

首页 资讯 问答 论坛 Cocos2d-x 开发者中心

新手入门 专题 新闻日历

站内搜索



主页 > 业界动态

CAShapeLayer和CAGradientLayer

发布于：2014-07-05 09:56

阅读数：28472

“两个动画效果来了解一下CALayer的两个重要的subClass：CAGradientLayer和CAShapeLayer。先看CAShapeLayer，我们做一个和Spark相机一样的圆形进度，每一段有一种颜色，标识不同时间段录的视频。”

阅读器

CAShapeLayer

CAGradientLayer

CAShapeLayer And CAGradientLayer

2个动画爱上CALayer

转自KooFrank's Blog

两个动画效果来了解一下CALayer的两个重要的subClass：CAGradientLayer和CAShapeLayer。微视录制视频的时候那个进度效果和Spark相机类似，但是个人还是比较喜欢Spark相机的录制的效果。

CAShapeLayer



我们做一个和Spark相机一样的圆形进度，每一段有一种颜色，标识不同时间段录的视频。

首先，我们创建一个UIView的子类叫RecordingCircleOverlayView这样看起来比较有意义，然后我们看到圆形进度条有一个底色是灰色的圆形轨迹，所以我们创建一个CAShapeLayer，然后提供一个CGPathRef给它的path属性，我们使用UIBezierPath这个类的bezierPathWithArcCenter:radius:startAngle:endAngle:clockwise:这个方法给CAShapeLayer提供path。

开发者通道

排行榜 代码库 图书库

网站库 发码区 工具库

招聘区 外包区 问答区

关注CocoaChina



关注微信



移动版

最近更新

- 1 OCaml 发布 iOS 7 版编译器 (OCaml) 2014-08-26
- 2 iOS 8 自动调整UITableView和UICollectionView 2014-08-25
- 3 对访问控制与protected的理解 2014-08-20
- 4 Facebook Shimmer 实现原理 2014-08-18
- 5 在企业内部分发 iOS 应用程序 2014-08-18
- 6 iOS中图形图像处理第一部分:位图图 2014-08-12
- 7 WWDC 2014 大会中的 Playground : 2014-08-11
- 8 Collection View 动画 2014-07-25
- 9 iTerm2 2.0版本已发布,添加大量新功能 2014-07-21
- 10 重制Skype应用中Action Sheet的动

图中我们可以看到有彩色一些片段, 我们使用另外一个CASHapeLayer和同样的CGPathRef作为背景层, 由于是同样的Path, 所以我们给UIBezierPath创建一个属性, 这样不用每次都重复创建。

```
01. CGPoint arcCenter = CGPointMake(CGRectGetMidY(self.bounds), CGRectGetMidX(self.bounds));
02. CGFloat radius = CGRectGetMidX(self.bounds) - insets.top - insets.bottom;
03.
04. self.circlePath = [UIBezierPath bezierPathWithArcCenter:arcCenter
05. radius:radius
06. startAngle:M_PI
07. endAngle:-M_PI
08. clockwise:NO];
```

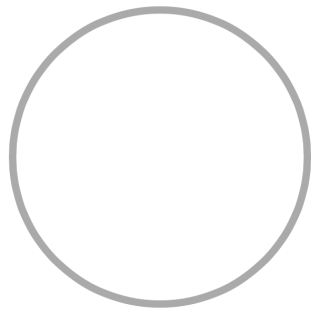
开始角度M_PI和结束角度-M_PI和Spark相机是一样的逆时针方向, 然后我们再创建一个背景层

```
01. CAShapeLayer *backgroundLayer = [CAShapeLayer layer];
02. backgroundLayer.path = self.circlePath.CGPath;
03. backgroundLayer.strokeColor = [[UIColor lightGrayColor] CGColor];
04. backgroundLayer.fillColor = [[UIColor clearColor] CGColor];
05. backgroundLayer.lineWidth = self.strokeWidth;
```

