



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Εφαρμοσμένων Μαθηματικών & Φυσικών Επιστημών
Τομέας Μαθηματικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κατσάνου Ελένη

Απεικόνιση Δυναμικών Γραφημάτων

Επιβλέπων: Αντώνιος Συμβώνης,
Καθηγητής, ΕΜΠ



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κατσάνου Ελένη

Απεικόνιση Δυναμικών Γραφημάτων

Επιβλέπων: Αντώνιος Συμβώνης,
Καθηγητής, ΕΜΠ

Τριμελής Επιτροπή

Αντώνιος Συμβώνης
Καθηγητής

Ιωάννης Κολέτσος
Επίκουρος Καθηγητής

Πέτρος Στεφανέας
Επίκουρος Καθηγητής

Αθήνα, Ιούλιος 2021

Περίληψη

Στην Θεωρία Γραφημάτων, τα γραφήματα χρησιμοποιούνται για την μοντελοποίηση δεδομένων που περιγράφουν σχέσεις μεταξύ οντοτήτων, ενώ δυναμικό γράφημα ονομάζεται ένα γράφημα το οποίο μεταβάλλεται με το χρόνο.

Στην παρούσα διπλωματική ασχοληθήκαμε με την ανάπτυξη ενός λογισμικού για την απεικόνιση και την δημιουργία δυναμικών γραφημάτων. Κατά την απεικόνιση, ο χρήστης έχει την δυνατότητα να παρακολουθήσει, σε μορφή βίντεο, την «πορεία» του γραφηματος στο χρόνο. Οι αλλαγές που μπορεί να γίνονται σε ένα δυναμικό γράφημα αφορούν προσθήκες/ διαγραφές κόμβων και ακμών, καθώς και αλλαγές βασικών τους χαρακτηριστικών (σχήμα, χρώμα, τοποθεσία, κλπ).

Τέλος, ο χρήστης μπορεί να δημιουργήσει ένα καινούριο δυναμικό γράφημα, να εισάγει σ' αυτό κόμβους και ακμές, αλλά και να τροποποιήσει τις ιδιότητες των κόμβων και των ακμών γραφικά μέσω της μπάρας του χρόνου.

Λέξεις Κλειδιά.

Θεωρία Γραφημάτων, Δυναμικό Γράφημα, Απεικόνιση, Επεξεργασία Γραφημάτων, Δημιουργία Γραφημάτων

Abstract

In Graph Theory, graphs are used to model data that describe relationships between entities, whilst dynamic (or temporal) graphs are graphs that change over time.

In this Diploma Thesis, we dealt with the development of a software for the visualization and creation of dynamic graphs. As for the visualization, the user can watch a video that displays the “course” of the dynamic graph over time. The changes that take place can be about the addition/deletion of nodes and edges, as well as alterations to their basic attributes (shape, color, position, etc).

Finally, the user can create a new dynamic graph, insert nodes and edges into it, but also edit the basic properties of nodes and edges via the time bar.

Keywords. Graph Theory, Dynamic Graph, Temporal Graph, Visualization, Graph Editing, Graph Creation

Ευχαριστίες

Πρώτα απ' όλα, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθήγητη της Διπλωματικής μου Εργασίας κ. Αντώνιο Συμβώνη, για την συνεχή επίβλεψη και καθοδήγησή του καθ' όλη την πορεία της εργασίας, αλλά και για τις συμβουλές του για τα επόμενα βήματά μου. Επίσης, θα ήθελα να ευχαριστήσω την Διδάκτορα Χρυσάνθη Ραυτοπούλου και την Επίκουρη Καθηγήτρια του Πανεπιστημίου της Ουτρέχτης Tamara Mchedlidze για την συνεχή και πολύτιμη βοήθεια τους στην εκπόνηση της Διπλωματικής μου Εργασίας. Στη Συνέχεια, θα ήθελα να ευχαριστήσω τα μέλη της επιτροπής, κ. Ιωάννη Κολέτσο, Επίκουρο Καθηγητή Ε.Μ.Π. και κ. Πέτρο Στεφανέα, Επίκουρο Καθηγητή Ε.Μ.Π. για την συμβολή τους στην Διπλωματική Εργασία.

Τέλος, θα ήθελα ιδιαίτερα να ευχαριστήσω την οικογένεια μου για την συμπαράσταση και την βοήθεια τους, αλλά και τους φίλους μου με τους οποίους πέρασα ωραίες στιγμές όλα αυτά τα χρόνια.

.....
Ελένη Κατσάνου

© (2021) Εθνικό Μετσόβιο Πολυτεχνείο. All rights Reserved. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σ' αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περιεχόμενα

Περίληψη	1
Abstract	2
Ευχαριστίες	3
1 Εισαγωγή	6
1.1 Βασικές έννοιες	6
1.2 Περιγραφή της Διπλωματικής	7
1.3 Τεχνολογικό Υπόβαθρο	9
1.4 Εφαρμογές απεικόνισης δυναμικών γραφημάτων	10
1.4.1 Διαφορές με την παρούσα Διπλωματική Εργασία	14
2 Γενικά Χαρακτηριστικά της Εφαρμογής	16
2.1 Αναπαράσταση Χρόνου	16
2.1.1 Αναπαράσταση χρόνου στο UI	17
2.2 Ιδιότητες Αντικειμένων (Κόμβων και Αχμών)	18
2.3 Το Γράφημα	22
2.4 Δεδομένα- Format *.grdyn	23
2.5 Στοιχεία του UI	30
2.5.1 Απεικόνιση του γραφήματος	32
2.5.2 Παράθυρο με ιδιότητες των αντικειμένων	35
2.5.3 Overview παράθυρο	37
3 Απεικόνιση δυναμικών γραφημάτων	39
3.1 Μετάβαση Γραφήματος	41
3.1.1 Λειτουργίες Βίντεο	44
3.2 Ομαλή μεταβολή των αντικειμένων του γραφήματος	44
3.3 Πυκνό χρονικό εύρος- Zoom Panel	49
3.4 Συγχρονισμός κινήσεων- Multithreading	51
4 Δημιουργία δυναμικών γραφημάτων (Dynamic Graph Editing)	54
4.1 Προσθήκη Κόμβου και Αχμής	54
4.1.1 Αλλαγή προεπιλεγμένων ρυθμίσεων	55
4.1.2 Τύποι Αντικειμένων	57
4.2 Επεξεργασία Κόμβου και Αχμής	61
5 Εφαρμογή: Hrafnkels Saga	76

6 Επίλογος	80
6.1 Σύνοψη	80
6.2 Συμπεράσματα	83
6.3 Μελλοντικές Επεκτάσεις	83
Βιβλιογραφία	85

1 Εισαγωγή

1.1 Βασικές έννοιες

Η θεωρία γραφημάτων είναι κλάδος των μαθηματικών και της θεωρητικής πληροφορικής και ασχολείται με τη μελέτη και την απεικόνιση γραφημάτων (ή γράφων). Τα γραφήματα χρησιμοποιούνται για την μοντελοποίηση δεδομένων που περιγράφουν σχέσεις μεταξύ οντοτήτων και σκόπος τους είναι να καταστήσουν τα δεδομένα αυτά πιο ευανάγνωστα και πιο εύκολα αξιοποιήσιμα. Ένα γράφημα, λοιπόν, αποτελείται από κόμβους, οι οποίοι αναπαριστούν τις οντότητες, και από ακμές, οι οποίες αναπαριστούν σχέσεις μεταξύ των κόμβων. Επιστημονικοί κλάδοι που χρησιμοποιούν τα γραφήματα είναι μεταξύ άλλων η χημεία και η φυσική, οι ανθρωπιστικές επιστήμες, η βιολογία κα. Ένα γράφημα, λοιπόν ορίζεται ως εξής:

Ορισμός 1.1. Ένα (Στατικό) Γράφημα $G := (V, E)$ αποτελείται από ένα πεπερασμένο σύνολο κόμβων V και ένα πεπερασμένο σύνολο ακμών $E \subseteq V \times V$. Συμβολίζουμε μια ακμή μεταξύ των κόμβων u και v του G με (u, v) .

Σε αυτή τη διπλωματική εργασία αντικείμενο μας είναι τα δυνάμικα γραφήματα. Ένα δυναμικό γράφημα είναι ένα γράφημα, δηλαδή αποτελείται από ένα σύνολο κόμβων V και ένα σύνολο ακμών E , το οποίο όμως μεταβάλλεται με το χρόνο. Σε βάθος χρόνου μπορεί να προστίθενται κόμβοι και ακμές, ενώ παράλληλα μπορούν να μεταβάλλονται οι ιδιότητες τους, όπως το χρώμα, το πάχος, το σχήμα κλπ. Ένα δυναμικό γράφημα λοιπόν ορίζεται ως εξής:

Ορισμός 1.2. Ένα Δυναμικό Γράφημα $G := (V, E, \tau)$ είναι ένα γράφημα το οποίο μεταβάλλεται σε σχέση με το χρόνο, όπου $\tau = (t_1, t_2, \dots, t_n)$ το σύνολο των χρονικών στιγμών του G . Οι προβολές του G είναι μια ακολουθία από στατικά γραφήματα $\Gamma := (G_1, G_2, \dots, G_n)$, όπου $G_i := (V_i, E_i)$ η προβολή του G την χρονική στιγμή t_i με $V_i \subseteq V$ και $E_i \subseteq E$.

Γενικά, δεν υπάρχει μόνο ένας τρόπος να αναπαραστήσει κάποιος ένα (στατικό) γράφημα. Οι δύο συνηθέστεροι τρόποι απεικόνισης γραφήματος είναι με πίνακα γειτνίασης και με node-link τρόπο [6]. Ο πίνακας γειτνίασης, είναι ένας πίνακας όπου στις γραμμές και στις στήλες του αντιστοιχούν όλοι οι κόμβοι του γραφήματος. Αν υπάρχει η ακμή (u_i, u_j) , όπου u_i, u_j κόμβοι, τότε το κελί i, j έχει 1, αλλιώς 0. Αν οι κόμβοι u_i, u_j συνδέονται με πολλαπλές ακμές, τότε το κελί (u_i, u_j) περιέχει το πλήθος των ακμών ανάμεσα στους δύο κόμβους. Αν το γράφημα είναι μη κατευθυνόμενο, τότε ο πίνακας γειτνίασης είναι συμμετρικός. Απ' την άλλη, μέσω node-link τρόπου, οι κόμβοι αναπαραστώνται γραφικά ως σχήματα και οι ακμές ως ευθύγραμμα τμήματα μεταξύ των δύο

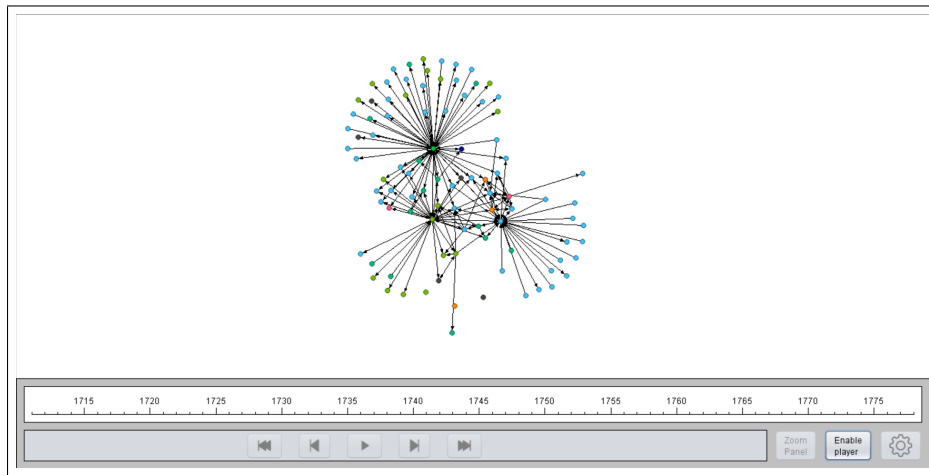
κόμβων. Οι κατευθυνόμενες ακμές, αναπαριστούνται με ευθύγραμμα τμήματα, όπου ένα βέλος δείχνει την κατεύθυνση της ακμής. Στο λογισμικό, τα γραφήματα έχουν επιλεχθεί να αναπαραστηθούν με node-link τρόπο και όχι με πίνακα γειτνίασης.

Όσον αφορά, τα δυναμικά γραφήματα, και πάλι, δεν υπάρχει μόνο ένας τρόπος απεικόνισης. Οι τρόποι που συναντιούνται πιο συχνά στη βιβλιογραφία [6] είναι δύο: time-to-space mapping και time-to-time mapping. Στον πρώτο τρόπο, οι προβολές του δυναμικού γραφήματος (G_1, G_2, \dots, G_n) των χρονικών στιγμών (t_1, t_2, \dots, t_n) απεικονίζονται διαδοχικά στο χώρο, π.χ. η μία δίπλα στην άλλη ή η μία κάτω από την άλλη. Στον δεύτερο τρόπο, τα γραφήματα «απλώνονται» στο χρόνο, δηλαδή τα (G_1, G_2, \dots, G_n) εμφανίζονται διαδοχικά το ένα μετά το άλλο, με κάποια χρονική καθυστέρηση ανάμεσα (animation). Στο πρόγραμμα, επιλέχθηκε ο δεύτερος τρόπος, δηλαδή να αναπαραστηθεί το δυναμικό γράφημα με time-to-time mapping.

1.2 Περιγραφή της Διπλωματικής

Αντικείμενο της παρούσας διπλωματικής είναι η υλοποίηση ενός λογισμικού για την απεικόνιση και την επεξεργασία δυναμικών γραφημάτων. Κατά την πλοήγηση στο λογισμικό, ο χρήστης έχει τη δυνατότητα είτε να δημιουργήσει ένα δυναμικό γράφημα από την αρχή είτε να εισάγει δεδομένα που αφορούν αντικείμενα ενός δυναμικού γραφήματος. Οι κόμβοι και οι ακμές έχουν ένα μοναδικό id, το οποίο, αν πρόκειται για εισαγωγή δεδομένων δίνεται από τον χρήστη μέσω των δεδομένων που εισάγει, ενώ αν πρόκειται για δημιουργία γραφήματος από την αρχή, το id δίνεται στο αντικείμενο από το πρόγραμμα. Επίσης, οι ακμές έχουν αφετηρία και τέρμα (Source Node και Target Node). Όλες οι ιδιότητες των κόμβων και των ακμών που αφορούν την απεικόνιση του δυναμικού γραφήματος είναι δυναμικής φύσης, δηλαδή μπορούν να αλλάζουν στο χρόνο. Τέτοιες ιδιότητες είναι το εύρος ζωής τους, το πότε είναι ορατά και πότε όχι, το χρώμα και το όνομα τους, ενώ για τους κόμβους δυναμικές ιδιότητες είναι, επιπλέον, το σχήμα, η θέση και η φωτογραφία τους και για τις ακμές είναι το πάχος, το είδος τους και το αν είναι ορατό το βέλος που υποδεικνύει την κατεύθυνση των κατευθυνόμενων ακμών.

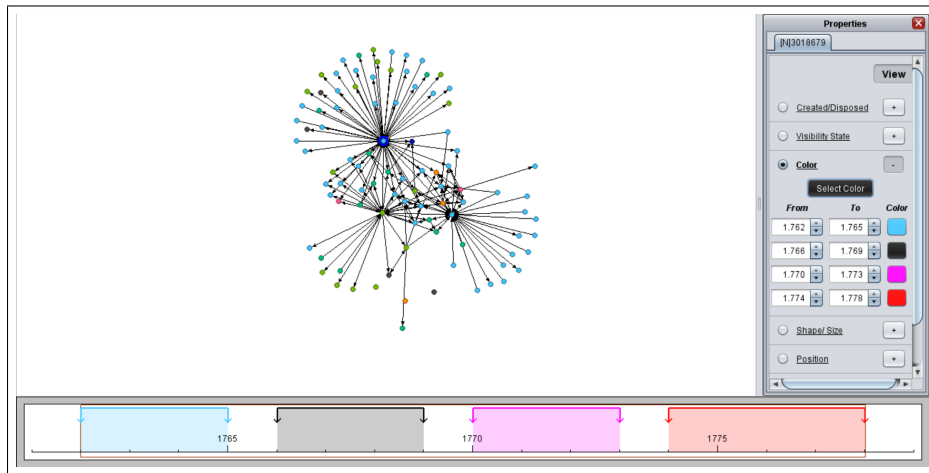
Η Διεπαφή Χρήστη (UI) του λογισμικού είναι χωρισμένη σε δύο μέρη, στο πάνω και στο κάτω. Το κάτω μέρος του UI αφορά την μπάρα του χρόνου και λειτουργίες του λογισμικού συσχετιζόμενες με την δυναμική απεικόνιση, ενώ το πάνω μέρος αφορά τον «καμβά», όπου ζωγραφίζεται το γράφημα. Τέλος, στο πάνω μέρος του UI αριστερά και δεξιά του γραφήματος ανοίγουν κάποια παράθυρα που περιέχουν πληροφορίες ή ρυθμίσεις και θα περιγραφούν εκτενέστερα παρακάτω.



