ANDROID platforma
lokaciono zasnovani servisi
LocationProvider, MapView

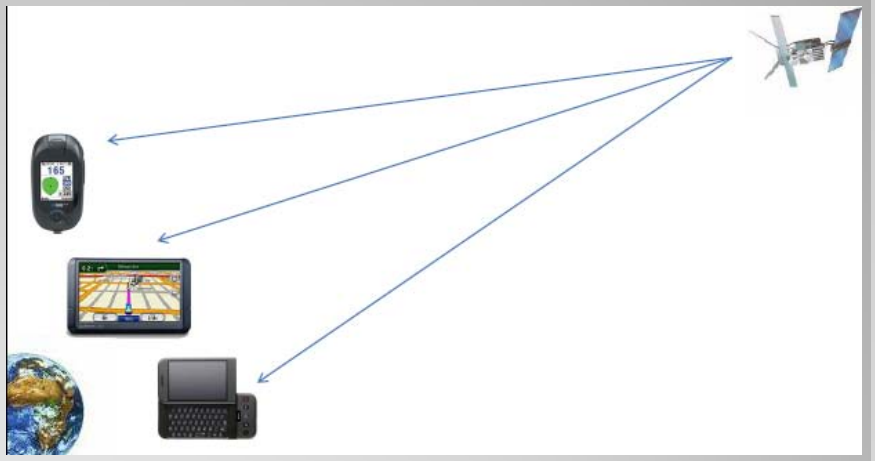
Mobilni i distribuirani informacioni sistemi

Mr Bratislav Predić

2012. godina

Location-based servisi

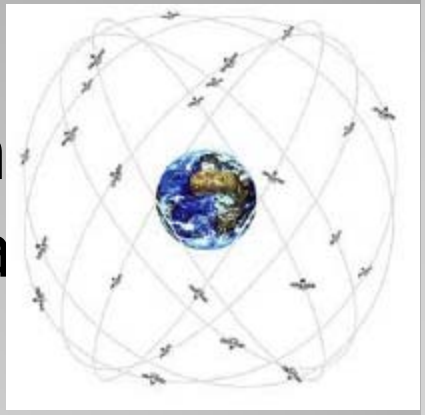
- Informacioni sistemi koji deo svoje funkcionalnosti baziraju na trenutnoj lokaciji korisnika
- Primeri:
  - Marketing
  - Zabava (igre)
  - Hitne službe
  - Poslovni sistemi (komercijala...)

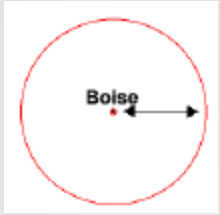


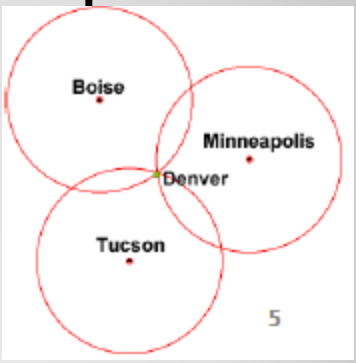
- Osnova lokaciono zasnovanih informacionih sistema
  - Jeftin i pouzdan sistem za lociranje korisnika

Global Positioning System - GPS

- 30-tak satelita u orbiti
- Inicijalno razvijen kao vojni sistem
- Sa bilo koje tačke na svetu vidljiva su barem 4 satelita u svakom trenutku
- Princip lociranja se zasniva na trilateraciji – merenju razdaljine do satelita
- Trilateracija u 2D i 3D prostoru – preseci krugova i sfera

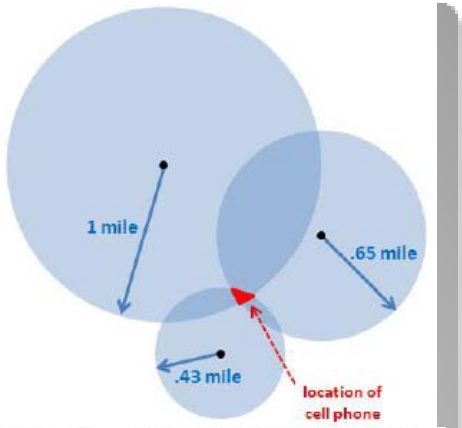




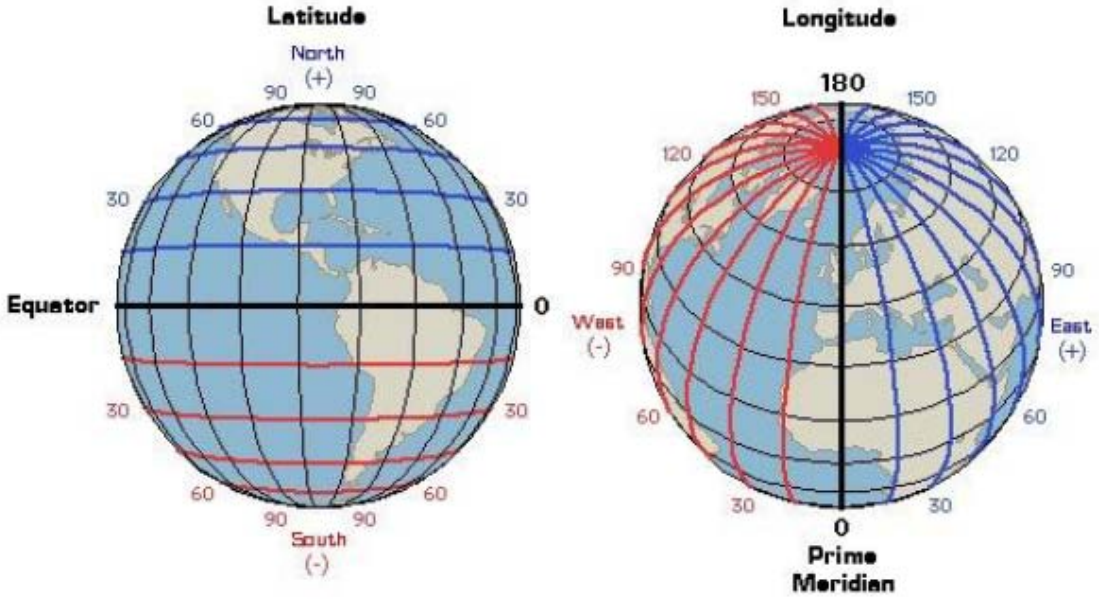


Lociranje u mobilnoj mreži

- Isti princip trilateracije se može primeniti i u mreži baznih stanica mobilne telefonije
- Geografski koordinatni sistem (sferni koordinatni sistem)



