MSI
Computer Science Department

Växjö University

Fall 2008 – LP2

Course Assignments for Graph Drawing DA4124 – Fall 08

2nd assignment

Deadline for this assignment is Dec 17, 2008.

All groups should develop a simple Java tool that is able to read an input file with a graph specification and to draw this graph (including the nodes' id) within a scrollable 2D drawing area. Each group will implement a different drawing algorithm or variations of one algorithm. A simple GUI is needed to allow the user to load input files (based on GraphML) easily and to modify options if possible. The latter issue depends on the implemented algorithm, e.g., the width W of the Coffman-Graham-Layering should be modifiable via the GUI. You should use the following graph specifications for testing:

`http://cs.msi.vxu.se/isovis/courses/fall08/da4124/assignments/dags-GraphML.zip` (DAGs)
`http://cs.msi.vxu.se/isovis/courses/fall08/da4124/assignments/digraphs-GraphML.zip` (digraphs)

Task 1 *GraphML (all groups)*
Implement a parser for GraphML input files (a specification of GraphML can be found on the course home page). This graph format is based on XML. Your tool should be able to parse simple graph specifications (i.e., nodes and edges), such as given in the ZIP-files with graph examples.

Task 2 *Tree Layout (Group D)*
Generalize the algorithm of Rheingold/Tilford, such that the resulting algorithm is able to draw general trees. Small subtrees should be equally distributed between larger subtrees. A good reference is: J. Walker. A Node-positioning Algorithm for General Trees. In Software-Practice and Experience, 20(7). pp. 685-705, 1990, (`http://cs.msi.vxu.se/isovis/courses/fall08/da4124/assignments/walker.pdf`); also if this algorithm doesn't work in linear time as stated. Implement the algorithm and embed it into a tool as described above. For testing, try to find suitable trees in `dags-GraphML.zip` or use your own tree specifications.

Task 3 *Hierarchical Layout of DAGs – Variant 1 (Group C)*
Implement the hierarchical approach for creating polyline drawings of acyclic digraphs. Implement the algorithm and embed it into a tool as described above.

- Layer Assignment: Longest Path Layering
- Crossing Reduction: Hybrid Approach discussed in the course
- Coordinate Assignment: Simple dummy node removal

Task 4 *Hierarchical Layout of DAGs – Variant 2 (Group A)*
Implement the hierarchical approach for creating polyline drawings of acyclic digraphs. Implement the algorithm and embed it into a tool as described above.

- Layer Assignment: Coffman/Graham Layering
- Crossing Reduction: Adjacent-Exchange AND Split (user can choose)
- Coordinate Assignment: Simple dummy node removal

Task 5 *Hierarchical Layout of DAGs – Variant 3 (Group B)*
Implement the hierarchical approach for creating polyline drawings of acyclic digraphs. Implement the algorithm and embed it into a tool as described above.

- Layer Assignment: Coffman/Graham Layering
- Crossing Reduction: Barycenter Method AND Median Method (user can choose)
- Coordinate Assignment: Simple dummy node removal

Task 6 *Hierarchical Layout of Digraphs – Variant 4 (Group E)*
Implement the hierarchical approach for creating polyline drawings of digraphs. Implement the algorithm and embed it into a tool as described above.

- Preprocessing: Cycle Removal
- Layer Assignment: Longest Path Layering
- Crossing Reduction: Barycenter Method
- Coordinate Assignment: Simple dummy node removal

Please prepare a demo of your tool as well as a short presentation (about 5-10 minutes) (PowerPoint, Latex, PDF, ...) on the most important aspects of your implementation like data structures, etc. You will present both during the second exercise at Dec 17th, 2008. Please, use your own notebook for the demo and the presentation! Send all files to Andreas Kerren via email by the given deadline! Any kind of plagiarism is not accepted and leads to the consequence that all group members will fail the course.