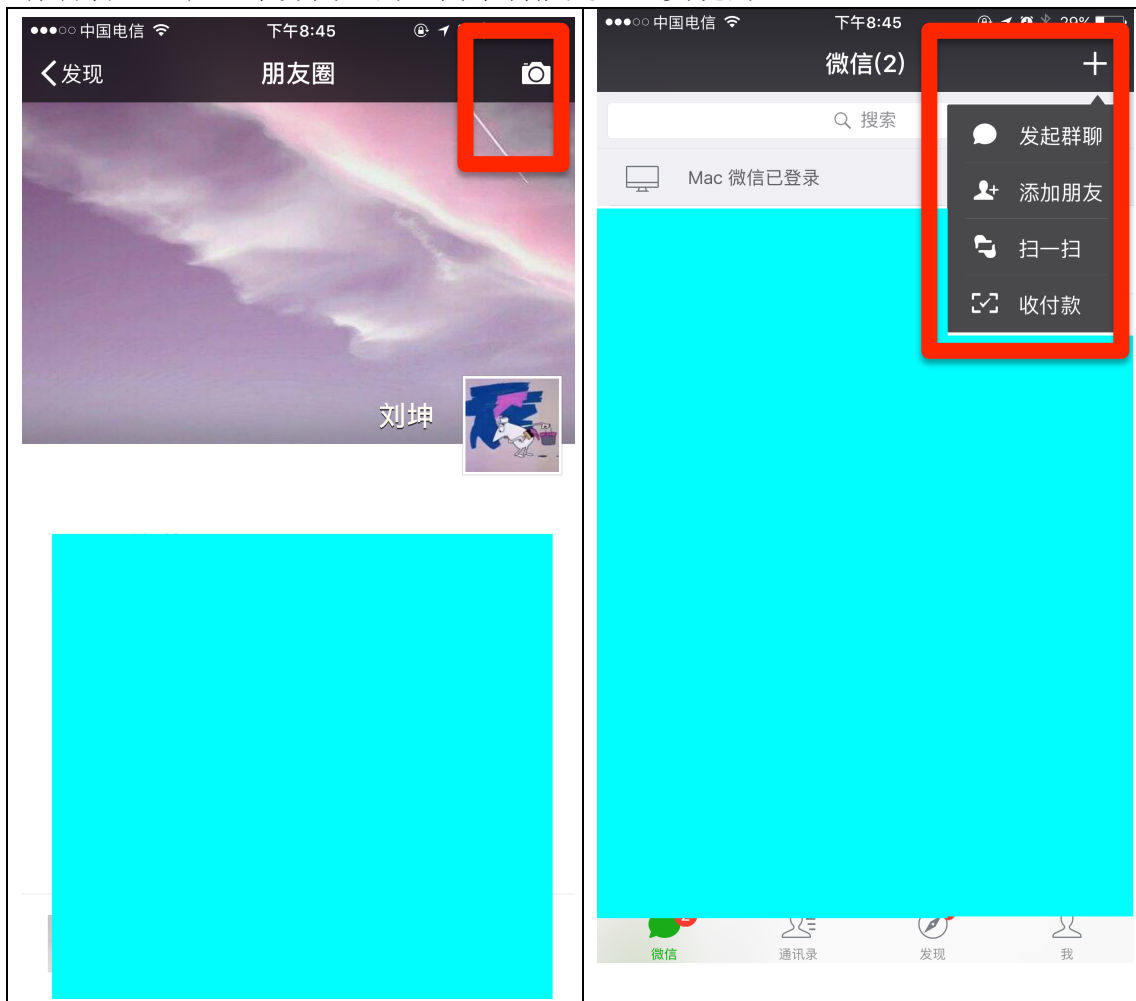


今天我们来练习如何创建按钮的监听和使用菜单栏功能。下面是两张微信的截图，你曾否想过在这个界面上的这两个功能是怎么实现的？



接下来大家跟我来一步一步实现按钮的单击监听和长按监听。

- 1、 首先，像往常一样创建一个空白的项目。
- 2、 然后，将我所给的 xml 文件复制到你 layout 文件当中。文件名 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:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.liuk.justjava.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="长按下面的二维码识别"
```

