

## 第五讲      导航的设计

### 使用说明

【源代码根目录】： 工程目录 NavigationUi

【记号】:

(@编程练习): 表明该实验是需要编将工程文件和实验报告一起提交。

(@团队编程练习): 表明该实验是以小组为单位完成的, 每个小组完成一份程序和报告即可, 报告和工程和其他的实验报告和工程最后要分开交。

本次实验的个人编程部分, 请建立在名为 KH5 的 project 中。团队编程练习建立 KHTD5 的工程。

上传方式: <http://disk.lehu.shu.edu.cn/index.aspx>

上传码:

android-sybk (本科)

android-syyjs (研究生)

【实验报告的要求】

- 1、文件名规范: 学号+姓名+实验名称.doc
- 2、内容格式见 实验报告格式.doc

## 目录

第五讲	导航的设计 .....	1
【课堂实验 KT5.1】	:Action Bar .....	3
【课堂实验 KT5.2】	:ViewPager 的用法.....	8
【课堂实验 KT5.3】	:为 ActionBar 增加 Tab .....	9
【课堂实验 KT5.4】	:Tab+Swipe 的实现 .....	12
【课堂实验 KT5.5】	:NavigationDrawer 的用法.....	12
【课堂实验 KT5.5】	:用 ToolBar 替代 ActionBar.....	15
【课后练习 KH5.1】	:app 需求分析 .....	18

## 【课堂实验 KT5.1】:Action Bar

【工程模块】: ActionBarSample

【工程描述】: 该程序展示 ActionBar 的基本用法。包括如何加入 ActionBar，获取对象、隐藏和显示；增加 ACTION button 和事件；设置 logo，title；设置向上返回的导航等。

【目标要求】:

- 掌握加入 ActionBar 的基本方法
- 掌握如下基本操作
  - a) 获取对象，显示隐藏 actionbar
  - b) 设置 logo,title,subtitle
  - c) 加入按钮和事件
  - d) 加入回溯按钮（UP）

【实验步骤】:

- 1、file→new→new module 创建一个新 module，选择 empty activity。
- 2、观察 androidmanifest.xml。注意斜体字的部分，theme 属性的设置，默认将在 app 中引入一个 ActionBar。其默认效果是显示@string/app\_name。

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.me.actionbarsample">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
```

- 3、在 activity\_main 中加入一个 button，并设置事件，来控制 actionbar 的显示和隐藏。代码如下：

```

public class MainActivity extends AppCompatActivity {
    ActionBar mAb;
    Button mBt;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mAb=getSupportActionBar(); //获取系统 ActionBar 对象

        mAb.setTitle("ActionBar 例子");
        mAb.setSubtitle("MainActivity");

        mAb.setDisplayHomeAsUpEnabled(true);
        mAb.setDisplayUseLogoEnabled(true);
        mAb.setLogo(R.drawable.ic_account_balance_black_18dp);

        mBt= (Button) findViewById(R.id.btshowbar);
        mBt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(mAb.isShowing()) {
                    mAb.hide();
                    mBt.setText("show bar");
                } else {
                    mAb.show();
                    mBt.setText("hide bar");
                }
            }
        });
    }
}

```

需注意的几点：

- `mAb=getSupportActionBar();` //获取系统 *ActionBar* 对象

来获取 *actionbar* 对象，由于要向下兼容（本例 `activity extends` 自 `AppCompatActivity` 而不是 `Activity`）因此用 `getSupportActionBar()` 而不是 `getActionBar()`；

- `Import` 的包也是 `import android.support.v7.app.ActionBar;` 而不是 `android.app.ActionBar;`
- 若要显示 `logo`，应将这三句话都写上：

```
mAb. setDisplayHomeAsUpEnabled(true);  
mAb. setDisplayUseLogoEnabled(true);  
mAb. setLogo(R.drawable.ic_account_balance_black_18dp);
```

4、运行并观察现象。

5、下面我们来创建另一个 activity 然后在该 activity 的 actionBar 上加入一个 up 图标，点击该图标则返回到主 activity。

6、创建 activity，命名为 SecondActivity，然后在 MainActivity 中加入一个按钮，点击按钮切换到 SecondActivity。

```
mBtSe = (Button) findViewById(R.id.btsecond);  
mBtSe.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent it=new Intent(MainActivity.this, SecondActivity.class);  
        startActivity(it);  
    }  
});
```

7、打开 androidmanifest.xml，在<SecondActivity>添加属性和元素如下：

