



http://ticktick.blog.51cto.com 【复制】 【订阅】

博客 | 写博文 | 帮助

首页 | C/C++ 编程 | Linux 编程 | Android 应用开发 | Android 内核开发 | 音视频/多媒体开发 | 嵌入式系统 | 开发经验总结 | 我的作品

tickTick 的BLOG

相关视频课程

更多



写留言

去学院学习

发消息

加好友链接

进家园 加好友

博客统计信息

51CTO推荐博客

用户名: tickTick

文章数: 153

评论数: 383

访问量: 1197282

无忧币: 4321

博客积分: 4732

博客等级: 8

注册日期: 2009-06-13

热门专题

更多>>



每天5分钟玩转  
OpenStack  
阅读量: 5863



【51CTO三周年】我在  
学院不得不说的收获  
阅读量: 12276



从菜鸟到老鸟-教你玩  
转Mac操作系统  
阅读量: 334784



QT学习之路: 从入门到  
精通  
阅读量: 1039539

热门文章

undefined reference问题总结

c++构造函数详解

C++串口编程实例

入门视频采集与处理（学会分析YUV..

c++异常处理机制示例及讲解

Doxygen详细介绍（三）（Doxygen..



七日成蝶-（C++系列第  
六部）-C++运算符重载  
2412人学习



零基础学通C语言系列  
大全之友元函数, 友元  
99人学习



这可能是你见过“最牛  
逼”的C++课程(共81课  
95531人学习

博主的更多文章>>

## 原创 Android开发实践：为什么要继承onMeasure()

2014-08-14 19:55:34

标签: 重载 Android 自定义 View ViewGroup

原创作品，允许转载，转载时请务必以超链接形式标明文章 [原始出处](#)、作者信息和本声明。否则将追究法律责  
任。 <http://ticktick.blog.51cto.com/823160/1540134>

Android开发中偶尔会用到自定义View，一般情况下，自定义View都需要继承View类的onMeasure方法，那么，为什么要继承onMeasure()函数呢？什么情况下要继承onMeasure()？系统默认的onMeasure()函数行为是怎样的？本文就探究这些问题。

首先，我们写一个自定义View，直接调用系统默认的onMeasure函数，看看会是怎样的现象：

```
1 package com.titctick.customview;
2
3 import android.content.Context;
4 import android.util.AttributeSet;
5 import android.view.View;
6
7 public class CustomView extends View {
8
9     public CustomView(Context context) {
10         super(context);
11     }
12
13     public CustomView(Context context, AttributeSet attrs) {
14         super(context, attrs);
15     }
16
17     @Override
18     protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
19         super.onMeasure(widthMeasureSpec, heightMeasureSpec);
20     }
21
22 }
```

1. 父控件使用match\_parent，CustomView使用match\_parent

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical">
6
7     <com.titctick.customview.CustomView
8         android:layout_width="match_parent"
9         android:layout_height="match_parent"
10        android:layout_margin="10dp"
11        android:background="@android:color/black"/>
12
13 </LinearLayout>
```

这里加了10dp的margin并且把View的背景设置为了黑色，是为了方便辨别我们的CustomView，效果如下：

谈谈RTP传输中的负载类型和时间戳

虚拟串口软件VSPM使用手册

搜索BLOG文章

搜索

最近访客

zhang..

zzy853

fdsfd..

fishc..

guoto..

秋水齐飞

dleho

wb57c..

w1412..

lison..

loveand5

ssun125

最新评论

- xhcx\_0321:

楼主，下载不了，下载下来的压缩包..
- Harry\_陆:

我想问一下，源码6.0 ROM升级的逻辑..
- hnraysir:

我是加你微信的eleso\*
- hnraysir:

public static void loge(String m..
- zxccxzzxz:

感谢楼主分享，正好遇到这个问题，..

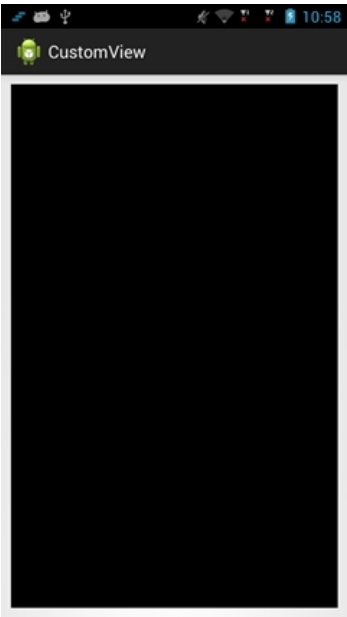
51CTO推荐博文

更多>>

- 我动了十年的老代码…
- 雾里散步——这次聊聊自己
- 【产品日记】V1.2版本发版——顺..
- 欢迎加入51CTO官方博客QQ群112163514
- 转变———大数据的抉择之路
- 关于密码那些事（一）
- 不懂技术的人请不要对懂技术的人..
- 北京，一个让屌丝望而却步的城市
- 追随你心 —— 聊聊VMC..
- 中小企业的服务器虚拟化探讨
- 是什么浪费了我工作的时间

友情链接

- IT精品课程
- 我的微博
- 我的微博
- Dian团队主页



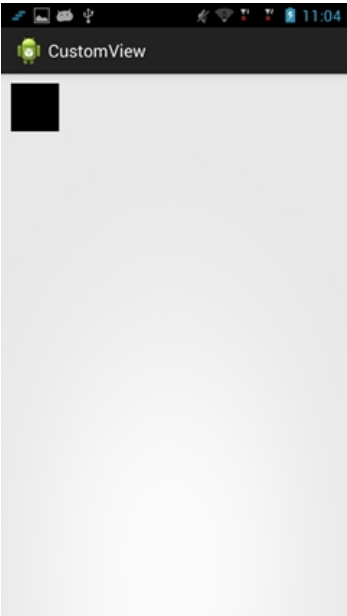
我们可以看到，默认情况下，如果父控件和CustomView都使用match\_parent，则CustomView会充满父控件。

2. 父控件使用match\_parent，CustomView使用wrap\_content

把layout文件中，CustomView的layout\_width/layout\_height替换为wrap\_content，你会发现，结果依然是充满父控件。

3. 父控件使用match\_parent，CustomView使用固定的值

把layout文件中，CustomView的layout\_width/layout\_height替换为50dp，你会发现，CustomView的显示结果为50dpx50dp，如图所示：



4. 父控件使用固定的值，CustomView使用match\_parent或者wrap\_content

那么，如果把父控件的layout\_width/layout\_height替换为50dp，CustomView设置为match\_parent或者wrap\_content，你会发现，CustomView的显示结果也是为50dpx50 dp。

5 结论

如果自定义的CustomView采用默认的onMeasure函数，行为如下：

(1) CustomView设置为 match\_parent 或者 wrap\_content 没有任何区别，其显示大小由父控件决定，它会填满整个父控件的空间。

(2) CustomView设置为固定的值，则其显示大小为该设定的值。

如果你的自定义控件的大小计算就是跟系统默认的行为一致的话，那么你就不需要重写onMeasure函数了。

## 6. 怎样编写onMeasure函数

系统默认的onMeasure函数的行为就讨论到这，下面也说说怎样重写onMeasure函数，以及onMeasure函数的基本原理，关键部分在代码中以注释的形式给出了，仅供参考：

