**Котошоп - примеры с градиентами**



[LinearGradient](http://developer.alexanderklimov.ru/android/catshop/gradient.php#lineargradient)  
[RadialGradient](http://developer.alexanderklimov.ru/android/catshop/gradient.php#radialgradient)  
[SweepGradient](http://developer.alexanderklimov.ru/android/catshop/gradient.php#sweepgradient)  
[Класс GradientDrawable](http://developer.alexanderklimov.ru/android/catshop/gradient.php#gradientdrawable)

Примеры основаны на шаблоне из статьи [Работаем с графикой. Основы](http://developer.alexanderklimov.ru/android/simplepaint.php). Создаём отдельный класс Draw2D.java и в нём экспериментируем.

**LinearGradient**

Линейный градиент изменяет цвет вдоль одной линии. Это самый стандартный градиент, часто используемый в практике. Менять цвет можно не только сверху вниз или слева направо, но и под углом.

package ru.alexanderklimov.test;

import android.content.Context;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.LinearGradient;

import android.graphics.Paint;

import android.graphics.RadialGradient;

import android.graphics.Shader;

import android.view.View;

public class Draw2D extends View {

Paint BackPaint = new Paint();

public Draw2D(Context context) {

super(context);

}

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

float w, h, cx, cy, radius;

w = getWidth();

h = getHeight();

cx = w / 2;

cy = h / 2;

if (w > h) {

radius = h / 4;

} else {

radius = w / 4;

}

canvas.drawRect(0, 0, w, h, BackPaint);

Paint MyPaint = new Paint();

MyPaint.setStyle(Paint.Style.FILL);

int shaderColor0 = Color.RED;

int shaderColor1 = Color.BLUE;

MyPaint.setAntiAlias(true);

Shader linearGradientShader;

linearGradientShader = new LinearGradient(0, 0, w, h, shaderColor1,

shaderColor0, Shader.TileMode.MIRROR);

MyPaint.setShader(linearGradientShader);

canvas.drawRect(0, 0, w, h, MyPaint);

linearGradientShader = new LinearGradient(cx, cy, cx + radius, cy

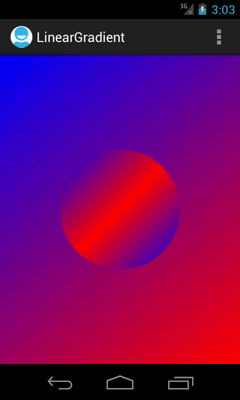
+ radius, shaderColor0, shaderColor1, Shader.TileMode.MIRROR);

MyPaint.setShader(linearGradientShader);

canvas.drawCircle(cx, cy, radius, MyPaint);

}

}



**RadialGradient**

По названию видно, что градиент идёт от центра к краям окружности вдоль радиуса.

package ru.alexanderklimov.test;

import android.content.Context;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.RadialGradient;

import android.graphics.Shader;

import android.view.View;

public class Draw2D extends View {

Paint BackPaint = new Paint();

public Draw2D(Context context) {

super(context);

}

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

float w, h, cx, cy, radius;

w = getWidth();

h = getHeight();

cx = w / 2;

cy = h / 2;

if (w > h) {

radius = h / 4;

} else {

radius = w / 4;

}

canvas.drawRect(0, 0, w, h, BackPaint);

Paint MyPaint = new Paint();

MyPaint.setStyle(Paint.Style.FILL);

float shaderCx = 0;

float shaderCy = 0;

float shaderRadius = w;

int shaderColor0 = Color.WHITE;

int shaderColor1 = Color.BLACK;

MyPaint.setAntiAlias(true);

Shader radialGradientShader;

radialGradientShader = new RadialGradient(shaderCx, shaderCy,

shaderRadius, shaderColor0, shaderColor1,

Shader.TileMode.MIRROR);

MyPaint.setShader(radialGradientShader);

canvas.drawRect(0, 0, w, h, MyPaint);

shaderCx = cx;

shaderCy = cy;

shaderRadius = radius;

shaderColor0 = Color.RED;

shaderColor1 = Color.BLUE;

radialGradientShader = new RadialGradient(shaderCx, shaderCy,

shaderRadius, shaderColor0, shaderColor1,

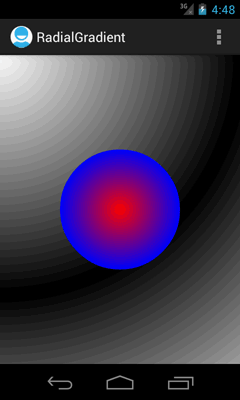
Shader.TileMode.MIRROR);

MyPaint.setShader(radialGradientShader);

canvas.drawCircle(cx, cy, radius, MyPaint);

}

}



**SweepGradient**

Ещё один красивый вид градиента - **SweepGradient**:

package ru.alexanderklimov.test;

import android.content.Context;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.SweepGradient;

import android.view.View;

public class Draw2D extends View {

Paint BackPaint = new Paint();

public Draw2D(Context context) {

super(context);

}

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

float w, h, cx, cy, radius;

w = getWidth();

h = getHeight();

cx = w / 2;

cy = h / 2;

if (w > h) {

radius = h / 4;

} else {

radius = w / 4;

}

canvas.drawRect(0, 0, w, h, BackPaint);

Paint MyPaint = new Paint();

MyPaint.setStyle(Paint.Style.FILL);

float shaderCx = cx;

float shaderCy = cy;

int shaderColor0 = Color.RED;

int shaderColor1 = Color.BLUE;

MyPaint.setAntiAlias(true);

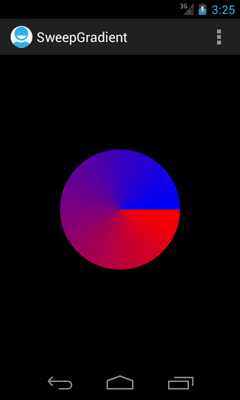
MyPaint.setShader(new SweepGradient(shaderCx, shaderCy, shaderColor0,

shaderColor1));

canvas.drawCircle(cx, cy, radius, MyPaint);

}

}



**Класс GradientDrawable**

Класс **GradientDrawable** позволяет закрашивать градиентом кнопки, фон и так далее. В качестве альтернативы можно использовать XML-файлы с элементом <shape>.

Закрасим фон элемента View.

package ru.alexanderklimov.simplepaint;

import android.content.Context;

import android.graphics.Canvas;

import android.graphics.drawable.GradientDrawable;

import android.graphics.drawable.GradientDrawable.Orientation;

import android.view.View;

public class Draw2D extends View {

public Draw2D(Context context) {

super(context);

}

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

GradientDrawable gradient = new GradientDrawable(

Orientation.BOTTOM\_TOP, new int[] { 0x805a54ff, 0x8014afff,

0x8045fd07, 0x80fffe05, 0x80f3af04, 0x80ff0b1a });

gradient.setBounds(0, 0, 480, 800);

gradient.draw(canvas);

}

}

