CMSC 436 Lab 2

Basic UI and Activities

Overview

- In this lab you will have two tasks

  - Create a simple UI with some text and a button that launches another Activity

  - Use Activity lifecycle callbacks to count the number of times an Activity is created, started, and resumed

Basic UI

- Information on how to build a simple UI can be found at

  http://developer.android.com/training/basics/firstapp/

- Check out "Building a Simple User Interface" and "Starting Another Activity"

Basic UI

- Create TextViews and a Button as shown at right

- You can make the LinearLayout arrange the widgets vertically by adding an android:orientation tag to the layout xml
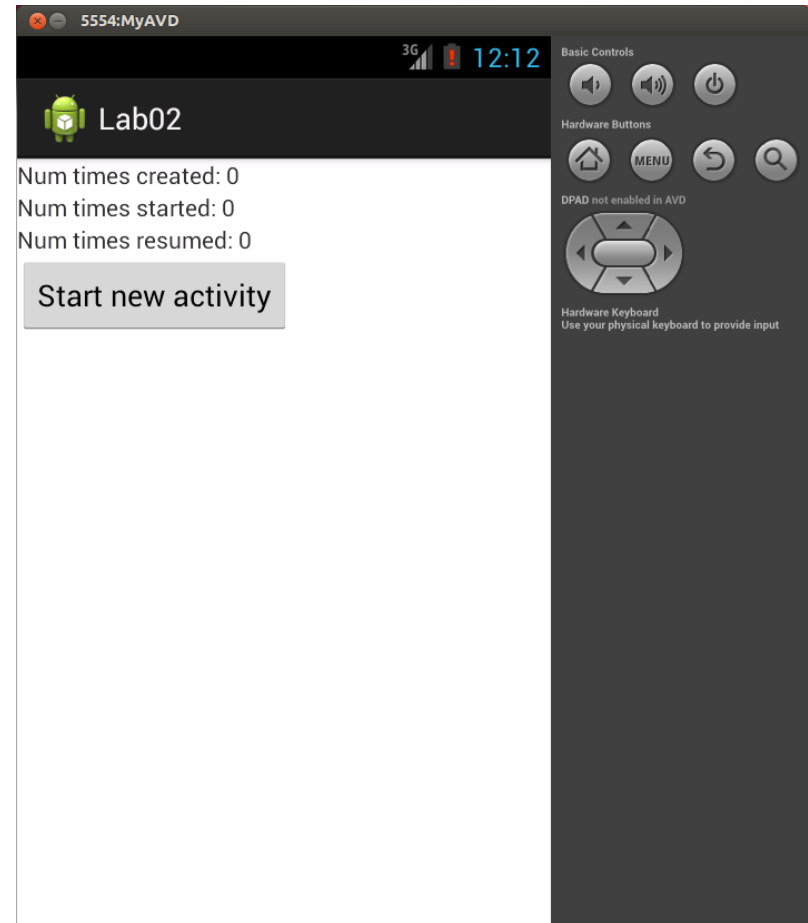
Basic UI

