flight, forKey: nil)

//动画执行完的位置

 balloon.position = CGPoint(x: -50.0, y: loginButton.center.y)





动画完成，可以使用同样的方式来处理bounds, position, transform.

将之前的动画全部换为Layer层动画