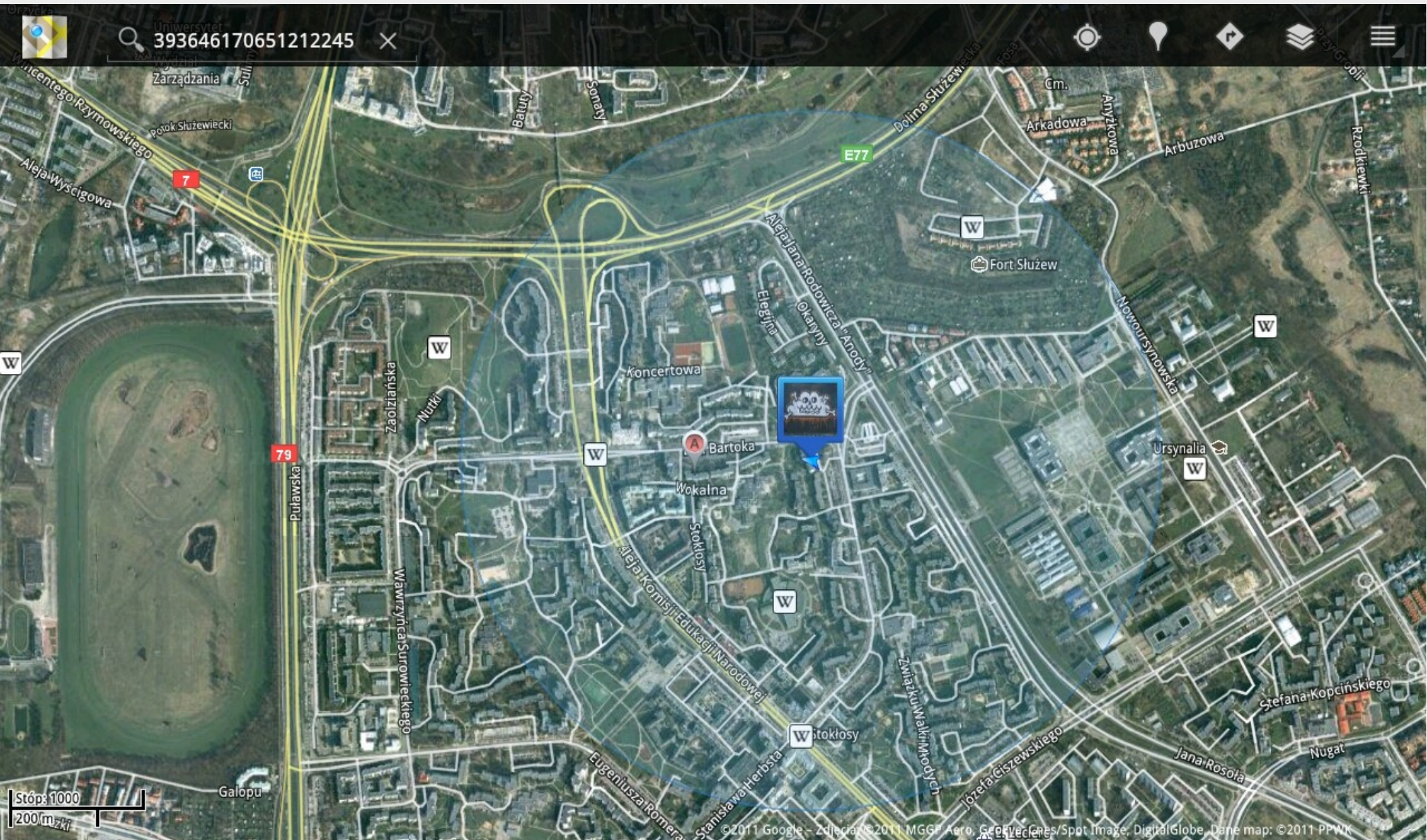


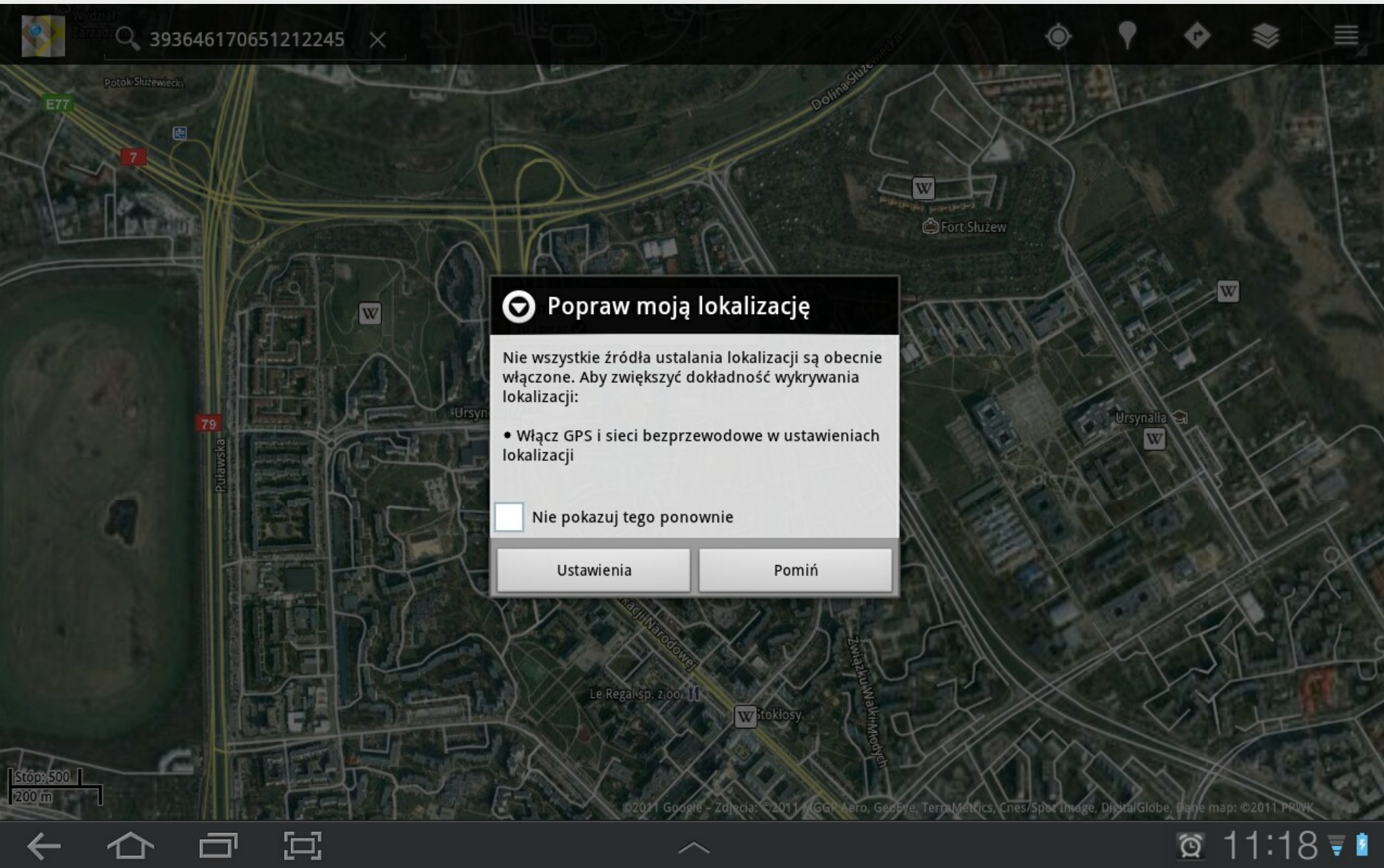
Geolokalizacja



Geolokalizacja - możliwości



Geolokalizacja - możliwości



Geolokalizacja - możliwości

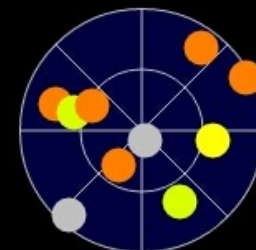
GPS Status

 GPS on

Accuracy (feet)

