

手机平台应用开发实验报告

学号： 1433198

班级： 周五下午班

姓名： 黄建武

实验名称： Lab4

一． 参考资料

<https://developer.android.com/guide/index.html> Android 官方文档

<http://www.cnblogs.com/meieiem/archive/2011/11/25/2263525.html> java 数组转 List

二． 实验环境

系统： Windows 10

工具： Android Studio 2.1.3

JDK： 1.8

minSdk： 19

targetSdk： 24

三． 实验目的

1. 掌握 Broadcast 编程基础
2. 掌握动态注册 Broadcast 和静态注册 Broadcast
3. 掌握 Notification 编程基础

四． 实验步骤

1. 阅读实验要求文档和课件。
2. 运行 TA 的 Demo 程序，了解实验要求细节。
3. 实现主页面的布局和点击跳转的逻辑。

实现了 OnClickListener 的接口，重写 onClick 函数，可以进行代码复用，无需每个按钮都实现一个 onClick 函数。

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    Button btn1, btn2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btn1 = (Button) findViewById(R.id.static_register);
        btn2 = (Button) findViewById(R.id.dynamic_register);
        btn1.setOnClickListener(this);
        btn2.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v == btn1) startActivity(new Intent(MainActivity.this, StaticRegisterActivity.class));
        else startActivity(new Intent(MainActivity.this, DynamicRegisterActivity.class));
    }
}

```

4. 根据上周的自定义适配器，自定义静态注册页面的 ListView 适配器 Fruit，构造函数接受一个 List<String>, 再根据不同的水果名，使用 Map 映射成不同的资源 ID，实现字符串和图片绑定，具体代码可看 Fruit.java 中的实现。

```

private void init() {
    String [] arr = new String[] {"Apple", "Banana", "Cherry", "Coco", "Kiwi", "Orange", "Pear", "Strawberry", "Watermelon"};
    final List<String> data = Arrays.asList(arr);
    Fruit adapter = new Fruit(this, data);
    ListView fruitlist = (ListView) findViewById(R.id.frutlist);
    assert fruitlist != null;
    fruitlist.setAdapter(adapter);
    fruitlist.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            sendBroadcast(new Intent("ml.huangjw.lab4.staticreceiver").putExtra("fruit", data.get(position)));
        }
    });
}

```

5. 实现静态注册页面的布局，设置适配器，发送广播和接收。

```

public void onReceive(Context context, Intent intent) {
    Bundle bundle = intent.getExtras();
    if (bundle != null) {
        String fruit = bundle.getString("fruit");
        NotificationManager nm = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
        Notification.Builder builder = new Notification.Builder(context);
        builder.setContentTitle("静态广播")
            .setContentText(fruit)
            .setPriority(Notification.PRIORITY_DEFAULT)
            .setLargeIcon(BitmapFactory.decodeResource(context.getResources(), map.get(fruit)))
            .setWhen(System.currentTimeMillis())
            .setSmallIcon(map.get(fruit))
            .setAutoCancel(true);
        PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, new Intent(context, MainActivity.class), 0);
        builder.setContentIntent(pendingIntent);
        Notification notification = builder.build();
        nm.notify(0, notification);
    }
}

```

Lagelcon 的参数是一个 Bitmap, 使用 BitmapFactory.decodeResource() 生成对应的图片的 Bitmap(),smallIcon 则可直接使用图片的 ID。

6. 对静态注册进行调试。

7. 实现动态注册页面的布局, 发送和接收广播。

有了静态广播的基础, 动态广播基本类似, 在点击时判断按钮状态, 注册或移除广播, 当接收到广播时多加一步进行判断

```
intent.getAction().equals("m1.huangjw.lab4.dynamicreceiver")
```

最后动态注册不需要再 AndroidManifest 里面进行注册 receiver, 一开始直接复制修改静态注册的代码, debug 了好久。

