Determining Your Orientation

The *orientation sensors* are a combination of a built-in compass that provides the yaw (heading) and the accelerometers that help determine pitch and roll.

If you've done a bit of trigonometry, you've got the skills required to calculate the device orientation based on the accelerometer values along the three axes. If you enjoyed trig as much as I did, you'll be happy to learn that Android does these calculations for you.
The device orientation is reported along all three dimensions, as illustrated in Figure 10-2:

❑ **Heading** The heading (also bearing or yaw) is the direction the device is facing around the *Z*-axis, where 0/360 degrees is North, 90 degrees is East, 180 degrees is South, and 270 degrees is West.

❑ **Pitch** Pitch represents the angle of the device around the *Y*-axis. The tilt angle returned shows 0 degrees when the device is fl at on its back, –90 degrees when standing upright (top of device pointing at the ceiling), 90 degrees when the device is upside down, and 180/–180 degrees when the device is face down.

❑ **Roll** The roll represents the device's sideways tilt between –90 and 90 degrees on the *X*-axis. The tilt is 0 degrees when the device is fl at on its back, –90 degrees when the screen faces left, and 90 degrees when the screen faces right.

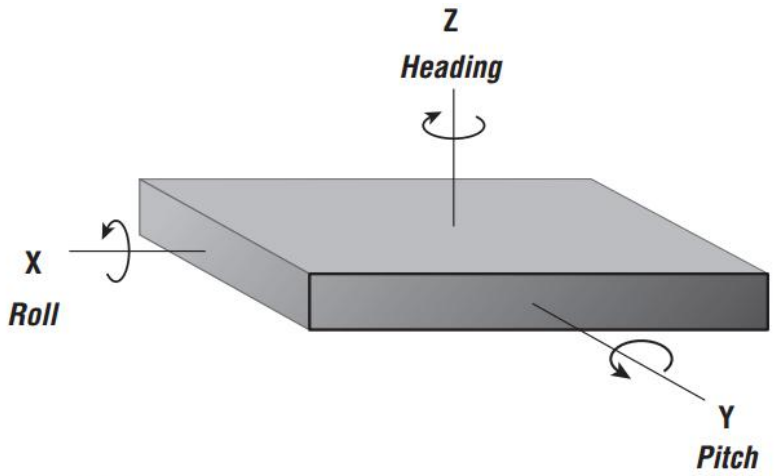

Figure 10-2

As implied by the preceding list, the Sensor Manager considers the device at rest (heading, pitch, roll at 0 degrees) when it is fl at on its back. To monitor device orientation, register a Sensor Listener with the Sensor Manager, specifying the SENSOR_ORIENTATION flag, as shown in the following code snippet:

```
SensorManager sm = (SensorManager)getSystemService(Context.SENSOR_SERVICE);
sm.registerListener(myOrientationListener,
SensorManager.SENSOR_ORIENTATION,
SensorManager.SENSOR_DELAY_NORMAL);
```

The onSensorChanged method in your SensorListener implementation will receive a float array containing the current orientation, along the three axes described above, whenever the device's orientation changes.

Within this float array, use the Sensor Manager constants DATA_X, DATA_Y, and DATA_Z to fi nd the roll, pitch, and heading (yaw) respectively. Use the corresponding RAW_DATA_* constants to find the unsmoothed / unfi ltered values as shown in the following code snippet:

```
SensorListener myOrientationListener = new SensorListener() {
public void onSensorChanged(int sensor, float[] values) {
if (sensor == SensorManager.SENSOR_ORIENTATION) {
float rollAngle = values[SensorManager.DATA_X];
float pitchAngle = values[SensorManager.DATA_Y];
```

```
float headingAngle = values[SensorManager.DATA_Z];
float raw_rollAngle = values[SensorManager.RAW_DATA_X];
float raw_pitchAngle= values[SensorManager.RAW_DATA_Y];
float raw_headingAngle = values[SensorManager.RAW_DATA_Z];
// TODO apply the orientation changes to your application.
}
}
public void onAccuracyChanged(int sensor, int accuracy) { }
};
```