Geography 575

Lab #1: Space-time Prop Symbol Maps w/ Leaflet & JQuery

Lab Objectives:

- Introduce you to JavaScript, JQuery, and the Leaflet.js mapping library
- Assemble and dynamically load a spatiotemporal information set in the GeoJSON format
- Implement operator primitives, including *pan*, *zoom*, *retrieve*, and *sequence*

Evaluation:

This lab is worth **50 points** toward the Lab Assignments evaluation item and is worth 10% of your course grade. A grading rubric is provided on the last page to inform your work.

Schedule of Deliverables:

• **February 10:** Lab #1 Assigned	//client contract begins
• **February 17:** Leaflet Tutorial & Dataset	//Activity #4
• **February 24:** Leaflet Tiles & Prop Symbol	//Activity #5
• **March 2:** Leaflet Operators	//Activity #6
• **March 9: Lab #1 Due in Full**	//contract deadline

Challenge Description

You have been invited by Professor Kris Olds to contribute dynamic content to the *World Regions in Global Context* course (Geography 340), an online course on regional development. The G340 course leverages web-based technologies to bring students and experts together from around the globe to discuss the dynamic geographic phenomena and processes (i.e., space over time) constituting the cultural and natural worlds. Professor Olds has requested that you contribute to the course material by designing and developing a web-based spatiotemporal visualization that "makes visible" some regional geographic phenomenon and process for structuring in-class discussion. Professor Olds has recommended that you map spatiotemporal information collected about cities—given his expertise in Urban Geography—making proportional symbols the appropriate thematic map approach; the specific geographic phenomenon/process portrayed by the spatiotemporal visualization remains your choice. Further, Professor Olds has requested you improve the user experience by including interactivity to (1) allow users to customize their experience with the map, (2) increase the amount of detail provided in the display upon request (i.e., overcoming the cartographic problematic), and (3) empower users with control over the map. The final spatiotemporal visualization should prompt hypotheses about the underlying drivers of the represented spatiotemporal pattern and/or process in order to promote online discussion in the course.

Editor's Notes from the World Regions *Class*

Your spatiotemporal visualization must include *at least 15* point locations, with each point location exhibiting variation across *at least 7* timestamps (e.g., days, months, years, decades). The scale and extent of your spatiotemporal dataset is not restricted. While you are required to design a proportional symbol map (i.e., not a choropleth map, dot density map, etc.), you may choose to map a spatiotemporal information set that is aggregated to units other than cities with permission from the course instructor. You are required to implement _at least 5_ interaction operators: (1) *pan*, (2) *zoom*, (3) *retrieve*, (4) *sequence*, and (5) a fifth operator of your choosing; instructions for implementing the first four are provided in this tutorial. Your spatiotemporal visualization must include a temporal and map legend to assist interpretation of the display.

You also are required to contextualize your map by placing it within a webpage and providing pertinent supplemental content. Give the map a good title, cite your data sources, and consider how you can use additional text to help the map tell a meaningful story. Your spatiotemporal visualization and web page must be unique; their appearance should not merely replicate that of the examples you are using.

Getting Started with Leaflet

Leaflet is one of many code libraries now available for publishing slippy maps to the web. Leaflet is a JavaScript library pioneered and maintained by Vladimir Agafonkin (http://agafonkin.com/en/), and quickly is growing in popularity within the web development community because it is both lightweight and open source (meaning you both can view how it functions and extend it to fit your needs). Maps produced using Leaflet can load a variety of basemap tile services and can draw vector features atop these tiles using the SVG (scalable vector graphics) standard. Because of the small file size and support of touch-based interactions, Leaflet is considered among the best web mapping libraries when designing for mobile devices.

The Leaflet library is an open-source project on GitHub (http://github.com/Leaflet/) and can be extended through numerous open plugins (http://leafletjs.com/plugins.html). Mapbox.js (http://www.mapbox.com/mapbox.js/) also builds on Leaflet and allows for simple loading and manipulation of custom tilesets. The goal of this tutorial is to provide you with a broad introduction to using Leaflet for Web Cartography. There are supportive communities for Leaflet on Twitter (https://twitter.com/search?q=%23leaflet) and Stack Overflow (http://stackoverflow.com/search?q=leaflet). Refer to these materials for additional background and guidance as you complete the tutorial.

Modules 4-6 walk you through your first Leaflet map. The code included in the modules is available for download through Learn@UW. If you would like a consolidated version, you can access our PDF write-up of using Leaflet for time series maps from Cartographic Perspectives.

Evaluation Rubric: Leaflet Challenge (50pts)

Representation (15)

(13-15 points) The basemap is appropriate for the map scenario; proportional symbols are styled consistently with the map and clearly represent the spread of the data without appearing cluttered; there are well-placed and highly usable proportional symbol and temporal legends; the temporal legend correctly updates the data representation.

(10-12 points) One or two of the above elements could have been better designed so as to promote more consistent styling or better usability.

(7-9 points) There are multiple bugs or flaws in the design of the above elements and/or some are designed inappropriately for the representation of the data.

(6 points or below) There are serious problems with the appropriateness, design consistency, and/or usability of the map representation elements.

Interaction (15)

(13-15 points) The Leaflet map successfully implements panning and zooming; retrieving through popups with space, time, and attribute data; sequencing through a slider or other appropriate UI element; and a fifth operator that is clearly visible to the map user.

(10-12 points)There are one or two minor bugs or one of the operators is not immediately obvious to the user. The fifth operator is not present.

(7-9 points) One of the provided operators remains unimplemented, or a couple operators are incomplete/buggy.

(6 points or below) Multiple operators are absent or have significant bugs/problems.

Design for Scenario (15)

(12-15 points) The overall web page design has a clear entry point for the user and visually tells a story; the dataset and overall design are appropriate to the assigned scenario; the design shows creativity and inventiveness. The additional operator makes sense for the map.

(8-11 points) The design and dataset could be somewhat more creative or usable or better aligned to the assigned scenario.

(7 points or below) The design and dataset are blasé, lack usability, or depart significantly from the assigned scenario.

Good Coding Practice (5)

(4-5 points) Your code follows practices introduced in class (variable names, elimination of redundancies, proper indentation). Your code has plenty of descriptive comments and is human readable. Your repository is properly pushed to GitHub with an informative description.

(2-3 points): Your coding practice needs improvement moving forward. Your code requires denser or more descriptive comments. Your repository is properly pushed to GitHub.

(1 point or less): Your code is simply pasted from the lab examples and is uncommented. You did not post you map to GitHub.

You must publish your interactive map to GitHub & submit a zipped directory to Learn@UW 1 hour before class on March 9th.