MSc in Advanced Computer Science
FHS Computer Science; Mathematics and Computer Science; Computer Science and Philosophy

DATA VISUALISATION

Hilary Term 2023

Submission deadline 12 noon, Tuesday 11th April 2023, via Inspera.

There is a total of 100 marks available for this paper, you should attempt all parts of the paper.

NB: You must not discuss this examination paper with anyone.

- The assignment has two questions. The first question (55 marks) is on the implementation of a visual analysis app or an interactive data-driven story in D3.js on a topic of your choice. The second question (45 marks) is a written report describing your project in detail, based on the foundations of Visual Analysis Design seen in the lectures.
- Please read the exam in full before addressing any of the questions, as this may help you to identify possible connections between its different parts.
- Please submit your written report as a `.pdf` file, typeset with a font size of at least 11 points, and using A4 paper format with all margins set to at least 2 cm. The maximum report length must not exceed 3000 words.
- The front page of your written report (not counting to the total word count) must clearly indicate the title of your project and your Candidate Number, with no other identifying information. It must also include the following statement: "I declare that, except where otherwise indicated, this mini-project is entirely my own work, and that it has not been previously submitted and/or assessed and is not due to be submitted on its entirety or in part for any other course, module or assignment", and fully abide by it.
- Please submit answers to all questions (code implementation and written report) as a single `.zip` file, named using your Candidate Number.

TURN OVER

Question 1

Implement a visual analysis app or interactive data-driven story (narrative visualisation) in D3.js that is ready to be deployed and that can be consumed by your intended audience.

The theme of your visualisation can draw from any topic. Keep your usage scenario in mind when designing your interface. Make sure that your visualisation's interface and functionality either: (a) allow a user to answer your proposed question(s), or (b) successfully communicates a message or series of messages.

Your visualisation should comply with the following guidelines and requirements:

- You are free to choose any publicly-available dataset(s). Useful resources for the identification of publicly-available datasets can be found in the 'Course materials' section of the course website. Include a URL linking to the source of your data.
- Your visualisation should be standalone and have an `index.html` file as entry point.
- Make sure that your code is well-organised, easy to read, and contains code comments.
- The use of custom backends (e.g., Node.js, Python, etc) and database systems (e.g., Postgres or MySQL) is not allowed. Although they could facilitate more powerful applications, in this mini-project you should put your efforts into the front-end development (using only JavaScript, D3.js, HTML, and CSS).

Your visualisation will be evaluated against the following criteria:

(a) Design. Your interface should consist of at least two different types of views/visualisation components (e.g., two bar charts would only count as a single component). At least one of your views should be an "innovative view" that is either: (a) a significant extension of an already used visualisation component in the taught coursework, or (b) a novel visualisation component. It cannot be a simple re-use of any visualisation component that you have already used in the taught coursework. The extension could consist of an adaptation of the visual encoding, or substantial new interactivity features. Use separate files for different visualisation components, using functions to promote code reuse. (25 marks)

(b) Functionality. Views should be coordinated with linked highlighting (i.e., an interaction event in one view should trigger a change in a different view). At least two UI widgets should allow users to filter the data or update certain views interactively (e.g., dropdown menus, checkboxes, range sliders), or provide more information (e.g., tooltips on hover). (20 marks)

(c) High level of style polish in final interface. Your interface should be as self-documenting as possible, with a meaningful title, appropriate labels and displayed text for panels, axes, and interactive HTML elements, legends documenting the meaning of visual encodings, and any necessary descriptions to facilitate interaction with your idioms. Make appropriate use of CSS for achieving a consistent styling across views. (10 marks)

Question 2

In addition to the implementation, you must submit a written report as PDF. **Your report should be a standalone document that fully describes your project, and it must not exceed 3000 words.** There is no hard limit on the number of screenshots you can include as you need to illustrate your project clearly, but try to keep it to a reasonable number.

Please include the following sections in this order:

(a) Overview

- Teaser image (screenshot) of your visualisation.
- Concise summary of your project (max. 250 words). This should consist of a few sentences describing what problem your visualisation is tackling and how. State the intended audience. Be brief and clear.

(5 marks)

(b) Data

- Include a URL linking to the source of your data and/or the associated metadata.
- Describe your original and derived data in both domain-specific and abstract language (dataset type, data types, and their range/cardinality), discussing clearly which features or subsets you have used directly in your application. If your dataset has many more than 20 attributes and you plan to visualise them all, then you may provide a high-level descriptor of groups of attributes (e.g., saying that the dataset contains *demographic attributes* instead of describing every single variable).
- Briefly describe your data preprocessing pipeline, if there is one. Any required data preprocessing (e.g., to derive new data, to join multiple datasets, etc) can be performed with either separate tools (e.g., R scripts, Python) or on the fly in JavaScript.

(5 marks)

(c) Goals and Tasks

- Description of your intended task(s) in both domain-specific and abstract language. You should aim for a total of four tasks. Do not discuss the visual encoding or interaction idioms chosen in this section.

(10 marks)

(d) Visualisation

- Describe the visualisation interface that you have built. What views are there and what do they allow users to do? For each view, describe your visual encoding choices and include the rationale for your design choices. You should analyse the visual encoding in detail, in terms of marks and channels, only for your innovative view. For the other views, if they have already been analysed in any of the lectures, you can describe the visual encoding very concisely – just explain how your data abstraction is mapped to the elements of the view.

- Provide rationale for your design choices, for both visual encoding and interaction, including appropriate choice of colour schemes. Why are they appropriate, in terms of the principles covered in this course, for your chosen data and tasks?
- How can users interact with your project within each view, and how are views linked?

(20 marks)

(e) Usage scenario

- Include a usage scenario walking through how your visualisation can be used during an interactive session, illustrated with screenshots of your system in action.

(5 marks)

(f) Credits

- Indicate any sources of inspiration, including any specific D3.js code blocks that you consulted or built upon. Explain what changes you made and their magnitude (e.g. unchanged vs. minor tweaks vs. major functionality additions) for any code that you built upon.

(Mandatory section to address plagiarism concerns, 0 marks)

LAST PAGE