```

        android:textSize="16dp"
        android:textColor="@android:color/background_dark"
        android:layout_marginTop="16dp"
        android:layout_marginLeft="8dp"
        android:id="@+id/textview_barcoderesult"/>

```

<TextView

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0"
        android:textSize="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginLeft="8dp"
        android:id="@+id/textView_numberOfOrder"/>

```

<Button

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="点击我"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginRight="8dp"
        android:background="@android:color/holo_blue_light"
        android:id="@+id/button_order"/>

```

<Button

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginRight="8dp"
        android:text="长按我"
        android:background="@android:color/holo_blue_dark"
        android:id="@+id/button_toast" />

```

<ImageView

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:id="@+id/imageview_barcode"/>

```

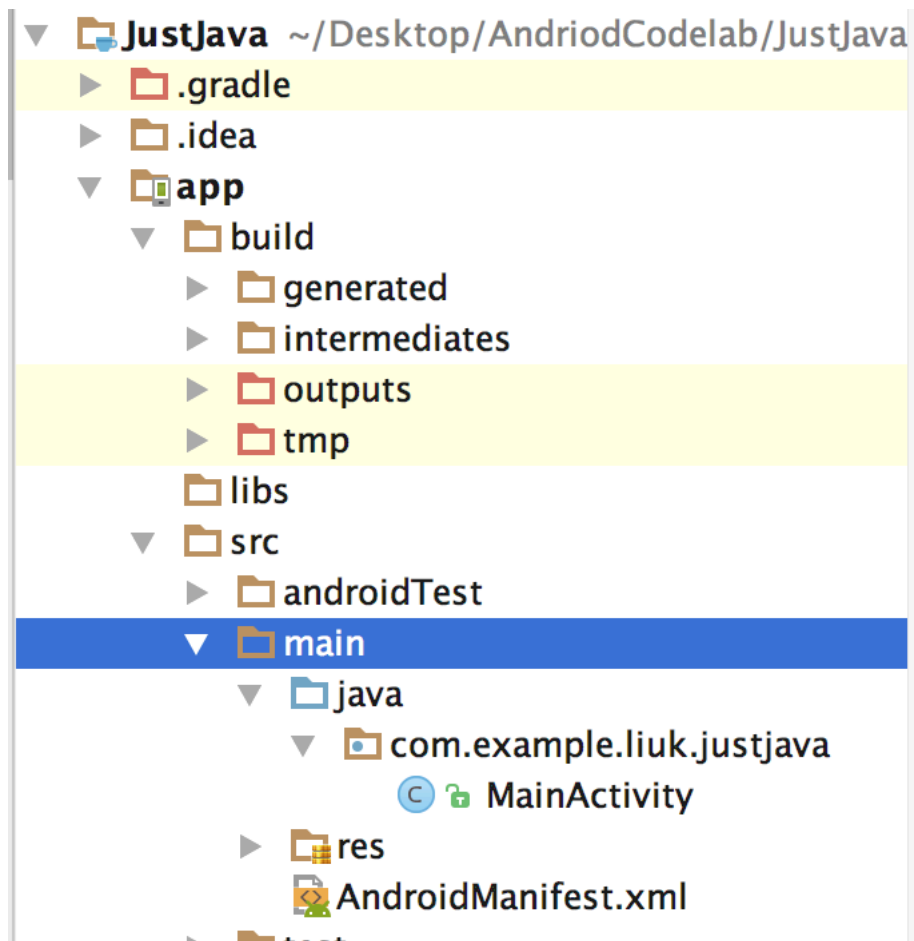
```

