Interactive Web Map of Alaska Native Historical Timelines

Robert Hollowood, Mary Kollander
CSCE A470 Final Report

May 2, 2024

Table of Contents

Interactive Web Map of Alaska Native Historical Timelines
Robert Hollowood, Mary Kollander

Abstract

This report outlines the development of an interactive web map displaying Alaska Native historical timelines with a focus on epidemic diseases from 1750 to 2020. The project was requested by Dr. Maria Williams, who also provided all associated research data. The web map is constructed from HTML/CSS, JavaScript, and GeoJSON with the integration of Leaflet, PruneCluster, and noUiSlider libraries. OpenTopoMap is used as the visual map layer, and the entire project is hosted by GitHub Pages.

1. Introduction

The client for this project is Maria Williams, chair of the UAA Alaska Native Studies program. Dr. Williams has a Ph.D. in Ethnomusicology and researches Alaska Native cultural practices (McCoy, 2013). She spearheaded the creation of The Alaska Native Reader, a collection of essays, poems, stories, and art from Alaska Native people of many walks of life, "from a Tlingit photographer to Athabascan and Yup'ik linguists, and from an Alutiiq mask carver to a prominent Native politician and member of Alaska's House of Representatives." (Williams, 2009).

Dr. Williams performs her own independent research into Alaska Native history. Over this research, she has compiled timelines of events regarding various aspects of said history, and wanted to share that research online in an explorable, interactive form. As Dr. Williams does not have a background in tech, she offered the project to the UAA CS&E department as a potential capstone project. Since she is active UAA faculty, we were able to regularly meet on campus to discuss the project.

2. Project Overview

The goal of this project was to create an online web map that should share Dr. Williams' research in an easy-to-explore, interactive format. This application will help spread knowledge about Alaska Native history, as well as give insight about how diseases spread and affect different areas and populations. It should be noted that the aim of this report is purely to discuss the technical details of the project; discussing the ramifications of the data itself is outside of our scope.

2.1 Provided Data

The original data provided to us for this project was a document detailing reported epidemic diseases in Alaska history. Multiple options for potential timelines were offered by Dr. Williams, but this one was chosen as it was the most verbose. The document is a loose collection of notes

in a bullet-point format; each entry starts with the time frame of the event and then describes a few details of the event, including its location. Some entries include a citation that leads to further information on the event. As some of the events described are over three centuries old, much of the provided data is naturally imprecise and/or incomplete; representing this data faithfully provided an additional challenge for the project.

3. Project Requirements

Dr. Williams did not have a clear product in mind when proposing this project; as such, it fell to us to determine many of the specifications of our project. Specifications also fluctuated over the course of the project, as Dr. Williams became more familiar with the shape of the project and determined what her own requirements were. Many of these specifications were added late as we determined work would continue over the summer; as such, the following specifications reflect our initial requirements for the project.

3.1 Functional Specifications

1. The application must be shown through a website accessible on the internet. This website should support both desktop and mobile browsers, and be directly accessible via a URL.

2. The application should primarily be composed of a "slippy map"- an interactive map that allows the user to zoom into the map and click-and-drag to move around. This map should be implemented using data from an open-source tilemap.

3. The application should mark points on the map signifying the location of events in Alaska Native history. These points should be generated from data baked into the website, with the user having no ability to modify them.

4. Interacting with a point on the map should display further information on that event. The provided information will always include a date, a location, and a brief description. The event may also have an associated image, or an academic source.

5. The points marked on the map should be easily filterable by time, allowing the user to only view points from a certain time period.

3.2 System Specifications

Our main system requirement (aside from "powerful enough to handle the application") was that the system should require no maintenance. After project completion we will not be able to perform maintenance ourselves, and it is unrealistic to expect Dr. Williams to hire help purely to maintain this application. As such, our solution needed to be "set and forget"; once deployed, it should be able to maintain itself without outside input. Aside from that, our only requirements

are that the system be responsive and that the application be able to run on most major devices and browsers without error.

