

sakulafly的专栏

目录视图

摘要视图

RSS 订阅

个人资料



williamzhang_zz

访问: 223874次

积分: 3249

等级: BLOG > 5

排名: 第4255名

原创: 65篇 转载: 6篇

译文: 4篇 评论: 41条

文章搜索

文章分类

[android杂项 \(2\)](#)[编辑器 \(4\)](#)[iOS开发 \(65\)](#)[GStreamer \(32\)](#)

文章存档

[2015年01月 \(1\)](#)[2014年11月 \(3\)](#)[2014年10月 \(1\)](#)[2014年09月 \(5\)](#)[2014年08月 \(1\)](#)

展开

阅读排行

[iOS真机UI调试利器——I](#)

(11315)

[iOS6和iOS7代码的适配\(](#)

(9954)

[Ubuntu下Sublime Text 2](#)

(8841)

[iOS6和iOS7代码的适配\(](#)

(6769)

[CSDN学院讲师招募, 诚邀您加入!](#)[博客Markdown编辑器上线啦](#)[PMBOK第五版精讲视频教程](#)[火星人敏捷开发1001问](#)

iOS的事件分发

分类: [iOS开发](#)

2014-01-25 20:29

2150人阅读

评论(0)

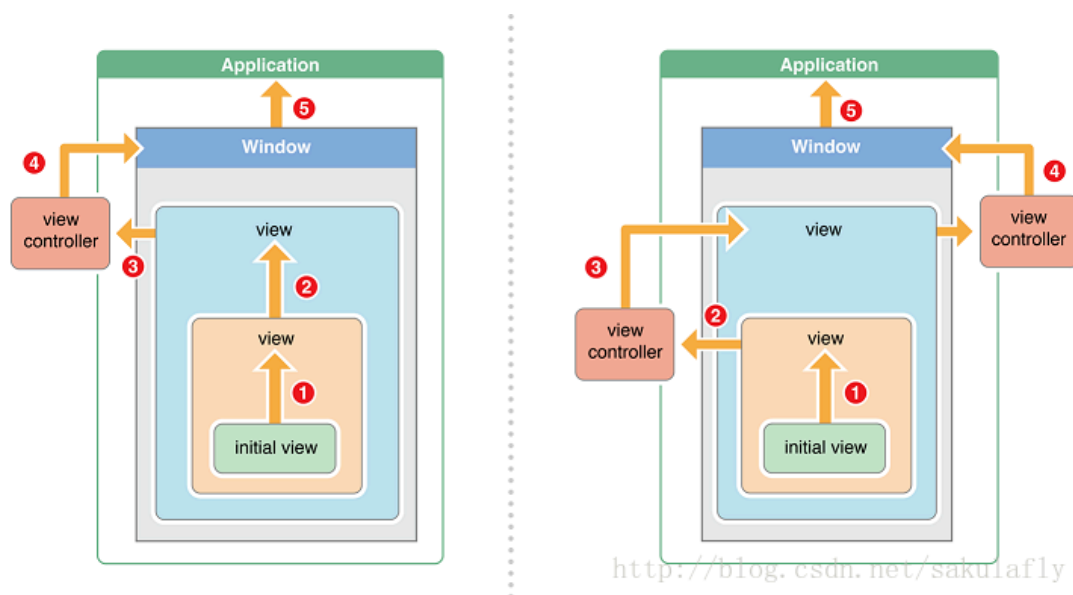
收藏

举报

iOS

移动平台上的开发主要关注数据以及数据的处理, 事件的处理以及UI。所以事件的分发处理是很重要的一个环节, 对于一个平台的优劣来说也是一项重要的参数。如果事件的分发设计的不好, 一些复杂的UI场景就会变得很难写甚至没法写。从小屏没有触摸的功能机开始到现在大屏多点触摸的智能机, 对于事件的分发处理基本思路都是一样的——链(设计模式中有个模式就是职责链chain of responsibility), 只是判定的复杂程度不同。

iOS中的事件有3类, 触摸事件(单点, 多点, 手势)、传感器事件(加速度传感器)和远程控制事件, 这里我介绍的是第一种事件的分发处理。



上面的这张图来自苹果的官方。描述了Responder的链, 同时也是事件处理的顺序。通过这两张图, 我们可以发现:

1. 事件顺着responder chain传递, 如果一环不处理, 则传递到下一环, 如果都没有处理, 最后回到UIApplication, 再不处理就会抛弃
2. view的下一级是包含它的viewController, 如果没有viewController则是它的superView
3. viewController的下一级是它的view的superView
4. view之后是window, 最后传给application, 这点iOS会比OS X简单(application就一个, window也一个)

iOS6和iOS7代码的适配(6276)
iOS6和iOS7代码的适配(6116)
iOS上应用如何兼容32位 (5871)
iOS下KVO的使用以及一 (5681)
android源码/内核下载 (4964)
在iOS中如何设置默认语 (4684)

