

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	专业 (方向)	计应
学号	15331421	姓名	郑佳锋
电话	13719325472	Email	1773643139@qq.com
开始日期	11.3	完成日期	11.4

一、 实验题目

Lab7

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:



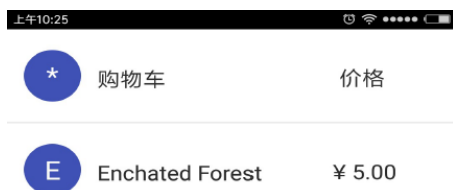
(3) 点击widget跳转到该商品详情界面:



(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。

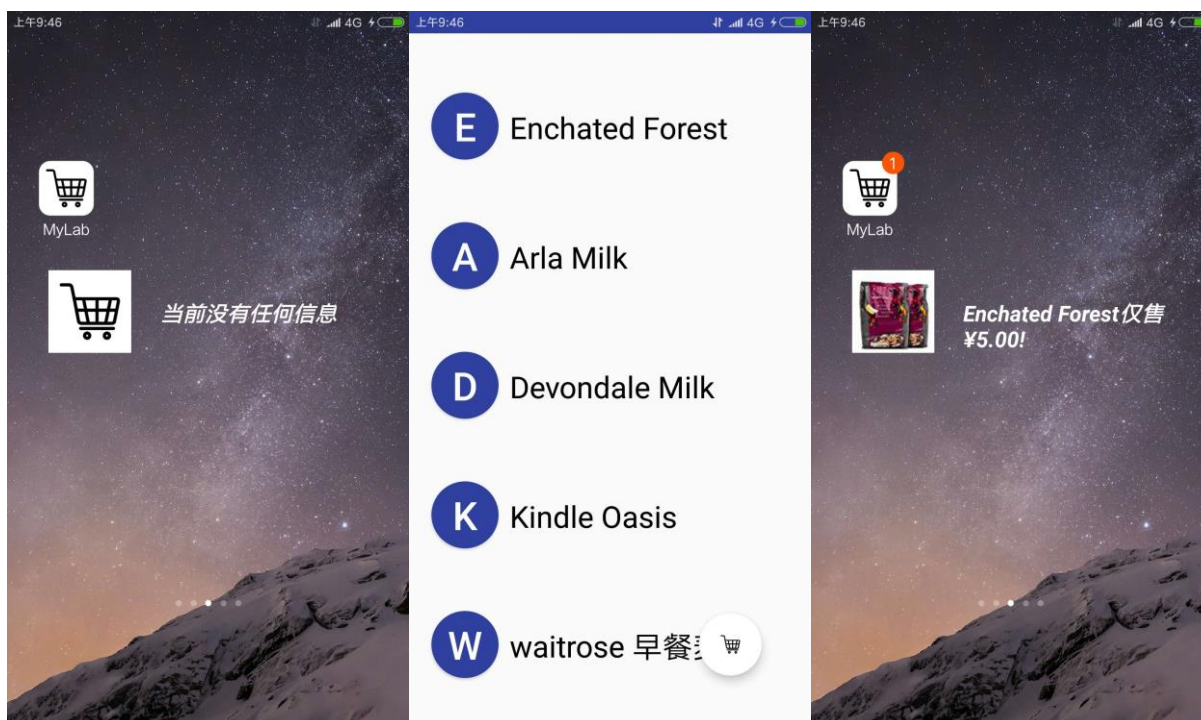
三、 实验结果

(1) 实验截图

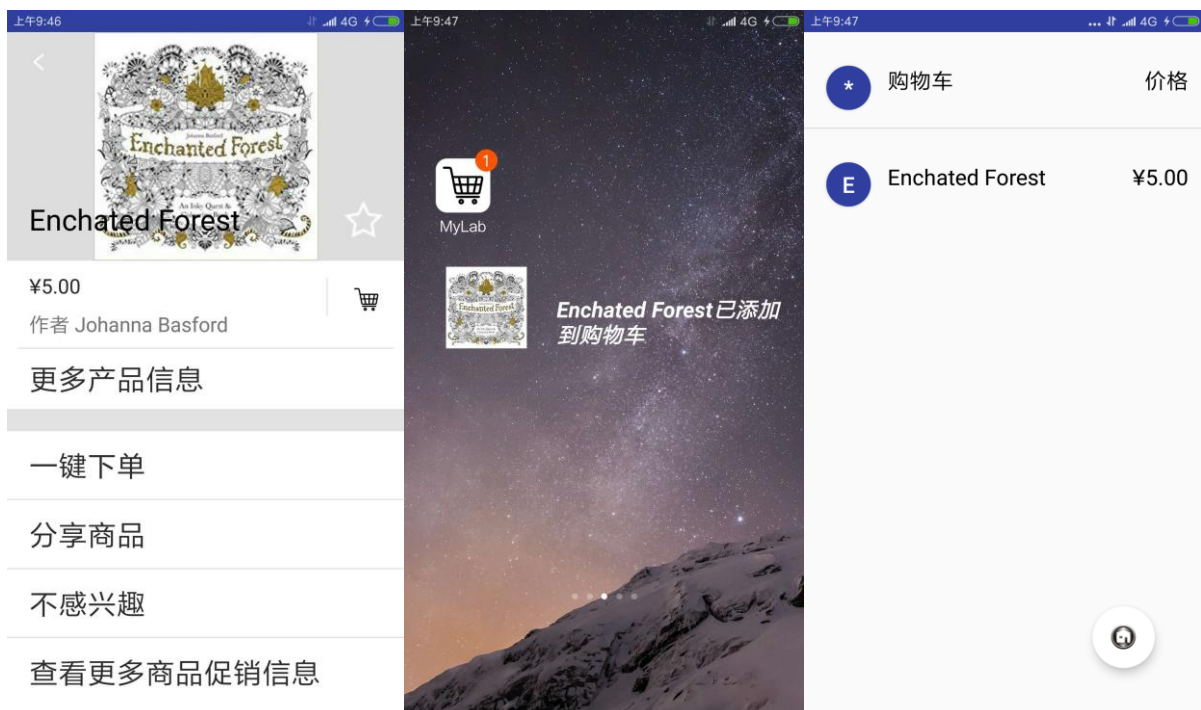
添加 widget :

点击 widget 启动应用 :

随机推荐一个商品 :



点击 widget 跳到详情界面 : 点击购物车更新 widget : 点击 widget 跳到购物车



(2) 实验步骤以及关键代码

1.创建 widget , 同时生成以下四个文件 :

example_appwidget_preview.png : widget 默认图片 , 可以在 m_widget_info.xml 文件改成其他。

m_widget_info.xml : 设置 widget 默认图片 , 最小宽度和高度 , 更新时间等等。

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:initialKeyguardLayout="@layout/m_widget"
4.     android:initialLayout="@layout/m_widget"
5.     android:minHeight="50dp"
6.     android:minWidth="300dp"
7.     android:previewImage="@mipmap/shoplist"
8.     android:resizeMode="horizontal|vertical"
9.     android:updatePeriodMillis="86400000"
10.    android:widgetCategory="home_screen|keyguard"></appwidget-provider>
```

m_widget.xml : 设置 widget 的布局 , 本实验中 widget 共有两个组件 , 一个 ImageView , 一个 TextView。

