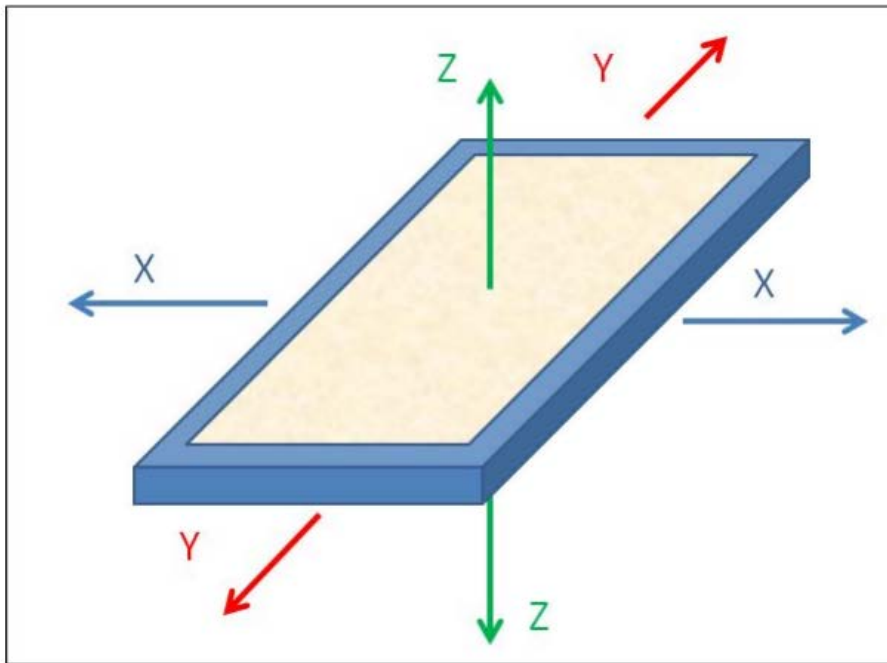


# Accelerometer in Android Java

Mr. Michaud

Marist School

# Steps for Accelerometer Setup



1. Declare a field for `SensorManager`
2. In `onCreate()` – call the function to enable the listener.
3. Define the Listener as a function.
4. Write the inner class that defines the `SensorEventListener` and reads / reacts to data.

# Field: SensorManager

```
// Sensor Manager  
private SensorManager sensorManager;
```

# onCreate()

```
// Enable the listener - We will write this later in the class  
enableAccelerometerListening();
```

# Define the Listener

```
private void enableAccelerometerListening() {  
    // Initialize the Sensor Manager  
    sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
    sensorManager.registerListener(sensorEventListener, sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),  
        SensorManager.SENSOR_DELAY_NORMAL);  
}
```

Note the `sensorEventListener` – we will define this later.

# SensorEventListener

Code to read and respond to sensor values goes here:

```
private SensorEventListener sensorEventListener =
    new SensorEventListener()
    {
        // Listens for Change in Acceleration, Displays, and Computes the Steps
        public void onSensorChanged(SensorEvent event)
        {
            // Gather the values from accelerometer
            float x = event.values[0];
            float y = event.values[1];
            float z = event.values[2];

            // Fetch the current y
            currentY = y;

            // Measure if a step is taken
            if ( Math.abs(currentY - previousY) > threshold ) {
                numSteps++;
                textViewSteps.setText(String.valueOf(numSteps));
            } // end if

            // Display the Values
            textViewX.setText(String.valueOf(x));
            textViewY.setText(String.valueOf(y));
            textViewZ.setText(String.valueOf(z));

            // Store the previous Y
            previousY = y;
        } // end onSensorChanged

        public void onAccuracyChanged(Sensor sensor, int accuracy)
        {
            // Empty - required by Class
        } // end onAccuracy Changed
    }; // ends private inner class sensorEventListener
```