桂林电子科技大学

序号：

课号：2023280

**Android应用开发**  实验报告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | 实验二 Activity、控件及布局 | | | | | |  | 辅导员意见：  成绩 辅导员  签 名 |
| 院 系 | 计算机与信息安全学院 | | 专业 | 计算机科学与技术 | | |
| 学 号 |  | | 姓名 |  | | |
| 实验日期 | 2021 | 年 | 11 | 月 | 26 | 日 |
|  |  | | | | | |

# Android 项目——消息发送

## 一、实验目的

1. 掌握 AndroidManifest.xml 配置清单⽂件的结构；

2. 掌握使⽤ Intent 显式启动 Activity 的⽅式；

3. 掌握如何为 Activity 添加返回⽗ Activity 的⽅法；

## 二、实验内容

**1.首先创建一个Android Studio工程项目**

打开Android Studio软件，点击创建新项目,将项目命名为xiaoxifasong，点击Finish创建。

**2.编写布局文件activity\_main.xml**

activity\_main.xml是APP项目的布局文件，可以在这里对APP的界面进行修改设计。

在此项目中，我们需要EditText、Button控件各一个。代码编写如下所示

如上代码所示，将Button按钮命名为send\_message，Text文本框命名为message，实现点击按钮发送文本框信息功能。为此我们需要在MainActivity.java的OnCreate文件中为Send Message 按钮添加点击事件监听器

**3.为按钮Send Message添加点击时间监听器、编译工程并查看运行结果**

在MainActivity.java的OnCreate方法中编写代码，该方法是在Activity创建时被系统调用，是一个Activity生命周期的开始。因此在此处编写方法相当于对方法进行了初始化，在APP启动时就部署好了方法。

EditText文本框通过findViewById（R.id.前端界面text文本框对应id）取到对应控件变量，Button按钮控件同理。然后Button变量可以调用setOnClickListener方法为按钮注册一个监听器，点击按钮就会实现监听器中的OnClick方法。对应代码如下



运行效果如下图所示

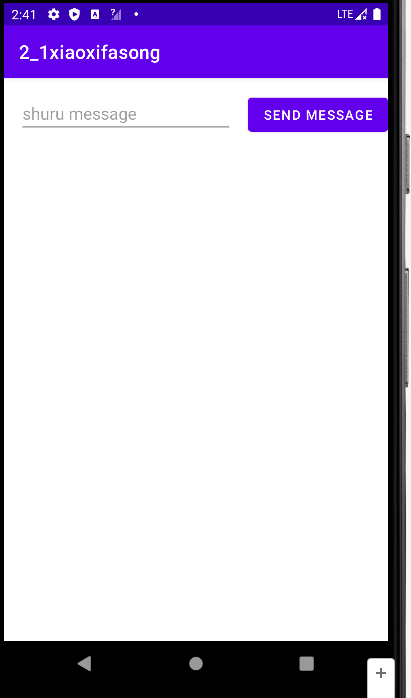


图1.1 运行结果

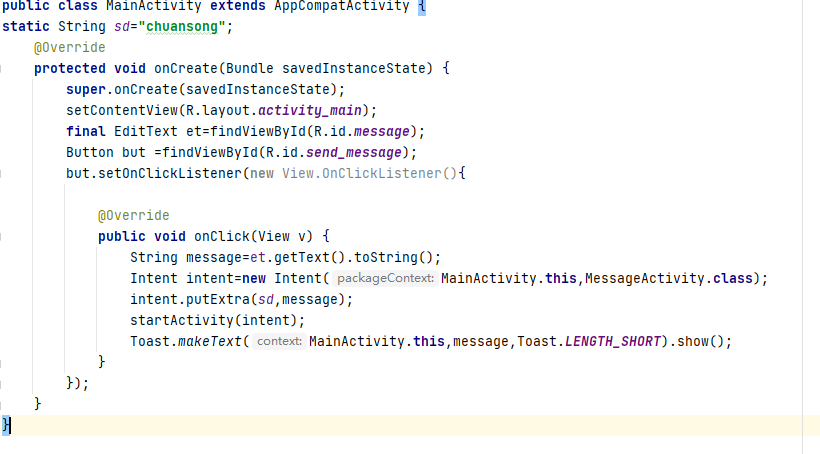
**4. 添加 MessageActivity 活动**

在 Android Studio 的 Project ⼯具窗⼜中右击 app 模块选择【New】->

【Activity】->【Empty Activity】。将新的活动命名为 MessageActivity，完成后，在 AndroidManifest.xml 配置⽂件中，你将看到两个已注册的 Activity

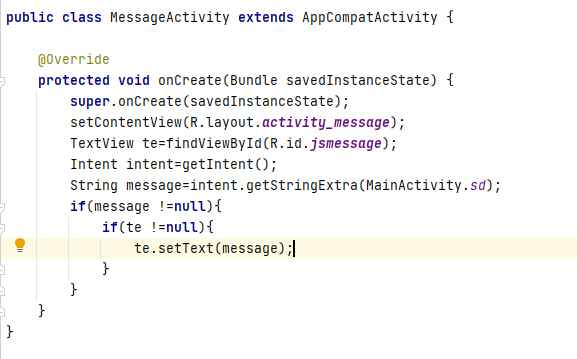
**5.修改MainActivity.java文件中的OnCreate方法**

修改OnCreate方法，实现在文本框输入字符串，点击sendmessage按钮后跳转到一个新页面，页面上显示字符串内容。为了实现页面跳转，我们使用Inter类，Intent是Android程序中各组件之间进行交互的一种重要方式，不仅可以指明当前组件想要执行的动作，还可以在不同组件之间传递数据。更改后MainActivity.java的OnCreate方法如下所示



