**Управляй мечтой!**

Рассмотрим пример анимации при помощи Object Animator и Value Animator. Мы создадим анимацию вращения автомобильного руля и проплывающих облачков.

Создадим новый проект с минимальной версией API 11. Создадим папку **res/drawable** и добавим в неё две картинки с облаком и рулём.

В этой же папке разместим два XML-файла, которые будут выводить изображение солнца (круг с градиентом) и травы (зелёный прямоугольник) через графические примитивы:

**sun.xml**

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

android:dither="true"

android:shape="oval" >

<gradient

android:endColor="#ffff6600"

android:gradientRadius="150"

android:startColor="#ffffcc00"

android:type="radial"

android:useLevel="false" />

<size

android:height="100dp"

android:width="100dp" />

</shape>

**ground.xml**

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

android:dither="true"

android:shape="rectangle" >

<solid android:color="#339933" />

</shape>

**window.xml**

Также добавим имитацию окна с рамкой. Сам прямоугольник будет прозрачным, а рамка будет серой.

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

android:shape="rectangle" >

<solid android:color="#00000000" />

<stroke

android:width="40dp"

android:color="#cccccc" />

</shape>

**Разработка дизайна**

Теперь, когда у нас есть необходимые графические ресурсы, можно приступать к дизайну приложения. Для начала добавим несколько строковых ресурсов:

<string name="wheel">Steering Wheel</string>

<string name="ground">The Ground</string>

<string name="window">Window Frame</string>

<string name="sun">The Sun</string>

<string name="cloud">A Cloud</string>

Создадим разметку:

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

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

android:id="@+id/car\_layout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#66ccff"

tools:context=".MainActivity" >

<ImageView

android:id="@+id/sun"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:contentDescription="@string/sun"

android:paddingLeft="100dp"

android:paddingTop="45dp"

android:src="@drawable/sun" />

<ImageView

android:id="@+id/cloud1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:contentDescription="@string/cloud"

android:paddingLeft="170dp"

android:paddingTop="70dp"

android:src="@drawable/cloud" />

<ImageView

android:id="@+id/cloud2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:contentDescription="@string/cloud"

android:paddingLeft="200dp"

android:paddingTop="90dp"

android:src="@drawable/cloud" />

<ImageView

android:id="@+id/ground"

android:layout\_width="fill\_parent"

android:layout\_height="200dp"

android:layout\_alignParentBottom="true"

android:contentDescription="@string/ground"

android:padding="40dp"

android:src="@drawable/ground" />

<ImageView

android:id="@+id/window"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:contentDescription="@string/window"

android:src="@drawable/window" />

<ImageView

android:id="@+id/wheel"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

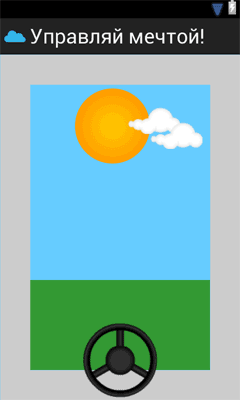
android:layout\_centerHorizontal="true"

android:contentDescription="@string/wheel"

android:padding="3dp"

android:src="@drawable/steering\_wheel" />

</RelativeLayout>



Обратите внимание, что ImageView с изображением руля мы добавили последним, поэтому изображение руля накладывается на окно с рамкой.

**Анимация**

Займёмся анимацией руля и солнца. Создадим новую папку **res/animator**.

Добавим в папку новый файл для анимация вращения руля.

**wheel\_spin.xml**

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

android:interpolator="@android:anim/accelerate\_decelerate\_interpolator"

android:ordering="sequentially" >

<objectAnimator

android:duration="3000"

android:propertyName="rotation"

android:repeatCount="infinite"

android:repeatMode="reverse"

android:valueTo="180"

android:valueType="floatType" />

</set>

Мы задали бесконечную анимацию вращения на 180 градусов с возвратом.

Давайте теперь запустим анимацию в коде, чтобы увидеть результат вживую. Добавим код в метод **onCreate()**

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

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

AnimatorSet wheelSet = (AnimatorSet) AnimatorInflater.loadAnimator(this, R.animator.wheel\_spin);

wheelSet.setTarget(wheel);

wheelSet.start();

}

Запустите проект и посмотрите на результат.

Для большей реалистичности добавим анимацию солца. В папке **res/animator** создадим новый файл:

**sun\_swing.xml**

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

android:interpolator="@android:anim/accelerate\_decelerate\_interpolator"

android:ordering="sequentially" >

<objectAnimator

android:duration="3000"

android:propertyName="x"

android:repeatCount="infinite"

android:repeatMode="reverse"

android:valueTo="-400"

android:valueType="floatType" />

</set>

В данном случае мы задали анимацию смещения объекта то в одну, то в другую стороны. Теперь напишем код в том же методе **onCreate()**:

// солнце

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

// загружаем анимацию движения солнца

AnimatorSet sunSet = (AnimatorSet) AnimatorInflater.loadAnimator(this, R.animator.sun\_swing);

// указываем объект солнца для анимации

sunSet.setTarget(sun);

// запускаем анимацию

sunSet.start();

Запустим приложение снова и видим, как солнце уходит за пределы экрана и возвращается обратно в соответствии с вращением руля.

Теперь займёмся анимацией неба - будем его затемнять, когда солнце уходит. На этот раз обойдёмя без xml-файлов анимации, а зададим анимацию в коде. Опять в том же методе **onCreate()** добавляем новый код:

ValueAnimator skyAnim = ObjectAnimator.ofInt

(findViewById(R.id.car\_layout), "backgroundColor",

Color.rgb(0x66, 0xcc, 0xff), Color.rgb(0x00, 0x66, 0x99));

skyAnim.setDuration(3000);

skyAnim.setRepeatCount(ValueAnimator.INFINITE);

skyAnim.setRepeatMode(ValueAnimator.REVERSE);

skyAnim.setEvaluator(new ArgbEvaluator());

skyAnim.start();

Снова запускаем приложение и смотрим на потрясающий эффект. Настоящее кино!

Единственное, что смущает - это неподвижные облака - белокрылые лошадки. Давайте их тоже анимируем. Продолжим код в том же методе:

// анимация первого облака

ObjectAnimator cloudAnim = ObjectAnimator.ofFloat(findViewById(R.id.cloud1), "x", -350);

cloudAnim.setDuration(3000);

cloudAnim.setRepeatCount(ValueAnimator.INFINITE);

cloudAnim.setRepeatMode(ValueAnimator.REVERSE);

cloudAnim.start();

// анимация второго облака

ObjectAnimator cloudAnim2 = ObjectAnimator.ofFloat(findViewById(R.id.cloud2), "x", -300);

cloudAnim2.setDuration(3000);

cloudAnim2.setRepeatCount(ValueAnimator.INFINITE);

cloudAnim2.setRepeatMode(ValueAnimator.REVERSE);

cloudAnim2.start();

Источник: [Android SDK: Creating a Simple Property Animation](http://mobile.tutsplus.com/tutorials/android/android-sdk-creating-a-simple-property-animation/) (там же можно найти исходники)