

# LSOVOMAN - Large-Scale Online Visualization of Austria's Media-Advertisement Networks

## Report

The following sections describe the problem and solutions for this project.

### Goal

Based on the works of Katharina Unger [1] the readability of the firework plot should be increased by providing a layout that constraints the children of a cluster within a unique area to avoid overlaps of cluster once the children are rendered. Furthermore, the user interaction is enhanced by giving the option to filter by date (quartal) to be able to explore the development of investments by time.

### Implementation / Solution

The main problem, constraining child-nodes to a unique area, is solved using a hierarchical constrained based layout proposed by T Dwyer et. al [2]. The input for the layout algorithm is, as previously utilised agglomerative clustering proposed by M. E. J. Newman [3]. The clustering algorithm yields hierarchical structured clusters, which are used to compute the optimal layout so that nodes that are part of a cluster do not intersect with nodes of another cluster, as well as no overlaps between nodes of the same cluster.

A problem that occurred using this algorithm was that the unique areas for each cluster degraded to very thin but long areas, as well as nodes moving to the boundaries of the areas lining up next to each other making the visualisation less appealing.

This problem is solved by modifying the weight of each individual node (which also increases the spatial extent in the algorithm) while also increasing the number of optimisation iterations without constraints while keeping the number of iterations with constraints the same.

These steps were done as a Team by Simon Pointner and Thorsten Korpitsch.

Filtering by quartal is implemented by iterating over all available edges and nodes and changing the rendering size relative to the investment amount over the full time

and the investment amount in the selected range. Changing the rendersize is done by modifying the buffers.

Simon Pointners report will further go into detail how cluttering was reduced by managing the visibility of inter cluster links.

The website is hosted using a django backend.

## References

[1] Interactive Visual Exploration of Large Bipartite Graphs using Firework Plots, Katharina Unger, 2020

[2] IPSep-CoLa: An incremental procedure for separation constraint layout of graphs, T Dwyer, Y Koren, K Marriott. Visualization and Computer Graphics, IEEE Transactions on 12 (5), 821-828, 2006

[3] Fast algorithm for detecting community structure in networks, M. E. J. Newman 2004

## User Documentation

The following sections describe necessities for running the program.

### Prerequisites

The project is built with python version 3.9. The following additional libraries are required to run the program.

Module	Version
Django	3.1.3
pandas	1.1.4
networkx	2.5
numpy	1.19.3
setuptools	50.3.2
Cython	0.29.21
pathlib	1.0.1
requests-html	0.10.0

enums	0.0.2
-------	-------

Also mind any dependencies of these modules need to be installed as well.  
Recommendation: Use a (python) virtual environment if you are developing.

## Repository

The repository can be found at <https://gitlab.cg.tuwien.ac.at/tkorpitsch/lsovoman> .

## Setting up the Backend

The database holds all information of media entities, legal entities and related investments. To create the database use the following commands in a terminal:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

To be able to update / create data you need to create a super user with the following command:

```
python manage.py createsuperuser
```

After entering the command you will be prompted for a username, email and password. This can be used to login on <http://localhost:8080/admin> (default, actual path depends on configurations).

## Starting the Server

To run the server enter the following command in a terminal:

```
python manage.py runserver localhost:8000 --noreload --nothreading
```

The options `--noreload` and `--nothreading` are necessary for enabling the django backend to execute javascript via requests-html which is used to compute the graph layout.

After starting the server it is available at <http://localhost:8080/> where the graph is visualised while the data can be updated at <http://localhost:8080/admin> .