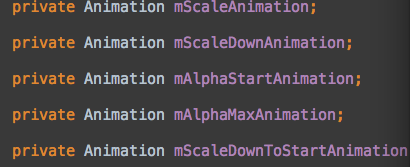
SwipeRefresh layout 读解

里面有一个MD 的progress drawable

5个animation （没用animator ）应该是为了兼容



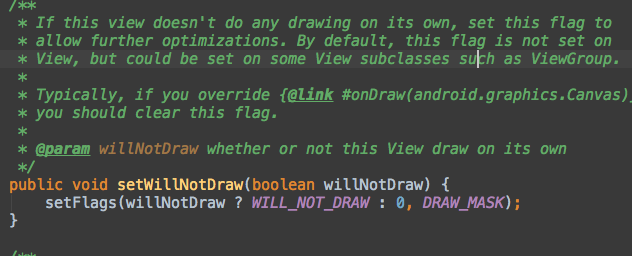
见其名 闻其意

一个animation listener



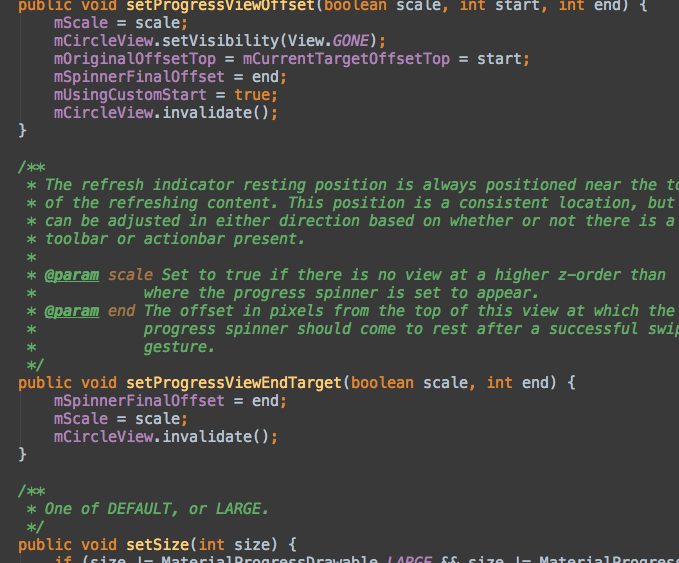
在end里面对判断刷新，进行progress 的设置和动画执行 并且调用refresh listener 如果不是刷新状态 progress 停止了动画隐藏了圆形背景

ok看public method 先看construct method



这个方法学到了 自定义view 的时候如果没有自定义绘制的东西（draw things with canvas）可以设置绝壁是为了性能（iOS也有类似的 不过是不重写drawRect）