- Clicking the button should launch another Activity as shown at right

- Return to the previous Activity by using the back button on your device or emulator
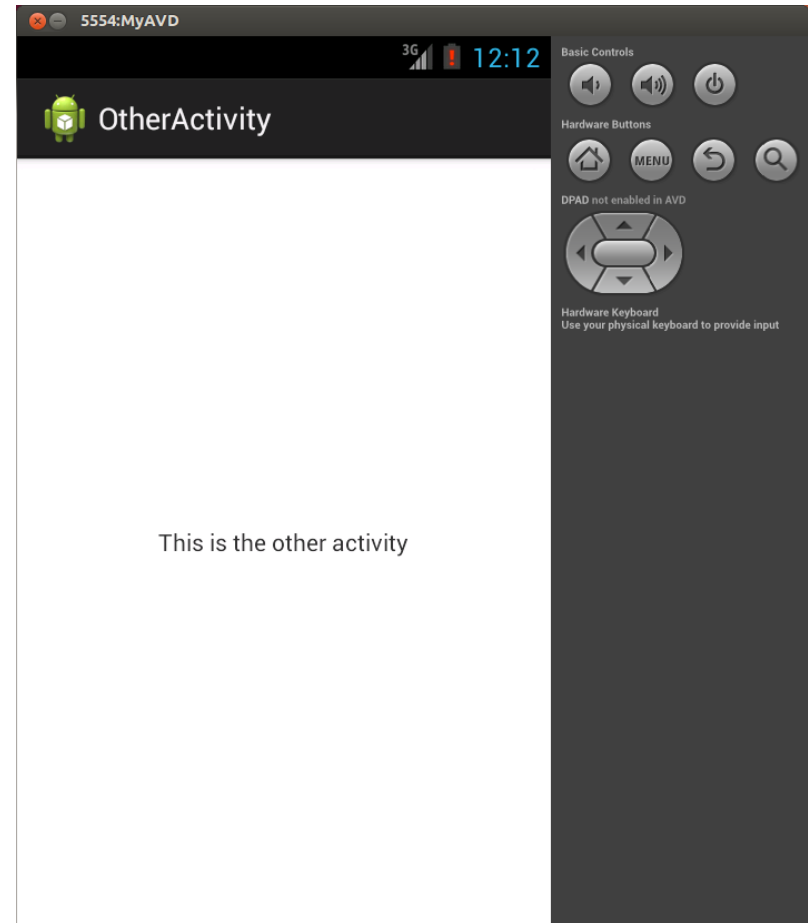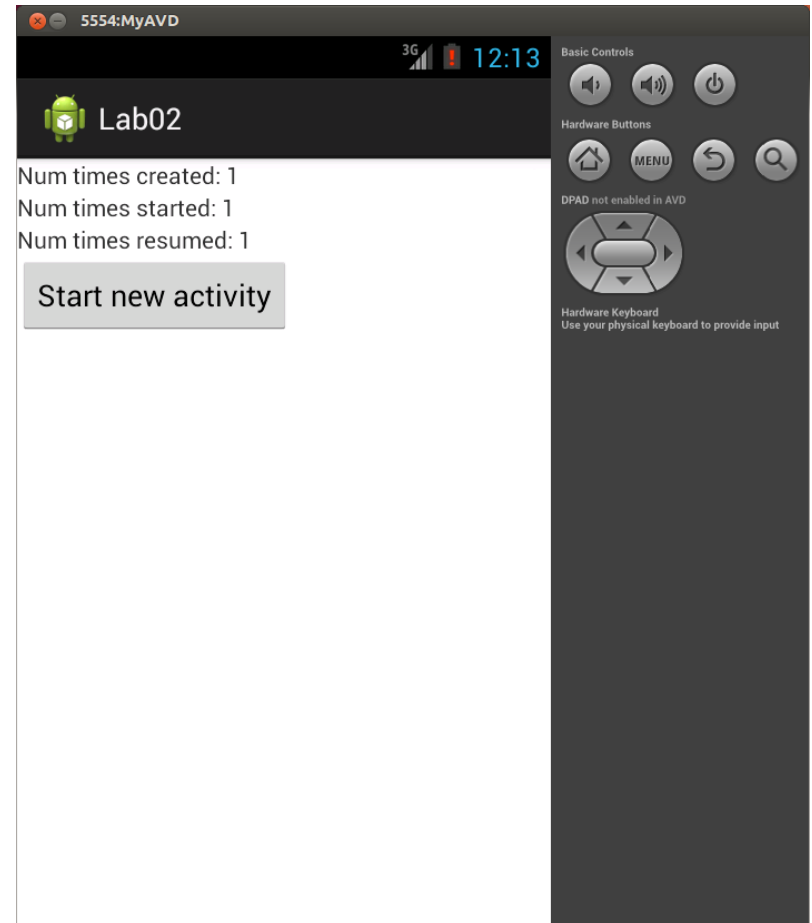
Activity Lifecycle

- Now you will use Activity lifecycle callbacks to count the number of times the main Activity is created, started, and resumed

- See http://developer.android.com/training/basics/activity-lifecycle/ for information on the lifecycle callbacks
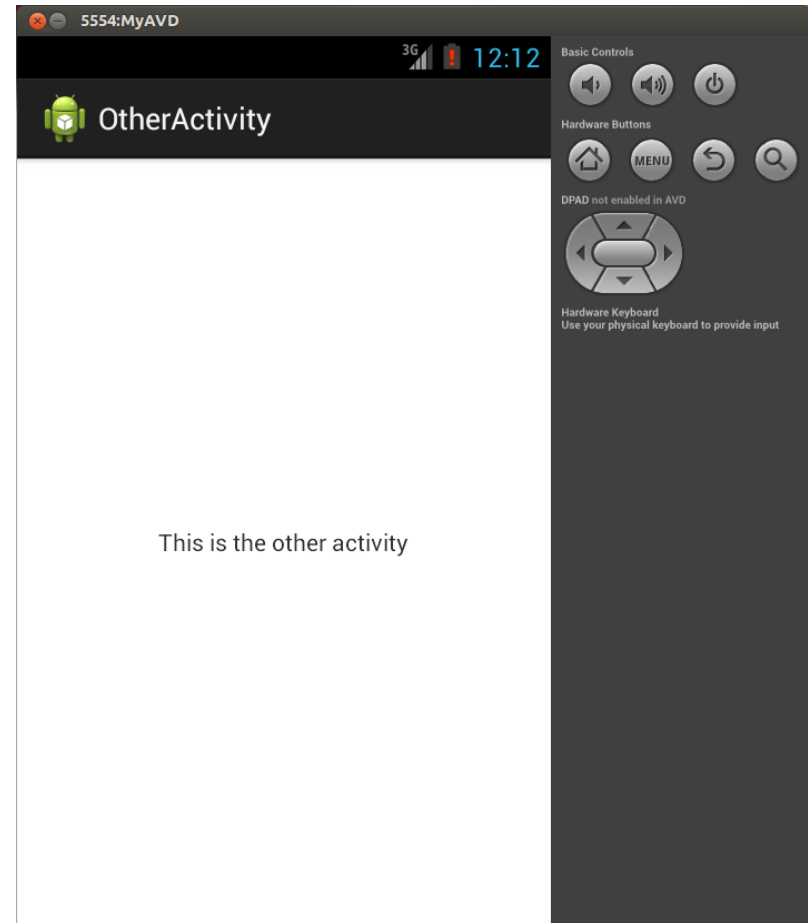
Activity Lifecycle

