CSE244 Lab06, Start with Event Driven Programming

Objectives of this lab

1. To recognize event-driven control
2. To understand that event handlers describe responses to events, not their causes
3. To be able to write event handlers for simple event-driven systems

Concepts

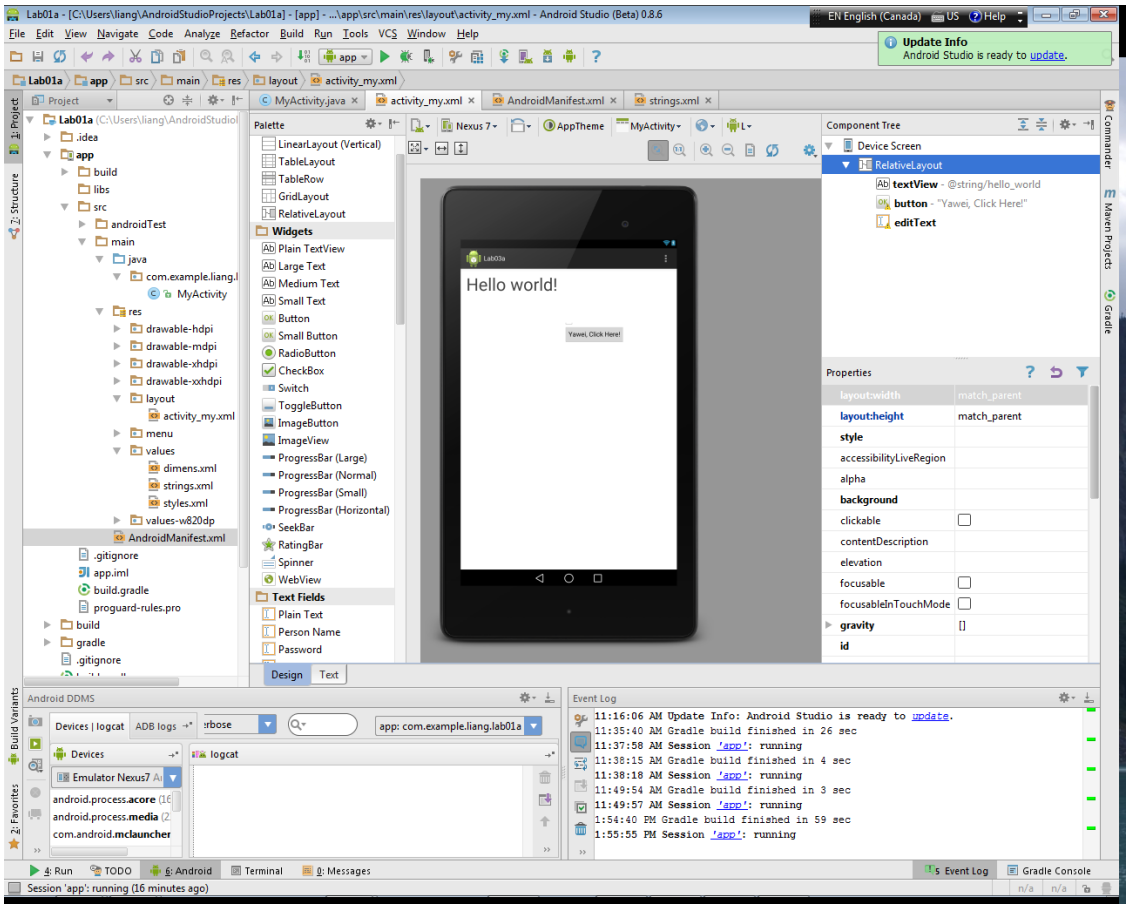
Event-driven programming focuses on how computer will act when certain events occur, for example, when a button is pressed. The activities of your program can vary depending on specific event occurred. This lab will deal with a couple of typical events and their responsive activities in Android Systems.

[**Exercise 1**] Echo Text Viewing after a Button is pressed.

