Accelerometer

Introduction

- Similarly to the iPhone, the android device is already bundled with an accelerometer.
- Using the accelerometer we can detect the movement and the rate of change in the speed of that movement.
- We cannot use the accelerometer for measuring the speed.

The `SensorManager` Class

- The `SensorManager` class allows us to get access to the various sensors on our handset.
- In order to get a `SensorManager` object we should call the `Context.getSystemService()` method passing over the `SENSOR_SERVICE` constant.

```
...
SensorManager sm = (SensorManager) getSystemService(SENSOR_SERVICE);
...
```

The `SensorManager` Class

- Once we get a `SensorManager` object we can register it with a `SensorEventListener` object.

```
...
SensorManager sm = (SensorManager) getSystemService(SENSOR_SERVICE);
SensorListener listener = new MySensorListener();
sm.registerListener(listener, SENSOR_ACCELEROMETER,SENSOR_DELAY_UI);
...
```

The third value passed over to this method must be one of the following: SENSOR_DELAY_NORMAL, SENSOR_DELAY_UI, SENSOR_DELAY_GAME or SENSOR_DELAY_FASTEST. This value indicates about the rate in which the listener will be notified about the events.

The `SensorListener` Interface

```
...
class MySensorListener implements SensorEventListener
{
    public void onAccuracyChanged(int sensor, int accuracy)
    {
        ...
    }
    public void onSensorChanged(SensorEvent event)
    {
        ...
    }
}
...
```

Accelerometer

Introduction

- Similarly to the iPhone, the android device is already bundled with an accelerometer.
- Using the accelerometer we can detect the movement and the rate of change in the speed of that movement.
- We cannot use the accelerometer for measuring the speed.

The `SensorManager` Class

- The `SensorManager` class allows us to get access to the various sensors on our handset.
- In order to get a `SensorManager` object we should call the `Context.getSystemService()` method passing over the `SENSOR_SERVICE` constant.

```
...
SensorManager sm = (SensorManager) getSystemService(SENSOR_SERVICE);
...
```

The `SensorManager` Class

- Once we get a `SensorManager` object we can register it with a `SensorEventListener` object.

```
...
SensorManager sm = (SensorManager) getSystemService(SENSOR_SERVICE);
SensorListener listener = new MySensorListener();
sm.registerListener(listener, SENSOR_ACCELEROMETER,SENSOR_DELAY_UI);
...
```

The third value passed over to this method must be one of the following: SENSOR_DELAY_NORMAL, SENSOR_DELAY_UI, SENSOR_DELAY_GAME or SENSOR_DELAY_FASTEST. This value indicates about the rate in which the listener will be notified about the events.

The `SensorListener` Interface

```
...
class MySensorListener implements SensorEventListener
{
    public void onAccuracyChanged(int sensor, int accuracy)
    {
        ...
    }
    public void onSensorChanged(SensorEvent event)
    {
        ...
    }
}
...
```