Droidikin A good place for Android™ developers

ImageSwitcher with auto switch code

Posted by pasquale on May 20, 2012 Go to comments Leave a comment (1)

This tutorial describe how to fading\sliding(or personalize the animation) images contained in the *drawable* folder using an *ImageSwitcher* and a *Runnable* to switch the images automatically.
ImageSwitcher is used to switch between images by playing an animation during the transition, this post describe how to do that using the java code or creating the animations in the *res/anim* resource folder.

The code

The *Activity* xml layout:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
```

RSS Feed

Search: type, hit enter SEARCH

Recent Posts

- Android Emulator List
- Create the Samsung Galaxy s2 Emulator
- Could not find class 'com.google.android.imageloader.ImageLoader' referenced from method…
- ImageSwitcher with auto switch code

Categories

- Errors
- Snippets
- Tutorials

Meta

- Log in
- Entries RSS
- Comments RSS
- WordPress.org

Tags

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        <ImageSwitcher android:id="@+id/imageswitcher"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:layout_margin="10dp"
           />
</LinearLayout>
```

Alpha Animation AVD Eclipse Emulator Galaxy Handler ImageLoader ImageSw itcher ImageView jar Library Libs Rotate Running S2 Samsung SDK Skin Slide

The *Activity* java code:

```
package com.droidikin.samples.imageswitcher;

import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.view.ViewGroup.LayoutParams;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageSwitcher;
import android.widget.ImageView;
import android.widget.ViewSwitcher.ViewFactory;

public class CodeActivity extends Activity implements ViewFactory {

        // A reference to the images contained into the drawable folder
        private Integer[] m_ImageIds = {
            R.drawable.img_01,
            R.drawable.img_02,
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```java
                R.drawable.img_03
            };

            // The ImageSwitcher
            ImageSwitcher m_ImageSwitcher;

            // The Handler used for manage the Runnable that switch the images
            Handler m_Handler = new Handler();

            // The index of the current image
            int m_imageIndex = 0;

        /** Called when the activity is first created. */
        @Override
        public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.use_code_activity);
            // assign the ImageSwitcher
            m_ImageSwitcher = (ImageSwitcher)findViewById(R.id.imageswitcher);
            m_ImageSwitcher.setFactory(this);
            // Create the Fade in animation
            Animation fadeIn = AnimationUtils.loadAnimation(this, android.R.ani
            fadeIn.setDuration(3000);
            // Create the Fade out animation
            Animation fadeOut = AnimationUtils.loadAnimation(this, android.R.an
            fadeOut.setDuration(3000);
            // Assign the two animations to the ImageSwitcher
            m_ImageSwitcher.setInAnimation(fadeIn);
            m_ImageSwitcher.setOutAnimation(fadeOut);

            // Start the Runnable
            m_Handler.post(m_UpdateTimeTask);
        }

            @Override
            public View makeView() {
                    ImageView i = new ImageView(this);
                    i.setBackgroundColor(0xFF000000);
                    i.setScaleType(ImageView.ScaleType.FIT_CENTER);
                    i.setLayoutParams(new ImageSwitcher.LayoutParams(LayoutPara
                    return i;
            }

            /*
             * Set the image to show into the ImageSwitcher, then post his exec
             */
            Runnable m_UpdateTimeTask = new Runnable() {
                        public void run() {
                                // Set the Image to show
                                m_ImageSwitcher.setImageResource(m_ImageIds[m_im
                                // Increment the index
                                m_imageIndex++;
                                // if necessary restart from the first image
                                if(m_imageIndex > (m_ImageIds.length-1)){
                                        m_imageIndex = 0;
                                }
                                // Set the execution after 5 seconds
                                m_Handler.postDelayed(this, (5 * 1000));
                        }
            };
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
80.
81.   }
```

In the code above you can see highlighted the lines from 40 to 48, where are defined and set to the *ImageSwitcher* the two animations.
Top