30₃

SNR

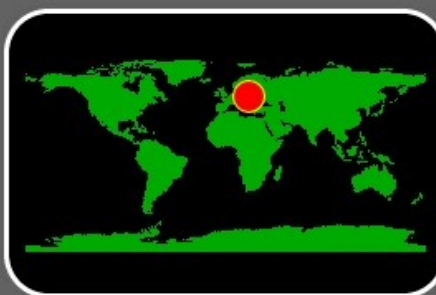
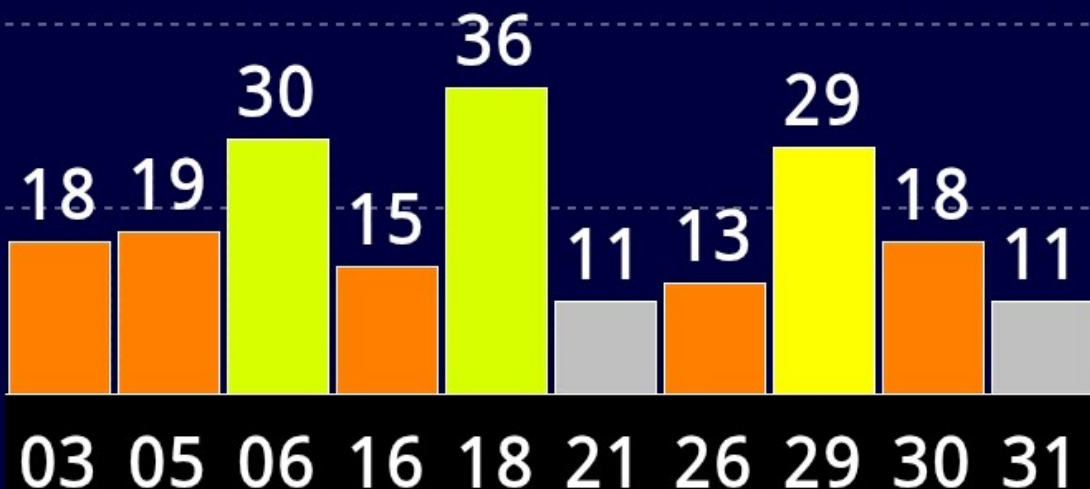