**6.编写MessageActivity.java文件**

在MessageActivity.java文件中，我们需要实现接收MainActivity.java文件传输过来的字符串。使用getIntent()方法即可做到。该⽅法取得启动 Activity 的 Intent 对象，再使⽤ Intent 对象的getStringExtra()⽅法取得传⼊的 Message 值



**7.编写activity\_message.xml 布局⽂件，编译、运行工程并查看运行结果**

编写activity\_message.xml的布局文件，此项目不需要对其进行过多设计，只需要使得字符串显示在屏幕正中央即可，因此我们只需要一个TextView控件。



最终效果如图

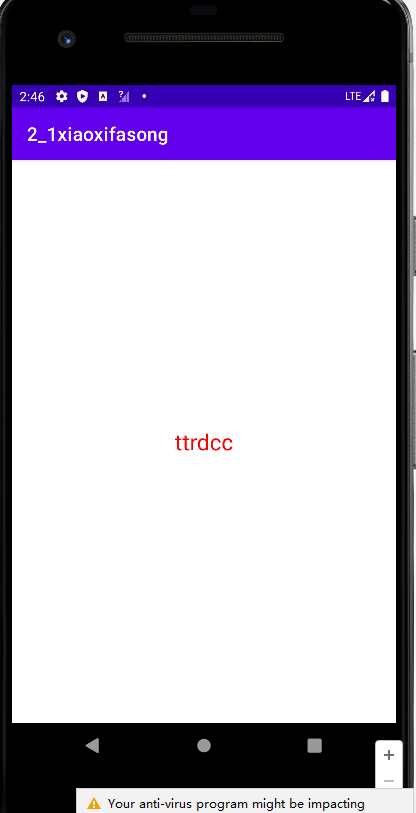


图1.2 最终运行结果

## 三、实验总结

通过这次实验懂得了如何设计Android的APP的界面布局，了解了EditText和Button等常用组件的属性。同时还知道如何为指定控件绑定监听器，为控件添加方法以完善APP功能。还懂得了页面之间的跳转以及页面与页面之间数据是如何传输的，成功完成了此次实验。

掌握了：

1.使⽤ ConstraintLayout （约束布局）进⾏布局管理；

2. 设置 EditText 单⾏⽂本模式属性；

3. 使⽤显式 Intent 启动 Activity ；

4. 使⽤ Intent 在 Activity 之间传递数据；

# Android 项目——计数器2

## 一、实验目的

1. 掌握 AndroidManifest.xml 配置清单⽂件的结构；

2. 掌握使⽤ Intent 显式启动 Activity 的⽅式；

3. 掌握如何为 Activity 添加返回⽗ Activity 的⽅法；

## 二、实验内容

**1.打开以前编写的Code2工程**

编译、部署该 App，运⾏时点击 COUNT 按钮后， TextView ⽂本控件的 计数发⽣变化。此时点击 AVD 虚拟机的 Rotate 按钮， TextView 的计数将清 零。导致该问题的原因是设备屏幕发⽣旋转时会导致 Android 系统的配置发⽣ 改变，从⽽引起当前的 Activity 被销毁后重新被创建，详细描述见 Activity 的 ⽣命周期。 Activity 被创建时，加载 activity\_main.xml 创建布局，在该布局⽂件中， 我们对 TextView 设置了⽂本的初始值为 0。因此，每当设备屏幕发⽣旋转时， TextView 的计数值都被清零。解决该问题的⽅法是每次 Activity 销毁时保存计 数值， Activity 被重新创建时在将保存的计数值设置为 TextView 显⽰的⽂本。

**2.更改主文件MainActivity.java**

onSaveInstanceState()是一个用于保存Activity状态的一个方法。我们可以用它来保存计数值。系统会先调用 onSaveInstanceState()，然后再使 Activity 变得易于销毁。系统会向该方法传递一个 Bundle，因此我们可以在其中使用putInt()方法以 "名字-值" 对形式保存 Activity 的计数值。自定义onSaveInstanceState()

方法如下

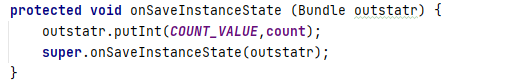


表2.1 onSaveInstanceState()调用示例代码

**3.新的Activity读取计数值**

新的Activity创建时，Bundle 同时传递给 onRestoreInstanceState() 和onCreate() 两个方法。我们使用上述的onRestoreInstanceState()这一方法从 Bundle 提取之前Activity保存的计数值并恢复该计数值。

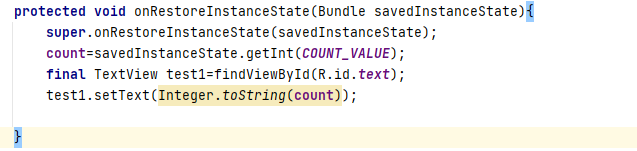


表2.2 onRestoreInstanceState()调用示例代码

**4.编译、运行工程并查看运行结果**

重新编译工程，查看运行结果



图2.1 计数器运行结果（竖屏）

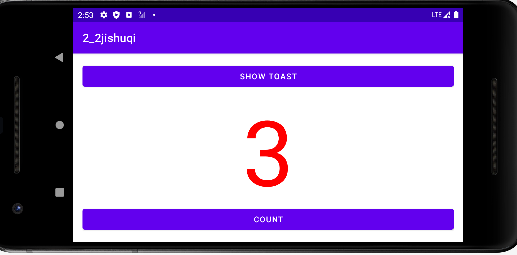


图2.2 计数器运行结果（横屏）

## 三、实验总结

通过这次实验懂得了如何通过onSaveInstanceState()方法保存Activity的状态。避免因为屏幕旋转导致APP产生一系列bug从而令用户体验大大下降。懂得了onSaveInstanceState()传输数据的一些方法和用onRestoreInstanceState()方法恢复Activity状态。

