Programming Handheld Systems

Sensors

Today's Topics

SensorManager & Sensor

SensorEvent & SensorEventListener

Filtering sensor values

Example applications

Sensors

Hardware devices that measure the physical environment

Motion

Position

Environment

Some Example Sensors

Motion – 3-axis Accelerometer

Position – 3-axis Magnetic field

Environment – Pressure

SensorManager

System service that manages sensors

Get instance with

```
getSystemService(
    Context.SENSOR_SERVICE )
```

Access a specific sensor with

```
SensorManager.
    getDefaultSensor(int type)
```

Some Sensor Type Constants

Accelerometer –

Sensor.TYPE_ACCELEROMETER

Magnetic field –

Sensor.TYPE_MAGNETIC_FIELD

Pressure –

Sensor.TYPE_PRESSURE

SensorEventListener

Interface for SensorEvent callbacks

SensorEventListener

Called when the accuracy of a sensor has changed

```
void onAccuracyChanged(
        Sensor sensor, int accuracy)
```

SensorEventListener

Called when sensor values have changed

```
void onSensorChanged(
                    SensorEvent event)
```

Registering for SensorEvents

Use the SensorManager to register/ unregister for SensorEvents

Registering for SensorEvents

To register a SensorEventListener for a given sensor

```
public boolean registerListener (
            SensorEventListener listener,
            Sensor sensor, int rate)
```

Registering for SensorEvents

Unregisters a listener for the sensors with which it is registered

```
public void unregisterListener (
        SensorEventListener listener,
        Sensor sensor)
```

SensorEvent

Represents a Sensor event

Data is sensor-specific

sensor type

time-stamp

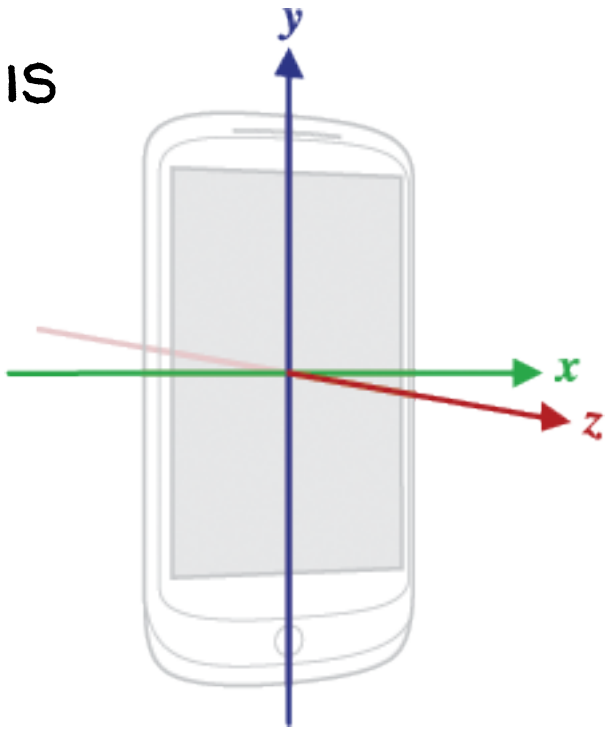
Accuracy

measurement data

Sensor Coordinate System

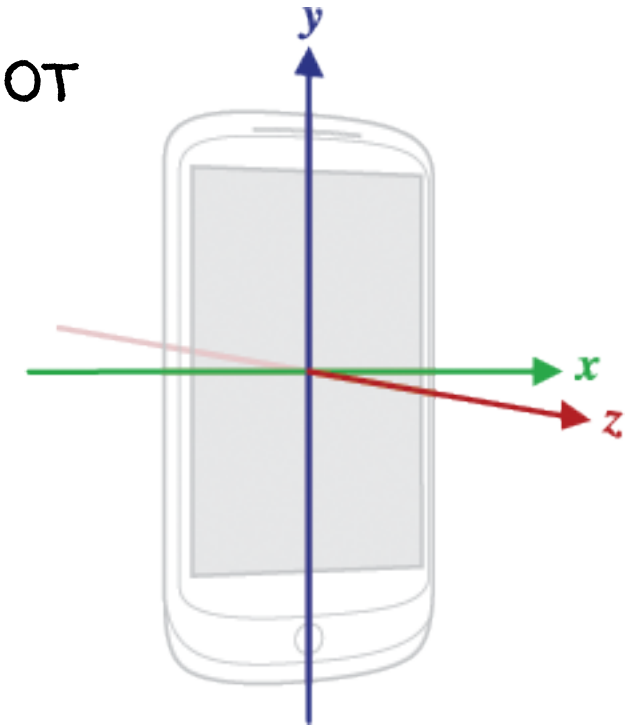
When default orientation is portrait & the device is lying flat, face-up on a table, axes run

