

Introduction to Android

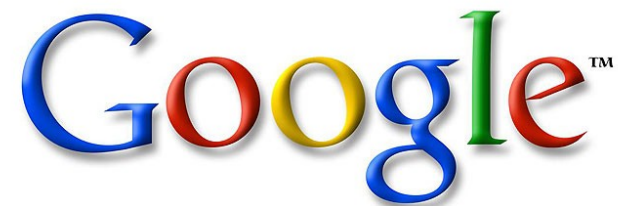
GOOGLE MAP& GPS

CS 436 Software Development on Mobile

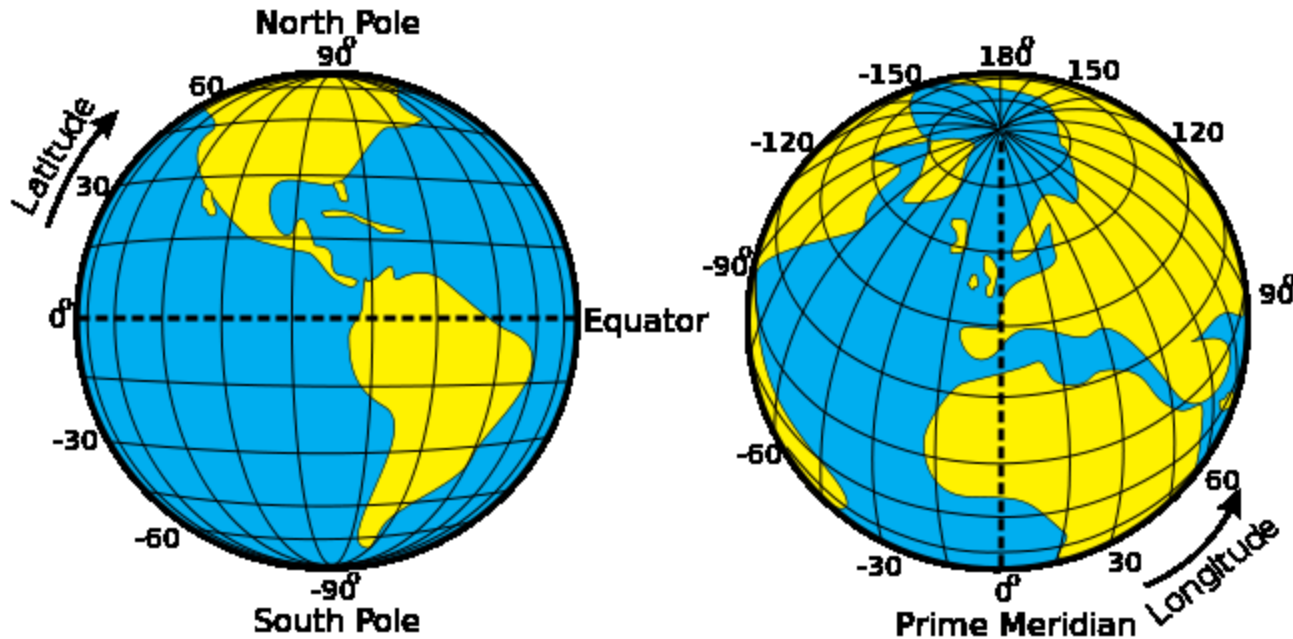
By Dr.Paween Khoenkaw



ANDROID



Geographic latitude and longitude

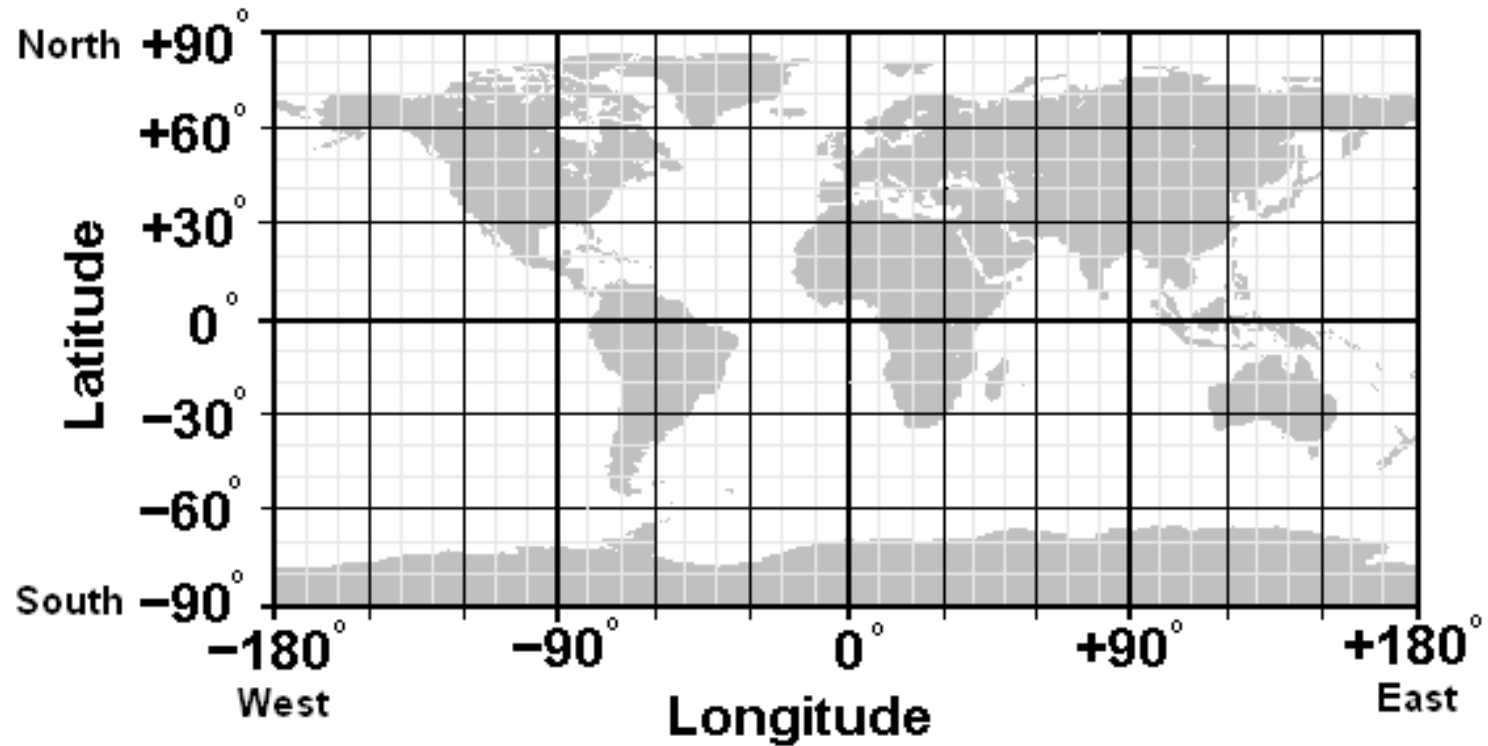


Latitude = X

Longitude=Y

Elevation=Z

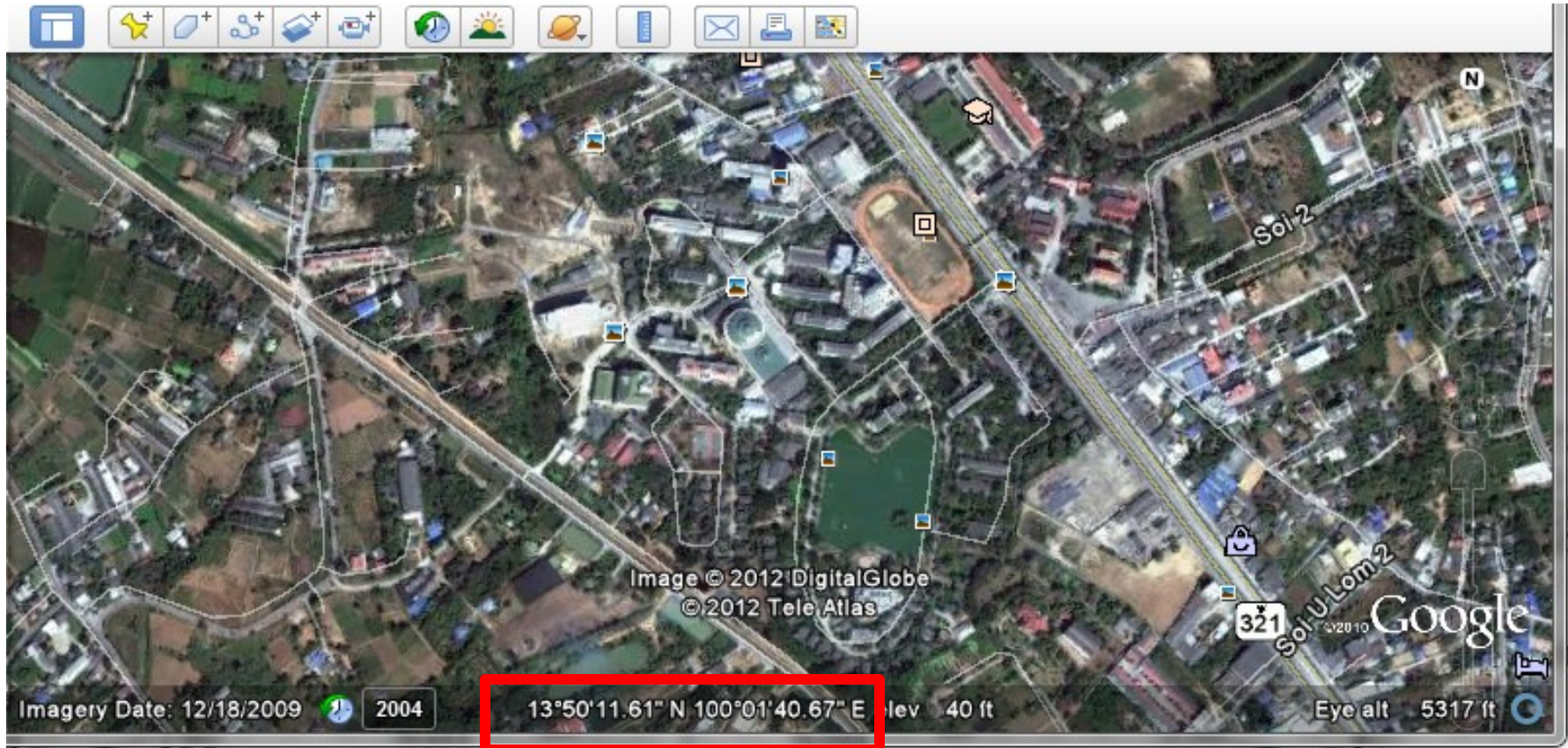
