Assignment 1: Story Map with GeoJSON and Mashups

Due: Oct 1st at 11:59pm EST

Worth: 20 points

In groups of three or four, create a story map using a geoweb data format.

Your storymap will utilize a geoweb mapping platform called Mapbox, which you will access via its application programming interface (Mapbox api). Mapbox is a development platform; it allows you to custom visualize your data. For your data, you will use the data format called GeoJSON. GeoJSON allows you to create complex sets of geospatial data with placemarks, polylines, polygons, attributes (e.g., descriptions, styling, and images), and other multimedia.

The choice of story is yours. The story should be interesting and dynamic and leverage different sets of data to enrich your narrative. It is easy to be overwhelmed by the possibilities a storymap opens to you. You should start simple and build from that. You also can draw on examples from numerous galleries to get a sense of what can be done and potentially where you can start.

Part of this assignment is coming to the realization that when you develop for the web, there is no one correct way to reach your goal. Often, you will be given a task and your clients will be less interested in the process than the final product. You will not be told what software and languages to use along the way; paying close attention to your client's expectations (i.e., a city government) is what is important.

Goals

- Compare and contrast features of traditional GIS and the geospatial web 2.0
- Appreciate how the story (and the capabilities of a particular platform) drives the application of the geoweb instead of the converse
- Create data in a tagging/notation language, GeoJSON, which is universally supported by the geoweb
- Understand what a code editor environment enables; how to properly use one; and how it differs from a traditional word processor
- Learn to document your application, both in the code and in a written report
- Mashup/embed one platform in another (e.g., Youtube in Mapbox)
- Experience client-server architecture and how parts of your application may be stored locally and parts in the cloud (or in multiple locations on the cloud)
- Make your story map aesthetically pleasing
- Individually reflect on the assignment

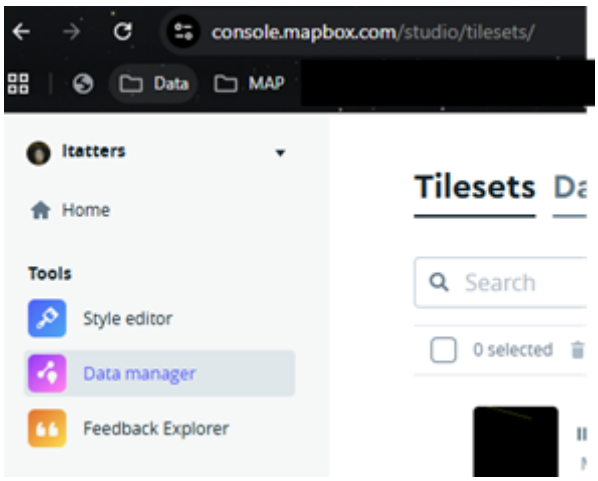
Note for your assignments: The temptation in all assignments is to let the experts or those with intermediate level of knowledge in the group handle the hard technical details. RESIST THIS TEMPTATION because it will hurt you in Part 2 of the final exam (the practicum), which is individual-based. As you know from working with GIS or programming, knowing the tech ONLY comes from doing (and occasionally, creatively failing) the tech. This also applies to any use of GPTs for coding. Always make sure you understand what the GPT is producing. And always attribute your use of GPTs.

Required, Recommended Software

- Rich Text Editor (e.g., Sublime Text), available on all GIC computers, free to download and install on your own.
- Cyberduck, available on all GIC computers, free to download and install on your own.
- Mapbox API
- Geojson.io

Tasks

1. Think of an interesting story that can be illustrated using geoweb applications, in this case Mapbox and GEOJson. This story should contain qualitative and quantitative content.
2. Find datasets and multimedia (e.g., images) to tell your story. The goal is for your story to include various kinds of data.
    a. One of those datasets should already be (or become) a shapefile. You will need to convert shapefile into geoJSON file
    b. You should have example of the four feature types: points, lines, 3D polygons, and areas. (Each will be ultimately saved as its own geoJSON file.)
    c. You should include an image for each 'chapter' (see below for discussion of chapter).
    d. You should include at least one video.
3. Based on your exploration and the required content, create a "storyboard". There are many examples from film. The goal here is NOT to be an artist but to sketch out the progression of the storymap. It also shows us your original vision(s).
4. Create four geoJSON files for your storymap: points.geojson; lines.geojson; 3D.geojson and areas.geojson. You could contain all the data in one file. But MapBox needs to create layers of different shape types in the same way ArcGIS does. Separate and smaller json files are much easier for you to debug and understand, (and easier for us to understand). Use geojson.io to create your files.
5. Login to your Mapbox account and create an Access Token and proceed to the Data manager tool on Mapbox. (Only one member of your group needs to use their token.)



