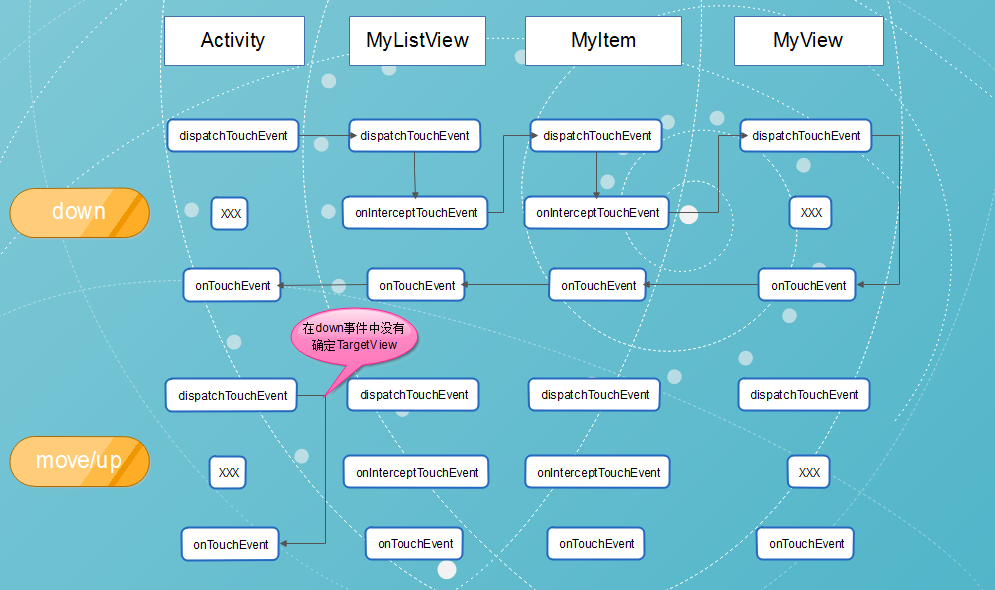
1. 处理事件的基本方法

View中diapatchTouchEvent、onTouchEvent

ViewGroup中dispatchTouchEvnet、onInterceptTouchEvent、onTouchEvent

二、事件传递流程

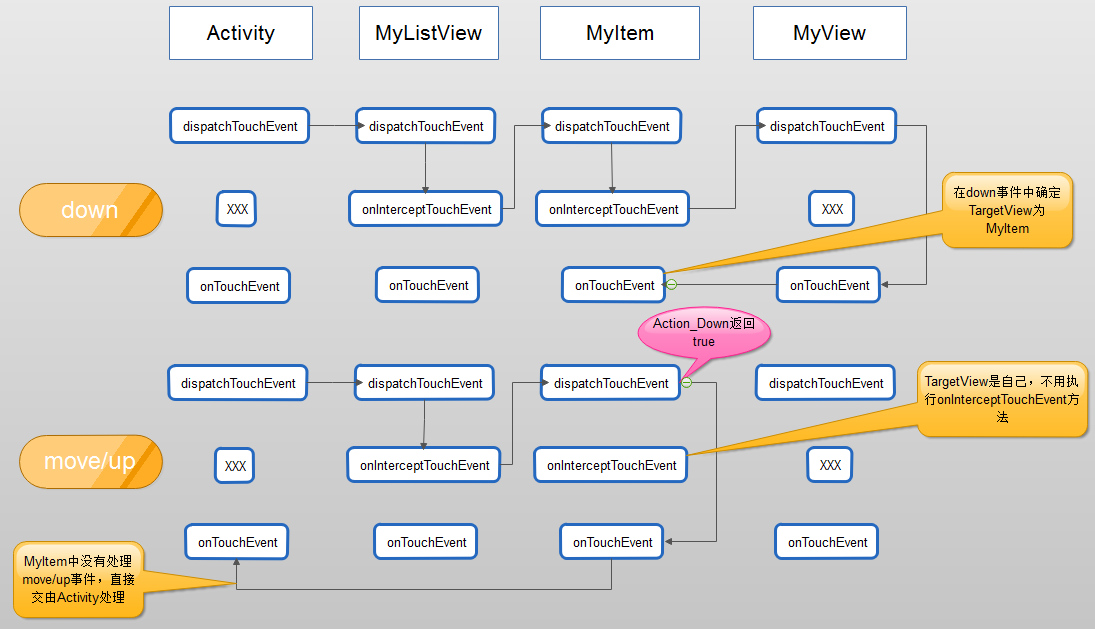
1. 基本流程：所有方法均返回默认值



1. 确定目标控件TargetView

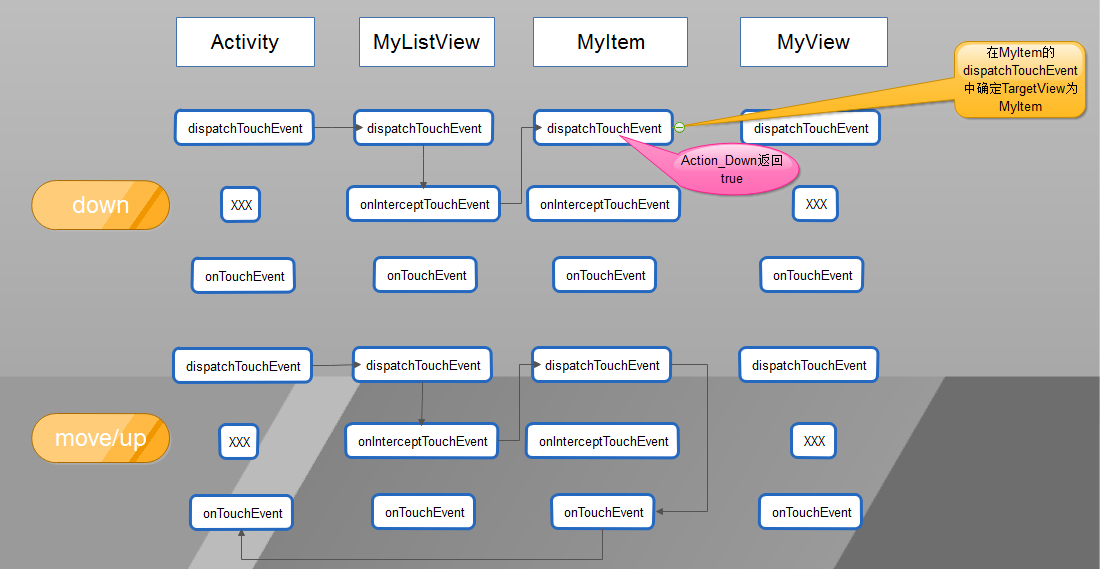
确定目标控件的动作只能在Action\_Down事件中，并且只能是在dispatchTouchEvent或者onTouchEvent方法中完成：

