

User Requirements

CS461 - Senior Software Engr Project

Group 36: Data Interactive Visualization Application (DIVA)

Instructor: D. Kevin McGrath

Instructor: Kirsten Winters

Fall 2018

Bhavya Parikh, Davian Lukman, Eli Laudi, Matthew L. Jansen, and Ryan Sisco

November 28th, 2018

Abstract

The purpose of this document is to explain the user requirements for our DIVA application. The user requirement will explain system purpose, system scope, system overview, and system requirements. There are assumptions and dependencies made in order to make the application. Lastly, the work flow of the making of the application is shown via Gantt chart.

CONTENTS

1	Introduction	3
1.1	System Purpose	3
1.2	System Scope	3
1.3	System Overview	3
1.3.1	System Context	3
1.3.2	System Functions	3
1.3.3	User Characteristics	4
1.4	Definitions, Acronyms, and Abbreviations	4
2	System Requirements	4
2.1	Functional Requirements	4
2.2	Usability Requirements	6
2.3	Performance Requirements	7
2.4	System Interface	7
3	Assumptions and Dependencies	7
4	Appendices	8
4.1	Gantt Chart	8
	References	9

1 INTRODUCTION

This section defines the scope and gives a brief, yet comprehensive overview of each module within the Data Interactive Visualization Application, or DIVA web application. Furthermore, this section defines the purpose of the DIVA web application and specifies the definitions of any acronyms or abbreviations that may be used.

1.1 System Purpose

The purpose of this document is to give a high level description of the DIVA web application. This document will also describe the reasoning behind the development of the DIVA web application. Furthermore, this document will clarify how the application will interact internally between individual modules, and with any applicable external applications or software.

1.2 System Scope

The DIVA web application is a utility which assists users and stakeholders in collecting, editing, and displaying large data sets in a 3D environment. Further, this application aids in identifying and interpreting the relationships between single objects, or groups of objects, through its interactive 3D user interface.

1.3 System Overview

This subsection will give a more specified overview of the DIVA web application. This overview will include defining the system's context, functionality, and user characteristics.

1.3.1 System Context

The DIVA web application is composed of several modules, or elements, of which interact with each other and/or the user in their own manner. The first notable functionality within the web application will be the importing module, which will import a CSV file, provided by the user through the employment of a file explorer. After this operation, the input data will be parsed using a parsing module, which converts the data from a CSV format to a JSON format, making the data easier to handle for later modules. Next, the JSON data will be handled by the applications pre-rendering module, which builds relationships between the different objects generated by the previous modules, and prepares the data to the stakeholder in a user-friendly manner. The purpose of this operation is to give the user the ability to edit any characteristics or properties related to the objects or relationships generated by the previous modules, and to finalize the JSON data before it is passed into the rendering module.

Every module defined can be grouped into a 'pre-rendering' framework, whereas the modules defined here on out can be grouped into a 'post-rendering' framework, which involves passing the user provided data into a 3D environment. The next element of the DIVA web application can be defined as the rendering module, which passes the JSON data into the WebGL API. The WebGL API will render the data in a 3D environment which the user can both interact with, and view the properties or characteristics of each object and relationship. The final module defined in the DIVA web application will be the export module, which gives the user the ability to download their 3D environment in a variety of formats, including pictures, animations, or movies.

1.3.2 System Functions

The DIVA web application will be capable of converting small and large sets of data (for example, a CSV file containing several thousand lines of data) from a CSV file type to JSON format. Additionally, the application will be capable of communicating the JSON data to a WebGL API, which can hold constant user interaction. Lastly, the web application will be capable of downloading the rendered 3D environment into their a picture (PNG), animation (GIF), or movie file (MOV, MP4) format.

This project is based greatly off of the interaction provided by the user, therefore a great proportion of the application's constraints are dependent on the usability of our software. However, the usability of the DIVA web application will not have hard dependencies on the technical know-how of the user, or in other words, the application user does not need to have any particular expertise in data visualization in order to use this system.

1.3.3 User Characteristics

There are several different types of users who may use the DIVA web application. The application is targeted more towards end users who will use this software to present large data sets to clients with the intention of conveying correlations and relationships in the data that could not be seen by simply viewing the raw data. However, this shouldn't restrict other potential users from taking advantage of the DIVA web application's capabilities. No matter if the user is presenting to clients or using it for recreational data visualization, the application user will be presented with the same interfaces. The DIVA web application does not support importing custom made modules for attempting to further the visualization capabilities.

1.4 Definitions, Acronyms, and Abbreviations

Parsing: process of analyzing a string of symbol.

API: Application Programming Interface; This is a set of functionalities, capabilities, or tools which assist in building software.

