

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	15M1	专业（方向）	移动互联网
学号	15352008	姓名	蔡荣裕
电话	13727021990	Email	897389207@qq.com
开始日期	2017/10/24	完成日期	2017/10/30

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。

1. widget 初始情况如右图 1。
2. 点击 widget 可以启动应用，并在 widget 随机推荐一个商品，如图 2
3. 点击 widget 跳转到该商品详情界面，如图 3



图 1



图 2



图 3

4. 点击购物车图标，widget 相应更新如图 4
5. 点击 widget 跳转到购物车界面，如图 5
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。



图 4

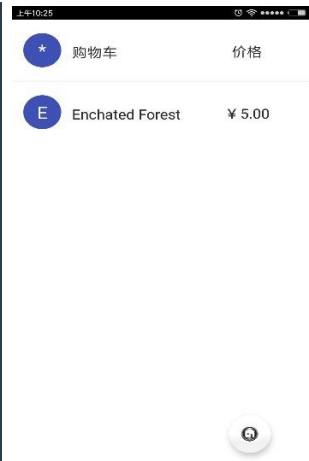
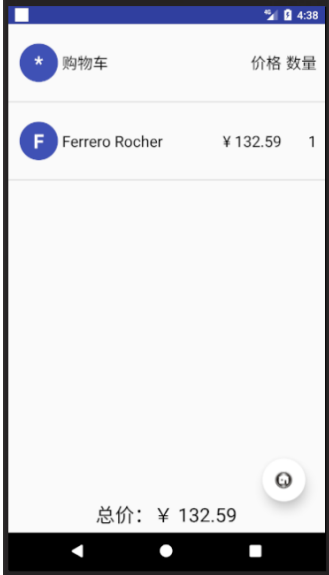
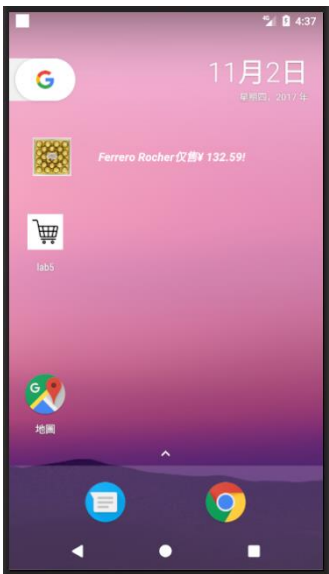
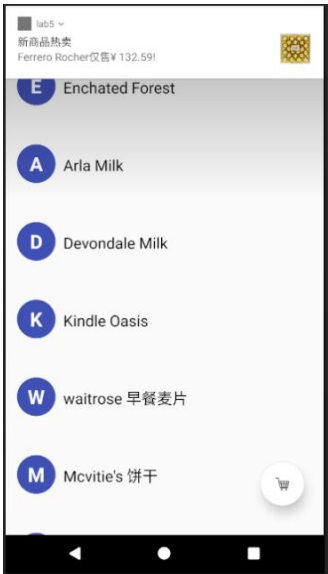


