CMSC436: Programming Handheld Systems

Fall 2017
The Fragment Class
Tablet UIs

Tablets have larger displays than phones do. They can support multiple UI panes / user behaviors at the same time.

The “1 activity – 1 thing the user can do” heuristic may not make sense for larger devices.
FragmentQuoteViewerWithActivity

Application uses two Activities

   Ones shows titles of Shakespeare plays & allows user to select one title

   The other shows a quote from the selected play
The Tragedy of Hamlet, Prince of Denmark

King Lear

Julius Caesar

Now cracks a noble heart. Good-night, sweet prince; And flights of angels sing thee to thy rest.
FragmentQuoteViewerWithActivity UI

This layout is reasonable on a phone
But inefficient on a larger device
The Tragedy of Hamlet, Prince of Denmark

King Lear

Julius Caesar
Now cracks a noble heart. Good-night, sweet prince; And flights of angels sing thee to thy rest.
Better Layout

Use two cooperating layout units on one screen
The Tragedy of Hamlet, Prince of Denmark

Now cracks a noble heart. Good-night, sweet prince; And flights of angels sing thee to thy rest.

King Lear

Julius Caesar
The Fragment Class

Represents a behavior / portion of UI within an Activity

Multiple Fragments can be embedded in an Activity to create a multi-pane UI

A single Fragment can be reused across multiple Activities
Fragment Lifecycle

Fragment lifecycle is coordinated with the lifecycle of its containing/hosting Activity

Fragments have their own lifecycles and receive their own callbacks
Fragment Lifecycle States

Resumed

  Fragment is visible in the hosting Activity

Paused

  Another Activity is in the foreground and has focus, this Fragment’s hosting Activity is still visible

Stopped

  The Fragment is not visible
Lifecycle Callback Methods
onAttach()

Activity is created
Fragment is first attached to its Activity
onCreate()

Initialize the Fragment
onCreateView()

Fragment sets up & returns its user interface View
onActivityCreated()

Containing Activity has completed onCreate() and the Fragment has been installed
Can now access hosting Activity
onStart()

Activity is started
Hosting Activity about
to become visible
onResume()

Activity is resumed
Hosting Activity is about to become visible and ready for user interaction
onPause()

Activity is paused
Hosting Activity is visible, but does not have focus
onStop()

Activity is stopped
Hosting Activity is no longer visible
onDestroyView()

Activity is destroyed
View previously created in `onCreateView()` has been detached from the Activity
Clean up view resources
onDestroy()

Fragment is no longer in use
Clean up Fragment resources
onDetach()

Fragment no longer attached to its activity
Null out references to hosting Activity
Adding Fragments to Activities

Two general ways to add Fragments to an Activity’s layout

- Declare it statically in the Activity’s layout file
- Add it programmatically using the fragmentManager
Fragment Layout

Layout can be inflated/implemented in `onCreateView()`

`onCreateView()` must return the View at the root of the Fragment’s layout

This View is added to the containing Activity
FragmentStaticLayout

Display titles and quotes side-by-side in two Fragments.

Fragments are statically added based on a layout file.
<table>
<thead>
<tr>
<th>The Tragedy of Hamlet, Prince of Denmark</th>
<th>Please select a Title.</th>
</tr>
</thead>
<tbody>
<tr>
<td>King Lear</td>
<td></td>
</tr>
<tr>
<td>Julius Caesar</td>
<td></td>
</tr>
</tbody>
</table>
public class QuoteViewerActivity extends Activity implements ListSelectionListener {
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.quote_activity);
        ...
    }
    ...
}
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:baselineAligned="false"
    android:orientation="horizontal"
    android:padding="@dimen/activity_margin">
    <fragment
        android:id="@+id/titles"
        class="course.examples.fragments.staticlayout.TitlesFragment"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="1" />
    <fragment
        android:id="@+id/details"
        class="course.examples.fragments.staticlayout_QUOTESFragment"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="2" />
</LinearLayout>
Adding Fragments Programmatically

While an Activity’s running you can add Fragments to its layout

Four-step process

1. Get reference to the FragmentManager
2. Begin a FragmentTransaction
3. Add the Fragment
4. Commit the FragmentTransaction
FragmentProgrammaticLayout

Display titles and quotes side-by-side in two Fragments
Layout file reserves space for Fragments
Fragments are programmatically added to layout at runtime
The Tragedy of Hamlet, Prince of Denmark

Now cracks a noble heart. Good-night, sweet prince; And flights of angels sing thee to thy rest.

