



资讯

论坛

代码

招聘

CVP



登录 | 注册

iOS开发 Swift App Store研究 产品设计 应用评测 游戏开发 苹果相关 WebApp 安卓相关 营销推广 业界动态 程序人生

What will your legacy be?

Find out more >



最省心的支付 SDK

首页 > iOS开发

Core Animation编程指南

2013-12-30 10:35 编辑: suiling 分类: iOS开发 来源: CocoaChina

0 75172

招聘信息: iOS开发工程师

图层的内容动画

Core Animation提供的基础设施让轻松创建复杂图层动画变得异常简单, Core Animation扩展了所有拥有图层的视图。例如改变图层框架矩形的尺寸, 改变其在屏幕上的位置, 应用旋转变换, 改变它的透明度。使用Core Animation初始化一个动画和改变属性一样简单, 你也可以显式的创建一个动画并设置动画的参数。

用简单的动画表现图层属性的变化

你可以以显式或隐式的执行简单的动画。隐式动画使用默认的定时器和动画属性展现动画。而显式动画需要你为动画对象配置一些参数。所以当默认的定时器能够很好的为你服务并且你所要的动画效果不需要太多代码时, 隐式动画则非常的适合你。

简单的动画包括改变一个图层的属性, 以及随着时间的推移, 让Core Animation以动画的形式展现这些属性的变化。图层定义了许多会影响图层可视外观的属性。改变这些属性是以动画方式展现外观变化的一种方式。例如将图层的透明度从1.0修改为0.0, 这将引起图层的淡出特效, 最后图层变为透明。

重要: 虽然你可以使用Core Animation接口直接让图层支持的视图产生动画, 但这样做经常需要额外的步骤。

为了触发隐式动画, 你所要做的是更新图层对象的属性。当更改的目标是图层树中的图层对象, 更改将立即反映到对象上。而图层对象的可视外观并不会立即发生变化, Core Animation将图层对象属性的变化当做是一个触发器, 用以创建和安排一个或多个可执行的隐式动画。如清单3-1那样修改图层对象属性, 将引起Core Animation为你创建动画对象, 动画对象将被安排在下次更新周期运行。

清单3-1 隐式动画的方式呈现图层属性的变化

```
01. theLayer.opacity = 0.0;
```

为了显式地使用动画对象呈现相同的变化, 创建一个CABasicAnimation对象并配置该对象的动画参数。在添加动画到图层之前, 你可以设置动画的开始值与结束值, 改变持续时间, 或任何其他动画参数。你指定你想要动画的属性的键路径, 接着设置动画参数。为了执行一个动画, 你使用addAnimation: forKey: 方法将动画对象添加到你想要展现动画的图层上。

清单3-2 显式动画的方式呈现图层属性的变化

```
01. CABasicAnimation* fadeAnimation = [CABasicAnimation animationWithKeyPath:@"opacity";
02. fadeAnimation.fromValue = [NSNumber numberWithFloat:1.0];
03. fadeAnimation.toValue = [NSNumber numberWithFloat:0.0];
04. fadeAnimation.duration = 1.0;
05. [theLayer addAnimation:fadeAnimation forKey:nil];
06.
07. theLayer.opacity = 0.0;
```

// 改变图层实际
// 记得更新图层树

注意: 创建一个显示动画, 推荐是赋值一个值给动画对象的fromValue属性。如果你没有为该属性指定值, Core Animation将使用图层的当前值作为开始值。如果已经更新了属性作为它的最终值, 这将致使fromValue属性值遭到干扰, 结果可能并不是你想要的。

不同于隐式动画, 隐式动画会更新图层对象的值。而显示动画不会更改图层树中的数据。显示动画仅是创建了一个动画。在动画结束之后, Core Animation从图层中移除该动画对象并使用当前的数据值重绘图层。如果你想让显示动画的改变成为永久性的, 如你在之前的例子中看到的, 你必须更新图层属性。

