

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	15352272	专业 (方向)	软件工程 (移动信息工程)
学号	15352272	姓名	彭国栋
电话	15626066058	Email	578291308@qq.com
开始日期		完成日期	

一、 实验题目

appwidget 及 broadcast 使用。

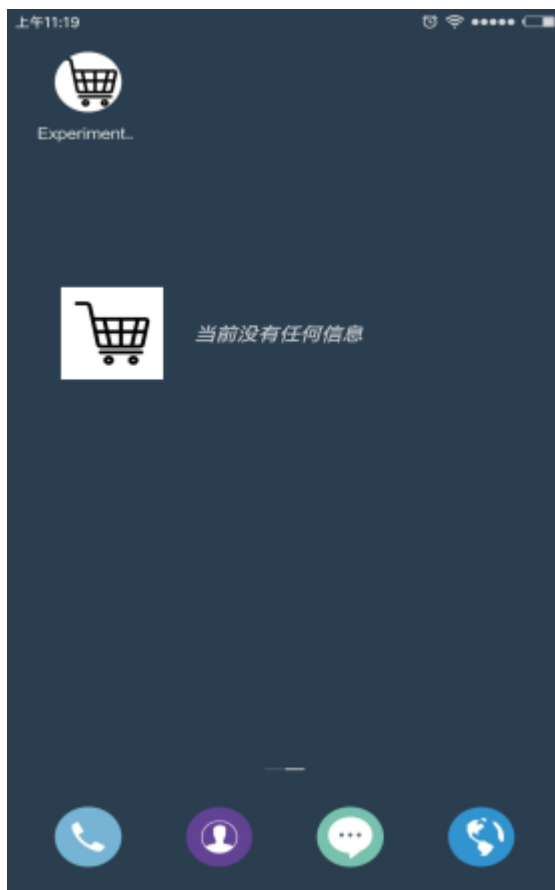
二、 实现内容

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：



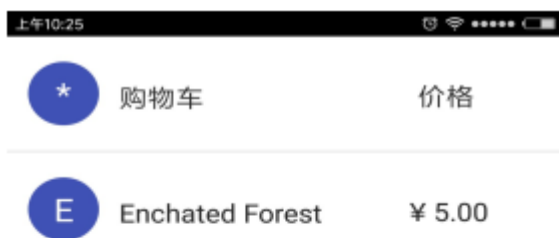
(3) 点击 widget 跳转到该商品详情界面：



(4) 点击购物车图标，widget 相应更新。



