
globsyn
Taking People To The Next Level

globsyn

globsyn
finishing school

Building dynamic UI with fragments

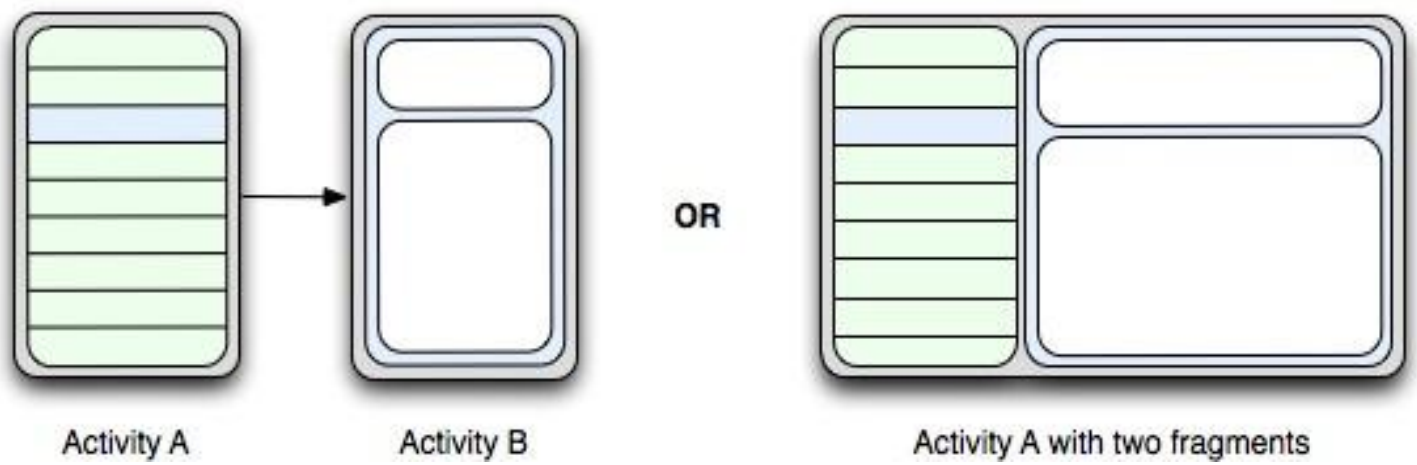
Building dynamic UI with fragments

- Topics to be covered in this session:
  - Fragments
  - Building dynamic UI with fragments - program

Introduction to Fragments

- An activity describes how a View should look like.
- Let us assume, that you have defined an activity for an mobile application.
- Now, the same application when run in a Tablet or a Computer might look awkward with a mobile based View.
- Fragments are used to overcome this problem. They are a collection of mini-activities each with their own set of Views.

Understanding Fragments Contd.



- Here in the first example, it is a mobile phone that is executing. Hence, Activity A and the called Activity B both are displayed in separate, single fragments.

- But, in the second figure, it is a Tablet and hence, Activity A has two Fragments one of which displays Activity B.

Fragments Inside Activity

- It lives in a ViewGroup inside the activity's view hierarchy
- Fragment has its own view layout.
- via XML: Insert a fragment into your activity layout by declaring the fragment in the activity's layout file, as a <fragment> element,
- via CODE: from your application code by adding it to an existing ViewGroup.

Methods Used in Fragments

- Some of the most common methods used in Fragments are,

.onAttach(Activity) – Used to associate a fragment with an activity.

.onCreate(Bundle) – Used to create the Fragment Bundle.

.onStart() – Starts the execution of the Fragment and makes it visible.

.OnCreateView(LayoutInflater,ViewGroup,Bundle) – This is used for the creation and return of the View that is associated with the Fragment.

Methods Used in Fragments Contd.

- OnPause() – Used to pause the interaction between the user and the fragment.
- OnStop()– The display of the fragment is stopped.
- onDestroyView() – It is used to destroy the View that is associated with the fragments. .
- OnDetach() – Detaches the user from the current Fragment and calls the fragment that is present immediately above.

The User Interface in Fragments

- Most fragments have an UI associated with them.
- They will also have a layout associated with them.
- What we are going to do is, create two fragments, one for Activity A and another for Activity B. So, now, when the display screen is large enough(i.e. in a Tablet) both the Views should be displayed together in a single Window itself.