Σχήμα 1: Η αναπαράσταση ενός γραφήματος στο λογισμικό

Η απεικόνιση του δυναμικού γραφήματος γίνεται με τη μορφή «βίντεο», το οποίο αποτελείται από τις προβολές του δυναμικού γραφήματος σε διαδοχικές χρονικές στιγμές. Διαθέσιμες στο λογισμικό είναι λειτουργίες βίντεο, όπως τα play/ pause, previous/ next, start/ end. Με το play/ pause ο χρήστης μπορεί να κάνει παύση του βίντεο σε κάποια χρονική στιγμή ή να το ξεκινήσει, με το previous/ pause απεικονίζεται στον κάμβά η προβολή του δυναμικού γραφήματος για την προηγούμενη ή την επόμενη χρονική στιγμή από την τρέχουσα, ενώ με το start/ end το βίντεο πηγαίνει στην πρώτη ή στην τελευταία στιγμή αντίστοιχα. Για την μετάβαση του δυναμικού γραφήματος από τη μια χρονική στιγμή στην επόμενη υπάρχει η επιλογή να γίνεται ομαλά, όπου τα αντικείμενα μεταβάλλονται βαθμιαία σε ένα χρονικό διάστημα ώσπου να φτάσουν στην τελική τους μορφή, ή να γίνεται απότομα, όπου το βίντεο μοιάζει με εναλλαγή φωτογραφιών, δηλαδή με εναλλαγή των προβολών του δυναμικού γραφήματος με μία καθυστέρηση ανάμεσα.

Κατά τη λειτουργία της επεξεργασίας ο χρήστης έχει τη δυνατότητα είτε να επεξεργαστεί ολόκληρο το γράφημα G , όπου εμφανίζονται όλοι οι κόμβοι και όλες οι ακμές των συνόλων V και E , είτε να επεξεργαστεί το γράφημα κάποια συγκεκριμένη χρονική στιγμή t_i , αλλά και να δει πληροφορίες για το αντικείμενο που ενδιαφέρεται οποιαδήποτε στιγμή. Η επεξεργασία αυτή αφορά τις δυναμικές ιδιότητες των αντικειμένων, όπου ο χρήστης μπορεί να μεγαλώσει ή να μικρύνει το εύρος ζωής τους, να επιλέξει ποιες χρονικές στιγμές γίνονται ορατά και ποιες εξαφανίζονται, να επεξεργαστεί σε όλο το χρονικό εύρος ζωής τους το χρώμα και το όνομα τους, αλλά και το σχήμα, τη θέση και τη φωτογραφία των κόμβων ή το πάχος, το είδος και το βέλος των ακμών. Η



Σχήμα 2: Η επεξεργασία του χρώματος ενός κόμβου.

επεξεργασίας μιας ιδιότητας στο λογισμικό γίνεται διαδραστικά μέσω της μπάρας του χρόνου και ο χρήστης μπορεί να επεξεργαστεί το διάστημα στο οποίο ισχύει μια τιμή ή να προσθέσει μια καινούρια τιμή για την ιδιότητα.

Στόχος της Διπλωματικής Εργασίας ήταν, απ' τη μία η επεξεργασία του δυναμικού γραφήματος να είναι εύχρηστη και να μπορεί ο χρήστης να μετατρέψει και να κατανοήσει τα δεδομένα με γραφικό και εύκολο τρόπο, και απ' την άλλη η δυναμική απεικόνιση να κάνει ομαλή την μετάβαση του γραφήματος από τη μία στιγμή στην επόμενη και, όπου επιλέγεται, να επισημαίνει τις αλλαγές που πραγματοποιούνται, χωρίς να υπερφορτώνει τον χρήστη με πληροφορίες.

1.3 Τεχνολογικό Υπόβαθρο

Στο κεφάλαιο αυτό περιγράφονται οι κυριότερες τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του λογισμικού.

Γλώσσα προγραμματισμού και προγραμματιστικό περιβάλλον: Το προγραμματιστικό περιβάλλον που χρησιμοποιήθηκε ήταν το Eclipse IDE και το πρόγραμμα γράφτηκε σε Java, στην έκδοση 1.8. Η Java είναι μια γλώσσα προγραμματισμού, η οποία βασίζεται σε χρήση αντικειμένων, είναι δηλαδή μια αντικειμενοστραφής γλώσσα προγραμματισμού.

OpenGL: Για τα γραφικά της εφαρμογής, δηλαδή για την απεικόνιση των κόμβων και

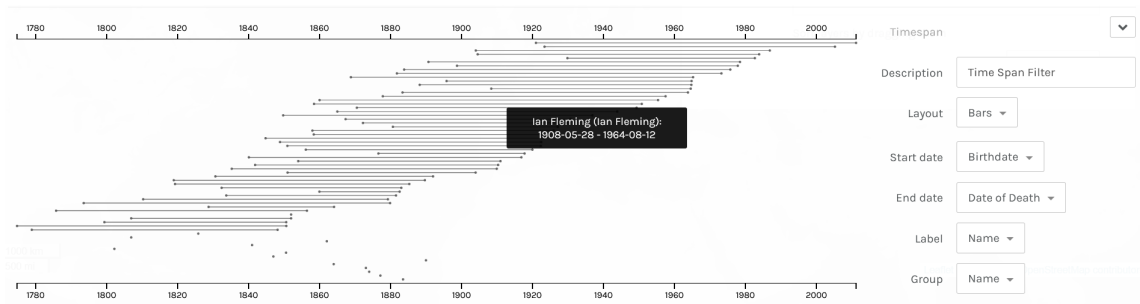
των ακμών, χρησιμοποιήθηκε το Java OpenGL 2. Το Java OpenGL είναι ένα API για την απόδοση 2D και 3D γραφικών και είναι μια «μετάφραση» του OpenGL που επιτρέπει την χρήση του σε Java, καθώς αρχικά είναι γραμμένο σε C++. Η χρήση του OpenGL προτιμήθηκε της κλάσης Swing της Java λόγω ταχύτητας. Για πείραμα, εισήχθη ένα δυναμικό γράφημα- κλίμα στο πρόγραμμα με εύρος 20 χρονικών στιγμών, το οποίο αποτελούνταν από 100 κόμβους και 100*99 ακμές, άρα από 10000 αντικείμενα. Στις 20 αυτές χρονικές στιγμές οι κόμβοι άλλαζαν χρώμα και θέση, ενώ οι ακμές άλλαζαν χρώμα, πάχος και είδος. Παρατηρήθηκε, λοιπόν, ότι το πρόγραμμα με τη χρήση του OpenGL απεικόνιζε το δυναμικό γράφημα κανονικά, ίσως μερικές φορές με μικρές καθυστερήσεις, ενώ με την χρήση της κλάσης Swing, πάγωνε και εν τέλει χρειαζόταν να τερματιστεί.

XML: Τέλος, δημιουργήθηκε ένα καινούριο format αρχείου για την εισαγωγή δεδομένων στο πρόγραμμα. Το αρχείο αυτό είναι γραμμένο στη γλώσσα XML [18]. Η γλώσσα XML είναι μια Επεκτάσιμη γλώσσα σήμανσης (Extensible Markup Language), η οποία χρησιμοποιείται για να περιγράψει δεδομένα. Είναι παρόμοια με την γλώσσα προγραμματισμού HTML, διότι και οι δύο γλώσσες χρησιμοποιούν ετικέτες (tags). Η διαφορά είναι ότι στην HTML οι ετικέτες έχουν πάντα το ίδιο νόημα και οι εντολές εκτελούνται πάντα με τον ίδιο τρόπο, ενώ η XML έχει μεγαλύτερη ελευθερία. Για παράδειγμα, η ετικέτα <table> στην HTML συμβολίζει πάντα έναν πίνακα. Η XML, λοιπόν, προσφέρει την ελευθερία να καθοριστούν και να αποκτήσουν νόημα οι ετικέτες από την αρχή.

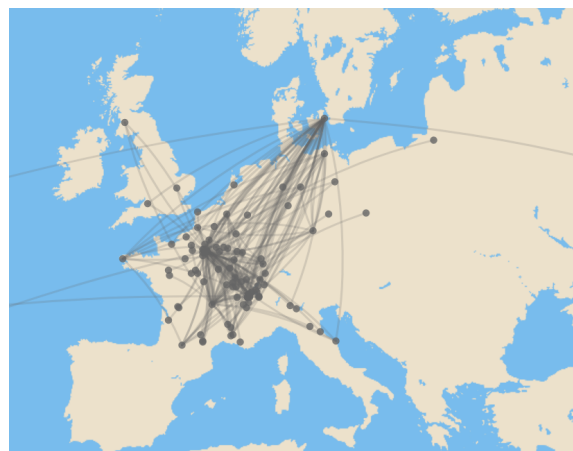
1.4 Εφαρμογές απεικόνισης δυναμικών γραφημάτων

Παρακάτω περιγράφονται κάποιες άλλες εφαρμογές για την απεικόνιση γραφημάτων:

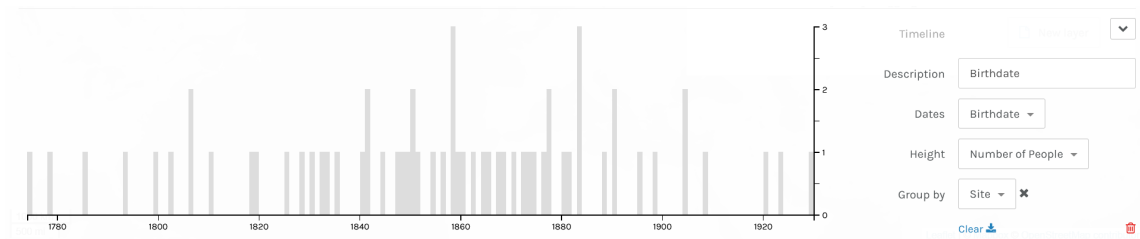
Palladio: Το Palladio [15] είναι ένα διαδικτυακό εργαλείο απεικόνισης γραφημάτων, σχεδιασμένο από το Humanities + Design Lab του πανεπιστημίου του Stanford. Προορίζεται περισσότερο για δεδομένα στις ανθρωπιστικές επιστήμες, και έχει τη δυνατότητα να απεικονίσει στον παγκόσμιο χάρτη γραφήματα από δεδομένα που αφορούν γεωγραφικές συντεταγμένες. Επίσης, δίνει την δυνατότητα στον χρήστη να φιλτράρει τα δεδομένα του και να δει ένα υπογράφημα, το οποίο περιλαμβάνει κόμβους και ακμές που πληρούν τις προϋποθέσεις που εκείνος έχει θέσει. Όσον αφορά τις δυναμικές πληροφορίες ενός γραφήματος, μπορεί κάποιος να δει διαγράμματα, όπως στα σχήματα 3 και 5 .



Σχήμα 3: Διάγραμμα όπου κάθε ευθεία απεικονίζει τη ζωή ενός ανθρώπου. Για παράδειγμα ο Ian Flemming έζησε από το 1908 μέχρι το 1964.



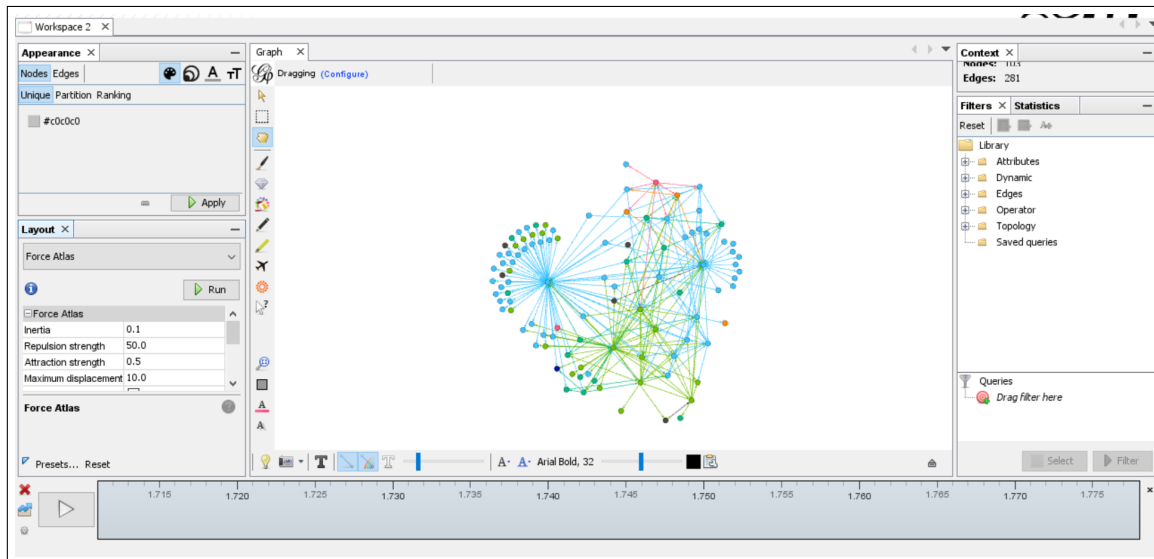
Σχήμα 4: Παράδειγμα γραφήματος στο Palladio



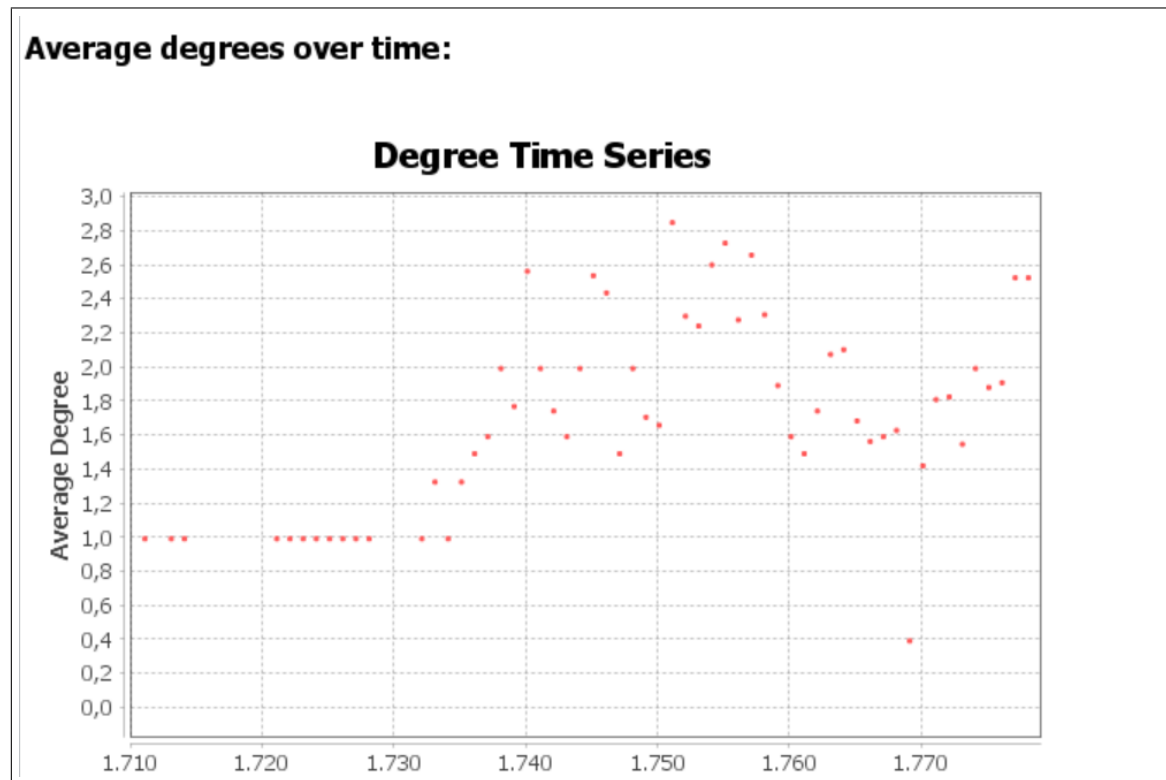
Σχήμα 5: Διάγραμμα που δείχνει αριθμό γεννήσεων ανα χρονιά

Gephi: Μια παρόμοια εφαρμογή με το λογισμικό που περιγράφεται σ' αυτή την διπλωματική είναι το Gephi [5]. Το Gephi προορίζεται για την απεικόνιση και την επεξεργασία στατικών και δυναμικών γραφημάτων. Όσον αφορά τα δυναμικά γραφήματα, η αναπαράσταση του χρόνου γίνεται με δύο τρόπους: με διαστήματα ή με διακριτές χρονικές

στιγμές. Η απεικόνιση των δυναμικών γραφημάτων μοιάζει με βίντεο, ενώ ένας δρομέας δείχνει το χρονικό διάστημα στο οποίο αντιστοιχεί το τρέχον γράφημα. Επίσης, μπορεί να εξάγει γραφήματα (διαγράμματα) με ιδιότητες του γραφήματος σε βάθος χρόνου, όπως πχ ο αριθμός των κόμβων ανά χρονική στιγμή, ο μέσος όρος των βαθμών των κόμβων κ.α. Άλλες λειτουργίες του Gephi είναι η εφαρμογή Layout αλγορίθμων στο γράφημα, φιλτράρισμα με βάση ιδιότητες των κόμβων και των ακμών, χρωματισμός των αντικειμένων του γραφήματος με βάση κάποια ιδιότητα (π.χ. ο βαθμός των κόμβων) κλπ.



