在Android应用开发过程中，系统默认的Button是由系统渲染和管理大小的。而我们看到许多成功的应用，都是有着酷炫的外观和使用体验的。因此，我们在开发产品的时候，需要对默认按钮进行美化。在本篇里，笔者结合实践经验，探讨一下自定义背景的按钮、自定义形状的按钮的实现方法。

首先看实现效果截图：



自定义背景的按钮目前有2种实现方式，矢量和位图。

## 1. 矢量图形绘制的方式

矢量图形绘制的方式实现简单，适用于对按钮形状和图案要求不高的场合。步骤如下：

(a) 在drawable文件夹下定义一个圆角矩形，外围轮廓线实线、内填充渐变色，xml代码如下。

*//bg\_default.xml*

*<?***xml version="1.0" encoding="utf-8"***?>*<**layer-list xmlns:android="http://schemas.android.com/apk/res/android"**>  
 <**item**>  
 <**shape android:shape="rectangle"**>  
 <**solid android:color="#FFEC7600"** />  
 <**corners  
 android:bottomLeftRadius="5dip"  
 android:bottomRightRadius="5dip"  
 android:topLeftRadius="5dip"  
 android:topRightRadius="5dip"** />  
 </**shape**>  
 </**item**>  
  
 <**item  
 android:bottom="1px"  
 android:left="1px"  
 android:right="1px"  
 android:top="1px"**>  
 <**shape**>  
 <**gradient  
 android:angle="90"  
 android:centerX="0.5"  
 android:centerY="0.5"  
 android:endColor="#FFFED69E"  
 android:startColor="#FFEC7600"  
 android:type="linear"** />  
 <**corners  
 android:bottomLeftRadius="5dip"  
 android:bottomRightRadius="5dip"  
 android:topLeftRadius="5dip"  
 android:topRightRadius="5dip"** />  
 </**shape**>  
 </**item**>  
</**layer-list**>

同样定义bg\_pressed.xml和bg\_selected.xml，内容相同，就是渐变颜色不同，用于按钮按下后的背景变化效果。

(b) 定义按钮按下后的效果变化描述文件drawable/bg.xml，代码如下。

*//bg.xml*

*<?***xml version="1.0" encoding="utf-8"***?>*<**selector xmlns:android="http://schemas.android.com/apk/res/android"**>  
  
 <**item android:drawable="@drawable/bg\_pressed" android:state\_pressed="true"** />  
 <**item android:drawable="@drawable/bg\_selected" android:state\_focused="true"** />  
 <**item android:drawable="@drawable/bg\_default"** />  
  
</**selector**>

(c) 在你需要的界面定义文件中，如layout/main.xml中定义一个Button控件。

1. <Button
2. android:layout\_width="120dip"
3. android:layout\_height="40dip"
4. android:text="矢量背景按钮" android:background="@drawable/bg\_alibuybutton" />

