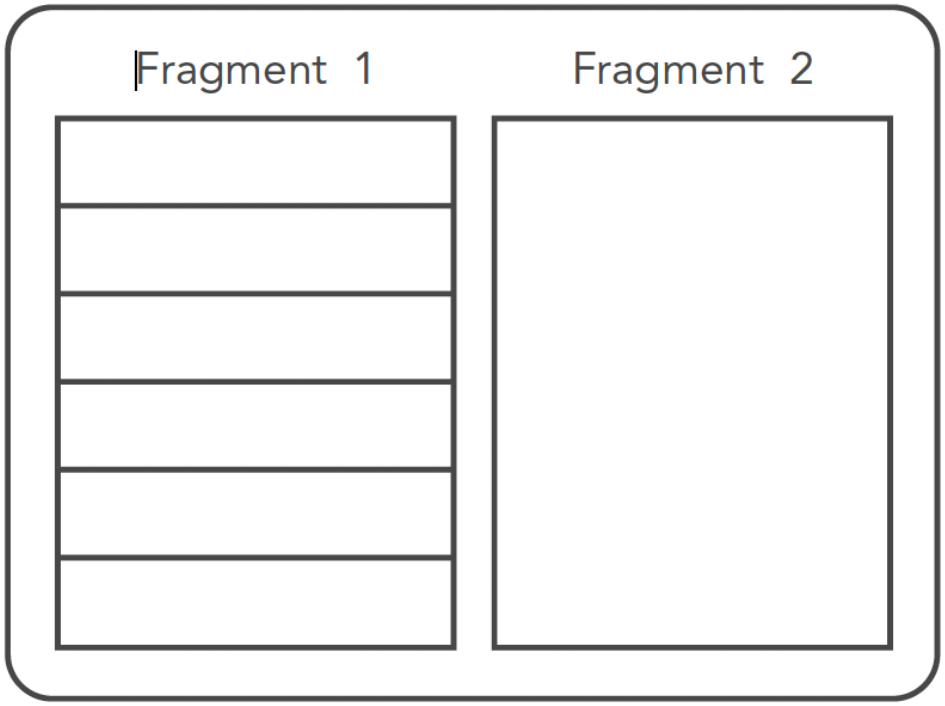
CIS 470
Mobile App Development

Lecture 5

Wenbing Zhao
Department of Electrical Engineering and Computer Science
Cleveland State University
wenbing@ieee.org

Fragments

- Activity is a container for views => typically fills the entire screen
- Fragments are introduced for large screen devices
  - One activity contains several mini-activities (fragments)



Using Fragments

- Create a new Android project and name it **Fragments**
- In the res/layout folder, add a new layout resource file and name it fragment1.xml. Populate it with the following code

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#00FF00">
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="This is fragment #1"
    android:textColor="#000000"
    android:textSize="25sp" />
</LinearLayout>
```

Using Fragments

- Also in the res/layout folder, add another new layout resource file and name it fragment2.xml
- Populate it as follows

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#FFFE00">
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="This is fragment #2"
    android:textColor="#000000"
    android:textSize="25sp" />
</LinearLayout>
```

Using Fragments

- In activity_main.xml, replace all with in the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:orientation="vertical"
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context="com.wenbing.fragments.MainActivity">
    <fragment
        android:name="com.username.fragments.Fragment1"
        android:id="@+id/fragment1"
        android:layout_weight="1"
        android:layout_width="fill_parent"
        android:layout_height="match_parent" />
    <fragment
        android:name="com. username.fragments.Fragment2"
        android:id="@+id/fragment2"
        android:layout_weight="1"
        android:layout_width="fill_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```

Using Fragments

- Under the <Your Package Name>/fragments package name, add two Java class files and name them Fragment1.java and Fragment2.java

Fagments1.java

```
package ....;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
public class Fragment1 extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater,
            ViewGroup container, Bundle savedInstanceState) {
        //---Inflate the layout for this fragment---
        return inflater.inflate(R.layout.fragment1, container, false);
    }
}
```

Using Fragments

Fagments2.java

```
package ....;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
public class Fragment2 extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater,
            ViewGroup container, Bundle savedInstanceState) {
        //---Inflate the layout for this fragment---
        return inflater.inflate(R.layout.fragment2, container, false);
    }
}
```

To draw the UI for a fragment, override the onCreateView() method. This method returns a View object

use a LayoutInflater object to inflate the UI from the specified XML file

The container argument refers to the parent ViewGroup, which is the activity in which you are trying to embed the fragment

Using Fragments

Adding Fragments Dynamics

- In the same project, modify the activity_main.xml file by commenting out the two <fragment> elements

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:orientation="vertical"
    ......
    tools:context="com. username.fragments.MainActivity">
<!--
    <fragment
        ......
    <fragment
        ......
-->
</LinearLayout>
```

Adding Fragments Dynamics

- Add the **bolded** lines in the following code to the MainActivity.java file

