

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	15 级	专业 (方向)	软件工程(移动)
学号	15352133	姓名	黄少豪
电话	13727024545	Email	328730316@qq.com
开始日期	2017.11.6	完成日期	2017.11.7

一、 实验题目

appwidget 及 broadcast 使用

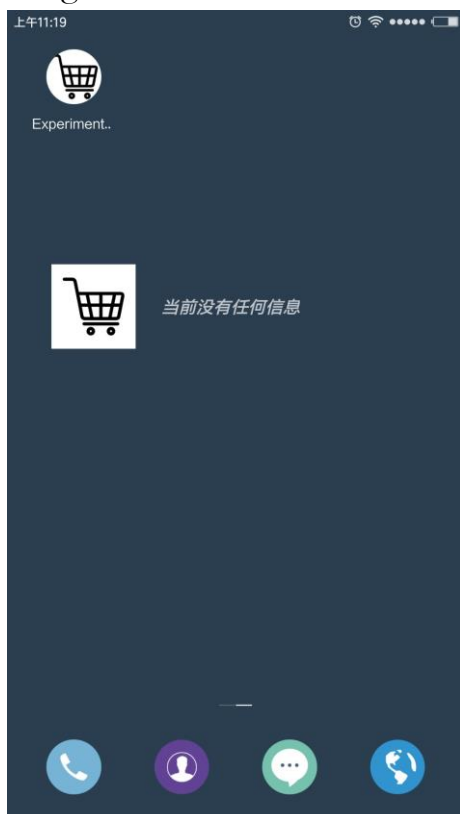
二、 实现内容

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：

(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:



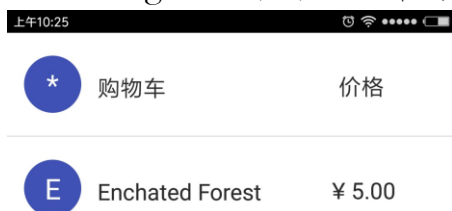
(3) 点击 widget 跳转到该商品详情界面:



