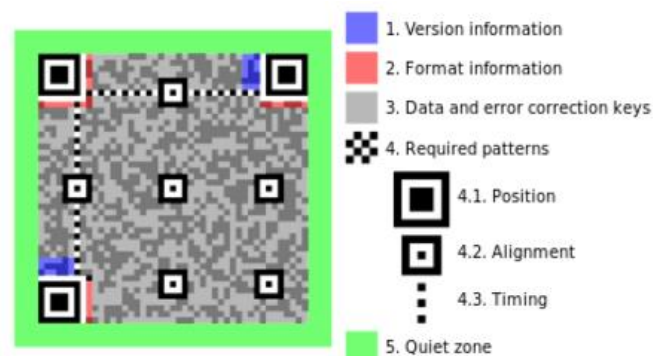


QR Code

1 Barcode basic knowledge

A barcode is an optical machine-readable representation of data relating to the object to which it is attached. Originally barcodes systematically represented data by varying the widths and spacings of parallel lines, and may be referred to as linear or one-dimensional (1D). Later two-dimensional (2D) codes were developed, using rectangles, dots, hexagons and other geometric patterns in two dimensions, usually called barcodes although they do not use bars as such. Barcodes originally were scanned by special optical scanners called barcode readers. Later applications software became available for devices that could read images, such as smartphones with cameras.

Nowadays the most applied barcode is QR code, integrated in applications like Wechat, Paypal, Alipay etc. QR code (abbreviated from Quick Response Code) is the trademark for a type of matrix barcode (or two-dimensional barcode) first designed for the automotive industry in Japan. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte/binary, and kanji) to efficiently store data; extensions may also be used. The QR Code system became popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC barcodes. Applications include product tracking, item identification, time tracking, document management, and general marketing. A QR code consists of black modules (square dots) arranged in a square grid on a white background, which can be read by an imaging device (such as a camera, scanner, etc.) and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data are then extracted from patterns that are present in both horizontal and vertical components of the image.



The above figure is a representation of the structure of QR code.

2 Get Prepared

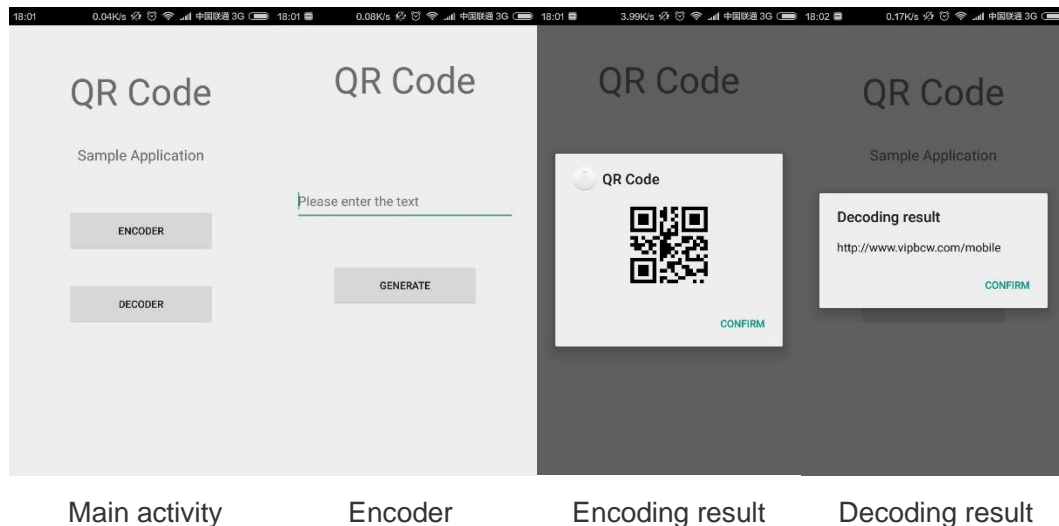
2.1 ZXing project

ZXing ("zebra crossing") is an open-source, multi-format 1D/2D barcode image processing library implemented in Java, with ports to other languages. Zxing provides the tool of processing different barcodes, such as QR Code, Data Matrix, Aztec and PDF 417.

