ImageSwitcher

[Базовый пример](http://developer.alexanderklimov.ru/android/views/imageswitcher.php#base)  
[Анимация затемнения](http://developer.alexanderklimov.ru/android/views/imageswitcher.php#animation)  
[Динамическая смена картинок](http://developer.alexanderklimov.ru/android/views/imageswitcher.php#dynamic)  
[Управление жестами](http://developer.alexanderklimov.ru/android/views/imageswitcher.php#gesture)  
[Слайд-шоу](http://developer.alexanderklimov.ru/android/views/imageswitcher.php#slide)

Компонент **ImageSwitcher** наследуется от класса **ViewSwitcher** и позволяет заменять текущий элемент **View** на другой элемент **View**. Самый распространённый вариант - использование**ImageView**. Одновременно на экране может отображаться только один элемент **View**. Во время перехода между элементами **View** могут использоваться анимации (плавные появления/исчезновения и вращения). В студии находится в разделе **Expert**.

Базовый пример

Дочерние элементы **View** создаются при помощи экземпляра класса **ViewFactory**. Например, элемент **ImageSwitcher** и его соответствующий экземпляр класса **ViewFactory** могут быть использованы для генерации элемента **ImageView** текущего вопроса викторины и для последующего переключения на элемент **ImageView** следующего вопроса, когда пользователь щёлкнет по кнопке **Вперёд** или **Назад**.

Создадим новый проект и добавим элементы в разметку.

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<ImageSwitcher

android:id="@+id/imageSwitcher"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="100dp"

android:onClick="onSwitcherClick">

<ImageView

android:id="@+id/imageView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/cat\_cage"/>

<ImageView

android:id="@+id/imageView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/cat\_tied"/>

</ImageSwitcher>

</LinearLayout>

Обратите внимание, что элементы **ImageView** являются дочерними и вложены в **ImageSwitcher**. Также у атрибута **onClick** задан обработчик **onSwitcherClick**. Осталось написать код:

package ru.alexanderklimov.imageswitcherdemo;

import ...

public class MainActivity extends Activity {

private ImageSwitcher mImageSwitcher;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mImageSwitcher = (ImageSwitcher)findViewById(R.id.imageSwitcher);

}

public void onSwitcherClick(View view){

mImageSwitcher.showNext();

}

}

Метод **showNext()** выводит следующий элемент. Вывод происходит циклически в одном направлении. Если у вас используются две картинки, то они будут по очереди сменять друг друга. Для смены изображения в обратную сторону используется метод **showPrevious()**.

Анимация затемнения

Мы рассмотрели простейший пример смены одного изображения на другое. Причём, смена происходит сразу, без задержки. Чтобы смена изображений происходила красиво, нужно подключить анимацию.

mImageSwitcher = (ImageSwitcher)findViewById(R.id.imageSwitcher);

Animation inAnimation = new AlphaAnimation(0, 1);

inAnimation.setDuration(2000);

Animation outAnimation = new AlphaAnimation(1, 0);

outAnimation.setDuration(2000);

mImageSwitcher.setInAnimation(inAnimation);

mImageSwitcher.setOutAnimation(outAnimation);

Добавив код к предыдущему примеру, вы получите плавную анимацию с использованием прозрачности. Первая картинка будет плавно исчезать, уступая второму изображению.

Динамическая смена картинок

Рассмотрим более продвинутый пример. Подготовьте несколько картинок в папке **res/drawable**. На экран активности добавим две кнопки для просмотра картинок **Вперёд** и **Назад**. На этот раз мы уже не используем дочерние элементы **ImageView**. Картинки будут загружаться динамически.

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<Button

android:id="@+id/buttonPrev"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:onClick="onClick"

android:text="&lt;&lt;"/>

<Button

android:id="@+id/buttonForward"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:onClick="onClick"

android:text=">>"/>

</LinearLayout>

<ImageSwitcher

android:id="@+id/imageSwitcher"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal">

</ImageSwitcher>

</LinearLayout>

Напишем код для управления галереей.

package ru.alexanderklimov.imageswitcherdemo;

import ...

public class MainActivity extends AppCompatActivity implements ViewSwitcher.ViewFactory {

private ImageSwitcher mImageSwitcher;

int position = 0;

private int[] mImageIds = { R.drawable.cat1, R.drawable.cat2,

R.drawable.cat3, R.drawable.cat4, R.drawable.cat5,

R.drawable.cat6, R.drawable.cat7, };