Creating Two Fragments

- The First Step involved is the creation of two fragments. Let us name them TutListFragment and TutViewerFragment. In the TutListFragment, you need to override the existing onListItemClick() and onCreate()

- However, on the TutViewerFragment, you need to override only the onCreateView() because, we are under the impression that the device is a Tablet and we are going to display the new fragment in the same activity itself while using TutViewerFragment

Code - Two Fragments with overriding

TutListFragment

```
@Override
public void onListItemClick(ListView l, View v, int
position, long id) {
    String[] links =
getResources().getStringArray(R.array.tut_links);
    String content = links[position];
    Intent showContent = new
Intent(getActivity().getApplicationContext(),
            TutViewerActivity.class);
    showContent.setData(Uri.parse(content));
    startActivity(showContent);
}
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setListAdapter(ArrayAdapter.createFromResource(g
etActivity()
            .getApplicationContext(), R.array.tut_titles,
            R.layout.list_item));
}
```

TutViewerFragment

```
@Override
public View onCreateView(LayoutInflater
inflater, ViewGroup container,
        Bundle savedInstanceState) {
    Intent launchingIntent =
getActivity().getIntent();
    String content =
launchingIntent.getData().toString();
    WebView viewer = (WebView)
inflater.inflate(R.layout.tut_view,
container, false);
    viewer.loadUrl(content);
    return viewer;
}
```

Creating Layouts for the Fragments

- Now, you have to create two different Layouts for our two different Fragments.

```
<?xml version="1.0" encoding="utf-8"?>
<fragment
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="com.mamlambo.tutorial.tutlist.TutListFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/tutlist_fragment">
</fragment>
```

```
<?xml version="1.0" encoding="utf-8"?>
<fragment
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="com.mamlambo.tutorial.tutlist.TutViewerFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/tutview_fragment">
</fragment>
```

Activity Class Update

- Since you have changed the layout of TutListFragment and TutViewerFragment, you need to update their activity class too.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.tutlist_fragment);
}
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.tutview_fragment);
}
```

Changing Communication

- To make two activities appear on the same page, you need to change the mode of communication between them.
- That is, you need to make the fragments independent of the activities. This can be done with the help of an actionlistener.
- Similarly, in TutViewerListener, you need to add updateUrl() method into the onCreateView() instead of other methods. By doing that, you could accommodate two activities on a single fragment.

How to Make Applications More Accessible?

- If you are a developed, you should be wondering how you can make your applications more accessible by using the features that Android provides.
- Add descriptive Text: If you add descriptive text to your controls, it can help people with physical and age-related limitations.
- Additional Inputs: If your application can get input from additional sources like a Trackpad or even through gestures instead of just through the Touchscreen, it would make the app extremely accessible.

Creating a Layout With Both Fragments and Providing Dynamic change

- Now, you need to create a LinearLayout with both the fragments. Thus, when a Landscape (Tablet) device is used, a different layout will be provided.
- When a mobile phone is used, the default layout will be given.
- After that, you need to add a dynamic choice option. That is, on run time, the app will detect if the device is a mobile phone or a tablet and it will display the suitable fragment. To do that, you need to make some changes in the TutSelected() method of the TutListFragment's activity.

Code : Single Layout With Multiple Fragments

```
<?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="horizontal">
   <fragment
      android:name="com.mamlambo.tutorial.tutlist.TutListFragment"
      android:layout_width="0dp"
      android:layout_height="match_parent"
      android:id="@+id/tutlist_fragment"
      android:layout_weight="45">
   </fragment>
   <fragment
      android:name="com.mamlambo.tutorial.tutlist.TutViewerFragment"
      android:layout_width="0dp"
      android:layout_height="match_parent"
      android:id="@+id/tutview_fragment"
      android:layout_weight="55">
   </fragment>
</LinearLayout>
```

Code : Dynamic Option

```
@Override
public void onTutSelected(String tutUrl) {
    TutViewerFragment viewer = (TutViewerFragment)
getFragmentManager()
            .findFragmentById(R.id.tutview_fragment);
    if (viewer == null || !viewer.isInLayout()) {
        Intent showContent = new Intent(getApplicationContext(),
                TutViewerActivity.class);
        showContent.setData(Uri.parse(tutUrl));
        startActivity(showContent);
    } else {
        viewer.updateUrl(tutUrl);
    }
}
```

Analysing the Output

- What happens now is simple. If a tablet accesses the app, activity A will be displayed in the same page as Activity B, that is both activities will appear in a single page (fragment)

- If a mobile phone accesses the app, Activity A will load Activity B that is, both of them will be shown in separate pages (Fragments)

Analysing the Output

- The device is a tablet and hence, both the pages(Activities) are displayed in a single fragment.

References

- Android Developers : *developer.android.com/training/basics/**fragments**/*
- MIT, USA : http://stuff.mit.edu/afs/sipb/project/android/docs/training/basics/fragments/index.html
- Google, Codes : *code.google.com/p/android/issues/detail?id=30604*
- YouTube : *www.youtube.com/watch?v=sCQ8EcM0HM0*

Taking People To The Next Level ...