

Objectives

With these assignments you will learn how to

- Generate hierarchical layouts in D3.js
- Handle graph structured data in D3.js
- Apply the Voronoi tessellation on a geographical data set

For your programming tasks, you will use D3.js together with JavaScript integrated in a HTML file. For the second task it is necessary to create a Python script for data preprocessing. First install node and a simple web server by typing

- `npm install http-server -g`

in your command line. In order to load e.g. CSV files you need to start a local server by typing

- `http-server -a 127.0.0.1 -o`

in the directory where your files are located. In each subtask you need to modify the provided HTML file, e.g., modify `Ex2c.html` for Exercise 2 c). If images are available, they serve only as an example and do not represent a binding solution. The sizes of the DOM objects, e.g., the radii of circles or color schemes, should be chosen in an appropriate way but are not restricted to one single possibility. Color schemes can be loaded from [colorbrewer¹](#), for details about D3.js you can refer to its documentation². Your solutions must be uploaded via Moodle till **January 27, 2021, 9am (UTC+1)** as one ZIP file that contains all HTML files and the respective data sources. The naming convention for this ZIP file is `sheet4__<Matrikelnummer1>__<Matrikelnummer2>.zip`.

Exercise 4.1: Visualizing Migration Flow Data (5 Points)

The data set *MigrationFlow.csv* contains the number of emigrants and immigrants between two countries. This fact can be represented by a directed, weighted graph. Create a visualization that shows the migration between continents in a time period of your choice. Your visualization should allow at least one interaction by the user. Make sure that the labels are appropriate and choose a suitable color scheme.

Exercise 4.2: Visualizing Hierarchical Structure of Source Code Files (10 Points)

We look at the C++ files of the *globjects* project³ as an example of hierarchically structured data provided by its file structure. First write a Python script to extract the Lines of Code (LoC) of each C++ file. Create a meaningful representation of this data that arranges the C++ files according to the folder structure and relative to the LoC as size. Your visualization should allow at least one interaction by the user. Make sure that the labels are appropriate and choose a suitable color scheme. In addition to the delivery of the html file for visualization, it is necessary to include the script for data generation and the data itself in the submission.

Exercise 4.3: Voronoi Tessellation of Geographic Data (10 Points)

The file *county_usa.csv* contains the administrative boundary lines of all US states, the file *postoffices_usa.csv* contains locations of post offices. Create a map of the entire USA in which the post offices are shown as dots. Then calculate the Voronoi tessellation of the USA with the post offices as central points. Enrich your visualization with at least two interaction techniques.

¹Colorbrewer2

²D3.js Documentation

³<https://github.com/cginternals/globjects>

Instructions

Pair Programming On these assignments, you are encouraged (not required) to work with a partner provided you practice pair programming. Pair programming „is a practice in which two programmers work side-by-side at one computer, continuously collaborating on the same design, algorithm, code, or test.“ One partner is driving (designing and typing the code) while the other is navigating (reviewing the work, identifying bugs, and asking questions). The two partners switch roles every 30–40 minutes, and on demand, brainstorm.

Violation of Rules A violation of rules results in grading the affected assignments with 0 points.

- Writing code with a partner without following the pair programming instructions listed above (e.g., if one partner does not participate in the process) is a serious violation of the course collaboration policy.
- Plagiarism represents a serious violation of the course policy.