GPS and Map



www.satsig.net

Coordinate system

1) Degree Minute Second System (DMS)



13°50'11.61"N

100°01'40.67"E

Coordinate system

1) Degree Minute Second System (DMS)

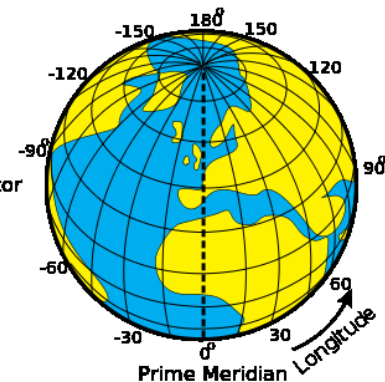
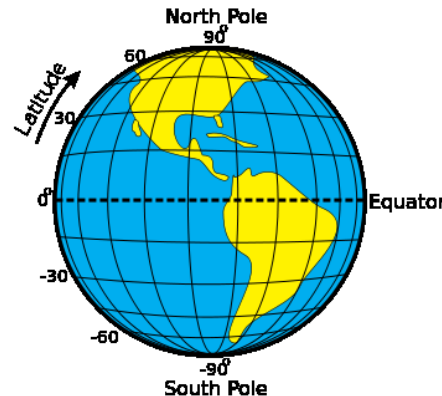
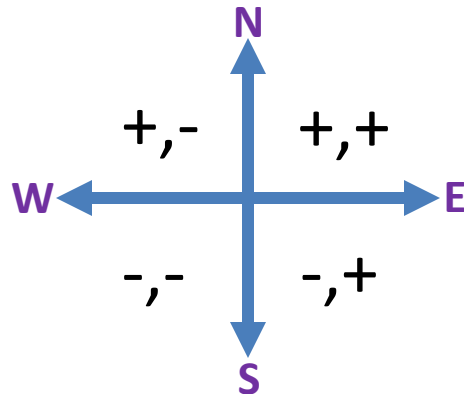
13°50'11.61"N 100°01'40.67"E