```
1. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2.     android:layout_width="match_parent"
3.     android:layout_height="match_parent"
4.     android:id="@+id/all_widget"
5.     android:background="#0000"
6.     android:padding="@dimen/widget_margin">
7.     <ImageView
8.         android:id="@+id/appwidget_image"
9.         android:layout_width="80dp"
10.        android:layout_height="80dp"
11.        android:src="@mipmap/shoplist"
12.        android:layout_marginLeft="12dp"
13.    />
14.     <TextView
15.         android:id="@+id/appwidget_text"
16.         android:layout_width="wrap_content"
17.         android:layout_height="wrap_content"
18.         android:background="#0000"
19.         android:contentDescription="@string/appwidget_text"
20.         android:text="@string/appwidget_text"
21.         android:textColor="#fff"
22.         android:textSize="15sp"
23.         android:textStyle="bold|italic"
24.         android:layout_marginLeft="115dp"
25.         android:layout_marginTop="25dp"
26.    />
27.
28. </RelativeLayout>
```

m_widget.java:重写 onUpdate 方法 , onUpdate 方法调用 updateAppWidget 方法 , 对创建的每一个 widget 进行更新。当 widget 未被点击时 , 图片为购物车 , 文字为 “当前没有任何信息” ; 点击后 , 利用 PendingIntent 启动应用 , 跳转到主页面。

