

# Android alkalmazásfejlesztés

Google Maps használata  
Alternatív térképek

**Sicz-Mesziár János**

sicz.mj@gmail.com

2011. március 31.

OE-NIK



# Google Maps

- ◎ Jól használható külső library. 😊
- ◎ Használatához regisztráció és az ÁSZF elfogadása kell!
- ◎ Google Maps API kulcs beszerzése:
  - Google API telepítés Android SDK Manager-ben
  - MD5 lenyomat készítés:

```
C:\Program Files\Java\<JDK_version>\bin>keytool.exe -list -alias  
androiddebugkey -keystore "c:\Users\<Username>\.android\debug.keystore"  
-storepass android -keypass android
```
  - API kulcsszerzés, MD5 lenyomat alapján:  
<http://code.google.com/intl/hu-HU/android/maps-api-signup.html>
- ◎ AndroidManifest-ben library használat hozzáadása:  
Application fül > Application Nodes > Add... > Uses Library  
Name: ***com.google.android.maps***
- ◎ AndroidManifest-ben internet használat kérés:  
Permissions fül > Add... > Uses Permission  
Name: ***android.permission.INTERNET***

# Google Maps a gyakorlatban

## ◎ MapView elhelyezése a layout-on:

```
<com.google.android.maps.MapView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:enabled="true"
    android:clickable="true"
    android:id="@+id/myMap"
    android:apiKey="<-- API KEY -->" />
```

## ◎ Google API target beállítása:

Projects menü > Properties > Android > Project Build

Target > Google APIs kiválasztása

**Emulátort is át kell állítani!!!**

## ◎ Osztályunk egy MapActivity leszármazott:

```
class MyMapActivity extends MapActivity{
    protected boolean isRouteDisplayed() {}
    MapView myMap = (MapView)findViewById(R.id.myMap);
}
```

# Zoom Controls, Map Controller

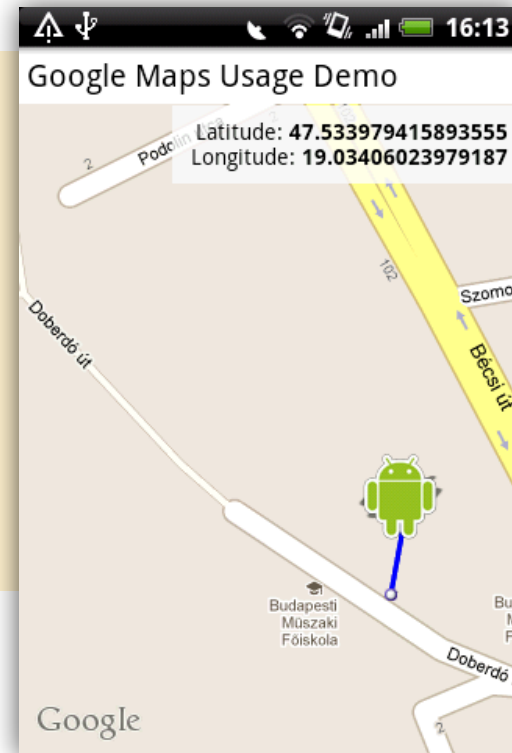
- ◉ Beépített +/- zoom engedélyezése:

```
myMap.setBuiltInZoomControls(true);  
myMap.getZoomButtonsController().setAutoDismissed(false); // Automatikus elrejtés tiltása
```

- ◉ Egyéni zoom gombokhoz saját ZoomControl kell.