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mImageSwitcher = (ImageSwitcher)findViewById(R.id.imageSwitcher);

mImageSwitcher.setFactory(this);

Animation inAnimation = new AlphaAnimation(0, 1);

inAnimation.setDuration(2000);

Animation outAnimation = new AlphaAnimation(1, 0);

outAnimation.setDuration(2000);

mImageSwitcher.setInAnimation(inAnimation);

mImageSwitcher.setOutAnimation(outAnimation);

mImageSwitcher.setImageResource(mImageIds[0]);

}

public void onClick(View v) {

switch (v.getId()) {

case R.id.buttonForward:

setPositionNext();

mImageSwitcher.setImageResource(mImageIds[position]);

break;

case R.id.buttonPrev:

setPositionPrev();

mImageSwitcher.setImageResource(mImageIds[position]);

break;

default:

break;

}

}

public void setPositionNext() {

position++;

if (position > mImageIds.length - 1) {

position = 0;

}

}

public void setPositionPrev() {

position--;

if (position < 0) {

position = mImageIds.length - 1;

}

}

@Override

public View makeView() {

ImageView imageView = new ImageView(this);

imageView.setScaleType(ImageView.ScaleType.FIT\_CENTER);

imageView.setLayoutParams(new

ImageSwitcher.LayoutParams(

LinearLayout.LayoutParams.MATCH\_PARENT, LinearLayout.LayoutParams.MATCH\_PARENT));

imageView.setBackgroundColor(0xFF000000);

return imageView;

}

}

Вначале мы прописали массив графических ресурсов, а также определили переменную *position* для определения текущей позиции.

Далее кроме знакомых нам методов для анимации, вызываем методы **setFactory()** и **setImageResource()**.

Метод **setFactory()** помогает указать подходящий класс **ViewFactory**, позволяющий менять изображения по очереди. Класс **ViewFactory** имеет всего один обязательный метод — метод**makeView()**. Этот метод должен возвращать **View** подходящего типа, в нашем примере определённым образом настроенный элемент **ImageView**.

Метод **setImageResource()** загружает изображение из ресурсов.

Вспомогательные методы **setPositionNext()** и **setPositionPrev()** позволяют установить значение для текущей позиции, которое используется в обработчиках нажатий кнопок.

Управление жестами

Если вы хотите управлять сменой картинок не через кнопки **Вперёд/Назад**, а движением пальца, то нужно подключить класс **GestureDetector**. Дополним предыдущий пример.

package ru.alexanderklimov.imageswitcherdemo;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.GestureDetector;

import android.view.MotionEvent;

import android.view.View;

import android.view.animation.AlphaAnimation;

import android.view.animation.Animation;

import android.widget.ImageSwitcher;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.ViewSwitcher;

public class MainActivity extends AppCompatActivity implements ViewSwitcher.ViewFactory, GestureDetector.OnGestureListener {

private ImageSwitcher mImageSwitcher;

int position = 0;

private int[] mImageIds = { R.drawable.cat1, R.drawable.cat2,

R.drawable.cat3, R.drawable.cat4, R.drawable.cat5,

R.drawable.cat6, R.drawable.cat7, };

private GestureDetector mGestureDetector;

private static final int SWIPE\_MIN\_DISTANCE = 120;

private static final int SWIPE\_MAX\_OFF\_PATH = 250;

private static final int SWIPE\_THRESHOLD\_VELOCITY = 100;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mImageSwitcher = (ImageSwitcher)findViewById(R.id.imageSwitcher);

mImageSwitcher.setFactory(this);

Animation inAnimation = new AlphaAnimation(0, 1);

inAnimation.setDuration(2000);

Animation outAnimation = new AlphaAnimation(1, 0);

outAnimation.setDuration(2000);

mImageSwitcher.setInAnimation(inAnimation);

mImageSwitcher.setOutAnimation(outAnimation);

mImageSwitcher.setImageResource(mImageIds[0]);

mGestureDetector = new GestureDetector(this, this);

}

public void onClick(View v) {

switch (v.getId()) {

case R.id.buttonForward:

setPositionNext();

mImageSwitcher.setImageResource(mImageIds[position]);

break;

case R.id.buttonPrev:

setPositionPrev();

mImageSwitcher.setImageResource(mImageIds[position]);

break;

default:

break;

