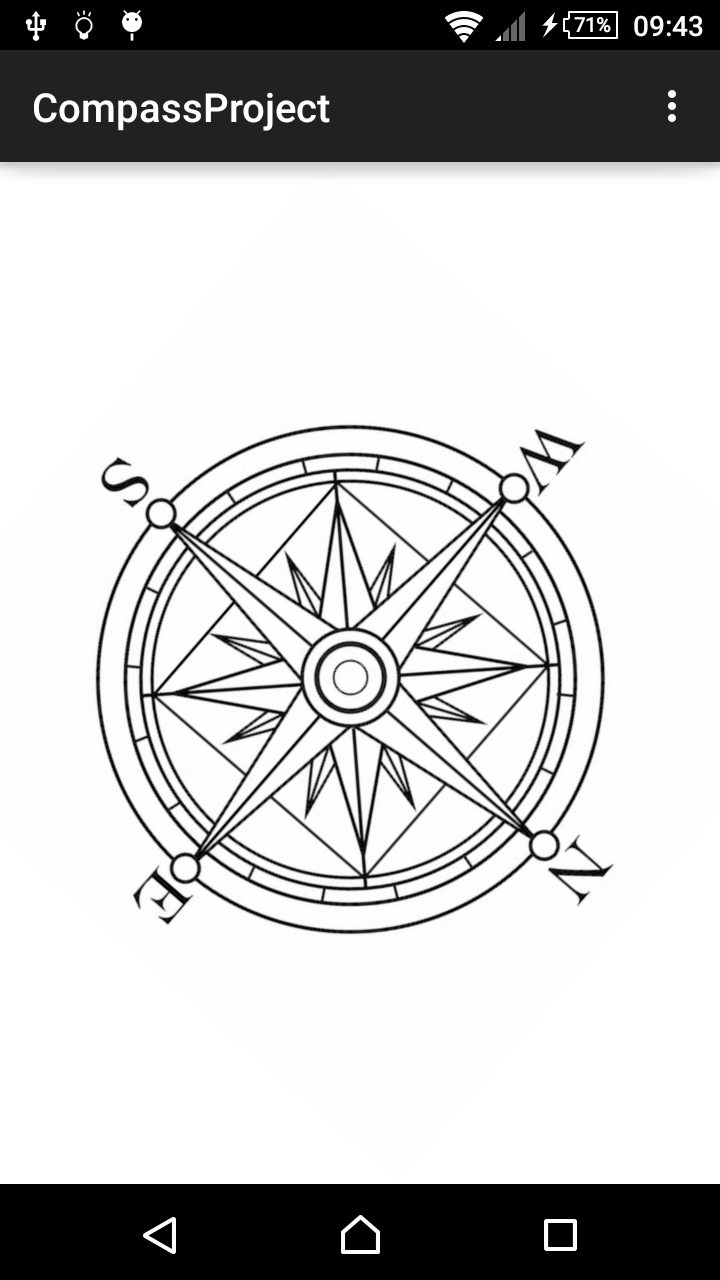
## Exercise 6.1: A Compass App

Following this exercise, you will build a small compass app that uses the sensors of a mobile device. For showing the compass we will use an image that is displayed with the letter “N” always pointing north.



Start with a new project and take the following steps:

* Load the file *compass.jpg* from the exercise materials and copy it via copy & paste to *res/drawable*.
* Create a layout *activity\_main.xml* similar to:

<**FrameLayout**

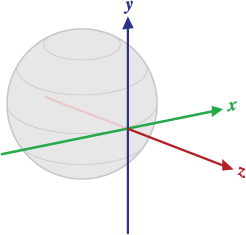
**xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" tools:context=".MainActivity"  
 android:background="#ffffffff"**>  
  
 <**ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/compass\_image"  
 android:layout\_gravity="center"  
 android:src="@drawable/compass"** />  
</**FrameLayout**>

* Register for sensor Sensor.TYPE\_ORIENTATION.

This is a virtual sensor whose value is computed from two physical sensors (accelerometer and magnet field sensor).

Ignore that the sensor is marked as *deprecated*!