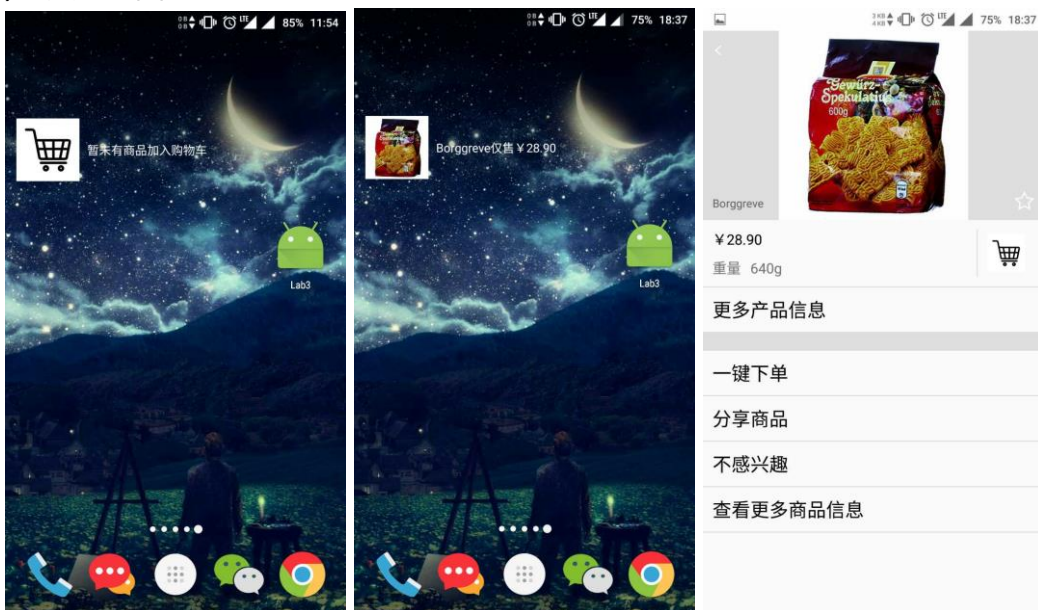
(5) 点击 widget 跳转到购物车界面：

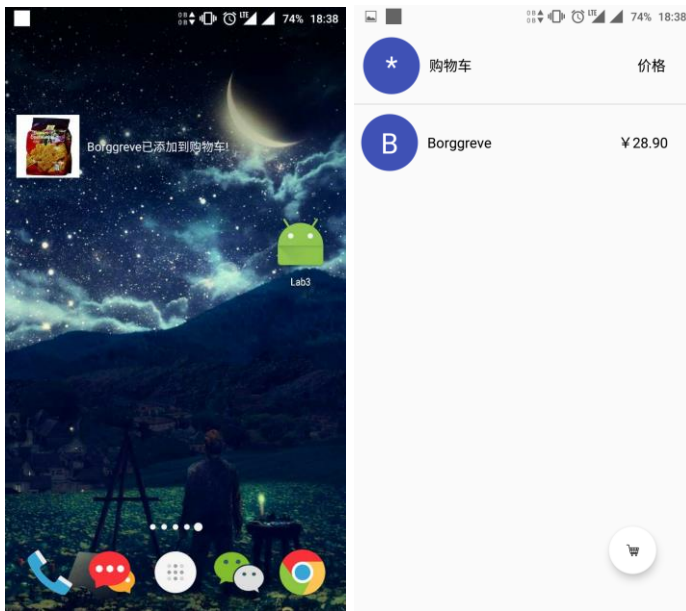


(6) 实现方式要求:启动时的 widget 的更新通过静态广播实现, 点击购物车图标时候 widget 的更新通过动态广播实现。

三、 课堂实验结果

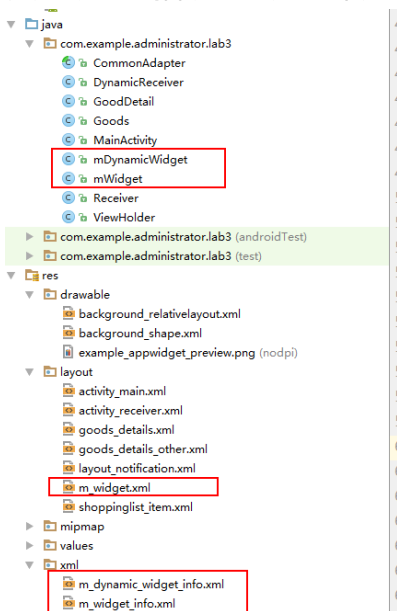
(1) 实验截图



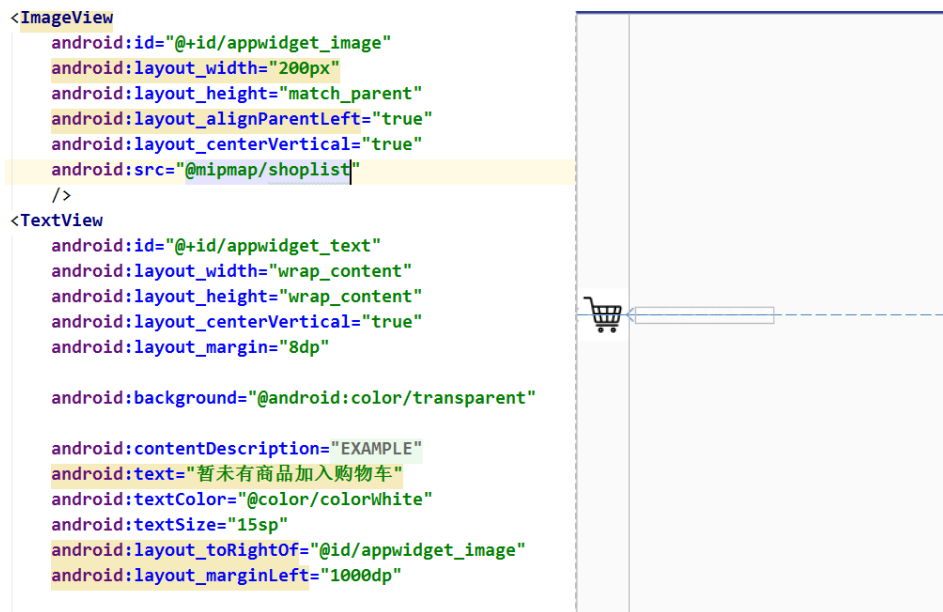


(2) 实验步骤以及关键代码

这次实验主要是利用新建一个 widget 来实现这项工作，由于需要使用静态广播和动态广播来实现所以和上次的代码类似，我新建了两个 widget 类，分别用于实现静态广播跳转到商品详情列表和实现动态广播跳转到购物车列表。



