Subject

We aim to visualize the *Layer Assignment* phase of the *Layered* layout algorithm step by step.

write summarizing stuff here, add source of layered paper

Our Approach

here should be some general information about things we do.

1. first we parse
2. then we apply the algorithm
3. then we visualize

write general information on your topic here, e.g what is you input, what is yout output, any methods you use.. but don't go into detail yet. this will happen in your very own chapter

Approach in detail

write a subchapter about your topic. go into details, if there is something you really want to mention, because its cool or so.. especially try to use stuff from lecture (sources, technical terms, etc.) if you want to give reasons for your solution.
Further: explain the input and output formats of your special topic -> iLearn has some MUSTs listed here, be sure you considered those.

Parsing

Die Parsing-Klasse ist dafür zustädnig das eigens erstelle Textformate einzulesen und in einen ElkGraph umzuwandeln

Syntax

Eine Node hinzufügen:
    node _<node name>

Eine Edge hinzufügen:
    edge_<start node> _<end node>
Wobei _ für beliebig viele Whitespaces steht, weiter muss jeder Knoten der von einer edge benutzt wird vorher mit dem *node* Keyword eingeführt werden

How to use the parser

```
try {
ElkNode testGraph = Parser.parse("testGraphs/testfile.txt");
}
catch (Exception e)
{
e.printStackTrace();
}
```

Example

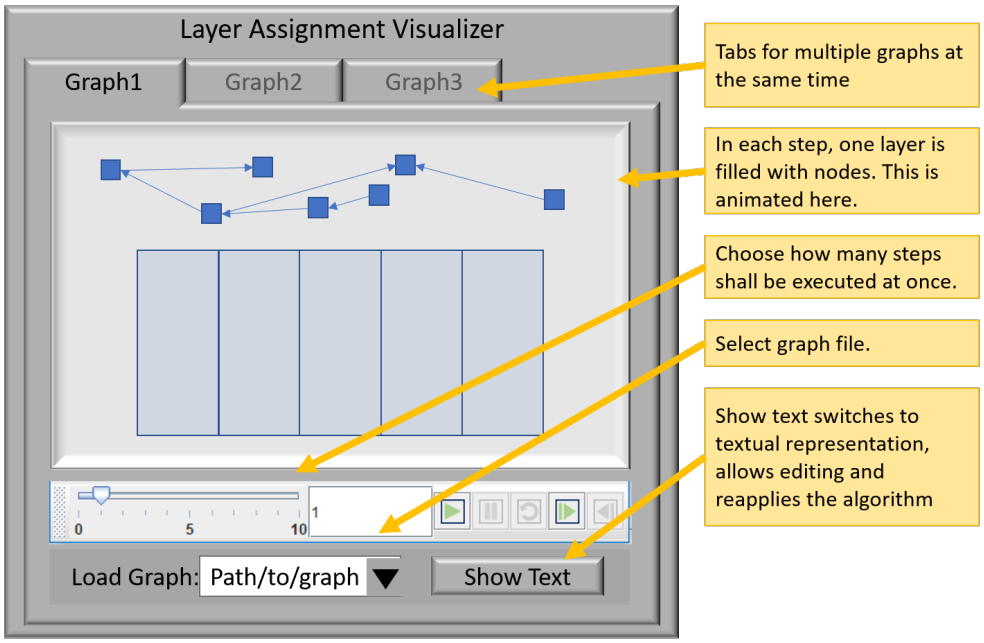
```
node n1
node n2
node n3


edge n1 n3
edge n2 n3
```