隐式和显示动画都会在当前运行循环周期结束之后开始执行, 并且当前的线程必须拥有一个用于执行动画的运行循环。如果你改变了图层的动画属性或者给图层添加了多个动画对象。所有这些动画将会同时被执行。例如将图层移动到屏幕之外的时候添加渐隐动画, 这两个动画是同时运行的。然而你可以让动画对象在一个特殊的时间开始执行。

用关键帧动画表现一个图层属性的变化

尽管一个基于属性的动画是从开始值到结束值改变一个属性, 一个CAKeyframeAnimation对象让你通过一个目标集合产生动画, 在某种程度上, 动画可能是线性的或者非线性。一个关键帧动画由一个目标数据值集合组成。每个值的时间应该是可达的。最简单的配置是你使用一个数组指定值和时间。对于一个图层位置的变化, 你可以有一个跟随路径的变化。动画对象使用你指定的关键帧并通过在某个值到另一个在给定时间区间内插值的方式构建动画。

图3-1显示了一个图层position属性的5秒动画。position是动画跟随一个路径。使用一个CGPathRef数据类型指定的。这个动画的代码如清单3-3所示。

热门资讯



这些 iOS 面试基础题目, 你都深入了解吗?

点击量 11455



自定义 push 和 pop 实现有趣的相册翻开效果

点击量 8078



Objective-C 的现代语法和新特性

点击量 7935



HTML5 崛起: 不再高冷, 不再小众

点击量 6461



月薪3万的程序员都避开了哪些坑

点击量 5997



CVP关东升: Swift精华教程汇总 (第一期)

点击量 5715



关于 @synchronized, 这儿比你想知道的还要

点击量 5542



【译】4个你需要知道的Asset Catalog的秘

点击量 5253



【译】17个提升iOS开发效率的必用工具

点击量 5209



程序员技术晋升非正式攻略

点击量 4514

综合评论

xib 和sb都是一坨屎 只有纯代码手动算Frame的开发方式才是正道
呵呵呵 评论了 新手向: 五分钟搭建App设置页面_纯代码+模型...

原文第一步, 有人说了不能搜索出结果.....结果照着翻译还是不行.....
a412739861 评论了 iOS 9学习系列: Search API...

落后了。
soleapp 评论了 Mozilla正式发布iOS版Firefox浏览器...

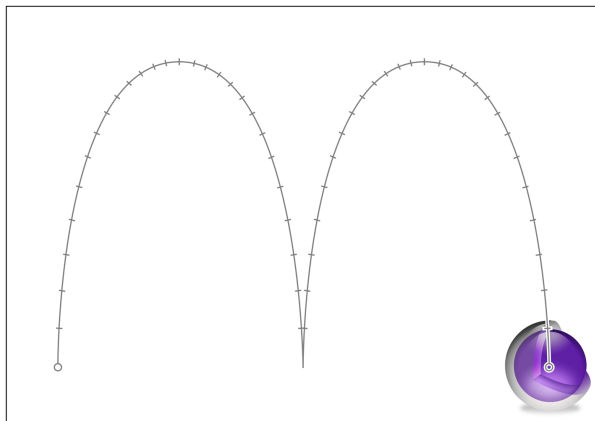


图3-1 一个图层的position属性的5秒关键帧动画

代码清单3-3 创建一个跳弹关键帧动画

```
01. // 创建一个实
02. CGMutablePathRef thePath = CGPathCreateMutable();
03. CGPathMoveToPoint(thePath, NULL, 74.0, 74.0);
04. CGPathAddCurveToPoint(thePath, NULL, 74.0, 500.0,
05.                        320.0, 500.0,
06.                        320.0, 74.0);
07. CGPathAddCurveToPoint(thePath, NULL, 320.0, 500.0,
08.                        566.0, 500.0,
09.                        566.0, 74.0);
10. CAKeyframeAnimation * theAnimation;
11.
12. theAnimation=[CAKeyframeAnimation animationWithKeyPath:@"position"];
13. theAnimation.path=thePath;
14. theAnimation.duration=5.0;
15. CGPathRelease(thePath);
16.
17. [theLayer addAnimation:theAnimation forKey:nil];
```