1. MyItem的onTouchEvent方法的Action\_Down中返回true，确定目标控件为MyItem，其他返回默认值

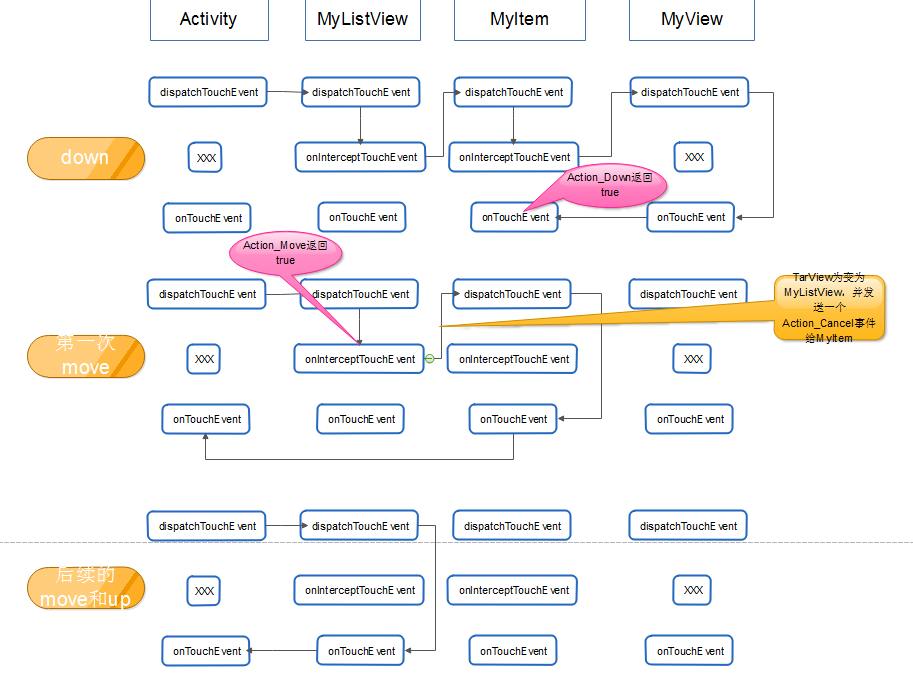


由上图，可知：

1. 在MyItem的onTouchEvent的Action\_Down中确定目标控件为MyItem
2. 后续的Action\_Move和Action\_Down就不会传递到子控件MyView中
3. 且dispatchTouchEvent直接将事件分发给onTouchEvent，而不用调用拦截方法onInterceptTouchEvent
4. 因为事件没有消耗，事件又直接交给Activity的onTouchEvent处理
5. MyItem的dispatchTouchEvent方法的Action\_Down返回true