We need to import these files to implement the QR Code encoder and decoder.

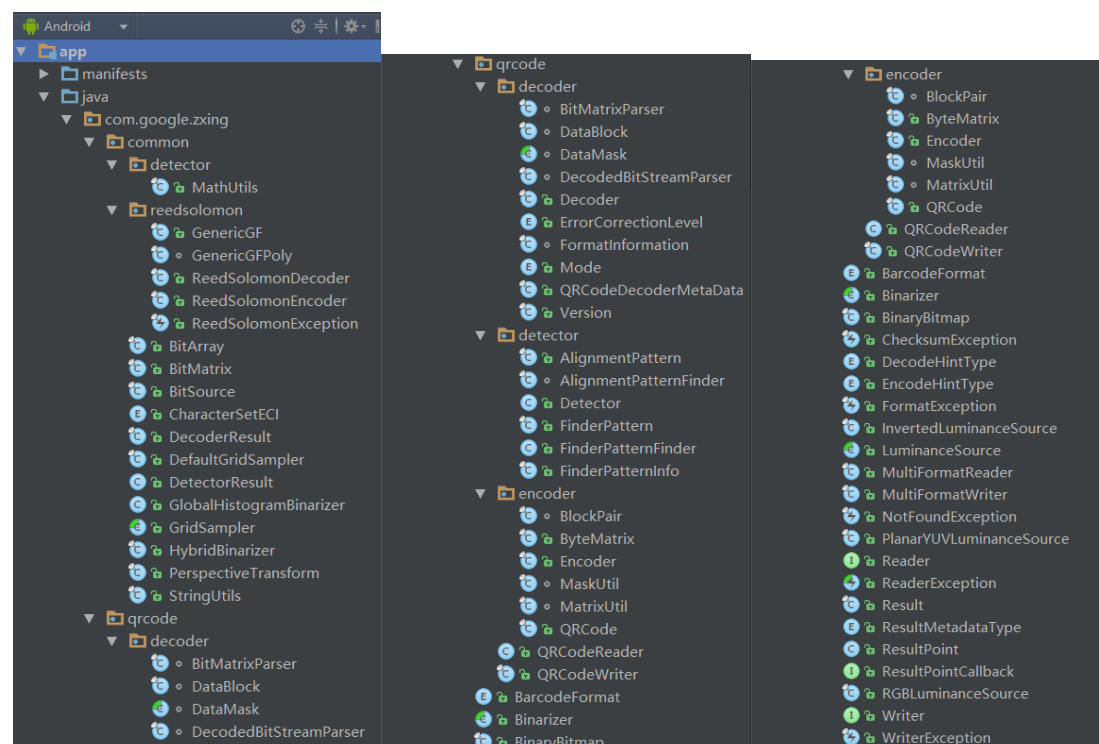
2.2 Introduction

In this experiment, we need to develop an application with the capability of QR code encoding and decoding. The expected application is show as below:



2.3 Library import

The opensource library can be downloaded from <https://github.com/zxing/zxing>. Then need to add these .jar files into our project. We can create new packages in the project to manage these files more easily. We need to paste all the needed files into the project first. The whole files needed is listed as follows:



2.4 Project source code

After importing source files, we can begin to write our own files. The project consists of three main activities, MainActivity.java, TestDecoder.java, and TestEncoder.java. MainActivity is the entrance of the two functions. TestDecoder will open the camera and decode the captured QR Code. Encoder transforms input contents into a QR Code.

