

个人资料



z1765887945

访问: 635次

积分: 27

等级: BLOG > 1

排名: 千里之外

原创: 2篇 转载: 0篇

译文: 0篇 评论: 0条

文章搜索

文章分类

android (2)

文章存档

2016年06月 (1)

2016年05月 (1)

阅读排行

android vector矢量图画图 (590)

Android ipc通信机制 (26)

评论排行

android vector矢量图画图 (0)

Android ipc通信机制 (0)

推荐文章

* 2016 年最受欢迎的编程语言是什么?

* Chromium扩展 (Extension) 的页面 (Page) 加载过程分析

* Android Studio 2.2 来啦

深度学习代码专栏 攒课--我的学习我做主 开启你的知识管理, 知识库个人图谱上线

android vector矢量图画图详解

标签: svg 矢量图 android vector pathdata

2016-05-08 15:09 598人阅读 评论(0) 收藏 举报

分类:

android (1)

版权声明: 本文为博主原创文章, 未经博主允许不得转载。

SVG是一种矢量图格式, 是Scalable Vector Graphics三个单词的首字母缩写。在xml文件中的标签是<vector>, 画出的图形可以像一般的图片资源使用, 例子如下:



```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportHeight="24.0"
    android:viewportWidth="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M22,16V4c0,-1.1 -0.9,-2 -2,-2H8c-1.1,0 -2,0.9 -2,2v12c0,1.1 0.9,2
2,2h12c1.1,0 2,-0.9 2,-2zm-11,-4l2.03,2.71L16,11l4,5H8l3,-4zM2,6v14c0,1.1 0.9,2 2,2h14v-2H4V6H2z"/>
</vector>
```



其所画出的图形为

于此同时, Android studio提供了丰富的图片资源, 可以右键module,new->vector asset选择, 如下:



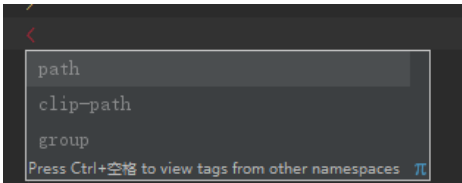
是不是很羡慕这些酷酷的图形, 当然自己也可以去动手去做。显而易见, 上面例子的重点是在PathData里面的那一大串数字:

```
android:pathData="M22,16V4c0,-1.1 -0.9,-2 -2,-2H8c-1.1,0 -2,0.9 -2,2v12c0,1.1 0.9,2,2h12c1.1,0 2,-0.9
2,-2zm-11,-4l2.03,2.71L16,11l4,5H8l3,-4zM2,6v14c0,1.1 0.9,2 2,2h14v-2H4V6H2z"/>
```

xml文件中:

* 手把手教你做音乐播放器（二）
技术原理与框架设计

* JVM 性能调优实战之：使用阿里开源工具 TProfiler 在海量业务代码中精确定位性能代码



共有三个标签，下面就是<path>部分：

先慢慢学习一些基本的语法：

- M: move to 移动绘制点，作用相当于把画笔落在哪一点。
- L: line to 直线，就是一条直线，注意，只是直线，直线是没有宽度的，所以你什么也看不到。

android:strokeColor="#333333" android:strokeWidth="10" 设置颜色和线宽

- Z: close 闭合，嗯，就是把图封闭起来。
- C: cubic bezier 三次贝塞尔曲线
- Q: quadratic bezier 二次贝塞尔曲线
- A: ellipse 圆弧