Latitude in GPS-Decimal notation: +90.00000 (North) to -90.000000 (South)
Longitude GPS-Decimal notation: +180.000000 (East) to -180.000000 (West)



Lociranje na Android platrormi

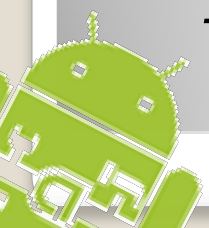
- Tri metode lociranja
  - GPS (satelitski, Epson Infineon prijemnik 3x3mm)
  - Cell tower (u mreži baznih stanica)
  - WiFi (participatorno opažanje – SVI korisnici učestvuju)
    Probajte sami: http://samy.pl/androidmap
- Android klase za lociranje
  - *Address*
  - *Criteria*
  - *Geocoder*
  - *GpsSatellite*
  - *GpsStatus*
  - *Location*
  - *LocationManager*
  - *LocationProvider*
  - *GpsStatus.Listener*
  - *GpsStatus.NmeaListener*
  - *LocationListener*

Android klase za lociranje

- **Location** klasa
  - Klasa predstavlja poziciju određenu u nekom trenutku
  - Sadrži podatke o: geog. dužini i širini, UTC timestamp i opciono nadmorsku visinu, brzinu i pravac kretanja
  - Klasa ima getter metode za sve ove podatke
  - Zanimljiva je statička metoda
    static void distanceBetween(...)
    Računa rastojanje između dve tačke u sfernom koordinatnom sistemu (nije elementaran problem)

Android klase za lociranje

- **LocationManager** klasa
  - Omogućava pristup Android sistemskim uslugama za geolociranje
  - Dva osnovna zadatka:
    - Periodično obaveštava o lokaciji telefona
    - Emituje Intent kada se korisnik nađe u blizini neke definisane tačke
  - Ne instancira se direktno `Context.getSystemService(Context.LOCATION_SERVICE)`
  - Konkretnu lokaciju generiše *LocationProvider*
  - *LocationProvider* se pribavlja direktno po imenu ili na osnovu definisanih kriterijuma lociranja *(Criteria* – potrošnja energije, preciznost itd.*)*



Android klase za lociranje

- **LocationListener** klasa
  - *LocationListener* se registruje sa *LocationManager*-om i po promeni lokacije dobija novu lokaciju
  - Registracija listener-a `requestLocationUpdates(Provider, minTime, minDistance, LocationListener)`
  - Bitne callback metode
    - *onLocationChanged(Location)*
      Nova lokacija korisnika dobijena
    - *onProviderDisabled(String)*, *onProviderEnabled(String)*
      Korisnik isključio/uključio location provider
    - *onStatusChanged(String, int, Bundle)*
      Promenjen status provider-a (GPS ne vidi dovoljno satelita npr.)

Android klase za lociranje

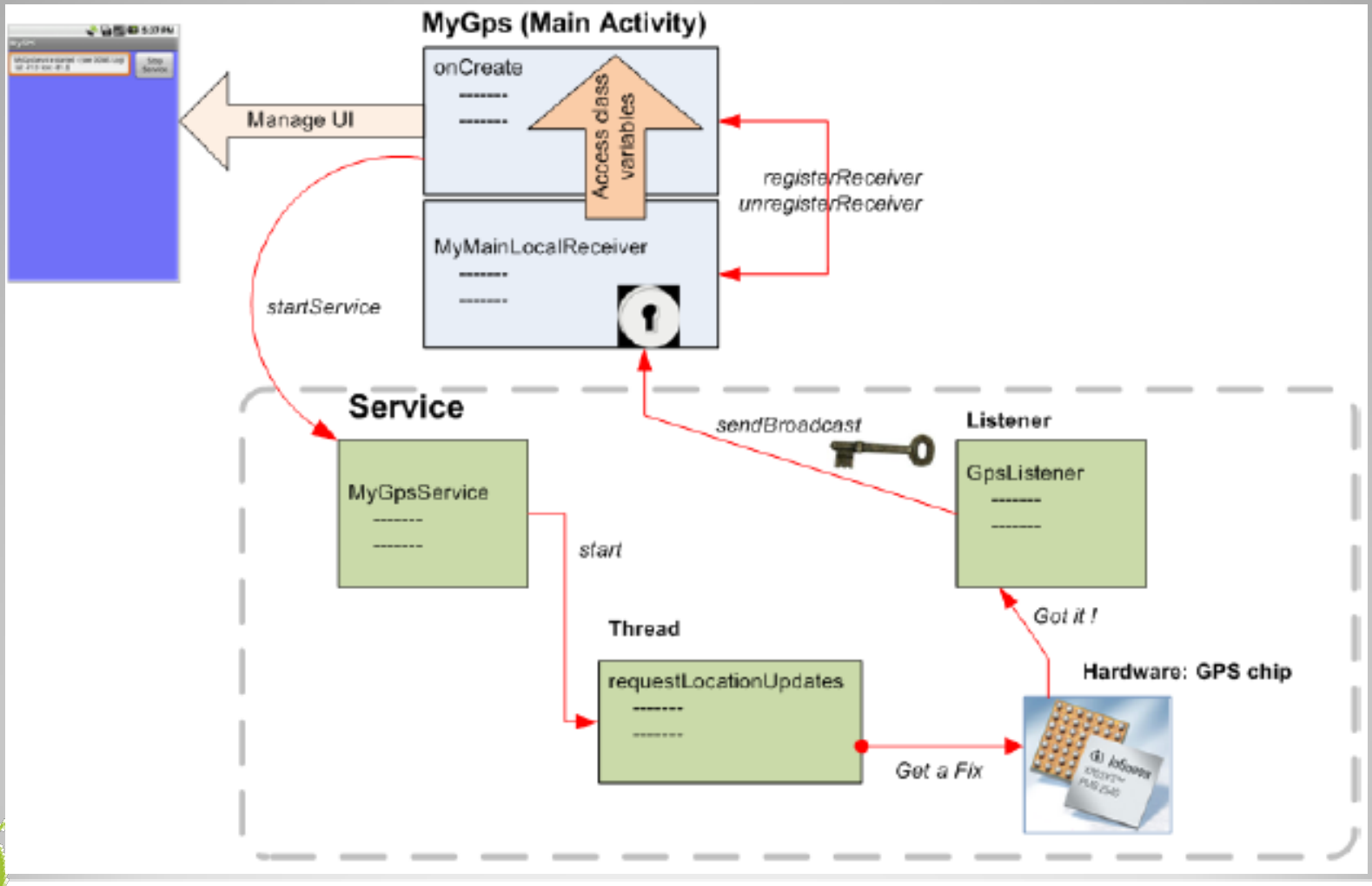
- **LocationProvider** klasa
  - Opisuje karakteristike konkretnog location provider-a (metode lociranja)
  - Osim preciznosti, lociranje korišćenjem provider-a može korisniku prouzrokovati dodatne troškove
  - A-GPS (assisted GPS – koristi mobilni paketni prenos podataka da preuzme ažuran almanah)
  - Metode za ispitivanje tipova podataka o lokaciji i kretanju koje provider vraća:
    - *getAccuracy()*
    - *supportsAltitude()*
    - *supportsBearing()*
    - *supportsSpeed()*

