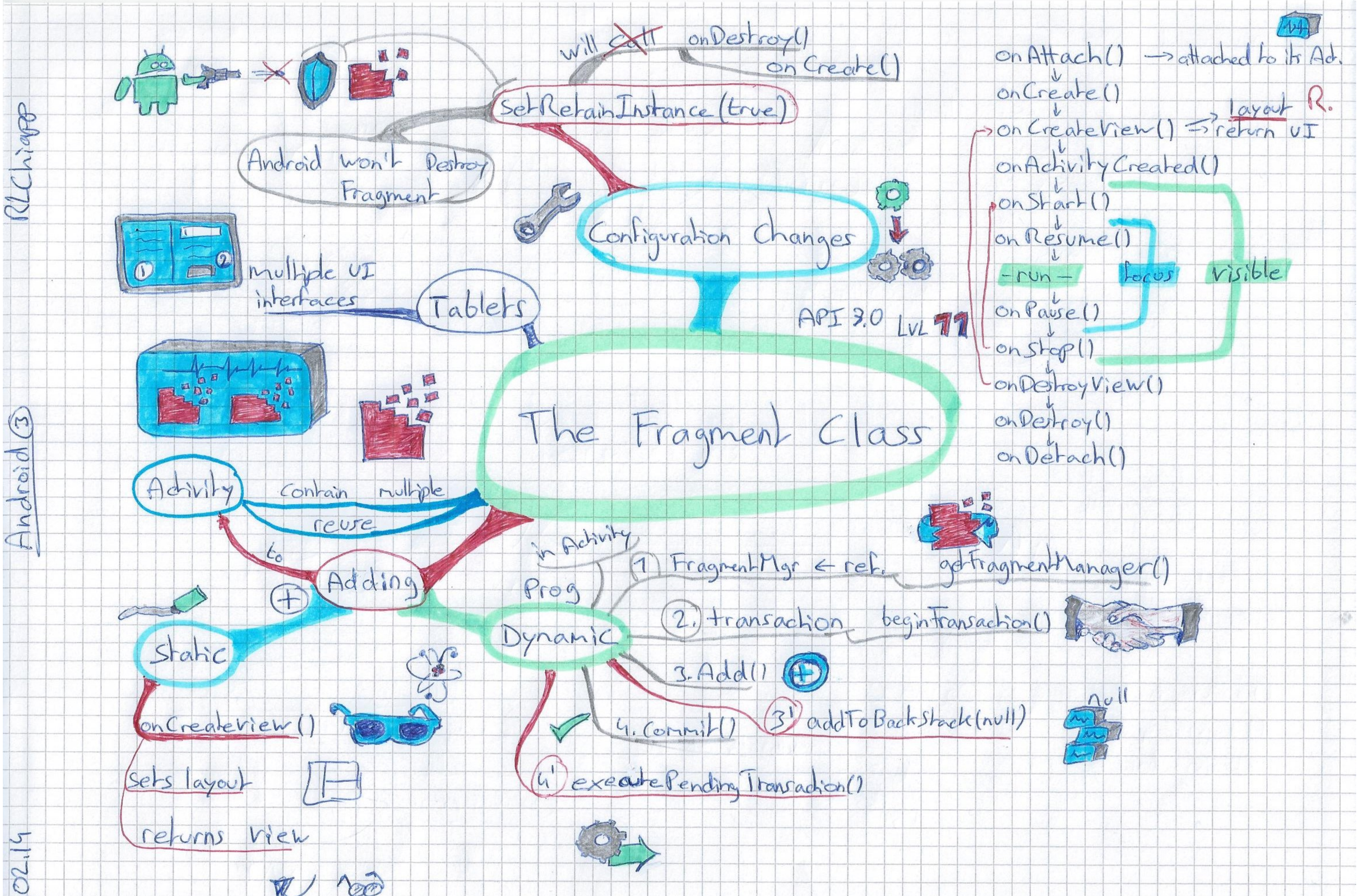


Romain Chiappinelli

Programming Handheld Systems

Adam Porter

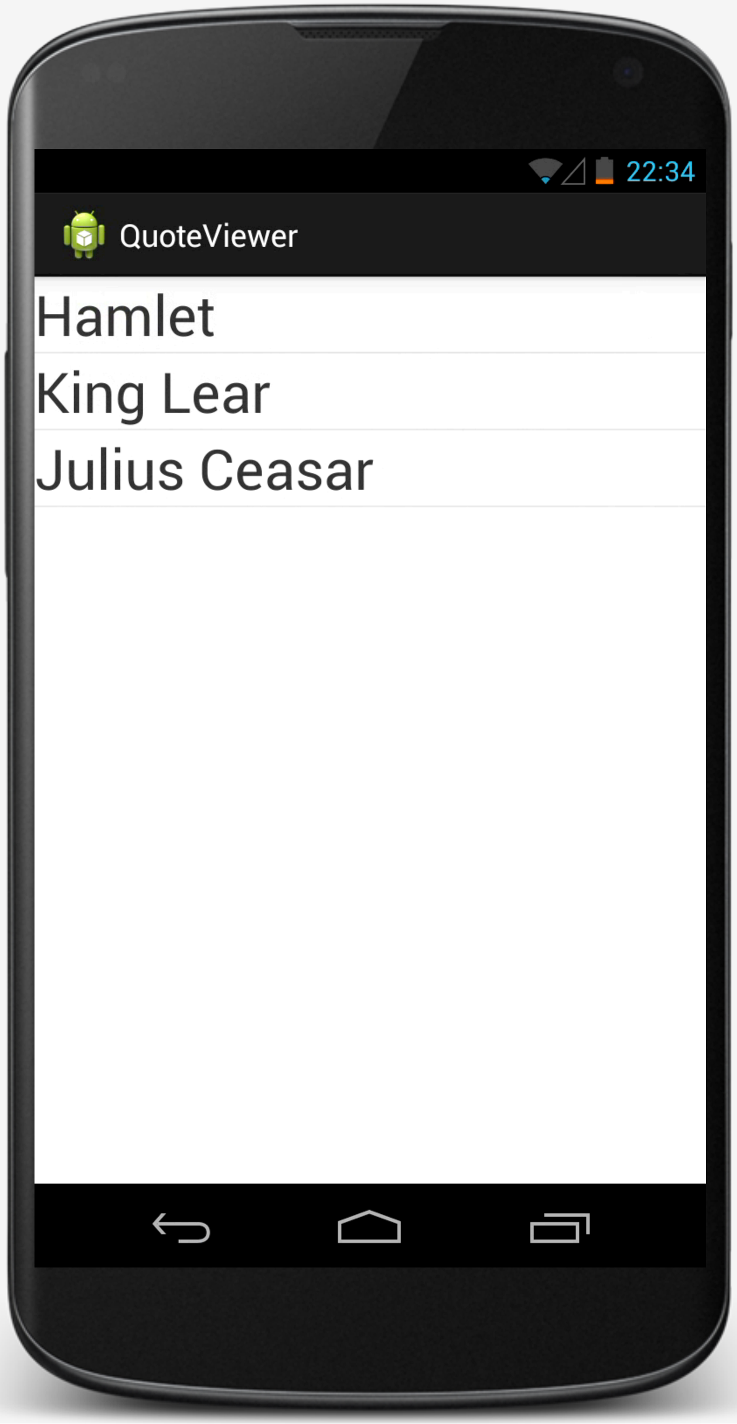
The Fragment Class

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 work for larger devices

QuoteViewer

- Application uses two Activities
  - Ones shows titles of Shakespeare plays & allows user to select one title
  - The other shows a quote from selected play



