**Noise шумовой эффект**

Продолжаем придумывать различные эффекты для изображений. Возьмём каждую точку изображения и применим к ней операцию OR с случайным цветом.

За основу возьмём [предыдущий пример](http://developer.alexanderklimov.ru/android/catshop/flip.php) и удалим из неё вторую кнопку.

package ru.alexanderklimov.invert\_1098;

butInvert.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

BitmapDrawable mydrawable = (BitmapDrawable) imageView.getDrawable();

Bitmap b = mydrawable.getBitmap();

b = applyNoiseEffect(b);

imageView.setImageBitmap(b);

Toast.makeText(getApplicationContext(),

"Сделано", Toast.LENGTH\_LONG).show();

}

});

public static Bitmap applyNoiseEffect(Bitmap source) {

final int COLOR\_MAX = 0xFF;

// получаем размеры изображения

int width = source.getWidth();

int height = source.getHeight();

int[] pixels = new int[width \* height];

// получаем массив пикселей из изображения

source.getPixels(pixels, 0, width, 0, 0, width, height);

// генерация случайных чисел

Random random = new Random();

int index = 0;

// проходим через каждый пиксель

for(int y = 0; y < height; ++y) {

for(int x = 0; x < width; ++x) {

// get current index in 2D-matrix

index = y \* width + x;

// получаем случайный цвет

int randColor = Color.rgb(random.nextInt(COLOR\_MAX),

random.nextInt(COLOR\_MAX), random.nextInt(COLOR\_MAX));

// применяем операцию OR

pixels[index] |= randColor;

}

}

// выходное изображение

Bitmap bmOut = Bitmap.createBitmap(width, height, source.getConfig());

bmOut.setPixels(pixels, 0, width, 0, 0, width, height);

return bmOut;

}

Источник: [Flea / Noise Effect](http://xjaphx.wordpress.com/2011/10/30/image-processing-flea-noise-effect/#1098)

 

**Р**