Tech Reviews

CS461 - Senior Software Engr Project

Instructor: D. Kevin McGrath

Instructor: Kristen Winters

Fall 2018

Davian Lukman

November 15th, 2018

Abstract

This document will talk about our reviews of technology or in this case, language and data structures used to benefits our project in CSV processing. Our project is making a 3D visualization of data with WebGL, which requires a data as an input in CSV file. The input of CSV files consist of two sections, such as format parsing, then go into system object. This document is intended to be read by people who have experienced in computer science, especially reading buffer and data structures. The conclusion of this document is that input can be done preferably using JSON, similar as system object can be done with JSON.

CSV Processing

Before going on further, the requirement of the project must be explained. Our project will implemented using WebGL[1], a JavaScript API to render interactive 3D graphics; therefore, the language appropriate for this are HTML for page and background, and JavaScript to execute shaders. The CSV file that we got is in form of Excel spreadsheet. This could be formatted into comma separated file or JSON using CSV converter found online if necessary.

CSV Processing in general involves two separate work, such as parsing the data and turning the data into system object. Both can be done in many ways for this project to be working; however, not everything can be efficient when applied. This document will review possible technologies or data structures for this project as a discussion to decide which method is the most efficient in time and memory of our program. TThere are three ways to format parsing a CSV file, which can be done with JavaScript with jQuery, JavaScript without jQuery, and JSON. Furthermore, there are also three ways to turn them into system object, such as JavaScript Hierarchy, JSON, and Array.

Format Parsing

Inputting CSV file can be done in so many ways; therefore, limiting numbers of way based on efficiency and simplicity for our project is required for future planning. Below is the list of suggestion for parsing a CSV file using JavaScript.

JavaScript with jQuery

Data parsing can be with JavaScript jQuery-CSV[2], called csv.toObject(filename). This function will also automatically map the input as system object, which is really nice because the function does both format parsing and turning them into system object with simultaneously. It is also possible to open local files without notifying request to the server; however, the are downside from using jQuery. The first problem with using jQuery is that it is difficult to set up because it requires addition file to JavaScript. Another problem with jQuery is that the functionality is new; therefore, not all browsers can support the use of it.

JavaScript without jQuery

On the other hand, JavaScript can also do data parsing without the help of jQuery with FileReader() function. Without the help of jQuery, moving the data into system object has to be done manually within the JavaScript. The code is simple, yet still efficient to the program. Unlike the jQuery that could only convert CSV file to arrays or map. The developer can use this method to freely choose which data structures could go along with the data, in this case will be explained below inside ”Turning values into System Object” section. The only downside from using this method is that it needs more configuration as it is not automatically parsed into system object.

JSON

JSON is another option that could be use to parse a data into system object. The JSON requires an additional file to be read similar as CSV file; therefore, the users need to use CSV converter found online or made by us if necessary to convert CSV to JSON file. JSON can be read using JavaScript by mentioning the json file names and the JavaScript source name. Then, the data can be parsed into the object. Using JSON will allow the developer to freely choose the data structures appropriate for this project. Furthermore, there are two ways of using JSON[3].

The first way to use JSON is creating a JavaScript object in a separate file. This actually not how the real JSON works because this is only a JavaScript object. However, the method of parsing this data into the system object is simple as calling a function JSON.parse(). On the other hand, the second way of using JSON is with jQuery getJSON() and creating a new XMLHttpRequest. The second method can be done synchronously or asynchronously. It requires two functions to load the JSON and parse the data. Using this method is suitable to achieve efficiency in coding.

Turning values into system object

The data that we have from parsing a CSV file can be processed by using Hierarchy, array, and JSON, assuming CSV data from parsing does not automatically map the data into a system object. An appropriate system object is needed to better match 3D visualization demonstration and relationship between data. Below are list of suitable data structures suggestion to be used for this project.

JavaScript Hierarchy

The concepts of hierarchy[4] are suitable to be used in our project simply because our project also using inheritance from IOC (Indicator of Compromises) to other properties, such as sections, and area. The use of hierarchy also benefits the 3D visualization that lets the users know the relation between each values or properties. The data structures can be easily set up by giving property to each values in the CSV files. Moreover, the data can be change inside the hierarchy as well if a data from CSV changes over time.

JSON

According to the JSON webpage[5], the JSON is built on two structures, which are collection of value pairs (struct, object), and ordered list of values (array, list). Similar to hierarchy, the concept of using JSON as an object is suitable to be used in our project. As for the code perspective, JSON is pretty simple, easy to understand, and easy to use. The only difference with hierarchy being only in the code and function itself. Similar to Hierarchy, the developer can also modify the values if there are changes over time as part of user interaction.

JavaScript Array

Turning CSV data into arrays is the last thing to do when using JavaScript because it is not as efficient as others. Using array to represent data is doable; however, the results would look more like a table, which is not what we wanted. However, using array can become necessary if needed. There are also no 2D arrays in JavaScript; but array within array can be set up to create a matrix. This is the last priority of data structures that would be used in the project because of inefficiency of the object.

Conclusion

The conclusion for this tech review is not final since further discussion is needed with the clients and other team members; however, from one's perspective, one can say that JSON with JavaScript is the most suitable language to be implemented for this project in term of CSV processing for simplicity and efficiency reasons. The later followed by JavaScript without jQuery for format parsing, and JavaScript hierarchy for turning data into system object.

References

[1] M. Bailey, “Webgl,” 2018, (Accessed on 11/15/2018. [Online]. Available: http://web.engr.oregonstate.edu/~mjb/cs550/Handouts/webgl.1pp.pdf

[2] E. Plaice, “A jquery csv parser plugin,” 2018, (Accessed on 11/15/2018. [Online]. Available: https://github.com/evanplaice/jquery-csv/

[3] CodePen, “Load json file locally using pure javascript,” 2018, (Accessed on 11/15/2018. [Online]. Available: https://codepen.io/KryptoniteDove/post/load-json-file-locally-using-pure-javascript

[4] wbamberg, “Details of the object model,” 2018, (Accessed on 11/15/2018. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Details_of_the_Object_Model

[5] D. Crockford, “Introducing json,” 2018, (Accessed on 11/15/2018. [Online]. Available: https://json.org/