```
<activity android:name=".SecondActivity"  
    android:parentActivityName=".MainActivity" >  
    <!-- Parent activity meta-data to support 4.0 and lower -->  
    <meta-data  
        android:name="android.support.PARENT_ACTIVITY"  
        android:value=".MainActivity" />  
</activity>
```

```
android:parentActivityName=".MainActivity"
```

标明 up 按钮要返回的上一级 activity。为了兼容低版本设备，应加上一个<meta-data>元素如上代码。

8、编辑 SecondActivity 的代码，如下，让它的 actionBar 出现一个 up 按钮。这个按钮默认是一个指向左边的箭头。

```

public class SecondActivity extends AppCompatActivity {
    ActionBar mAb;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        mAb=getSupportActionBar(); //获取系统 ActionBar 对象
        mAb.setTitle("ActionBar 例子");
        mAb.setSubtitle("SecondActivity");

        mAb.setDisplayHomeAsUpEnabled(true);
        mAb.setDisplayUseLogoEnabled(true);
        mAb.setLogo(R.drawable.ic_account_balance_black_18dp);

        mAb.setDisplayHomeAsUpEnabled(true);
    }
}

```

9、运行程序，观察现象

10、下面，我们来为 actionbar 加上事件按钮也就是所谓的 action。首先在 res/下建一个 menu 目录。右击 res->new→Directory。

11、右击 menu，new→new menu resource file，建立一个 menu 资源文件，比如：menu\_main。则在 menu 目录下创建了一个 menu\_main.xml 文件，编辑文件如下：

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/action1"
        android:icon="@drawable/ic_accessibility_white_24dp"
        android:title="@string/app_name"
        app:showAsAction="ifRoom"/>
    <item
        android:id="@+id/actionnever"
        android:icon="@mipmap/ic_launcher"
        android:title="@string/app_name"
        app:showAsAction="never"/>
</menu>

```

在菜单资源文件中，我们加入了两个 item，这两个 item 就是两个菜单项。注意属性 app:showAsAction 的值（ifRoom 表示，只要 actionbar 有空间，该按钮就显示，否则就隐藏在最右边的 overflow 中，never 则表示无论有没有空间都隐藏在 overflow 中）。

12、接着把该菜单和 ActionBar 绑定，方法就是实现回调函数

onCreateOptionsMenu(Menu menu)和 onOptionsItemSelected(MenuItem item)

编辑 MainActivity.java 加入如下代码:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {
        case R.id.action1:
            Toast.makeText(getApplicationContext().getApplicationContext(), "action1",
                Toast.LENGTH_SHORT).show();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {

    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
```

onCreateOptionsMenu(Menu menu)回调函数用于将 menu 资源文件绑定到 actionBar，onOptionsItemSelected 函数则实现菜单项的选中事件。

13、运行程序，观察结果。

## 【课堂实验 KT5.2】:ViewPager 的用法

【工程模块】: SwipeSample

【工程描述】: 该程序展示了使用 ViewPager 类实现 swipe 效果的基本方法。

【目标要求】:

理解 adapter 设计模式的方法论

掌握 viewpager 的基本使用方法

【实验步骤】:

- 1、运行程序，观察现象。
- 2、理解分析代码

【知识点注释】

- 1、viewpager 派生自 viewgroup，因此其行为可以参考 viewgroup
- 2、该类还在开发中，因此后期的用法可能会调整
- 3、和 fragment 的绑定有两种方式

[FragmentPagerAdapter](#)

Fragment 个数固定.

[FragmentStatePagerAdapter](#)

Fragment 个数不能确定的情况下用



```
mDemoCollectionPagerAdapter =
    new DemoCollectionPagerAdapter(
        getSupportFragmentManager());
mViewPager = (ViewPager) findViewById(R.id.pager);
mViewPager.setAdapter(mDemoCollectionPagerAdapter);

public class DemoCollectionPagerAdapter extends FragmentStatePagerAdapter {
    public DemoCollectionPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int i) {
        Fragment fragment = new DemoObjectFragment();
        Bundle args = new Bundle();
        // Our object is just an integer :-P
        args.putInt(DemoObjectFragment.ARG_OBJECT, i + 1);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public int getCount() {
        return 100;
    }