Android lociranje - primer

- Specifičnosti primera
  - GPS prijemnik nije sinhroni tip uređaja, ne postoji funkcija koja sinhrono vraća trenutnu lokaciju
  - Obzirom na ovo kašnjenje zahtev je smešten u pozadinski servis
  - I servis se izvršava u istom procesu kao i glavna aktivnost pa u servisu moramo napraviti poseban thread
  - Kombinujemo i mogućnost slanja lokacije SMS porukom
  - Kako bi testirali slanje SMS poruka radimo sa dva emulatora

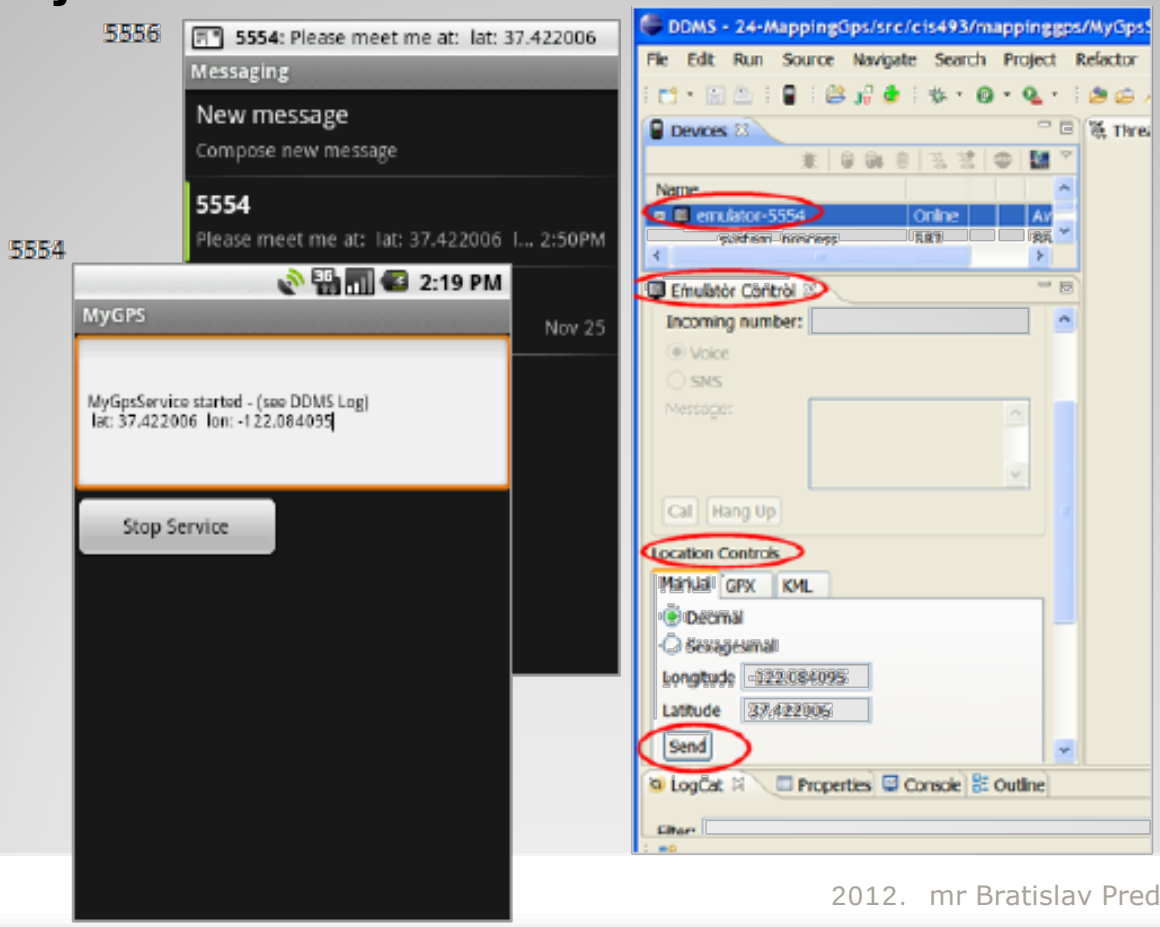
Android lociranje - primer

- Arhitektura primera



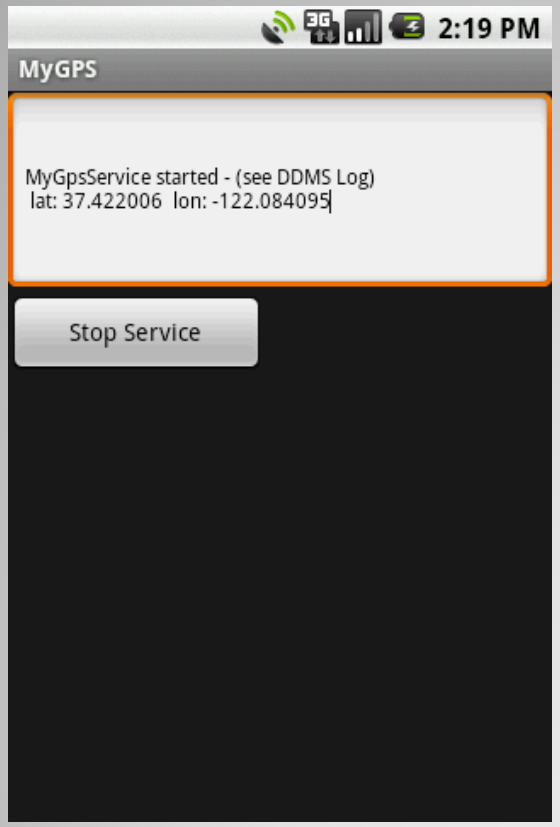
Android lociranje - primer

- U emulator okruženju moguće je ručno unositi geografske koordinate simuliranom GPS prijemniku



Android lociranje - primer

- Layout




```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  android:id="@+id/widget32"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical"
  xmlns:android=
   "http://schemas.android.com/apk/res/android”>
  <EditText
    android:id="@+id/txtMsg"
    android:layout_width="fill_parent"
    android:layout_height="120px"
    android:textSize="12sp”>
  </EditText>
  <Button
    android:id="@+id/btnStopService"
    android:layout_width="151px"
    android:layout_height="wrap_content"
    android:text="Stop Service”>
  </Button>
</LinearLayout>
```

