

# 中山大学数据科学与计算机学院本科生实验报告

## ( 2017 年秋季学期 )

课程名称：手机应用平台开发

任课教师：刘宁

年级	15 级软件工程	专业 ( 方向 )	计算机应用
学号	15331236	姓名	马朝露
电话	15521122675	Email	1073627941@qq.com
开始日期	2017/10/30	完成日期	2017/11/2

### 一、 实验题目

appwidget 及 broadcast 使用

### 二、 实现内容

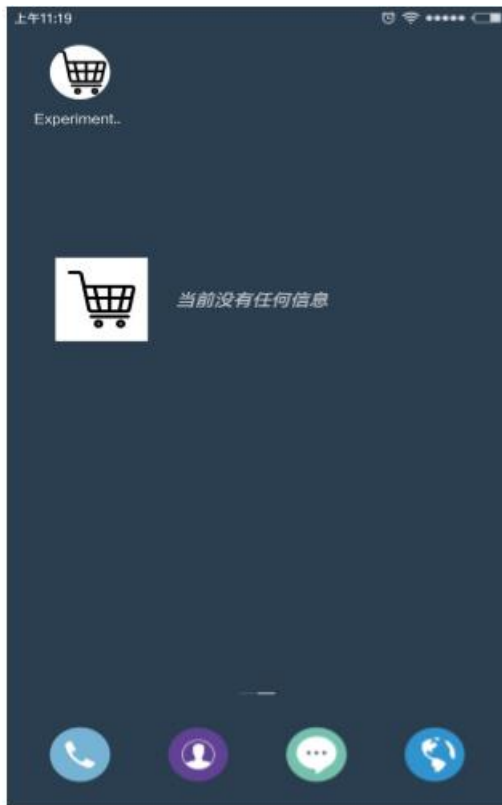
【实验目的】 1、掌握 AppWidget 编程基础

2、掌握 Broadcast 编程基础

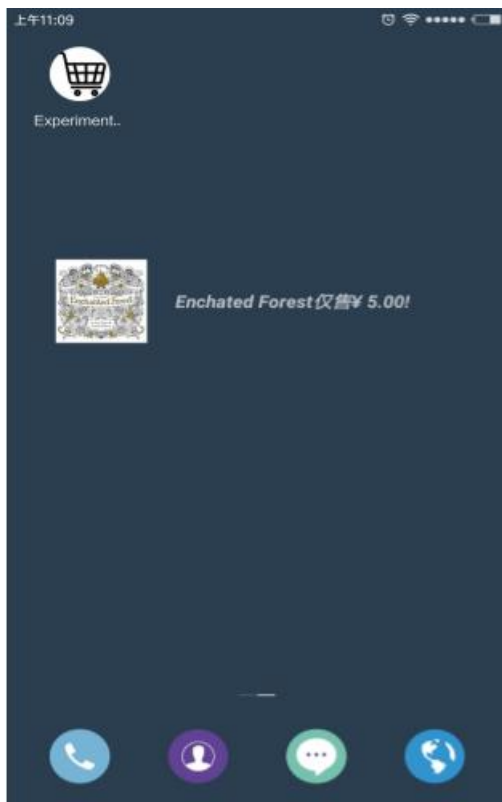
3、掌握动态注册 Broadcast 和静态注册 Broadcast

【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)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:



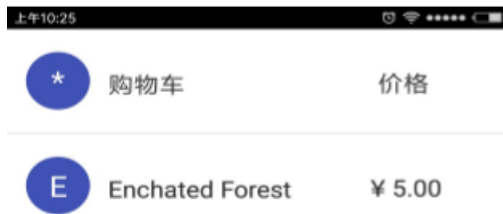
(3)点击 widget 跳转到该商品详情界面:



(4)点击购物车图标，widget 相应更新:



(4)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

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：



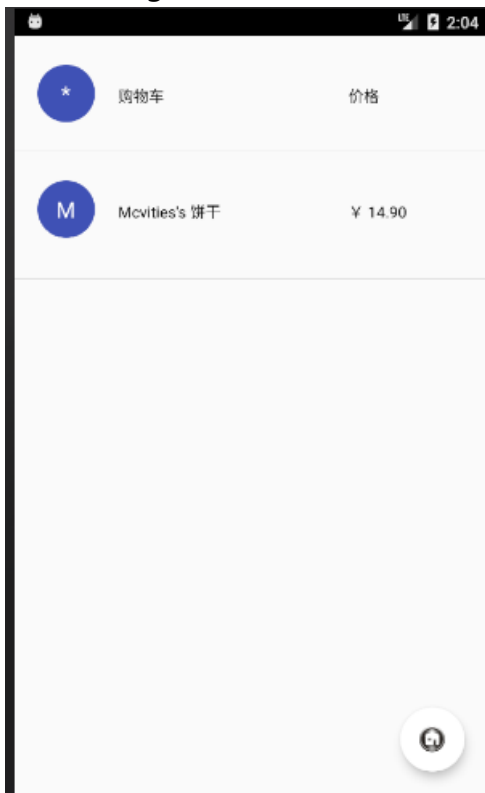
3.点击 widget 跳转到该商品详情界面：



4.点击购物车图表，widget 相应更新：



5.点击 widget 跳转到购物车界面：



(2) 实验步骤以及关键代码  
【widget】

```

private int shop_image,
static void updateAppWidget(Context context, AppWidgetManager appWidgetManager,
                             int appWidgetId) {

    CharSequence widgetText = "当前没有任何信息";
    // Construct the RemoteViews object
    RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.widget);
    views.setTextViewText(R.id.appwidget_text, widgetText);

    // Instruct the widget manager to update the widget
    appWidgetManager.updateAppWidget(appWidgetId, views);
}

@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) {
        Intent i = new Intent(context, lab3.class); // 首先, 在onUpdate()方法定义一个Intent, 用来开启Activity 的Intent
        // 第一个参数上下文, 第二个参数-要开启的Activity类

        // 顾名思义, 就是组件名称, 通过调用Intent中的setComponent方法, 我们可以打开另外一个应用中的Activity或者服务。
        ComponentName me = new ComponentName(context, Widget.class);
        // 用该Intent实例化一个PendingIntent
        PendingIntent pi = PendingIntent.getActivity(context, 0, i, PendingIntent.FLAG_UPDATE_CURRENT);
        // 实例化RemoteView, 其对应相应的Widget布局
        RemoteViews updateView = new RemoteViews(context.getPackageName(), R.layout.widget);
        updateView.setOnClickPendingIntent(R.id.widget_item, pi); // 设置点击事件
        appWidgetManager.updateAppWidget(me, updateView);
    }
}

@Override

```

widget内容的初始化在updateAppWidget中，每次都要用RemoteView来显示widget，每次改变widget内容后都需要updateAppWidget。onUpdate(Context, AppWidgetManager, int[]) 间隔调用此方法去更新 App Widget，间隔时间的设置是在AppWidgetProviderInfo中。，重写onUpdate方法，为Widget添加事件，使得点击能够启动应用。这里需要使用到一种用户程序访问主屏幕和修改特定区域内容的方法：RemoteView 架构 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。

## 【静态广播】

使用随机数:



```

Random random=new Random();
int random_int = random.nextInt(10);
random_name = name_[random_int];
if (random_name == "Enchated Forest") {
    random_image = R.mipmap.enchatedforest;
    random_price="¥ 5.00";
    random_birth="作者 Johanna Basford";
    random_initial="E";
} else if (random_name == "Arla Milk") {
    random_image=R.mipmap.arla;
    random_price="¥ 59.00";
    random_birth="产地 德国";
    random_initial="A";
} else if (random_name == "Devondale Milk") {
    random_image=R.mipmap.devondale;
    random_price="¥ 79.00";
    random_birth="产地 澳大利亚";
    random_initial="D";
} else if (random_name == "Kindle Oasis") {
    random_image=R.mipmap.kindle;
    random_price="¥ 2399.00";
    random_birth="版本 8GB";
    random_initial="K";
} else if (random_name == "waitrose 早餐麦片") {
    random_price="¥ 179.00";
    random_image=R.mipmap.waitrose;
    random_birth="重量 2Kg";
}

```

传递数据及发送广播：

```

Intent intentBroadcast = new Intent();
intentBroadcast.setAction("MystaticFliter");
Bundle bundlebroadcast = new Bundle();
bundlebroadcast.putString("random_name",random_name);
bundlebroadcast.putInt("random_image",random_image);
bundlebroadcast.putString("random_price",random_price);
bundlebroadcast.putString("random_birth",random_birth);
bundlebroadcast.putString("random_initial",random_initial);
intentBroadcast.putExtras(bundlebroadcast);
sendBroadcast(intentBroadcast);

```

接收器：

```

@Override
public void onReceive(Context context, Intent intent) {
    // TODO: This method is called when the BroadcastReceiver is receiving
    // an Intent broadcast.
    super.onReceive(context, intent);
    AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(context);
    if (intent.getAction().equals("MystaticFliter")) {

        Bundle bundle=intent.getExtras();
        random_name = bundle.getString("random_name");
        random_price = bundle.getString("random_price");
        random_image = bundle.getInt("random_image");
        random_birth = bundle.getString("random_birth");
        random_initial = bundle.getString("random_initial");
        //Toast.makeText(context, "You hit me!" +random_price+random_initial, Toast.LENGTH_LONG).show();
        RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.widget);
        views.setTextViewText(R.id.appwidget_text, random_name+"仅售"+random_price+"!");
        views.setImageViewResource(R.id.appwidget_imageview, random_image);
        ComponentName me = new ComponentName(context, Widget.class);
        appWidgetManager.updateAppWidget(me, views);
        Intent clickintent = new Intent(context, details.class);
        Bundle bundle_ = new Bundle();
        bundle_.putString("initial_1", random_initial);
        bundle_.putString("name2", random_name);
        bundle_.putString("birth", random_birth);
        bundle_.putString("price2", random_price);
        bundle_.putInt("image", random_image);
        clickintent.putExtras(bundle_);
        PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, clickintent, PendingIntent.FLAG_CANCEL_CURRENT);
        views.setOnClickPendingIntent(R.id.widget_item, pendingIntent);
        appWidgetManager.updateAppWidget(me, views);
    }
}