(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。

三、 课堂实验结果

(1) 实验截图



Android Emulator - Nexus_6_API_25:5554



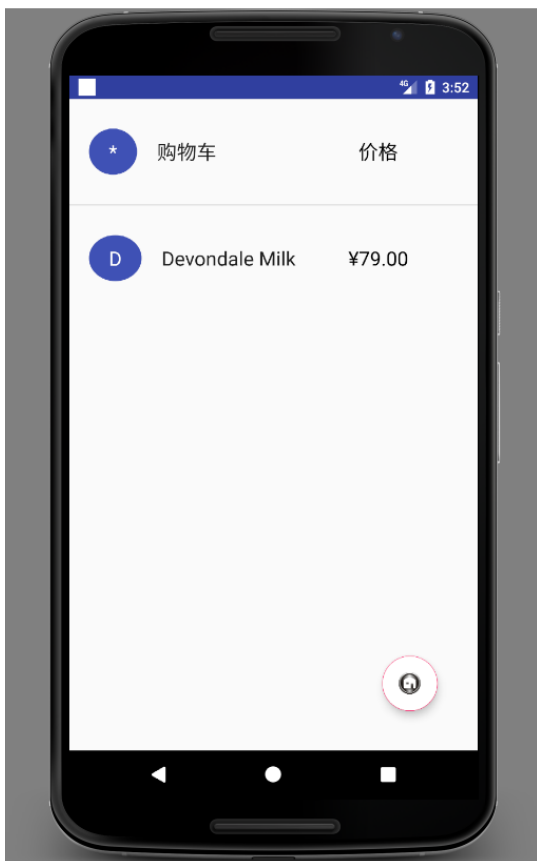
Android Emulator - Nexus_6_API_25:5554



Android Emulator - Nexus_6_API_25:5554

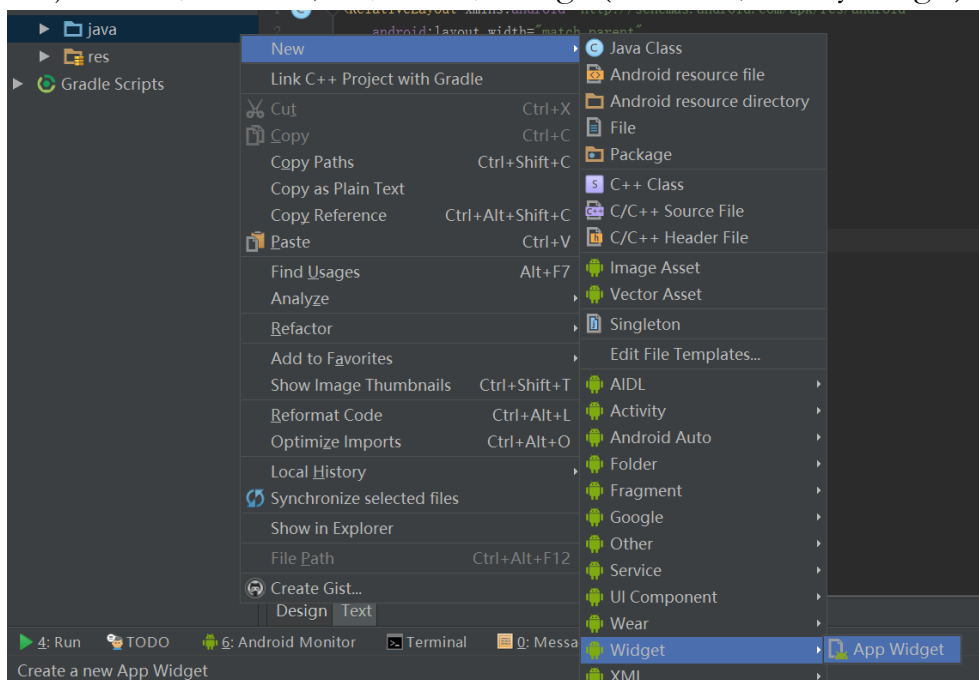


Android Emulator - Nexus_6_API_25:5554



(2) 实验步骤以及关键代码

1. 在 java 文件夹右击，新建一个 widget(这里名字为 MyWidget)



2. 在 my_widget.xml 中设置 widget 的布局(使用 RelativeLayout)

```
<ImageView
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/appwidget_image"
    android:src="@mipmap/shoplist"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"/>

<TextView
    android:id="@+id/appwidget_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@id/appwidget_image"
    android:layout_marginLeft="20dp"
    android:layout_centerVertical="true"
    android:contentDescription="@string/appwidget_text"
    android:text="@string/appwidget_text"
    android:textColor="#ffffff"
    android:textSize="15sp"
    android:textStyle="bold|italic"/>
```

3. 在 my_widget_info 中设置整个 widget 的样式与属性

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

4. 修改 AndroidManifest.xml，为 receiver 添加 action 表示 widget 监听名字为“STATICATION”的广播(静态广播)。

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

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

5. 在 MyWidget.java 中重写 MyWidget 类的 onUpdate 方法，表示在 Widget 初始化时为其配置点击事件，点击后启动应用。

```
@Override
public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {
    if (views == null) views = new RemoteViews(context.getPackageName(), R.layout.my_widget);
    //views是一个RemoteViews，负责在桌面中显示widget
    Intent i = new Intent(context, MainActivity.class);
    //i是一个intent，表示从当前界面跳转至应用主界面
    PendingIntent pi = PendingIntent.getActivity(context, 0, i, PendingIntent.FLAG_UPDATE_CURRENT);
    //pi是一个PendingIntent，包装i表示i这个intent需要满足某些条件才会启动
    views.setOnClickPendingIntent(R.id.appwidget_image, pi);
    //为widget的图片设置点击事件，点击后会启动pi包装的intent
    views.setOnClickPendingIntent(R.id.appwidget_text, pi);
    //为widget的文字设置点击事件，点击后会启动pi包装的intent
    if (me == null) me = new ComponentName(context, MyWidget.class);
    appWidgetManager.updateAppWidget(me, views);
    //使用appWidgetManager的updateAppWidget方法使views生效
}
```

6. 重写 MyWidget 类的 onReceive 方法，使其接收名字为“STATICATION”的广播(静态广播)。


```

@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("STATICATION")) {
        int pos = bundle.getInt("pos");
        String name = MainActivity.listItems.get(pos).get("name").toString();
        String price = MainActivity.listItems.get(pos).get("price").toString();
    }
}

```

广播会发送一个 pos 的整型数据，表示商品在商品列表的位置，调用主界面的商品列表获取商品的 name 与 price。

```

if (views == null) views = new RemoteViews(context.getPackageName(), R.layout.my_widget);
views.setTextViewText(R.id.appwidget_text, name + "仅售" + price);
views.setImageViewResource(R.id.appwidget_image, MainActivity.pic.get(pos));

Intent i = new Intent(context, InfoActivity.class);
i.putExtra("pos", pos);
PendingIntent pi = PendingIntent.getActivity(context, 0, i, PendingIntent.FLAG_UPDATE_CURRENT);

views.setOnClickPendingIntent(R.id.appwidget_image, pi);
views.setOnClickPendingIntent(R.id.appwidget_text, pi);

if (me == null) me = new ComponentName(context, MyWidget.class);
appWidgetManager.updateAppWidget(me, views);

