Android 属性动画和视图动画的区别 - 简书

> 提起动画首先想到的是动画片，童年最爱的黑猫警长，葫芦娃，邋遢大王，大脸猫和蓝皮鼠，四驱兄弟，七龙珠等等，当时觉着好神奇，他们是怎么动的呢。动画其实就是一帧帧的画面顺序播放，只...

0.2882019.08.03 07:10:20 字数 2,123 阅读 817

提起动画首先想到的是动画片，童年最爱的黑猫警长，葫芦娃，邋遢大王，大脸猫和蓝皮鼠，四驱兄弟，七龙珠等等，当时觉着好神奇，他们是怎么动的呢。动画其实就是一帧帧的画面顺序播放，只要画面切换的够快, 会我们眼中形成视觉残留的效果，人眼不会感觉到突兀也就会认为画面是连续运动的，至于 24 帧什么的这里就不说了，现在的手机普遍支持 60 帧每秒，微鲸的 vr 设备好像 87 帧每秒，只要游戏本身不卡顿，玩起来还是挺流畅的。动画可以实现各种各样的特效，android 平台开发过程中动画也相当重要。
Android 平台提供了强大的动画框架，使我们在完成复杂的特效时不用自己改变空间的位置或者设置属性通过简单的动画代码就可以实现，android3.0 之前，主要包括两种动画方式：补间动画（Tween Animation）和帧动画（Frame Animation 或者 Drawable Animation），这两种动画统称为 view 动画，针对视图动画存在的不足，3.0 之后 google 增加了属性动画（Property Animation）。之后动画就被分成了 View Animation 和 Property Animation。

关于动画的一道面试题：
在应用中常看到不断变化的数字怎么实现?

视图动画：

```
补间动画是视图动画的一种，Tween中文意思是在两者之间，中文翻译成补间还是挺贴合意思的，view从一个位置的特定状态变化到另外一个位置，中间过程我们开发者不
```

补间动画可以（仅可以）完成 view 的位置、大小、旋转、透明度等一系列简单的变换，可通过 xml 文件或者代码实现，通常补件动画都是利用 xml 文件实现，属性设置简单明了，又可以重复使用。

```
基于View的渐变动画，她只改变了View的绘制效果，而实际属性值未变。比如动画移动一个控件的位置，但控件实际位置仍未改变，如果我们此时想选中控件，它的位置仍
```

- alpha 渐变透明度动画效果
- scale 渐变尺寸伸缩动画效果
- translate 画面转换位置移动动画效果
- rotate 画面转移旋转动画效果
  举例：放在 res/anim / 文件夹下：

```
<?xml version="1.0" encoding="utf-8"?>
    <scale xmlns:android="http://schemas.android.com/apk/res/android"
        android:fromXScale="0.0"
        android:toXScale="1.1"
        android:fromYScale="0.0"
        android:toYScale="1.1"
        android:pivotX=30"
        android:pivotY="20"
        android:duration="500" />
```

补间动画使用很方便，但也有很大的局限性，首先动画虽然执行了但是只是我们看到的，控件真实的位置透明度等都没有变化，如果我们想对一个已经移动的控件执行点击事件，如果控件移动的位置比较大点击界面上看到的控件是无法触发事件的。

补间动画只能针对设计好的特定的几种属性执行动画，对于自定义的属性则不太好完成（或者说根本不支持）。

2 帧动画（Frame Animation 或者 Drawable Animation）：

帧动画也是 view 动画的一种，帧动画是通过读取 xml 文件中设置的一系列 Drawable，以类似幻听片的方式展示这些 drawable, 就形成了动画效果，当然也可以利用代码实现帧动画。可能大家觉着帧动画不太常用，其实类似的原理可以借鉴，类似 android 手机开机的很多动画效果就是类似帧动画，加载一系列图片，实现开机的动画效果。

```
在代码中定义动画帧，使用AnimationDrawable类；XML文件能更简单的组成动画帧，在res/drawable文件夹，使用<animation-list>采用<item>来定义不同的帧。
```

