# 概述

Android提供了几种动画类型：View Animation、Drawable Animation、Property Animation。View Animation相当简单，不过只能支持简单的缩放、平移、旋转、透明度基本的动画，且有一定的局限性。比如：你希望View有一个颜色的切换动画；你希望可以使用3D旋转动画；你希望当动画停止时，View的位置就是当前的位置；这些View Animation都无法做到。这就是Property Animation产生的原因，本篇详细介绍Property Animation的用法。

# 2、相关API

Property Animation故名思议就是通过动画的方式改变对象的属性了，我们首先需要了解几个属性和相关的类：

1、属性

Duration动画的持续时间，默认300ms。

Time interpolation：时间差值，乍一看不知道是什么，但是我说LinearInterpolator、AccelerateDecelerateInterpolator，大家一定知道是干嘛的了，定义动画的变化率。

Repeat count and behavior：重复次数、以及重复模式；可以定义多少次；重复时重头开始，还是反向。

Animator sets：动画集合，你可以定义一组动画，一起执行或者顺序执行。

Frame refresh delay：帧刷新延迟，对于你的动画，多久刷新一次帧；默认为10ms，但最终依赖系统当前的状态，基本不用管。

1. 相关的类：

ObjectAnimator：动画的执行类，后面详细介绍；

ValueAnimator：动画的执行类，后面详细介绍；

AnimatorSet：用于控制一组动画的执行：线性，一起，每个动画的先后执行等；

AnimatorInflater：用户加载属性动画的Xml文件；

TypeEvaluator：类型估值，主要用于设置动画操作属性的值；

TimeInterpolator：时间差值，上面已经介绍。

总的来说，属性动画就是，动画的执行类来设置动画操作的对象的属性、持续时间，开始和结束的属性值，时间差值等，然后系统会根据设置的参数动态的变化对象的属性。

# ObjectAnimator实现动画

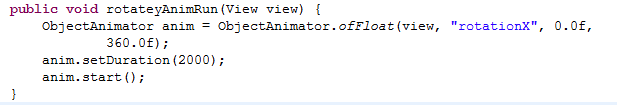
之所以选择ObjectAnimator为第一，这个实现最简单，一行代码，秒秒钟实现动画，下面看个例子：

布局文件：



很简单，就一个TextView和两个Button；

Activity代码（重要部分）：



效果见Demo

是不是一行代码就能实现简单的动画。

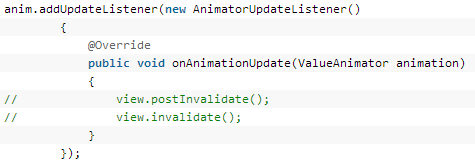
对于ObjectAnimator

1. 提供ofInt、ofFloat、ofObject，这几个方法都是设置动画作用的元素、作用的属性、动画开始、结束、已经中间的任意个属性值。

当对于属性，只设置一个的时候，会认为当然对象该属性的值为开始（getPropName放射获取），然后设置的值为终点。如果设置两个，则一个为开始、一个为结束

动画更新过程中，会不断调用setPropName更新元素的属性，所有使用ObjectAnimator更新某个属性，必须得有getter（设置一个属性值的时候）和setter方法

1. 如果你操作对象的该属性方法里面，比如上例的setRotationX如果内部没有调用view的重绘，则需要你自己按照下面方式手动调用：



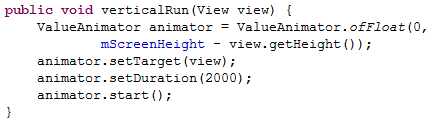
1. 看了上面的例子，因为设置的操作的属性只有一个，那么如果我希望一个动画能够让View既可以缩小、又能够淡出（3个属性scaleX,scaleY,alpha），只使用ObjectAnimator咋弄？