图 5

四、 课堂实验结果

（一） 实验截图：

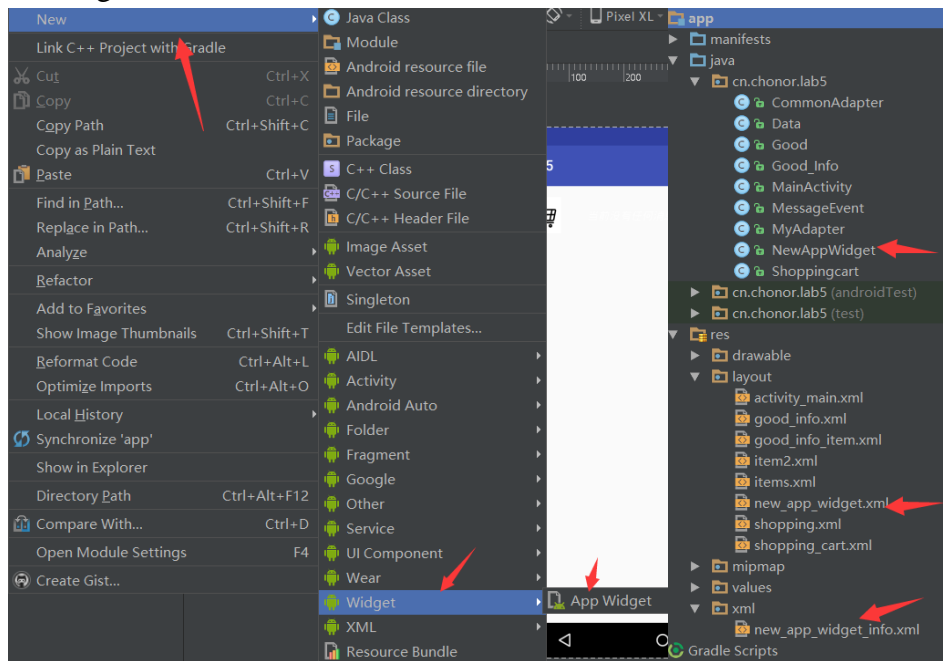
1. 虚拟机上效果如下：



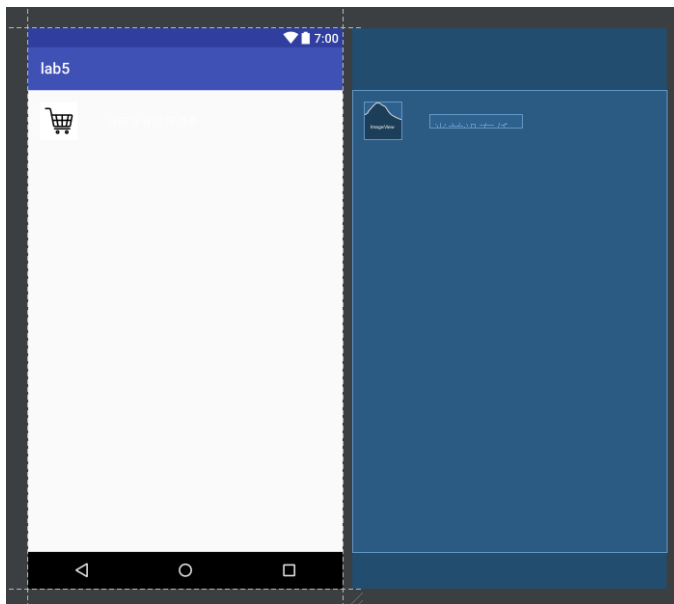
(二) 关键步骤:

1. 创建一个 widget。

- i. Android studio 已经提供了一个 widget 模板给我们使用，所以我们只需要先创建一个 widget 模板之后，在模板的基础上进行修改即可。



- ii. widget 界面修改如下，直接展示 XML 的效果



- iii. 修改 widget_info.xml 文件，设置 widget 的大小属性和布局

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:initialKeyguardLayout="@layout/new_app_widget"
    android:initialLayout="@layout/new_app_widget"
    android:minHeight="40dp"
    android:minWidth="250dp"
    android:previewImage="@mipmap/ic_launcher_round"
    android:resizeMode="horizontal|vertical"
    android:updatePeriodMillis="86400000"
    android:widgetCategory="home_screen|keyguard"></appwidget-provider>
```

此处设置最小宽度为 4 格，最小高度为 1 格。

2. 实现点击 widget 启动并推荐一个商品。

此处基于上次的实验，我们只需要实现通过点击初始 widget 启动，启动后基于上次实验，会发出一个推送商品的广播，所以此时我们只要接收广播然后更新 widget 信息。

- i. 实现 widget 启动应用，此时我们需要在 widget 被添加后第一次更新时，设置需要 widget 信息，和添加点击事件。此时我们只需要在 onUpdate 进行修改，并设置一个 flag 进行是否是第一次更新的判断。

```
@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){ //第一次更新
            Intent i = new Intent(context, MainActivity.class); //添加跳转事件
            PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, i, PendingIntent.FLAG_UPDATE_CURRENT);
            RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.new_app_widget);
            views.setTextViewText(R.id.appwidget_text, "当前没有任何消息"); //设置初始信息
            views.setImageViewResource(R.id.appwidget_image, R.mipmap.ic_launcher);
            views.setOnClickPendingIntent(R.id.appwidget, pendingIntent); //设置点击事件
            ComponentName me = new ComponentName(context, NewAppWidget.class);
            AppWidgetManager.getInstance(context).updateAppWidget(me, views); //更新
        } else
            updateAppWidget(context, appWidgetManager, appWidgetId);
    }
}
```

- ii. 应用开始后 widget 随机推荐一个商品，此时我们只需要接收应用运行后发出的随机推送商品的静态广播即可。

首先要现在 manifest 中给 widget 添加静态广播接收

