

eclipse\_xu

路漫漫其修远兮 吾将上下而求索

目录视图

我的新书

Android群英传:神兵利器，包含了Android开发中的工具使用技巧



查看详情

网上购买：

京东 天猫 亚马逊 当当

友情链接 - 我们的博客

聊科技 游戏 电影 美食 请访问 HZtalk  
Android开发者 极客头条  
极客导航

我写的书

Android群英传，包含了Android开发技巧与提高



查看详情

网上购买：

京东 天猫 亚马逊 当当

本书源码：Github下载

我的Github

欢迎Follow、Fork、Star

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曲折的兼容之路

标签: android Vector svg 动画

2016-07-06 13:29 9513人阅读 评论1

分类： Android ( 186 )

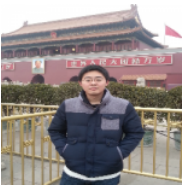
版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?) [+]

Android Vector曲折的兼容之路

两年前写书的时候，就在研究Android L提出的Vector，可研究下来发现，完全不具备兼容性，相信这也是它没有被广泛使用的不懈努力，现在Vector终于迎来了它的春天。

个人资料



eclipse\_xu

关注

发私信

访问：987358次

积分：14720

等级：BLOG > 7

排名：第527名

原创：236篇

转载：18篇

译文：7篇

评论：2382条

我的视频

Android Studio第一次亲密接触

Android UI模板设计

Android Bmob后端服务

Android 属性动画分析

Android 图像处理

Android 图像处理番外篇

Android BaseAdapter

Android AsyncTask

Android 异步加载


Android 摄像头


Android 新年心愿

Android 二维码

我的微信公众号


为你推荐最新的博文~更有惊喜等着你



 给我写信

文章搜索


博客专栏



Android异步任务处理从零开始

文章：8篇

阅读：22731



android拼图游戏开发记录

文章：6篇

阅读：57131



在文章后面，会给出本文的Demo和效果图，并开源在Github

Vector Drawable

Android 5.0发布的时候，Google提供了Vector的支持。Vector Drawable相对于普通的Drawable来说，有以下几个好处：

• Vector图像可以自动进行适配，不需要通过分辨率来设置不同的图片

• Vector图像可以大幅减少图像的体积，同样一张图，用Vector来实现，可能只有PNG的几十分之一

• 使用简单，很多设计工具，都可以直接导出SVG图像，从而转换成Vector图像

• 功能强大，不用写很多代码就可以实现非常复杂的动画

• 成熟、稳定，前端已经非常广泛的进行使用了

Vector图像刚发布的时候，是只支持Android 5.0+的，对于Android pre-L的系统来说，并不能使用，所以，可以说那时候的V  
过自从AppCompat 23.2之后，Google对p-View的Android系统也进行了兼容，也就是说，Vector可以使用于Android 2.1以  
com.android.support:appcompat-v7:23.2.0以上的版本就可以了，这时候，Vector应该算是迎来了它的春天。

如何获得Vector图像

概念

首先，需要讲解两个概念——SVG和Vector。

SVG，即Scalable Vector Graphics 矢量图，这种图像格式在前端中已经使用的非常广泛了，详见  
WIKI：[https://en.wikipedia.org/wiki/Scalable\\_Vector\\_Graphics](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics)

Vector，在Android中指的是Vector Drawable，也就是Android中的矢量图，详  
见：<https://developer.android.com/reference/android/graphics/drawable/VectorDrawable.html>

http://blog.csdn.net/eclipsxys/article/details/51838119

2/15



Android UI开发详解

文章：8篇

阅读：43473

我的微博

微博



Tomcat的猫

加关注

为啥没我的书 [泪] //@博文视点Broadview: 小伙伴们，快来继续吧

慕课网：#慕课网分享#【九月图书慕课网第二波送书！】慕课网强势推出实战课程，还原一线互联网公司工作流程。工具+内容，课程实用性极强，帮你突破职场瓶颈：<http://t.cn/Rq5Zfky> 慕课网联合@博文视点Broadview 九月

TA 的粉丝 (4244)

全部»



0fxHz



零度西瓜



柴姐wm



XXXXXTf



Le丶Incr



博文视点

文章分类

Flex (1)

编程基础 (31)

Android (187)

VBA (10)

数据结构 (7)

数据库 (12)

其他 (13)

C# (7)

文章存档

2016年09月 (2)

2016年08月 (2)

2016年07月 (3)

2016年06月 (1)

2016年05月 (1)

展开

阅读排行

Android Design Support Li... (47404)

Android UI开发详解之Action... (40293)

Eclipse，到了说再见的时候了... (30110)

Android Studio导入Project... (29175)

对不起！我来晚了！——《A... (26571)

我也来开发2048之终极奥义 (25632)

解放双手——Android自动化... (25053)

Gradle自定义插件 (23964)

