**Анимации между активностями**

[Анимация при старте активности](http://developer.alexanderklimov.ru/android/animation/overridependingtransition.php#oncreate)  
[Метод overridePendingTransition()](http://developer.alexanderklimov.ru/android/animation/overridependingtransition.php#overridependingtransition)  
[Анимация при помощи XML](http://developer.alexanderklimov.ru/android/animation/overridependingtransition.php#xmlanimation)

**Анимация при старте активности**

Прежде чем обсуждать, как создавать анимацию между активностями, рассмотрим пример создания анимации для самой первой активности, которая загружается при запуске приложения.

Создадим файл анимации **rotate.xml** в папке **res/anim/**:

<?xml version="1.0" encoding="utf-8"?>

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

android:interpolator="@android:anim/linear\_interpolator" >

<rotate

android:duration="1000"

android:fromDegrees="0"

android:pivotX="50%"

android:pivotY="50%"

android:repeatCount="1"

android:repeatMode="reverse"

android:startOffset="0"

android:toDegrees="360" />

</set>

Добавим в активность два ImageView:

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

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

android:id="@+id/screen"

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=".TestActivity" >

<ImageView

android:id="@+id/icon"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/ic\_launcher" />

<ImageView

android:id="@+id/bigicon"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:src="@drawable/ic\_launcher" />

</LinearLayout>

Нам нужно запустить анимацию в методе onCreate() и, возможно, в onResume():

package ru.alexanderklimov.startanimation;

import android.os.Bundle;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.ImageView;

import android.app.Activity;

public class MainActivity extends Activity {

ImageView bigIcon;

ImageView icon;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

bigIcon = (ImageView) findViewById(R.id.bigicon);

icon = (ImageView) findViewById(R.id.icon);

Animation animRotateIn\_icon = AnimationUtils.loadAnimation(this,

R.anim.rotate);

icon.startAnimation(animRotateIn\_icon);

}

@Override

protected void onResume() {

super.onResume();

Animation animRotateIn\_big = AnimationUtils.loadAnimation(this,

R.anim.rotate);

bigIcon.startAnimation(animRotateIn\_big);

}

}

**Метод overridePendingTransition()**

Если вы хотите задать собственную анимацию при переключении между активностями, то можете сделать это несколькими способами.

У класса **Activity** есть метод **overridePendingTransition()** и он должен идти сразу после вызова метода **startActivity()** или **finish()**. Метод принимает два параметра: ресурс анимации для запускающей активности и ресурс анимации для уходящей активности.

Таким образом вам нужно создать в папке **res/anim** два файла анимации и передать их идентификаторы в метод.

Предположим, первый файл будет **diagonaltranslate.xml**:

<?xml version="1.0" encoding="utf-8"?>

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

<translate

android:duration="1100"

android:fromXDelta="100%p"

android:fromYDelta="100%p"

android:toXDelta="0"

android:toYDelta="0" />

</set>

Второй файл - **alpha.xml**:

<?xml version="1.0" encoding="utf-8"?>

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

<alpha

android:duration="300"

android:fromAlpha="1.0"

android:toAlpha="0.0" />

</set>

В одном из старых проектов я нашёл метод **startActivity()** и применил этот способ:

Intent questionIntent = new Intent(MainActivity.this,

AboutActivity.class);

startActivityForResult(questionIntent, CHOOSE\_THIEF);

overridePendingTransition(R.anim.diagonaltranslate,R.alpha);

Запускаем проект и переходим на другую активность. Новая активность будет расти из нижнего правого угла по диагонали, а текущая активность будет становиться прозрачной. Тут всё зависит от вашей фантазии, придумайте собственные сценарии анимации. Для примера я использовал достаточно большие значения *duration*, на самом деле не стоит увлекаться и старайтесь указывать короткие интервалы.

Ещё пример анимации, когда экран сдвигается влево, а при возвращении обратно - вверх. Привожу только файлы анимации.