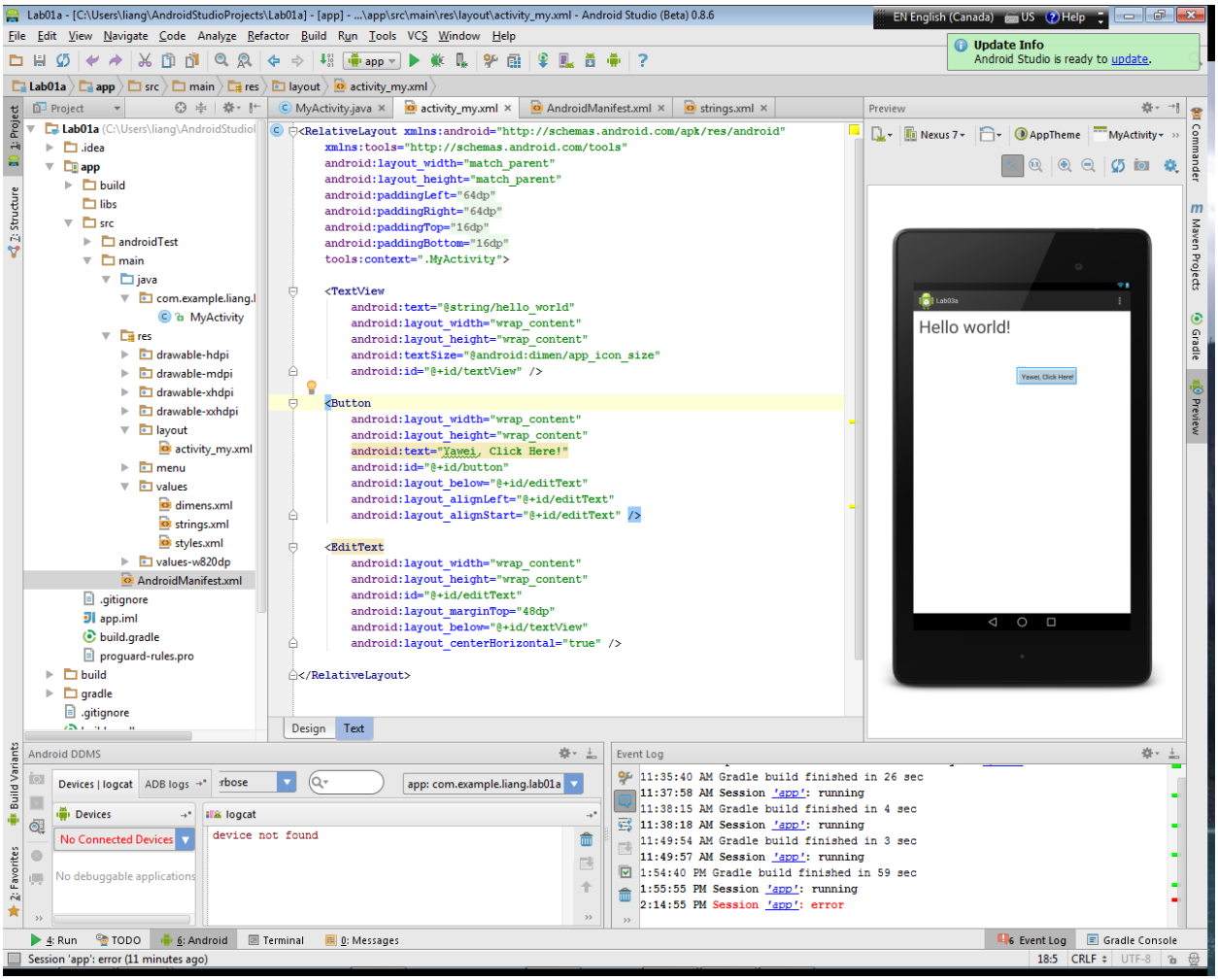
Step 1: Start a new project like we did in Lab05.

Step 2: Add a field for entering data and a button to trigger a display activity to your app.

- Find the `activity_main.xml` file in the `res/layout` folder. That is the file that was displayed when the app was created.
- Select the `Text Fields` section of the palette. Choose the first type of `EditText` field and drag it onto the canvas. As you drag it, you see the word "`Plain Text`". This will be the box you could type in a message or number, and later, we will echo this message/number in the `TextView` which shows "`Hello World!`" originally (a TextView will be inserted be default, in case you delete it, to add it back, select the `Plain TextView` from the palette and drag and drop it on your canvas.).
- Choose a `button` from the Form Widgets section of the palette and dag it onto the app canvas. You can change the label on the button by clicking it and type what you choose. This button will be listened if there is any click on it happens, then, we are going to echo the message/number into the `TextView` box.



If Text view of the activity_main.xml is chosen, then, you should see its contents change accordingly.



- Save your change.

Step 4: Change the Java code, i.e., MyActivity.java, to start the display activity as soon as the button is clicked.