1. dispatchTouchEvent的Action\_Down返回true，目标控件设置为MyItem，并且将事件消费，不往下传递
2. 修改目标控件TargetView
3. 验证onInterceptTouchEvent修改目标控件

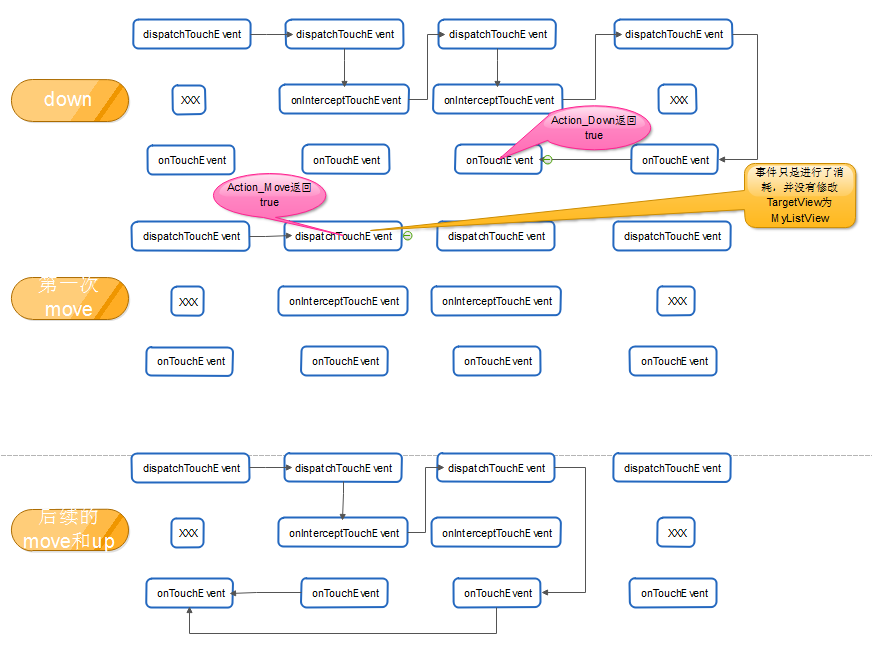


在down事件传递中，MyItem的onTouchEvent返回true，确定目标控件为MyItem

在第一次move事件时，在MyListView的onInterceptTouchEvent返回true，此时MyListView会将move动作给为Action\_Cancel，并继续往下传递，用于告诉之前的目标控件MyItem，后续是事件会拦截，不用它处理