这构造里通过createProgressView 添加了进度view 下拉滑动使用了nestedScrollinghhelper



这几个方法是修改默认参数以呈现更丰富的UI



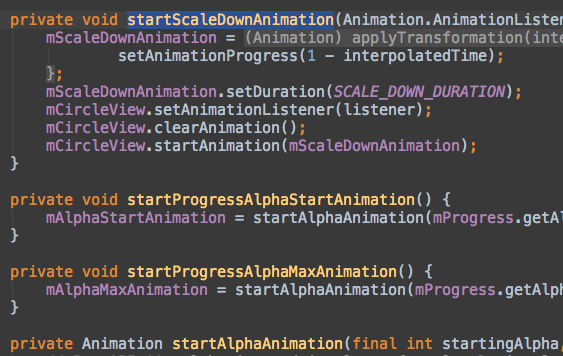
动画结束的listener

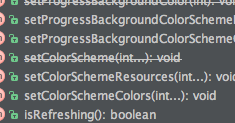


更改刷新状态。

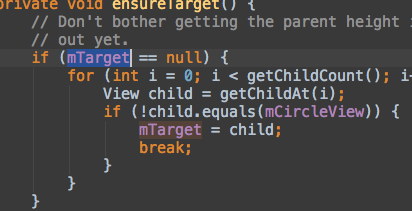


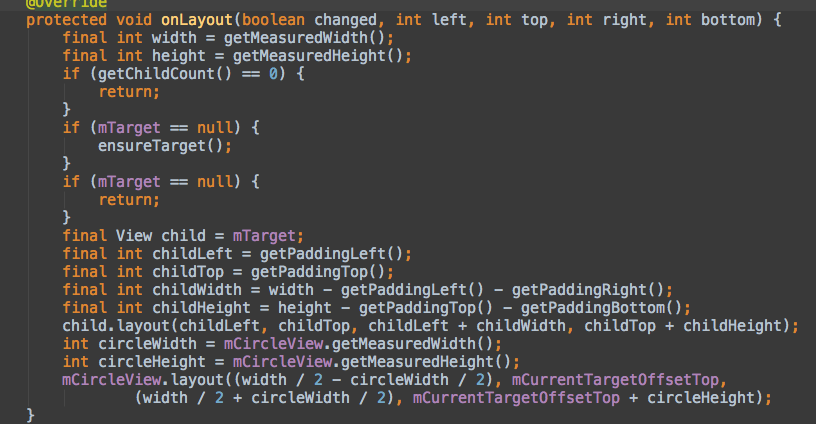
开始动画 并且掉回调

这些是动画启动 在onTouchEvent的时候会执行



又是一些配置参数的方法

获取子view （不要circleview



onLayout 方法没有子view 就直接返回了

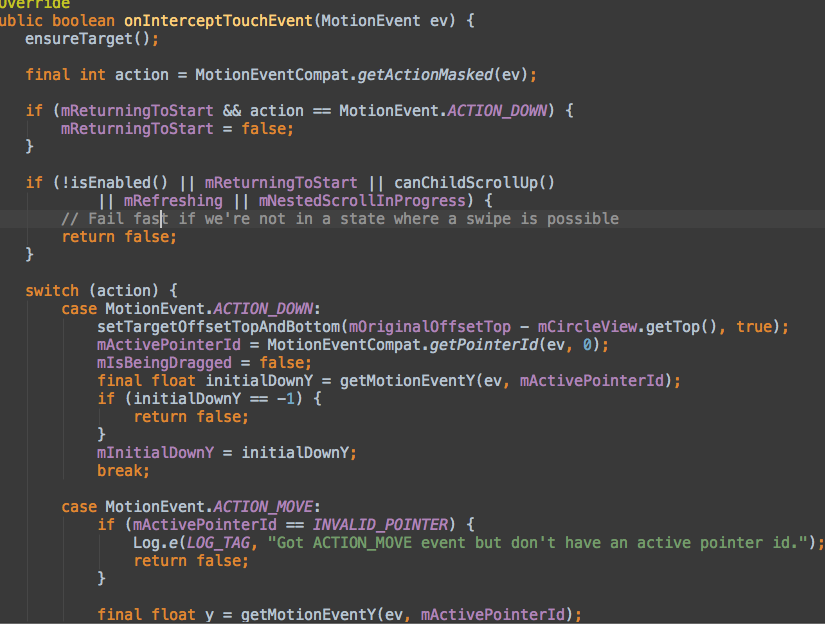
layout child view at index 0 应该是layout 了子布局的跟布局然后会层层调用到最后view的layout

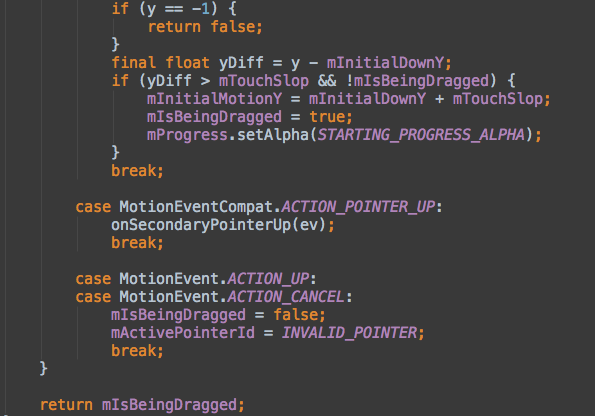


onMeasure方法

获取的根布局，根布局measure ，circleview measure

获取到circleview 在子view 的index 并且记录下来=

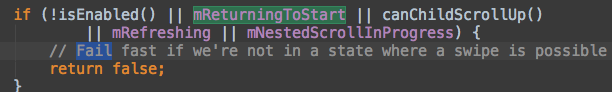




onintercept touchevent

down 事件级return to start 为false 不拦截





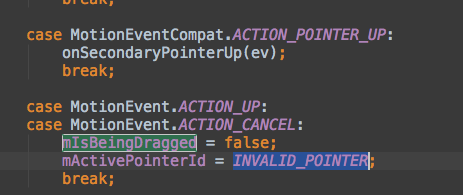
所有一下情况都不拦截。View 不可用，正在返回到开始，子view 可以上滑，真正刷新，正在滑动。

通过以上过程。

1.进入down 重设circleview 的offset 获取到y 如果-1 不拦截。记录down 的y

2.进入move 获取到y 如果y <-1 直接返回false 不拦截 计算偏移量y如果偏移量大于最小可识别的值，并且不正在被滑动，记录移动的初始化的y（down y+最小识别）并且设置为正在被drag