Android lociranje - primer

- Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="cis493.mappinggps"
      android:versionCode="1"
      android:versionName="1.0">
    <application
                android:icon="@drawable/icon"
                android:label="@string/app_name"
                android:debuggable="true"  >

        <activity android:name=".MyGPS"
                android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <service
                android:name="MyGpsService">
        </service>

    </application>

    <uses-sdk android:minSdkVersion="2" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

</manifest>
```

Android lociranje - primer

```
publicclassMyGPSextendsActivity {
  Button btnStopService;
  TextView txtMsg;
  Intent intentMyService;
  ComponentName service;
  BroadcastReceiver receiver;

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    txtMsg= (TextView) findViewById(R.id.txtMsg);
    // initiate the service
    intentMyService= new Intent(this, MyGpsService.class);
    service= startService(intentMyService);
    txtMsg.setText("MyGpsServicestarted -(see DDMS Log)");
    // register & define filter for local listener
    IntentFilter mainFilter= new IntentFilter(GPS_FILTER);
    receiver= new MyMainLocalReceiver();
    registerReceiver(receiver, mainFilter);
    btnStopService= (Button) findViewById(R.id.btnStopService);
```

Android lociranje - primer

```
  btnStopService.setOnClickListener(newOnClickListener() {
  public void onClick(View v) {
    try{
      stopService(new Intent(intentMyService) );
      txtMsg.setText("After stopingService: \n“ + service.getClassName());
      btnStopService.setText("Finished");
      btnStopService.setClickable(false);
    } catch(Exception e) {
      Log.e("MYGPS", e.getMessage() );
    }
  }
 });
}//onCreate

  @Override
  protected void onDestroy() {
    super.onDestroy();
    try{
      stopService(intentMyService);
      unregisterReceiver(receiver);
    } catch(Exception e) {
      Log.e("MAIN-DESTROY>>>", e.getMessage() );
    }
    Log.e("MAIN-DESTROY>>>", "Adios");
  }// onDestroy
```

Android lociranje - primer

```
Private class MyMainLocalReceiver extends BroadcastReceiver{
  @Override
  public void onReceive(Context localContext, Intent callerIntent) {
    double latitude = callerIntent.getDoubleExtra("latitude",-1);
    double longitude = callerIntent.getDoubleExtra("longitude",-1);
    Log.e("MAIN>>>", Double.toString(latitude));
    Log.e("MAIN>>>", Double.toString(longitude));
    String msg= " lat: "+ Double.toString(latitude) + " "
        + " lon: "+ Double.toString(longitude);
    txtMsg.append("\n"+ msg);
    //testing the SMS-texting feature
    texting(msg);
  }
}//MyMainLocalReceiver
```

- BroadcastReceiver iz servisa dobija koordinate preko Intent-a
- Dobijene koordinate prikazuje na ekranu i šalje ih SMS porukom

Android lociranje - primer

```
  private void texting(String msg){
    try{
      SmsManagersmsMgr= SmsManager.getDefault();
      // Parameter of sendTextMessage are:
      // destinationAddress, senderAddress,
      // text, sentIntent, deliveryIntent)
      //---------------------------------------------------
      smsMgr.sendTextMessage("5556", "5551234",
         "Please meet me at: "+ msg, null, null);
    } catch(Exception e) {
      Toast.makeText(this, "texting\n"+ e.getMessage(), 1).show();
    }
  }// texting
}//MyGPS
```

- Ne koristimo poslednja dva argumenta – intent-i koji se šalju kada je poruka poslata i kada je primljena (receipt)

Android lociranje - primer

- Servis

```
Public class MyGpsService extends Service {
  String GPS_FILTER= "cis493.action.GPS_LOCATION";
  Thread triggerService;
  LocationManager lm;
  GPSListener myLocationListener;
  boolean isRunning= true;

  @Override
  public Ibinder onBind(Intent arg0) {
    return null;
  }
  @Override
  public void onCreate() {
    super.onCreate();
  }
  @Override
  public void onStart(Intent intent, intstartId) {
    super.onStart(intent, startId);
    Log.e("<<MyGpsService-onStart>>", "I am alive-GPS!");

    // New thread for GPS interfacing
```

Android lociranje - primer

```
  triggerService = new Thread(new Runnable() {
    public void run() {
      try{
        Looper.prepare();
        // try to get your GPS location using the LOCATION.SERVICE provider
        lm = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        // This listener will catch and disseminate location updates
        myLocationListener= new GPSListener();
        long minTime= 10000; // frequency update: 10 seconds
        float minDistance= 50; // frequency update: 50 meter
        lm.requestLocationUpdates(//request GPS updates
                 LocationManager.GPS_PROVIDER,
                 minTime,
                 minDistance,
                 myLocationListener);
        Looper.loop(); -- Creates message loop for a thread
      } catch(Exception e) {
         Log.e("MYGPS", e.getMessage() );
      }
    }// run
  });
  triggerService.start();
}// onStart
```

http://rxwen.blogspot.com/2010/08/looper-and-handler-in-android.html

Android lociranje - primer

```
 private class GPSListener implements LocationListener{
  public void onLocationChanged(Location location) {
    //capture location data sent by current provider
    double latitude = location.getLatitude();
    double longitude = location.getLongitude();
    //assemble data bundle to be broadcasted
    Intent myFilteredResponse= new Intent(GPS_FILTER);
    myFilteredResponse.putExtra("latitude", latitude);
    myFilteredResponse.putExtra("longitude", longitude);
    Log.e(">>GPS_Service<<", "Lat:"+ latitude + " lon:"+ longitude);
    //send the location data out
    sendBroadcast(myFilteredResponse);
  }

