

Android事件分发机制

1.被分发的对象

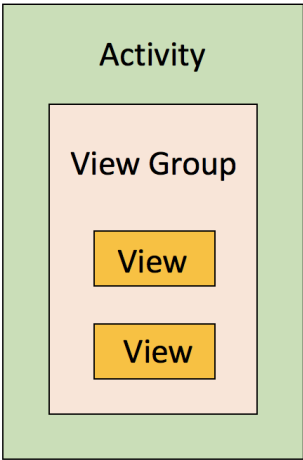
被分发的对象是那些？被分发的对象是用户触摸屏幕而产生的点击事件，事件主要包括：按下、滑动、抬起与取消。这些事件被封装成MotionEvent对象。该对象中的主要事件如下表所示：

事件	触发场景	单次事件流中触发的次数
MotionEvent.ACTION_DOWN	在屏幕按下时	1次
MotionEvent.ACTION_MOVE	在屏幕上滑动时	0次或多次
MotionEvent.ACTION_UP	在屏幕抬起时	0次或1次
MotionEvent.ACTION_CANCEL	滑动超出控件边界时	0次或1次

按下、滑动、抬起、取消这几种事件组成了一个事件流。事件流以按下为开始，中间可能有若干次滑动，以抬起或取消作为结束。

2.分发事件的组件

分发事件的组件，也称为分发事件者，包括Activity、View和ViewGroup。它们三者的一般结构为：



从上图中可以看出，Activity包括了ViewGroup，ViewGroup又可以包含多个View。

组件	特点	举例
Activity	安卓视图类	如MainActivity
ViewGroup	View的容器，可以包含若干View	各种布局类
View	UI类组件的基类	如按钮、文本框

3.分发的核心方法

负责对事件进行分发的方法主要有三个，分别是：

- dispatchTouchEvent (

- onTouchEvent ()
- onInterceptTouchEvent () 。

它们并不存在于所有负责分发的组件中，其具体情况总结于下面的表格中：

组件	dispatchTouchEvent	onTouchEvent	onInterceptTouchEvent
Activity	存在	存在	不存在
ViewGroup	存在	存在	存在
View	存在	存在	不存在

从表格中看，dispatchTouchEvent,onTouchEvent方法存在于上文的三个组件中。而onInterceptTouchEvent为ViewGroup独有。这些方法的具体作用在下文作介绍。

4. 事件分发过程

- 向下传播：Activity包括Layout，事件从Activity向Layout传播被称作‘向下传播’。Layout包含若干View，事件从Layout向其子View传播，也被称为‘向下传播’。
- 向上传播：与‘向下传播’相反。
- 分发方法 `dispatchTouchEvent`

从方法的名称中可以看出该方法主要是负责分发，是安卓事件分发过程中的核心。事件是如何传递的，主要就是看该方法，理解了这个方法，也就理解了安卓事件分发机制。

在了解该方法的核心机制之前，需要知道一个结论：

- 如果某个组件的该方法返回TRUE,则表示该组件已经对事件进行了处理，不用继续调用其余组件的分发方法，即停止分发。
- 如果某个组件的该方法返回FALSE,则表示该组件不能对该事件进行处理，需要按照规则继续分发事件。在不复写该方法的情况下，除了一些特殊的组件，其余组件都是默认返回False的。后续有例子说明。
- ViewGroup独有的 `onInterceptTouchEvent` 方法

ViewGroup的该方法与Activity的类似，只是新添了一个onInterceptTouchEvent方法。当事件传入时，首先会调用onInterceptTouchEvent。

- 如果该方法返回了FALSE（表示不拦截），则交给子View去调用dispatchTouchEvent () 方法
- 如果该方法返回了TRUE（表示拦截），则直接交给该ViewGroup对象的onTouchEvent(ev)方法处理，具体是否能处理以onTouchEvent()的实际情况为准。

