

Aplicatii mobile pentru afaceri

Master SIA

Lect. Octavian Dospinescu

Tematica generala curs

- GPS
- GoogleMaps
 - Marcare puncte pe harta (map overlay);
 - Marcarea traseelor pe harta

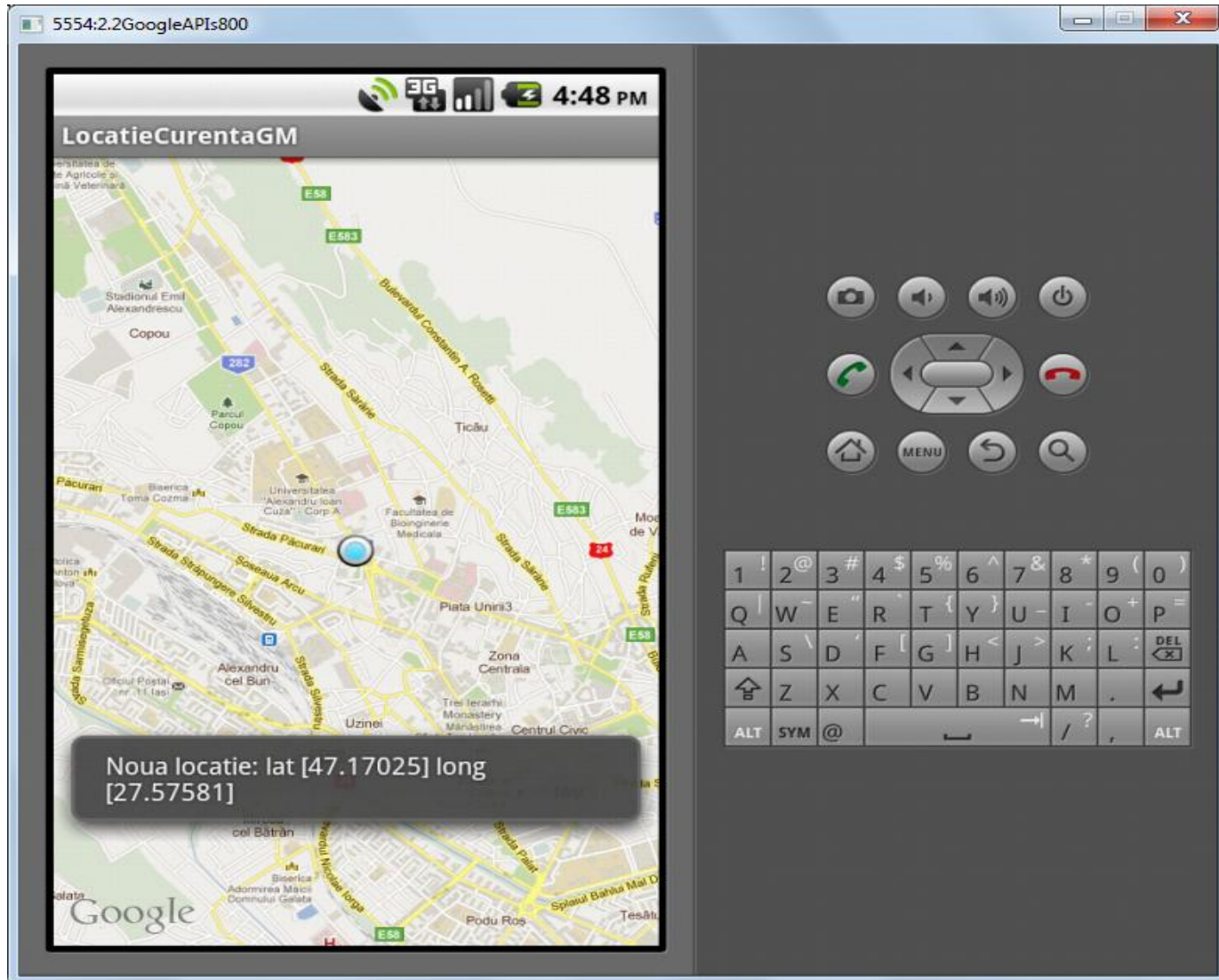
Tematica generala curs

- Manipulare GoogleMaps
- Clase specifice
 - MyLocationOverlay
 - MapActivity
 - MapView

Scopuri

- afisare pe harta a locatiei curente si marcarea acestei locatii
- actualizare locatie curenta in functie de deplasarea in spatiu a dispozitivului
- marcarea trasee pe harta
- integrari ulterioare cu alte componente si servicii

Scopuri (reprezentate vizual 😊)



(c) Octavian Dospinescu & Marian Perca
2012-2013 pentru Master SIA

Precizari de inceput

- utilizatorul interactioneaza cu un **MapActivity**
- acel MapActivity foloseste un **MapView** (care a fost definit in main.xml);
- pe o harta pot fi mai multe puncte “de marcaj” (map overlays);
- fiecare MapView are o colectie de puncte de tip Overlays;
- colectia se poate obtine/folosi prin metoda **.getOverlays();**

