

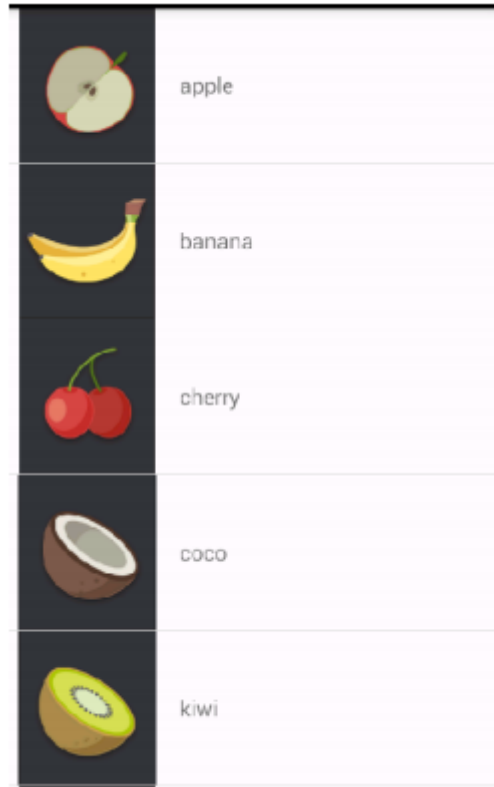
实验三 Intent, Bundle 的使用及 ListView 的应用

【实验目的】

1. 复习 Button 点击事件处理
2. 了解 Intent, Bundle 机制，并且使用 Intent, Bundle 实现页面跳转
3. 了解 ListView 使用及其 Item 点击事件处理

【实验内容】

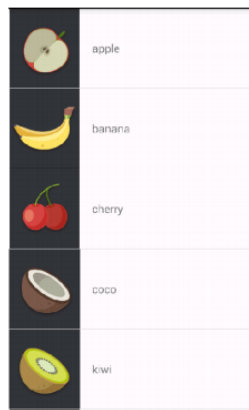
1. 进入主页面后，使用 ListView 实现如下页面：



2. 点击其中任何一项水果，跳转到另外一个活动，使用 Intent 转换活动，并且使用 Bundle 传递数据（必须使用 Bundle），跳转页面如下：



其中水果名为 Bundle 传递进来数据，点击 Back 按钮，返回原来的 Activity

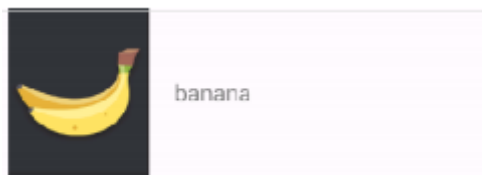


【参考】

1. ListView 是 Android 中最常用的控件之一，同时也是比较难用的控件。与前面学习的控件相比，ListView 用法相对复杂，因此能够使得手机程序界面丰富度和灵活性大大提高。

ListView 以列表形式动态展示数据，其中列表中的项（item）不仅仅是文字，同时也可以图片和文字。因此，ListView 需要给列表中的每一项添加布局文件（xml），从而控制每一项展示方式。

2. 本实验中需要用到布局文件（xml）对 ListView 每一项进行安排（自己设计 xml 文件）：



可以看出，每一项都有一张图片和该水果的名称，因此，考虑使用一个类，对 Fruit 进行封装（传说中的面向对象，原来可以这样玩）：

```
public class Fruit {  
    private String name;  
    private int imageId;  
  
    public Fruit(String name, int imageId) {  
        this.name = name;  
        this.imageId = imageId;  
    }  
  
    public int getImageId() {  
        return imageId;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

然后用线性表，将所有的 Fruit 对象储存起来：

```
private List<Fruit> fruitList = new ArrayList<Fruit>();
```

那么 `imageId` 是什么呢？在 `ListView` 中，每一项（`Item`）都需要图片和水果名称，原来 `imageId` 存储的是每一张图片的 `id`（`android` 为每一个文件都使用了一个唯一的 `id`，定义在 `R` 文件中）：

```
fruitList.add(new Fruit("apple", R.mipmap.apple));
```

然后怎么办？接下来需要创建一个自定义的适配器，继承自 `ArrayAdapter`，并且泛型指定为 `Fruit` 类，重写该类中的方法：

```
public class FruitAdapter extends ArrayAdapter<Fruit> {
```

然后怎么用？请百度自己查询

3. 在活动之间跳转，几乎是每一个 `android` 程序必备的，而活动之间的跳转是通过 `Intent` 实现的，`android` 组件之间传递数据，都是通过 `Intent` 来实现：

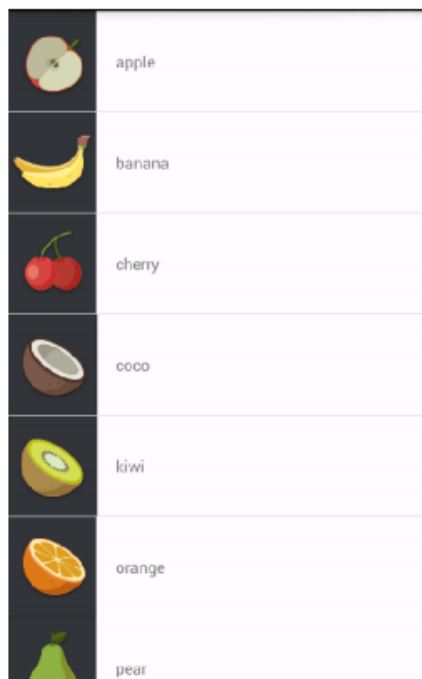
```
Intent intent = new Intent(MainActivity.this, SecondActivity.class);  
  
startActivity(intent);
```

因此，需要新建一个 `Activity`，并且在 `AndroidManifest.xml` 进行注册，否则程序会崩溃（`android` 所有的组件只有注册后才能够使用）：

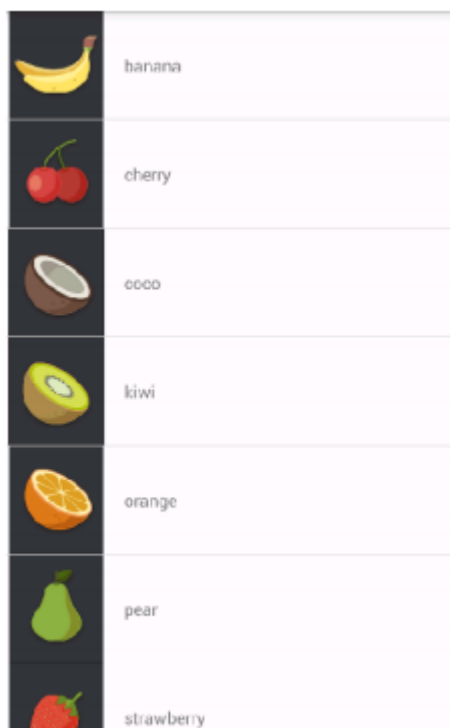
```
<activity  
    android:name=".SecondActivity"  
    android:label="@string/app_name" />
```

【扩展项】

在原来实验基础上，添加功能，使得长按 item 后自动删除该 item，比如：



长按 apple 后，该 item 被删除，如下图：



提示： `setOnItemLongClickListener`