Hamlet

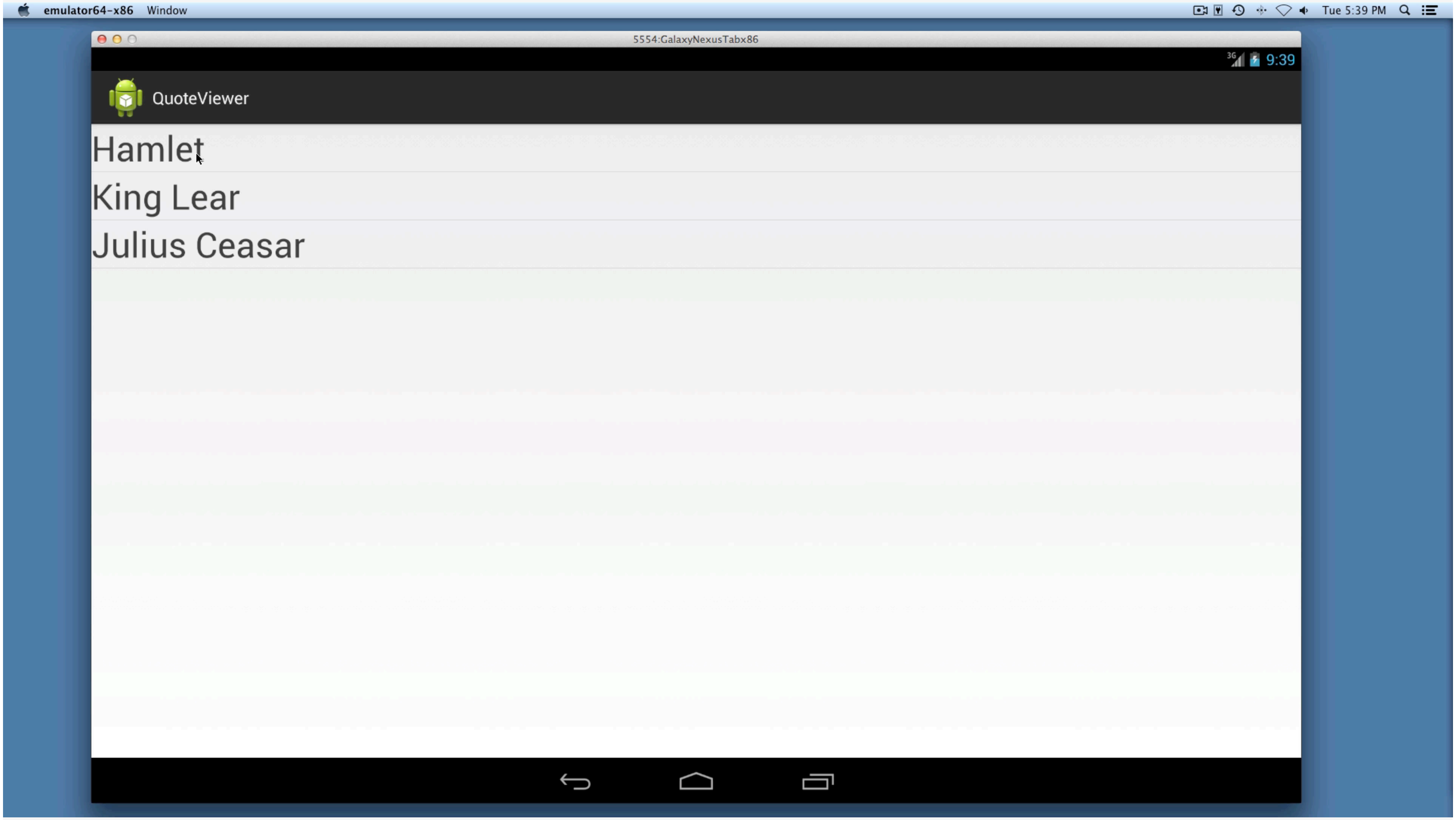
King Lear

Julius Ceasar

The QuoteViewer UI

This layout is reasonable on a phone

But inefficient on a larger device



QuoteViewer

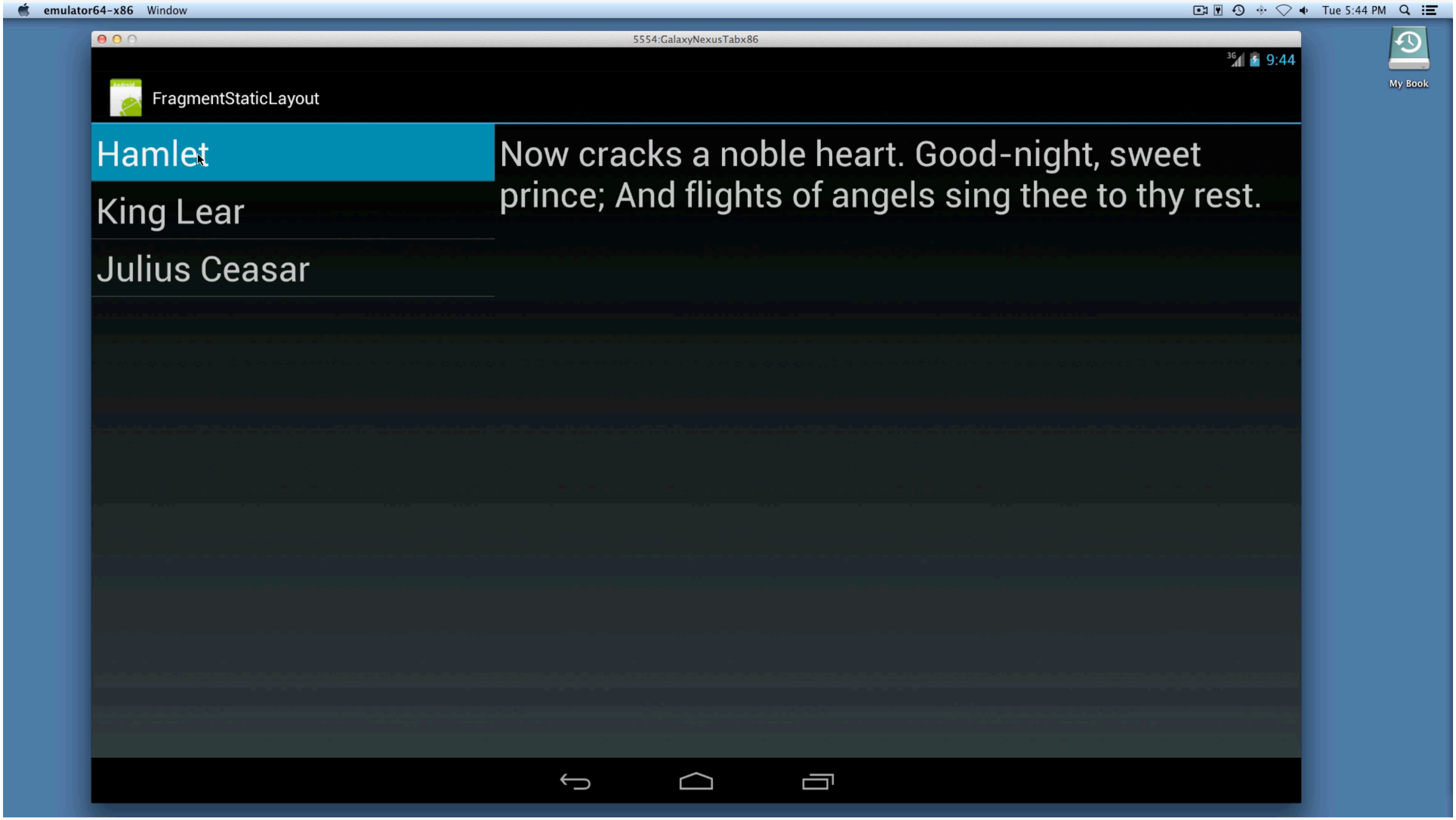
Hamlet

King Lear

Julius Ceasar

Better Layout

Use two cooperating layout units on one screen



FragmentStaticLayout

Hamlet

King Lear

Julius Ceasar

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

Fragment

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 Activity

Fragments have their own lifecycles and receive their own callbacks

Fragment Lifecycle States

Resumed

Fragment is visible in the running activity

Paused

Another activity is in the foreground and has focus, containing activity is visible

Stopped

The fragment is not visible

Lifecycle Callback Methods

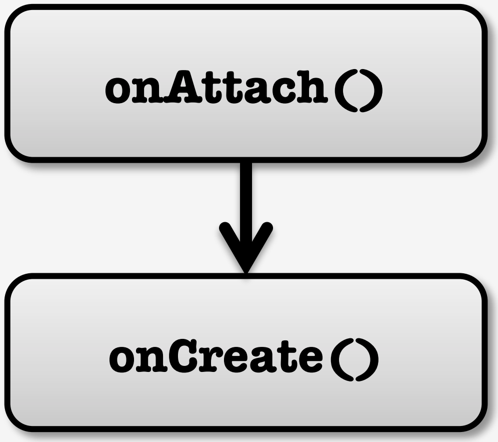
Activity Created: onAttach()

Fragment is first attached to its activity

onAttach()

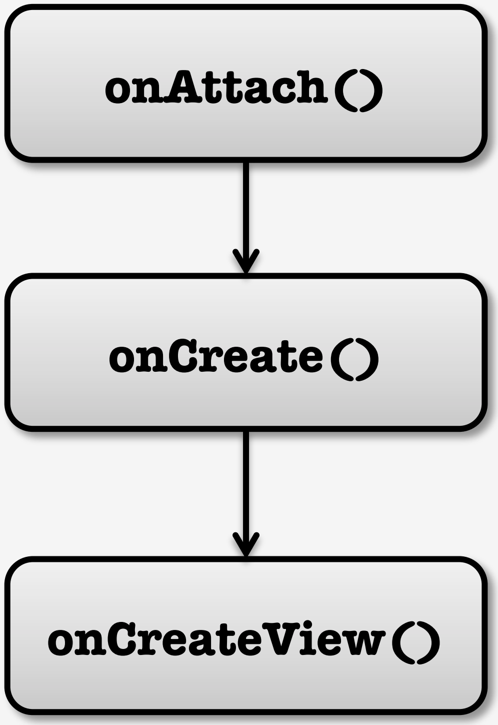
onCreate()

Initialize the fragment



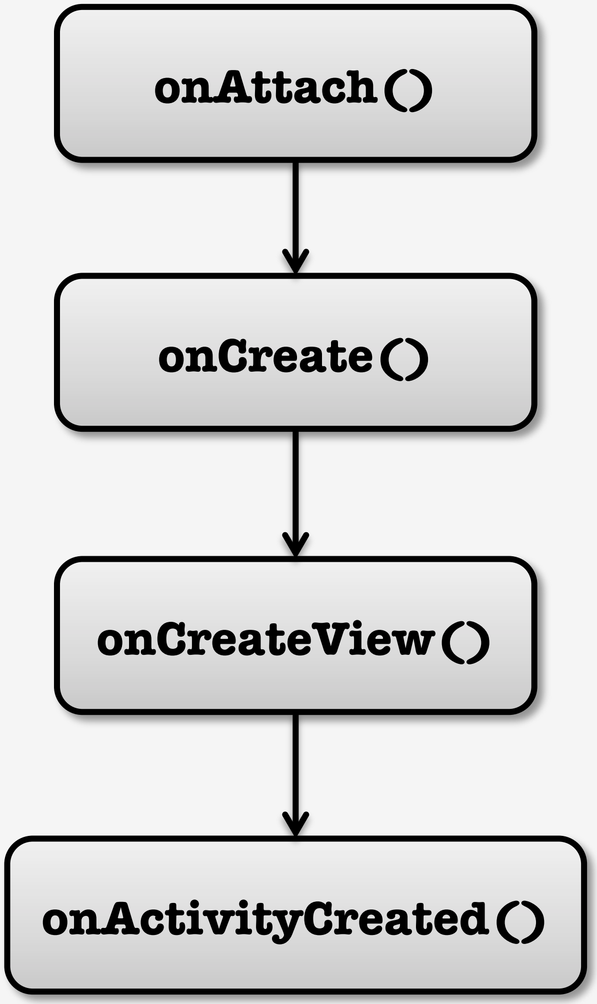
onCreateView()

Fragment sets up & returns its user interface



onActivityCreated()

Containing activity has completed onCreate() and the fragment has been installed



onAttach()

onCreate()

onCreateView()

onActivityCreated()

