Делаем анимированную кнопку с помощью ImageView

|  |
| --- |
|  |

Кнопки являются самым часто используемым элементом любого Android приложения. Сегодня мы немного расширим свои способности в области создания кнопок и сделаем анимированную кнопку.

Создаем новый проект, выбираем Blank Activity. Основные шаги создания будут следующими:

1) создаем интерфейс программы, состоящий всего то из одного элемента ImageView, который и будет играть роль кнопки;

2) создаем анимацию из набора изображений;

3) ну и напоследок кодим в MainActivity.java.

Начнем с работы с интерфейсом. Открываем файл **activity\_main.xml** и добавляем туда элемент **ImageView**, задав ему определенные рамки, чтоб он в итоге не занимал нам весь экран:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/activity\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="#FFFFFF"  
 android:weightSum="1"**>  
  
  
 <**ImageView  
 android:clickable="true"  
 android:id="@+id/imageView"  
 android:onClick="animButtonClick"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="172dp"  
 android:layout\_gravity="center"  
 android:layout\_marginLeft="50pt"  
 android:layout\_marginTop="50pt"  
 android:layout\_marginBottom="50pt"  
 android:layout\_marginRight="50pt"  
 android:background="@drawable/anim\_button"  
 android:layout\_weight="0.06"** />  
  
  
</**LinearLayout**>

Пока что Android Studio ругает нас на строчке  android:background="@drawable/anim\_button", чтобы это исправить, переходим к выполнению второго пункта плана: создание анимации. Как вы наверное уже догадались, анимация в приложении будет представлена xml файлом anim\_button.xml. Перед тем, как его создать, добавим в проект приложения несколько изображений, которые и будут составлять анимацию. Я не слишком крутой дизайнер (мягко говоря), поэтому если вам не понравиться мой набор [картинок](http://learn-android.ru/Files/ImageAnimation.rar), можете подобрать что-нибудь свое. Изображения нужно поместить в папке res/drawable проекта.

Теперь в этой же папке **res/drawable** создаем новый xml файл по имени **anim\_button.xml**и добавим в него наши картинки:

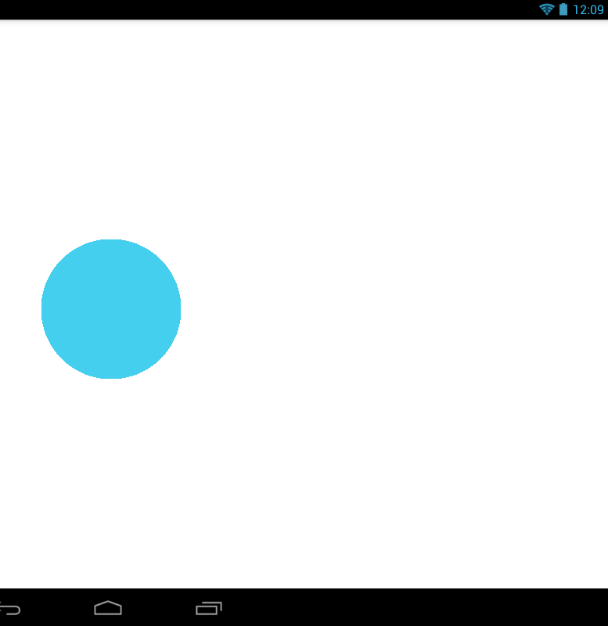
<**animation-list xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/loading"  
 android:oneshot="true"** >  
 <**item android:drawable="@drawable/ba1" android:duration="10"**/>  
 <**item android:drawable="@drawable/ba2" android:duration="10"**/>  
 <**item android:drawable="@drawable/ba3" android:duration="10"**/>  
 <**item android:drawable="@drawable/ba4" android:duration="10"**/>  
 <**item android:drawable="@drawable/ba3" android:duration="10"**/>  
 <**item android:drawable="@drawable/ba2" android:duration="10"**/>  
 <**item android:drawable="@drawable/ba1" android:duration="10"**/>  
  
</**animation-list**>

Команда **duration**регулирует продолжительность показа одного изображения в миллисекундах, можете подобрать оптимальное для вас значение.

Переходим к работе в файле **MainActivity.java**. Нам нужно объявить и инициализировать переменные ImageView и AnimationDrawable, настроить файл anim\_button.xml в качестве анимации для элемента ImageView и настроить запуск самой анимации при нажатии на элемент ImageView. Чтобы все это сделать, добавляем следующий код:

**import** android.app.Activity;  
**import** android.graphics.drawable.AnimationDrawable;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.ImageView;  
  
**public class** MainActivity **extends** Activity {  
  
 *//Объявляем использование ImageView* ImageView **anim\_image**;  
 *//Объявляем использование анимации AnimationDrawable* AnimationDrawable animation;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 *//Связываем объект ImageView с нашим элементом ImageView  
 //и задаем ему на фон созданную анимацию:* **anim\_image** = (ImageView) findViewById(R.id.***imageView***);  
 **anim\_image**.setBackgroundResource(R.drawable.***anim\_button***);  
  
 *//Загружаем объект анимации:* animation = (AnimationDrawable)**anim\_image**.getBackground();  
  
 *//Выставляя значение false, добиваемся бесконечного  
 //повторения анимации (true - только 1 повторение):* animation.setOneShot(**true**);}  
  
  
 *//Метод обработки нажатия на ImageView:* **public void** animButtonClick(View v) {  
 Log.*d*(**"animButton"**, **"Click"**);  
 animation.stop();  
 animation.start();  
  
 }  
}

Ну вот и все на этот раз, запускаем и смотрим что получилось:



Кнопочка с ImageView работает нормально. Чтобы анимация повторилась 1 раз, в строке   
**animation.setOneShot(false)** нужно выставить значение **true**.

Наша анимация не сильно красива, поэтому сложно увидеть ей применение, но если создать красивую анимацию, то использование такого элемента в интерфейсе программы может оказаться очень полезным и приятным. Кстати, похожую работу мы выполняли в уроке по созданию фреймовой [анимации](http://learn-android.ru/news/sozdanie_prostoj_frejmovoj_animacii_v_android_prilozhenii/2015-01-18-24.html).