



常州大学
CHANGZHOU UNIVERSITY

实 验 报 告

第 二 次实验

课程名称： 移动终端软件开发

实验名称： Intent 的调用

学 院： 信息数理学院 班 级： 计算机 162

学 号： 16430121 姓 名： 林锦雄

完成日期： 2019.5.5

指导教师： 高晋树 成 绩：

一、实验目的

1. 了解 Intent 的组成;
2. 掌握使用 Intent 启动 Activity 的方法;
3. 掌握获取 Activity 返回值的方法。

二、实验过程

在实验一的工程基础上，添加 Activity，利用 Intent 实现 Activity 组件之间的相互调用。

1、建立信息显示界面

创建新的 StudentDetailActivity，界面如图 2.1 所示。

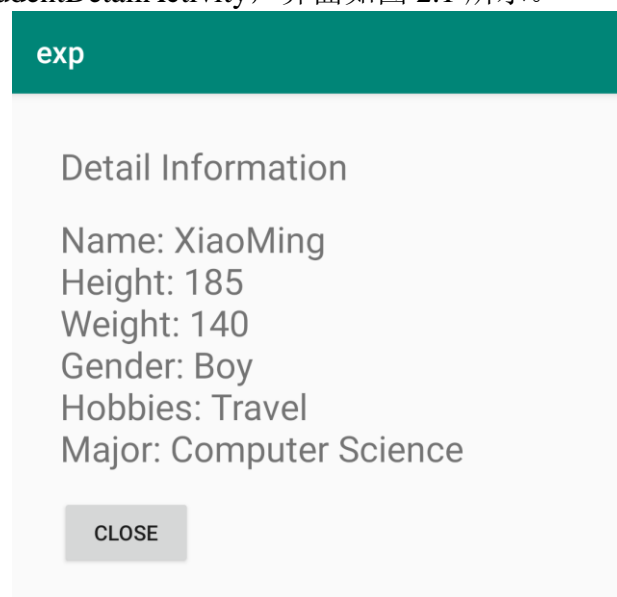


图 2.1 信息显示界面

点击列表中的“查看”菜单项将会启动图 2.1 的信息显示界面，此时界面上应该显示出所选择查看的姓名的所有字段信息。

点击“返回”按钮将会回到基本信息界面。

```
1. @Override
2. protected void onCreate(Bundle savedInstanceState) {
3.     super.onCreate(savedInstanceState);
4.     setContentView(R.layout.activity_student_detail);
5.     TextView view = findViewById(R.id.studentDetailTextView);
6.     view.setText(getIntent().getSerializableExtra("student").toString());
7.     findViewById(R.id.closeDetailButton).setOnClickListener(new View.OnClickListener() {
8.         @Override
9.         public void onClick(View v) {
10.             MainActivity.this.finish();
11.         }
12.     });
13. }
```

