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

姓名	学号	班级	电话	邮箱
张子豪	15352427	15M1	15989046143	ahzzh1998@163.com

1.实验题目

appwidget 及broadcast 使用

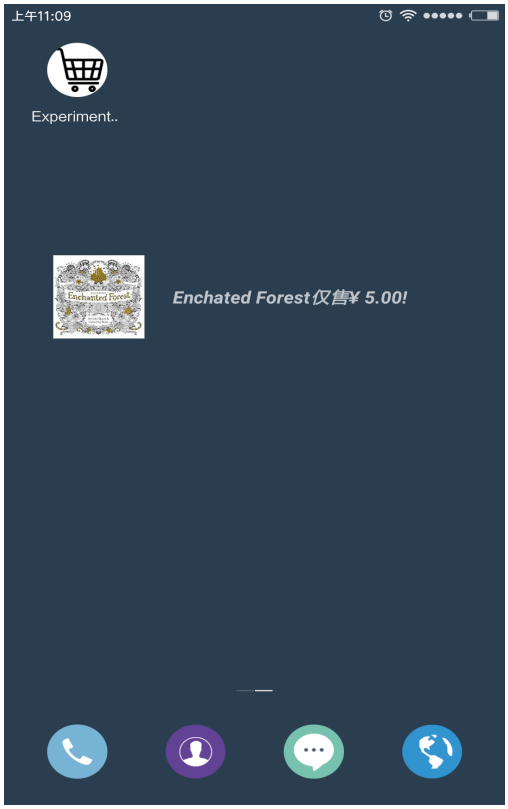
2.实现内容

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：

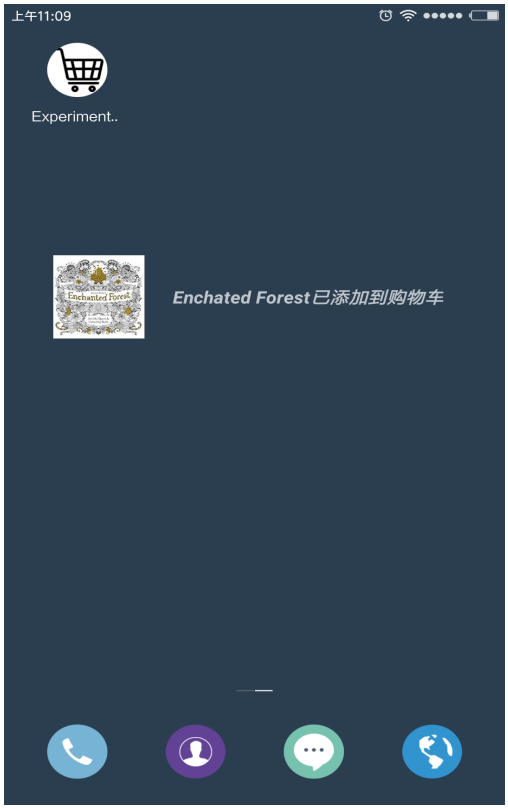
(1)widget 初始情况如下：

(2)点击widget可以启动应用，并在widget随机推荐一个商品:

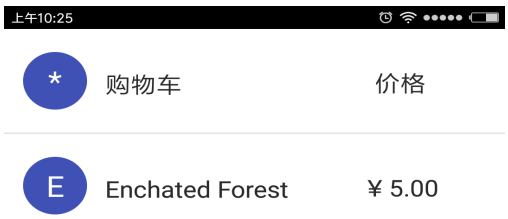


(3)点击widget跳转到该商品详情界面:

(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:



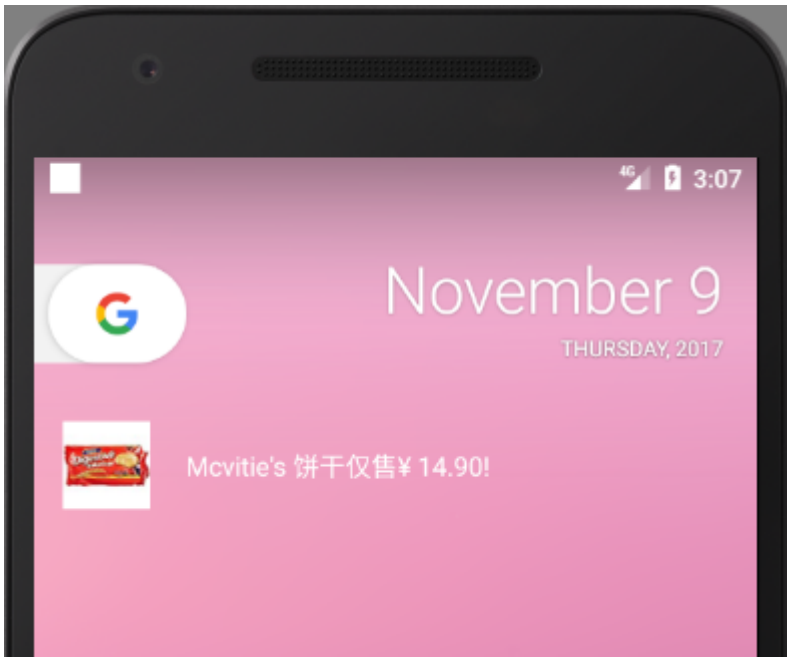
(6)实现方式要求:启动时的widget的更新通过静态广播实现，点击购物车图标时候widget的更新通过动态广播实现。

