## **Android Animation**

Sang Shin
Michèle Garoche
www.javapassion.com
"Learn with Passion!"



#### Disclaimer

- Portions of this presentation are modifications based on work created and shared by the Android Open Source Project
  - http://code.google.com/policies.html
- They are used according to terms described in the Creative Commons 2.5 Attribution License
  - http://creativecommons.org/licenses/by/2.5/

## **Topics**

- Types of animation
  - > Frame by Frame animation
  - > Tween animation
- Interpolator
- Layout animation

# Types of Animation

## **Two Types of Animation**

- Frame animation (Frame by frame animation)
  - Creates an animation by showing a sequence of images in order
  - Uses AnimationDrawable class
- Tween animation
  - Creates an animation by performing a series of transformations on a single image
  - Uses Animation class
  - More frequently used than Frame animation

## Frame Animation

# **Steps for Doing Frame Animation**

- Create frames (images) in the /res/drawable directory
- Create Animation resource XML file in which the frames are specified as <item> under <animation-list>
- Write Code that
  - Loads the Animation resource XML file into AnimationDrawable object
  - > Starts/stops AnimationDrawable object

## **Animation Resource XML file**

The XML file consists of an <animation-list>
 element as the root node and a series of child
 <item> nodes that each define a frame

```
<?xml version="1.0" encoding="UTF-8"?>
<animation-list
    xmlns:android="http://schemas.android.com/apk/res/android"
    id="selected" android:oneshot="false">
    <item android:drawable="@drawable/ball1" android:duration="50" />
    <item android:drawable="@drawable/ball2" android:duration="50" />
    <item android:drawable="@drawable/ball3" android:duration="50" />
    <item android:drawable="@drawable/ball4" android:duration="50" />
    <item android:drawable="@drawable/ball5" android:duration="50" />
    <item android:drawable="@drawable/ball6" android:duration="50" />
    </animation-list>
```

#### Code That Loads & Starts Frame Animation

```
AnimationDrawable frameAnimation;
public void onCreate(Bundle icicle) {
  super.onCreate(icicle);
  setContentView(R.layout.framebyframe);
  // Load the ImageView that will host the animation and
  // set its background to the AnimationDrawable XML resource.
  ImageView img = (ImageView) findViewById(R.id.my image view);
  // Load animation resource file /res/anim/simple animation.xml
  img.setBackgroundResource(R.anim.simple animation);
  // Get the background, which has been set to an AnimationDrawable object.
  frameAnimation = (AnimationDrawable) img.getBackground();
public boolean onTouchEvent(MotionEvent event) {
  if (event.getAction() == MotionEvent.ACTION DOWN) {
     if (animationStarted) {
       frameAnimation.stop():
       animationStarted = false:
    else{
       frameAnimation.start();
       animationStarted = true;
    return true:
  return super.onTouchEvent(event);
```



## **Transformations in Tween Animation**

- TransitionAnimation
  - Position change
- RotateAnimation
  - > Rotation
- ScaleAnimation
  - > Scaling
- AlphaAnimation
  - > Transparancy

## **Steps of Tween Animation**

- Step #1: Create XML animation resource file in /res/anim/ directory
  - You specify one more more transformations in the resource file
- Step #2: Load the XML animation resource file into Animation object
  - > Animation object captures the transformation logic
- Step #3: Start the animation
  - Select animation target and apply the transformation

# Create Animation Resource File: A single transformation

- The XML file is located under /res/anim/ directory
- Example below is for Rotate transformation

## **Set of transformations**

- Sequential transformation is done with <set>
- Example below Translate transformation followed by Rotate transformation

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:shareInterpolator="false">
   <translate
       android:fromXDelta="0"
       android:toXDelta="100"
       android:fromYDelta="0"
       android:toYDelta="150"
       android:duration="1000" />
    <rotate
       android:interpolator="@android:anim/accelerate decelerate interpolator"
       android:fromDegrees="0"
       android:toDegrees="270" android:toYScale="0.0"
       android:pivotX="50%"
       android:pivotY="50%" android:startOffset="700"
       android:duration="700" />
</set>
```

# Load Animation Resource & Start Animation



# What is Interpolator?

- An interpolator defines the rate of change of an animation.
- This allows the basic transformation (animation) effects (alpha, scale, translate, rotate) to be accelerated, decelerated, repeated, etc.
- You can set an interpolator either in XML animation resource file or programmatically

## **Built-in Android Interpolators**

- android.R.anim.accelerate interpolator
- android.R.anim.decelerate\_interpolator
- android.R.anim.accelerate\_decelerate\_interpolator
- android.R.anim.anticipate interpolator
- android.R.anim.overshoot interpolator

# **Setting Interpolator**

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate interpolator">
  <alpha
    android:fromAlpha="0.0"
    android:toAlpha="1.0"
    android:duration="100"/>
  <scale
    android:fromXScale="0.5" android:toXScale="1.5"
    android:fromYScale="0.5" android:toYScale="1.5"
    android:pivotX="50%" android:pivotY="50%"
    android:duration="200" />
  <scale
    android:fromXScale="1.5" android:toXScale="1.0"
    android:fromYScale="1.5" android:toYScale="1.0"
    android:pivotX="50%" android:pivotY="50%"
    android:startOffset="200"
    android:duration="100"/>
</set>
```

## **Setting Interpolator Programmatically**

# Layout Animation

## What is Layout Animation?

- Animations are applied when components are being laid out
  - When components are added or removed from layouts, these animations are triggered.

## **Steps for Layout Animation**

- 1.Create "layout animation" resource files under /res/anim/ directory
- 2.Add android:layoutAnimation=@anim/<layoutanimation-resource-file> attribute to the layout
- Load the layout resource file as you normally do

## 1. Create Animation XML Resources

 Create it under /res/anim/ directory - let's call it /res/anim/layout\_bottom\_to\_top\_slide.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layoutAnimation
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:delay="500%"
    android:animationOrder="reverse"
    android:animation="@anim/slide_right" />
```

The above uses /res/anim/slide\_right.xml below

# 2. Create Layout Resource File

 Layout resource file now has android:layoutAnimation attribute

```
<ListView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/list"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layoutAnimation="@anim/layout_bottom_to_top_slide" />
```

## 3. Load Layout resource

 There is nothing different in this code from the normal way of loading layout resource

# Thank you!



Check JavaPassion.com Codecamps!
http://www.javapassion.com/codecamps
"Learn with Passion!"