2、建立专业设置界面

创建新的 MajorSettingActivity，界面如图 2.2 所示。

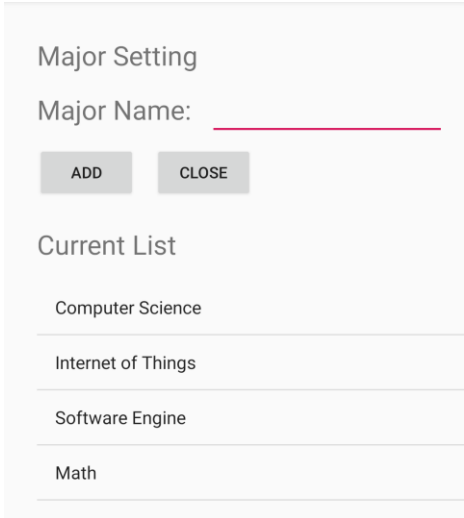


图 2.2 专业设置界面

点击选项菜单的“专业设置”子菜单项将会启动图 2.2 的专业设置界面，此时界面上应该显示出专业 Spinner 控件中列出的所有专业信息。

点击“添加”按钮则把 EditText 控件中输入的专业名称加入下面的 ListView 控件列表，如果 EditText 控件为空，则要给出相应提示。

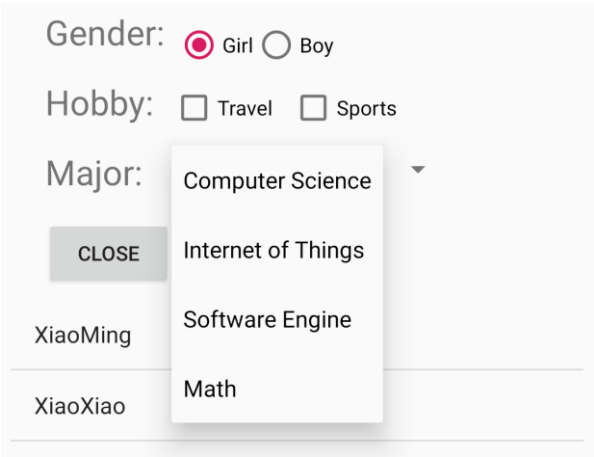


图 2.3 主界面 Spinner

点击“返回”按钮将会回到的基本信息界面，此时在专业设置界面上做的任何改变都应该给出广播消息提示，并在专业 Spinner 控件中体现出来。

3、添加快捷菜单

为专业设置界面上的 ListView 控件添加一个快捷菜单，包含“修改”和“删除”两个菜单项，效果如图 2.3 所示。

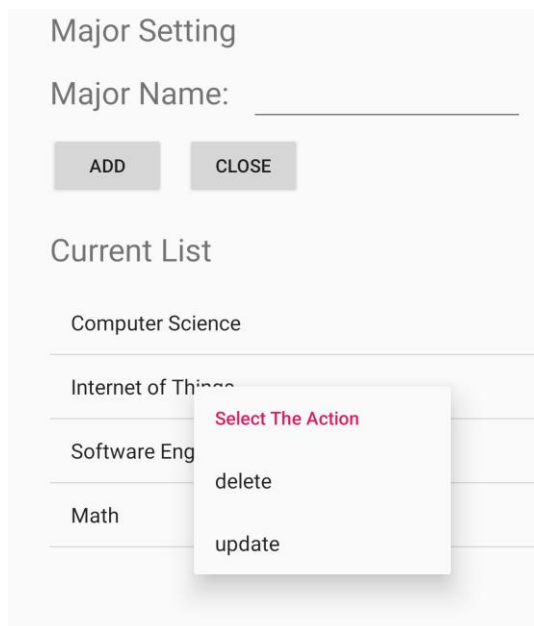


图 2.4 专业设置快捷菜单

点击“修改”弹出对话框让用户输入新的名称然后点 **Update** 进行更新，同时更新下面的 **ListView** 控件列表，如果 **EditText** 控件为空，则要给出相应提示。

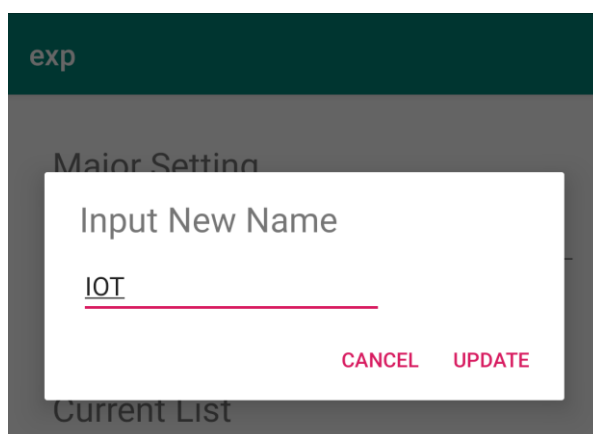


图 2.5 修改专业设置对话框

点击“删除”菜单项则删除所选择项的专业名称，同时更新下面的 **ListView** 控件列表。

三、实验总结

通过这次实验，学习了 **Intent** 的组成，以及使用 **Intent** 启动 **Activity** 的方法，如何进行 **Activity** 之间使用 **Intent** 传递值的方法，练习了 **Activity** 的通过隐式启动的方法，掌握获取 **Activity** 返回值的方法。学习了如何使用对话框优化界面友好性。

四、思考练习

如何实现 Android 程序的开机自启动？

只要让 Service 响应 BOOT_COMPLETED 的广播，再增加 RECEIVE BOOT_COMPLETED 的权限就可以实现开机自启动。

```
1. <intent-filter>
2.     <action android:name="android.intent.action.BOOT_COMPLETED" />
3.     <category android:name="android.intent.category.DEFAULT" />
4. </intent-filter>
```