## Térkép irányítása MapController-rel:

```
MapController mc = myMap.getController();  
mc.setCenter(  
    new GeoPoint(47533571, 19033851));  
mc.setZoom(17);  
mc.scrollBy(120, 120); // Pixelben értendő  
mc.animateTo(new GeoPoint(...));
```



# Pontok és pozíciók elhelyezése

## ⦿ Kell egy ItemizedOverlay<OverlayItem> leszármazott:

- Hozzá egy konstruktor, amiben az alap Marker-t megadjuk
- Egy listába felgyűjtjük a réteghez tartozó pontokat
- Előírt/ajánlott metódusok megvalósítása:
  - OverlayItem createltem(int i) // i.-dik elem
  - int size() // listánk elemszáma
  - + : void addOverlay(OverlayItem overlay) // hozzáadás

## ⦿ Overlay hozzáadása a térképhez

```
MyItemizedOverlay pointsOverlay =  
    new MyItemizedOverlay(pointDefaultMarker); //Drawable  
List<Overlay> mapOverlays = myMap.getOverlays();  
mapOverlays.add(pointsOverlay);
```

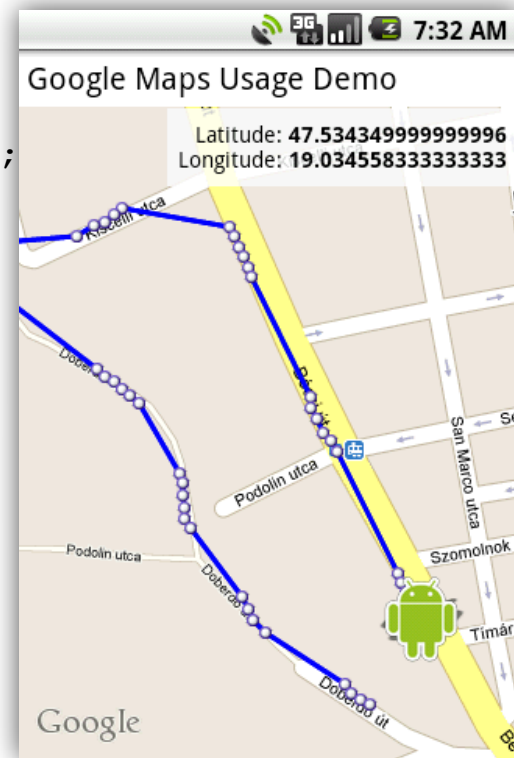
## ⦿ Overlay feltöltése adatokkal

```
OverlayItem item = new OverlayItem(new GeoPoint(  
    (int)(location.getLatitude() * 1E6),  
    (int)(location.getLongitude() * 1E6)  
), "Hello point", "Deatils...");  
pointsOverlay.addOverlay(item);  
myMap.invalidate();
```

# Rajzolás térképre

## © Adott Overlay draw() metódusának felüldefiniálása:

```
public void draw(Canvas canvas, MapView mapView, boolean shadow) {
    if(overlays.size() > 0 && projection != null){
        Point startPoint = new Point();
        projection.toPixels(overlays.get(0).getPoint(), startPoint);
        line.reset(); // Az első pont hozzáadása a Path-hoz = Kezdőpont
        line.moveTo(startPoint.x, startPoint.y);
        // Végig megyünk a pontokon és hozzáadjuk a Path-hoz
        for(OverlayItem i : overlays){
            Point point = new Point();
            projection.toPixels(i.getPoint(), point);
            line.lineTo(point.x, point.y);
        }
        // És akkor a lényeg, a rajzolás 😊
        canvas.drawPath(line, p);
    }
    super.draw(canvas, mapView, shadow);
}
```



# Oda-vissza geo kódoló

⊙ Lényege, hogy koordinátából címet, illetve címből koordinátát fejtünk vissza.

