**PRACTICAL #**

**OBJECT:**

Working with Fragments in Android

**THEORY:**

A fragment is an independent Android component, typically used by activities. A fragment runs in the context of an activity, but has its own life cycle and typically its own user interface. Fragments allow to design very flexible user interfaces, as they can be dynamically added and removed from an activity.

Fragments can also be defined without a user interface, known as headless fragments.

Android devices exists in a variety of screen sizes and densities. You can build single-pane layouts for phones and multi-pane layouts for tablets.

The typical example is a list of items in an activity. On a tablet you see the details immediately on the same screen on the right hand side if you click on item. On a smartphone you jump to a new detail screen.

This lab assumes that you have two fragments (master and detail), but you can have more. We will also have one main activity and one detailed activity. On a tablet the main activity contains both fragments in its layout, on a mobile device it only contains the main fragment (shown in Figures 1 and 2).



Figure 1: Master (1) and details (2) fragments on mobile device

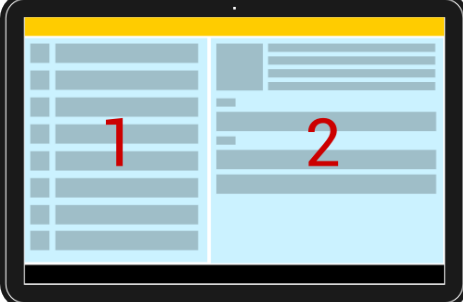


Figure 2: Master (1) and details (2) fragments on Tablet

It is possible to define in the layout file of an activity that it contains fragments (static definition). You can also modify the fragments of an activity at runtime (dynamic definition).

To display different fragments in your activities based on the actual display space you can:

a) Use one activity, which displays two fragments for tablets and on phone. In this case change at runtime the fragments displayed by the activity whenever necessary.

b) Use separate activities to host each fragment on a handset. An activity may include multiple fragments for a certain configuration and for other configurations a different number. If a detailed fragment is there, the main activity tells the fragment that it should update itself. If a detail fragment is not available, the main activity starts the detailed activity.

This demonstrates how to use fragments in a standard Android application. We create the project RSS Reader. RSS is a type of web feed which allows users to access updates to online content in a standardized, computer-readable format.

The application uses different numbers of fragments depending on portrait and landscape mode.

In portrait mode the main activity shows one fragment. From this fragment the user can navigate to another activity which contains another fragment.

In landscape mode the main activity shows both fragments side by side.

**Create a new Android project with the following specifications:**

**Application Name: RSS Reader**

**Template: Empty Activity**

**Backward compatibility: Not selected**

Make sure that you activity extends Activity to ensure you are not using the backwards compatibility mode. Since the fragments are not supported in older version of Android.

**Create a new layout file called fragment\_rssitem\_detail.xml in the res/layout/ folder.**

*<?xml version="1.0" encoding="utf-8"?>*

*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"*

*android:layout\_width="match\_parent"*

*android:layout\_height="match\_parent"*

*android:orientation="vertical" >*

*<TextView*

*android:id="@+id/detailsText"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="match\_parent"*

*android:layout\_gravity="center\_horizontal|center\_vertical"*

*android:layout\_marginTop="20dip"*

*android:text="Default Text"*

*android:textAppearance="?android:attr/textAppearanceLarge"*

*android:textSize="30dip" />*

*</LinearLayout>*

**Create a new layout file called fragment\_rsslist\_overview.xml.**

*<?xml version="1.0" encoding="utf-8"?>*

*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"*

*android:layout\_width="match\_parent"*

*android:layout\_height="match\_parent"*

*android:orientation="vertical" >*

*<Button*

*android:id="@+id/updateButton"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:text="Press to update"/>*

*</LinearLayout>*

**Create fragment classes:**

Create the following classes which are used as fragments. Start with the DetailFragment class:

*import android.app.Fragment;*

*import android.os.Bundle;*

*import android.view.LayoutInflater;*

*import android.view.View;*

*import android.view.ViewGroup;*

*import android.widget.TextView;*

*public class DetailFragment extends Fragment {*

*@Override*

*public View onCreateView(LayoutInflater inflater, ViewGroup container,*

*Bundle savedInstanceState) {*

*View view = inflater.inflate(R.layout.fragment\_rssitem\_detail,*

*container, false);*

*return view;*

*}*

*public void setText(String text) {*

*TextView view = (TextView) getView().findViewById(R.id.detailsText);*

*view.setText(text);*

*}*

*}*

Create the MyListFragment class. Despite its name it will not display a list of items. Instead it will just have a button which allow to send the current time to the details fragment.

*import android.app.Fragment;*

*import android.content.Context;*

*import android.os.Bundle;*

*import android.view.LayoutInflater;*

*import android.view.View;*

*import android.view.ViewGroup;*

*import android.widget.Button;*

*public class MyListFragment extends Fragment {*

*private OnItemSelectedListener listener;*

*@Override*

*public View onCreateView(LayoutInflater inflater, ViewGroup container,*

*Bundle savedInstanceState) {*

*View view = inflater.inflate(R.layout.fragment\_rsslist\_overview,*

*container, false);*

*Button button = (Button) view.findViewById(R.id.updateButton);*

*button.setOnClickListener(new View.OnClickListener() {*

*@Override*

*public void onClick(View v) {*

*updateDetail("testing");*

*}*

*});*

*return view;*

*}*

*public interface OnItemSelectedListener {*

*void onRssItemSelected(String text);*

*}*

*@Override*

*public void onAttach(Context context) {*

*super.onAttach(context);*

*if (context instanceof OnItemSelectedListener) {*

*listener = (OnItemSelectedListener) context;*

*} else {*

*throw new ClassCastException(context.toString()*

*+ " must implement MyListFragment.OnItemSelectedListener");*

*}*

*}*

*// triggers update of the details fragment*

*public void updateDetail(String uri) {*

*// create fake data*

*String newTime = String.valueOf(System.currentTimeMillis() + uri);*

*// send data to activity*

*listener.onRssItemSelected(newTime);*

*}*

*}*

Change the main layout file:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:baselineAligned="false"

android:orientation="horizontal" >

<fragment

android:id="@+id/listFragment"

android:layout\_width="0dp"

android:layout\_weight="1"

android:layout\_height="match\_parent"

class="com.example.android.rssreader.MyListFragment"

tools:layout="@layout/fragment\_rsslist\_overview">

</fragment>

<fragment

android:id="@+id/detailFragment"

android:layout\_width="0dp"

android:layout\_weight="2"

android:layout\_height="match\_parent"

class="com.example.android.rssreader.DetailFragment"

tools:layout="@layout/fragment\_rssitem\_detail">

</fragment>

</LinearLayout>

Change the RssfeedActivity class so that it can act as call back for the list fragment and update the detailed fragment:

import android.app.Activity;

import android.os.Bundle;

public class RssfeedActivity extends Activity implements MyListFragment.OnItemSelectedListener{

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_rssfeed);

}

@Override

public void onRssItemSelected(String text) {

DetailFragment fragment = (DetailFragment) getFragmentManager()

.findFragmentById(R.id.detailFragment);

fragment.setText(text);

}

}

**Verify the application:**

Start your application. Both fragments should be displayed both in landscape and portrait mode. You can use the emulator controls to change the orientation of the emulator. If you press the button in the ListFragment, the DetailFragment gets updated.

**ACTIVITIES**

**Activity 1**

Create a projects called TechArticles. The master fragment contains the article titles. Upon touching an article title, a detail fragment opens to show details.

**REVIEW QUESTIONS**

1. What are fragments?
2. What are different types of fragment subclasses available?
3. What are the advantages of using fragments?
4. How are fragments displayed on a phone and a tablet?