法是不是很不错，可能会说使用AnimatorSet啊，这一看就是一堆动画塞一起执行，但是我偏偏要用一个ObjectAnimator实例实现呢~下面看代码：

把设置属性的那个字符串，随便写一个该对象没有的属性，就是不管~~咱们只需要它按照时间插值和持续时间计算的那个值，我们自己手动调用

# ValueAnimator实现动画

和ObjectAnimator用法很类似，简单看一下用view垂直移动的动画代码：



给你的感觉是不是，坑爹啊，这和ValueAnimator有毛线区别~但是仔细看，你看会发现，没有设置操作的属性~~也就是说，上述代码是没有任何效果的，没有指定属性~

这就是和ValueAnimator的区别之处：ValueAnimator并没有在属性上做操作，你可能会问这样有啥好处？我岂不是还得手动设置？

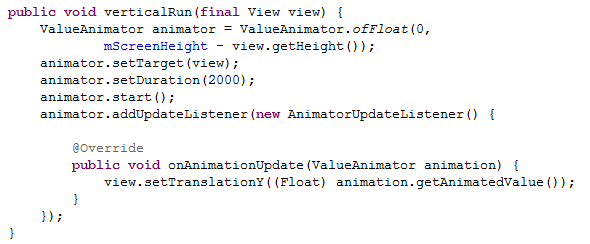
好处：不需要操作的对象的属性一定要有getter和setter方法，你可以自己根据当前动画的计算值，来操作任何属性，记得上例的那个【我希望一个动画能够让View既可以缩小、又能够淡出（3个属性scaleX,scaleY,alpha）】吗？其实就是这么个用法

实例：

布局文件：



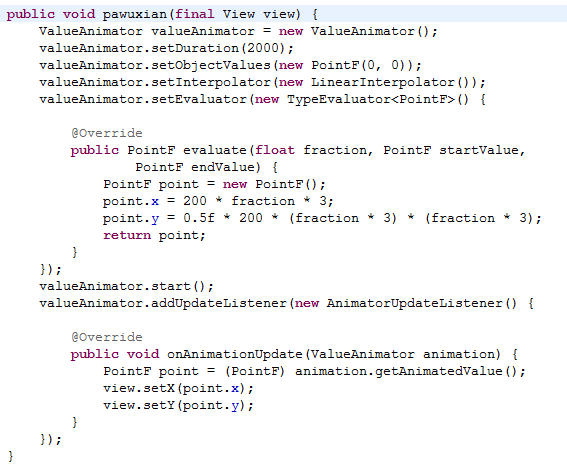
左上角一个TextView，底部两个按钮~我们先看下落体的代码：



与ObjectAnimator不同的就是我们自己设置元素属性的更新~虽然多了几行代码，但是貌似提高灵活性

下面再来一个例子，如果我希望小球抛物线运动【实现抛物线的效果，水平方向100px/s，垂直方向加速度200px/s\*s 】，分析一下，貌似只和时间有关系，但是根据时间的变化，横向和纵向的移动速率是不同的，我们该咋实现呢？此时就要重写TypeValue的时候了，因为我们在时间变化的同时，需要返回给对象两个值，x当前位置，y当前位置：

代码：



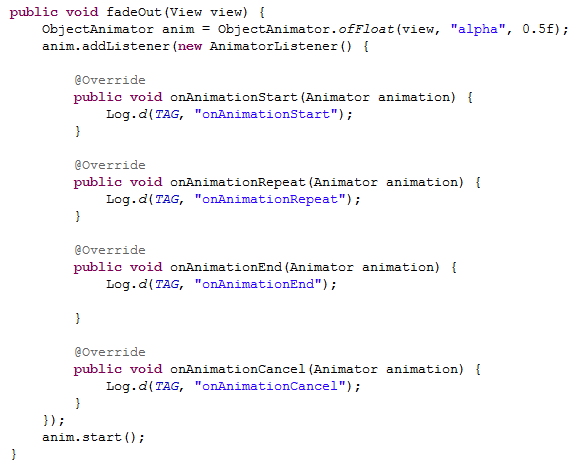
可以看到，因为ofInt,ofFloat等无法使用，我们自定义了一个TypeValue，每次根据当前时间返回一个PointF对象，（PointF和Point的区别就是x,y的单位一个是float,一个是int;RectF,Rect也是）PointF中包含了x,y的当前位置～然后我们在监听器中获取，动态设置属性