**res/anim/left\_out.xml**

<?xml version="1.0" encoding="utf-8"?>

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

android:interpolator="@android:anim/linear\_interpolator">

<translate

android:fromXDelta="0"

android:toXDelta="-100%p"

android:duration="500"/>

</set>

**res/anim/right\_in.xml**

<?xml version="1.0" encoding="utf-8"?>

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

android:interpolator="@android:anim/linear\_interpolator">

<translate

android:fromXDelta="100%p"

android:toXDelta="0"

android:duration="500"/>

</set>

**res/anim/top\_out.xml**

<?xml version="1.0" encoding="utf-8"?>

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

android:interpolator="@android:anim/linear\_interpolator">

<translate

android:fromYDelta="0"

android:toYDelta="-100%p"

android:duration="500"/>

</set>

**res/anim/bottom\_in.xml**

**<?xml version="1.0" encoding="utf-8"?>**

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

**android:interpolator="@android:anim/linear\_interpolator">**

**<translate**

**android:fromYDelta="100%p"**

**android:toYDelta="0"**

**android:duration="500"/>**

**</set>**

**Анимация при помощи XML**

Кстати, есть способ использовать анимацию без единой строчки Java-кода, только XML.

Создайте в папке **res/anim** четыре файла:

**activity\_down\_up\_enter.xml**

<?xml version="1.0" encoding="utf-8"?>

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

android:interpolator="@android:anim/bounce\_interpolator" >

<translate

android:duration="2000"

android:fromYDelta="100%"

android:toYDelta="0" />

</set>

**activity\_down\_up\_exit.xml**

<?xml version="1.0" encoding="utf-8"?>

<set

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

android:interpolator="@android:anim/bounce\_interpolator">

<translate

android:fromYDelta="0%"

android:toYDelta="-100%"

android:duration="2000" />

</set>

**activity\_down\_up\_close\_enter.xml**

<?xml version="1.0" encoding="utf-8"?>

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

android:interpolator="@android:anim/decelerate\_interpolator" >

<alpha

android:duration="1000"

android:fromAlpha="0.2"

android:toAlpha="1.0" />

</set>

**activity\_down\_up\_close\_exit.xml**

<?xml version="1.0" encoding="utf-8"?>

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

android:interpolator="@android:anim/decelerate\_interpolator" >

<scale

android:duration="900"

android:fromXScale="1.0"

android:fromYScale="1.0"

android:pivotX="50%"

android:pivotY="50%"

android:toXScale="0.0"

android:toYScale="0.0" />

</set>

Теперь пропишем стили на основе наших файлов.

**res/values/styles.xml**

<resources>

<style name="AppTheme" parent="android:Theme.Light" />

<style name="DownUpActivity" parent="@android:style/Animation.Activity">

<item name="android:activityOpenEnterAnimation">@anim/activity\_down\_up\_enter</item>

<item name="android:activityOpenExitAnimation">@anim/activity\_down\_up\_exit</item>

<item name="android:activityCloseExitAnimation">@anim/activity\_down\_up\_close\_exit</item>

<item name="android:activityCloseEnterAnimation">@anim/activity\_down\_up\_close\_enter</item>

</style>

</resources>

Создадим файл для темы.

**res/values/themes.xml**

<?xml version="1.0" encoding="utf-8"?>

<resources>

<style

name="AnimationActivity"

parent="@android:style/Theme.Light">

<item name="android:windowAnimationStyle">@style/DownUpActivity</item>

</style>

</resources>

Остался последний шаг. Пропишем темы в манифесте для двух активностей, между которыми будем переключаться.

<activity

...

android:label="@string/title\_activity\_test"

android:theme="@style/AnimationActivity" >

...

</activity>

<activity

android:name="ru.alexanderklimov.animation.SecondActivity"

android:label="@string/title\_activity\_second"

android:theme="@style/AnimationActivity" >

...

</activity>

Создайте в коде переход на второй экран и запустите проект. При нажатии на кнопку, которая запускает вторую активность, вы увидите, что второй экран приложеиня появляется с анимацией движения снизу вверх с пристукиванием. При переходе обратно на первый экран, вторая активность будет уменьшаться в размерах, сворачиваясь по центру.

Рекомендую вживую посмотреть на эффекты, вам понравится.