  public void onProviderDisabled(String provider) {}
  public void onProviderEnabled(String provider) {}
  public void onStatusChanged(String provider, int status, Bundle extras) {}
 };//GPSListenerclass
}// MyService3
```

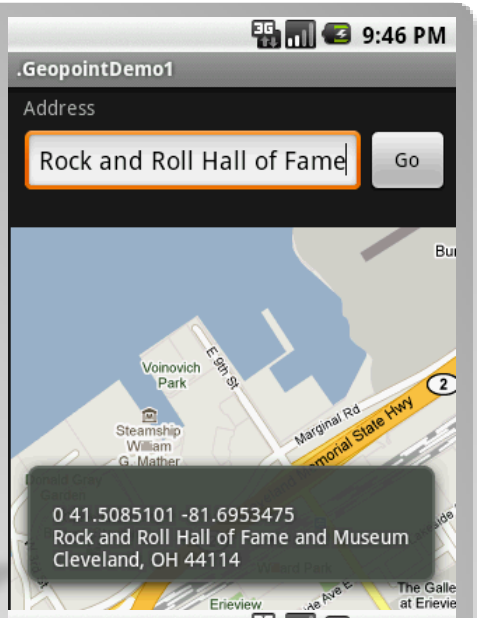
Dodatni lokacioni servisi

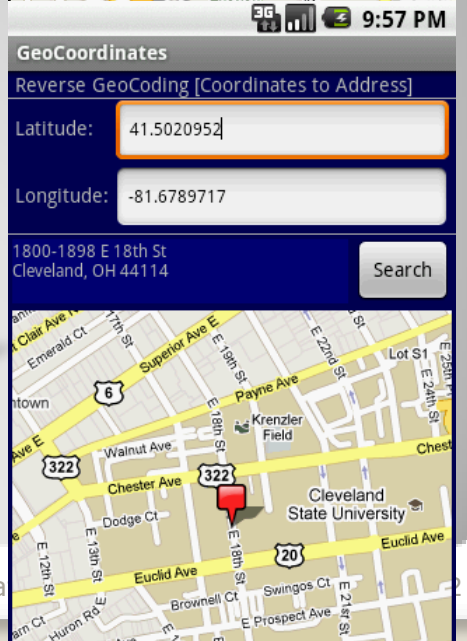
- Google nudi dodatne lokacione servise
  - Geokodiranje
    Vraća koordinate na osnovu adrese

```
Geocoder gc= new Geocoder(this);
// 5 najpribližnijih lokacija
List<Address> lstFoundAddresses =
        gc.getFromLocationName(txtStreetAddress, 5);
```

  - Reverzno geokoridanje
    Vraća adresu na osnovu koordinata

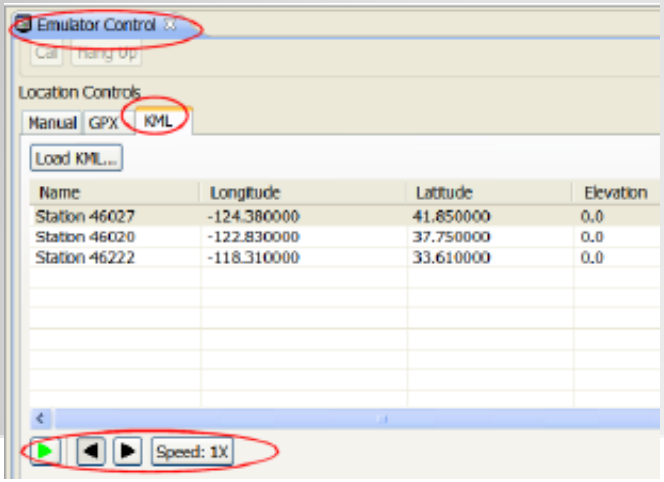
```
Geocoder gc = new Geocoder(context, Locale.US);
List<Address> streets =
        gc.getFromLocation(latitude, longitude, 1);
```





Koordinate u Android emulatoru

- U Android emulatoru koordinate se mogu ručno uneti ili se može uneti putanja kao:
  - GPX datoteka
    često ručni GPS prijemnici mogu snimati putanju kretanja u ovaj format
  - KML datoteka
    XML format Google Earth



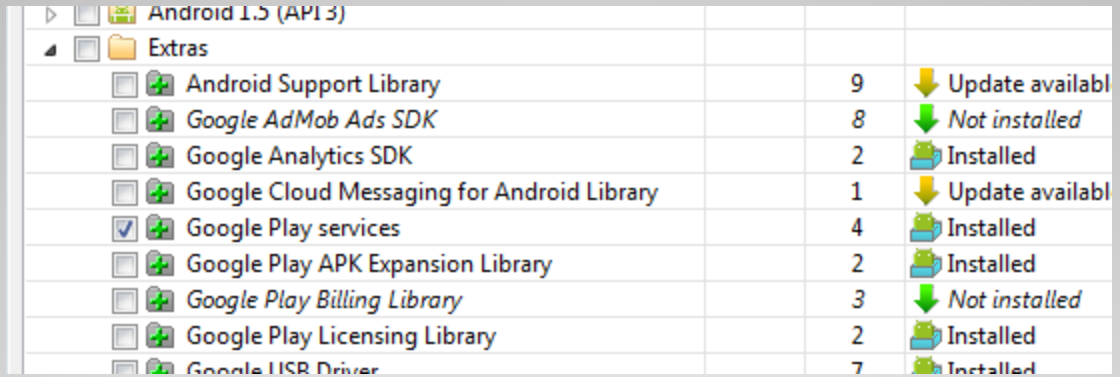

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
<Placemark>
        <name>Station 46027</name>
        <description>Off the coast of Lake Earl</description>
        <Point>
        <coordinates>-124.38,41.85,0</coordinates>
        </Point>
</Placemark>
<Placemark>
        <name>Station 46020</name>
        <description>Outside the Golden Gate</description>
        <Point>
        <coordinates>-122.83,37.75,0</coordinates>
        </Point>
</Placemark>
<Placemark>
        <name>Station 46222</name>
        <description>San Pedro Channel</description>
        <Point>
        <coordinates>-118.31,33.61,0</coordinates>
        </Point>
</Placemark>
</kml>
```

Android i mapiranje

- Android koristi Google Maps eksternu biblioteku za rad sa mapama
- Klase iz ovog paketa potpuno transparentno
  - Preuzimaju (download) segmente mape
  - Iscrtavaju mapu
  - Keširaju segmente mape na lokalnom uređaju
- Google Maps V2 – potpuno novi koncept
  - V1 je deprecated, map key nije moguće dobiti
- Glavna klasu u ovom paketu je **MapView**, podklasa *ViewGroup*
- MapView klasa obrađuje sve događaje za navigaciju po mapi (zoom, pan...)

