THE UNIVERSITY OF MELBOURNE

COMP90018

Mobile Computing Systems Programming

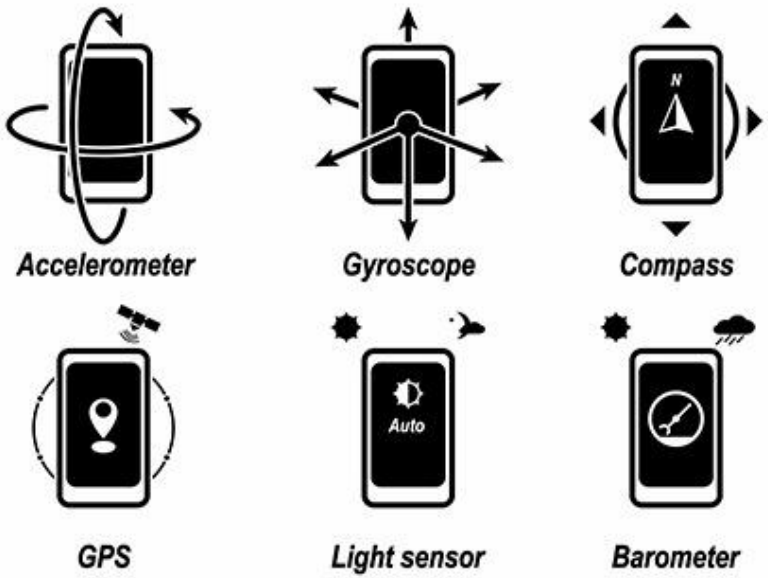
WEEK 4 – Sensors

Atiq Shaikh,

atiq.shaikh@unimelb.edu.au

Sensor

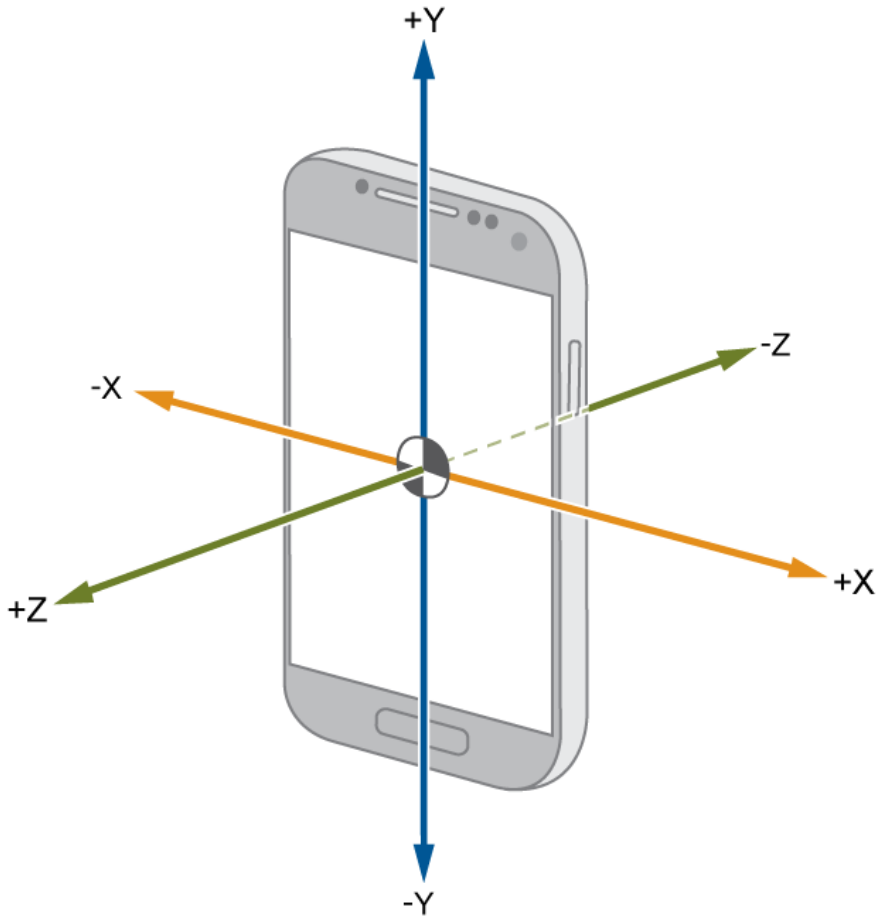
Sensor



Most Android-powered devices have ***built-in sensors*** that measure motion, orientation, and ***various environmental conditions***.

https://developer.android.com/guide/topics/sensors/sensors_overview

Sensor – Motion Sensors



Motion sensors

These sensors measure ***acceleration forces and rotational forces*** along three axes.