4. System Design

For our web map, HTML is used to define the structural hierarchy of the user interface. Additionally, CSS is used for styling and device compatibility. JavaScript is used as the dynamic scripting language to handle interactivity. The entire project is hosted online using GitHub Pages.

4.1 Data Storage & Formatting

GeoJSON formatting is used to manage our marker data. This format is used for encoding geographic data structures using JSON. Geometry is represented as a JSON object with a "type" member specifying the type of geometry and coordinates represented as arrays. Each node's geographical location, event description, time range, and reference link is stored in a singular GeoJSON file. Figure 1 shows how this information is formatted.

```
{
  "type": "Feature",
  "properties": {
    "name": "m1",
    "description": "Event description",
    "startDate": 1750,
    "endDate": 2020,
    "dataRef": "",
    "imageUrl": ""
  },
  "geometry": {
    "type": "Point",
    "coordinates": [-201.53, 56.13]
  }
}
```

Figure 1: GeoJSON formatting for an individual marker.

As our data was not provided in GeoJSON format, converting the data was also part of our work on the project. As doing this by hand would be tedious, we created a simple feature editor (shown in figure 2) for ourselves capable of reading/writing GeoJSON files, creating new features, and editing/deleting existing features. This is a simple script written using PySimpleGui to create the visual interface. This was not created as an all-purpose GeoJSON file creator, but instead a tool specifically built to generate the features for this project; as such, there are no immediate plans to share this tool.

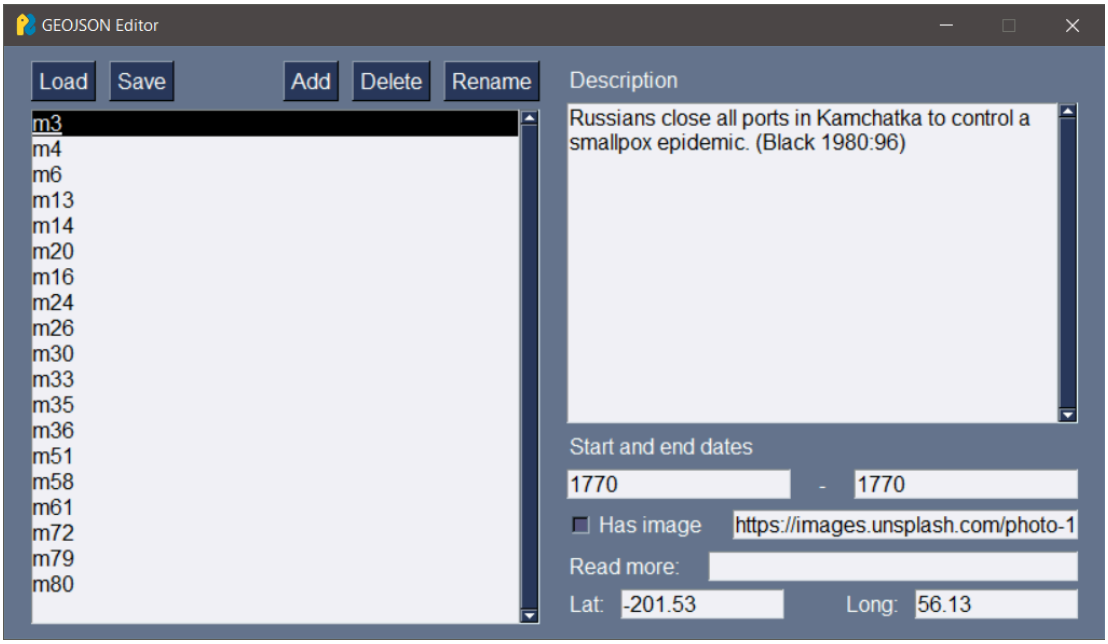


Figure 2: A screenshot of the feature editor.