```
broadcast.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if (broadcast.getText().toString().equals("Register Broadcast")) {  
            broadcast.setText(R.string.UnregisterBroadcast);  
            registerReceiver(dynamicReceiver, new IntentFilter("m1.huangjw.lab4.dynamicreceiver"));  
        } else {  
            broadcast.setText(R.string.RegisterBroadcast);  
            unregisterReceiver(dynamicReceiver);  
        }  
    }  
});
```

8. 对动态注册进行调试。

9. 完善样式, 设置应用程序图标, 取消注册页面的标题栏。

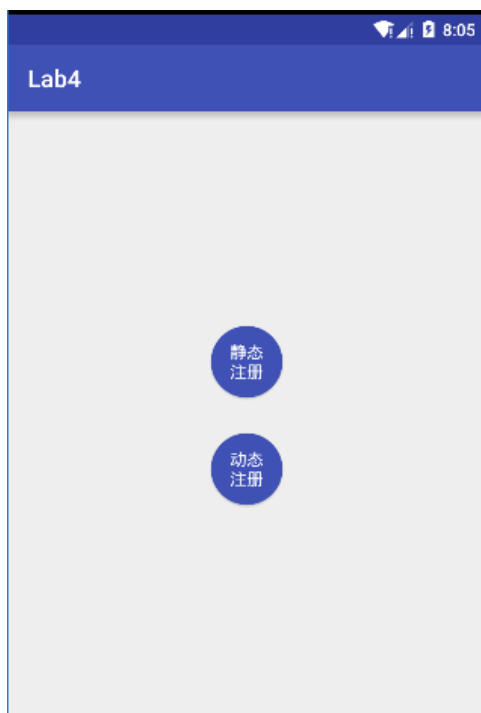
主页面有标题栏而其他两个没有, 一开始直接设置应用的主题为 NoActionBar, 结果三个页面都没有标题栏了, 最后直接在注册文件中设置对应 Activity 的样式。

```
<activity android:name=".StaticRegisterActivity"
    android:launchMode="singleInstance"
    android:theme="@style/Theme.AppCompat.Light.NoActionBar">
</activity>

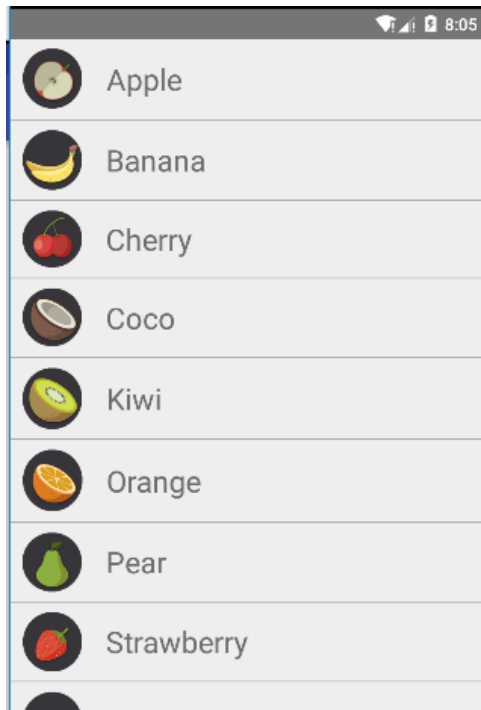
<activity android:name=".DynamicRegisterActivity"
    android:launchMode="singleInstance"
    android:theme="@style/Theme.AppCompat.Light.NoActionBar">
</activity>
```

五. 实验结果截图

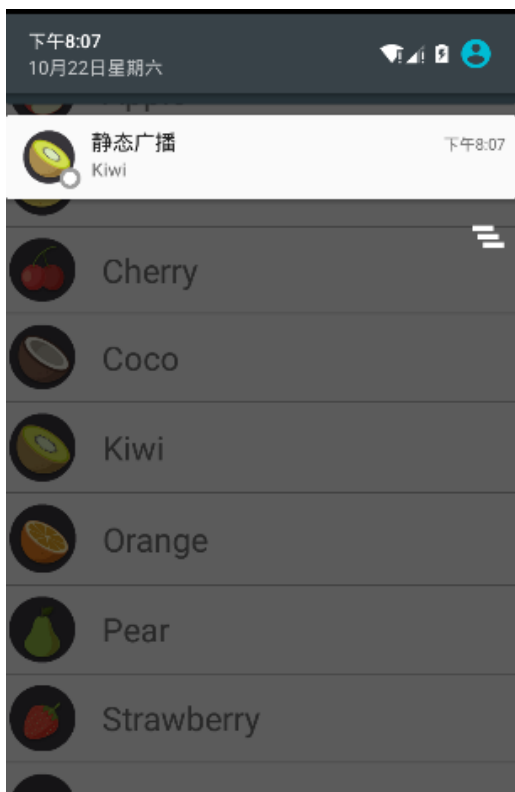
主页面



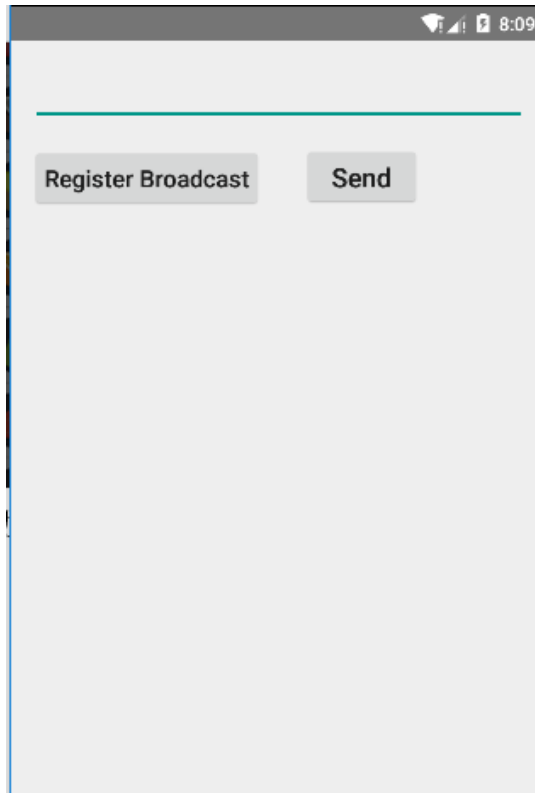
静态注册页面



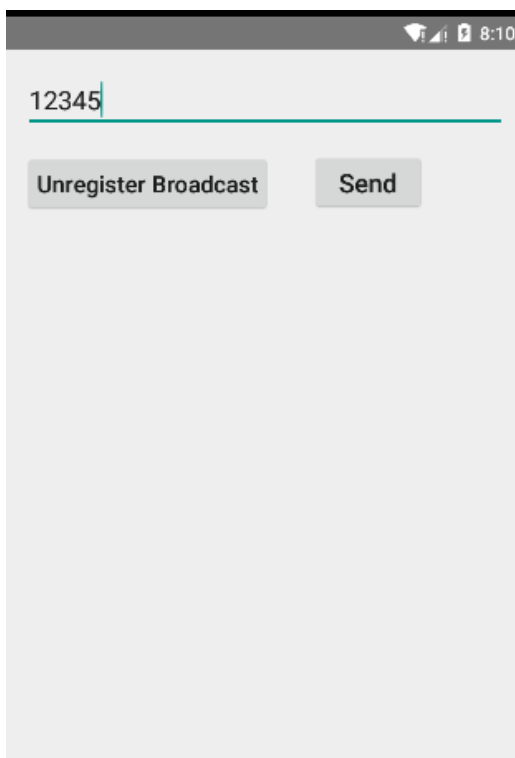
点击后发送广播，点击通知回到主界面



动态注册页面



注册广播



广播接收器已注册，点击发送，产生通知，点击回到主界面



六. 实验过程遇到的问题

1. 点击不同的水果通知里会产生不同的 icon, 在设置小图标时可以直接传 int 型的参数, 想了个方法, 把字符串和对应的图片用 map 进行映射