// 创建一个动画

// 为图层添加

指定关键帧的值

关键帧的值是关键帧动画中最重要的部分。这些值定义了动画在整个执行期间内的行为。指定关键帧值的主要方式以对象数组作为它的值。但是对于包含CGPoint数据类型（比如图层的anchorPoint属性和position属性），你可以指定一个CGPathRef数据类型替代。

当指定一个关键帧值数组，你放到数组中内容依赖于属性需要的数据类型。你可以直接添加一些对象到数组中。然而一些对象必须在添加到数组中之前被转换为id类型，所有标量类型或结构体必须被包装为对象，比如：

- 对于属性类型为CGRect（例如bounds和frame属性），使用NSValue对象包装每一个矩形。
- 对于图层的变换属性，使用NSValue包装每一个CATransform3D矩阵。动画这个属性将引起关键帧动画给图层轮流应用每个变换矩阵。
- 对于borderColor属性，在添加到数组之前，转换CGColorRef数据类型为id类型。
- 对于属性为CGFloat类型，在添加到数组之前，使用NSNumber包装每个值。
- 为了动画图层的内容属性，指定一个CGImageRef数据类型属性。

对于一个CGPoint数据类型的属性，你可以创建一个点（使用NSValue对象包装）数组，或者使用CGPathRef对象指定跟踪的路径。当你指定一个点数组，关键帧动画对象在每一个连续的点之间绘制一条线，并沿着这些线移动。当你指定一个CGPathRef对象，动画起始于路径的开始点并跟随路径线移动，这包括沿着任何曲面。你可以使用开放的或者封闭的路径。

指定关键帧动画的定时器

关键帧动画的定时与步调比基本动画来的要复杂。以下是几个用于控制定时和步调的属性：

- calculationMode属性定义了计算动画定时的算法。该属性值会影响其他与定时相关属性的使用方式。
- 线性和曲线动画，动画的calculationMode属性被设置为kCAAnimationLinear或CAAnimationCubic，属性值被用于提供定时器信息以生成动画。这些模式值让你最大化控制动画的定时器。
- 节奏动画，动画的calculationMode属性被设置为kCAAnimationPaced或kCAAnimationCubicPaced，这些属性值不依赖于keyTimes或timingFunctions属性提供的额外定时器值。相反，定时器值被隐式地计算以提供一个常速率动画。
- 离散动画，动画的calculationMode属性被设置为kCAAnimationDiscrete，该值将引起动画属性从一个关键帧跳到另一个没有任何补间动画的下一个关键帧。计算模式使用keyTimes属性值，但忽略timingFunctions属性。
- keyTimes属性为应用在每一关键帧指定应用到每一个关键帧上的计时器。该属性只在calculationMode属性被设置为kCAAnimationLinear，kCAAnimationDiscrete，kCAAnimationCubic时被使用。它不应用在节奏动画中。
- timingFunctions属性指定使用在每一个关键帧部分的定时曲线(该属性替换了继承的timingFunction属性)。

如果你想自己处理动画的定时，可以使用kCAAnimationLinear或kCAAnimationCubic模式与keyTimes和timingFunctions属性。keyTimes定义了应用在每一关键帧的时间点。所有中间值的定时由定时函数控制，定时函数允许你对各个部分应用缓入或缓出曲线定时。如果你不指定任何定时函数，动画将会是线性的。

停止一个隐式动画的运行

动画通常直到运行结束才会停止，但是你也可以根据需要使用以下技术提前停止动画：

小编呢？

liuyan3176 评论了 苹果眼里的二次元游戏是怎样的？ ...

这什么文章啊，一遍遍重复多少内容啊，凑字数吗？？

lixuwensmile 评论了 苹果眼里的二次元游戏是怎样的？ ...

干活，但国内死翘翘

wyxy2005 评论了 赛式方法论（上）：为什么你的游戏做不完、做得烂？ ...