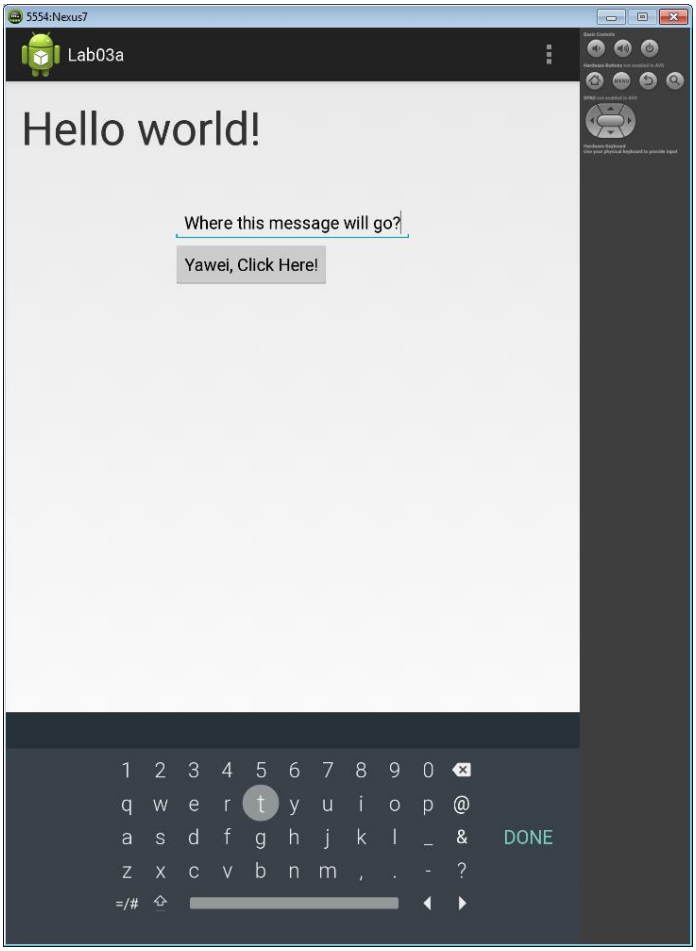
- First, import those classes:

```
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View.OnClickListener;  // yawei add this
import android.view.View;        // yawei add this
import android.widget.Button;    // yawei add this
import android.widget.EditText; // yawei add this
import android.widget.TextView; // yawei add this
```

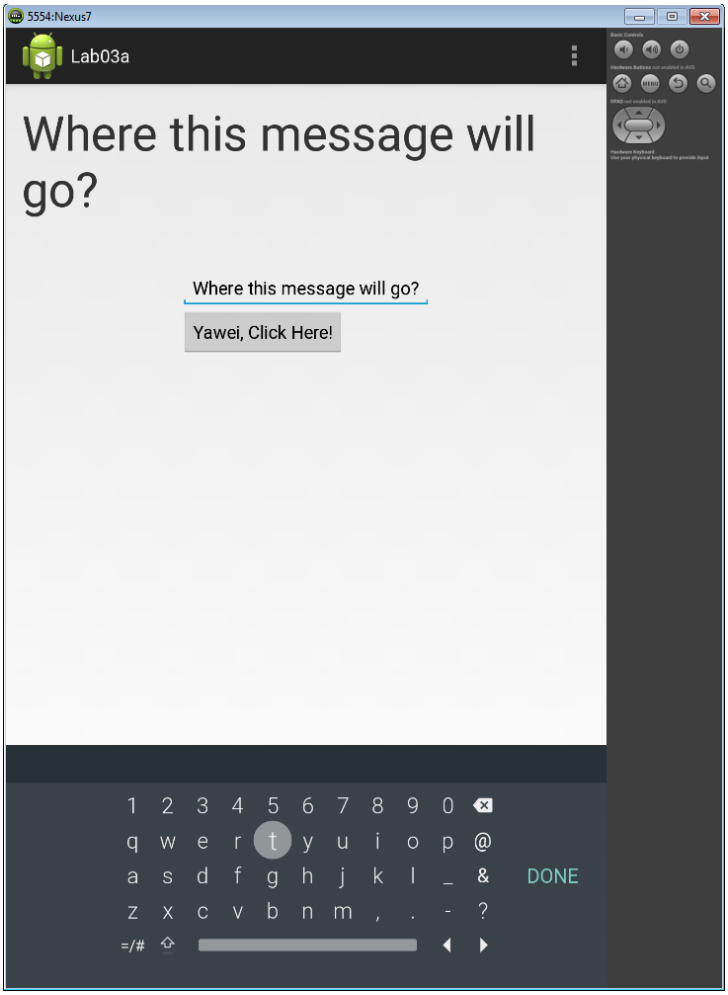
- Change the content of `OnCreate()` method. The code to update the `TextView` from the `EditText` field occurs on the line: `t.setText(e.getText());`

```
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my);
        // yawei add the following lines
        final EditText e = (EditText)findViewById(R.id.editText);
        final TextView t = (TextView)findViewById(R.id.textView);
        Button b = (Button)findViewById(R.id.button);
        b.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                t.setText(e.getText());
            }
        });
    }
```

Step 5: Run you app and you will see the following after you type a message in the EditView box and click on the button.



After button is clicked, you will see:



[**Exercise 2**] Build an app which reads a real value in from a user via `EditText` representing temperature, listens to one of the two buttons to convert the temperature to its corresponding Celsius or Fahrenheit, and out put the result in the `TextView`.