This category includes *accelerometers, gravity sensors, gyroscopes, and rotational vector sensors*.

https://developer.android.com/guide/topics/sensors/sensors_motion.html

Sensor – Environmental sensors

Environmental sensors

These sensors measure various ***environmental parameters***, such as ambient air temperature and pressure, illumination, and humidity.

This category includes *barometers, photometers, and thermometers*.

https://developer.android.com/guide/topics/sensors/sensors_environment.html

Sensor – Position sensors

Position sensors

These sensors measure the ***physical position of a device***.

This category includes *orientation sensors and magnetometers*.

https://developer.android.com/guide/topics/sensors/sensors_position.html

Sensors


Accelerometer
Gyroscope
Magnetometer
Barometer
Proximity
Light sensor
Touch screen
GPS
WiFi
Bluetooth
GSM/CDMA Cell
NFC: Near Field
Camera (front)
Camera (back)
Source: Internet

Sensors in detail

1. **Proximity:** Detect proximity or closeness of object to a mobile
2. **Magnetometer:** Detect magnetic field around the phone
3. **Accelerometer:** Keep track of orientation of device
4. **Gyroscope:** Keep track of the device has been rotated
5. **Camera:** Capture picture or image

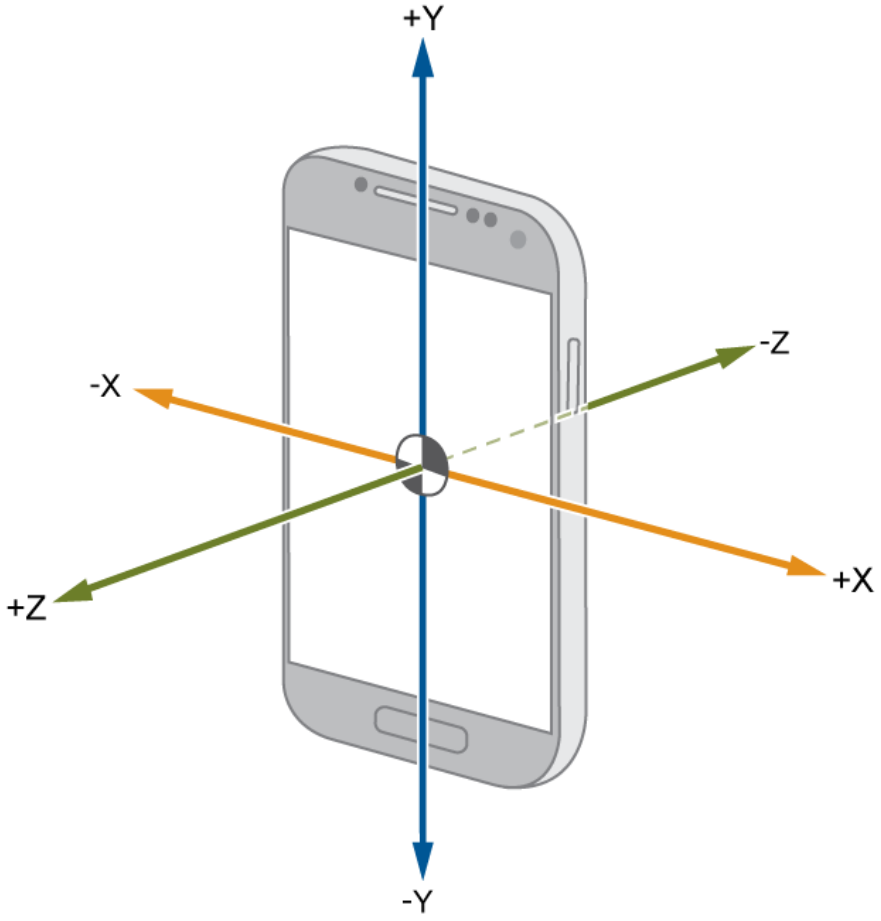
Sensors in detail

1. **Light sensor:** Measure intensity of light falling on the device
2. **Temperature sensor:** Show ambient temperature near a device
3. **Barometer:** Measure air pressure near device
4. **GPS:** Detect location of the device
5. **Compass:** Measure acceleration force that is applied to the device

THE UNIVERSITY OF MELBOURNE

Compass

Accelerometer



Measures the ***acceleration force*** in m/s^2 that is applied to a device on all three physical axes (x, y, and z), including the force of gravity.