每个命令都有大小写形式，大写代表后面的参数是绝对坐标，小写表示相对坐标，相对于上一个点的位置。参数之间用空格或逗号隔开。

命令详解：

- M (x y) 把画笔移动到x,y，要准备在这个地方画图了。
- L (x y) 直线连到x,y，还有简化命令H(x) 水平连接、V(y)垂直连接。
- Z，没有参数，连接起点和终点
- C(x1 y1 x2 y2 x y)，控制点 (x1,y1) (x2,y2)，终点x,y。
- Q(x1 y1 x y)，控制点 (x1,y1)，终点x,y
- C和Q会在下文做简单对比。
- A(rx ry x-axis-rotation large-arc-flag sweep-flag x y)
- android:pathData=" M50,50 a10,10 1,1 0 1,0" />
rx ry 椭圆半径
x-axis-rotation x轴旋转角度
large-arc-flag 为0时表示取小弧度，1时取大弧度 （舍取的时候，是要长的还是短的）
sweep-flag 0取逆时针方向，1取顺时针方向

L的用法：

android:pathData=" M10,0 L10,40 40,40" />

把画笔放在（10,0）位置，连线10，40点 在连线40，40点。。。于是，一个直角三角形出来了~这里没有写z，没什么关系。



Q和C的对比： 详细了解贝塞尔曲线：

<http://www.cnblogs.com/jay-dong/archive/2012/09/26/2704188.html>

Q

android:pathData="M0,0 q30,90 80,20"/>



控制点1， 30,90：

控制点2， 80,20：

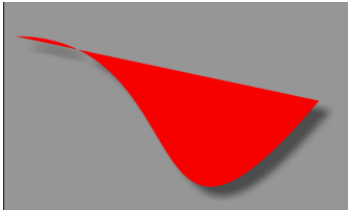
C

android:pathData=" M0,0 c0,0 30,90 80,20" />
C 第一控制点 (0, 0) 第二控制点 (30, 90) 结束点 (80, 20) 或 c 第一控制点 第二控制点 结束点



现在修改第一个控制点:

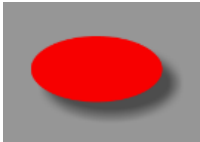
android:pathData=" M0,0 c50,0 30,90 80,20" />



a:
这么多 数字, 怎么看啊, 可以直接拉到下面看作用。

android:pathData=" M50,50 a10,5 0,1 0 1,0" />
以50, 50为起点, 逆时针画

椭圆图形, x轴半径10, y轴半径5



转动x轴~~~

android:pathData=" M50,50 a10,5 90,1 0 1,0" />



我想要椭圆上半段, 此处修改为x轴半径的两倍
android:pathData=" M50,50 a10,5 90,1 0 20,0" />

椭圆左半段
android:pathData=" M50,50 a10,5 90 1 0 0 10" />

椭圆右半段
android:pathData=" M50,50 a10,5 90 1 1 0 10" />

th
android:fillColor="#fff70000" 下
android:pathData=" M50,50 a10,5 0 1 0 1 0" />
<path
android:fillColor="#FFF22420" 上
android:pathData=" M50,50 a10,5 0 1 1 1 0" />
<path
android:fillColor="#fff57000"右
android:pathData=" M50,50 a10,5 0 1 1 1 1" />
<path
android:fillColor="#FF323243"左
android:pathData=" M50,50 a10,5 0 1 0 0 1" />

将为您关闭此次推广展示 [撤销](#)

此推广有什么问题？

视觉干扰

内容不宜

不感兴趣

其它

Baidu 百度



出现上面的情况可以想到是因为，起始点50,50在椭圆中的位置不同。那么，再修改一下。

```
android:pathData=" M50,50 a10,5 0 1 1 0 7" /> 
```

