

不曾拥有

每天积累一点点

博客园

首页

博问

闪存

新随笔

联系

订阅

管理

公告

昵称: 不曾拥有
园龄: 3年2个月
粉丝: 21
关注: 0
[+加关注](#)

<	2012年12月						>
日	一	二	三	四	五	六	
25	26	27	28	29	30	1	
2	<u>3</u>	4	5	6	7	8	
9	10	11	12	13	14	15	
16	17	18	19	<u>20</u>	21	22	
23	24	25	26	27	28	29	
30	31	1	2	3	4	5	

搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

随笔分类(44)

[@Selector\(2\)](#)
[ARC相关\(1\)](#)
[ASIHTTPRequest库\(1\)](#)
[AsyncSocket\(1\)](#)
[Cocos2d-x\(4\)](#)
[C指针\(2\)](#)
[DrawingPrintingIOS 学习\(3\)](#)
[DTCoreText\(5\)](#)
[iPad开发基础教程学习笔记\(3\)](#)
[NSRange\(1\)](#)
[RCLabel\(1\)](#)
[UIActionSheet\(1\)](#)
[UIColor\(1\)](#)

随笔-45 文章-24 评论-16

CABasicAnimation用法

CABasicAnimation 自己只有三个property fromValue toValue ByValue

当你创建一个 **CABasicAnimation** 时,你需要通过-**setFromValue** 和-**setToValue** 来指定一个开始值和结束值。当你增加基础动画到层中的时候,它开始运行。当用属性做动画完成时,例如用位置属性做动画,层就会立刻 返回到它的初始位置

记住当你做动画时,你至少使用了 **2** 个对象。这些对象都是层本身,一个层或者层继承的对象,和在先前 的例子中你分配给层的

CABasicAnimation 对象。因为你给动画对象设定了最后的值(目的地),但是并不意味着当动画完成的时候,层的属性就改变成了最后的值。当动画完成时,你必须显示的设定层的属性,这样动画结束后,你的层才能真正的到你设定的属性值上。

你可以简单的停止动画到你结束的点上,但是这仅仅是一个视觉效果。层实际的值仍然是一样的。要真的 改变内部的值,就像刚才所说的你必须显示的设定那个属性。例如,显示的设定位置的属性,你需要在层中调用-**setPosition** 方法。但是,这会造成一点问题。

如果你通过-**set** 这个方法显示的设定了层属性的值,那么默认的动画将被执行,而非之前你设定的动画。在表 3-9 中演示了你设置位置的方法。注意到了,我们使用 **position** 已经创建了基础动画,但是我们在层上显示的调用了-**setPosition** 方法,就覆盖了我们设定的动画,使我们设定的基础动画完全没用了。如果你使用了这个代码,你会看到虽然我们的层结束的时候放到了正确的位置,但是它使用的是默认的 **0.25** 秒,而非我们在 动画里显示设定的 **5** 秒钟。

```
1 CABasicAnimation *animation =  
2 [CABasicAnimation animationWithKeyPath:@"position"]; [  
3 [layer setPosition:endpoint];  
4 [layer addAnimation:animation forKey:nil];
```

因此现在问题出来了,你怎么能使用我们设定的动画呢?看表 3-9 的最后一行,注意到 forKey:这个参数 是被设定为 nil。这就是为什么动画不能覆盖默认动画的原因。如果你改变最后一行为[layer addAnimation:animation forKey:@"position"],动画将会按照我们设定的时间工作。这告诉了层当需要做动画时,使用我们给关键路

UILabel(1)
UITableView(4)
UITextField(2)
XML(1)
核心动画Core Animation(7)
考试学习系统项目文档(2)
数据结构与算法(1)

文章分类(24)

CALayer(2)
iOS(3)
NSDictionary(1)
three20(1)
UIAlertView(1)
UIBezierPath(1)
UIFont(1)
UIImage(7)
UIView(2)
UIViewController(3)
xcode(2)

最新评论

1. Re:利用IOS画图功能画出五角星, 并且可以调整五角星的填充范围
你好, 有代码么发下, 我复制过来没有效果, 58954986@qq.com

--boysimple

2. Re:对自定义UITableViewCell的理解
我觉得bool 应该放在viewDidLoad里面

--怨瑞玛

3. Re:对自定义UITableViewCell的理解
第一种方法加不加if条件判断都会执行, 何必呢?

--怨瑞玛

4. Re:关键帧动画-五角星动画-在层上画出五角星

@羊羊羊你搞懂了吗, 兄弟...

