Find The Word Mobile Application
Technical Manual

Institiúid Teicneolaíochta Cheatharlach

INSTITUTE *of* TECHNOLOGY
CARLOW

At the Heart of South Leinster

Student Name: Daniel Duda
Student Number: C00208007
Supervisor: Paul Barry
Date: 09.04.2019

Introduction

The purpose of this document is to outline the code produced by author of the Word Finder Android application.

How to install

Copy the .apk file to your android device. Usually it is necessary to allow installation of application from other sources. It can be done in device settings and can be undone after application installation. After application is installed when starting camera preview user must allow application to use camera and if user want to save word searches the user must allow write external storage permission.

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

   private static String TAG = MainActivity.class.getSimpleName().toString().trim();

   private Button openImagesButton;
   private Button whiteButton;
   private Button blackButton;
   private Button greenButton;
   private Button redButton;
   private Button blueButton;
   private Button colorPickerButton;
   private Switch flashLightSwitch;
   private Switch drawOverlaySwitch;
   private Switch drawTextSwitch;

   private Spinner spinner;

   private int requestedHeight;
   private int requestedWidth;
   private static Context appContext;

   //user input
   private EditText editText;
   //dark transparent layout to show when showing image
   private RelativeLayout back_dim_layout;

   Map<String, Integer> textColorMap = new HashMap<>();
   private TextView wordsTextView;

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

       openImagesButton = findViewById(R.id.open_images_button);
       editText = (EditText) findViewById(R.id.editText);

       appContext = getApplicationContext();


//https://developer.android.com/reference/android/graphics/drawable/ShapeDrawable.html
       whiteButton = findViewById(R.id.white_button);
       blackButton = findViewById(R.id.black_button);
       greenButton = findViewById(R.id.green_button);
       redButton = findViewById(R.id.red_button);
       blueButton = findViewById(R.id.blue_button);
```

```
        colorPickerButton = findViewById(R.id.color_picker_button);
        flashLightSwitch = (Switch) findViewById(R.id.flash_light_switch);
        drawOverlaySwitch = (Switch) findViewById(R.id.draw_overlay_switch);
        drawTextSwitch = (Switch) findViewById(R.id.draw_text_switch);

        spinner = (Spinner) findViewById(R.id.resolution_spinner);

        flashLightSwitch.setOnCheckedChangeListener(switchListener);
        drawOverlaySwitch.setOnCheckedChangeListener(switchListener);
        drawTextSwitch.setOnCheckedChangeListener(switchListener);
        back_dim_layout = (RelativeLayout)
MainActivity.this.findViewById(R.id.bac_dim_layout);

        openImagesButton.setOnClickListener(openImagesLIstener);

        wordsTextView = (TextView) findViewById(R.id.textView);
        wordsTextView.setMovementMethod(new ScrollingMovementMethod());
    }

    @Override
    public void onResume() {
        super.onResume();
        Log.d(TAG, "onResume");
        supportedResolutionsSpiner();
        drawOverlaySwitch.setChecked(true);
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent resultData) {

        if (requestCode == READ_REQUEST_CODE && resultCode == Activity.RESULT_OK) {

            Uri uri = null;
            if (resultData != null) {
                uri = resultData.getData();

                showPopupImage(uri);
            }
        }
    }
```

```
    // https://developer.android.com/reference/java/io/FileOutputStream.html
    // https://developer.android.com/guide/topics/providers/document-provider.html#open
    private Bitmap getBitmapFromUri(Uri uri) throws IOException {
        ParcelFileDescriptor parcelFileDescriptor =
                getContentResolver().openFileDescriptor(uri, "r");
        FileDescriptor fileDescriptor = parcelFileDescriptor.getFileDescriptor();
        Bitmap image = BitmapFactory.decodeFileDescriptor(fileDescriptor);
        parcelFileDescriptor.close();
        return image;
    }

    public static Context getAppContext() {
        return appContext;
    }

    public void openOcrActivity(View v) {

        //https://developer.android.com/guide/components/intents-filters
        //https://developer.android.com/guide/components/intents-common   tu zdjecia
jak otworzyc

        if(textColorMap!=null && !textColorMap.isEmpty()) {

            Intent intent = new Intent(MainActivity.this, OcrActivity.class);
            intent.putExtra("map", (Serializable) textColorMap);
            intent.putExtra("height", requestedHeight);
            intent.putExtra("width", requestedWidth);
            startActivity(intent);
        }
        else{
            Toast.makeText(MainActivity.this, "The word list is empty. You need to add
at least one word to start word search."    , Toast.LENGTH_LONG).show();
        }
    }

    private static final int READ_REQUEST_CODE = 42;

    private View.OnClickListener openImagesLIstener = new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            openImagesGallery();
        }
    };

    public void openImagesGallery() {
        //https://developer.android.com/guide/topics/providers/document-provider
        Intent intent = new Intent(Intent.ACTION_OPEN_DOCUMENT);
```

