

Systemtest: Graph - Normale Fälle

System unter Test	https://github.com/Graphs-Visualization-Service	+ Bestanden
Package	gvs.testster.graph	- Nicht Bestanden
Datum	04.12.17	

Testergebnis ✔ 75%

Test-Nr.	Test File	Test Beschreibung	Erwartetes Resultat	Ergebnis	Bemerkungen
1	RelativeGraph.java	Darstellung eines Graphen mit relativen Vertices, deren Positionen vom User fixiert sind.	Koordinaten der Vertices werden prozentual interpretiert.	+	
2	DefaultGraph.java	Darstellung eines Graphen mit default Vertices, deren Positionen vom User nicht angegeben werden.	Koordinaten der Vertices werden vom Layouter für den ersten Graphen berechnet. Für weitere Graphen der gleichen Session werden die Koordinaten übernommen.	+	
3	RelativeGraph.java DefaultGraph.java	Darstellung von gerichteten Edges.	Edge wird als Pfeil dargestellt. Der Schnittpunkt des Pfeils mit dem Vertex ist korrekt berechnet und verändert sich beim Dragging des Vertex entsprechend.	+	Es wäre allenfalls etwas schöner (v.a. bei kurzen Edges), wenn das Label mittig zur Linie "ohne Pfeil" wäre
4	RelativeGraph.java DefaultGraph.java	Darstellung von ungerichteten Edges ohne Pfeilangabe.	Edge wird als Linie dargestellt. Der Schnittpunkt der Linie mit dem Vertex ist korrekt berechnet und verändert sich beim Dragging des Vertex entsprechend.	+	
5	RelativeGraph.java DefaultGraph.java	Darstellung von ungerichteten Edges mit Pfeilangabe.	Edge wird als Pfeil dargestellt. Der Schnittpunkt des Pfeils mit dem Vertex ist korrekt berechnet und verändert sich beim Dragging des Vertex entsprechend.	+	
6	IconGraph.java	Darstellung eines Graphen mit Vertices, welche teilweise ein Label mit Icon und teilweise nur ein Icon enthalten.	Korrekte Darstellung des Graphen.	+	
7	RelativeGraph.java DefaultGraph.java	Darstellung einer Abfolge von Graphen mit Styleänderungen.	Korrekte Darstellung der Styles.	+	
8	RelativeGraph.java DefaultGraph.java	Darstellung einer Abfolge von Graphen mit Hinzufügen und Entfernen von Vertices.	Korrekte Darstellung des Graphen. Neu hinzugekommene Vertices werden gelayoutet. Gleichbleibende Vertices übernehmen die Positionen des vorherigen Graphen.	-	Bei DefaultGraphen wird der neue Vertex nicht per physics engine plziert, sondern erscheint bei 0,0 Nach betätigen von Autolayout i.o.
9	RelativeGraph.java DefaultGraph.java	Abspeicherung einer Graph Session.	Die Session wird korrekt gespeichert. Der Speicherort wird vom User gewählt. Die korrekte Dateiendung (*.gvs) wird gesetzt. Die berechneten Koordinaten werden mitgespeichert.	+	
10	relativeGraph.gvs	Laden einer Graph Session	Die Session wird korrekt geladen. Es werden nur Dateien mit der *.gvs Endung angezeigt. Die gespeicherten Koordinaten werden übernommen. Die Graphen können mit AutoLayout neu gelayoutet werden.	+	

11	RelativeGraph.java DefaultGraph.java	Replay einer Graph Session	Die Session wird korrekt abgespielt. Während dem Abspielen wird der Play Button zum Pause Button. Nur der Pause und der Cancel Button sind während dem Abspielen aktiv. Graphen können von jedem Step aus abgespielt werden. Der Step wird beim Pausieren beibehalten. Beim Cancel springt die Session auf den ersten Graphen.	+	
12	RelativeGraph.java DefaultGraph.java	Durchsteppen eines Graphs	Die Step Buttons (First, Prev, Next, Last) navigieren korrekt durch die Session. Bei den ersten und letzten Graphen sind die korrekten Buttons aktiviert/deaktiviert. Die Progressbar zeigt den korrekten Füllstand.	+	
13	RelativeGraph.java DefaultGraph.java	DragSupport eines Graphen	Vertices können herum gezogen werden. Dabei bewegen sich die angehängten Edges entsprechend mit. Beim Ziehen eines Vertex an den rechten oder unteren Rand, wird der gesamte Graph herausgezoomt um Platz zu schaffen.	+	
14	RelativeGraph.java DefaultGraph.java	AutoLayout eines Graphen	Die Koordinaten der Vertices werden neu berechnet. Bei aktivierter Random Layout Option werden die Vertices zufällig oder bei deaktivierter Option immer gleich "zufällig" positioniert. Während dem Layoutprozess sind sämtliche anderen Controls deaktiviert.	-	Nach dem layouten werden die StepButtons nicht korrekt deaktiviert
15	StyleTester.java	Sämtliche unterstützten Styles für Edges und Nodes werden übertragen und dargestellt (Farben, Liniendicke, Linienstyle)	Alle Styles werden wie im Client definiert dargestellt	-	Dicke Linie (FAT) scheint nur für ArrowHeads und nicht für Linien zu funktionieren. Die Edge Label Farbe ändert teilweise nicht deterministisch auf Weiss.
16	-	Autolayout nachdem alle Vertices mit dem Mauszeiger manuell positioniert wurden	Eine entsprechende Meldung erscheint, dass alle Nodes bereits positioniert sind.	-	Führt zu inkorrekt aktivierung der StepButtons

