mapsforge

OpenStreetMap rendering on Android

Computer Science Institute
Freie Universität Berlin

Agenda

- Quick introduction of the *mapsforge* project
- OpenStreetMap on Android

- Our approach to mobile map rendering
- Our free map library for Android
- Our free map application for Android
- Hello, MapView

- Routing with OpenStreetMap data
- Questions and live demo

About mapsforge

- The mapsforge project
  - initiated at the computer science institute of Freie Universität Berlin
  - was started in 2008
  - provides free mapping and navigation tools
  - is based on OpenStreetMap and free software
  - currently involves 10 students and scientific staff
  - since April 2010 at Google Project Hosting
  - uses the GPL3 license for all source code

http://mapsforge.org

What we do

- Creating various tools for
  - Map rendering and overlays
  - Route planning
  - POI indexing and search
  - Spatial data formats and algorithms

- Focus is on mobile devices
  - Currently only Android
  - In the future also other platforms

- We aim for maximum flexibility
  - Configurable map extraction and rendering
  - Configurable navigation and routing
  - Online and offline services

OpenStreetMap on Android

- The OpenStreetMap (OSM) project
  - Creative Commons license
  - Map data can be downloaded as XML files
  - Rendered tiles available as PNG files
- Many applications for OSM on Android exist
  - Not all of them are free software
  - Only few can work completely offline
  - Only very few render map images on the device
- Advantages of rendering OpenStreetMap offline
  - No internet connection required
  - Saving money and privacy
  - Much more flexible than static tiles

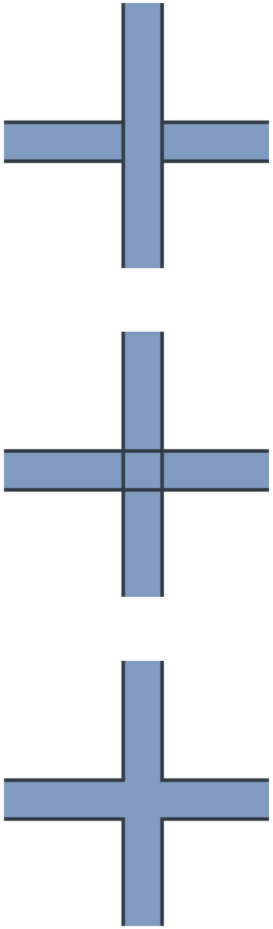
Our approach to mobile map rendering

- All map data is stored on the device
  - Preprocessing of OSM data with a converter
    - Input: Raw OSM map data (Berlin: ~170 MB)
    - Output: Single binary file
      (Berlin: 15 MB, full details including names)
    - No pre-rendering of images or tiles
    - Finally the map file is copied to SD card
- Tile based rendering
  - Tile = small, rectangular part of a map
  - De facto standard for online maps
  - Advantages
    - Incremental map rendering
      → better user experience
    - Allows caching
      → higher frame rate

U Kaiserdamm
S Messe Nord / ICC
S Westkreuz

Challenges of mobile map rendering

- Rendering a map tile is difficult
  - Get all map **data** that **intersects** it
  - Correct handling of **overlapping** map objects
  - Fast but accurate map **projection**
  - **Filter** map objects by zoom level
  - Collision free **label placement**
  - **Scheduling** of all required tiles
  - …



Android specific challenges

- Memory Management
  - Only 16 MB of RAM per process (or 24 MB, depending on the device)
  - Return value of `Runtime.freeMemory()` not reliable
  - Be aware of memory leaks

- Issues with Android activity lifecycle
  - No single exit point (application maybe killed by OS)
  - Delayed execution of `onStop()` and `onDestroy()`
    → potential race conditions
    → potential OutOfMemoryException
  - Handling of configuration change (e.g. orientation, language)

Android specific challenges (continued)

- Performance tuning for Dalvik VM
  - Know your development platform
    → Designing for Performance
    → A JIT Compiler for Android's Dalvik VM (since version 2.2)