掌握了：

1. Activity ⽣命周期的各状态；

2. 使⽤ AVD 虚拟机的 Rotate 按钮旋转设备屏幕；

3. 使⽤ onSaveInstanceState ⽅法保存 Activity 状态；

4. 使⽤ onRestoreInstanceState ⽅法恢复 Activity 状态；

# Android 项目——登录界面

## 一、实验目的

1. 掌握布局的组合使⽤⽅式；

2. 掌握 Vector Asset 资源的使⽤⽅式；

3. 掌握 Image View 、View 等控件的使⽤⽅式；

## 二、实验内容

**1.新建名为**xinwenyueduqi**的Android工程**

打开Android Studio软件，点击创建新项目,将项目命名xinwenyueduqi，点击Finish完成创建。

在本次项目中，我们需要设计一个APP登录页面，来熟悉Android Studio内置资源的引入和使用以及某些常用控件的使用方式。具体页面如下图所示

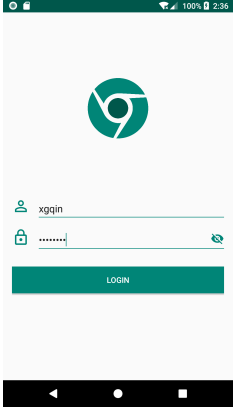


图3.1 登录页面效果

**2.设计页面布局**

由上述登录效果图可以知道，我们需要以下三个控件布局

* ImageView，显⽰ Logo 、用户名、密码、显示/隐藏密码样式四个图标；
* EditText，用于输⼊⽤户名和密码的文本框；
* Button，用于登录的一个按钮。

根据登录效果图把页面分为3个部分，我们均使用RelativeLayout

* 页⾯顶部，包含APP的 Logo。
* 页⾯中部，包含用户名、密码输⼊框等。
* 页⾯底部，包含登录所需按钮。

页面整体布局采用LinearLayout布局，使用RelativeLayout的属性android:layout\_weight指定页面顶部和底部元素的高度。页面中部则视情况调整。

**3.加入Image Asset资源**

右键res包下的mipmap目录，选择【New】->【Image Asset】菜单，在弹出的Asset Studio中，设置icon的名称，在Soucre Asset中选择本地icon路径

