

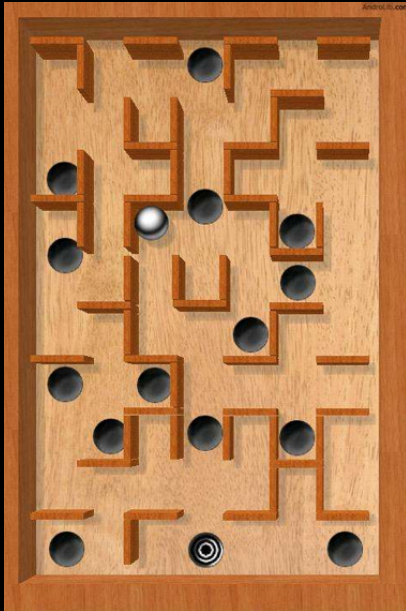
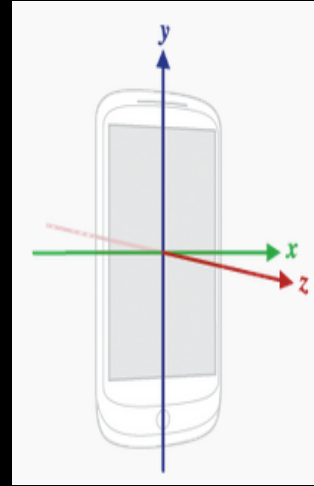
Assignment 5

