

5COSC005W MOBILE APPLICATION DEVELOPMENT

Lecture 10: More on User Interfaces - Working with Fragments

Dr Dimitris C. Dracopoulos

Module Web page:

<https://dracopd.users.ecs.westminster.ac.uk/DOCUM/courses/5cosc005w/5cosc005w.html>

What are Fragments?

Using an activity a user can perform a single task per screen. This might not be necessarily ideal, especially for larger screens.

- A Fragment (from Android 3.0 - API 11) represents a behavior or a portion of user interface in an Activity.
- Multiple fragments can be used in a single activity to build a multi-pane UI.
- Fragments can be re-used across multiple activities.

A fragment has to be embedded within an activity and its lifecycle is affected by the activity's lifecycle.

An example

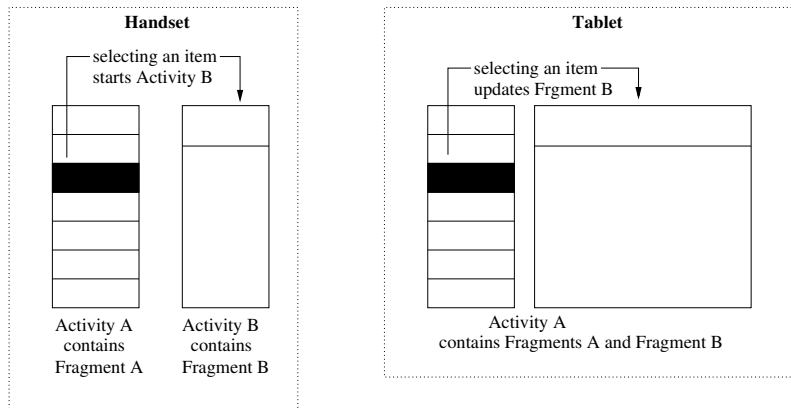


Figure 1: An example of an application in which 2 separate UI modules implemented by fragments can be embedded in a single activity for a tablet design but separated for a handset design.

Creating a Fragment

A class has to be created which is a subclass of class `Fragment` or another subclass of class `Fragment`.

The following lifecycle methods have to be implemented at least:

- `onCreate()`: Called when creating the fragment. Initialize essential components of the fragment that you want to retain when the fragment is paused or stopped, then resumed.
- `onCreateView()`: Called it is time for the fragment to draw its UI.
- `onPause()`: First indication that the user is leaving the fragment (though it does not always mean the fragment is being destroyed). Commit any changes that should be persisted beyond the current user session (because the user might not come back).

Subclasses of class Fragment

Common subclasses of class `Fragment` which you might want to subclass to create your own fragment:

- `DialogFragment`
- `ListFragment`
- `PreferenceFragment`

Adding a user interface to a Fragment

```
public class ExampleFragment extends Fragment {
    public View onCreateView(LayoutInflater inflater,
                            ViewGroup container,
                            Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.example_fragment,
                               container, false);
    }
}
```

Note: If your fragment is a subclass of `ListFragment`, you do not need to override this as the parent class implementation creates and returns a `ListView`.

Adding a Fragment to an activity

- Statically (using the activity's XML layout files)
- Programmatically

Statically adding a Fragment

Use the activity's layout XML:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment android:name="com.example.news.ArticleListFragment"
        android:id="@+id/list"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
    <fragment android:name="com.example.news.ArticleReaderFragment"
        android:id="@+id/viewer"
        android:layout_weight="2"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
</LinearLayout>
```


Programmatically adding a Fragment

Specify a ViewGroup to add the fragment:

```
FragmentManager fragmentManager = getFragmentManager()
FragmentManager.beginTransaction()
    fragmentManager.beginTransaction();
```

```
ExampleFragment fragment = new ExampleFragment();
fragmentTransaction.add(R.id.fragment_container, fragment);
```

```
/* optionally add the transaction to the back stack in case
   the user presses the Back button to get back to the fragment */
fragmentTransaction.addToBackStack(null);
```

```
// Commit the transaction
fragmentTransaction.commit();
```

Replacing or removing a Fragment

The `replace()` or `remove()` methods can be used instead of `add()`. For example, to replace whatever fragment is in the fragment container view:

```
// Create new fragment and transaction
Fragment newFragment = new ExampleFragment();
FragmentManager transaction =
    getSupportFragmentManager().beginTransaction();

/* Replace whatever is in the fragment_container view with
   this fragment and add the transaction to the back stack */
transaction.replace(R.id.fragment_container, newFragment);
transaction.addToBackStack(null);

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

How a Fragment communicates with its activity?

