layoutSub

DCSnail-杨牛 关注
2020-02-02 09:24:00 4770 收藏 1

文章标签： drawRect layoutSub 曲线绘制 曲线绘制

探究前了解

首先理解自己要绘制曲线的核心文件时对应的是这一个文件，是关于layoutSubviews和drawRect的触发时机的，今天,就根据这篇文章来做一些测试,这是一些文章中的结论。
请先以下配方全动起来了解自己个人复习：

一、layoutSubviews在以下情况下会被调用：
1. init初始化不会触发layoutSubviews。
2. addSubview会触发layoutSubviews。
3. 改变一个UIView的Frame会触发layoutSubviews，当然即便是frame的值设置前后发生了变化。
4. 滚动一个UiScrollView引起UIView的重新布局会触发layoutSubviews。
5. 旋转Screen会触发父UIView上的layoutSubviews事件。
6. 直接调用setNeedsLayout 或者 layoutIfNeeded。

二、drawRect在以下情况下会被调用：
1. 如果在UIView初始化时没有设置size大小，将直接导致drawRect不被自动调用。drawRect 被用来在Controller->loadView, Controller->viewDidLoad两个方法之间做调用,所以不用担心在视图控制器中,因为设置view的frame和drawRect就开始画了,这样可以在控制器中设置一些数据给View,而不会在drawRect被重复调用的前提下。
2. 依方法名调用setNeedsDisplay时,所以可以发现调用setNeedsDisplay时,即会系统自动调用drawRect方法。
3. 通过设置contentMode属性值为UIViewContentModeRedraw，那么将在每次设置或更改frame的时候自动调用drawRect。
4. 直接调用sizeToFit()，或者setNeedsDisplayInRect触发drawRect，但是有个前提条件是rect不能为0。

以上这些是：四大时机。

三、drawRect方法使用注意点：
1. 若使用UIKit框架，只能在drawRect 方法中获取相应的contextRef并绘图，如果在其他方法中获取将获取到一个invalidate的ref并且不能用于画图。drawRect 方法不是在主线程上调用，这通过调用setNeedsDisplay 或 setNeedsDisplayInRect 以及其自身调该方法。
2. 若使用coreGraphics框架，只能在drawInContext:中（或在于drawRect 过程），或者在delegate中的相应方法绘制。同样也是调用setNeedsDisplay等间接调用以上方法
3. 若需实时画图，不建议使用gestureRecognizer，只能使用touchbegan等方法来调用setNeedsDisplay实时刷新屏幕

探究

layoutSubviews调用时机测试

```
#import <UIKit/UIKit.h>

@interface TestView : UIView

@end
```

```
#import "TestView.h"

@implementation TestView

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self)
    {
        NSLog(@"initWithFrame:%@",_NSStringFromCGRect(frame));
    }
    return self;
}

- (void)layoutSubviews
{
    NSLog(@"layoutSubviews %@", self);
    [super layoutSubviews];
}

@end
```

```
#import "RootViewController.h"
#import "TestView.h"

@interface RootViewController ()

{

@property (nonatomic, strong) NSTimer    *timer;
@property (nonatomic, strong) TestView   *largeView;
@property (nonatomic, strong) TestView   *smallView;

@end

@implementation RootViewController


- (void)viewDidLoad
{
    [super viewDidLoad];

    // 1. init初始化不会触发layoutSubviews 【正确的】
    // 2. addSubview会触发layoutSubviews 【不完全正确,当frame为初值不会触发的】
    // 3. 改变一个UIVIEW的frame 会 发生变化调用layoutSubviews 【正确】

    //  [self test_1];
    //  [self test_2];
    //  [self test_3];

    // 4. 滚动一个UiScrollView会触发了layoutSubviews(错误,不用滚动也触发)
    //  [self test_4];

    // 5. 改变一个UIVIEW的时候也会触发父UIView上的layoutSubviews事件
    //  [self test_5];
}

- (void)test_1
{
    /*
    解释:

    在[initWithFrame:]方法,设置父亲Frame宽为(0,0), (0, 0),长宽都设置为任何的东西,
    所以即便设置了test,也没必要调用时,同时引起了addSubview调用触发layoutSubviews原理
    调用,只有当解决掉frame为零情况下的时候才会触发layoutSubviews
    */

    TestView *test = [TestView new];
    [self.view addSubview:test];
}

- (void)test_2
{
    TestView *test = [TestView new];
    test.frame = CGRectMake(0, 0, 100, 100);
    [self.view addSubview:test];
}

- (void)test_3
{
    /*
    解释:

    layoutSubviews这个方法会每天调用,要被父类告知的时候才执行的方法
    */

    TestView *test = [TestView new];
    test.frame = CGRectMake(0, 0, 50, 50);
    UIView *myNewView = [[UIView alloc] initWithFrame:CGRectMake(0, 0, 50, 50)];
    [test addSubview:myNewView];
}

- (void)test_4
{
    CGRect rect = self.view.bounds;
    CGFloat height = rect.size.height;
    CGFloat width = rect.size.width;

    UIScrollView *rootScroll = [[UIScrollView alloc] initWithFrame:self.view.bounds];
    NSArray *data          = @[@"", @"", @"", @""];
    [data enumerateObjectsUsingBlock:^(id obj, NSUInteger idx, BOOL *stop) {
        TestView *tmp      = [[TestView alloc] initWithFrame:CGRectMake(width*idx, 0,
                                                                        width, height)];
        [rootScroll addSubview:tmp];
    }];
    rootScroll.contentSize = CGSizeMakeWidth * data.count, height);
    [self.view addSubview:rootScroll];
}

- (void)test_5
{
    _timer = [NSTimer scheduledTimerWithTimeInterval:1.f
                                             target:self
                                           selector:@selector(timerEvent:)
                                           userInfo:nil
                                            repeats:YES];
    _largeView = [[TestView alloc] initWithFrame:self.view.bounds];
    [self.view addSubview:_largeView];

    _smallView = [[TestView alloc] initWithFrame:CGRectMake(0, 0, 100, 100)];
    [_largeView addSubview:_smallView];
}

- (void)timerEvent:(id)sender
{
    _smallView.frame = CGRectMake(arc4random()%100 + 20,
                                  arc4random()%100 + 20,
                                  arc4random()%100 + 20,
                                  arc4random()%100 + 20);
    NSLog(@"_smallView %@", _smallView);
    NSLog(@"_smallView %@", _largeView);
}

@end
```