Activity Started: onStart()

Hosting Activity about to become visible

onStart ()

Activity Resumed: onResume()

Hosting Activity is about to become visible and ready for user interaction

onResume()

Activity Paused: onPause()

Hosting Activity is visible, but does not have focus

onPause()

Activity Stopped: onStop()

Hosting Activity is no longer visible

onStop ()

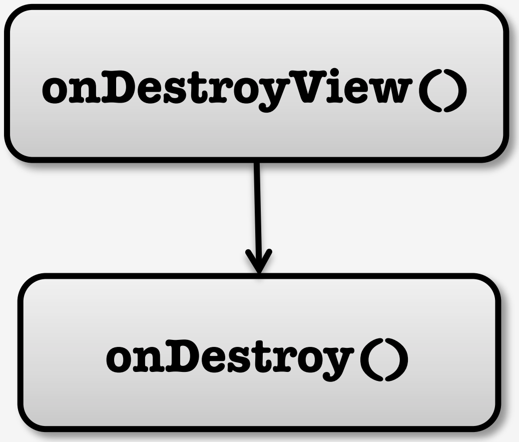
Activity Destroyed: onDestroyView()

- View previously created by onCreateView() has been detached from the Activity
- Typical Actions
  - Clean up view resources

onDestroyView()

onDestroy()

- Fragment is no longer in use
- Typical Actions
  - Clean up Fragment resources

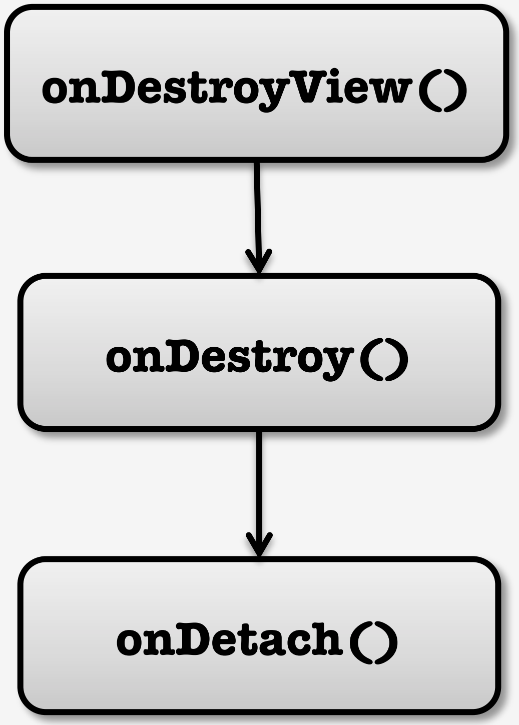


onDestroyView()

onDestroy()

onDetach()

- Fragment no longer attached to its activity
- Typical Actions
  - 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

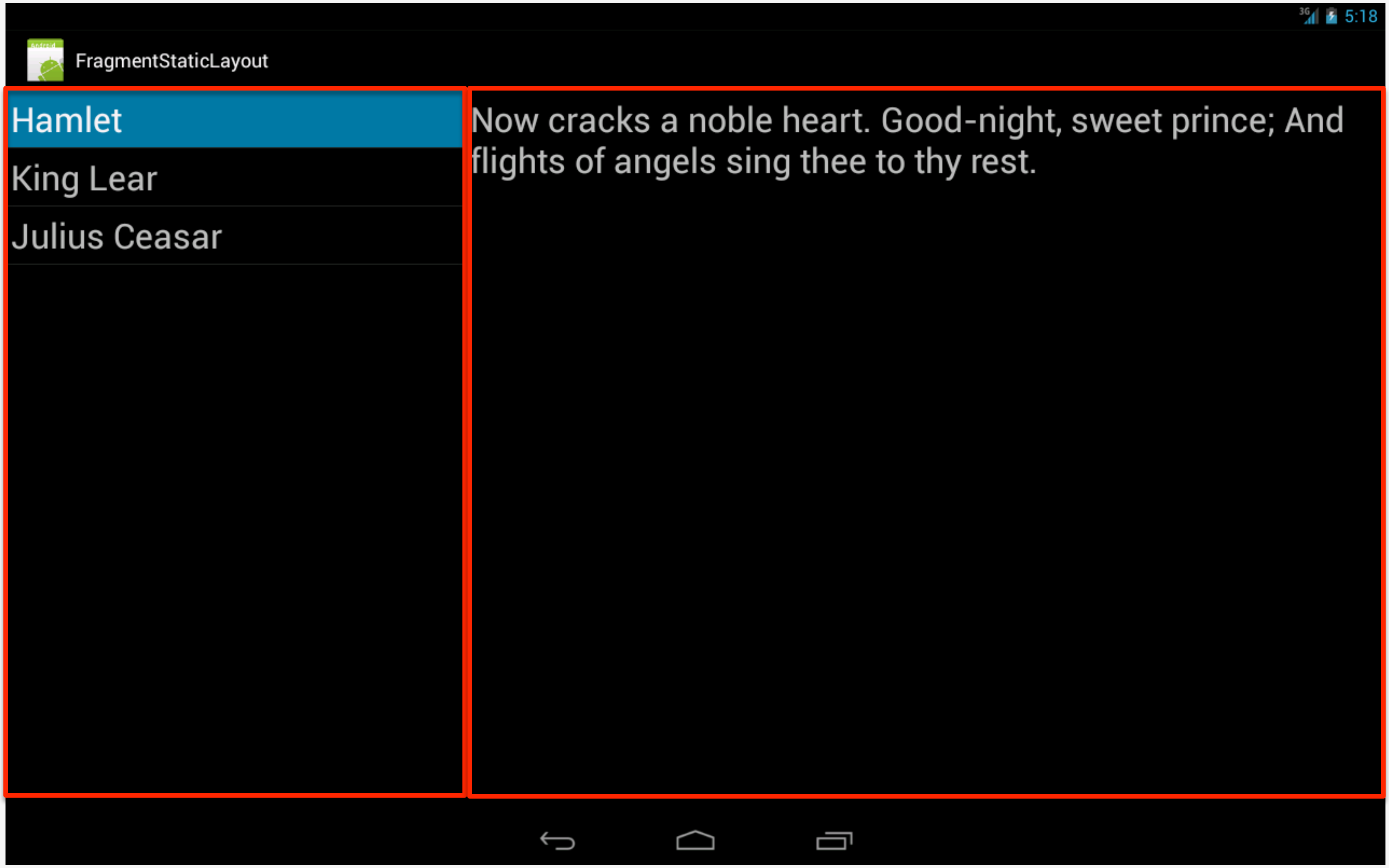
Add it programmatically using the fragmentManager

Fragment Layout

Layout can be inflated/ implemented in onCreateView()

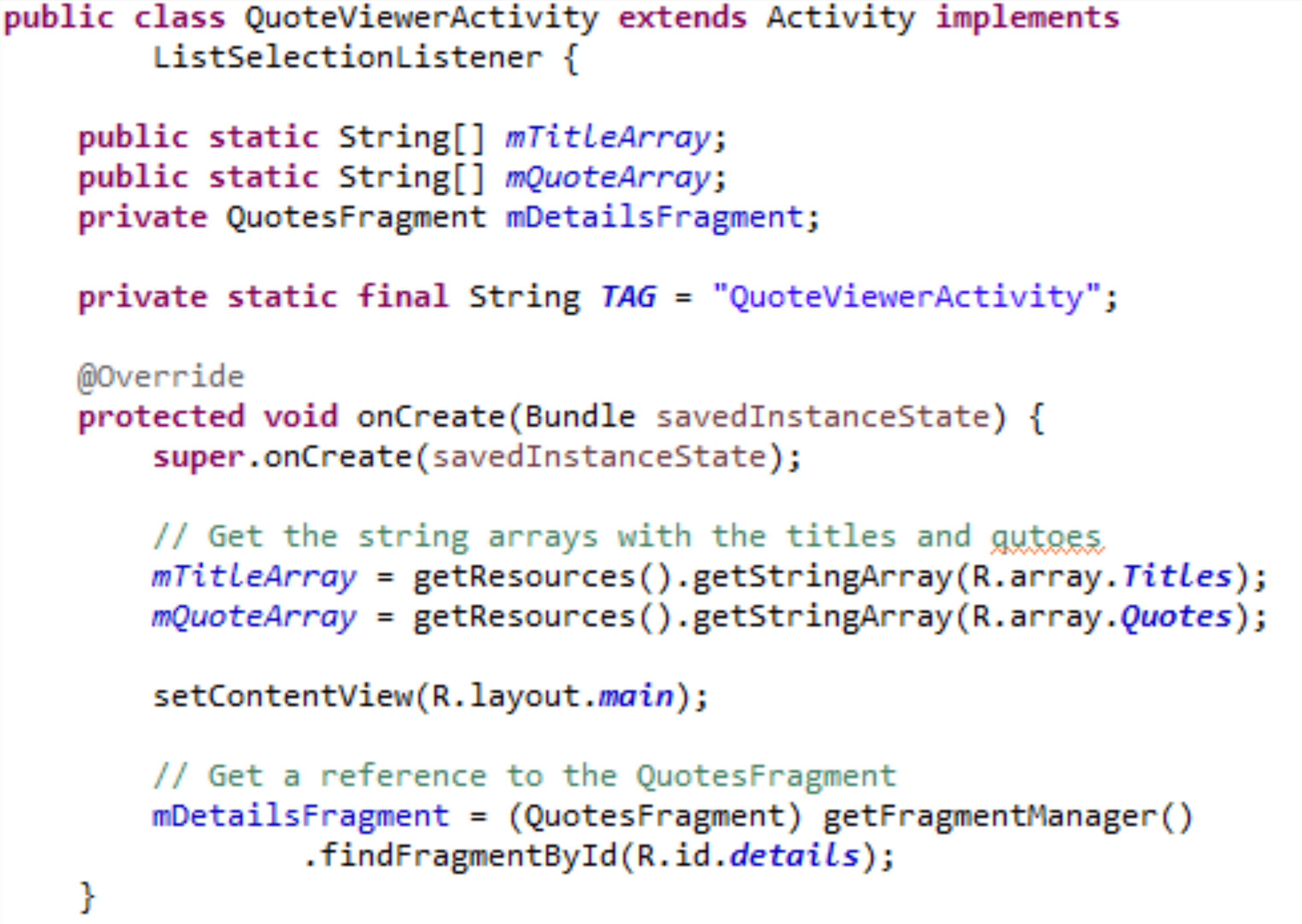
onCreateView() must return the View at the root of the Fragment's layout