In View

10

In Use

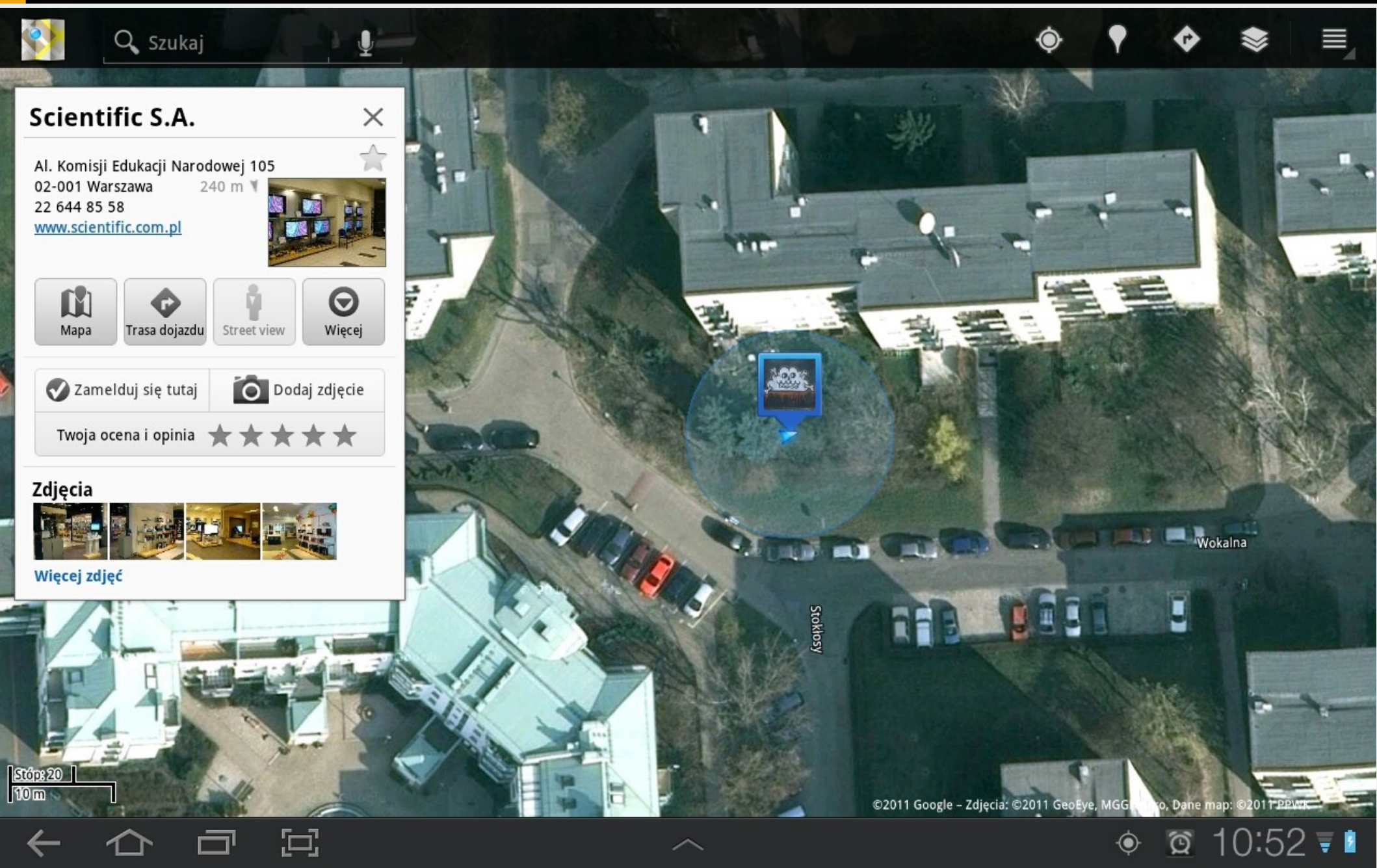
9



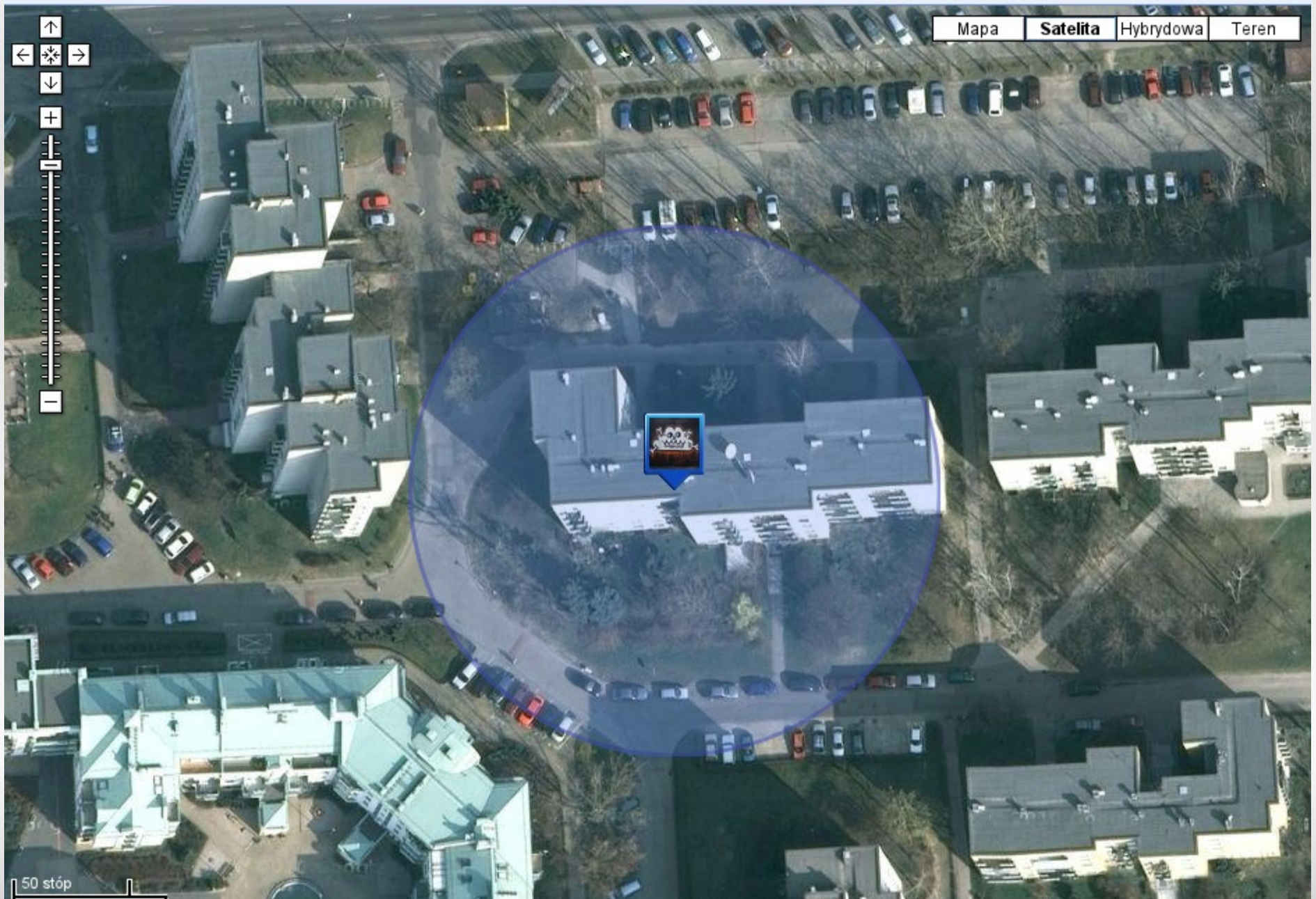
0

10:50

Geolokalizacja - możliwości



[Korzystaj ze Współrzędnych Google na swoim telefonie](#)



Geolokalizacja - możliwości


Zamelduj się

Tu i teraz

Wokalna 6, 01-001 Warszawa



Dodaj komentarz do miejsca zameldowania

 Znajomi we Współrzędnych



Zamelduj się tutaj

Następnym razem:

Melduj się tu automatycznie

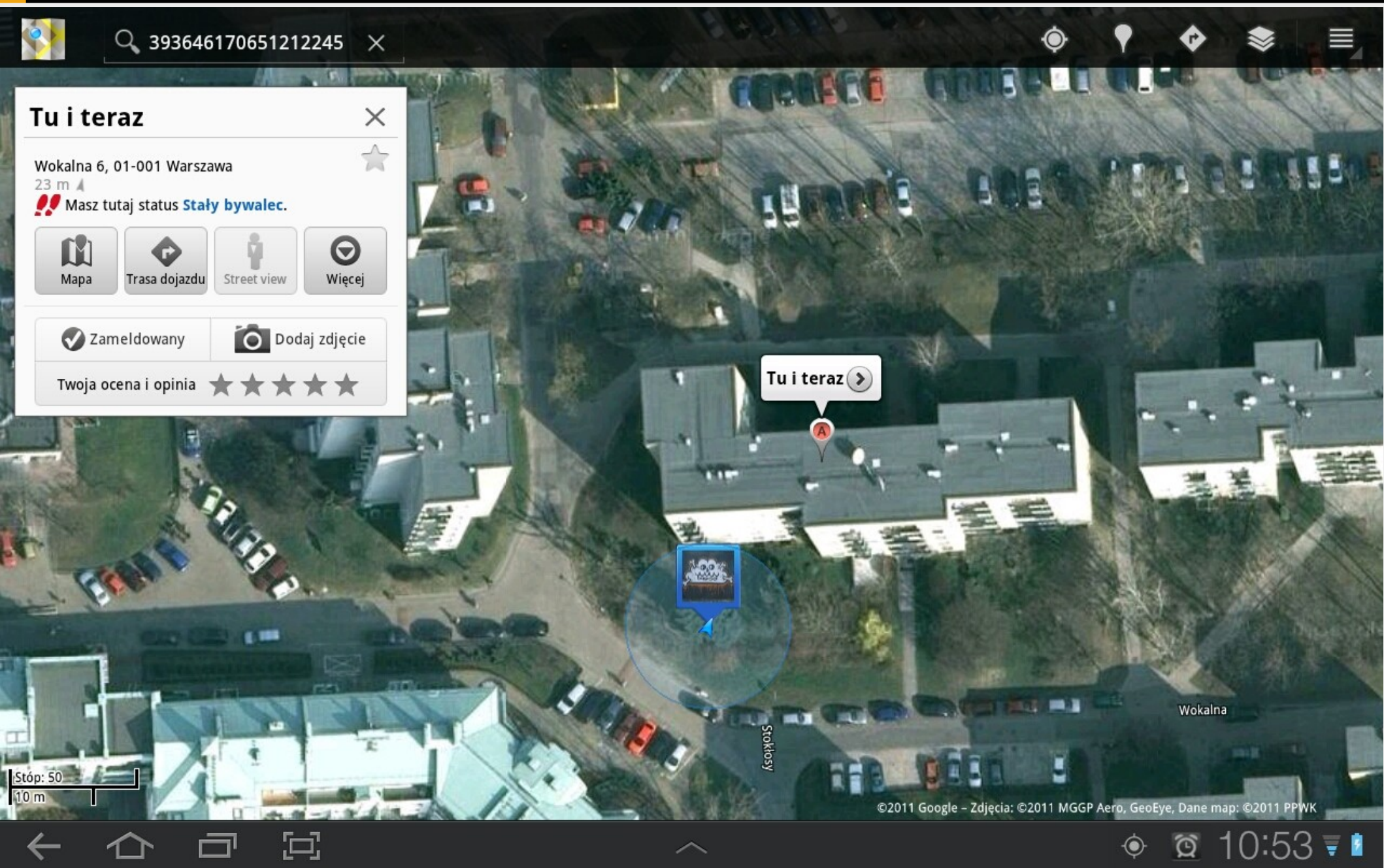
Przypomnij mi, aby się tutaj zameldować



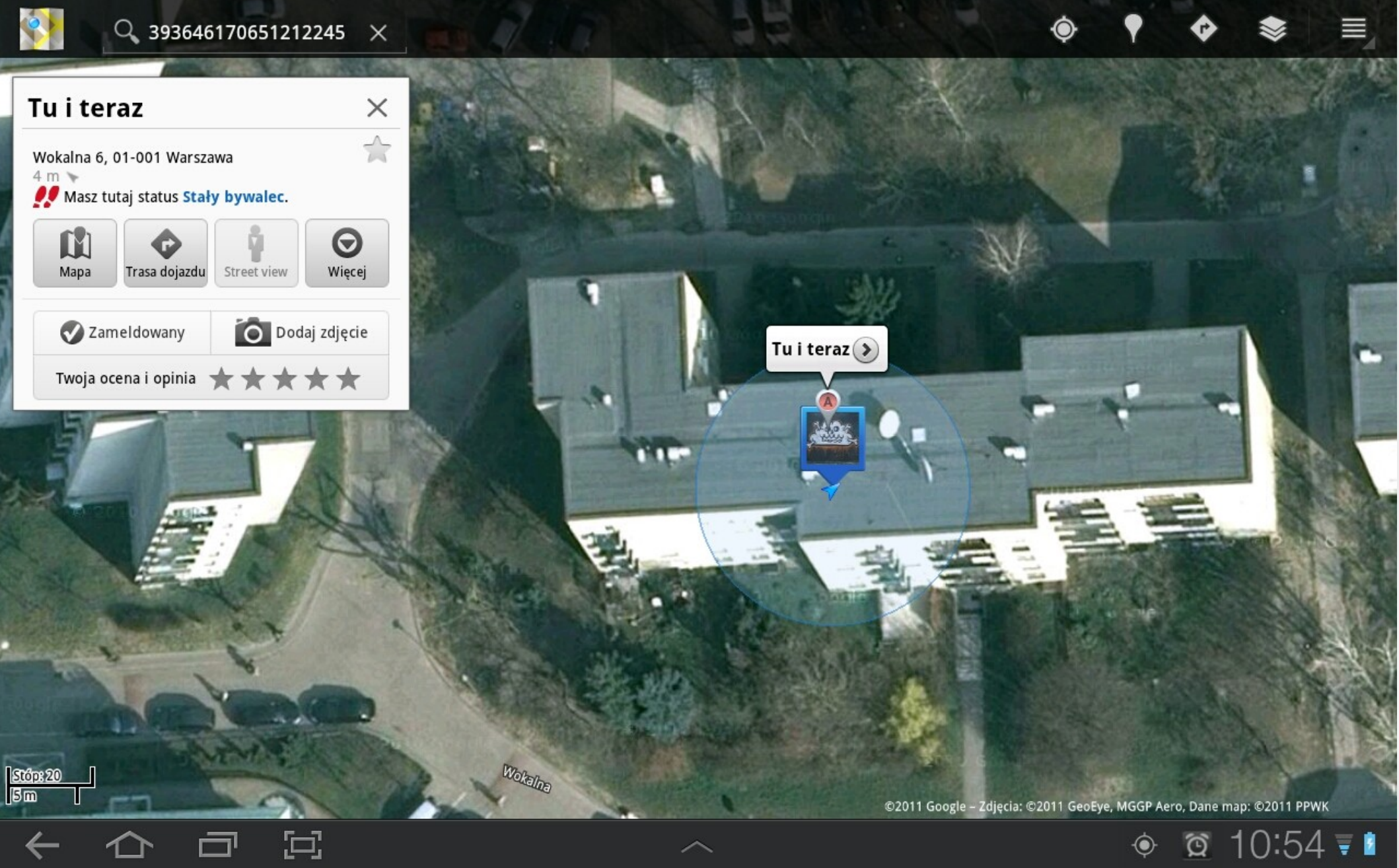
10:53



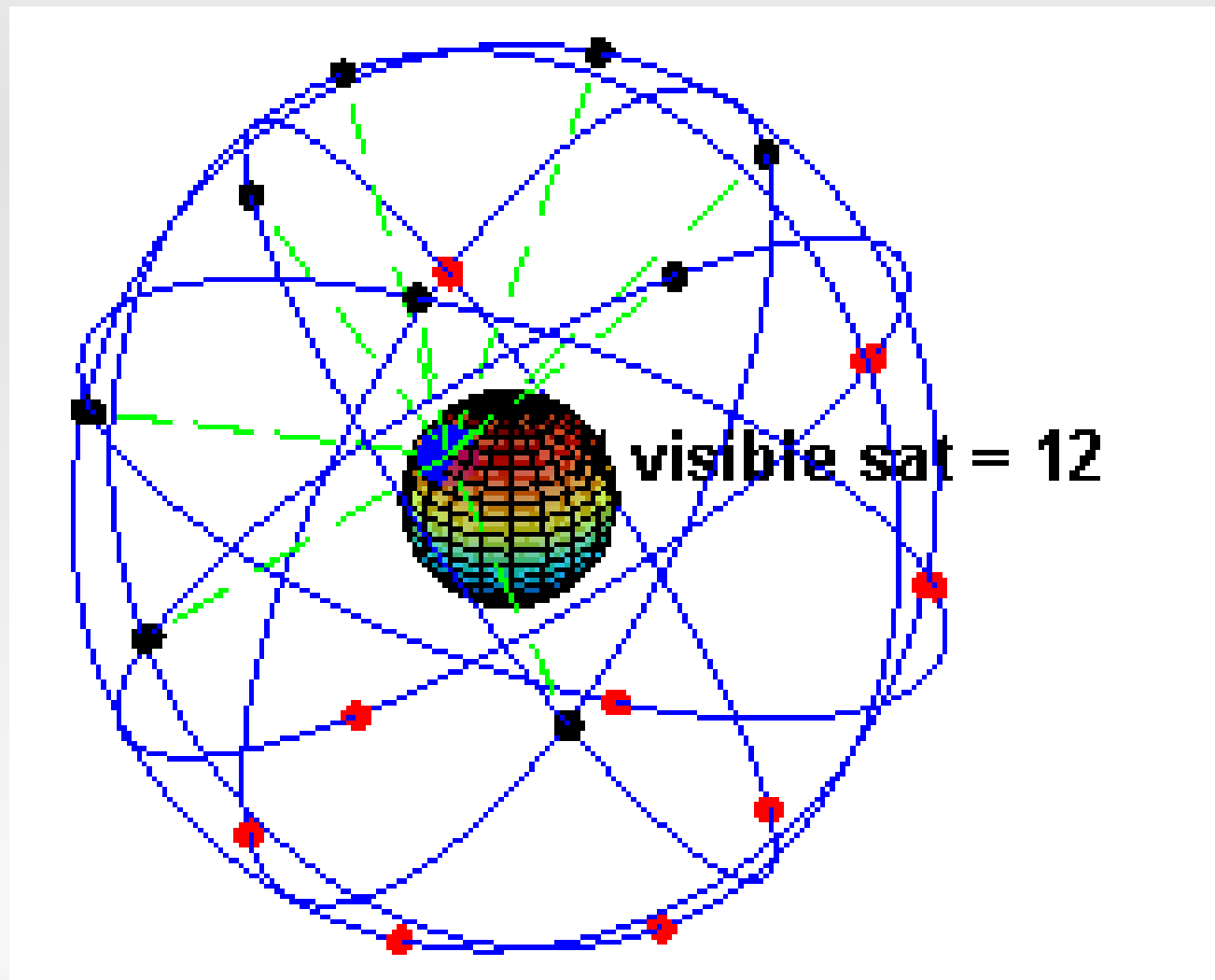
Geolokalizacja - możliwości



Geolokalizacja - możliwości



Technologie satelitarne



Technologie satelitarne

- GPS - Global Positioning System (USA) 31 satelitów wokół Ziemi na średniej orbicie okołoziemskiej (MEO 3000km-30000km) - 20200km
- GLONASS (ros. ГЛОНАСС , Глобальная навигационная спутниковая система; Globalnaja Nawigacionnaja Sputnikowaja Sistiema) 24 satelity docelowo 30 MEO - 19 100 km, aktualnie zastosowania cywilne (iphone, android)
- GALILEO (Europa) – 27+3 satelitów do MEO - 23 616 km – w trakcie budowy (2017-18)
- BEIDOU(Chiny) – w trakcie budowy 35 satelitów do 2020 (aktualnie 3+1 satelitów geostacjonarnych)

Dokładność lokalizacji

Usługi:

- android.permission.ACCESS_FINE_LOCATION – używana przez system GPS, stacje bazowe GSM oraz WiFi
- android.permission.ACCESS_COARSE_LOCATION – używana przez GSM lub WiFi. Niezbyt dokładna

w celu aktywacji w pliku konfiguracyjnym AndroidManifest.xml należy dopisać element `<uses-permission>` dla każdego typu usługi

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