Σχήμα 6: Το UI του Gephi



Σχήμα 7: Το διάγραμμα ενός δυναμικού γραφήματος για τον μέσο βαθμό των κόμβων κάθε στιγμή στο Gephi

GraphDiaries: Ένα άλλο λογισμικό το οποίο περιγράφεται από τους Benjamin Bach, Emmanuel Pietriga και Jean-Daniel Fekete είναι το GraphDiaries [4]. Το συγκεκριμένο λογισμικό εστιάζει στην απεικόνιση δυναμικών γραφημάτων και, πιο συγκεκριμένα, στην μετάβαση του δυναμικού γραφήματος από τη μία χρονική στιγμή στην επόμενη. Το GraphDiaries προτείνει η μετάβαση του γραφήματος G_i στο γράφημα G_{i+1} να γίνεται με τον εξής τρόπο: πρώτα αποχρωματίζονται ομαλά (fade-out) οι κόμβοι του γραφήματος G_i που δεν υπάρχουν στο γράφημα G_{i+1} , μετά οι εναπομείναντες κόμβοι ανατοποθετούνται με Layout αλγόριθμο και τέλος χρωματίζονται ομαλά (fade-in) οι «καινούριοι» κόμβοι του γραφήματος G_{i+1} .

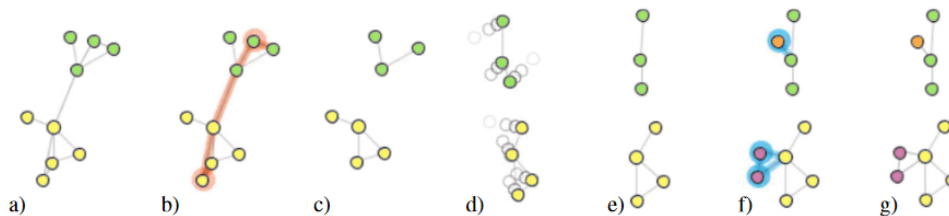


Fig. 1. Staged transitions with change highlighting (node fill colors describe arbitrary, domain-specific, attributes of those nodes): a) initial state, b) element removal (red halos), c) remaining elements only, d) layout adaption, e) remaining elements at their new position, f) element addition (blue halos), and g) final state.

1.4.1 Διαφορές με την παρούσα Διπλωματική Εργασία

- Το Gephi, παρέχει τη δυνατότητα να προσθέσει κάποιος λειτουργίες στο πρόγραμμα μέσω plugins. Μια αρχική σκέψη λοιπόν, ήταν, αν θα μπορούσαν οι λειτουργίες του λογισμικού, που περιγράφεται σ' αυτή τη διπλωματική, να υλοποιηθούν μέσω plugins και το τέλικό συμπέρασμα ήταν ότι δεν μπορούσαν να υλοποιηθούν όλες. Οι λειτουργίες, λοιπόν, που ήταν οι πιο δύσκολες, και εν τέλει αποτρεπτικές, ήταν οι λειτουργίες που αφορούν την μετάβαση του δυναμικού γραφήματος από τη μία χρονική στιγμή στην επόμενη. Πιο συγκεκριμένα, το Gephi υλοποιεί την απεικόνιση αποκλειστικά μέσω των δεδομένων του κόμβου, κάτι το οποίο αποβαίνει αποτρεπτικό, διότι αν θέλει κάποιος να απεικονίσει μια προσωρινή κατάσταση, δηλαδή μια μετάβαση με ομαλό τρόπο, θα πρέπει να τροποποιήσει τα δεδομένα του χρήστη, κάτι το οποίο δεν ήταν επιθυμητό. Για παράδειγμα, η διαγραφή ενός κόμβου μπορεί να απεικονιστεί με fade-out του χρώματος του, κάτι το οποίο στο Gephi θα σήμαινε προσθήκη των ενδιαμέσων χρωμάτων στα δεδομένα του χρήστη. Η επιθυμητή υλοποίηση, όμως, ήταν αυτή η μετάβαση του γραφήματος να γίνεται «εσωτερικά» και να μην επηρεάζονται τα δεδομένα του χρήστη.
- Όπως προαναφέρθηκε, στο GraphDiaries η μετάβαση του δυναμικού γραφήματος γίνεται με fade-out και fade-in στο χρώμα των κόμβων και των ακμών που αφαιρούνται και προστίθενται από μια χρονική στιγμή στην επόμενη. Στο λογισμικό που περιγράφεται στην παρούσα Διπλωματική Εργασία, ο χρήστης μπορεί να επιλέξει για ποιες ιδιότητες των αντικειμένων η μετάβαση θα γίνει ομαλά και για ποιες όχι (βλ. κεφάλαιο 3.2). Εκτός από την αφαίρεση και την προσθήκη κόμβων και ακμών, δηλαδή, κατά τη μετάβαση του δυναμικού γραφήματος μπορεί να μεταβάλλεται το χρώμα των αντικειμένων, το σχήμα, η θέση και η φωτογραφία των κόμβων ή το πάχος των ακμών. Όταν μεταβάλλεται κάποια ιδιότητα των αντικειμένων, εκτός από την θέση των κόμβων, τα αντικείμενα περιβάλλονται από ένα φωτιστέφανο (halo) διαφορετικού χρώματος για κάθε ιδιότητα, ώστε να

γίνει αντιληπτή η μετάβαση από τον χρήστη. Όσον αφορά τη θέση των κόμβων, η μετάβαση γίνεται μέσω μιας ομαλής κίνησης του κόμβου από το ένα σημείο στο άλλο.

- Μια λειτουργία για την εισαγωγή αντικειμένων στο δυναμικό γράφημα που δεν συναντάει κανείς σε άλλες εφαρμογές είναι η «παλέτα» αντικειμένων (κεφάλαιο 4.1.2). Την παλέτα αυτή μπορεί κάποιος είτε να την δημιουργήσει μέσω του προγράμματος είτε να την εισάγει έτοιμη και αποτελείται από τύπους κόμβων και ακμών. Κάθε τύπος αντικειμένου, είναι στην πραγματικότητα ένα «σετ» ρυθμίσεων για τις ιδιότητες ενός αντικειμένου και χρησιμοποιείται για την γρήγορη εισαγωγή κόμβων και ακμών με όμοιες ιδιότητες. Για παράδειγμα, αν τα δεδομένα του χρήστη αφορούν ανθρώπους και ο χρήστης θέλει να αναπαραστήσει τις γυναίκες με πορτοκαλί κύκλο και τους άντρες με πράσινο τετράγωνο, αρκεί να δημιουργήσει δύο τύπους κόμβου Woman και Man με αυτές τις ιδιότητες ο καθένας και να επιλέγει τον αντίστοιχο τύπο για κάθε εισαγωγή.
- Τέλος, σε αντίθεση με άλλα προγράμματα, στο λογισμικό έχουμε επιλέξει όλες οι ιδιότητες που αφορούν την απεικόνιση των αντικειμένων να είναι δυναμικής φύσης, δηλαδή να μπορούν να αλλάζουν σε βάθος χρόνου, κάτι που δίνει στον χρήστη την ελευθερία να αναπαραστήσει τις μεταβολές στα δεδομένα του με δικό του τρόπο. Επιπλέον, η εισαγωγή και η επεξεργασία κόμβων και ακμών γίνεται διαδραστικά, με ευκολό και εύχρηστο τρόπο, χωρίς να χρειάζεται να επεξεργαστεί ο χρήστης τις ιδιότητες των αντικειμένων «χειροκίνητα», δηλαδή να πληκτρολογήσει κάπου όλες τις τιμές που παίρνει μια ιδιότητα σε όλο το εύρος ζωής του αντικειμένου.

2 Γενικά Χαρακτηριστικά της Εφαρμογής

Σε αυτό το κεφάλαιο θα περιγραφούν μέρη του λογισμικού που δεν έχουν να κάνουν με την επεξεργασία ή την απεικόνιση δυναμικών γραφημάτων. Το πρώτο μέρος αυτού του κεφαλαίου αφορά τον τρόπο που επιλέχθηκε να αναπαραστηθεί ο χρόνος στο πρόγραμμα, δηλαδή τι θεωρείται χρονική στιγμή, ποια είναι η εσωτερική αναπαράσταση της χρονικής στιγμής, κοκ. Το δεύτερο αφορά τα αντικείμενα ενός δυναμικού γραφήματος, δηλαδή τους κόμβους και τις ακμές, και ποιες είναι οι ιδιότητες που μπορούν να έχουν αυτά, αλλά και πώς επηρεάζονται σε σχέση με τον χρόνο. Στο τρίτο μέρος δίνονται πληροφορίες για το δυναμικό γράφημα γενικά, δηλαδή το πώς αποθηκεύονται τα αντικείμενα κλπ. Στο τέταρτο μέρος θα αναφερθεί ο τρόπος που μπορεί να εισάγει κάποιος δεδομένα στο πρόγραμμα, αλλά και ποια δομή θα πρέπει να ακολουθούν αυτά τα δεδομένα. Τέλος, περιγράφονται μέρη του λογισμικού που αφορούν την διέπαψη του χρήστη (UI).

2.1 Αναπαράσταση Χρόνου

Η αναπαράσταση του χρόνου στο πρόγραμμα είναι διακριτής φύσης και αφορά ένα πεπερασμένο σύνολο από χρονικές στιγμές. Για τα δεδομένα υπάρχουν δύο επιλογές: μπορούν να δίνονται είτε σε **ακέραιους αριθμούς**, είτε σε **ημερομηνίες**. Για την πρώτη περίπτωση ο κάθε ακέραιος αριθμός θεωρείται μια διακριτή χρονική στιγμή, ενώ δεν είναι απαραίτητο ο πρώτος ακέραιος να είναι το μηδέν ή το ένα ή κάποιος θετικός αριθμός, αλλά ορίζεται από τα δεδομένα που δίνει ο χρήστης. Αν, για παράδειγμα, στα δεδομένα υπάρχουν οι ακέραιοι $\{-3, \dots, 15\}$, τότε η πρώτη χρονική στιγμή του γραφήματος είναι το -3 , η δεύτερη το -2 κλπ.

Από την άλλη, στην περίπτωση των ημερομηνιών, σαν χρονική στιγμή θεωρείται μία ημερολογιακή ημέρα. Στο πρόγραμμα, όμως, η χρονική στιγμή αποθηκεύεται και πάλι σαν ένας ακέραιος αριθμός, ο οποίος υπολογίζεται ως εξής: όταν ο χρήστης δημιουργεί ένα γράφημα, αποθηκεύεται στο πρόγραμμα η ημερομηνία της πρώτης δημιουργίας. Στη συνέχεια, ο ακέραιος αριθμός για κάθε ημερολογιακή ημέρα προκύπτει από τη διαφορά ημερών της ημερομηνίας προς εισαγωγή από την ημερομηνία δημιουργίας. Αν, δηλαδή, κάποιος χρήστης χρησιμοποιεί το λογισμικό την 20/05/2021 και εισάγει την ημερομηνία 10/05/2021, τότε η συγκεκριμένη ημερομηνία θα αποθηκευτεί ως ο ακέραιος -10 , ενώ αν εισάγει την ημερομηνία 30/05/2021, τότε η συγκεκριμένη χρονική στιγμή θα αντιστοιχηθεί στον ακέραιο 10. Ο χρήστης δεν ενημερώνεται κάπου για αυτόν τον υπολογισμό, αφού στο UI εμφανίζεται μόνο η δοθείσα ημερομηνία, η οποία προκύπτει από την αντίθετη διαδικασία υπολογισμού, δηλαδή προστίθεται στην ημερομηνία δημιουργίας ο ακέραιος που έχει αποθηκευτεί στο πρόγραμμα. Ο λόγος που αποθηκεύεται η

ημερομηνία δημιουργίας του γραφήματος είναι για να αποφευχθούν λάθη, τα οποία θα προέκυπταν από την αλλαγή ημερομηνίας (π.χ. αν αλλάξει η μέρα ενώ ο χρήστης χρησιμοποιεί το λογισμικό).

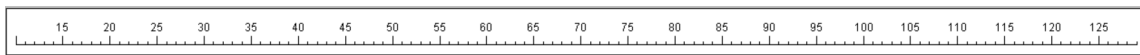
Για τους παραπάνω υπολογισμούς, δημιουργήθηκε η κλάση **DateConverter**, η οποία έχει στατικές μεθόδους μετατροπής ακεραίου (integer) σε ημερομηνία (τύπου String ή Date) ή αντίθετα. Ένας άλλος τρόπος για την μετατροπή ημερομηνίας σε ακέραιο θα ήταν η χρήση την κλάσης Date της βιβλιοθήκης της Java. Η συγκεκριμένη κλάση αποθηκεύει εσωτερικά μια ημερομηνία ως τον αριθμό των milliseconds που έχουν περάσει από την 01/01/1970. Το αποτέλεσμα αυτής της διαφοράς είναι, εν δυνάμει, ένας πολύ μεγάλος αριθμός, ο οποίος θα πρέπει να αποθηκευτεί σαν long. Για τις ανάγκες του λογισμικού όμως, προτιμήθηκε να αναπαραστούνται οι χρονικές στιγμές σαν ακέραιοι αριθμοί τύπου integer.

Και στις δύο περιπτώσεις, λοιπόν, οι χρονικές στιγμές αναπαριστούνται και αποθηκεύονται ως ακέραιοι αριθμοί τύπου integer. Σε κάθε τέτοιο ακέραιο αριθμό ή διάστημα ακεραίων, δηλαδή σε κάθε χρονική στιγμή t_i , αντιστοιχίζεται η προβολή G_i του δυναμικού γραφήματος. Από χρονική στιγμή σε επόμενη, η μετάβαση ανάμεσα στα γραφήματα αυτά αφορά τις αλλαγές που μπορεί να συμβούν μεταξύ των γραφημάτων αυτών. Μία τέτοια αλλαγή μπορεί να είναι η αφαίρεση ή η προσθήκη κόμβων και ακμών. Άλλες αλλαγές έχουν να κάνουν με τις ιδιότητες των αντικείμενων, δηλαδή με την αλλαγή χρώματος, σχήματος ή θέσης των κόμβων ή με την αλλαγή πάχους ή είδους των ακμών. Οι ιδιότητες των αντικειμένων περιγράφονται αναλυτικά σε επόμενο κεφάλαιο (Κεφάλαιο 2.2).

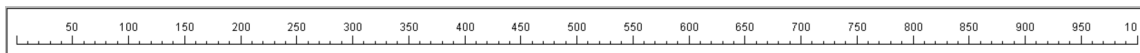
2.1.1 Αναπαράσταση χρόνου στο UI

Στο UI ο χρόνος αναπαριστάται με έναν βαθμονομημένο χάρακα στο κάτω μέρος. Ο χάρακας αυτός είναι ένα αντικείμενο της κλάσης TimelineDrawer, η οποία επεκτείνει την κλάση JComponent της βιβλιοθήκης της Java και είναι υπεύθυνη για όλους τους υπολογισμούς που αφορούν την βαθμονόμηση του χάρακα, αλλά και για την απεικόνιση του. Η βαθμονόμηση του χάρακα γίνεται με τέτοιο τρόπο, ώστε κάθε διάστημα να καταλαμβάνει το λιγότερο 10 pixel. Συνεπώς, όλος ο χάρακας μπορεί να αποτελείται το πολύ από $\frac{(\text{πλάτος})}{10}$ διαστήματα. Η κλίμακα του υπολογίζεται στην συνέχεια με βάση το εύρος των χρονικών στιγμών του γραφήματος. Υπολογίζεται, δηλαδή, πόσοι ακέραιοι «χωράνε» σε κάθε διάστημα του χάρακα, και συνεπώς σε πόσα διαστήματα ακριβώς θα είναι χωρισμένος ο χάρακας, ώστε να μην παραβαίνεται ο παραπάνω «κανόνας». Αν πρόκειται για ακέραιους αριθμούς, το πλήθος των χρονικών στιγμών ανα διάστημα μπο-

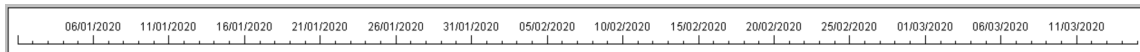
ρεί να είναι ίσο με ένα, πολλαπλάσιο του πέντε ή πολλαπλάσιο του 10. Τα διαστήματα, δηλαδή, του χάρακα μπορεί να αντιστοιχούν σε έναν, πέντε, δέκα, πενήντα, εκατό κοκ. ακεραίους. Απ' την άλλη, αν πρόκειται για ημερομηνίες, τότε κάθε διάστημα μπορεί να είναι είτε μια μέρα είτε ένας μήνας είτε ημερολογιακά χρόνια, όπου το πλήθος τους υπολογίζεται όμοια με τους ακέραιους αριθμούς (δηλαδή ένας χρόνος, πολλαπλάσιοι του 5 ή πολλαπλάσιοι του 10). Συνεπώς, μπορεί κάθε διάστημα να μην αποτελείται από τον ίδιο αριθμό ημερών και άρα χρονικών στιγμών. Για παράδειγμα, υπάρχουν μήνες που έχουν 28, 30 ή 31 μέρες, και άρα δεν θα έχουν όλα τα διαστήματα το ίδιο μήκος. Επίσης, ένα δίσεκτο έτος έχει μία μέρα παράπανω, οπότε στο διάστημα που του αντιστοιχεί, θα συνυπολογιστεί και αυτή η έξτρα μέρα. Στα σχήματα 8, 9, 10 και 11 εμφανίζονται κάποια τέτοια παραδείγματα.