```
        // Filter to only show results that can be "opened", such as a
        // file (as opposed to a list of contacts or timezones)
        intent.addCategory(Intent.CATEGORY_OPENABLE);

        // Filter to show only images, using the image MIME data type.
        // If one wanted to search for ogg vorbis files, the type would be "audio/ogg".
        // To search for all documents available via installed storage providers,
        // it would be "*/*".
        intent.setType("image/*");

        startActivityForResult(intent, READ_REQUEST_CODE);
    }

    public void setColour(View v) {

        switch (v.getId()) {

            case R.id.white_button:

//https://developer.android.com/reference/android/view/View.html#getBackgroundTintList
()
                editText.setTextColor(whiteButton.getBackgroundTintList());
                editText.setBackgroundColor(Color.GRAY);
                break;
            case R.id.black_button:

                editText.setTextColor(blackButton.getBackgroundTintList());
                editText.setBackgroundColor(Color.WHITE);
                break;
            case R.id.green_button:

                editText.setTextColor(greenButton.getBackgroundTintList());
                editText.setBackgroundColor(Color.WHITE);
                break;
            case R.id.red_button:

                editText.setTextColor(redButton.getBackgroundTintList());
                editText.setBackgroundColor(Color.WHITE);
                break;
            case R.id.blue_button:

                editText.setTextColor(blueButton.getBackgroundTintList());
                editText.setBackgroundColor(Color.WHITE);
                break;

            case R.id.color_picker_button:

                startColorPicker();
                break;
        }
```

```
    }

    // https://github.com/yukuku/ambilwarna
    public void startColorPicker() {
        AmbilWarnaDialog colorPicker = new AmbilWarnaDialog(this, Color.WHITE, new
AmbilWarnaDialog.OnAmbilWarnaListener() {
            @Override
            public void onCancel(AmbilWarnaDialog dialog) {

            }

            @Override
            public void onOk(AmbilWarnaDialog dialog, int color) {

                colorPickerButton.setBackgroundColor(color);
                ColorDrawable color0 = (ColorDrawable)
colorPickerButton.getBackground();
                editText.setTextColor(color0.getColor());
                editText.setBackgroundColor(Color.WHITE);
                if (color == Color.WHITE) {
                    editText.setBackgroundColor(Color.GRAY);
                }
            }
        });
        colorPicker.show();
    }

    public void addWord(View v) {

        if(TextUtils.isEmpty(editText.getText().toString().trim())) {
            Toast.makeText(this, "Input text is empty.", Toast.LENGTH_SHORT).show();
            editText.setText("");
        }
        else{

            // https://developer.android.com/reference/java/util/StringTokenizer
            String[] result = editText.getText().toString().split("\\s+");

            if(result.length != 1){
                Toast.makeText(this, "One word input only allowed.",
Toast.LENGTH_SHORT).show();
            }
            else{

                textColorMap.put(editText.getText().toString().trim(),
editText.getCurrentTextColor());

                SpannableStringBuilder builder = new SpannableStringBuilder();

                for (Map.Entry<String, Integer> entry : textColorMap.entrySet()) {
```

```
                    String keyWord = entry.getKey();
                    int colorValue = entry.getValue();


//https://developer.android.com/reference/android/text/SpannableString.html

//https://developer.android.com/reference/android/text/SpannableStringBuilder
                    SpannableString spannableString = new SpannableString(keyWord);
                    spannableString.setSpan(new ForegroundColorSpan(colorValue), 0,
spannableString.length(), 0);
                    if (colorValue == Color.WHITE) {
                        //
https://developer.android.com/reference/android/text/style/BackgroundColorSpan
                        spannableString.setSpan(new BackgroundColorSpan(Color.GRAY), 0,
spannableString.length(), Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
                    }
                    builder.append(spannableString);
                    builder.append(" \n");
                }