```
<receiver android:name=".NewAppWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="cn.chonor.lab5.staticreceiver" />
    </intent-filter>
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/new_app_widget_info" />
</receiver>
```

然后我们在 widget 类中添加静态广播接收者，所以此时需要重写 onReceive 函数进行接收，接收后需要更新信息和设置点击事件。

```
@Override
public void onReceive(Context context, Intent intent) {
    final String action = intent.getAction();
    if(action.equals(STATICACTION)) { //判断是静态广播
        flag = false;
        Bundle extras = intent.getBundleExtra("mainActivity");
        int position = extras.getInt("position"); //获得数据位置
        Intent i = new Intent(context, Good_Info.class); //设置跳转跳
        Bundle bundle = new Bundle();
        bundle.putInt("position", position); //设置跳转时传输数据
        bundle.putParcelableArrayList("data", data.getGood_list());
    }
}
```

```

        bundle.putParcelableArrayList("cart", data.getCart_list());
        i.putExtra("mainActivity", bundle);
        RemoteViews views = new RemoteViews(context.getPackageName(),
R.layout.new_app_widget);//更新 widget
        PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, i,
PendingIntent.FLAG_UPDATE_CURRENT);
        views.setTextViewText(R.id.appwidget_text, data.getGood_list_index(postiton).getGoodName() + "
仅售" + data.getGood_list_index(postiton).getGoodPrice() + "元");
        views.setImageResource(R.id.appwidget_image, data.ID[data.map.get
(data.getGood_list_index(postiton).getGoodName())]);
        views.setOnClickPendingIntent(R.id.appwidget, pendingIntent);//设置点击事件
        ComponentName me = new ComponentName(context, NewAppWidget.class);
        AppWidgetManager.getInstance(context).updateAppWidget(me, views);//更新
    }
    super.onReceive(context, intent);
}

```

此处需要的开启应用后,widget 推荐商品已经完成。

3. 点击购物车更新 widget, 点击 widget 转跳到购物车界面。

- i. 此处 widget 内增加一个静态函数, 然后在商品信息界面接收到动态广播后调用此函数进行 widget 更新。

```

static void upDataDynamic(Context context, PendingIntent pendingIntent, String names) {
    Data data=new Data();
    RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.new_app_widget);
    views.setTextViewText(R.id.appwidget_text, names+"已添加到购物车");//更新信息
    views.setImageResource(R.id.appwidget_image, data.ID[data.map.get(names)]);
    views.setOnClickPendingIntent(R.id.appwidget, pendingIntent);//设置点击事件
    ComponentName me = new ComponentName(context, NewAppWidget.class);
    AppWidgetManager.getInstance(context).updateAppWidget(me, views);//更新
}

```

- ii. 之后在动态广播接收时调用这个函数进行更新

