

Android 补间动画

作者: [RustFisher](#)

原文链接: <https://ld246.com/article/1628665968431>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

补间动画：通过使用 **Animation** 对单张图片执行一系列转换来创建动画。
在 XML 中定义的动画，用于对图形执行旋转、淡出、移动和拉伸等转换。
动画文件放在 `res/anim/`，该文件名将用作资源 ID。

移动示例

先用一个例子建立直观的认识。新建一个动画xml，实现从左到右的效果。

`move_hor_1.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate
    android:duration="200"
    android:toXDelta="-100%" />

  <translate
    android:duration="400"
    android:startOffset="200"
    android:toXDelta="100%" />
</set>
```

要使用这个动画，需要用到 `AnimationUtils.loadAnimation(getContext(), R.anim.move_hor_1)`;
加载动画。得到对象 `mAnimation`

交给 `View.startAnimation(mAnimation)`来启动。

上面的动画例子效果图



仔细看一下动画xml里的内容

- `set`里有2个`translate`
- `translate`表示移动动作
 - `duration`是这个动作的执行时长（毫秒）
 - `toXDelta`表示横行移动

- `startOffset`表示动画开始执行多久后再执行这个动作

实际上，第二个 `translate`周期是400毫秒，它“等了”200毫秒才开始执行。把View从左边移动到了右边。

缩放示例

除了位移，还可以执行缩放效果。



可以看到示例种先放大再缩小回去。主要是用了 `android:repeatMode="reverse"`和 `android:repeatCount="1"`。

完整动画xml如下

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <scale
    android:duration="600"
    android:fromXScale="1"
    android:fromYScale="1"
    android:pivotX="50%"
    android:pivotY="50%"
    android:repeatCount="1"
    android:repeatMode="reverse"
    android:toXScale="1.35"
    android:toYScale="1.35" />
</set>
```

`scale`里面我们遇到了几个元素 `pivotX`, `repeatCount`, `repeatMode`

- `android:pivotX` 在对象缩放时要保持不变的 X 坐标; `pivotY`同理。
- `android:repeatCount` 动画的重复次数。设为 "-1" 表示无限次重复，也可设为正整数。
 - 例如，值 "1" 表示动画在初次播放后重复播放一次，因此动画总共播放两次。默认值为 "0"，表不重复。

- `android:repeatMode` 动画播放到结尾处的行为。 `android:repeatCount` 必须设置为正整数或 "-1"，该属性才有效。
 - 设置为 "reverse" 可让动画在每次迭代时反向播放，
 - 设置为 "repeat" 则可让动画每次从头开始循环播放。

多个动画元素的例子

一个 `<set>` 集合里可以包含多个元素。



录制gif的出来发现快结束时总有一个抖动，是录制的问题。

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <rotate
        android:duration="500"
        android:fromDegrees="0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toDegrees="360" />

    <translate
        android:duration="500"
        android:fromXDelta="0%"
        android:toXDelta="50%" />

    <alpha
        android:duration="500"
        android:fromAlpha="1"
        android:toAlpha="0.5" />

    <set
        android:interpolator="@android:anim/decelerate_interpolator"
        android:startOffset="500">
```

```
<rotate
  android:duration="400"
  android:fromDegrees="360"
  android:pivotX="100%"
  android:pivotY="50%"
  android:toDegrees="0" />
```

```
<translate
  android:duration="400"
  android:fromXDelta="0%"
  android:toXDelta="-50%" />
```

```
<alpha
  android:duration="400"
  android:fromAlpha="0.5"
  android:toAlpha="1" />
```

```
</set>
</set>
```

<set>里放置着多个元素，可以把 <set> 放进 <set> 里。

我们分成出发和回来两个动作

出发：

第一个 <rotate> 是从自己的中心位置开始。

<translate> 水平向右走了50%。

回来：

后面 <rotate> 的 pivotX 需要考虑到出发时水平移动了50%，因此 pivotX 取值100%。

而 <translate> 从0%走到-50%即可。

动画xml元素

下面总结一下各个属性

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
  android:interpolator="@[package:]anim/interpolator_resource"
  android:shareInterpolator=["true" | "false"] >
  <alpha
    android:fromAlpha="float"
    android:toAlpha="float" />
  <scale
    android:fromXScale="float"
    android:toXScale="float"
    android:fromYScale="float"
    android:toYScale="float"
    android:pivotX="float"
    android:pivotY="float" />
  <translate
```

```
    android:fromXDelta="float"
    android:toXDelta="float"
    android:fromYDelta="float"
    android:toYDelta="float" />
<rotate
    android:fromDegrees="float"
    android:toDegrees="float"
    android:pivotX="float"
    android:pivotY="float" />
<set>
    ...
</set>
</set>
```