The main activity and its layout file are listed below:

MainActivity.java

```
package edu.sjtu.zhusy54.qrcode;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (resultCode == RESULT_OK) {
            Bundle bundle = data.getExtras();
            String scanResult = bundle.getString("result");
            new AlertDialog.Builder(MainActivity.this)
                .setTitle("Decoding result")
                .setMessage(scanResult)
                .setPositiveButton("Confirm", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        dialog.dismiss();
                    }
                }).show();
        }
    }

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

        Button encoder = (Button)findViewById(R.id.btn_encoder);
```

```

        Button decoder = (Button)findViewById(R.id.btn_decoder);
        View.OnClickListener myListener = new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent;
                switch (v.getId()){
                    case R.id.btn_encoder:
                        intent = new Intent(MainActivity.this,
TestEncoder.class);
                        startActivity(intent);
                        break;
                    case R.id.btn_decoder:
                        intent = new Intent(MainActivity.this,
TestDecoder.class);
                        startActivityForResult(intent, 0);
                        break;
                    default:
                        break;
                }
            }
        };
        encoder.setOnClickListener(myListener);
        decoder.setOnClickListener(myListener);
    }
}

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<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:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="edu.sjtu.zhusy54.qrcode.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:paddingTop="40dp"

```

```

        android:text="QR Code"
        android:textSize="50dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="40dp"
        android:layout_gravity="center_horizontal"
        android:text="Sample Application"
        android:textSize="20dp"
    />
    <Button
        android:layout_width="200dp"
        android:layout_height="60dp"
        android:id="@+id/btn_encoder"
        android:layout_marginTop="60dp"
        android:layout_gravity="center_horizontal"
        android:text="Encoder"/>
    <Button
        android:layout_width="200dp"
        android:layout_height="60dp"
        android:id="@+id/btn_decoder"
        android:layout_marginTop="40dp"
        android:layout_gravity="center_horizontal"
        android:text="Decoder"/>
</LinearLayout>

```

The encoder and its layout file are listed below:

TestEncoder.java

```

package edu.sjtu.zhusy54.qrcode;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import com.google.zxing.BarcodeFormat;
import com.google.zxing.MultiFormatWriter;

```

```

import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;

/**
 * Created by Syman-Z on 2016/2/25.
 */
public class TestEncoder extends Activity {
    EditText textContent;

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

        textContent = (EditText)findViewById(R.id.gen_content);
        Button genBtn = (Button)findViewById(R.id.btn_generate);
        genBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    String contentString = textContent.getText().toString();
                    if (!contentString.equals("")) {
                        BitMatrix matrix = new
MultiFormatWriter().encode(contentString,
                                BarcodeFormat.QR_CODE, 300, 300);
                        int width = matrix.getWidth();
                        int height = matrix.getHeight();
                        int[] pixels = new int[width * height];

                        for (int y = 0; y < height; y++) {
                            for (int x = 0; x < width; x++) {
                                if (matrix.get(x, y)) {
                                    pixels[y * width + x] = Color.BLACK;
                                }
                            }
                        }

                        Bitmap bitmap = Bitmap.createBitmap(width, height,
                                Bitmap.Config.ARGB_8888);
                        bitmap.setPixels(pixels, 0, width, 0, 0, width, height);
                        ImageView image1 = new ImageView(TestEncoder.this);
                        image1.setImageBitmap(bitmap);
                        new AlertDialog.Builder(TestEncoder.this)
                                .setTitle("QR Code")
                                .setIcon(android.R.drawable.ic_dialog_info)
                                .setView(image1)

```

```

        .setPositiveButton("Confirm", new
DialogInterface.OnClickListener(){
            @Override
            public void onClick(DialogInterface dialog,
int which) {
                dialog.dismiss();
            }
        })
        .show();
//        Bitmap qrCodeBitmap =
EncodingHandler.createQRCode(contentString, 350);
//        qrImgImageView.setImageBitmap(qrCodeBitmap);
    }else {
        Toast.makeText(TestEncoder.this, "Text can not be empty",
Toast.LENGTH_SHORT).show();
    }

    } catch (WriterException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
});
}
}

```

encoder.xml

```

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:paddingTop="40dp"
        android:paddingBottom="80dp"
        android:text="QR Code"
        android:textSize="50dp"/>

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"