评论排行

iOS真机UI调试利器—— (11)
Android Ubuntu平台下AI (7)
GStreamer基础教程10—— (4)
GStreamer基础教程11—— (3)
深入Blocks分析 (2)
iOS中Blocks的介绍 (2)
GStreamer iOS教程4—— (2)
xcode上整理字符串资源 (2)
iOS中如何呼出另一个应 (2)
C语言 (2)

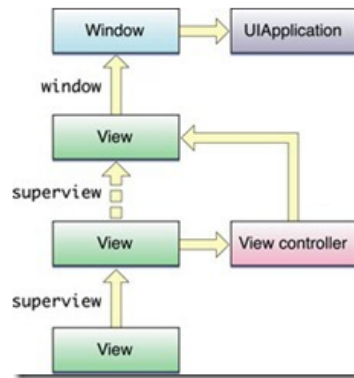
推荐文章

* 【ShaderToy】开篇
* FFmpeg源代码简单分析:
avio_open2()
* 技能树之旅: 从模块分离到测试
* Qt5官方demo解析集36——
Wiggly Example
* Unity3d HDR和Bloom效果 (高
动态范围图像和泛光)
* Android的Google官方设计指南

最新评论

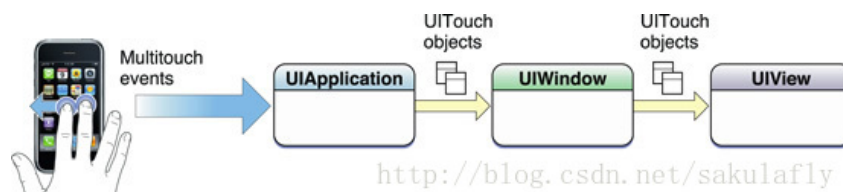
iOS中Blocks的介绍
巾令页: 讲的挺详细的!
iOS运行时工具-cycript
u010943824: 执行 ./cycript -r
127.0.0.1:8888出现以下错误:
*** _syscall(...
Android Ubuntu平台下ADB驱动
hejiehejiehe: 大哥, 我什么lsusb
找不到设备呀, 我开了调试模式
实战iOS应用从32位升级到64位
simba_nana: 你好, 怎么在cmd
模式下用file命令检查库文件是否
支持64位呢, 谢谢
GStreamer基础教程02——GStre
tomorrow_opal: customize翻译
成定制更好些吧?
如何使用TestFlight进行Beta测试
qq_14962031: 邀请我, 谢谢
iOS8.1.2yuanchao521.521@QQ.cc
深入Blocks分析
软件攻城师: 抄袭书上的
GStreamer基础教程13——播放
刘二毛: 楼主, 我用的playbin2,
确实seek, 回放时画面会卡住不
动, 但声音依然有, 请教楼主“回
放是很多...
iOS中如何呼出另一个应用
菜鸟_工程师: 哥们 你好 你这个
帖子帮到了我 谢谢您, 那您了解
如何从Safari浏览器 跳转到一个
A...

总结出来传递规则是这样的:

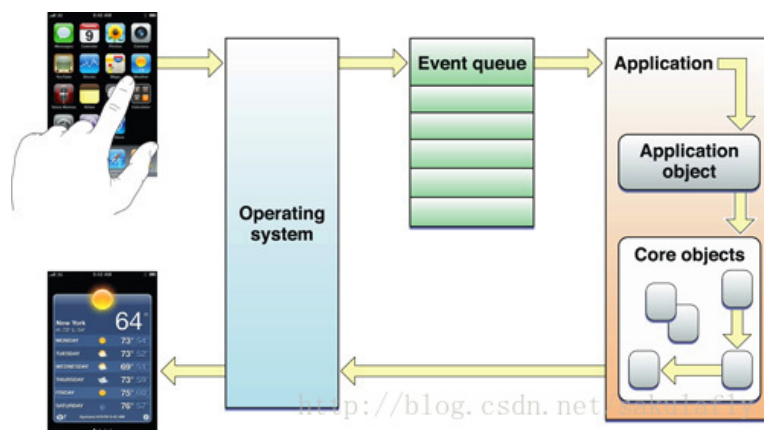


这样事件就会从first responder逐级传递过来, 直到被处理或者被抛弃。

由于UI的复杂, 这个responder chain是需要根据事件来计算的。比如, 我现在在一个view内加入了2个Button, 先点击了一个, 则first responder肯定是这个点击过的button, 但我下面可以去点击另一个button, 所以显然, 当触摸事件来时, 这个chain是需要重新计算更新的, 这个计算的顺序是事件分发的顺序, 基本上是分发的反过来。