DIVA: Data Interactive Visualization Application; The name of the web application under development.

3D: Three Dimensional; Any mention regarding 3D is three dimensional related to the rendering. Meaning that there will be three axis x, y, and z that will represent each data from the CSV file. For Example, each country type will have different x or y position on the visualization. *CSV*: Comma Separated Value file; A delimited text file that uses a comma to separate values. A CSV file stores tabular data in plain text. Each line of the file is a data record.

WebGL: Web Graphics Library; Allows rendering using the graphics card on a JavaScript web application.

HTML: Hypertext Markup Language; The standard markup language for creating web pages and web applications.

JavaScript: JavaScript; This is a programming language commonly used in web development.

CSS: Cascading Style Sheets; A style sheet language used for describing the presentation of a document written in a markup language like HTML.

JSON: JavaScript Object Notation; An open-standard file format that uses human-readable text to transmit data objects consisting of attributevalue pairs and array data types

IOC: Indicator of Compromise; An indicator or identification which consist of IP address and hash value of compromise.

Interactive: allow two flow information between the users and interface.

GIF: Graphic Interchange Format. A GIF is a format which plays multiple frames in a loop without sound.

MP4: MP4 is a file format created as a multimedia container format designed to store audiovisual data.

MOV: Apple Quicktime Movie file; A common multimedia container file format developed by Apple and compatible with both Macintosh and Windows platforms [1].

PNG: Portable Network Graphics is a raster graphics file format that supports loss-less data compression[2].

FPS: Frames per second; Is a unit that measures display device performance. It consists of the number of complete scans of the display screen that occur each second. This is the number of times the image on the screen is refreshed each second, or the rate at which an imaging device produces unique sequential images called frames [3].

2 SYSTEM REQUIREMENTS

Here, we will define the system requirements of the DIVA web application, of which characterize the functional boundaries of our system. These requirements will assist in describing the properties and behaviors of our web application.

2.1 Functional Requirements

This section will define the functional requirements of the DIVA web application. These requirements shall illustrate the integral actions of our project.

ID: FR1

TITLE: File input

DESCRIPTION: The application shall have an import module. This module shall give the user the ability to open a file explorer to browse the files in their local directory, in order to find a CSV file to upload to the DIVA web application.

RATIONALE: This functionality exists to allow the user to provide input data to later render into a 3D environment.

DEPENDENCIES: None.

ID: FR2

TITLE: CSV Parsing

DESCRIPTION: The application shall parse the data provided by the user in the CSV file, into JSON format. The collection of names and values within each individual JSON element will include the characteristics and properties that are included in an individual row of the CSV file.

RATIONALE: This functionality exists to convert the user input data into a lightweight format, which can later be edited by the user, and processed by the rendering framework of the application.

DEPENDENCIES: FR1.

ID: FR3

TITLE: JSON editing

DESCRIPTION: The application shall hold a user interface which allows the user to edit the JSON elements created through the CSV parsing module.

RATIONALE: This functionality exists to convert the user input data into a lightweight format, which can later be edited by the user, and processed by the rendering framework of the application.

DEPENDENCIES: FR1, FR2.

ID: FR4

TITLE: Adding and removing JSON elements

DESCRIPTION: The application shall allow the user to add new JSON elements to the data set that they input using the CSV file input module. Additionally, the user shall have the ability to remove JSON elements from the data set.

RATIONALE: This functionality exists to provide the application user flexibility when choosing which data points or relationships they would like to render, and how they would like to be rendered.

DEPENDENCIES: FR1, FR2.

ID: FR5

TITLE: Time-line integration

DESCRIPTION: The application shall provide the user the ability to define a time-line of events, of which each element of the JSON collection can be placed within a certain time frame.

RATIONALE: This functionality exists so that the user can create a story of their data, such that the development or evolution of their data can be perceived.

DEPENDENCIES: FR1, FR2.

ID: FR6

TITLE: Rendering the JSON elements using WebGL

DESCRIPTION: Once the user is done editing, adding, and removing JSON elements, the application shall pass these elements into the WebGL API for rendering. This will process the data provided by the user and render it into a 3D environment.

RATIONALE: This functionality exists so that the input data can be viewed in a 3D environment.

DEPENDENCIES: FR1, FR2, FR5.

ID: FR7

TITLE: User interaction with WebGL interface

DESCRIPTION: The application shall allow the user to browse or interact with the data displayed in the rendered 3D environment. This way, the user of the application can get more than one physical viewpoint of the rendered data.

RATIONALE: This functionality exists so that the user can provide different viewpoints of the data to their intended audience.