向产品宣战——开发者眼中的... (23173)

Android拼图游戏开发全纪录1 (17795)

评论排行

因此，可以说Vector就是Android中的SVG实现，因为Android中的Vector并不是支持全部的SVG语法，也没有必要，因为完整的，但已经支持的SVG语法已经够用了，特别是Path语法，几乎是Android中Vector的标配，详细可以参考：<http://www.w3.c>

Vector语法简介

Android以一种简化的方式对SVG进行了兼容，这种方式就是通过使用它的Path标签，通过Path标签，几乎可以实现SVG中的其复杂一点，但这些东西都是可以通过工具来完成的，所以，不用担心写起来会很复杂。

Path指令解析如下所示：

1. 支持的指令：
- M = moveto(M X,Y)：将画笔移动到指定的坐标位置
  - L = lineto(L X,Y)：画直线到指定的坐标位置
  - H = horizontal lineto(H X)：画水平线到指定的X坐标位置
  - V = vertical lineto(V Y)：画垂直线到指定的Y坐标位置
  - C = curveto(C X1,Y1,X2,Y2,ENDX,ENDY)：三次贝赛曲线
  - S = smooth curveto(S X2,Y2,ENDX,ENDY)
  - Q = quadratic Belzier curve(Q X,Y,ENDX,ENDY)：二次贝赛曲线
  - T = smooth quadratic Belzier curveto(T ENDX,ENDY)：映射
  - A = elliptical Arc(A RX,RY,XROTATION,FLAG1,FLAG2,X,Y)：弧线
  - Z = closepath()：关闭路径

2. 使用原则:
- 坐标轴为以(0,0)为中心，X轴水平向右，Y轴水平向下
  - 所有指令大小写均可。大写绝对定位，参照全局坐标系；小写相对定位，参照父容器坐标系
  - 指令和数据间的空格可以省略
  - 同一指令出现多次可以只用一个

注意，‘M’处理时，只是移动了画笔，没有画任何东西。它也可以在后面给出上同时绘制不连续线。

关于这些语法，开发者需要的并不是全部精通，而是能够看懂即可，其它的都可以交给工具来实现。

从PNG到SVG

- 设计师

要从一般使用的PNG图像转换到SVG图像，对于设计师来说，并不是一件难事，因为大部分的设计工具（PS、Illustrator等等）像，如PNG、JPG，当然，也包括SVG，因此，设计师可以完全按照原有的方式进行设计，只是最后导出的时候，选择SVG即可

- 程序员

不要求开发者都去学习使用这些设计工具，开发者可以利用一些工具，自己转换一些比较基础的图像，<http://inloop.github.io>，一个非常牛逼的网站，可以在线将普通图像转换为Android Vector Drawable。如图所示：

我也来开发2048之终极奥义	(377)
Android拼图游戏开发全纪录5	(213)
向产品宣战——开发者眼中的...	(174)
对不起！我来晚了！——《A...	(156)
把抽奖活动写成一篇技术博客...	(148)
《Android群英传》勘误	(115)
Android拼图游戏开发全纪录3	(114)
Eclipse，到了说再见的时候了...	(102)
Android Design Support Li...	(70)
Android群英传:神兵利器——...	(69)