⊙ Geo kódoló:

```
Geocoder geoCoder =  
    new Geocoder(getBaseContext(), Locale.getDefault());  
List<Address> addresses = geoCoder.getFromLocation(  
    p.getLatitudeE6()/1E6, p.getLongitudeE6()/1E6, 1);  
// addresses.get(0).getAddressLine(i);
```

⊙ Visszakódoló:

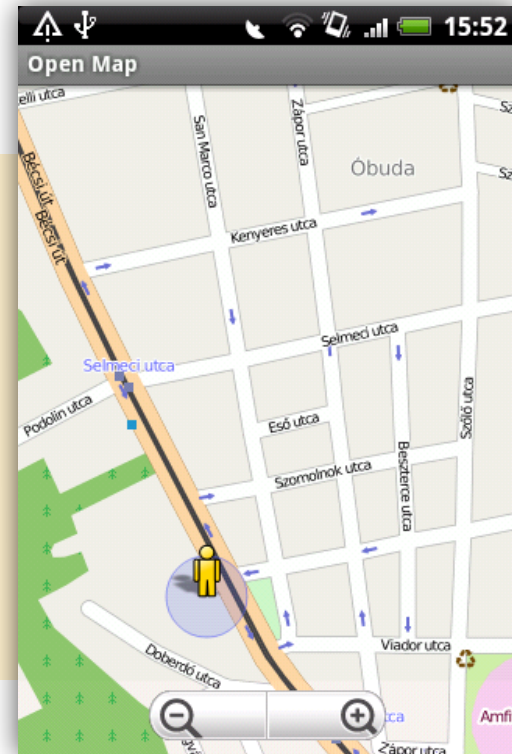
```
List<Address> addresses =  
    geoCoder.getFromLocationName(„Budapest, Bécsi út”, 5);  
p = new GeoPoint(  
    (int) (addresses.get(0).getLatitude() * 1E6),  
    (int) (addresses.get(0).getLongitude() * 1E6)  
);
```

# OpenStreetMap

- ⦿ Egy ingyenes alternatíva a Google Maps kiváltására.
- ⦿ Online és offline térkép adatbázis, ∃ vektorgrafikus is.
- ⦿ Android támogatás ([osmdroid](#)): hasonló MapView osztály, hasonló metódusokkal.
- ⦿ Nem kell API kulcs.

## Mint Google Maps esetén:

```
MapView myMap = new MapView(this, 256);  
MapController mc = myMap.getController();  
mc.setCenter(new GeoPoint(47533571,  
    19033851));  
mc.setZoom(17);
```





# Választható feladatok! 😊



## GPS alapú ébresztő | GPS Tracker program

GPS adatok elérése  
Térkép használata

# GPS alapú ébresztő

- ⊙ Legyen egy GPS alapú ébresztő óra, mely a következőkre képes.
- ⊙ Térképen kiválasztunk egy tetszőleges pontot.
- ⊙ Ezt a pontot rögzítsük, és helyezzük el a térképen!
- ⊙ Ha a kiválasztott pont közelébe értünk (pl.: távolság < 25 méter), akkor a telefon rezegjen!
- ⊙ Ez a távolság beállítható legyen egy külön beállítási felületen!
- ⊙ Az aktuális távolságot írja ki, valamelyik sarokba. Ennek ismeretében adjon egy közelítő becslést, a hátralévő időről! (Pl.: az utolsó 10 sebesség adatból)
- ⊙ Opcionális: ha már egyszer hatókörön belül voltam, de utána kívülre kerültem, akkor adjon valamilyen hangot is.
  - Emulátorhoz használhatóak a rögzített koordináták:  
<http...>

# GPS Tracker alkalmazás

- ⊙ Készíts egy GPS tracker alkalmazást, a következő képességekkel.
- ⊙ A kapott GPS koordinátákat tárolja le egy adatbázisba vagy egy fájlba (pl.: szabványos XML formátumban)
- ⊙ Közben rajzolja ki az eddig megtett útvonalat.
- ⊙ A korábban letárolt adat visszatölthető legyen, ami azt jelenti, hogy kirajzolja a térképre a koordinátákat.
- ⊙ Adatokból írjunk ki néhány extra információt, mint
  - átlag sebesség,
  - maximum sebesség,
  - mozgási idő,
  - megtett összes távolság