Iason Manoloudis

3D Traffic Sign Rendering in OSM2World

Contact Information

Github: https://github.com/JayGhb
Mentor: Tobias Knerr
OpenStreetMap account name: JasonManoloudis

Overview

OpenStreetMap is a data provider for various services that work with map data. ”Taking advantage” of the exported OSM data structure, OSM2World is a converter that renders 3D models of the world, based on such. As the project is still under construction and some OSM tags styles are not yet supported, it is the scope of this project to expand its codebase and include various aspects of traffic sign modules to its rendering capabilities.

Current state/ Deliverables

OSM2World has been a functional project for a long time and this project aims at expanding it. As for now, it is only possible to render traffic signs only if those have been explicitly mapped using a seperate node (not by marking their effects on a way). Moreover, a small number of signs and their textures are supported, consisting exclusively of German ones whose national IDs remain hardcoded in the TrafficSignModule.java class. Lastly, there is currently no rendering or texture support for other traffic modules such as traffic lights, mirrors whereas road marking can also be further enchanced.

This Google Summer of Code application focuses on 6 specific tasks to be completed throughout the summer. Further information on them is given in the *Timeline* section below.

Those are, by name:

- Font rendering for signs with custom text
- Add support for common human-readable traffic sign values
- Create an easier way for users to import traffic sign catalogues
- SVG texture support
- Rendering of destination signs
- ”Guessing” the location of a sing based on the road attributes
- (time permitting) Rendering of traffic mirrors & lights - road marking enchancement

Timeline

Note: For the following, a rating of 5 stars implies a top-priority project and a rating of 1 star, a lowest-priority one.

April 10 - May 26 *(Application Review - Community Bonding period)*

- Implement small existing TODOs - get to know the codebase better
- Read and experiment on Java Graphics2D, measuring text sizes in pixels, using SVGs in Java projects, JOGL and more
- Further mapping with JOSM, familiarizing with every aspect of traffic sign, way, destination tagging
- Reading ducomentations to speed up and ease the upcoming coding

As this timeframe is not part of the official GSoC coding process, it remains somewhat abstract and is not broken down any further. More investigating than the one mentioned, may take part, depending on the difficulty and importance of the various topics.

May 27 - June 10 *(Font rendering for signs with custom text)*

☆☆☆☆☆ A possible way of implementation would begin with adding Java Font support. The Java Font class represents fonts, which are used to render text in a visible way. Following this, the implementation steps would be:

1. Measuring text size in pixels
2. Create a BufferedImage in the appropriate size (Graphics2D createGraphics())
3. Draw the text (Graphics2D drawString())
4. Save the image
5. Support several OSM2World output formats (obj,pov,png,direct via JOGL)

For that, existing texture classes (e.g. TextureData, JOGLTextureManager) will have to be refactored to be able to support different formats like raster, vector graphic file format, procedural textures for natural elements, and more.

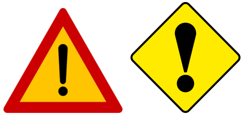
June 11 - June 24 *(Human-readable traffic sign values - 1st evaluation)*

☆☆☆☆☆ This task would include changing parts mostly of the TrafficSignModule.java class. Specifically, right now the code includes:

```
     List<TrafficSignType> types = new ArrayList<TrafficSignType>(signs.length);

     for (String sign : signs) {

     sign = sign.trim();
     sign = sign.replace('-', '_');

     try {
        types.add(TrafficSignType.valueOf(country + '_' + sign));
     } catch (IllegalArgumentException e) {
        // not a supported traffic sign type
     }
}
```

which matches each sign contained in "String[] signs" to one in the list (currently enum) of supported types and later creates a visual representation of them. This however assumes that all sign values begin with their country prefix (e.g. DE_101) while this is not the case for human-readable ones. Necessities for the implementation of this task would be:

- Refactoring the existing code to solve the above-mentioned problem
- Add more human-readable sign values mentioned in the OSM Wiki
- Add a configuration option for which country's style of the sign to assume, as same sings have different representations in different countries. An example of this can be found in the *Warning* signs of Greece (left) and Ireland (right)



June 25 - July 9 *(Traffic sign catalogues)*

☆☆☆☆☆ An easier - more mapper-friendly way of adding different countries' traffic signs catalogues will be implemented. This includes:

- Users would be able to add signs catalogues through modifying the properties file

- Source code must include support for reading and rendering them based on their textures
- More catalogues are to be added by the applicant to expand the existing sign support and demonstrate the use of the new feature

July 10 - July 22 *(SVG texture support - 2nd evaluation)*

★★★★☆ Why adding Scalable Vector Graphics texture support would be benefitial:

- SVG is just XML - smaller file sizes than those of existing png textures translate into smaller loading time
- SVG can be created/edited with any text editor which makes it easily accessible to more mappers worldwide
- There are available Java libraries, well documented and widely used, that make support implementation easier (Batik, SVGSalamander).

During the specified time period, SVG texture support will be implemented in OSM2World and SVG textures will be added to/replace some of the existing ones. After the task's completion, related Wikis and webpages will be updated with information on how users will easily be able to add more on their own.

July 23 - August 6 *(Render destination signs)*

★★★★☆ Rendering of destination signs is considered (by the applicant) to be a task of higher difficulty so part of the time will most likely be given to investigation.

- Get accustomed to existing destination tagging techniques, destination relation and more
- Implement support for destination tags from OSM data
- Use existing relevant work from community members as base idea for rendering implementation (e.g. OSMLaneVisualizer)

August 7 - August 19 *(Guessing positions of not explicitly defined sings - Final evaluation)*

★★★★☆ As mentioned in the relevant OSM Wiki: *"It should however be assumed that the physical location of the sign is at the beginning and / or the end of the affected section"*. Given that, OSM2World rendered would assume, based also on the nature (effect) of the sign, that its location would be either the beggining/end of the affected section or, in case of way (road), somewhere next to the road. To make the implementation plan less abstract, time will be given to investigating "navigation", code-wise, from node to node to on a way to the point that there are no more outbound segments and the precise begging/end of the road can be specified.

August 20 - August 26 *(End of the program)*

During the final days, community members will be invited to review and give feedback on the complete project, minor bugs will be fixed and remaining documentation/wiki updates will be written.

During the whole coding process, regular Wikis and/or Medium articles are required to document the work and update the community. Those are scheduled to 2 per specified time period, as each time period consists of 2 weeks and they are considered enough to cover each week's work without resulting in an overly extended post.

Time devoted/ About me

I am a final year undergraduate student at the School of Informatics, Aristotle University of Thessaloniki, Greece. Having experience on various aspects of Informatics from courses, personal work and my time working abroad, I see this as a great opportunity to enchance both OSM2World and my existing skillset on technologies I have always been interested, like OpenGL and Java (repo links on highlighted words). Working on developing with an open source organization is a first for me and judging from the interactions with the community so far, OpenStreetMap seems like a great place to begin this journey.

Before the beginning of summer, I will be set free from my University obligations as I am expected to graduate, meaning I am more than willing to devote at least 40 hours per week on something that I finally do, not because it is obligatory.
I hope I will be able to work with you all and hopefully meet the community on State of the Map 2019 as a mapper and -why not- as a fellow contributor.

Iason Manoloudis.