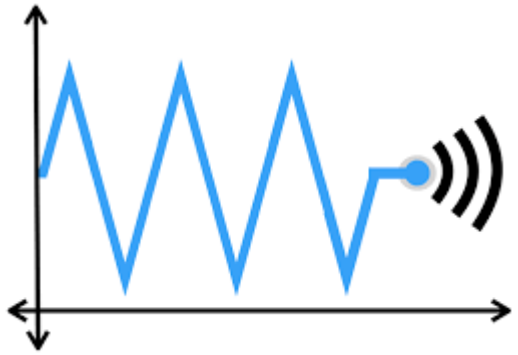
https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ACCELEROMETER

Magnetometer

Measures the ambient ***geomagnetic field*** for all three physical axes (x, y, z) in μT.

https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_MAGNETIC_FIELD

SensorEventListener



Used for ***receiving notifications*** from the ***SensorManager*** when there is new sensor data.

https://developer.android.com/reference/android/hardware/SensorEventListener

For Reading – Calculate Acceleration

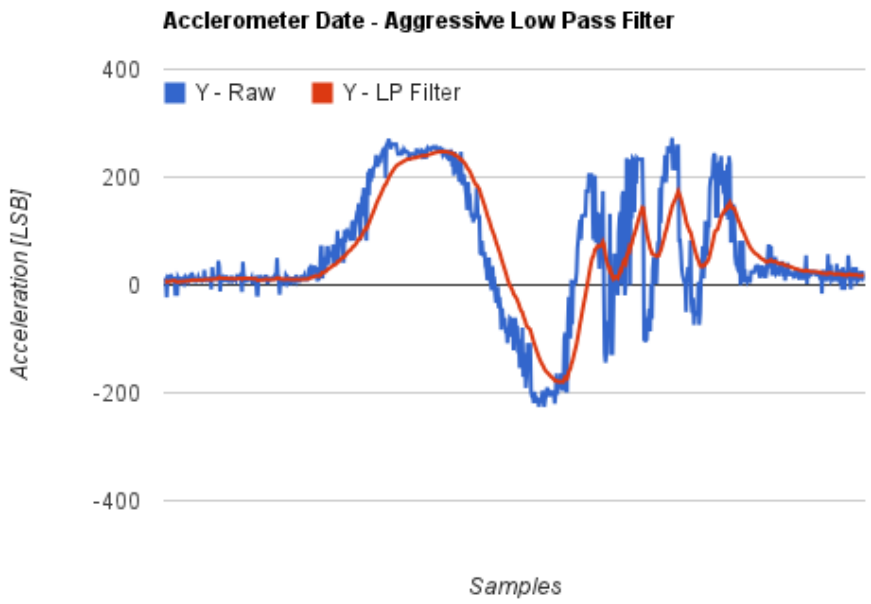
Conceptually, an acceleration sensor determines the acceleration that is applied to a device (A_d) by measuring the ***forces that are applied to the sensor itself (F_s)*** using the following relationship:

$$A_D = -(\frac{1}{mass}) \sum F_S$$

However, the ***force of gravity*** is always influencing the measured acceleration according to the following relationship:

$$A_D = -g - (\frac{1}{mass}) \sum F_S$$

For Reading – Low-Pass Filter



A low pass filter allows the low frequencies to pass while blocking the high frequencies.

http://philstech.blogspot.com/2012/04/quadcopter-accelerometer-data-filtering.html