还是赞一个吧！静态cell这个还是很适合新手的！

447690182@qq.c 评论了 新手向：五分钟搭建App设置页面_纯代码 + 模型...

mark一下，很精彩

love270628124 评论了 iOS开发——UI组件（个人整理） ...

mark一下，很精彩

love270628124 评论了 iOS开发——UI组件（个人整理） ...

傻子都知道吧

_wangyk 评论了 新手向：五分钟搭建App设置页面_纯代码 + 模型...

相关帖子

为啥cocoaChina代码库专区不能下载代码了

新手一枚，这是我的APP近期销售数据的结果，不知道苹果啥时候能给我打钱？有啥需要注意的么

ios 如何区分2g 3g 4g WiFi，各位大神

谁知道ITC新版本怎么查看用户评论？

走一走瞧一瞧，iOS8下获取相册所有图片

UIBarButtonItem这个警告是怎么回事呢

cocos2dx3.8.1不能读取cocos2.3.2的csb? ??

ReactiveCocoa只支持iOS8以上的系统么？

如何防止数据库被破解

为了从图层上移除单独的动画对象，调用图层的`removeAnimationForKey`：方法移除你的动画对象。该方法使用的键要与调用`addAnimation: forKey:`方法传入的键一致。你指定的键必须不为`nil`。

为了移除图层的所有动画对象。调用图层的`removeAllAnimations`方法。该方法立即会移除所有进行中的动画，并使用图层当前的状态信息重绘图层。

注意：你不能直接移除图层的隐式动画。

当你从图层上移除一个动画，Core Animation通过使用图层当前的值重绘图层做出响应。因为当前的值通常是动画的结束值，这可能会引起图层的外观突然的跳跃。如果你想图层的外观仍然在动画最后一帧出现的地方。你可以检索在呈现树中的对象的最终值，并设置这些值到图层树中的对象。

关于临时暂停动画的信息，请参阅“清单5-4”。

同时动画多个属性变化

如果你想同时给一个图层对象应用多个动画，你可以使用`CAAnimationGroup`对象将这些动画放在一个组里。通过使用单独的配置点，使用组对象简化了对多个动画对象的管理。应用于动画组的定时器和持续值将使用相同的值覆盖单个动画对象。

清单3-4显示了如何使用相同定时和持续时间的动画组执行两个与边框相关的动画。

清单3-4 同时执行两个动画

```
01.
02. CAKeyframeAnimation* widthAnim = [CAKeyframeAnimation
03. animationWithKeyPath:@"borderWidth"]; // Animatic
04. NSArray* widthValues = [NSArray arrayWithObjects:@1.0, @10.0, @5.0, @30.0, @0.5, @15.0, @2.0,
05. widthAnim.values = widthValues;
06. widthAnim.calculationMode = kCAAnimationPaced;
07.
08. CAKeyframeAnimation* colorAnim = [CAKeyframeAnimation
09. animationWithKeyPath:@"borderColor"]; // Animatic
10. NSArray* colorValues=[NSArray arrayWithObjects:(id)[UIColorgreenColor].CGColor, (id)[UIColororangeColor].CGColor];
11. colorAnim.values = colorValues;
12. colorAnim.calculationMode = kCAAnimationPaced
13.
14. CAAnimationGroup* group = [CAAnimationGroup animation];
15. group.animations = [NSArray arrayWithObjects:colorAnim, widthAnim, nil]; // Animatic
16. group.duration = 5.0;
17. [myLayer addAnimation:group forKey:@"BorderChanges"];
```

对于分组动画的更高级的方式是使用一个事务对象。事务提供了更加灵活的方式，允许你创建内嵌的动画集合以及给每一个动画赋值不同的动画参数。

检测一个动画的结束

Core Animation提供对动画开始与结束的检测支持。这些通知是执行所有与动画相关的内务处理的最佳时刻。比如说你可能使用开始通知设置一些相关状态信息，使用对应的结束通知清理这些状态。