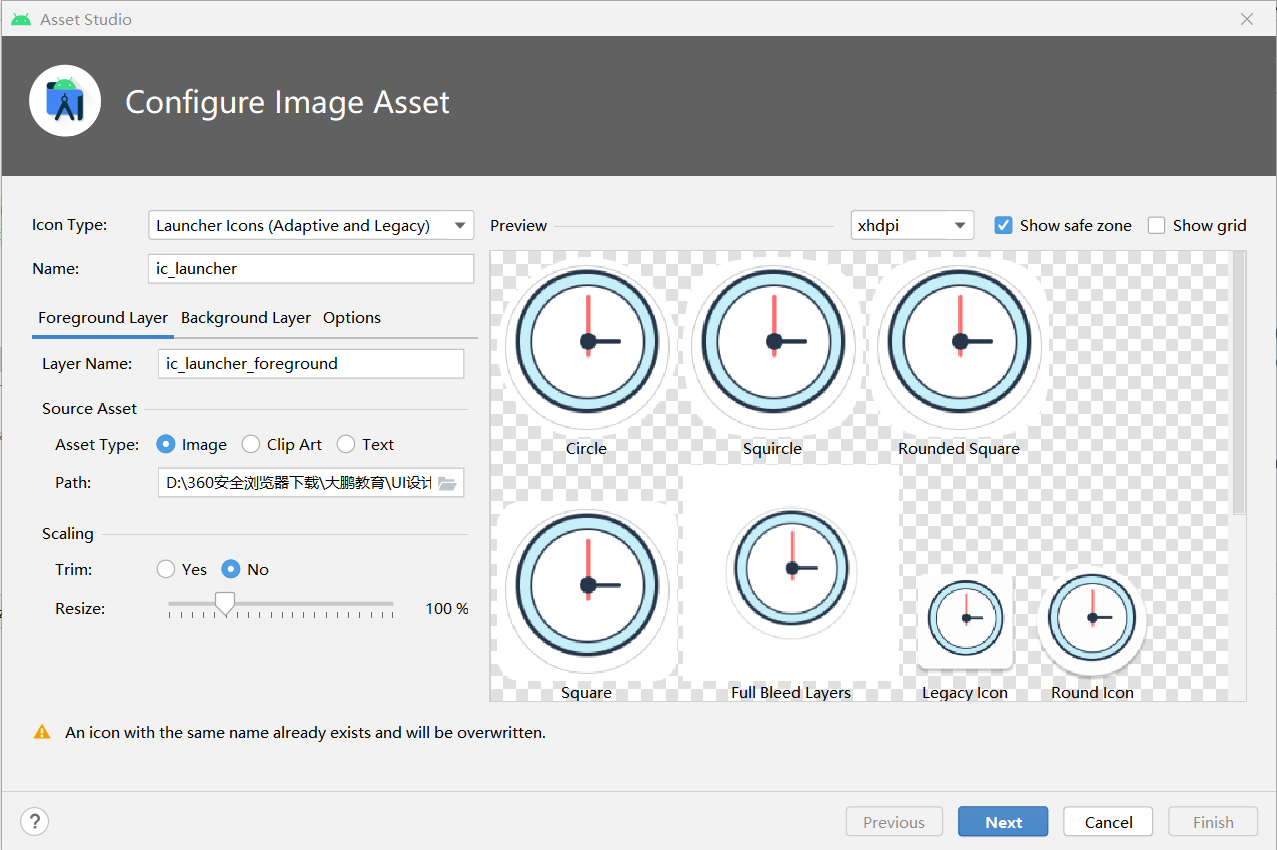


图3.2 Configure Image Asset页面

**4.加入Vector Asset资源**

右键res包下的drawable目录，选择【New】->【Vector Asset】菜单，在弹出的Vector Studio中，双击Clip Art下的图标，在弹出的窗口中搜索person outline，选择图标并点击OK完成Vector Asset资源的引入。同理引入其他所需图标。

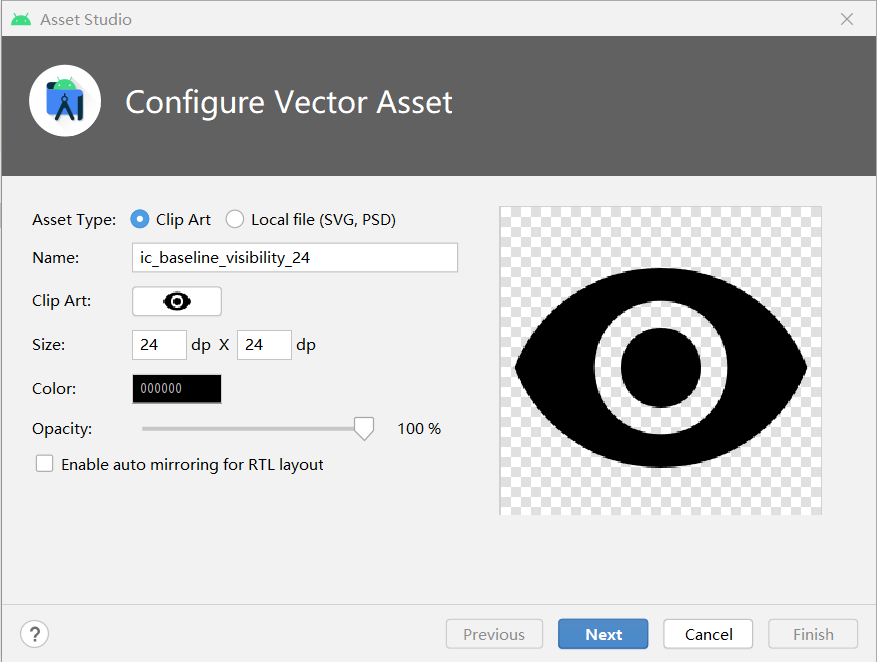


图3.3 Configure Vector Asset页面

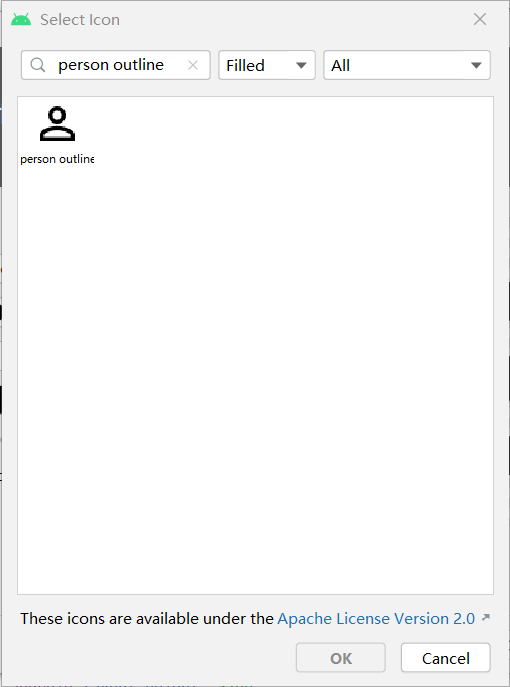


图3.4 select icon页面

**5.编写 LoginActivity.java文件实现密码明文切换图标**

在LoginActivity.java文件中为控件et\_pwd和iv\_pwd\_switch绑定监听器，使得点击iv\_pwd\_switch控件可以实现密码控件et\_pwd的显示/隐藏。