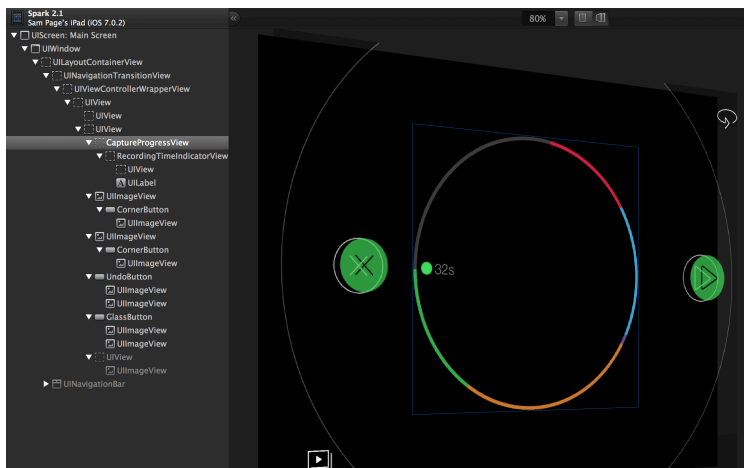
然后我们把backgroundLayer添加为RecordingCircleOverlayView的subLayer

```
01. [self.layer addSublayer:backgroundLayer];
```

如果我们build运行成功的话应该是这样的。



现在我们需要一个方法来实现开始和停止进度, 如果我们回头去看Spark Camera, 我们需要按下手指才会开始松开结束, 首先UITapGestureRecognizer和UIControlEventTouchUpInside没有方法检测按下和松开, 但是我们可以用UIControlEventTouchDown, 但是我们在Reveal里面并没有看到它是这么做的, 所以最后决定使用复写UIResponder的touchesBegan:WithEvent: and touchesEnded:WithEvent:方法来实现。



有个这个方法后, 我们可以控制CASHapeLayer的strokeEnd的属性大小来实现动画效果, 首页我们先设置它的值为0然后把这个layer添加作为子类。

```
01. CAShapeLayer *progressLayer = [CAShapeLayer layer];
02. progressLayer.path = self.circlePath.CGPath;
03. progressLayer.strokeColor = [[self randomColor] CGColor];
04. progressLayer.fillColor = [[UIColor clearColor] CGColor];
05. progressLayer.lineWidth = self.strokeWidth;
06. progressLayer.strokeEnd = 0.f;
```

2014-07-18

推荐内容

热点内容



iTerm2 2.0版本已发布, 添加大量新功能, 更易于使用



如何做出优秀的App Store应用截图, 多图讲述, 超详细!



WWDC2014观感兼回答iOS初学者的困惑



WWDC 2014 iOS 8游戏相关的十个重大更新



iOS应用国际化教程 (2014版)

然后我们发现多个CAShapeLayer分别代表不同的段, 而且每个CAShapeLayer都有自己的strokeEnd,所以
我们创建一个数组, 把每一个CAShapeLayer添加到数组里。

```
01. [self.progressLayers addObject:progressLayer];
```

继而我们还需要一个属性代表当前的正在增加可以动画的片段, 所以我们添加一个属性来记录当前的进度的layer。

```
01. self.currentProgressLayer = progressLayer;
```

所以最后方法看起来是这样的。

```
01. - (void)addNewLayer
02. {
03.     CAShapeLayer *progressLayer = [CAShapeLayer layer];
04.     progressLayer.path = self.circlePath.CGPath;
05.     progressLayer.strokeColor = [[self randomColor] CGColor];
06.     progressLayer.fillColor = [[UIColor clearColor] CGColor];
07.     progressLayer.lineWidth = self.strokeWidth;
08.     progressLayer.strokeEnd = 0.f;
09.
10.     [self.layer addSublayer:progressLayer];
11.     [self.progressLayers addObject:progressLayer];
12.
13.     self.currentProgressLayer = progressLayer;
14. }
```

