

Core Animation — — By Dorayo

隐式动画

Core Animation 可以这样理解——屏幕上的任何东西都可以（或者可能）做动画(0.25s)。动画并不需要你在Core Animation中手动打开，相反需要明确地关闭，否则他会一直存在，这就是隐式动画。

事务 (CATransaction)

事务实际上是Core Animation用来包含一系列属性动画集合的机制，任何用指定事务去改变可以做动画的图层属性都不会立刻发生变化，而是当事务一旦提交的时候开始用一个动画过渡到新值。

CATransaction没有属性或者实例方法，并且也不能用+alloc和-init方法创建它。但是可以用+begin和+commit分别来入栈或者出栈。(UIView收尾式动画)

问题：但当你改变一个属性，Core Animation是如何判断动画类型和持续时间的呢？

=> [事务的设置](#) [图层行为](#)

图层行为

我们知道Core Animation通常对CALayer的所有属性（可动画的属性）做动画，但是UIView把它关联的图层的这个特性关闭了。为了更好说明这一点，我们需要知道隐式动画是如何实现的：

我们把改变属性时CALayer自动应用的动画称作行为，当CALayer的属性被修改时候，它会调用-actionForKey:方法，传递属性的名称，接下来：

- 图层首先检测它是否有委托，并且是否实现CALayerDelegate协议指定的-actionForLayer:forKey方法。如果有，直接调用并返回结果。
- 如果没有委托，或者委托没有实现-actionForLayer:forKey方法，图层接着检查包含属性名称对应行为映射的actions字典。
- 如果actions字典没有包含对应的属性，那么图层接着在它的style字典接着搜索属性名。
- 最后，如果在style里面也找不到对应的行为，那么图层将会直接调用定义了每个属性的标准行为的-defaultActionForKey:方法。

总结：

- UIView关联的图层禁用了隐式动画，对这种图层做动画的唯一办法就是使用UIView的动画函数，或者继承UIView，并覆盖-actionForLayer:forKey:方法，或者直接创建一个显式动画。

- 对于单独存在的图层，我们可以通过实现图层的`-actionForLayer:forKey:`委托方法，或者提供一个`actions`字典来控制隐式动画。

呈现树与模型树

显式动画

属性动画

- `CABasicAnimation`
- `CAKeyframeAnimation`

它不限制于设置一个起始和结束的值，而是可以根据一连串随意的值来做动画。

虚拟属性 eg:`transform.rotation`

动画组动画

- `CAAnimationGroup`

转场动画

- `CATransition`

属性动画只对图层的可动画属性起作用，所以如果要改变一个不能动画的属性（比如图片），或者从层级关系中添加或者移除图层，属性动画将不起作用，这时可能就需要转场动画了。

过渡动画首先展示之前的图层外观，然后通过一个交换过渡到新的外观