                //https://developer.android.com/reference/android/widget/Toast.html
                Toast toast = Toast.makeText(getApplicationContext(), builder,
Toast.LENGTH_SHORT);
                toast.setGravity(Gravity.TOP | Gravity.CENTER_HORIZONTAL, 0, 0);
                toast.show();

                wordsTextView.setText("");
                wordsTextView.append(builder);
                editText.setText("");
            }
        }
    }

    CompoundButton.OnCheckedChangeListener switchListener = new
CompoundButton.OnCheckedChangeListener() {
        public void onCheckedChanged(CompoundButton v, boolean isChecked) {
            switch (v.getId()) {
                case R.id.flash_light_switch:

                    if (isChecked) {
                        if
(MainActivity.this.getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA_
FLASH)) {
                            CameraSource.setUsingTorchMode(Boolean.TRUE);
                        } else {
                            Toast.makeText(MainActivity.this, "Your device don't have
flash light", Toast.LENGTH_SHORT).show();
                            flashLightSwitch.setChecked(false);
                        }
```

```
                    } else {

                        CameraSource.setUsingTorchMode(Boolean.FALSE);
                    }
                    break;
                case R.id.draw_overlay_switch:
                    if (isChecked) {
                        TextGraphic.setDrawOverlay(Boolean.TRUE);
                    } else {
                        TextGraphic.setDrawOverlay(Boolean.FALSE);
                    }

                    if (drawOverlaySwitch.isChecked() == false &&
drawTextSwitch.isChecked() == false) {
                        Toast.makeText(MainActivity.this, "DrawOverlay and DrawText are
not set. To find word you need at least one set.  ", Toast.LENGTH_SHORT).show();
                    }
                    break;
                case R.id.draw_text_switch:
                    if (isChecked) {
                        TextGraphic.setDrawText(Boolean.TRUE);

                    } else {
                        TextGraphic.setDrawText(Boolean.FALSE);
                    }
                    if (drawOverlaySwitch.isChecked() == false &&
drawTextSwitch.isChecked() == false) {
                        Toast.makeText(MainActivity.this, "DrawOverlay and DrawText are
not set. To find word you need at least one set.  ", Toast.LENGTH_SHORT).show();
                    }
                    break;
            }// endswitch
        }
    };

    public void showPopupImage(Uri uri) {

        //
https://developer.android.com/training/camera/photobasics.html#TaskScalePhoto
        //
https://developer.android.com/reference/android/graphics/BitmapFactory.Options#inSampl
eSize

        LayoutInflater layoutInflater = (LayoutInflater)
MainActivity.this.getSystemService(MainActivity.LAYOUT_INFLATER_SERVICE);
        final View popupView = layoutInflater.inflate(R.layout.image_popup, null);

        final PopupWindow popupWindow = new PopupWindow(popupView,
ViewGroup.LayoutParams.WRAP_CONTENT, WindowManager.LayoutParams.WRAP_CONTENT);
        popupWindow.setFocusable(true);
```

```
        popupWindow.setBackgroundDrawable(new ColorDrawable());

        Bitmap bitmap = null;
        try {
            bitmap = getBitmapFromUri(uri);
        } catch (IOException e) {
            e.printStackTrace();
        }

        ImageView imageView = (ImageView)
popupView.findViewById(R.id.picture_imageView);

        imageView.setImageBitmap(bitmap);

        popupWindow.showAtLocation(MainActivity.this.findViewById(R.id.ad_word_button),
Gravity.CENTER, 0, 0);

        popupWindow.setOnDismissListener(new PopupWindow.OnDismissListener() {
            @Override
            public void onDismiss() {
                back_dim_layout.setVisibility(View.GONE);
                openImagesGallery();
            }
        });

        back_dim_layout.setVisibility(View.VISIBLE);
    }

    private void supportedResolutionsSpiner() {

        List<SupportedResolution> supportedResolutionList = new
ArrayList<SupportedResolution>();

        SupportedResolution supportedResolution = new SupportedResolution(640, 480);
        supportedResolutionList.add(supportedResolution);

        supportedResolution = new SupportedResolution(960, 720);
        supportedResolutionList.add(supportedResolution);

        supportedResolution = new SupportedResolution(1024, 768);
        supportedResolutionList.add(supportedResolution);

        supportedResolution = new SupportedResolution(1280, 720);
        supportedResolutionList.add(supportedResolution);

        supportedResolution = new SupportedResolution(1280, 960);
        supportedResolutionList.add(supportedResolution);

        supportedResolution = new SupportedResolution(1280, 1024);
```

```
        supportedResolutionList.add(supportedResolution);

        supportedResolution = new SupportedResolution(1920, 1080);
        supportedResolutionList.add(supportedResolution);