Although a Fragment is implemented independently from its Activity and can be used within multiple activities, there are 2 ways in which it can communicate with the activity it is embedded into:

- Use the `getActivity()` method

```
View listView = getActivity().findViewById(R.id.list);
```

Similarly the activity can obtain a reference to the fragment using `findFragmentById()` or `findFragmentByTag()`:

```
ExampleFragment fragment = (ExampleFragment)  
    getFragmentManager().findFragmentById(R.id.example_fragment);
```

- Create event callbacks to the activity using an interface

Creating event callbacks to the activity

- Define an interface inside the fragment
- Require that the host activity implements it

Now, when the activity receives a callback through the interface, it can share the information with other fragments in the layout if it needs to.

Example of creating event callbacks to the activity

A news application has two fragments in an activity:

- 1 Fragment A to show a list of articles
- 2 Fragment B to display an article

Fragment A must tell the activity when a list item is selected so that the activity can tell fragment B to display the article.

The `OnArticleSelectedListener` interface is declared inside fragment A:

```
public class FragmentA extends ListFragment {
    ...
    // Container Activity must implement this interface
    public interface OnArticleSelectedListener {
        public void onArticleSelected(int index);
    }
    ...
}
```

Example of creating event callbacks to the activity (cont'd)

To ensure that the host activity implements this interface, fragment A's `onAttach()` callback method (which the system calls when adding the fragment to the activity) instantiates an instance of `OnArticleSelectedListener` by casting the `Activity` that is passed into `onAttach()`:

```
public class FragmentA extends ListFragment {
    OnArticleSelectedListener mListener;
    ...
    public void onAttach(Activity activity) {
        super.onAttach(activity);
        try {
            mListener = (OnArticleSelectedListener) activity;
        } catch (ClassCastException e) {
            throw new ClassCastException(activity.toString() +
                " must implement OnArticleSelectedListener");
        }
    }
    ...
}
```

Example of creating event callbacks to the activity (cont'd)

```
public static class FragmentA extends ListFragment {
    OnArticleSelectedListener mListener;
    ...
    public void onListItemClick(ListView l, View v, int position,
                                long id) {
        // Send the event and Uri to the host activity
        mListener.onArticleSelected(position);
    }
    ...
}
```

Now the activity know which article was selected (i.e. it knows the position of the list item selected) and can instruct from within its implementation of the `onArticleSelected(int pos)` method the second Fragment B to display the selected article.

The Shakespeare Example

An activity displays a list of Shakespeare play titles using a fragment. A second fragment is used to show a summary of the play when selected from the list.

Based on the screen configuration (landscape or portrait) the 2 fragments can be displayed simultaneously (landscape) or a second activity created which displays the second fragment.

The Shakespeare Example (cont'd)

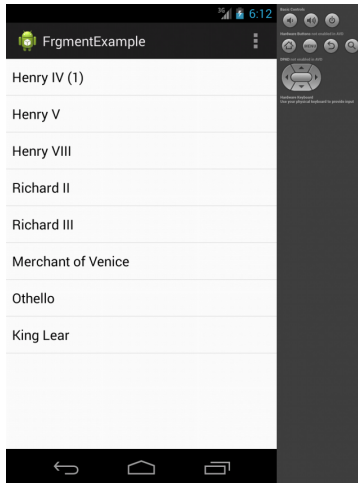


Figure 2: Portrait mode for the Shakespeare example.

The Shakespeare Example (cont'd)

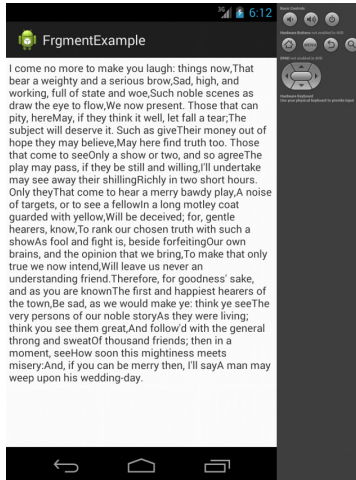


Figure 3: Portrait mode for the Shakespeare example shown once a title has been selected.