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

\* 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

最新评论

Android Studio集成Bug管理系统

包子超人：@u012764110:同问

Android拼图游戏开发全纪录5

qq\_33625205：771435689@qq.com 求源码LZ，谢谢

自律给你自由——设计布局的新姿势

天风隼：凑合

自律给你自由——设计布局的新姿势

ysy950803：@x359981514:那也还好，现在应该也基本没有4.4以下的机子了。

自律给你自由——设计布局的新姿势

eclipse\_xu：@ysy950803:最低适配API9 几乎没有问题~

自律给你自由——设计布局的新姿势

ysy950803：这个应该有API版本要求吧，太低了用不了。

自律给你自由——设计布局的新姿势

sumsear：@x359981514:好有道理

自律给你自由——设计布局的新姿势

gongpan\_1986：看上去挺好的，用代码写布局几年了，这个目前非常不习惯

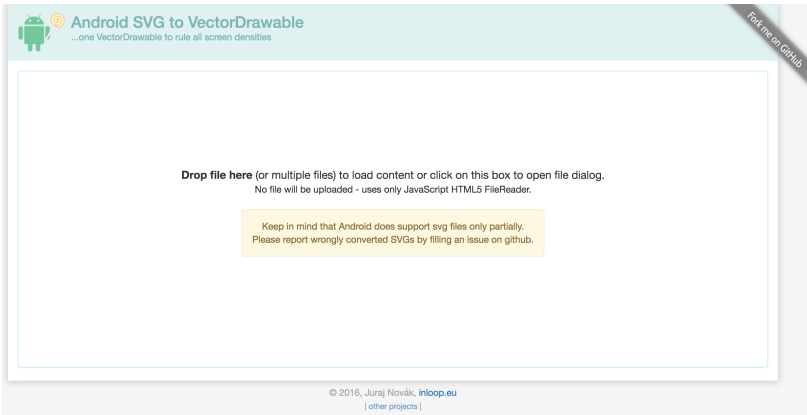
自律给你自由——设计布局的新姿势

乱在长安：@x359981514:+1

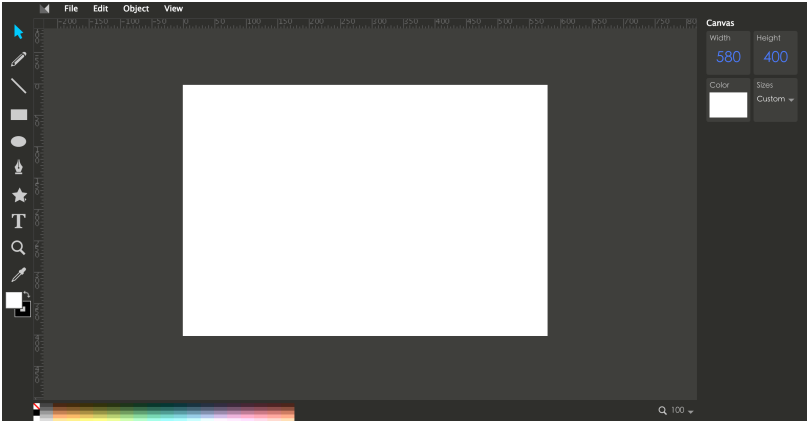
自律给你自由——设计布局的新姿势

eclipse\_xu：@zhiwenyan:小磁铁

公众号

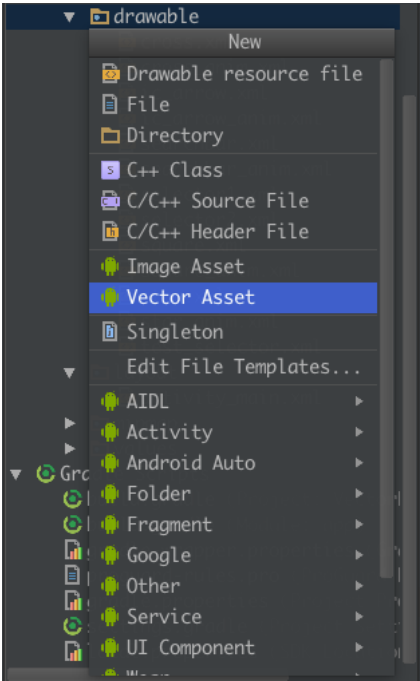


或者，还可以使用SVG的编辑器来进行SVG图像的编写，例如<http://editor.method.ac/>

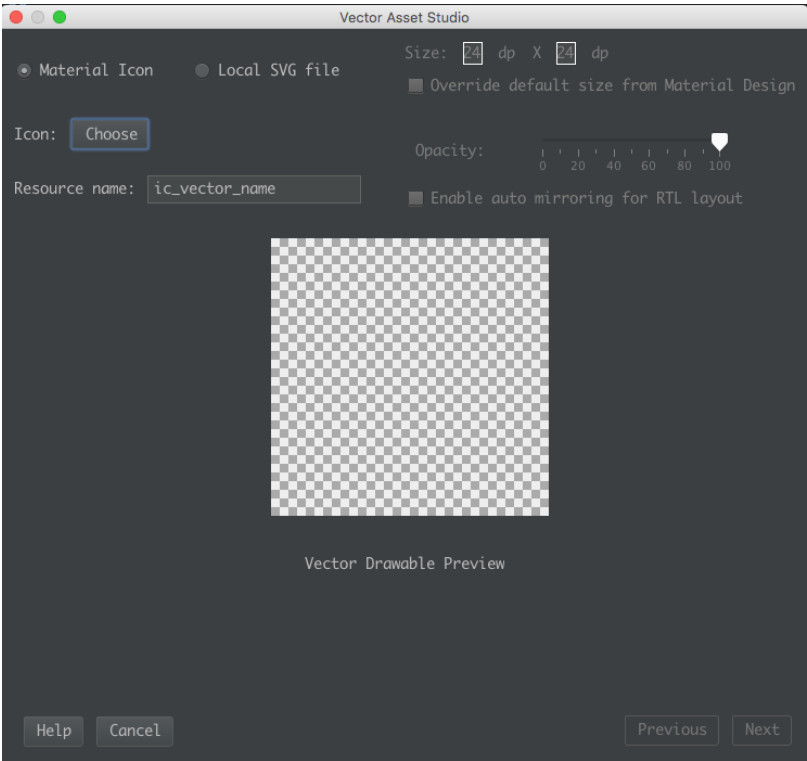


使用Android Studio

利用Android Studio的Vector Asset，可以非常方便的创建Vector图像，甚至可以直接通过本地的SVG图像来生成Vector图像



进去之后，就可以生成Vector图像，如图所示：



## Google的兼容之路

### 只兼容L+

Vector是在Android L中提出来的新概念，所以在刚开始的时候是只兼容L+的。

### Gradle Plugin 1.5的兼容

从Gradle Plugin 1.5开始，Google支持了一种兼容方式，即在Android L之上，使用Vector，而在L之下，则使用Gradle将Vec

Android gradle plugin 1.5发布以后，加入了一个跟VectorDrawable有关的新功能。Android build tools 提供了另外一如果编译的版本是5.0之前的版本，那么build tools 会把VectorDrawable生成对应的png图片，这样在5.0以下的版本则使而在5.0以上的版本中则使用VectorDrawable.在build.gradle添加generatedDensities配置，可以配置生成的png图片的密

### AppCompat23.2的兼容

从AppCompat23.2开始，Google开始支持在低版本上使用Vector。

## 静态Vector图像

我们有很多方法能够得到这些Vector，那么如何使用它们呢，Android 5.0以上的使用就不讲了，不太具有普遍代表性，我们从起。

