**Color Depth уменьшаем глубину цвета**

В графических редакторах используется уменьшение глубины цвета.

Реализуем это на Android. За основу возьмём [предыдущий пример](http://developer.alexanderklimov.ru/android/catshop/flip.php) и удалим из неё вторую кнопку. Как и прежние операции, связанные с обработкой каждого пикселя изображения, данный процесс занимает значительное время.

Перейдем к коду.

butDemo.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

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

Bitmap b = mydrawable.getBitmap();

b = decreaseColorDepth(b, 32);

imageView.setImageBitmap(b);

setTitle("Offset: 32");

}

});

public static Bitmap decreaseColorDepth(Bitmap src, int bitOffset) {

// получаем размер изображения

int width = src.getWidth();

int height = src.getHeight();

// подготавливаем новое изображение

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

// информация о цвете пикселя

int A, R, G, B;

int pixel;

// проходим через каждый пиксель

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

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

// получаем цвет пикселя

pixel = src.getPixel(x, y);

A = Color.alpha(pixel);

R = Color.red(pixel);

G = Color.green(pixel);

B = Color.blue(pixel);

// устанавливаем смещение цвета

R = ((R + (bitOffset / 2)) - ((R + (bitOffset / 2)) % bitOffset) - 1);

if(R < 0) { R = 0; }

G = ((G + (bitOffset / 2)) - ((G + (bitOffset / 2)) % bitOffset) - 1);

if(G < 0) { G = 0; }

B = ((B + (bitOffset / 2)) - ((B + (bitOffset / 2)) % bitOffset) - 1);

if(B < 0) { B = 0; }

// устанавливем цвет пикселя на новом изображении

bmOut.setPixel(x, y, Color.argb(A, R, G, B));

}

}

// возвращаем финальное изображение

return bmOut;

}

Попробуйте самостоятельно поиграться с различными значениями.

Идея статьи: [Image Processing – Decreasing Color Depth](http://xjaphx.wordpress.com/2011/06/21/image-processing-decreasing-color-depth/#1098)

 

**Р**