4.2 Leaflet & OpenTopoMap

Leaflet is a lightweight, open-source JavaScript library used to create interactive web maps. The library is known for its customizability, ease of use, and platform-friendly nature. The project uses Leaflet's built-in set of tools for handling map-related functionalities including the following:

- Map marker popups
- Map boundaries
- Zooming, panning, dragging, & scaling
- Tile layer implementation and switching

The web map uses various performance features offered by Leaflet including hardware acceleration, CSS integration, smart polyline/polygon rendering with dynamic clipping, tap delay elimination, and full browser support. The majority of these features are automatically implemented. To create the map, the *L.map* method is used with inputs for map configuration, initial view coordinates, and initial zoom level. The map configuration and initialization is demonstrated in Figure 3.

```
// Map configuration
const mapConfig = {
        layers: [createTileLayer(OTM_TILE_LAYER)],
        maxBounds: MAX_BOUNDS,
        maxBoundsViscosity: MAX_BOUNDS_VISCOSITY,
        minZoom: calculateMinZoom(),
        maxZoom: MAX_ZOOM
};

// Initialize map
export const map = L.map('map', mapConfig).setView(INITIAL_COORDINATES, calculateMinZoom());
```

Figure 3: Leaflet map configuration and initialization method.

The visual map layer is integrated using a variant of OpenStreetMap (OSM) called OpenTopoMap (OTM), which uses a topological design instead of the default street layer provided by OSM. The geographical data itself is provided freely by the OpenStreetMap Foundation (OSMF) under the Open Data Commons Open Database License (ODbL).

4.3 PruneCluster & NoUiSlider

PruneCluster is a JavaScript library used for fast and realtime marker clustering, designed as an alternative to the default Leaflet marker clustering library. The decision to use this library stems from its ability to handle large datasets while maintaining low memory consumption, especially on mobile devices. The clustering algorithm included in the library takes inspiration from collision detection in physical engines. Additionally, the PruneCluster library supports integration with Leaflet, allowing all the customization of Leaflet markers to transfer over to PruneCluster markers. Figure 4 demonstrates how PruneCluster handles marker clustering in the web map, combining markers that have close proximity. Once selected, the camera zooms in to show the included markers.

Figure 4: PruneCluster marker handling in the web map.

The timeline range slider is implemented using the noUiSlider JavaScript library, a customizable and easy-to-use library used for creating range sliders in web applications. NoUiSlider allows for tooltip popups on each range handle which is used to show the currently selected years. Additionally, the included *mergeTooltips* function is used to prevent the pop ups from overlapping. Figure 5 shows the final design of the timeline range slider in the web map.

Figure 5: The implemented timeline range slider.

4.4 GitHub Pages

The web map uses GitHub Pages for online hosting. GitHub Pages is a static site hosting service that publishes a public website directly from a repository on GitHub from HTML, CSS, and JavaScript files. This service handles the entire build process of the project automatically; all changes published to the active branch are automatically pushed to the web page.

Sites hosted via GitHub Pages are subject to usage limits such as a 1 GB size limit, 10 minute deployment timeout, soft 100 GB per month bandwidth limit, and a soft limit of 10 builds per hour. The web map has not yet exceeded any of these limitations, and it is not expected to in the future. However, if this does occur, GitHub Support will not immediately cut off the website; an email notice is sent first suggesting strategies for reducing the site's impact on the servers, providing an appropriate amount of time to make adjustments to reduce server load.

4.5 System Architecture

As previously stated, the system primarily relies on HTML/CSS, JavaScript, and GeoJSON to provide the base of the interactive map, which is then hosted using GitHub Pages. Figure 6 shows a diagram of the system architecture, highlighting the dependencies of each component.