--喷跑的小蜗牛1992

5. Re:关键帧动画-五角星动画-在层上画出五角星

我也不太理解block, 到底定义的时什么, 哪位大神能指点指点吗?

--喷跑的小蜗牛1992

阅读排行榜

径指定的新动画。

下面是一些继承的游泳的属性

Autoreverses

当你设定这个属性为 YES 时,在它到达目的地之后,动画的返回到开始的值,代替了直接跳转到 开始的值。

Duration

Duration 这个参数你已经相当熟悉了。它设定开始值到结束值花费的时间。期间会被速度的属性所影响。RemovedOnCompletion 这个属性默认为 YES,那意味着,在指定的时间段完成后,动画就自动的从层上移除了。这个一般不用。

假如你想要再次用这个动画时,你需要设定这个属性为 NO。这样的话,下次你在通过-set 方法设定动画的属性时,它将再次使用你的动画,而非默认的动画。

Speed

默认的为 1.0.这意味着动画播放按照默认的速度。如果你改变这个值为 2.0,动画会用 2 倍的速度播放。这样的影响就是使持续时间减半。如果你指定的持续时间为 6 秒,速度为 2.0,动画就会播放 3 秒钟---一半的 持续时间。

BeginTime

这个属性在组动画中很有用。它根据父动画组的持续时间,指定了开始播放动画的时间。默认的是 0.0.组 动画在下个段落中讨论“Animation Grouping”。

TimeOffset

如果一个时间偏移量是被设定,动画不会真正的可见,直到根据父动画组中的执行时间得到的时间都流逝了。

RepeatCount

默认的是 0,意味着动画只会播放一次。如果指定一个无限大的重复次数,使用 1e100f。这个不应该和 repeatDuration 属性一块使用。

RepeatDuration

这个属性指定了动画应该被重复多久。动画会一直重复,直到设定的时间流逝完。它不应该和 repeatCount 一起使用。

1. CABasicAnimation用法(18004)
2. 对自定义UITableViewCell的理解(8566)
3. <原>AsyncSocket使用心得(7595)
4. <转>UITableView当数据很少的时候, 去掉多余的cell分割线(3714)
5. <原>DTCoreText学习(一) - DTAttributedTextCell原理(3456)

评论排行榜

1. <原>DTCoreText学习(一) - DTAttributedTextCell原理(3)
2. 关键帧动画 - 五角星动画 - 在层上画出五角星(3)
3. 利用IOS画图功能画出五角星, 并且可以调整五角星的填充范围(2)
4. <原>DTCoreText学习(三) - 自定义DTAttributedTextCell(2)
5. 对自定义UITableViewCell的理解(2)

推荐排行榜

1. Cocos2d-x中的分辨率学习(1)
2. <原>关键帧动画CAKeyframeAnimation(1)

下面这段英文摘自苹果官方文档, 将的是**fromValue toValue ByValue** 如何使用

The interpolation values are used as follows:

- Both fromValue and toValue are non-nil. Interpolates between fromValue and toValue.
- fromValue and byValue are non-nil. Interpolates between fromValue and (fromValue + byValue).
- byValue and toValue are non-nil. Interpolates between (toValue - byValue) and toValue.
- fromValue is non-nil. Interpolates between fromValue and the current presentation value of the property.
- toValue is non-nil. Interpolates between the current value of keyPath in the target layer's presentation layer and toValue.
- byValue is non-nil. Interpolates between the current value of keyPath in the target layer's presentation layer and that value plus byValue.
- All properties are nil. Interpolates between the previous value of keyPath in the target layer's presentation layer and the current value of keyPath in the target layer's presentation layer.