DEPENDENCIES: FR1, FR2, FR5, FR6.

ID: FR8

TITLE: Exporting Media

DESCRIPTION: The application shall provide a module to the user, allowing them to export their rendered 3D environment to either a picture (PNG), animation (GIF), or movie file (MOV/MP4) type.

RATIONALE: This functionality exists so that the user can place their 3D environment where they choose (for example, in a slide show). Additionally, this allows the user to present their data in an offline format.

DEPENDENCIES: FR1, FR2, FR5, FR6.

2.2 Usability Requirements

This section will define the usability requirements of the DIVA web application. These requirements will assist in defining the ease of use for the user.

ID: UR1

TITLE: File Explorer

DESCRIPTION: The application shall provide the user a file explorer option when using the file upload module to upload a CSV file.

RATIONALE: This requirement exists so that the user is easily able to upload a CSV file to the application.

DEPENDENCIES: None.

ID: UR2

TITLE: Pre-render element editing

DESCRIPTION: The application shall provide an easy to navigate, and easy to edit structure of JSON elements.

RATIONALE: This requirement exists so that the user is easily able to customize which data points they would like to render.

DEPENDENCIES: FR1, FR2.

ID: UR3

TITLE: 3D environment interaction

DESCRIPTION: The application shall provide an easy to navigate environment, of which the user can interact with their 3D environment in.

RATIONALE: This requirement exists so that non-technical users are able to explore all available options included in the WebGL API.

DEPENDENCIES: FR1, FR2, FR5, FR6.

ID: UR4

TITLE: Export module

DESCRIPTION: The application shall provide the user with a drop-down menu when choosing which file format they would like to download their 3D environment in. Furthermore, the application shall display a clearly visible 'export' button, which when clicked, will provide the user with a file explorer modal to pick where they would like their file exported to in their local directory.

RATIONALE: This requirement exists so that the ability to export media files of their 3D environment is clear and easy to do.

DEPENDENCIES: FR1, FR2, FR5, FR6.

2.3 Performance Requirements

This section will define the performance requirements of the DIVA web application. These requirements aid in characterizing our project with solid performance specifications in order to uphold user satisfaction when the application will be in use.

ID: PR1

TITLE: Data limits

DESCRIPTION: The application shall be able to handle parsing, processing, and rendering any given CSV file with a maximum of 10,000 lines of data.

RATIONALE: This requirement exists to act as an upper limit for future unit and integration testing of the application.

DEPENDENCIES: FR1 - FR8.

ID: PR2

TITLE: FPS limits

DESCRIPTION: The application shall be able to consistently keep a 30 FPS rendering of the input data set after being rendered into a 3D environment.

RATIONALE: This requirement exists so that the user can expect their rendered 3D environment to not have any diminishing qualities in regards to graphic buffering, up to the data limit defined in PR1.

DEPENDENCIES: FR1 - FR8, PR1.

2.4 System Interface

The general system interface is a web application, which can be accessed via a browser such as Chrome, Firefox, etc. The application shall be accessible across Windows, Linux, and Mac, as well as Android and IOS with compatible browsers [4].

Specifically, the DIVA web application will contain external interfaces available to the user, as well as internal interfaces available to the modules which collectively make up the application. The external interfaces will include all elements available to the user, including the CSV file import module, JSON element editing/adding/removing module, 3D environment interaction module, and the media export module. The internal interfaces are defined as the links which internal modules will use to communicate with each other. These will include the modules which parse the CSV file, send the JSON elements to the WebGL API, and convert the user's screen into one or more snapshots which are later converted to media files.

3 ASSUMPTIONS AND DEPENDENCIES

- 1) The CSV file the user uploads has at least 3 columns.
- 2) System is error free.
- 3) Assume the user is using a browser that supports the application.
- 4) Assume that the user has a medium to high-end graphic card to maintain 30 FPS.
- 5) The assumption of our project to be considered finish is the user can safely input a CSV file into the Web app. Then, the web app can successfully take the data and render to make a 3D visualization. Furthermore, the user can download the visualization in different format, such as GIF, PNG, or MOV.

REFERENCES

- [1] FileInfo, "Mov," 2018, (Accessed on 11/26/2018. [Online]. Available: <https://fileinfo.com/extension/mov>
- [2] ISO, "Png," 2018, (Accessed on 11/27/2018. [Online]. Available: <https://www.iso.org/standard/29581.html>
- [3] Techopedia, "Fps," 2018, (Accessed on 11/28/2018. [Online]. Available: <https://www.techopedia.com/definition/7297/frames-per-second-fps>
- [4] Mfluehr, "Webgl," 2018, accessed on 11/28/2018. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API