**Snow Effect эффект снега**

Продолжаем придумывать различные эффекты для изображений. На этот раз зададим некоторый порог для цвета. И если значение цвета в случайной точке больше заданного порога, то в точке устанавливаем белый цвет. Таким образом получаем эффект снега.

Попробуем сделать это на Android. 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 = applySnowEffect(b);

imageView.setImageBitmap(b);

Toast.makeText(getApplicationContext(),

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

}

});

public static Bitmap applySnowEffect(Bitmap source) {

final int COLOR\_MAX = 0xFF;

// get image size

int width = source.getWidth();

int height = source.getHeight();

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

// get pixel array from source

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

// random object

Random random = new Random();

int R, G, B, index = 0, thresHold = 50;

// iteration through pixels

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

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

// get current index in 2D-matrix

index = y \* width + x;

// get color

R = Color.red(pixels[index]);

G = Color.green(pixels[index]);

B = Color.blue(pixels[index]);

// generate threshold

thresHold = random.nextInt(COLOR\_MAX);

if(R > thresHold && G > thresHold && B > thresHold) {

pixels[index] = Color.rgb(COLOR\_MAX, COLOR\_MAX, COLOR\_MAX);

}

}

}

// output bitmap

Bitmap bmOut = Bitmap.createBitmap(width, height, Bitmap.Config.RGB\_565);

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

return bmOut;

}

Источник: [Snow Effect](http://xjaphx.wordpress.com/2011/10/30/image-processing-snow-effect/#1098)

 

**Р**