### pre-L版本兼容

VectorDrawableCompat依赖于AAPT的一些功能，它能保持最近矢量图使用的添加的属性ID，以便他们可以被pre-L版本之前

在Android 5.0之前使用Vector，需要aapt来对资源进行一些处理，这一过程可以在aapt的配置中进行设置，如果没有启用这样的设备上运行就会发生android.content.res.Resources\$NotFoundException。

首先，你需要在项目的build.gradle脚本中，增加对Vector兼容性的支持，代码如下所示：

使用Gradle Plugin 2.0以上：

```
1  android {
2
3      defaultConfig {
4          vectorDrawables.useSupportLibrary = true
5      }
6  }
```

```
6    }  
    }
```

使用Gradle Plugin 2.0以下，Gradle Plugin 1.5以上：

```
1  android {  
2      defaultConfig {  
3          // Stops the Gradle plugin's automatic rasterization of vectors  
4          generatedDensities = []  
5      }  
6      // Flag to tell aapt to keep the attribute ids around  
7      aaptOptions {  
8          additionalParameters "--no-version-vectors"  
9      }  
10 }
```

像前面提到的，这种兼容方式实际上是先关闭AAPT对pre-L版本使用Vector的妥协，即在L版本以上，使用Vector，而在pre-L版本的PNG图片，generatedDensities这个数组，实际上就是要生成PNG的图片分辨率的数组，使用appcompat后就不需要这样

当然，最重要的还是添加appcompat的支持：

```
1  compile 'com.android.support:appcompat-v7:23.4.0'
```

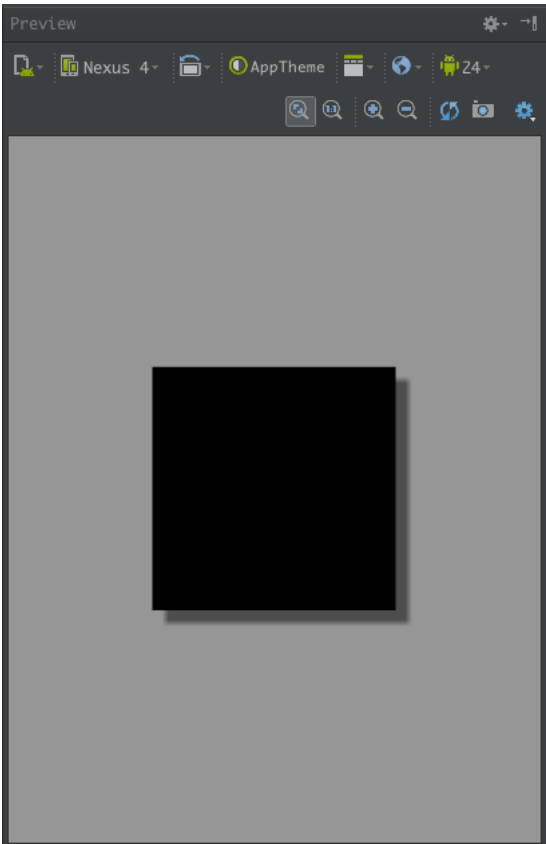
同时，确保你使用的是AppCompatActivity而不是普通的Activity。

## Vector图像

一个基本的Vector图像，实际上也是一个xml文件，如下所示：

```
1  <vector xmlns:android="http://schemas.android.com/apk/res/android"  
2      android:width="200dp"  
3      android:height="200dp"  
4      android:viewportHeight="500"  
5      android:viewportWidth="500">  
6  
7      <path  
8          android:name="square"  
9          android:fillColor="#000000"  
10         android:pathData="M100,100 L400,100 L400,400 L100,400 z"/>  
11  
12 </vector>
```