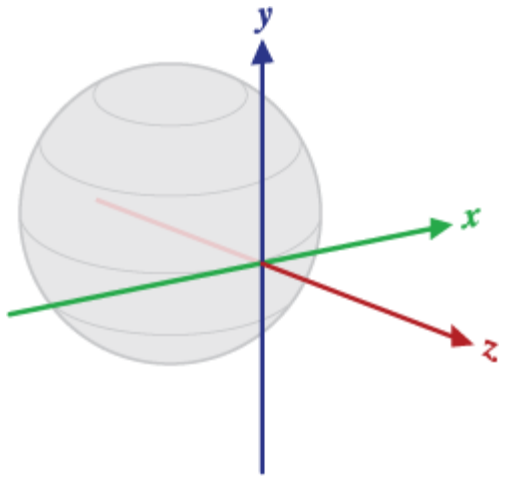
For Reading – Low-Pass/High-Pass Filter


```
public void onSensorChanged(SensorEvent event){
    // In this example, alpha is calculated as t / (t + dT),
    // where t is the low-pass filter's time-constant and
    // dT is the event delivery rate.

    final float alpha = 0.8;

    // Isolate the force of gravity with the low-pass filter.
    gravity[0] = alpha * gravity[0] + (1 - alpha) * event.values[0];
    gravity[1] = alpha * gravity[1] + (1 - alpha) * event.values[1];
    gravity[2] = alpha * gravity[2] + (1 - alpha) * event.values[2];

    // Remove the gravity contribution with the high-pass filter.
    linear_acceleration[0] = event.values[0] - gravity[0];
    linear_acceleration[1] = event.values[1] - gravity[1];
    linear_acceleration[2] = event.values[2] - gravity[2];
}
```

To measure the real acceleration of the device can be achieved by applying a ***high-pass filter***. Conversely, a ***low-pass filter*** can be used to isolate the force of gravity.

https://developer.android.com/guide/topics/sensors/sensors_motion#java

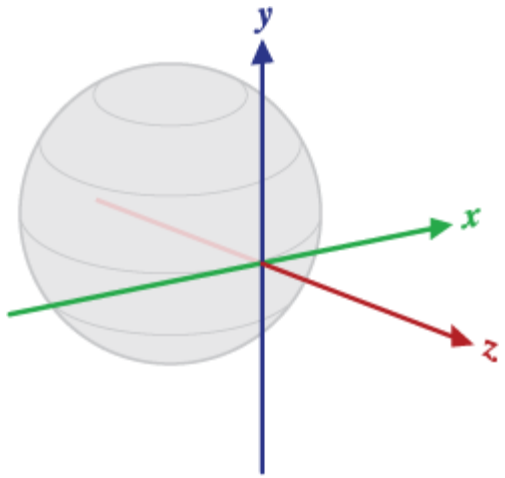
For Reading – getRotationMatrix



Computes the ***inclination matrix I*** as well as the ***rotation matrix R*** transforming a vector from the device coordinate system to the world's coordinate system.

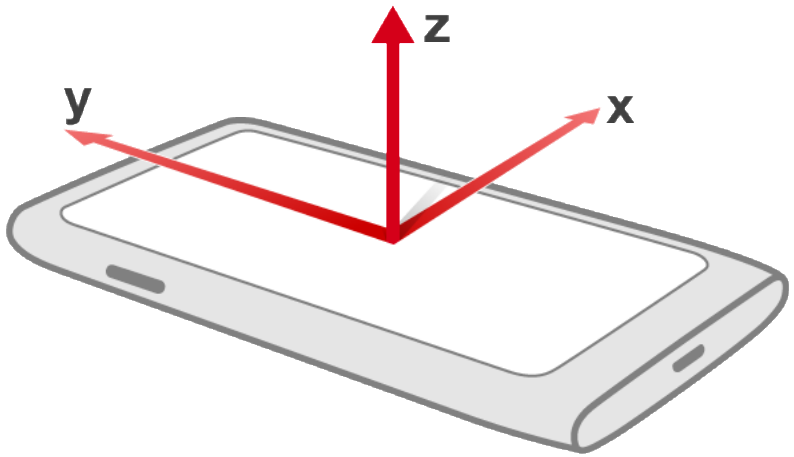
https://developer.android.com/reference/android/hardware/SensorManager.html#getRotationMatrix(float[],%20float[],%20float[],%20float[])

For Reading – getRotationMatrix



R is the identity matrix when the device is aligned with the world's coordinate system, that is, when the device's X axis points toward East, the Y axis points to the North Pole and the device is facing the sky.

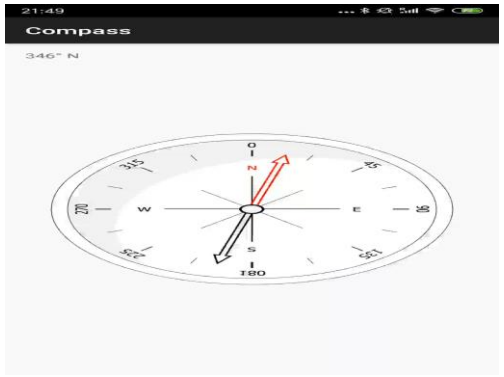
For Reading – getOrientation



Computes the ***device's orientation*** based on the ***rotation matrix***.