Android Google Maps V2

- Android Google Maps API V2 zavisi od *Google Play Service SDK* ekstenzije
- Instalira se kroz SDK manager



- Projekat može da se importuje u Eclipse workspace kao biblioteka

```
<android-sdk-folder>/extras/google/google_play_services/libproject/google-play-services_lib
```

- Voditi računa o TargetSDK

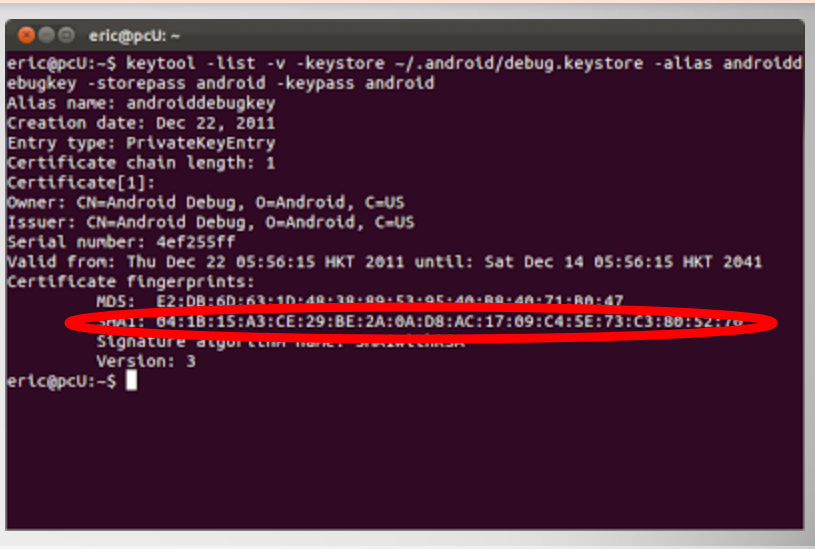
http://android-er.blogspot.com/2012/12/a-simple-example-using-google-maps.html

Android Google Maps V2

- Korišćenje Google resursa se prati kroz registraciju aplikacijakorišćenjem ključeva
- SVAKI development računar ima keystore
  - Debug
  - Release

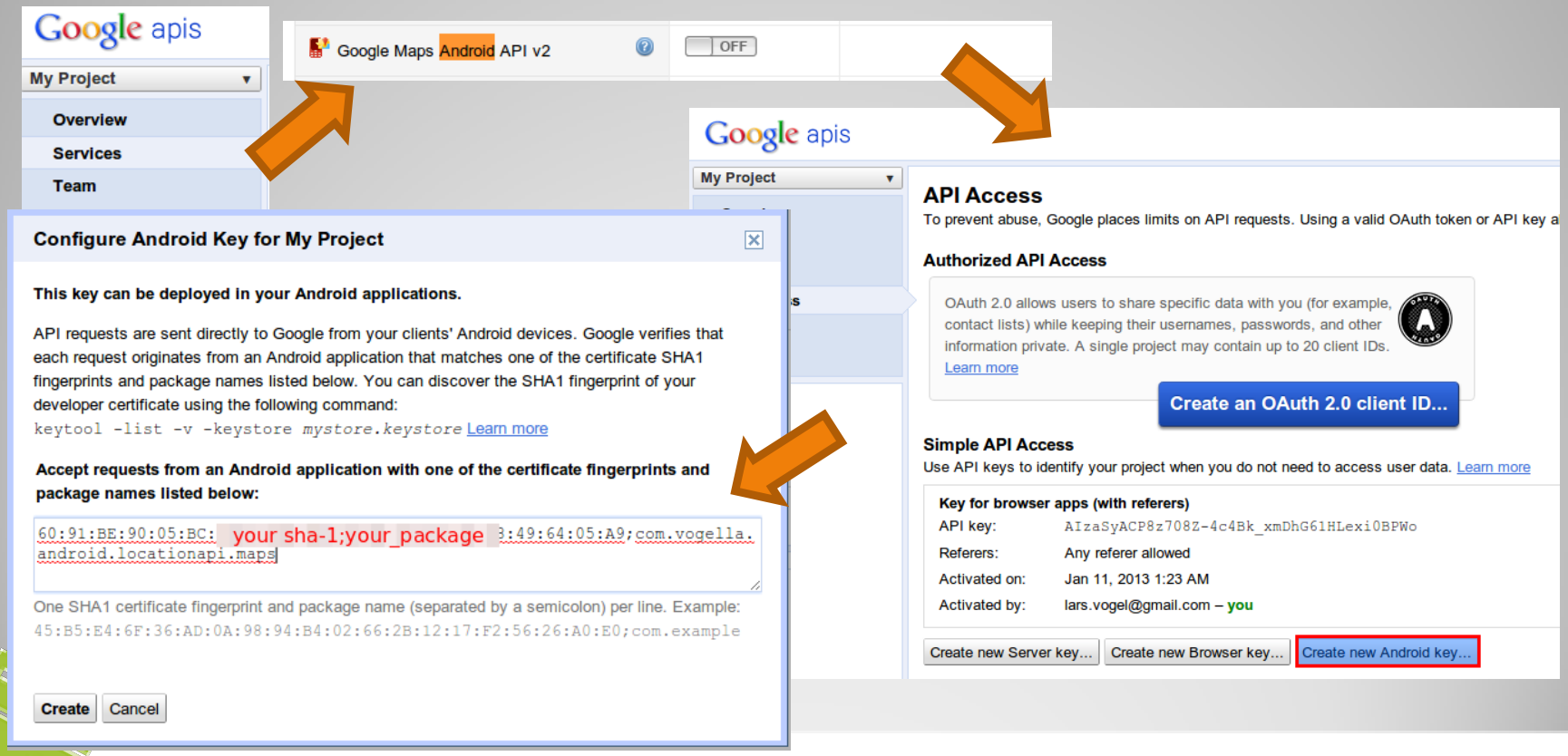
```
keytool -list -v -keystore <USER_HOME>/.android/debug.keystore -alias
androiddebugkey -storepass android -keypass android
```

- Bitan je SH1 fingerprint



Android Google Maps V2

- Centralizovano upravljanje svim Google servisima kroz Google API Console
  - https://code.google.com/apis/console



Android Google Maps V2

- Neophodne izmene u manifestu

```
<permission
    android:name="com.example.MAPS_RECEIVE"
    android:protectionLevel="signature" />
```

Custom permission štićen potpisom aplikacije. Verovatno se koristi kao endpoint za komunikaciju od strane Google Maps API

```
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true" />
```

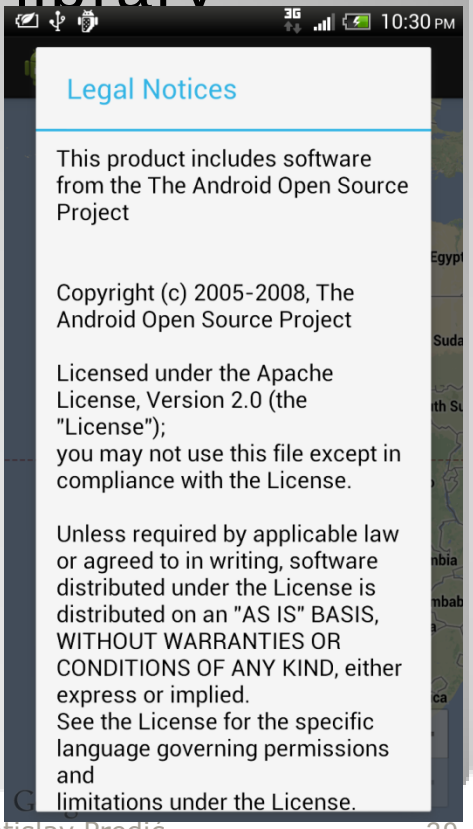
Google Maps API V2 koristi OpenGLES v2 za render mape

```
<uses-permission android:name="com.example.MAPS_RECEIVE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<application
    …
    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="your_apikey" />
</application>

</manifest>
```