其他的方法 还是属性等 都是继承而来的

我们可以通过animationWithKeyPath键值对的方式来改变动画

animationWithKeyPath的值:

transform.scale = 比例轉換

transform.scale.x = 闊的比例轉換

transform.scale.y = 高的比例轉換

transform.rotation.z = 平面圖的旋轉

opacity = 透明度

margin

zPosition

backgroundColor 背景颜色

cornerRadius 圆角

borderWidth

bounds

contents

contentsRect

cornerRadius

frame

hidden

mask

masksToBounds

opacity

position

shadowColor

shadowOffset

shadowOpacity

shadowRadius

下面是一些例子

```
1 CABasicAnimation *pulse = [CABasicAnimation animation];
2 pulse.timingFunction = [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseInEaseOut];
3 pulse.duration = 0.5 + (rand() % 10) * 0.05;
4 pulse.repeatCount = 1;
5 pulse.autoreverses = YES;
6 pulse.fromValue = [NSNumber numberWithInt:0.8];
7 pulse.toValue = [NSNumber numberWithInt:1.2];
8 [self.ui_View.layer addAnimation:pulse forKey:nil];
9
10 // bounds
11
12 CABasicAnimation *anim = [CABasicAnimation animation];
13 anim.duration = 1.f;
14 anim.fromValue = [NSValue valueWithCGRect:CGRectMake(0, 0, 100, 100)];
15 anim.toValue = [NSValue valueWithCGRect:CGRectMake(50, 50, 150, 150)];
16 anim.byValue = [NSValue valueWithCGRect:CGRectMake(0, 0, 100, 100)];
17 // anim.toValue = (id)[UIColor redColor].CGColor;
18 // anim.fromValue = (id)[UIColor blackColor].CGColor;
19
20 anim.timingFunction = [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseInEaseOut];
```

```
21     anim.repeatCount = 1;
22     anim.autoreverses = YES;
23
24     [ui_View.layer addAnimation:anim forKey:nil];
25     //cornerRadius
26
27     CABasicAnimation *anim2 = [CABasicAnimation anim
28     anim2.duration = 1.f;
29     anim2.fromValue = [NSNumber numberWithFloat:0.f]
30     anim2.toValue = [NSNumber numberWithFloat:20.f];
31     anim2.timingFunction = [CAMediaTimingFunction fu
32     anim2.repeatCount = CGFLOAT_MAX;
33     anim2.autoreverses = YES;
34
35     [ui_View.layer addAnimation:anim2 forKey:@"corne
36
37     //contents
38
39     CABasicAnimation *anim = [CABasicAnimation animation
40     anim.duration = 1.f;
41     anim.fromValue = (id)[UIImage imageNamed:@"1.jpg
42     anim.toValue = (id)[UIImage imageNamed:@"2.png"]
43     //     anim.byValue = (id)[UIImage imageNamed:@"3.pn
44     //     anim.toValue = (id)[UIColor redColor].CGColor;
45     //     anim.fromValue = (id)[UIColor blackColor].CGColor
46
47     anim.timingFunction = [CAMediaTimingFunction fun
48     anim.repeatCount = CGFLOAT_MAX;
49     anim.autoreverses = YES;
50
51     [ui_View.layer addAnimation:anim forKey:nil];
52
53
54
55     [ui_View.layer setShadowOffset:CGSizeMake(2,2)];
56     [ui_View.layer setShadowOpacity:1];
57     [ui_View.layer setShadowColor:[UIColor grayColor
58     //
59     CABasicAnimation *anim = [CABasicAnimation anima
60     anim.duration = 1.f;
61     anim.toValue = (id)[UIColor redColor].CGColor;
62     anim.fromValue = (id)[UIColor blackColor].CGCol
63
64     anim.timingFunction = [CAMediaTimingFunction fun
65     anim.repeatCount = CGFLOAT_MAX;
66     anim.autoreverses = YES;
67
68     [ui_View.layer addAnimation:anim forKey:nil];
69
70     CABasicAnimation *_anim = [CABasicAnimation anim
```

```
71     _anim.duration = 1.f;
72     _anim.fromValue = [NSValue valueWithCGSize:CGSize
73     _anim.toValue = [NSValue valueWithCGSize:CGSizeM
74
75     _anim.timingFunction = [CAMediaTimingFunction fu
76     _anim.repeatCount = CGFLOAT_MAX;
77     _anim.autoreverses = YES;
78
79     [ui_View.layer addAnimation:_anim forKey:nil];
80
81
82     CABasicAnimation *_anim1 = [CABasicAnimation ani
83     _anim1.duration = 1.f;
84     _anim1.fromValue = [NSNumber numberWithFloat:0.5
85     _anim1.toValue = [NSNumber numberWithFloat:1];
86
87     _anim1.timingFunction = [CAMediaTimingFunction f
88     _anim1.repeatCount = CGFLOAT_MAX;
89     _anim1.autoreverses = YES;
90
91     [ui_View.layer addAnimation:_anim1 forKey:nil];
92
93
94
95     CABasicAnimation *_anim2 = [CABasicAnimation ani
96     _anim2.duration = 1.f;
97     _anim2.fromValue = [NSNumber numberWithFloat:10]
98     _anim2.toValue = [NSNumber numberWithFloat:5];
99
100    _anim2.timingFunction = [CAMediaTimingFunction f
101    _anim2.repeatCount = CGFLOAT_MAX;
102    _anim2.autoreverses = YES;
103
104    [ui_View.layer addAnimation:_anim2 forKey:nil];
```

