**Склеиваем два изображения**

Алгоритм склеивания двух изображений в одно будет следующим - вычисляем ширину у каждой картинки и складываем их, чтобы получить общую ширину новой картинки. По высоте алгоритм другой - смотрим, у какого изображения высота больше и её принимаем за основу. Далее создаётся новый **Bitmap**, а затем новый **Canvas**. В результате получим новую картинку, состоящую из двух выбранных изображений, взятых из Галереи.

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:orientation="vertical"

tools:context=".MainActivity" >

<Button

android:id="@+id/buttonLoadimage1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Load Image 1" />

<TextView

android:id="@+id/textViewSourceUri1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<Button

android:id="@+id/buttonLoadimage2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Load Image 2" />

<TextView

android:id="@+id/textViewSourceUri2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<Button

android:id="@+id/buttonProcessing"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Processing" />

<ImageView

android:id="@+id/imageViewResult"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

</LinearLayout>

Код

package ru.alexanderklimov.test;

import java.io.FileNotFoundException;

import android.app.Activity;

import android.content.Intent;

import android.graphics.Bitmap;

import android.graphics.Bitmap.Config;

import android.graphics.BitmapFactory;

import android.graphics.Canvas;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

public class MainActivity extends Activity {

private Button mLoadFirstImageButton, mLoadSecondImageButton;

private TextView mFirstSourceUriTextView, mSecondSourceUriTextView;

private Button mProcessingButton;

ImageView imageResult;

final int RQS\_IMAGE1 = 1;

final int RQS\_IMAGE2 = 2;

private Uri mSource1, mSource2;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mLoadFirstImageButton = (Button) findViewById(R.id.buttonLoadimage1);

mLoadSecondImageButton = (Button) findViewById(R.id.buttonLoadimage2);

mFirstSourceUriTextView = (TextView) findViewById(R.id.textViewSourceUri1);

mSecondSourceUriTextView = (TextView) findViewById(R.id.textViewSourceUri2);

mProcessingButton = (Button) findViewById(R.id.buttonProcessing);

imageResult = (ImageView) findViewById(R.id.imageViewResult);

mLoadFirstImageButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// Выбираем картинку из Галереи

Intent intent = new Intent(

Intent.ACTION\_PICK,

android.provider.MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI);

startActivityForResult(intent, RQS\_IMAGE1);

}

});

mLoadSecondImageButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View arg0) {

// Выбираем картинку из Галереи

Intent intent = new Intent(

Intent.ACTION\_PICK,

android.provider.MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI);

startActivityForResult(intent, RQS\_IMAGE2);

}

});

mProcessingButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

if (mSource1 != null && mSource2 != null) {

Bitmap processedBitmap = processingBitmap();

if (processedBitmap != null) {

imageResult.setImageBitmap(processedBitmap);

Toast.makeText(getApplicationContext(), "Done",

Toast.LENGTH\_LONG).show();

} else {

Toast.makeText(getApplicationContext(),

"Something wrong in processing!",

Toast.LENGTH\_LONG).show();

}

} else {

Toast.makeText(getApplicationContext(),

"Select both image!", Toast.LENGTH\_LONG).show();

}

}

});

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (resultCode == RESULT\_OK) {

switch (requestCode) {

case RQS\_IMAGE1:

mSource1 = data.getData();

mFirstSourceUriTextView.setText(mSource1.toString());

break;

case RQS\_IMAGE2:

mSource2 = data.getData();

mSecondSourceUriTextView.setText(mSource2.toString());

break;

}

}

}

private Bitmap processingBitmap() {

Bitmap bitmap1 = null;

Bitmap bitmap2 = null;

Bitmap newBitmap = null;

try {

bitmap1 = BitmapFactory.decodeStream(getContentResolver()

.openInputStream(mSource1));

bitmap2 = BitmapFactory.decodeStream(getContentResolver()

.openInputStream(mSource2));

// Вычисляем ширину для новой картинки

int width = bitmap1.getWidth() + bitmap2.getWidth();

int height;

if (bitmap1.getHeight() >= bitmap2.getHeight()) {

height = bitmap1.getHeight();

} else {

height = bitmap2.getHeight();

}

Config config = bitmap1.getConfig();

if (config == null) {

config = Bitmap.Config.ARGB\_8888;

}

newBitmap = Bitmap.createBitmap(width, height, config);

Canvas newCanvas = new Canvas(newBitmap);

newCanvas.drawBitmap(bitmap1, 0, 0, null);

newCanvas.drawBitmap(bitmap2, bitmap1.getWidth(), 0, null);

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

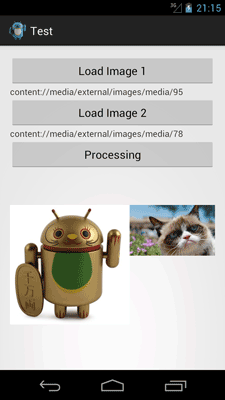
}

return newBitmap;

}

}

Идея: [Combine bitmap side-by-side](http://android-er.blogspot.ru/2013/08/combine-bitmap-side-by-side.html)



**Cклеиваем половинки одной картинки**

Изменим пример. Возьмём половинку картинки, отразим её вертикально и получим новый результат. Чтобы не менять сильно код от первого примера, оставим разметку с двумя кнопками для загрузки из Галереи, хотя нам понадобится только одна кнопка для этой цели.

В коде поменяем только метод **processingBitmap()**. Учтите, что в примере идёт проверка на загрузку второго изображения. Либо удалите эту проверку, либо загрузите какое-нибудь изображение как заглушку, которое не будет участвовать в примере.

private Bitmap processingBitmap() {

Bitmap tempBitmap;

Bitmap bitmapMaster = null;

try {

tempBitmap = BitmapFactory.decodeStream(getContentResolver()

.openInputStream(mSource1));

// create a mirror bitmap

Matrix matrixMirror = new Matrix();

matrixMirror.preScale(-1.0f, 1.0f);

Bitmap bitmapMirror = Bitmap.createBitmap(tempBitmap, 0, 0,

tempBitmap.getWidth(), tempBitmap.getHeight(),

matrixMirror, false);

// define half/half area

Rect rect1Half = new Rect(0, 0, tempBitmap.getWidth() / 2,

tempBitmap.getHeight());

Rect rect2Half = new Rect((tempBitmap.getWidth() / 2) + 1, 0,

tempBitmap.getWidth(), tempBitmap.getHeight());

Config config = tempBitmap.getConfig();

if (config == null) {

config = Bitmap.Config.ARGB\_8888;

}

// craete the master bitmap

bitmapMaster = Bitmap.createBitmap(tempBitmap.getWidth(),

tempBitmap.getHeight(), config);

Canvas canvasMaster = new Canvas(bitmapMaster);

// merge bitmap half/half

canvasMaster.drawBitmap(tempBitmap, rect1Half, rect1Half, null);

canvasMaster.drawBitmap(bitmapMirror, rect2Half, rect2Half, null);

imageResult.setImageBitmap(bitmapMaster);

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return bitmapMaster;

}

Иногда получаются интересные эффекты от этого способа. На рисунке представлен пример знакомого нам кота-андроида, у которого в результате эффекта появилось два хвоста. Брр.