修改了右边椭圆的代码



现在取的是大弧度，所以看到这样的效果，如果 7改为10（也就是y轴半径的两倍）这刚好会在 一半的位置。

现在取小弧度看看，

```
android:pathData=" M50,50 a10,5 0 0 1 0 7" /> 
```

，可以看到小弧度 顺时针画图。



再修改为逆时针，

```
android:pathData=" M50,50 a10,5 0 0 0 0 7" /> 
```



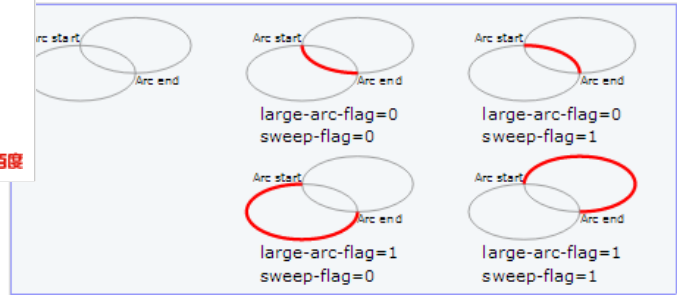
椭圆的属性 差不多讲解完成了，如下

```
android:pathData=" M50,50 a10,5 0 0 0 0 7"/>
```

- 10, 5 为椭圆x, y轴半径
- 第一个0 为 x轴旋转角度
- 第二个0 为取大小弧度，0为小，1为大
- 第三个0 为顺逆时针，0为逆1为顺

- 第四个0 为修改修改起始点在椭圆中的位置，y轴.
- 四个 7 为修改修改起始点在椭圆中的位置，x轴。

是前辈留下的图：



<path>里面还有哪些属性那？

将为您关闭此次推广展示

[撤销](#)

此推广有什么问题？

视觉干扰

内容不宜

不感兴趣

其它

Baidu 百度

```
android:fillAlpha
android:name
android:pathData
android:fillColor
android:strokeAlpha
android:strokeColor
android:strokeLineCap
android:strokeLineJoin
android:strokeMiterLimit
android:strokeWidth
android:trimPathEnd
android:trimPathOffset
android:trimPathStart
Press Ctrl+空格 to view tags from other namespaces
```

- **android:name** 定义该 **path** 的名字，这样在其他地方可以通过名字来引用这个路径
- **android:pathData** 和 **SVG** 中 **d** 元素一样的路径信息。
- **android:fillColor** 定义填充路径的颜色，如果没有定义则不填充路径
- **android:strokeColor** 定义如何绘制路径边框，如果没有定义则不显示边框
- **android:strokeWidth** 定义路径边框的粗细尺寸
- **android:strokeAlpha** 定义路径边框的透明度
- **android:fillAlpha** 定义填充路径颜色的透明度
- **android:trimPathStart** 从路径起始位置截断路径的比率，取值范围从 **0** 到**1**
- **android:trimPathEnd** 从路径结束位置截断路径的比率，取值范围从 **0** 到**1**
- **android:trimPathOffset** 设置路径截取的范围 **Shift trim region (allows showed region to include the start and end), in the range from 0 to 1.**
- **android:strokeLineCap** 设置路径线帽的形状，取值为 **butt, round, square.**
- **android:strokeLineJoin** 设置路径交界处的连接方式，取值为 **miter,round,bevel.**
- **android:strokeMiterLimit** 设置斜角的上限， **Sets the Miter limit for a stroked path.**

下面详细讲一下 **android:strokeLineCap** ， **android:strokeLineJoin** 两个属性

```
android:pathData="M200,200 1100,300
                M300,200 1-100,300
```

再没有添加这两条属性前：



将为您关闭此次推广展
示 [撤销](#)

此推广有什么问题？

视觉干扰

内容不宜

不感兴趣

其它

Baidu 百度

添加语句：**android:strokeLineCap="round"**后可以看到有三个点改变了格式（左下角是图形结束点，并没有改变）



最后添加: android:strokeLineJoin="round" 左下角也做了改变, 如下



这xml开始部分的代码是做什么的那?