有两种不同的方式获取关于动画状态的通知：

使用`setCompletionBlock`：方法添加一个完成块给当前的事务。当事务中的所有动画完成后，事务将执行你的完成块。

给`CAAnimation`对象赋值一个代理，该代理实现了`animationDidStart`：方法和`animationDidStop: finished:`代理方法。

如果你想将两个动画链接在一起，使得当第一个动画结束之后启动第二个动画。不要使用动画通知。相反，使用动画对象的`beginTime`属性在希望的时间启动动画。为了将两个动画链接在一起，设置第二个动画的开始时间为第一个动画的结束时间。

如何动画有图层支持视图

如果一个图层属于一个图层支持视图，创建动画的推荐做法是使用由UIKit或AppKit提供的基于视图的动画接口。有几种使用Core Animation接口直接动画图层的方式，但如何创建这些动画依赖于目标平台。

iOS中图层更改的规则

因为iOS视图底层总有一个图层，UIView类从图层对象派生了大部分的数据。因此，对图层属性的更改会通过视图对象自动表现出来。此行为代表你可以使用Core Animation或UIView接口完成对图层属性的更改。

如果你想使用Core Animation初始化动画，你必须在一个基于视图的动画块内部执行所有Core Animation调用。UIView类默认是关闭图层动画的，你在动画块之外所做的改变都不是动画，但是你可动画块中重新启用图层动画。清单3-5展示了一个关于如何隐式改变图层opacity和显式改变图层的position例子，在例子中myNewPosition变量被事先计算并被块所捕获。两个动画都开始于相同的时间，但透明度动画使用默认的定时，而位置动画使用动画对象指定的定时。

清单3-5 动画附属在iOS视图上的图层

```
01. [UIView animateWithDuration:1.0 animations:^(
02.
03. myView.layer.opacity = 0.0;
04. // Change t
05. CABasicAnimation* theAnim=[CABasicAnimation animationWithKeyPath:@"position"];
06. theAnim.fromValue = [NSValue valueWithCGPoint:myView.layer.position]; // Change the positio
07. theAnim.toValue = [NSValue valueWithCGPoint:myNewPosition];
08. theAnim.duration = 3.0;
09. [myView.layer addAnimation:theAnim forKey:@"AnimateFrame"];
10. }];
```

记住将更新视图约束作为动画的一部分

如果你使用基于约束的布局规则管理你的视图位置，你必须移除任何可能对动画的配置部分造成干扰的约束。约束将影响所有你对视图的位置和尺寸的改变。也会影响到视图和它的子视图的关系。如果你正在动画改变所有这些项目，你可以移除这些约束后再做改变，接着应

微博



CocoaChina

加关注

【Mac App Store 出错 部分应用需要重装解决】11月13日上午消息，昨晚，苹果公司的Mac版应用商店(Mac App Store)发生安全证书错误，导致部分应用需要重装才能解决。很多用户在社交网络上报告这个问题，一些自Mac App Store下载并首次打开的应用时，会弹出“损坏”的提示。<http://t.cn/RUlmxoL>



18分钟前

转发 | 评论

【苹果Apple Pencil发货迟遭抨击：不如三星和微软】iPad Pro的下单、开箱、评测，是这两天苹果新闻的主

What will your legacy be?

Find out more >



用需要的新约束。

构建图层层级

大多数时候，使用图层的最佳方式是联合使用图层和视图对象。你可能需要向视图添加另外的图层对象以增加视图层级。你可能为了获得更佳的性能或者单独使用视图实现你需要的功能会比较困难的情况下而选择使用图层。因此你需要知道如何管理你创建的图层的层次。

重要：在OS X 10.8及之前的版本中，推荐的做法是最小化图层层次的使用，尽量使用支持图层的视图。引进图层重绘策略的OS X版本让你可以自定义支持图层的视图的行为以及获得与之前使用独立图层的那种性能。

