

[首页](#) [ANDROID](#) [互联网](#) [杂乱无章](#) [科技资讯](#) [程序员人生](#) [程序员笑话](#) [编程技术](#) [网址导航](#)

## 3.4 TouchListener PK OnTouchEvent + 多点触碰

分类 [Android 基础入门教程](#)

**Android 基础入门教程(Q群号 : 153836263)**

### 本节引言：

如题，本节给大家带来的是**TouchListener**与**OnTouchEvent**的比较，以及多点触碰的知识点！TouchListener是基于监听的，而OnTouchEvent则是基于回调的！下面通过两个简单的例子来加深大家的理解！

### 1.基于监听的TouchListener

代码示例：

实现效果图：



实现代码：main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

- 1.0 Android基础入门教程
- 1.0.1 2015年最新Android基...
- 1.1 背景相关与系统架构分析
- 1.2 开发环境搭建
- 1.2.1 使用Eclipse + ADT + S...
- 1.2.2 使用Android Studio开...
- 1.3 SDK更新不了问题解决
- 1.4 Genymotion模拟器安装
- 1.5.1 Git使用教程之本地仓...
- 1.5.2 Git之使用GitHub搭建...
- 1.6 .9(九妹)图片怎么玩
- 1.7 界面原型设计
- 1.8 工程相关解析(各种文件...
- 1.9 Android程序签名打包
- 1.11 反编译APK获取代码&...
- 2.1 View与ViewGroup的概念
- 2.2.1 LinearLayout(线性布局)
- 2.2.2 RelativeLayout(相对布...
- 2.2.3 TableLayout(表格布局)
- 2.2.4 FrameLayout(帧布局)
- 2.2.5 GridLayout(网格布局)
- 2.2.6 AbsoluteLayout(绝对...
- 2.3.1 TextView(文本框)详解
- 2.3.2 EditText(输入框)详解
- 2.3.3 Button(按钮)与ImageB...
- 2.3.4 ImageView(图像视图)
- 2.3.5.RadioButton(单选按钮...
- 2.3.6 开关按钮ToggleButton...
- 2.3.7 ProgressBar(进度条)
- 2.3.8 SeekBar(拖动条)
- 2.3.9 RatingBar(星级评分条)
- 2.4.1 ScrollView(滚动条)

```
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MyActivity">
```

```
<ImageView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imgtouch"
    android:background="@drawable/touch"/>
```

```
</RelativeLayout>
```

[反馈](#)[2.4.2 Date & Time组件\(上\)](#)[2.4.3 Date & Time组件\(下\)](#)[2.4.4 Adapter基础讲解](#)[2.4.5 ListView简单实用](#)[2.4.6 BaseAdapter优化](#)[2.4.7 ListView的焦点问题](#)[2.4.8 ListView之checkbox错...](#)[2.4.9 ListView的数据更新问题](#)[2.5.0 构建一个可复用的自定...](#)[2.5.1 ListView Item多布局的...](#)[2.5.2 GridView\(网格视图\)的...](#)[2.5.3 Spinner\(列表选项框\)...](#)[2.5.4 AutoCompleteTextVie...](#)[2.5.5 ExpandableListView\(...](#)[2.5.6 ViewPager\(翻转视图\)...](#)[2.5.7 Toast\(吐司\)的基本使用](#)[2.5.8 Notification\(状态栏通...](#)[2.5.9 AlertDialog\(对话框\)详解](#)[2.6.0 其他几种常用对话框基...](#)[2.6.1 PopupWindow\(悬浮框...](#)[2.6.2 菜单\(Menu\)](#)[2.6.3 ViewPager的简单使用](#)[2.6.4 DrawerLayout\(官方侧...](#)[3.1.1 基于监听的事件处理机制](#)[3.2 基于回调的事件处理机制](#)[3.3 Handler消息传递机制浅析](#)[3.4 TouchListener PK OnTo...](#)[3.5 监听EditText的内容变化](#)[3.6 响应系统设置的事件\(Co...](#)[3.7 AsyncTask异步任务](#)[3.8 Gestures\(手势\)](#)[4.1.1 Activity初学乍练](#)[4.1.2 Activity初窥门径](#)[4.1.3 Activity登堂入室](#)[4.2.1 Service初涉](#)[4.2.2 Service进阶](#)[4.2.3 Service精通](#)[4.3.1 BroadcastReceiver牛...](#)[4.3.2 BroadcastReceiver庖...](#)[4.4.1 ContentProvider初探](#)