The Shakespeare Example (cont'd)

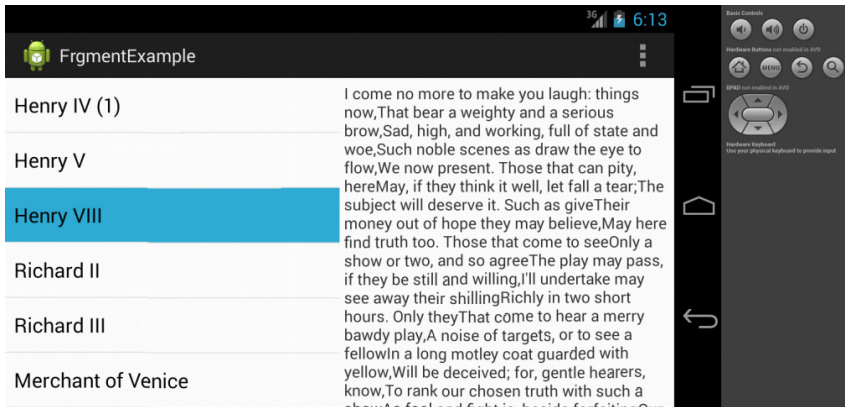


Figure 4: Landscape mode for the Shakespeare example.

The Shakespeare Example (cont'd)

The MainActivity:

```
package com.example.frgmentexample;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        /* Inflate the menu; this adds items to the action bar
           if it is present. */
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

The Shakespeare Example (cont'd)

The layout of the main activity (portrait mode):

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment class="com.example.frgmentexample.TitlesFragment"
        android:id="@+id/titles"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</FrameLayout>
```

The Shakespeare Example (cont'd)

The layout of the main activity (landscape mode - it should be placed in the res/layout-land directory):

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <fragment class="com.example.frgmentexample.TitlesFragment"
        android:id="@+id/titles"
        android:layout_weight="1"
        android:layout_width="0px"
        android:layout_height="match_parent" />

    <FrameLayout android:id="@+id/details"
        android:layout_weight="1"
        android:layout_width="0px"
        android:layout_height="match_parent"/>
</LinearLayout>
```



```
// Populate list with our static array of titles.
setListAdapter(new ArrayAdapter<String>(getActivity(),
    android.R.layout.simple_list_item_activated_1,
    Shakespeare.TITLES));

// Check to see if we have a frame in which to embed the details
// fragment directly in the containing UI.
View detailsFrame = getActivity().findViewById(R.id.details);
mDualPane = detailsFrame != null &&
    detailsFrame.getVisibility() == View.VISIBLE;

if (savedInstanceState != null) {
    // Restore last state for checked position.
    mCurCheckPosition = savedInstanceState.getInt("curChoice", 0);
}

if (mDualPane) {
    // In dual-pane mode, the list view highlights the selected item
    getListView().setChoiceMode(ListView.CHOICE_MODE_SINGLE);
    // Make sure our UI is in the correct state.
    showDetails(mCurCheckPosition);
}}
```


The Shakespeare Example (cont'd)

```
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putInt("curChoice", mCurCheckPosition);
}
```

```
public void onItemClick(AdapterView l, View v,
                        int position, long id) {
    showDetails(position);
}
```

```
/**
 * Helper function to show the details of a selected item,
 * either by displaying a fragment in-place in the current UI,
 * or starting a whole new activity in which it is displayed. */
void showDetails(int index) {
    mCurCheckPosition = index;

    if (mDualPane) {
        // We can display everything in-place with fragments,
        // so update the list to highlight the selected item
        //and show the data.
    }
}
```

The Shakespeare Example (cont'd)

```
getListView().setItemChecked(index, true);

// Check what fragment is currently shown, replace if needed.
DetailsFragment details = (DetailsFragment)
    fragmentManager().findFragmentById(R.id.details);
if (details == null || details.getShownIndex() != index) {
    // Make new fragment to show this selection.
    details = DetailsFragment.newInstance(index);

    // Execute a transaction, replacing any existing fragment
    // with this one inside the frame.
    FragmentTransaction ft = fragmentManager().beginTransaction();
    ft.replace(R.id.details, details);

    ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
    ft.commit();
}
} else {
    // Otherwise we need to launch a new activity to display
    // the dialog fragment with selected text.
    Intent intent = new Intent();
```

The Shakespeare Example (cont'd)

```
intent.setClass(getActivity(), DetailsActivity.class);  
    intent.putExtra("index", index);  
    startActivity(intent);  
}  
}  
}
```

The Shakespeare Example (cont'd)

The DetailsFragment.java:

```
package com.example.frgmentexample;

import android.app.Fragment;
import android.os.Bundle;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ScrollView;
import android.widget.TextView;

public class DetailsFragment extends Fragment {
    /**
     * Create a new instance of DetailsFragment, initialized to
     * show the text at 'index'.
     */
    public static DetailsFragment newInstance(int index) {
        DetailsFragment f = new DetailsFragment();
    }
}
```

The Shakespeare Example (cont'd)