将图层排列为图层层级

图层层级在许多方式上都与视图的层级相似。你嵌入一个图层到另一个图层中，在嵌入的图层（称为sublayer）与父图层（称为superlayer）之间将创建了一个父子关系。这个父子关系会影响许多子图层的许多方面。比如当图层的内容位于它的父级内容之上，他的位置则被指定为相对于它父级的坐标系，并且也会被任何应用于父级的变换所影响。

添加、插入、移除子图层

每个图层对象都拥有添加、插入和移除子图层的方法。表4-1是对这些方法和行为的总结。

行为	方法	描述
添加图层	addSublayer:	给当前的图层添加一个新的子层对象。子层被加入到当前图层的子层列表的末尾。图层将显示在所有与其有相同zPosition属性值的同胞图层的顶部。
插入图层	insertSublayer: above: insertSublayer: atIndex: insertSublayer: below:	将子图层插入到子图层层次中指定的索引或者相对于其他子图层的位置。当插入到其他图层的上方或下方，你仅需指定子图层在子图层数组中的位置。图层的可视性主要由它的zPosition属性决定，然后是由其在子图层数组中的位置决定。
移除图层	removeFromSuperlayer	将子图层从父图层中移除
交换图层	replaceSublayer: with	将某个子图层与另外的图层交换位置。如果你插入的图层已经在另外的图层层次中，首先它将从图层之前的图层层次中移除

当需要操作图层对象时你可以使用前述的这些方法。你不能使用这些方法去排列属于layer-backed视图的图层。然而，一个layer-backed视图可以作为你创建的独立图层的父图层。

子图层的位置与尺寸

当添加和插入一个子图层，在它显示到屏幕之前你必须设置子图层的尺寸和位置。你可以在子图层被添加到图层层次之后更改其位置与尺寸，但应该养成在创建图层的时候就设置这些值的习惯。

你使用bounds属性设置一个子图层的尺寸，使用position属性设置子图层在它的父图层内的位置。边界矩形的原点几乎总是（0，0），尺寸可以是任何尺寸值。Position属性被解释为相对于图层的锚点，默认是锚点被定为在图层的中心。如果你不给这些属性赋值，Core Animation将设置图层的初始宽和高都为0，图层的position属性值为（0，0）。

```
01. myLayer.bounds = CGRectMake(0,0,100,100);
02. myLayer.position = CGPointMake(200,200);
```

重要：你的图层宽和高总是设置整型数。

图层层级对动画的影响

一些superlayer属性会影响应用到它的子图层的动画的行为。speed属性即是如此的属性。对于动画来说，speed是一个倍增器。默认该属性的值被设为1.0，但将该值改为2.0将引起动画运行速度是原始速度的两倍，因此花费一半的时间就动画就结束了。该属性影响的不仅是设置该属性的图层也会影响图层的子图层。像这样的改变也会倍增。如果子图层和它的父图层都拥有2.0的速度，则在子图层上动画的运行将是它的原始速度的4倍。

图层的大部分属性的变化仅影响所有被它包含的子图层。比如对图层应用一个旋转变换将旋转该图层以及所有它的子图层。相似地，改变一个图层的透明度也会改变它的子图层透明度。

调整你的图层层次的布局

Core Animation对调整子图层的尺寸和位置有多个选项，以应对它的子图层的改变。在iOS中，layer-backed视图的普遍使用，使得图层层次的创建变得次要。仅仅是手动布局被支持。对于OS X，有多种轻松管理你的视图层次的选项可用。

图层级的布局仅仅相关与如果你正在构建的图层层次使用的是你创建的独立图层。如果你的app的图层都是与图层相关，使用基于图层的布局支持对你的视图尺寸与位置的更新，以应对尺寸与位置的变化。

手动规划你的图层层次

在iOS和OS X中，你可以通过在父图层的代理对象上实现layoutSublayersOfLayer:方法手动处理布局。你使用该方法调整任何当前内嵌到图层中的子图层的位置和尺寸。当手动布局更新时，对每个子图层的位置执行必要的计算取决于你。