13 °

50/60 Minute

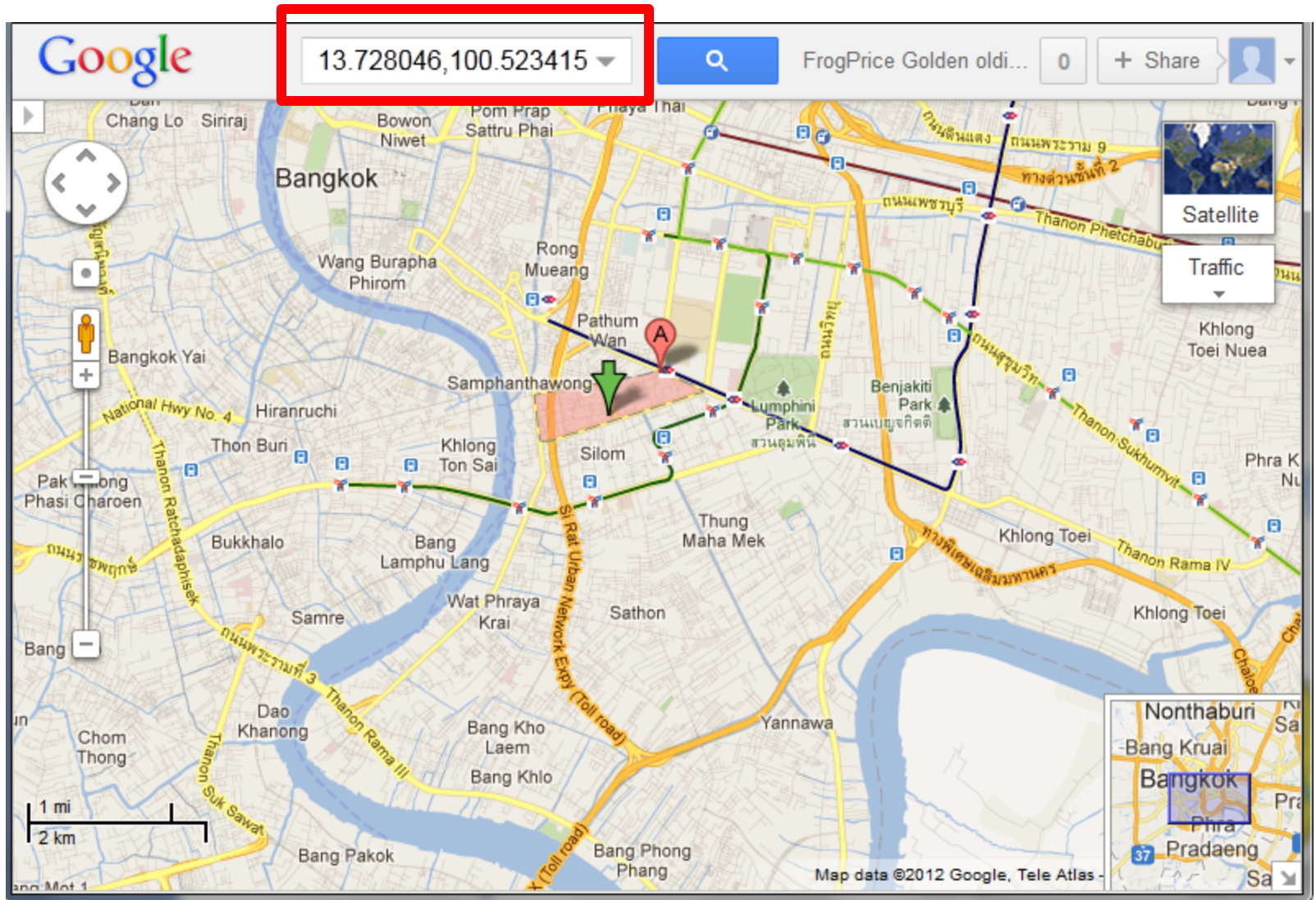
11.61/(60^2) Minute

$$\text{Degree decimal} = 13 + (50 * 60) + (11.61 / 3600) \\ = 13.336336 ^\circ$$



Coordinate system

2) Degree decimal



Coordinate system

3) Geopoint E6

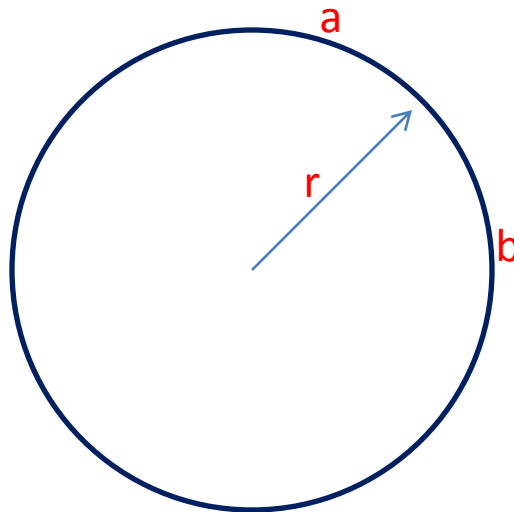
$$E6 = \left\| DegreeDecimal \times 10^6 \right\|$$

System	Value	Data type
DMS	13°50'11.61"N	String
Degree decimal	13.336336	Float
E6	13336336	Integer