6. Upload your GEOJson file to Mapbox Data Manager as a new dataset. Use one new dataset per shape type (i.e., points, lines, polygons, and 3D Polygons).

7. When you have created and then uploaded your datasets, they must be exported within Mapbox as new tilesets, which transforms your raw JSON data into a Mapbox layer. Use the 'Export' button in the top-right corner of the Dataset view, and choose 'Export to a new tileset'.
8. When your tilesets have been created, you will now need to create a new style in Mapbox. Mapbox Data Manager uses the term 'style' in lieu of ArcGIS Online's 'map'. Here you order your multiple layers of geospatial data and manage your display preferences.
    a. Insert each of your tileset 'layers' into your new style
        HINT: You must choose to import point layer as type 'Symbol'. Otherwise your points will be limited to small circles. You can then use the icon panel to assign types.
    b. Customize each tileset layer as required (e.g., icons, symbology) to tell your story
    c. Mapbox chooses to store your content (files, style) on its site. You will be building your 'storyboard' in a separate .js file. Mapbox will generate a style URL so you can link to the content it stores to the storymap you create.
9. Because you're not yet sufficiently knowledgeable about Javascript, we have done some of the work for you.
    a. Extract all files from the 384-A1-2025.zip
    b. Save them all to a new folder
    c. Remember those images you chose for your chapters and background? Save them into the 'images' folder.
10. The storyboard—the non-data portion of your project—exists in a separate Javascript file, called config.js. Edit *config.js* to complete your storymap. Here are some instructions in a github about Mapbox story maps.
    a. Change the style url to that of the new style you created in Step 9.
    b. Add your Mapbox access token.
    c. Choose whether or not to display a marker at the center of each chapter location.
    d. (Believe it or not, Mapbox's style doesn't control all aspects of your display preferences.) Choose a theme for the story text. There are light and dark options.
    e. Choose where your story's text sidebar should be aligned over the map. Options are center, left, right.

```
{
   style: 'mapbox://styles/mapbox/streets-v11',
   accessToken: 'YOUR_ACCESS_TOKEN',
   showMarkers: true,
   alignment: 'left',
   title: 'Story Title Goes Here',
   subtitle: 'A subtitle going into more detail goes here',
   byline: 'By a Digital Storyteller',
   footer: 'Sources and citations, etc. live down at the bottom of the story',
   chapters: [...]
}
```

    f. A chapter is a stop on the storymap in which there is information (e.g., data, images, and text) for the user. Add as many chapters in config.js as needed. Keep in mind you will need to tell the code where to position your 'camera' for each chapter (where it stops). To do this you need to focus one chapter at a time on each area of the style you have

created by inputting the chapter coordinates. You'll also need a comma between each section, but no comma at the end. Here is what a chapter looks like:

```
{
   id: 'identifier',
   alignment: 'right',
   hidden: false,
   title: 'Title',
   image: 'images/YOUR_FILE_HERE.png',
   description: 'Copy these sections to add to your story.',
   location: {
      center: [-77.020636, 38.886900],
      zoom: 13.5,
      pitch: 60,
      bearing: -43.2
   },
   mapAnimation: 'flyTo',
   rotateAnimation: true,
   callback: '',
   onChapterEnter: [],
   onChapterExit: []
}
```

g. Give each section a unique name in the section id property. Do not use spaces in the name (e.g., use _ or - instead). (The inability to use spaces in coding, except for strings, is so common that developers often learn never to use them.)
h. Fill out a title, and description properties for the narrative content of your story. The description section supports HTML tags.
i. The location values tell the 'camera' where to position itself. The following link will help you find and set these values.
j. Link the images you found to associate with your datasets to each chapter by replacing YOUR_FILE_HERE with the filename of the image