This View is added to the containing Activity



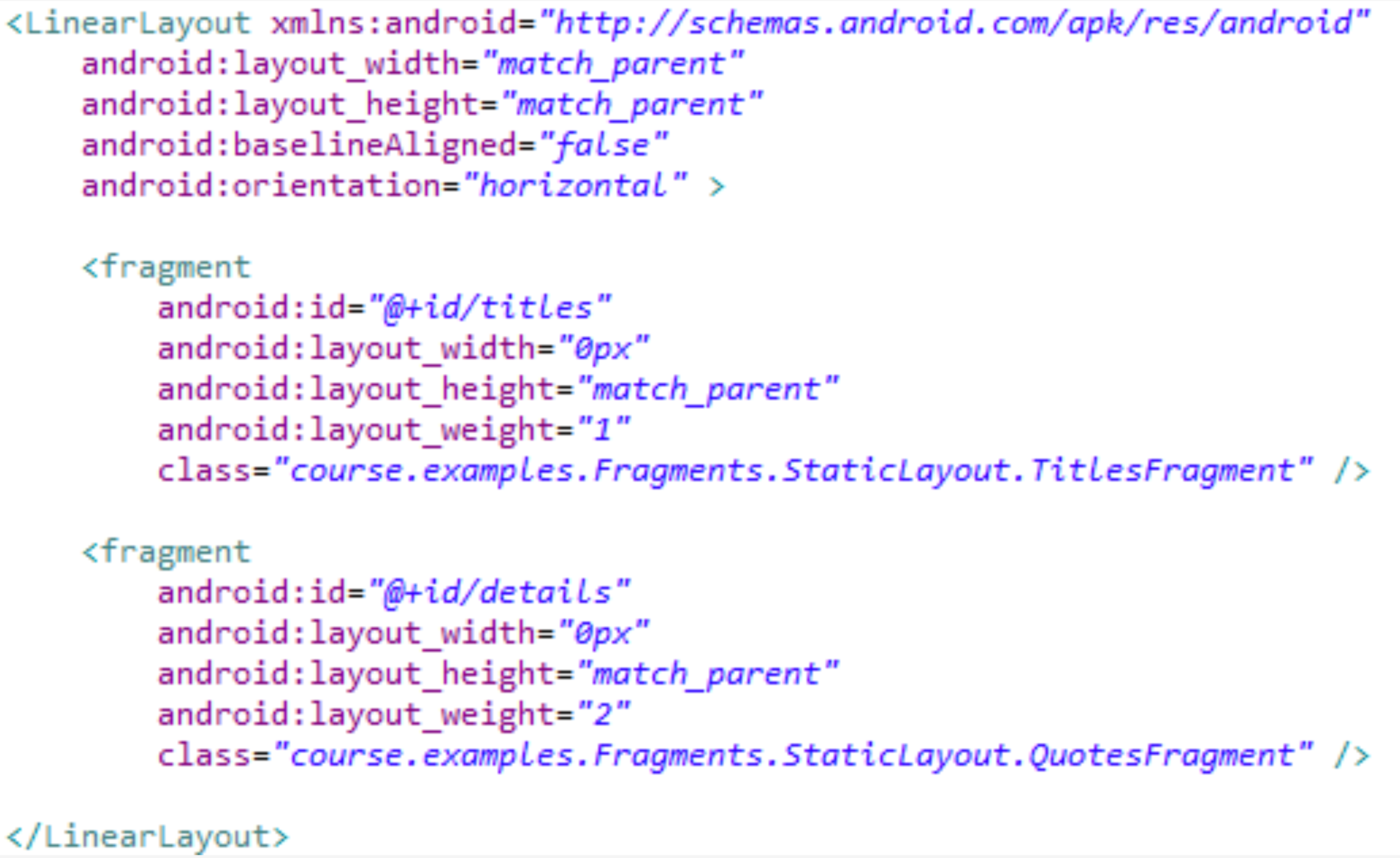
Now cracks a noble heart. Good-night, sweet prince; And flights of angels sing thee to thy rest.

FragmentStaticLayout



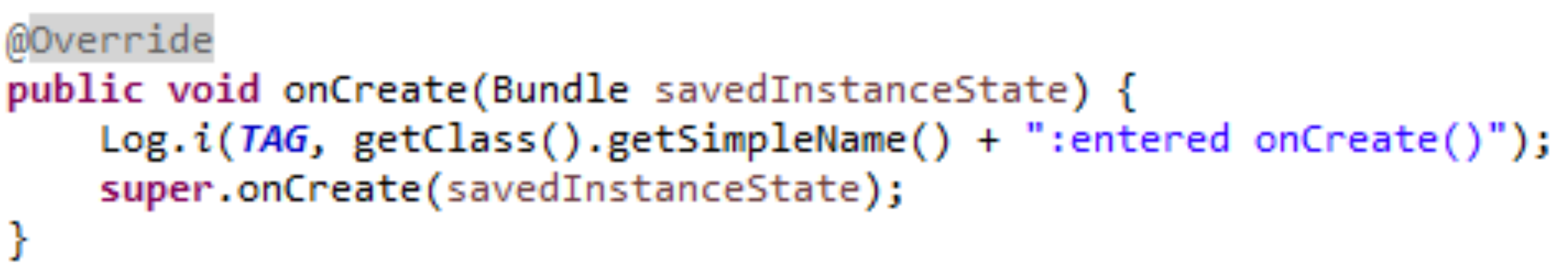
```java
public class QuoteViewerActivity extends Activity implements
        ListSelectionListener {

    public static String[] mTitleArray;
    public static String[] mQuoteArray;
    private QuotesFragment mDetailsFragment;

    private static final String TAG = "QuoteViewerActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Get the string arrays with the titles and qutoes
        mTitleArray = getResources().getStringArray(R.array.Titles);
        mQuoteArray = getResources().getStringArray(R.array.Quotes);

        setContentView(R.layout.main);

        // Get a reference to the QuotesFragment
        mDetailsFragment = (QuotesFragment) getFragmentManager()
                .findFragmentById(R.id.details);
    }
```

FragmentStaticLayout

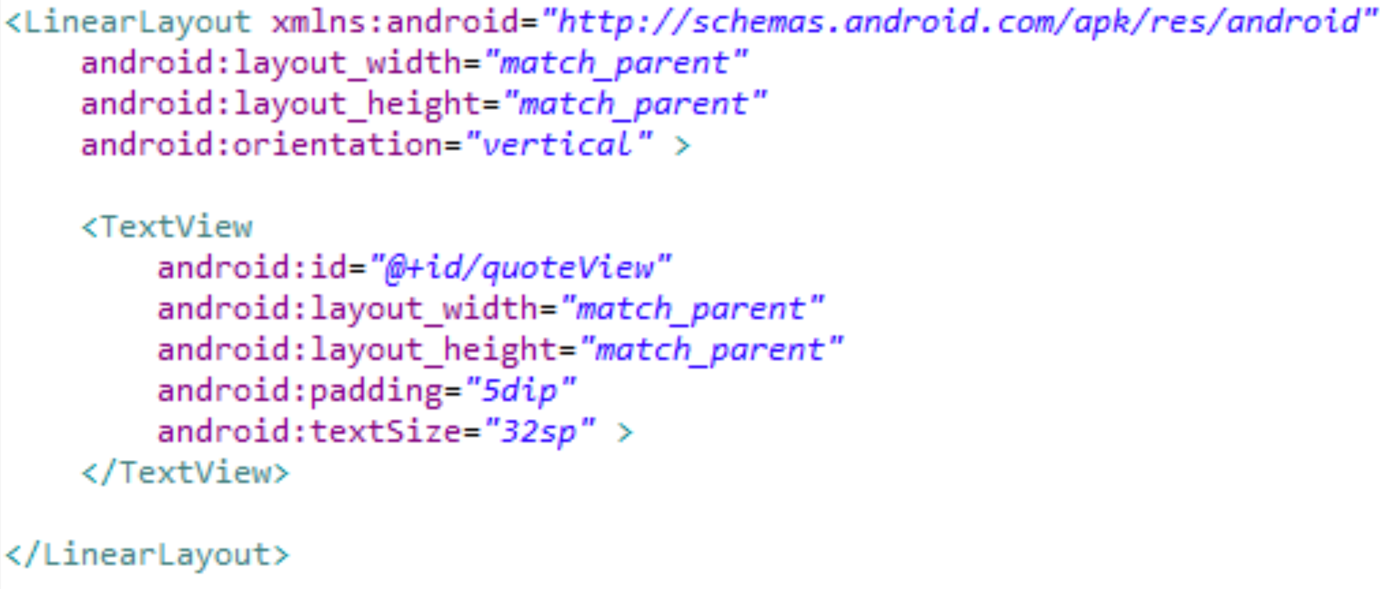


```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:baselineAligned="false"
    android:orientation="horizontal" >

    <fragment
        android:id="@+id/titles"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="1"
        class="course.examples.Fragments.StaticLayout.TitlesFragment" />

    <fragment
        android:id="@+id/details"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="2"
        class="course.examples.Fragments.StaticLayout.QuotesFragment" />

</LinearLayout>
```

FragmentStaticLayout



```java
@Override
public void onCreate(Bundle savedInstanceState) {
    Log.i(TAG, getClass().getSimpleName() + ":entered onCreate()");
    super.onCreate(savedInstanceState);
}
```



```xml
<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/quoteView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="5dip"
        android:textSize="32sp" >
    </TextView>

</LinearLayout>
```

