

实验四 AppWidget 使用及 Broadcast 实现

【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

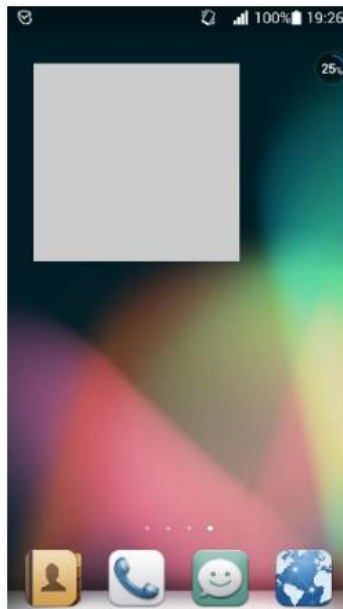
【实验内容】

实现以下功能：

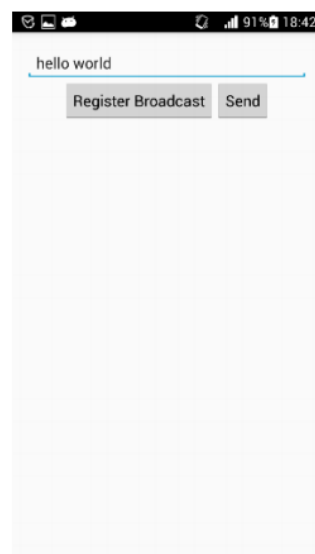
- a) 主页面可以编辑广播的信息，点击发送广播的按钮发送广播
- b) 主页面上设置一个按钮进行广播接收器的注册与注销
- c) 广播接收器若已被注册，发送出的广播信息能够及时更新桌面上 Widget 上文字内容
- d) 点击 Widget 可以跳转回主页面

参考截图：

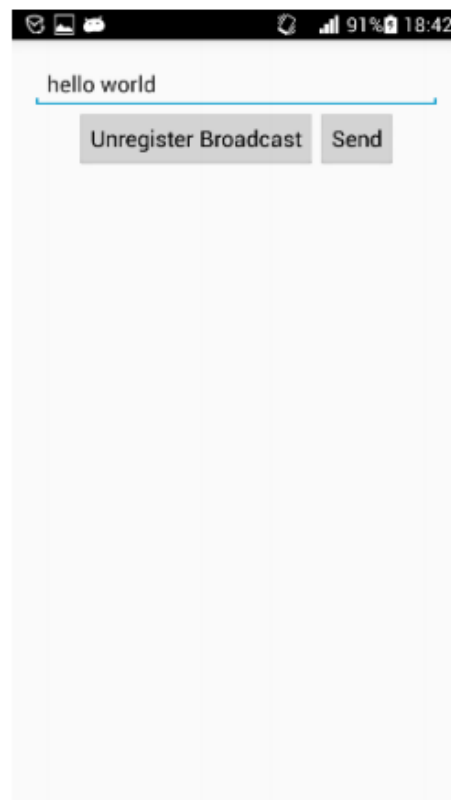
初始状态：



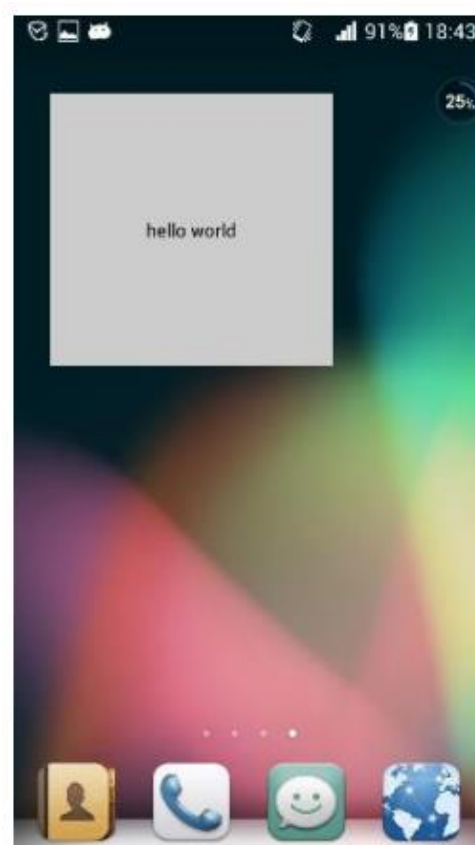
未注册广播：



已注册广播:

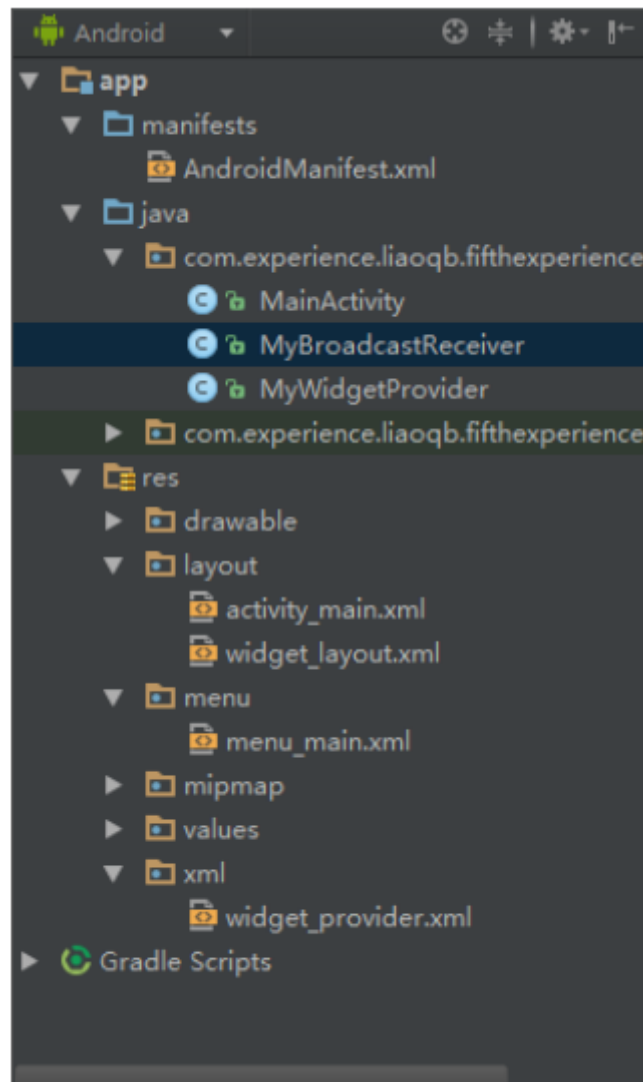


已注册广播时点击发送按钮:



【参考】

1. 参考文件目录如下：



2. 创建 Widget

- 1) 创建 Widget 布局文件 widget_layout.xml，布局中只有一个 TextView 用于显示文字
- 2) 在/res/xml（如果没有 xml 文件夹，自己创建一个）下创建类型为 AppWidget Provider 的 Android xml 文件 widget_provider.xml，并且编辑该文件，设置其大小属性和布局，如下图：

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="200dp"
    android:minHeight="80dp"
    android:initialLayout="@layout/widget_layout" />
```

其中，minWidth 为最小宽度，minHeight 为最小高度，initialLayout 为初始布局

- 3) 创建 AppWidgetProvider 子类（这里命名为 MyWidgetProvider），重写 onUpdate 方

法，为 Widget 添加事件，使得能够返回主页面：

```
public class MyWidgetProvider extends AppWidgetProvider {
    @Override
    public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {
        super.onUpdate(context, appWidgetManager, appWidgetIds);

        Intent intent = new Intent(context, MainActivity.class);
        PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, intent, 0);

        // use RemoteView to update Widget
        RemoteViews remoteViews = new RemoteViews(context.getPackageName(), R.layout.widget_layout);

        remoteViews.setOnClickPendingIntent(R.id.widget, pendingIntent);
        // update AppWidget
        appWidgetManager.updateAppWidget(appWidgetIds, remoteViews);
    }
}
```

并且在 AndroidManifest.xml 中注册 Widget：

```
<receiver
    android:name=".MyWidgetProvider"
    android:label="FifthExperience">
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widget_provider"/>

    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE"/>
    </intent-filter>
</receiver>
```

3. 使用动态方法创建 BroadcastReceiver：

1) 实现 BroadcastReceiver 子类（这里命名为 MyBroadcastReceiver），并且重写 onReceive 方法：

```
public class MyBroadcastReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        // use RemoteViews to update Widget
        RemoteViews remoteViews = new RemoteViews(context.getPackageName(), R.layout.widget_layout);
        // set editText
        remoteViews.setTextViewText(R.id.widget, intent.getStringExtra("message"));

        AppWidgetManager.getInstance(context).updateAppWidget(new ComponentName(context.getApplicationContext(),
            MyWidgetProvider.class), remoteViews);
    }
}
```

2) 创建主页面，设置注册和注销按钮，使其能够注册或者注销广播器

3) 设置按钮点击事件，如果广播注册，则能够点击后发送广播

```
public void onClick(View v) {  
    Intent intent = new Intent("SYSU_ANDROID_2015");  
  
    intent.putExtra("message", editText.getText().toString());  
  
    sendBroadcast(intent);  
}
```

4) 注意在 Android 主界面中将 launchMode 设置为 singleInstance, 使得点击 Widget 后不会另外新建一个 MainActivity:

```
<activity  
    android:name=".MainActivity"  
    android:launchMode="singleInstance"  
    android:label="FifthExperience">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN"/>  
  
        <category android:name="android.intent.category.LAUNCHER"/>  
    </intent-filter>  
</activity>
```

【扩展项】

在原来代码基础上, 使用静态 BroadcastReceiver 实现同样的功能。