Tutorial 6 – Accelerometer

1. Create a new project and name it “Accelerometer.” Choose API 21 or so, then keep the rest of the selections generic, no need to change them.
2. Create three different Large Texts on your screen. The best way to do this is to have them aligned vertically with some space between them. In order from top to bottom, have them display X, Y, and Z. For the id’s, set them to xText, yText, and zText. Feel free to modify the text color or background color. Our exquisite design is complete!
3. Open up the MainActivity.java file so we can start coding. As always, we need to import a few packages:

**import** android.hardware.Sensor;  
**import** android.hardware.SensorManager;  
**import** android.hardware.SensorEvent;  
**import** android.hardware.SensorEventListener;  
**import** android.widget.TextView;

1. Once we have imported these packages, we need to declare objects for our TextViews, Sensor, and Sensor Manager.

**private** TextView **xText**, **yText**, **zText**;  
**private** Sensor **mySensor**;  
**private** SensorManager **SM**;

1. Now, within the onCreate method we must create our sensorManager and Accelerometer Sensor, as well as “register” our Sensor Listener:

*// Create our Sensor Manager***SM** = (SensorManager)getSystemService(***SENSOR\_SERVICE***);  
  
*// Accelerometer Sensor***mySensor** = **SM**.getDefaultSensor(Sensor.***TYPE\_ACCELEROMETER***);  
  
*// Register sensor Listener***SM**.registerListener(**this**, **mySensor**, SensorManager.***SENSOR\_DELAY\_NORMAL***);

I happened to come across a little “Easter Egg” on my way through programming this. I find that when registering the Listener, if you are typing “SensorManger.G” there is a sensor manager for the gravity of the first Death Star. Amazing.

1. Continuing on – we need to assign our TextView objects to their corresponding id’s. We have done this a few times, so this should seem familiar:

*// Assign TextViews***xText** = (TextView)findViewById(R.id.***xText***);  
**yText** = (TextView)findViewById(R.id.***yText***);  
**zText** = (TextView)findViewById(R.id.***zText***);

1. Now, to be able to use the sensor, we need to implement a class named SensorEventListener:

**public class** MainActivity **extends** Activity **implements** SensorEventListener{

// All other previous code

}

1. Now that we’ve implemented SensorEventListener, we need to call its methods. Find a blank area beneath onCreate and hold the Alt key, then press Insert. This will bring up a little menu, which you will select “Implement Methods.” Click OK to populate those methods. In the method onAccuracyChanged, add a comment saying “//not in use” or so forth.
2. In the onSensorChanged method, we will be changing the text that is shown on the screen each time the sensor updates. The data from the sensor is output in the format of an array, where 0 is the X-axis, 1 is the Y-axis, and 2 is the Z-axis. Place the following code inside the onSensorChanged method:

xText.setText(**"X: "** + event.**values**[0]);  
**yText**.setText(**"Y: "** + event.**values**[1]);  
**zText**.setText(**"Z: "** + event.**values**[2]);

1. Last but not least, you will have to export this program to your phone, because the emulator will not give you readings. If you don’t have an android phone, or cannot get it to work out – I have provided this lesson with a video and show the results at the end (I may have added some extra coloring as well).

Video Link: <https://www.youtube.com/watch?v=YrI2pCZC8cc&index=5&list=PLFVlCGwfyegYi8G0yxIVlGfjT3xGzCZOz>

Github Link: <https://github.com/amyork/Android_Studio_Tut_6_Accelerometer>