قيبطت ةجلاعم روصلا في ديوردنأ لهس

وتسب بب الذاكرة، تست هلك [illegible] المتجهة الصور أن حيث [illegible] معقدًا يكون أن يمكن أن ديوردنأ في الصور مع التعامل فيئات فيه تأتي الذي المكان هو ذلك دقة. [illegible] التقطيع وتطلب التوجه، في مشاكل [illegible] التحويلات com.lzw.flower.utils. حزمة من الفئات من قويًا زوجًا أريك سأكون المقال، هذا في مفيدة. Crop و BitmapUtils مثل المساعدة عرض تستسن اهترفي شيء للحن ما هلعفي لك طريقة ونعرض اهامدختسا في اسيعاري مشاعك. دعونا وغوص!

[illegible]: الأساسية الصور معالجة أداة فئة BitmapUtils هي مجموعة من الطرق الثابتة لتعديل كائنات Bitmap. بشرح: متبوعة الكاملة، الشيفرة إليك

```
package com.lzw.flower.utils;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Matrix;
import android.net.Uri;
import android.provider.MediaStore;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

public class BitmapUtils {
  public static Bitmap convertGreyImg(Bitmap img) {
    int width = img.getWidth();
    int height = img.getHeight();
    int[] pixels = new int[width * height];
    img.getPixels(pixels, 0, width, 0, 0, width, height);
    int alpha = 0xFF << 24;
    for (int i = 0; i < height; i++) {
      for (int j = 0; j < width; j++) {
        int grey = pixels[width * i + j];
        int red = ((grey & 0x00FF0000) >> 16);
        int green = ((grey & 0x0000FF00) >> 8);
        int blue = (grey & 0x000000FF);
        grey = (int) ((float) red * 0.3 + (float) green * 0.59 + (float) blue * 0.11);
        grey = alpha | (grey << 16) | (grey << 8) | grey;
```

```
      pixels[width * i + j] = grey;
    }
  }
  Bitmap result = Bitmap.createBitmap(width, height, Bitmap.Config.ARGB_8888);
  result.setPixels(pixels, 0, width, 0, 0, width, height);
  return result;
}

public static Bitmap toGreyImg(Bitmap bitmapOrg) {
  Bitmap bitmapNew = bitmapOrg.copy(Bitmap.Config.ARGB_8888, true);
  if (bitmapNew == null) {
    return null;
  }
  for (int i = 0; i < bitmapNew.getWidth(); i++) {
    for (int j = 0; j < bitmapNew.getHeight(); j++) {
      int col = bitmapNew.getPixel(i, j);
      int alpha = col & 0xFF000000;
      int red = (col & 0x00FF0000) >> 16;
      int green = (col & 0x0000FF00) >> 8;
      int blue = (col & 0x000000FF);
      int gray = (int) ((float) red * 0.3 + (float) green * 0.59 + (float) blue * 0.11);
      int newColor = alpha | (gray << 16) | (gray << 8) | gray;
      bitmapNew.setPixel(i, j, newColor);
    }
  }
  return bitmapNew;
}

public static void saveBitmapToPath(Bitmap bitmap, String imagePath) {
  FileOutputStream out = null;
  File file = new File(imagePath);
  if (file.getParentFile().exists() == false) {
    file.getParentFile().mkdirs();
  }
  try {
    out = new FileOutputStream(imagePath);
    bitmap.compress(Bitmap.CompressFormat.PNG, 100, out);
    out.flush();
  } catch (Exception e) {
    e.printStackTrace();
```

```
    } finally {
      try {
        if (out != null) out.close();
      } catch (IOException e) {
        e.printStackTrace();
      }
    }
}

public static Bitmap rotateBitmap(Bitmap source, float angle) {
    Matrix matrix = new Matrix();
    matrix.postRotate(angle);
    return Bitmap.createBitmap(source, 0, 0, source.getWidth(), source.getHeight(), matrix, true);
}

public static Uri getResourceUri(int resId) {
    return Uri.parse("android.resource://com.lzw.flower/" + resId);
}

public static Bitmap getBitmapByUri(Context cxt, Uri uri) throws IOException {
    return MediaStore.Images.Media.getBitmap(cxt.getContentResolver(), uri);
}

public static int calInSampleSize(BitmapFactory.Options options, int reqWidth) {
    int w = options.outWidth;
    int h = options.outHeight;
    int inSampleSize = 1;
    if (w > reqWidth && reqWidth > 0) {
      inSampleSize = Math.round(w / reqWidth);
    }
    return inSampleSize;
}

public static Bitmap decodeSampledBitmapFromPath(String path, int reqWidth) {
    BitmapFactory.Options options = new BitmapFactory.Options();
    options.inJustDecodeBounds = true;
    BitmapFactory.decodeFile(path, options);
    int inSampleSize = calInSampleSize(options, reqWidth);
    options.inJustDecodeBounds = false;
    options.inSampleSize = inSampleSize;
```

```
    return BitmapFactory.decodeFile(path, options);
  }

  public static Bitmap decodeFileByHeight(String path, int reqH) {
    BitmapFactory.Options opt = new BitmapFactory.Options();
    opt.inJustDecodeBounds = true;
    BitmapFactory.decodeFile(path, opt);
    int scale = calInSampleSizeByHeight(opt, reqH);
    opt.inSampleSize = scale;
    opt.inJustDecodeBounds = false;
    Bitmap bm = BitmapFactory.decodeFile(path, opt);
    return bm;
  }

  public static int calInSampleSizeByHeight(BitmapFactory.Options options, int reqHeight) {
    int h = options.outHeight;
    int inSampleSize = 1;
    if (h > reqHeight) {
      inSampleSize = Math.round(h * 1.0f / reqHeight);
    }
    return inSampleSize;
  }
}
```