set

容纳其他动画元素（`<objectAnimator>`、`<valueAnimator>` 或其他 `<set>` 元素）的容器。代表 **AnimatorSet**。

可以放置多个动画元素，组合在一起。每个set可定义自己的 `ordering` 属性。

- `android:interpolator` 插值器资源。要应用于动画的 **Interpolator**。该值必须是对指定插值器的源的引用（而不是插值器类名称）。可使用平台提供的默认插值器资源，也可创建自己的插值器资源。
- `android:shareInterpolator` 布尔值。如果要在所有子元素中共用同一插值器，则为 “true” 。

alpha

淡入或淡出动画。对应 **AlphaAnimation** 类。

- `android:fromAlpha` 浮点数。起始不透明度偏移，0.0 表示透明，1.0 表示不透明。
- `android:toAlpha` 浮点数。结束不透明度偏移，0.0 表示透明，1.0 表示不透明。

scale

缩放动画，调整大小的动画；对应 **ScaleAnimation**。

指定 `pivotX` 和 `pivotY`，来指定View向外（或向内）扩展的中心点。

例如，如果这两个值为 0、0（左上角），则所有扩展均向右下方向进行。

- `android:fromXScale` 浮点数。起始 X 尺寸偏移，其中 1.0 表示不变。
- `android:toXScale` 浮点数。结束 X 尺寸偏移，其中 1.0 表示不变。
- `android:fromYScale` 浮点数。起始 Y 尺寸偏移，其中 1.0 表示不变。
- `android:toYScale` 浮点数。结束 Y 尺寸偏移，其中 1.0 表示不变。
- `android:pivotX` 浮点数。在对象缩放时要保持不变的 X 坐标。
- `android:pivotY` 浮点数。在对象缩放时要保持不变的 Y 坐标。

translate

移动动画。垂直或水平移动。或者水平和垂直移动一起。对应 **TranslateAnimation**。

支持采用以下三种格式之一的以下属性：从 -100 到 100 的以 “%” 结尾的值，表示相对于自身的百分比；从 -100 到 100 的以 “%p” 结尾的值，表示相对于其父项的百分比；不带后缀的浮点值，表示对值。

属性：

- **android:fromXDelta** float或百分比。起始 X 偏移。表示方式：相对于正常位置的像素数（例如 "5"），相对于元素宽度的百分比（例如 "5%"），或相对于父项宽度的百分比（例如 "5%p"）。
- **android:toXDelta** float或百分比。结束 X 偏移。表示方式同**fromXDelta**
- **android:fromYDelta** float或百分比。起始 Y 偏移。表示方式：相对于正常位置的像素数（例如 "5"），相对于元素高度的百分比（例如 "5%"），或相对于父项高度的百分比（例如 "5%p"）。
- **android:toYDelta** float或百分比。结束 Y 偏移。表示方式同**fromYDelta**

rotate

旋转动画。对应**RotateAnimation**。

属性：

- **android:fromDegrees**浮点数。起始角度位置，以度为单位。
- **android:toDegrees**浮点数。结束角度位置，以度为单位。
- **android:pivotX**float或百分比。旋转中心的 X 坐标。表示方式：相对于对象左边缘的像素数（例如 "5"），相对于对象左边缘的百分比（例如 "5%"），或相对于父级容器左边缘的百分比（例如 "5%p"）。
- **android:pivotY**float或百分比。旋转中心的 Y 坐标。表示方式：相对于对象上边缘的像素数（例如 "5"），相对于对象上边缘的百分比（例如 "5%"），或相对于父级容器上边缘的百分比（例如 "5%p"）。

参考

- Tween动画指南 <https://developer.android.com/guide/topics/resources/animation-resource#Tween>
- res简介 https://an.rustfisher.com/android/resource/resources_overview/