Due Tonight at 11:59 PM

Motion Sensors

I couldn't think of a clever
quip this week

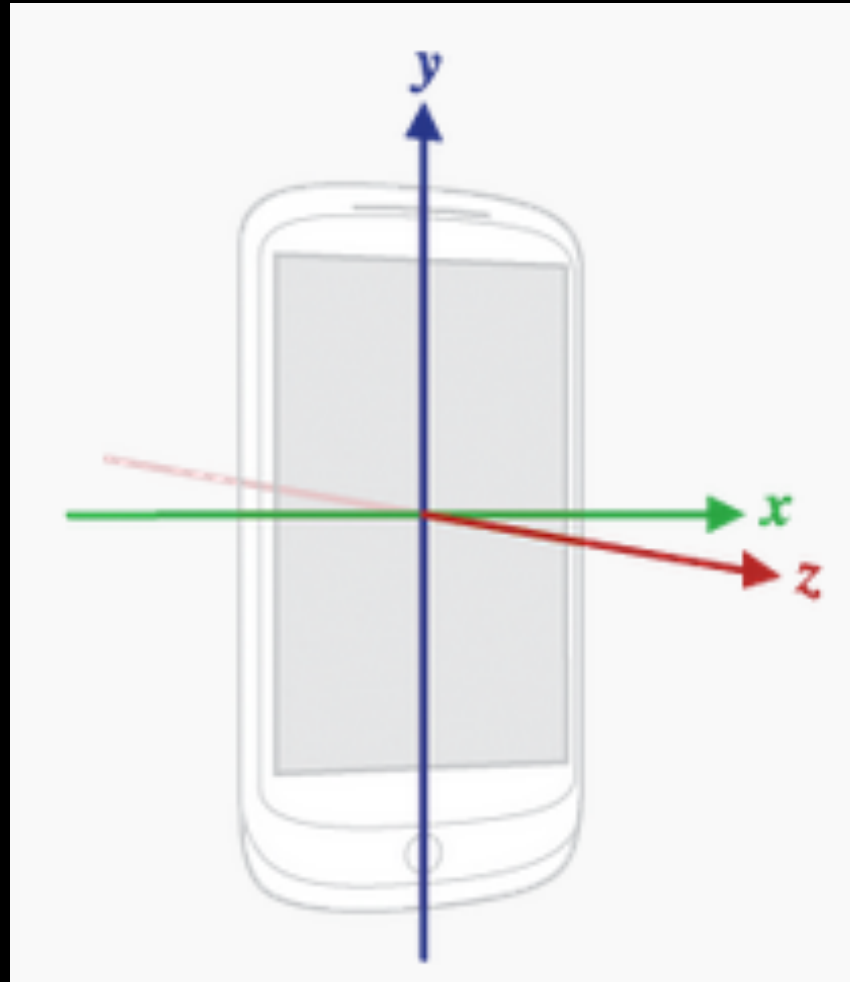
Motion Sensors



Not Just Games



Coordinate System

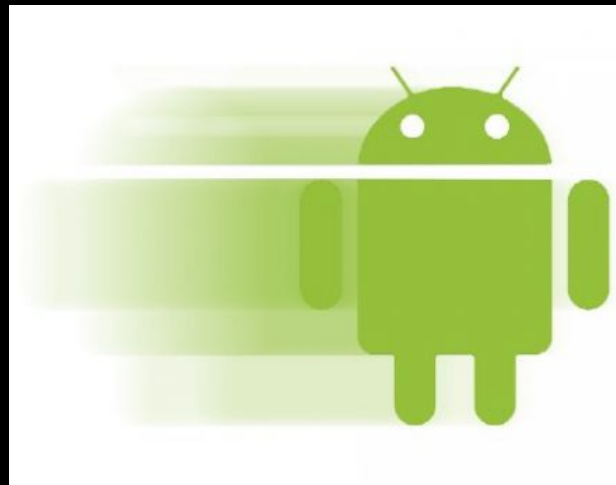


Motion Sensors

- Accelerometer
- Gyroscope
- Gravity
- Linear Acceleration
- Rotation Vector

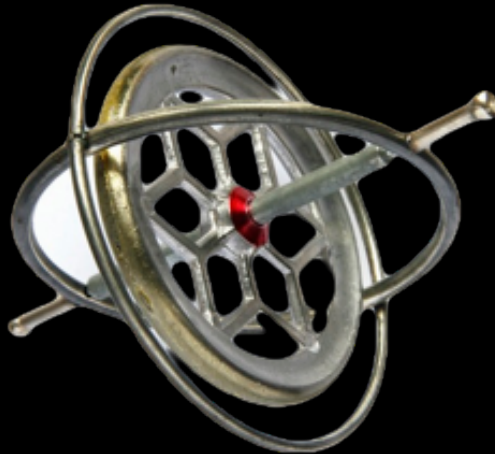
Accelerometer

- Measures acceleration forces on each axis
- Always hardware based
- Doesn't filter out gravity
- Values in m/s^2



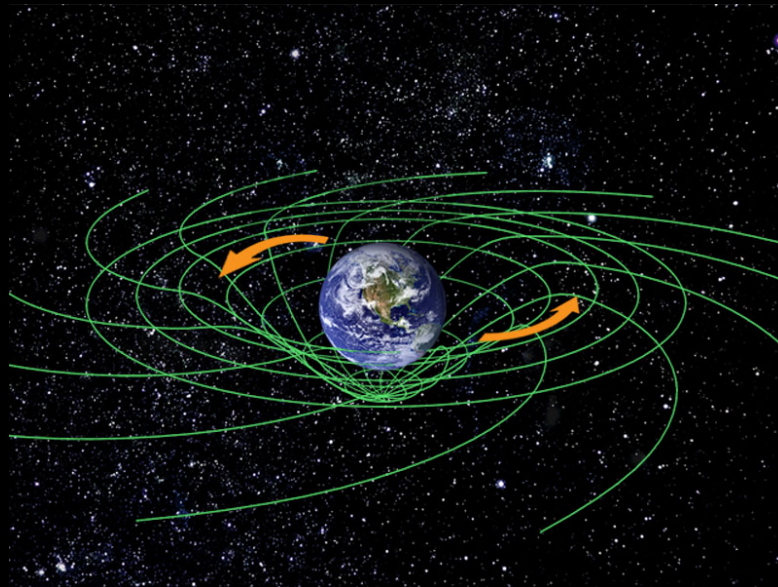
Gyroscope

- Measures rate of rotation on each axis
- Always hardware based
- All values zero at rest
- Measured in Rad/s



Gravity

- Measures the force of gravity along each axis
- May be software supported using accelerometer and gyroscope
- Magnitude should always be 9.81 m/s^2
- m/s^2