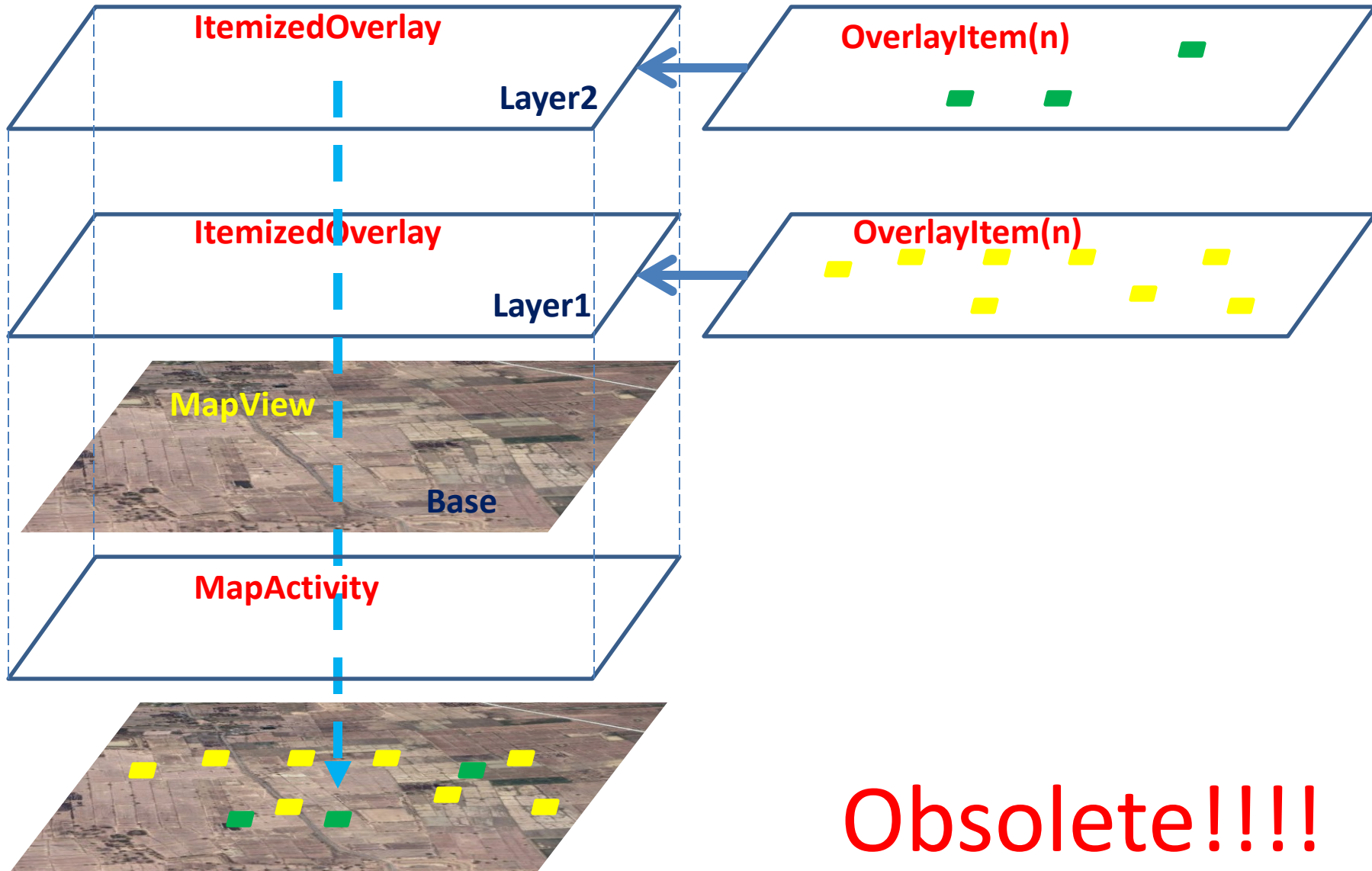
The haversine formula

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

- d is the distance between the two points
- r is the radius of the sphere,
- ϕ_1, ϕ_2 latitude of point 1 and latitude of point 2
- λ_1, λ_2 longitude of point 1 and longitude of point 2



Google API MapView



Google API MapView

Maps API key signup

```
$ keytool -list -v -alias alias_name -keystore my-release-key.keystore
```

User: android

Password:android

<http://developers.google.com/maps/documentation/android/maps-api-signup>

Obsolete!!!!

GPS & A-GPS

Global Positioning System

GPS – How it works

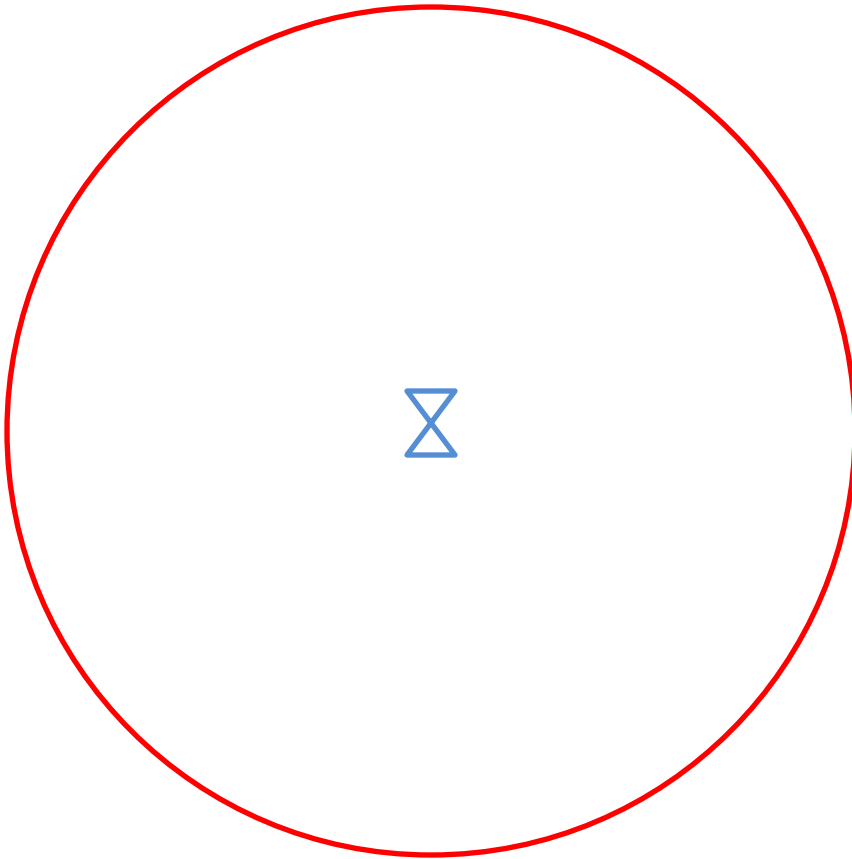
- All satellites have clocks set to exactly the same time.
- All satellites know their exact position in space from data
- Each satellite transmits its position and a time signal.
- The signals travel to the receiver delayed
- The receiver calculates the distance to each satellite



