GestureDetector 和 SimpleOnGestureListener

OnGestureListener 、OnDoubleTapListener，GestureDectector有一个内部类 SimpleOnGestureListener，这个内部类提供了更方便的响应不同手势的类。

接口方法：

@Override  
        public boolean onSingleTapUp(MotionEvent e) {  
            return false;  
        }

@Override  
        public void onLongPress(MotionEvent e) {  
              
        }

  /\*\*  
         \* @param e1 The first down motion event that started the scrolling.  
           @param e2 The move motion event that triggered the current onScroll.  
           @param distanceX The distance along the X axis(轴) that has been scrolled since the last call to onScroll. This is NOT the distance between e1 and e2.  
           @param distanceY The distance along the Y axis that has been scrolled since the last call to onScroll. This is NOT the distance between e1 and e2.  
                       无论是用手拖动view，或者是以抛的动作滚动，都会多次触发 ,这个方法在ACTION\_MOVE动作发生时就会触发 参看GestureDetector的onTouchEvent方法源码

distanceX表示在x轴方向上的距离，不是e1和e2的距离（也不是在x轴方向上的分量）   
distanceY表示在y轴方向上滑动的距离，不是e1和e2的距离（也不是在y轴方向上的分量）

@Override  
        public boolean onScroll(MotionEvent e1, MotionEvent e2,  
                float distanceX, float distanceY) {  
            return false;  
        }

/\*\*  
         \* @param e1 第1个ACTION\_DOWN MotionEvent 并且只有一个  
         \* @param e2 最后一个ACTION\_MOVE MotionEvent   
         \* @param velocityX X轴上的移动速度，像素/秒    
         \* @param velocityY Y轴上的移动速度，像素/秒  
         \* 这个方法发生在ACTION\_UP时才会触发 参看GestureDetector的onTouchEvent方法源码  
         \*   
         \* \*/  
        @Override  
        public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX,  
                float velocityY) {  
            return false;  
        }

        @Override  
        public void onShowPress(MotionEvent e) {  
              
        }  
        @Override  
        public boolean onDown(MotionEvent e) {  
            return false;  
        }  
        @Override  
        public boolean onDoubleTap(MotionEvent e) {  
            return false;  
        }  
        @Override  
        public boolean onDoubleTapEvent(MotionEvent e) {  
            return false;  
        }

/\*\*  
         \* 这个方法不同于onSingleTapUp，他是在GestureDetector确信用户在第一次触摸屏幕后，没有紧跟着第二次触摸屏幕，也就是不是“双击”的时候触发  
         \* \*/  
        @Override  
        public boolean onSingleTapConfirmed(MotionEvent e) {  
            return false;  
        }

1. 点击屏幕上的某项的执行流程  有两种情况，一种是时间很短，一种时间稍长

时间很短：onDown--------》onSingleTapUp--------》onSingleTapConfirmed

时间稍长：onDown--------》onShowPress------》onSingleTapUp--------》onSingleTapConfirmed

2. 长按事件

onDown--------》onShowPress------》onLongPress

3.抛：手指触动屏幕后，稍微滑动后立即松开

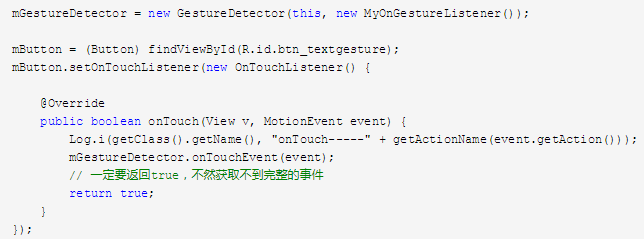
onDown-----》onScroll----》onScroll----》onScroll----》………----->onFling

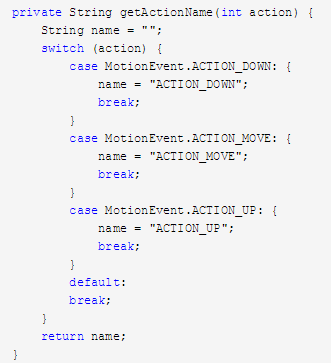
4.拖动

onDown------》onScroll----》onScroll------》onFiling

注意：有的时候会触发onFiling，但是有的时候不会触发，个人理解是人的动作不标准所致。

例子：





class MyOnGestureListener extends SimpleOnGestureListener {

60 @Override

61 public boolean onSingleTapUp(MotionEvent e) {

62 Log.i(getClass().getName(), "onSingleTapUp-----" + getActionName(e.getAction()));

63 return false;

64 }

65

66 @Override

67 public void onLongPress(MotionEvent e) {

68 Log.i(getClass().getName(), "onLongPress-----" + getActionName(e.getAction()));

69 }

70

71 @Override

72 public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {

73 Log.i(getClass().getName(),

74 "onScroll-----" + getActionName(e2.getAction()) + ",(" + e1.getX() + "," + e1.getY() + ") ,("

75 + e2.getX() + "," + e2.getY() + ")");

76 return false;

77 }

78

79 @Override

80 public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {

81 Log.i(getClass().getName(),

82 "onFling-----" + getActionName(e2.getAction()) + ",(" + e1.getX() + "," + e1.getY() + ") ,("

83 + e2.getX() + "," + e2.getY() + ")");

84 return false;

85 }

86

87 @Override

88 public void onShowPress(MotionEvent e) {

89 Log.i(getClass().getName(), "onShowPress-----" + getActionName(e.getAction()));

90 }

91

92 @Override

93 public boolean onDown(MotionEvent e) {

94 Log.i(getClass().getName(), "onDown-----" + getActionName(e.getAction()));

95 return false;

