

中山大学移动信息工程学院本科生实验报告

(2017 年秋季学期)

课程名称：移动应用开发

任课教师：

年级	15 级	专业 (方向)	软件工程 (移动信息工程)
学号	15352223	姓名	刘朝开
电话	15626489095	Email	1209354383@qq.com
开始日期	2017.11.5	完成日期	2017.11.5

一、 实验题目

appwidget 及 broadcast 使用

二、 实现内容

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:

widget 初始情况如下：



点击 widget 可以启动应用，并在 widget 随机推荐一个商品：



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:



购物车

价格



Enchated Forest

¥ 5.00



(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。

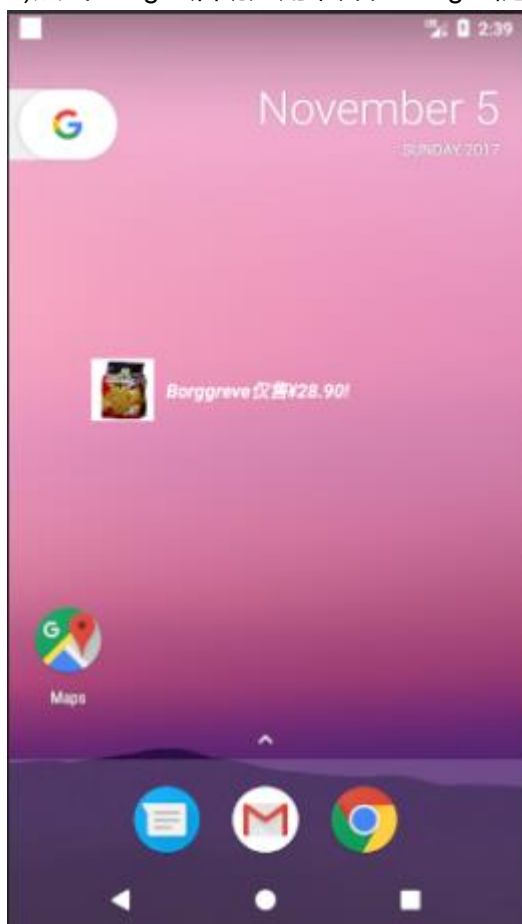
三、 课堂实验结果

实验截图

1) widget 初始情况如下：



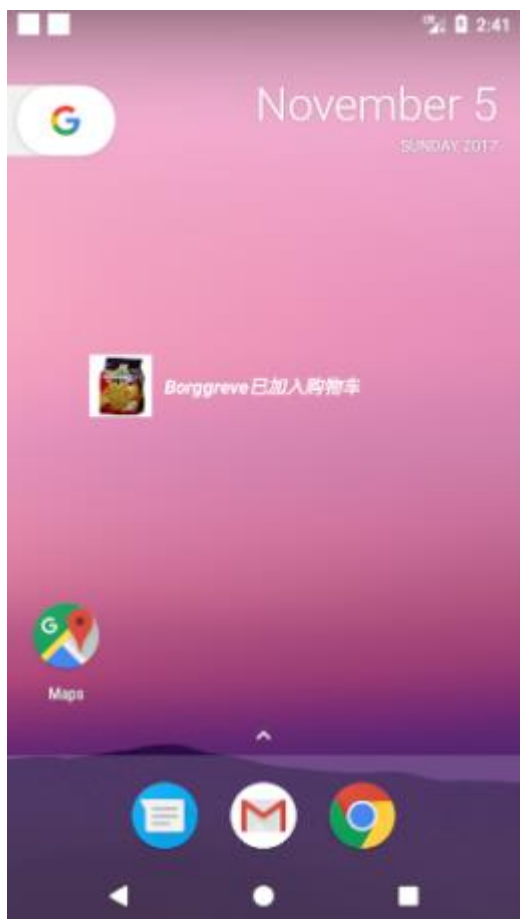
2) 点击 widget 启动应用，并在 widget 随机推荐一个商品:



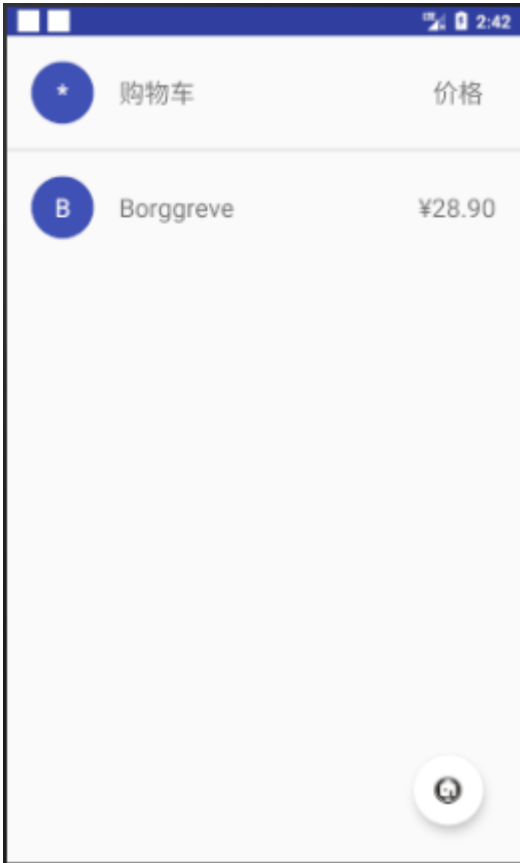
3)点击 widget 跳转到该商品详情界面:



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:



(1) 实验步骤以及关键代码

1) 新建 widget 命名为 Shoppingwidget,生成三个文件：Shoppingwidget.java, shoppingwidget.xml, shopping_widget_info.xml。

在 AndroidManifest.xml 文件中静态注册广播：

```
<receiver android:name=".ShoppingWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="com.example.liuchaokai.lab3_listview_recyclerview.STARTAPP"/>
    </intent-filter>
</receiver>
```

在 shopping.java 中动态注册广播。

```
//注册widget广播
intentFilter_w = new IntentFilter();
intentFilter_w.addAction("Order.Product");
pOrder_w = new ShoppingWidget();
registerReceiver(pOrder_w, intentFilter_w);
```

2) 编辑布局文件 shoppingwidget.xml 如下，包括一个 ImageView 用于显示商品图片，和一个 TextView，用于显示商品信息。

```

<ImageView
    android:id="@+id/shopwidget_view"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:src="@drawable/shoplist"
    android:layout_marginStart="50dp"
    android:layout_centerVertical="true"/>
<TextView
    android:id="@+id/shopwidget_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#00000000"
    android:contentDescription="@string/appwidget_text"
    android:layout_toEndOf="@+id/shopwidget_view"
    android:layout_marginStart="10dp"
    android:layout_alignTop="@+id/shopwidget_view"
    android:layout_marginTop="15dp"
    android:layout_alignBottom="@+id/shopwidget_view"
    android:layout_centerVertical="true"
    android:text="当前没有任何信息"
    android:textColor="#ffffff"
    android:textSize="15sp"
    android:textStyle="bold|italic" />

```

3) 编辑 shopping_widget_info.xml 文件如下：

```

<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:initialKeyguardLayout="@layout/shopping_widget"
    android:initialLayout="@layout/shopping_widget"
    android:minHeight="50dp"
    android:minWidth="300dp"
    android:previewImage="@drawable/shoplist"
    android:resizeMode="horizontal|vertical"
    android:updatePeriodMillis="86400000"
    android:widgetCategory="home_screen|keyguard">
</appwidget-provider>

```

minWidth 和 minHeight 分别指定了 App Widget 布局需要的最小宽度和高度，分别为 300dp 和 50dp。

previewImage 用于指定预览图。这里设为 shoplist。

resizeMode 指定了 widget 的调整尺寸的规则。这里设为可以横向和纵向拉伸。

updatePeriodMillis 它定义了 widget 的更新频率，实际最小更新时间为 30Minutes。

widgetCategory 指定了 widget 能显示的地方：这里设置可以显示在 home Screen 和 lock screen。

4) 编辑 ShoppingWidget.java 文件。主要需要重载的函数是 onReceive 函数和 onUpdate 函数：onUpdate 函数，在 widget 被添加或者更新时被调用，第一次被点击时进入 MainActivity 界面。

```

@Override // widget 被添加 || 更新时调用
public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {
    RemoteViews updateViews = new RemoteViews(context.getPackageName(), R.layout.shopping_widget);
    Intent intent = new Intent(context, MainActivity.class);
    PendingIntent pi = PendingIntent.getActivity(context, requestCode: 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);
    updateViews.setOnClickPendingIntent(R.id.shopwidget, pi);
    appWidgetManager.updateAppWidget(appWidgetIds, updateViews);
}