        // https://developer.android.com/guide/topics/ui/controls/spinner.html
        ArrayAdapter<SupportedResolution> dataAdapter = new
ArrayAdapter<SupportedResolution>(this, android.R.layout.simple_spinner_item,
supportedResolutionList);

dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner.setAdapter(dataAdapter);
        spinner.setOnItemSelectedListener(onItemSelectedListener);
        spinner.setSelection(3);
    }

    OnItemSelectedListener onItemSelectedListener = new OnItemSelectedListener() {

        @Override
        public void onItemSelected(AdapterView<?> adapterView, View view, int i, long
l) {

            SupportedResolution supportedResolution = (SupportedResolution)
adapterView.getItemAtPosition(i);
            requestedHeight = supportedResolution.getHeight();
            requestedWidth = supportedResolution.getWidth();
        }

        @Override
        public void onNothingSelected(AdapterView<?> adapterView) {
        }
    };
}
```

OcrActivity.java

```
public class OcrActivity extends AppCompatActivity {

    private static Context ocrActivityContextContext;

    private static final int GET_PERMISSION_REQUEST = 0;
    private static final int WRITE_EXTERNAL_STORAGE_PERMISSION = 1;
    private CameraSource cameraSource = null;
    private CameraSourcePreview preview;
    private GraphicOverlay graphicOverlay;

    private static String TAG = MainActivity.class.getSimpleName().toString().trim();
    private GestureDetectorCompat gestureDetector;

    // map store data(word(string), color(int)) send via intent from mainActivity
    public static Map<String, Integer> mapOcr = new HashMap<>();
    int requestedHeight;
    int requestedWidth;

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

        //keep the screen active
        //https://developer.android.com/training/scheduling/wakelock
        //getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);

        ocrActivityContextContext = this;

//https://developer.android.com/training/camera/cameradirect.html#TaskRestartPreview
        //https://developer.android.com/guide/topics/media/camera.html#custom-camera
        //https://developer.android.com/training/camera/cameradirect#java
        Intent intent = getIntent();
        mapOcr = (Map<String, Integer>) intent.getSerializableExtra("map");
        requestedHeight = intent.getIntExtra("height", 1080);
        requestedWidth = intent.getIntExtra("width", 1920);

        graphicOverlay = (GraphicOverlay) findViewById(R.id.graphics_overlay);
        preview = (CameraSourcePreview) findViewById(R.id.camera_source_preview);

        if (getPermissions()) {
            createCameraSource();
            startCameraSource();
        }
        gestureDetector = new GestureDetectorCompat(this, new MyGestureListener());
```

```
    }

    @Override
    public void onResume() {
        super.onResume();
        Log.d(TAG, "onResume");
        startCameraSource();
    }

    /**
     * Stops the camera.
     */
    @Override
    protected void onPause() {
        super.onPause();
        if (preview != null) {
            preview.stop();
        }
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        if (cameraSource != null) {
            cameraSource.release();
        }
    }

    private void createCameraSource() {

        if (cameraSource == null) {
            cameraSource = new CameraSource(this, graphicOverlay);
            cameraSource.setRequestedPreviewHeight(requestedHeight);
            cameraSource.setRequestedPreviewWidth(requestedWidth);
            cameraSource.setFacing(CameraSource.CAMERA_FACING_BACK);
            cameraSource.setMachineLearningFrameProcessor(new
TextRecognitionProcessor());
        }
    }

    private void startCameraSource() {
        if (cameraSource != null) {
            try {
                if (preview == null) {
                    Log.d(TAG, "resume: Preview is null");
                }
                if (graphicOverlay == null) {
                    Log.d(TAG, "resume: graphOverlay is null");
                }
                preview.start(cameraSource, graphicOverlay);
```

```
            } catch (IOException e) {
                Log.e(TAG, "Unable to start camera source.", e);
                cameraSource.release();
                cameraSource = null;
            }
        }
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        this.gestureDetector.onTouchEvent(event);
        return super.onTouchEvent(event);
    }

    //https://developer.android.com/training/gestures/detector#java

//https://developer.android.com/reference/android/view/GestureDetector.OnDoubleTapList
ener
    class MyGestureListener extends GestureDetector.SimpleOnGestureListener {
        private static final String DEBUG_TAG = "Gestures";