```

```

        android:paddingTop="40dp"
        android:id="@+id/gen_content"
        android:layout_gravity="center_horizontal"
        android:hint="Please enter the text"/>

<Button
    android:layout_width="200dp"
    android:layout_height="60dp"
    android:id="@+id/btn_generate"
    android:layout_marginTop="60dp"
    android:layout_gravity="center_horizontal"
    android:text="Generate"/>

</LinearLayout>

```

The decoder and its layout file are listed below:

TestDecoder.java

```

package edu.sjtu.zhusy54.qrcode;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.res.Configuration;
import android.graphics.PixelFormat;
import android.hardware.Camera;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.util.SparseArray;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.view.Window;
import android.view.WindowManager;

import com.google.zxing.BinaryBitmap;
import com.google.zxing.PlanarYUVLuminanceSource;
import com.google.zxing.Reader;
import com.google.zxing.Result;
import com.google.zxing.common.HybridBinarizer;
import com.google.zxing.qrcode.QRCodeReader;

import java.util.List;

/**

```



```

* Created by Syman-Z on 2016/2/25.
*/
public class TestDecoder extends Activity implements SurfaceHolder.Callback {
    // 定义对象
    private SurfaceView mSurfaceview = null; // SurfaceView 对象: (视图组件)视频
    显示
    private SurfaceHolder mSurfaceHolder = null; // SurfaceHolder 对象: (抽象接
    口)SurfaceView 支持类
    private Camera mCamera =null; // Camera 对象, 相机预览
    private boolean bIfPreview=false, storeFlag=true, stored=false;
    private int mPreviewWidth=720, mPreviewHeight=1280, colorNum, unitC;
    // private int mPreviewWidth=480, mPreviewHeight=800, colorNum, unitC;

    //用于接收解码好的数据
    private Handler uiHandler;
    private Bundle mBundle;
    private boolean isFirstFrame=true;
    private byte[] tmpBytes;
    private int frameNum, totalFrm, byteLenth, frameCnt=0;
    // Map<Integer, byte[]> myMap = new SparseArray<Integer, byte[]>();
    // SparseArray<int[]> myMap = new SparseArray<int[]>();
    SparseArray<byte[]> myMap = new SparseArray<byte[]>();

    static Context context;

    // InitSurfaceView
    private void initSurfaceView()
    {
        mSurfaceview = (SurfaceView) this.findViewById(R.id.preview_view);
        mSurfaceHolder = mSurfaceview.getHolder(); // 绑定 SurfaceView, 取得
        SurfaceHolder 对象
        mSurfaceHolder.addCallback(TestDecoder.this); // SurfaceHolder 加入回调接
        口
        //mSurfaceHolder.setFixedSize(720, 1280); // 预览大小設置
        // mSurfaceHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);// 設置顯
        示器類型, setType 必须设置
        mSurfaceHolder.setFormat(PixelFormat.TRANSPARENT);
    }

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

```

```

        //无 title
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        //设置全屏
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.decoder);

        initSurfaceView();

    }

    /* 【SurfaceHolder.Callback 回调函数】 */
    public void surfaceCreated(SurfaceHolder holder)
    // SurfaceView 启动时/初次实例化，预览界面被创建时，该方法被调用。
    {
        mCamera = Camera.open();// 开启摄像头（2.3 版本后支持多摄像头,需传入参数）
        try
        {
            Log.i("TAG", "SurfaceHolder.Callback: surface Created");
            mCamera.setPreviewDisplay(mSurfaceHolder);//set the surface to be
used for live preview
        }catch (Exception ex)
        {
            if(null != mCamera)
            {
                mCamera.release();
                mCamera = null;
            }
            Log.i("TAG"+"initCamera", ex.getMessage());
        }
        mCamera.setPreviewCallback(new Camera.PreviewCallback(){

            @Override
            public void onPreviewFrame(byte[] data, Camera camera) {
//                try {
//                    System.out.println("decoding frame!!!");
//                    MultiFormatReader formatReader = new MultiFormatReader();
//                    LuminanceSource source = new PlanarYUVLuminanceSource(data,
720, 1280, 60, 340, 600, 600, false);
//                    Binarizer binarizer = new HybridBinarizer(source);
//                    BinaryBitmap binaryBitmap = new BinaryBitmap(binarizer);
//                    Map hints = new HashMap();
//                    hints.put(EncodeHintType.CHARACTER_SET, "UTF-8");
//                    Result result = formatReader.decode(binaryBitmap, hints);

