# Practical 3: Tricorder

The principle class of the Tricorder application is TricorderActivity class

```
public class TricorderActivity
        extends MapActivity implements SensorEventListener
{
  private MapView map_view;
  private MapOverlay map_overlay;
  private MapController map_ctrl;

  private SensorManager sensor_manager;
  private Sensor sensor_magnet;

  private LocationManager loc_manager;
  private LocationListener loc_listener;
  private Location my_location;

  private MediaRecorder media_recorder;

  ...
}
```

The map_view is part of the Google Map API, I've implemented it together with an overlay (map_overlay) that gives ability to draw custom graphics, and a map controller (map_ctrl) that supports a method to centre the map on ones current position. sensor_manager together with sensor_magnet indicates that I am measuring magnetic field strength, in this application the magnitude of field strength in the x and y directions (relative to the phone) is used to determine the geographic direction that the phone is pointing. The Location related classes together listen for GPS updates, loc_listener is written to centre the map on the (lat,lon) position on each update. Finally, media_recorder is used to make continuous measurements on the ambient volume.

Figure 1 shows a screenshot. Initially, the application will centre the map-view on the last known gps position, or if one cannot be found it will temporarily centre the application on the Computer Lab. At once GPS updates are found the map will continue to remain centred and coordinates are drawn near the top-left. A small blue circle is also drawn on the current location. Based on the x and y field-strengths the angle with respect to the x-axis can be calculated based on the following conversions.

$$\theta = tan^{-1}(x/y) \quad (1)$$

$$\theta = \pi + tan^{-1}(x/y) \quad (2)$$

$$\theta = \pi + tan^{-1}(x/y) \quad (3)$$

$$\theta = 2\pi + tan^{-1}(x/y) \quad (4)$$

Depending on whether the angle lies in quadrant (1), (2), (3) or (4) as the equations are numbered, and shown in red in figure 2. The angle $\theta$ together with the volume and the (lat,lon) positions are all written to the map overlay (from various parts of the TricorderActivity) which is programmed to redraw on each update of the magnetic sensor, as follows

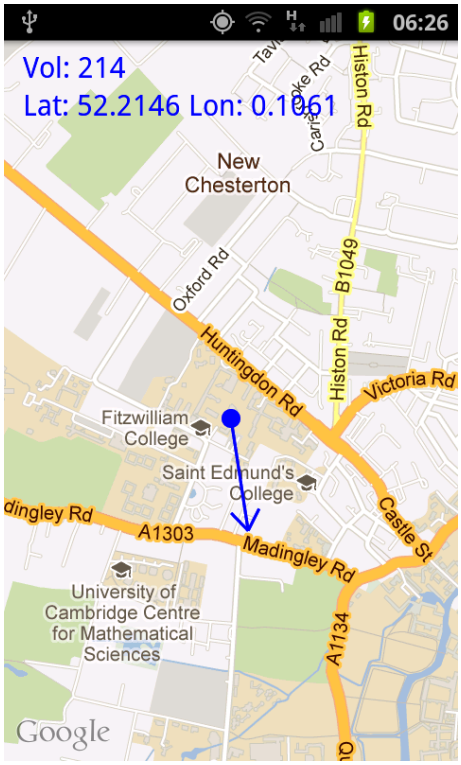


Figure 1: The Tricorder shows position on Google map, direction that the phone is pointing, and a measure of the ambient noise.

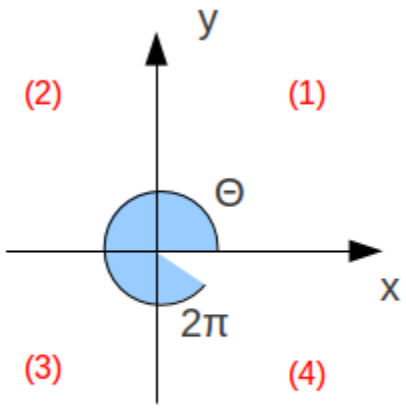


Figure 2: A trigonometric conversion is necessary to find the angle of rotation relative to the x-axis (between 0 and $2\pi$).

```
map_overlay.lat = my_location.getLatitude();
map_overlay.lon = my_location.getLongitude();
...

map_overlay.volume  = media_recorder.getMaxAmplitude();
...

map_overlay.angle = angle;
map_view.postInvalidate();
```

The map_overlay was added to the map view's list of overlays (List<Overlay> mapOverlays) earlier in the code. postInvalidate() is method the belongs to the Google API which will first redraw the Google Map, and then draw all the overlays (in order) that belongs to the list (in my case only one). All the custom graphics shown in the screenshot are organised in the overlay's draw() method.