绘制流程参考黑马

View,SurfaceView,GLSurfaceView有什么区别

 View：显示视图，内置画布，提供图形绘制函数、触屏事件、按键事件函数等；**必须在UI主线程内更新画面，速度较慢**

 SurfaceView：基于view视图进行拓展的视图类，更适合2D游戏的开发；**是view的子类，类似使用双缓机制，在新的线程中更新画面所以刷新界面速度比view快。**

 GLSurfaceView：基于SurfaceView视图再次进行拓展的视图类，专用于3D游戏开发的视图；**是SurfaceView的子类，openGL专用。**

View缺乏双缓冲机制

当程序需要更新View上的图像时，程序必须重绘View上显示的整张图片

新线程无法直接更新View组件

<http://www.cnblogs.com/0616--ataozhijia/p/4003380.html> 自定义控件和 绘制流程

一、View结构原理

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。

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

     draw操作利用前两部得到的参数，将视图显示在屏幕上，到这里也就完成了整个的视图绘制工作。子类也不应该修改该方法，因为其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。

二、View类的构造方法

创建自定义控件的3种主要实现方式:

1）继承已有的控件来实现自定义控件: 主要是当要实现的控件和已有的控件在很多方面比较类似, 通过对已有控件的扩展来满足要求。

2）通过继承一个布局文件实现自定义控件，一般来说做组合控件时可以通过这个方式来实现。

    注意此时不用onDraw方法，在构造广告中通过inflater加载自定义控件的布局文件，再addView(view)，自定义控件的图形界面就加载进来了。

3）通过继承view类来实现自定义控件，使用GDI绘制出组件界面，一般无法通过上述两种方式来实现时用该方式。

三、自定义View增加属性的两种方法：

1）在View类中定义。通过构造函数中引入的AttributeSet 去查找XML布局的属性名称，然后找到它对应引用的资源ID去找值。

案例：实现一个带文字的图片（图片、文字是onDraw方法重绘实现

明确需求，确定你想实现的效果 2.确定是使用组合控件的形式还是全新自定义的形式，组合控件即使用多个系统控件来合成一个新控件，你比如titilebar，这种形式相对简单， [参考](http://blog.csdn.net/chdjj/article/details/21193995) 3.如果是完全自定义一个view的话，你首先需要考虑继承哪个类，是View呢，还是ImageView等子类。 4.根据需要去复写View#onDraw、View#onMeasure、View#onLayout方法 5.根据需要去复写dispatchTouchEvent、onTouchEvent方法 6.根据需要为你的自定义view提供自定义属性，即编写attr.xml,然后在代码中通过TypedArray等类获取到自定义属性值 7.需要处理滑动冲突、像素转换等问题

另外一个回答

* 编写attr.xml文件
* 在layout布局文件中引用，同时引用命名空间
* 在自定义控件中进行读取（构造方法拿到attr.xml文件值）
* 覆写onMeasure()方法
* 覆写onLayout()方法
* 理解Activity，View,Window三者关系
* 这个问题真的很不好回答。所以这里先来个算是比较恰当的比喻来形容下它们的关系吧。Activity像一个工匠（控制单元），Window像窗户（承载模型），View像窗花（显示视图）LayoutInflater像剪刀，Xml配置像窗花图纸。  
  1：Activity构造的时候会初始化一个Window，准确的说是PhoneWindow。  
  2：这个PhoneWindow有一个“ViewRoot”，这个“ViewRoot”是一个View或者说ViewGroup，是最初始的根视图。  
  3：“ViewRoot”通过addView方法来一个个的添加View。比如TextView，Button等  
  4：这些View的事件监听，是由WindowManagerService来接受消息，并且回调Activity函数。比如onClickListener，onKeyDown等