<!--android:src="@drawable/barcode_test"-->

```

</LinearLayout>

- 3、在 MainActivity 的 onCreate 方法中加入以下的语句，文件目录如下：
app/src/main/java/MainActivity



文件 step1

```
final Button button = (Button) findViewById(R.id.button_order);  
final Button buttonToast = (Button) findViewById(R.id.button_toast);
```

```
button.setOnClickListener(new View.OnClickListener(){
```

```
    @Override
```

```
    public void onClick(View v){
```

```
        TextView numberOfOrders = (TextView) findViewById(R.id.textView_numberOfOrder);
```

```
        numberOfOrders.setText(""+(++countA));
```

```
//这个地方是用来写当检测到点击事件的时候会执行的语句。
```

```
    }
```

```
});
```

```
buttonToast.setOnLongClickListener(new View.OnLongClickListener(){
```

```
    @Override
```

```
    public boolean onLongClick(View v) {
```

```
        Toast.makeText(MainActivity.this,"这是一个长按事件！",Toast.LENGTH_LONG).show();
```

```
//这个地方是用来写当检测到长点击事件的时候会执行的语句。在这个方法中你可以写任意合法的语句，
```

```
//但是其他的地方是固定的，好了再回到刚才你阅读的地方。
```

```
        return false;
```

```

    }
};

```

这个时候会出现错误，因为我们没有定义 `countA` 这个变量，需要你在下面图示位置加入变量：

```
private int countA = 0;
```

```

public class MainActivity extends AppCompatActivity {
    private int countA = 0;
    //TextView txtView = (TextView) findViewById(R.id.textview_barcoderesult);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

```

现在就已经完成对按钮的监听啦，运行你的 app，点击“点击我按钮”计数器会加一，而长按“长按我”这个按钮，会在 app 底部出现提示“这是一个长按事件”，这其实和微信那个长按按钮编辑家纯文本是一样的。

这个时候我们回到代码部分，看看我写的注释。

- 其实不只有 `button` 可以设定监听，所有继承自 `View` 的类都可以，就像你遗传自你父亲母亲的那些基因，只要是显性的都可以使用。所以我们来看看有哪些 `View` 可以调用监听？

Known Direct Subclasses

[AnalogClock](#), [ImageView](#), [KeyboardView](#), [MediaRouteButton](#), [ProgressBar](#), [Space](#), [SurfaceView](#), [TabItem](#), [TextView](#), [TextureView](#), [ViewGroup](#), [ViewStub](#)

Known Indirect Subclasses

[AbsListView](#), [AbsSeekBar](#), [AbsSpinner](#), [AbsoluteLayout](#), [ActionMenuView](#), [AdapterView<T extends Adapter>](#), [AdapterViewAnimator](#), [AdapterViewFlipper](#), and 107 others.

这些类都可以设置监听，接下来让我们在 app 中实现一个长按识别二维码：

- 二维码图片放入文件夹 `app/src/main/res/drawable`
- 在 `activity_main.xml` 文件中的注释加入 `<!--android:src="@drawable/barcode_test"--> ImageView` 的 `</>` 中。

- 将下面的代码加入到你刚才的 `Button` 对象下面（如图）：
文件 `step2`

```

final ImageView barcodeImageView =
    (ImageView)findViewById(R.id.imageview_barcode);
ImageView myImageView = (ImageView) findViewById(R.id.imageview_barcode);
final TextView txtView = (TextView) findViewById(R.id.textview_barcoderesult);
final Bitmap myBitmap = BitmapFactory.decodeResource(
    getApplicationContext().getResources(),
    R.drawable.barcode_test);
myImageView.setImageBitmap(myBitmap);

```

