2012

Open Source project

Graph Generator

Manual EN

Beta 0.9

About project

The Graph Generator project is a new open source project which manages generating of graphs of various types. These graphs can be used as an input data for testing your graph algorithms

Graph generator generates graphs according to its user wishes, e.g. complete graphs, discrete graphs, trees, simple acyclic graphs, etc.

The output file format is optional. Graph generator offers well-known formats for graph representation - Trivial Graph Format, Graph Modelling Language and DOT for graph visualization in Graphviz software.

Technologies

- Java SE6 – application is executable on various platforms
- Swing – simple GUI supported in Java SE6
- XML – common universal format for text representation of data objects
- DOT – file format used for representation of graph objects, widely used in GraphViz software for graph visualization

Description of program

After executing file `/dist/Graph-generator.jar` main window is opened(shown in Figure 1). First you have to choose vertex count of the required graph(In the case you are going to generate bipartite graph leave this field empty, vertex count will be filled below). The specified value must be positive integer otherwise warn message beside the field and in the log will be shown(similarly as in the Figure 2).

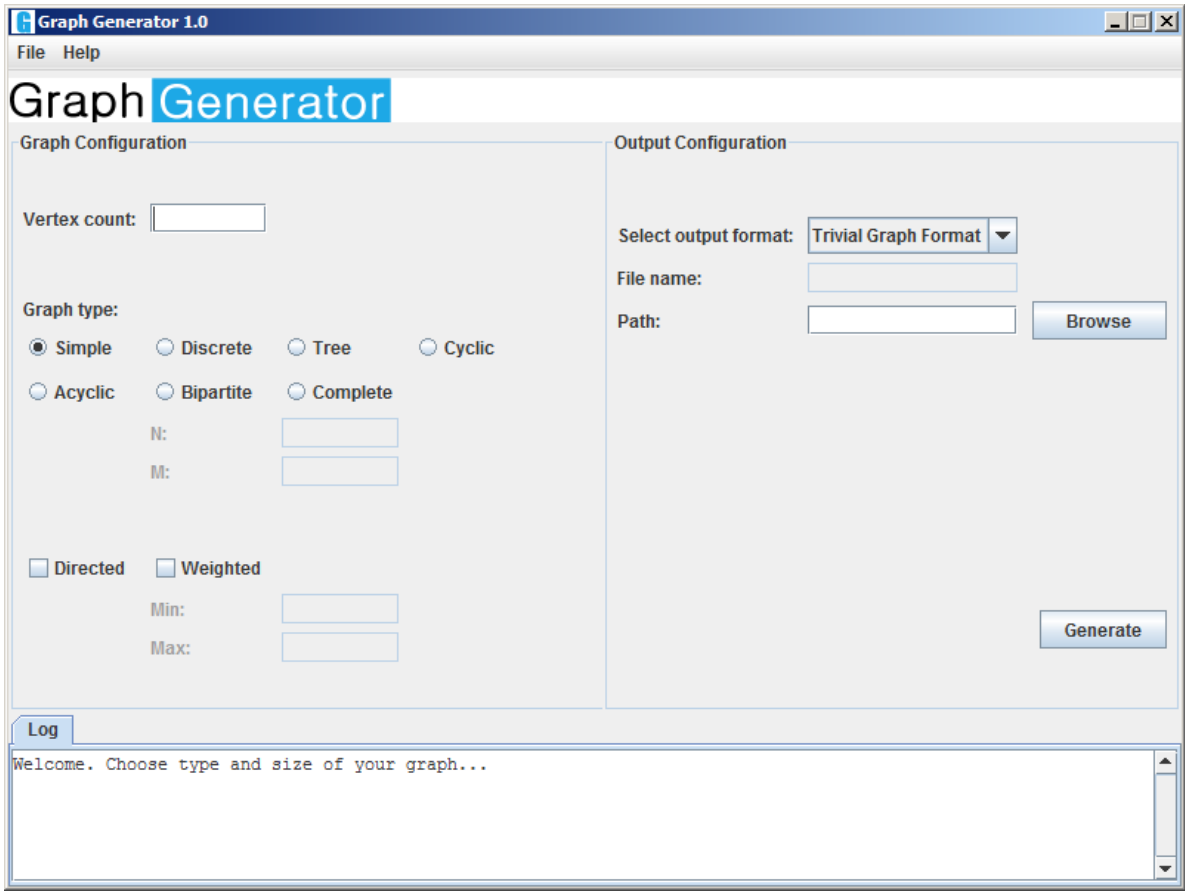

Figure 1 – Main window

Second step is choosing of graph type. Currently six different types are supported:

- **Simple** – Simple graph (connected graph without multiedges)
- **Discrete** – Discrete graph (graph with no edges)
- **Tree** – Simple acyclic graph
- **Cyclic** –Graph with one or more cycles
- **Acyclic -** Graph without cycles

- **Bipartite** – Bipartite graph. In this case please fill in vertex count of both graph parts as a positive integer.
- **Complete** – Complete graph (every vertex is connected to every other vertex)

