

Objectives

With these assignments you will learn how to

- Discuss a given visualization
- Create a force-directed graph in D3.js
- Work with geographical data
- Work with two different frameworks for 3D visualization techniques

Your solution for exercise 1 and 2 should be included as a single file names *Exercise 1.pdf* and *Exercise 2.pdf* in your final submission. In exercise 3 and 4 of your programming tasks, you will use D3.js together with JavaScript integrated in a HTML file. Additional Python scripts for data processing must be included in your submission as well. 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. 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 **February 22, 2021, 9am (UTC+1)** as one ZIP file that contains files required and the respective data sources. The naming convention for this ZIP file is `sheet5_<Matrikelnummer1>_<Matrikelnummer2>.zip`.

Exercise 5.1: Evaluation of a 3D visualization for real phenomena I/II (2 Points)

The given visualization *rainstation.gif* represents punctual rainfall data for the State of Arizona in the USA. Identify at least two problems that affect the readability and informative value. Provide suggestions for their improvement. Further information may be found at *United States Geological Survey - Rainfall Mapping*³.

Exercise 5.2: Evaluation of a 3D visualization for real phenomena II/II (4 Points)

Met.3D⁴ is an open-source visualisation tool for the interactive, three-dimensional visualisation of numerical ensemble weather forecasts and similar numerical atmospheric model data sets. Evaluate the given visualisation *Rautenhaus2015.png* in terms of effectiveness (visual representation) and expressiveness (visual presentation and graphical integrity) of the data shown. Consider factors of perception as well as technical implementation. Further information may be found at *Rautenhaus et.al.*⁵.

Exercise 5.3: May the force be with you (5 Points)

The dataset *starwars-full-interactions.json*⁶ describes the connections of different characters of the Star Wars film saga as an undirected graph. The nodes are the characters, which are described by a *name*, the *number* of scenes in which they appear during episodes 1-7 and a *color*, which should represent the color of the nodes in your visualization. The edges are given by their *source* and *target*, which index the characters starting at 0. The weight of each edge is contained in value and

¹Colorbrewer2

²D3.js Documentation

³United States Geological Survey - Rainfall Mapping

⁴Met.3D

⁵Rautenhaus et.al. (2015). Three-dimensional visualization of ensemble weather forecasts.

⁶GitHub evelinag/StarWars-social-network

describes the frequency of the common scenes. Create a force-directed graph using D3.js. Define a mapping for the weights of the graphs and the weights of each node on visual variables. Your solution should contain at least two interactions techniques.

Exercise 5.4: Location-Based Insights in Berlin (10 Points)

The dataset *Luebeck.geojson*⁷ contains the administrative boundaries of Lübeck. An additional data set *stores-attr.csv*, provided by *Targomo GmbH*, contains punctual semantic data of various businesses or shops (for example, attractiveness) in Lübeck. Design and implement a 2D⁸ visualization using *d3*⁹ in which both data sets are spatially combined and which represents an aspect chosen by yourself. The visualization should allow users to interact spatially as well as semantically by means of at least two interaction techniques.

Exercise 5.5: Treemap Visualization (10 Points)

In this task you are asked to create a 2.5D treemap using the module *treemap-min* of the the project *bitcoin*¹⁰. Write a Python script *crawling.py* that extracts all the C++ and Python source code files of the master branch. The size of the cuboids should represent the Lines of Code of the files. Extend your script so that two additional statements about the developers' activities are extracted and map them to the visual variables height and color. The configuration of your solution must be done in the file *index.html*. Your submission must contain the files *crawling.py* and *index.html*. For starting the treemap module you need to start a local server by typing

- `python -m http.server 8080`

in the directory where your files are located. You can then view the treemap by typing

- `localhost:8080`

in your address line.

Exercise 5.6: Bonus: WebGL 3D Scatter Plot (10 Points)

The module *scatter-min* provides all building blocks for building your own scatter plot. In this task you are asked to modify the file *index.html* to visualize the dataset *example3D.csv* with your scatter plot. Your implementation should enable the following features

- Rendering of coordinate planes
- Representation of projection lines of a point on the coordinate planes
- Labeling of selected points
- Scaling of the point size depending on the depth

Your submission must contain the file *index.html*. For starting the treemap module you need to start a local server by typing

- `python -m http.server 8080`

in the directory where your files are located. You can then view the treemap by typing

- `localhost:8080`

in your address line.

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,

⁷<http://opendatalab.de/projects/geojson-utilities/>

⁸or on a non-mandatory basis a 3D Visualization

⁹d3.js Project

¹⁰<https://github.com/bitcoin/bitcoin>

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.