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportHeight="24.0"
    android:viewportWidth="24.0">
```

先看看有哪些属性,

```
vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:height="64dp"
    android:width="64dp"
    android:viewportHeight="600"
    android:viewportWidth="600"
    android:name="..."
    android:alpha="..."
    android:autoMirrored="..."
    android:tint="..."
    android:tintMode="..."
    app:srcCompat="@drawable/..."
    app:srcCompat="@drawable/..."
```

将为您关闭此次推广展示 [撤销](#)

此推广有什么问题?

视觉干扰

内容不宜

不感兴趣

其它

Baidu 百度

- android:name 定义该drawable的名字
- android:width 定义该 drawable 的内部(intrinsic)宽度,支持所有 Android 系统支持的尺寸, 通常使用 dp
- android:height 定义该 drawable 的内部(intrinsic)高度,支持所有 Android 系统支持的尺寸, 通常使用 dp
- android:viewportWidth 定义矢量图视图的宽度, 视图就是矢量图 path 路径数据所绘制的虚拟画布
- android:viewportHeight 定义矢量图视图的高度, 视图就是矢量图 path 路径数据所绘制的虚拟画布
- android:tint 定义该 drawable 的 tint 颜色。默认是没有 tint 颜色的
- android:tintMode 定义 tint 颜色的 Porter-Duff blending 模式, 默认值为 src_in
- android:autoMirrored 设置当系统为 RTL (right-to-left) 布局的时候, 是否自动镜像以图片。比如 阿拉伯语。
- android:alpha 该图片的透明度属性

<group>里面可以定义多了<path>，这样可以方便管理多个<path>

```
android:translateX
android:name
android:pivotX
android:pivotY
android:rotation
android:scaleX
android:scaleY
android:translateY
appNs      adds application namespace
Press Ctrl+空格 to view tags from other namespaces
```

- **android:name** 定义 **group** 的名字
- **android:rotation** 定义该 **group** 的路径旋转多少度，这样图片就被旋转了，注意写数字的时候别晕了。
- **android:pivotX** 定义缩放和旋转该 **group** 时候的 **X** 参考点。该值相对于 **vector** 的 **viewport** 值来指定的。
- **android:pivotY** 定义缩放和旋转该 **group** 时候的 **Y** 参考点。该值相对于 **vector** 的 **viewport** 值来指定的。
- **android:scaleX** 定义 **X** 轴的缩放倍数
- **android:scaleY** 定义 **Y** 轴的缩放倍数
- **android:translateX** 定义移动 **X** 轴的位移。相对于 **vector** 的 **viewport** 值来指定的。
- **android:translateY** 定义移动 **Y** 轴的位移。相对于 **vector** 的 **viewport** 值来指定的。

<clip-path>定义当前绘制的剪切路径，就是图像的一部分剪切下来。注意，clip-path 只对当前的 group 和子 group 有效。

```
android:name
android:pathData
appNs      adds application namespace
Press Ctrl+空格 to view tags from other namespaces
```

<clip-path android:pathData="M200,200 h200 v150 h-200 v-150" />

原图为上面的 叉 ， 剪切后为：



将为您关闭此次推广展示 [撤销](#)

此推广有什么问题？

视觉干扰

内容不宜

不感兴趣

其它

Baidu 百度

顶 0 踩 0

下一篇 Android ipc通信机制

我的同类文章

android（1）		
• Android ipc通信机制	2016-06-09	阅读 26

参考知识库



Android知识库
12674 关注 | 1190 收录

猜你在找

- 精通iOS移动开发 (Xcode7&Swift2;)：视图与视图控制
- 如何使用Android的VectorDrawable类绘制矢量图
- NinePatch图片制作从入门到精通
- Android 关于SVG矢量图支持
- iOS移动开发从入门到精通 (Xcode7 & Swift2)
- 通过android studio 支持各版本使用矢量图 SVG
- C/C++高级开发实战 快速掌握C/C++ 适合iOS/Android
- Android矢量图VectorDrawable及动画
- 在VC2015里学会使用tinymce库
- Android使用SVG矢量图打造酷炫动效



查看评论

暂无评论

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

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved

将为您关闭此次推广展示 [撤销](#)

此推广有什么问题？

视觉干扰

内容不宜

不感兴趣

其它

Baidu 百度