        @Override
        public boolean onDoubleTap(MotionEvent event) {

            if (checkWriteExternalStoragePermission()) {

                Bitmap bitmap = takeScreenshot(preview);
                saveBitmap(bitmap);
            } else {
                Toast toast = Toast.makeText(OcrActivity.this, "Write external
permissions required to save screenshoot go to Settings and grand permission ",
Toast.LENGTH_LONG);
                toast.show();
            }
            return true;
        }
    }
    //  ********************************************** SCRENSHOOT
**************************************************************************
    public void saveBitmap(Bitmap bitmap) {


//https://developer.android.com/reference/android/os/Environment.html#getExternalStora
gePublicDirectory%28java.lang.String%29
        File imageFolder = new File(Environment.getExternalStorageDirectory() +
File.separator + "Pictures/wordFinder/");
        if (!imageFolder.exists()) {
            imageFolder.mkdirs();
        }
```

```
        //https://developer.android.com/reference/java/text/SimpleDateFormat
        SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("yyyy-MM-dd_HH-mm-ss");
        String currentDateandTime = simpleDateFormat.format(new Date());
        String imageName = currentDateandTime + ".jpg";

        File imagePath = new File(imageFolder, imageName);
        FileOutputStream fileOutputStream;

        try {
            //https://developer.android.com/reference/java/io/FileOutputStream
            fileOutputStream = new FileOutputStream(imagePath);
            bitmap.compress(Bitmap.CompressFormat.JPEG, 50, fileOutputStream);
            Toast.makeText(this, "Screenshot saved", Toast.LENGTH_SHORT).show();
        } catch (FileNotFoundException e) {
            //e.printStackTrace();
            Log.e("FileNotFoundException", e.getMessage(), e);
        }
        MediaScannerConnection.scanFile(OcrActivity.this, new
String[]{imagePath.getPath()}, null, null);
    }


//https://codelabs.developers.google.com/codelabs/sceneform-intro/index.html?index=..%
2F..%2Fio2018#14
    //https://codelabs.developers.google.com/codelabs/sceneform-intro/index.html#0
    public static Bitmap takeScreenshot(View view) {

        int width = view.getWidth() / 2;
        int height = view.getHeight() / 2;

        // https://developer.android.com/training/camera/photobasics
        Bitmap bitmap = Bitmap.createBitmap(width, height, Bitmap.Config.ARGB_8888);
        Canvas canvas = new Canvas(bitmap);

        Drawable backgroundDrawable = view.getBackground();

        if (backgroundDrawable != null) {
            backgroundDrawable.draw(canvas);
        } else {
            canvas.drawColor(Color.WHITE);
        }
        view.draw(canvas);//Manually render this view (and all of its children) to the
given Canvas
        return bitmap;
    }

//********PERMISSIONS******

    // https://developer.android.com/training/permissions/requesting.html
```

```
    private boolean getPermissions() {

        if (ContextCompat.checkSelfPermission(OcrActivity.this, CAMERA) ==
PackageManager.PERMISSION_GRANTED) {
            return Boolean.TRUE;
        } else {
            ActivityCompat.requestPermissions(OcrActivity.this, new String[]{CAMERA,
WRITE_EXTERNAL_STORAGE}, GET_PERMISSION_REQUEST);
        }
        return Boolean.FALSE;
    }

    private boolean checkWriteExternalStoragePermission() {

        if (ContextCompat.checkSelfPermission(OcrActivity.this, WRITE_EXTERNAL_STORAGE)
== PackageManager.PERMISSION_GRANTED) {
            return Boolean.TRUE;
        } else {
            return Boolean.FALSE;
        }
    }

    private boolean getWriteExternalStoragePermission() {

        if (ContextCompat.checkSelfPermission(OcrActivity.this, WRITE_EXTERNAL_STORAGE)
== PackageManager.PERMISSION_GRANTED) {
            return Boolean.TRUE;
        } else {
            ActivityCompat.requestPermissions(OcrActivity.this, new
String[]{WRITE_EXTERNAL_STORAGE}, WRITE_EXTERNAL_STORAGE_PERMISSION);
        }
        return Boolean.FALSE;
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        // check if user allowed camera permissions neccesary for application to work
        if (requestCode == GET_PERMISSION_REQUEST) {
            if (grantResults[0] != PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(getApplicationContext(), "Camera permission denied
application will not work", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "Camera permission granted",
Toast.LENGTH_SHORT).show();
                createCameraSource();
                startCameraSource();
```