**6.编译、运行APP并查看运行结果**

可以看到当点击iv\_pwd\_switch控件后，密码显示/隐藏。

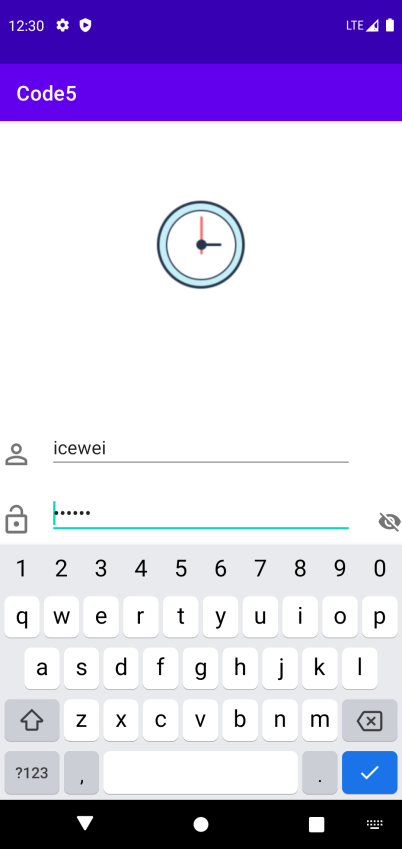


图3.5 运行结果（密码隐藏）

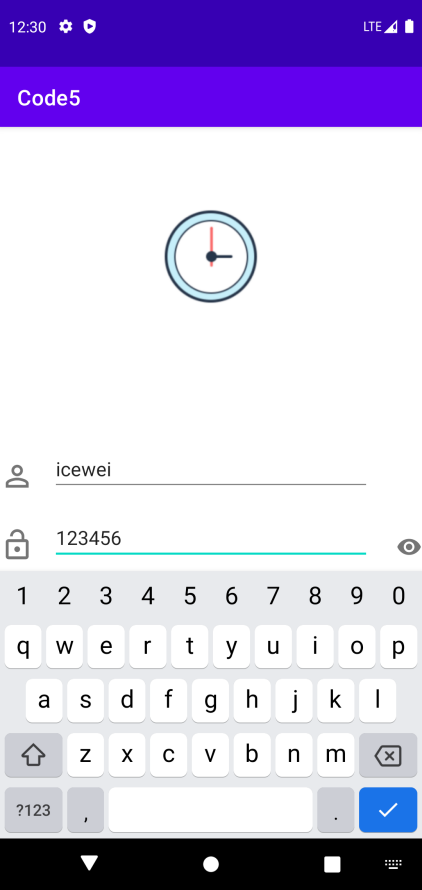


图3.6 运行结果（密码显示）

## 三、实验总结

通过这次实验懂得了如何引入系统的Image Asset和Vector Asset资源，设置APP的启动图标和设计用户名、密码等图标。还通过该实验了解了如何实现文本框内文字的显示和隐藏。

掌握了：

1. 使⽤ LinearLayout ， Relative Layout 进⾏页⾯布局；

2. 使⽤ Vector Asset 添加图标资源；

3. 使⽤ Image Asset 导⼊应⽤启动图标；

# Android 项目——新闻列表

## 一、实验目的

1. 掌握 ListView 控件的基本⽤法；

2. 掌握 ListView ⾃定义 Item 布局的⽅法；

3. 掌握 ArrayAdapter 的基本⽤法及⾃定义 Adapter 的⽅法；

4. 掌握 CardView 控件的基本⽤法；

## 二、实验内容

此次实验要求实现一个新闻列表页面，具体布局如下图所示。

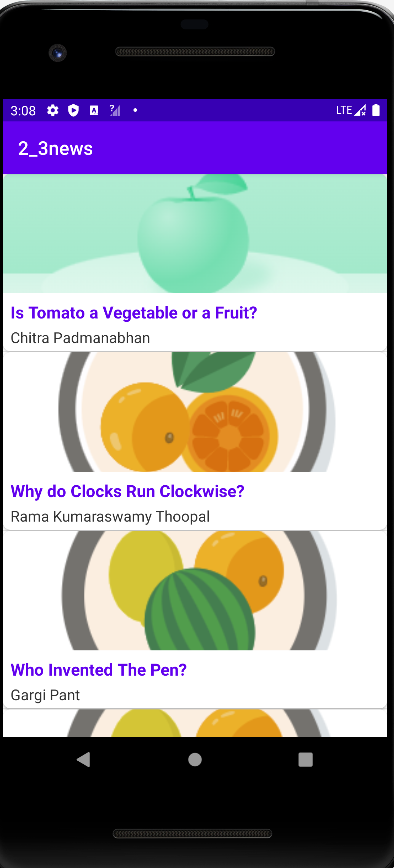


图4.1 新闻列表具体页面

**1.创建名为newszuizhong的Android项目**

打开Android Studio软件，点击创建新项目,将项目命名为**newszuizhong**，点击Finish完成创建。

**2.设计页面布局**

首先我们需要设计页面整体布局，由于是一个新闻列表，所以我们需要使用ListView组件呈现新闻。



表4.1 activity\_main.xml主要代码

简便起见，我们将所有的新闻数据作为字符串资源放⼊ arrays.xml ⽂件 中。在⼯程窗⼜中右击 app 模块⽂件夹，选择【New】->【Android Resource File】 菜单，弹出 New Resource File 窗⼜，设置⽂件名为 arrays 。在 array.xml ⽂件中加⼊ titles 及 authors 两个字符串数组，标签下内容自定义。

**3.构建ListView适配器ArrayAdapter**

在安卓中，设配器就是把数据变成符合界面风格的形式，并且通过ListView显示出来。也就是说，适配器是数据和界面之间的桥梁。常见的ListView适配器有ArrayAdapter和SimpleAdapter两种适配器。

我们首先设置ListView组件的ArrayAdapter适配器。

表4.2 实现ArrayAdapter适配器代码

|  |
| --- |
| public class MainActivity extends AppCompatActivity {   private String[] titles = null;  private String[] authors = null;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);   titles = getResources().getStringArray(R.array.titles);  authors = getResources().getStringArray(R.array.authors);   ArrayAdapter<String> adapter = new ArrayAdapter<String>(  MainActivity.this,android.R.layout.simple\_list\_item\_1, titles);   ListView lvNewsList = findViewById(R.id.news\_list);  lvNewsList.setAdapter(adapter);  } } |

在上述代码中，我们显示用getResources()方法取到系统资源，然后使用getStringArray()根据标签的ID来取到标签下的信息并转化成字符串。随后new 一个ArrayAdapter<String>初始化设配器。最后调用setAdapter()方法为ListView设置适配器。