```

1. static void updateAppWidget(Context context, AppWidgetManager appWidgetManager,
2.                             int appWidgetId) {
3.
4.     CharSequence widgetText = context.getString(R.string.appwidget_text);
5.     // Construct the RemoteViews object
6.     RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.m_widget);
7.     views.setTextViewText(R.id.appwidget_text, widgetText);
8.     Intent intent = new Intent(context, MainActivity.class);
9.     intent.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, appWidgetId); // Identifies th
    e particular widget...
10.    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
11.    PendingIntent pendIntent = PendingIntent.getActivity(context, 0, intent, PendingInte
    nt.FLAG_UPDATE_CURRENT);
12.    //RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.m_widget);
13.    views.setOnClickPendingIntent(R.id.all_widget, pendIntent);
14.    //appWidgetManager.updateAppWidget(appId, views);
15.    // Instruct the widget manager to update the widget
16.    appWidgetManager.updateAppWidget(appWidgetId, views);
17. }
18. @Override
19. public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidget
    Ids) {
20.     final int N = appWidgetIds.length;
21.     for (int i = 0; i < N; i++) {
22.         int appWidgetId = appWidgetIds[i];
23.         updateAppWidget(context, appWidgetManager, appWidgetId);
24.     }
25. }

```

2. 静态广播：

当启动应用后，widget 会随机推荐一件商品，所以现在 MainActivity 中发送广播。

```

1. Random randomForWidget = new Random();
2. int ranNumForWidget = randomForWidget.nextInt(mDatas.size());
3. Intent intentForWidget = new Intent();
4. intentForWidget.setAction("mywidget");
5. intentForWidget.putExtra("name", mDatas.get(ranNum));
6. intentForWidget.putExtra("price", name_price.get(mDatas.get(ranNum)));
7. intentForWidget.putExtra("picture", name_pic.get(mDatas.get(ranNumForWidget)));
8. sendBroadcast(intentForWidget);

```

广播中 setAction 为 “mywidget”，所以需要在 AndroidManifest.xml 中将 m_widget 的广播接收过滤器设置为 “mywidget”。

```

1. <receiver android:name=".mWidget">
2.     <intent-filter>
3.         <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
4.         <action android:name="mywidget" />
5.     </intent-filter>
6.     <meta-data
7.         android:name="android.appwidget.provider"
8.         android:resource="@xml/m_widget_info" />
9. </receiver>

```

当 m_widget 类接收到广播后，重写 onReceive 方法，在该方法中将接收到的 intent 中的名称和图片设置成 widget 的文字和图片，并修改 widget 的点击事件，使之被点击后传递 intent 到详情界面 (NewAcitvity) 并打开它。

```

1.  super.onReceive(context, intent);
2.  AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(context);
3.  Bundle bundle = intent.getExtras();
4.  if(intent.getAction().equals("mywidget"))
5.  {
6.      RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.m_widget);
7.      views.setTextViewText(R.id.appwidget_text, intent.getStringExtra("name") + "仅售"
8.      "+intent.getStringExtra("price") + "!");
9.      views.setImageViewResource(R.id.appwidget_image,Integer.parseInt(intent.getStringExtra("picture")));
10.      Intent newIntent = new Intent(context, NewActivity.class);
11.      newIntent.putExtra("passname",intent.getStringExtra("name"));
12.      newIntent.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, new ComponentName(context, m
13.      Widget.class)); // Identifies the particular widget...
14.      newIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
15.      PendingIntent pendIntent = PendingIntent.getActivity(context, 0, newIntent, PendingI
16.      ntent.FLAG_UPDATE_CURRENT);
17.      //RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.m_widget);
18.      views.setOnClickPendingIntent(R.id.all_widget, pendIntent);
19.      appWidgetManager.updateAppWidget(new ComponentName(context, mWidget.class), views);
20.  }

```

3. 动态广播：

在详情界面 (NewActivity) 的购物车点击事件中发送广播，广播的名称设置为“widgetshop”：

```

1.  shopbus.setOnClickListener(new View.OnClickListener() {
2.      @Override
3.      public void onClick(View v) {
4.          Toast.makeText(NewActivity.this, "商品已加入购物车", Toast.LENGTH_SHORT).show();
5.          EventBus.getDefault().post(
6.              new FirstEvent(passname, name_price.get(passname)));
7.          Intent intent = new Intent();
8.          intent.setAction("shop");
9.          intent.putExtra("name",passname);
10.         intent.putExtra("picture", name_pic.get(passname));
11.         sendBroadcast(intent);
12.         Intent widgetIntent = new Intent();
13.         widgetIntent.setAction("widgetShop");
14.         widgetIntent.putExtra("name",passname);
15.         widgetIntent.putExtra("picture", name_pic.get(passname));
16.         sendBroadcast(widgetIntent);
17.     }
18. });

```

接着在 NewActivity 的 onCreate 方法中注册广播：