在后续的move和up事件时，此时的目标控件改为MyListview，所有一会传递到MyListView为止

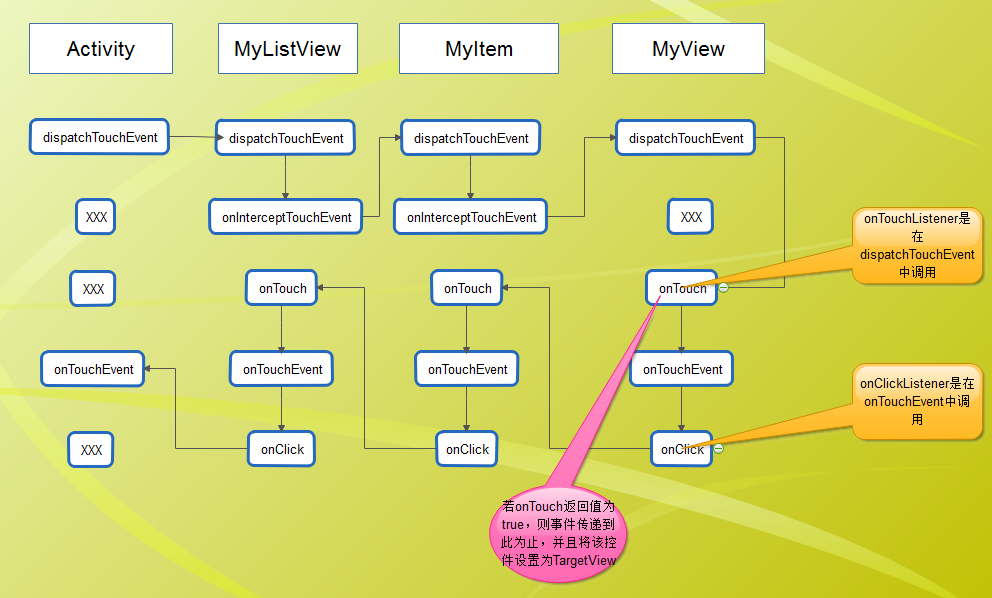
1. 验证dispatchTouchEvent不可修改目标控件



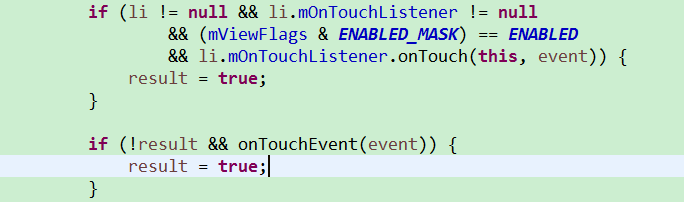
在Action\_Down事件中确定了目标控件为MyItem

在第一次move中我们返回true，后续的move返回false，通过第三个事件传递图可见目标控件并没有更改，仍然是MyItem

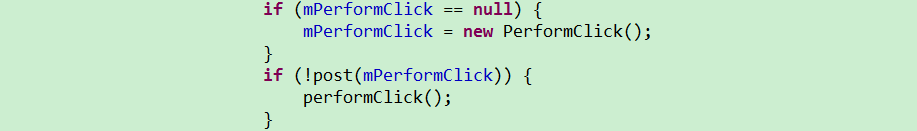
1. onTouchListener与onClickListener调用的时机



在dispatTouchEvent方法中先调用回调方法onTouch，如onTouch返回true，则后续的onTouchEvent不执行，并且将该View设置为目标控件

