



학생 여러분 반갑습니다.  
다른 친구들이 입장할 때까지  
조금 기다려 주십시오.

곧 모바일 프로그래밍 수업을  
시작합니다.

음소거(🔇)가 되었는지 확인 바랍니다.

모바일 프로그래밍  
화목(1,2교시)/ 화목(3,4교시)  
정윤현 (AI/소프트웨어학부)



# Mobile Programming

## Android Programming

### Chap 10. Sensors

Prof. Younhyun Jung  
Email) [younhyun.jung@gachon.ac.kr](mailto:younhyun.jung@gachon.ac.kr)



# Sensors

- Android sensors:

- Accelerometer (가속센서)
- Gyroscope sensor (자이로스코프센서)
- Gravity sensor (중력센서)
- Light sensor (조도 센서)
- Linear Acceleration sensor (선형 가속 센서)
- Proximity sensor (근접 센서)
- Temperature sensor (온도 센서)
- Orientation sensor (방향 센서)
- etc..

+ magnet



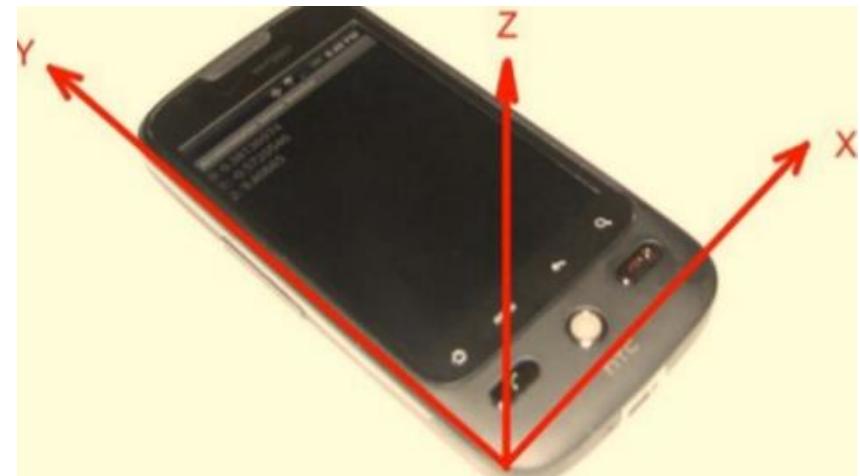
Ref : Android Meetup @ Tech Jini



# Accelerometer

→ 가속도 센서

- Provides the total force applied to the device ↳ 가만히 있을 때  
↳ 1G
- When device is stationary, this gives +1g (gravitational force) reading in one axis and its components on other axis
- All forces acting on the device will be detected by the accelerometers
- The measurement unit is SI  
(International System of Units) m/s<sup>2</sup>
- How to separate Gravity and Linear acceleration?
  - Used in gaming to make it interactive.
  - etc..



# Gravity Sensor and Linear Acceleration



- Gravity sensor is not a separate hardware but it's a virtual sensor based on the accelerometers, acceleration is removed from accelerometer data.  
 $\hookrightarrow$  가속도계 - 가속도 제거 = 중력.

- Similarly, Linear acceleration is not a separate hardware but it's based on the accelerometers, gravity is removed from accelerometer data.

$$\hookrightarrow \text{가속도계} - \text{중력} = \frac{\text{Linear Accelerator}}{\text{Accelerator}}$$

# Orientation sensor



- Measure degrees of rotation that a device makes around all three physical axes (x,y,z).
- There is no hardware sensor for orientation sensor but combination of two sensor (gravity sensor and magnetic field sensor) determine the orientation of device.
  - Magnetic field sensor : it can detect the Earth's magnetic field and thus tell us where north is. This sensor is also referred to as the compass

→ 힘 센서 Device 방향 구하기



# Gyroscope sensor

- Gyroscopes measure the rate of rotation around an axis. When the device is not rotating, the sensor values will be zeros.
- It gives us 3 values:
  - **Pitch** (around X axis)
  - **Roll** (around Y axis)
  - **Azimuth** (around Z axis)



# Light Sensor

조도.