3.课堂实验结果

- 实验截图

开机后widget初始化如下：

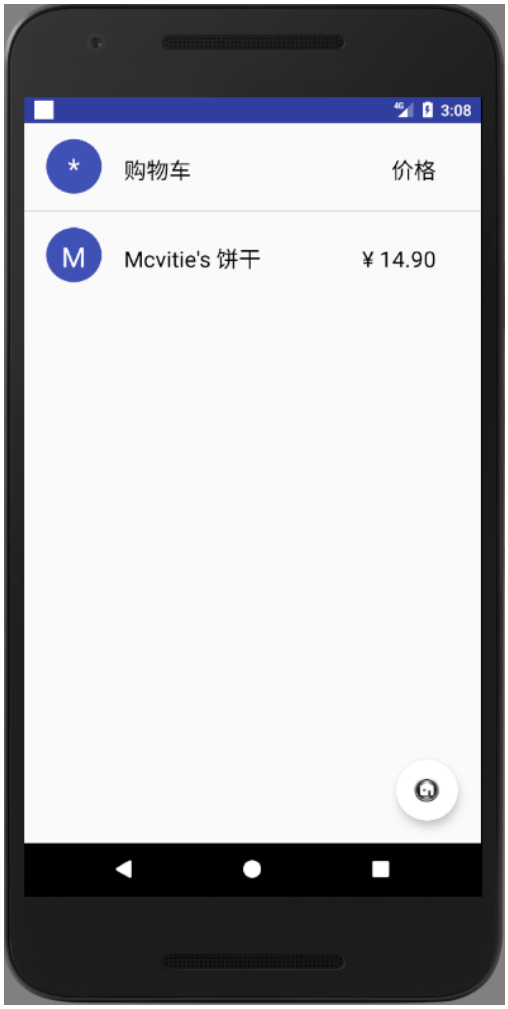
点击widget，进入app后再返回桌面，可以看到widget已经更新：



点击widget，进入商品详情：

添加到购物车后，返回桌面，可以看到widget已经更新：

点击widget进入购物车：



- 实验步骤以及关键代码

  在Android studio中新建一个widget类后，AS会自动补充好主要组件，典型的widget一般有三个组件——一个边框，一个框架和图形控件。在AndroidManifest中也会自动注册这个widget。

  1. widget的布局文件

     本次实验使用的widget只有一幅图片和一个文本，因此只需要一个ImageView和一个TextView

```
<ImageView
        android:id="@+id/appwidget_img"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:src="@mipmap/shoplist"/>
    <TextView
        android:id="@+id/appwidget_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="当前没有任何信息"
        android:textColor="#ffffff"
        android:textSize="15sp"
        android:layout_alignLeft="@+id/appwidget_img"
        android:layout_marginLeft="70dp"
        android:layout_alignTop="@+id/appwidget_img"
        android:layout_marginTop="15dp"/>
```

2. AppWidgetProviderInfo

AppWidgetProviderInfo定义了应用程序的小部件的基本属性,如最小尺寸布局,其最初的布局资源,多久更新应用程序的小部件,和(可选)配置活动启动创建时间等。

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:initialKeyguardLayout="@layout/m_widget"  //定义锁屏时应用widget的布局
    android:initialLayout="@layout/m_widget"  //定义了widget的XML布局
    android:minHeight="50dp"  //指定widget所占据的高
    android:minWidth="300dp"  //指定widget所占据的宽
    android:previewImage="@mipmap/shoplist"  //指定用户第一次看见预览界面的widget的图片
    android:resizeMode="horizontal|vertical"  //指定widget可以拖动水平/垂直布局网格改变大
小
    android:updatePeriodMillis="86400000"  //设置widget更新时间
    android:widgetCategory="home_screen|keyguard"  //指定widget是否可以显示在主屏幕和锁屏
上
></appwidget-provider>
```

3. widget没有任何信息时点击widget进入app

完成这个要求需要重写onUpdate方法，onUpdate在widget更新时触发，这里使用了一个flag变量来记录widget是否是第一次被启用。