## MainActivity.java

```
public class MyActivity extends ActionBarActivity {

    private ImageView imgtouch;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my);

        imgtouch = (ImageView) findViewById(R.id.imgtouch);
        imgtouch.setOnClickListener(new View.OnClickListener() {

            @Override
            public boolean onTouch(View v, MotionEvent event) {

                Toast.makeText(getApplicationContext(), "你通过监
听器模式:OnClickListener摸了伦家~", Toast.LENGTH_LONG).show();

                return true;
            }
        });
    }
}
```

### 代码解析：

就是简单的设置一个ImageView,然后setOnClickListener,重写onTouch方法即可!很简单,其实这个在帧布局那一节已经有个例子了,还记得那个随手指移动的萌妹子吗?

## OnTouchListener相关方法与属性:

**onTouch(View v, MotionEvent event):**这里面的参数依次是触发触摸事件的组件,触碰事件event 封装了触发事件的详细信息,同样包括事件的类型、触发时间等信息。比如event.getX(),event.getY()我们也可以对触摸的动作类型进行判断,使用event.getAction()再进行判断;如:

event.getAction == MotionEvent.ACTION\_DOWN : 按下事件

event.getAction == MotionEvent.ACTION\_MOVE:移动事件

event.getAction == MotionEvent.ACTION\_UP:弹起事件

## 2.基于回调的onTouchEvent( )方法

同样是触碰事件,但是onTouchEvent更多的是用于自定义的view,所有的view类中都重写了该方法,而这种触摸事件是基于回调的,也就是说:如果我们返回的值是false的话,那么事件会继续向外传播,由外面的容器或者Activity进行处理!当然还涉及到了手势(Gesture),这个我们会在后面进行详细的讲解!onTouchEvent其实和onTouchListener是类似的,只是处理机制不用,前者是回调,后者是监听模式!

**代码示例：** 定义一个简单的view，绘制一个蓝色的小圆，可以跟随手指进行移动

**实现代码：** MyView.java

```
public class MyView extends View{
    public float X = 50;
    public float Y = 50;

    //创建画笔
    Paint paint = new Paint();

    public MyView(Context context, AttributeSet set)
    {
        super(context, set);
    }

    @Override
    public void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        paint.setColor(Color.BLUE);
        canvas.drawCircle(X, Y, 30, paint);
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        this.X = event.getX();
        this.Y = event.getY();
        //通知组件进行重绘
        this.invalidate();
        return true;
    }
}
```

**main.xml:**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
```

4.4.2 ContentProvider再探...

4.5.1 Intent的基本使用

4.5.2 Intent之复杂数据的传递

5.1 Fragment基本概述

5.2.1 Fragment实例精讲—...

5.2.2 Fragment实例精讲—...

5.2.3 Fragment实例精讲—...

5.2.4 Fragment实例精讲—...

5.2.5 Fragment实例精讲—...

6.1 数据存储与访问之——文...

6.2 数据存储与访问之——S...

6.3.1 数据存储与访问之——...

6.3.2 数据存储与访问之——...

7.1.1 Android网络编程要学...

7.1.2 Android Http请求头与...

7.1.3 Android HTTP请求方...

7.1.4 Android HTTP请求方...

7.2.1 Android XML数据解析

7.2.2 Android JSON数据解析

7.3.1 Android 文件上传

7.3.2 Android 文件下载 ( 1 )

7.3.3 Android 文件下载 ( 2 )

7.4 Android 调用 WebService

7.5.1 WebView(网页视图)基...

7.5.2 WebView和JavaScip...

7.5.3 Android 4.4后WebVie...

7.5.4 WebView文件下载

7.5.5 WebView缓存问题

7.5.6 WebView处理网页返...

7.6.1 Socket学习网络基础准备

7.6.2 基于TCP协议的Socket...

7.6.3 基于TCP协议的Socket...

7.6.4 基于UDP协议的Socke...

8.1.1 Android中的13种Draw...

8.1.2 Android中的13种Draw...

8.1.3 Android中的13种Draw...

8.2.1 Bitmap(位图)全解析 P...

8.2.2 Bitmap引起的OOM问题

8.3.1 三个绘图工具类详解

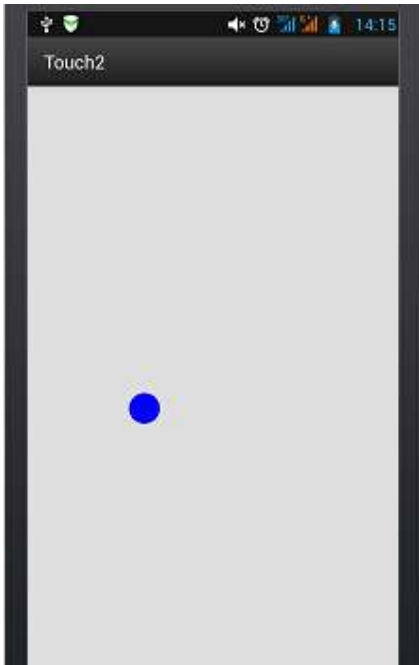
8.3.2 绘图类实战示例

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MyActivity">
        <example.jay.com.touch2.MyView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </RelativeLayout>