```

这部分我直接在widget类中实现了，首先接收来自主界面的intent，更新widget，然后设置点击事件，传递intent给详情界面。

过滤部分：

```

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

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

```

## 【动态广播】

注册及注销：

```

@Override
protected void onStart() {
    super.onStart();
    // 实例化IntentFilter对象
    IntentFilter filter = new IntentFilter();
    filter.addAction("DYNAMICATION");
    dreceiver = new dynamicReceiver();
    // 注册广播接收
    registerReceiver(dreceiver, filter);
}

@Override
protected void onStop() {

    unregisterReceiver(dreceiver);
    super.onStop();
}

```

传递数据及发送：

```

@Override
public void onClick(View v) {
    /*Map<String, Object> map = new HashMap<String, Object>();
    map.put("initial_1", s3);
    map.put("name_1", s1);
    map.put("price", s2);
    list.add(map);
    shopList.mDatas_ = list;
    s. this, "商品已加入购物车", Toast.LENGTH_SHORT). show(); */
    final class Bundle extends BaseBundle {
        Bundle() {
            intent();
            addAction("dynamic");
        }
        Bundle bundle_ = new Bundle();
        bundle_.putString("shop_name", s1);
        bundle_.putInt("shop_image", shop_image);
        intent.putExtras(bundle_);
        sendBroadcast(intent);
        EventBus.getDefault().postSticky(new MessageEvent(s3, s1, s2));
    }
}

Bundle bundle=this getIntent().getExtras();

```

接收器：

```

public class dynamicReceiver extends BroadcastReceiver {
    private String shop_name;
    private int shop_image;
    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
        // an Intent broadcast.
        AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(context);
        if (intent.getAction().equals("DYNAMICATION")) {
            Bundle bundle=intent.getExtras();
            shop_name = bundle.getString("shop_name");
            shop_image = bundle.getInt("shop_image");
            RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.widget);
            views.setTextViewText(R.id.appwidget_text, shop_name+"已添加到购物车");
            views.setImageViewResource(R.id.appwidget_imageview,shop_image);
            ComponentName me = new ComponentName(context,Widget.class);
            appWidgetManager.updateAppWidget(me, views);
            Intent clickintent = new Intent(context,shopList.class);
            PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, clickintent, 0);
            views.setOnClickPendingIntent(R.id.widget_item,pendingIntent);
            appWidgetManager.updateAppWidget(me, views);
            //Toast.makeText(context, "You hit me!"+"shop_name, Toast.LENGTH_LONG). show();
            /*Intent clickintent = new Intent(context,shopList.class);
            PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, clickintent, 0);
            NotificationManager notifyManager = (NotificationManager) context.getSystemService(Context.NOTIFICATION);
            Notification.Builder builder = new Notification.Builder(context);
            builder.setContentTitle("马上下单").
                setLargeIcon(BitmapFactory.decodeResource(context.getResources(),shop_image)).
                setSmallIcon(shop_image).setContentText(shop_name+"已添加到购物车").
                setContentIntent(pendingIntent).setAutoCancel(true);
            Notification notify = builder.build();
            notifyManager.notify(0, notify);*/
        }
    }
}

```

### (3) 实验遇到困难以及解决思路

#### 1. 不能添加 widget

解决方法：

Widget 设置的宽度太宽。改为课件中的 300dp 即可

#### 2. 虽然接受到了静态广播，但是 widget 没有改变，

解决方法：每次改变 widget 的内容都要调用 appWidgetManager.updateAppWidget()函数

## 四、实验思考及感想

这次的作业和上次的代码大体差不多，但是却是不同的内容。学会了安卓小部件 widget。有进一步的理解了广播事件。同时印象深刻的是 RemoteViews。RemoteViews 并不是一个真正的 View，它没有实现 View 的接口，而只是一个用于描述 View 的实体。比如：创建 View 需要的资源 ID 和各个控件的事件响应方法。RemoteViews 会通过进程间通信机制传递给 AppWidgetHost。

AppWidgetHostView 是真正的 View，但它只是一个容器，用来容纳实际的 AppWidget 的 View。这个 AppWidget 的 View 是根据 RemoteViews 的描述来创建。

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。