Создание простой фреймовой анимации в Android приложении

|  |
| --- |
| [Главная](http://learn-android.ru/index.html) » Создание простой фреймовой анимации в Android приложении |
| С помощью стандартных возможностей Android Studio можно создавать довольно симпатичную анимацию. Что интересно, есть несколько способов создания анимации, возможности не ограничиваются каким то один методом. Один из таких способов был показан в уроке [создания перехода](http://learn-android.ru/news/osnovy_raboty_s_perekhodami/2015-01-14-13.html), там мы получили красивую анимацию шаров с помощью перехода.  На этот раз рассмотрим еще один интересный способ создания эффекта анимации. Этот способ напоминает нам сам принцип эффекта анимации - есть набор картинок, кадров, которые имеют похожие изображения с малыми отличиями между ними:  http://developeroleg.ucoz.ru/Images/Studies/anim.jpg  Так ведь делаются и мультфильмы, и фильмы. Благодаря быстрой прокрутке (показу) поочередно таких изображений и возникает эффект анимации. Чем больше таких мало отличающихся картинок, чем меньше шаг между изменениями, так скажем, то тем плавнее и качественнее выглядит анимация, видео. Вот и говорят - качество видео сколько то кадров в секунду.  Вот именно такой эффект мы и будем использовать. Неплохо бы сделать анимацию с теми рожицами выше, ну да ладно, сделаем, как обычно, с шариками.  Итак, создаем новый проект, выбираем **Blank Activity**. Начнем с заготовлению шариков для анимации. В папке приложения **drawable**создаем новый файл **circle1.xml**. Создаем в этом файле кружок, задаем его цвет и размер:  <shape xmlns:android="http://schemas.android.com/apk/res/android"  android:shape="oval"  android:dither="true">  <gradient  android:startColor="#FFA07A"  android:endColor="#FFA07A"  android:type="linear"/>  <size  android:height="150dp"  android:width="150dp"/>  </shape>  Теперь создадим таким же образом еще, ну 5 штук шариков, которые будут использоваться для создания анимации. Создаем файлы **circle2.xml**,**circle3.xml**, **circle4.xml**, **circle5.xml**,**circle6.xml**в той же папке **drawable**. Разница между шариками будет только в их цвете. Нет смысла приводить 6 экземпляров одинакового кода, разница, как уже сказано, только в цвете:  **circle2.xml**:  android:startColor="#FF7F50"  android:endColor="#FF7F50"  **circle3.xml**:  android:startColor="#FF6347"  android:endColor="#FF6347"  **circle4.xml**:  android:startColor="#FF4500"  android:endColor="#FF4500"  **circle5.xml**:  android:startColor="#FF8C00"  android:endColor="#FF8C00"  **circle6.xml**:  android:startColor="#FFA500"  android:endColor="#FFA500"  Вот так, с разноцветными шариками закончили. Теперь с этих с xml файлов с изображениями нужно сделать анимацию. Делается это так. В этой же папке **drawable**создаем еще один файл **anim.xml**. В нем нам нужно создать такую штуку, как **<animation-list>** и наполнить ее нашими изображениями:  <animation-list  android:oneshot="false"  xmlns:android="http://schemas.android.com/apk/res/android">  <item  android:drawable="@drawable/circle1"  android:duration="100"/>  <item  android:drawable="@drawable/circle2"  android:duration="100"/>  <item  android:drawable="@drawable/circle3"  android:duration="100"/>  <item  android:drawable="@drawable/circle4"  android:duration="100"/>  <item  android:drawable="@drawable/circle5"  android:duration="100"/>  <item  android:drawable="@drawable/circle6"  android:duration="100"/>  </animation-list>  Строка **android:oneshot="false"** настраивает количество повторов анимации. Если стоит значение **"true"** то анимация повториться только один раз, если **"false"** - то выполняется бесконечный повтор. Строка **duration**означает продолжительность показа каждого объекта, у нас выставлено 100 мс = 0,1 секунды.  Переходим к редактированию внешнего вида нашего приложения. Открываем файл **activity\_main.xml**. Создаем здесь всего один элемент - **ImageView**, который и будет изображать нашу анимацию:  <**RelativeLayout** 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"  tools:context=".MainActivity"  android:id="@+id/bg">  <**ImageView**  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/an"  android:layout\_centerInParent="true"/>  </**RelativeLayout**>  Остается выполнить последний штрих, открываем файл **MainActivity.java**. Добавим к строкам импорта парочку таких:  import **android.graphics.drawable.AnimationDrawable**;  import **android.widget.ImageView**;  В методе onCreate создадим ссылку на наш элемент в **activity\_main.xml**, настроим анимацию **anim.xml**ему в качестве фона, создадим саму анимацию и запустим ее:  import **android.app.Activity**;  import **android.graphics.drawable.AnimationDrawable**;  import **android.os.Bundle**;  import **android.widget.ImageView**;  **public** **class** MainActivity **extends** Activity {  **@Override**  **protected** **void** onCreate(**Bundle** *savedInstanceState*) {  super.onCreate(savedInstanceState);  setContentView(**R**.layout.activity\_main);  *//Ссылаемся на созданный ImageView:*  **ImageView** mAnimation = (**ImageView**)findViewById(**R**.id.an);  *//Настраиваем ему нашу анимацию в качестве фона:*  mAnimation.setBackgroundResource(**R**.drawable.anim);  *//Создаем саму анимацию:*  **AnimationDrawable** animation = (**AnimationDrawable**)mAnimation.getBackground();  *//Ну и запускаем анимацию:*  animation.start();  }  }  Вот и все, анимация готова! Мы создали фреймовую анимацию и использовали ее в нашем приложении. Ничего сложного, но с помощью этого метода можно делать неплохие, красивые вещи. Кстати, можете сделать так же, но с использованием тех синих рожиц вверху, думаю, будет очень прикольно.  **Тестируйте на устройстве/эмуляторе с Android 4.2+**, на версии ОС 2.3 эта анимация у меня не заработала, на планшете с Android 4.2 - заработало. |