Create the animations using the *res/anim* resource folder files

Like the code example above, we must define two animations and assign they to the *ImageSwitcher*, but in this case we do that into the layout Xml file of our *Activity*:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
     android:layout_width="fill_parent"
     android:layout_height="fill_parent"
     android:orientation="vertical" >
        <ImageSwitcher android:id="@+id/imageswitcher"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:layout_margin="10dp"
              android:inAnimation="@android:anim/fade_in"
              android:outAnimation="@android:anim/fade_out"
           />
</LinearLayout>
```

It works!, but the animation duration is the default of the inherit animation, then, if we need a specific duration, we must define our own animation.

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Below an example of a Fade In animation with a duration of 3 seconds.

```xml
<?xml version="1.0" encoding="utf-8"?>
<alpha xmlns:android="http://schemas.android.com/apk/res/android"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:duration="3000"  />
```

Then in the *Activity* layout xml, change the line 10 and 11:

```xml
 1. <?xml version="1.0" encoding="utf-8"?>
 2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
 3.      android:layout_width="fill_parent"
 4.      android:layout_height="fill_parent"
 5.      android:orientation="vertical" >
 6.         <ImageSwitcher android:id="@+id/imageswitcher"
 7.               android:layout_width="wrap_content"
 8.               android:layout_height="wrap_content"
 9.               android:layout_margin="10dp"
10.               android:inAnimation="@anim/fade_in"
11.               android:outAnimation="@anim/fade_out"
12.            />
13. </LinearLayout>
```

Where *@anim/fade_in* and *@anim/fade_out* are the names of the created animations.

Top

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Use a different animation

Into the default animations list is contained also the sliding animation, then we can set it directly in our *Activity* layout xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
     android:layout_width="fill_parent"
     android:layout_height="fill_parent"
     android:orientation="vertical" >
        <ImageSwitcher android:id="@+id/imageswitcher"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:layout_margin="10dp"
              android:inAnimation="@android:anim/slide_in_left"
              android:outAnimation="@android:anim/slide_out_right"
           />
</LinearLayout>
```

But if we want to completely manage the animation behavior we can create our animation into the *res/anim* folder, for example changing the duration of the sliding animation:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
      android:interpolator="@android:anim/accelerate_interpolator"
     >
    <translate
```

```
            android:fromXDelta="0"
            android:toXDelta="100%p"
           android:duration="2000" />
</set>
```

Or creating a rotate animation:

```
<?xml version="1.0" encoding="utf-8"?>
<rotate   xmlns:android="http://schemas.android.com/apk/res/android"
   android:fromDegrees="0"
   android:interpolator="@android:anim/linear_interpolator"
   android:toDegrees="360"
   android:pivotX="50%"
   android:pivotY="50%"
   android:duration="1000"
   android:startOffset="0" />
```

Or a rotation with alpha effect animation:

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

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
            android:duration="3000"
            android:startOffset="0"
          />
          <alpha
                android:interpolator="@android:anim/accelerate_interpolator"
                android:fromAlpha="0.0"
                android:toAlpha="1.0"
                android:duration="3000"
          />
</set>
```

We can create infinitely animations(also via code, but I prefer the xml way: is reusable and easy to switch), the limits are our imagination and competences, and probably the time 😉

Tutorials   Alpha, Animation, Handler, ImageSwitcher, ImageView, Rotate, Running, Slide

Could not find class 'com.google.android.imageloader.ImageLoader', referenced from method… →

Leave a comment ?   1 Comments.

Hanson January 10, 2013 at 2:49 pm   Reply

😃 Thanks this is what i needed

Leave a Comment

open in browser PRO version   Are you a developer? Try out the HTML to PDF API   pdfcrowd.com

	NAME
	EMAIL
	Website URL

NOTE - You can use these HTML tags and attributes:
`<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <strike> <strong>`

SUBMIT

Copyright © 2013 Droidikin | Powered by WordPress and zBench

↑ Top

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com