ما يحتويه؟

- تحويل الألوان إلى رمادية:
  - convertGreyImg: تستخدم في الكاميرا لتحويل الصورة إلى رمادية. تقريبية، لكنها سريعة.
  - toGreyImg: تستخدم في الكاميرا لتحويل الصورة إلى رمادية. تعتمد على 0.3R + 0.59G + 0.11B
- عمليات الملفات:
  - saveBitmapToPath: تحفظ الصورة في المسار المحدد، وتخلق المجلدات إذا لزم الأمر.
- التحويلات:
  - rotateBitmap: تدير الصورة باستخدام Matrix بسيط. فعال ولكن ليس مرنًا.
- التحميل والتخزين المؤقت:
  - getBitmapByUri و getResourceUri: تحمّل الصور من الموارد أو المسارات.
  - decodeFileByHeight و decodeSampledBitmapFromPath: تغيير حجم الصور بكفاءة مع الحفاظ على نسبة العرض إلى الارتفاع، مما يقلل من استخدام الذاكرة.

□□□□: عيطعتت قيقد دام دختساباب أدوات ديوردنأ ةيلصألا ةئف Crop ديفتست من ةين عيطعتت ديوردنأ. ةيلصألا كيلإ
:ةرفيشلا

```
package com.lzw.flower.utils;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.provider.MediaStore;
import com.lzw.flower.base.App;

import java.io.File;

public class Crop {
  public static void startPhotoCrop(Activity cxt, Uri uri, String outputPath, int resultCode) {
    Intent intent = new Intent("com.android.camera.action.CROP");
    intent.setDataAndType(uri, "image/*");
    int w = App.drawWidth;
    int h = App.drawHeight;
    int factor = gcd(w, h);
    int w1 = w / factor;
    int h1 = h / factor;
    intent.putExtra("crop", "true")
        .putExtra("aspectX", w1)
        .putExtra("aspectY", h1)
        .putExtra("scale", true)
        .putExtra("outputX", w)
        .putExtra("outputY", h)
        .putExtra("outputFormat", Bitmap.CompressFormat.PNG.toString());
    intent.putExtra("noFaceDetection", true);
    intent.putExtra("return-data", false);
    Uri uri1 = Uri.fromFile(new File(outputPath));
    intent.putExtra(MediaStore.EXTRA_OUTPUT, uri1);
    cxt.startActivityForResult(intent, resultCode);
  }

  static int gcd(int a, int b) {
```

```
    if (b == 0) {
      return a;
    } else {
      return gcd(b, a % b);
    }
  }
}
```

ما حدث هنا؟

- startPhotoCrop: يبدأ عملية اقتطاع الصورة، ويحدد Uri مع النظام، ويسمح باستخدام مساحة العرض إلى الارتفاع المطلوبة، ويمرر الأساسية. في App فئة مثل آخر مكان في تعريفها تم App.drawHeight و App.drawWidth أن يفترض الإخراج.
- gcd: خوارزمية تكرارية لحساب أكبر عامل مشترك، أن لضمان نسبة العرض إلى الارتفاع في الأدوات هذه استخدام يمكنك هكذا البسيط. شكلها

استخدام أمثلة شيء: لكل جمع أندرويد: تطبيق في الأدوات هذه استخدام يمكنك هكذا

```
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import com.lzw.flower.utils.BitmapUtils;
import com.lzw.flower.utils.Crop;

public class MainActivity extends AppCompatActivity {
    private static final int REQUEST_CROP = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //
        Bitmap original = BitmapFactory.decodeResource(getResources(), R.drawable.sample);
        Bitmap grey = BitmapUtils.convertGreyImg(original);
        BitmapUtils.saveBitmapToPath(grey, "/sdcard/DCIM/grey_image.png");

        //
        Bitmap scaled = BitmapUtils.decodeSampledBitmapFromPath("/sdcard/DCIM/photo.jpg", 200);
```

```
        //
        Uri imageUri = Uri.fromFile(new File("/sdcard/DCIM/photo.jpg"));
        Crop.startPhotoCrop(this, imageUri, "/sdcard/DCIM/cropped.png", REQUEST_CROP);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_CROP && resultCode == RESULT_OK) {
            //                  "/sdcard/DCIM/cropped.png"
        }
    }
}
```

ملاحظات: تأكد من الإذن بالتخزين في AndroidManifest.xml و على للعمليات التشغيل وقت في التحقق من المسار المشار إليه في Crop يجب أن تعرّف drawWidth و drawHeight.

رائعة الأدوات هذه لماذا

1. **الفعالية**: طرق التعديل تمنع OutOfMemoryError عند العمل مع الصور الكبيرة.
2. **المرونة**: من واسعة مجموعة يغطي والقطع التحويلات، الرمادية، الألوان الحالات.
3. **البساطة**: إنشاء. إلى تحتاج لا سهلًا الكامل تجعل الثابتة الطرق

الخاتمة في Crop و BitmapUtils نبني كنت سواء الصور. معالجة إلى يحتاج أندرويد تطبيق لأي رائعة بداية نقطة هي قم حاولها، احتياجاتك. تغطي الشيفرة هذه المستخدم، قبل من تقطيع تضيف أو المعرض، في الصور تفيض أو صور، حرر معك! تعمل كيف لي وقل احتياجاتك، حسب بتعديلها

أدناه! أفكارك اشارك أندرويد؟ في الصور معالجة في واجهتها التي التحديات هي ما