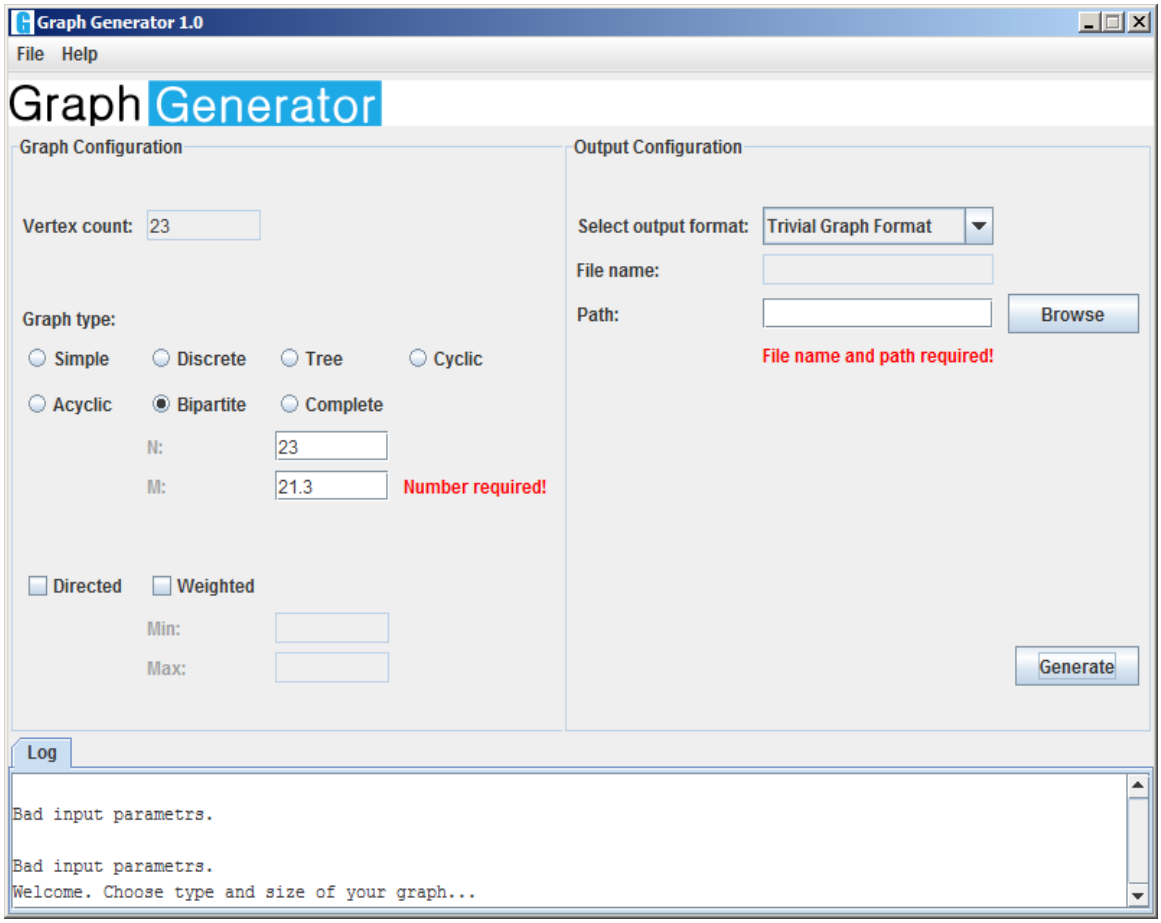

Figure 2 – Data validation and warning messages in log

In next step you choose between Directed and Undirected graph, you can also choose the range of edge weight (interval $\langle min, max \rangle$) as two real numbers.

Last two steps of setting are choosing output format of generated graph and output file placement. Currently 5 common formats for text representation of graphs are supported:

- **Trivial Graph Format –** simple format where every line of output is pair of vertex IDs, an edge exists between theese two vertices
- **DOT** – file format used for representation of graph objects, widely used in GraphViz software for graph visualization. Visualization in GraphViz of graph generated by Graph Generator is illustrated on Figure 3.
- **XML** – Standard format for text representation of structured data
- **Incidence Matrix –** basic representation of graph

- **Adjacency Matrix -** basic representation of graph

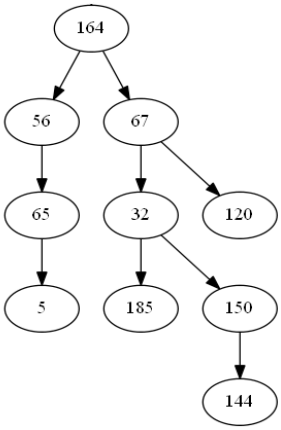

Figure 3 – Graph visualization in GraphViz

After choosing output file format click the **Browse** button and choose the path and the name for the output file similarly to the Figure 4.