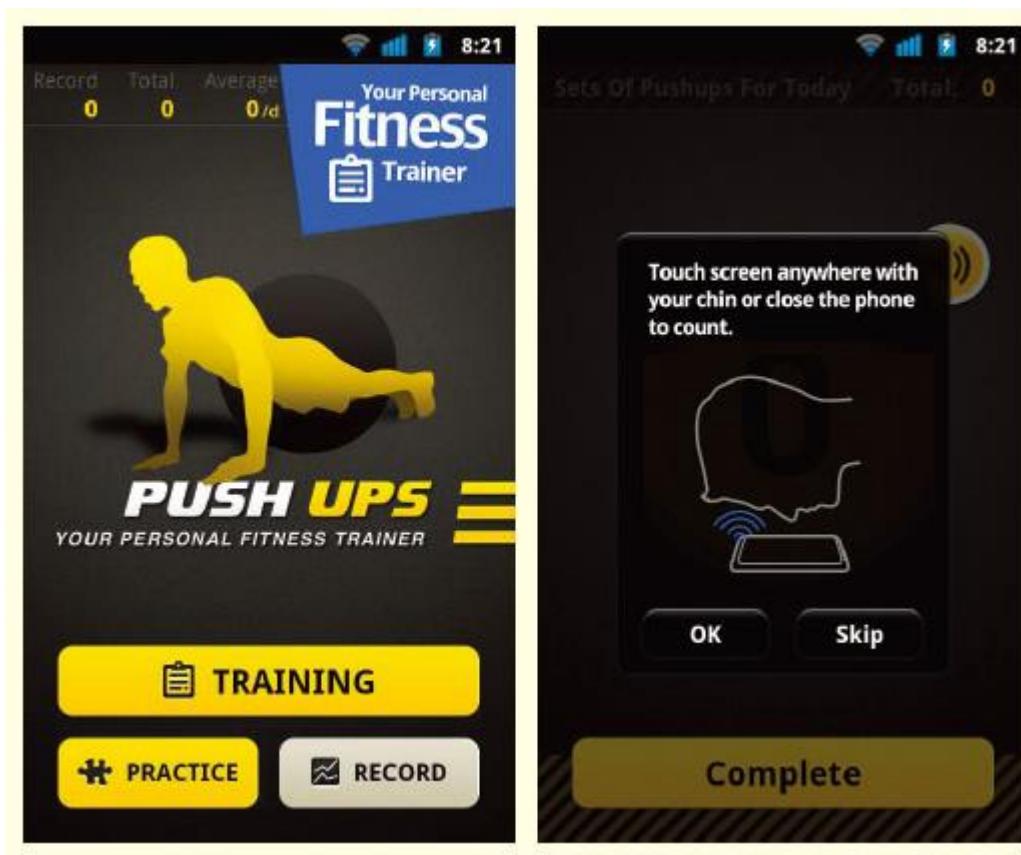
- Located at front of mobile near to front facing camera.
- Gives a reading of the light level detected by the light sensor of the device. As the light level changes, the sensor reading change.
- The units of the data are in SI lux units.
- Range is from 0 to maximum value for sensor.



