Data Visualization
Milestone 2

Zacharie Mizeret, Andrija Kolić, Ahmed Reda Seghrouchni, Mehdi Berrada

I. Project Goal

The goal of this data visualization project is to provide users with an engaging and informative experience that allows them to explore natural disasters around the world in a unique way. Using WebGL technology and 3D modeling, users will be able to fly around the globe and view various natural disasters in a visually compelling manner. The visualization aims to provide a new perspective on the scope and impact of natural disasters by allowing users to freely navigate the globe. This milestone report outlines the progress made thus far in developing the visualization, including a breakdown of the essential and optional steps required for its creation.

II. Tools

A. *Data Cleaning*

In this project, a crucial task was to ensure that the data set was properly prepared and organized before any visual representation could take place. To achieve this goal, we employed a combination of advanced tools for data cleaning and preprocessing. The most important part was to insure we have usable spatial data for all (or at least most) events. This was achieved by using the Nominatim geocoder service to first map countries to their latitude and longitude coordinates, and after to map more specific locations in the same way. We used the geopy library to easily interact with this service. Additionaly, we made use of the Python programming language and its broad collection of libraries such as Seaborn, Pandas, and NumPy. These tools provided us with a variety of highly specialized functions and instruments essential for managing and analyzing our large dataset, containing 15,827 cases and 43 variables.

B. *Data Visualization*

Our data visualization project is implemented using a variety of tools and libraries, each playing a critical role in creating a seamless and engaging user experience. First, we use gl-matrix, a JavaScript library for matrix and vector operations, to manipulate and transform the data used for the visualizations. Second, we use regl, a WebGL wrapper library that simplifies many of the low-level WebGL API calls and provides a more user-friendly interface. We also use the standard WebGL API directly for some of the more advanced features. Finally, we leverage the libraries provided by the Computer Graphics course at EPFL, specifically for loading and displaying 3D models and textures. These libraries provide a convenient way to manage and display complex visual elements, such as meshes and textures, allowing us to create a visually appealing and immersive data visualization that showcases natural disasters around the world.

III. Goal Breakdown

A. *Data Cleaning*

The goal of data cleaning in this dataset is to transform raw data into a format that can be analyzed and visualized effectively. This involves :

- Handling missing values (i.e. Null entries)
- Removing duplicates
- Correcting data format errors : Changing ”Bahamas (the)” to ”Bahamas”
- Figure out the best way to process the location data, since it’s not always in the same format (e.g. sometimes it’s a list of regions, which obviously can’t be mapped out at all together)

Additionally, we merged the original dataset that doesn’t contain geographical coordinates with another dataset containing latitude and longitude coordinates, to have a location file that we will use to display the event. We found out that the ”Location” column contains a lot of Null entries and we plan to look into this issue more in depth for Milestone 3.

B. *Data Visualization*

The data visualization component of this project can be divided into two categories: core and optional steps. The core steps are those that are necessary to achieve the project’s primary objective, while the optional steps are additional measures that can be taken to enhance the overall visual appeal and user experience. They are the following:

Core Steps:

1) Generate Spherical Mesh: To create the visualization, the first step is to generate a spherical mesh according to WebGL standards.
2) Create Camera and Environment Variables: Next, the camera and environment variables need to be set up to define the simulation environment’s properties.
3) Implement Drawing Pipeline for the Globe: The drawing pipeline for the globe needs to be implemented by creating vertex and fragment shaders and defining texture coordinates for the mesh.
4) Load Plane 3D Model: The plane 3D model needs to be loaded, and its associated data needs to be prepared using libraries such as gl-matrix or regl.
5) Create Drawing Pipeline for the Plane: The drawing pipeline for the plane needs to be created, including

vertex and fragment shaders and properties such as lighting, material properties, and textures.

6) Create Assets for Each Disaster Type: For each type of natural disaster, 3D models can be created using modeling software such as Blender or Maya.
7) Place Assets on the Globe: All assets need to be placed on the globe according to the dataset, and animations can be added to represent the relevant natural disaster.
8) Timeline Manipulation: A timeline must be placed on the page to allow the user to change the current year of disasters they want displayed on the globe.

Optional Steps:

1) Apply Textures to the Plane: To enhance the visualization, textures need to be applied to the plane, including the base map and any overlays.
2) Apply Normal Mapping: In order to create a more realistic look for both the globe and the plane, normal mapping can be applied to their textures. This technique creates the illusion of depth and surface detail, which is especially important for showcasing natural disasters and their impact. Normal mapping provides a more visually appealing result compared to regular ambient lighting.

So far he project has successfully accomplished the rendering of both the globe and the plane, including the creation of the camera and the necessary variables for the simulation environment. Additionally, textures have been loaded and applied to both objects.

There are a few remaining steps to complete for the project. These include creating and placing the disaster assets on the globe, implementing the ability to scroll through time on the timeline, and any additional optional steps that may be left.

IV. Sketches

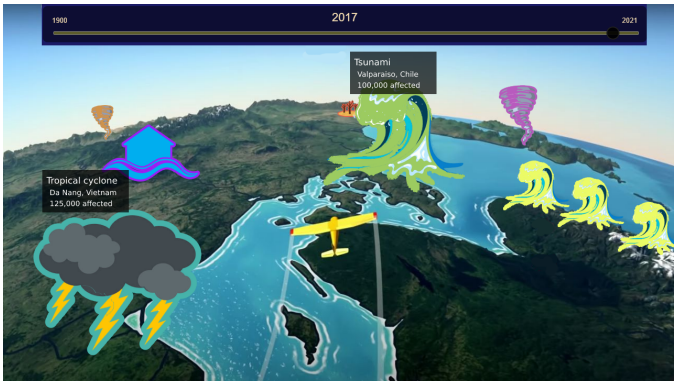

Fig. 1. Sketch of a possible visual representation of our project.

Disasters that happened at the selected time range are displayed over their geographical location, represented by simple icons and short textual description.

V. Prototype

You can find the prototype in the *project* directory of the repository. In order to run it locally you need to start an HTTP server (e.g. live-server) in the *project* directory.

Once you open the webpage, you can navigate around the globe using the W, A and D keys. You can also interact with the timeline slider at the top of the screen. Right now no disasters are rendered so there's sadly not much to see.