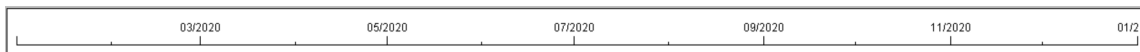
Σχήμα 8: Η μπάρα του χρόνου για τις χρονικές στιγμές [10, 129]



Σχήμα 9: Η μπάρα του χρόνου για τις χρονικές στιγμές [0, 1000] με 10 ακέραιους αριθμούς διάστημα



Σχήμα 10: Η μπάρα του χρόνου για [01/01/2020, 15/03/2020]



Σχήμα 11: Η μπάρα του χρόνου για [01/01/2020, 01/01/2021], με διάστημα ένα μήνα

2.2 Ιδιότητες Αντικειμένων (Κόμβων και Ακμών)

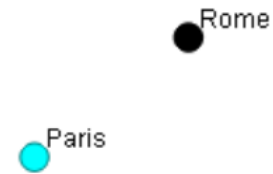
Οι ιδιότητες των αντικειμένων χωρίζονται σε δύο κατηγορίες: στις στατικές και στις δυναμικές. Οι στατικές ιδιότητες παίρνουν μία μονο τιμή, η οποία δεν αλλάζει με το χρόνο. Οι κόμβοι και οι ακμές έχουν σαν ιδιότητα ένα μοναδικό id, το οποίο είναι αλφαριθμητικό και το οποίο είναι στατικό. Όσον αφορά τις ακμές, στατική ιδιότητα αποτελεί και το αν είναι κατευθυνόμενη ή όχι, αλλά και από ποιον κόμβο ξεκινούν και σε ποιον τερματίζουν (source και target).

Παρακάτω θα περιγράψουμε τις ιδιότητες που αφορούν την απεικόνιση του γραφήματος, δηλαδή τις δυναμικές ιδιότητες. Περιέχουν πληροφορίες, δηλαδή, για το πώς θα «ζωγραφιστούν» οι κόμβοι και οι ακμές του γραφήματος εν τέλει. Για τα αντικείμενα έχουν δημιουργηθεί τρεις κλάσεις στο λογισμικό: η Element, η Node και η Edge. Η κλάση Element αφορά τις ιδιότητες και των δύο αντικειμένων, ενώ οι κλάσεις Node και Edge την επεκτείνουν. Για τις ιδιότητες επιλέχθηκε να εισάγεται και να αποθηκεύεται μόνο η στιγμή που αλλάζει τιμή κάποια ιδιότητα. Για παράδειγμα, αν ένας κόμβος έχει κόκκινο χρώμα στο διάστημα [0, 3] και μπλε χρώμα στο διάστημα [4, 7], τότε η πληροφορία που αποθηκεύεται είναι: (χρονική στιγμή 0, κόκκινο), (χρονική στιγμή 4, μπλε). Παραλείπεται, λοιπόν, να αποθηκευτεί η ίδια πληροφορία για διαφορετική χρονική στιγμή.

Οι κόμβοι και οι ακμές είναι αντικείμενα τα οποία «ζουν» για κάποιο χρονικό διάστημα και μέσα σ' αυτό το διάστημα μπορούν να γίνονται ορατοί, να εξαφανίζονται, να αλλάζουν χρώμα ή να αλλάζει κάποια άλλη ιδιότητα τους. Υπάρχει, λοιπόν, ένας διαχωρισμός, μεταξύ του πότε ζει το αντικείμενο και του πότε είναι ορατό. **Για τη ζωή των αντικειμένων** υπάρχουν δύο πεδία, το created και το disposed, τα οποία είναι τύπου integer και τα οποία καθορίζουν την χρονική στιγμή που το αντικείμενο γεννιέται και τη χρονική στιγμή που πεθαίνει, καθορίζουν λοιπόν το διάστημα «ζωής» τους. Αν ένα αντικείμενο γεννιέται την χρονική στιγμή 10 και πεθαίνει την χρονική στιγμή 20, τότε θεωρείται ότι ζει μέχρι την στιγμή 19, ότι δηλαδή την χρονική στιγμή 20 δεν υφίσταται. Ανάμεσα σ' αυτό το διάστημα το αντικείμενο μπορεί να γίνεται ορατό και να εξαφανίζεται, αλλά και να αλλάζει ιδιότητες ανεξάρτητα από το πότε είναι ορατό, μπορεί δηλαδή να αλλάζει χρώμα σε χρονική στιγμή που δεν είναι ορατό. Όπως αναφέρθηκε για τις υπόλοιπες ιδιότητες, έτσι και για την πληροφορία του **πότε είναι ορατό ένα αντικείμενο**, αποθηκεύεται μόνο η χρονική στιγμή που αλλάζει η κατάσταση του, δηλαδή η χρονική στιγμή που ένα αντικείμενο γίνεται ορατό ή μη. Για όλες τις δυναμικές ιδιότητες, λοιπόν, των κόμβων ή των ακμών ο τρόπος που αποθηκεύονται οι τιμές τους είναι δυάδες του τύπου (χρονική στιγμή, τιμή ιδιότητας) ταξινομημένες ως προς την χρονική στιγμή. Για το αν είναι ορατός ένας κόμβος ή μια ακμή, η πληροφορία αποθηκεύεται ως δυάδα integer αριθμού και boolean τιμής, π.χ. (0, true). Κάτι το οποίο, λοιπόν, κατά κάποιο τρόπο απαγορεύεται είναι η πληροφορία (2, true), (3, true), αφού αρκεί μόνο η πρώτη δυάδα για να περιγράψει το αν είναι ορατό το αντικείμενο την χρονική στιγμή 3.

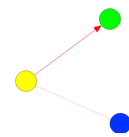
Οι ιδιότητες που έχουν και τα δύο αντικείμενα του γραφήματος κοινά, δηλαδή στοιχεία της κλάσης Element, είναι το πότε γίνονται ορατά ή όχι (visibility state), το όνομα (label) και το χρώμα (color). Όσον αφορά **το όνομα**, η τιμή της ιδιότητας στην δυάδα

(χρονική στιγμή, τιμή ιδιότητας), είναι ένα διάνυσμα με δύο τιμές, όπου στην πρώτη θέση μπαίνει το όνομα του αντικειμένου και στη δεύτερη θέση μπαίνει μια τιμή αλήθειας που περιγράφει αν το συγκεκριμένο όνομα είναι ορατό την συγκεκριμένη χρονική στιγμή. Θα μπορούσαν, δηλαδή, δύο διαδοχικές τιμές για το όνομα να είναι (0, [Κώστας, true]) και (1, [Κώστας, false]).



Για τους κόμβους, ο τρόπος που ζωγραφίζονται τα ονόματα είναι ακριβώς δεξιά τους, ενώ για τις ακμές τα ονόματα τους δεν εμφανίζονται καθόλου. Η πρόσθετη ιδιότητα, λοιπόν, για την ορατότητα του ονόματος των κόμβων και των ακμών αφορά, στην πραγματικότητα, μόνο τους κόμβους. Αν ένας κόμβος δεν έχει όνομα για κάποιο διάστημα, τότε το διάνυσμα παίρνει τις τιμές [false], δηλαδή κενό αλφαριθμητικό, όπου δεν είναι ορατό.

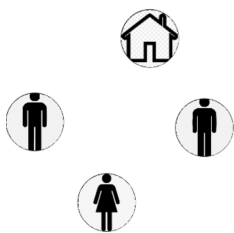
Για το **χρώμα** των αντικειμένων, αποθηκεύονται δυάδες του τύπου (χρονική στιγμή, χρώμα), όπου το χρώμα είναι τύπου Color της βιβλιοθήκης της Java σε r, g, b μορφή. Το χρώμα αυτό είναι τελικά το χρώμα, στο οποίο θα απεικονιστεί ο κόμβος ή η ακμή για την συγκεκριμένη χρονική στιγμή.



Οι πρόσθετες ιδιότητες που έχουν οι κόμβοι ανεξάρτητα από τις ακμές, δηλαδή στοιχεία της κλάσης Node, είναι το σχήμα (shape), η θέση (position) και η φωτογραφία τους (icon). Όσον αφορά **το σχήμα των κόμβων**, υπάρχουν δύο επιλογές, η έλλειψη (ellipse) και το ορθογώνιο παραλληλόγραμμο (rectangle). Και για τα δύο σχήματα, πρέπει να αποθηκεύονται παράλληλα το ύψος και το πλάτος τους, τα οποία είναι τύπου float. Για την αποθήκευση του σχήματος χρησιμοποιείται η κλάση Shape της Java και, πιο συγκεκριμένα, οι κλάσεις Ellipse και Rectangle, που επεκτείνουν την κλάση Shape. Οι τιμές για την συγκεκριμένη ιδιότητα του κόμβου μπορεί, για παράδειγμα, να πάρουν τη μορφή (10, ellipse 10 x 10), (14, rectangle 15 x 20).

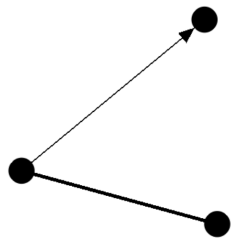
Για **την θέση των κόμβων** χρησιμοποιείται η κλάση Point της Java και έχει μορφή καρτεσιανών συντεταγμένων, όπου τα x, y είναι τύπου float. Η θέση ενός κόμβου είναι λοιπόν ένα σημείο στο επίπεδο και αφορά το γεωμετρικό κέντρο του σχήματος που έχει ο κόμβος εκείνη τη στιγμή. Αν, δηλαδή, ένας κόμβος βρίσκεται στη θέση (100, 100) και είναι κύκλος ακτίνας 10, τότε ζωγραφίζεται με το κέντρο του στην θέση (100, 100) και ακτίνα 10. Οι συντεταγμένες, όπως και οι διαστάσεις των σχημάτων αφορούν pixel, τα οποία όμως προσαρμόζονται, έτσι ώστε να είναι καλά ορατό το γράφημα στην οθόνη. Ένας κύκλος ακτίνας 10 δεν είναι, δηλαδή, σίγουρο ότι θα ζωγραφιστεί, εν

τέλει, με ακτίνα 10 pixel, λόγω του zoom που μπορεί να έχει γίνει κατά την διαδικασία της παραπάνω προσαρμογής. (βλ. κεφάλαιο 2.5 για περαιτέρω πληροφορίες).



Τελευταία ιδιότητα για τους κόμβους είναι η **φωτογραφία (icon)** που τοποθετείται μέσα στο σχήμα του κόμβου. Για την φωτογραφία δεν αποθηκεύεται στο πρόγραμμα η ίδια η φωτογραφία, αλλά ένα αλφαριθμητικό String που «δείχνει» τη θέση της φωτογραφίας στον υπολογιστή. Ένα τέτοιο String μπορεί για παράδειγμα να είναι το `C:/Users/30693/Documents/image.png`. Οι φωτογραφίες πρέπει να είναι σε format .png, .jpg ή .jpeg. Αν ένας κόμβος δεν έχει φωτογραφία για ένα συγκεκριμένο χρονικό διάστημα, τότε είναι αποθηκευμένο ένα κενό αλφαριθμητικό String. Οι φωτογραφίες τροποποιούνται και «κόβονται» με τέτοιο τρόπο, ώστε να ταιριάζουν ακριβώς πάνω στο σχήμα του κόμβου. Δηλαδή, για έναν κυκλικό κόμβο ακτίνας 10, κόβεται από τη φωτογραφία ο μέγιστος δυνατός κύκλος από το κέντρο της, ενώ για ένα ορθογώνιο παραλληλόγραμμο 20 x 15, κόβεται το μέγιστο δυνατό ορθογώνιο παραλληλόγραμμο με αυτή την αναλογία διαστάσεων. Αυτή η τροποποίηση γίνεται μόνο μία φορά για κάθε φωτογραφία και σχήμα και το αποτέλεσμα της αποθηκεύεται σε έναν φάκελο με όνομα `NodeIcons`, ο οποίος τοποθετείται στον φάκελο που βρίσκεται η φωτογραφία. Η επεξεργασία, δηλαδή, που περιγράφηκε παραπάνω, εκτελείται μόνο αν, κατά την απεικόνιση του κόμβου διαπιστωθεί από το πρόγραμμα ότι για το συγκεκριμένο σχήμα που έχει ο κόμβος και για την συγκεκριμένη φωτογραφία που πρέπει να τοποθετηθεί, δεν έχει γίνει η επεξεργασία. Αυτή η διαπίστωση γίνεται με τη χρήση του `HashMap images`, το οποίο έχει ως κλειδί ένα αντικείμενο της κλάσης `Key` και ως τιμή ένα αλφαριθμητικό String, το οποίο «δείχνει» την θέση της επεξεργασμένης φωτογραφίας στον υπολογιστή. Ένα αντικείμενο της κλάσης `Key`, έχει δύο πεδία, το ένα είναι το σχήμα και το άλλο είναι η θέση της (πρωτότυπης) φωτογραφίας στον υπολογιστή. Στο `HashMap images` αποθηκεύονται, λοιπόν, όλοι οι συνδυασμοί φωτογραφίας- σχήματος που έχει συναντήσει το πρόγραμμα κάθε στιγμή, αλλά και η θέση της αντίστοιχης επεξεργασμένης φωτογραφίας. Αν δύο κόμβοι με το ίδιο σχήμα χρησιμοποιούν την ίδια φωτογραφία, τότε η επεξεργασία και η αποθήκευση γίνεται μόνο μία φορά και στην συνέχεια η φωτογραφία «διαβάζεται» από τον φάκελο. Κάθε στιγμή, δηλαδή, στον φάκελο `NodeIcons` υπάρχουν οι φωτογραφίες που ήδη έχει χρειαστεί να απεικονιστούν και να τοποθετηθούν σε κάποιον κόμβο. Επίσης, κατά την επεξεργασία της φωτογραφίας, μειώνεται και η ανάλυση της, διότι για φωτογραφίες πολύ υψηλής ανάλυσης μειώνεται η ταχύτητα απεικόνισής τους. Τέλος, τις χρονικές στιγμές, που τοποθετείται κάποια φωτογραφία σε έναν κόμβο, δεν φαίνεται το χρώμα του κόμβου, αφού αυτό καλύπτεται από την φωτογραφία.

Όσον αφορά τις ακμές, οι πρόσθετες ιδιότητες που έχουν είναι το πάχος (width), το είδος (style) και, για τις κατευθυνόμενες ακμές, το αν θα φαίνεται το βέλος που προσδιορίζει την κατεύθυνση της (arrow visibility). Το πάχος της ακμής αποθηκεύεται



σαν ένας αριθμός τύπου float, οπότε μια δυάδα για την ιδιότητα του πάχους θα έχει τη μορφή (χρονική στιγμή, float αριθμος), ή για παράδειγμα (4, 3). Αυτός ο αριθμός αναφέρεται και πάλι σε pixel, αλλά όπως αναφέρθηκε και παραπάνω στο σχήμα του κόμβου, η τελική απεικόνιση της ακμής δεν είναι απαραίτητο ότι θα γίνει με τα συγκεκριμένα pixel.

Για το είδος της ακμής, υπάρχουν 3 επιλογές: συνεχείς (solid), διακεκομμένες με τελείες (dotted) ή διακεκομμένες με παύλες (dashed). Αυτή η πληροφορία αποθηκεύεται σαν αλφαριθμητικό String, όπου η ιδιότητα μπορεί να πάρει μία από τις παραπάνω τιμές (solid, dotted ή dashed). Στο σχήμα 12 φαίνονται παραδείγματα τέτοιων ακμών.



Σχήμα 12: Solid, dotted και dashed ακμή

Τέλος, όταν πρόκειται για κατευθυνόμενες ακμές, υπάρχει και η ιδιότητα του **αν είναι ορατό το βέλος** που υποδεικνύει την κατεύθυνση της. Αυτή η πληροφορία αποθηκεύεται με μία τιμή αλήθειας, δηλαδή τύπου boolean που παίρνει τιμές true ή false. Τέτοια δυάδες μπορεί να έχουν την μορφή (3, true), (5, false) κοκ.

Προφανώς, μια ακμή δεν μπορεί να ζει χωρίς κάποιον κόμβο, οπότε τις χρονικές στιγμές που ζει μια ακμή θα πρέπει να ζουν και οι δύο κόμβοι που αποτελούν αρχή και τέλος της ακμής. Συνεπώς, τα πεδία created και disposed της ακμής, θα πρέπει να βρίσκονται ανάμεσα στα αντίστοιχα πεδία των δύο κόμβων. Αντίστοιχα, αν μια ακμή είναι ορατή σε χρονικές στιγμές που οι κόμβοι δεν είναι ορατοί, τότε η ακμή δεν εμφανίζεται καθόλου στο γράφημα.

2.3 Το Γράφημα

Το δυναμικό γράφημα είναι ένα αντικείμενο της κλάσης Graph. Οι κόμβοι που το αποτελούν αποθηκεύονται σε μία ArrayList nodes. Παράλληλα με τη λίστα αυτή δημιουργείται και ένα HashMap<String, Integer> nodeId, το οποίο έχει ως κλειδί το id του κόμβου (String) και ως τιμή τη θέση του κόμβου στη λίστα nodes. Δίνεται, δηλαδή,

ένα καινούριο `id` στους κόμβους, το οποίο είναι ακέραιος αριθμός που «δείχνει» τη θέση του κόμβου στη λίστα. Αυτός ο τρόπος αποθήκευσης βοηθάει, διότι η ανάζητηση ενός κόμβου στο γράφημα γίνεται σε σταθερό χρόνο. Όμοια αποθηκεύονται και οι ακμές του γραφήματος.