显示如图所示：



这里需要解释下这里的几个标签：

- android:width \ android:height:定义图片的宽高
- android:viewportHeight \ android:viewportWidth:定义图像被划分的比例大小，例如例子中的500，即把200dp大小Path标签中的坐标，就全部使用的是这里划分后的坐标系统。

这样做有一个非常好的作用，就是将图像大小与图像分离，后面可以随意修改图像大小，而不需要修改PathData中的坐标。

- android:fillColor:PathData中的这些属性就不详细讲了，与Canvas绘图的属性基本类似。

在控件中使用

有了静态的Vector图像，就可以在控件中使用了。

可以发现，这里我们使用的都是普通的ImageView，好像并不是AppCompatActivity，这是因为使用了AppCompatActivity后，ImageView转换为AppCompatActivity。

ImageView\ImageButton

对于ImageView这样的控件，要兼容Vector图像，只需要将之前的android:src属性，换成app:srcCompat即可，示例代码如下

```
1 <ImageView
2     android:id="@+id/iv"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     app:srcCompat="@drawable/vector_image"/>
```

在代码中设置的话，代码如下所示：

```
1 ImageView iv = (ImageView) findViewById(R.id.iv);
2 iv.setImageResource(R.drawable.vector_image);
```

setBackgroundResource也是可以设置Vector的API

Button

Button并不能直接使用app:srcCompat来使用Vector图像，需要通过Selector来进行使用，首先，创建两个图像，用于Selecto示：

selector1.xml

```
1 <vector xmlns:android="http://schemas.android.com/apk/res/android"
2     android:width="24dp"
3     android:height="24dp"
4     android:viewportHeight="24.0"
5     android:viewportWidth="24.0">
6     <path
7         android:fillColor="#FF000000"
8         android:pathData="M14.59,8L12,10.59 9.41,8 8,9.41 10.59,12 8,14.59 9.41,16 12,13.41 14.59,16 16
9 </vector>
```

selector2.xml

```
1 <vector xmlns:android="http://schemas.android.com/apk/res/android"
2     android:width="24dp"
3     android:height="24dp"
4     android:viewportHeight="24.0"
5     android:viewportWidth="24.0">
6     <path
7         android:fillColor="#FF000000"
8         android:pathData="M11,15h2v2h-2zM11,7h2v6h-2zM11.99,2C6.47,2 2,6.48 2,12s4.47,10 9.99,10C17.52,
9 </vector>
```

selector.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <selector xmlns:android="http://schemas.android.com/apk/res/android">
3     <item android:drawable="@drawable/selector1" android:state_pressed="true"/>
4     <item android:drawable="@drawable/selector2"/>
5 </selector>
```

非常简单，只是把普通的Selector中的图像换成了Vector图像而已，接下来，在Button中使用这个Selector即可：

```
1 <Button
2     android:id="@+id/btn"
3     android:layout_width="70dp"
4     android:layout_height="70dp"
5     android:background="@drawable/selector"/>
```

然后运行，如果你认为可以运行，那就是太天真了，都说了是兼容，怎么能没有坑呢，这里就是一个坑.....

这个坑实际上是有历史渊源的，Google的一位开发者在博客中写到：

First up, this functionality was originally released in 23.2.0, but then we found some memory usage and Configur so we it removed in 23.3.0. In 23.4.0 (technically a fix release) we’ ve re-added the same functionality but behind need to manually enable.

实际上，他们的这个改动，就影响了类似DrawableContainers ( DrawableContainers which reference other drawables res only a vector resource ) 这样的类，它的一个典型，就是Selector ( StateListDrawable也是 )。这个开发者在文中提到的flag 放在Activity的前面就可以了：

```
1 static {
2     AppCompatDelegate.setCompatVectorFromResourcesEnabled(true);
3 }
```

开启这个flag后，你就可以正常使用Selector这样的DrawableContainers了。同时，你还开启了类似android:drawableLeft这的使用权限，以及RadioButton的使用权限，以及ImageView’ s src属性。

RadioButton

RadioButton的Button同样可以定义，代码如下所示：



```
1 <RadioButton
2     android:layout_width="50dp"
3     android:layout_height="50dp"
4     android:button="@drawable/selector"/>
```

动态Vector基础

动态Vector才是Android Vector Drawable的精髓所在

动态的Vector需要通过animated-vector标签来进行实现，它就像一个粘合剂，将控件与Vector图像粘合在了一起，一个基础的下所示：

```
1 <animated-vector
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:drawable="@drawable/XXXXX1">
4
5     <target
6         android:name="left"
7         android:animation="@animator/XXXXX2"/>
8
9 </animated-vector>
```

实际上这里面只有两个重点是需要关注的，XXXXX1和XXXXX2。一个具体的示例如下所示：