好了，我们已经分别讲解了ValueAnimator和ObjectAnimator实现动画；二者区别；如何利用部分API，自己更新属性实现效果

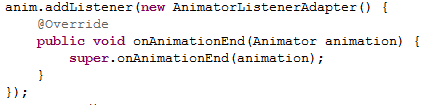
# 监听动画事件

对于动画，一般都是一些辅助效果，比如我要删除个元素，我可能希望是个淡出的效果，但是最终还是要删掉，并不是你透明度没有了，还占着位置，所以我们需要知道动画如何结束。

所以我们可以添加一个动画的监听:



这样就可以监听动画的开始、结束、被取消、重复等事件~但是有时候会觉得，我只要知道结束就行了，这么长的代码我不能接受，那你可以使用AnimatorListenerAdapter



AnimatorListenerAdapter继承了AnimatorListener接口，然后空实现了所有的方法~

animator还有cancel()和end()方法：cancel动画立即停止，停在当前的位置；end动画直接到最终状态。

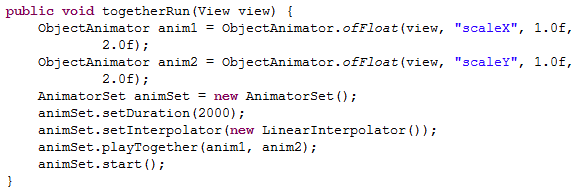
# AnimatorSet的使用

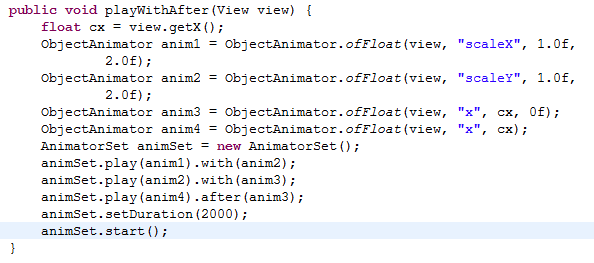
实例：

布局文件：



代码：





写了两个效果：

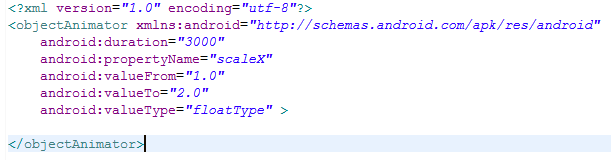
第一：使用playTogether两个动画同时执行，当然还有playSequentially依次执行~~

第二：如果我们有一堆动画，如何使用代码控制顺序，比如1，2同时；3在2后面；4在1之前等~就是效果2了

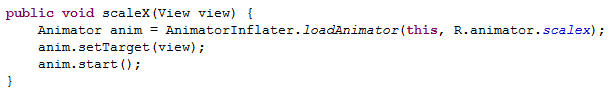
# 使用XML文件来创建属性动画

大家肯定都清楚，View Animator、Drawable Animator都可以在anim文件夹下创建动画，然后在程序中使用，设置在Theme中设置为属性值。当然了，属性动画其实也可以在文件中声明：

首先在res下建立animator文件夹，然后建立res/animator/scalex.xml



代码：



使用AnimatorInflater加载动画的资源文件，然后设置目标，就ok，这只是单纯的横向放大一倍，如果希望纵向与横向同时缩放呢？则可以这么定义属性文件：



使用set标签，有一个orderring属性设置为together,【还有另一个值：sequentially（表示一个接一个执行）】。