GPS i Android

Interfejs `LocationListener` (`android.location`)

- `onLocationChanged(Location location)` – metoda wywoływana w momencie zmiany położenia.
- `onProviderDisabled(String provider)` – wyłączenie się dostawcy usługi
- `onProviderEnabled(String provider)` – włączenie się dostawcy usługi
- `onStatusChanged(String provider, int status, Bundle extras)` – metoda wywoływana w momencie zmiany stanu dostawcy (np. gdy dostawca stracił możliwość pobierania nowych danych).

```
public static final int OUT_OF_SERVICE = 0;
```

```
public static final int TEMPORARILY_UNAVAILABLE = 1;
```

```
public static final int AVAILABLE = 2;
```

Location Manager – przykład

```
private double[] getGPSArray() {
    LocationManager lm = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE); //pobranie managera
    List<String> providers = lm.getProviders(true); //wylistowanie dostawców

    Location l = null;

    for (int i = providers.size() - 1; i >= 0; i--) {
        l = lm.getLastKnownLocation(providers.get(i)); //sprawdzenie lokalizacji
        if (l != null)
            break;
    }

    double[] gps = new double[4];
    if (l != null) {
        gps[0] = l.getLatitude();
        gps[1] = l.getLongitude();
        gps[2] = l.getAltitude();
        gps[3] = l.getAccuracy();
    }
    return gps;
}
```


Location Manager cd

Utworzenie obiektu managera

```
LocationManager locationManager = (LocationManager)  
getSystemService(Context.LOCATION_SERVICE);
```

Utworzenie obiektu słuchacza

```
final MyLocationListener locationManager = new  
MyLocationListener();
```

Uruchomienie nasłuchiwanie zmian (5000ms, 10M dokładności)

```
locationManager.requestLocationUpdates(LocationManager.GPS_  
PROVIDER,  
5000, 10, locationManager);
```

Implementacja Słuchacza

```
class MyLocationListener implements LocationListener {

    private double latitude = 0;
    private double atitude = 0;
    private double lognitude = 0;

    @Override
    public void onLocationChanged(Location loc) {
        latitude = loc.getLatitude();
        atitude = loc.getAltitude();
        lognitude = loc.getLongitude();
    }

    public double[] getLocation() {
        double[] locations = { latitude, lognitude, atitude };
        return locations;
    }

    @Override
    public void onProviderDisabled(String arg0) {}

    @Override
    public void onProviderEnabled(String arg0) {}

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {}
}
```