下面是一些应用

几个可以用来实现热门APP应用PATH中menu效果的几个方法

```
+(CABasicAnimation *)opacityForever_Animation:(float)time
// 永久闪烁的动画
```

```
{
```

```
    CABasicAnimation *animation=[CABasicAnimation
    animationWithKeyPath:@"opacity"];
```

```
    animation.fromValue=[NSNumber numberWithFloat:1.0];
```

```
        animation.toValue=[NSNumber numberWithFloat:0.0];

        animation.autoreverses=YES;

        animation.duration=time;

        animation.repeatCount=FLT_MAX;

        animation.removedOnCompletion=NO;

        animation.fillMode=kCAFillModeForwards;

        return animation;

    }

+ (CABasicAnimation *)opacityTimes_Animation:
(float)repeatTimes durTimes:(float)time; // 有闪烁次数的动
画

{

    CABasicAnimation *animation=[CABasicAnimation
animationWithKeyPath:@"opacity"];

    animation.fromValue=[NSNumber numberWithFloat:1.0];

    animation.toValue=[NSNumber numberWithFloat:0.4];

    animation.repeatCount=repeatTimes;

    animation.duration=time;

    animation.removedOnCompletion=NO;

    animation.fillMode=kCAFillModeForwards;

    animation.timingFunction=[CAMediaTimingFunction
functionWithName:kCAMediaTimingFunctionEaseIn];

    animation.autoreverses=YES;

    return animation;

}
```

```
+(CABasicAnimation *)moveX:(float)time X:(NSNumber *)x /
/ 横向移动

{

    CABasicAnimation *animation=[CABasicAnimation
animationWithKeyPath:@"transform.translation.x"];

    animation.toValue=x;

    animation.duration=time;

    animation.removedOnCompletion=NO;

    animation.fillMode=kCAFillModeForwards;

    return animation;

}

+(CABasicAnimation *)moveY:(float)time Y:(NSNumber *)y /
/ 纵向移动

{

    CABasicAnimation *animation=[CABasicAnimation
animationWithKeyPath:@"transform.translation.y"];

    animation.toValue=y;

    animation.duration=time;

    animation.removedOnCompletion=NO;

    animation.fillMode=kCAFillModeForwards;

    return animation;

}

+(CABasicAnimation *)scale:(NSNumber *)Multiple origin:
(NSNumber *)originMultiple durTimes:(float)time Rep:
(float)repeatTimes / / 缩放

{
```



```
        CABasicAnimation *animation=[CABasicAnimation
animationWithKeyPath:@"transform.scale"];

        animation.fromValue=originMultiple;

        animation.toValue=Multiple;

        animation.duration=time;

        animation.autoreverses=YES;

        animation.repeatCount=repeatTimes;

        animation.removedOnCompletion=NO;

        animation.fillMode=kCAFillModeForwards;

        return animation;
    }

+ (CAAnimationGroup *)groupAnimation: (NSArray
*)animationAry durTimes:(float)time Rep:
(float)repeatTimes // 组合动画