96 }

97

98 @Override

99 public boolean onDoubleTap(MotionEvent e) {

100 Log.i(getClass().getName(), "onDoubleTap-----" + getActionName(e.getAction()));

101 return false;

102 }

103

104 @Override

105 public boolean onDoubleTapEvent(MotionEvent e) {

106 Log.i(getClass().getName(), "onDoubleTapEvent-----" + getActionName(e.getAction()));

107 return false;

108 }

109

110 @Override

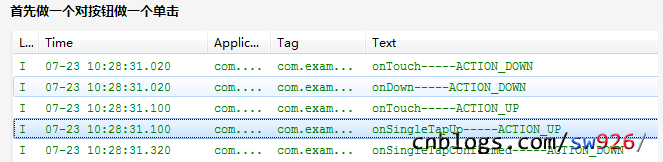
111 public boolean onSingleTapConfirmed(MotionEvent e) {

112 Log.i(getClass().getName(), "onSingleTapConfirmed-----" + getActionName(e.getAction()));

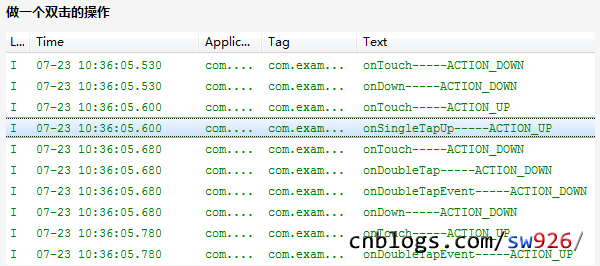
113 return false;

114 }

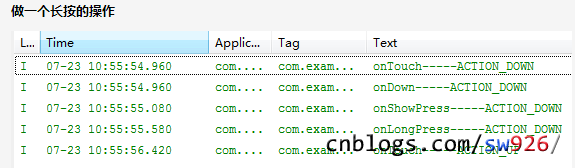
115 }



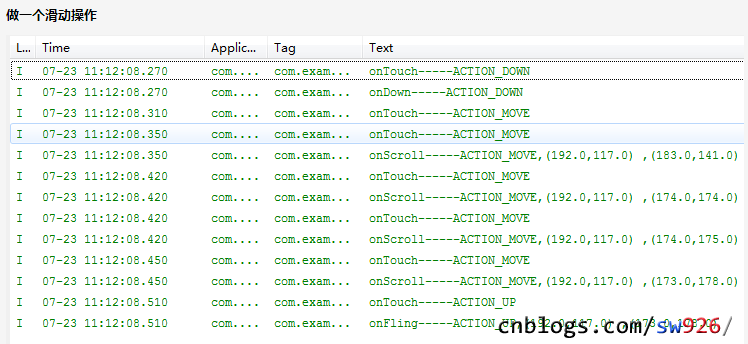
onSingleTapUp被调用，说明发生了单击事件，onSingleTapConfirmed被调用，说明确认发生了一个单击事件，不是双击的事件。需要注意的是onSingleTapUp已经是一click事件，onSingleTapUp触发的时候是**ACTION\_UP**事件。onSingleTapConfirmed是在用户手指**离开屏幕**后触发，所有up并不是所有触屏事件的结束。



首先发生了一个onSingleTapUp，说明完成了一次单击事件，然后发生了onDoubleTap，至此，一次双击事件已经完成。我们可以看到，onDoubleTap发生的时候是**ACTION\_DOWN**事件，也就是说双击事件是第二次按下屏幕的时候触发，而不是第二次离开屏幕的时候触发，在onDoubleTap发生之后，就可以在onDoubleTapEvent中监听到双击事件发生后从按下到弹起的所有触屏事件。onDoubleTap发生后不会触发onSingleTapUp和onSingleTapConfirmed。



onLongPress实在**ACTION\_DOWN**时发生，onLongPress发生后在up之前不会用其他事件触发，可以在**onShowPress**处理状态的改变，比如按钮的按下状态。



onScroll事件是拖动，onFling是抛。结合log来了解一下。首先是ACTION\_DOWN，之后多次ACTION\_MOVE，移动超过一定距离，触发了onScroll，如果onScroll被触发，**在up之前是不会有长按，单击，双击等事件的**

boolean  onSingleTapConfirmed(MotionEvent e)  
boolean  onSingleTapUp(MotionEvent e)  
上面这两个函数都是在touch down后又没有滑动（onScroll），又没有长按（onLongPress），然后Touchup时触发。  
  
  
void  onLongPress(MotionEvent e)  
Touch了不移动一直Touch down时触发  
  
  
boolean  onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY)  
Touch了滑动时触发。  
  
  
boolean  onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY)  
Touch了滑动一点距离后，up时触发。  
  
  
void  onShowPress(MotionEvent e)  
Touch了还没有滑动时触发  
（与onDown，onLongPress比较，onDown只要Touch down一定立刻触发。  
而Touchdown后过一会没有滑动先触发onShowPress再是onLongPress。  
所以Touchdown后一直不滑动，onDown->onShowPress->onLongPress这个顺序触发。）  
  
  
boolean  onDown(MotionEvent e)  
Touch down时触发  
  
  
boolean  onDoubleTap(MotionEvent e)  
双击的第二下Touch down时触发  
  
  
boolean  onDoubleTapEvent(MotionEvent e)  
双击的第二下Touch down和up都会触发，可用e.getAction()区分。