- Avoid garbage collection
  - Consider object pools
  - Class variables instead of local variables

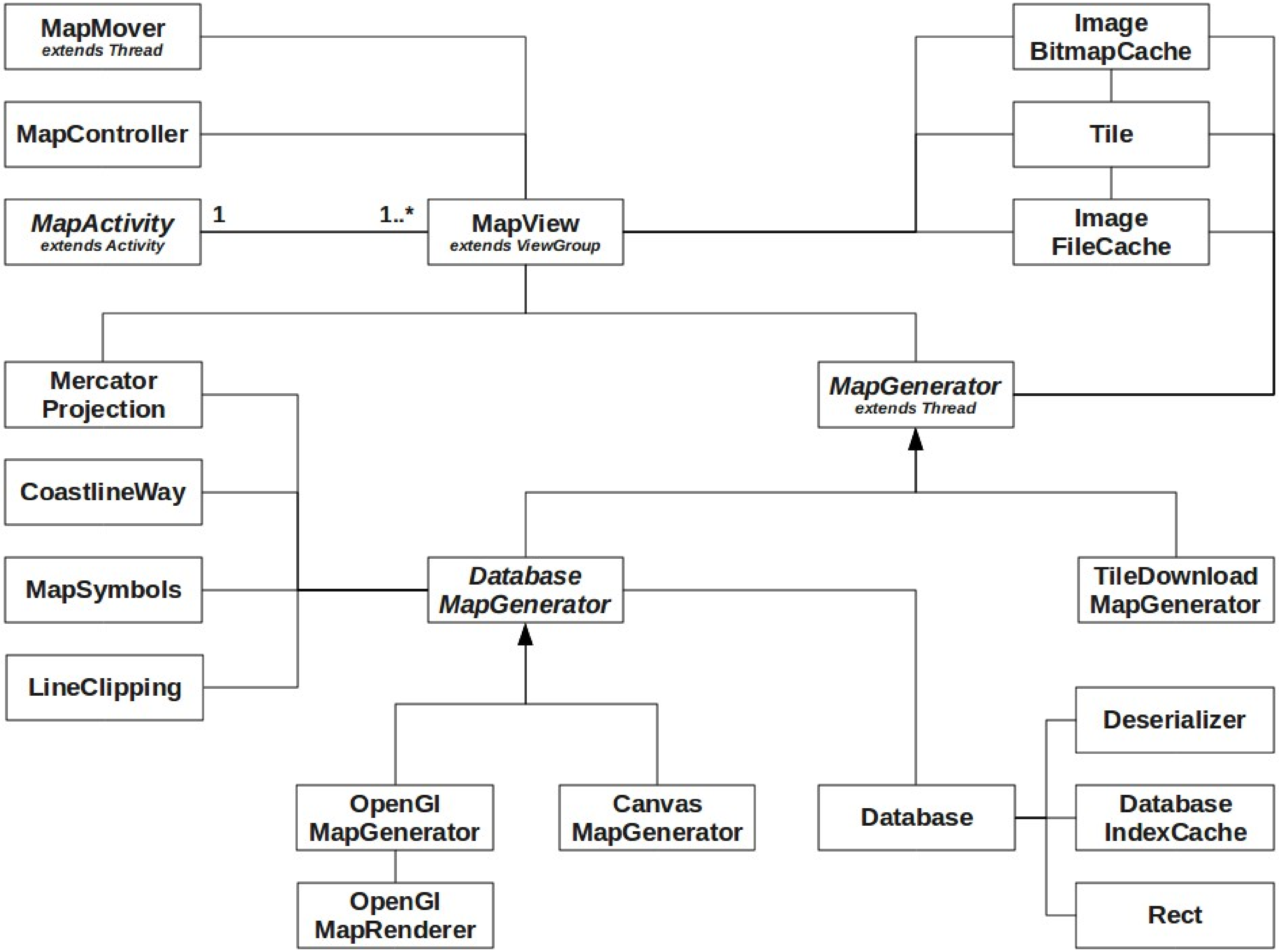
- Multiple Android versions and devices
  - Be aware of market fragmentation (regarding different versions)
  - Different screen resolutions and densities
  - Device-specific bugs :-(

