

## 个人资料



xu\_fu

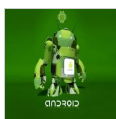


访问: 160473次  
积分: 2460  
等级: [BLOG > 5](#)  
排名: 第6819名

原创: 74篇  
转载: 7篇  
译文: 5篇  
评论: 116条

## 文章搜索

## 博客专栏



**Android动效实验室**  
文章: 10篇  
阅读: 27078

## 文章分类

[Android](#) (47)  
[设计模式](#) (4)  
[cocos2d-x](#) (1)  
[Java](#) (1)  
[Swift](#) (2)

来源: [android](#)

[2015年04月](#) (5)  
[2015年03月](#) (4)  
[2015年02月](#) (1)  
[2015年01月](#) (1)  
[2014年12月](#) (3)

[展开](#)[Markdown博文大赛清新开启](#) [天天爱答题 一大波C币袭来](#) [寻找Java大牛!](#) [一次拿下软考, 我自有锦囊妙计!](#)

## Android - View绘图原理总结

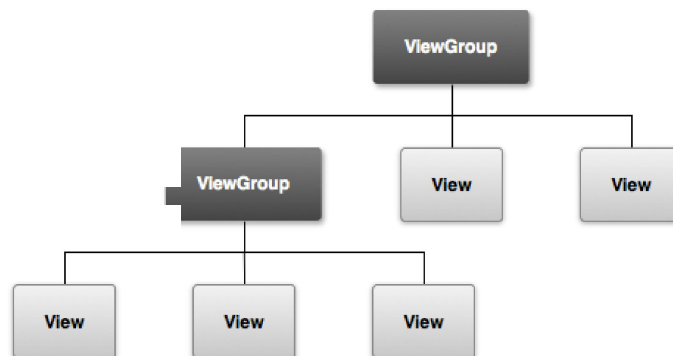
分类: [Android](#)

2012-08-04 19:03

8929人阅读

[评论\(1\)](#)[收藏](#)[举报](#)[android](#) [layout](#) [扩展](#) [工作](#) [图形](#)[目录\(?\)](#) [\[+\]](#)

Android系统的视图结构的设计也采用了组合模式，即View作为所有图形的基类，Viewgroup对View继承扩展为视图容器类，由此就得到了视图部分的基本结构--树形结构



## View定义了绘图的基本操作

基本操作由三个函数完成：measure()、layout()、draw()，其内部又分别包含了onMeasure()、onLayout()、onDraw()三个子方法。具体操作如下：

## 1、measure操作

measure操作主要用于计算视图的大小，即视图的宽度和长度。在view中定义为final类型，要求子类不能修改。measure()函数中又会调用下面的函数：

(1) onMeasure(), 视图大小的将在这里最终确定，也就是说measure只是对onMeasure的一个包装，子类可以覆写onMeasure()方法实现自己的计算视图大小的方式，并通过setMeasuredDimension(width, height)保存计算结果。

## 2、layout操作

layout操作用于设置视图在屏幕中显示的位置。在view中定义为final类型，要求子类不能修改。layout()函数中有两个基本操作：

(1) setFrame ( l,t,r,b )，l,t,r,b即子视图在父视图中的具体位置，该函数用于将这些参数保存起来；

(2) onLayout(), 在View中这个函数什么都不会做，提供该函数主要是为viewGroup类型布局子视图用的；

## 3 draw操作

在屏幕上，到这里也就完成了整个的视图绘制工作。子类也应该修改该方法，因为内部定义了绘图的基本操作：

(1) 绘制背景；

(2) 如果要视图显示渐变框，这里会做一些准备工作；

(3) 绘制视图本身，即调用onDraw()函数。在view中onDraw()是个空函数，也就是说具体的视图都要覆写该函数来实现自己的显示（比如TextView在这里实现了绘制文字的过程）。而对于ViewGroup则不需要实现该函数，因为作为容器是“没有内容”的，其包含了多个子view，而