- When the Activity is first loaded it will have been created, started, and resumed one time each
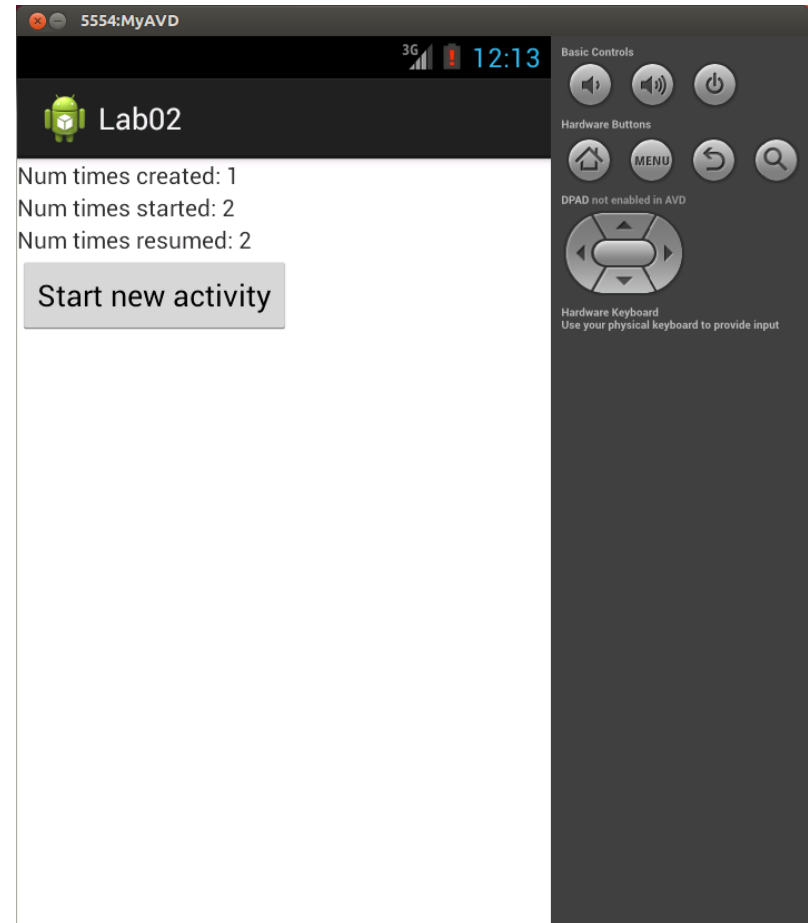
Activity Lifecycle

- Clicking the button to start a new activity will pause and stop the main Activity since the main Activity is completely covered
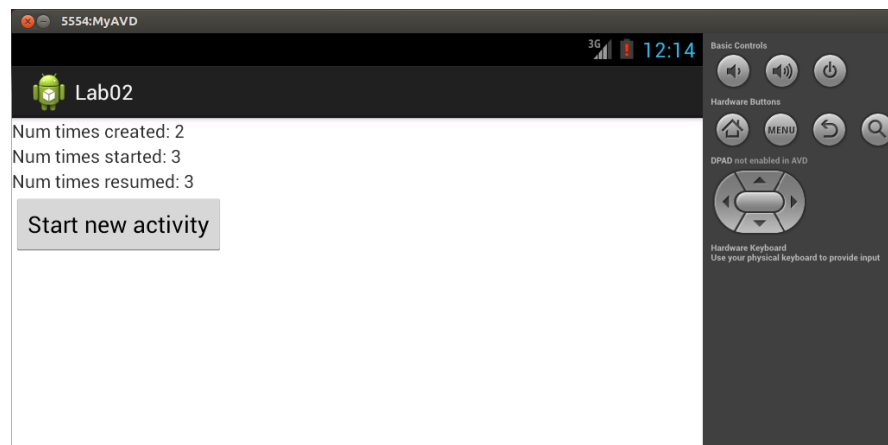
Activity Lifecycle

- Returning back to the main Activity will cause it to be started and resumed, but not created

Activity Lifecycle

- Rotating will cause the main Activity to be created, started, and resumed

- The emulator can be rotated with numeric keypad 7 and 9 or ctrl-F12

Activity Lifecycle

- To preserve the counts when the Activity is destroyed and created you will need to save them in a Bundle

- See "Recreating an Activity" at
  http://developer.android.com/training/basics/activity-lifecycle/
  for information on storing and retrieving data with a Bundle

Activity Lifecycle

- Finally, note that you can gain access to widgets defined in the layout xml by calling findViewById, such as

    TextView textView1 = (TextView) findViewById(R.id.textView1);
    textView1.setText("foo");