```
map = new HashMap<>();
map.put("Apple", R.mipmap.apple);
map.put("Banana", R.mipmap.banana);
map.put("Cherry", R.mipmap.cherry);
map.put("Coco", R.mipmap.coco);
map.put("Kiwi", R.mipmap.kiwi);
map.put("Orange", R.mipmap.orange);
map.put("Pear", R.mipmap.pear);
map.put("Strawberry", R.mipmap.strawberry);
map.put("Watermelon", R.mipmap.watermelon);
```

点击对应的 item 时通知就设置对应的图片, 可是 Map 的值类型不能设置为 int, 于是改为 Map<String, Integer>, 解决了设置

小 Icon 的问题。可是当设置大 Icon 时却行不通了，设置大 Icon 的函数参数没有 int 类型，无法使用这种方法。setLargeIcon 函数的参数是一个 Bitmap，BitmapFactory.decodeResource()函数可以传入图片 id 就可以生成 Bitmap。

2. 动态页面还没注册点击 send 就有通知产生，debug 了很久才发现是自己不理解动态广播，直接复制静态广播部分，导致动态广播在配置文件里已经注册了，相当于静态注册了，所以点 send 就直接产生了广播。

3. 换了 API24 后 Button 的文字全部变成了大写，上网查询后得知是 API 的问题，需要给 Button 指定属性

android:textAllCaps="false"，文字才显示正常。

七. 思考与总结

这次实验难度适中，只是自己还没弄清楚静态广播和动态广播就开始实验，导致出现了一些不该犯的错误，照搬静态广播的代码，在设置 action 和 filter 是前后不一致等问题，都使广播错，浪费了很多时间。通过这次实验，学习到了 Broadcast 和 Notification 编程的基础，也暴露出了一些问题，上周刚实现的自定义适配器这周就有点生疏了，只记得大概。这周的 Notification 代码也较复杂，课后需要花时间理解记忆，免得上机考手足无措。