**iOS开发UI篇—核心动画(关键帧动画)**

**一、简单介绍**

是CApropertyAnimation的子类，跟CABasicAnimation的区别是：CABasicAnimation只能从一个数值(fromValue)变到另一个数值(toValue)，而CAKeyframeAnimation会使用一个NSArray保存这些数值

属性解析：

values：就是上述的NSArray对象。里面的元素称为”关键帧”(keyframe)。动画对象会在指定的时间(duration)内，依次显示values数组中的每一个关键帧

path：可以设置一个CGPathRef\CGMutablePathRef,让层跟着路径移动。path只对CALayer的anchorPoint和position起作用。如果你设置了path，那么values将被忽略

keyTimes：可以为对应的关键帧指定对应的时间点,其取值范围为0到1.0,keyTimes中的每一个时间值都对应values中的每一帧.当keyTimes没有设置的时候,各个关键帧的时间是平分的

说明：CABasicAnimation可看做是最多只有2个关键帧的CAKeyframeAnimation

**二、代码示例**

第一种方式：

代码：

[](javascript:void(0);)

1 //

2 // YYViewController.m

3 // 10-核心动画（关键帧动画1）

4 //

5 // Created by apple on 14-6-21.

6 // Copyright (c) 2014年 itcase. All rights reserved.

7 //

8

9 #import "YYViewController.h"

10

11 @interface YYViewController ()

12 @property (weak, nonatomic) IBOutlet UIView \*customView;

13

14 @end

15

16 @implementation YYViewController

17

18

19 -(void)touchesBegan:(NSSet \*)touches withEvent:(UIEvent \*)event

20 {

21 //1.创建核心动画

22 CAKeyframeAnimation \*keyAnima=[CAKeyframeAnimation animation];

23 //平移

24 keyAnima.keyPath=@"position";

25 //1.1告诉系统要执行什么动画

26 NSValue \*value1=[NSValue valueWithCGPoint:CGPointMake(100, 100)];

27 NSValue \*value2=[NSValue valueWithCGPoint:CGPointMake(200, 100)];

28 NSValue \*value3=[NSValue valueWithCGPoint:CGPointMake(200, 200)];

29 NSValue \*value4=[NSValue valueWithCGPoint:CGPointMake(100, 200)];

30 NSValue \*value5=[NSValue valueWithCGPoint:CGPointMake(100, 100)];

31 keyAnima.values=@[value1,value2,value3,value4,value5];

32 //1.2设置动画执行完毕后，不删除动画

33 keyAnima.removedOnCompletion=NO;

34 //1.3设置保存动画的最新状态

35 keyAnima.fillMode=kCAFillModeForwards;

36 //1.4设置动画执行的时间

37 keyAnima.duration=4.0;

38 //1.5设置动画的节奏

39 keyAnima.timingFunction=[CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseInEaseOut];

40

41 //设置代理，开始—结束

42 keyAnima.delegate=self;

43 //2.添加核心动画

44 [self.customView.layer addAnimation:keyAnima forKey:nil];

45 }

46

47 -(void)animationDidStart:(CAAnimation \*)anim

48 {

49 NSLog(@"开始动画");

50 }

51

52 -(void)animationDidStop:(CAAnimation \*)anim finished:(BOOL)flag

53 {

54 NSLog(@"结束动画");

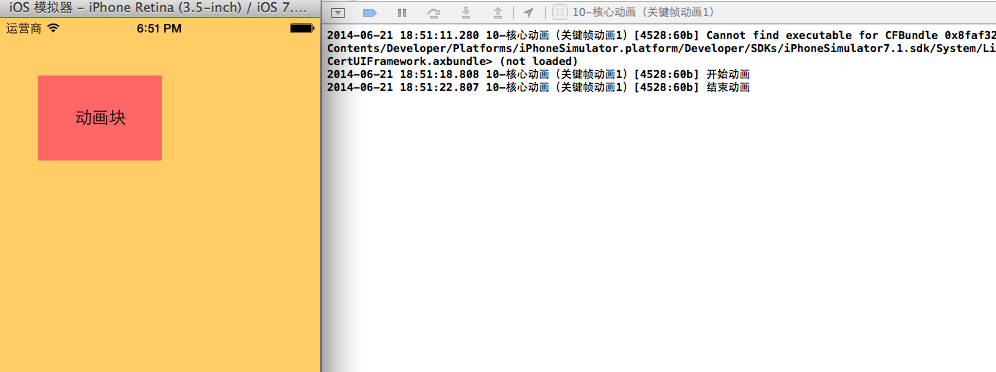
55 }

56 @end

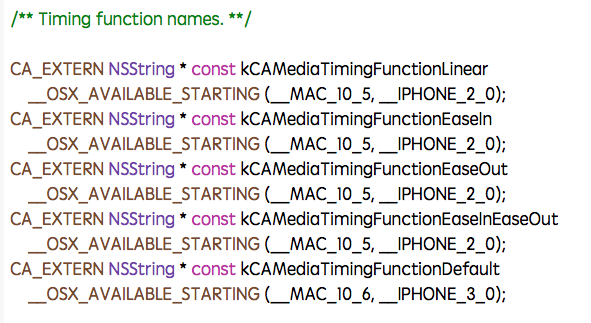
[](javascript:void(0);)

说明：这个项目在storyboard中拖入了一个view，并和控制器中的custom进行了关联。

效果和打印结果：



补充：设置动画的节奏



第二种方式（使用path）让layer在指定的路径上移动（画圆）：

代码：

[](javascript:void(0);)