Orbit 11,000 miles above Earth
6 visible sats from any point on Earth

GPS & A-GPS

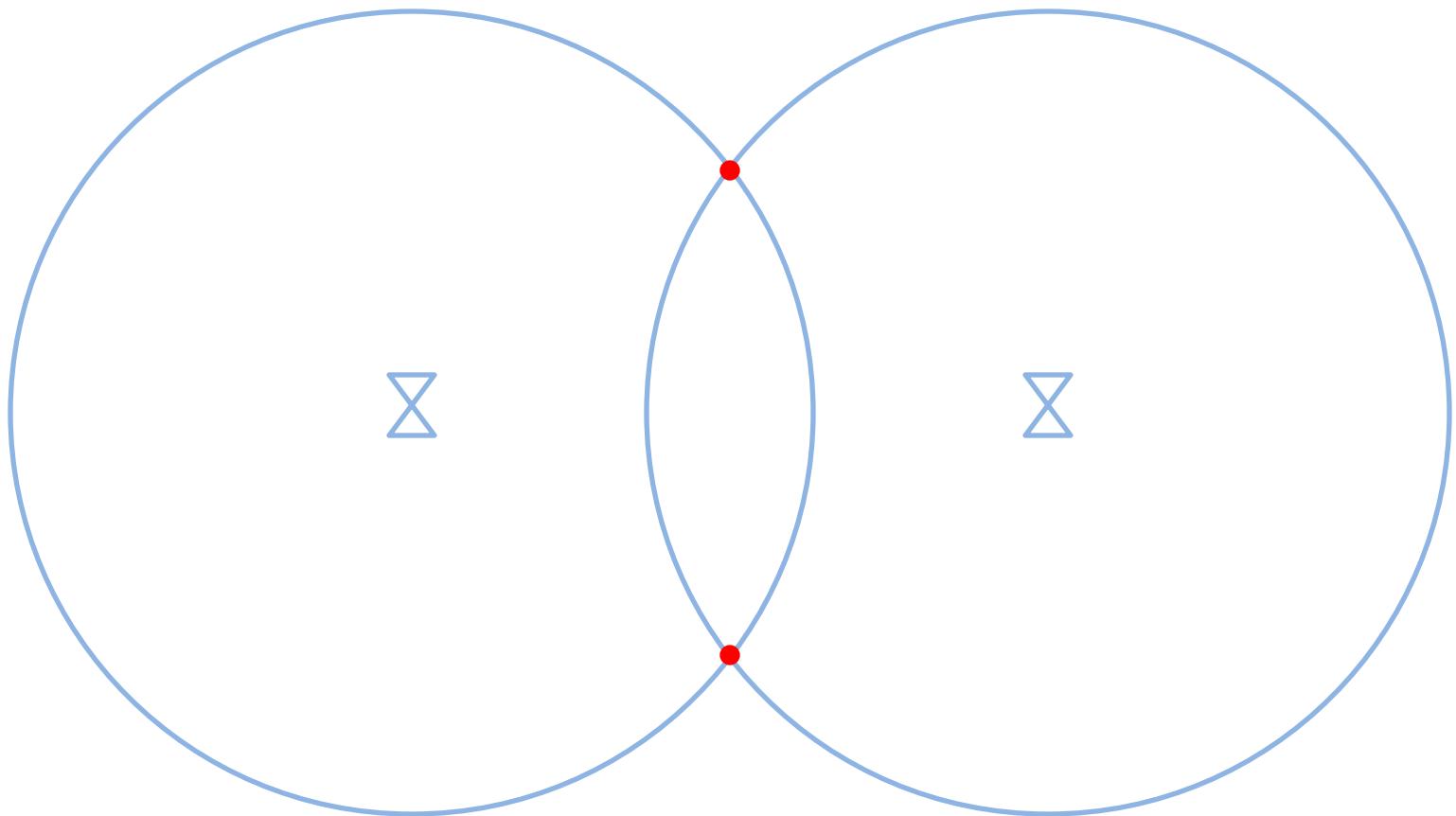
GPS Trilateration



1 Satellite

GPS & A-GPS

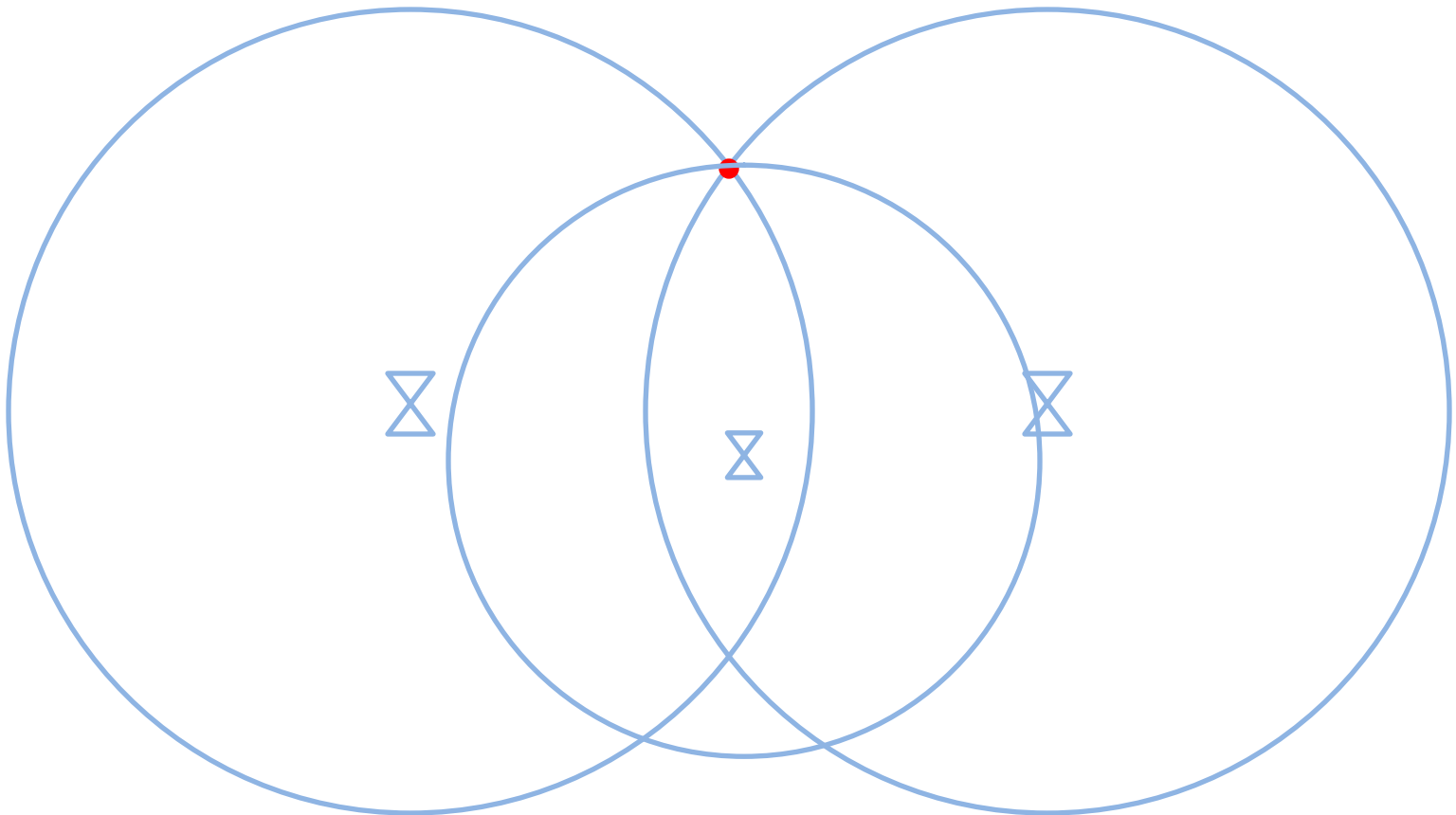
GPS Trilateration



2 Satellites

GPS & A-GPS

GPS Trilateration



3 Satellites

GPS & A-GPS

GPS Trilateration



4 Satellites

GPS & A-GPS

GPS ERROR

