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： 郑贵锋

年级	大三	专业（方向）	移动互联网
学号	15352408	姓名	张镓伟
电话	13531810182	Email	709075442@qq.com
开始日期	2017.11.5	完成日期	2017.11.7

一、 实验题目

appwidget 及 broadcast 使用

二、 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

三、 实验内容

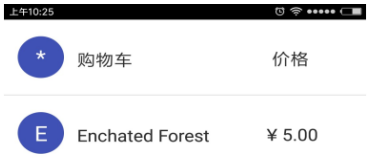
实现一个 Android 应用，实现静态广播、动态广播两种改变 widget 内容的方法。在上次实验的基础上进行修改，所以一些关于静态动态广播的内容会简略。具体要求:
(1)初始情况如下：

(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。



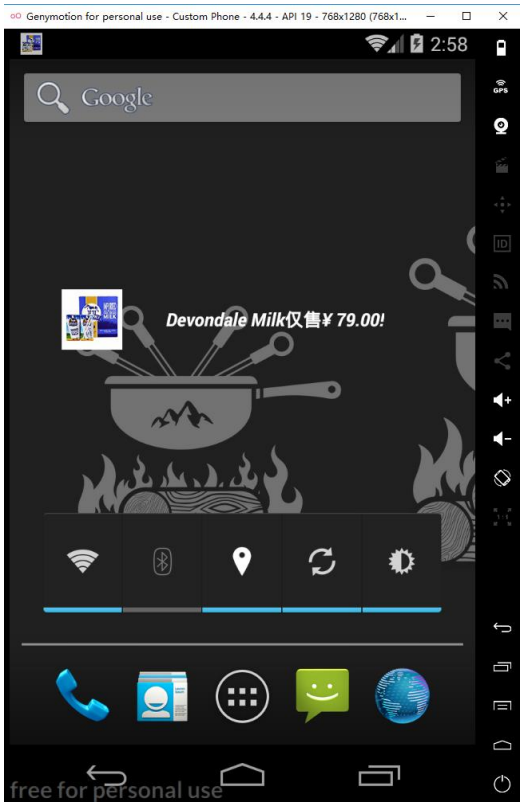
(6)实现方式要求:启动时的 widget 的更新通过静态广播实现，点击购物车图标时候 widget 的更新通过动态广播实现。

