## 1.1 事件类型

主要的事件类型有:

ACTION\_DOWN: 表示用户开始触摸.

ACTION\_MOVE: 表示用户在移动(手指或者其他)

ACTION\_UP:表示用户抬起了手指

ACTION\_CANCEL:表示手势被取消了,一些关于这个事件类型的讨论见:http://stackoverflow.com/questions/11960861/what-causes-a-motionevent-action-cancel-in-android

还有一个不常见的:

ACTION\_OUTSIDE: 表示用户触碰超出了正常的UI边界.

但是对于多点触控的支持,Android加入了以下一些事件类型.来处理,如另外有手指按下了,

有的手指抬起来了.等等:

ACTION\_POINTER\_DOWN:有一个非主要的手指按下了.

ACTION\_POINTER\_UP:一个非主要的手指抬起来了

## 1.2 事件发生的位置,x,y轴

getRawX() 获得事件发生时,触摸的中间区域在屏幕的X轴的值.

getRawY() 获得事件发生时,触摸的中间区域在屏幕的Y轴的值.

在多点触控中还可以通过:

getX(int pointerIndex) ,来获得对应手指事件的发生位置(相对于父布局). 获得Y轴用getY(int pointerIndex)

## 1.3 MotionEvent 中getAction与getActionMasked区别

mAction的低8位即0x00到0xff用来表示动作的类型信息。

例如：MotionEvent#ACTION\_DOWN的值是 0,即0x00。

MotionEvent#ACTION\_UP的值是 1，即0x01。

Android是支持多点触控的。MotionEvent带有触控点索引信息，在mAction的第二个8位中存储。例如，如果mAction的值是0x0000，则表示是第一个触控点的ACTION\_DOWN操作。

如果mAction的值是0x0100呢，则表示是第二个触控点的ACTION\_DOWN操作。

第三个的ACTION\_DOWN呢？相信你可以推出来是0x0200。

总而言之，mAction时的低8位（也就是0-7位）是动作类型信息。

mAction的8-15位呢，是触控点的索引信息。（即表示是哪一个触控点的事件）。

动作类型只要0-255就可以了，动作类型，mPointer也是。这样做的好处，是节约内容。如onMeasure中的MeasureSpec，也采用类似的存储方式。

## 1.4 多点触摸事件

 ACTION\_POINTER\_DOWN:代表用户**又**使用一个手指触摸到屏幕上，每次增加一个触摸点，就会收到ACTION\_POINTER\_DOWN。第一个触摸点是ACTION\_ DOWN事件。

 ACTION\_POINTER\_UP:代表用户的一个手指离开了触摸屏，但是还有其他手指还在触摸屏上。它与ACTION\_UP的区别就是，它是在多个触摸点中的一个触摸点消失时（此时，还有触摸点存在，也就是说用户还有手指触摸屏幕）产生，而ACTION\_UP最后一个触摸点消失时才会触发此事件。