```

public class DynamicReceiver extends BroadcastReceiver {
    public static final String DYNAMICATION= "cn.chonor.lab5.dynamicreceiver";
    public DynamicReceiver() {}
    @Override
    public void onReceive(Context context, Intent intent) {
        if(intent.getAction().equals(DYNAMICATION)) {
            Intent intent1 =new Intent (context,Shoppingcart.class);//点击事件和传输
            PendingIntent pi = PendingIntent.getActivity(context, 0, intent1, PendingIntent.FLAG_CANCEL_CURRENT);
            NewAppWidget.upDataDynamic(context,pi,names);
        }
    }
}

```

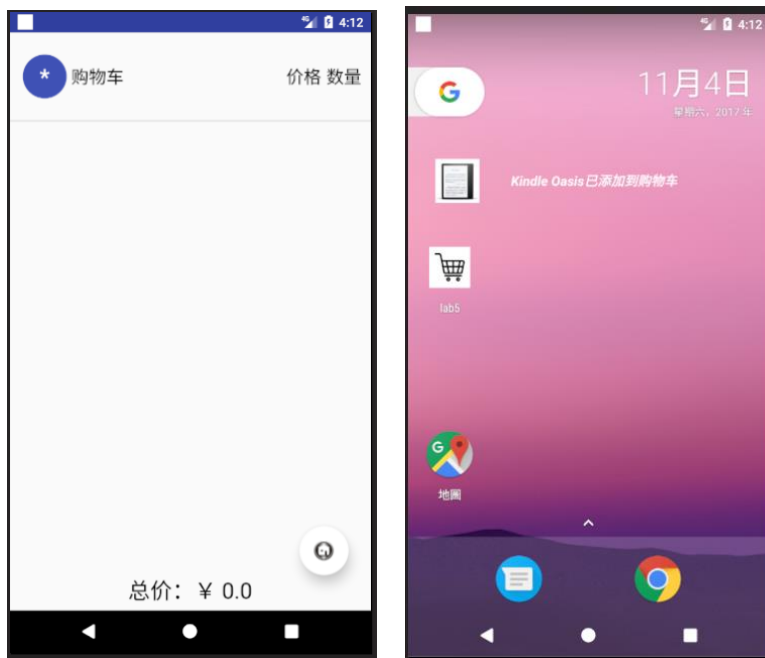
(三) 实验中遇到的困难和解决思路：

1. 动态广播接收的时候无法再 widget 内注册接收器，解决方法：在 widget 的 java 文件内写一个静态函数，在商品详情界面的时候进行接收广播的同时调用静态函数修改 widget。
2. 在静态函数修改 widget 的时候，点击事件无效，但是可以修改。之后发现是 context 问题，直接改传一个 PendingIntent 使得点击事件可以正常使用。

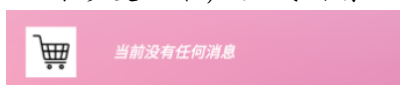
五、课后实验结果

本次实验只有 widget 的使用，无额外扩展。

修复了一下不合理的地方，当我们清空购物车列表的此时 widget 还是显示了推送信息，但是此时我们打开的是一个空的购物车。



此时修复一下，当我们清空购物车的时候还原 widget 为初始状态。



此时在更新里添加判断。

```
static void upDataDynamic(Context context, PendingIntent pendingIntent, String names) {
    Data data = new Data();
    RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.new_app_widget);
    if (names.equals("")) { // 购物车清空
        views.setTextViewText(R.id.appwidget_text, "当前没有任何消息"); // 设置初始信息
        views.setImageResource(R.id.appwidget_image, R.mipmap.ic_launcher);
    } else { // 动态广播更新
        views.setTextViewText(R.id.appwidget_text, names + "已添加到购物车"); // 更新信息
        views.setImageResource(R.id.appwidget_image, data.ID[data.map.get(names)]);
    }
    views.setOnClickPendingIntent(R.id.appwidget, pendingIntent); // 设置点击事件
    ComponentName me = new ComponentName(context, NewAppWidget.class);
```

```
AppWidgetManager.getInstance(context).updateAppWidget(me, views);//更新  
}
```

购物车被删除为空时调用更新

```
builder.setPositiveButton("确定", new DialogInterface.OnClickListener() {  
    @Override //设置确定按钮动作  
    public void onClick(DialogInterface dialogInterface, int i) {  
        data.removeCart_list_index(position);  
        myAdapter.notifyDataSetChanged();  
        Toast.makeText(getApplicationContext(), "成功删除 " + tmp, Toast.LENGTH_SHORT).show();  
        if(data.getCart_list().size()==0) {//购物车为空  
            Intent intent1 = new Intent(Shoppingcart.this, MainActivity.class);//点击事件和传输  
            PendingIntent pi = PendingIntent.getActivity(Shoppingcart.this, 0, intent1, PendingIntent.FLAG_CANCEL_CURRENT);  
            NewAppWidget.upDataDynamic(Shoppingcart.this, pi, "");  
        }  
    }  
}
```

六、实验思考及感想

本次实验是 widget 使用，主要就是涉及到 widget 的创建，更新，和接收广播。Widget 本身就是一个接收者，所有我们直接使用 onReceive 就可以接收了。

实验中因为 android studio 本身已经为我们内置了一个模板了，我们可以直接使用模板进行代码，而且其对于 widget 的配置也是可以通过我们输入几个参数就自动配置好的。剩下的就是我们写一下如何接收到之前实验中发出的广播然后进行更新。Widget 的用法比较局限，所以没想到什么可以进行拓展的地方。