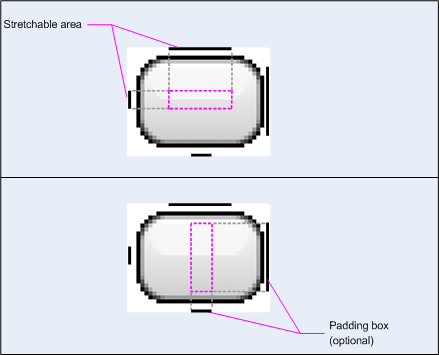
这样，自定义背景的按钮就可以使用了，在实现onClick方法后就可以响应操作。

## 2. 9-patch图片背景方式

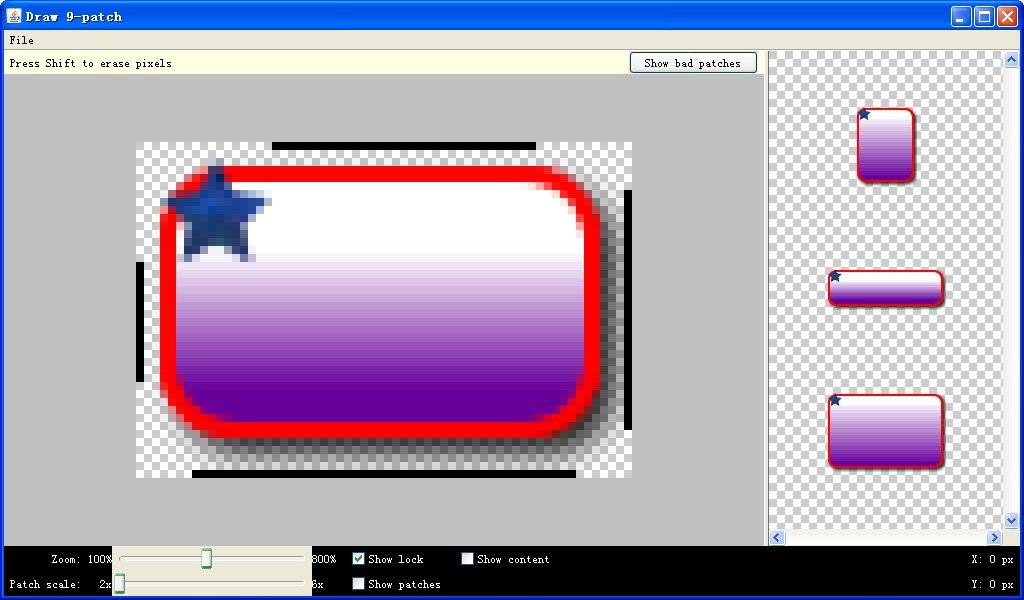
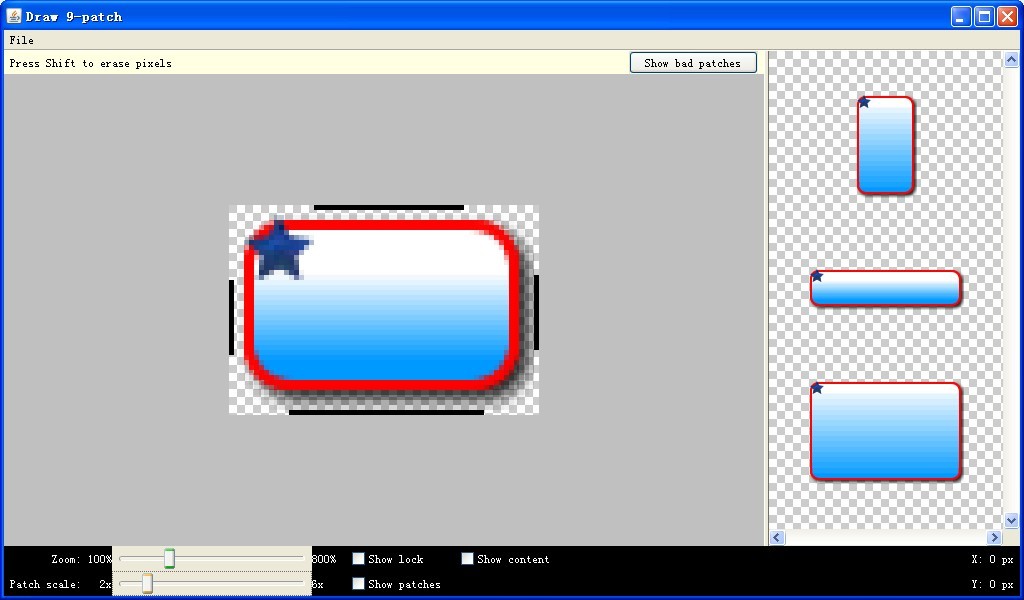
在Android的设计过程中，为了适配不同的手机分辨率，图片大多需要拉伸或者压缩，这样就出现了可以任意调整大小的一种图片格式“.9.png”。这种图片是用于Android开发的一种特殊的图片格式，它的好处在于可以用简单的方式把一张图片中哪些区域可以拉伸，哪些区域不可以拉伸设定好，同时可以把显示内容区域的位置标示清楚。此种方法相对复杂繁琐，但可以制作出更多、更复杂样式的按钮图样。

什么是9-patch格式呢？

9-patch格式，是在Android中特有的一种PNG图片格式，以"\*\*\*.9.png"结尾。此种格式的图片定义了可以伸缩拉伸的区域和文字显示区域，这样，就可以保证在Android开发中对非矢量图进行拉伸而仍然保持美观。如果使用位图而没有经过9-patch处理的话，效果就会像第一张截图中的“普通图片背景按钮”那样被无情的拉伸，影响效果。Android中大量用了这种技术，默认按钮的背景就是用了类似的方法实现的。我们看一下google官方的描述：



该格式相对于一般PNG图片来说，多了上下左右各一条1px的黑线。左、上黑线隔开了9个格子，当中一个格子(见上图Strechable Area区域)声明为可以进行拉伸。右、下两条黑线所定义的Paddingbox区域是在该图片当做背景时，能够在图片上填写文字的区域。每条黑线都是可以不连续的，这样就可以定义出很多自动拉伸的规格。Android sdk中提供了设置的工具，启动命令位于：$ANDROID\_SDK/tools/draw9patch.bat，使用它对于原始PNG进行设置9-patch格式，非常方便，如下图。



draw9patch工具的右侧是能够看到各方向拉伸后的效果图，你所要做的就是在图上最外侧一圈1px宽的像素上涂黑线。

注意，在draw9patch.bat第一次运行时，sdk2.2版本上会报错：java.lang.NoClassDefFoundError:org/jdesktop/swingworker/SwingWorker。需要下载[swing-worker-1.1.jar](http://code.google.com/p/dividere/downloads/detail?name=swing-worker-1.1.jar&can=2&q=) ，放入$android\_sdk/tools/lib路径下，成功运行。

