



中山大學
SUN YAT-SEN UNIVERSITY

《手机平台应用开发》 Broadcast

学 院 名 称 : 数据科学与计算机学院

成 员 : 陈伟宸

时 间 : 2016 年 10 月 19 日

一、实验环境

操作系统：Windows 10

IDE：Android Studio 2.1.2

二、实验过程

1. 新建一个 empty project
2. 新建 StaticActivity.java、StaticReceiver.java、DynamicActivity.java、DynamicReceiver.java、static_layout.xml、item.xml、dynamic_layout.xml，在 AndroidManifest.xml 进行注册，并将 Activity 的运行模式设置为 singleInstance

```
<activity android:name=".MainActivity"
    android:launchMode="singleInstance">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".StaticActivity"
    android:launchMode="singleInstance">
    <intent-filter>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
<activity android:name=".DynamicActivity"
    android:launchMode="singleInstance">
    <intent-filter>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
<receiver android:name=".StaticReceiver">
    <intent-filter>
        <action android:name="STATICACTION"/>
    </intent-filter>
</receiver>
```

3. 设计开始页面。

在 activity_main.xml 中添加两个按钮，并设置好样式。

```
<Button
    android:id="@+id/static_register"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="静态注册"
    android:gravity="center"
    android:layout_marginTop="120dp"
    android:layout_gravity="center" />
<Button
    android:id="@+id/dynamic_register"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="动态注册"
    android:gravity="center"
    android:layout_marginTop="100dp"
    android:layout_gravity="center"/>
```

在 MainActivity.java 中，添加对上面新建的按钮的点击事件监听器，使它们分别跳转到 StaticActivity 和 DynamicActivity 中

```
static_register.setOnClickListener((view) -> {
    Intent intent = new Intent(MainActivity.this, StaticActivity.class);
    startActivity(intent);
});

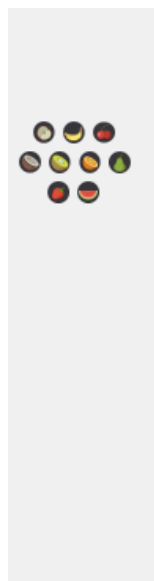
dynamic_register.setOnClickListener((view) -> {
    Intent intent = new Intent(MainActivity.this, DynamicActivity.class);
    startActivity(intent);
});
```

4. 设计静态注册页面。

在 static_layout.xml 中添加一个 ListView

```
<ListView
    android:id="@+id/fruit_list"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</ListView>
```

在 StaticActivity.java 中导入所有的水果名和图片，并使用 SimpleAdapter 适配刚刚创建的 ListView



```
List<Map<String, Object>> data = new ArrayList<>();
final String[] fruitsName = new String[] { "Apple", "Banana", "Cherry", "Coco", "Kiwi",
    "Orange", "Pear", "Strawberry", "Watermelon" };
final int[] fruitsImg = new int[] { R.mipmap.apple, R.mipmap.banana, R.mipmap.cherry,
    R.mipmap.coco, R.mipmap.kivi, R.mipmap.orange, R.mipmap.pear,
    R.mipmap.strawberry, R.mipmap.watermelon };
for (int i = 0; i < fruitsName.length; i++) {
    Map<String, Object> temp = new LinkedHashMap<>();
    temp.put("name", fruitsName[i]);
    temp.put("image", fruitsImg[i]);
    data.add(temp);
}

ListView fruit_list = (ListView) findViewById(R.id.fruit_list);
SimpleAdapter simpleAdapter = new SimpleAdapter(this, data, R.layout.item,
    new String[] { "name", "image" },
    new int[] { R.id.fruit_name, R.id.fruit_img });
fruit_list.setAdapter(simpleAdapter);
```

添加点击 ListView 中的 Item 的事件监听器，使用 Bundle 和 Intent 用于传递参数，并广播

```
fruit_list.setOnItemClickListener((adapterView, view, i, l) -> {
    Intent intent = new Intent("STATICACTION");
    Bundle bundle = new Bundle();
    bundle.putString("fruit_name", fruitsName[i]);
    bundle.putInt("fruit_img", fruitsImg[i]);
    intent.putExtras(bundle);
    sendBroadcast(intent);
});
```

4. 实现 StaticReceiver。

首先获得传递的参数，并使用它们设置标题、正文和图标等

```

Bundle bundle = intent.getExtras();

NotificationManager manager = (NotificationManager) context
    .getSystemService(Context.NOTIFICATION_SERVICE);
Notification.Builder builder = new Notification.Builder(context);

builder.setContentTitle("静态广播")
    .setContentText(bundle.getCharSequence("fruit_name"))
    .setLargeIcon(BitmapFactory.decodeResource(context.getResources(),
        bundle.getInt("fruit_img")))
    .setSmallIcon(bundle.getInt("fruit_img"));

```

实现点击回到主页面

```

Intent mIntent = new Intent(context, MainActivity.class);
PendingIntent mPendingIntent = PendingIntent.getActivity(context, 0, mIntent, 0);
builder.setContentIntent(mPendingIntent);
Notification notification = builder.build();
manager.notify(0, notification);

```

5. 设计动态注册页面。

在 dynamic_layout.xml 中添加一个 EditText 和一个 LinearLayout

用于存放两个 Button

```

<EditText
    android:id="@+id/edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <Button
        android:id="@+id/register_unregister"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Register Broadcast"
        android:layout_marginLeft="10dp"
        android:textAllCaps="false"
        android:tag="register"/>
    <Button
        android:id="@+id/send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:textAllCaps="false"
        android:text="Send"/>

```

在 DynamicActivity.java 中添加对第一个按钮的点击事件监听器，判断当前动作应该是注册还是注销，实现注册和注销，并能够在按钮上正确显示当前可以执行的操作

```

send.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(register_unregister.getTag().toString().equals("unregister")) {
            Intent intent = new Intent("DYNAMICACTION");
            Bundle bundle = new Bundle();
            bundle.putString("msg", editText.getText().toString());
            intent.putExtras(bundle);
            sendBroadcast(intent);
        }
    }
});

```

添加对 Send 按钮的点击事件监听器，传递 EditText 中的内容并进

行广播

6. 实现 DynamicReceiver。

首先获得传递的参数，并使用它们设置标题、正文等

```
Bundle bundle = intent.getExtras();

NotificationManager manager = (NotificationManager) context
    .getSystemService(Context.NOTIFICATION_SERVICE);
Notification.Builder builder = new Notification.Builder(context);

builder.setContentTitle("动态广播")
    .setContentText(bundle.getString("msg"))
    .setLargeIcon(BitmapFactory.decodeResource(context.getResources(), R.mipmap.dynamic))
    .setSmallIcon(R.mipmap.dynamic);
```

实现点击回到主页面

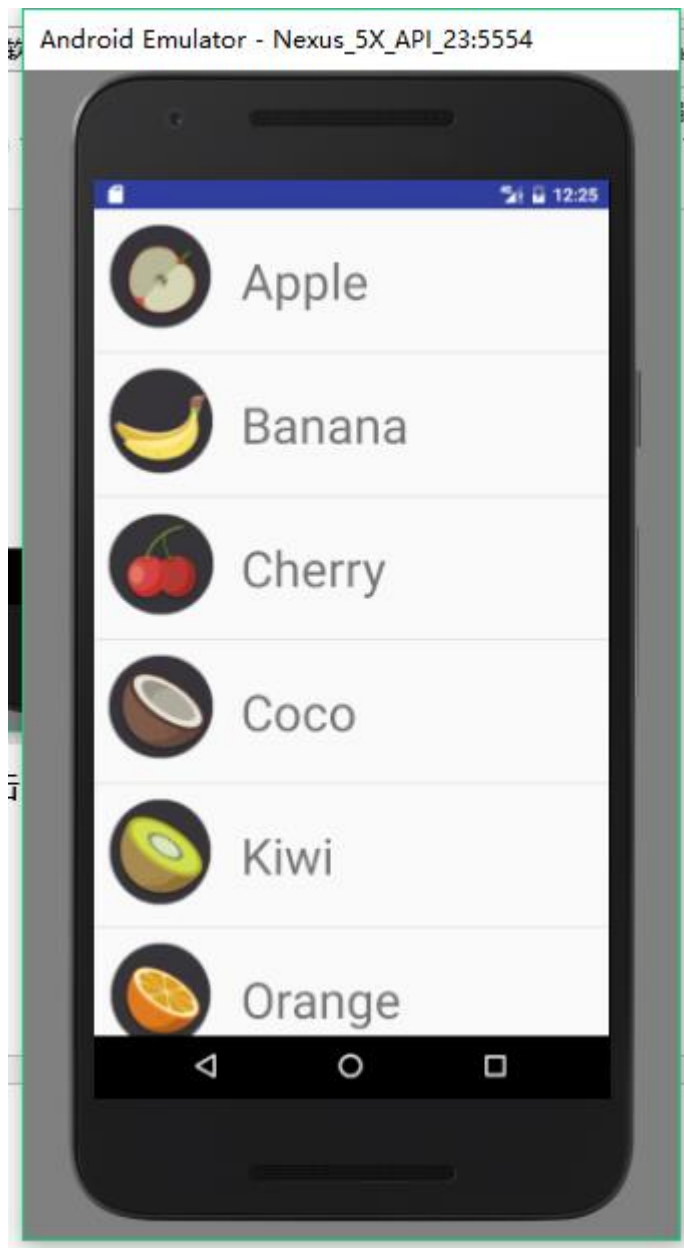
```
Intent mIntent = new Intent(context, MainActivity.class);
PendingIntent mPendingIntent = PendingIntent.getActivity(context, 0, mIntent, 0)
builder.setContentIntent(mPendingIntent);
Notification notification = builder.build();
manager.notify(0, notification);
```

三、实验结果

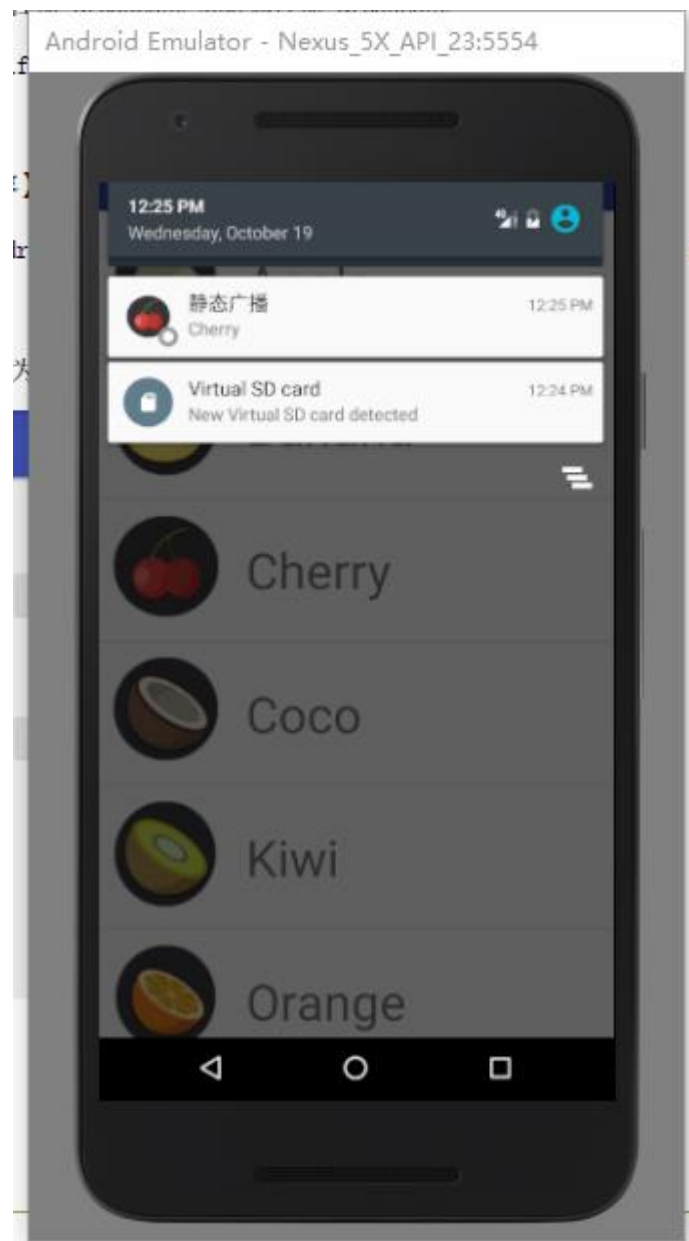
1. 运行程序



2. 点击“静态注册”



3. 点击 Cherry 并查看通知栏



4. 点击该通知



5. 点击“动态注册”



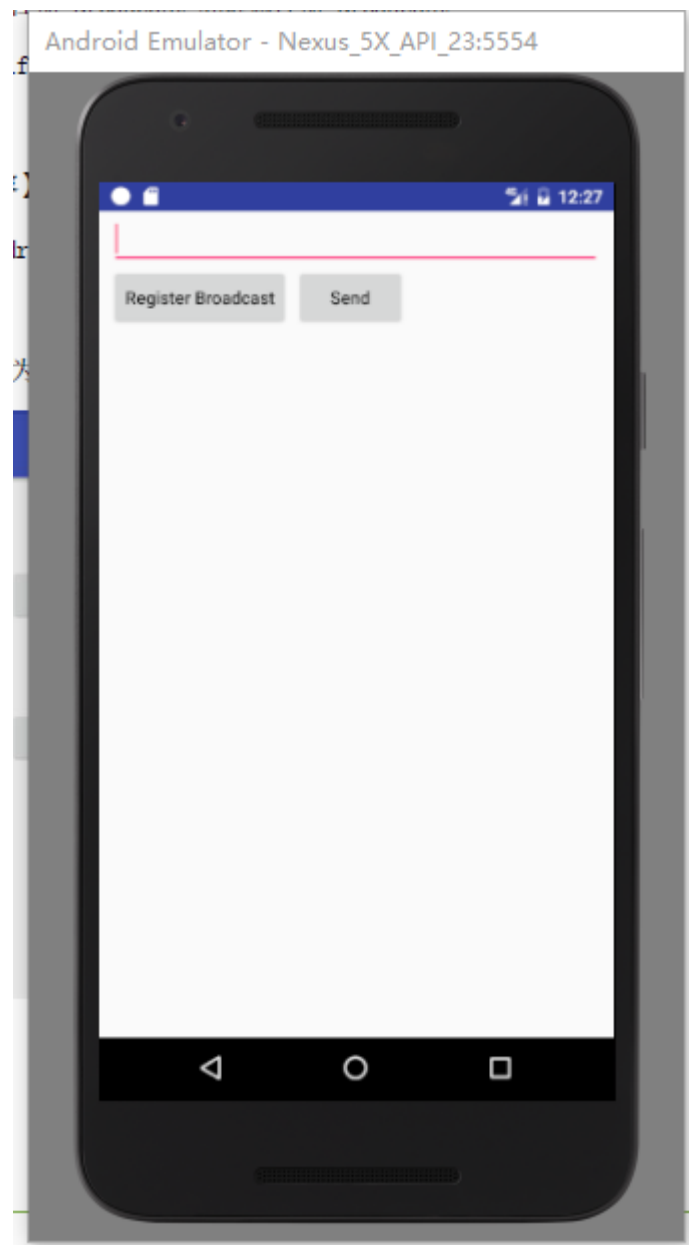
注意，通知栏小图标发生了变化

6. 查看通知栏



7. 点击该通知





6. 点击 "Send" , 什么也没有发生

7. 点击 "Register Broadcast"



8. 输入 “Splendidsky”，并点击 “Send”

四、实验心得

Broadcast，顾名思义就是广播，通过广播可以唤醒某个 Activity 处理某个事件。

Activity 需要通过注册才能处理某个广播事件，注册分为静态注册和动态注册。静态注册通过在 AndroidManifest.xml 中硬编码实现，无法注销；而动态注册使用 java 实现，在不需要的时候可以注销。

另外，广播时可以传递一些消息。

要操作通知栏需要先取得通知栏的 `manager`，再通过 `Notification.Builder` 来创建新的通知。