

# *Développement Application Android*

## *Capteurs*

Anthony Chomienne

CPE Lyon



2023

# DÉVELOPPEMENT APPLICATION ANDROID — PLAN

1 GEOLOCALISATION

2 CARTOGRAPHIE

3 CAPTEURS

4 CONCLUSION

# PERMISSIONS

- `<uses-feature  
android:name="android.hardware.location.gps"/>`
- `<uses-permission android:name=  
"android.permission.ACCESS_COARSE_LOCATION"/>`
- `<uses-permission android:name=  
"android.permission.ACCESS_FINE_LOCATION"/>`

# PERMISSION À DEMANDER 1/2

```
if (ActivityCompat.checkSelfPermission( this,
    Manifest.permission.ACCESS_FINE_LOCATION) !=
    PackageManager.PERMISSION_GRANTED) {
    String[] permissions =
        {Manifest.permission.ACCESS_FINE_LOCATION};
    ActivityCompat.requestPermissions(this,
        permissions, 1);
}
```

## PERMISSION À DEMANDER 2/2

```
@Override
public void onRequestPermissionsResult(int
    requestCode, @NonNull final String[]
    permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode,
        permissions, grantResults);
    if (grantResults[0] ==
        PackageManager.PERMISSION_GRANTED) {
        //on a la permission
    } else {
        //afficher un message d'erreur
    }
}
```

# DEMANDER LA LOCALISATION 1/2

```
LocationManager manager = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);
manager.requestLocationUpdates(
    LocationManager.GPS_PROVIDER, 120, 100,
    myLocationListener);
manager.requestLocationUpdates(
    LocationManager.NETWORK_PROVIDER, 120, 100,
    myLocationListener);
```

## DEMANDER LA LOCALISATION 2/2

```
LocationListener myLocationListener = new
    LocationListener() {
        @Override
        public void onLocationChanged(Location newLocation) {

        }

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

        }

        @Override
        public void onProviderEnabled(String provider) {

        }

        @Override
        public void onProviderDisabled(String provider) {

        }
    }
```

# ARRÊTER LA LOCALISATION

```
manager.removeUpdates(myLocationListener);
```



# DÉVELOPPEMENT APPLICATION ANDROID — PLAN

## 1 GEOLOCALISATION

## 2 CARTOGRAPHIE

- Google Maps
- Open Street Map

## 3 CAPTEURS

## 4 CONCLUSION

# DÉVELOPPEMENT APPLICATION ANDROID — PLAN

## 1 GEOLOCALISATION

## 2 CARTOGRAPHIE

- Google Maps
- Open Street Map

## 3 CAPTEURS

## 4 CONCLUSION

# GRADLE (APP)

```
dependencies {  
    implementation  
        'com.google.android.gms:play-services-maps:17.0.1'  
}
```

# GRADLE (PROJET)

```
buildscript {  
    //....  
    dependencies {  
        classpath  
            'com.google.android.libraries.mapsplatform.  
            secrets-gradle-plugin:secrets-gradle-plugin:2.0.0'  
    }  
}
```

# ANDROIDMANIFEST.XML: PERMISSIONS

- android.permission.ACCESS\_NETWORK\_STATE
- android.permission.INTERNET
- android.permission.WRITE\_EXTERNAL\_STORAGE

# PERMISSION: META-DATA

```
<meta-data
    android:name= "com.google.android.gms.version"
    android:value=
        "@integer/google_play_services_version" />
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="${MAPS_API_KEY}" />
```

# LOCAL.PROPERTIES

```
MAPS_API_KEY=Votre_Clef_D_API
```

# LAYOUT

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android=
    "http://schemas.android.com/apk/res/android"
        android:name=
            "com.google.android.gms.maps.SupportMapFragment"
        android:id= "@+id/map"
        android:layout_width= "match_parent"
        android:layout_height= "match_parent"/>
```



```
SupportMapFragment mapFragment =  
    SupportMapFragment.newInstance();  
FragmentManager manager = getSupportFragmentManager();  
manager.beginTransaction()  
    .replace(R.id.map, mapFragment)  
    .commit();
```

# ACTIVITY

```
OnMapReadyCallback callback = new
    OnMapReadyCallback() {
        @Override
        public void onMapReady(GoogleMap map)
        {
            //la carte est chargé on peut interagir
            avec map
        }
    }

MapFragment mapFragment =
    getSupportFragmentManager().
        findFragmentById(R.id.map);
mapFragment.getMapAsync(callback);
```

# INTERACTION

- `GoogleMap.setLocationSource(LocationSource);`
- Deux méthodes à surcharger
- `public void activate(OnLocationChangeListener listener);`
- `public void deactivate();`
- Appeler la fonction `onLocationChanged` du `OnLocationChangeListener` reçu précédemment via `activate`
- `public void onLocationChanged(Location newLocation);`

# MARKER ET AUTRES ÉLÉMENTS 1/3

- `MarkerOptions options;`
- `options.position(LatLng position);`
- `options.icon(Bitmap icon);`
- `options.title(String title);`

## MARKER ET AUTRES ÉLÉMENTS 2/3

- `PolygonOptions options;`
- `options.add(LatLng position);`
- `options.addAll(Iterable<LatLng> positions);`
- `options.fillColor(int color);`
- `options.strokeColor(int color);`

## MARKER ET AUTRES ÉLÉMENTS 3/3

- `CircleOptions options;`
- `options.center(LatLng position);`
- `options.radius(double radius);`
- `options.title(String title);`
- `options.fillColor(int color);`
- `options.strokeColor(int color);`

# DÉVELOPPEMENT APPLICATION ANDROID — PLAN

## 1 GEOLOCALISATION

## 2 CARTOGRAPHIE

- Google Maps
- Open Street Map

## 3 CAPTEURS

## 4 CONCLUSION

# GRADLE (APP)