King Lear

Julius Caesar
protected void onCreate(Bundle savedInstanceState) {

    // Get a reference to the FragmentManager
    FragmentManager fragmentManager = getFragmentManager();
    if (null == fragmentManager.findFragmentById(R.id.title_frame)) {
        // Begin a new FragmentTransaction
        FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
        // Add the TitleFragment
        fragmentTransaction.add(R.id.title_frame, new TitlesFragment());
        // Add the QuoteFragment
        mQuoteFragment = new QuotesFragment();
        fragmentTransaction.add(R.id.quote_frame, mQuoteFragment);
        // Commit the FragmentTransaction
        fragmentTransaction.commit();
    } else {
        mQuoteFragment = (QuotesFragment) fragmentManager.findFragmentById(R.id.quote_frame);
    }
}
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/activityFrame"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:baselineAligned="false"
    android:orientation="horizontal"
    android:padding="@dimen/activity_margin">
    <FrameLayout
        android:id="@+id/title_frame"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1" />
    <FrameLayout
        android:id="@+id/quote_frame"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2" />
</LinearLayout>
Dynamic Layout

Fragment transactions allow you to dynamically change your app’s user interface.

Can make the interface more fluid & take better advantage of available screen space.
FragmentDynamicLayout

Starts with a single Fragment
Changes to two-Fragment layout when user selects a title
FragmentDynamicLayout

The Tragedy of Hamlet, Prince of Denmark
King Lear
Julius Caesar
The Tragedy of Hamlet, Prince of Denmark

King Lear

Julius Caesar

Now cracks a noble heart. Good-night, sweet prince; And flights of angels sing thee to thy rest.
protected void onCreate(Bundle savedInstanceState) {

    // Get a reference to the FragmentManager
    mFragmentManager = getFragmentManager();
    mQuoteFragment = (QuotesFragment) mFragmentManager.findFragmentById(R.id.quote_fragment_container);
    mTitleFragment = (TitlesFragment) mFragmentManager.findFragmentById(R.id.title_fragment_container);

    if (null == mFragmentManager.findFragmentById(R.id.title_fragment_container)) {
        // Start a new FragmentTransaction
        FragmentTransaction fragmentTransaction = mFragmentManager.beginTransaction();
        mTitleFragment = new TitlesFragment();
        // Add the TitleFragment to the layout
        fragmentTransaction.add(R.id.title_fragment_container, mTitleFragment);
        // Commit the FragmentTransaction
        fragmentTransaction.commit();
    }
}
public void onListSelection(int index) {

    // If the QuoteFragment has not been created, create and add it now
    if (null == mFragmentManager.findFragmentById(R.id.quote_fragment_container)) {

        // Start a new FragmentTransaction
        FragmentTransaction fragmentTransaction = mFragmentManager.beginTransaction();
            .beginTransaction();

        ...
    }
mQuoteFragment = new QuotesFragment();
// Add the QuoteFragment to the layout
fragmentTransaction.add(R.id.quote_fragment_container,
    mQuoteFragment);
// Add this FragmentTransaction to the backstack
fragmentTransaction.addToBackStack(null);
// Commit the FragmentTransaction
fragmentTransaction.commit();
// Force Android to execute the committed FragmentTransaction
mFragmentManager.executePendingTransactions();
}
// Tell the QuoteFragment to show the quote string at position index
mQuoteFragment.showQuoteAtIndex(index);
Configuration Changes

If you call `setRetainInstance(true)` on Fragment, Android won’t destroy the Fragment on configuration changes.
Configuration Changes

Results in some changes to lifecycle callback sequence
onDestroy() will not be called
onCreate() will not be called
FragmentStaticConfigLayout

Essentially the same as FragmentStaticLayout

Focus here is on how Fragments are saved and restored on configuration changes
FragmentStaticConfigLayout

In landscape mode
Both Fragments use a large font
TitleFragment takes more horizontal space & allows long titles to span multiple lines
In portrait mode
Both Fragments use a smaller font
TitleFragment will use less space and will ellipsize long titles, limiting them to a single line
Now cracks a noble heart. Good-night, sweet prince; And flights of angels sing thee to thy rest.
public class QuotesFragment extends Fragment {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Retain this Fragment across Activity reconfigurations
        setRetainInstance(true);
    }

    // Called to create the content view for this Fragment
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        // Inflate the layout defined in quote_fragment.xml. The last parameter is false because the returned
        // view does not need to be attached to the container ViewGroup
        return inflater.inflate(R.layout.quote_fragment, container, false);
    }
}
// Set up some information about the mQuoteView TextView

@Override
public void onActivityCreated(Bundle savedInstanceState) {
    Log.i(TAG, getClass().getSimpleName() + ":onActivityCreated()");
    super.onActivityCreated(savedInstanceState);

    mQuoteView = getActivity().findViewById(R.id.quoteView);
    mQuoteArrayLen = QuoteViewerActivity.mQuoteArray.length;

    showQuoteAtIndex(mCurrIdx);
}

Next

User Interface classes