Aplikacija koristi registrovan Google Maps V2 API key (**ne može se koristiti V1 key**)

Android Google Maps V2

- Google Maps API V2 radi sa novijim tipovima UI widget-a
  - *MapFragment* (od API level 11 – Android 3.0)
  - Za starije uredjaje se koristi support library
    - *SupportMapFragment*
- Glavna aktivnost nasleđuje *FragmentActivity* a ne *Activity*
- Sve aplikacije koje koriste ovaj API **moraju** da prikažu *Google Play Services Legal Notices*



Android Google Maps V2

- Klasa *GooglePlayServicesUtil* nudi helper metodu za preuzimanje aktuelnog teksta licence

```
String LicenseInfo = GooglePlayServicesUtil.getOpenSourceSoftwareLicenseInfo(
        getApplicationContext());
AlertDialog.Builder LicenseDialog = new AlertDialog.Builder(MainActivity.this);
LicenseDialog.setTitle("Legal Notices");
LicenseDialog.setMessage(LicenseInfo);
LicenseDialog.show();
```

- Google Maps API V2 zavisi od Google Plaz Services komponente – treba proveriti

```
@Override
 protected void onResume() {
  super.onResume();
  int resultCode = GooglePlayServicesUtil.isGooglePlayServicesAvailable(getApplicationContext());
  if (resultCode == ConnectionResult.SUCCESS){
   Toast.makeText(getApplicationContext(), "isGooglePlayServicesAvailable SUCCESS",
     Toast.LENGTH_LONG).show();
  }else{
   GooglePlayServicesUtil.getErrorDialog(resultCode, this, RQS_GooglePlayServices);
  }
```

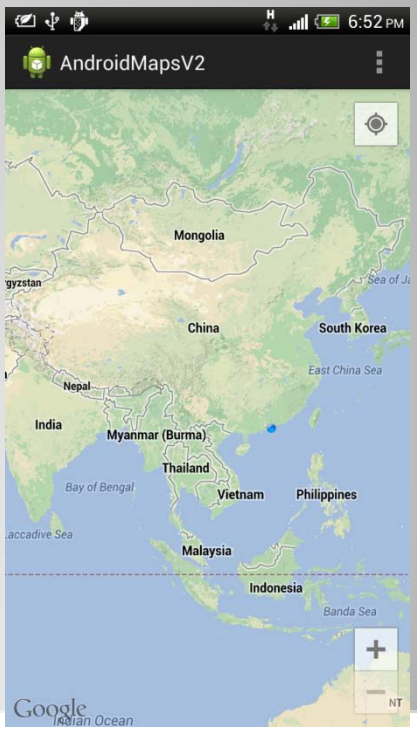
Android Google Maps V2

- Za upravljanje mapom se koristi GoogleMap objekat koga treba preuzeti iz fragmenta

```
FragmentManager myFragmentManager = getSupportFragmentManager();
SupportMapFragment mySupportMapFragment
    = (SupportMapFragment)myFragmentManager.findFragmentById(R.id.map);
myMap = mySupportMapFragment.getMap();
```

- Moguće je uključiti i opciju prikaza korisnikove trenutne lokacije

```
myMap.setMyLocationEnabled(true);
```



Android Google Maps V2

- Postavljanje tipa mape
  - MAP_TYPE_NONE
    Bez pozadinske mape
  - MAP_TYPE_NORMAL
    Obične mape
  - MAP_TYPE_SATELLITE
    Satelitski snimak bez labela
  - MAP_TYPE_HYBRID
    Satelitski snimak sa transparentnim ulicama
  - MAP_TYPE_TERRAIN
    Reljef


```
FragmentManager myFragmentManager = getSupportFragmentManager();
SupportMapFragment mySupportMapFragment
          = (SupportMapFragment)myFragmentManager.findFragmentById(R.id.map);
myMap = mySupportMapFragment.getMap();
myMap.setMyLocationEnabled(true);
//myMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
//myMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
//myMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
myMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
```

Android Google Maps V2

- Detekcija klika na mapu
- *FragmentActivity* treba da implementira *OnMapClickListener*
- Override-ujemo metodu *onMapClick(LatLng point)* i uvezujemo listener sa
  - *myMap.setOnMapClickListener(this)*

```
public class MainActivity extends FragmentActivity implements OnMapClickListener{
  private GoogleMap myMap;
  ...
  myMap.setOnMapClickListener(this);
  ...
  @Override
  public void onMapClick(LatLng point) {
    tvLocInfo.setText(point.toString());
    myMap.animateCamera(CameraUpdateFactory.newLatLng(point));
  }
  ...
```

Android Google Maps V2

- Dodavanje markera na mapu
- Ne radi se kroz overlay slojeve kao u V1 API
- Poseban listener
  - *OnMapLongClickListener*
- *FragmentActivity* implementira ovaj listener i override-uje metodu
  - *onMapLongClick(LatLng point)*
- Listener se vezuje za mapu sa
  - *myMap.setOnMApLongClickListener(this)*
- Marker se dodaje direktno na mapu


```
myMap.addMarker(new MarkerOptions().position(point).title(“title”)
```