```

实现效果图：



用手指触摸进行移动~

### 3.多点触碰

原理类的东西：

所谓的多点触碰就是多个手指在屏幕上进行操作，用的最多的估计是放大缩功能吧，比如很多的图片浏览器都支持缩放！理论上Android系统本身可以处理多达256个手指的触摸，当然这取决于手机硬件的支持；不过支持多点触摸的手机一般支持2-4个点，当然有些更多！我们发现前面两点都有用到MotionEvent，MotionEvent代表的是一个触摸事件；前我们可以根据event.getAction() & MotionEvent.ACTION\_MASK来判断是哪种操作，除了上面介绍的三种单点操作外，还有两个多点专用的操作：

MotionEvent.ACTION\_POINTER\_DOWN:当屏幕上已经有一个点被按住，此时再按下其他点时触发。

MotionEvent.ACTION\_POINTER\_UP:当屏幕上有多个点被按住，松开其中一个点时触发（即非最后一个点被放开时）。

简单的流程大概是这样：

当我们一个手指触摸屏幕 ——> 触发ACTION\_DOWN事件

接着有另一个手指也触摸屏幕 ——> 触发ACTION\_POINTER\_DOWN事件,如果还有其他手指触摸，继续触发

有一个手指离开屏幕 ——> 触发ACTION\_POINTER\_UP事件，继续有手指离开，继续触发

- 8.3.3 Paint API之—— Mask...
- 8.3.4 Paint API之—— Xferm...
- 8.3.5 Paint API之—— Xferm...
- 8.3.6 Paint API之—— Xferm...
- 8.3.7 Paint API之—— Xferm...
- 8.3.8 Paint API之—— Xferm...
- 8.3.9 Paint API之—— Color...
- 8.3.10 Paint API之—— Colo...
- 8.3.11 Paint API之—— Colo...
- 8.3.12 Paint API之—— Path...
- 8.3.13 Paint API之—— Sha...
- 8.3.14 Paint几个枚举/常量值...
- 8.3.15 Paint API之——Type...
- 8.3.16 Canvas API详解(Part 1)
- 8.3.17 Canvas API详解(Part...
- 8.3.18 Canvas API详解(Part...
- 8.4.1 Android动画合集之帧...
- 8.4.2 Android动画合集之补...
- 8.4.3 Android动画合集之属...
- 8.4.4 Android动画合集之属...
- 9.1 使用SoundPool播放音...
- 9.2 MediaPlayer播放音频与...
- 9.3 使用Camera拍照
- 9.4 使用MediaRecord录音
- 10.1 TelephonyManager(电...
- 10.2 SmsManager(短信管理...
- 10.3 AudioManager(音频管...
- 10.4 Vibrator(振动器)
- 10.5 AlarmManager(闹钟服务)
- 10.6 PowerManager(电源服...
- 10.7 WindowManager(窗口...
- 10.8 LayoutInflater(布局服务)
- 10.9 WallpaperManager(壁...
- 10.10 传感器专题(1)——相...
- 10.11 传感器专题(2)——方...
- 10.12 传感器专题(3)——加...
- 10.12 传感器专题(4)——其...
- 10.14 Android GPS初涉
- 11.0 《2015最新Android基...