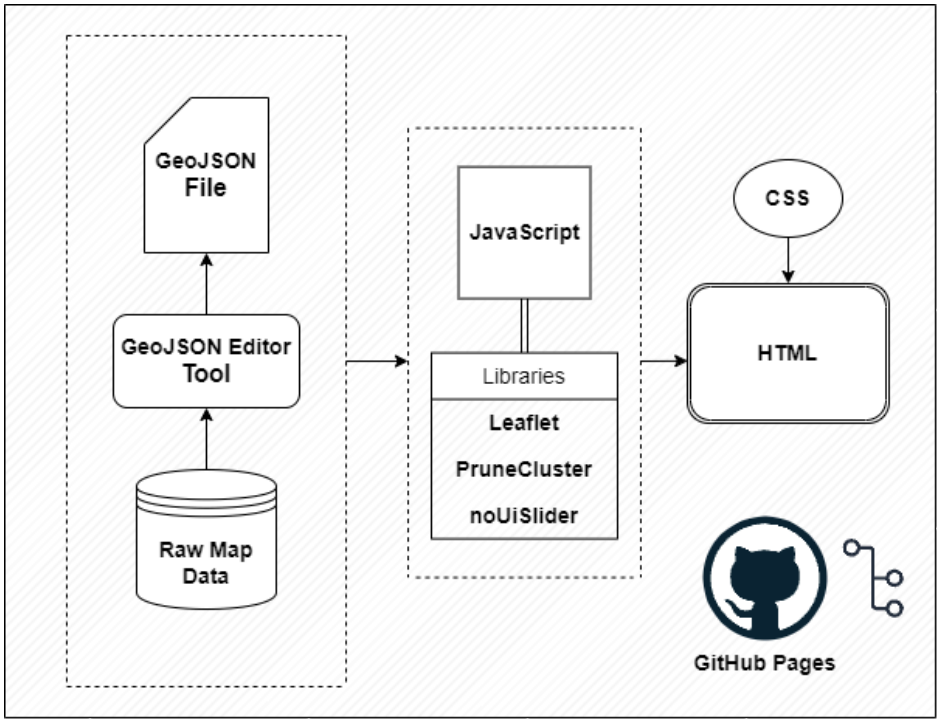


Figure 6: Diagram of the system's architecture.

The raw map data initially stored in a Google Docs file is converted to GeoJSON format via the GeoJSON Editor Tool. JavaScript retrieves this data and provides functionality for the HTML framework. CSS is used in conjunction with HTML to ensure optimal compatibility across various web browsers and devices, including mobile platforms. The entire framework is hosted from GitHub Pages.

5. Development Process

The development lifecycle of our project relied heavily on the bi-weekly meetings with Dr. Williams, which we treated as deadlines. The purpose of the meetings was to show our progress on the project and receive feedback / suggestions for the current design. The two-week period in between these meetings acted similarly to an agile development sprint in which certain goals were set and expected to be completed by the deadline.

5.1 Testing and Debugging

The testing phase of our project did not proceed as initially planned; we tested the web map heavily during each phase of development to make sure that all components were working as expected on various devices and browsers. Elements that were found to be incompatible became top priority until they were fixed. Overall, the project did not have a primary testing phase as this was spread throughout all phases of development.

5.2 Challenges

In the first stages of the project, much of the code had to be rewritten until a stable version was produced. The web map went through three full rewrites until all components worked together with no performance loss. This primarily revolved around the timeline slider, as quickly updating the markers initially caused an unacceptable amount of lag. The most notable rewrite was implementing the PruneCluster library; the code had already been designed to fully support Leaflet markers and clustering, but ended up causing too many issues to keep. In all, the map, timeline range slider, and markers all relied on this library and had to be rewritten.

6. Results

As of the submission of this report, the application meets all functional specifications we initially laid out, but work still needs to be done to import all the data into the map. Work on features requested by the client is also still in progress. That being said, Dr. Williams has been pleased with the progress so far and continues to support our efforts going forward. While the data currently in the system is incomplete it is still significant, and already starts to serve the purpose of sharing information about Alaska Native history.