Αν θέλει κάποιος να προσθέσει έναν κόμβο στο γράφημα πρέπει πρώτα να δημιουργήσει το αντικείμενο του κόμβου και μετά να το εισάγει στο γράφημα. Η κλάση `Graph` έχει τις μεθόδους `addNode(String id)`, `addEdge(String id)`, `removeNode(String id)` και `removeEdge(String id)`. Με τις μεθόδους αυτές, εισάγονται και αφαιρούνται κόμβοι και ακμές στο γράφημα. Κατά τη διαγραφή ενός κόμβου ή μιας ακμής, δεν διαγράφεται και η αντίστοιχη θέση στη λίστα, αλλά αντικαθίσταται ο κόμβος ή η ακμή με το κενό `null` και διαγράφεται η αντίστοιχη εγγραφή στο `HashMap`. Αυτό γίνεται, διότι αν διαγραφόταν η θέση στη λίστα, θα έπρεπε για όλους τους κόμβους που βρίσκονται σε επόμενες θέσεις να διαγραφεί η εγγραφή που τους αντιστοιχεί στο `HashMap` και να προστεθεί καινούρια με τον νέο δείκτη. Όταν διαγράφεται ένας κόμβος, διαγράφονται αυτόματα και οι ακμές του.

Το εύρος τιμών του γραφήματος υπολογίζεται με βάση τη ζωή των κόμβων. Η ελάχιστη τιμή του, δηλαδή η πρώτη χρονική στιγμή του δυναμικού γραφήματος, είναι η χρονική στιγμή που δημιουργείτε ο πρώτος κόμβος, δηλαδή η ελάχιστη τιμή από τα πεδία `created` όλων των κόμβων. Αντίστοιχα, η τελευταία στιγμή του δυναμικού γραφήματος είναι η χρονική στιγμή που πεθαίνει ο τελευταίος κόμβος, δηλαδή η μέγιστη τιμή από τα πεδία `disposed` όλων των κόμβων μειωμένο κατά 1 (αφού διαφορετικά την τελευταία χρονική δε θα ζούσε κανένας κόμβος).

2.4 Δεδομένα- Format *.grdyn

Για την ανάγνωση δεδομένων από το πρόγραμμα δημιουργήθηκε ένα καινούριο `format` αρχείου, το οποίο περιέχει κωδικοποιημένες όλες τις πληροφορίες του δυναμικού γραφήματος που εισάγεται. Ένα αρχείο αυτής της μορφής είναι γραμμένο στη γλώσσα `xml` και πρέπει να τελειώνει σε `.grdyn`. Οι πληροφορίες που δίνονται είναι χωρισμένες σε τέσσερα μέρη: οι γενικές πληροφορίες του γραφήματος, μια δήλωση των ιδιοτήτων των κόμβων και των ακμών, οι πληροφορίες για τους κόμβους (π.χ. χρώμα, σχήμα κλπ.) και για τις ακμές (π.χ. χρώμα, πάχος κλπ.). Παρακάτω δίνεται μια εκτενής περιγραφή της δομής αυτών των αρχείων και των απαραίτητων πληροφοριών του δυναμικού γραφήματος προς εισαγωγή.

Graph:

```

<graph defaultedgetype="directed" timeformat="integer">
  <attributes class="node">
    ...
  </attributes>
  <attributes class="edge">
    ...
  </attributes>
  <nodes>
    ...
  </nodes>
  <edges>
    ...
  </edges>
</graph>

```

* Για εξοικονόμηση μνήμης, αν όλες οι ακμές του γραφήματος είναι κατευθυνόμενες (ή αντίστοιχα μη-κατευθυνόμενες), τότε μπαίνει στο tag του γραφήματος η ιδιότητα defaultedgetype, η οποία παίρνει τις τιμές directed, undirected και mixed. Στην περίπτωση του mixed κάθε ακμή έχει και την αντίστοιχη ιδιότητα.

* Το γράφημα μπορεί να έχει δύο τιμές για το timeformat: integer και date (σε μορφή yyyy-MM-dd).

Attributes:

```

<attributes class="node">
  <attribute id="place_name" title="Place Name" type="string" default=" "/>
  <attribute id="gn_country" title="Country" type="string" default="Country"/>
  ...
  <attribute id="dynamic_str" title="Strongly Connected" type="integer"/>
</attributes>
<attributes class="edge">
  <attribute id="ee_bio_id" title="ee bio id" type="string"/>
  ...
</attributes>

```

* Τα attributes του γραφήματος χωρίζονται σε δύο κατηγορίες, σ' αυτά των κόμβων και σ' αυτά των ακμών.

- * Κάθε attribute πρέπει να έχει ένα id, ένα όνομα (title) και τι τύπου είναι (type), ενώ προαιρετικά μπορεί να δίνεται μια default τιμή για το συγκεκριμένο attribute.
- * Το id των attribute δεν μπορεί να περιέχει κενά και τα σύμβολα / \ : " * ? < > |
- * Το type μπορεί να πάρει την τιμή int, float, double, String και boolean.

Nodes:

```

<nodes>
  <node id="3038261" created="1760" disposed="1778">
    <state>
      <visible timestamp="1760" value="true"/>
      <visible timestamp="1761" value="false"/>
      <visible timestamp="1776" value="true"/>
    </state>
    <attvalues>
      <attvalue for="place_name" value="Albi" timestamp="1760"/>
      <attvalue for="gn_country" value="France" timestamp="1760"/>
      ...
      <attvalue for="dynamic_str" value="0" timestamp="1760"/>
    </attvalues>
    <visvalues>
      <label value="Albi" timestamp="1760" visible="true"/>
      <shape value="ellipse" width="4.0" height="4.0" timestamp="1760"/>
      <shape value="rectangle" width="4.0" height="6.0" timestamp="1776"/>
      <position value="(-155.98499;-147.12442)" timestamp="1760"/>
      <color value="(61;197;255)" timestamp="1760"/>
      <color value="(145;182;67)" timestamp="1776"/>
    </visvalues>
  </node>
  <node>
    ...
  </node>
</nodes>

```

- * Πρώτο στοιχείο του κόμβου πρέπει να είναι το id του, το timestamp που δημιουργήθηκε (created) και το timestamp που «πέθανε» (disposed). Αν κάποιο από αυτά τα πεδία είναι κενό, τότε ο κόμβος αυτός δεν εισάγεται καθόλου στο γράφημα, και ο χρήστης ενημερώνεται με αντίστοιχο μήνυμα.

- * Μέσα στο state-block μπαίνουν τα timestamps στα οποία αλλάζει δυναμική κατάσταση ο κόμβος, δηλαδή γίνεται ορατός ή όχι (visible value="true").
- * Όλα τα timestamps, τα οποία προσδιορίζουν πότε ο κόμβος γίνεται ορατός ή όχι πρέπει να είναι μέσα στο διάστημα [created,disposed), αλλιώς το timestamp δεν εισάγεται στον κόμβο
- * Τα timestamps πρέπει να δίνονται με ταξινομημένη αύξουσα σειρά, αλλιώς ο κόμβος δεν εισάγεται στο γράφημα, βγάζοντας αντίστοιχο μήνυμα.
- * Ακολουθούν οι τιμές για τα user defined attributes του κόμβου ταξινομημένα πρώτα ως προς attribute και μετά ως προς τα timestamps.
- * Στο επόμενο block δίνονται τα timestamps που αφορούν την απεικόνιση του κόμβου, δηλαδή στοιχεία για το όνομα, το σχήμα και το μέγεθος, τη θέση και το χρώμα του κόμβου ταξινομημένα ως προς ιδιότητα και μετά ως προς timestamp.
- * Για το σχήμα οι επιλογές είναι έλλειψη (ellipse) και ορθογώνιο παραλληλόγραμμο (rectangle) και πρέπει να δίνεται το πλάτος και το ύψος του εκάστοτε σχήματος. Το χρώμα δίνεται σε r, g, b μορφή. Για το όνομα του κόμβου πρέπει να δίνεται και η πληροφορία αν το συγκεκριμένο όνομα είναι ορατό (visible) για το συγκεκριμένο timestamp.
- * Οι default τιμές για τις ιδιότητες που αφορούν την απεικόνιση είναι: για το σχήμα κύκλος, για το πλάτος και το ύψος 10 (κύκλος ακτίνας 5), για το χρώμα μαύρο και η θέση του κόμβου παίρνει τυχαία τιμή μέσα στα «σύνορα» του υπόλοιπου γραφήματος.
- * Υπάρχει η επιλογή προσθήκης φωτογραφίας στον κόμβο, όπου η δήλωση γίνεται ως εξής:

```
<icon value="image.png" timestamp="1776"/>
```

Οι φωτογραφίες πρέπει να βρίσκονται στον ίδιο φάκελο, όπου βρίσκεται και το αρχείο *.grdyn.

- * Αντίστοιχα, αν για κάποιο attribute δεν δίνεται η τιμή για την πρώτη χρονική στιγμή, τότε δίνεται η τιμή null ή η default τιμή για το attribute, αν δίνεται.

Edges:

```

<edges>
  <edge id="18165" source="2643743" target="2660646" created="1768"
  disposed="1769">
    <state>
      <visible timestamp="1768" value="true"/>
    </state>
    <attvalues>
      <attvalue for="ee_bio_id" value="keategeorg002130" timestamp="1768"/>
      ...
    </attvalues>
    <visvalues>
      <color value="(0;0;0)" timestamp="1768"/>
      <width value="1.0" timestamp="1768"/>
      <style value="solid" timestamp="1768"/>
      <arrow value="true" timestamp="1768"/>
    </visvalues>
  </edge>
  <edge>
    ...
  </edge>
</edges>

```

- * Πρώτο στοιχείο της ακμής πρέπει να είναι το id της, το id του source κόμβου και το id του target κόμβου, το timestamp που δημιουργήθηκε (created) και το timestamp που «πέθανε» (disposed) καθώς και η πληροφορία αν η ακμή είναι κατευθυνόμενη ή όχι, αν αυτό δεν αναγράφεται στις ιδιότητες του γραφήματος.
- * Αν κάποιο από αυτά τα πεδία είναι κενό, τότε η συγκεκριμένη ακμή δεν εισάγεται καθόλου στο γράφημα, ενημερώνοντας το χρήστη με αντίστοιχο μήνυμα.
- * Όσον αφορά τα δυναμικά χαρακτηριστικά και τα user defined attributes, η δομή που ακολουθείται είναι ακριβώς ίδια με τη δομή που ακολουθείται στους κόμβους, η οποία περιγράφηκε παραπάνω.
- * Ιδιότητες που αφορούν την απεικόνιση των ακμών είναι το όνομα της, το πάχος της (default τιμή 1, με τιμές > 0), το είδος ευθείας, το αν φαίνεται το βέλος στην συγκεκριμένη χρονική στιγμή και το χρώμα της το οποία πάλι πρέπει να δίνεται σε r, g, b μορφή.

! Το χρώμα πρέπει να δίνεται σε r, g, b μορφή, όπου $0 \leq r, g, b \leq 255$, με τη μορφή $\{r; g; b\}$. Διαθέσιμα χρώματα με τα ονόματά τους: black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow

! Η θέση του κόμβου πρέπει να δίνεται σε μορφή $\{x; y\}$

! Όλες οι ιδιότητες των κόμβων (εκτός από το id του) και των ακμών (εκτός από τα id, source-id, target-id και directed) είναι δυναμικές, οπότε για κάθε συμβάν (δηλαδή αλλαγή της τιμής κάποιου attribute) θα πρέπει να δίνεται και το timestamp, στο οποίο αυτό πραγματοποιήθηκε. Αν αυτό δεν δίνεται, τότε το συγκεκριμένο συμβάν δεν εισάγεται στον κόμβο/ακμή με κανέναν τρόπο.

Reserved Words:

> graph

– defaultedgetype = {directed, undirected, mixed} –timeformat = {integer, datetime}

> attributes

– class = {node, edge}

> attribute

– id –title –type

> nodes

> node

– id –created –disposed

> state

> visible

– timestamp –value = {true, false}

> attvalues

> attvalue

- for –value –timestamp
- > visvalues
 - > label
 - value –timestamp –visible = {true, false}
 - > shape
 - value = {ellipse, rectangle} –width –height –timestamp
 - > position
 - value = { ; } –timestamp
 - > color
 - value = { ; ; } –timestamp
 - > icon
 - value –timestamp
- > edges
 - > edge
 - id –source –target –directed = {true, false} –created –disposed
 - > state
 - > visible
 - timestamp –value = {true, false}
 - > attvalues
 - > attvalue
 - for –value –timestamp
 - > visvalues
 - > label

- value –timestamp –visible = {true, false}
- > color
 - value = { ; ; } –timestamp
- > width
 - value –timestamp
- > style
 - value={solid, dashed, dotted} –timestamp
- > arrow
 - value={true, false} –timestamp

Γιατί όχι κάποιο άλλο format;

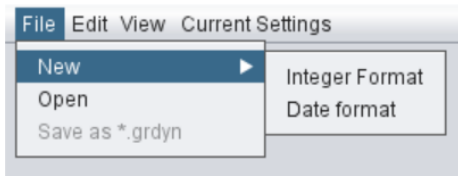
Ο λόγος που δημιουργήθηκε καινούριο format από την αρχή και δεν χρησιμοποιήθηκε κάποιο άλλο format για δυναμικά γραφήματα είναι ότι δεν επιτρέπουν την εισαγωγή δυναμικών ιδιοτήτων των κόμβων και των ακμών. Ένα format δυναμικών γραφημάτων που ήταν «υποψήφιο» ήταν το .gexf, το οποίο δημιουργήθηκε για το Gephi [5] και είναι επίσης γραμμένο σε XML. Αυτό, όμως, δεν επέτρεπε την εισαγωγή δυναμικών πληροφοριών για τις ιδιότητες των κόμβων και των ακμών. Για παράδειγμα, το χρώμα των κόμβων στο συγκεκριμένο format δηλώνεται ως

```
<viz:color r="239" g="173" b="66" a="0.6"/>
```

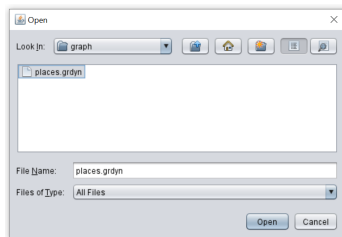
οπότε δεν μπορεί ένα αντικείμενο να λάβει διαφορετικές τιμές για το χρώμα σε διαφορετικές χρονικές στιγμές. Αντίστοιχα ισχύει και για τις υπόλοιπες ιδιότητες που έχουν συμπεριληφθεί στο πρόγραμμα, είτε πρόκειται για ιδιότητες κόμβων είτε ακμών.

2.5 Στοιχεία του UI

Όταν ανοίγει κάποιος την εφαρμογή έχει δύο επιλογές: η μία είναι να εισάγει ένα αρχείο .gdfjn με πληροφορίες για ένα δυναμικό γράφημα και η άλλη είναι να δημιουργήσει ένα καινούριο γράφημα από την αρχή. Με την δεύτερη επιλογή, ο χρήστης καλείται να επιλέξει για την αναπαραστάση του χρόνου που επιθυμεί, δηλαδή αν θέλει το δυναμικό γράφημα να αφορά ακέραιες χρονικές στιγμές ή να αφορά ημερομηνίες.



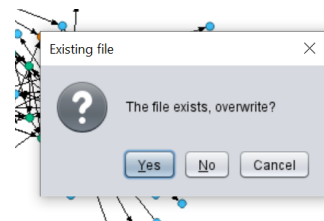
Και στις δύο περιπτώσεις δημιουργείται ένα γράφημα με έναν μόνο κόμβο, ο οποίος είναι ορατός για όλο το διάστημα «ζωής» του, είναι κύκλος ακτίνας 10, έχει μαύρο χρώμα, βρίσκεται στη θέση (0.0, 0.0), έχει κενό όνομα και κενή φωτογραφία. Αν ο χρήστης επιλέξει την πρώτη αναπαράσταση των ακέραιων αριθμών, ο κόμβος αυτός ζει στο διάστημα [0, 10], ενώ σε αντίθετη περίπτωση ο κόμβος αυτός ζει στο διάστημα [20/05/2021, 29/05/2021].



Αν ο χρήστης επιλέξει να εισάγει αρχείο, τότε ανοίγει μια αναζήτηση της κλάσης `JFileChooser` της Java, έτσι ώστε να μπορέσει να επιλέξει το αρχείο που επιθυμεί. Το πρόγραμμα αποδέχεται το αρχείο μόνο αν τελειώνει σε `.grdyn`, αλλιώς εμφανίζεται ανάλογο προειδοποιητικό μήνυμα. Το διάβασμα του αρχείου γίνεται μέσω της κλάσης `ImportFromGrdyn`, η οποία δημιουργεί τους κόμβους και τις ακμές με βάση τις πληροφορίες του αρχείου και τα εισάγει στο δυναμικό γράφημα. Η συγκεκριμένη κλάση ελέγχει επίσης το αρχείο για το αν παραβιάζει τους κανόνες που αναφέρθηκαν στο κεφάλαιο 2.4. Αν βρεθούν λάθη, ανοίγει ένα παράθυρο, όπου ενημερώνεται ο χρήστης για όλα τα λάθη που βρέθηκαν και για την σημασία που έχει το λάθος για το αντικείμενο προς εισαγωγή, δηλαδή για το αν εισήχθη στο γράφημα ή όχι.