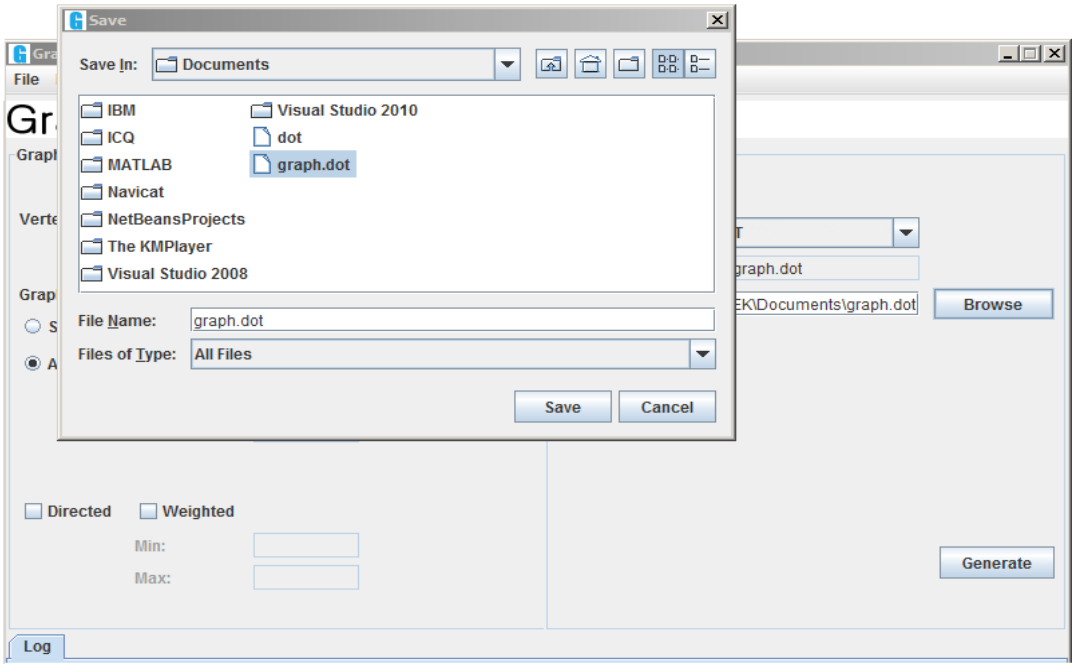

Figure 4 – Output file save

Aftef choosing output file, click the button **save** and after check of filled parameters click the button **Generate** for start of graph generating. During the run of program the application state is being reported in the log. After completion of generating the path to the output file is stated in log too.

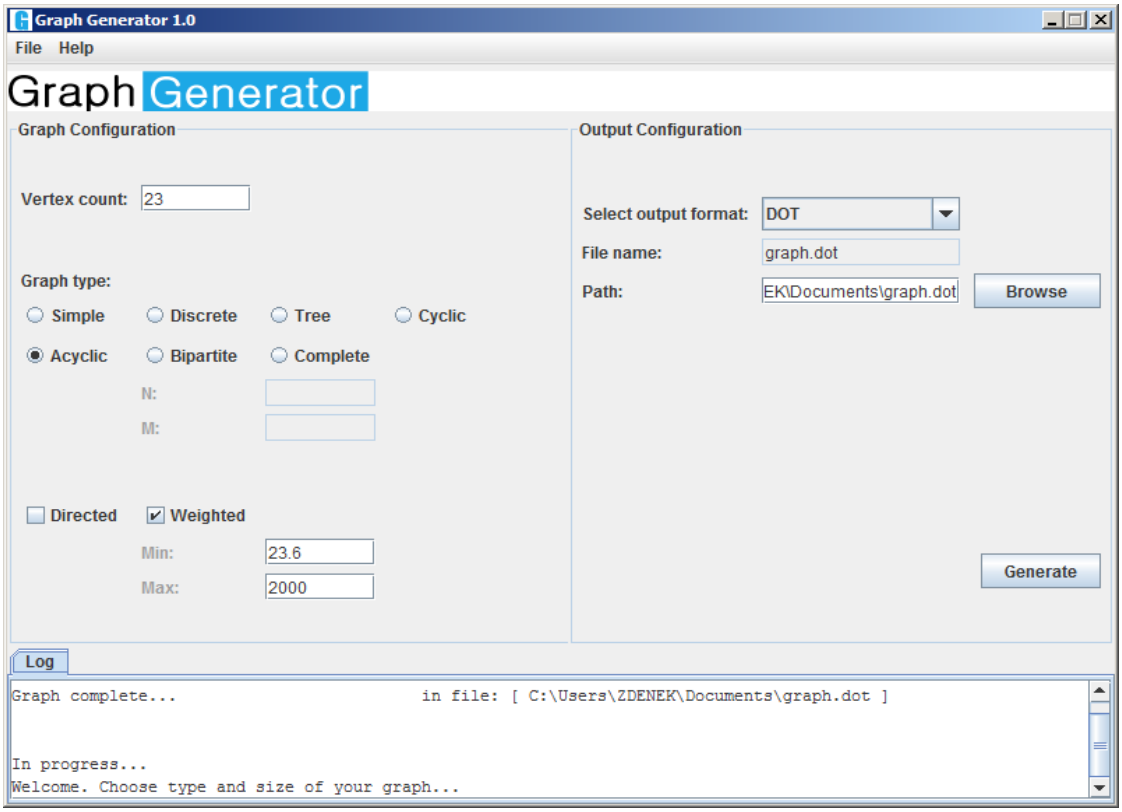

Figure 5 – Program log

Links:

Wiki pages:	https://github.com/dusatzde/Graph-Generator/wiki
Project website:	http://dusatzde.github.com/Graph-Generator/
GitHub repo:	https://github.com/dusatzde/Graph-Generator
Mailing List:	https://groups.google.com/forum/?fromgroups#!forum/graph-generator