Анимация на Android

**Среда программирования:**

Android Studio

Создадим полноэкранное приложение.

Наше приложение будет работать по следующему принципу:  
На каждом шаге прорисовки (**onDraw(Canvas)**) получаем переменную **curTime**, которая будет хранить текущее время. В зависимости от ее значения, объекты, которые мы рисуем, меняют свои положение и свойства. Под конец метода прорисовки вызываем **invalidate()**, который заставляет **MyView**вызвать функцию прорисовки заново.

**MyView:**

**package** me.graphica.canvasanimation;

**import** android.animation.ArgbEvaluator;

**import** android.content.Context;

**import** android.graphics.Canvas;

**import** android.graphics.Color;

**import** android.graphics.Paint;

**import** android.view.View;

**public** **class** MyView **extends** [View](http://www.google.com/search?hl=en&q=allinurl%3Aview+java.sun.com&btnI=I%27m%20Feeling%20Lucky) {

**private** **final** **int** CIRCLE\_COUNT = 20; *//количество окружностей*

**private** **final** **float** CIRCLE\_ANIMATION = 0.5f; *//величина поворота окружностей на каждый кадр*

**private** **final** **long** RECT\_ANIMATION = 1\_000; *//длительность анимации прямоугольника = 1 секунда*

**private** **final** **long** COLOR\_ANIMATION = 1\_000; *//длительность анимации смены цветов = 1 секунда*

**private** **int** mWidth; *//ширина view*

**private** **int** mHeight; *//высота view*

**private** [Paint](http://www.google.com/search?hl=en&q=allinurl%3Apaint+java.sun.com&btnI=I%27m%20Feeling%20Lucky) mCirclePaint;

**private** [Paint](http://www.google.com/search?hl=en&q=allinurl%3Apaint+java.sun.com&btnI=I%27m%20Feeling%20Lucky) mRectPaint;

**private** ArgbEvaluator mArgbEvaluator; *//класс для анимации смены цветов*

**private** **long** mStartTime; *//время начала анимации*

**private** **float** mCircleDegree; *//текущий угол поворота окружностей*

**public** MyView([Context](http://www.google.com/search?hl=en&q=allinurl%3Acontext+java.sun.com&btnI=I%27m%20Feeling%20Lucky) context) {

**super**(context);

mCirclePaint = **new** [Paint](http://www.google.com/search?hl=en&q=allinurl%3Apaint+java.sun.com&btnI=I%27m%20Feeling%20Lucky)();

mCirclePaint.setAntiAlias(**true**);

mCirclePaint.setColor([Color](http://www.google.com/search?hl=en&q=allinurl%3Acolor+java.sun.com&btnI=I%27m%20Feeling%20Lucky).WHITE);

mCirclePaint.setStyle([Paint](http://www.google.com/search?hl=en&q=allinurl%3Apaint+java.sun.com&btnI=I%27m%20Feeling%20Lucky).[Style](http://www.google.com/search?hl=en&q=allinurl%3Astyle+java.sun.com&btnI=I%27m%20Feeling%20Lucky).FILL);

mRectPaint = **new** [Paint](http://www.google.com/search?hl=en&q=allinurl%3Apaint+java.sun.com&btnI=I%27m%20Feeling%20Lucky)(mCirclePaint);

mRectPaint.setColor([Color](http://www.google.com/search?hl=en&q=allinurl%3Acolor+java.sun.com&btnI=I%27m%20Feeling%20Lucky).RED);

mArgbEvaluator = **new** ArgbEvaluator();

}

@Override

**protected** **void** onSizeChanged(**int** w, **int** h, **int** oldw, **int** oldh) { *//вызывается при изменении размера view*

mWidth = w; *//запоминаем ширину view*

mHeight = h; *//запоминаем высоту view*

mStartTime = getTime(); *//начинаем отсчет времени*

}

**private** **long** getTime() { *//возвращает текущее время в миллисекундах*

**return** [System](http://www.google.com/search?hl=en&q=allinurl%3Asystem+java.sun.com&btnI=I%27m%20Feeling%20Lucky).nanoTime() / 1\_000\_000;

}

@Override

**protected** **void** onDraw([Canvas](http://www.google.com/search?hl=en&q=allinurl%3Acanvas+java.sun.com&btnI=I%27m%20Feeling%20Lucky) canvas) {

colorAnimation(canvas); *//демонстрация анимации цвета*

rotateAnimation(canvas); *//демонстрация анимации поворота*

moveAnimation(canvas); *//демонстрация анимации движения*

invalidate(); *//заставляет перерисовывать view, пока view находится в поле зрения пользователя*

}

**private** **void** rotateAnimation([Canvas](http://www.google.com/search?hl=en&q=allinurl%3Acanvas+java.sun.com&btnI=I%27m%20Feeling%20Lucky) canvas) {

canvas.save(); *//запоминаем текущее состояние canvas*

**int** centerX = mWidth / 2;

**int** centerY = mHeight / 2;

canvas.translate(centerX, centerY); *//рисуем окружности вокруг центра view*

mCircleDegree += CIRCLE\_ANIMATION; *// += 0.5f*

canvas.rotate(mCircleDegree); *//устанавливаем угол поворота*

**int** rad = [Math](http://www.google.com/search?hl=en&q=allinurl%3Amath+java.sun.com&btnI=I%27m%20Feeling%20Lucky).min(mWidth, mHeight) / 4; *//расстояние между противоположными окружностями*

**int** cir\_rad = rad / 12; *//радиус самих окружностей*

**for** (**int** i = 0; i < CIRCLE\_COUNT; i++) { *//вычисляет положение и рисует 20 окружностей*

**double** t = (2 \* [Math](http://www.google.com/search?hl=en&q=allinurl%3Amath+java.sun.com&btnI=I%27m%20Feeling%20Lucky).PI / CIRCLE\_COUNT) \* i;