Clasa **MyLocationOverlay**

- Constructor – 2 argumente necesare:
 - Contextul (de regula **this** sau **getApplicationContext()**);
 - MapView-ul caruia ii este destinat

- Exemplu:

```
locatiaMea = new MyLocationOverlay(this, harta);
```

Clasa **MyLocationOverlay**

- Constructorul nu adauga automat punctul de marcaj in cadrul MapView-ului;
- Trebuie folosita o metoda **explicita** de adaugare a punctului de marcaj in colectia de marcaje ale mapView-ului.

```
hartaGM.getOverlays().add(locatiaMea);
```


Clasa MyLocationOverlay

- Metode de interes:
 - `.getMyLocation()` – ofera obiectul care contine locatia memorata;
 - `.enableMyLocation()` – activeaza cererea de obtinere a locatiei;
 - `.disableMyLocation()` – dezactiveaza cererea de obtinere a locatiei;
 - `.runOnFirstFix()` – este metoda care ne permite sa rulam o secventa personalizata de indata ce este gasita o locatie

Arhitectura generala a aplicatiei

- Fisierul main.xml contine descrierea MapView-ului:

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.maps.MapView
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/hartaGM"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:clickable="true"
  android:apiKey="cheia proprie ☺"/>
```

Arhitectura generala a aplicatiei

- Drepturi specifice in AndroidManifest.xml:
 - android.permission.INTERNET
 - android.permission.ACCESS_FINE_LOCATION
 - import biblioteca:
`<uses-library android:name="com.google.android.maps"/>`

Arhitectura generala a aplicatiei

- In clasa MapActivity declaram membri specifici pentru MapView si MyLocationOverlay

```
public class LocatieCurentaGMActivity extends MapActivity {  
  
    MapView hartaGM = null;  
    MyLocationOverlay locatiaMea = null;  
    LocationManager locManager = null;  
    LocationListener locListener = null;  
    //... ..  
}
```

Arhitectura generala a aplicatiei

- In metoda **OnCreate()** definim comportamentul initial al aplicatiei

@Override

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    // realizam referinta catre harta din xml  
    hartaGM = (MapView) findViewById(R.id.hartaGM);  
    // afisam controalele pentru zoom  
    hartaGM.setBuiltInZoomControls(true);  
    // setam nivelul detalieri  
    hartaGM.getController().setZoom(15);  
    // initializez obiectul Overlay  
    locatiaMea = new MyLocationOverlay(this, hartaGM);  
    // il adaug pe harta  
    hartaGM.getOverlays().add(locatiaMea);  
    hartaGM.postInvalidate(); //asigura redesenarea hartii  
    locManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
```

Arhitectura generala a aplicatiei

- Definim un listener specific pentru managerul de locatii (locManager)

```
locListener = new LocationListener() {  
    public void onLocationChanged(Location location) {  
        afiseazaLocatie(location);  
    }  
    public void onProviderDisabled(String provider) {  
    }  
    public void onProviderEnabled(String provider) {  
    }  
    public void onStatusChanged(String provider, int status,  
        Bundle extras) {  
    }  
};
```

Arhitectura generala a aplicatiei

- Definim o procedura care va fi invocata atunci cand se modifica locatia curenta (vezi slide anterior)

```
private void afiseazaLocatie(Location locatie) {
    if (locatie != null) {
        double lat = locatie.getLatitude();
        double lng = locatie.getLongitude();
        GeoPoint pctLocatie = new GeoPoint((int) (lat * 1000000),
            (int) (lng * 1000000));

        Toast.makeText(getBaseContext(), "Noua locatie: lat [" + lat + "] long [" + lng + "]",
            Toast.LENGTH_SHORT).show();

        hartaGM.getController().animateTo(pctLocatie);
    }
}
```

Arhitectura generala a aplicatiei

- Interceptam evenimentul onResume() si definim comportamentul aplicatiei

```
@Override
    public void onResume() {
        super.onResume();
        // preiau ultima locatie si afisez pe harta daca exista
        Location ultimaLocatieCunoscuta = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        afiseazaLocatie(ultimaLocatieCunoscuta);

        // solicit utimele informatii privind locatia
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
            locListener);
        locatiaMea.enableMyLocation();
        // de indata ce gasesc o locatie voi centra harta in acel punct
        locatiaMea.runOnFirstFix(new Runnable() {
            public void run() {

                hartaGM.getController().setCenter(locatiaMea.getMyLocation());
            }
        });
    }
}
```


Implementare practica 😊

Posibilitatea de a obtine 1.5 puncte
in plus la nota finala.... sunteti
interesati?

Trasee pe harta

- O prezentare succinta in PowerPoint privind conceptele principale
- O implementare simpla
- Link util:
 - <http://stackoverflow.com/questions/2176397/drawing-a-line-path-on-google-maps>
- Termen: 29 dec. 2012
- Maxim (primele) 2 persoane 😊