四、 课堂实验结果

（1） 实验截图

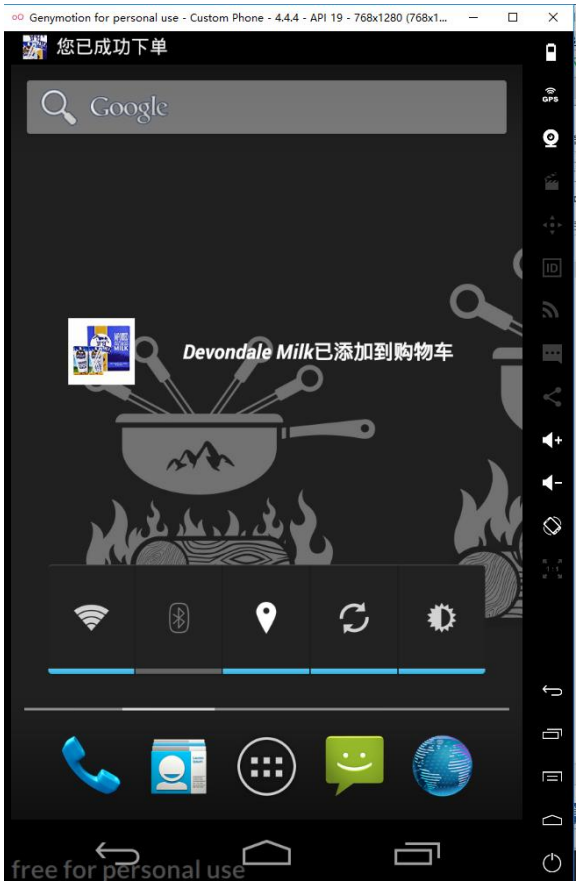
1. 初始情况

2.点击 widget 启动应用并随机推荐商品

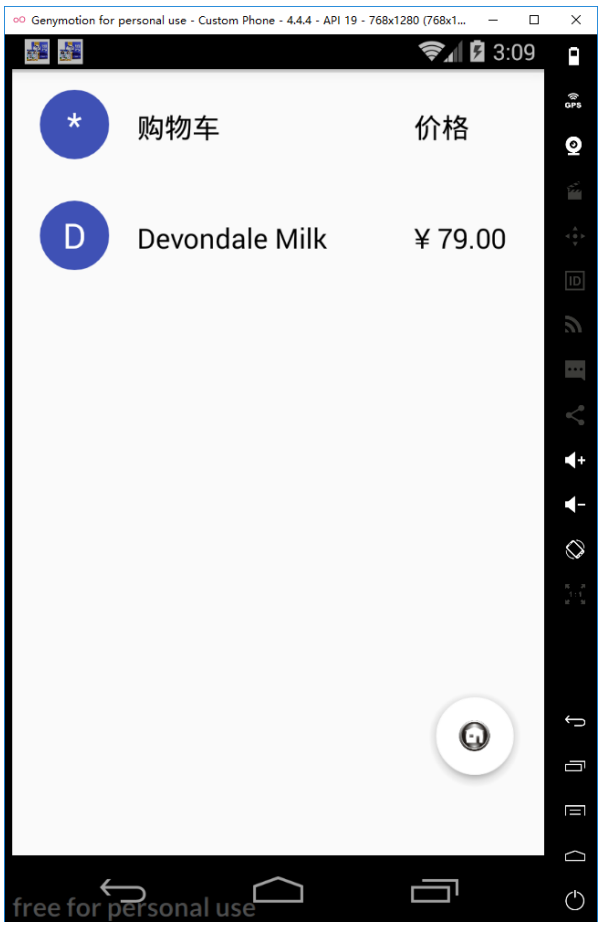


3.点击 widget，进入商品详情

4. 点击购物车图标，widget 相应更新



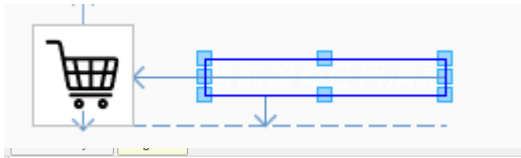
5. 点击 widget 进入购物车列表



（2） 实验步骤以及关键代码

Widget 布局文件：

1.布局是一个 imageview 和一个 textview



```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/appwidget">

    <TextView
        android:id="@+id/appwidget_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/appwidget_image"
        android:layout_marginBottom="15dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="36dp"
        android:layout_toEndOf="@+id/appwidget_image"
        android:contentDescription="EXAMPLE"
        android:text="当前没有任何消息"
        android:textColor="#ffffff"
        android:textSize="15sp"
        android:textStyle="bold|italic" />

    <ImageView
        android:id="@+id/appwidget_image"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_marginStart="15dp"
        android:layout_marginTop="15dp"
        app:srcCompat="@mipmap/shoplist"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true" />

</RelativeLayout>
```

Widget 初始化：

1. 新建一个 Widget 类，重写它的 onUpdate 方法，重写这个方法的目的是 初始化 widget 启动时的内容，通过 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。

```
@Override
public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {
    // There may be multiple widgets active, so update all of them
    for (int appWidgetId : appWidgetIds) {
        if(flag){
            init(context);
        }else
            updateAppWidget(context, appWidgetManager, appWidgetId);
    }
}

private void init(Context context){
    Intent i = new Intent(context, MainActivity.class);
    PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, i, PendingIntent.FLAG_UPDATE_CURRENT);
    RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.m_widget);
    views.setTextViewText(R.id.appwidget_text, "当前没有任何消息");
    views.setImageViewResource(R.id.appwidget_image, R.mipmap.shoplist);
    views.setOnClickPendingIntent(R.id.appwidget, pendingIntent);
    ComponentName me = new ComponentName(context, mWidget.class);
    AppWidgetManager.getInstance(context).updateAppWidget(me, views);
}
```