Επίσης, μπορεί κάποιος να εξάγει το γράφημα του σε μορφή `.grdyn`. Αν κάποιος κάνει αυτή την επιλογή, τότε και πάλι ανοίγει μια αναζήτηση της κλάσης `JFileChooser` της Java, έτσι ώστε να μπορέσει να επιλέξει το αρχείο που επιθυμεί. Αν το αρχείο υπάρχει ήδη, εμφανίζεται ενημερωτικό μήνυμα που ρωτάει τον χρήστη αν επιθυμεί να συνεχίσει με την επιλογή του, αφού το προηγούμενο αρχείο θα χαθεί.

Διαφορετικά δημιουργείται ένα καινούριο αρχείο με το όνομα της επιλογής του χρήστη. Υπεύθυνη για την εξαγωγή του γραφήματος σε αρχείο είναι η κλάση `ExportToGrdyn`, η οποία «γράφει» στο προαναφερθέν αρχείο όλες τις πληροφορίες του γραφήματος, στη δομή που περιγράφηκε στο κεφάλαιο 2.4.

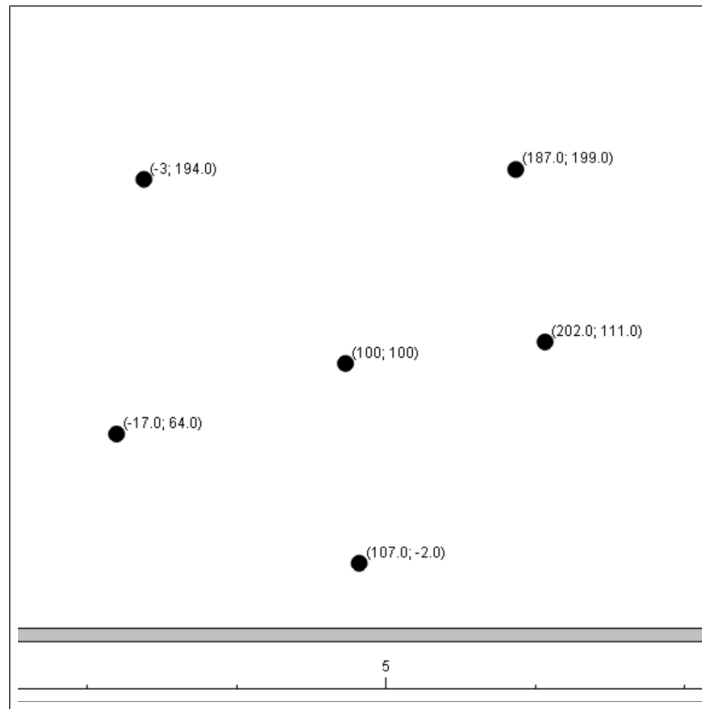


2.5.1 Απεικόνιση του γραφήματος

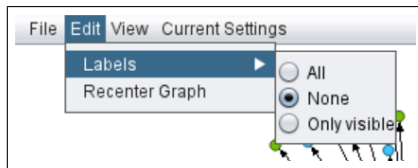
Στη συνέχεια, το δυναμικό γράφημα απεικονίζεται στον «καμβά» στο πάνω μέρος του UI. Οι κόμβοι και οι ακμές του γραφήματος απεικονίζονται με την πρώτη τιμή για κάθε ιδιότητα. Για παράδειγμα, αν ένας κόμβος είναι αρχικά μαύρος και μετά γίνεται κόκκινος, τότε θα απεικονιστεί με μαύρο χρώμα. Στο κάτω μέρος του UI εμφανίζεται η μπάρα του χρόνου, δηλαδή ο βαθμονομημένος χάρακας που περιγράφηκε παραπάνω, με όλο το εύρος χρονικών στιγμών του δυναμικού γραφήματος. Επίσης, κάτω από τον χάρακα εμφανίζονται κάποια ανενεργά κουμπιά που αντιστοιχούν σε κάποιες λειτουργίες βίντεο της δυναμικής απεικόνισης.

Η απεικόνιση του γραφήματος, όπως προαναφέρθηκε, γίνεται με τη χρήση του OpenGL και υλοποιείται μέσω της κλάσης Canvas, η οποία επεκτείνει την κλάση Drawer. Η κλάση Drawer επεκτείνει την κλάση GLJPanel και υλοποιεί την διεπαφή GLEventListener του Java OpenGL. Η διεπαφή GLEventListener έχει τέσσερις μεθόδους: `init`, `display`, `reshape` και `dispose`, και είναι εκείνη που είναι υπεύθυνη για την απεικόνιση του αντικειμένου (rendering). Η `init` καλείται πριν ζωγραφιστεί το αντικείμενο προς απεικόνιση (εδώ το γράφημα), η μέθοδος `display` περιέχει όλες τις εντολές για την απεικόνιση του αντικειμένου, η μέθοδος `reshape` καλείται όταν ο χώρος απεικόνισης αλλάζει μέγεθος και η μέθοδος `dispose` καλείται όταν καταστρέφεται το αντικείμενο. Η κλάση GLJPanel του Java OpenGL είναι επέκταση του JPanel της Swing βιβλιοθήκης της Java.

Η κλάση Drawer, η οποία είναι υπεύθυνη για την απεικόνιση του γραφήματος, παίρνει ένα αντικείμενο της κλάσης Graph και «ζωγραφίζει» τους κόμβους και τις ακμές του. Πριν την απεικόνιση του γραφήματος, η κλάση Drawer υπολογίζει κάποιες παραμέτρους, έτσι ώστε το γράφημα να απεικονιστεί στο κέντρο της οθόνης, ανεξάρτητα από τις συντεταγμένες των κόμβων, δηλαδή ανεξάρτητα από το πόσο «απλωμένοι» είναι οι κόμβοι στο επίπεδο ή από τη θέση τους. Αυτές οι παράμετροι υπολογίζονται στη μέθοδο `init` και αφορούν το `zoom` και το πόσο πρέπει να μετακινηθεί το γράφημα αριστερά- δεξιά ή πάνω- κάτω, έτσι ώστε να «χωρέσει» και να τοποθετηθεί το γράφημα σε μια περιοχή 400x400 pixel στο κέντρο της οθόνης. Το κέντρο των συντεταγμένων, δηλαδή το σημείο (0.0, 0.0) δεν μπορεί λοιπόν να προσδιοριστεί, αφού εξαρτάται από τις συντεταγμένες των κόμβων, δηλαδή από το πόσο έχει μετακινηθεί το γράφημα. Πριν την μετακίνηση αυτή, το σημείο (0.0, 0.0) βρίσκεται στην πάνω αριστερά γωνία της οθόνης. Η μέθοδος `display` εφαρμόζει αυτές τις μετατροπές και περιέχει εντολές για την απεικόνιση των κόμβων και των ακμών του γραφήματος. Διαπερνά, δηλαδή, όλους τους κόμβους και όλες τις ακμές του γραφήματος και τα ζωγραφίζει με τις εκάστοτε ιδιότητες (χρώμα, σχήμα, πάχος κλπ.).

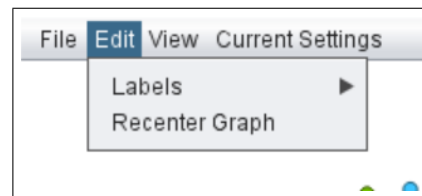


Σχήμα 13: Μέρος του καμβά μετά την εισαγωγή ενός γραφήματος με τις αναγραφόμενες συντεταγμένες



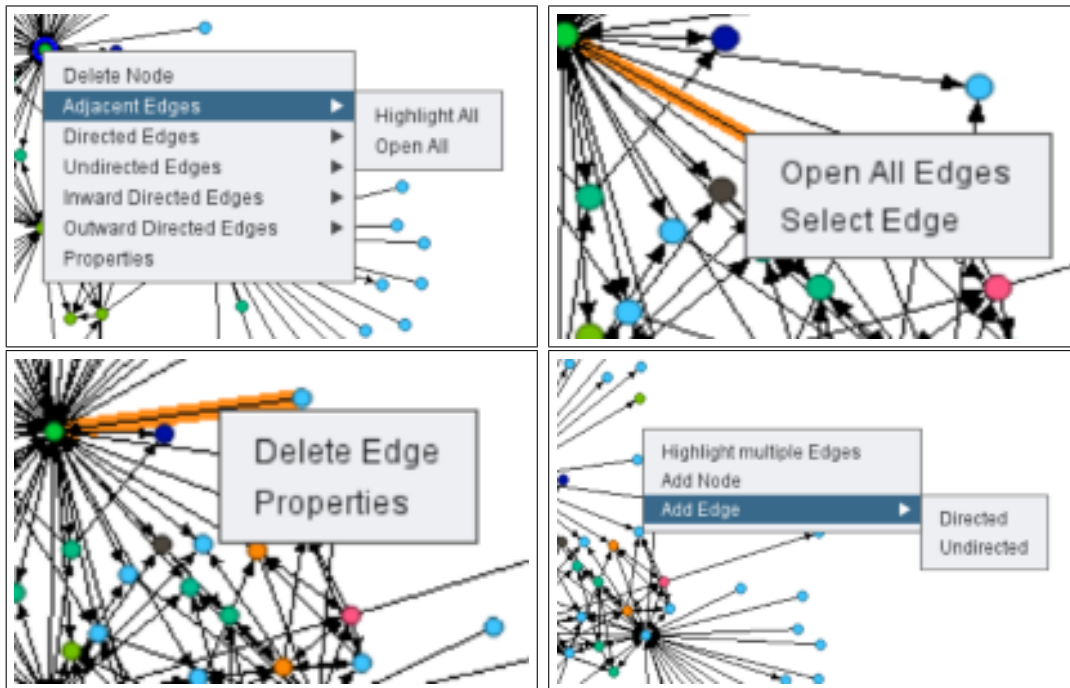
Για το όνομα των κόμβων, υπάρχουν τρεις επιλογές: να φαίνονται όλα, να μην φαίνεται κανένα ή να φαίνονται μόνο αυτά που είναι ορατά. Στην τρίτη επιλογή εμφανίζονται μόνο τα ονόματα που η δυάδα τους έχει τη μορφή (χρονική στιγμή, [Όνομα, true]) (βλ. κεφ. 2.2).

Όλες τις «διαδραστικές» λειτουργίες του γραφήματος, τις χειρίζεται η κλάση Canvas. Ο χρήστης με τη χρήση του ποντικιού του μπορεί να ζουμάρει (zoom-in/zoom-out) ή να σύρει ολόκληρο το γράφημα πατώντας παρατεταμένα το δεξί πλήκτρο του ποντικιού. Στο μενού της εφαρμογής υπάρχει η επιλογή "Recenter Graph", με την οποία το γράφημα επανέρχεται στο κέντρο της οθόνης. Επίσης, ο χρήστης μπορεί να επιλέξει ένα αντικείμενο είτε είναι κόμβος είτε είναι ακμή (ο επιλεγμένος κόμβος εμφανίζεται με μπλε και η επιλεγμένη ακμή με πορτοκαλί περίβλημα),



ενώ με δεξί κλικ ανοίγουν οι εξής επιλογές, οι οποίες φαίνονται και στο σχήμα 14:

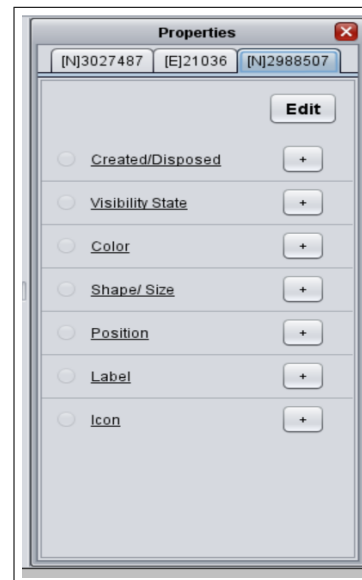
1. αν το δεξί κλικ γίνει πάνω σε κάποιον κόμβο, τότε ανοίγουν οι παρακάτω επιλογές. Η πρώτη επιλογή αφορά την διαγραφή του κόμβου (βλ. κεφ. 4). Στη συνέχεια κάποιος έχει την επιλογή είτε να επιλέξει όλες τις ακμές του κόμβου είτε να τις «ανοίξει» στο παράθυρο με τις ιδιότητες των αντικειμένων (βλ. κεφ. 2.5.2) και το ίδιο για όλες τις κατευθυνόμενες, μη-κατευθυνόμενες, έσω και έξω ακμές του κόμβου. Τέλος, ο χρήστης μπορεί να «ανοίξει» τον κόμβο στο παράθυρο με τις ιδιότητες των αντικειμένων.
2. αν το δεξί κλικ γίνει πάνω σε κάποια ακμή, τότε:
 - (i) αν η επιλεγμένη ακμή είναι πολλαπλή (π.χ. αν υπάρχουν περισσότερες από μια ακμές ανάμεσα σε δύο κόμβους), τότε μπορεί κάποιος να «ανοίξει» όλες τις ακμές, ή να επιλέξει κάποια από τις ακμές μέσα από ένα παράθυρο επιλογής. Αν επιλέξει κάποια ακμή, τότε για αυτήν ανοίγουν οι επιλογές που περιγράφονται στο 2(ii).
 - (ii) διαφορετικά οι επιλογές είναι δύο: είτε να διαγράψει κάποιος την ακμή (βλ. κεφ. 4), είτε να την «ανοίξει» στο παράθυρο με τις ιδιότητες των αντικειμένων.
3. αν το δεξί κλικ γίνει σε κάποιο άλλο σημείο του καμβά, τότε υπάρχει η επιλογή να επιλεγούν οι πολλαπλές ακμές και η επιλογή να προσθέσει κάποιος κόμβους ή ακμές στο γράφημα (βλ. κεφ. 4).



Σχήμα 14: Οι επιλογές που εμφανίζονται όταν γίνεται δεξί κλικ σε κόμβο, σε ακμή και στο γράφημα

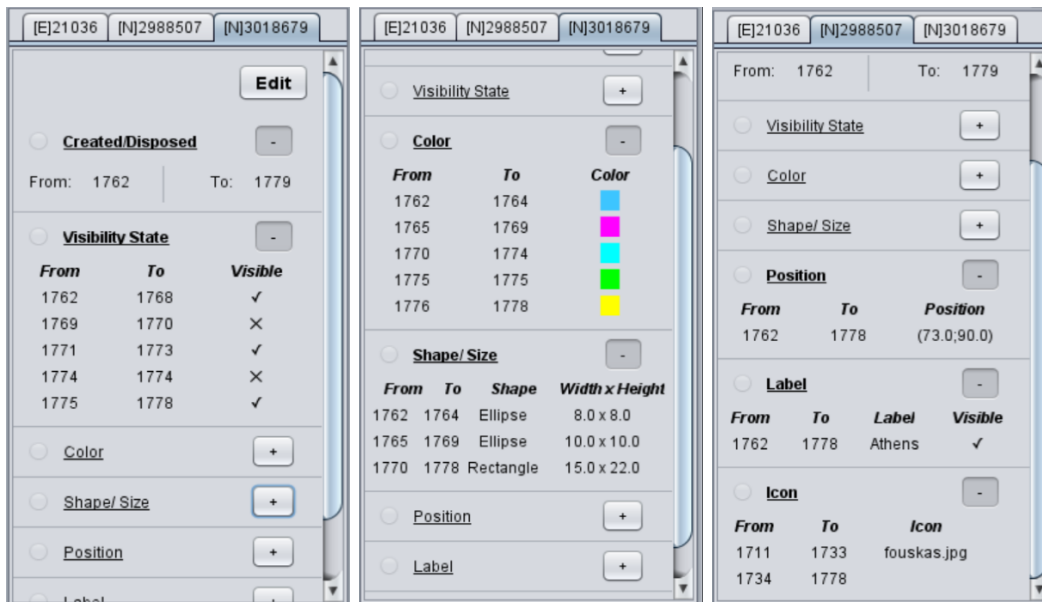
2.5.2 Παράθυρο με ιδιότητες των αντικειμένων

Πατώντας διπλό κλικ πάνω σε έναν κόμβο ή σε μία ακμή, ή πατώντας δεξί κλικ και στην συνέχεια επιλέγοντας την επιλογή Properties (Σχήμα 14), ανοίγει στο δεξί μέρος του UI ένα παράθυρο με όνομα "Properties", το οποίο περιέχει καρτέλες (tabs). Ο χρήστης μπορεί να μεγαλώσει ή να μικρύνει το πλάτος αυτού του παραθύρου σέρνοντας το αριστερά- δεξιά. Κάθε καρτέλα αντιστοιχεί σε ένα αντικείμενο, είτε κόμβο είτε ακμή, και σαν όνομα καρτέλας τοποθετείται το id του αντικειμένου μαζί με το γράμμα [E], αν πρόκειται για ακμή, ή το γράμμα [N], αν πρόκειται για κόμβο. Η καρτέλα αυτή περιλαμβάνει πληροφορίες για το αντικείμενο, δηλαδή αναγράφονται όλες οι τιμές για όλες τις ιδιότητες του αντικειμένου, αλλά και τις χρονικές στιγμές που αυτές «ισχύουν». Κάθε φορά που ο χρήστης πατάει διπλό κλικ πάνω σε ένα αντικείμενο (ή επιλέγει το Pro-

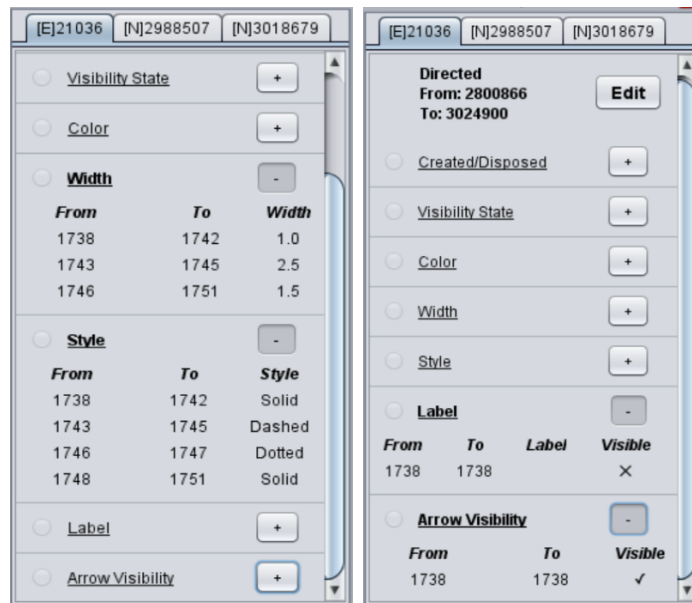


perties) προστίθεται μια καινούρια καρτέλα στο παράθυρο, η οποία αντιστοιχεί στο επιλεγμένο αντικείμενο. Αν το αντικείμενο υπάρχει ήδη σαν καρτέλα, τότε «ανοίγει» η συγκεκριμένη καρτέλα, δηλαδή επιλέγεται το αντικείμενο και εμφανίζονται οι ιδιότητες του. Από την άλλη, αν κάποιος πατήσει πάνω σε μία καρτέλα, τότε επιλέγεται το αντίστοιχο αντικείμενο και στο καμβά, δηλαδή εμφανίζεται το μπλέ περίβλημα αν πρόκειται για κόμβο ή το πορτοκαλί αν πρόκειται για ακμή. Αν θέλει να κλείσει κάποιος κάποια καρτέλα, αρκεί να πατήσει δεξί κλικ πάνω της, ενώ αν θέλει να κλείσει όλο το παράθυρο, αρκεί να πατήσει το κόκκινο x στην πάνω δεξιά γωνία.

