**Highlight Image подсветка**

Для опыта нам понадобится картинка с прозрачным фоном, чтобы создать ореол вокруг изображения. За основу возьмём [предыдущий пример](http://developer.alexanderklimov.ru/android/catshop/flip.php) и удалим из неё вторую кнопку.

Перейдем к коду.

butDemo.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

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

Bitmap b = mydrawable.getBitmap();

b = doHighlightImage(b);

imageView.setImageBitmap(b);

}

});

public static Bitmap doHighlightImage(Bitmap src) {

// создадим новый битмап, который станет итоговым

Bitmap bmOut = Bitmap.createBitmap(src.getWidth() + 96,

src.getHeight() + 96, Bitmap.Config.ARGB\_8888);

// подключаем холст

Canvas canvas = new Canvas(bmOut);

// установим цвет по умолчанию

canvas.drawColor(0, PorterDuff.Mode.CLEAR);

// создадим размытие для прозрачности

Paint ptBlur = new Paint();

ptBlur.setMaskFilter(new BlurMaskFilter(15, Blur.NORMAL));

int[] offsetXY = new int[2];

// получим прозрачный слепок из изображения

Bitmap bmAlpha = src.extractAlpha(ptBlur, offsetXY);

// готовимся к рисованию

Paint ptAlphaColor = new Paint();

ptAlphaColor.setColor(0xFFFFFFFF);

// закрашиваем цветом цветной слепок (bitmap)

canvas.drawBitmap(bmAlpha, offsetXY[0], offsetXY[1], ptAlphaColor);

// освобождаем ресурсы

bmAlpha.recycle();

// рисуем исходник

canvas.drawBitmap(src, 0, 0, null);

// возвращаем финальный рисунок

return bmOut;

}

Идея статьи: [Image Processing – Highlight Image On The Fly](http://xjaphx.wordpress.com/2011/06/20/image-processing-highlight-image-on-the-fly/#1098)

 