首先新建了 widget 文件之后，需要设置 widget 的布局，使用一个 ImageView 和一个 TextView 来实现简单布局：



然后再设置两个 widget 的 info.xml 文件中的布局文件



然后再设置 AndroidManifest.xml 中的 widget 里的动态广播名字：

```

<receiver android:name=".mWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="MystaticFliters" />
    </intent-filter>

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

```

接着就可以分别实现动态广播和静态广播的 widget 了。

静态广播实现的 widget，跳转到商品详情：

```

public void onReceive(Context context, Intent intent) {
    /*重写onReceive函数即可，实现方式类似于上次的通知实现*/
    super.onReceive(context, intent);
    Intent intentForStaticWidget=new Intent();
    temp=(Goods) intent.getExtras().get("random");
    /*由于功能上没有什么不同，所以可以借用上次通知的时候采用的静态广播来实现*/
    appWidgetManager=AppWidgetManager.getInstance(context);
    if(intent.getAction().equals(STATICACTION))
    {
        intentForStaticWidget.setClass(context,GoodDetail.class);//设置跳转
        intentForStaticWidget.putExtra("goods",temp);//设置传输的key和对应的商品
        mPendingIntent= PendingIntent.getActivity(context,0,intentForStaticWidget,PendingIntent.FLAG_UPDATE_CURRENT);
        remoteViews=new RemoteViews(context.getPackageName(),R.layout.m_widget);

        remoteViews.setImageViewResource(R.id.appwidget_image,temp.getImageid());
        remoteViews.setTextViewText(R.id.appwidget_text,temp.getName()+"仅售"+temp.getPrice());
        remoteViews.setOnClickPendingIntent(R.id.appwidget_relativeLayout,mPendingIntent);
        ComponentName componentName=new ComponentName(context,mWidget.class);
        appWidgetManager.updateAppWidget(componentName,remoteViews);/*更新widget*/
    }
}

```

动态广播会需要重新发送一个广播来实现：

在 MainActivity 里：

```

/*发送动态Widget的过滤器*/
mDynamicWidgetReceiver=new mDynamicWidget();
registerReceiver(mDynamicWidgetReceiver,intentFilter);//注册接收器

```

并在 onDestroy 函数中，注销：

```

protected void onDestroy()
{
    super.onDestroy();
    unregisterReceiver(receiver);
    unregisterReceiver(mDynamicWidgetReceiver);
    EventBus.getDefault().unregister(this);//注销订阅者
}

```

然后在 mDynamicWidget 中重写 onReceive 即可

```
public void onReceive(Context context, Intent intent) {
    super.onReceive(context, intent);
    Intent intentForDynamicWidget=new Intent();
    temp=(Goods) intent.getExtras().get("dynamic");
    appWidgetManager=AppWidgetManager.getInstance(context);
    if(intent.getAction().equals(DYNAMICACTION))
    {
        intentForDynamicWidget.setClass(context,MainActivity.class);//设置跳转界面
        mPendingIntent= PendingIntent.getActivity(context,0,intentForDynamicWidget,PendingIntent.FLAG_UPDATE_CURRENT);
        remoteViews=new RemoteViews(context.getPackageName(),R.layout.m_widget);
        remoteViews.setImageViewResource(R.id.appwidget_image,temp.getImageid());
        remoteViews.setTextViewText(R.id.appwidget_text,temp.getName()+"已添加到购物车!");
        remoteViews.setOnClickPendingIntent(R.id.appwidget_relativeLayout,mPendingIntent);
        ComponentName componentName=new ComponentName(context,mWidget.class);
        appWidgetManager.updateAppWidget(componentName,remoteViews);
    }
}
```

(3) 实验遇到困难以及解决思路

一开始在使用动态广播实现跳转到购物车列表的时候，以为可以和静态广播一样，不需要重新设置发送的广播，后面发现并没有反应，之后在仔细研究和推敲之后才意识到，还是需要在启动程序的时候注册接收器，只是可以使用同样的过滤方式。最后才解决无法跳转到购物车的问题。

四、 课后实验结果

这次实验并没有做特别的创新。

五、 实验思考及感想

这次实验和上次的实验很类似，感觉代码相似度都和上次的通知实验很像，只是这次感觉有点不足的是仍然采用了在 widget 中使用重写 onReceive 的方式实现，而感觉可以考虑通过设置 updateAppWidget 和 onUpdate 函数，然后再在 onReceive 上面调用 onUpdate 函数会更好。