```
            }// check if user allowed write external storage permissions not necessary
for application to work but needed to save screenshots, can also be granted from
settings
            if (grantResults[1] != PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(this, "Application need external storage to save
screenshots", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "External storage permission granted you can save
screenshots", Toast.LENGTH_SHORT).show();

            }
        }
        // handle write external storage request
        if (requestCode == WRITE_EXTERNAL_STORAGE_PERMISSION) {
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(this, "External storage permission granted you can save
screenshots", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "Application need external storage to save
screenshots", Toast.LENGTH_SHORT).show();
            }
        }
    }
    //**********PERMISSIONS***********

    public static Context getOcrActivityContext() {
        return ocrActivityContextContext;
    }
}
```

CameraSource.java

Original file:
https://github.com/firebase/quickstart-android/blob/master/mlkit/app/src/main/java/com/google/firebase/samples/apps/mlkit/common/CameraSource.java

Added/changed code:

```
if (usingTorchMode == Boolean.TRUE) {
    parameters.setFlashMode(Camera.Parameters.FLASH_MODE_TORCH);
} else {
    parameters.setFlashMode(Camera.Parameters.FLASH_MODE_OFF);
}
public SurfaceTexture getDummySurfaceTexture() {
    return this.dummySurfaceTexture;
}
public float getRequestedFps() {
    return requestedFps;
}
public void setRequestedFps(float requestedFps) {
    this.requestedFps = requestedFps;
}
public int getRequestedPreviewWidth() {
    return requestedPreviewWidth;
}
public void setRequestedPreviewWidth(int requestedPreviewWidth) {
    this.requestedPreviewWidth = requestedPreviewWidth;
}
public int getRequestedPreviewHeight() {
    return requestedPreviewHeight;
}
public void setRequestedPreviewHeight(int requestedPreviewHeight) {
    this.requestedPreviewHeight = requestedPreviewHeight;
}
public boolean isRequestedAutoFocus() {
    return requestedAutoFocus;
}
public void setRequestedAutoFocus(boolean requestedAutoFocus) {
    this.requestedAutoFocus = requestedAutoFocus;
}
private static Boolean usingTorchMode = Boolean.FALSE;
public static void setUsingTorchMode(Boolean mode) {
```

```
    usingTorchMode = mode;
}
public static Boolean getUsingTorchModeTorchMOde() {
    return usingTorchMode;
}
```

CameraSourcePreview.java

Original file:https://github.com/firebase/quickstart-android/blob/master/mlkit/app/src/main/java/com/google/firebase/samples/apps/mlkit/common/CameraSourcePreview.java

Added/changed code:

```
//int childHeight = (int) (((float) layoutWidth / (float) width) * height);
int childHeight = layoutHeight;

 //childWidth = (int) (((float) layoutHeight / (float) height) * width);
 childWidth = layoutWidth;
```

GraphicOverlay.java

Original file:
https://github.com/firebase/quickstart-android/blob/master/mlkit/app/src/main/java/com/google/firebase/samples/apps/mlkit/common/GraphicOverlay.java

No changes made.

BItmapUtils.java

Original file:
https://github.com/firebase/quickstart-android/blob/master/mlkit/app/src/main/java/com/google/firebase/samples/apps/mlkit/common/BitmapUtils.java

No changes made.

CameraImageGraphic.java

Original file:
https://github.com/firebase/quickstart-android/blob/master/mlkit/app/src/main/java/com/google/firebase/samples/apps/mlkit/common/CameraImageGraphic.java

No changes made.

FrameMetadata.java

Original file:
https://github.com/firebase/quickstart-android/blob/master/mlkit/app/src/main/java/com/google/firebase/samples/apps/mlkit/common/FrameMetadata.java

No changes made.

VisionImageProcessor.java

Original file:
https://github.com/firebase/quickstart-android/blob/master/mlkit/app/src/main/java/com/google/firebase/samples/apps/mlkit/common/VisionImageProcessor.java

No changes made.

TextGraphic.java

Original file:
https://github.com/firebase/quickstart-android/blob/master/mlkit/app/src/main/java/com/google/firebase/samples/apps/mlkit/java/textrecognition/TextGraphic.java

Added/changed code:

```
private static boolean drawOverlay;
private static boolean drawText;