Our free map library for Android

- Library for rendering map tiles on different zoom levels
  - Implements important classes and methods from the Google Maps for Android API
  - High level of detail and rendering quality
    - Subset of ~170 OpenStreetMap tags
    - Each tag can be rendered individually with name and symbol
  - Intelligent caching of tiles in RAM and on SD card
  - Integrated map scale and zoom controls

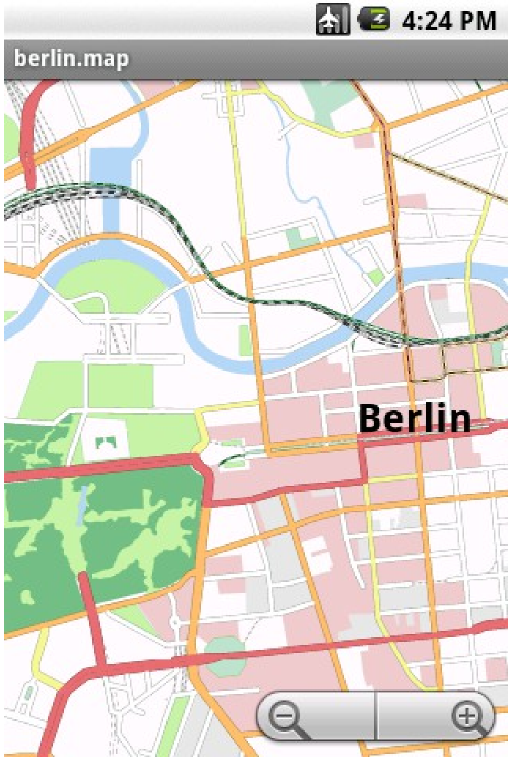
- Key class is the MapView
  - Handles user input via touchscreen, keys or trackball
  - Assembles tiles into one big image on the screen
  - Achieves 30-40 FPS (depending on cache hits)

Architectural View of Map Rendering



Our free map application for Android

- Working title: AdvancedMapViewer
  - Runs on all Android versions ≥ 1.5
  - Uses our map library and MapView
  - Serves as demo application and test case
- Some features
  - Select map file in a file browser
  - Configuration via preferences
  - Follow current GPS position
  - Enter new coordinates
  - Currently included languages: English, German, Finnish
- You can download the APK file and several preprocessed map files at our website



Hello, MapView (Google Version)

```
import android.os.Bundle;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapView;

public class HelloMapView extends MapActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        MapView mapView = new MapView(this, "Your Maps API Key");
        mapView.setClickable(true);
        mapView.setBuiltInZoomControls(true);
        setContentView(mapView);
    }

    @Override
    protected boolean isRouteDisplayed() {
        return false;
    }
}
```

AndroidManifest.xml:

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

Hello, MapView (mapsforge Version)

```
import android.os.Bundle;
import ~~com.google.android.maps.MapActivity~~ org.mapsforge.android.map.MapActivity;
import ~~com.google.android.maps.MapView~~ org.mapsforge.android.map.MapView;

public class HelloMapView extends MapActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        MapView mapView = new MapView(this, ~~"Your Maps API Key"~~);
        mapView.setClickable(true);
        mapView.setBuiltInZoomControls(true);
        mapView.setMapFile("/sdcard/berlin.map");
        setContentView(mapView);
    }

~~    @Override~~
~~    protected boolean isRouteDisplayed() {~~
~~        return false;~~
~~    }~~
}
```

