TextSwitcher

Компонент **TextSwitcher** находится в разделе **Transitions** и предназначен для анимированной смены текста.

Если использовать обычный **TextView** и выводить в нём новые сообщения, то текст будет выводиться мгновенно - предыдущий текст исчезает, а новый появляется вместо него. Но, если вам нужно сделать красиво, то **TextSwitcher** поможет вам.

Общий принцип использования следующий - с помощью интерфейса **ViewFactory** и его метода **makeView()** создаются дочерние элементы **TextView**, которые переключаются между собой с анимацией. В методе **setFactory()** указываем реализованный интерфейс. В завершение указываем два типа анимации в метода **setInAnimation()** и **setOutAnimation()**.

Разместим на форме кнопку и **TextSwitcher**:

<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="wrap\_content"

android:layout\_height="wrap\_content"

android:onClick="OnClick"

android:text="Начать счёт" />

<TextSwitcher

android:id="@+id/textSwitcher"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

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

</LinearLayout>

Теперь переходим к коду:

package ru.alexanderklimov.testapplication;

import android.graphics.Color;

import android.os.Bundle;

import android.support.v7.app.ActionBarActivity;

import android.view.Gravity;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.TextSwitcher;

import android.widget.TextView;

public class MainActivity extends ActionBarActivity implements TextSwitcher.ViewFactory {

private TextSwitcher mTextSwitcher;

private int mCounter = 0;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mTextSwitcher = (TextSwitcher) findViewById(R.id.textSwitcher);

mTextSwitcher.setFactory(this);

Animation inAnimation = AnimationUtils.loadAnimation(this,

android.R.anim.fade\_in);

Animation outAnimation = AnimationUtils.loadAnimation(this,

android.R.anim.fade\_out);

mTextSwitcher.setInAnimation(inAnimation);

mTextSwitcher.setOutAnimation(outAnimation);

updateCounter();

}

public void onClick(View v) {

mCounter++;

updateCounter();

}

private void updateCounter() {

mTextSwitcher.setText(String.valueOf(mCounter));

}

@Override

public View makeView() {

TextView textView = new TextView(this);

textView.setGravity(Gravity.CENTER | Gravity.CENTER\_HORIZONTAL);

textView.setTextSize(70);

textView.setTextColor(Color.RED);

return textView;

}

}

Данный пример немного похож на [пример подсчета ворон](http://developer.alexanderklimov.ru/android/android3.php), только на этот раз результаты выводим не в текстовом поле, а в **TextSwitcher**. На скриншоте этого не увидеть, поэтому нужно попробовать пример на эмуляторе или устройстве. Если приглядеться, то можно увидеть у цифры 6 проступающую следующую цифру 7.



На что обратить внимание - мы используем системные анимации **android.R.anim.fade\_in** и **android.R.anim.fade\_out**. Поэтому нам не нужно создавать файлы XML для анимации и сохранять их в **res/anim**, как мы это делали в некоторых уроках. Метод **setFactory** позволяет указать нужный класс. Метод **makeView()** должен возвращать элемент **View** подходящего типа, в нашем случае **TextView**. Заметьте, что в методе мы формируем своеобразный шаблон (размер, цвет, позиция) для **TextView**, а вывод содержимого формируется в других методах.