https://developer.android.com/reference/android/hardware/SensorManager.html#getOrientation(float%5B%5D,%2520float%5B%5D)

Compass – Demonstration



THE UNIVERSITY OF MELBOURNE

Barometer

Barometer

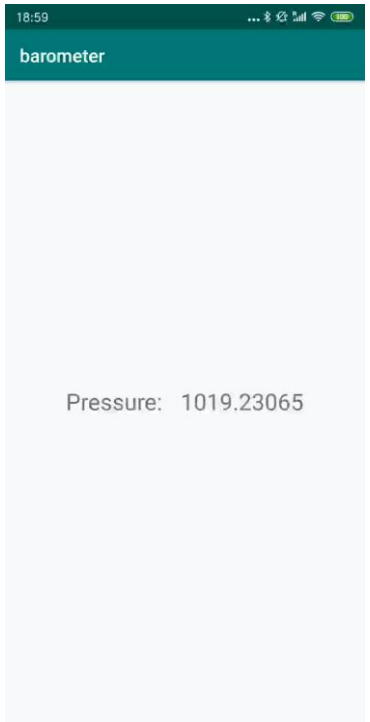
A ***barometer*** is a scientific instrument that is used to measure air pressure in a certain environment. Pressure tendency can forecast short term changes in the weather.

Barometer Sensor

Barometers are used for ***monitoring the ambient air pressure*** in hPa or mbar.

https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_PRESSURE

Barometer Demonstration



THE UNIVERSITY OF MELBOURNE

Location

Location

One of the unique features of mobile applications is ***location awareness***.

Mobile users bring their devices with them ***everywhere***, and adding location awareness to your app offers users a more ***contextual experience***.

Location Demonstration – Native

Step 1:

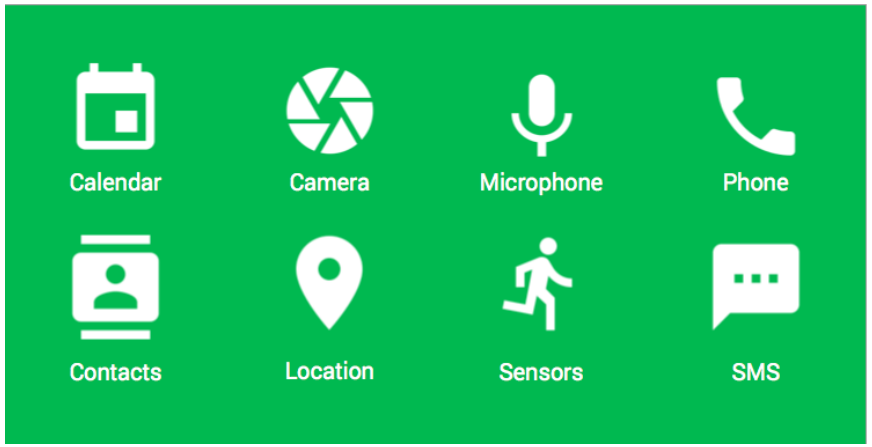
1. You need a ***LocationManager***
2. A Service does the low-level sensing : ***Context.LOCATION_SERVICE***
3. Register a ***LocationListener***

Location Demonstration – Native

Step 2 (Permission Check For Location):