为了让它可以有动画, 我们有两个重点, 其一我们可以使用rotation transform属性, 但是我们使用CAShapeLayer的strokeStart和strokeEnd结合起来实现动画, 其二停止动画后我们可以使用截图当前的状态同时移除动画, 这样就可以保留每个状态的颜色。为了实现这些, 我们使用CABasicAnimation和CALayer的属性presentationLayer, 直接上代码。

```
01. - (void)updateAnimations
02. {
03.     CGFloat duration = self.duration * (1.f - [[self.progressLayers firstObject] strokeEnd]);
04.     CGFloat strokeEndFinal = 1.f;
05.
06.     for (CAShapeLayer *progressLayer in self.progressLayers)
07.     {
08.         CABasicAnimation *strokeEndAnimation = nil;
09.         strokeEndAnimation = [CABasicAnimation animationWithKeyPath:@"strokeEnd"];
10.         strokeEndAnimation.duration = duration;
11.         strokeEndAnimation.fromValue = @(progressLayer.strokeEnd);
12.         strokeEndAnimation.toValue = @(strokeEndFinal);
13.         strokeEndAnimation.autoreverses = NO;
14.         strokeEndAnimation.repeatCount = 0.f;
15.         strokeEndAnimation.fillMode = kCAFillModeForwards;
16.         strokeEndAnimation.removedOnCompletion = NO;
17.         strokeEndAnimation.delegate = self;
18.         [progressLayer addAnimation:strokeEndAnimation forKey:@"strokeEndAnimation"];
19.
20.         strokeEndFinal -= (progressLayer.strokeEnd - progressLayer.strokeStart);
21.
22.         if (progressLayer != self.currentProgressLayer)
23.         {
24.             CABasicAnimation *strokeStartAnimation = nil;
25.             strokeStartAnimation = [CABasicAnimation animationWithKeyPath:@"strokeStart"];
26.             strokeStartAnimation.duration = duration;
27.             strokeStartAnimation.fromValue = @(progressLayer.strokeStart);
28.             strokeStartAnimation.toValue = @(strokeEndFinal);
29.             strokeStartAnimation.autoreverses = NO;
30.             strokeStartAnimation.repeatCount = 0.f;
31.             strokeStartAnimation.fillMode = kCAFillModeForwards;
32.             strokeStartAnimation.removedOnCompletion = NO;
33.             [progressLayer addAnimation:strokeStartAnimation forKey:@"strokeStartAnimation"];
34.         }
35.     }
36.     CABasicAnimation *backgroundLayerAnimation = [CABasicAnimation animationWithKeyPath:@"strokeStart"];
37.     backgroundLayerAnimation.duration = duration;
38.     backgroundLayerAnimation.fromValue = @(self.backgroundLayer.strokeStart);
39.     backgroundLayerAnimation.toValue = @(1.f);
40.     backgroundLayerAnimation.autoreverses = NO;
41.     backgroundLayerAnimation.repeatCount = 0.f;
42.     backgroundLayerAnimation.fillMode = kCAFillModeForwards;
43.     backgroundLayerAnimation.removedOnCompletion = NO;
44.     backgroundLayerAnimation.delegate = self;
45.     [self.backgroundLayer addAnimation:backgroundLayerAnimation forKey:@"strokeStartAnimation"];
```

```
46. } }
```

上面代码中我们看到我们遍历了所有的CAShapeLayer,给每个strokeEnd添加了CABasicAnimation动画,然后给不是当前的layer的strokeStart属性添加了一个动画。再来看看duration,假设一圈代表45秒钟,这个意味着每次停止之后又开始的话duration肯定是减少的,所以用duration代表一圈剩余的可以录制的时间,再看strokeEndFinal,假设有很多段,肯定不是每个段的strokeEnd都是1所以这个是用来标识每段可以达到的最终距离一圈为(0-1)。最后我们需要更新background layer除去有彩色段剩余的地方。

你可能注意到上面的代码里面并没有移除动画,所以对于显示每一个CAShapeLayer我们设置都是通过layers的presentationLayer设置strokeStart和strokeEnd,然后移除CAShapeLayer上的所有动画。

presentationLayer在文档中是这么说的:

“While an animation is in progress, you can retrieve this object and use it to get the current values for those animations.”

所以把上面所说的结合起来,代码应该是这样的。

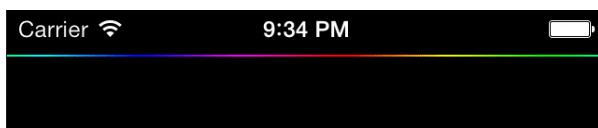
```
01. - (void)removeAnimations
02. {
03.     for (CAShapeLayer *progressLayer in self.progressLayers)
04.     {
05.         progressLayer.strokeStart = [progressLayer.presentationLayer strokeStart];
06.         progressLayer.strokeEnd = [progressLayer.presentationLayer strokeEnd];
07.         [progressLayer removeAllAnimations];
08.     }
09.     self.backgroundLayer.strokeStart = [self.backgroundLayer.presentationLayer strokeStart];
10.     [self.backgroundLayer removeAllAnimations];
11. }
```

