**Rotate**

Используйте атрибуты **fromDegrees** и **toDegrees** внутри тега **<rotate>**, чтобы задать начальный и конечный углы поворота вокруг опорной точки. Опорная точка указывается с помощью атрибутов **pivotX** и **pivotY**, которые содержат процентные значения относительно ширины и высоты объекта **Drawable** соответственно.

Рассмотрим примеры вращения объекта. Создадим проект, в котором элемент ImageView будет постоянно вращаться либо относительно своего центра, либо вокруг своего угла. В папке **res/anim** создаем два файла:

**rotate\_center.xml**

<?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="2000"

android:fromDegrees="0"

android:pivotX="50%"

android:pivotY="50%"

android:startOffset="0"

android:toDegrees="360" />

</set>

**rotate\_corner.xml**

<?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="2000"

android:fromDegrees="0"

android:pivotX="0%"

android:pivotY="0%"

android:startOffset="0"

android:toDegrees="360" />

</set>

Разместим на форме две кнопки и ImageView:

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

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<Button

android:id="@+id/rotatecenter"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Вращение относительно центра" />

<Button

android:id="@+id/rotatecorner"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Вращение относительно угла картинки" />

<ImageView

android:id="@+id/imageView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="120dp"

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

</LinearLayout>

Теперь код для анимации:

package ru.alexanderklimov.testapp;

import android.app.Activity;

import android.os.Bundle;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.Animation.AnimationListener;

import android.view.animation.AnimationUtils;

import android.widget.Button;

import android.widget.ImageView;

public class TestAppActivity extends Activity {

ImageView myImageView;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

myImageView = (ImageView) findViewById(R.id.imageView1);

Button buttonRotateCenter = (Button) findViewById(R.id.rotatecenter);

Button buttonRotateCorner = (Button) findViewById(R.id.rotatecorner);

final Animation animationRotateCenter = AnimationUtils.loadAnimation(

this, R.anim.rotate\_center);

final Animation animationRotateCorner = AnimationUtils.loadAnimation(

this, R.anim.rotate\_corner);

buttonRotateCenter.setOnClickListener(new Button.OnClickListener() {

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

myImageView.startAnimation(animationRotateCenter);

}

});

buttonRotateCorner.setOnClickListener(new Button.OnClickListener() {

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

myImageView.startAnimation(animationRotateCorner);

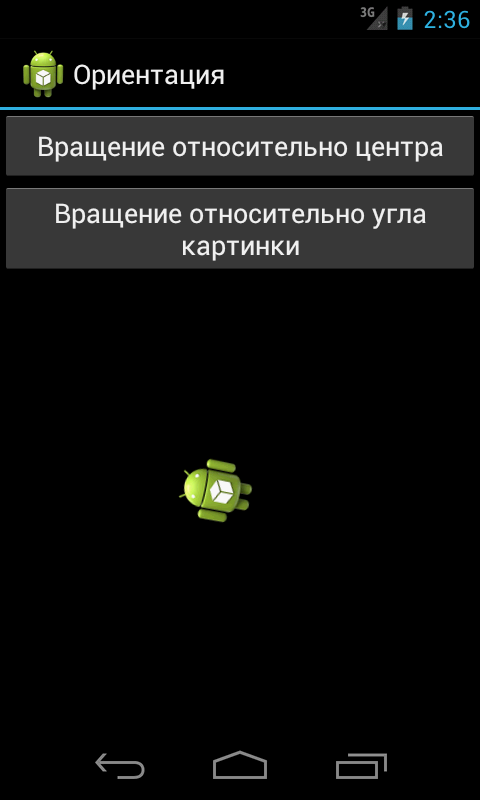
}

});

}

}

Запускайте проект и вращайте картинку.



**Смотри также**

[Класс RotateAnimation](http://developer.alexanderklimov.ru/android/animation/rotateanimation.php)