**Guided Project 1 –**

Problem Statement: The goal of project 1 was to use Python code to effectively read, format, and transform well known text (WKT) data from a JavaScript Object Notion (JSON) file. JSON files cannot always be imported to ArcGIS pro with the preexisting tools provided by ESRI. Therefore, to solve this problem, my main objectives were:

1. Utilize python to explore JSON data.
2. Format a python script to transform the JSON to a shapefile (shp).
3. Develop a python toolbox (.pyt) providing a tool that will read JSON data and create a shp file based on the JSON data

I began by creating a jupyter notebook to explore the data and metadata for the JSON file of interest. The jupyter notebook is a good tool for testing blocks of code, by isolating code into “cells”. This is an effective way of testing our scripts before we implement them into a user interface format. The following steps can be viewed in the file “pROJECT1.ipynb”. These steps are summarized as follows:

* The json file “no\_tax.json” was opened
* Keys were identified as “meta” and “data”
* Fieldnames for the json file “no\_tax.json” were printed:
  + sid
  + id
  + position
  + created\_at
  + created\_meta
  + updated\_at
  + updated\_meta
  + meta
  + the\_geom
  + OBJECTID
  + ID
  + Cluster Letter
  + Shape.STArea()
  + Shape.STLength()
* These fields were tested by creating a polygon from the geometry field.
* Field names were explored for a created feature class, “notax\_fc.shp”
* Fields “fc\_fullname, field\_name, field\_type” were added to the feature class
* Fields “Shape, geom” were removed from the feature class
* The last cell was used to develop and test code for an ArcGIS python toolbox that would read a json file and convert the json to a shapefile ready to import to a map project in ArcGIS. This code did not seem to work inside the notebook.

I created a python toolbox (pyt) from the catalog pane in ArcGIS pro and began formatting the code from VScode. This code can be viewed from the file “Project\_1.pyt”. After multiple attempts at the code not working, I utilized ChatGPT to see if it was able to identify any problems with my code. Minor errors with tabs and typos were identified by ChatGPT (These typos are noted to the side of where they were corrected, “#Corrected ‘displayName’”) (Figure 1).

A screen shot of a computer program

Description automatically generated

**Figure 1: code defining parameters of “Project1tool”**

This toolbox now provides an easy-to-use method for a user to create a shape from a JSON file. The user simply needs to follow these steps (Figure 2):

A screenshot of a computer

Description automatically generated

**Figure 2: Graphic User Interface (GUI) for “Project1tool” found in “Project\_1.pyt” (pre-run)**

* 1. From the catalog pane, expand the toolbox titled “Project\_1.pyt” and select the tool (script) titled “Project1tool”.
  2. Browse, enter, or copy/paste the file path to the desired JSON file in the parameter “Input JSON file”.
  3. Browse or enter the desired file path for the output feature class in the “Output Feature Class” parameter.
  4. Browse or enter the desired workspace into the “Workspace” parameter. This should be the location of your environment that the JSON file can be found.
  5. Click “Run”

A successful “Run” of this tool should result in the following messages (Figure 3):

A screenshot of a computer

Description automatically generated

**Figure 3: Result of successful run of Project1tool**

The output feature class can be found at the location determined in the “Output Feature Class” parameter, to be imported and viewed on a map (Figure 4).

A map of a lake

Description automatically generated

**Figure 4: Output feature class created from successful run of Project1tool**

This tool makes it possible to visualize JSON data and to further explore the geometry that it represents. The tool makes it possible for the user to assign values based on the fields of the shapefile attributes so that the shapefile can be used to communicate on a dimension not possible by the raw JSON data, alone. For example, symbology can be changed to focus on certain aspects of the data so that we may derive different aspects of meaning from the data(Figure 5):

A map of a city

Description automatically generated

**Figure 5: Map layout of New Orleans tax data (e.g. Symbology based on cluster length)**

Works Cited

City of New Orleans. (2018). *New Orleans land parcels* [Data set]. Data.gov. <https://catalog.data.gov/dataset/?q=new+orleans+land+parcels&sort=views_recent+desc&res_format=CSV&res_format=JSON&groups=local&publisher=data.nola.gov&ext_location=&ext_bbox=&ext_prev_extent=&page=1>

Wang, L. (2024, December 3). Guided Project 1 [Lecture]. GIS programming Geog 4057, Louisiana State University

OpenAI. (2024, December 11). Assistance with Python code for Project 1. ChatGPT. https://chat.openai.com/