```
// Supply index input as an argument.
Bundle args = new Bundle();
args.putInt("index", index);
f.setArguments(args);

return f;
}

public int getShownIndex() {
    return getArguments().getInt("index", 0);
}

public View onCreateView(LayoutInflater inflater,
                          ViewGroup container, Bundle savedInstanceState) {
    if (container == null) {
        // We have different layouts, and in one of them this
        // fragment's containing frame doesn't exist. The fragment
        // may still be created from its saved state, but there is
        // no reason to try to create its view hierarchy because it
        // won't be displayed. Note this is not needed -- we could
        // just run the code below, where we would create and return
        // the view hierarchy; it would just never be used.
    }
}
```

The Shakespeare Example (cont'd)

```
        return null;
    }
```

```
ScrollView scroller = new ScrollView(getActivity());
TextView text = new TextView(getActivity());
int padding = (int)TypedValue.applyDimension(
    TypedValue.COMPLEX_UNIT_DIP,
    4, getActivity().getResources().getDisplayMetrics());
text.setPadding(padding, padding, padding, padding);
scroller.addView(text);
text.setText(Shakespeare.DIALOGUE[getShownIndex()]);
return scroller;
}
```

The Shakespeare Example (cont'd)

The Shakespeare.java file:

```
package com.example.frgmentexample;

public final class Shakespeare {
    /**
     * Our data, part 1.
     */
    public static final String[] TITLES =
    {
        "Henry IV (1)",
        "Henry V",
        "Henry VIII",
        "Richard II",
        "Richard III",
        "Merchant of Venice",
        "Othello",
        "King Lear"
    };
};
```

The Shakespeare Example (cont'd)

```
/**
 * Our data, part 2.
 */
public static final String[] DIALOGUE =
{
    "So shaken as we are, so wan with care," +
    "Find we a time for frightened peace to pant," +
    "And breathe short-winded accents of new broils" +
    "To be commenced in strands afar remote.",

    "Hear him but reason in divinity," +
    "And all-admiring with an inward wish" +
    "You would desire the king were made a prelate:" +
    "Hear him debate of commonwealth affairs,",

    "I come no more to make you laugh: things now," +
    "That bear a weighty and a serious brow," +
    "Sad, high, and working, full of state and woe," +
    "Such noble scenes as draw the eye to flow,",

    "First, heaven be the record to my speech!" +
    "In the devotion of a subject's love," +
```


The Shakespeare Example (cont'd)

"Tendering the precious safety of my prince," +
"And free from other misbegotten hate," +
"Come I appellat to this princely presence.",

"Now is the winter of our discontent" +
"Made glorious summer by this sun of York;" +
"And all the clouds that lour'd upon our house" +
"In the deep bosom of the ocean buried.",

"To bait fish withal: if it will feed nothing else," +
"it will feed my revenge. He hath disgraced me, and" +
"hindered me half a million; laughed at my losses," +
"mocked at my gains, scorned my nation, thwarted my" +
"bargains, cooled my friends, heated mine" +
"enemies; and what's his reason? I am a Jew. Hath",

"Virtue! a fig! 'tis in ourselves that we are thus" +
"or thus. Our bodies are our gardens, to the which" +

The Shakespeare Example (cont'd)

```
"our wills are gardeners:",
```

```
"Blow, winds, and crack your cheeks! rage! blow!" +
```

```
"You cataracts and hurricanoes, spout" +
```

```
"Till you have drench'd our steeples"
```

```
};
```

```
}
```

The Shakespeare Example (cont'd)

The DetailsActivity.java file:

```
package com.example.frgmentexample;
```

```
import android.app.Activity;
```

```
import android.content.res.Configuration;
```

```
import android.os.Bundle;
```

```
public class DetailsActivity extends Activity {
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        if (getResources().getConfiguration().orientation
```

```
            == Configuration.ORIENTATION_LANDSCAPE) {
```

```
            // If the screen is now in landscape mode, we can show the
```

```
            // dialog in-line with the list so we don't need this activity.
```

```
            finish();
```

```
            return;
```

```
        }
```

The Shakespeare Example (cont'd)

```
if (savedInstanceState == null) {  
    // During initial setup, plug in the details fragment.  
    DetailsFragment details = new DetailsFragment();  
    details.setArguments(getIntent().getExtras());  
    getFragmentManager().beginTransaction().add(  
        android.R.id.content, details).commit();  
}  
}  
}
```

The Shakespeare Example (cont'd)

In the manifest file add the `DetailsActivity` activity and change the minimum SDK version to 11.

Further Reading

<http://developer.android.com/guide/components/fragments.html>