Widget onReceive：

重写 Widget onReceive 方法，当接收到对应广播时进行数据处理。这里分为动态广播和静态广播。其实内容跟上一次实验的 StaticReceiver 和 DynamicReceiver 基本上一致的。只是没有了 notification。这里换成了用 RemoteViews 和 AppWidgetManager 去更新显示的信息。

1. 静态广播

```
public void onReceive(Context context, Intent intent) {
    super.onReceive(context, intent);
    final String action = intent.getAction();
    if(action.equals(STATICACTION)) {
        flag = false;
        Bundle bundle = intent.getExtras();
        Name = bundle.getString("Name");
        Price = bundle.getString("Price");
        Info = bundle.getString("Info");

        Intent mIntent = new Intent(context, ItemsDetails.class);
        mIntent.putExtra("Name",Name);
        mIntent.putExtra("Price",Price);
        mIntent.putExtra("Info",Info);

        RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.m_widget);
        PendingIntent pendingIntent = PendingIntent.getActivity(context, 1, mIntent, PendingIntent.FLAG_UPDATE_CURRENT);
        views.setTextViewText(R.id.appwidget_text, Name + "仅售" + Price + "!");
        views.setImageViewResource(R.id.appwidget_image, ItemImage.getImg(Name));
        views.setOnClickPendingIntent(R.id.appwidget, pendingIntent);
        ComponentName me = new ComponentName(context, mWidget.class);
        AppWidgetManager.getInstance(context).updateAppWidget(me, views);
    }
```

2.动态广播

```
else if(action.equals(DYNAMICACTION)) {
    Bundle bundle = intent.getExtras();
    Name = bundle.getString("Name");

    Intent mIntent = new Intent(context, MainActivity.class);
    RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.m_widget);
    PendingIntent pendingIntent = PendingIntent.getActivity(context, 2, mIntent, PendingIntent.FLAG_UPDATE_CURRENT);
    views.setTextViewText(R.id.appwidget_text, Name + "已添加到购物车");
    views.setImageViewResource(R.id.appwidget_image, ItemImage.getImg(Name));
    views.setOnClickPendingIntent(R.id.appwidget, pendingIntent);
    ComponentName me = new ComponentName(context, mWidget.class);
    AppWidgetManager.getInstance(context).updateAppWidget(me, views);
}
```

广播的注册：

1.由于我将通过 boardcast 类的静态广播以及通过 widget 类的静态广播的广播名都设成一样的了，所以 java 代码不需要改动，只需要在 AndroidManifest.xml 中增加一个注册 widget 的静态广播即可。

```
<receiver
    android:name=".StaticReciver"
    android:enabled="true"
    android:exported="true">
    <intent-filter>
        <action android:name="com.example.freedom.lab5.MyStaticFliter" />
    </intent-filter>
</receiver>
```

新增部分

```
<receiver android:name=".mWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="com.example.freedom.lab5.MyStaticFliter" />
    </intent-filter>

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

2.动态广播需在 java 代码中新增注册

```
//动态广播注册
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
registerReceiver(dynamicWidgetReceiver, dynamic_filter);
```

新增

注销同理

（3） 实验遇到困难以及解决思路

这次实验还比较简单，基本将上一次的实验代码改一改就可以了。我一开始动态广播是没有更新到 widget 消息的，后来我发现是忘记注册广播了（由于静态部分 java 代码一个注册可以在两个地方用，所以动态部分我想当然也这样认为了）。实际上虽然广播名一样，但是用到的处理动态广播信息的类不一样，所以我们如果要同时保留 notification 和 widget 的话就需要注册两次。

五、 实验思考及感想

这次实验学习了 widget 的使用。Widget 是一种可以被放在其他应用中并周期性更新的应用视图。在本次实验里，它就是我们的桌面小部件。个人感觉这个东西可能更方便的可以在不打开应用的情况下，接收相关的消息。可以一直存在于桌面你指定的特定位置，不会像通知栏的消息会被刷走。