Layer Assignment Algorithm

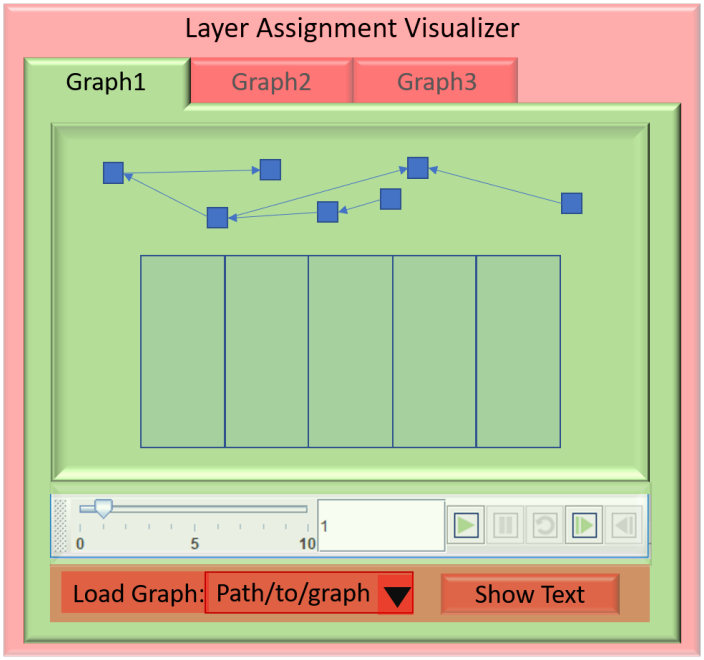
Die Klasse LayerAssignment enthällt eine Reihe hilfreicher Funktionen, die Wichtigste ist allerdings die Methode assignLayers. assignLayers nimmt einen ElkGraphen, führt das Layerassignment an ihm durch und gibt eine Bearbeitungshistorie in Form einer ArrayList vom typen MyGraph zurück. Wobei MyGraph eine von uns entwickelte vereinfachte Graphenstruktur ist. Jeder MyGraph in der Liste stellt also einen ßnapshotäus dem Layerassingments des Graphen dar. Durch Seiteneffekte beinflusst die Methode auch den eingegebenen ElkGraphen, so dass er am Ende der Methode als vollständig gelayerter Graph vorliegt. Dazu haben wir jedem Knoten 3 Propertys gegeben: LAYER: Ein Integer wert größer oder gleich 1, der das Layer angibt, in dem sich der Knoten befindet
IS_DUMMY: Ein boolscher wert, der True ist, wenn es sich bei dem Knoten um einen von uns eingefügten Dummyknoten handelt. Zu beachten ist, dass auch Edges die Property IS_DUMMY haben.
POSITION_IN_LAYER: Gibt an, an welcher Stelle im Layer sich der Knoten befindet, diese Property muss in der Crossingminimisation optimiert werden.
Beim einfügen der Dummyknoten geht der Algorithmus wie folgt vor: Wenn eine Kante k ein Layer übersprngt, fügen wir für jedes Layer zwischen den Layern des Startknotens von k und des Endknotens von k einen neuen Dummyknoten in dem jeweiligen Layer ein. Nun fügen wir Dummykanten so ein, dass von jedem Dummyknoten eine Dummykante zum Dummyknoten im nächsthöheren Layer führt. Desweiteren soll eine Dummykante vom höchsten Dummyknoten zum Zielknotne von k führen. Zuletzt wird noch der Zielknoten von k auf den Dummyknoten im Layer direkt über dem Startknoten von k gesetzt.
(Hier vlt zeichnung)
MyGraph ist eine vereinfachte Graphenstruktur. Sie enthällt nur eine Liste von Knoten und eine Liste von Kanten.

Visualize

In der Visualisierung soll der Algorithmus für das Layer Assignment durch eine Schritt für Schritt durchführbare Animation dargestellt werden. Die Visualisierung soll voraussichtlich so aussehen:



Momentan sind noch nicht alle Graphischen Elemente umgesetzt. Die im folgenden Bild grün markierten Bereiche wurden bereits implementiert, sind jedoch teilweise noch nicht mit Funktionalität ausgestattet. Rote Bereiche wurden noch nicht umgesetzt. **Note: Zum testen nur den Step Forwad Knopf verwenden (step size ist einstellebar). Die anderen Knöpfe funktionieren grade nicht oder nicht richtig.**



Schedule

Date	Thomas	Jonas	Francesca
18.6.18	-	-	Window mockup (no functionality)
18.6.18	Parsing finished (no cycle chek)	-	-
18.6.18	-	First approach for Layer assignment	-
19.6.18	First Deadline/Demo		
30.6.18	-	-	Animations applied to LA algorithm
30.6.18	-	Added some tests	-
30.6.18	Cycle check	-	-
7.7.18	Complete Documentation		
9.7.18	Last Checks?		
10.7.18	Finished		

Daten eintragen

Technical Details

things like class diagrams maybe?