```

1.  IntentFilter filter1 = new IntentFilter();
2.  filter1.addAction("widgetShop");
3.  registerReceiver(widgetReceiver,filter1);

```

在 onDestroy 方法中解除注册：

```

1.  protected void onDestroy(){
2.      super.onDestroy();
3.      unregisterReceiver(widgetReceiver);

```



```
4. }
```

在 `m_widget` 类中继续完善 `onReceive` 方法，实质接收到 Action 为 “widgetshop” 的广播时，将 widget 的文本和图片修改为添加到购物车商品的文字和图片，同时修改点击事件，点击后跳转到购物车页面 (`MainActivity`)。

```
1. if(intent.getAction().equals("widgetShop")){
2.     RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.m_widget);
3.     views.setTextViewText(R.id.appwidget_text, intent.getStringExtra("name") + "已添加到购物车");
4.     views.setImageViewResource(R.id.appwidget_image,Integer.parseInt(intent.getStringExtra("picture")));
5.     Intent newIntent = new Intent(context, MainActivity.class);
6.     newIntent.putExtra("widget","widget");
7.     newIntent.putExtra("passname",intent.getStringExtra("name"));
8.     newIntent.setAction(Intent.ACTION_MAIN);
9.     newIntent.addCategory(Intent.CATEGORY_LAUNCHER);
10.    newIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK| Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
11.    newIntent.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, new ComponentName(context, mWidget.class)); // Identifies the particular widget...
12.    PendingIntent pendIntent = PendingIntent.getActivity(context, 0, newIntent, PendingIntent.FLAG_UPDATE_CURRENT);
13.    //RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.m_widget);
14.    views.setOnClickPendingIntent(R.id.all_widget, pendIntent);
15.    appWidgetManager.updateAppWidget(new ComponentName(context, mWidget.class), views);
16. }
```

在 `MainActivity` 中接收 intent，将商品列表设置为不可见，购物车列表设置为可见。

```
1. Intent intent2 = getIntent();
2. Bundle bundle = intent2.getExtras();
3. if(bundle!=null){
4.     if (bundle.getString("widget") != null) {
5.         myRecView.setVisibility(View.INVISIBLE);
6.         shopList.setVisibility(View.VISIBLE);
7.         floatBtn.setImageResource(R.mipmap.mainpage);
8.         tag = 1;
9.         Map<String, String> temp = new LinkedHashMap<String, String>();
10.        temp.put("goods", intent2.getStringExtra("passname"));
11.        temp.put("price", name_price.get(intent2.getStringExtra("passname")));
12.        temp.put("shopButton", intent2.getStringExtra("passname").substring(0,1));
13.        data.add(temp);
14.        simpleAdapter.notifyDataSetChanged();
15.    }
16. }
```

(3) 实验遇到困难以及解决思路

本次实验困扰我的点主要在于 `Activity` 的生存周期。比如，点击 widget 打开商品详情页面，点击购物车图标后，可以跳转到购物车页面，但是购物车却空空。而打开应用后，跳转到详情界面，点击购物车图标，就可以正常添加到购物车。经过分析，才发现之前都是习以为常的在跳转到哪个页面就在哪个页面注册广播，所以这次点击 widget 打开商品详情页面后 `MainActivity` 并

未被激活，所以发送的广播就无效。于是，我就把 **MainActivity** 中的注册和反注册广播搬到 **NewActivity** 中，这样就能正常接收广播了。类似的问题在这两次实验中都有遇到，所以我就去了解了一下 **Activity** 生命周期的相关知识。在每次发送广播或者使用 **eventBus** 的时候，要考虑 **Activity** 是否已经存在栈中，若是，是新创建实例还是把它移到栈顶等等……这些都是我们在编程使需要考虑的问题。

四、 实验思考及感想

本次实验的主要内容是 **widget** 的使用。在平时使用 **android** 手机的过程中，我很少使用到这个组件，甚至不知道这个组件就叫 **widget**。学习了这次课程后，才发现手机的大部分 **app** 都有 **widget** 这个功能，有些还挺实用的。

Widget 实现起来不是很难。创建一个类后该有的组件基本都自动创建了，只需要在这些组件里设计自己想要的组件即可。通过这次实验也更加明白了 **Activity** 之间的信息传递。