```
1 <animated-vector
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:drawable="@drawable/ic_arrow">
4
5     <target
6         android:name="left"
7         android:animation="@animator/anim_left"/>
8
9     <target
10        android:name="right"
11        android:animation="@animator/anim_right"/>
12
13 </animated-vector>
```

这里表示目标图像是drawable/ic\_arrow，对left、right分别使用了anim\_left、anim\_right动画。这里的name属性，就是在静态者path标签的name属性。

animated-vector标签在现在的Android Studio中实际上是会报错的，但这个并不影响编译和运行，属于Android Studio的兼容性问题。

目标图像

XXXXX1是目标Vector图像，也就是静态的Vector图像，例如：

```
1 <vector xmlns:android="http://schemas.android.com/apk/res/android"
2     android:width="120dp"
3     android:height="120dp"
4     android:viewportHeight="24.0"
5     android:viewportWidth="24.0">
6
7     <group android:name="left">
8         <path
9             android:fillColor="#FF000000"
10            android:pathData="M9.01,14L2,14v2h7.01v3L13,15l-3.99,-4v3"/>
11     </group>
12
13     <group android:name="right">
14         <path
15             android:fillColor="#FF000000"
16            android:pathData="M14.99,13v-3L22,10L22,8h-7.01L14.99,5L11,9l3.99,4"/>
17     </group>
18
19 </vector>
```

可以发现，这里的Vector图像比之前我们看见的要多了一个group标签。group标签的作用有两个：

- 对Path进行分组，由于我们后面需要针对Path进行动画，所以可以让具有同样动画效果的Path在同一个Group中
- 拓展动画效果，单个的path标签是没有translateX和translateY属性的，因此无法使用属性动画来控制path translateY，我们需要先将相关的path标签元素包裹在一个个的group标签中。

动画效果

XXXXX2实际上就是模板要实现的动画，动画效果实际上就是基础的属性动画，例如：

anim\_left.xml

```
1 <objectAnimator
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:duration="1000"
4     android:interpolator="@android:interpolator/anticipate_overshoot"
5     android:propertyName="translateX"
6     android:repeatCount="infinite"
7     android:repeatMode="reverse"
8     android:valueFrom="0"
9     android:valueTo="-10"
10    android:valueType="floatType"/>
```

anim\_right.xml

```
1 <objectAnimator
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:duration="1000"
4     android:interpolator="@android:interpolator/anticipate_overshoot"
5     android:propertyName="translateX"
6     android:repeatCount="infinite"
7     android:repeatMode="reverse"
8     android:valueFrom="0"
9     android:valueTo="10"
10    android:valueType="floatType"/>
```

在代码中使用

```
1 ImageView imageView = (ImageView) findViewById(R.id.iv);
2 AnimatedVectorDrawableCompat animatedVectorDrawableCompat = AnimatedVectorDrawableCompat.create(
3     this, R.drawable.square_anim
4 );
5 imageView.setImageDrawable(animatedVectorDrawableCompat);
6 ((Animatable) imageView.getDrawable()).start();
```

动态Vector兼容性问题

向下兼容问题

一说到兼容，就不得不提到坑，几乎所有的为了兼容而做的改动，都会留下一些不可填满的坑，动态Vector动画也不例外，虽然像进行了Android 2.1以上的兼容，但对于动态Vector动画，还是有很多限制的，例如：

- Path Morphing，即路径变换动画，在Android pre-L版本下是无法使用的。
- Path Interpolation，即路径插值器，在Android pre-L版本只能使用系统的插值器，不能自定义。
- Path Animation，即路径动画，这个一般使用贝塞尔曲线来代替，所以没有太大影响。

向上兼容问题

除了在低版本上的兼容性问题，在L版本以上，也存在兼容性问题，即继承了AppCompatActivity的界面，如果直接设置Image Path Morphing动画是无法生效的，因为默认的AppCompatActivity已经默认使用ImageViewCompat给转换了，但是AnimatedVectorDrawableCompat是不支持Path Morphing动画的，所以，在AppCompatActivity界面里面就无效了。

解决办法很简单，即使用代码来给ImageView添加动画：

```
1 ImageView imageView = (ImageView) view;
2 AnimatedVectorDrawable morphing = (AnimatedVectorDrawable) getDrawable(morphing);
3 imageView.setImageDrawable(morphing);
```



99元特价机票

德赛西威怎么样

千炮捕鱼电脑版

固态硬盘排行榜