    @Override
    public CharSequence getPageTitle(int position) {
        return "OBJECT " + (position + 1);
    }
}
```

## 【课堂实验 KT5.3】:为 ActionBar 增加 Tab

【工程模块】: TabActionBarSample

【工程描述】: 该程序展示了如何为 ActionBar 增加上 tab。

【目标要求】:

掌握添加 tab 的方法

掌握 tab 点击事件的写法

**【实验步骤】:**

- 1、运行程序，观察现象。
- 2、分析理解代码

**【知识点注释】:**

- 1、 让 ActionBar 拥有 tab 风格,可用方法: `setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);`  
注意这个方法, 对由 Toolbar 构成的 actionBar 无效
- 2、 然后调用 `addTab` 添加 tab 即可, 可以添加上标题和 icon。
- 3、 `actionBar.addTab(`

```
        actionBar.newTab()  
            .setText("Tab " + (i + 1))  
            .setTabListener(tabListener));
```

- 4、 实现选项卡单击事件接口

```
ActionBar.TabListener tabListener = new ActionBar.TabListener() {
```

```
final ActionBar actionBar = getSupportActionBar();

// Specify that tabs should be displayed in the action bar.
actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

// Create a tab listener that is called when the user changes tabs.
ActionBar.TabListener tabListener = new ActionBar.TabListener() {

    public void onTabSelected(ActionBar.Tab tab, FragmentTransaction ft) {
        // show the given tab
        ContentFragment cf = new ContentFragment();
        Bundle args = new Bundle();
        args.putInt(ContentFragment.ARG_POSITION, tab.getPosition());
        cf.setArguments(args);
        FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();

        // Replace whatever is in the fragment_container view with this fragment,
        // and add the transaction to the back stack so the user can navigate back
        transaction.replace(R.id.fragment_container, cf);
        transaction.addToBackStack(null);

        // Commit the transaction
        transaction.commit();
    }

    public void onTabUnselected(ActionBar.Tab tab, FragmentTransaction ft) {
        // hide the given tab
    }

    public void onTabReselected(ActionBar.Tab tab, FragmentTransaction ft) {
        // probably ignore this event
    }
};

// Add 3 tabs, specifying the tab's text and TabListener
for (int i = 0; i < 3; i++) {
    actionBar.addTab(
        actionBar.newTab()
            .setText("Tab " + (i + 1))
            .setTabListener(tabListener));
}

;
```

## 【课堂实验 KT5.4】:Tab+Swipe 的实现

【工程模块】: TabSwipeSample

【工程描述】: 展示了 TAB 和 swipe 相结合的 UI 设计。

【目标要求】:

掌握相关回调事件的写法。

【实验步骤】:

1、运行程序观察并分析代码。

【知识点注释】

1、Tab 和 swipe 的实现可参见上面的例子，把它们相结合就是实现两个事件。

当用户点击 tab 的时候，swipe 应该同时滚动到相应的页面，而当用户滑动页面时候，当前 tab 也应相应变化。因此，只要实现两个回调函数即可。

```
mViewPager.setOnPageChangeListener(new ViewPager.SimpleOnPageChangeListener() {
    @Override
    public void onPageSelected(int position){
        actionBar.setSelectedNavigationItem(position);
    }
});
.....
ActionBar.TabListener tabListener = new ActionBar.TabListener() {
    public void onTabSelected(ActionBar.Tab tab, FragmentTransaction ft) {
        // show the given tab
        mViewPager.setCurrentItem(tab.getPosition());
    }
}
```

2、本例中 viewpage 的 adapter 继承自 FragmentpageAdapter

## 【课堂实验 KT5.5】:NavigationDrawer 的用法

【工程模块】: NavigationDrawerSample

【工程描述】: 展示了 drawer 的例子。

【目标要求】:

掌握使用 drawer 的基本方法。

【实验步骤】:

5、运行程序观察并分析代码。

【知识点注释】

一、DrawerLayout 是 NavigationDrawe 的主类，可以在 layout 中中加入相应标签。