Systemtest: Graph - Spezielle Fälle

System unter Test	https://github.com/Graphs-Visualization-Service	+	Bestanden
Package	gvs.testers.graph	-	Nicht Bestanden
Datum	04.12.17		

Testergebnis ✔ 80%

Test-Nr.	Test File	Test Beschreibung	Erwartetes Resultat	Ergebnis	Bemerkungen
1	LongLabels.java	Darstellung eines Graphen mit langen Vertex Labels.	Die langen Labels werden abgekürzt. Beispiel: LongLabel -> Lo....el	+	
2	mixedGraph.java	Darstellung eines Graphen der Default Vertices und relative Vertices enthält.	?	-	wird vom XML Schema nicht erkannt. Kein Absturz, nur Fehler in Console. User Feedback!
3	emptyGraph.java	Übermittlung eines Graphen der keine Vertices enthält.	Leerer Screen wird dargestellt. Name der Session erscheint im Dropdown Menü.	+	Braucht es hier eine UserInfo à la "empty Graph"?
4	nullEdges.java	Übermittlung eines Graphen der Edges enthält, welche null als Vertex-Start oder Ende besitzen.	Edges sollen nicht angezeigt werden. Es kommt zu keinen Fehlern im GVS 2.0 UI	+	
5	emptyString.java	Darstellung eines Graphen mit Vertices, welche leere Strings als Labels besitzen.	Vertices werden trotzdem angezeigt. Sie besitzen kein Label.	+	Allenfalls könnte man eine minWidth einstellen, dass in so einem Fall schöne Kreise entstehen statt "Eier"

Systemtest: Tree - Normale Fälle

System unter Test	https://github.com/Graphs-Visualization-Service	+ Bestanden
Package	gvs.testee.tree	- Nicht Bestanden
Datum	04.12.17	

Testergebnis  92%

Test-Nr.	Test File	Test Beschreibung	Erwartetes Resultat	Ergebnis	Bemerkungen
1	BinaryTree.java	Darstellung eines Baums mit binären Nodes. Dabei sollen gewisse Nodes nur ein linkes oder nur ein rechtes Kind besitzen.	Platzierung der Nodes wird vom Layouter richtig berechnet. AutoLayout sowie Dragging ist deaktiviert.	+	
2	MultipleRoots.java	Darstellung eines Forests	Die einzelnen Bäume überlappen sich nicht.	+	
3	BinaryTree.java	Darstellung einer Abfolge von Trees mit Hinzufügen und Löschen von Nodes.	Korrekte Darstellung der Session	+	
4	BinaryTree.java	Darstellung einer Abfolge von Trees mit Styleänderungen.	Korrekte Darstellung der Styles. Edges übernehmen immer den Style des Kind Nodes.	+	
5	DefaultTree.java	N-ary Trees werden von GVS 2.0 Lib nicht unterstützt.	Exception im Client	-	Wenn noch Zeit ist könnten wir das ergänzen, Layoutalgorithmus sollte funktionieren. Exception ist eine Cast Exception. Besser wäre hier etwas das klar macht, dass der Fehler mangelnde Implementierung von DefaultTrees ist.
6	BinaryTree.java	Abspeicherung einer Tree Session.	Die Session wird korrekt gespeichert. Der Speicherort wird vom User gewählt. Die korrekte Dateieindung (*.gvs) wird gesetzt.	+	
7	test.gvs	Laden einer Tree Session	Die Session wird korrekt geladen. Es werden nur Dateien mit der *.gvs Endung angezeigt.	+	
8	BinaryTree.java	Replay einer Tree Session	Die Session wird korrekt abgespielt. Während dem Abspielen wird der Play Button zum Pause Button. Nur der Pause und der Cancel Button sind während dem Abspielen aktiv. Graphen können von jedem Step aus abgespielt werden. Der Step wird beim Pausieren beibehalten. Beim Cancel springt die Session auf den ersten Graphen.	+	
9	BinaryTree.java	Durchsteppen eines Trees	Die Step Buttons (First, Prev, Next, Last) navigieren korrekt durch die Session. Bei den ersten und letzten Graphen sind die korrekten Buttons aktiviert/deaktiviert. Die Progressbar zeigt den korrekten Füllstand.	+	
10	BinaryTree.java	DragSupport eines Trees	DragSupport für Trees ist deaktiviert.	+	
11	BinaryTree.java	AutoLayout eines Trees	AutoLayout für Trees ist deaktiviert.	+	
12	StyleTester.java	Sämtliche unterstützten Styles für Edges und Nodes werden übertragen und dargestellt (Farben, Liniendicke, Linienstyle)	Alle Styles werden wie im Client definiert dargestellt	+	
13	ClusterSplitter	Darstellung überfüllter Trees	Baum wird trotz vieler Nodes schön dargestellt.	+	