```
import android.app.Activity;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.util.DisplayMetrics;
public class MainActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        FragmentManager fragmentManager = getFragmentManager();
        FragmentTransaction fragmentTransaction =
             fragmentManager.beginTransaction();
        //---get the current display info---
        DisplayMetrics display = this.getResources().getDisplayMetrics();
         int width = display.widthPixels;  int height = display.heightPixels;
.....
```

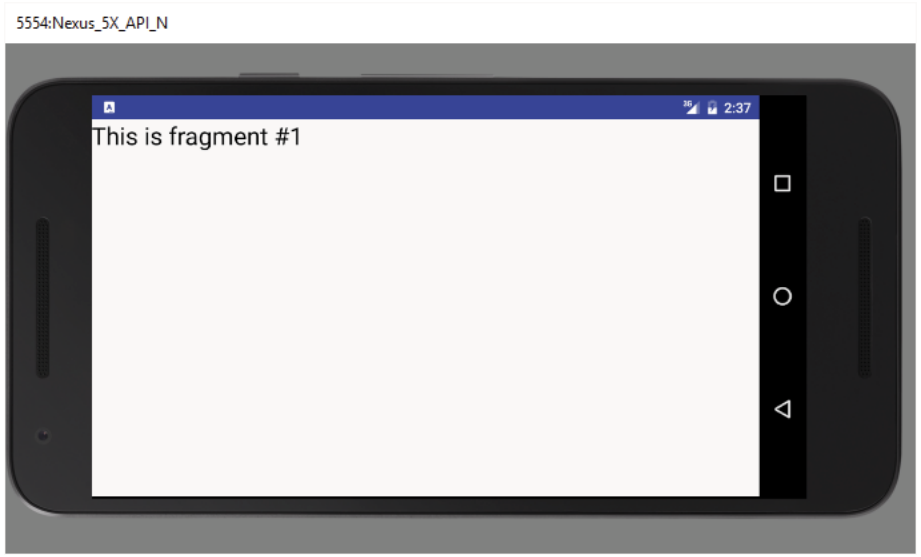
Remove:
setContentView(R.layout.*activity_main*);

Adding Fragments Dynamics

```
        if (width> height)
        {
            //---landscape mode---
            Fragment1 fragment1 = new Fragment1();
            // android.R.id.content refers to the content view of the activity
            fragmentTransaction.replace(android.R.id.content, fragment1);
        }
        else
        {
            //---portrait mode---
            Fragment2 fragment2 = new Fragment2();
            fragmentTransaction.replace(android.R.id.content, fragment2);
        }
        fragmentTransaction.commit();
    }
}
```

FragmentTransactionclass to perform fragment transactions (such as add, remove, or replace) in your activity

Adding Fragments Dynamics



Interactions between Fragments

- An activity might contain two or more fragments working together to present a coherent UI to the user
  - E.g.: the user taps on an item in that fragment, details about the selected item might be displayed in another fragment
- Continue in the same project, add bolded statements to Fragment1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#00FF00">
<TextView
    android:id="@+id/lblFragment1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="This is fragment #1"
    android:textColor="#000000"
    android:textSize="25sp" />
</LinearLayout>
```

Interactions between Fragments

- Add the following bolded lines to fragment2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    ......
<TextView
    ..... />
<Button
    android:id="@+id/btnGetText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Get text in Fragment #1"
    android:textColor="#000000"
    android:onClick="onClick" />
</LinearLayout>
```

Interactions between Fragments

- In activity_main.xml, uncomment the two fragments:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:orientation="vertical"
    .....
    tools:context="com. username.fragments.MainActivity">
    <fragment
        android:name="com.username.fragments.Fragment1"
        android:id="@+id/fragment1"
        android:layout_weight="1"
        android:layout_width="fill_parent"
        android:layout_height="match_parent" />
     <fragment
        android:name="com. username.fragments.Fragment2"
        android:id="@+id/fragment2"
        android:layout_weight="1"
        android:layout_width="fill_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```

Interactions between Fragments

- Modify the MainActivity.java file by commenting out the code added in the previous step, and **add setContentView() back**

```
public class MainActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        /*
        FragmentManager fragmentManager = getFragmentManager();
        FragmentTransaction fragmentTransaction =
             fragmentManager.beginTransaction();
        //---get the current display info---
        DisplayMetrics display = this.getResources().getDisplayMetrics();
         int width = display.widthPixels;  int height = display.heightPixels;
.....
```

Interactions between Fragments

```
        if (width> height)
        {
            //---landscape mode---
            Fragment1 fragment1 = new Fragment1();
            // android.R.id.content refers to the content view of the activity
            fragmentTransaction.replace(android.R.id.content, fragment1);
        }
        else
        {
            //---portrait mode---
            Fragment2 fragment2 = new Fragment2();
            fragmentTransaction.replace(android.R.id.content, fragment2);
        }
        fragmentTransaction.commit();
        */
    }
}
```

Interactions between Fragments

- Add the bolded statements to the Fragment2.java

Fagments2.java

```
public class Fragment2 extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater,
      ViewGroup container, Bundle savedInstanceState) {.
    }
    @Override
    public void onStart() {
        super.onStart();
        //---Button view---
        Button btnGetText = (Button)getActivity().findViewById(R.id.btnGetText);
        btnGetText.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                TextView lbl = (TextView)
                        getActivity().findViewById(R.id.lblFragment1);
                Toast.makeText(getActivity(), lbl.getText(),
                        Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

```
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
```

Challenge Task#3

- Lame version: refactor the contact app to use three fragments instead of three activities, with the top fragment being the main fragment that displays the contact list
- More interesting version (still quite artificial) – notepad app with custom keypad
  - Top fragment: display the note entered
  - Bottom fragment: display several rows of letters, numbers, and symbols (each as a button) for text input; when a user push a button, the corresponding input is appended in the top fragment