Adding Fragments Dynamically

While an Activity's running you can add a Fragment to it's layout

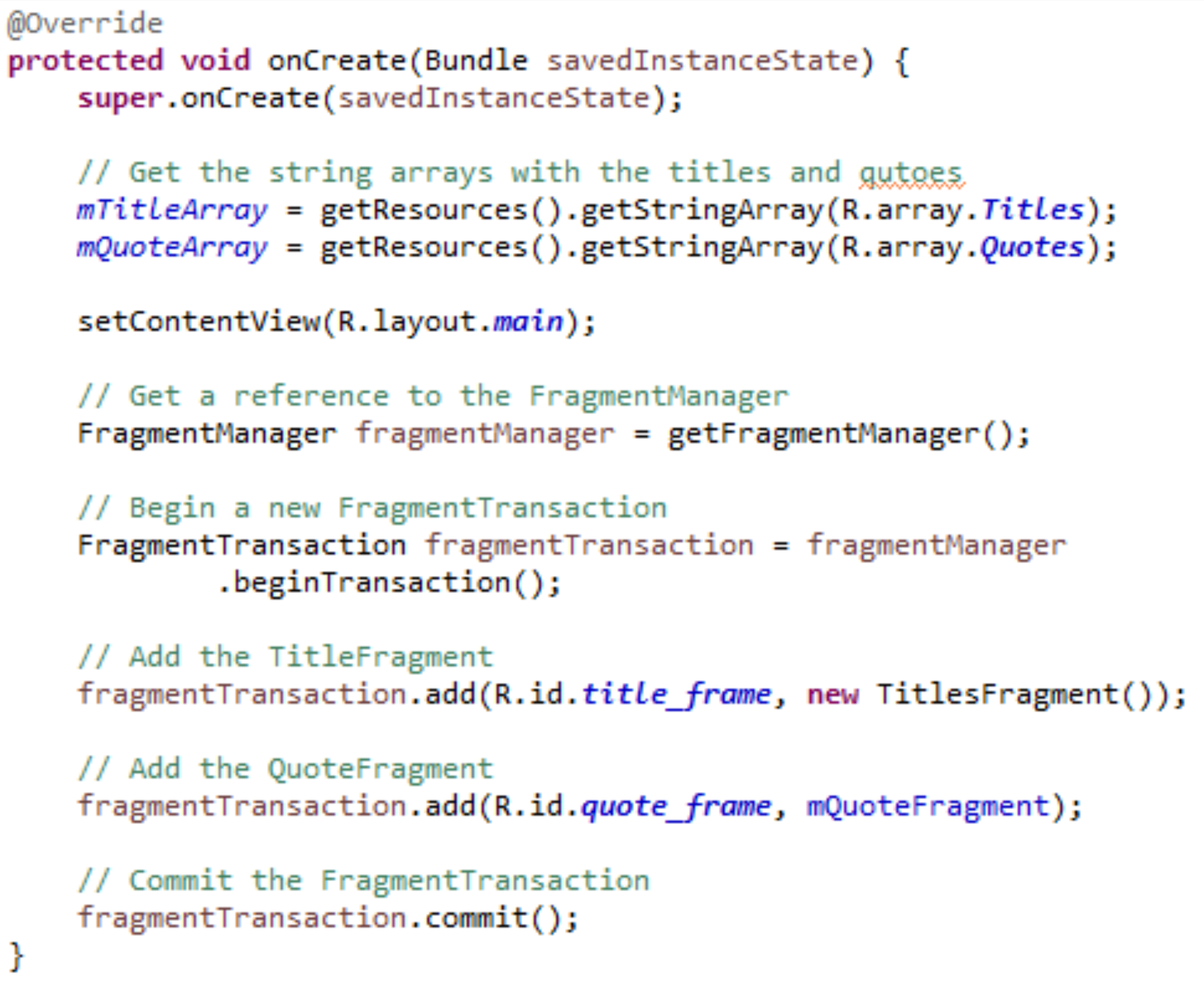
Get reference to the FragmentManager

Begin a FragmentTransaction

Add the Fragment

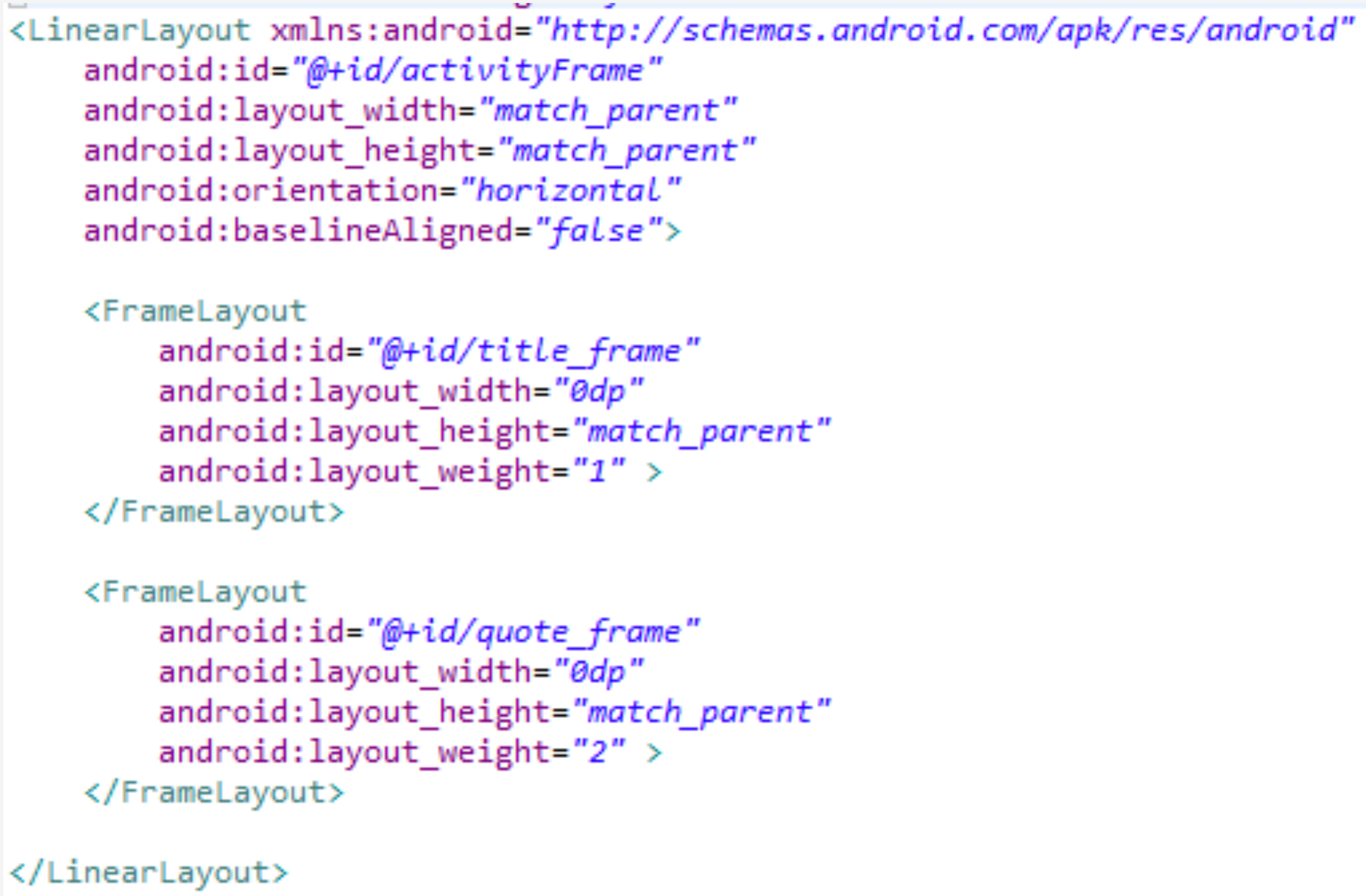
Commit the FragmentTransaction

FragmentProgrammaticLayout

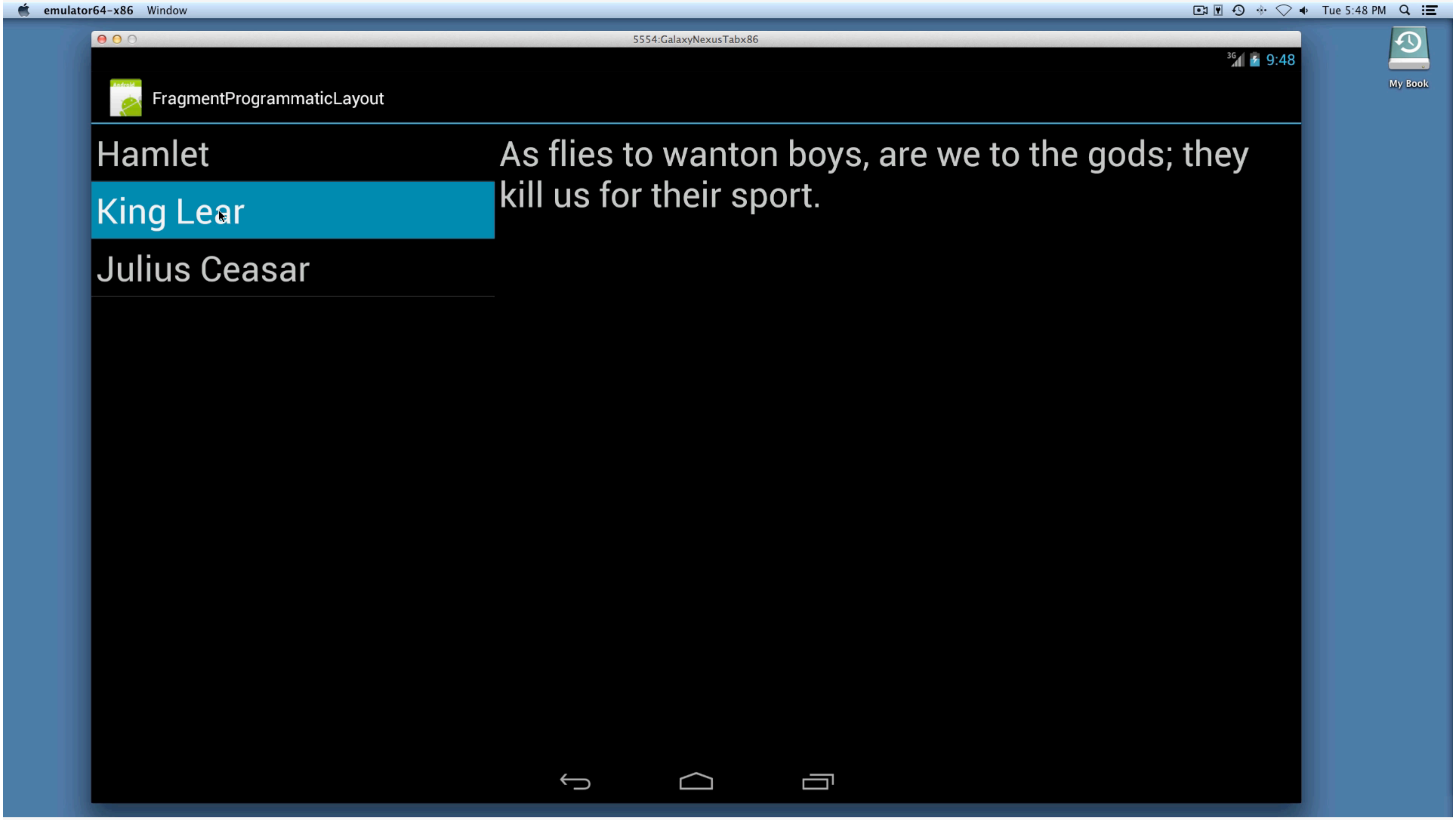


```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Get the string arrays with the titles and qutoes
    mTitleArray = getResources().getStringArray(R.array.Titles);
    mQuoteArray = getResources().getStringArray(R.array.Quotes);

    setContentView(R.layout.main);

    // Get a reference to the FragmentManager
    FragmentManager fragmentManager = getFragmentManager();

    // Begin a new FragmentTransaction
    FragmentTransaction fragmentTransaction = fragmentManager
            .beginTransaction();

    // Add the TitleFragment
    fragmentTransaction.add(R.id.title_frame, new TitlesFragment());