```

```

//          Intent resultIntent = new Intent();
//          Bundle bundle = new Bundle();
//          bundle.putString("result", result.toString());
//          resultIntent.putExtras(bundle);
//          TestDecoder.this.finish();
////          this.setResult(RESULT_OK, resultIntent);
//      } catch (Exception e) {
//          e.printStackTrace();
//      }
        int previewWidth =
camera.getParameters().getPreviewSize().width;
        int previewHeight =
camera.getParameters().getPreviewSize().height;

        PlanarYUVLuminanceSource source = new PlanarYUVLuminanceSource(
            data, previewWidth, previewHeight, 0, 0, previewWidth,
            previewHeight, false);
        BinaryBitmap bitmap = new BinaryBitmap(new
HybridBinarizer(source));

        Reader reader = new QRCodeReader();
        // Reader reader = new MultiFormatReader();
        try {

            Result result = reader.decode(bitmap);
            String text = result.getText();

            Intent intent = new Intent();
            Bundle bundle = new Bundle();
            bundle.putString("result", result.toString());
            intent.putExtras(bundle);
            setResult(RESULT_OK, intent);
            finish();
        } catch (Exception e) {
            e.printStackTrace();
//            Toast.makeText(getApplicationContext(), "Not Found",
Toast.LENGTH_SHORT).show();
        }
    }
});
}

    public void surfaceChanged(SurfaceHolder holder, int format, int width, int
height)

```

```

// 当 SurfaceView/预览界面的格式和大小发生改变时，该方法被调用
{
    Log.i("TAG", "SurfaceHolder.Callback: Surface Changed");
    //mPreviewHeight = height;
    //mPreviewWidth = width;
    initCamera();
    mCamera.cancelAutoFocus();//只有加上了这一句，才会自动对焦。
}

public void surfaceDestroyed(SurfaceHolder holder)
// SurfaceView 销毁时，该方法被调用
{
    Log.i("TAG", "SurfaceHolder.Callback: Surface Destroyed");
    if(null != mCamera)
    {
        mCamera.setPreviewCallback(null); //!! 这个必须在前，不然退出出错
        mCamera.stopPreview();
        bIfPreview = false;
        mCamera.release();
        mCamera = null;
    }
}

/* 【2】 【相机预览】 */
private void initCamera()//surfaceChanged 中调用
{
    Log.i("TAG", "going into initCamera");
    if (bIfPreview)
    {
        mCamera.stopPreview();//stopCamera();
    }
    if(null != mCamera)
    {
        try
        {
            /* Camera Service settings*/
            Camera.Parameters parameters = mCamera.getParameters();
            // parameters.setFlashMode("off"); // 无闪光灯
            parameters.setPictureFormat(PixelFormat.JPEG); //Sets the image
format for picture 设定相片格式为 JPEG，默认为 NV21
            parameters.setPreviewFormat(PixelFormat.YCbCr_420_SP); //Sets
the image format for preview picture, 默认为 NV21
            /* 【ImageFormat】 JPEG/NV16(YCrCb format, used for Video)/NV21(YCrCb
format, used for Image)/RGB_565/YUY2/YU12*/