New!

```
AndroidManifest.xml:
~~<uses-permission android:name="android.permission.INTERNET" />~~
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

What comes next

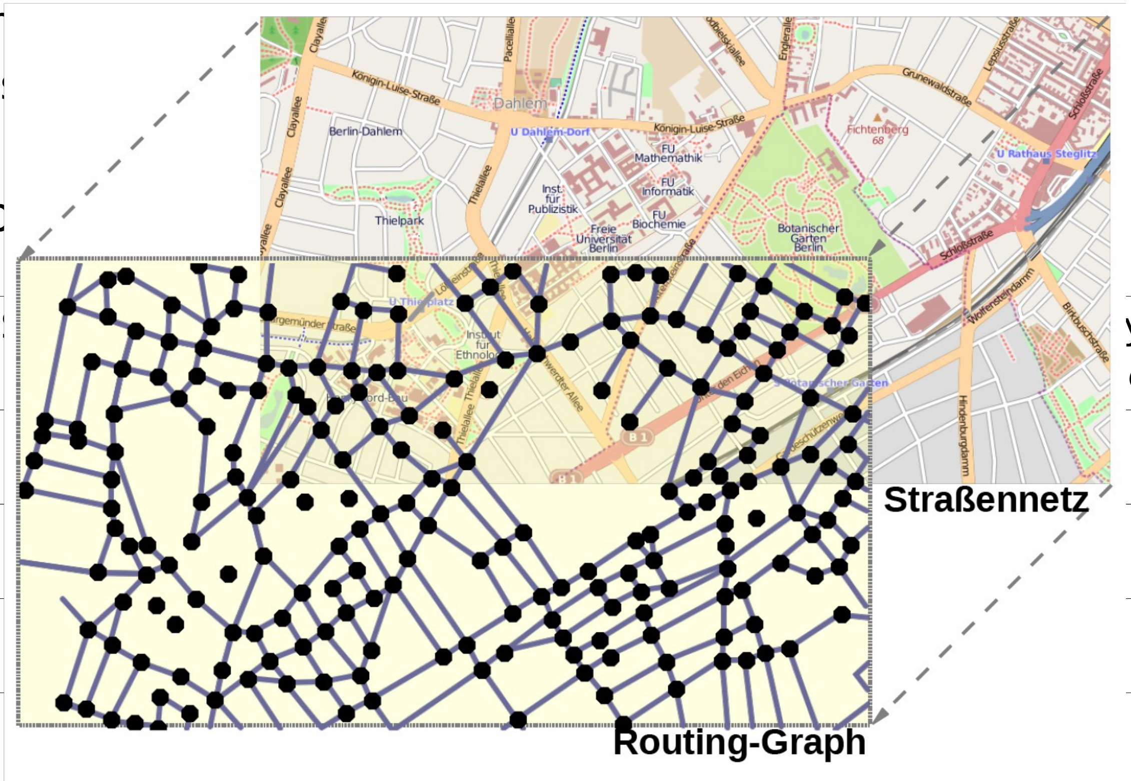
- Map rendering module
  - Fix known issues
  - Improve Overlay API
  - Use OpenGL for map rendering
    - Hardware acceleration
    - Better drawing controll
    - Maps in 3D?
  - Custom render styles and symbols
  - Better label placement algorithms
- Navigation modules
  - Offline routing with turn-by-turn navigation
  - Offline search and geocoding
- Support for other platforms

Routing with OpenStreetMap data

- Data quality and coverage