**4.编译、运行和查看运行结果**

初步运行效果如下图所示

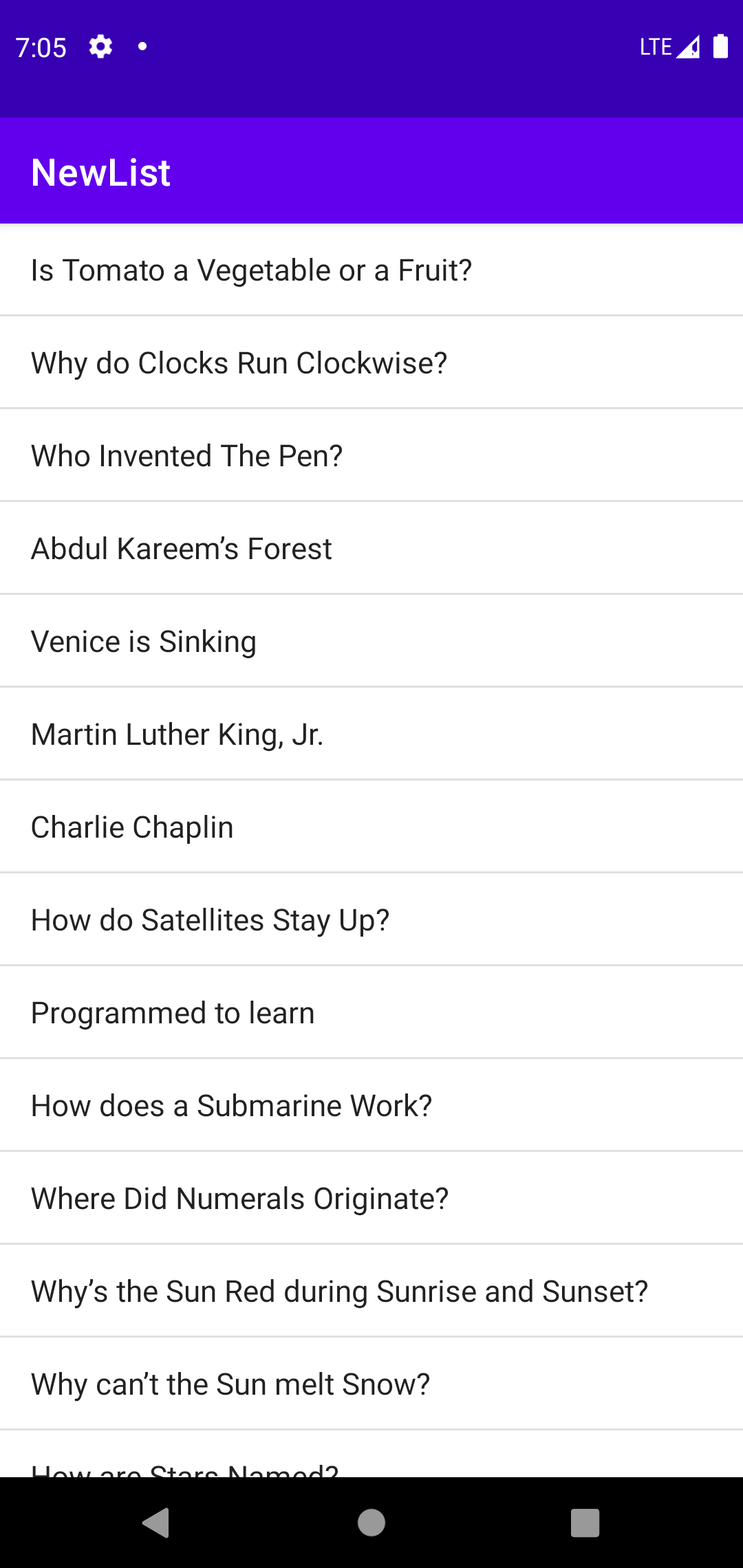


图4.2 ArrayAdapter作适配器运行结果

**5.构建适配器SimpleAdapter**

ArrayAdapter 作为 ListView 的适配器使⽤时，只能⽤于 Item 中包含 1 个 TextView 的情况。如果要在列表中显⽰新闻的标题及作者，则需要使 ⽤ SimpleAdapter 作为适配器。

首先我们得了解SimpleAdapter中参数所表达的意思：

表4.3 SimpleAdapter方法

|  |
| --- |
| SimpleAdapter(Context context, List<? extends Map<String,?>>data,int resource,String[] from,int[] to); |

表4.4 SimpleAdapter参数解析

|  |  |
| --- | --- |
| Context context | 表示上下文对象，就是要展示 ListView 的 Activity，或者是通过 getApplicationContext() 得到的上下文对象 |
| List<?extends Map<String,?>>data | 用于在列表中显示的数据，传入的数据必须是 List<? extends Map<String,?>> 的实现类 |
| int resource | ListView 中显示的每行子 View 的资源文件，就是位于 layout 文件夹中的 .xml 布局文件 |
| String[] from | 表示 Map<String,Object> 中存放的 Key 键值，因为它要通过键值才能找到对应的 Value，也就是布局要显示的内容 |
| int[] to | 表示要显示出来的 resource 布局文件的 R.id.xx 值，它和 from 中的数据源选项要一一对应 |