# Usages!

→ *근접센서.*

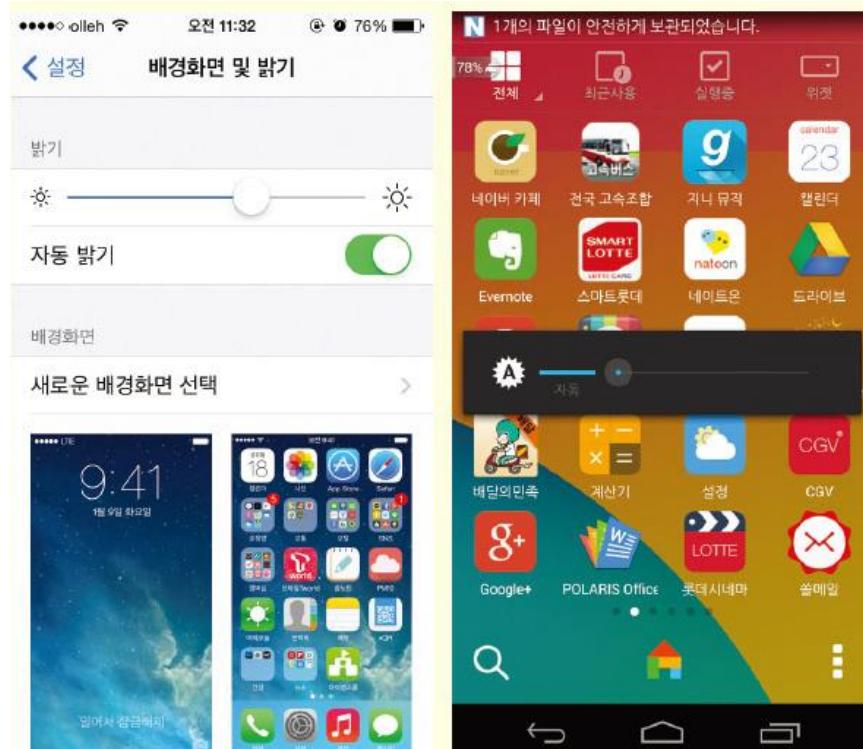
- Proximity sensor





# Usages!

- Light sensor





# Usages!

- Gravity sensor





# Usages!

- Gyroscope sensor





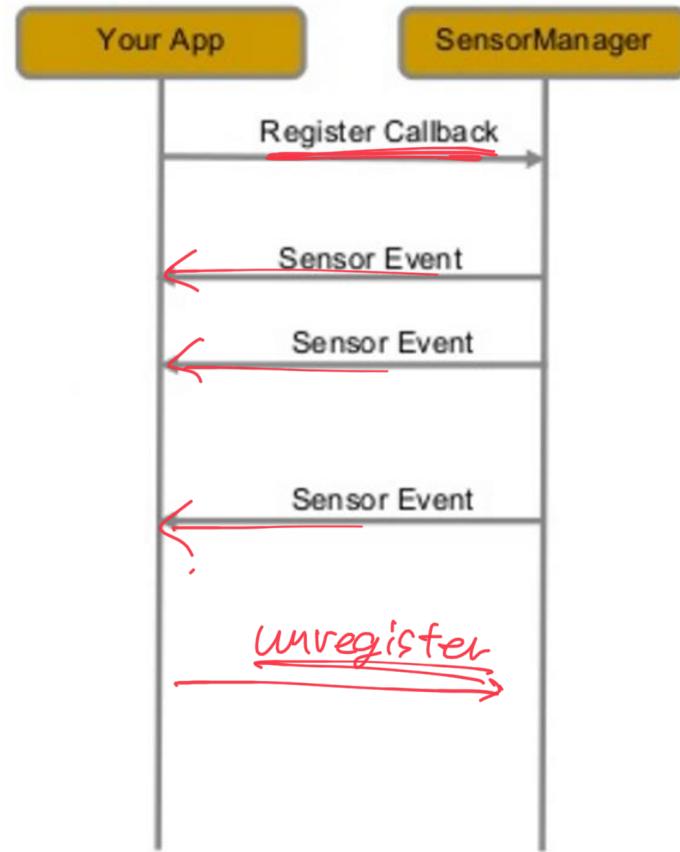
# Sensor Framework

- Android has provided sensor Framework managed by following classes
  - Sensor → HW를 대변하는 Class
  - SensorManager → 서비스 관리.
  - SensorEvent
  - SensorEventListener
- Framework has provided call back to obtain sensor data.



# Sensor Framework

- SensorEventListener  
async call back  
*비동기*
- Call back retrieve data at different duration
  - Normal 20ms
  - Game 20ms
  - Fast 0ms
- Accuracy of retrieved data can be Low, Medium or High





# Sensor Framework

- Create SensorManager

```
msM = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
```

- Create Sensor Object

```
mySensor = msM.getDefaultSensor (SENSOR.TYPE_GRAVITY);
```

- TYPE VALUE : TYPE\_ACCELEROMETER, TYPE\_GYROSCOPE, TYPE\_GRAVITY, TYPE\_LIGHT, etc..

- Make Sensor Listener

```
public SensorEventListener mySensorListener = new SensorEventListener(){  
    @Override  
    public void onSensorChanged(SensorEvent event){  
        mTx1.setText(Float.toString(event.values[0]));  
        Log.i("TAG", Float.toString(event.values[0]));  
    }  
    @Override  
    public void onAccuracyChanged(Sensor sensor, int accuracy){}  
};
```

↳ 정밀도가 - 코드 or OS에 의해 변경되었을 때.



# Sensor Framework

- How to Get a type of sensor

```
if(event.sensor.getType()==Sensor.TYPE_GRAVITY){....}
```

어떤 센서인지

- How to retrieve the value of sensor

```
mSM.registerListener(mySensorListener, mySensor,  
SensorManager.SENSOR_DELAY_NORMAL);
```

EventListener 등록

GAME, ...

- registerListener(센서리스너클래스, 센서 객체, 리스너의 반응속도)
- 반응속도 : SENSOR\_DELAY\_FASTEST, GAME, UI, NORMAL

- Release sensor Listener

```
@Override  
protected void onPause(){  
    super.onPause();  
    mSM.unregisterListener(mySensorListener);  
}
```

//

해제.