Android Google Maps V2

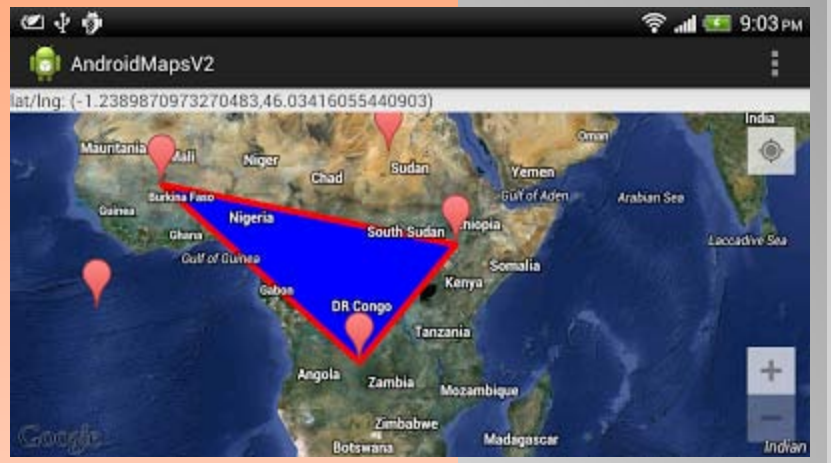
- Klik na marker i crtanje polilinije na mapi
- Aktivnost treba d aimplementira *OnMarkerClickListener*


```
@Override
 public boolean onMarkerClick(Marker marker) {
  if(markerClicked){
   if(polyline != null){
    polyline.remove();
    polyline = null;
   }
   rectOptions.add(marker.getPosition());
   rectOptions.color(Color.RED);
   polyline = myMap.addPolyline(rectOptions);
  }else{
   if(polyline != null){
    polyline.remove();
    polyline = null;
   }
   rectOptions = new PolylineOptions().add(marker.getPosition());
   markerClicked = true;
  }
  return true;
 }
```

Android Google Maps V2

- Poligon se crta na praktično identičan način

```
@Override
 public boolean onMarkerClick(Marker marker) {

  if(markerClicked){

   if(polygon != null){
    polygon.remove();
    polygon = null;
   }

   polygonOptions.add(marker.getPosition());
   polygonOptions.strokeColor(Color.RED);
   polygonOptions.fillColor(Color.BLUE);
   polygon = myMap.addPolygon(polygonOptions);
  }else{
   if(polygon != null){
    polygon.remove();
    polygon = null;
   }

   polygonOptions = new PolygonOptions().add(marker.getPosition());
   markerClicked = true;
  }

  return true;
 }
```



Android Google Maps V2

- Premeštanje markera (draggable marker)
  - Marker mora da bude označen kao *draggable*
    marker.draggable(true)
  - Implementirati *OnMarkerDragListener* u aktivnosti
  - Listener se postavnja za mapu
    myMap.setOnMarkerDragListener(this)
  - Override-ovati tri metode listenera
    - onMarkerDrag, onMarkerDragStart i onMarkerDragEnd

Android Google Maps V2

- GoogleMap widgetpodržava standarne kontrole
  - Zoom, kompas, myLocation

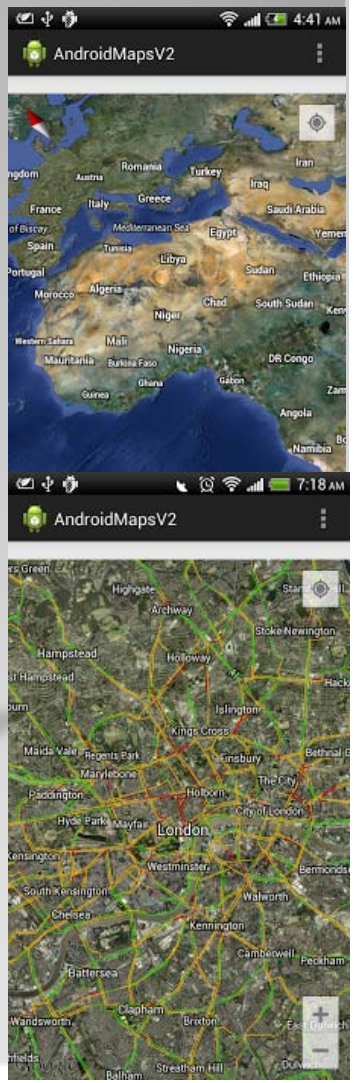
```
myMap.getUiSettings().setZoomControlsEnabled(true);
myMap.getUiSettings().setCompassEnabled(true);
myMap.getUiSettings().setMyLocationButtonEnabled(true);
```

- Omogućavanje map gestures kontrole (multitouch)

```
myMap.getUiSettings().setRotateGesturesEnabled(true);
myMap.getUiSettings().setScrollGesturesEnabled(true);
myMap.getUiSettings().setTiltGesturesEnabled(true);
myMap.getUiSettings().setZoomGesturesEnabled(true);
//or myMap.getUiSettings().setAllGesturesEnabled(true);
```

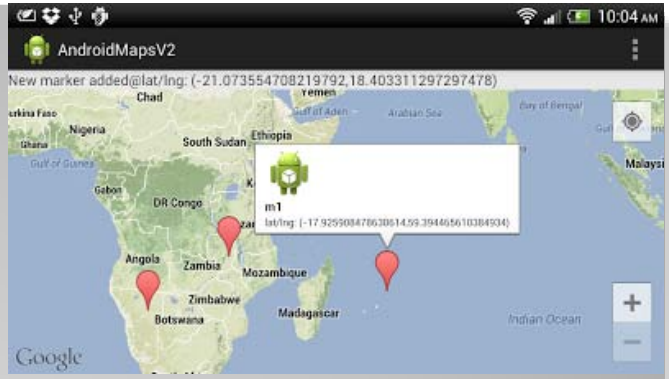
- Traffic layer

```
myMap.setTrafficEnabled(true);
```



Android Google Maps V2

- Custom info za markere
- Kroz implementaciju *InfoWindowAdapter*
- Dodati layout za info prikaz




```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal“>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginRight="5dp"
            android:adjustViewBounds="true"
            android:src="@drawable/ic_launcher"/>
        <TextView
            android:id="@+id/title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="12dp"
            android:textStyle="bold"/>
        <TextView
            android:id="@+id/snippet"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="10dp"/>
    </LinearLayout>
</LinearLayout>
```

```
public class MainActivity extends Activity
 implements OnMapLongClickListener{

class MyInfoWindowAdapter implements InfoWindowAdapter{
  private final View myContentsView;
  MyInfoWindowAdapter(){
   myContentsView = getLayoutInflater().inflate(
              R.layout.custom_info_contents, null);
  }

  @Override
  public View getInfoContents(Marker marker) {
    TextView tvTitle =
((TextView)myContentsView.findViewById(R.id.title));
    tvTitle.setText(marker.getTitle());
    TextView tvSnippet =
((TextView)myContentsView.findViewById(R.id.snippet));
    tvSnippet.setText(marker.getSnippet());

    return myContentsView;
  }

  @Override
  public View getInfoWindow(Marker marker) {
   // TODO Auto-generated method stub
   return null;
  }

 }
```

Alternative

- OpenStreetMap
  - Potpuno besplatan servis
  - http://www.openstreetmap.org/
- OpenStreetMap na Androidu
  - *osmdroid* framework
  - Potpuno imitira GoogleMaps V1 API