1. Location Sensing requires ***Permission*** from users;
2. API 23 or above needs ***runtime Permission*** Check.

Location Demonstration – Native



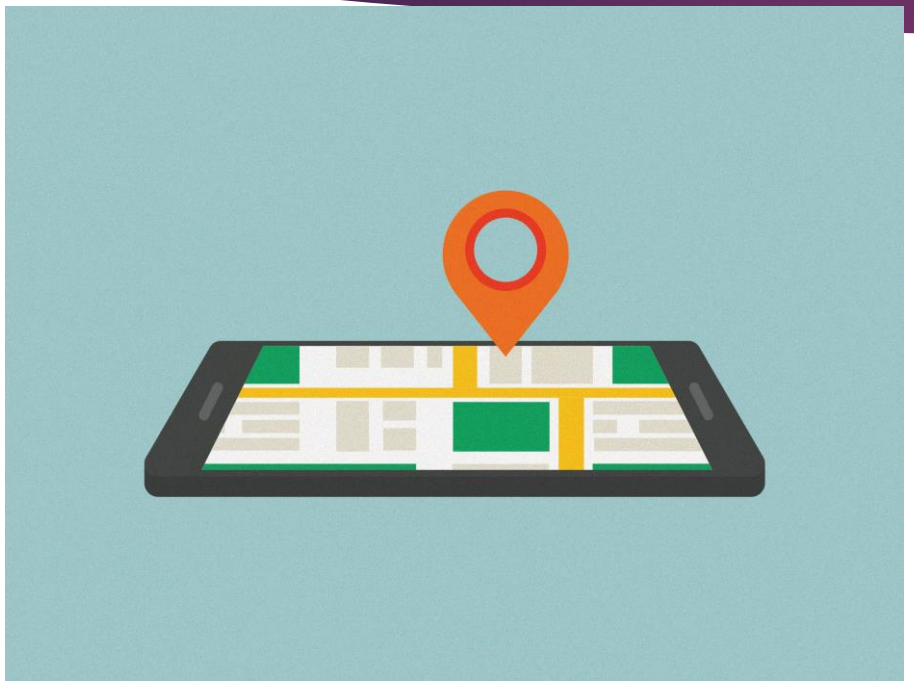
Required Perimissions:

```
<uses-feature
android:name="android.hardware.location.network"
/>

<uses-feature
android:name="android.hardware.location.gps" />
```

Location Demonstration – Native

```
2019-07-30 20:28:28.209 8521-8521/io.cluo29.github.activitylocation D/haha: both available location
2019-07-30 20:28:28.210 8521-8521/io.cluo29.github.activitylocation D/haha: latitude: -37.79904687
    longitude144.96274112
2019-07-30 20:28:28.273 8521-8521/io.cluo29.github.activitylocation D/haha: network Location latitude -37.7988979
    longitude:144.9626403
2019-07-30 20:28:48.457 8521-8521/io.cluo29.github.activitylocation D/haha: network Location latitude -37.7988602
    longitude:144.9626434
2019-07-30 20:29:09.473 8521-8521/io.cluo29.github.activitylocation D/haha: network Location latitude -37.7988441
    longitude:144.9626431
```

Location Demonstration – Google API



The ApiDemos repository on GitHub includes samples that demonstrate the ***use of location*** on a map.

https://github.com/googlemaps/android-samples/blob/master/ApiDemos

Location Demonstration – Google API

Get an API Key

☆☆☆☆☆

★ **New Users:** Before you can start using the Google Maps Platform APIs and SDKs, you must sign up and create a billing account. To learn more, see Get Started with Google Maps Platform.

To use the Maps SDK for Android you must have an API key. The API key is a unique identifier that is used to authenticate requests associated with your project for usage and billing purposes. To learn more, see the API Key Best Practices and the FAQs.

Get the API key

You must have at least one API key associated with your project.

To get an API key:

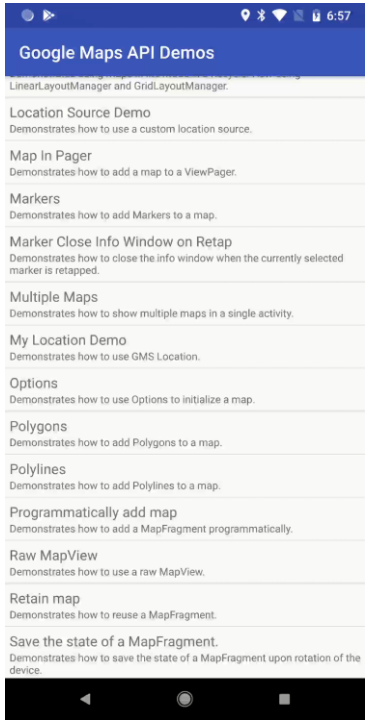
1. Go to the Google Cloud Platform Console.
2. From the Project drop-down menu, select or create the project for which you want to add an API key.
3. From the ≡ Navigation menu, select **APIs & Services > Credentials**.
4. On the **Credentials** page, click **Create credentials > API key**.
   The **API key created** dialog displays your newly created API key (an encrypted string).
5. Click **Close.**
   The new API key is listed on the **Credentials** page under **API keys**.
   (Remember to restrict the API key before using it in production.)

★ **Note:** You can use the same API key for your Maps SDK for Android and Places SDK for Android apps.

Get your own API key for ***Google Map API***:

https://developers.google.com/maps/documentation/android-sdk/get-api-key

Location Demonstration – Google API



THE UNIVERSITY OF MELBOURNE

Thank you