#### 阅读排行

Android图片下载缓存库f	(13791)
Android教程示例BitmapI	(13551)
Android - View绘图原理.	(8893)
Android之Bitmap大图加:	(8402)
Android图片高斯模糊的-	(7779)
Android瀑布流控件——	(6306)
Android实现书籍翻页效!	(5678)
Android自拍相机应用——	(4694)
Android系统截屏功能提!	(4412)
ViewPager	(4013)

#### 评论排行

Android瀑布流控件——	(21)
Android教程示例BitmapI	(18)
Android实现书籍翻页效!	(14)
Android系统截屏功能提!	(12)
Android闪闪发光字体效!	(9)
Android保存图片到系统I	(9)
Android图片高斯模糊的-	(5)
Android图片下载缓存库f	(4)
Android属性动画初步	(3)
流媒体Live555（五）——	(3)

#### 推荐文章

- \* [【ShaderToy】开篇](#)
- \* [FFmpeg源代码简单分析: avio\\_open2\(\)](#)
- \* [技能树之旅: 从模块分离到测试](#)
- \* [Qt5官方demo解析集36——Wiggly Example](#)
- \* [Unity3d HDR和Bloom效果（高动态范围图像和泛光）](#)
- \* [Android的Google官方设计指南](#)

#### 最新评论

- Android自拍相机应用——图片的 tyong: good
- Android系统截屏功能提取 asd2341: 只能截本activity，通知栏空白
- Android Easing Interpolator—— sengeiou: 顶一个！！！！
- Android系统截屏功能提取 aguica: 好评
- Android保存图片到系统图库 hunan1111: 为什么小米手机还是没有刷新出来呢（系统图库中查看该文件夹下没有），但是在我用文件管理查看指定的文件夹...
- Android保存图片到系统图库 hunan1111: 为什么小米手机还是没有刷新出来呢（系统图库中查看该文件夹下没有），但是在我用文件管理查看指定的文件夹...
- ViewPager dujiehong: 思路挺清晰的，感谢分享
- Android系统截屏功能提取 xu\_fu: @loveheronly:在takeScreenShot里面得到截图的bitmap，这里就可以保存文...
- Android系统截屏功能提取 loveheronly: 你好，怎么保存截图的图片呢

子View已经实现了自己的绘制方法，因此只需要告诉子view绘制自己就可以了，也就是下面的dispatchDraw()方法;

（4）绘制子视图，即dispatchDraw()函数。在view中这是个空函数，具体的视图不需要实现该方法，它是专门为容器类准备的，也就是容器类必须实现该方法；

（5）如果需要（应用程序调用了setVerticalFadingEdge或者setHorizontalFadingEdge），开始绘制渐变框；

（6）绘制滚动条；

从上面可以看出自定义View需要最少覆写onMeasure()和onDraw()两个方法。

### ViewGroup中的扩展操作：

首先Viewgroup是一个抽象类。

#### 1、对子视图的measure过程

（1）measureChildren()，内部使用一个for循环对子视图进行遍历，分别调用子视图的measure()方法；

（2）measureChild()，为指定的子视图measure，会被measureChildren调用；

（3）measureChildWithMargins()，为指定子视图考虑了margin和padding的measure；

以上三个方法是ViewGroup提供的3个对子view进行测量的参考方法，设计者需要实际中首先覆写onMeasure()，之后再对子view进行遍历measure，这时候就可以使用以上三个方法，当然也可以自定义方法进行遍历。

#### 2、对子视图的layout过程

在ViewGroup中onLayout()被定义为abstract类型，也就是具体的容器必须实现此方法来安排子视图的布局位置，实现中主要考虑的是视图的大小及视图间的相对位置关系，如gravity、layout\_gravity。

#### 3、对子视图的draw过程

（1）dispatchDraw()，该方法用于对子视图进行遍历然后分别让子视图分别draw，方法内部会首先处理布局动画（也就是说布局动画是在这里处理的），如果有布局动画则会为每个子视图产生一个绘制时间，之后再有一个for循环对子视图进行遍历，来调用子视图的draw方法（实际为下边的drawChild()）；