最后,还有一个问题是我们需要确保我们完成了动画以后手指按下不要保持添加layer和更新动画这些操作,所以我们可以设置一个代理方法像这样,就大功告成了。

```
01. - (void)animationDidStop:(CAAnimation *)anim finished:(BOOL)flag
02. {
03.     if (self.hasFinishedAnimating == NO && flag)
04.     {
05.         [self removeAnimations];
06.         self.finishedAnimating = flag;
07.     }
08. }
```

最后你可以在[github](#)上面下载这个项目。

CAGradientLayer



首页我们创建一个UIView的子类,然后我们使用CAGradientLayer作为默认的CALayer。

```
01. + (Class)layerClass {
02.     return [CAGradientLayer class];
03. }
```

CAGradientLayer是CALayer的一个子类,添加了一些额外的属性,我们将是使用colors,startPoint,endPoint这些来创建一个有梯度的动画。

现在有几个方法来实现这种彩色的效果,一种是我现在将要使用的创建一个包含UIColor的数组,有不同的色调的值,在你的initWithFrame方法里添加一下代码:

```
01. // Use a horizontal gradient
02. CAGradientLayer *layer = (id)[self layer];
03. [layer setStartPoint:CGPointMake(0.0, 0.5)];
04. [layer setEndPoint:CGPointMake(1.0, 0.5)];
05.
06. // Create colors using hues in +5 increments
07. NSMutableArray *colors = [NSMutableArray array];
08. for (NSInteger hue = 0; hue <= 360; hue += 5) {
09.
10.     UIColor *color;
```

```
11.         color = [UIColor colorWithHue:1.0 * hue / 360.0
12.                     saturation:1.0
13.                     brightness:1.0
14.                     alpha:1.0];
15.         [colors addObject:(id)[color CGColor]];
16.     }
17.     [layer setColors:[NSArray arrayWithArray:colors]];
```

现在运行你可以看见一个水平光谱图，下一步创建移动的效果，我们可以遍历这个颜色的数组使用layer animation,一个动画结束的时候会前面的颜色方法最后重复这个进度，方法是这样：

```
01. - (void)performAnimation {
02.     // Move the last color in the array to the front
03.     // shifting all the other colors.
04.     CAGradientLayer *layer = (id)[self layer];
05.     NSMutableArray *mutable = [[layer colors] mutableCopy];
06.     id lastColor = [[mutable lastObject] retain];
07.     [mutable removeLastObject];
08.     [mutable insertObject:lastColor atIndex:0];
09.     [lastColor release];
10.     NSArray *shiftedColors = [NSArray arrayWithArray:mutable];
11.     [mutable release];
12.
13.     // Update the colors on the model layer
14.     [layer setColors:shiftedColors];
15.
16.     // Create an animation to slowly move the gradient left to right.
17.     CABasicAnimation *animation;
18.     animation = [CABasicAnimation animationWithKeyPath:@"colors"];
19.     [animation setValue:shiftedColors];
20.     [animation setDuration:0.08];
21.     [animation setRemovedOnCompletion:YES];
22.     [animation setFillMode:kCAFillModeForwards];
23.     [animation setDelegate:self];
24.     [layer addAnimation:animation forKey:@"animateGradient"];
25. }
26.
27. - (void)animationDidStop:(CAAnimation *)animation finished:(BOOL)flag {
28.     [self performAnimation];
29. }
```

为了增加一个标识进度的进行，我们可以使用mask属性来屏蔽一部分，在头文件中添加两个属性：

```
01. @property (nonatomic, readonly) CALayer *maskLayer;
02. @property (nonatomic, assign) CGFloat progress;
```

然后在initWithFrame:里面添加：

```
01. maskLayer = [CALayer layer];
02. [maskLayer setFrame:CGRectMake(0, 0, 0, frame.size.height)];
03. [maskLayer setBackgroundColor:[UIColor blackColor] CGColor];
04. [layer setMask:maskLayer];
```

创建一个宽度为0的mask覆盖整个View，mask的颜色不重要，当我们progress属性更新的时候我们会增加它的宽度，所以复写setProgress:方法像下面这样：

```
01. - (void)setProgress:(CGFloat)value {
02.     if (progress != value) {
03.         // Progress values go from 0.0 to 1.0
04.         progress = MIN(1.0, fabs(value));
05.         [self setNeedsLayout];
06.     }
07. }
08.
09. - (void)layoutSubviews {
10.     // Resize our mask layer based on the current progress
11.     CGRect maskRect = [maskLayer frame];
12.     maskRect.size.width = CGRectGetWidth([self bounds]) * progress;
13.     [maskLayer setFrame:maskRect];
14. }
```

现在当我们设置progress值的时候我们要确保它在0到1之间，然后下一步在layoutSubviews里面我们重新定义mask的值。

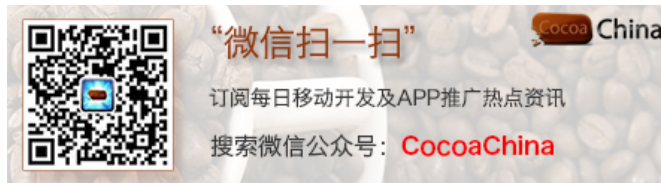
参考：

[Spark Camera's recording meter](#)

[Animated progress view with CAGradientLayer](#)

CocoaChina是全球最大的苹果开发中文社区，官方微信每日定时推送各种精彩的研发教程资源和工具，介绍app推广营销经验，最新企业招聘和外包信息，以及Cocos2d引擎、Cocos Studio开发工具包的最新动态及培训信息。关注微信可以第一时间了解最新产品和服务动态，微信在手，天下我有！

请搜索微信号“CocoaChina”关注我们！



(8)

看过此文章的用户还看过

iOS动画库Pop和Canvas各自的优势和劣势是什么？



先说 Canvas。Canvas 的目的是「Animate in Xcode Without Code」。开发者可以通过在 Storyboard 中指定 User Defined Runtime Attributes 来实现一些 Canvas 中...

访问人数：3371 查看详细



iOS开发中一些常见的并行处理

本文主要探讨一些常用多任务的最佳实践。包括Core Data的多线程访问，UI的并行绘制，异步网络请求以及一些在运行态内存吃紧的情况下处理大文件的方案等。??其实编...

访问人数：5676 查看详细



iOS 7交互式过渡

iOS 7新加入了一个介于ViewController之间的过渡的实现方法。本文将介绍如何利用自定义的过渡实现如图所示效果。在这个例子中，我们将在两个viewContoller的转换...

访问人数：3296 查看详细

3条评论

最新 最早 最热



宋尚永

mark

2015年1月30日 回复 顶 转发



iCode

mark

2015年10月7日 回复 顶 转发



陈天石

mark,必须看

3月11日 回复 顶 转发

社交帐号登录: 微信 微博 QQ 人人 更多»



说点什么吧...

发布

<div><p>苹果开发中文站</p></div> <div>网站地图</div> <div>关于我们</div> <div>联系我们</div> <div>合作云平台: 又拍云</div> <div>京公网安备 11010502011183</div> <div>京ICP备 11006519号 京ICP证 100954号</div> <div>Copyright © 2008-2013 CocoaChina.com</div>	<div>资讯频道</div> <div>游戏开发</div> <div>App Store研究</div> <div>iOS开发</div> <div>游戏开发</div> <div>Cocos引擎</div> <div>业界动态</div> <div>产品设计</div> <div>程序人生</div>	<div>开发者论坛</div> <div>论坛</div> <div>技术问答</div> <div>开发者中心</div> <div>代码库</div> <div>工具库</div>	<div>开发者平台</div> <div>开发者平台</div> <div>关注微信 每日推荐</div> <div></div>	<div>关注我们</div> <div>扫一扫 浏览移动版</div> <div></div>
	<div>友情链接</div> <div>Cocos引擎中文官网 cocos2d-x 摩点众筹 美图秀秀 iPhone 版 苹果开发者中心 手游那点事 雷锋网 工程师爸爸 iPad网址导航 麦芽地 Nooidea.com 装傻充愣 啃苹果论坛 苹果fans 苹果发烧友 9RIA天地会 苹果发烧友 泰然网 eoe开发者社区 维以不永伤 远景苹果主题 游戏邦 爱应用 人人都是产品经理 9秒社团 源码天堂 游戏陀螺 推酷网 SegmentFault</div>			