```
4  if (morphing != null) {
5      morphing.start();
6  }
```

注意不要使用AnimatedVectorDrawableCompat即可。

抽取string兼容问题

开发者有时候为了代码简洁可能会把Vector图像中的pathData放到string.xml中，然后在Vector图像中引用string。

但这种方式如果通过生成png来兼容5.0以下机型的话，会报pathData错误，编译器不会去读取string.xml，只能把pathData文件中也是一样，这也是为了兼容做出的牺牲吗，不得而知。

其它兼容问题

其它非常奇怪、诡异、不能理解的兼容性问题，只能通过版本文件夹的方式来进行兼容了，例如drawable-v21，分资源在两个文件夹下，这样在21以上版本，会使用drawable-v21的资源，而其它会使用drawable下的资源。

动态Vector进阶

用好ObjectAnimator

所谓Vector动画进阶，实际上就是在利用ObjectAnimator的一些属性，特别是trimPathStart、trimPathEnd这两个针对Vector属性不兼容pre-L）。

这两个属性的官方文档如下所示：

1	android:trimPathStart
2	The fraction of the path to trim from the start, in the range from 0 to 1.
3	android:trimPathEnd
4	The fraction of the path to trim from the end, in the range from 0 to 1.
5	android:trimPathOffset
6	Shift trim region (allows showed region to include the start and end), in the range from 0 to 1.

其实很简单，就是一个图像的截取，设置一个比例即可，即当前绘制多少比例的图像，其余部分不绘制，Start和End分别就是从始算，大家参考几个例子就能理解了。

理解Path Morph

Path Morph动画是Vector动画的一个高级使用，说到底，也就是两个PathData的转换，但是这种转换并不是随心所欲的，对于行Path Morph的前提是，它们具有相同个数的关键点，即两个路径的变换，只是关键点的坐标变化，掌握了这一个基本原理，就容易了。

学习Vector

在Github上我开源了一个Vector的动画Demo库，地址如下所示：

<https://github.com/xuyisheng/VectorDemo>

这个Demo分为两部分，一部分是可以兼容Android pre-L版本和L+版本的Vector动画，另一部分（通过ActionBar的按钮切换动画。

每个Vector动画，基本都包含四部分内容，即：

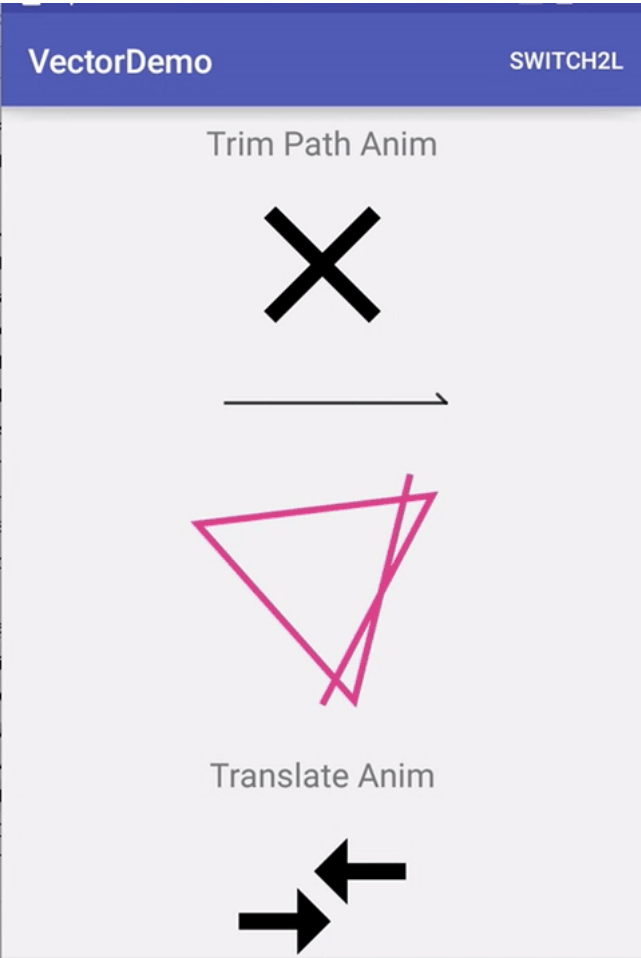
- Vector：图像资源
- Animated-vector：动画、图像粘合剂
- ObjectAnimator：动画资源
- 代码：启动动画

每个Vector动画通过这四个部分去进行分析，就非常清晰了。





这里展示下Demo的效果图：



Vector性能问题

有读者在文章后面留言，询问VectorDrawable的性能问题，这里解释一下。

- 1. Bitmap的绘制效率并不一定会比Vector高，它们有一定的平衡点，当Vector比较简单时，其效率是一定比Bitmap高的，高效率，Vector需要更加简单，PathData更加标准、精简，当Vector图像变得非常复杂时，就需要使用Bitmap来代替了
- 2. Vector适用于ICON、Button、ImageView的图标等小的ICON，或者是需要的动画效果，由于Bitmap在GPU中有缓存所以Vector图像不能做频繁的重绘
- 3. Vector图像过于复杂时，不仅仅要注意绘制效率，初始化效率也是需要考虑的重要因素
- 4. SVG加载速度会快于PNG，但渲染速度会慢于PNG，毕竟PNG有硬件加速，但平均下来，加载速度的提升弥补了绘制的时

Google的这个视频中，已经对Vector的效率问题做了解释，可以参考下：

<https://www.youtube.com/watch?v=wIFVIlstKmA&feature=youtu.be&t=6m3s>

参考

<https://medium.com/@shemag8/animated-vector-drawable-e4d7743d372c#.3vkt12j20>

<https://github.com/jpuderer/AnimatedButton>

顶  
12

踩  
0

技术、售后、段子



- 上一篇 Android Studio集成Bug管理系统
- 下一篇 贝塞尔曲线开发的艺术



我的同类文章

Android ( 186 )

• 自律给你自由——设计布局的新姿势