设置ListView数据源

使⽤ List<Map > dataList; 替换掉之前定义的 titles、authors 数组。将数据源的构造操作放⼊onotData()⽅法中。

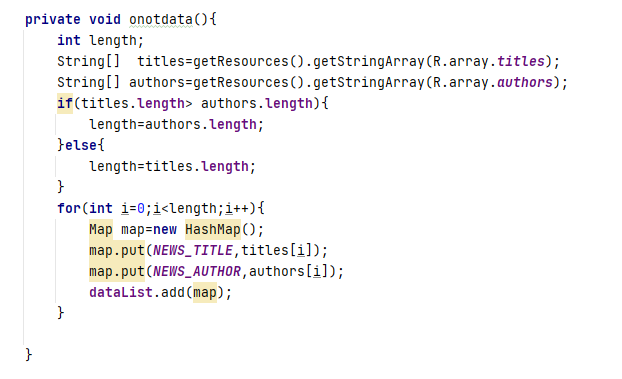


表4.5 设置ListView数据源代码

在上述代码中，我们使用了HashMap()类。哈希表（HashMap）是种数据结构，应用于键值对的保存和查找，删除等操作（用来索引的对象叫做key（键），其对应的对象叫做value（值））。所以我们将每个title和author以键值对的形式链接在一起并存储到数据源dataList中。

然后我们开始编写SimpleAdapter适配器。

表4.6 实现SimpleAdapter适配器代码



**6.编译、运行APP并查看运行结果**

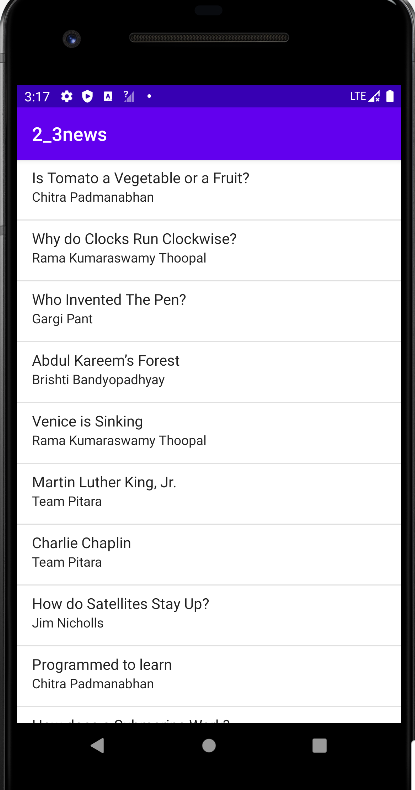


图4.3 SimpleAdapter作适配器运行结果

**7.自定义Item布局**

要实现如效果图一样的图片、标题、作者排版，我们需要自定义Item布局。为此我们可以使用CardView卡片布局。

CardView的基本属性：

表4.7 CardView基本属性解析

|  |  |
| --- | --- |
| app:cardBackgroundColor | 设置背景颜色 |
| app:cardCornerRadius | 设置圆角大小 |
| app:cardElevation | 设置z轴的阴影 |
| app:cardMaxElevation | 设置z轴的最大高度值 |
| app:cardUseCompatPadding | 是否使用CompatPadding |
| app:cardPreventCornerOverlap | 是否使用PreventCornerOverlap |
| app:contentPadding | 设置内容的padding |
| app:contentPaddingLeft | 设置内容的左padding |
| app:contentPaddingTop | 设置内容的上padding |
| app:contentPaddingRight | 设置内容的右padding |
| app:contentPaddingBottom | 设置内容的底padding |

首先自定义list\_item.xml，右键res包中的layout目录，选择【New】->【Layout Rescouse File】,在File Name中写入list\_item.xml然后点击OK生成新的布局文件。

其次我们需要到build.gradle配置文件下添加CardView依赖

表4.8 CardView依赖

|  |
| --- |
| implementation 'com.android.support:cardview-v7:28.0.0' |

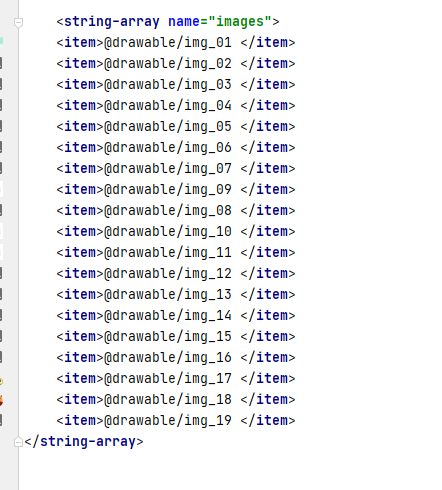
**8.编辑list\_item.xml布局文件**

在list\_item.xml文件中我们需要写入三个RelativeLayout组件（图片、标题和作者）进行布局管理。新闻标题及作者对应的 TextView 控件则使⽤分别使⽤ layout\_below 属性位于新闻图⽚及新闻标题下⽅，并设置合理的 padding 属性。

**9.引入图片资源文件**

把图片资源放到res包下的drawable下。然后再arrays.xml文件下引用。

表4.9 arrays.xml引用图片代码



**10.构造新闻适配器NewsAdapte**r

新建一个NesAdapter类，继承父类ArrayAdapter。

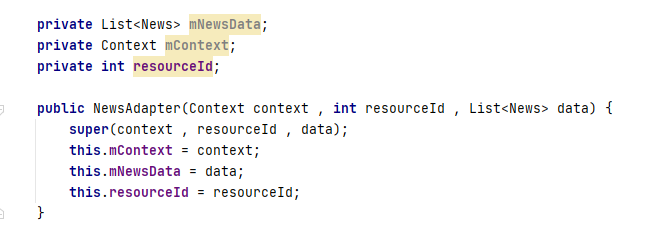
NewsAdapter 类的构造器需要三个参数，分别为：

表4.10 ArrayAdapter参数解析

|  |  |
| --- | --- |
| Context context | context 上下⽂⽤于 NewsAdapter 类中 LayoutInflater 加载布局时使⽤ |
| int resourceId | ⽤于设置 ListView 每个 Item 项时的布局 |
| List<News> data | ⽤于传递 News 对象列表 |