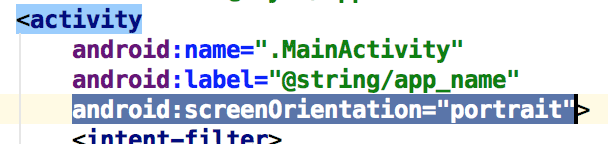
* In onSensorChanged() use the measured information.



* + The first of the three obtained values (values[0]) is the rotation around the z-axis. The z-axis points from earth center, its rotation angle is the angle from the magnetic north pole.
  + Use the method ImageView.setRotation(), to rotate the compass image.

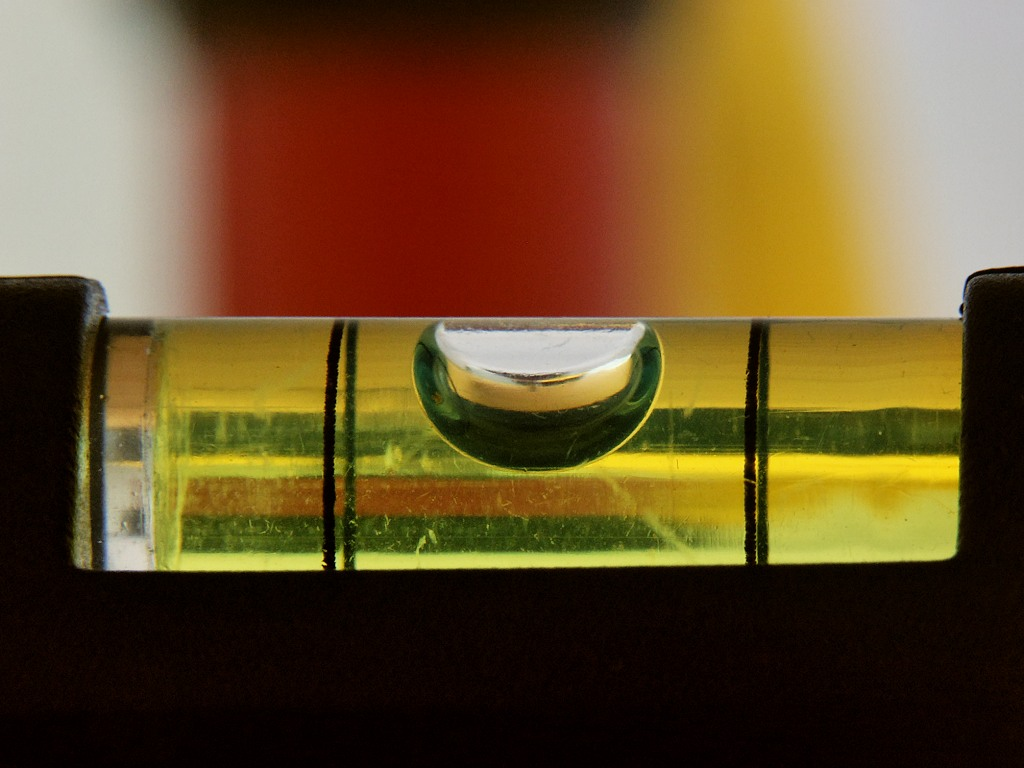
Hint: You might need to invert the value, f.e.   
setRotation(-values[0])!

* To prevent your UI from rotation from portrait to landscape when moving the phone add the attribute android:screenOrientation="portrait" to your activity in AndroidManifest.xml



* It is best to test your app on a **real device**. The emulator allows to simulate sensors in a limited way (in emulator select … 🡪 Virtual sensors), but for the magnetometer this functionality does not seem to work right now.

## Exercise 6.2: Write a Spirit Level App



A spirit level (dt. “Wasserwaage”) is a measuring instrument that indicates whether a surface is horizontal (level). Traditional instruments use a tube with a small bubble inside a fluid.

In this little project you write your own spirit level app!

To do so, you will need to use the TYPE\_ACCELEROMENTER sensor.

You are free to implement the UI in an individual way (increasing difficulty), for example:

* Use text-output
* Use two linear sliders (see theft detector from lecture) in horizontal and vertical direction to visualize the measurement.
* Build the visualization using each an image for the background and an image for the “bubble”
* Build a 2D-visualization with a bubble moving along two axes

You may also get inspired by searching for existing apps on Google Play: <https://play.google.com/store/search?q=spirit%20level>