2016-09-22

阅读 7507

• 《Android群英传:神兵利器》勘误

2016-08-28

阅读 1053

• PathMeasure之迷径追踪

2016-07-22

阅读 4898

• Android Studio集成Bug管理系统

2016-06-08

阅读 2802

• 一个字符解决Gradle aar编译参数传递...

2016-04-29

阅读 2419

• 推送 从入门到放弃

2016-09-

• Android群英传:神兵利器——出版祭

2016-08-

• 贝塞尔曲线开发的艺术

2016-07-

• 隐藏Boss——ddmlib使用入门

2016-05-

• 30天入门 Android 开发，Google 与...

2016-03-

更多文章

参考知识库



Android知识库  
12674 关注 | 1190 收录

猜你在找

Android高级界面控件难点精讲  
Android《自定义控件》视频，震…  
Android之动画全讲  
Android自定义控件系列之九宫格…  
Android移植基础

Python配置第三方库Theano的曲…  
反弹之路曲折  
电商老大的短腿阿里巴巴曲折的…  
项目曲折进行之路第一篇  
项目曲折进行之路前夕篇



MetLife  
大都會人壽

美國最大人壽保險公司\*  
專業財務管理團隊 助您人生升級

\*連續2015年南德保險獎得主，連續13年A++最高信評

立即聯絡我們進行  
免費財務狀況分析

查看评论



情随事迁666  
api 17并没有出现android.content.res.Resources\$NotFoundException。

14楼 20



Albert凯尔  
666 SVG挺强大的，希望更多的应用能使用

13楼 20



Aven Huang  
svg虽然很6，但是效率问题也不容忽视，特别是Android的Vector，效率更低，比自己解析Svg文件慢了去了，原因就是Android官方将简单的Rect、Circle都用path来绘制了，我试过，时间差距从2倍到20倍不等，如果要使用svg，还是推荐自己解析svg文件，但解析过程的效率还是要想办法解决的。

12楼 20



Aven Huang  
svg虽然很6，但是效率问题也不容忽视，特别是Android的Vector，效率更低，比自己解析Svg文件慢了去了，原因就是Android官方将简单的Rect、Circle都用path来绘制了，我试过，时间差距从2倍到20倍不等，如果要使用svg，还是推荐自己解析svg文件，但解析过程的效率还是要想办法解决的

11楼 20



eclipse\_xu  
回复Aven Huang：想和你沟通一下这个问题 不知道方便留个联系方式吗



eclipse\_xu  
回复Aven Huang：效率问题你是如何检测的？

技术、售后、段子





踏驴行者  
回复eclipse\_xu：同问，想知道效率问题



大苞米  
666

10楼 20



业精于勤而荒于嬉  
这个牛，

9楼 20



刘某人程序员  
get

8楼 20



灰翼  
mark一下，以后没准用得上

7楼 20



AnalyzeSystem  
算是我看过的最全面的了，跪谢了

6楼 20



gouyuhua2008  
写得不错，10086个赞

5楼 20



SunnyCoffee  
很强大。感觉不久就会出现大批的矢量图库，解决适配和动画效果的问题。

4楼 20



tiantang\_1991  
好厉害

3楼 20



aa51513  
微信群过来的，非常感谢大牛，学习了

2楼 20



nwsuafer  
32个赞，医生

1楼 20

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	VPI	
IE10	Eclipse	CRM	JavaScript	数据库	Ubuntu	NFC	WAP	jQuery	BI	HTML5	Spring	Apa
HTML	SDK	IIS	Fedora	XML	LBS	Unity	Splashtop	UML	components	Windows Mobile	Rai	
Cassandra	CloudStack	FTC	coremail	OPhone	CouchBase	云计算	iOS6	技术、售后、段子				
Compuware	大数据	aptech	Perl	Tornado	Ruby	Hibernate	ThinkPHP					
Cloud Foundry	Redis	Scala	Django	Bootstrap								



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

京ICP10000000号

CSDN.NET, All Rights Reserved



99元特价机票



德赛西威怎么样



千炮捕鱼电脑版



固态硬盘排行榜