    // Add the QuoteFragment
    fragmentTransaction.add(R.id.quote_frame, mQuoteFragment);

    // Commit the FragmentTransaction
    fragmentTransaction.commit();
}
```

FragmentProgrammaticLayout



```
<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:orientation="horizontal"
    android:baselineAligned="false">

    <FrameLayout
        android:id="@+id/title_frame"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1" >
    </FrameLayout>

    <FrameLayout
        android:id="@+id/quote_frame"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2" >
    </FrameLayout>

</LinearLayout>
```



FragmentProgrammaticLayout

Hamlet

King Lear

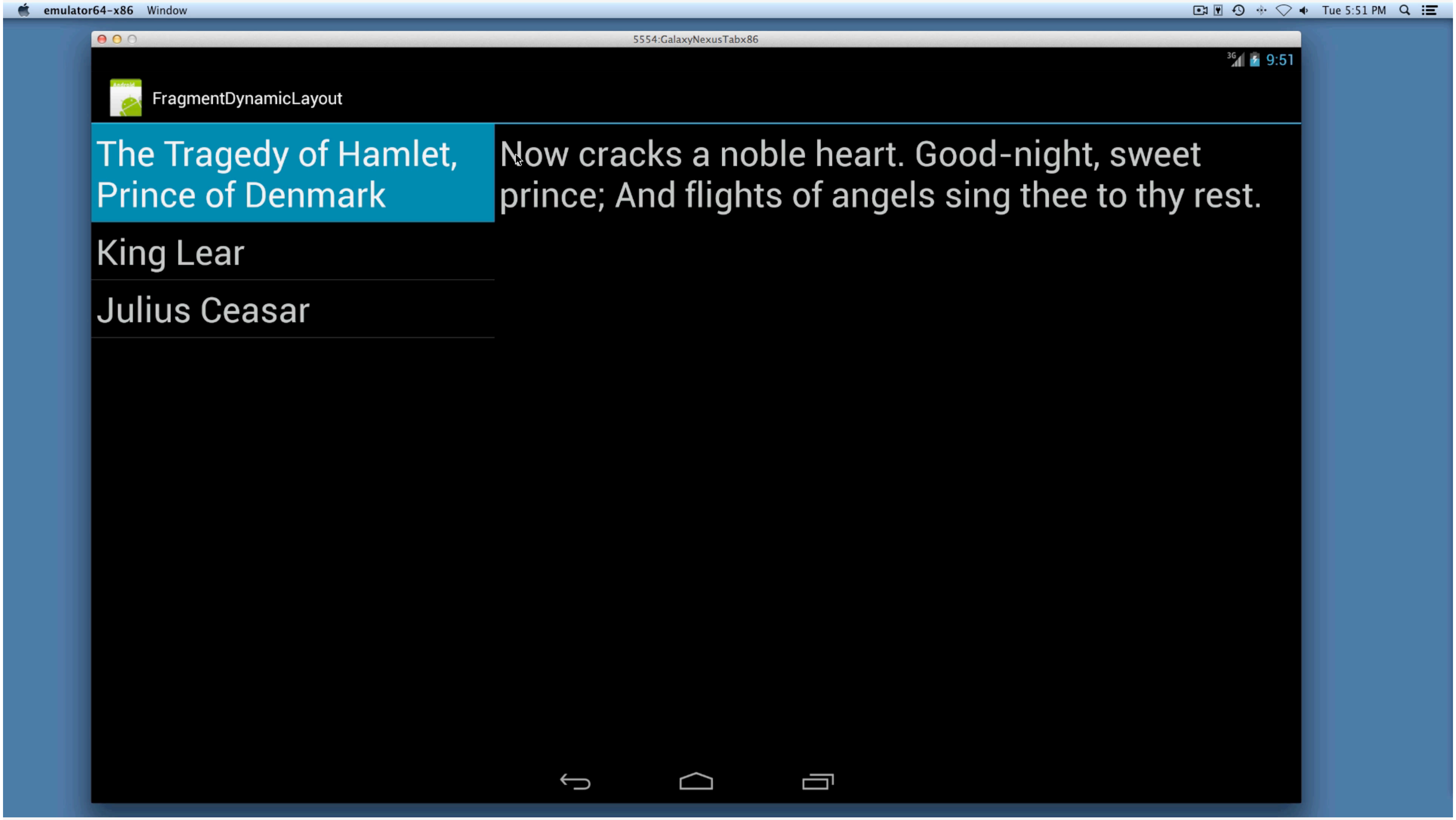
Julius Ceasar

As flies to wanton boys, are we to the gods; they kill us for their sport.

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



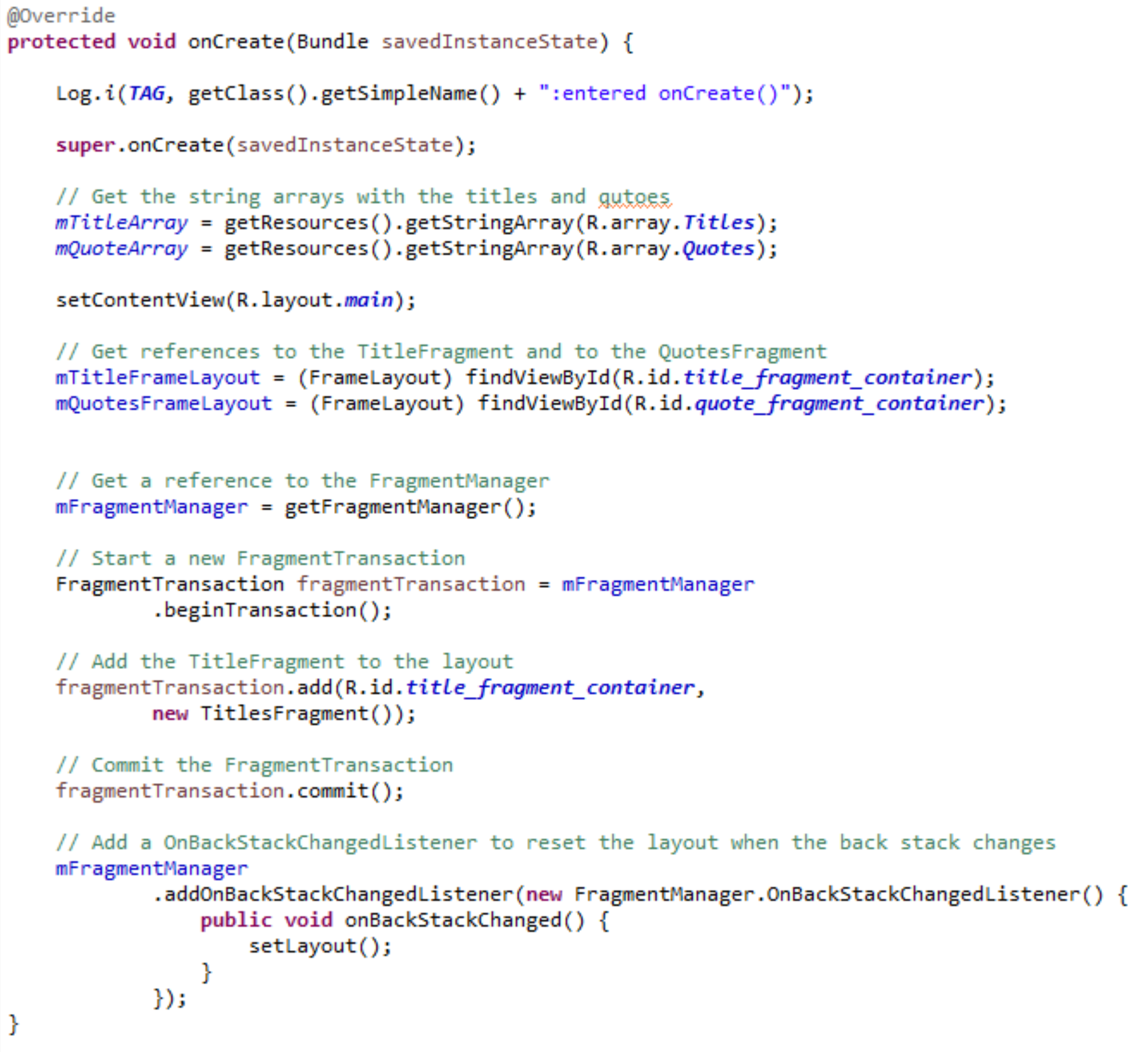
The Tragedy of Hamlet, Prince of Denmark