无论是哪种事件, 都是系统本身先获得, 是iOS系统来传给UIApplication的, 由Application再决定交给谁去处理, 所以如果我们要拦截事件, 可以在UIApplication层面或者UIWindow层面去拦截。



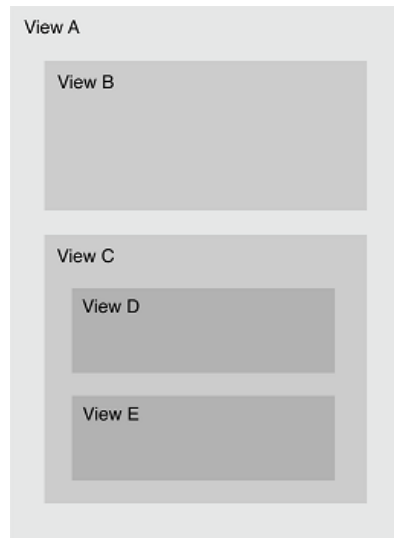
UIView是如何判定这个事件是否是自己应该处理的呢? iOS系统检测到一个触摸操作时会打包一个UIEvent对象, 并放入Application的队列, Application从队列中取出事件后交给UIWindow来处理, UIWindow会使用hitTest:withEvent:方法来递归的寻找操作初始点所在的view, 这个过程成为hit-test view。

hitTest:withEvent:方法的处理流程如下: 调用当前view的pointInside:withEvent:方法来判定触摸点是否在当前view内部, 如果返回NO, 则hitTest:withEvent:返回nil; 如果返回YES, 则向当前view内的subViews hitTest:withEvent:消息, 所有subView的遍历顺序是从数组的末尾向前遍历, 直到有subView返回非空对象或遍历完

iOS6和iOS7代码的适配(6) ———
nihao38: 牛!!!但有个问题怎么解决, 程序名称的多语言没有切换过来, 怎么办?

成。如果有subView返回非空对象, hitTest方法会返回这个对象, 如果每个subView返回都是nil, 则返回自己。

好了, 我们还是看个例子:



这里ViewA包含ViewB和ViewC, ViewC中继续包含ViewD和ViewE。假设我们点击了viewE区域, 则hit-test View判定过程如下:

1. 触摸在A内部, 所以需要检查B和C
2. 触摸不在B内部, 在C内部, 所以需要检查D和E
3. 触摸不在D内部, 但在E内部, 由于E已经是叶子了, 所以判定到此结束

我们可以运行一段代码来验证, 首先从UIView继承一个类myView, 重写里面的

```
[objc] C 80
01. - (UIView *)hitTest:(CGPoint)point withEvent:(UIEvent *)event
02. {
03.     UIView *retView = nil;
04.     NSLog(@"hitTest %@ Entry! event=%@", self.name, event);
05.
06.     retView = [super hitTest:point withEvent:event];
07.     NSLog(@"hitTest %@ Exit! view = %@", self.name, retView);
08.
09.     return retView;
10. }
```

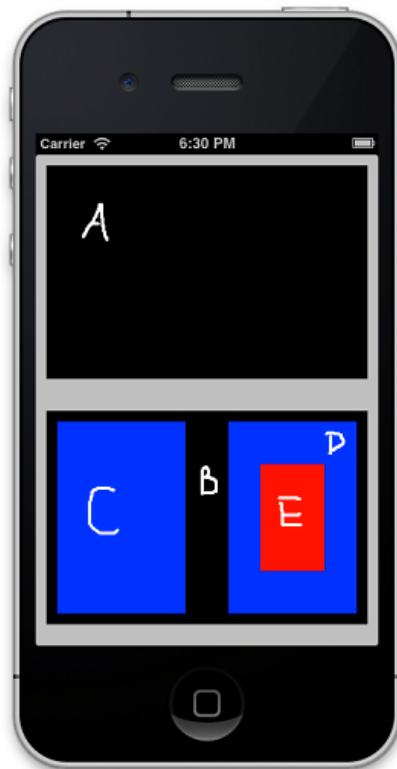
```
[objc] C 80
01. - (BOOL)pointInside:(CGPoint)point withEvent:(UIEvent *)event
02. {
03.     BOOL ret = [super pointInside:point withEvent:event];
04.     // if ([self.name isEqualToString:@"viewD"]) {
05.     //     ret = YES;
06.     // }
07.     if (ret) {
08.         NSLog(@"pointInside %@ = YES", self.name);
09.     } else {
10.         NSLog(@"pointInside %@ = NO", self.name);
11.     }
12.
13.     return ret;
}
```