{

    CAAnimationGroup *animation=[CAAnimationGroup
animation];

    animation.animations=animationAry;

    animation.duration=time;

    animation.repeatCount=repeatTimes;

    animation.removedOnCompletion=NO;

    animation.fillMode=kCAFillModeForwards;

    return animation;
}

+ (CAKeyframeAnimation *)keyframeAniamtion:
```

```
(CGMutablePathRef)path durTimes:(float)time Rep:
(float)repeatTimes // 路径动画

{

    CAKeyframeAnimation *animation=[CAKeyframeAnimation
animationWithKeyPath:@"position"];

    animation.path=path;

    animation.removedOnCompletion=NO;

    animation.fillMode=kCAFillModeForwards;

    animation.timingFunction=[CAMediaTimingFunction
functionWithName:kCAMediaTimingFunctionEaseIn];

    animation.autoreverses=NO;

    animation.duration=time;

    animation.repeatCount=repeatTimes;

    return animation;

}

+(CABasicAnimation *)movepoint:(CGPoint )point // 点移动

{

    CABasicAnimation *animation=[CABasicAnimation
animationWithKeyPath:@"transform.translation"];

    animation.toValue=[NSValue valueWithCGPoint:point];

    animation.removedOnCompletion=NO;

    animation.fillMode=kCAFillModeForwards;

    return animation;

}

+(CABasicAnimation *)rotation:(float)dur degree:
(float)degree direction:(int)direction repeatCount:
```

```
(int)repeatCount // 旋转

{

    CATransform3D rotationTransform =
    CATransform3DMakeRotation(degree, 0, 0,direction);

    CABasicAnimation* animation;

    animation = [CABasicAnimation
    animationWithKeyPath:@"transform"];

    animation.toValue= [NSValue
    valueWithCATransform3D:rotationTransform];

    animation.duration= dur;

    animation.autoreverses= NO;

    animation.cumulative= YES;

    animation.removedOnCompletion=NO;

    animation.fillMode=kCAFillModeForwards;

    animation.repeatCount= repeatCount;

    animation.delegate= self;

    return animation;

}
```

分类: [核心动画Core Animation](#)

好文要顶

关注我

收藏该文



不曾拥有

关注 - 0

粉丝 - 21

[+加关注](#)

0

0

(请您对文章做出评价)

« 上一篇: [<转>iOS SDK中使用NSXMLParser解析XML](#)

» 下一篇: [CALayer 以及时间模型<转>](#)

posted @ 2012-12-20 16:22 不曾拥有 阅读(18004) 评论(0) 编辑
收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】融云即时通讯云—专注为 App 开发者提供IM云服务

【福利】极光推送-富媒体+大数据驱动精准推送, 完全免费开放啦



最新IT新闻:

- 徐小平: 三个月内就能离职的一定不是牛人
 - OneNote 11月大更新 iOS版加入Force Touch
 - ASP.NET 5开发者的五个阶段
 - 马化腾vs马云: 微信干翻新浪微博后又要将支付宝拉下王座?
 - 联想6年来首次亏损 移动业务或带其走向复苏
- » 更多新闻...

【视频】0基础1个月快速入门HTML5开发

HTML、CSS、JS、jQuery、Bootstrap、Egret全面学习, 一块补差!



最新知识库文章:

- 被误解的MVC和被神化的MVVM
 - 再谈设计和编码
 - 什么时候应该避免写代码注释?
 - 持续集成是什么?
 - 人, 技术与流程
- » 更多知识库文章...