将鼠标放在白色边框，点住左键不放拖动，会出现黑色线条如图。

解释：

1号黑色条位置向下覆盖的区域表示图片横向拉伸时，只拉伸该区域

2号黑色条位置向右覆盖的区域表示图片纵向拉伸时，只拉伸该区域

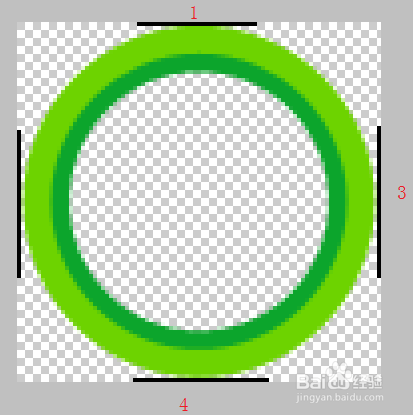
3号黑色条位置向左覆盖的区域表示图片纵向显示内容的区域

4号黑色条位置向上覆盖的区域表示图片横向显示内容的区域

注： 没有黑色条的位置覆盖的区域是图片拉伸时保持不变

5、如果失误多选了部分，可按住shift键，点击鼠标左键去掉黑色条

6、右边3个黑色图形区域为显示当前拉伸效果

[](http://jingyan.baidu.com/album/3065b3b683eab3becff8a409.html?picindex=4)

此种方法实现的步骤如下。

(a) 使用draw9patch.bat作完图片后，得到两张按钮背景，分别是正常和按下状态下的，命名为bg\_btn.9.png和bg\_btn\_2.9.png。

(b) 编写图片使用描述文件bg\_9patchbutton.xml。

1. // in bg\_9patchbutton.xml
2. <?xml version="1.0" encoding="UTF-8"?>
3. <selector xmlns:android="http://schemas.android.com/apk/res/android">
4. <item android:state\_pressed="true"
5. android:drawable="@drawable/bg\_btn\_2" />
6. <item android:state\_focused="true"
7. android:drawable="@drawable/bg\_btn\_2" />
8. <item android:drawable="@drawable/bg\_btn" />
9. </selector>

(c) 在界面定义文件 layout/main.xml中添加Button、ImageButton按钮控件的定义。Button、ImageButton都是可以使用背景属性的。

1. <Button
2. android:layout\_width="120dip"
3. android:layout\_height="40dip"
4. android:text="9-patch图片背景按钮"
5. android:background="@drawable/bg\_9patchbutton" />
6. <Button
7. android:layout\_width="200dip"
8. android:layout\_height="40dip"
9. android:text="9-patch图片背景按钮"
10. android:background="@drawable/bg\_9patchbutton" />
11. <Button
12. android:layout\_width="120dip"
13. android:layout\_height="80dip"
14. android:text="9-patch图片背景按钮"
15. android:background="@drawable/bg\_9patchbutton" />
16. <ImageButton
17. android:layout\_width="120dip"
18. android:layout\_height="40dip"
19. android:src="@drawable/bg\_9patchbutton"
20. android:scaleType="fitXY"
21. android:background="@android:color/transparent" />

以上2种实现按钮的美化，都是标准的矩形按钮为基础。在一些应用中我们可以看到漂亮的自定义形状的异形按钮，这是怎么实现的呢？经过一番研究和实践，找出了一种方便的方法，就是使用ImageButton加上9-patch就可以实现漂亮的自动延伸效果。

## 3. 自定义形状、颜色、图样的按钮的实现

步骤如下。

(a) 设计一张自定义形状风格背景的图片，如下图。



(b) 未点击和按下后的状态各做一张，形成一套图片，如下图。

https://bbsmax.ikafan.com/static/L3Byb3h5L2h0dHAvaW1hZ2VzMC5jbmJsb2dzLmNvbS9ibG9nMjAxNS83MzQ0NjcvMjAxNTA3LzE3MTcxNjI1NTYzNjc1OS5naWY=.jpgforward.png

https://bbsmax.ikafan.com/static/L3Byb3h5L2h0dHAvaW1hZ2VzMC5jbmJsb2dzLmNvbS9ibG9nMjAxNS83MzQ0NjcvMjAxNTA3LzE3MTcxNjQ4NDU0NTMwMC5naWY=.jpgforward2.png

(c) 创建和编写按钮不同状态的图片使用描述文件drawable/ib\_forward.xml

1. // ib\_forward.xml
2. <?xml version="1.0" encoding="UTF-8"?>
3. <selector xmlns:android="http://schemas.android.com/apk/res/android">
4. <item android:state\_pressed="true"
5. android:drawable="@drawable/forward2" />
6. <item android:state\_focused="true"
7. android:drawable="@drawable/forward2" />
8. <item android:drawable="@drawable/forward" />
9. </selector>

(d) 在界面定义文件 layout/main.xml中添加ImageButton按钮控件的定义。

1. // in layout/main.xml
2. <ImageButton
3. android:layout\_width="80dip"
4. android:layout\_height="40dip"
5. android:src="@drawable/ib\_forword"
6. android:scaleType="fitXY"
7. android:background="@android:color/transparent" />

版权归http://blog.csdn.net/xjanker2/archive/2011/03/04/6222690.aspx所有