如果你实现一个自定义图层子类，你的子类可以覆盖layoutSublayers方法并使用该方法（而不是代理）去处理任何布局任务。如果你需

要对你自定义的图层类内的子图层的定位进行完全的控制的情况下，你只需要覆盖这个方法。

子图层与裁剪

与视图不同的，一个父图层不会自动地裁剪超出其边界的子图层的内容。相反，默认情况下，父图层允许他的子图层完全的被显示。然而，通过设置图层的masksToBounds属性为YES，你可以重新启用裁减功能。

如果圆角半径被指定的话，一个图层的裁减蒙版形状包括图层的圆角半径。图4-3说明了masksToBounds属性如何影响一个拥有圆角半径的图层。当masksToBounds属性被设置为NO，整个子图层被显示，尽管它已经超出了它的父图层的边界。将masksToBounds属性设为YES，将引起超出父图层的子图层内容被裁减掉。

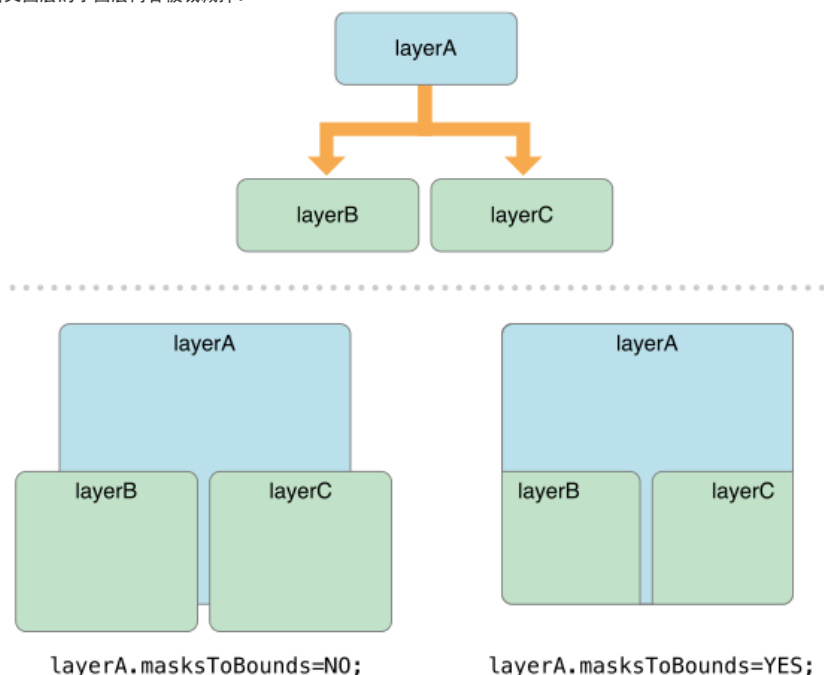


图4-3 子图层的裁减区域设为父图层的边界

在图层之间转换坐标值

你可能需要将某图层中的一个坐标值转换为屏幕坐标位置相同而处在不同图层的一个坐标值。CALayer提供了一组简单的转换方法以应对这种情况：

ØconvertPoint: fromLayer:

ØconvertPoint: toLayer:

ØconvertRect: fromLayer:

ØconvertRect: toLayer:

除了转换点和矩形的值，你也可以在两个图层之间转换时间，使用convertTime: fromLayer: 与convertTime: toLayer: 方法。每个图层定义了自身的本地时空，使用本地时空与其他系统同步动画的开始和结束。这些时空默认被同步；然而，如果你改变了图层的动画速度，图层的时空也相应地发生变化。你可以使用时间转换方法考虑这样的因素，确保两个图层的时间是同步的。



微信扫一扫

订阅每日移动开发及APP推广热点资讯

公众号: CocoaChina

我要投稿

收藏文章

分享到:

相关资讯

What will your legacy be?
Find out more >

UNIVERSITY OF
BATH

我来说两句



您还没有登录！请 [登录](#) 或 [注册](#)

所有评论 (0)