当最后一个手指离开屏幕 ——> 触发ACTION\_UP事件

而且在整个过程中，ACTION\_MOVE事件会一直不停地被触发

我们可以通过event.getX(int)或者event.getY(int)来获得不同触摸点的位置：比如event.getX(0)可以获得第一个接触点的X坐标，event.getX(1)获得第二个接触点的X坐标这样... 另外，我们还可以在调用MotionEvent对象的getPointerCount()方法判断当前有多少个手指在触摸~

## 代码示例：

好吧，我们来写个最常见的单指拖动图片，双指缩放图片的示例吧：

实现效果图：



实现代码：

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <ImageView
        android:id="@+id/img_test"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="matrix"
        android:src="@drawable/pic1" />

</RelativeLayout>
```

## MainActivity.java

```
package com.jay.example.edittextdemo;
```

```
import android.app.Activity;
import android.graphics.Matrix;
import android.graphics.PointF;
import android.os.Bundle;
import android.util.FloatMath;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnTouchListener;
import android.widget.ImageView;

public class MainActivity extends Activity implements OnTouchListener {

    private ImageView img_test;

    // 缩放控制
    private Matrix matrix = new Matrix();
    private Matrix savedMatrix = new Matrix();

    // 不同状态的表示:
    private static final int NONE = 0;
    private static final int DRAG = 1;
    private static final int ZOOM = 2;
    private int mode = NONE;

    // 定义第一个按下的点, 两只接触点的重点, 以及出事的两指按下的距离:
    private PointF startPoint = new PointF();
    private PointF midPoint = new PointF();
    private float oriDis = 1f;

    /*
     * (non-Javadoc)
     *
     * @see android.app.Activity#onCreate(android.os.Bundle)
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        img_test = (ImageView) this.findViewById(R.id.img_test);
    }

    img_test.setOnTouchListener(this);

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        ImageView view = (ImageView) v;
        switch (event.getAction() & MotionEvent.ACTION_MASK) {
            // 单指
            case MotionEvent.ACTION_DOWN:
                matrix.set(view.getImageMatrix());
                savedMatrix.set(matrix);
```

```

        startPoint.set(event.getX(), event.getY());
        mode = DRAG;
        break;
    // 双指
    case MotionEvent.ACTION_POINTER_DOWN:
        oriDis = distance(event);
        if (oriDis > 10f) {
            savedMatrix.set(matrix);
            midPoint = middle(event);
            mode = ZOOM;
        }
        break;
    // 手指放开
    case MotionEvent.ACTION_UP:
    case MotionEvent.ACTION_POINTER_UP:
        mode = NONE;
        break;
    // 单指滑动事件
    case MotionEvent.ACTION_MOVE:
        if (mode == DRAG) {
            // 是一个手指拖动
            matrix.set(savedMatrix);
            matrix.postTranslate(event.getX() - startPoint.x, event.getY() - startPoint.y);
        } else if (mode == ZOOM) {
            // 两个手指滑动
            float newDist = distance(event);
            if (newDist > 10f) {
                matrix.set(savedMatrix);
                float scale = newDist / oriDis;
                matrix.postScale(scale, scale, midPoint.x, midPoint.y);
            }
        }
        break;
    }
    // 设置ImageView的Matrix
    view.setImageMatrix(matrix);
    return true;
}

// 计算两个触摸点之间的距离
private float distance(MotionEvent event) {
    float x = event.getX(0) - event.getX(1);
    float y = event.getY(0) - event.getY(1);
    return FloatMath.sqrt(x * x + y * y);
}

// 计算两个触摸点的中点
private PointF middle(MotionEvent event) {
    float x = event.getX(0) + event.getX(1);
    float y = event.getY(0) + event.getY(1);
    return new PointF(x / 2, y / 2);
}

```

```
}  
  
}
```

本节小结：

好的，关于TouchListener和OnTouchEvent以及多点触碰就到这里~

大家都在玩手机而我却用手机学英语

邮箱订阅《每日英语》

每天坚持五分钟, 随时随地学英语!

立即行动



在线实例	字符集&工具	最新更新
<ul style="list-style-type: none"><li>· HTML 实例</li><li>· CSS 实例</li><li>· JavaScript 实例</li><li>· Ajax 实例</li><li>· jQuery 实例</li><li>· XML 实例</li><li>· Java 实例</li></ul>	<ul style="list-style-type: none"><li>· HTML 字符集设置</li><li>· HTML ASCII 字符集</li><li>· HTML ISO-8859-1</li><li>· HTML 实体符号</li><li>· HTML 拾色器</li><li>· JSON 格式化工具</li></ul>	<ul style="list-style-type: none"><li>· Swift 正式开源</li><li>· PHP 7 正式发布</li><li>· Shell 编程快速入门</li><li>· Shell 文件包含</li><li>· Shell 输入/输出...</li><li>· Shell printf 命令</li><li>· Shell 基本运算符</li></ul>

- 站点信息
- 意见反馈
  - 免责声明
  - 关于我们
  - 文章归档

关注微信



Copyright © 2013-2015 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1