```

<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <!-- The main content view -->
    <FrameLayout
        android:id="@+id/content_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <!-- The navigation drawer -->
    <ListView android:id="@+id/left_drawer"
        android:layout_width="240dp"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:choiceMode="singleChoice"
        android:divider="@android:color/transparent"
        android:dividerHeight="0dp"
        android:background="#111"/>
</android.support.v4.widget.DrawerLayout>

```

该例中 `FrameLayout` 是作为内容 `view` 的区域，必须是第一个子标签，这样在 UI 中，其 `z` 轴位置出于顶端。且 `width`，`height` 属性要被设为 `match_parent`。

作为 `Drawview` 的 `listview` 必须有一个 `layout_gravity` 属性，会根据不同语言的左右显示方式来显示。其 `width` 不要超过 `320dp`，这样可以给内容 `view` 留出部分显示空间。

二、使用 `DrawerLayout` 的一般过程如下：

1、创建 `DrawerLayout` 对象（可以如上在 `layout` 中加入）

2、获得 `DrawerLayout` 中的 `listview` 对象，并设置内容（`setAdapter()`）。和设置点击事件接口（`mDrawerList.setOnItemClickListener`）

4、在接口类中，创建具体的内容 `fragment` 并嵌入到 `DrawerLayout` 中的内容 `view` 中  
`fragmentManager.beginTransaction()`

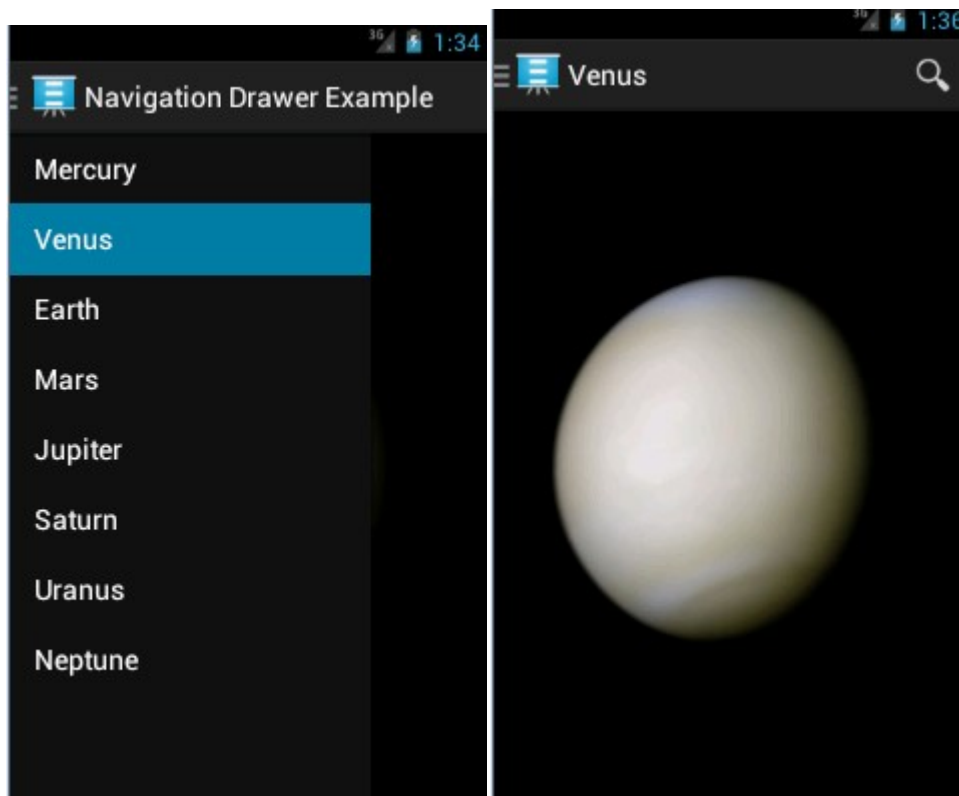
`.replace(R.id.content_frame, fragment)`

`.commit();`

5、创建 `ActionBarDrawerToggle` 对象，关联该 `darwerlayout`。使得在 `actionbar` 上可以通过按钮显示和隐藏 `drawer`。

```
mDrawerToggle = new ActionBarDrawerToggle(  
    this,                /* host Activity */  
    mDrawerLayout,        /* DrawerLayout object */  
    R.drawable.ic_drawer, /* nav drawer icon to replace 'Up' caret */  
    R.string.drawer_open,  /* "open drawer" description */  
    R.string.drawer_close  /* "close drawer" description */  
) {  
  
    /** Called when a drawer has settled in a completely closed state. */  
    public void onDrawerClosed(View view) {  
        getActionBar().setTitle(mTitle);  
    }  
  
    /** Called when a drawer has settled in a completely open state. */  
    public void onDrawerOpened(View drawerView) {  
        getActionBar().setTitle(mDrawerTitle);  
    }  
};  
  
