UPS Visual Analytics for Large-Scale Dynamic Flow Networks
Code Report

Rhys Newbury[1], Shenzhou Zhang[1], Jesse Bolton[1]
[1]Monash University

October 28, 2019

1 Introduction

During this project, a visualization tool was created for UPS. This tool attempts to effectively present origin-destination flow data, using a novel visualization system which leverages VR. This document describes the documentation put in place for the code developed for this project. Furthermore, we describe the limitations.

2 User Guide

We have designed a user guide, for the end user. This user guide was designed to engage the user and make the software attractive. For the basis of the designed we used SNES Super Mario World [1].
To maximize the effectiveness, we tried to minimize the amount of words and explain features with diagrams. This sets out clear instructions for the user to interact with the software.

Figure 1: Example Page from the User Guide

3 Technical User Guide

The technical user guide, instructs future developers on how to start the project. This was designed to target users with Unity and programming experience. This was designed as a getting started guide for the developer, rather then a full guide to the code. This document describes how to use the prefabs to create a basic visualisation tool using our software and Unity.

4 Code Documentation

To document the code and guide future developers on the inner workings of the code, we use Doxygen. Doxyegen is a tool used to create HTML files from source code. This tool also generates UML diagrams, allowing future developers to see dependencies and control flow. This was chosen over static UML, as it automatically updates as changes are made. This will be significantly more useful for future developers, rather than static diagrams.

5 Limitations and Further Work

This project has several limitations mostly due to the limited resources in this project.
The main limitation was the rendering of the flow lines. As the flow lines need to be recalculated and re-rendered every frame in which the objects have moved. This causes the frame rate to drop significantly below 15fps where the recommend level is around 75fps [2]. This could cause motion sickness in the users. However, this drop in frame rate only occurs when moving a large amount of flow lines quickly. This issue could be overcome by using the GPU to render the lines with less Game Objects. This technique was used previously and successfully by Yang *et al.* [3] However, our team did not have the resources to implement this.
Another limitation of our software system is that the system can only be run on very specialised and expensive machines. Furthermore, our system can only be run on Windows machines, due to the dependencies of some libraries which are only available to them [4]. These requirements put a large restriction on the portability of the system. This means that the software is not widely available to every day users in UPS. However, if the libraries were available in other operating systems, we believe our software could be easily ported.
To support mesh colliders in Unity, the geometrical properties of the map need to be simplified. This is done through an external tool [5].However, the user can still gain meaningful insights from the simplified map. For further work, it would be useful to remove the need to simplify the map, either by extending the Unity colliders or by providing an interface to the simplify the geometry automatically.

5.1 Improvements and Further Works

Further work for this project, will be evaluating the usefulness of this software, with a focus on how users interact with the space around them. This intends to examine whether users will make use of the 3D environment or are more inclined to place everything on a plane. This can be examined through a user study. This can then inform further decisions about how to best make use of the software and Virtual Reality systems as a whole.
In terms of software features, we would recommend a series of extra features, which the team did not get to complete.

- Moving Data between facilities
- Time Scale for data
- Details for flows
- Improving Frame Rate
- Automated Class Breaks
- Different amount of class separators
- Automated map simplification
- Implementing a database

5.1.1 Moving Data between facilities

To support the UPS analyst, it would be useful to be able to visualise the effect of moving packages between facilities. In the Project Proposal, we proposed two methods for achieving this. Firstly, allowing the user to move data manually through interaction. Another possible feature would be an automatic balancing algorithm, where the algorithm could propose different configuration for package flows. This would allow UPS, to visualise how to move flows to maximise their efficiency.

5.1.2 Time Scale for data

To allow for UPS to visualise how the data changes over time, we would recommend the implementation of a time slider. This could be realised as an InteractableObject. This would provide another level of data filtering to the user. The software was designed with this feature in mind, therefore this function could be easily integrated in to the software.

5.1.3 Details for flows

There is currently no method for gaining details about individual flows. This does not fit with the design principle 'Schneiderman's Mantra' [6]. To improve this software, we recommend this feature be implemented such that the user can find important details about individual flows.

5.1.4 Improving Frame Rate

Further work, will be improving the frame rate such that the frame rate does not drop when flows are being moved around. This could be done through GPU rendering.

5.1.5 Automated Class Breaks

Currently, the user has to specify class breaks manually. This can lead to class breaks which do not display information in an effective manner [7]. The process of deciding class breaks, should be automated, only allowing the user to select the algorithm(equal-interval, quantities, natural breaks) for finding class breaks. The team did not have to implement this feature, however the Legend class can easily be extended to support this functionality.

5.1.6 Different amount of class separators

Currently the legend enforces the data has 7 classes. This should be allowed to be changed from a range from 2-7 [7]. Enforcing the range, means the users of the software need to use a reasonable amount of classes such that the classed choropleth maps are readable. This feature should be easy to implement in to the Legend class, however the team ran out of time to complete this.

5.1.7 Automated map simplification

The maps need to be preprocessed, such that they can be visualised. This process needs to be completed manually. However this process could be made, automatic, such that when a user is added a map, it could be preprocessed to simplify the map or this could be done at run time by Unity. Completing this at run time would be a better option, as it would allow the user to specify a simplification level such that it fits their needs.

5.1.8 Implementing a database

To speed up the system, a database could be used rather then a CSV file. Databases have a series of advantages compared to CSV files, especially during querying of large datasets. A larger dataset will be introduced if the time dependent dataset is integrated. This may extend past the scalability limit of CSV files. Therefore, a useful extension to this software would be integrating with a database, specifically SQL as it is natively supported in C# [8]. Furthermore, this would allow for easier updates to the database for the UPS staff.

6 References

References

[1] 2019. URL `https://www.nintendo.co.jp/clvs/manuals/common/pdf/CLV-P-SAAAE.pdf`.

[2] Navid Farahani, Robert Post, Jon Duboy, Ishtiaque Ahmed, Brian J. Kolowitz, Teppituk Krinchai, Sara E. Monaco, Jeffrey L. Fine, Douglas J. Hartman, and Liron Pantanowitz. Exploring virtual reality technology and the oculus rift for the examination of digital pathology slides. *Journal of pathology informatics*, 7:22–22, May 04, 2016. ISSN 2229-5089. URL `https://www.ncbi.nlm.nih.gov/pubmed/27217972`. 27217972[pmid].

[3] Y. Yang, T. Dwyer, B. Jenny, K. Marriott, M. Cordeil, and H. Chen. Origin-destination flow maps in immersive environments. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):693–703, Jan 2019. ISSN 1077-2626.

[4] 2019. URL `https://store.steampowered.com/app/250820/SteamVR/`.

[5] 2019. URL https://github.com/mbloch/mapshaper.

[6] Brooke Fitzgerald. Schneiderman's mantra, 2016. URL https://hampdatavisualization.wordpress.com/2016/02/26/schneidermans-mantra/.

[7] 2019. URL https://www.axismaps.com/guide/univariate/choropleth/.

[8] 2019. URL https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqlconnection?view=netframework-4.8.