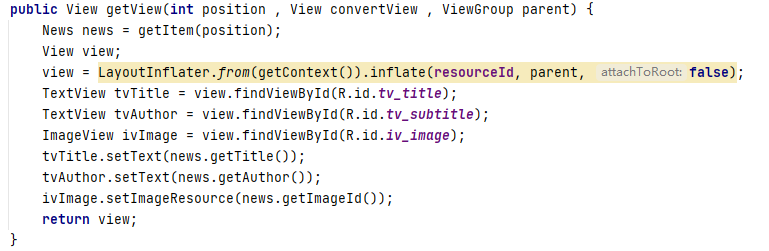
NewsAdapter构造器代码如下：

表4.12 NewsAdapter构造器代码



此外我们还需要实现getView方法，getView ⽅法是⽤于 ListView 显⽰某⼀位置 Item 时进⾏回调的⽅法。在 getView ⽅法中调⽤ LayoutInflater 加载指定的 resourceId 布局，第二个参数则为加载的布局对应的⽗容器控件对象，第三个参数表⽰是否将加载的布局加⼊⽗容器控件中，在此情况下要选择 false。

表4.13 getView方法实现

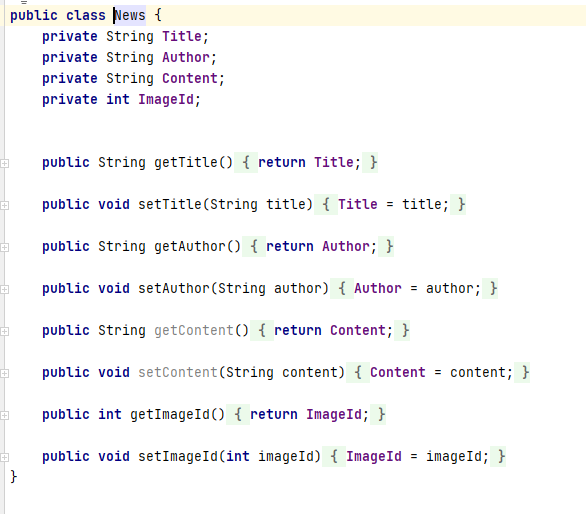


**11.再次修改数据源的构造方式**

在 MainActivity.java文件中 中修改从 arrays.xml 中加载新闻标题、作者、新闻标题图的⽅式。

在之前的initData()方法中，我们使用了HashMap()类来绑定新闻的标题和作者，现在我们新建一个News类来存储新闻信息。

表4.14 News类



在inotdata()方法中更改newsList添加数据方式

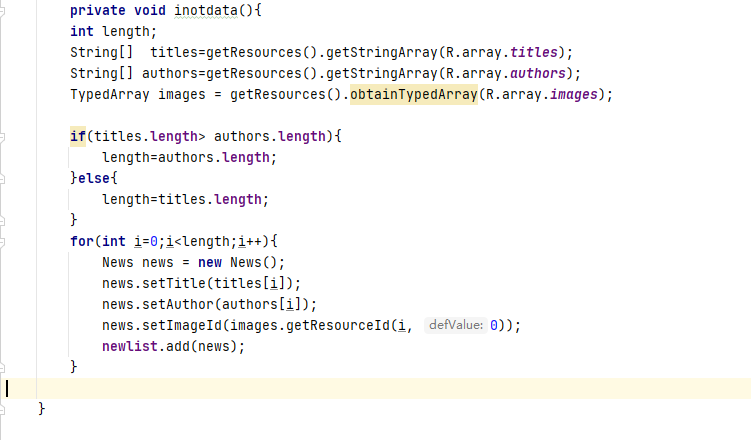


表4.15 inotdata()方法实现代码

最后我们在onCreate()方法中更改适配器

表4.16 更改适配器



12.编译、运行APP并查看APP运行结果

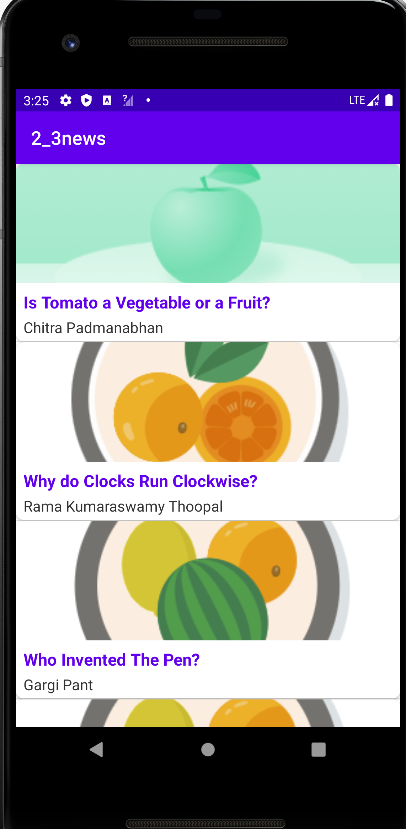


图4.4 最终运行结果

## 三、实验总结

通过这次实验懂得了使用ListView显示数据，如何使用自定义适配器ArrayAdapter和SimpleAdapter来展示新闻内容。还学会了如何使用CardView卡片布局。

掌握了：

1. 使⽤ ListView 进⾏数据显⽰；

2. ⾃定义 ListView 的 Item 布局；

3. ⾃定义 ArrayAdapter 构造适配器；

4. 使⽤ CardView 卡⽚布局展⽰数据；