- X – right to left
- Y – top to bottom
- Z – Down to up



Sensor Coordinate System

Coordinate system does not change when device orientation changes



SensorRawAccelerometer

Displays the raw values read from the device's accelerometer


17:36
SensorRawValues
Raw X: 1.187027
Raw Y: 9.5707855
Raw Z: -0.06745911

Demonstration of the SensorRawAccelerometer project in the IDE

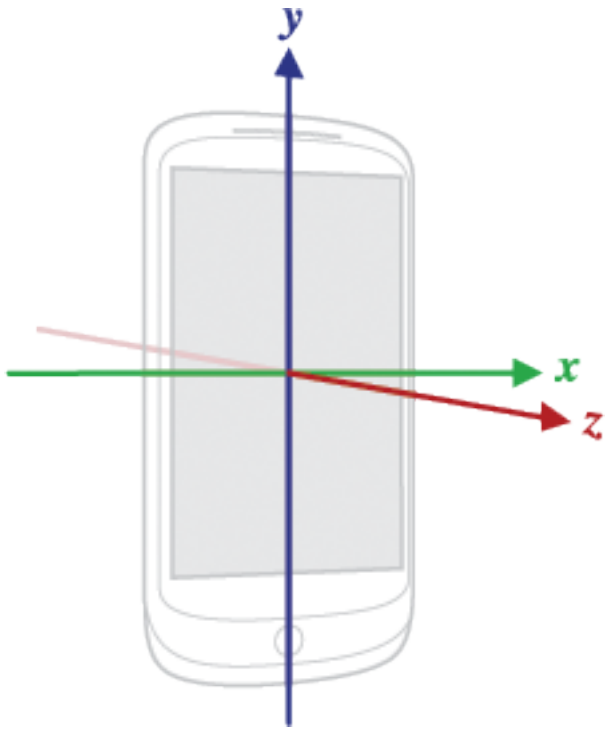
Accelerometer Values

If the device were standing straight up, the accelerometer would ideally report:

$X \approx 0$ m/s^2

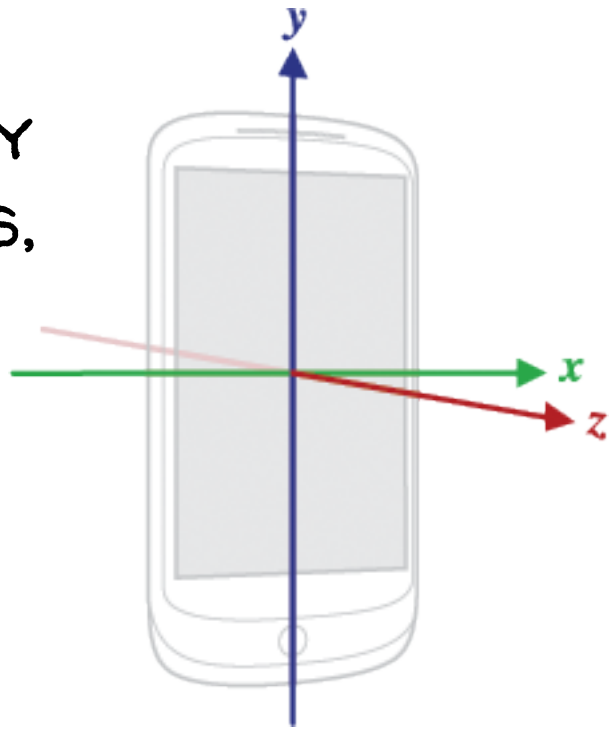
$Y \approx 9.81$ m/s^2

$Z \approx 0$ m/s^2



Accelerometer values

But these values will vary due to natural movements, non-flat surfaces, noise, etc.



