

# 实验五

## appwidget 及broadcast 使用

### 【实验目的】

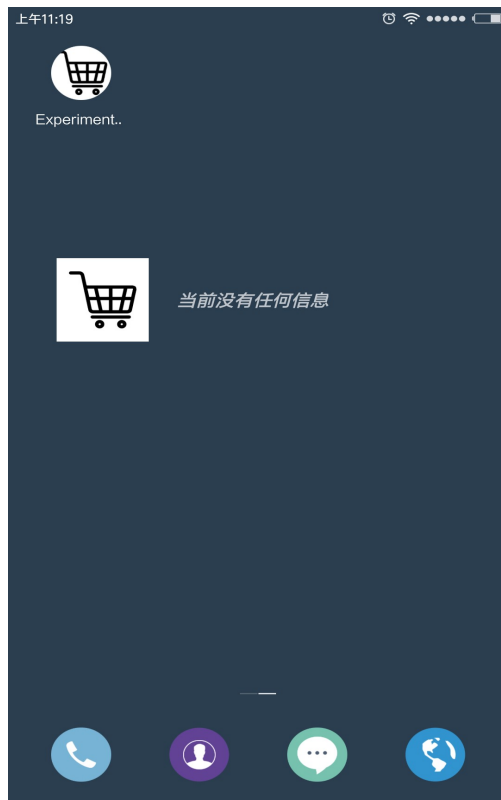
- 1、掌握 AppWidget 编程基础
- 2、掌握 Broadcast 编程基础
- 3、掌握动态注册 Broadcast 和静态注册 Broadcast

### 【实验内容】

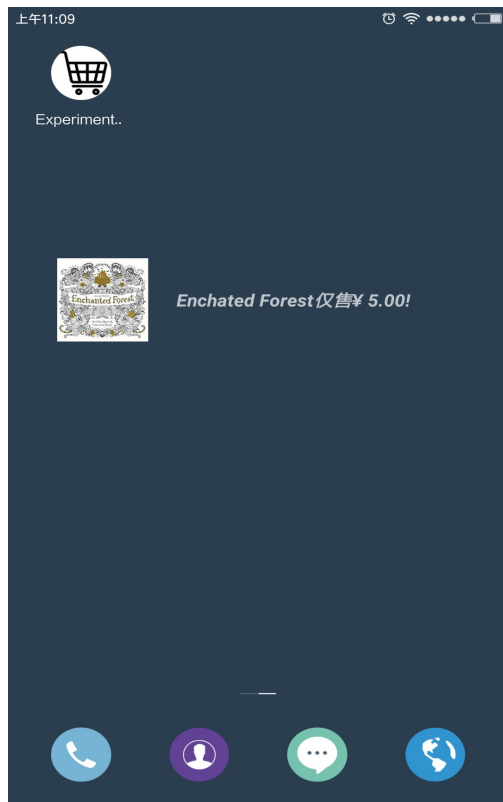
实现一个Android 应用，实现静态广播、动态广播两种改变widget 内容的方法。在上次实验的基础上进行修改，所以一些关于静态动态广播的内容会简略。

具体要求:

(1)widget 初始情况如下：



(2)点击widget可以启动应用，并在widget随机推荐一个商品:



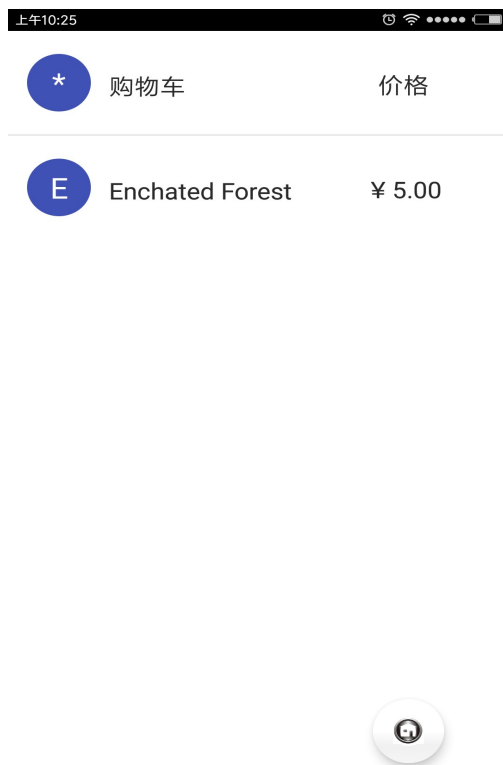
(3)点击widget跳转到该商品详情界面:



(4))点击购物车图标，widget相应更新:



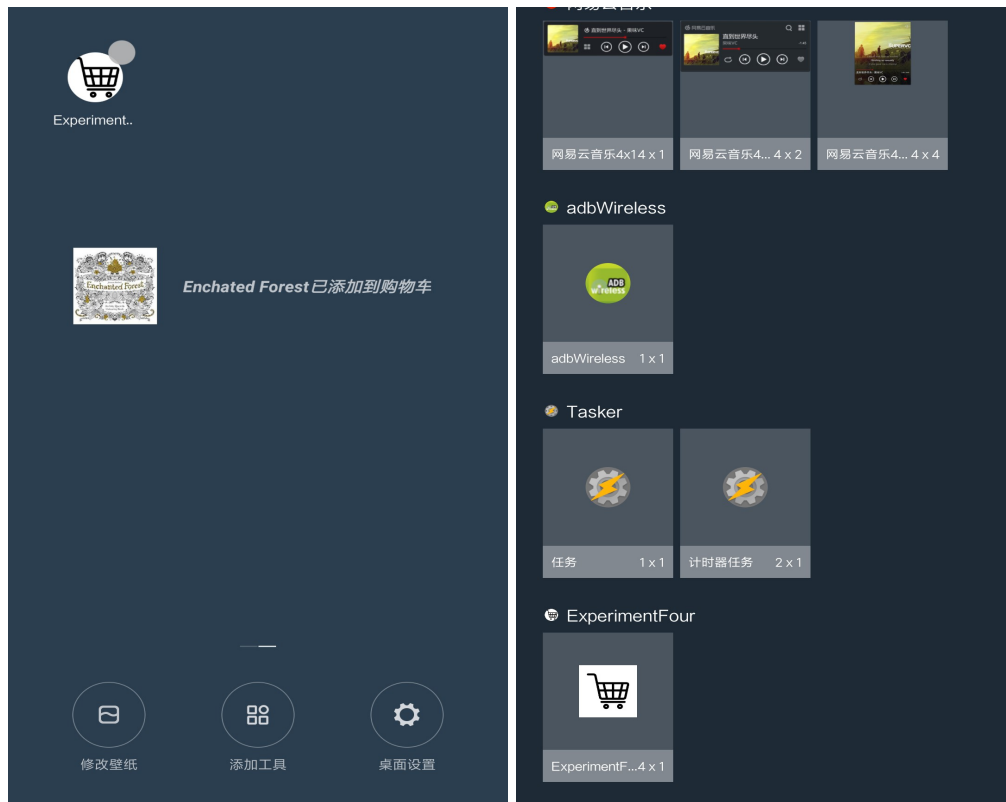
(5)点击widget跳转到购物车界面:



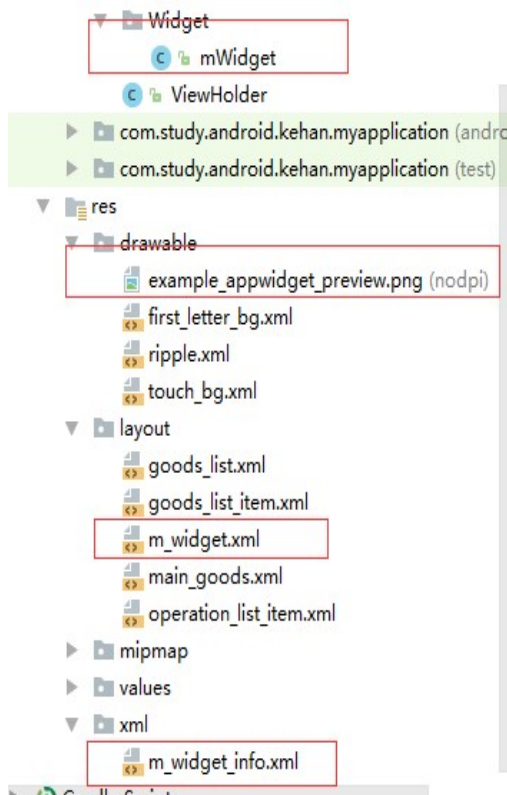
(6)实现方式要求:启动时的widget的更新通过静态广播实现，点击购物车图标时候widget的更新通过动态广播实现。

【参考内容】

(1)Widget widget 的添加：长按菜单键，点击widgets 选项。找到对应的widget 将其拖入桌面。对于不同的API 版本显示会稍有不同。



典型的Android Widget 有三个主要组件，一个边框、一个框架和图形控件以及其他元素。在 Android Studio 中创建Widget 类后，会直接生成相关文件。



1.Widget 布局文件m\_widget.xml，布局中有一个ImageView，一个TextView。要求：文字颜色为白色，大小为15sp，整体背景为透明。最后效果如下：