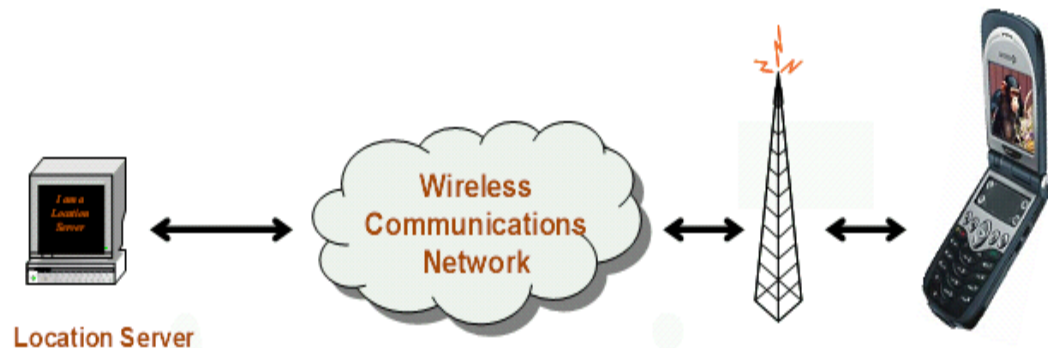
- Standard Positioning Service (SPS): Civilian Users

<u>Source</u>	<u>Amount of Error</u>
➤ Satellite clocks:	1.5 to 3.6 meters
➤ Orbital errors:	< 1 meter
➤ Ionosphere:	5.0 to 7.0 meters
➤ Troposphere:	0.5 to 0.7 meters
➤ Receiver noise:	0.3 to 1.5 meters
➤ Multipath:	0.6 to 1.2 meters
➤ Numerical errors	1 meter