}

}

public void setPositionNext() {

position++;

if (position > mImageIds.length - 1) {

position = 0;

}

}

public void setPositionPrev() {

position--;

if (position < 0) {

position = mImageIds.length - 1;

}

}

@Override

public View makeView() {

ImageView imageView = new ImageView(this);

imageView.setScaleType(ImageView.ScaleType.FIT\_CENTER);

imageView.setLayoutParams(new

ImageSwitcher.LayoutParams(

LinearLayout.LayoutParams.MATCH\_PARENT, LinearLayout.LayoutParams.MATCH\_PARENT));

imageView.setBackgroundColor(0xFF000000);

return imageView;

}

@Override

public boolean onTouchEvent(MotionEvent event) {

return mGestureDetector.onTouchEvent(event);

}

@Override

public boolean onDown(MotionEvent e) {

return false;

}

@Override

public void onShowPress(MotionEvent e) {

}

@Override

public boolean onSingleTapUp(MotionEvent e) {

return false;

}

@Override

public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {

return false;

}

@Override

public void onLongPress(MotionEvent e) {

}

@Override

public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {

try {

if (Math.abs(e1.getY() - e2.getY()) > SWIPE\_MAX\_OFF\_PATH)

return false;

// справа налево

if (e1.getX() - e2.getX() > SWIPE\_MIN\_DISTANCE

&& Math.abs(velocityX) > SWIPE\_THRESHOLD\_VELOCITY) {

setPositionNext();

mImageSwitcher.setImageResource(mImageIds[position]);

} else if (e2.getX() - e1.getX() > SWIPE\_MIN\_DISTANCE

&& Math.abs(velocityX) > SWIPE\_THRESHOLD\_VELOCITY) {

// слева направо

setPositionPrev();

mImageSwitcher.setImageResource(mImageIds[position]);

}

} catch (Exception e) {

// nothing

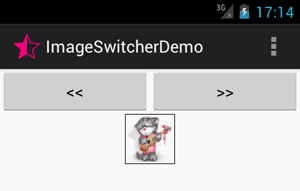
return true;

}

return true;

}

}



Слайд-шоу

По такому же принципу можно устроить анимацию в виде слайд-шоу.

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity">

<Button

android:id="@+id/button"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:onClick="onClick"

android:text="Далее"/>

<ImageSwitcher

android:id="@+id/imageSwitcher"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"/>

</LinearLayout>

Код. Не забудьте подготовить несколько картинок в ресурсах.

package ru.alexanderklimov.imageswitcherdemo;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

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.LinearLayout;

import android.widget.ViewSwitcher;

public class MainActivity extends AppCompatActivity {

private ImageSwitcher mImageSwitcher;

// массив картинок

private int[] mImageIds = {R.drawable.cat1, R.drawable.cat2, R.drawable.cat3,

R.drawable.cat4, R.drawable.cat5};

private int mCurIndex;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mImageSwitcher = (ImageSwitcher) findViewById(R.id.imageSwitcher);

Animation slideInLeftAnimation = AnimationUtils.loadAnimation(this,

android.R.anim.slide\_in\_left);

Animation slideOutRight = AnimationUtils.loadAnimation(this,

android.R.anim.slide\_out\_right);

mImageSwitcher.setInAnimation(slideInLeftAnimation);

mImageSwitcher.setOutAnimation(slideOutRight);

mImageSwitcher.setFactory(new ViewSwitcher.ViewFactory() {

@Override

public View makeView() {

ImageView imageView = new ImageView(MainActivity.this);

imageView.setScaleType(ImageView.ScaleType.FIT\_CENTER);

LayoutParams params = new ImageSwitcher.LayoutParams(

LinearLayout.LayoutParams.MATCH\_PARENT, LinearLayout.LayoutParams.MATCH\_PARENT);

imageView.setLayoutParams(params);

return imageView;

}

});

mCurIndex = 0;

mImageSwitcher.setImageResource(mImageIds[mCurIndex]);

}

public void onClick(View v) {

if (mCurIndex == mImageIds.length - 1) {

mCurIndex = 0;

mImageSwitcher.setImageResource(mImageIds[mCurIndex]);

} else {

mImageSwitcher.setImageResource(mImageIds[++mCurIndex]);

}

}

}

Реклама