**double** x = rad \* [Math](http://www.google.com/search?hl=en&q=allinurl%3Amath+java.sun.com&btnI=I%27m%20Feeling%20Lucky).cos(t);

**double** y = rad \* [Math](http://www.google.com/search?hl=en&q=allinurl%3Amath+java.sun.com&btnI=I%27m%20Feeling%20Lucky).sin(t);

canvas.drawCircle((**float**) x, (**float**) y, cir\_rad, mCirclePaint);

}

canvas.restore(); *//восстанавливаем canvas*

}

**private** **void** moveAnimation([Canvas](http://www.google.com/search?hl=en&q=allinurl%3Acanvas+java.sun.com&btnI=I%27m%20Feeling%20Lucky) canvas) {

canvas.save();

**float** rectWidth = mWidth / 10;

**float** rectHeight = mHeight / 10;

**long** curTime = getTime() - mStartTime; *//время с момента начала анимации в мс*

**float** fraction = (**float**) (curTime % RECT\_ANIMATION) / RECT\_ANIMATION; *//дробь от 0 до 1, где 0 - начало анимации, 1 - ее завершение*

**if** ((curTime / RECT\_ANIMATION) % 2 == 1)

fraction = 1 - fraction; *//делает обратную анимацию, если она завершилась*

canvas.translate((mWidth - rectWidth) \* fraction, mHeight - rectHeight);

canvas.drawRect(0, 0, rectWidth, rectHeight, mRectPaint);

canvas.restore();

}

**private** **void** colorAnimation([Canvas](http://www.google.com/search?hl=en&q=allinurl%3Acanvas+java.sun.com&btnI=I%27m%20Feeling%20Lucky) canvas) {

**long** curTime = getTime() - mStartTime;

**float** fraction = (**float**) (curTime % COLOR\_ANIMATION) / COLOR\_ANIMATION;

**if** ((curTime / COLOR\_ANIMATION) % 2 == 1)

fraction = 1 - fraction;

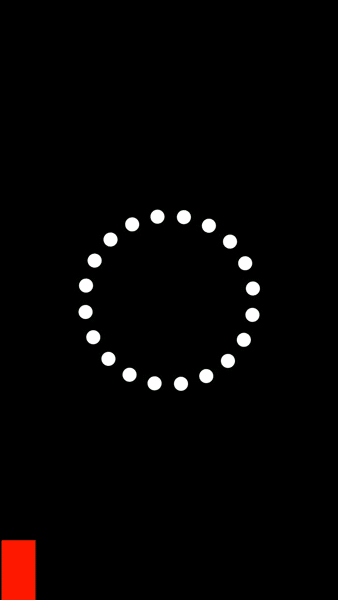
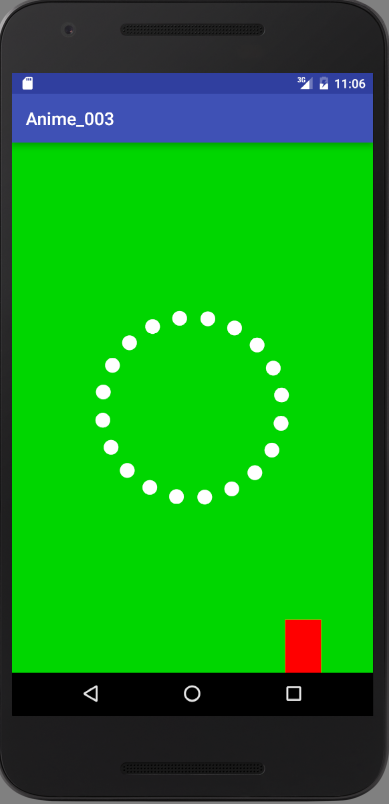
**int** color = (**int**) mArgbEvaluator.evaluate(fraction, [Color](http://www.google.com/search?hl=en&q=allinurl%3Acolor+java.sun.com&btnI=I%27m%20Feeling%20Lucky).BLACK, [Color](http://www.google.com/search?hl=en&q=allinurl%3Acolor+java.sun.com&btnI=I%27m%20Feeling%20Lucky).GREEN); *//evaluate вычислаяет промежуточный цвет между двумя цветами*

*//в зависимоти от fraction = [0, 1]*

canvas.drawColor(color);

}

}

Приложение выглядит следующим образом:  
 

Главная программа имеет вид:

MainActivity.java

**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(**new** MyView(**this**));  
 }  
}

Обратите внимание, что размер и положение графических объектов зависят только от ширины (**mWidth**) и высоты (**mHeight**) экрана. Таким образом наше приложение будет корректно отображаться на всех Android устройствах.

Меняя следующие строки

**private** **final** **int** CIRCLE\_COUNT = 20; *//количество окружностей*

**private** **final** **float** CIRCLE\_ANIMATION = 0.5f; *//величина поворота окружностей на каждый кадр*

**private** **final** **long** RECT\_ANIMATION = 1\_000; *//длительность анимации прямоугольника = 1 секунда*

**private** **final** **long** COLOR\_ANIMATION = 1\_000; *//длительность анимации смены цветов = 1 секунда*

мы можем легко изменить поведение нашего приложения.

Например,

**private** **final** **int** CIRCLE\_COUNT = 10; *//количество окружностей*

**private** **final** **float** CIRCLE\_ANIMATION = 1f; *//величина поворота окружностей на каждый кадр*

**private** **final** **long** RECT\_ANIMATION = 1\_000; *//длительность анимации прямоугольника = 1 секунда*

**private** **final** **long** COLOR\_ANIMATION = 500; *//длительность анимации смены цветов = 0,5 секунды*