// Set the drawer toggle as the DrawerListener  
mDrawerLayout.setDrawerListener(mDrawerToggle);  
  
getActionBar().setDisplayHomeAsUpEnabled(true);  
getActionBar().setHomeButtonEnabled(true);
```

【问题】（@编程练习）用 drawer 实现一个如下相册功能，点击选项，切换照片。照片文字可自行替换。



## 【课堂实验 KT5.5】:用 Toolbar 替代 ActionBar

【工程模块】: ToolbarSample

【工程描述】: 展示了用 Toolbar 替代 actionbar 的例子。

【目标要求】:

掌握 toolbar 的基本用法。

【实验步骤】:

1、运行程序观察并分析代码。

【知识点注释】

ToolBar 是 5.0 后引入的,其用法灵活,功能强大,有两种用法,一种是替代原来的 actionBar。

`setSupportActionBar(mToolbar)`

另一种则是按照普通的 viewgroup 来用。

可以在 activity 中加入多个 toolbar.

注意,若采用第一种用法,即调用 `setSupportActionBar(mToolbar)`

用 toolbar 替换 actionBar, 绑定菜单的方法和 actionBar 类似, 需要实现回调函数 `onOptionsItemSelected()`

, 同时 toolbar 自带的绑定 menu 的方法就无效了。

`mToolbar.setNavigationIcon(R.drawable.mic);`

而菜单事件可以像 actionBar 一样通过实现 `onOptionsItemSelected()` 来执行, 也可用 toolbar 自带的函数来完成:

```
mToolbar.setOnMenuItemClickListener(new Toolbar.OnMenuItemClickListener() {  
    // @Override
```

```
//      public boolean onOptionsItemSelected(MenuItem item) {  
//          switch (item.getItemId()) {  
//              case R.id.mic:  
//                  return true;  
//  
//              default:  
//                  return false;  
//          }  
//      }  
//  }  
//  });
```

本例中还实现了一个 `searchview` 和 `actionprovider`



```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item
    android:id="@+id/mic"
    android:icon="@drawable/mic"
    android:title="@string/app_name"
    app:showAsAction="ifRoom"/>
    <item android:id="@+id/action_share"
      android:title="share"
      app:showAsAction="ifRoom"
      app:actionProviderClass="android.support.v7.widget.ShareActionProvider"/>
    <item android:id="@+id/action_search"
      android:title="search"
      android:icon="@drawable/mic"
      app:showAsAction="ifRoom|collapseActionView"
      app:actionViewClass="android.support.v7.widget.SearchView" />
    <item
      android:id="@+id/mic1"
      android:icon="@drawable/mic"
      android:title="@string/app_name"
      app:showAsAction="never"/>
  </menu>

```

```
@Override
```

```

public boolean onCreateOptionsMenu(Menu menu) {

    getMenuInflater().inflate(R.menu.toolbar_menu, menu);
    MenuItem menuItem1 = menu.findItem(R.id.action_share);
    ShareActionProvider sap=(ShareActionProvider)
MenuItemCompat.getActionProvider(menuItem1);
    sap.setShareIntent(getDefaultIntent());

    return true;
}

private Intent getDefaultIntent() {
    Intent intent=new Intent(Intent.ACTION_SEND);
    intent.setType("image/*");
    return intent;
}

```

## 【课后练习 KH5.1】:app 需求分析

【要求】: (@下周提交 12 月 30 前) 文档: 学号+姓名+KH51.DOC

分析一个熟悉的 app (例如: 大众点评、携程、酷我、微信), 逆向工程其需求报告。请写明主要功能列表。

绘出 UI 层次关系图, 标明调用关系。分析其每一个窗口 UI 的构造方法, 判断其合理性。

## 【课后练习 KH5.2】:UI 综合练习

【要求】: (@团队练习, 下下周提交 1 月 6 日前) 文档+工程: 第 x 组+KH52.RAR

假如你要设计一个菜谱 app, 主要为用户推荐菜谱, 除了能分类按照季节等推荐并展示各类菜谱外, 还能让用户收藏菜谱。用户还可以搜索感兴趣的菜谱。请为该 app 设计 UI:

- A、列出全部可能用到的窗口, 并简要说明个窗口的作用
- B、画出窗口结构图, 标明调用关系。
- C、用原型工具, 或者绘图工具完成全部 UI 的设计。
- D、使用本章所学内容, 设计其首页+另外的主要功能窗口 (不少于 2 个)