```

onReceive 函数主要接收两个广播执行相应的操作：

第一条广播：接收打开 App 广播，根据产生的随机数相应地改动 widget 信息。并设计点击事件为打开对应商品的详情界面。

```

if(intent.getAction().equals("com.example.liuchaokai.lab3_listview_recyclerview.STARTAPP"))
{
    //获取信息
    Bundle gBundle = intent.getExtras();
    int pIcon = gBundle.getInt( key: "pIcon");
    String pTile = gBundle.getString( key: "pTile");
    String pText = gBundle.getString( key: "pText");
    String pName = gBundle.getString( key: "pName");
    Log.d( tag: "tag", msg: "hihihi111");
    //设置显示内容
    RemoteViews updateViews = new RemoteViews(context.getPackageName(),R.layout.shopping_widget);
    Intent intent1 = new Intent(context,Shopping.class);
    intent1.putExtra( name: "name",pName);

    intent1.addCategory(Intent.CATEGORY_LAUNCHER);

    PendingIntent pi = PendingIntent.getActivity(context, requestCode: 0,intent1,PendingIntent.FLAG_UPDATE_CURRENT);
    updateViews.setImageViewResource(R.id.shopwidget_view,pIcon);
    updateViews.setTextViewText(R.id.shopwidget_text,pText);
    updateViews.setOnClickPendingIntent(R.id.shopwidget,pi);
    ComponentName me = new ComponentName(context,ShoppingWidget.class);
    AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(context);
    appWidgetManager.updateAppWidget(me,updateViews);
}

```

第二条广播：接收点击了购物车按钮的信息，并相应地改动 widget 的信息。并设计 widget 点击事件为跳转到购物车列表界面。

```

else if(intent.getAction().equals("Order.Product"))
{
    Bundle gBundle = intent.getExtras();
    int pIcon = gBundle.getInt( key: "pIcon");
    String pTile = gBundle.getString( key: "pTile");
    String pText = gBundle.getString( key: "pText");
    String pName = gBundle.getString( key: "pName");
    //设置显示内容
    RemoteViews updateViews = new RemoteViews(context.getPackageName(),R.layout.shopping_widget);
    Intent intent1 = new Intent(context,MainActivity.class);
    intent1.putExtra( name: "flag", value: 1);

    intent1.addCategory(Intent.CATEGORY_LAUNCHER);

    PendingIntent pi = PendingIntent.getActivity(context, requestCode: 0,intent1,PendingIntent.FLAG_UPDATE_CURRENT);
    updateViews.setImageViewResource(R.id.shopwidget_view,pIcon);
    updateViews.setTextViewText(R.id.shopwidget_text,pText);
    updateViews.setOnClickPendingIntent(R.id.shopwidget,pi);
    ComponentName me = new ComponentName(context,ShoppingWidget.class);
    AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(context);
    appWidgetManager.updateAppWidget(me,updateViews);
}

```

RemoteViews：它是一个远程视图。App Widget 中的视图，都是通过 RemoteViews 转换表现的。在 RemoteViews 的构造函数中，通过传入 layout 文件的 id 来获取“layout 文件对应的视图 (RemoteViews)”；调用 RemoteViews 中的方法能对 layout 中的组件进行设置 widgetViews.setOnClickPendingIntent(R.id.widget_btn, calendarIntent)来设 ID 对应的点击响应事件。

ComponentName：就是组件名称，通过调用 Intent 中的 setComponent 方法，我们可以打开另外一个应用中的 Activity 或者服务。实例化一个 ComponentName 需要两个参数，第一个参数是要启动应用的包名称，这个包名称是指清单文件中列出的应用的包名称，第二个参数是你启动的 Activity 或者 Service 的全称（包名+类名）。

AppWidgetManager：主要是对 AppWidget 进行管理，掌控 AppWidget 的更新信息等。

（2）实验遇到困难以及解决思路

困难：第一次将 widget 拖出界面无法响应点击事件，并且在拖出或者删除 widget 时会出现程序崩溃的现象。

分析：onReceive 函数除了接收我发送的广播之外，还会接收到系统发送的广播。一开始我是将接收信息的 gBundle 变量放在 if 语句即我所声明的广播之外的。如下图：

```
//获取信息
Bundle gBundle = intent.getExtras();
int pIcon = gBundle.getInt( key: "pIcon");
String pTitle = gBundle.getString( key: "pTitle");
String pText = gBundle.getString( key: "pText");
String pName = gBundle.getString( key: "pName");
Log.d( tag: "tag", msg: "hihihi111");
```

由于刚将 widget 拖出界面时，系统发出了一条广播但是没有对应的 Bundle 信息，而强行去 get，会导致程序崩溃。

解决方案：将上面的代码放入我自己设计的广播内容中，即 if 语句中。

四、 课后实验结果

五，实验思考及感想

这次试验基本承接着上次广播的实验，与上次实验的内容极为相似，所以做起来难度不是太大。关键要去理解的是两部分内容：一是 shopping_widget_info.xml 文件中各个变量设置的参数的理解。二是理解 widget 中各个函数的作用，什么时候被调用等。还有 RemoteViews，AppWidgetManager，ComponentName 的参数以及作用。

作业要求：

1. 命名要求: 学号_姓名_实验编号，例如 15330000_林 XX_lab1。
2. 实验报告提交格式为 pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按 0 分处理。