Για να δει κάποιος τις τιμές που παίρνει μια ιδιότητα, αρκεί να πατήσει πάνω στο κουμπί με το +, το οποίο όταν είναι πατημένο παίρνει την τιμή -. Στα σχήματα 15 και 16 φαίνεται ο τρόπος με τον οποίο αναγράφονται οι ιδιότητες των αντικειμένων.



Σχήμα 15: Οι ιδιότητες ενός κόμβου



Σχήμα 16: Οι ιδιότητες μιας ακμής

2.5.3 Overview παράθυρο

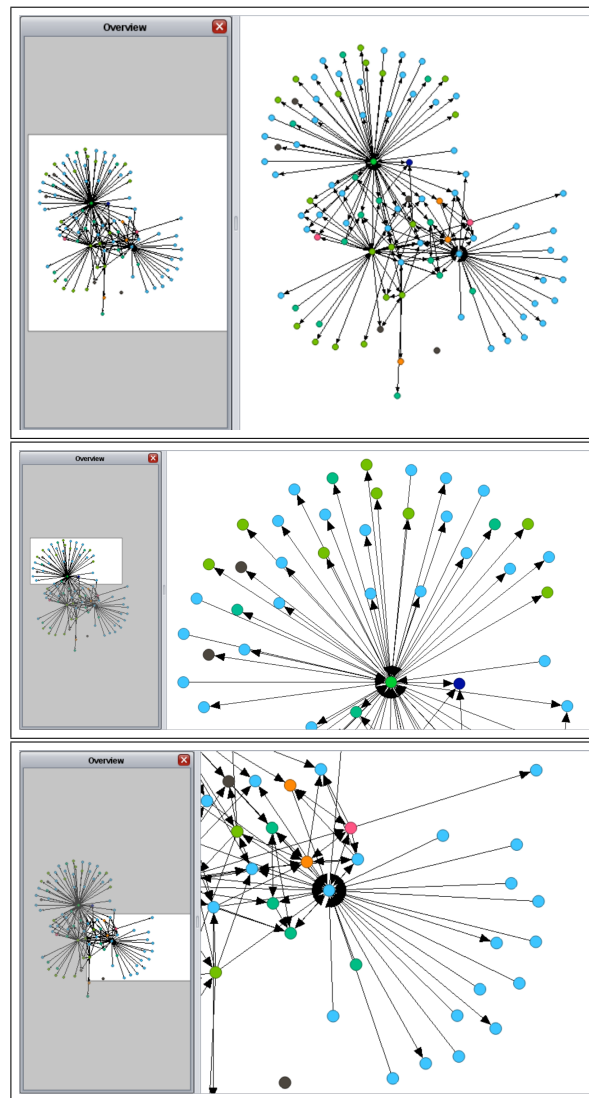


Στο μενού της εφαρμογής μπορεί να βρει κάποιος την επιλογή Overview. Αν την επιλέξει, τότε εμφανίζεται στο αριστερό μέρος του UI ένα καινούριο παράθυρο, το οποίο θα μπορούσε κανείς να παρομοιάσει με χάρτη. Πιο συγκεκριμένα, στο κέντρο του παραθύρου αυτού απεικονίζεται όλο το γράφημα που έχει ζωγραφιστεί εκείνη τη στιγμή στον καμβά, ανεξάρτητα από το αν έχει γίνει ζουμ ή από το αν έχει μετακινηθεί το γράφημα αριστερά- δεξιά ή πάνω- κάτω. Επίσης, τα μέρη του γραφήματος που είναι ορατά εμφανίζονται μέσα σε ένα άσπρο τετράγωνο, ενώ τα υπόλοιπα καλύπτονται από ένα διαφανές γκρι χρώμα.

Για την απεικόνιση του παραθύρου αυτού υπεύθυνη είναι η κλάση ViewingWindow, η οποία επεκτείνει κι αυτή την κλάση Drawer (βλ. κεφάλαιο 2.5.1). Μέσω του Overview παραθύρου, ο χρήστης μπορεί

- να «κεντράρει» το άσπρο τετράγωνο, δηλαδή το ορατό κομμάτι του γραφήματος, κάνοντας κλικ σε κάποιο σημείο,
- να το μεγαλώσει και να το μικρύνει, δηλαδή να κάνει ζουμ, κουνώντας τον τροχό του ποντικιού,

- ή να το σύρει, μεταφέροντας το.



Σχήμα 17: Στιγμιότυπα του Overview παράθυρου

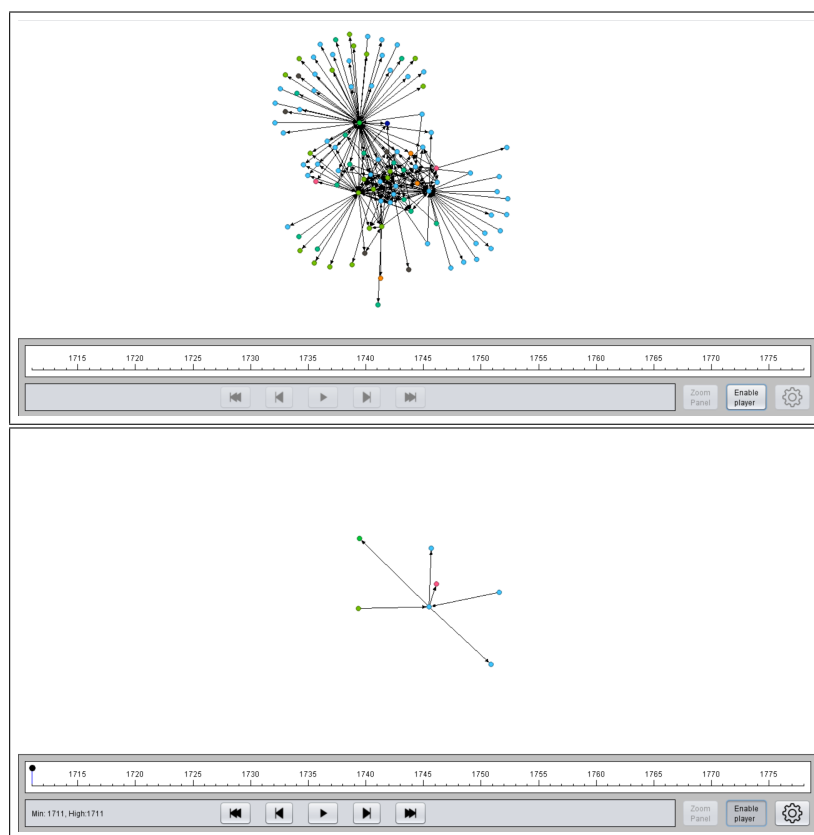
3 Απεικόνιση δυναμικών γραφημάτων

Στο κεφάλαιο αυτό θα περιγράψουμε την λειτουργία του λογισμικού που αφορά την απεικόνιση ενός δυναμικού γραφήματος. Η μετάβαση του δυναμικού γραφήματος από μία χρονική στιγμή στην επόμενη αφορά τις αλλαγές που πραγματοποιούνται στα αντικείμενα του, δηλαδή στους κόμβους και στις ακμές του. Οι αλλαγές αυτές μπορεί να αφορούν είτε τη ζωή των αντικειμένων είτε τις ιδιότητες που αφορούν την απεικόνιση τους, όπως αυτά περιγράφηκαν στο κεφάλαιο 2.2. Στη συγκεκριμένη διπλωματική μας ενδιαφέρει η μετάβαση του γραφήματος από τη μία χρονική στιγμή στην επόμενη να γίνεται με όμαλο τρόπο. Οι αλλαγές, δηλαδή, που πραγματοποιούνται στους κόμβους και στις ακμές γίνονται μέσα σε ένα χρονικό διάστημα και όχι ξαφνικά. Αν για παράδειγμα ένας κόμβος αλλάζει θέση, τότε η μεταβολή αυτή δεν θα γίνει ξαφνικά, αλλά ο κόμβος θα κινηθεί ανάμεσα στις δύο αυτές θέσεις ομαλά. Η απεικόνιση του δυναμικού γραφήματος φαίνεται, έτσι, σαν μια ταινία, όπου κάθε frame της αντιστοιχεί σε μια μικρή μεταβολή του γραφήματος, ώσπου η μετάβαση να ολοκληρωθεί και τα αντικείμενα του γραφήματος να πάρουν την τελική τους μορφή.

Όπως προαναφέρθηκε, όταν η λειτουργία της δυναμικής απεικόνισης δεν είναι ενεργοποιημένη απεικονίζονται στον «καμβά» του UI όλοι οι κόμβοι και όλες οι ακμές του δυναμικού γραφήματος, ανεξαρτήτως από το πότε είναι ορατοί και πότε όχι. Κατα τη λειτουργία της δυναμικής απεικόνισης, λοιπόν, απεικονίζεται στον καμβά η προβολή του δυναμικού γραφήματος G_i για μία χρονική στιγμή t_i , δηλαδή απεικονίζονται μόνο τα αντικείμενα του γραφήματος, τα οποία ζουν και είναι ορατά την t_i . Στο συγκεκριμένο κεφάλαιο θεωρούμε ότι μία χρονική στιγμή δεν αποτελείται απαραίτητα από έναν ακέραιο αριθμό ή μία ημερομηνία (βλ. κεφ. 2.1), αλλά μπορεί να είναι διάστημα αυτών, αφού στο λογισμικό έχουμε επιλέξει να υπάρχει η επιλογή της απεικόνισης για διάστημα ακεραίων (ή ημερολογιακών ημερών). Παράλληλα με την απεικόνιση του γραφήματος στον καμβά, υπάρχει μια αντιστοιχία με τον δρομέα που εμφανίζεται στην μπάρα του χρόνου, αφού ο δρομέας μας «δείχνει» την χρονική στιγμή απεικόνισης. Ο χρήστης έχει τη δυνατότητα είτε να «πλοηγηθεί» στις χρονικές στιγμές μετακινώντας τον δρομέα στην μπάρα του χρόνου είτε να παρακολουθήσει μια ταινία, η οποία αποτελείται από την μετάβαση του δυναμικού γραφήματος σε διαδοχικές χρονικές στιγμές. Όπως αναφέραμε στο κεφάλαιο 2.1, η αναπαράσταση του χρόνου ενός δυναμικού γραφήματος μπορεί να αφορά είτε ακέραιους αριθμούς είτε ημερομηνίες, αλλά για λόγους συντομίας θα αναφερόμαστε μόνο στους ακέραιους στην συνέχεια, αφού και στις δύο περιπτώσεις η εσωτερική αναπαράσταση είναι η ίδια.

Τη λειτουργία της απεικόνισης μπορεί να την ενεργοποιήσει κάποιος πατώντας το κουμπί

“Enable player” στο κάτω μέρος του UI. Αυτόματα, ενεργοποιούνται τα κουμπιά που αφορούν τις λειτουργίες βίντεο (βλ. κεφ. 3.1.1), αλλά και το κουμπί, με το οποίο μπορεί κάποιος να ρυθμίσει κάποιες παραμέτρους για την μετάβαση του δυναμικού γραφήματος. Επίσης, αυτόματα με το κουμπί Enable player εμφανίζεται στην μπάρα του χρόνου ο δρομέας, ο οποίος «δείχνει» στην πρώτη χρονική στιγμή του δυναμικού γραφήματος, ενώ στον καμβά εμφανίζεται το δυναμικό γράφημα στην πρώτη χρονική στιγμή (G_1). Ο χρήστης μπορεί να σύρει τον δρομέα αριστερά- δεξιά, έτσι ώστε να απεικονιστεί στον καμβά το γράφημα της αντίστοιχης χρονικής στιγμής. Τέλος, δίπλα από τα κουμπιά για τις λειτουργίες του βίντεο, αναγράφεται η χρονική στιγμή στην οποία βρίσκεται το βίντεο, είτε πρόκειται για διάστημα είτε όχι.



Σχήμα 18: Ο καμβάς του UI πριν και μετά την ενεργοποίηση της απεικόνισης του δυναμικού γραφήματος



Σχήμα 19: Το κάτω μέρος του UI πριν και μετά την ενεργοποίηση της απεικόνισης του δυναμικού γραφήματος

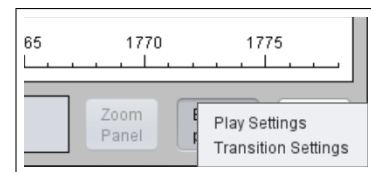
3.1 Μετάβαση Γραφήματος

Στο κεφάλαιο αυτό δεν θα ασχοληθούμε με την ομαλή μετάβαση του γραφήματος από μια χρονική στιγμή στην επόμενη, αλλά θα θεωρήσουμε την απεικόνιση του γραφήματος σαν διαδοχικές φωτογραφίες, όπου η κάθε φωτογραφία αντιστοιχεί σε μία χρονική στιγμή. Για την μετάβαση του δυναμικού γραφήματος στην επόμενη χρονική στιγμή, υπάρχουν τρεις παράμετροι:

1. το μήκος του διαστήματος (Interval Length),
2. η καθυστέρηση μετάβασης (Delay) και
3. το βήμα (Step).

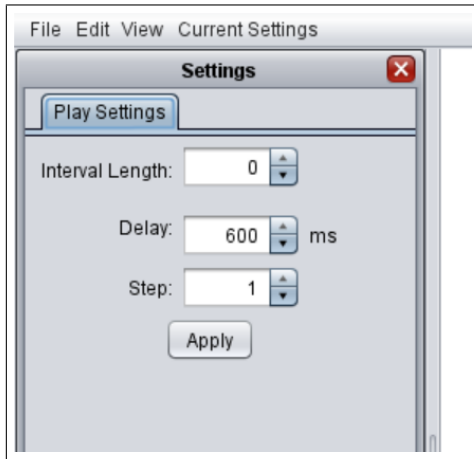
Οι παράμετροι Delay και Step χρησιμοποιούνται και από το Gephi [5] για την μετάβαση του δυναμικού γραφήματος.

Για να ρυθμίσει κάποιος αυτές τις παραμέτρους, αρκεί να επιλέξει τα Play Settings, από τις ρυθμίσεις στο κάτω δεξιά μέρος του UI. Τότε ανοίγει ένα παράθυρο στο αριστερό μέρος, όπου κάποιος μπορεί να επιλέξει τις τιμές που επιθυμεί για κάθε μία από τις τρεις παραμέτρους.