The followings are the formula for the conversion.

```
Celsius to Fahrenheit:   °C × 1.8 + 32 = °F
Fahrenheit to Celsius:   (°F − 32) / 1.8 = °C
```

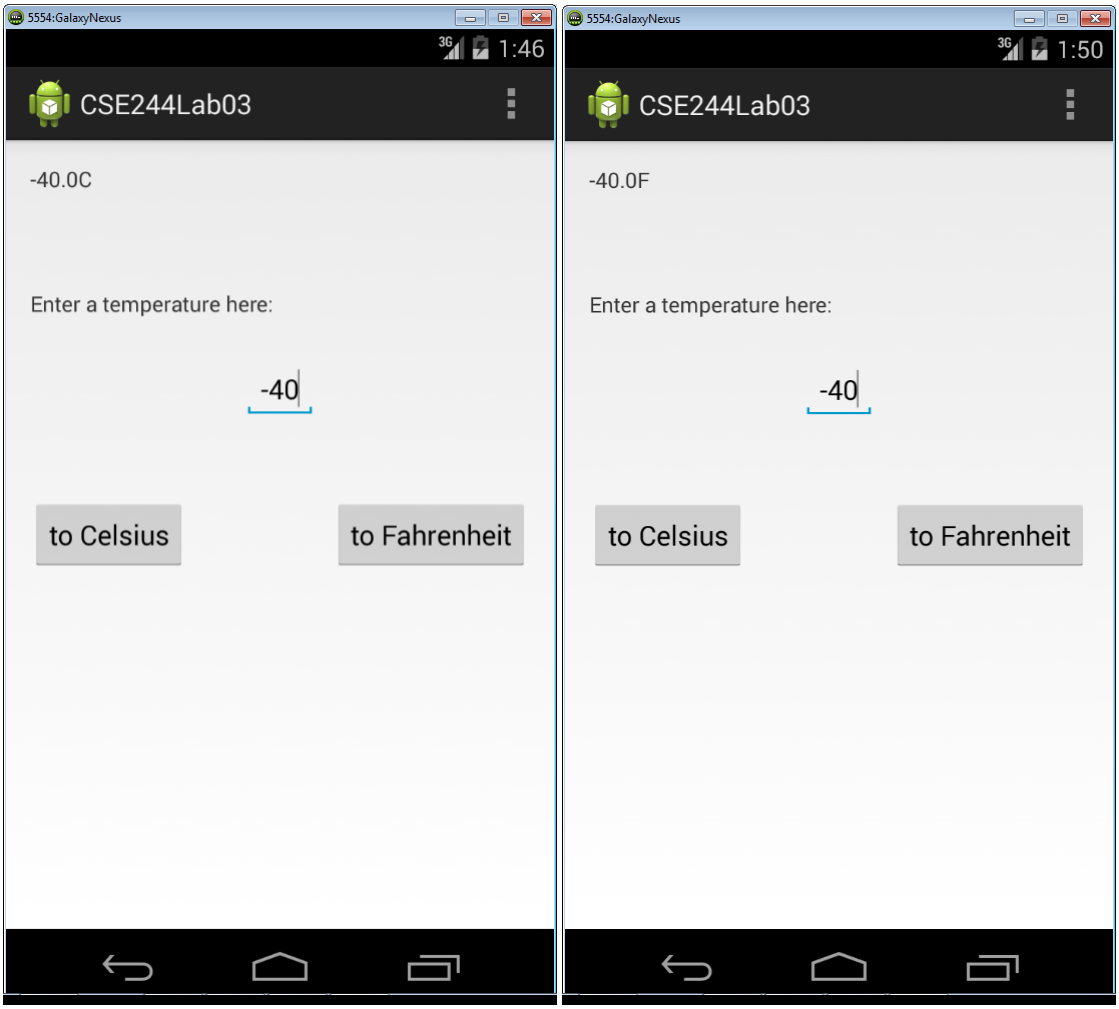
To convert the text in `EditText` to a double value, you have to use the following Java code:

```
String s = e.getText().toString();    // this will get a string
double n = Double.parseDouble(s);     // this converts string s to double
```

To convert a `double` value back to `String`, you need to do this:

```
String.valueOf(n)
```

Here are two sample screenshots of the app:



Deliverables

Submit the Java code, the xml file inserted in a word file, and a screenshot of your GUI for Exercise 2 to liang-y@rmc.ca before the deadline.

Deadline

Nov. 16, 2018.

Marking Scheme

No.	Item to check	Mark assigned	**Mark gained**
1	2 buttons, 1 `EditText`, 1 `TextView` designed and labelled properly.	3	
2	2 click listeners are programmed.	2	
3	Celsius to Fahrenheit calculated properly	1	
4	Fahrenheit to Celsius calculated properly	1	
5	Results are displayed in `TextView`	1	
6	A screenshot of your GUI	2	
	Total:	10	