



What is grApp?

GrApp is a web-based and open-source graph editor. Its main feature is a smoothly animated visualization, which allows users to modify and analyze the graphs interactively.



Why do we need grApp?

- Smooth and interactive graph visualization and modification in 2D and 3D
- Fully web-based and thus platform-independent
- Open-source
- Interface for user-created executables (interacting with the displayed graph data via .gml)
- Import, export and conversion of 10 different graph formats
- Graph6 webinterface



GrApp supports also a basic 3D layout of the graph.



Using grApp:

You can import graphs manually by using the input field or per file drag-and-drop on the drag-n-drop canvas field. GrApp accepts the graph formats graph6, sparse6, GML, graphML, Leda, Edge List, GEXF, CSV, TSV and AdjacencyMatrices (further formats like graphML, GEXF and DOT are experimental). Some of these can also be exported by using the "Save as"-button. Additionally, graphs can be exported as a JPG image.





GrApp supports interactive functionalities such as
moving, adding and deleting (subsets of) vertices and edges,

fixing the coordinates of a subset of vertices while dynamically layouting the remaining vertices (e.g., to frame planar graphs in an outer face),
setting individual (right-click) and global visual styles of vertices and edges.

-		edge curves	0
Enable Interaction Edge visibility	N N	edge width	1
Show vertex labels Fix on move	X X	vertex radius	1
Fix all vertices Color background	\checkmark	force collide	0
Color all vertices Color all edges		edge length	1



GrApp supports basic curved edges.



Integration of Executables

grApp supports user-made executables that may interact with the displayed graph and its current layout, such as here a radial layout algorithm. The interface to the graph is provided by exchanging a .gml-specification of the graph, which the executable has to read in and output again (after possibly modifying it). Arbitrary executables may be integrated from the field "commands" in the sidebar. The command-line parameters for this executable can be set by clicking the parameters button.



GrApp's layout algorithms and interacting with the graph

For displaying interactions with the graph smoothly, grApp uses the force-layout algorithm of d3.js continuously. This assigns forces such as gravity or repulsiveness among the set of edges and vertices. Once the forces of the vertices and edges have been defined by the force-graph algorithm, the simulation can generate different layouts according to the forces applied. The user may create clusters of vertices with different specifications. Some of them can be with fixed positions, which means no forces will be applied, while others repel or attract each other. The fixed positions can be set from the checkboxes on the sidebar. The yellow area surrounding some of the vertices and edges indicates that they are selected by the user. In this way, attributes can be assigned to multiple elements in bulk by right clicking on any of the selected elements. All of the key-bindings for interaction can be found in the instructions or by clicking on the "key-bindings" button on the toolbar.





Graph that illustrates a christmas tree.



Graph6-Webinterface

You can share a graph displayed in grApp as a link by using the share substrated button on the toolbar. A web-link with the graph encoded in graph6 format will be generated (see below) and is available for anyone that has this link as the grApp server is online. You can easily send the link to colleagues and friends, which will be able to open the visualized abstract graph by clicking on the link. Other programs may use the graph6-webinterface as well for output purposes: they only need to attach the graph6 encoded string to the grApp base web-link in order to have a clickable-link to the visualized graph.

Share link

bject 🕨 r

https://www3.tuhh.de/e11/schmidt/grApp/graph/?g6=~?B_s????????????????@??O??_

te Graph Data	JSON Editor		
Tree -	You can insert, remove or change custom fields of the		
odes ► 3 ► graphics ► fill	graph file from the ISON Editor page where the graph		
object {2}	graph lie from the JSON Editor page where the graph		
▼ nodes [224]	displayed in the JSON format. You can even filter the		
▶ 0 {8}	data or search for specific vertices by a certain attribute		
▶ 1 {9}	using the buttons in the toolbar		



In the example it is shown how you can easily change color of a vertex.

After you have finished modifying, click on the update graph button, which will redirect you to the editor page and will render the modified state of the graph.

Under the hood

GrApp uses the force-layout used by d3.js. JavaScript, HTML and CSS, are used for building the user interface, while Python and the Django framework runs on the server side. NetworkX enhances the application with further graph algorithms and file parsers.

Contributors: Ivelin Ivanov Jens M. Schmidt



Official page: https://www3.tuhh.de/e11/schmidt/grApp/ github page: https://github.com/iviv62/grapp