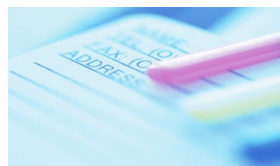




广东岭南职业技术学院
LINGNAN INSTITUTE OF
TECHNOLOGY



Android 任务驱动式教程



广东岭南职业技术学院
LINGNAN INSTITUTE OF
TECHNOLOGY

任务5

Animation





- 什么是 Animations
- Animations 的分类
- Animations 的使用方法
- LayoutAnimationController 的使用方法
- ListView 与 Animations 结合使用
- AnimationListener 的使用方法





什么是 Animations



Animations提供了一系列的动画效果，这些效果可以应用在绝大多数的控件；





○ 两种动画

- Tweened Animation(渐变动画)
- Frame By Frame (逐帧动画)



Animations从总体来说可以分为两大类：

第一类：Tweened Animations

该类Animations提供了旋转，移动，伸展和淡出等等效果；

第二类：Frame-by-Frame Animations

这一类Animations可以创建一个Drawable序列，这些Drawable可以按照指定的时间间歇一个一个的显示；



○ Tweened Animation

● 4 种类型

- **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** 画面转换位置移动动画效果。



使用 Tweened Animations 的步骤

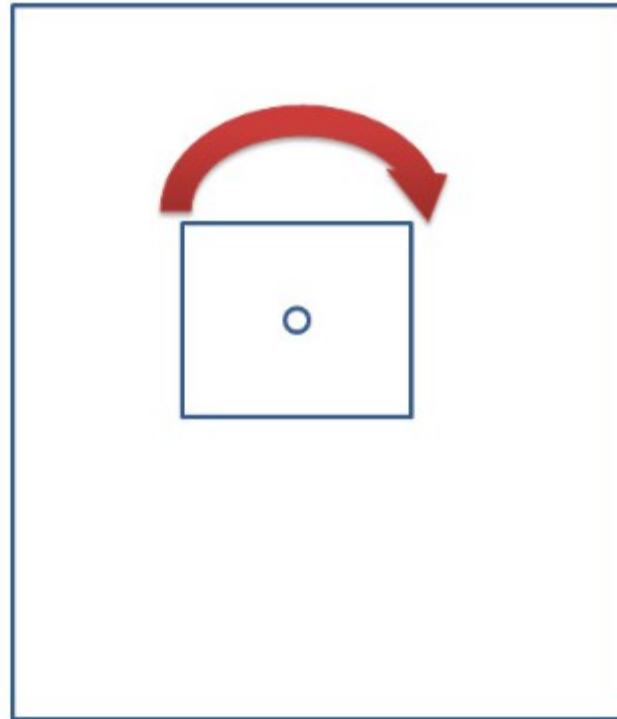


1. 创建一个AnimationSet对象；
2. 根据需要创建相应的Animation对象；
3. 根据软件动画的需求，为Animation对象设置相应的数据；
4. 将Animation对象添加到AnimationSet对象当中；
5. 使用控件对象开始执行AnimationSet；