```

final Button button = (Button) findViewById(R.id.button_order);
final Button buttonToast = (Button) findViewById(R.id.button_toast);
final ImageView barcodeImageView = (ImageView) findViewById(R.id.imageview_barcode);
ImageView myImageView = (ImageView) findViewById(R.id.imageview_barcode);
final TextView txtView = (TextView) findViewById(R.id.textview_barcoderesult);
final Bitmap myBitmap = BitmapFactory.decodeResource(
    getApplicationContext().getResources(),
    R.drawable.barcode_test);
myImageView.setImageBitmap(myBitmap);

```

4、 将以下代码放在 `onCreate` 方法里面：

文件 `step3.txt`

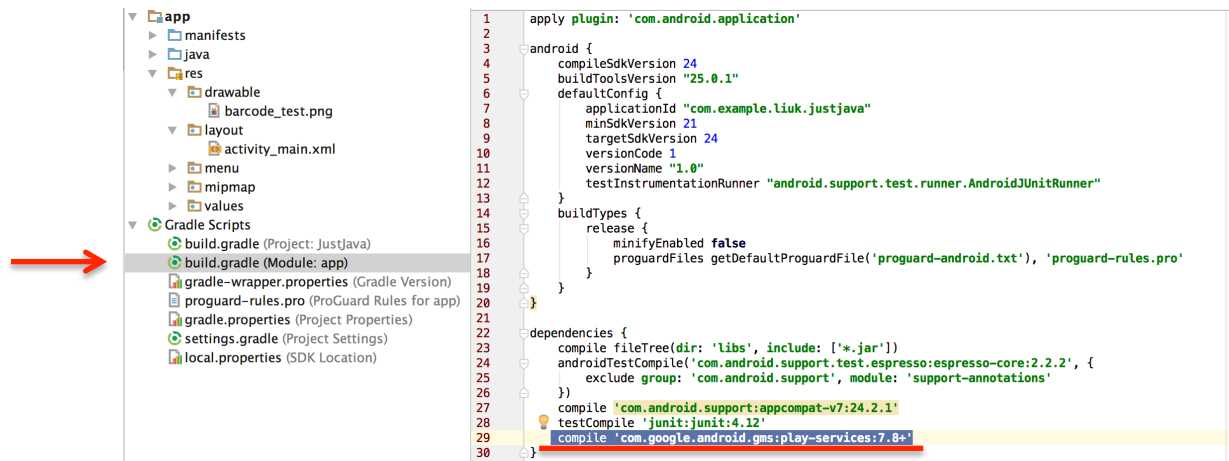
```

barcodeImageView.setOnLongClickListener(new View.OnLongClickListener(){
    @Override
    public boolean onLongClick(View v) {
        Toast.makeText(MainActivity.this, "正在识别二维码",
            Toast.LENGTH_SHORT).show();
        BarcodeDetector detector =
            new BarcodeDetector.Builder(getApplicationContext())
                .setBarcodeFormats(Barcode.DATA_MATRIX |
Barcode.QR_CODE)
                .build();
        if(!detector.isOperational()){
            txtView.setText("Could not set up the detector!");
            return true;
        }
        Frame frame = new Frame.Builder().setBitmap(myBitmap).build();
        SparseArray<Barcode> barcodes = detector.detect(frame);
        Barcode thisCode = barcodes.valueAt(0);
        txtView.setText(thisCode.rawValue);
        return false;
    }
});

