CS 298 Summary of Intermediate Results

On-the-fly Map Generator for OpenStreetMap Data Using WebGL
Advisor:Dr. Chris Pollett

Description:

This is a continuation of the CS 298 Project, started last Spring (Spring 2015). This project is an approach to create an On-the-fly Map Generator For Openstreetmap Data Using WebGL.

Last semester, I started with drawing a canvas on top of Google Maps and tried plotting the latitude and longitude coordinates queried within a bounded box, which was my first deliverable. I also tried drawing specific geometry shapes- polygons depicting buildings and specific areas. I, however failed, as I was not successful in drawing a bunch of polygons together in a single draw call, but one.

This semester, I will continue working from where I left, try to draw many polygons in single draw call (using drawElements() call) within the bounding box and expand it to bigger bounding boxes. I will also try to generate map tiles accordingly. In addition, I will also do my other deliverables-resizing the map according to zoom levels and adding ability to do filtering on the OSM data and generating tiles accordingly. Finally, I have aimed to do a comparison between the traditional tile servers and my project, which generates map tiles on-the-fly.

Schedule:

Week 1(Aug 25- Aug 30, 2015)	Talk in detail about the project to professor. Get suggestions on improvement.
Week 2(Aug 31 – Sep 7, 2015)	Research on improving the existing code. Continue coding.
Week 3(Sep 7- Sep 14, 2015)	Continue coding. Start researching on generating map tiles on-the-fly.
Week 4(Sep 14 – Sep 21, 2015)	**Finish Deliverable 1: Create Map tiles for places of interest.**
Week 5(Sep 21- Sep 28, 2015)	Continue coding, do improvements if any. Start researching on how to resize map tiles according to zoom levels
Week 6(Sep 28 – Oct 5, 2015)	Continue coding.
	Continue coding. Do improvements if any.
Week 7(Oct 5 – Oct 12, 2015)	**Finish Deliverable 2: Resize map tiles according to zoom levels.** Start

	researching on generating tiles according to the data being filtered.
Week 8(Oct 12 – Oct 19, 2015)	Continue coding.
Week 9(Oct 19 – Oct 26, 2015)	Continue coding.
Week 10(Oct 26 – Nov 2, 2015)	**Finish Deliverable 3: Add the ability to do filtering on the OSM data returned and generate tiles accordingly.** Start studying/comparing traditional tile servers with the current project.
Week 11(Nov 2 – Nov 9, 2015)	Start writing the CS 298 report.
Week 12(Nov 9 – Nov 16, 2015)	Work on the draft version of the CS 298 Report - Submit to Advisor and Committee.
Week 13(Nov 16 – Nov 23, 2015)	**Work on the final version of the CS 298 Report - Submit to Advisor and Committee.**
Week 14(Nov 23 – Nov 30, 2015)	**Choose a date and hall for defense. Defend the project.**

Key Deliverables:

My deliverables for this project in order are:

- Code:
    - Draw Maps on the browser (for a given query, the Postgres database returns the corresponding OSM data and the client generates and renders tiles).
    - Add a feature/functionality for the map thus generated to be resizable according to zoom levels.
    - Add an ability to do filtering on the OSM data queried (different ways of drawing the same data) - feature/functionality.
- Report:
    - A detailed report on the proposed changes and my findings.

Innovations and Challenges:

- Rendering tiles on the client for the OSM data retrieved from the Postgres database.
- Filtering of the queried data and rendering tiles accordingly.

References:

[1] Segmentation of OpenStreetMap Data: Generating, Merging, and Distributing Tiles. Jan Behrens. University of Bremen, September 2011.

[2] Web-based 4D visualization of marine geo-data using WebGL. Bernd Resch, Ralf Wohlfahrt and Christoph Wosniok. Cartography and Geographic Information Science, 2014.

[3] Programming 3D applications with HTML5 and WebGL. Tony Parisi. O'Reilly Media, 2014.

[4] http://www.chromeexperiments.com/detail/mapsgl/

[5] https://www.mapbox.com/blog/vector-tiles/

[6] https://github.com/twpayne/tilecloud

[7] http://sourceforge.net/projects/gmapstilegen/

[8] https://github.com/samsargent/Google-ImageMap-Tile-Generator