广东岭南职业技术学院
LINGNAN INSTITUTE OF
TECHNOLOGY

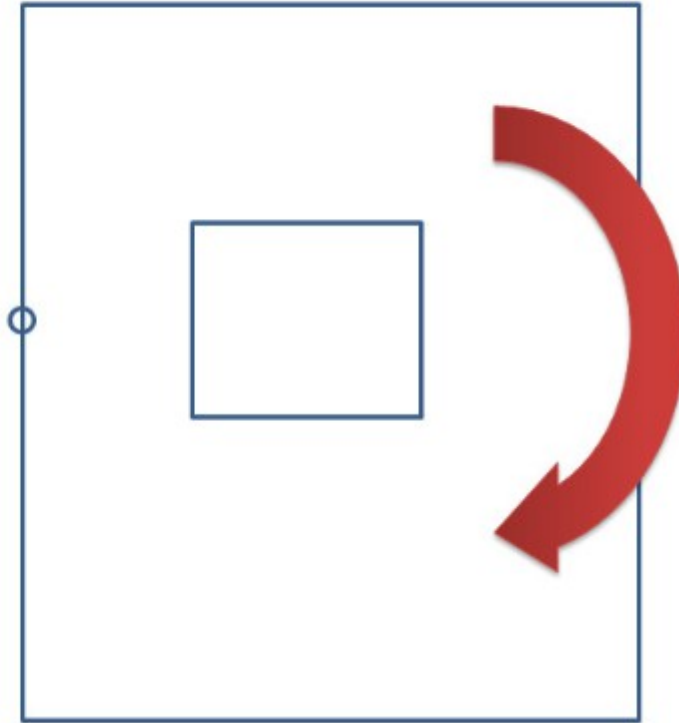
Animation.RELATIVE_TO_SELF





广东岭南职业技术学院
LINGNAN INSTITUTE OF
TECHNOLOGY

Animation.RELATIVE_TO_PARENT





各种 Animation 的主要属性

ANIMATION TYPE	ATTRIBUTES	VALID VALUES
Alpha	fromAlpha/toAlpha	Float from 0 to 1
Scale	fromXScale/toXScale	Float from 0 to 1
	fromYScale/toYScale	Float from 0 to 1
	pivotX/pivotY	String of the percentage of graphic width/height from 0% to 100%
Translate	fromX/to X	Float from 0 to 1
	fromY/toY	Float from 0 to 1
Rotate	fromDegrees/toDegrees	Float from 0 to 360
	pivotX/pivot Y	String of the percentage of graphic width/height from 0% to 100%



Tween Animations 的通用属性



1.setDuration(long durationMills)

设置动画持续时间（单位：毫秒）

2.setFillAfter(boolean fillAfter)

如果fillAfter的值为true；则动画执行后，控件将停留在执行结束的状态；

3.setFillBefore(boolean fillBefore)

如果fillBefore的值为true；则动画执行或，控件将回到动画执行之前的状态；

4.setStartOffset(long startOffset)

设置动画执行之前的等待时间；

5.setRepeatCount(int repeatCount)

设置动画重复执行的次数；



Tween Animations 的通用属性



1.setDuration(long durationMills)

设置动画持续时间（单位：毫秒）

2.setFillAfter(boolean fillAfter)

如果fillAfter的值为true；则动画执行后，控件将停留在执行结束的状态；

3.setFillBefore(boolean fillBefore)

如果fillBefore的值为true；则动画执行或，控件将回到动画执行之前的状态；

4.setStartOffSet(long startOffSet)

设置动画执行之前的等待时间；

5.setRepeatCount(int repeatCount)

设置动画重复执行的次数；



○ Tweened Animation

● xml 属性

- **duration** 动画持续时间，时间以毫秒为单位。
- **fillAfter** 当设置为 **true**，该动画转化在动画结束后被应用，即动画结束时画面停留在最后一帧。
- **fillBefore** 当设置为 **true**，该动画转化在动画开始前被应用即动画结束时画面停留在第一帧。
- **interpolator** 设置动画渲染器，有 3 中渲染器：
 - **accelerate_decelerate_interpolator** 动画加速减器
 - **accelerate_interpolator** 动画加速器
 - **decelerate_interpolator** 动画减速器



- 1.在res文件夹下面新建一个名为anim的文件夹；
- 2.创建xml文件，并首先加入set标签，改标签如下：

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator">
</set>
```
- 3.在该标签当中加入rotate，alpha，scale或者translate标签；
- 4.在代码当中使用AnimationUtils当中装载xml文件，并生成Animation对象；



Alpha 的 xml 文件编写方法



```
<alpha  
  android:fromAlpha="0.1"  
  android:toAlpha="1.0"  
  android:duration="3000" />
```



Rotate 的 xml 文件编写方法



```
<rotate  
    android:fromDegrees="0"  
    android:toDegrees="+350"  
    android:pivotX="50%"  
    android:pivotY="50%"  
    android:duration="3000" />
```

android:pivotX的值共有三种设置方法：

1. android:pivotX="50"这种方法使用绝对位置定位；
2. android:pivotX="50%"这种方法相对于控件本身定位；
3. android:pivotX="50%p"这种方法相对于控件的父控件定位；



Translate 的 xml 文件编写方法



```
<translate  
    android:fromXDelta="50%"  
    android:toXDelta="100%"  
    android:fromYDelta="0%"  
    android:toYDelta="100%"  
    android:duration="2000" />
```



Scale 的 xml 文件编写方法