14. `}`

在viewDidLoad方法中手动加入5个view，都是myView的实例。

```
[objc] C 80
01. - (void)viewDidLoad
02. {
03.     [super viewDidLoad];
04.
05.     _viewA = [[myView alloc] initWithFrame:CGRectMake(10, 10, 300, 200) Color:
[UIColor blackColor] andName:@"viewA"];
06.     [self.view addSubview:_viewA];
07.     [_viewA release];
08.
09.     _viewB = [[myView alloc] initWithFrame:CGRectMake(10, 240, 300, 200) Color:
[UIColor blackColor] andName:@"viewB"];
10.     [self.view addSubview:_viewB];
11.     [_viewB release];
12.
13.     _viewC = [[myView alloc] initWithFrame:CGRectMake(10, 10, 120, 180) Color:
[UIColor blueColor] andName:@"viewC"];
14.     [_viewB addSubview:_viewC];
15.     [_viewC release];
16.
17.     _viewD = [[myView alloc] initWithFrame:CGRectMake(170, 10, 120, 180) Color:
[UIColor blueColor] andName:@"viewD"];
18.     [_viewB addSubview:_viewD];
19.     [_viewD release];
20.
21.     _viewE = [[myView alloc] initWithFrame:CGRectMake(30, 40, 60, 100) Color:
[UIColor redColor] andName:@"viewE"];
22.     [_viewD addSubview:_viewE];
23.     [_viewE release];
24.
25. }
```

这个样式如下：



当我点击viewE的时候，打印信息如下：

```
2014-01-25 18:32:46.538 eventDemo[1091:c07] hitTest viewB Entry! event=<UITouchesEvent: 0x8d0cae0>
timestamp: 6671.26 touches: {(
)}
2014-01-25 18:32:46.538 eventDemo[1091:c07] pointInside viewB = YES
2014-01-25 18:32:46.539 eventDemo[1091:c07] hitTest viewD Entry! event=<UITouchesEvent: 0x8d0cae0>
timestamp: 6671.26 touches: {(
)}
2014-01-25 18:32:46.539 eventDemo[1091:c07] pointInside viewD = YES
2014-01-25 18:32:46.539 eventDemo[1091:c07] hitTest viewE Entry! event=<UITouchesEvent: 0x8d0cae0>
timestamp: 6671.26 touches: {(
)}
2014-01-25 18:32:46.540 eventDemo[1091:c07] pointInside viewE = YES
2014-01-25 18:32:46.540 eventDemo[1091:c07] hitTest viewE Exit! view = <myView: 0x8c409f0; frame = (30 40;
60 100); layer = <CALayer: 0x8c40a90>>
2014-01-25 18:32:46.540 eventDemo[1091:c07] hitTest viewD Exit! view = <myView: 0x8c409f0; frame = (30 40;
60 100); layer = <CALayer: 0x8c40a90>>
2014-01-25 18:32:46.541 eventDemo[1091:c07] hitTest viewB Exit! view = <myView: 0x8c409f0; frame = (30 40;
60 100); layer = <CALayer: 0x8c40a90>>
2014-01-25 18:32:46.541 eventDemo[1091:c07] touchesBegan viewE
2014-01-25 18:32:46.624 eventDemo[1091:c07] touchesEnded viewE
```

从打印信息可以看到，先判断了viewB，然后是viewD，最后是viewE，但事件就是直接传给了viewE。

上一篇

iOS中的retainCount

下一篇

iOS6和iOS7代码的适配(5)——popOver

顶

1

踩

0

主题推荐

ios

移动平台

远程控制

设计模式

多点触摸

猜你在找

- iOS拦截控件事件处理后继续执行原来的消息传递流程

UINavigationController表示UINavigationController中的控件

GStreamer基础教程01Hello World

iOS开发UIView之间常用视图之间切换方式

查看MFC源代码
- GStreamer基础教程02GStreamer概念

ios协议和委托

关于开源中国代码托管的一些问题

QT命名空间使用

代码戒律Objective-C最佳实践

准备好了么？跳吧！

更多职位尽在 CSDN JOB

事件响应技术员（深圳）	我要跳槽	iOS开发工程师	我要跳槽
平安科技(深圳)有限公司	10-15K/月	筑客网络技术（上海）有限公司	8-15K/月
Android、IOS架构师	我要跳槽	iOS开发工程师	我要跳槽

新浪微博

20-40K/月

浙江大华技术股份有限公司

8-20K/月



坚持一个月 听懂CNN

上班时我也偷偷练口语，每天只要5分钟，30天后真的不一样！

>

立即行动



查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

apttech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