6.1 Current Version

The application is currently live at https://mgkollander.github.io/ak-history-webmap/, and is expected to remain hosted by GitHub Pages for the foreseeable future. If requested by Dr. Williams, the domain name will be changed to something more appropriate for the project.

6.2 Future Work

As mentioned previously, the base design we initially planned has been completed; the next step we will be prioritizing is importing the rest of the markers to the map. After this is complete, we will begin work on our list of optional features listed below in order of priority.

1. Populate a new GeoJSON file with cultural map boundaries sourced from the UAF Digital Archives. These boundaries will be attached to a new tile layer.

2. Separate markers into two types; a pin-like marker for an exact location, and a circular marker that contains a geographical range for wider area events. The second type will display the region affected when it is selected.

3. Overhaul marker popups to create a window on half the screen, while panning to display the entire range of the selected marker. When displayed, the timeline slider will be locked until the popup is exited. This will make the device much easier to use on mobile devices.

4. Create a menu / about button that displays general information about the project, including a link to the GitHub repository, authors, licensing, and attributions. There will also be a section for different accessibility options, such as increasing text size or enabling color blind modes.

5. Add a lock to the timeline slider that holds the current range when you move the handles. This will let users view events in a certain span of time without needing to constantly go back and forth between the toggles. For example, a lock from 1950 to 2000 will allow the user to only see events between 50 years. If one handle is moved, the other accommodates to equal 50 years.

7. Summary and Conclusions

Our web map was developed using Leaflet for Dr. Maria Williams to help share her research into Alaska Native history. The project is feature-complete as defined by our initial specifications, but work is still to meet requirements determined later. Regardless, Dr. Williams is excited about the progress thus far and will continue to support development going forward. We have been invited to present our poster on May 22nd as part of the Alaska Native Showcase for the Board of Regents.

This project has been a joy to work on. Amidst all the heavy, highly technical content that we learned during this semester, having a project that was conceptually simple but tricky to implement effectively was a wonderful change of pace. Having a chance to take our time to make something that works well rather than just “works” was a valuable experience, and may just be the most important lesson learned this semester. Working with the data itself was a treat, too; being able to see the spread of diseases over the state as we worked to add the points for it record by record was eye-opening.

In the future, we believe our testing process could be improved. For a decent duration of the project we did not test the map with a full marker dataset, instead testing features in a vacuum. By not testing our application under a full load, we did not find the issues with Leaflet’s in-built feature clustering until partway through development. Earlier testing under conditions closer to the final version could have helped us avoid refactoring our code as often as we did.

8. References

Williams, M. S. T. (2010b). The Alaska Native reader: history, culture, politics. Choice Reviews Online, 47(08), 47–4593. https://doi.org/10.5860/choice.47-4593

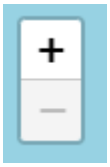
Appendix A: User Manual

Accessing the Web Map

The web map can be accessed through https://mgkollander.github.io/ak-history-webmap/, or through the mgkollander/ak-history-webmap repository on GitHub. It is compatible with most available browsers and devices.

Controls

Zoom controls are located in the top left corner of the screen. Additionally, the map can be zoomed using touch controls (pinching), or by scrolling the mouse wheel.



The current map view can be moved by either dragging with your finger, clicking and dragging with the mouse, or by using the arrow keys.

Viewing a Marker

To see an event for a marker, it can either be tapped on or clicked on. The popup can then be closed by either pressing/clicking the “x” in the top right corner, or touching/clicking on the map outside of the popup.

Using the Timeline Slider

The handles on the timeline slider can be dragged using either touch controls or the mouse.

Changing the Map Type

The topographic map on which the points are displayed can be changed to a simpler street map via the icon in the top right. When you press on the icon, radio buttons are revealed. Click the button of your preferred map to make the change.