但是依旧推荐使用 xml，具体如下：
<animation-list> 必须是根节点，包含一个或者多个 < item > 元素，属性有：

· android:oneshot true 代表只执行一次，false 循环执行。
· <item> 类似一帧的动画资源。
<item> animation-list 的子项，包含属性如下：
· android:drawable 一个 frame 的 Drawable 资源。
· android:duration 一个 frame 显示多长时间。
文件放在 res/drawable / 目录下

```
<?xml version="1.0" encoding="utf-8"?><animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot=["true" | "false"] >
```

```
    <item
        android:drawable="drawable_resource"
        android:duration="integer" /></animation-list>
```

帧动画是最简单的动画，用的比较少。

属性动画在视图动画之后推出，API 11 以上才能够使用，是为了弥补视图动画存在的问题，从名字可以看出属性动画和视图动画的不同，视图动画主要针对视图起作用，

对数值的操作：

```
ValueAnimator animator = ValueAnimator.ofInt(0,700);

 animator.setDuration(1000);

 animator.start();
```

对 view 的操作：

```
ObjectAnimator animator = ObjectAnimator.ofFloat(view,"alpha",1,0,1);

animator.setDuration(1000);

animator.start();
```

- 第一个参数用于指定这个动画要操作的是哪个控件
- 第二个参数用于指定这个动画要操作这个控件的哪个属性
- 第三个参数是可变长参数，这个就跟 ValueAnimator 中的可变长参数的意义一样了，就是指这个属性值是从哪变到哪

添加监听：

```
ValueAnimator animator = ValueAnimator.ofInt(0,400);

animator.setDuration(1000);

animator.addUpdateListener(new ValueAnimator.AnimatorUpdateListener() {

    @Override

   public void onAnimationUpdate(ValueAnimator animation) {

       int valcur = (int)animation.getAnimatedValue();

    }

});

animator.start();
```

由于出现的时间不同，视图动画和属性动画有几个显而易见的不同：

视图动画早于属性动画，视图动画在 API 1 里面就已经存在，属性动画直到 API3.0 才出现，视图动画所在的包名为 android.view.animation, 属性动画为 android.animation, 可见视图动画只针对 view 起作用；试图动画中用到的类一般以 Animation 结尾，而属性动画则以 Animator 结尾。

为什么引入视图动画：

帧动画可以通过顺序播放资源来实现动画的，补间动画可以实现控件的渐入渐出、移动、旋转和缩放效果。但类似利用动画改变 View 的背景或者改变一个数值或者改变一个对象的属性视图动画就无法完成此类工作了，所以视图动画存在局限性。属性动画还支持监听动画过程，在动画过程中自己操作控件进行改变。

视图动画的文件在 android.view.animation 下，通过学习也可以知道，视图动画只能作用于视图，实现类似缩放、旋转功能，动画效果比较固定；而属性动画可以通过改变 View 的属性完成动画，利用 setxxx() 和 getxxx() 函数可以对 Object 的任意属性改变，从而可以实现视图动画实现不了的功能。可以通过在 object 中添加属性的 set 函数，在 ondraw 方法中操作属性就可以完成动画

属性动画通过改变 view 属性实现动画，而视图动画虽然利用动画改变了 view 的位置和大小，但 view 真正的属性没有改变，这就会导致许多问题，视图动画过后我们再去操作 view 则得不到响应，这就是为什么自定义控件利用动画改变控件位置后还有调用 layout（）设置 view 位置的原因，这样才能真正的改变 view 的位置，而属性动画就可以一步到位的完成。

总结如下：

（1）属性动画比视图动画更强大，不但可以实现缩放、平移等操作，还可以自己定义动画效果，监听动画的过程，在动画过程中或完成后做响应的动作。
（2）属性动画不但可以作用于 View，还能作用于 Object。
（3）属性动画利用属性的改变实现动画，而视图动画仅仅改变了 view 的大小位置，但 view 真正的属性没有改变。

完！后续会对属性动画，视图动画，插值器，自定义属性动画等进行系列介绍。

Animation 动画概述和执行原理
Android 动画之补间动画 TweenAnimation
Android 动画之逐帧动画 FrameAnimation
Android 动画之插值器简介和系统默认插值器
Android 动画之插值器 Interpolator 自定义
Android 动画之视图动画的缺点和属性动画的引入
Android 动画之 ValueAnimator 用法和自定义估值器
Android 动画之 ObjectAnimator 实现补间动画和 ObjectAnimator 自定义属性
Android 动画之 ObjectAnimator 中 ofXX 函数全解析 - 自定义 Property，TypeConverter，TypeEvaluator