```
@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){//第一次启用widget，设置为没有任何商品信息且点击进入app
            RemoteViews views=new
RemoteViews(context.getPackageName(),R.layout.m_widget);
            Intent intent=new Intent(context,MainActivity.class);
            PendingIntent
pendingIntent=PendingIntent.getActivity(context,0,intent,PendingIntent.FLAG_UPDATE_CURR
ENT);
            views.setOnClickPendingIntent(R.id.appwidget_text,pendingIntent);
            ComponentName me=new ComponentName(context,mWidget.class);
            appWidgetManager.updateAppWidget(me,views);
            flag=true;
        }
        else updateAppWidget(context, appWidgetManager, appWidgetId);
    }
}
```

4. 静态广播随机产生商品推送

发生广播：

```
Random random=new Random();
int rand=random.nextInt(name.length);
Bundle bundle=new Bundle();
bundle.putInt("key",rand);
Intent widget_broadcast=new Intent("widget_broadcast");//widget广播
widget_broadcast.putExtras(bundle);
sendBroadcast(widget_broadcast);
```

接收广播在widget类中重写onReceive方法：

```
@Override
public void onReceive(Context context,Intent intent){
    super.onReceive(context,intent);
    AppWidgetManager appWidgetManager=AppWidgetManager.getInstance(context);
    if(intent.getAction().equals("widget_broadcast")){
        Bundle tmp=intent.getExtras();
        int pos=tmp.getInt("key");//获取产生的随机数
        RemoteViews views=new RemoteViews(context.getPackageName(),R.layout.m_widget);
        views.setTextViewText(R.id.appwidget_text,name[pos]+"仅售"+price[pos]+"!");
        views.setImageViewResource(R.id.appwidget_img,img[pos]);
        Intent i=new Intent(context,ProductActivity.class);
        Bundle bundle=new Bundle();
        bundle.putInt("key",pos);
        i.putExtras(bundle);
        PendingIntent
pendingIntent=PendingIntent.getActivity(context,0,i,PendingIntent.FLAG_UPDATE_CURRENT);
        views.setOnClickPendingIntent(R.id.appwidget_text,pendingIntent);
        ComponentName me=new ComponentName(context,mWidget.class);
        appWidgetManager.updateAppWidget(me,views);
    }
}
```

5. 动态广播点击进入购物车

   由于上次实验中已经实现了动态广播，只需要在自定义的BroadcastReceiver类的onReceive方法中加上更新widget的代码就可以了，进入购物车后设置商品列表不可见也是上一次实验的内容，因此这里不再赘述：

```
RemoteViews views=new RemoteViews(context.getPackageName(),R.layout.m_widget);
views.setTextViewText(R.id.appwidget_text,name[pos]+"已添加到购物车");
views.setImageViewResource(R.id.appwidget_img,img[pos]);
Intent mainIntent=new Intent(context,MainActivity.class);
PendingIntent
pendingIntent=PendingIntent.getActivity(context,0,mainIntent,PendingIntent.FLAG_UPDATE_
CURRENT);
views.setOnClickPendingIntent(R.id.appwidget_text,pendingIntent);
ComponentName me=new ComponentName(context,mWidget.class);
appWidgetManager.updateAppWidget(me,views);
```

- 实验遇到困难以及解决思路

在写静态广播的时候，出现了一个很奇怪的事，我的onReceive函数会执行两次，也就是说我收到了两个静态广播。我在MainActivity中取消静态广播的发送后，依然能够收到一个静态广播。

解决方法是在mWidget的注册代码中再增加一个action，可能是因为编译器自动命名的action的问题，在receiver -> intent-filter中再增加一个action，然后发送广播的时候使用这个action的name，就解决了这个问题。

```
<receiver android:name=".mWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" /> //编译器自动命名的action
        <action android:name="widget_broadcast"/> //新增加的action
    </intent-filter>

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

4.实验思考及感想

经过这次实验我了解到了widget的使用，widget的view需要通过RemoteViews来转换表现，传入layout来获取一个RemoteView，然后通过修改这个RemoteView就能修改widget的显示，以及对应的点击响应事件。RemoteViews实际上是widget资源视图的集合工具管理者。当widget指定其具体的AppWidgetProvider后，AppWidgetProvider通过创建RemoteViews来加载视图，其RemoteViews将会调用setRemoteViewsAdapter来设置内部适配器，此适配器也将会继续获取widget管理器调用updateAppWidget()方法，此方法有会用远程视图工厂（RemoteViewsFactroy）来初始化数据并调用其onDataSetChanged（）来通知适配器更新数据。

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。