**PRACTICAL # 14**

**OBJECT:**

Working with Sensors

**THEORY:**

Android devices have built-in sensors that measure motion, orientation, and other environmental condition. The android platform supports three broad categories of sensors:

Motion Sensors

Environmental sensors

Position sensors

Some of the sensors are hardware based and some are software based sensors. Hardware-based sensors are physical components built into a handset or tablet device. They derive their data by directly measuring specific environmental properties, such as acceleration, geomagnetic field strength, or angular change. Software-based sensors are not physical components and derive their data from one or more of the hardware-based sensors. The linear acceleration sensor and the gravity sensor are examples of software-based sensors.

**SensorManager**

Android allows to access the data from these sensors and use it in application. Some of the important classes to manage sensors are SensorManager and Sensor. To use sensors, instantiate the object of

SensorManager class. It can be achieved as follows.

*SensorManager sensorManager = (SensorManager)this.getSystemService(SENSOR\_SERVICE);*

Now you can instantiate any sensor by calling getDefaultSensor() method of the SensorManager class. Before getting data from the sensor, check if the required sensor is available on the device.

*Sensor sensorGyroscope = sensorManager.getDefaultSensor(Sensor.TYPE\_GYROSCOPE);*

*if (sensorGyroscope!=null){*

*Toast.makeText(this, "sensor gyro init", Toast.LENGTH\_LONG).show();*

*}else{*

*Toast.makeText(this, "sensor gyro not available", Toast.LENGTH\_LONG).show();*

*}*

**Monitoring Sensor Events**

If the sensor is available and the object is initialized, register its listener and override two methods which are onAccuracyChanged and onSensorChanged.

*sensorManager.registerListener(new SensorEventListener() {*

*@Override public void onSensorChanged(SensorEvent event) {*

*}*

*@Override public void onAccuracyChanged(Sensor sensor, int accuracy) {*

*}*

*}, sensorLight, SensorManager.SENSOR\_DELAY\_NORMAL);*

**Reading Sensor Data**

Different sensors measure their specific kinds of measurements. An acceleration sensor measures the acceleration applied to the device, including the force of gravity.

*public void initAccelerometer(){*

*Sensor sensorAccelerometer = mSensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER);*

*mSensorManager.registerListener(new SensorEventListener() {*

*@Override public void onSensorChanged(SensorEvent event) {*

*mTextViewSensorAccelerometer.setText("x:*

*"+event.values[0]+"\ny: "+event.values[1]+"\nz: "+event.values[2]);*

*}*

*@Override*

*public void onAccuracyChanged(Sensor sensor, int accuracy)*

*{*

*}*

*}*

*,sensorAccelerometer, SensorManager.SENSOR\_DELAY\_NORMAL*

*);*

*}*

**Using Google Play filters to target specific sensor configurations**

If you are publishing your application on Google Play you can use the <uses-feature> element in your manifest file to filter your application from devices that do not have the appropriate sensor configuration for your application. The following example filters devices that do not have an accelerometer:

*<uses-feature android:name="android.hardware.sensor.accelerometer"android:required="true" />*

**ACTIVITIES**

**Activity 1**

Create a simple game that uses accelerometer to show a ball moving relative to the motion of the phone.

**REVIEW QUESTIONS**

1. Which sensors are available in Android?
2. How do you access a sensor’s data?
3. What is the difference between hardware based and software based sensors?
4. What are sensor events?