设置progres 的alpha



up不处理 最后是对特殊情况（电话进来等）进行处理（将state 都设置为false跟invalid

默认返回

是否真正被滑动 简单的就是说down 记录y不拦截 move 如果偏移量大于最小值 记录为正在被拉 拦截所有事件



该方法被重写主要做了版本判断<21以及子view 的判断为AbsListView时活着跟布局无法nestedscroll的时候什么都不做，不然就可以拦截事件。



在onNestedPreScroll 里面经过计算最后调用了上面的方法来移动circleview

并且

根据偏移量来不断执行动画，clear 原先的执行新的= =原来这样。。学到了



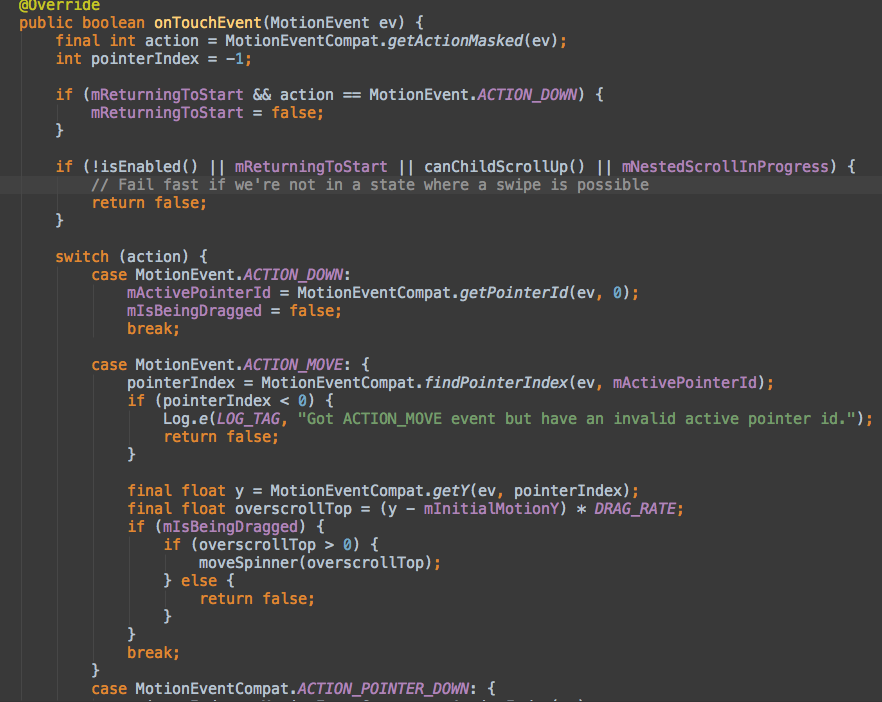
移动spinner 的方法

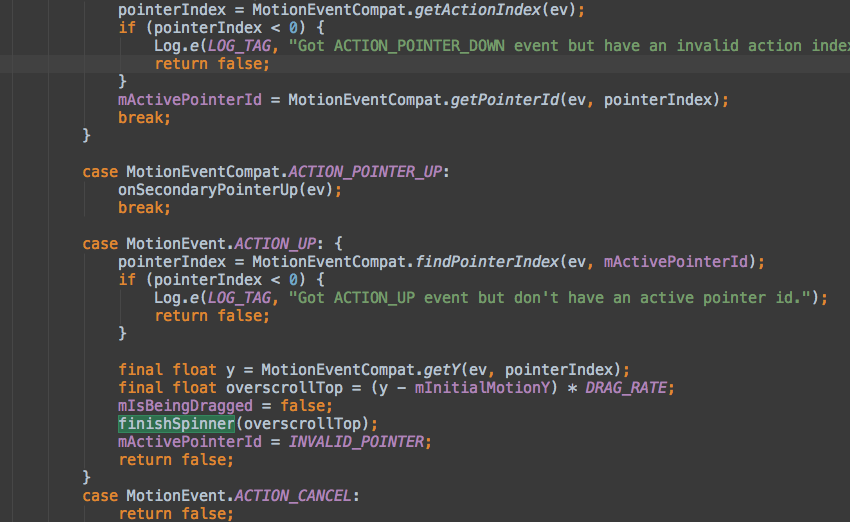


是结束spinner 的方法

而这两个方法在touchevent 跟 nested scroll 里面都有调用 肯定是版本兼容的做法。

Finish里面同样执行了动画（旋转 spinner ）的动画





onTouchEvent 方法！！

跟onInterceptTouchEvent 做了类似的状态过滤

接下来

1. down 事件 获取pointerid 然后设置drag 状态false
2. move 事件 获取pointerid 的 index 获取 y 计算 y 于在onInterceptTouchEvent记录的出事y 的偏移量。然后判断是否正在滑动，在判断 移动的y 是否大于0 是则移动spinner 否则不处理事件。
3. up事件执行了finish spinner 的动画

最后是一些执行动画的方法没有仔细去看，，，，太复杂了 ok 大概撸完了 学到不少