1 #import "YYViewController.h"

2

3 @interface YYViewController ()

4 @property (weak, nonatomic) IBOutlet UIView \*customView;

5

6 @end

7

8 @implementation YYViewController

9

10

11 -(void)touchesBegan:(NSSet \*)touches withEvent:(UIEvent \*)event

12 {

13 //1.创建核心动画

14 CAKeyframeAnimation \*keyAnima=[CAKeyframeAnimation animation];

15 //平移

16 keyAnima.keyPath=@"position";

17 //1.1告诉系统要执行什么动画

18 //创建一条路径

19 CGMutablePathRef path=CGPathCreateMutable();

20 //设置一个圆的路径

21 CGPathAddEllipseInRect(path, NULL, CGRectMake(150, 100, 100, 100));

22 keyAnima.path=path;

23

24 //有create就一定要有release

25 CGPathRelease(path);

26 //1.2设置动画执行完毕后，不删除动画

27 keyAnima.removedOnCompletion=NO;

28 //1.3设置保存动画的最新状态

29 keyAnima.fillMode=kCAFillModeForwards;

30 //1.4设置动画执行的时间

31 keyAnima.duration=5.0;

32 //1.5设置动画的节奏

33 keyAnima.timingFunction=[CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseInEaseOut];

34

35 //设置代理，开始—结束

36 keyAnima.delegate=self;

37 //2.添加核心动画

38 [self.customView.layer addAnimation:keyAnima forKey:nil];

39 }

40

41 -(void)animationDidStart:(CAAnimation \*)anim

42 {

43 NSLog(@"开始动画");

44 }

45

46 -(void)animationDidStop:(CAAnimation \*)anim finished:(BOOL)flag

47 {

48 NSLog(@"结束动画");

49 }

50 @end

[](javascript:void(0);)

说明：可以通过path属性，让layer在指定的轨迹上运动。

停止动画：

[](javascript:void(0);)

1 #import "YYViewController.h"

2

3 @interface YYViewController ()

4 @property (weak, nonatomic) IBOutlet UIView \*customView;

5 - (IBAction)stopOnClick:(UIButton \*)sender;

6

7 @end

8

9 @implementation YYViewController

10

11

12 -(void)touchesBegan:(NSSet \*)touches withEvent:(UIEvent \*)event

13 {

14 //1.创建核心动画

15 CAKeyframeAnimation \*keyAnima=[CAKeyframeAnimation animation];

16 //平移

17 keyAnima.keyPath=@"position";

18 //1.1告诉系统要执行什么动画

19 //创建一条路径

20 CGMutablePathRef path=CGPathCreateMutable();

21 //设置一个圆的路径

22 CGPathAddEllipseInRect(path, NULL, CGRectMake(150, 100, 100, 100));

23 keyAnima.path=path;

24

25 //有create就一定要有release

26 CGPathRelease(path);

27 //1.2设置动画执行完毕后，不删除动画

28 keyAnima.removedOnCompletion=NO;

29 //1.3设置保存动画的最新状态

30 keyAnima.fillMode=kCAFillModeForwards;

31 //1.4设置动画执行的时间

32 keyAnima.duration=5.0;

33 //1.5设置动画的节奏

34 keyAnima.timingFunction=[CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseInEaseOut];

35

36 //2.添加核心动画

37 [self.customView.layer addAnimation:keyAnima forKey:@"wendingding"];

38 }

39

40 - (IBAction)stopOnClick:(UIButton \*)sender {

41 //停止self.customView.layer上名称标示为wendingding的动画

42 [self.customView.layer removeAnimationForKey:@"wendingding"];

43 }

44 @end

[](javascript:void(0);)



点击停止动画，程序内部会调用  [self.customView.layer removeAnimationForKey:@"wendingding"];停止self.customView.layer上名称标示为wendingding的动画。

**三、图标抖动**

**代码示例：**

[](javascript:void(0);)

1 //

2 // YYViewController.m

3 // 12-图标抖动

4 //

5 // Created by apple on 14-6-21.

6 // Copyright (c) 2014年 itcase. All rights reserved.

7 //

8

9 #import "YYViewController.h"

10 #define angle2Radian(angle) ((angle)/180.0\*M\_PI)

11

12 @interface YYViewController ()

13 @property (weak, nonatomic) IBOutlet UIImageView \*iconView;

14

15 @end

16

17

18 @implementation YYViewController

19

20 -(void)touchesBegan:(NSSet \*)touches withEvent:(UIEvent \*)event

21 {

22 //1.创建核心动画

23 CAKeyframeAnimation \*keyAnima=[CAKeyframeAnimation animation];

24 keyAnima.keyPath=@"transform.rotation";

25 //设置动画时间

26 keyAnima.duration=0.1;

27 //设置图标抖动弧度

28 //把度数转换为弧度 度数/180\*M\_PI

29 keyAnima.values=@[@(-angle2Radian(4)),@(angle2Radian(4)),@(-angle2Radian(4))];

30 //设置动画的重复次数(设置为最大值)

31 keyAnima.repeatCount=MAXFLOAT;

32

33 keyAnima.fillMode=kCAFillModeForwards;

34 keyAnima.removedOnCompletion=NO;

35 //2.添加动画

36 [self.iconView.layer addAnimation:keyAnima forKey:nil];

37 }

38

39 @end

[](javascript:void(0);)

说明：图标向左向右偏转一个弧度（4），产生抖动的视觉效果。

程序界面：