官方文档中介绍：

> You should not call this method directly. If you want to force a layout update, call the setNeedsLayout method instead to do so prior to the next drawing update. If you want to update the layout of your views immediately, call the layoutIfNeeded method.

layoutSubviews 测试后的结论：

1.一个view基于能够自己调用layoutSubviews,如果要调用,需要调用 setNeedsLayout或者 layoutIfNeeded
2.如果view的frame为零,即便设置addSubview也不会调用layoutSubviews
3.如果一个view的frame非零size值后后发生了改变,那么layoutSubviews也会被触发。重新设置 frame 但 size 不变的话,是不会触发的

drawRect调用时机测试

```
#import <UIKit/UIKit.h>

@interface AView : UIView

@end
```

```
#import "AView.h"

@implementation AView

- (void)drawRect:(CGRect)rect {
    // Drawing code

    NSLog(@"DrawRect");

}

- (void)layoutSubviews
{
    NSLog(@"layoutSubviews");
}

@end
```

```
#import "ViewController.h"
#import "AView.h"
@interface ViewController ()
@property (nonatomic, strongWview *wv;
@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    self.v = [[AView alloc]initWithFrame:CGRectMake(0, 0, 300, 200)];
    self.v.backgroundColor = [UIColor greenColor];
    [self.view addSubview:_v];

}

- (void)viewWillAppear:(BOOL)animated
{
}
- (void)viewDidAppear:(BOOL)animated
{
}
- (void)viewDidDisappear:(BOOL)animated
{
}
- (IBAction)click:(id)sender {
//    self.v.backgroundColor = [UIColor redColor];
    self.v.frame = CGRectMake(50, 10, 200, 200);
//    [self.v setNeedsLayout];
}

@end
```

drawRect 测试后的结论:

1.如果在UIView初始化时没有设置size大小，将直接导致drawRect不被自动调用。drawRect 调用是在Controller->viewWillAppear, Controller->viewDidAppear 两方法之间调用的,所以这样可以在控制器中设置一些曲线View drawRt时候重要用到某些变量值。
2.直接调用setNeedsDisplay，或者setNeedsDisplayInRect触发drawRect，但是有个前提条件是frame的size不能为0
3.当 setSubview 时,frame 中的 size 为 0时,不会触发 drawRect 方法
4.改变 view 的 size 时时候,不会触发 drawRect,但是要改其颜色变化的时候,会触发 drawRect
5.当一个View第一次显示或当一个事件发生，该事件导致view的可视部分发现时，才会被调用。

drawRect官方文档:

> The default implementation of this method does nothing. Subclasses that use technologies such as Core Graphics and UIKit to draw their view's content should override this method and implement their drawing code there. You do not need to override this method if your view sets its content in other ways. For example, you do not need to override this method if your view just displays a background color or if your view sets its content directly using the underlying layer object.