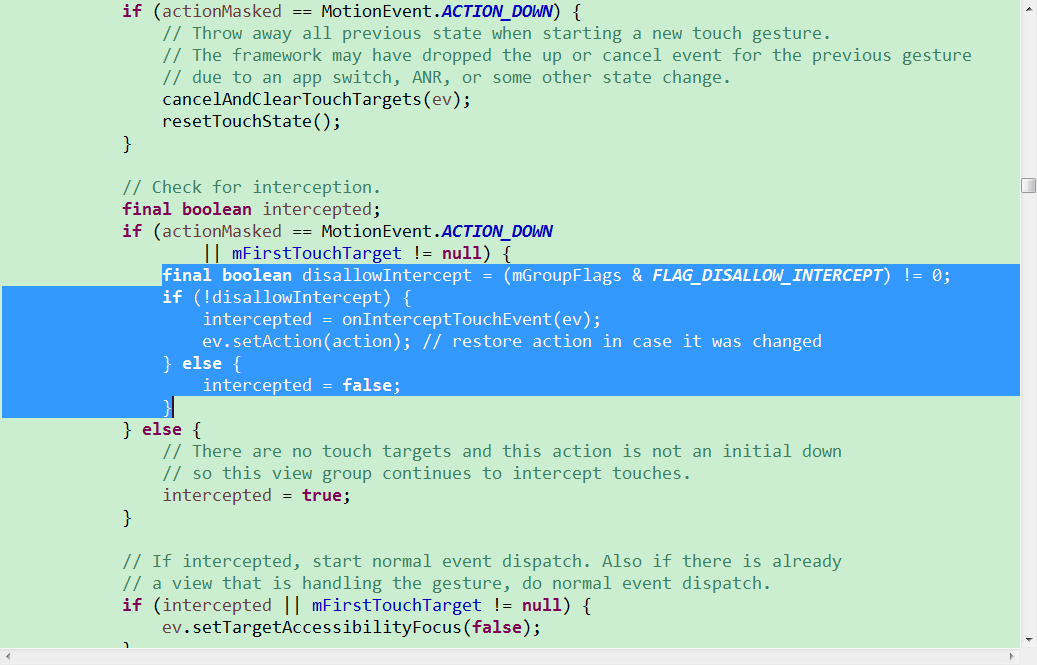


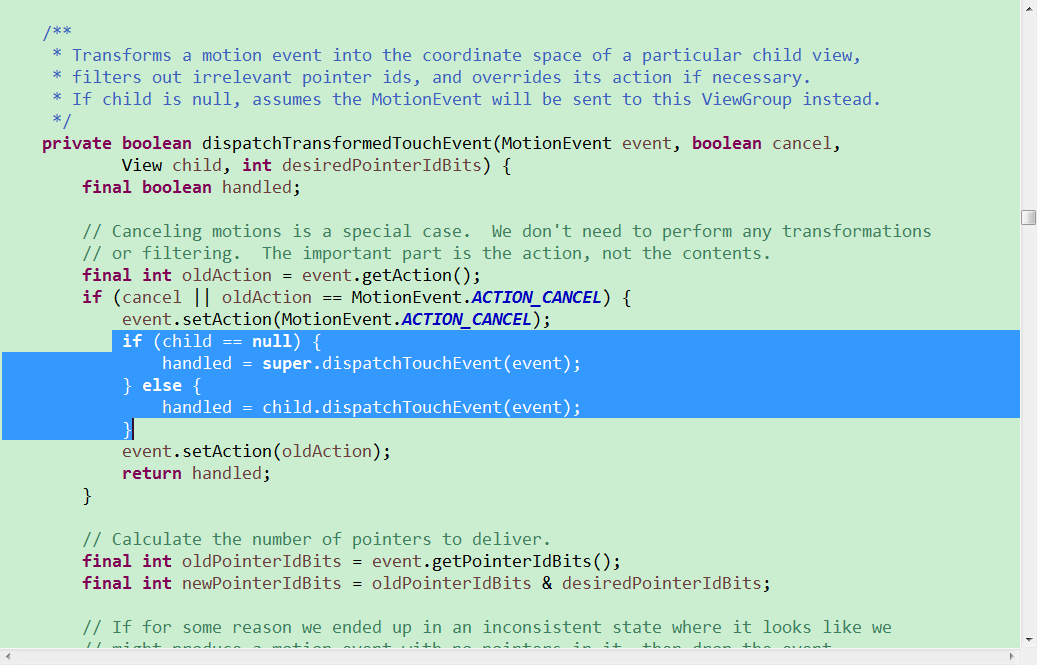
在onTouchEvent方法中调用点击回调

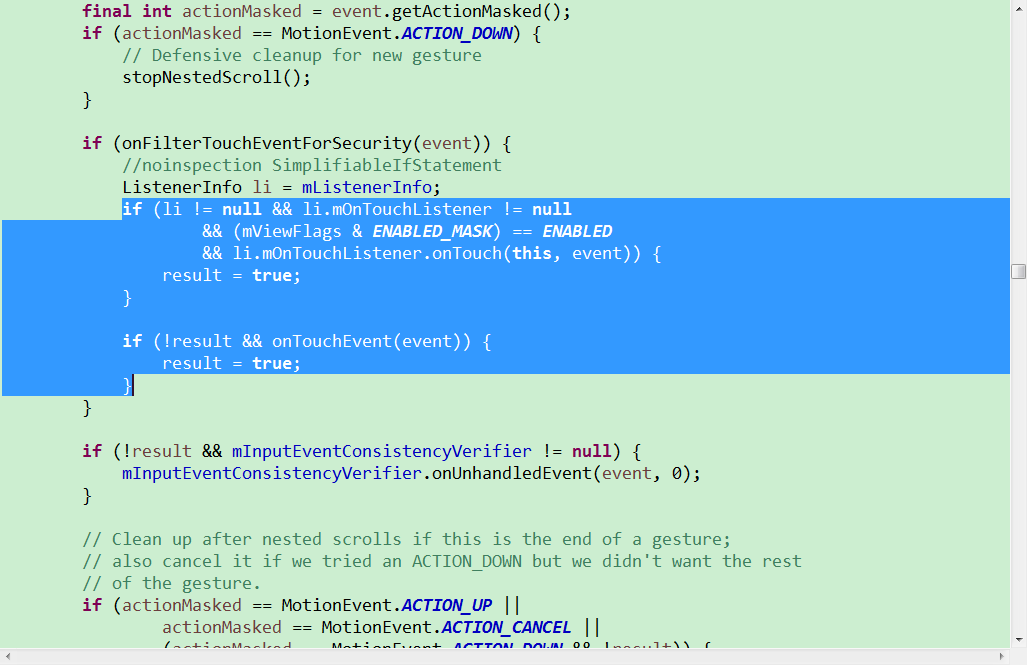


ViewGroup.java中

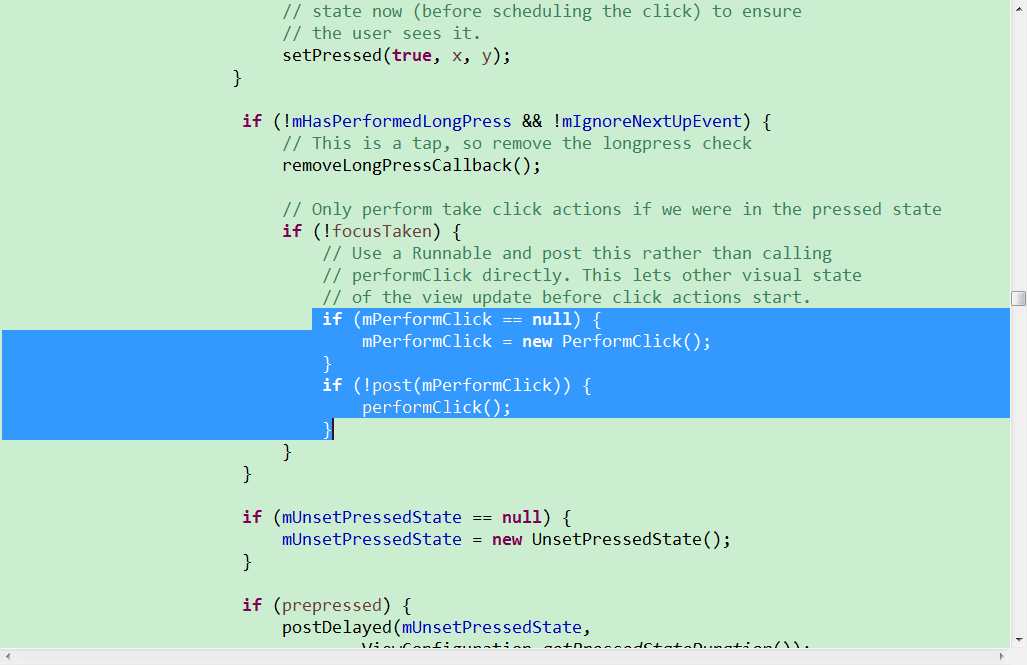
dispatchTouchEvent中会一次调用onInterceptTouchEvent、child.dispatchTouchEvent/super.dispatchTouchEvent、onTouchListener、onTouchEvent