GPS & A-GPS

Network Assisted GPS

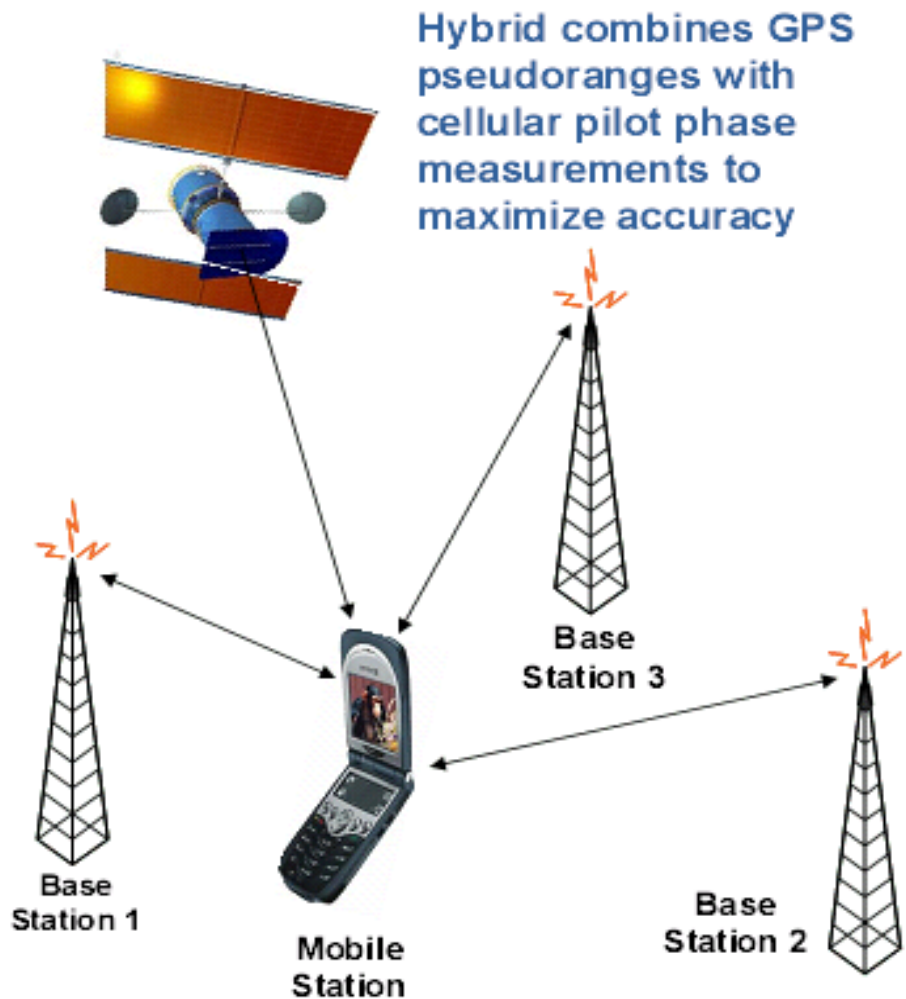
- AGPS
 - Assisted-GPS means that a Location Server assists a wireless device client to produce location fixes
 - TO IMPROVE PERFORMANCE AND REDUCE COST!
 - More accurate location fixes
 - Higher yield
 - Faster Time to Fix
 - Lower Battery consumption
 - Lower terminal device costs



GPS & A-GPS

Network Assisted GPS

- **Assisted-GPS** allows integration with other positioning methods
- The most important is **AFLT**
 - Basically, AFLT trilaterates to terrestrial towers, instead of GPS satellites
 - Is a key gap-filler when GPS is not available
 - Possible to combine terrestrial and satellite range measurements for a “hybrid” solution
- **Key Benefit:**
 - Improved yield, particularly in challenging environments



GPS & A-GPS

Android GPS

1) NmeaListener, onNmeaReceived

Return NMEA 0183 Protocol

```
$GPGGA,092750.000,5321.6802,N,00630.3372,W,1,8,1.03,61.7,M,55.2,M,,*76
$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.03,1.38*0A
$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,290,20,08,54,157,30*70
$GPGSV,3,2,11,02,39,223,19,13,28,070,17,26,23,252,,04,14,186,14*79
$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,,*76
$GPRMC,092750.000,A,5321.6802,N,00630.3372,W,0.02,31.66,280511,,,A*43
$GPGGA,092751.000,5321.6802,N,00630.3371,W,1,8,1.03,61.7,M,55.3,M,,*75
$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.03,1.38*0A
$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,290,20,08,54,157,30*70
$GPGSV,3,2,11,02,39,223,16,13,28,070,17,26,23,252,,04,14,186,15*77
$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,,*76
$GPRMC,092751.000,A,5321.6802,N,00630.3371,W,0.06,31.66,280511,,,A*45
```

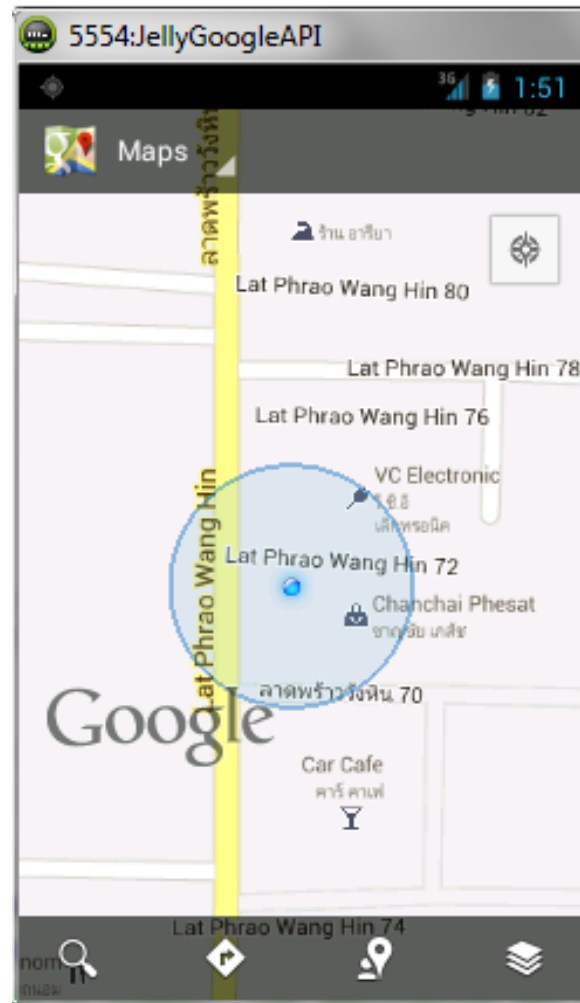
2) LocationChanged

Return Lat Long Eve Speed Clock

Location Service

Case study: GPS & GoogleMap

Location Service



Project: Android_demo_gps

GPS and MapView

- 1) Ask for permission for GPS data
- 2) Create LocationManager object
- 3) Set update interval
- 4) Implement LocationListener
- 5) Get coordinate data

GPS and MapView

-Ask for permission for GPS data

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
  package="com.example.android_demo_gps"  
  android:versionCode="1"  
  android:versionName="1.0" >
```

```
<uses-sdk  
  android:minSdkVersion="8"  
  android:targetSdkVersion="15" />
```

```
<application  
  android:icon="@drawable/ic_launcher"  
  android:label="@string/app_name"  
  android:theme="@style/AppTheme" >  
  <activity  
    android:name=".DemoGPS"  
    android:label="@string/title_activity_demo_gps" >  
    <intent-filter>  
      <action android:name="android.intent.action.MAIN" />  
  
      <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
  </activity>
```

```
</application>
```

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

```
</manifest>
```

AndroidManifest.xml

GPS and MapView

- Create LocationManager object

Global

