Sensors

Sensors Overview

- Sensors are designed for diverse needs
- motion, orientation and environmental conditions
- Android-powered devices have built-in sensors
- e.g. three dimensional device movement, positioning or changes in ambient environment
- e.g. game might track readings from a device's gravity sensor to infer complex user gestures such as tilt, shake, rotation or swing
- a weather application might use a device's temperature and humidity sensor
- Reference: Android Developer Sensor

Types of sensors

- Motion sensors - measures acceleration forces and rotational forces along three axes e.g. accelerometer, gravity sensor, gyroscope and rotational vector sensor
- Environmental sensors - measure various environmental parameters, such as temperature, pressure, illumination and humidity. e.g. barometers, photometers and thermometers
- Position sensors - measure the physical position of a device. This category includes orientation sensors and magnetometers.

Sensor Framework

Using Sensor framework,

- determine which sensors are available on a device
- determine individual sensor's capabilities
- acquire raw sensor data
- register and unregister sensor event listeners

Identifying Sensors

```
private SensorManager sm;

public void onClick(View v) {
  String LOG_TAG="Sensor";
  sm=(SensorManager)getSystemService(Context.
      SENSOR_SERVICE);
  List<Sensor> ls = sm.getSensorList(Sensor.TYPE_ALL)
      ;
  for(Sensor s : ls) {
    Log.i(LOG_TAG, "Sensor Name :"+s.getName()+" Type
         :"+s.getType());
  }
```

other options for getSensorList()

1. TYPE_ACCELEROMETER, TYPE_PROXIMITY, TYPE_LIGHT
2. there may be multiple sensors for a single task
3. Information about Sensors can be obtained

Reading sensor data

```
public class MainActivity extends Activity implements
     SensorEventListener, OnClickListener {
  private SensorManager sm;
  private Sensor s;
  private TextView xtv,ytv,ztv;
  private boolean monitoring=false;
  private Button b;

  @Override
  protected void onCreate(Bundle b) {
    super.onCrete(b);
    setContentView(R.layout.activity_main);
```

Reading sensor data

```
  sm=(SensorManager)getSystemService(Context.
      SENSOR_SERVICE);
  s=sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
  xtv=(Textview)findViewById(R.id.xtv);
  ytv=(Textview)findViewById(R.id.ytv);
  ztv=(Textview)findViewById(R.id.ztv);
  b = (Button)findViewById(R.id.button);
  b.setOnClickListener(this);
}
@Override
public final void onAccuracyChanged(Sensor s, int
    accuracy) {
  //Do something on sensor accuracy change
}
```

Reading sensor data

```
@Override
public final void onSensorChanged(SensorEvent se) {
  if(monitoring) {
    float []values = se.values;
    //Movement
    float accx = values[0];
    float accy = values[1];
    float accz = values[2];

    xtv.setText(String.valueOf(accx));
    ytv.setText(String.valueOf(accy));
    ztv.setText(String.valueOf(accz));
  }
}

@Override
protected void onResume() {
  super.onResume();
  sm.registerListener(this,s,SensorManager.
      SENSOR_DELAY_NORMAL);
}
```

Reading sensor data

```
  @Override
  protected void onPause() {
    super.onPause();
    sm.unregisterListener(this);
  }

  @Override
  public void onClick(View v) {
    Toast.makeText(this,"Displaying Accelerometer
        data ",Toast.LENGTH_SHORT).show();
    monitoring = true;
  }
}
```