Linear Acceleration

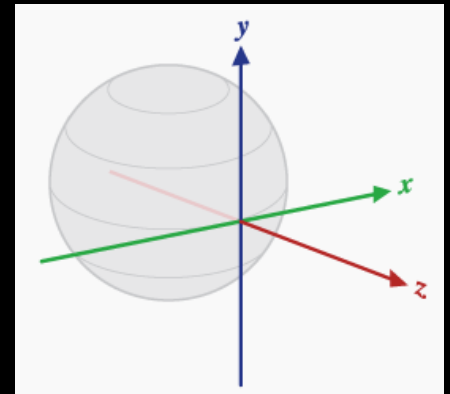
- Measures acceleration along each axis **EXCLUDING** gravity
- May be software supported via Accelerometer and Gravity sensors
- Measured in m/s^2



Rotation Vector

- Measures components of rotation vector
- Uses a slightly different coordinate system
- Unitless (vector)

- X is defined as the vector product $Y \times Z$. It is tangential to the ground at the device's current location and points approximately East.
- Y is tangential to the ground at the device's current location and points toward the geomagnetic North Pole.
- Z points toward the sky and is perpendicular to the ground plane.



Accessing Sensors

- Accessed via SensorManager
- All represented using Sensor class and accessed via SensorEventListener

```
private SensorManager mSensorManager;  
private Sensor mSensor;  
...  
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ROTATION_VECTOR);
```

Processing Sensor Data

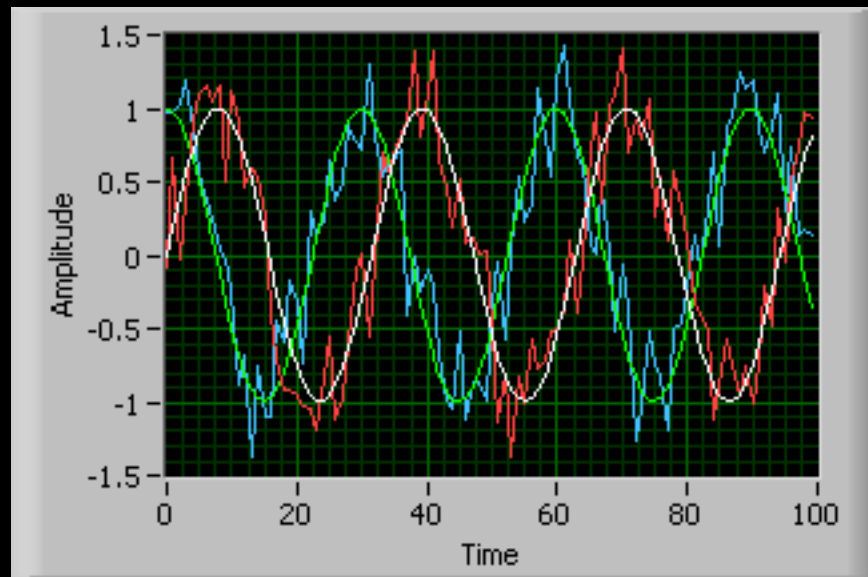
- All sensors provide 3 float values.
- Meaning of these values based on sensor.

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

    float x, y, z;
    x = event.values[0];
    y = event.values[1];
    z = event.values[2];
}
```

High Pass Filter

- Sensors can have lots of noise.
- Filter smooths out highs and lows.



High Pass Filter

- Applies a filter based on previous values
- Prevents rapid jumps in values
- Overly simple smoothing function

```
float filter = 0.1f;

filteredValues[0] = (event.values[0] * filter) + (filteredValues[0] * (1 - filter));
filteredValues[1] = (event.values[1] * filter) + (filteredValues[1] * (1 - filter));
filteredValues[2] = (event.values[2] * filter) + (filteredValues[2] * (1 - filter));

String unit = units[mode];
value1.setText("" + format.format(filteredValues[0]) + " " + unit);
value2.setText("" + format.format(filteredValues[1]) + " " + unit);
value3.setText("" + format.format(filteredValues[2]) + " " + unit);
```

Questions?