  - High quality and coverage in metropolitan regions
  - High quality due to imports of official or commercial data for some regions

- Routing graph extraction
  - Conditior
  - Heuristic

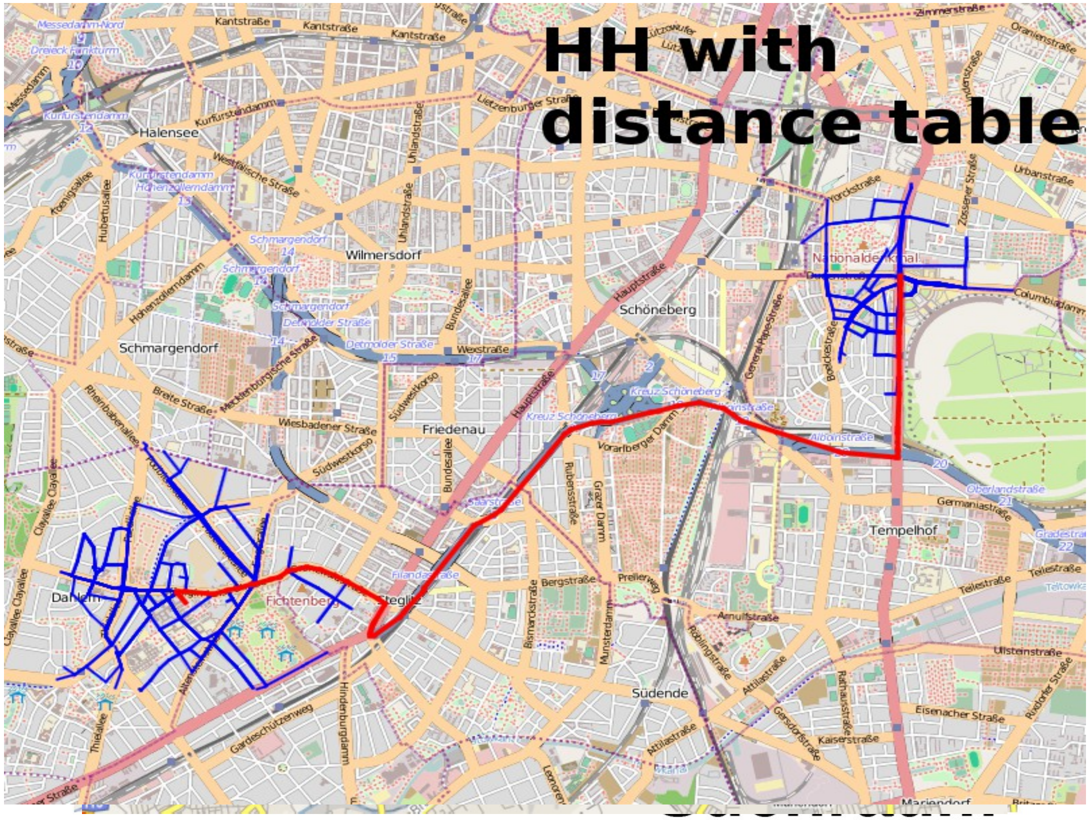
- Size of grap



Map data (O	y es)
Berlin	
Germany	
Europe	

Routing in large road networks

- Beneficial properties of road networks (as routing graphs)
  - Vertices have low degree (~2,5 in average)
  - Spatial information of vertices available
  - Hierarchical structure



Highway Hierarchies – Query times

- HH-Routing in Germany

With DT	Nodes	Edges
G0‘	2260320	5575736
G1	818981	1831940
G1‘	160058	826952
G2	116432	387502