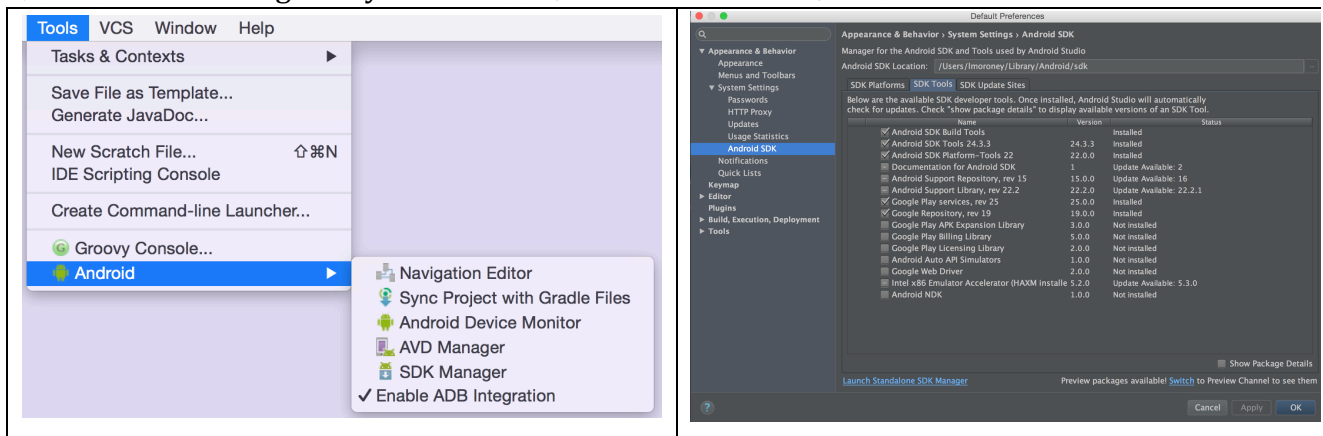
```

这个时候出现问题了，不要慌，在下面这个文件里加入：

`compile 'com.google.android.gms:play-services:7.8+'`

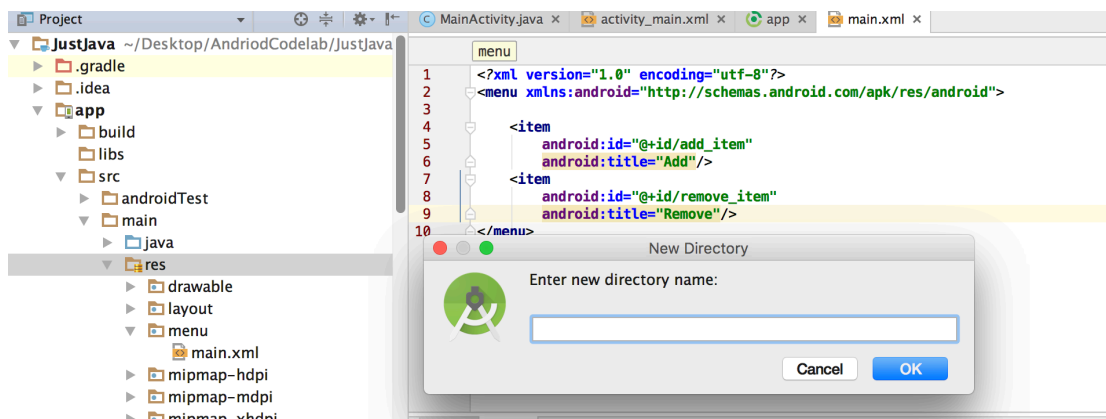


并且确定你的 Google Play Services 版本在 26 以上，升级步骤如下。

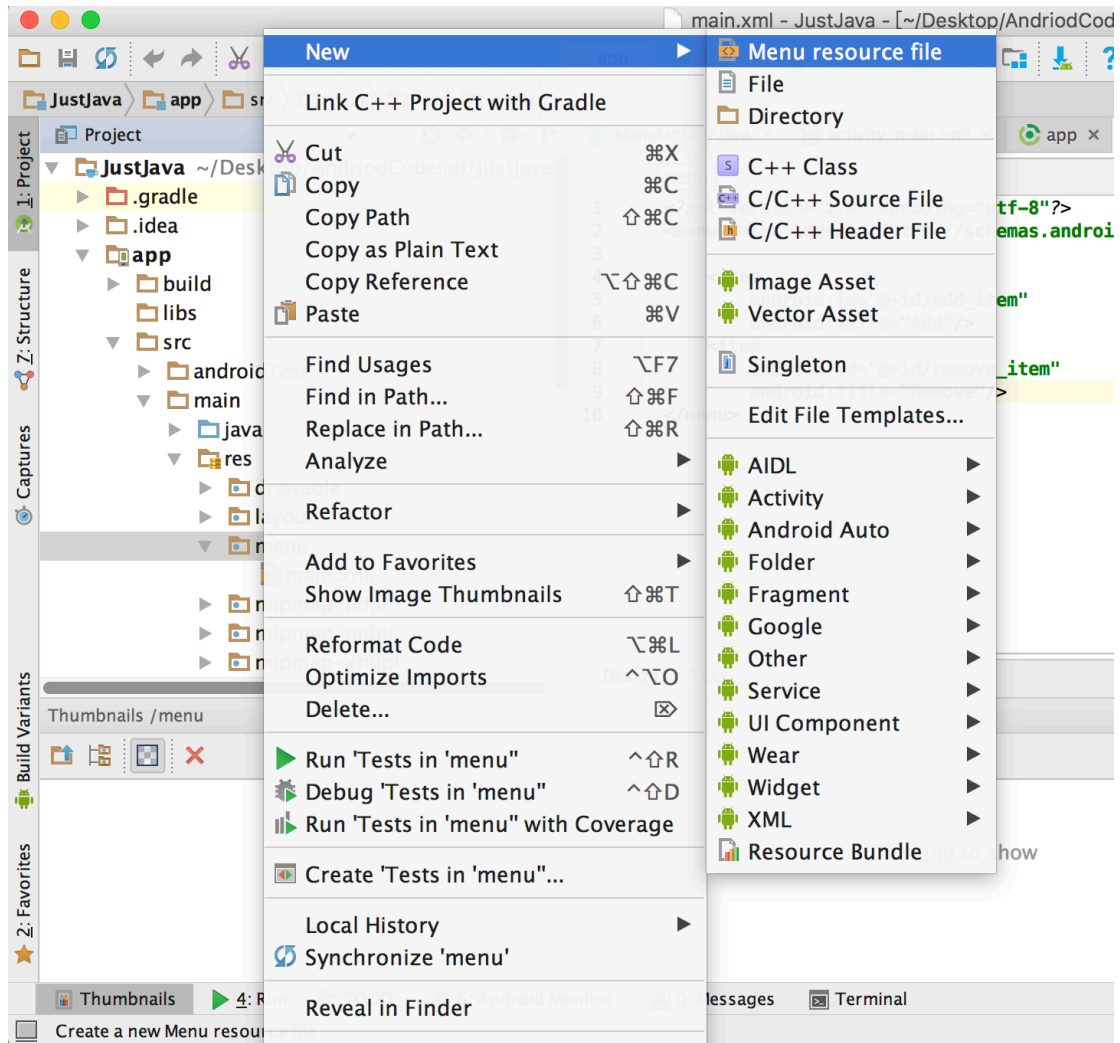


当问题解决，你的 **app** 就实现里以上的功能。

最后，让我们实现菜单功能：在 **app/src/main/res** 上右键 **new->Directory** 名叫 **menu**。



右键 menu new->menu resource file,取名 main。



在 main.xml 中加入以下语句：

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/add_item"
        android:title="Add"/>

    <item
        android:id="@+id/remove_item"
        android:title="Remove"/>

</menu>
```

再在 MainActivity 中重写以下方法。

文件 step4.txt

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    if(item.getItemId()==R.id.add_item){  
        Toast.makeText(MainActivity.this,"Add success",Toast.LENGTH_LONG).show();  
    }else if(item.getItemId()==R.id.remove_item){  
        ((TextView)findViewById(R.id.textview_barcoderesult)).setText("NULL");  
        Toast.makeText(MainActivity.this,"Remove success",Toast.LENGTH_LONG).show();  
    }  
    return true;  
}
```

这时再去运行 app，就会有右上角的菜单栏里