onTouchEvent中调用onClickListener、onLongClickListener等



三、常见的滑动控件

ScrollView/ HorizontalScrollView事件默认处理情况

处理所有事件

Action\_Down:不拦截

Action\_Move:当竖向滑动距离大于某个值时，拦截

ListView

处理所有事件

Action\_Down:不拦截、但处理

Action\_Move:当竖向滑动距离大于某个值时，拦截

ViewPager

处理所有事件

Action\_Down:不拦截

Action\_Move:当横向滑动距离大于某个值时，拦截

以上滑动容器都不能设置onClickListener后者设置了也不响应，因为在它们的onTouchEvent中没有调用super.onTouchEvent

1. ScrollView中嵌入ListView

问题：只能响应ScrollView的滑动，ListView滑动不响应

Down事件：ScrollView、ListView都不拦截，事件在ListView的onTouchEvent消费掉，此时目标控件为ListView

Move事件：ScrollView判断滑动距离大于mTouchSlop时，拦截，此时目标控件为ScrollView，后续所有的事件都不经过ListView

所有表现为ScrollView处理所有的move事件，即ScrollView能滑动，ListView不能滑动

方法：

1. ScrollView的onInterceptTouchEvent返回false
2. ListView的onTouchEvent中调用parent.requestDisallowIntercept是ScrollView不拦截
3. 将ListView实现onTouchListener接口，并设置给ScrollView，事件每次传递给ScrollView时，都会在onTouchEvent之前执行onTouch方法，我们就是在onTouch方法中处理ListView的滑动

2. ListView嵌套HorizontalScrollView、ViewPager

问题：因内外控件滑动方向不一致，原生控件不产生冲突，但是当需要ListView的OnItemClickListener时，该监听器无效

原因：HorizontalScrollView、ViewPager中消费掉了down事件，ListView获取到down事件并进行消费，

尝试解决的方法：

1. 在HorizontalScrollView、ViewPager的OnClickListner中调用ListView的OnItemClickListener，结果失败

原因：HorizontalScrollView、ViewPager的onTouchEvent中并没有调用View的onClick方法，故这两个控件没有点击事件（设置了也无效）