# Exercise

- activity\_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
    <TextView  
        android:id="@+id/textView2"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_alignParentLeft="true"  
        android:layout_alignParentTop="true"  
        android:text="Gravity Values"  
        android:textAppearance="?android:attr/textAppearanceLarge" />  
  
    <TextView  
        android:id="@+id/textView1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_alignParentLeft="true"  
        android:layout_below="@+id/textView2"  
        android:text="NO RETURN" />  
  
</RelativeLayout>
```



# Exercise

- MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    private TextView mTxt1;  
    private SensorManager mSM;  
    private Sensor myGravity;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        mTxt1 = (TextView) findViewById(R.id.textView1);  
        mSM = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
        myGravity = mSM.getDefaultSensor(Sensor.TYPE_GRAVITY);  
        mSM.registerListener(mySensorListener, myGravity,  
            SensorManager.SENSOR_DELAY_NORMAL);  
    }  
  
    @Override  
    protected void onStart() {  
        super.onStart();  
        mSM.registerListener(mySensorListener, myGravity,  
            SensorManager.SENSOR_DELAY_GAME);  
    }  
}
```

```
package org.androidtown.listview.sensors;  
  
import android.content.Context;  
import android.hardware.Sensor;  
import android.hardware.SensorEvent;  
import android.hardware.SensorEventListener;  
import android.hardware.SensorManager;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.util.Log;  
import android.widget.TextView;
```

20ms

onPause → 돌아온 때!

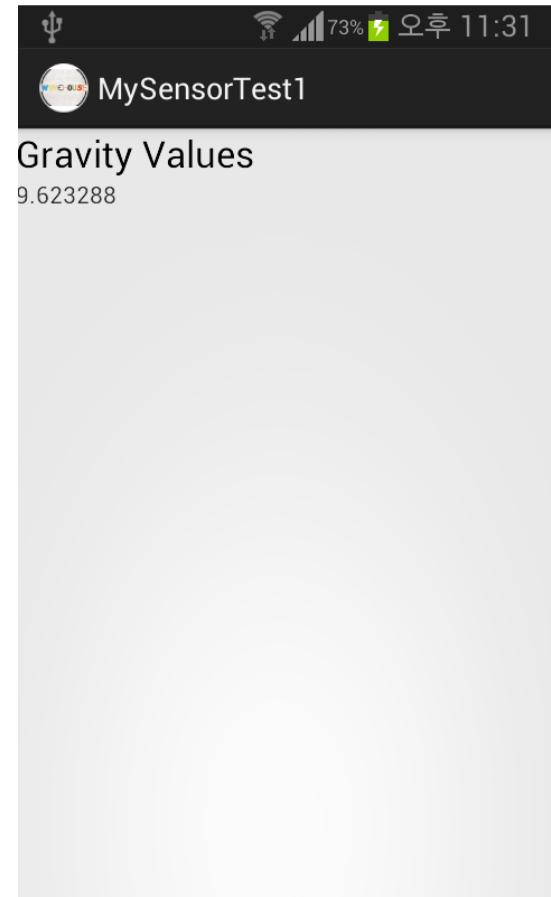
SENSOR\_SERVICE

↳ getDefaultSensor (Sensor  
· TYPE\_GRAVITY)



# Exercise

```
@Override  
protected void onPause() {  
    super.onPause();  
  
    mSM.unregisterListener(mySensorListener);  
}  
  
public SensorEventListener mySensorListener = new SensorEventListener() {  
  
    @Override  
    public void onSensorChanged(SensorEvent event) {  
  
        if(event.sensor.getType() == Sensor.TYPE_GRAVITY)  
        {  
            mTxt1.setText(Float.toString(event.values[0]));  
            Log.i("TAG", Float.toString(event.values[0]));  
        }  
    }  
  
    @Override  
    public void onAccuracyChanged(Sensor sensor, int accuracy) {}  
};  
}
```





# Do it more!

- MainActivity.java

⇒ 센서들이 무엇들이 있는지  
그동 풀어.

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        SensorManager sm = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
        List<Sensor> arSensor = sm.getSensorList(Sensor.TYPE_ALL);  
  
        String result = "갯수 : " + arSensor.size() + "\n\n";  
  
        for(Sensor s : arSensor) {  
            result += ("이름 = " + s.getName() + ",\n 형식 = " + s.getType() +  
",\n 제조사 = " + s.getVendor() +  
",\n 전원 = " + s.getPower() + ",\n 해상도 = " +  
s.getResolution() + ",\n 범위 = " +  
s.getMaximumRange() + "\n\n");  
        }  
        TextView txtResult = (TextView) findViewById(R.id.result);  
        txtResult.setText(result);  
    }  
}
```

# Do it more!



- activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/scr"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >
        <TextView
            android:id="@+id/result"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:textSize="8pt" />
    </LinearLayout>
</ScrollView>
```