TextGraphic(GraphicOverlay overlay, FirebaseVisionText.Element text, int textColor) {
   super(overlay);
   this.text = text;

   //if(drawOverlay){
   rectPaint = new Paint();
   rectPaint.setColor(textColor);
   rectPaint.setStyle(Paint.Style.STROKE);
   rectPaint.setStrokeWidth(STROKE_WIDTH);
   // }

   textPaint = new Paint();
   textPaint.setColor(textColor);
   textPaint.setTextSize(TEXT_SIZE);
}


@Override
public void draw(Canvas canvas) {
   if (text == null) {
       throw new IllegalStateException("Attempting to draw a null text.");
   }
```

```
    RectF rect = new RectF(text.getBoundingBox());

    // Draws the bounding box around the TextBlock.
    rect.left = translateX(rect.left);
    rect.top = translateY(rect.top);
    rect.right = translateX(rect.right);
    rect.bottom = translateY(rect.bottom);

    if (drawOverlay && drawText) {
        canvas.drawRect(rect, rectPaint);
        canvas.drawText(text.getText(), rect.left, rect.bottom, textPaint);
    } else if (drawText) {
        // Renders the text at the bottom of the box.
        canvas.drawText(text.getText(), rect.left, rect.bottom, textPaint);
    } else if (drawOverlay) {
        canvas.drawRect(rect, rectPaint);
    }

}

public static void setDrawOverlay(Boolean draw) {

    drawOverlay = draw;
}

public static void setDrawText(Boolean draw) {

    drawText = draw;
}
```

TextRecognitionProcessor.java

Original file:
https://github.com/firebase/quickstart-android/blob/master/mlkit/app/src/main/java/com/google/firebase/samples/apps/mlkit/java/textrecognition/TextRecognitionProcessor.java

Added/changed code:

```
private Map<String, Integer> map = OcrActivity.mapOcr;
private String word;
private int color;

private boolean checkMapSize(){
```

```
    if (map.size() == 1){
        word = map.entrySet().iterator().next().getKey();
        color = map.entrySet().iterator().next().getValue();
        return Boolean.TRUE;
    }
    return Boolean.FALSE;
}

for (int k = 0; k < elements.size(); k++) {

    if(checkMapSize()){

        if (elements.get(k).getText().equalsIgnoreCase(word)) {

            GraphicOverlay.Graphic textGraphic = new TextGraphic(graphicOverlay,
elements.get(k), color);
            graphicOverlay.add(textGraphic);
        }
    }
    else{
        for (Map.Entry<String, Integer> entry : map.entrySet()) { // to chyba
najszybsze

            System.out.println(entry.getKey() + "/" + entry.getValue());

            if (elements.get(k).getText().equalsIgnoreCase(entry.getKey())) {

                GraphicOverlay.Graphic textGraphic = new TextGraphic(graphicOverlay,
elements.get(k), entry.getValue());
                graphicOverlay.add(textGraphic);
            }
        }
    }

