LayerBased Algorithm Specification/Documantation

Bennet Bleßmann, Sven Korfmann

June 1, 2018

Contents

1 Import

This Section will provide Instructions on how to import this Project into Eclips and IntelliJ Idea.

1.1 Clone

As a first step you should clone the project to your device. We assume you have cloned the project to the folder: ∼/git/AGD-Project

1.2 Install JDK-10

1.3 Eclipse

InteliJ IDEA Setup 1.4.1

1.3.1 Setup Eclipse to use JDK10

1.3.2 Import Project to Eclipse

1.4 IntelliJ IDEA

1.4.1 Import to InteliJ IDEA

2 Guidlines

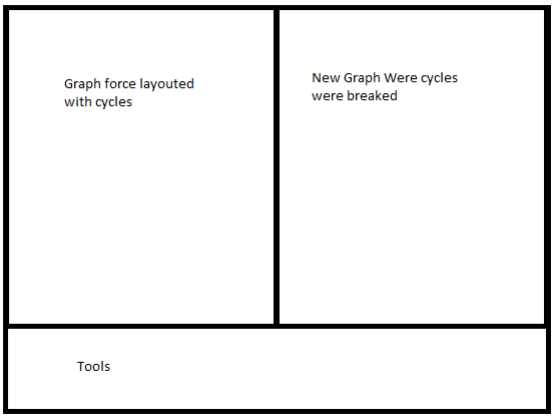
- The Base package is *de.webtwob.agd.projekt.cyclebreake* and all packages from here on will be relative.
- Project will be conform to MVC pattern.
- Model will be in the package *model*.
- View will be in the package *view*.
- Controler will be in the package *controls*.
- Code Comments and this Specification should be in English.
- Options that make the Algorithm not deterministic should be advanced.
- By default the Algorithm should be deterministic.
- When needing randomness use a fixed/default seed, resetting for each invocation.
- Avoid magic values, prefer options with defaults if applicable.

3 Model

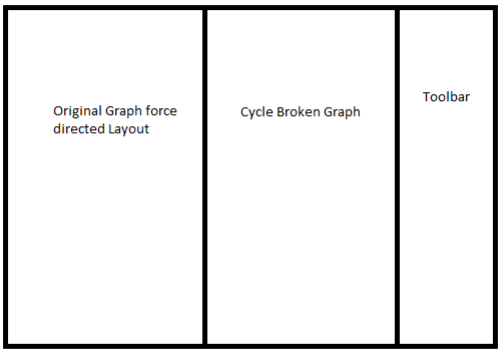
- Reads the graph from a file.
- Performs cycle break (Greedy) (in another thread).
- Can return graph with Property that edges were reversed.
- The state of the Graph per "Step" will be saved .

4 View

The UI will be looking like:



or



- The left Graph will bew showing the original graph (concernig edges).
- The right graph will have it's nodes at the same place as the left graph.
- Nodes will be a default size
- Nodes in both graphs will be coloured dynamically
    - Currently (both) used node will be coloured green
    - Sources (left graph) will be colured (yellow maybe?)
    - Sinks (left graph) will be colured (red maybe?)
- Edges of the current active Node may be coloured (right graph) (in vs. out-degree)
    - Incoming edges are coloured (blue?)
    - Outgoing edges are coloures (purple?)

5 Controller

- A step forward and step backward button
- An option to choose how many steps to go
- An option to choose Level of detail (Maybe with steps?)
- A toggle button that enables autoplay
- Maybe an option to choose how fast autolpay is.
- Maybe a dropdown list with cycle-break algorithms

6 Classes

7 Options

8 Minimal Viable Product

- A Cycle Break algorithm (Greedy)
- Load a graph from a file (In your own input format)
- Present a drawing of the graph
- Be able to configure the verbosity of the animation (step forwards or backwards, run continuously, speed, level of detail, ...)
- Be well-structured, i.e. in terms of separation of concerns (drawing, layout calculation, ...)
- implement your visualization application in Java Swing
- use the ElkGraph as the input format for your visualization.
- comment the code