11. The description variable is where you need to "mashup" content and be creative. The code will translate standard HTML code you write within it. Be careful to use only " " characters in your html and not single quotes. [You are not required to use css to add style[1]]

    HINT: You may write out your code on multiple lines. However, the description variable requires it to be on one, so just put everything on the same line (e.g., "<h1>title</h1>
 Hello this is my description ").

    a. Include at least one of the following HTML tags at some point in your descriptions:
        i. Headings https://www.w3schools.com/html/html_headings.asp
        ii. Links https://www.w3schools.com/html/html_links.asp
        iii. Tables https://www.w3schools.com/html/html_tables.asp
        iv. Lists https://www.w3schools.com/html/html_lists.asp
    b. In some chapters, you need to embed video from YouTube or Vimeo using their "embed" code

[1] (but if you really want to try you need to use INLINE CSS https://www.w3schools.com/CSS/css_howto.asp)

    c. Upload any multimedia that is not accessed via a URL to neogeoweb.ca, via the ftp client, cyberduck.
12. Story telling is an art. Try to find the right balance between creativity and elegance to convey information. The flexibility of geoJSON through markers, extrusions, and styles provides considerable opportunities to present information effectively.
13. Go the 'Datasets' tab on Mapbox Data Manager https://console.mapbox.com/studio/datasets and use the ⋮ to download the modified geoJSON. Zip these files and your *config.js* and include them in your submission.
14. Zoom to and display the best chapter design in the storymap and take a screenshot.
15. Take a screenshot of an example textbox containing your custom html tags like a table/list/video.
16. Document your *config.js* with the use of comment tags.
17. Use an FTP client such as CyberDuck to upload your files (e.g., https://neogeoweb.ca/group#/A1.html).

Documentation

1. Collectively write documentation https://neogeoweb.ca/group#/A1.html (report) for your storymap. Imagine a client (e.g., a city government) and address it to them. Imagine their 'ask' and answer why you chose the story, the elements of the story, and the style?
    a. Document the process of creation. Add citations, not for the content, but for the choices you made. The citations can be academic, the grey literature, or reference to examples ("I saw an ESRI storymap that had an inset map and I decided to apply it")
    b. All submissions should be single-spaced.
    c. All screenshots, tables, figures must be included in the body of the text and not in appendices. Tables and figures typically are illustrations of work and not the entirety. You need not include a print out of all the code as an appendix.
    d. Include your storyboard.
    e. If you used a GPT for any part of the storymap, how you used it, what prompt(s) you supplied, and what purposes did the automation serve that a human couldn't. If you did not use a GPT you don't need to mention anything, although you can elect to document why you didn't want to use a GPT (mention the reasons why).
    f. Manage your client's expectations: what were you not able to accomplish for them.
2. Individually (i.e., each person in the group), reflect on the assignment.
    a. What (e.g., chapters, data handling, coding, soft skills) were you responsible for?
    b. What were the different geoJSON (e.g., interesting objects), tilesets and Mapbox Style customization techniques (e.g., colour) that you used to create the story?
    c. What had you hoped to do originally versus what were you able to accomplish in the time allotted or what the platform seemed to allow? Add citations (see above).
    d. Sometimes storytelling doesn't actually need maps or interactivity but they're there anyway. How did the dynamic and static spatial data augment the narrative process for your particular story?
    e. The individual reflection should be written in paragraphs (e.g., as opposed to bullet points) and be no more than two pages.
3. All individual reflections must be appended to the written documentation. The group submits one report, which includes each reflection.

Upload everything to your group's folder

- Geojsons.zip[2]
- Index.html
- Config.js[3]
- Report
- Url of the finished story (e.g., http://neogeoweb.ca/Group#/A1/A1.html)

Make sure you test websites on a machine different from the one on which you created them. You are learning about client-server distribution of resources. We often find missing images or broken code (e.g., “x”s where an image should appear) when we run the code because some of the content only exists on local host.

Five points off per day for late assignments.

[2] McGill’s email system is notorious for rejecting .zip files. To fix this, simply append a letter or number to the geojsons.zip (geojsons.zip1)

[3] McGill’s email system is notorious for rejecting .js files. To fix this, simply add the config.js to the geojsons.zip folder