```
LocationManager lm;  
lm = (LocationManager) getSystemService(LOCATION_SERVICE);
```

OnCreate

- Set update interval

onResume

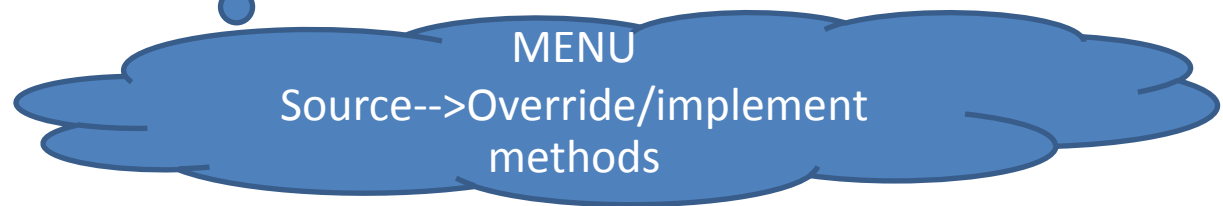
```
lm.requestLocationUpdates(LocationManager.GPS_PROVIDER,  
1000, 10f, this);
```

Update interval (milisecond)

GPS and MapView

- Implement LocationListener

**public class DemoGPS extends Activity implements
LocationListener {**



- Get coordinate data

@Override

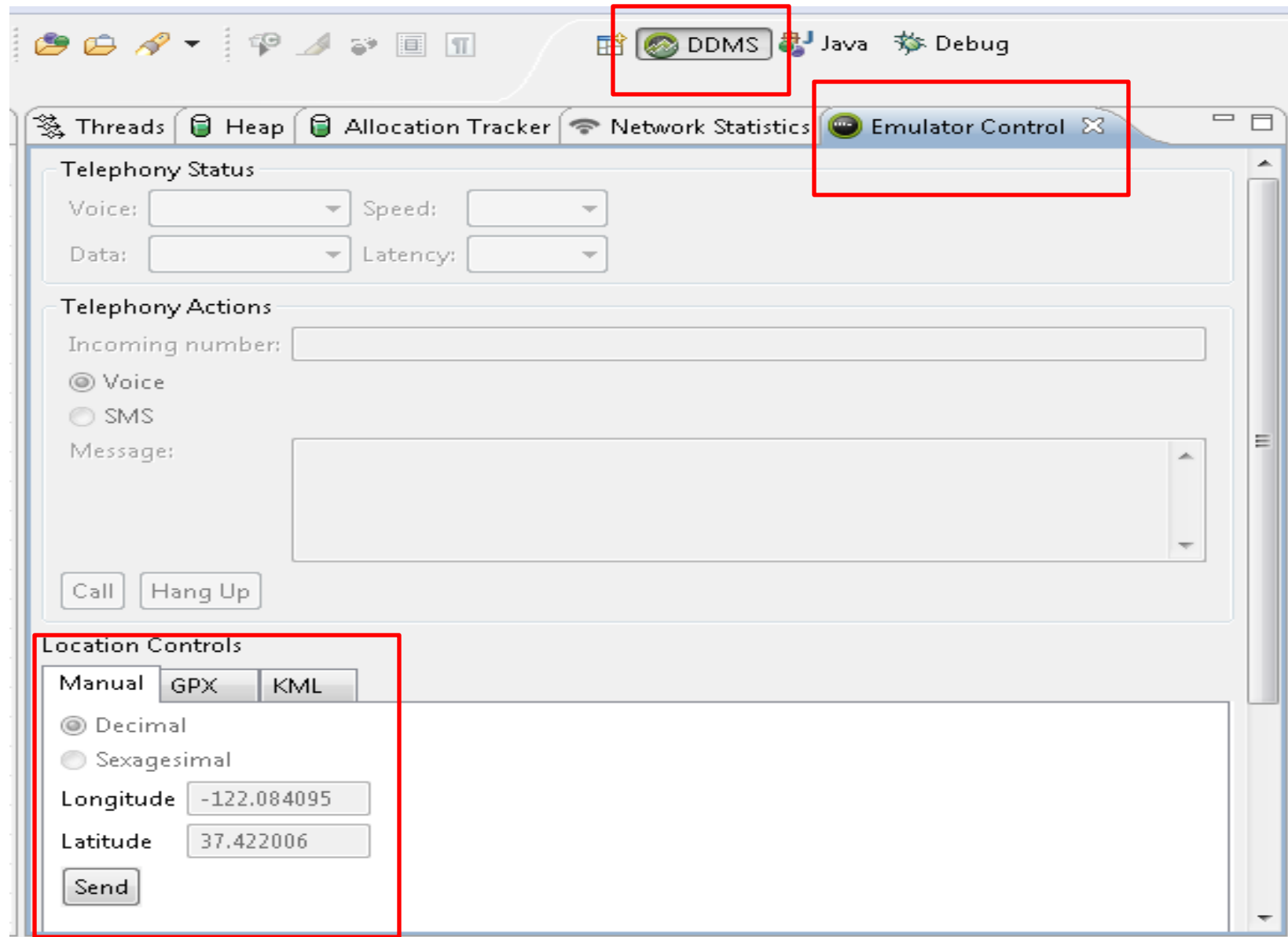
public void onLocationChanged(Location **arg0) {**

Lat=arg0.getLatitude();

Long=arg0.getLongitude();

}

Use DDMS to control emulator



GoogleMap

- 1)Ask for permission for internet access
- 2)Create URI for geo data with coordinate
- 3)Create intent to view URI
- 4)Start intent

**** You need GoogleAPI to enable map service ****

GoogleMap

- Ask for permission for internet access

```
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".DemoGPS"
        android:label="@string/title_activity_demo_gps" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
```

</application>

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<uses-permission android:name="android.permission.INTERNET"/>

</manifest>

GoogleMap

Create URI for geo data with coordinate

```
Uri uri = Uri.parse("geo:" + Lat + "," + Long);
```

Create intent to view URI

```
Intent intent = new Intent(Intent.ACTION_VIEW,uri);
```

Start intent

```
startActivity(intent);
```

Thank you 😊