Filtering Accelerometer values

Two common transforms

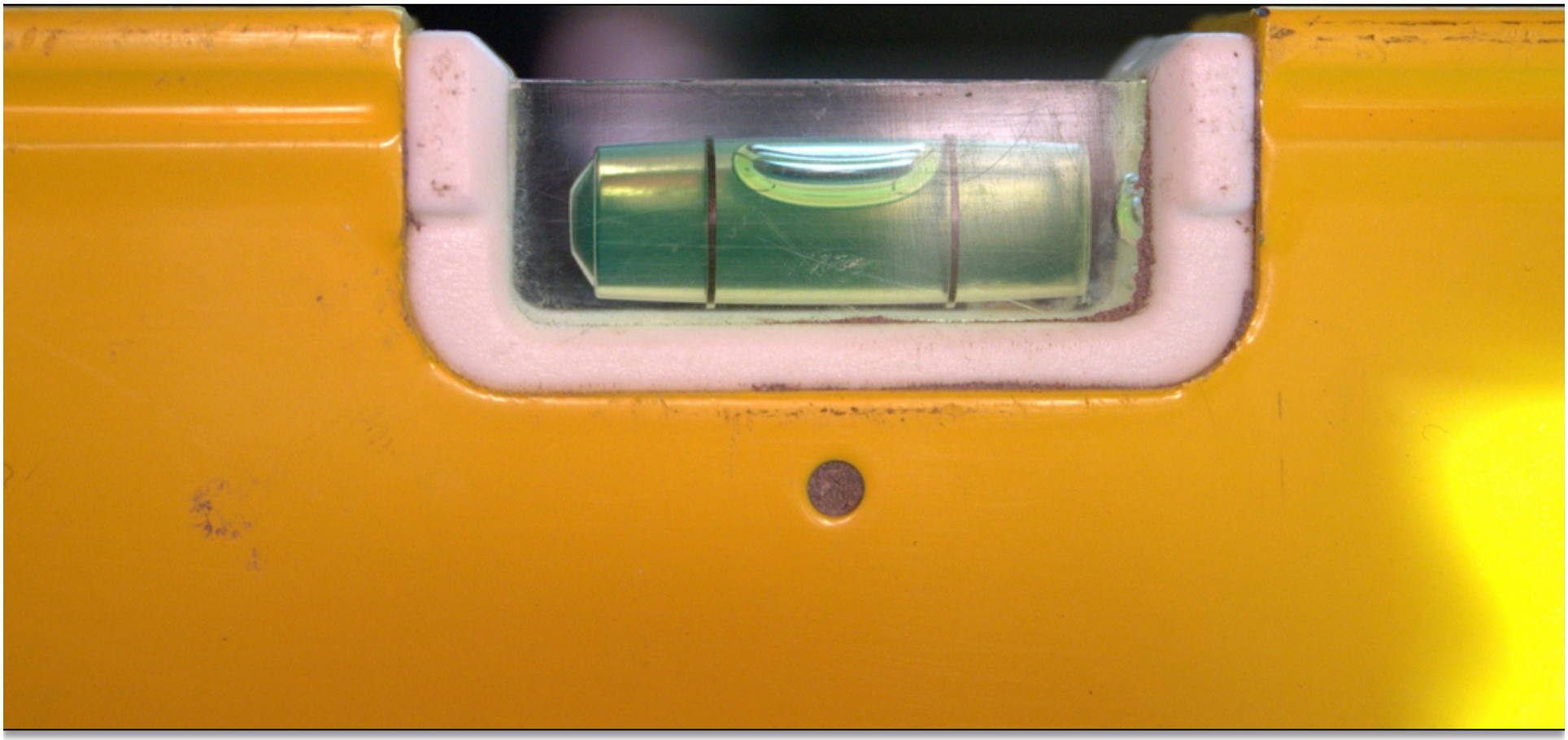
Low-pass filter

High-pass filter

Low-pass filters

Deemphasize transient force changes

Emphasize constant force components



Carpenter's Level

High-pass filters

Emphasize transient force changes

Deemphasize constant force components

Percussion Instrument

SensorFilteredAccelerometer

Applies both a low-pass and a high-pass filter to raw accelerometer values

Displays the filtered values


17:40
SensorFiltererdValues
Raw X: 0.030166626
Raw Y: 9.675522
Raw Z: 0.20509338
LowPass X 0.025992874
LowPass Y: 9.705229
LowPass Z: 0.18410519
HighPass X 0.004173752
HighPass Y:-0.029706955
HighPass Z:0.020988196

Demonstration of the SensorFilteredAccelerometer project in the IDE

SensorCompass

Uses the device's accelerometer and magnetometer to orient a compass


17:43
Compass

Demonstration of the SensorCompass project in the IDE

Next Time

Maps & Location