Grayscale - градации серого

В графических редакторах есть операция преобразования картинки в градации серого цвета (Photoshop: Image | Mode | Grayscale).

Попробуем сделать это на Android. За основу возьмём [предыдущий пример](http://developer.alexanderklimov.ru/android/catshop/matrix.php#prescale) и удалим из неё вторую кнопку.

Перейдем к коду. Каждый пиксель изображения содержит информацию о четырёх каналах: Alpha (прозрачность), Red (красный), Green (зелёный), Blue (синий). Значение прозрачности мы не трогаем, а остальные значения цветов вычисляем по формуле: 0xFF – CurrentValue.

Существует специальный стандарт (можно поискать в Википедии), который описывает, как должно происходит подобное преобразование. Вкратце, измемение цвета точки меняется по формуле: 30% Red + 59% Green + 11 % Blue.

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 = doGreyscale(b);

imageView.setImageBitmap(b);

Toast.makeText(getApplicationContext(),

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

}

});

public static Bitmap doGreyscale(Bitmap src) {

// Константы для изменения цвета точки

final double GS\_RED = 0.299;

final double GS\_GREEN = 0.587;

final double GS\_BLUE = 0.114;

// Создадим новый bitmap с теми же настройками, как у источника

Bitmap bmOut = Bitmap.createBitmap(src.getWidth(), src.getHeight(), src.getConfig());

// информация о цветах

int A, R, G, B;

int pixel;

// Получаем размеры изображения

int width = src.getWidth();

int height = src.getHeight();

// проходим через каждый пиксель

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

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

// работаем с отдельным пикселем

pixel = src.getPixel(x, y);

// получаем значение каждого цветового канала

A = Color.alpha(pixel);

R = Color.red(pixel);

G = Color.green(pixel);

B = Color.blue(pixel);

// применяем конвертацию

R = G = B = (int)(GS\_RED \* R + GS\_GREEN \* G + GS\_BLUE \* B);

// применяем новые значения к той же точке

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

}

}

// Получаем новое изображение

return bmOut;

}

Запускаем проект и щелкаем на кнопку, чтобы получить изображение в градациях серого. Напоминаю, что это ресурсоёмкие вычисления, требующие много времени - необходимо пройтить по каждой точке изображения и заменить его цвет.

 

Смотри также [ColorMatrixColorFilter: Grayscale](http://developer.alexanderklimov.ru/android/catshop/colorfilters.php" \l "grayscale)