King Lear

Julius Ceasar

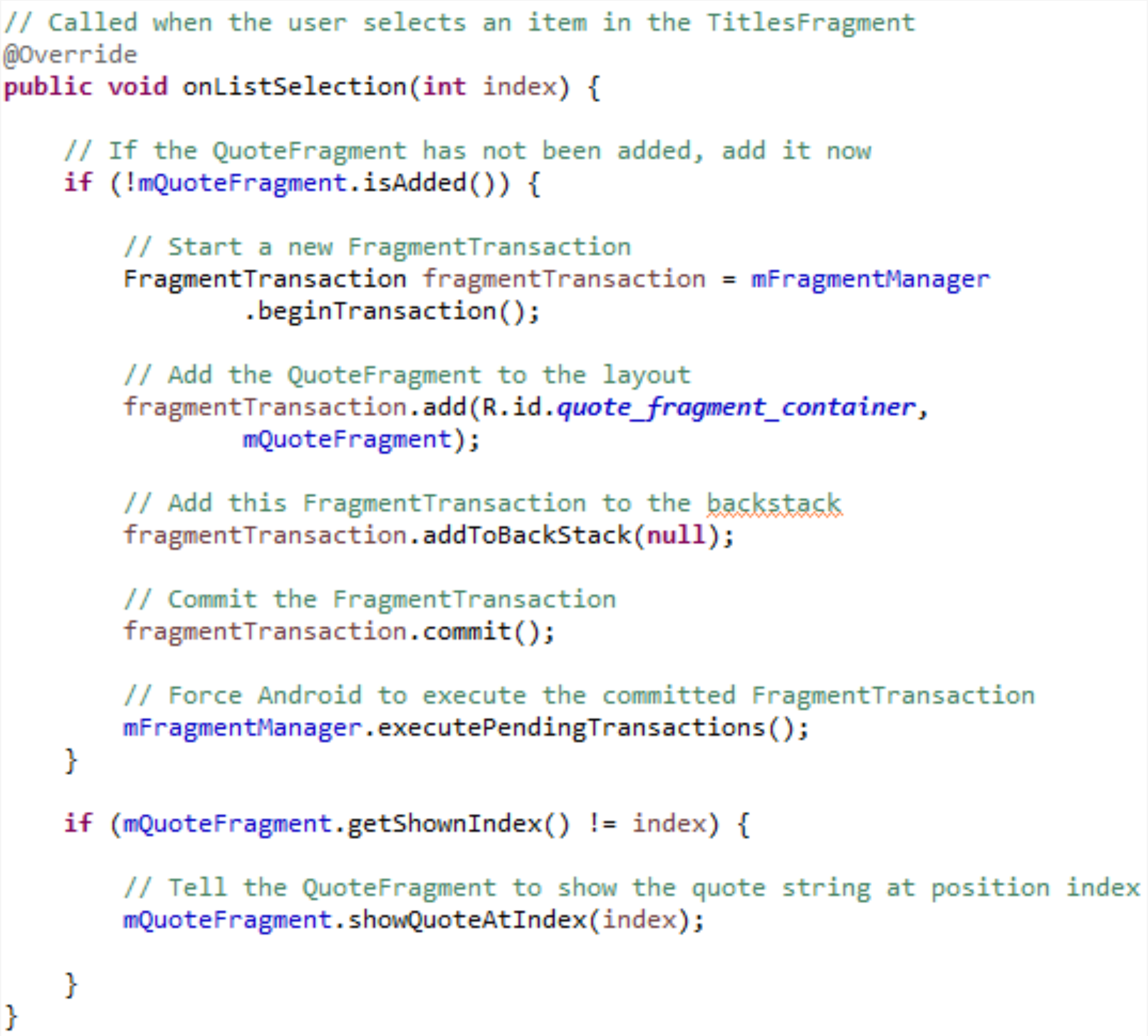
Now cracks a noble heart. Good-night, sweet prince; And flights of angels sing thee to thy rest.

FragmentDynamicLayout



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

    Log.i(TAG, getClass().getSimpleName() + ":entered onCreate()");

    super.onCreate(savedInstanceState);

    // Get the string arrays with the titles and qutoes
    mTitleArray = getResources().getStringArray(R.array.Titles);
    mQuoteArray = getResources().getStringArray(R.array.Quotes);

    setContentView(R.layout.main);

    // Get references to the TitleFragment and to the QuotesFragment
    mTitleFrameLayout = (FrameLayout) findViewById(R.id.title_fragment_container);
    mQuotesFrameLayout = (FrameLayout) findViewById(R.id.quote_fragment_container);


    // Get a reference to the FragmentManager
    mFragmentManager = getFragmentManager();

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

    // Add the TitleFragment to the layout
    fragmentTransaction.add(R.id.title_fragment_container,
            new TitlesFragment());

    // Commit the FragmentTransaction
    fragmentTransaction.commit();

    // Add a OnBackStackChangedListener to reset the layout when the back stack changes
    mFragmentManager
            .addOnBackStackChangedListener(new FragmentManager.OnBackStackChangedListener() {
                public void onBackStackChanged() {
                    setLayout();
                }
            });
}
```

FragmentDynamicLayout



```
// Called when the user selects an item in the TitlesFragment
@Override
public void onListSelection(int index) {

    // If the QuoteFragment has not been added, add it now
    if (!mQuoteFragment.isAdded()) {

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

        // 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();
    }

    if (mQuoteFragment.getShownIndex() != index) {

        // Tell the QuoteFragment to show the quote string at position index
        mQuoteFragment.showQuoteAtIndex(index);