3.ListView中的item为SwipeLayout（一个自定义的可左右滑动的控件）

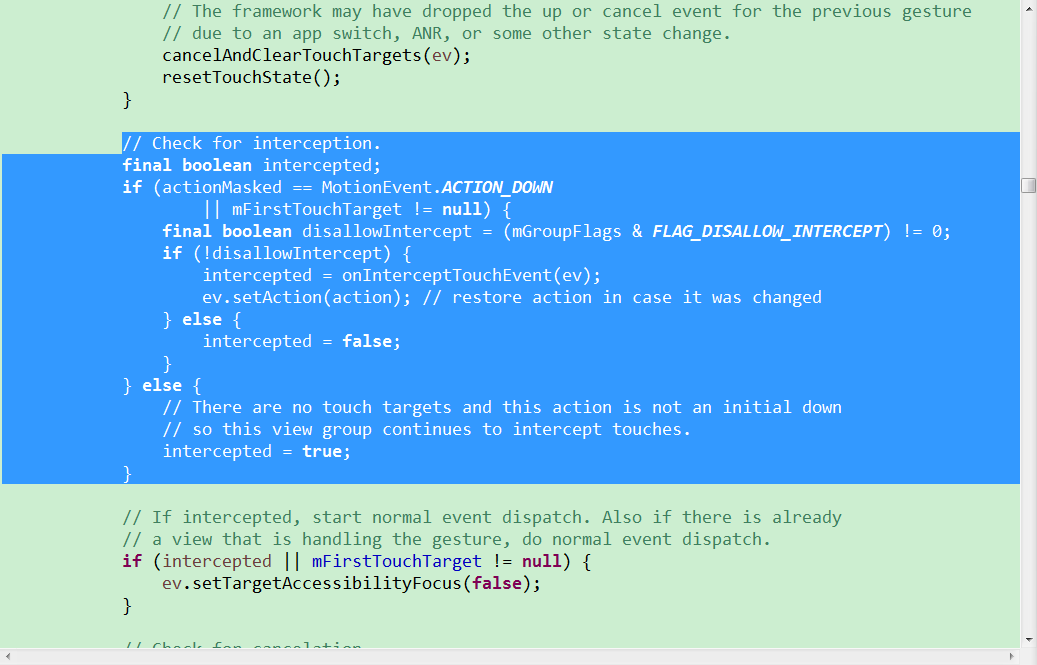
问题：同样是ListView的onItemClickListener无法响应

方法1：丢弃原有的事件传递，我们将事件全部拦截在ListView中，再SwipeLayout中自定义一个方法（如：onSiwpe（MotionEvent）），将事件传递给SwipeLayout，这样事件在ListView和SwipeLayout中都可以进行各自的消费（即利用两次）

方法2：SwipeLayout不消费任何事件,并将SwipeLayout的左右移动交给ListView处理，ListView使用scrollTo、scrollBy移动SwipeLayout

方法3：给SwipeLayout设置OnClickListener，并在onClick中回调OnItemClickListener(即利用SwipeLayout来响应ListView的接口)

四、分析ViewGroup源码中dispatchTouchEvent调用拦截的过程

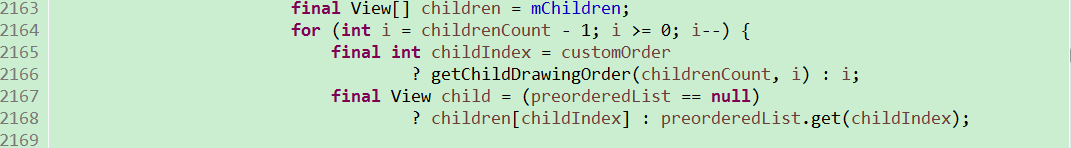


当一个Action\_Down传递到该ViewGroup时，此时mTouchTarget为null，disallowIntercept为flase，执行拦截方法onInterceptTouchEvent

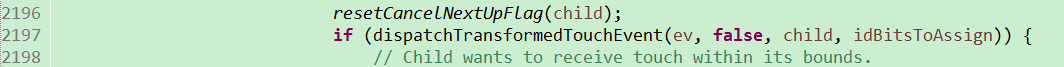
当一个Action\_Move/Action\_up传递到该ViewGroup时，此时mTouchTarget（mTouchTarget在Action\_Down中已经确定，若点击的在子控件child上，那么mTouchTarget就会赋值为child，否则为null）有确定值：

A．若mTouchTarget为子控件child，如果child设置了requestDisallowIntercept方法请求该ViewGroup不拦截，那么标志位intercepted置为false，否则会执行该ViewGroup的interceptTouchEvent方法询问是否拦截

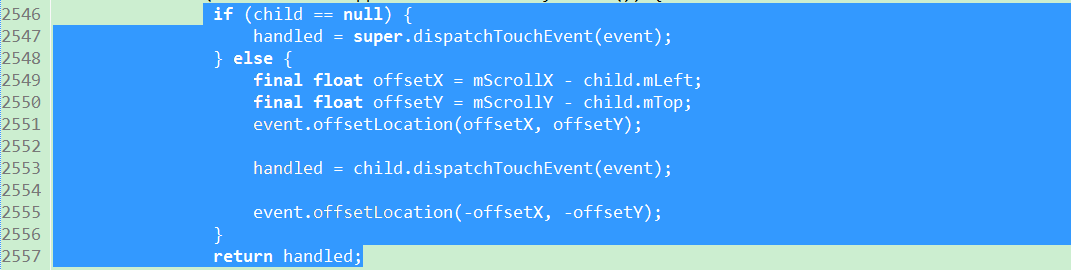
ViewGroup的dispatchTouchEvent方法中寻找目标控件TargetView的源码分析:



如上图，倒序的遍历ViewGroup中的子控件，并在该遍历for循环中调用下图代码，一旦找到了符合条件的子控件，那么就会break终止循环



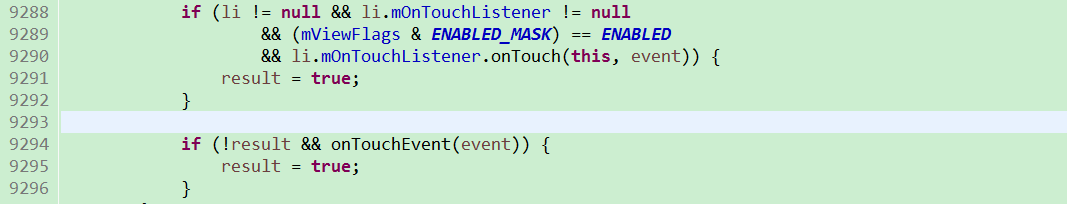
我们进入dispatchTransformedTouchEvent中，如下图是其中一些代码