}
```

VisionProcessorBase.java

Original file:
https://github.com/firebase/quickstart-android/blob/master/mlkit/app/src/main/java/com/google/firebase/samples/apps/mlkit/java/VisionProcessorBase.java

No changes made.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">


    <android.support.constraint.ConstraintLayout
        android:id="@+id/main_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <Button
            android:id="@+id/color_picker_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="8dp"
            android:layout_marginBottom="16dp"
            android:onClick="setColour"
            android:text="Color Picker"
            android:textSize="14sp"
            app:layout_constraintBottom_toTopOf="@+id/green_button"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.483"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <Button
            android:id="@+id/white_button"
            android:layout_width="39dp"
            android:layout_height="41dp"
            android:layout_marginStart="8dp"
            android:layout_marginTop="80dp"
            android:layout_marginEnd="8dp"
            android:background="@drawable/oval_shape"
            android:backgroundTint="@color/white"
            android:onClick="setColour"
            app:layout_constraintEnd_toStartOf="@+id/black_button"
            app:layout_constraintHorizontal_chainStyle="packed"
            app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/black_button"
    android:layout_width="42dp"
    android:layout_height="42dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="8dp"
    android:background="@drawable/oval_shape"
    android:backgroundTint="@color/black"
    android:onClick="setColour"
    app:layout_constraintEnd_toStartOf="@+id/green_button"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toEndOf="@+id/white_button"
    app:layout_constraintTop_toTopOf="parent" />


<Button
    android:id="@+id/red_button"
    android:layout_width="42dp"
    android:layout_height="42dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="8dp"
    android:background="@drawable/oval_shape"
    android:backgroundTint="@color/red"
    android:onClick="setColour"
    app:layout_constraintEnd_toStartOf="@+id/blue_button"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toEndOf="@+id/green_button"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/blue_button"
    android:layout_width="42dp"
    android:layout_height="42dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="8dp"
    android:background="@drawable/oval_shape"
    android:backgroundTint="@color/blue"
    android:onClick="setColour"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toEndOf="@+id/red_button"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/green_button"
```

```
        android:layout_width="42dp"
        android:layout_height="42dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="80dp"
        android:layout_marginEnd="8dp"
        android:background="@drawable/oval_shape"
        android:backgroundTint="@color/green"
        android:onClick="setColour"
        app:layout_constraintEnd_toStartOf="@+id/red_button"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toEndOf="@+id/black_button"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="0dp"
        android:layout_height="60dp"
        android:layout_marginTop="136dp"
        android:layout_marginEnd="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toStartOf="@+id/ad_word_button"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintHorizontal_chainStyle="spread"
        app:layout_constraintHorizontal_weight="4"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageButton
        android:id="@+id/ad_word_button"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="136dp"
        android:onClick="addWord"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/button2"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintHorizontal_weight="1"
        app:layout_constraintStart_toEndOf="@+id/editText"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@android:drawable/ic_input_add" />

    <Button
        android:id="@+id/button2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="136dp"
        android:onClick="openOcrActivity"
        android:text="Start"
        android:textSize="18sp"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintHorizontal_weight="2"
    app:layout_constraintStart_toEndOf="@+id/ad_word_button"
    app:layout_constraintTop_toTopOf="parent" />/>

<Button
    android:id="@+id/open_images_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="56dp"
    android:layout_marginEnd="8dp"
    android:text="Open Images"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.978"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button2" />


<TextView
    android:id="@+id/textView"
    android:layout_width="137dp"
    android:layout_height="185dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="36dp"
    android:scrollbars="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/resolution_spinner" />


<Switch
    android:id="@+id/flash_light_switch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="28dp"
    android:layout_marginBottom="172dp"
    android:text="Fashlight"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/open_images_button" />

<Switch
    android:id="@+id/draw_overlay_switch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
```

```
        android:layout_marginEnd="4dp"
        android:layout_marginBottom="116dp"
        android:text="Draw Overlay"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/flash_light_switch" />

    <Switch
        android:id="@+id/draw_text_switch"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="24dp"
        android:layout_marginBottom="64dp"
        android:text="Draw Text"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/draw_overlay_switch" />


    <Spinner
        android:id="@+id/resolution_spinner"
        android:layout_width="150dp"
        android:layout_height="38dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView3" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="20dp"
        android:text="Preview resolution:"
        android:textColor="@color/black"
        android:textSize="16sp"
        app:layout_constraintEnd_toEndOf="@+id/resolution_spinner"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText" />


</android.support.constraint.ConstraintLayout>

<RelativeLayout
    android:id="@+id/bac_dim_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentStart="true"
```

```
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:background="#C0000000"
        android:visibility="gone"></RelativeLayout>

</RelativeLayout>
```

Activity_ocr.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:id="@+id/ocr_layout"
   tools:context=".MainActivity">

   <ie.student.wordfinder.CameraSourcePreview
       android:id="@+id/camera_source_preview"
       android:layout_width="match_parent"
       android:layout_height="match_parent"
       app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintEnd_toEndOf="parent"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintTop_toTopOf="parent">

       <ie.student.wordfinder.GraphicOverlay
           android:id="@+id/graphics_overlay"
           android:layout_width="match_parent"
           android:layout_height="match_parent" />

   </ie.student.wordfinder.CameraSourcePreview>

</android.support.constraint.ConstraintLayout>
```

Image_popup.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="wrap_content">
```

```
    <ImageView
        android:id="@+id/picture_imageView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
     />


</RelativeLayout>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ie.student.wordfinder">

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" />

    <permission android:name="android.permission.FLASHLIGHT"
        android:permissionGroup="android.permission-group.HARDWARE_CONTROLS"
        android:protectionLevel="normal" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".OcrActivity" android:theme="@style/FullscreenTheme">

        </activity>
        <activity android:name=".MainActivity"
            android:windowSoftInputMode="adjustNothing">

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```