```
dependencies {  
    implementation  
        'org.osmdroid:osmdroid-android:6.1.10'  
}
```



# PERMISSIONS

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

la permission WRITE\_EXTERNAL\_STORAGE est à demander à l'utilisateur

# LAYOUT

```
<org.osmdroid.views.MapView  
    android:id="@+id/map_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
/>
```

## ACTIVITY/FRAGMENT

```
//Récupération du context dans le cas d'une  
Activity  
Context contex = this; //marche aussi avec  
getApplicationContext();  
//Récupération du context dans le cas d'un  
Fragment  
Context contex = binding.getRoot().getContext();  
Configuration.getInstance().load(context,  
    PreferenceManager.  
        getDefaultSharedPreferences(context));
```

# FRAGMENT OU ACTIVITY

```
@Override
public void onCreate(Bundle savedInstanceState) {
    //...
    //...
    binding.mapView.setTileSource(
        TileSourceFactory.MAPNIK); //source des
        données cartographique
}
```

## FRAGMENT OU ACTIVITY 2/2

```
@Override  
public void onResume() {  
    super.onResume();  
    binding.mapView.onResume();  
}
```

```
@Override  
public void onPause() {  
    super.onPause();  
    binding.mapView.onPause();  
}
```

# GEOLOCALISATION

```
this.mLocationOverlay = new  
    MyLocationNewOverlay(new  
        GpsMyLocationProvider(context), binding.mapView);  
this.mLocationOverlay.enableMyLocation();  
binding.mapView.getOverlays()  
    .add(this.mLocationOverlay);
```

# MARKER

```
Marker marker = new Marker(map);
marker.setPosition(new
    GeoPoint(latitude, longitude));
marker.setAnchor(Marker.ANCHOR_CENTER,
    Marker.ANCHOR_CENTER);
marker.setIcon(getResources()
    .getDrawable(R.drawable.ic_launcher));
marker.setTitle("Start point");
binding.mapView.getOverlays().add(marker);
```

# POLYGON

```
List<GeoPoint> geoPoints = new ArrayList<>();  
Polygon polygon = new Polygon();  
polygon.setFillPaint()  
    .setColor(Color.parseColor("#1EFFE70E"));  
polygon.setPoints(geoPoints);  
polygon.setTitle("A sample polygon");
```



# DÉVELOPPEMENT APPLICATION ANDROID — PLAN

1 GEOLOCALISATION

2 CARTOGRAPHIE

3 CAPTEURS

4 CONCLUSION

# SENSORMANAGER

```
SensorManager sensorManager = (SensorManager)
    getSystemService(Context.SENSOR_SERVICE);
if (sensorManager.getDefaultSensor(
    Sensor.TYPE_MAGNETIC_FIELD) != null) {
    // Capteur disponible
} else {
    // Capteur absent!
}
```

# SENSOREventListener

```
SensorEventListener listener = new SensorEventListener() {  
    @Override  
    public final void onAccuracyChanged(Sensor sensor, int  
        accuracy) {  
        // Que faire si la précision change  
    }  
  
    @Override  
    public final void onSensorChanged(SensorEvent event) {  
        //la taille de values dépend du capteur  
        //lumière/proximité 1 seule valeur  
        //gyroscope, accelero,... 3 (Axes X,Y,Z)  
        float X = event.values[0];  
        float Y = event.values[1];  
        float Z = event.values[2];  
    }  
};
```

# UTILISATION

```
@Override
protected void onResume() {
    super.onResume();
    Sensor sensor = sensorManager.getDefaultSensor(
        Sensor.TYPE_MAGNETIC_FIELD);
    sensorManager.registerListener(this, sensor,
        SensorManager.SENSOR_DELAY_NORMAL);
}

@Override
protected void onPause() {
    super.onPause();
    sensorManager.unregisterListener(this);
}
```

# DÉVELOPPEMENT APPLICATION ANDROID — PLAN

1 GEOLOCALISATION

2 CARTOGRAPHIE

3 CAPTEURS

4 CONCLUSION

# SOURCES ET BIBLIO

- <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- **Part de marché Mobile en France Janvier 2015**
- <http://opensignal.com/reports/2015/08/android-fragmentation/>
- <http://developer.android.com/>
- <http://www.statista.com/>
- <https://medium.com/@ankit.sinhal/mvc-mvp-and-mvvm-design-pattern-6e169567bbad>

Licence : 