    }
}
```

Configuration Changes

If you call setRetainInstance(true), 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

Functionality is similar to previous examples

FragmentStaticConfigLayout

In landscape mode

Both Fragments use a large font

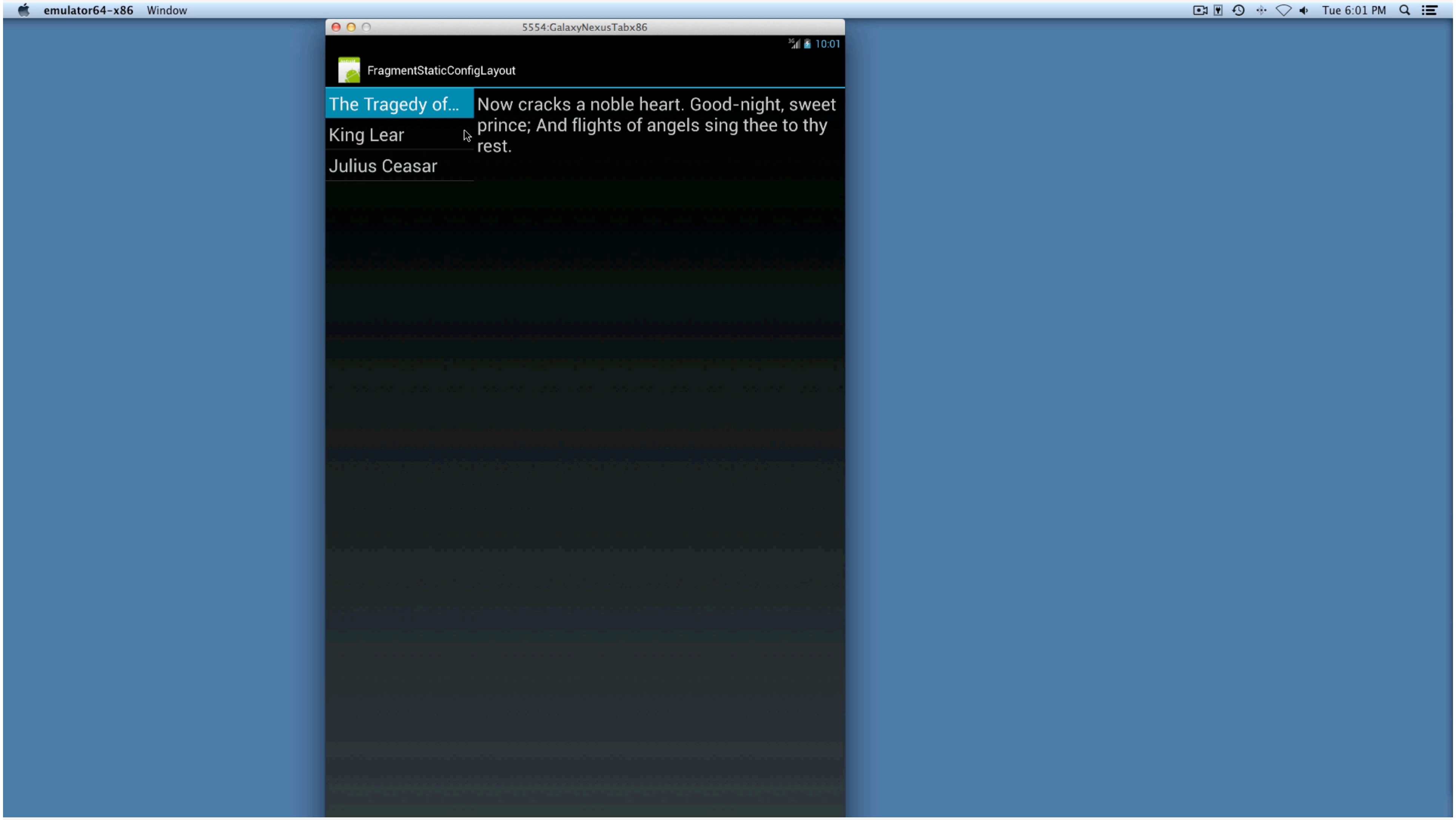
TitleFragment takes more horizontal space & allows long titles to span multiple lines

FragmentStaticConfigLayout

In portrait mode

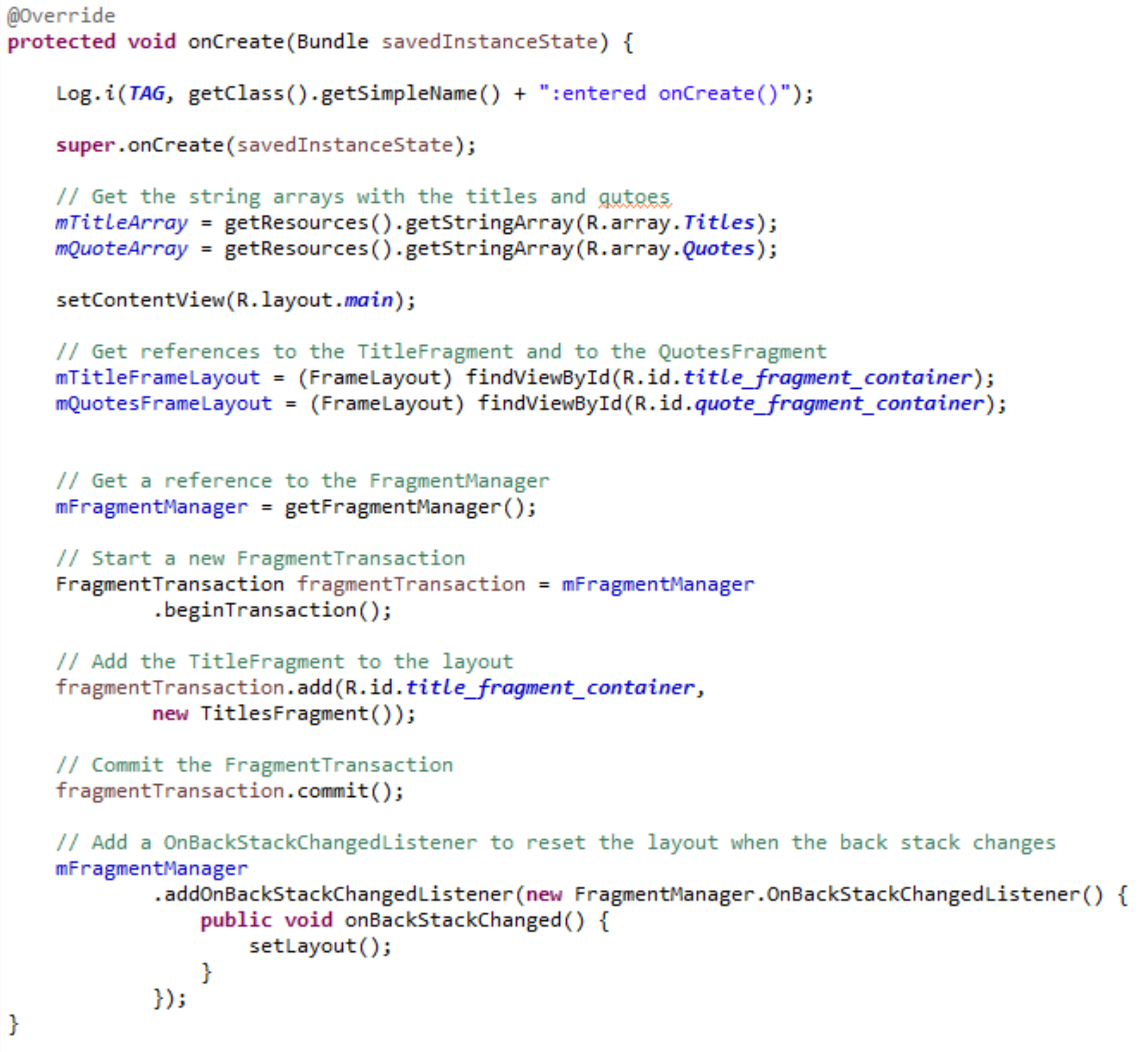
Both Fragments use a smaller font

TitleFragment will use less space and will ellipsize long titles, limiting them to a single line



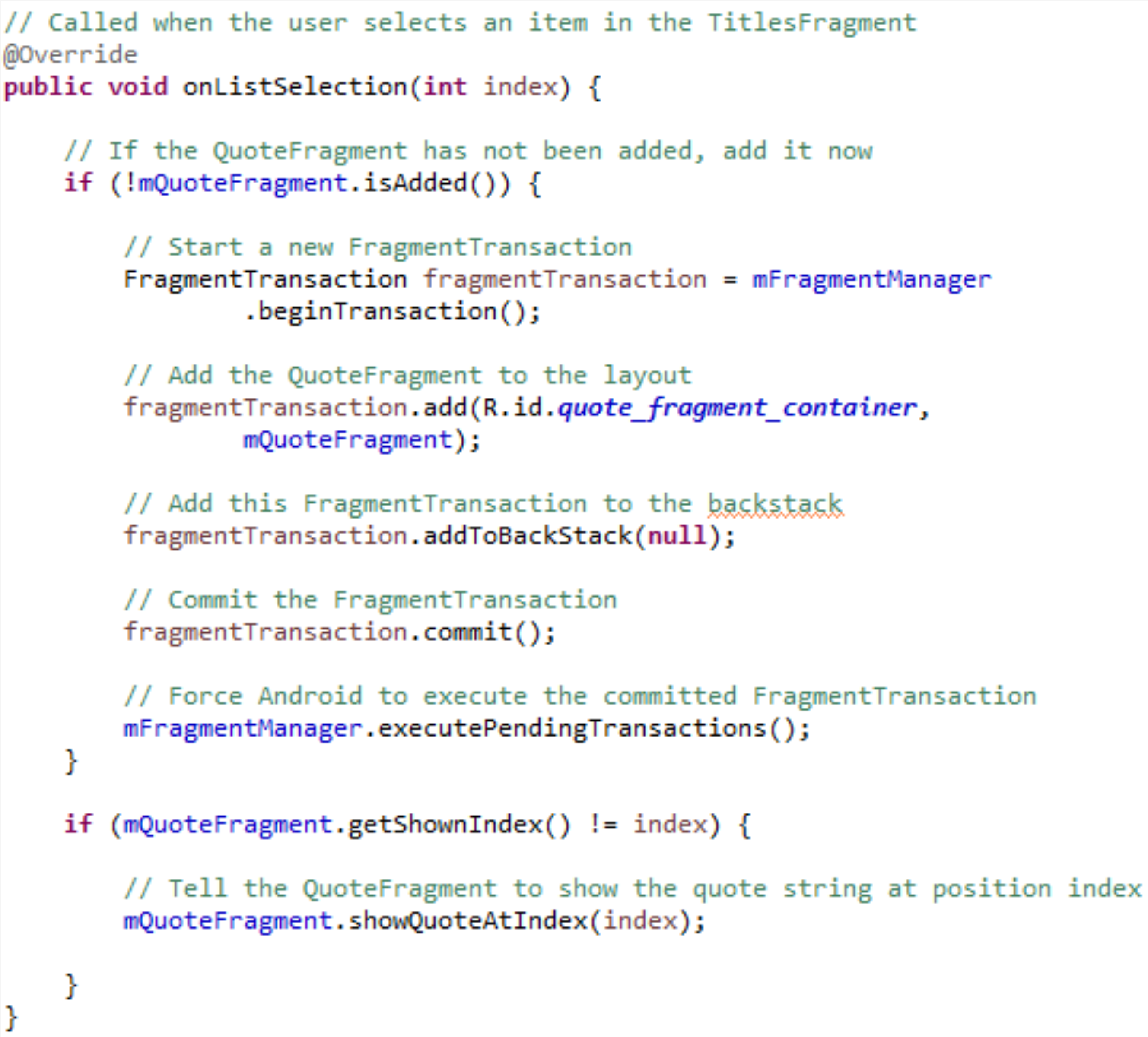
Now cracks a noble heart. Good-night, sweet prince; And flights of angels sing thee to thy rest.

FragmentDynamicLayout



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

    Log.i(TAG, getClass().getSimpleName() + ":entered onCreate()");

    super.onCreate(savedInstanceState);

    // Get the string arrays with the titles and qutoes
    mTitleArray = getResources().getStringArray(R.array.Titles);
    mQuoteArray = getResources().getStringArray(R.array.Quotes);

    setContentView(R.layout.main);

    // Get references to the TitleFragment and to the QuotesFragment
    mTitleFrameLayout = (FrameLayout) findViewById(R.id.title_fragment_container);
    mQuotesFrameLayout = (FrameLayout) findViewById(R.id.quote_fragment_container);


    // Get a reference to the FragmentManager
    mFragmentManager = getFragmentManager();

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

    // Add the TitleFragment to the layout
    fragmentTransaction.add(R.id.title_fragment_container,
            new TitlesFragment());

    // Commit the FragmentTransaction
    fragmentTransaction.commit();

    // Add a OnBackStackChangedListener to reset the layout when the back stack changes
    mFragmentManager
            .addOnBackStackChangedListener(new FragmentManager.OnBackStackChangedListener() {
                public void onBackStackChanged() {
                    setLayout();
                }
            });
}
```

FragmentDynamicLayout



```
// Called when the user selects an item in the TitlesFragment
@Override
public void onListSelection(int index) {

    // If the QuoteFragment has not been added, add it now
    if (!mQuoteFragment.isAdded()) {

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

        // 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();
    }

    if (mQuoteFragment.getShownIndex() != index) {

        // Tell the QuoteFragment to show the quote string at position index
        mQuoteFragment.showQuoteAtIndex(index);

    }
}
```

Next Time

User Interface classes