（2）drawChild()，该方法用于具体调用子视图的draw方法，内部首先会处理视图动画（也就是说视图动画是在这里处理的），之后调用子视图的draw()。

从上面分析可以看出自定义viewGroup的时候需要最少覆写onMeasure()和onLayout()方法，其中onMeasure方法中可以直接调用measureChildren等已有的方法，而onLayout方法就需要设计者进行完整的定义；一般不需要覆写以dispatchDraw()和drawChild()这两个方法，因为上面两个方法已经完成了基本的事情。但是可以通过覆写在该基础之上做一些特殊的效果，比如

```
[java] view plain copy print ?
01.  @Override
02.  protected void dispatchDraw(Canvas canvas) {
03.      // TODO Auto-generated method stub
04.      //
05.      //可以在这里先做一些处理，包括对传入的canvas
06.      //
07.      super.dispatchDraw(canvas); //这里会调用drawChild绘制子视图
08.      //
09.      //所有子视图都绘制完成后这里还可以做一些处理，比如绘制阴影什么的
10.      //
11.
12.  }
```

### 其他

从以上分析可以看出View树的绘制是一个递归的过程，从ViewGroup一直向下遍历，直到所有的子view都完成绘制，那这一切的源头在什么地方（是谁最发起measure、layout和draw的）？当然就是在View树的源头了——ViewRoot！，ViewRoot中包含了窗口的总容器DecorView，ViewRoot中的performTraversal()方法会依次调用decorView的measure、layout、draw方法，从而完成view树的绘制。

#### invalidate()方法

invalidate()方法会导致View树的重新绘制，而且view中的状态标志mPrivateFlags中有一个关于当前视图是否需要重绘的标志位DRAWN，也就是说只有标志位DRAWN置位的视图才需要进行重绘。当视图调用invalidate()方法时，首先会将当前视图的DRAWN标志置位，之后有一个循环调用parent.invalidateChildInParent()，这样会导致从当前视图依次向上遍历直到根视图

Android保存图片到系统图库  
xu\_fu: @baidu\_27000181: bmp  
是一堆二进制数据，先创建一个  
文件，然后把bmp的数据写到了  
这个...

ViewRoot，这个过程会将需要重绘的视图标记DRAWN置位，之后ViewRoot调用performTraversals()方法，完成视图的绘制过程。

参考书籍《Android内核剖析》——柯元旦

上一篇 设计模式学习（4）  
下一篇 数据结构知识整理（一）

主题推荐 android 应用程序 布局 动画 结构

猜你在找

- LayoutInflater 详解
- Android VelocityTracker简介
- Android笔记 IntentService与服务区别
- 分享45个android实例源码很好很强大收藏吧
- android仿win8 metro磁贴布局
- 【精品课程】竟有如此全面的软考资料？“碉堡”
- 【精品课程】HTML 5全掌控
- 【精品课程】Cocos2d-x实战-手把手教你上线项目-
- 【精品课程】Swift语言进阶
- 【精品课程】深入浅出MySQL入门必备

准备好了么？跳吧！ 更多职位尽在 CSDN JOB

Android视频工程师	我要跳槽	Android软件开发工程师	我要跳槽
国广星空视频科技（北京）有限公司	15-20K/月	深圳市聚领信息科技有限公司	7-12K/月
资深Android开发工程师	我要跳槽	Android开发工程师	我要跳槽
杭州随笔记网络技术有限公司	12-25K/月	上海彩亿信息技术有限公司	7-18K/月


下载 超级数据恢复软件

shuju360.net

硬盘、U盘、SD卡的数据都可以轻松恢复 ¥89元/套。立即下载，免费试用！

查看评论

1楼 lucherr 2013-03-26 20:40发表

 不错 学习了， thanks for sharing

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker  
OpenStack VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC  
WAP jQuery BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML  
LBS Unity Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra  
CloudStack FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App  
SpringSide Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP  
HBase Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap