中山大學
SUN YAT-SEN UNIVERSITY

《手机平台应用开发（与 Google 共建）实验》

实验五：appwidget 及 broadcast 使用

实验报告

学 院 名 称 ：	数据科学与计算机学院
专　　业	软件工程（计应）
学 生 姓 名 ：	张凯鑫
学　　号 ：	14331362
班　　级 ：	周三上午 4-5 节、周五下午 7-8 节

【实验目的】

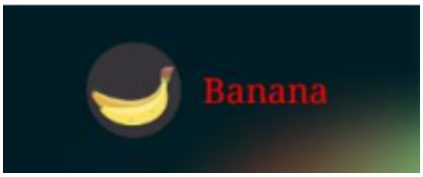
1. 掌握 AppWidget 编程基础；
2. 掌握 Broadcast 编程基础；
3. 掌握动态注册 Broadcast 和静态注册 Broadcast；

【实验内容】

实现一个 Android 应用，实现静态广播、动态广播两种改变 widget 内容的方法。
具体要求：

（1）该界面为应用启动后看到的界面（下图左），widget 初始情况如下（下图右）：

（2）点击静态注册按钮，跳转至如下界面（下图左），点击表单项目，如 banana，widget 会发生变化（下图右）。点击 Widget 上的图片可以跳转回主页面：



（3）点击动态注册按钮，跳转至如下界面。

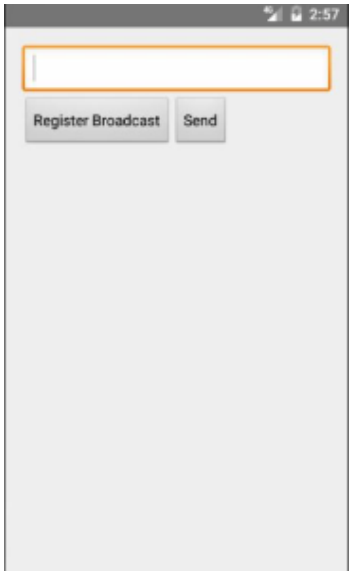
实现以下功能：

a) 可以编辑广播的信息，点击 Send 按钮发送广播。

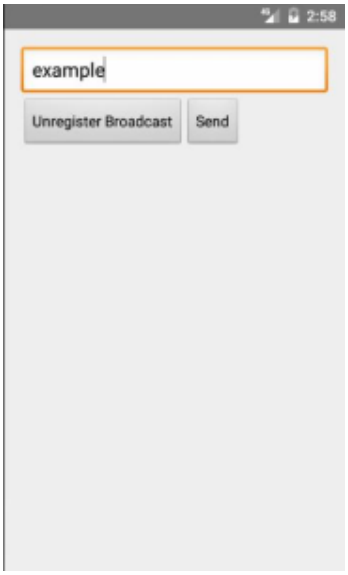
b) 设置一个按钮进行广播接收器的注册与注销。

c) 广播接收器若已被注册，发送出的广播信息能够及时更新桌面上 widget 上文字内容及更新为默认 dynamic 图片。

d) 点击 widget 上的图片可以跳转回主界面。



（未注册广播）



（已注册广播）

（已注册后点击发送）

注：在设置按钮内容的时候注意大小写问题。

【实验过程】

（关键实现代码见实验结果，与实验结果截图一起分析，此处暂不列出）

1. 在实验四的 Android Studio 项目上修改，命名改为：ExperimentFive。
2. 创建 Widget 类的 java 文件 WidgetDemo.java，自动生成 widget_demo_info.xml（放于 res/xml 目录下），生成 widget_demo.xml（放于 res/layout 目录下）。
3. 修改 widget_demo.xml，在其中添加控件 ImageView, TextView，设计文字大小为 20sp，颜色为红，整体背景透明，初始文字为“Widget”，初始图片为“苹果”。
4. 重写 WidgetDemo.java 中的重写 onUpdate 方法，为 widget 添加点击事件，使其能回到主页面。
5. 重写 WidgetDemo.java 中的 onReceive 方法，用于处理静态注册广播点击表单项目时 widget 的文字和图片更改，以及点击事件处理。
6. 重写 DynamicActivity.java 中的 onReceive 方法，用于处理动态注册广播发送消息后 widget 的文字和图片更改，以及点击事件处理。
7. 运行并调整实验代码直到实验完成。

【实验结果】

实验效果图如下（实验代码详见 lab5_code 文件夹）：

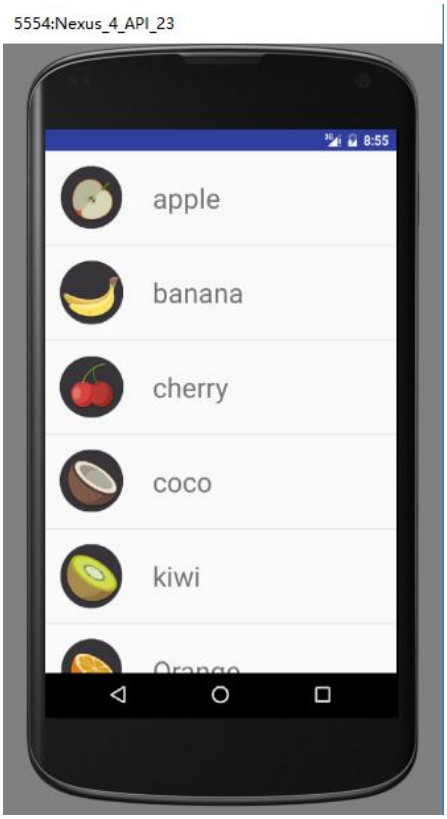
主界面

widget 初始状态

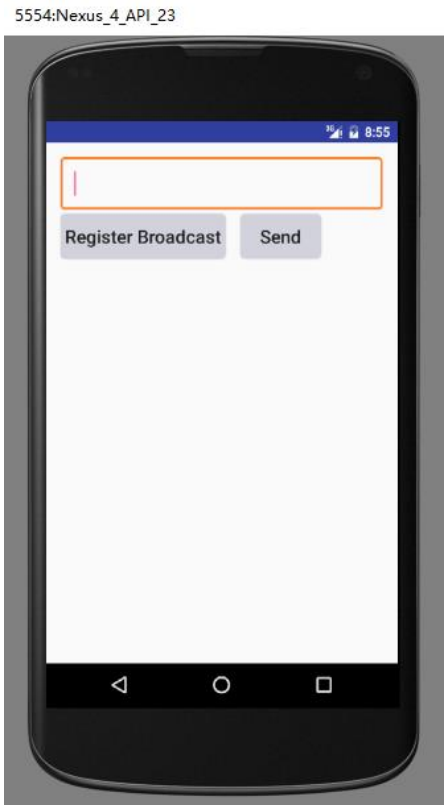
修改 WidgetDemo.java 代码，重写 onUpdate 方法，为 Widget 添加事件，使得每次创建 widget 后，图片和文字为为设置的“apple”和“Widget”，点击图片能够返回主界面：

```
public class WidgetDemo extends AppWidgetProvider{
    @Override
    public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {
        super.onUpdate(context, appWidgetManager, appWidgetIds);
        Intent clickInt = new Intent(context, MainActivity.class);
        PendingIntent pi = PendingIntent.getActivity(context, 0, clickInt, 0);
        RemoteViews rv = new RemoteViews(context.getPackageName(), R.layout.widget_demo);
        rv.setOnClickPendingIntent(R.id.widget_image, pi);
        appWidgetManager.updateAppWidget(appWidgetIds, rv);
    }
}
```

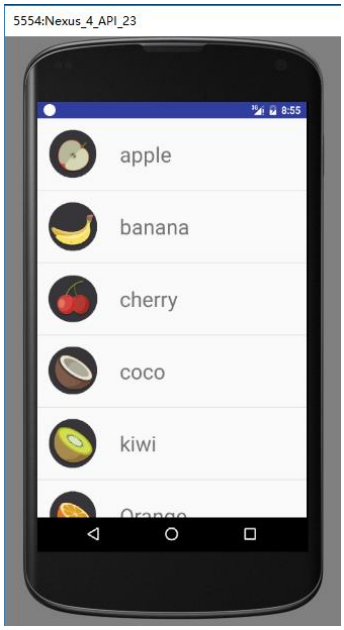
静态注册界面



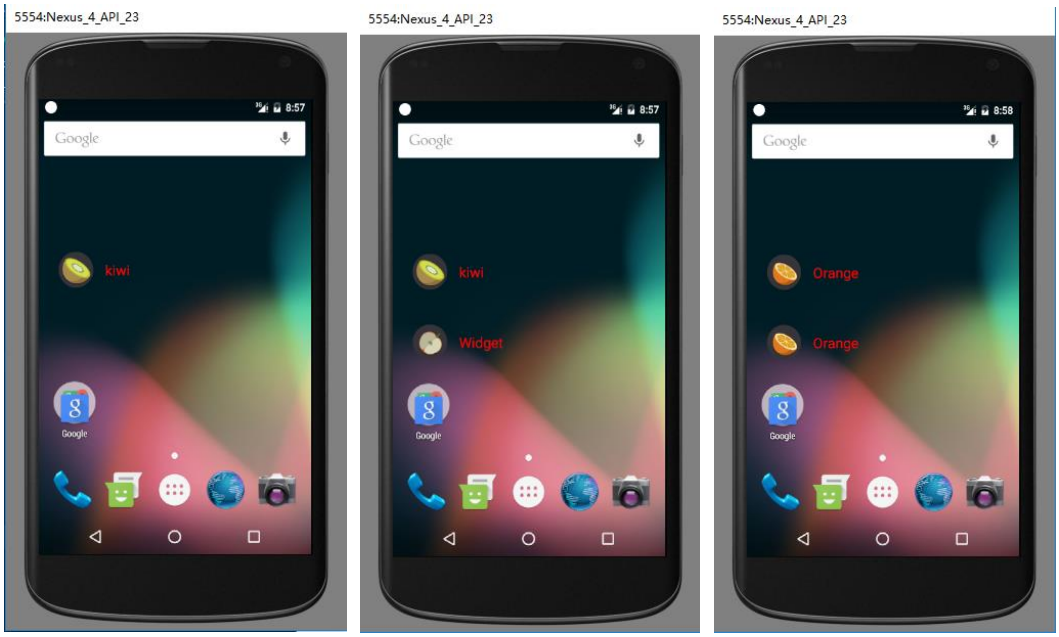
动态注册界面



静态注册示例：点击表单中任一项（如 banana）发送静态广播，返回主页，可看到 widget 的文字和图片随之改变。点击 widget 图片，跳转到主界面。（实验过程截图分别如下：）



接收到静态注册的广播时，widget 的图片和文字不是固定不变的，而是与被点击的表单中的项的图片文字一样，即与广播内容一致。另外，下面截图说明，新创建的 widget 保持初始图片和文字，而在接收到广播后，两个 widget 一起跟着改变内容。



关键代码截图如下：

在 AndroidManifest.xml 中静态注册广播：

```
<receiver android:name=".WidgetDemo">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE"/>
        <action android:name="com.example.kaixin.experimentfour.staticreceiver"/>
    </intent-filter>

    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widget_demo_info"/>
</receiver>
```

在 StaticActivity.java 中使用自定义 Adapter 配置 ListView 并设置表单的项的点击事件为发送静态广播，同时传递点击项的图片和文字消息：

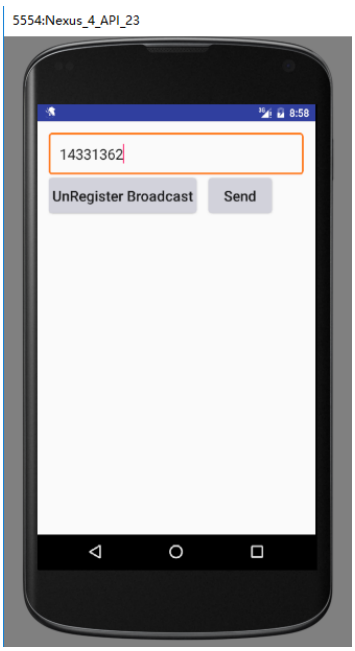
```
FruitAdapter adapter = new FruitAdapter(StaticActivity.this, R.layout.item, fruitList);
ListView fruit_lv = (ListView)findViewById(R.id.fruit_list);
fruit_lv.setAdapter(adapter);
fruit_lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Intent intent = new Intent("com.example.kaixin.experimentfour.staticreceiver");
        intent.putExtra("name",fruitList.get(position).getName());
        intent.putExtra("imageId", fruitList.get(position).getImageId());
        sendBroadcast(intent);
    }
});
```

在 WidgetDemo.java 中接收静态广播，重写 onReceive 函数：

```java
@Override
public void onReceive(Context context, Intent intent) {
    super.onReceive(context, intent);
    RemoteViews rv = new RemoteViews(context.getPackageName(), R.layout.widget_demo);
    if (intent.getAction().equals("com.example.kaixin.experimentfour.staticreceiver")) {
        String name = intent.getStringExtra("name");
        int imageId = intent.getIntExtra("imageId", 0);
        rv.setImageViewResource(R.id.appwidget_image, imageId);
        rv.setTextViewText(R.id.appwidget_text, name);
        Intent clickInt = new Intent(context, MainActivity.class);
        PendingIntent pi = PendingIntent.getActivity(context, 0, clickInt, 0);
        rv.setOnClickPendingIntent(R.id.appwidget_image, pi);

        AppWidgetManager appWidgetManger = AppWidgetManager.getInstance(context);
        int[] appIds = appWidgetManger.getAppWidgetIds(new ComponentName(context, WidgetDemo.class));
        appWidgetManger.updateAppWidget(appIds, rv);
    }
}
```

动态注册示例：点击按钮“Register Broadcast”注册动态广播，按钮文字变换为“UnRegister Broadcast”，输入框中输入要广播的信息，（如学号 14331362），点击按钮”Send”发送动态广播。返回手机主页，查看 widget 文字和图片，点击 widget 图标，跳转到主界面。截图依次如下：



点击按钮“UnRegister Broadcast”注销动态广播，按钮文字变换为“Register Broadcast”，输入框中输入要广播的信息，（如姓名 zhangkaixin），点击“send”按键，查看桌面 widget，发现不变，结果截图依次如下：

关键代码截图：

在 DynamicActivity.java 中设置按键“（Un）Register Broadcast”的点击事件：“Register Broadcast”按键点击事件为动态注册广播，并更改按键名：“UnRegisterBroadcast”按键点击事件为动态注销广播，并更改按键名：

```
btn_register.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (btn_register.getText().toString().equals("Register Broadcast")) {
            dynamicReceiver = new DynamicReceiver();
            IntentFilter dynamic_filter = new IntentFilter();
            dynamic_filter.addAction("com.example.kaixin.experimentfour.dynamicreceiver");
            registerReceiver(dynamicReceiver, dynamic_filter);
            btn_register.setText("UnRegister Broadcast");
        } else if (btn_register.getText().toString().equals("UnRegister Broadcast")) {
            unregisterReceiver(dynamicReceiver);
            btn_register.setText("Register Broadcast");
        }
    }
});
```

在 DynamicActivity.java 中处理“Send”按键的点击事件，输入字符串为空时，不处理，否则发送广播（注意：当广播未注册时，发送是不成功的，这点不需要再做判断）：

```
btn_send.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!TextUtils.isEmpty(input.getText())) {
            Intent intent = new Intent("com.example.kaixin.experimentfour.dynamicreceiver");
            intent.putExtra("name", input.getText().toString());
            sendBroadcast(intent);
        }
    }
});
```

在 DynamicReceiver 中接收动态广播，重写 onReceive 函数：

```
RemoteViews rv = new RemoteViews(context.getPackageName(), R.layout.widget_demo);
rv.setImageViewResource(R.id.appwidget_image, R.mipmap.dynamic);
rv.setTextViewText(R.id.appwidget_text, name);
Intent clickInt = new Intent(context, MainActivity.class);
PendingIntent pi = PendingIntent.getActivity(context, 0, clickInt, 0);
rv.setOnClickPendingIntent(R.id.appwidget_image, pi);

AppWidgetManager appWidgetManger = AppWidgetManager.getInstance(context);
int[] appIds = appWidgetManger.getAppWidgetIds(new ComponentName(context, WidgetDemo.class));
appWidgetManger.updateAppWidget(appIds, rv);
```

【遇到的问题与解决方法】

1、模拟器上 Nexus_4API_23 添加 widget 并不是长按菜单键触发选择，一开始一直找不到添加 widget 的方法，后来发现是长按桌面触发选择，点击 widgets 选项，找到对应的 widget 将其拖入桌面即可。

2、一开始不知道 Widget 可以直接创建，Widget 的文件是创建普通 java 文件之后写的，附带着的文件也是自己创建并写入的，产生了一些问题，比如手机桌面添加 widget 后，不能正常显示图片和文字，只是一个小小的灰色的方框，但点击方框后可跳转到主界面。后来直接删除相关代码，创建 widget 类，系统自动生成相关文件（WidgetDemo.java 、widget_demo.xml 和 widget_demo_info.xml ），并在 AndroidManifest.xml 中注册，重写 onUpdate 后，手机添加 widget，图片、文字显示正常，并可实现点击跳转到主界面。

3、实验过程中，widget 的初始文字和图片一开始设置为默认的“example”和“example_appwidget_preview”，之后按实验要求更改为“Widget”和“apple”，代码中已经修改，但在模拟器上运行一直不改变，最后发现，每次更改 widget 内容，最好将模拟器上的程度卸了重装。

4、本次实验动态注册广播界面有一个 EditText 控件，与上次实验不同，这个不是一条线而是一个方框输入，网上了解了一下，可以通过设置 background 来实行，类似于设置 Button 的 background，可以改变边框颜色、大小、角的弧度等，成功实现。

5、本次实验是在实验四代码的基础上更改的，一开始直接注释掉了 Notification 部分，后来实现 widget 功能后，想把 Notification 的功能补充进来，但发现动态部分没问题，静态部分 Notification 不能生效，最后发现是静态广播注册出错，Notification 静态接收广播的类为 StaticReceiver.java，在一开始被我处理掉了，补上相关代码如下，Notification 功能也实现了

```
<receiver android:name=".StaticReceiver">
    <intent-filter>
     <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
     <action
android:name="com.example.kaixin.experimentfour.staticreceiver" />
    </intent-filter>

</receiver>
```

【实验心得与体会】

1. 本次实验初步掌握了 AppWidget 的创建和简单使用，主要是 AppWidget 的 onUpdate 和 onReceive 方法，onUpdate 的实现基本遵循这样的流程：一是创建 RemoteViews，二是调用 AppWidgetManager 的 updateAppWidget 去更新 widget。 onUpdate 方法一般是当 Widget 被拖动到桌面上时被调用。onReceive 方法一般是在接收到广播后被调用，一般 onReceive 的实现流程和 onUpdate 类似：创建 RemoteViews 来设置所要修改的 widget 内容、和调用 updateAppWidget 方法更新。
2. 点击 widget 跳转到其他活动，这里用到的是 pendingIntent，这是一个特殊的 Intent，它不会马上执行，直到被触发。
3. RemoteView 架构允许用户程序更新主屏幕的 View，点击 Widget 激活点击事件，Android 会将其转发给用户程序，由 AppWidgetProviders 类处理，使用用户程序可更新主屏幕 Widget。
4. 本次实验还清楚了 EditText 的边框效果通过设置 background 实现的，因为出错（见“遇到的问题与解决方法”），对静态注册广播也有了更深入的了解。因为是在实验四的基础上实现的，很多内容其实已经实现，本次的实验相对简单。