```
<scale android:fromXScale="1.0"  
    android:toXScale="0.0"  
    android:fromYScale="1.0"  
    android:toYScale="0.0"  
    android:pivotX="50%"  
    android:pivotY="50%"  
    android:duration="2000" />
```



○ Tweened Animation

● xml 属性

- **repeatCount** 动画的重复次数。
- **repeatMode** 动画重复的行为，默认值是 **restart**，如果设置为 **reverse**，表示偶数次显示动画时会做与动画文件定义的方向相反的动作。
- **startOffset** 动画之间的时间间隔，从上次动画停多少时间开始执行下个动画。
- **zAdjustment** 动画的 **ZOrder** 的改变，**0**：保持 **ZOrder** 不变，**1**：保持在最上层，**-1**：保持在最下层。



Frame Animation

- **Frame Animation 的设计思想是在 onDraw 方法中使用 invalidate() 方法不断刷新 View 的方式实现旋转动画。**
- **逐帧动画是通过 Android 中的 android.graphics.drawable.AnimationDrawable 类来实现。**



什么是 AnimationSet



1. AnimationSet 是 Animation 的子类;
2. 一个 AnimationSet 包含了一系列的 Animation;
3. 针对 AnimationSet 设置一些 Animation 的常见属性 (如 startOffset, duration 等等), 可以被包含在 AnimationSet 当中的 Animation 集成;





什么是 Interpolator



Interpolator定义了动画变化的速率，在Animations框架当中定义了以下几种Interpolator

AccelerateDecelerateInterpolator:在动画开始与结束的地方速率改变比较慢，在中间的时候加速

AccelerateInterpolator:在动画开始的地方速率改变比较慢，然后开始加速

CycleInterpolator:动画循环播放特定的次数，速率改变沿着正弦曲线

DecelerateInterpolator:在动画开始的地方速率改变比较慢，然后开始减速

LinearInterpolator:在动画的以均匀的速率改变



Frame-By-Frame Animations



在res/drawable当中创建一个XML文件，用于定义 Animations的动画序列：

```
<animation-list
xmlns:android="http://schemas.android.com/apk/res/android"
android:oneshot="false">
<item android:drawable="@drawable/nv1"
    android:duration="500" />
<item android:drawable="@drawable/nv2"
    android:duration="500" />
<item android:drawable="@drawable/nv3"
    android:duration="500" />
<item android:drawable="@drawable/nv4"
    android:duration="500" />
</animation-list>
```



Frame-By-Frame Animations



1.为ImageView设置背景资源:

```
imageView.setBackgroundResource(R.drawable.anim_nv);
```

2.通过ImageView得到AnimationDrawable:

```
AnimationDrawable animationDrawable =  
    (AnimationDrawable)imageView.getBackground();
```

3.开始执行动画:

```
animationDrawable.start();
```




什么是 LayoutAnimationController



1. LayoutAnimationController 用于为一个 layout 里面的控件，或者是一个 ViewGroup 里面的控件设置动画效果；
2. 每一个控件都有有相同的动画效果；
3. 这些控件的动画效果在不同的时间现实出来；
4. LayoutAnimationController 可以在 xml 文件当中设置，也可以在代码当中进行设置；



1. 在res/anim文件夹当中创建一个新文件，名为list_anim_layout.xml文件：

```
<layoutAnimation  
xmlns:android="http://schemas.android.com/apk/res/  
android"  
android:delay="0.5"  
android:animationOrder="random"  
android:animation="@anim/list_anim"/>
```

2. 在布局文件当中为ListView添加如下配置：

```
android:layoutAnimation="@anim/list_anim_layout"
```





1. 创建一个Animation对象:

可以通过装载xml文件, 或者直接使用Animation的构造函数创建Animation对象;

2. 使用如下代码创建LayoutAnimationController对象:

```
LayoutAnimationController lac =  
    new LayoutAnimationController(animation);
```

3. 设置控件显示的顺序:

```
lac.setOrder(LayoutAnimationController.ORDER_NOR  
MAL);
```

4. 为ListView设置LayoutAnimationController属性:

```
listView.setLayoutAnimation(lac);
```





什么是 AnimationListener



- 1.AnimationListener是一个监听器;
- 2.该监听器在动画执行的各个阶段会得到通知,从而调用相应的方法;
- 3.主要包含以下的三个方法:
 - 1.onAnimationEnd(Animation animation)
 - 2.onAnimationRepeat(Animation animation)
 - 3.onAnimationStart(Animation animation)