```

接着设置 views，views 是一个 RemoteViews 对象，将 views 的图片换成商品图片，文字换成“商品名字仅售价格”的格式；为 views 的文字与图片设置点击事件，使其被点击后能跳转至商品详情界面。

7. 动态广播更新 widget 状态

在上一个实验的基础上修改 DynamicReceiver 类，在其 onReceive 方法中添加如下代码：

```

if (views == null) views = new RemoteViews(context.getPackageName(), R.layout.my_widget);
Intent i = new Intent(context, MainActivity.class);
PendingIntent pi = PendingIntent.getActivity(context, 0, i, PendingIntent.FLAG_UPDATE_CURRENT);
views.setOnClickPendingIntent(R.id.appwidget_image, pi);
views.setOnClickPendingIntent(R.id.appwidget_text, pi);
views.setTextViewText(R.id.appwidget_text, MainActivity.listItems.get(pos).get("name").toString() + "已添加到购物车");
if (me == null) me = new ComponentName(context, MyWidget.class);
if (appWidgetManager == null) appWidgetManager = AppWidgetManager.getInstance(context);
appWidgetManager.updateAppWidget(me, views);

```

这里其实就是收到广播后修改 RemoteViews 对象，使桌面的 Widget 状态发生改变(桌面的 Widget 由 RemoteViews 对象控制)。创建一个 intent 对象表示跳转至购物车界面；使用 PendingIntent 对象包装这个 intent 对象；为 Widget 的图片与文字创建点击事件表示点击后触发 PendingIntent 对象。

8. 静态广播的发送与动态广播接收器的注册，这里直接沿用上一个实验的代码

```
//静态广播发送
public void initBroadcast(){
    Random random = new Random();
    int pos = random.nextInt(10) + 1;
    Bundle bundle = new Bundle();
    bundle.putInt("pos", pos);
    Intent intentBroadcast = new Intent("STATICATION");
    intentBroadcast.putExtras(bundle);
    sendBroadcast(intentBroadcast);
}
```

```
//动态广播接收注册
dynamicReceiver = new DynamicReceiver();
dynamic_filter = new IntentFilter();
dynamic_filter.addAction("DYNAMICATION");
registerReceiver(dynamicReceiver, dynamic_filter);
```

(3) 实验遇到困难以及解决思路

1. 在 AndroidStudio 中创建 Widget 的问题

一开始不知道原来可以直接右击 java 文件夹新建 widget，于是自己手动创建各个文件但出现各种问题无法运行。解决方法就是直接右击 java 文件夹新建 widget，AndroidStudio 会自动帮你创建需要的各个文件。

2. Widget 在手机桌面的排版问题

不同商品的图片文字组合在桌面 widget 的显示无法固定，一些是正常显示，但也有一些排版很奇怪不符合要求。后来查看图片属性发现图片的宽高大小是不一样，所以导入图片显示时才会出现不同的问题。在 widget 的 xml 文件中设置一下 ImageView 的 width 与 height 即可解决问题。

3. 动态广播更新 widget 状态的问题

主要还是没有理清静态与动态的概念，一开始我的实现方法就是直接在 MyWidget 类的 onReceive 方法中再加一个 if 语句，在接收到名字为“DYNAMICATION”的广播后执行操作。但后来发现这依然是静态广播的方法。区分静态与动态主要是看广播接收器对象是如何创建的。在 xml 文件中描述创建的是静态的，在 java 文件中创建的就是动

态的。这里 widget 的创建是在 AndroidManifest.xml 文件中描述的，所以是静态的。要实现动态接收广播，首先接收器要在 java 文件中创建的，这里使用的是上一个实验的 BroadcastReceiver 对象。在这个对象的 onReceive 函数中加上对 RemoteViews 对象的处理即可实现动态广播改变 widget 状态了。

四、 课后实验结果

五、 实验思考及感想

1. 通过本次实验，加深了对静态与动态区别的体会。之前一直对静态与动态的区别感到模糊，现在算是理清。静态是在应用运行前就已确定好，是在 xml 文件中描述创建的；动态是在运行时才能确定是否创建，是在 java 文件中创建的。静态的好处是对象的创建与 java 文件无关，当存在非常多的对象时，在 xml 文件中描述创建的对象会比较好管理；动态的好处就是对象可以通过一些条件判断来确定是否创建，在某种程度上可以节约资源。
2. 大体理清了 widget 的运行机制。Widget 在 android 中是使用 RemoteViews 对象进行管理的。Widget 需要什么布局，里面的 view 需要显示什么，需要有什么动作都可以在 RemoteViews 对象中进行设置。