该方法的默认实现并不会做任何事情。子类使用诸如Core Graphics和UIKit技术绘制其视图内的内容应该重写该方法，并且把实现的代码写在该方法中。如果你程序的内容是通过其他方式设置的，那么你不需要重写该方法。例如，如果你的程序仅仅只设置一个背景颜色，那不需要重写该方法或者你的程序内容是直接使用layer对象设置的，也不需要调用该方法。

> By the time this method is called, UIKit has configured the drawing environment appropriately for your view and you can simply call whatever drawing methods and functions you need to render your content. Specifically, UIKit creates a graphics context for drawing and adjusts the transform of that context so that its origin matches the origin of your view's bounds rectangle. You can get a reference to the graphics context using the UIGraphicsGetCurrentContext function, but do not establish a strong reference to the graphics context because it can change between calls to the drawRect: method.

当调用该方法时，UIKit框架已经为你的绘图环境设置好适合的视图环境，你可以轻松的调用任何相关方法和函数来渲染你的程序内容。特别地，UIKit会创建用于绘图的一个图形上下文，然后调整该上下文的形变，以使该图形上下文的原点与你程序的bounds的原点相匹配。你可以通过使用UIGraphicsGetCurrentContext函数获得该图形上下文的引用，但是千萬不要在调用这上下文期间保持对该图形上下文的强引用，因为上下文会改变。

> If you subclass UIView directly, your implementation of this method does not need to call super. However, if you are subclassing a different view class, you should call super at some point in your implementation.

如果你直接继承UIView的子类，该方法的实现不需要调用super。但是，如果你创建的是不同的view类型，你需要在实现代码的某个时候调用super。

> This method is called when a view first displayed or when an event occurs that invalidates a visible part of the view. You should never call this method directly yourself. To invalidate part of your view, and thus cause that portion to be redrawn, call the setNeedsDisplay or setNeedsDisplayInRect: method instead.

当一个View第一次显示或当一个事件发生，该事件导致View的可视部分无效时，就会调用。永远不要手动调用该方法。为了使控件的某个部分无效，并且因此导致其不部分重绘，应该调用setNeedsDisplay或者setNeedsDisplayInRect:方法来代替。

参考资料：
When is layoutSubviews called?

打赏

文章目录

- KVO的实现
- 在drawRect中绘制曲线时的layoutSubviews
- 测试后的结论:
- drawRect调用时机测试
- drawRect 测试后的结论:
- drawRect官方文档:
- 相关推荐

打赏 评论
文章目录 打赏鼓励作者一下

KVO的实现 792
在drawRect中绘制曲线时的layoutSubviews 1k+
测试后的结论: 921
drawRect调用时机测试 1k+
drawRect 测试后的结论: 1k+
drawRect官方文档: 1k+
drawRect 1k+
相关推荐
UIView的layoutSubviews和drawRect方法何时调用,学习... 8-14
最为两方法最最原始的,layoutSubviews方法在初始时,是drawRect方法被调用最始。layoutSubviews以下情况下会被调用：1. init初始化不会触发layoutSubviews。2. addSubview会触发layoutSubviews。3. 改变一个UIVIEW的frame 会 发生变化调用layoutSubviews 【正确】
关于layoutSubviews和drawRect的区别 anshanxiu2009的博客 ©1.3k
layoutSubviews 在调用的最最多 (1) 在 init initWithFrame: 方法初始时,并且frame不为 CGRectZero的是触发 (2) addSubview的时候调用。
iOS开发之drawRect的作用和应用机制 maoxiaokuo的博客 ©1k+
drawRect 与 layoutSubviews的区别时机 ,iOS开发之drawRect的作用和应用机制,可以用来自己绘图一个UIview的视图。重要写一个扩展UIView的类，并重写drawRect方法，在drawRect方法中,调用UIKit的一系列的绘图函数,即可在UIVIew上画出自己想要的图形。 ,drawRect 与 layoutSubviews 的区别时机。 2019年晚秋至冬重温iOS第二篇用机器学习... ,drawRect方法中,CGRectZero是一个宽和高为0的矩形,所以drawRect方法不会被调用。 ,layoutSubviews在以下情况下会被调用：1. init初始化不会触发layoutSubviews。 ,layoutSubviews 和 drawRect 的区别 ,layoutSubviews 和 drawRect 的区别 ,layoutSubviews 和 drawRect 的区别