TECHNISCHE UNIVERSITÄT CHEMNITZ

GRAPP 2022
17th International Conference on Computer Graphics Theory and Applications
Online Streaming 6th – 8th February 2022

Development of a Platform-independent Renderer for the Rendering of OpenStreetMap Indoor Maps in Flutter

Julia Richter, Robin Thomas, David Lange, Thomas Graichen, and Ulrich Heinkel
Professorship Circuit and System Design, Chemnitz University of Technology, Germany, Contact: julia.richter@etit.tu-chemnitz.de

MOTIVATION

Available renderers have at least one of the following weaknesses:

- Not platform-independent
- Not freely available / not Open Source
- No seamless integration of indoor into outdoor maps: Indoor elements are rendered over outdoor elements
  - ➢ occlusions

Our solution:

- Extension of mapsforge_flutter with indoor rendering
- Platform-independent
- Open Source
- Seamless integration of indoor elements into outdoor maps
  - ➢ Map elements are subjects to same layout algorithm
  - ➢ No occlusions

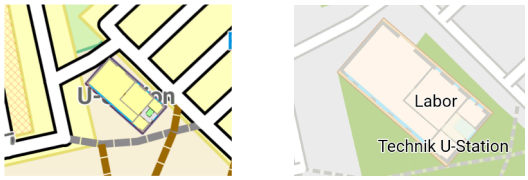

Fig. 1: Example of rendered map with (available) and without (our solution) occlusions

METHODS

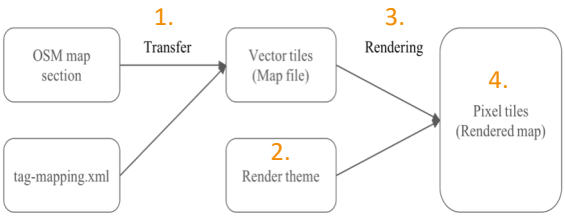

Fig. 2: Overview: From OSM indoor data to rendered maps (pixel tiles).

1. Transfer of required indoor data with *level* or *repeat_on* tag into Mapsforge vector tiles
2. Render theme extension with render rules for indoor elements
3. Extension of the rendering library with an additional match check to an indoor level variable
4. Extension of the tile identification by an indoor level value for retrieval from the cache

RESULTS AND DISCUSSION

Open Source Repository: Schwartz, M. (2020).
GitHub - mikes222/mapsforge flutter: A port of mapsforge for flutter
https://github.com/mikes222/mapsforge_flutter

Platform-independence and Performance

- Average tile render duration
- Tested on the four zoom levels: 14, 16, 18 and 20
- One example map (Chemnitz University of Technology)
- Four platforms:

Device	Desktop computer		Fairphone 3 (GSMArena, 2021b)	iPhone 6 (GSMArena, 2021a)
OS	Windows 10	Linux 5.4.101-1; Wayland - GNOME - Manjaro	Android 10 Kernel 4.9.218	iOS 12.5.1
CPU	Intel(R) Core(TM) i7-7700K CPU @ 4.20 GHz		Octa-core (4x1.8 GHz Kryo 250 Gold & 4x1.8 Ghz Kryo 250 Silver)	Dual-core 1.4 GHz Typhoon (ARM v8-based)
GPU	AMD Radeon RX 590		Adreno 506	PowerVR GX6450 (quad-core graphics)

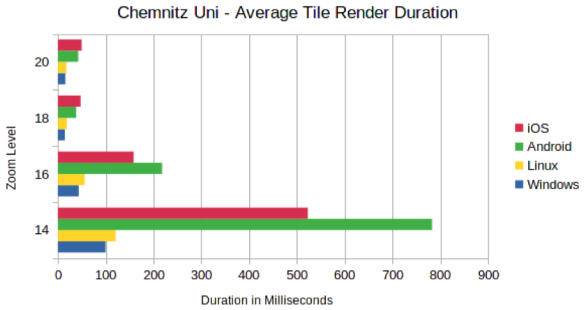

Fig. 3: Averagre tile render duration for an example building

Discussion:

- Rendering time increases significantly at a lower zoom level because there significantly more elements per tile have to be rendered than at higher zoom levels.
- Desktop applications perform better than mobile applications with a maximum average render time of 120 ms per tile. However, desktops usually have a larger viewport than mobile devices and have to render more tiles than mobile devices.

Rendering Output

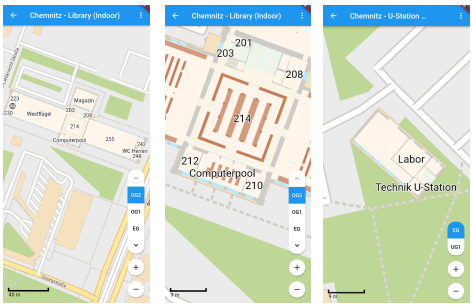

Fig. 4: Examples of maps rendered with the developed Flutter indoor rendering

Discussion:

- Rendering output for a variety of open and freely available OSM indoor maps
- Seamless integration of indoor into outdoor maps
- Level selection and automatic recognition of the currently available levels was developed in Flutter as a proof of concept

FUTURE WORK

- Labels or symbols at tile boundaries are cut off due to their direct rendering into tiles
  - ➢ *Tile boundaries must be taken into consideration when rendering labels and sysmbols*
- Flutter does not allow rendering in separate thread: parallel rendering of separate tiles is impossible and the rendering can block the user interface
  - ➢ *Enabling of multi threading*
  - ➢ *Hardware accelerated rendering*
- Appropriate representation of underground elements and building parts exceeding the outline of a building

This project is funded by the European Social Fund (ESF) with the project ID 100382183.

Europäische Union
Europa fördert Sachsen. ESF Europäischer Sozialfonds

Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des vom Sächsischen Landtag beschlossenen Haushaltes.