2.Widget 内容提供者文件m\_widget\_info.xml，编辑该文件，设置其大小属性和布局，如下图：

```
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/a:
    android:initialKeyguardLayout="@layout/m_widget"
    android:initialLayout="@layout/m_widget"
    android:minHeight="50dp"
    android:minWidth="300dp"
    android:previewImage="@mipmap/shoplist"
    android:resizeMode="horizontal|vertical"
    android:updatePeriodMillis="86400000"
    android:widgetCategory="home_screen|keyguard" />
```

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

3.修改mWidget.java 代码，重写 onUpdate 方法，为 Widget 添加事件，使得点击能够启动应用。

这里需要使用到一种用户程序访问主屏幕和修改特定区域内容的方法：RemoteView 架构。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，使得用户程序可更新主 屏幕Widget。

```
RemoteViews updateViews = new RemoteViews(context, getPackageName(), R.layout.m_widget); //实例化RemoteView, 其对应相应的Widget布局
Intent i = new Intent(context, GoodsList.class);
PendingIntent pi = PendingIntent.getActivity(context, requestCode: 0, i, PendingIntent.FLAG_UPDATE_CURRENT);
updateViews.setOnClickPendingIntent(R.id.widget, pi); //设置点击事件
ComponentName me = new ComponentName(context, mWidget.class);
appWidgetManager.updateAppWidget(me, updateViews);
```

pendingIntent 是一种特殊的Intent。主要的区别在于Intent 的执行立刻的，而 pendingIntent 的执行不是立刻的。本次使用方法类的静态方法为 getActivity(Context,int, Intent, int)，对应Intent 的跳转到一个activity 组件的操作。

(2)静态广播部分:

使用随机数:

```
Random random=new Random();
random.nextInt(n); //返回一个0到n-1的整数
```

利用bundle 和intent 将图片与文字内容发送出去:

```
Intent intentBroadcast = new Intent(STATICACTION); // 定义Intent
intentBroadcast.putExtras(bundle);
sendBroadcast(intentBroadcast);
```

参考代码中的STATICACTION 为自己设定的广播名称。由于是静态注册，所以需要在AndroidManifest.xml 中进行注册。注意这里用的是receiver是widget，当然可以不使用widget，使用别的receiver也可以。

```
<receiver android:name=".Widget.mWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="com.example.ex4.MyStaticPliter" />
    </intent-filter>

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

在Widget 类中重写onReceive 方法，这里需要使用到RemoteView 以及Bundle。当接收到对应广播时进行数据处理。

```
@Override
public void onReceive(Context context, Intent intent) {
    super.onReceive(context, intent);
    AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(context);
    Bundle bundle = intent.getExtras();
    if (intent.getAction().equals(STATICACTION)) {
    }
}
```

if 条件语句中主要用到的函数为：setTextViewText、setImageViewResource。之后使用AppWidgetManager 类对Widget 进行更新。

(3)动态广播部分 1.实现 BroadcastReceiver 子类（这里命名为DynamicReceiver），并且重写onReceive 方法，修改方法与静态广播类中类似。

```
public class DynamicReceiver extends BroadcastReceiver {
    private static final String DYNAMICACTION = "com.example.ex4.MyDynamicPliter"; //动态广播的Action字符串

    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(DYNAMICACTION)) { //动作检测
            Bundle bundle = intent.getExtras();
            //TODO:添加Notification部分
        }
    }
}
```

2.注册广播关键代码:

```
IntentFilter dynamic_filter = new IntentFilter();
dynamic_filter.addAction(DYNAMICACTION); //添加动态广播的Action
registerReceiver(dynamicReceiver, dynamic_filter); //注册自定义动态广播消息
```

注销广播关键代码:

```
unregisterReceiver(dynamicReceiver);
```

其中dynamicReceiver 为我们之前创建的DynamicReceiver 类。用registerReceiver与unregisterReceiver 分别对其进行注册与注销。

3.发送方法与静态注册时一直，仅需修改广播名称即可。（使用sendBroadcast(intent)）

#### 【检查内容】

- 1、布局显示是否正常。
- 2、静态广播：启动应用Widget是否有随机推荐商品。
- 3、动态广播：点击购物车后，Widget是否提示商品已加入购物车。
- 4、点击widget是否正确跳到对应的界面。

#### 【提交说明】

- 1、 deadline：下一次实验课前一天晚上12点
- 2、 提交作业地址：ftp://edin.sysu.edu.cn
- 3、 文件命名及格式要求：学号姓名labX.zip（姓名中文拼音均可）
- 4、 目录结构：

```
15331111_huashen_lab1 --
|
-- lab1实验报告.pdf
|
-- lab1_code (包含项目代码文件)
```