Systemtest: Tree - Spezielle Fälle

System unter Test	https://github.com/Graphs-Visualization-Service	+ Bestanden
Package	gvs.testers.tree	- Nicht Bestanden
Datum	04.12.17	

Testergebnis

✔ 100%

Test-Nr.	Test File	Test Beschreibung	Erwartetes Resultat	Ergebnis	Bemerkungen
1	LongLabels.java	Darstellung eines Trees mit langen Vertex Labels.	Die langen Labels werden abgekürzt. Beispiel: LongLabel -> Lo....el	+	
2	EmptyString.java	Darstellung eines Trees mit Vertices, welche leere Strings als Labels besitzen.	Nodes werden trotzdem angezeigt. Sie besitzen kein Label.	+	Allenfalls könnte man eine minWidth einstellen, dass in so einem Fall schöne Kreise entstehen statt "Eier"
3	RemoveRoot.java	Darstellung einer Tree Abfolge, wenn die Root entfernt wird. Dabei wird in der GVS 2.0 Lib die Struktur GVSTreeWithRoot benutzt.	Leerer Screen wird dargestellt. Name der Session erscheint im Dropdown Menü.	+	Braucht es hier eine UserInfo à la "empty Graph"?
4	RemoveRootCollection.java	Darstellung einer Tree Abfolge, wenn die Root entfernt wird. Dabei wird in der GVS 2.0 Lib die Struktur GVSTreeWithCollection benutzt.	Leerer Screen wird dargestellt. Name der Session erscheint im Dropdown Menü.	+	Braucht es hier eine UserInfo à la "empty Graph"?
5	CyclicRootCollection.java	Darstellung eines Trees mittels Collection. Die Knoten haben Zyklen untereinander.	Die Zyklen werden Clientseitig erkannt und die Übertragung abgebrochen. Der Server wird freigegeben.	+	

Systemtest: GVS UI Allgemein


System unter Test	https://github.com/Graphs-Visualization-Service	+	Bestanden
Package	gvs.test.*	-	Nicht Bestanden
Datum	04.12.17		

Testergebnis ✔ 100%

Test-Nr.	Test File	Test Beschreibung	Erwartetes Resultat	Ergebnis	Bemerkungen
1	-	Löschen einer Session	Die Session wird nicht mehr angezeigt und ist im Dropdown Menü nicht mehr aufgelistet. Anzeige wechselt zur aktuellsten Session.	+	
2	-	Löschen der letzten Session	Das Fenster wechselt zur Startansicht. Sämtliche Session spezifischen Controls werden ausgeblendet und das Background-Logo wird eingeblendet. Die Speicher und Delete Buttons sind deaktiviert.	+	
3	relativeGraph.gvs	Laden einer bereits geladenen Session	Die Session wird nicht ein zweites Mal geladen. Ist gerade eine andere Session aktiv, wird zur bestehenden Session gewechselt, welche der geladenen Session entspricht.	+	
4	emptySessionName.java	Darstellen einer Session mit leerem Namen	Session wird mit leeren String ins Dropdown aufgenommen. Alles funktioniert wie gewohnt.	+	
5	-	Wechseln der aktuellen Session über die Dropdown Liste	Die ausgewählte Session wird angezeigt	+	

Systemtest: Client - Server Connection

System unter Test	https://github.com/Graphs-Visualization-Service	+	Bestanden
Package	gvs.test.*	-	Nicht Bestanden
Datum	04.12.17		

Testergebnis  100%

Test-Nr.	Test File	Test Beschreibung	Erwartetes Resultat	Ergebnis	Bemerkungen
1	relativeGraph.java	Versuchter Verbindungsaufbau eines zweiten Clients	Die Verbindung mit dem ersten Client wird nicht gestört. Der zweite Client kommt in die Warteschlange.	+	
2	watchdogTest.java	Verbindungsabbruch des Clients (z.B. NullPointerException)	Der reservierte Server wird wieder frei gegeben.	+	