G2‘	34587	322354
G3	28308	148948
G3‘	10127	138184
G4	8570	67238
G4‘	3261	69252

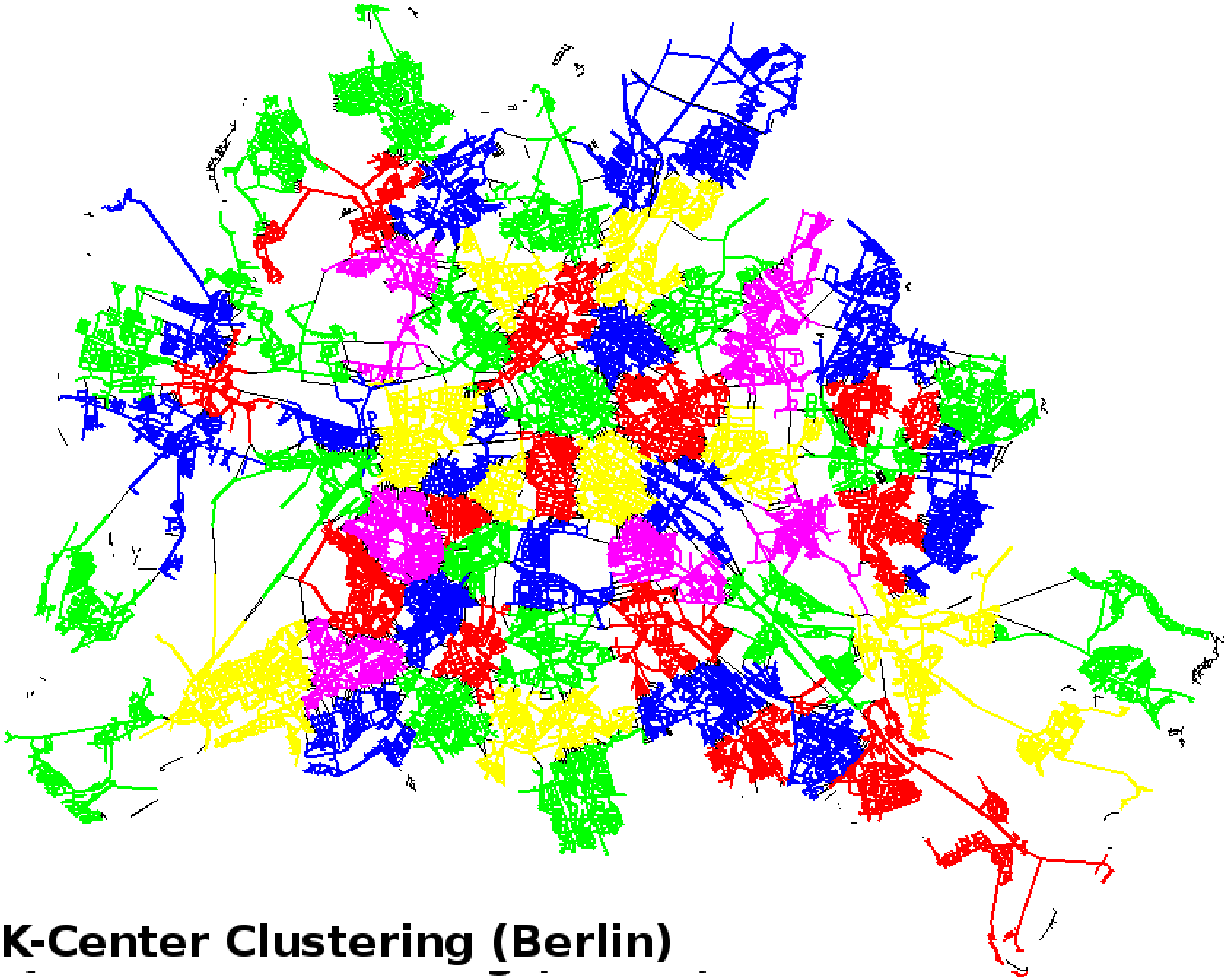
Avg. query time:	2.45 ms
Req. memory:	163 MB
Preprocessing:	131 min.

Without DT	Nodes	Edges
G0‘	2260320	5575736
G1	818981	1831940
G1‘	160058	826952
G2	116432	387502
G2‘	34587	322354
G3	28308	148948
G3‘	10127	138184
G4	8570	67238
G4‘	3261	69252
G5	2826	29052
G5‘	1046	28448
G6	878	4866
G6‘	293	4690
G7	243	1128
G7‘	44	772
G8	6	6
G8‘	0	0

Avg. query time:	3.8 ms
Req. memory:	123 MB
Preprocessing:	157 min.

Mobile Highway Hierarchies routing

- Proble ze I/O
- Soluti
  - Pa
  - Uti



K-Center Clustering (Berlin)

Thank you!

http://mapsforge.org