```
1 package com.titcktick.customview;
2
3 import android.content.Context;
4 import android.util.AttributeSet;
5 import android.view.View;
6
7 public class CustomView extends View {
8
9     private static final int DEFAULT_VIEW_WIDTH = 100;
10    private static final int DEFAULT_VIEW_HEIGHT = 100;
11
12    public CustomView(Context context) {
13        super(context);
14    }
15
16    public CustomView(Context context, AttributeSet attrs) {
17        super(context, attrs);
18    }
19
20    @Override
21    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
22
23        int width = measureDimension(DEFAULT_VIEW_WIDTH, widthMeasureSpec);
24        int height = measureDimension(DEFAULT_VIEW_HEIGHT, heightMeasureSpec);
25
26        setMeasuredDimension(width, height);
27    }
28
29    protected int measureDimension( int defaultSize, int measureSpec ) {
30
31        int result = defaultSize;
32
33        int specMode = MeasureSpec.getMode(measureSpec);
34        int specSize = MeasureSpec.getSize(measureSpec);
35
36        //1. layout给出了确定的值，比如：100dp
37        //2. layout使用的是match_parent，但父控件的size已经可以确定了，比如设置的是具体的值或者match_parent
38        if (specMode == MeasureSpec.EXACTLY) {
39            result = specSize; //建议：result直接使用确定值
40        }
41        //1. layout使用的是wrap_content
42        //2. layout使用的是match_parent，但父控件使用的是确定的值或者wrap_content
43        else if (specMode == MeasureSpec.AT_MOST) {
44            result = Math.min(defaultSize, specSize); //建议：result不能大于specSize
45        }
46        //UNSPECIFIED,没有任何限制，所以可以设置任何大小
47        //多半出现在自定义的父控件的情况下，期望由自控件自行决定大小
48        else {
49            result = defaultSize;
50        }
51
52        return result;
53    }
54 }
```

这样重载了onMeasure函数之后，你会发现，当CustomView使用match\_parent的时候，它会占满整个父控件，而当CustomView使用wrap\_content的时候，它的大小则是代码中定义的默认大小100x100像素。当然，你也可以根据自己的需求改写measureDimension()的实现。

关于onMeasure的讨论就介绍到这儿了，有任何疑问欢迎留言或者来信lujun.hust@gmail.com交流，或者关注我的新浪微博 @卢\_俊 获取最新的文章和资讯。

本文出自 “Jhuster的专栏” 博客，请务必保留此出处<http://ticktick.blog.51cto.com/823160/1540134>

分享至:

收藏 +



上一篇

写给新入职的毕业生们

下一篇

Android开发实践：自定义ViewGroup的onLayout(..)



相关文章

- android游戏开发(四)自定义view
- android自定义View的用法
- android自定义控件实例
- Android自定义View研究(六)--View中的原点坐..
- Android自定义View研究(五)--View的大小

职位推荐

- 执行主策
- Android开发工程师（高级）
- 安卓开发工程师
- IOS开发工程师
- IOS逆向开发工程师

文章评论

- [1楼]

祝你幸福365

回复
- 2015-03-28 15:18:32
- 写的很好，很有用
- 

[2楼]

w9xhcn

回复

2015-07-23 22:53:58

感谢！！

发表评论

昵 称 [登录](#) [快速注册](#)

验证码： 请点击后输入验证码 [博客过2级，无需填写验证码](#)

内 容

Copyright By 51CTO.COM 版权所有

51CTO 技术博客

