Ben Gorgan

5/7/2023

**Final Report**

**GeoJSON Visualizer**

The project I chose to work on is the GEOJSON Visualizer. GeoJSON is an open standard format designed for representing simple geographical features, along with their non-spatial attributes. It is based on the JSON format. The features include points (therefore addresses and locations), line strings (therefore streets, highways and boundaries), polygons (countries, provinces, tracts of land), and multi-part collections of these types. GeoJSON features need not represent entities of the physical world only; mobile routing and navigation apps, for example, might describe their service coverage using GeoJSON.

The GeoJSON format differs from other GIS standards in that it was written and is maintained not by a formal standards organization, but by an Internet working group of developers. The GeoJSON format working group and discussion were begun in March 2007[[4]](https://en.wikipedia.org/wiki/GeoJSON#cite_note-4) and the format specification was finalized in June 2008.

In April 2015 the Internet Engineering Task Force founded the *Geographic JSON working group*which released GeoJSON as RFC 7946 in August 2016.

**Project Deliverables**

I was able to develop an application that accomplished the following deliverables.The application includes a graphical user interface that the user will be able to use to upload a GeoJSON File.

My application uses information contained in the GeoJSON file to create a legible plot that displays the information within the file.

**Tools**

There are several tools that I used to help create this application. To help develop the user interface I will use the Tkinter framework (<https://docs.python.org/3/library/tkinter.html>). The tkinter package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems.

Another tool that I used was GeoPandas, which I used to help interpret the data from the GeoJSON file (<https://geopandas.org/en/stable/>). GeoPandas is an open source project to make working with geospatial data in python easier. GeoPandas extends the datatypes used by pandas to allow spatial operations on geometric types. Geometric operations are performed by shapely. Geopandas further depends on fiona for file access and matplotlib for plotting.

I also made use of the matplotlib to help generate the plots. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. This python package allowed me to integrate the GeoPandas data with the tkinter interface.

I developed this project in the Visual Studio Code development environment (<https://code.visualstudio.com/>). Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging. First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

**Review**

Overall, this project was not too difficult to implement. I made use of great tools that made interpreting the GeoJSON files quite trivial. If I had more time to work on this project, I would add additional features that allow you to fully delve into the capabilities of GeoPandas. Going through the documentation, the built in functionality is quite robust, and it allows you to do much more than I have implemented in this project. The component I would be most interested in working with is the explore functionality that allows you to explore the GeoJSON file interactively, allowing you to zoom in and out, and browse through a complete map. There are also options to dynamically fill in the plot based on different tables that can be stored within the file to create some quite amazing plots. If I had more time, I would also improve the interface of the application, making it look a little better. I would also add ways to detect errors, if the wrong file type was selected.