当目标子控件child为空时，表示点击的位置不在子控件的范围内，那么分发给自己，即调用super.dispatchTouchEvent，即最后调用到View.dispatchTouchEvent中。

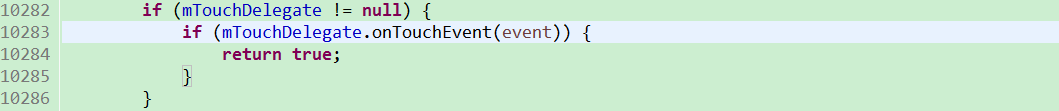
当目标子控件child不为空时，表示点击的位置在子控件的范围内，那么就将事件分发给子控件child.dispatchTouchEvent。

分析View中dispatchTouchEvent的源码

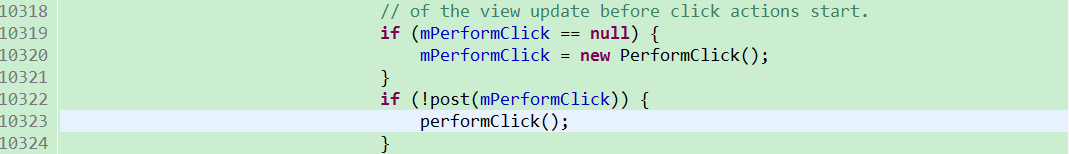


首先询问是否有OnTouchListener，若有查看是否消费，若消费事件就此结束，若不消费，传递给onTouchEvent处理

分析View中onTouchEvent的源码



若设置了触摸代理TouchDelegate，那么所有事件全部交由事件代理来处理，事件就此终止



若没有设置TouchDelegate，那么事件就给onTouchEvent自己处理，其中OnClickListener就是在Action\_up中调用并消费事件

伪代码

ViewGroup的伪代码

dispatchTouchEvent(event){

If(!disallowIntercept){

interceptTouchEvent();-------------------------------调用拦截方法

}

…

for(倒序遍历所有子控件){

if(当前时间的坐标在该子控件范围内){

mTargetView=该子控件;

break;

}

}

…

If(mTargetView==null){

handle=Super.dispatchTouchEvent();--------------调用父类的分发，即View的分发事件

}else{

handle=mTargetView.dispatchTouchEvnet();----调用子控件的分发

}

…

onTouchEvent();

return handle;------------------------------------------------此处的返回值会影响目标控件的确定

}

View的伪代码

dispatchTouchEvent(event){

if(设置OnTouchListener并且消费事件){

reslut=true;

}

if(!reslut&&onTouchEvent()){-----------------------调用onTouchEvent方法，消费则结果为true

reslut=true

}

return result;--------------------------------------------此处的返回值会影响目标控件的确定

}

结论：

1. 只有在dispatchTouchEvent和onTouchEvent方法中的Action\_Down事件才能确定目标控件
2. 只能在onInterceptTouchEvent方法中才能修改目标控件，且拦截的控件会发个Action\_Cancel给子控件，并将拦截的事件交给Activity
3. 如果确定了目标控件，dispatchTouchEvent和onTouchEvent返回false，那么事件交给Activity
4. 如果没有确定目标控件，dispatchTouchEvent和onTouchEvent返回false，那么事件交给父控件

Opengl事件传递与原生的不同点：

1. 目标控件的确定

在android原生中，onTouchEvent和dispatchTouchEvent的down事件都可以确定目标控件

在opengl中，在onTouchEvent的down事件可以中确定，在dispatchTouchEvent的down事件中不能确定

1. 目标控件确定后，interceptTouchEvent方法是否执行

在android原生中，一旦确定了目标控件，那么后续的事件就不会执行该目标控件的interceptTouchEvent方法

在opengl中则会执行interceptTouchEvent方法

1. 目标控件的修改

在android原生控件中，通过interceptTouchEvent修改目标控件时，目标控件会发一个cancel事件给原来的目标控件的dispatcheTouchEvent方法；后续事件不会执行目标控件的interceptTouchEvent方法；后续的move事件都是只执行一次onTouchEvent方法

在opengl中，目标控件会发送一个cancel事件给原来的目标控件的onTouchEvent方法；后续事件还是会执行interceptTouchEvent方法；后续的move事件都是执行两次onTouchEvent方法

1. dispatchTouchEvent返回值

在android原生控件中，如果有目标控件，dispatchTouchEvent中返回true直接消费，返回false直接交给Activity处理；如果没有目标控件，交由父控件处理

在opengl中，不管有没有目标控件，只要dispatchTouchEvent方法中返回true或者false，后续所有的事件全部由Activity处理，不会往下分发