```

```

        // 【调试】获取 camera 支持的 PictrueSize, 看看能否设置??
        List<Camera.Size> pictureSizes =
mCamera.getParameters().getSupportedPictureSizes();
        List<Camera.Size> previewSizes =
mCamera.getParameters().getSupportedPreviewSizes();
        List<Integer> previewFormats =
mCamera.getParameters().getSupportedPreviewFormats();
        List<String> focusModes = parameters.getSupportedFocusModes();
//        List<Integer> previewFrameRates =
mCamera.getParameters().getSupportedPreviewFrameRates();
        Log.i("TAG"+"initCamera", "cyy support parameters is ");
        Camera.Size psize = null;
        for (int i = 0; i < pictureSizes.size(); i++)
        {
            psize = pictureSizes.get(i);
            Log.i("TAG"+"initCamera", "PictrueSize,width: " + psize.width
+ " height" + psize.height);
        }
        for (int i = 0; i < previewSizes.size(); i++)
        {
            psize = previewSizes.get(i);
            Log.i("TAG"+"initCamera", "PreviewSize,width: " + psize.width
+ " height" + psize.height);
        }
        Integer pf = null;
        for (int i = 0; i < previewFormats.size(); i++)
        {
            pf = previewFormats.get(i);
            Log.i("TAG"+"initCamera", "previewformates:" + pf);
        }
        String fm;
        for (int i = 0; i < focusModes.size(); i++)
        {
            fm = focusModes.get(i);
            Log.i("TAG"+"initCamera", "previewformates:" + fm);
        }

//        List<int[]> supportedPreviewFpsRange =
mCamera.getParameters().getSupportedPreviewFpsRange();
        List<Integer> supportedPreviewFrameRates =
mCamera.getParameters().getSupportedPreviewFrameRates();
        int fr;
        for (int i = 0; i < supportedPreviewFrameRates.size(); i++)
        {

```

```

        fr = supportedPreviewFrameRates.get(i);
        Log.i("TAG"+"initCamera", "previewformates:" + fr);
    }
    List<int[]>
range=mCamera.getParameters().getSupportedPreviewFpsRange();
//
    Log.i("TAG", "range:"+range.size());
    for(int j=0;j<range.size();j++) {
        int[] r=range.get(j);
        for(int k=0;k<r.length;k++)
        {
            Log.i("TAG"+"initCamera", "supportedPreviewFps:"+r[k]);
        }
    }

//
parameters.setFocusMode(Camera.Parameters.FOCUS_MODE_CONTINUOUS_PICTURE);//1 连续对焦

parameters.setFocusMode(Camera.Parameters.FOCUS_MODE_CONTINUOUS_VIDEO);//自动对焦

//
    parameters.setPreviewFpsRange(29950, 30000);
    // 设置拍照和预览图片大小
//
    parameters.setPictureSize(4160, 2336); //honor 7 指定拍照图片的大小
//
    parameters.setPictureSize(2048, 1536); //nubia Z7 MAX 指定拍照图片的大小
//
    parameters.setPictureSize(3264, 1836); //nexus4 指定拍照图片的大小
    parameters.setPictureSize(1280, 720); //最优拍照图片的大小
    parameters.setPreviewSize(mPreviewHeight, mPreviewWidth); // 指定preview 的大小
//
    parameters.setPreviewFrameRate(20);
    //这两个属性 如果这两个属性设置的和真实手机的不一样时, 就会报错

    // 横竖屏镜头自动调整
    if (this.getResources().getConfiguration().orientation != Configuration.ORIENTATION_LANDSCAPE)
    {
        parameters.set("orientation", "portrait"); //
        parameters.set("rotation", 90); // 镜头角度转 90 度 (默认摄像头是横拍)

        mCamera.setDisplayOrientation(90); // 在 2.2 以上可以使用
    } else// 如果是横屏
    {
        parameters.set("orientation", "landscape"); //

```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />

</LinearLayout>

```

2.5 Activity & permission registration

Before running our application, we have to register the activities and ask for relative permissions. There is an Android Manifest in every android project. It stores the basic information of an application. If we need to get some information outside the app, such as sensors, Wi-Fi, file I/O, etc. , we need to ask the operating system for these permissions. Moreover, the operating system need to know every activity in a project. These information should be added in the manifest. The manifest in this project is listed below:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sjtu.zhusy54.qrcode">

    <uses-permission android:name="android.permission.VIBRATE" /> <!-- 震动权限 -->
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera" /> <!-- 使用照相机权限 -->
    <uses-feature android:name="android.hardware.camera.autofocus" /> <!-- 自动聚焦权限 -->

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>

```