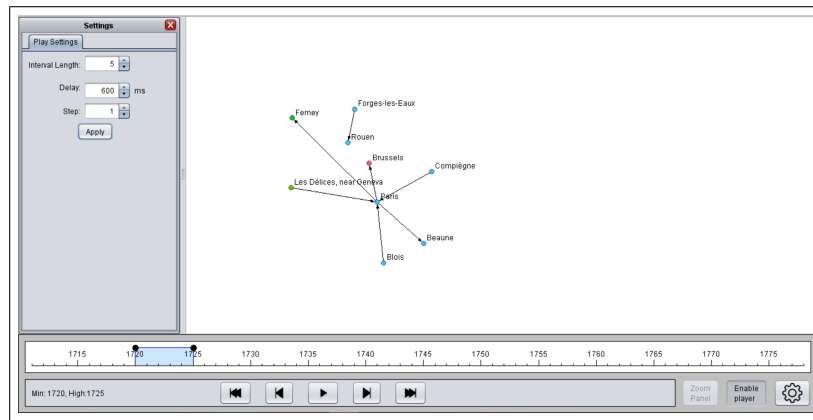
Η πρώτη παράμετρος αφορά το μήκος του διαστήματος, αφορά δηλαδή το πλήθος των ακεραίων που περιλαμβάνονται σε μία χρονική στιγμή της απεικόνισης. Η προεπιλεγμένη τιμή είναι το 0, οπότε κάθε χρονική στιγμή t_i περιλαμβάνει έναν ακεραίο. Αν το μήκος είναι 1, τότε κάθε χρονική στιγμή περιλαμβάνει δύο διαδοχικούς ακεραίους κοκ. Το διάστημα αυτό απεικονίζεται στην μπάρα του χρόνου με ένα μπλέ διαφανές ορθογώνιο παραλληλόγραμμο, το οποίο περιέχει όλους τους ακεραίους που συμπεριλαμβάνονται στο

διάστημα. Αν η χρονική στιγμή, λοιπόν, αφορά διάστημα ακεραίων, τότε το γράφημα που απεικονίζεται στον καμβά περιέχει όλους τους κόμβους και όλες τις ακμές που είναι



ορατοί τουλάχιστον για έναν ακέραιο που συμπεριλαμβάνεται στο διάστημα. Αν, δηλαδή, το μήκος του διαστήματος είναι ίσο με 0 και V_1, V_2, \dots, V_5 είναι τα σύνολα των κόμβων τις στιγμές t_1, t_2, \dots, t_5 του δυναμικού γραφήματος (τα V_1, V_2, \dots, V_5 έχουν στοιχεία από το σύνολο V των κόμβων του δυναμικού γραφήματος), τότε, αν το μήκος του διαστήματος γίνει 4, τότε το σύνολο των κόμβων $V_{[1,5]}$ που απεικονίζεται στον καμβά θα ταυτίζεται με την ένωση των V_1, V_2, \dots, V_5 , δηλαδή $V_{[1,5]} = V_1 \cup V_2 \cup \dots \cup V_5$. Οι κόμβοι, σ' αυτήν την περίπτωση, και αντίστοιχα και οι ακμές, απεικονίζονται με την τελευταία τιμή που παίρνει η κάθε ιδιότητα τους σ'

αυτό το διάστημα. Για παράδειγμα, αν η τρέχουσα απεικόνιση αφορά το διάστημα $[1, 5]$ και ένας κόμβος είναι μπλε στο διάστημα $[1, 3]$ και κίτρινος στο διάστημα $[3, 7]$, τότε ο κόμβος θα απεικονιστεί με κίτρινο χρώμα, δηλαδή θα απεικονιστεί με την τελευταία τιμή που παίρνει το χρώμα του μέσα στο διάστημα $[1, 5]$.



Σχήμα 20: Το γράφημα και η μπάρα του χρόνου για μήκος διαστήματος 5.

Η δεύτερη και η τρίτη παράμετρος αφορούν το «βίντεο» που αποτελείται από τις προβολές του δυναμικού γραφήματος σε διαδοχικές χρονικές στιγμές και τον τρόπο που γίνεται η μετάβαση από την μια χρονική στιγμή στην επόμενη. Η δεύτερη παράμετρος, δηλαδή η καθυστέρηση μετάβασης, αφορά τον χρόνο, όπου το βίντεο «παγώνει» σε κάθε χρονική στιγμή, δηλαδή σε κάθε γραφήμα G_i , προτού κάνει την μετάβαση στην

επόμενη. Η προεπιλεγμένη τιμή για την συγκεκριμένη παράμετρο είναι 600 ms. Από την άλλη, η τρίτη παράμετρος, δηλαδή το βήμα, αφορά το πλήθος των ακεραίων που μεσολαβούν ανάμεσα σε δύο χρονικές στιγμές της δυναμικής απεικόνισης. Δηλαδή, αν το βήμα είναι ίσο με 1 (step=1), τότε το «βίντεο» θα πάει στην αμέσως επόμενη χρονική στιγμή, ενώ αν το βήμα είναι 2 (step=2), τότε το βίντεο θα πάει π.χ. από την t_1 στην t_3 κοκ.

Παράδειγμα 3.1. Αν ένα δυναμικό γράφημα, λοιπόν, έχει χρονικό εύρος $[1; 50]$, και οι ρυθμίσεις των παραμέτρων είναι οι προεπιλεγμένες, τότε το σύνολο των χρονικών στιγμών είναι το $\{t_1 = 1, t_2 = 2, \dots, t_{50} = 50\}$, οπότε οι προβολές του δυναμικού γραφήματος είναι αντίστοιχα οι $\{G_1, G_2, \dots, G_{50}\}$. Η ροή του βίντεο έχει την μορφή:

$$G_1 \text{ για } 600 \text{ ms} - > G_2 \text{ για } 600 \text{ ms} - > \dots - > G_{50} \text{ για } 600 \text{ ms}$$

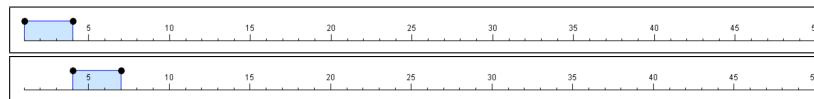
Παράδειγμα 3.2. Αν για το δυναμικό γράφημα του παραδείγματος 3.1 οι ρυθμίσεις που έχουν επιλεγεί είναι μήκος διαστήματος 3, 1 s για την καθυστέρηση μετάβασης και βήμα 1, τότε το σύνολο των χρονικών στιγμών είναι το $\{t_{[1,4]} = [1, 4], t_{[2,5]} = [2, 5], \dots, t_{[47,50]} = [47, 50]\}$, οπότε οι προβολές του δυναμικού γραφήματος είναι αντίστοιχα οι $\{G_{[1,4]}, G_{[2,5]}, \dots, G_{[47,50]}\}$. Η ροή του βίντεο έχει την μορφή:

$$G_{[1,4]} \text{ για } 1 \text{ s} - > G_{[2,5]} \text{ για } 1 \text{ s} - > \dots - > G_{[47,50]} \text{ για } 1 \text{ s}$$

Παράδειγμα 3.3. Αν για το δυναμικό γράφημα του παραδείγματος 3.1 οι ρυθμίσεις που έχουν επιλεγεί είναι μήκος διαστήματος 3, 1 s για την καθυστέρηση μετάβασης και βήμα 3, τότε το σύνολο των χρονικών στιγμών είναι το $\{t_{[1,4]} = [1, 4], t_{[4,7]} = [4, 7], \dots, t_{[46,49]} = [46, 49]\}$, οπότε οι προβολές του δυναμικού γραφήματος είναι αντίστοιχα οι $\{G_{[1,4]}, G_{[4,7]}, \dots, G_{[46,49]}\}$. Η ροή του βίντεο έχει την μορφή:

$$G_{[1,4]} \text{ για } 1 \text{ s} - > G_{[4,7]} \text{ για } 1 \text{ s} - > \dots - > G_{[46,49]} \text{ για } 1 \text{ s}$$

Ο δρομέας σε όλη τη διάρκεια του «βίντεο» ακολουθεί μια ομαλή πορεία πάνω στον βαθμονομημένο χάρακα, όπου την απόσταση ανάμεσα στα δύο pixel που αντιστοιχούν στις δύο χρονικές στιγμές την διανύει στο χρόνο της καθυστέρησης μετάβασης. Αν ο δρομέας είναι διάστημα, τότε σαν τρέχον pixel θεωρείται το πάνω άκρο του διαστήματος. Στο παράδειγμα 3.3, δηλαδή, η απόσταση που θα διανυθεί σε 1 δευτερόλεπτο είναι η απόσταση ανάμεσα στο σημείο 4 και 7 πάνω στον χάρακα (Σχήμα 21).



Σχήμα 21: Δύο διαδοχικές χρονικές στιγμές στην μπάρα του χρόνου για το παράδειγμα 3.3

3.1.1 Λειτουργίες Βίντεο

Οι λειτουργίες του βίντεο που υποστηρίζονται από το λογισμικό είναι 5:

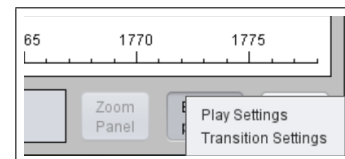
1. play/ pause
2. step back
3. step forward
4. go to start
5. go to end

Με το play/ pause ξεκινάει το βίντεο από την τρέχουσα χρονική στιγμή, ή γίνεται παύση του. Με το step back και step forward το βίντεο πηγαίνει ένα βήμα πίσω ή ένα βήμα μπροστά, όπου το βήμα είναι αυτό των ρυθμίσεων και δεν είναι απαραίτητα ίσο με 1. Τέλος, με το go to start και go to end το βίντεο πηγαίνει στην πρώτη και στην τελευταία στιγμή αντίστοιχα. Υπεύθυνα για αυτές τις πέντε λειτουργίες είναι τα πέντε κουμπιά που βρίσκονται κάτω από τη μπάρα του χρόνου (Σχήμα 19).

3.2 Ομαλή μεταβολή των αντικειμένων του γραφήματος

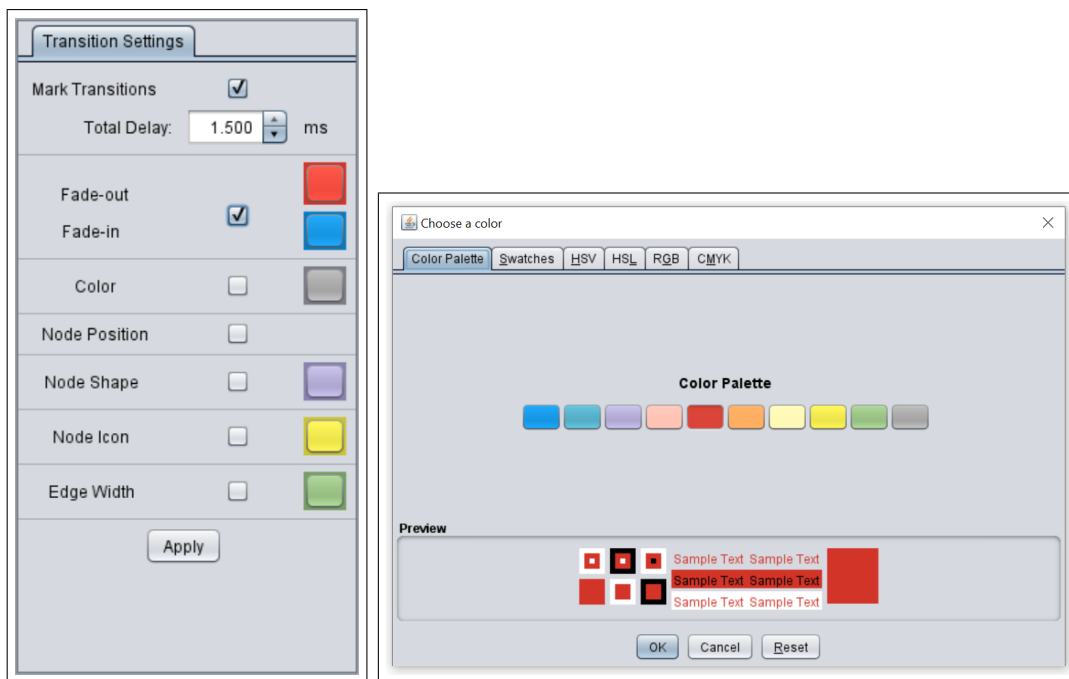
Μέχρι τώρα θεωρήσαμε την μετάβαση του δυναμικού γραφήματος σαν εναλλαγή από φωτογραφίες, όπου η κάθε φωτογραφία είναι η προβολή του δυναμικού γραφήματος σε μία χρονική στιγμή. Στο κεφάλαιο αυτό θα περιγράψουμε πώς γίνεται η μετάβαση αυτή ομαλά, δηλαδή τον τρόπο με τον οποίο απεικονίζονται οι αλλαγές που γίνονται στο δυναμικό γράφημα από μια χρονική στιγμή t_i στην επόμενη t_{i+1} , δηλαδή από την προβολή G_i στην G_{i+1} .

Στο κεφάλαιο 3.1 αναφέραμε ότι η καθυστέρηση μετάβασης καθορίζεται από την παράμετρο delay (play delay). Σ' αυτό το κεφάλαιο εισάγουμε μια καινούρια παράμετρο, την transition delay. Η παράμετρος αυτή καθορίζει τον χρόνο, κατά τον οποίο οι κόμβοι και οι ακμές μεταβάλλονται ομαλά ώστε να φτάσουν στην τελική τους κατάσταση. Η πλήρης μετάβαση, λοιπόν, από μια χρονική στιγμή στην επόμενη χρειάζεται (play delay+transition delay) χρόνο για να ολοκληρωθεί, όπου ο χρόνος που η ταινία «παγώνει» σε μία χρονική στιγμή προηγείται του χρόνου μετάβασης. Τις ρυθμίσεις για την μετάβαση του γραφήματος μπορεί να τις βρει κάποιος κάτω δεξιά στο κουμπί με το γράναζι, όπου υπάρχουν οι επιλογές Play Settings και Transition Settings. Αν για την μετάβαση του δυναμικού γραφήματος από τη χρονική στιγμή t_i στην χρονική στιγμή t_{i+1} ο δρομέας πρέπει να κινηθεί



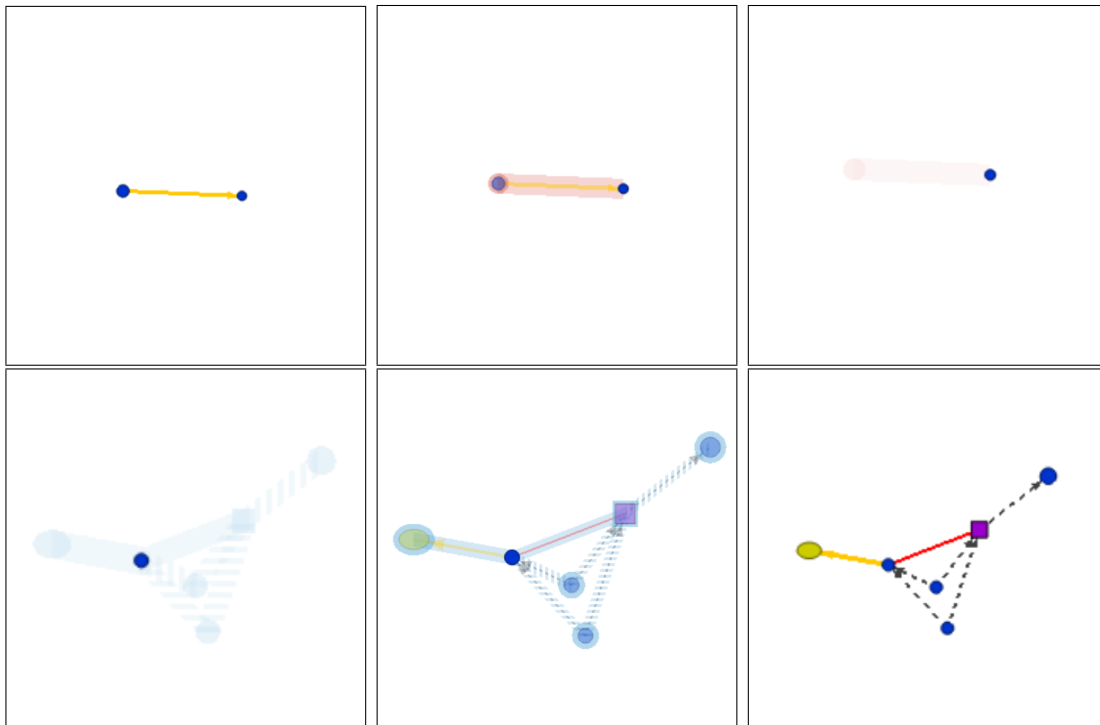
από το σημείο P_i στο σημείο P_{i+1} του βαθμονομημένου χάρακα, τότε η απόσταση αυτή χωρίζεται στα δύο, όπου το πρώτο μέρος αντιστοιχεί στην καθυστέρηση play delay και το δεύτερο μέρος αντιστοιχεί στην καθυστέρηση transition delay. Αν, για παράδειγμα οι παραμέτροι παίρνουν τις τιμές 1 και 2 δευτερόλεπτα αντίστοιχα, τότε το 1/3 της απόστασης $P_i P_{i+1}$ αφιερώνεται στο play delay και τα άλλα 2/3 αφιερώνονται στο transition delay.

Όσον αφορά τον τρόπο που γίνεται αυτή η ομαλή μετάβαση, όταν στο δυναμικό γράφημα προστίθενται ή αφαιρούνται κόμβοι και ακμές ή όταν αλλάζει κάποια ιδιότητα τους τότε τα αντικείμενα αυτά περιβάλλονται από ένα φωτοστέφανο (halo), του οποίου το χρώμα μπορεί κάποιος να το επιλέξει από τις ρυθμίσεις. Ο χρήστης μπορεί, επίσης, να επιλέξει για ποιες από τις ιδιότητες των αντικειμένων θα εμφανιστεί το halo και θα γίνει η μετάβαση ομαλά. Για τις ιδιότητες που δεν είναι ενεργοποιημένη αυτή η λειτουργία, η απεικόνιση του κόμβου ή της ακμής αλλάζει απότομα. Αν, για παράδειγμα, ένας κόμβος αλλάξει σχήμα από έλλειψη σε ορθογώνιο παραλληλόγραμμο και η αντίστοιχη λειτουργία δεν είναι ενεργοποιημένη, τότε ο κόμβος θα αλλάξει σχήμα απότομα στη μέση του transition delay.

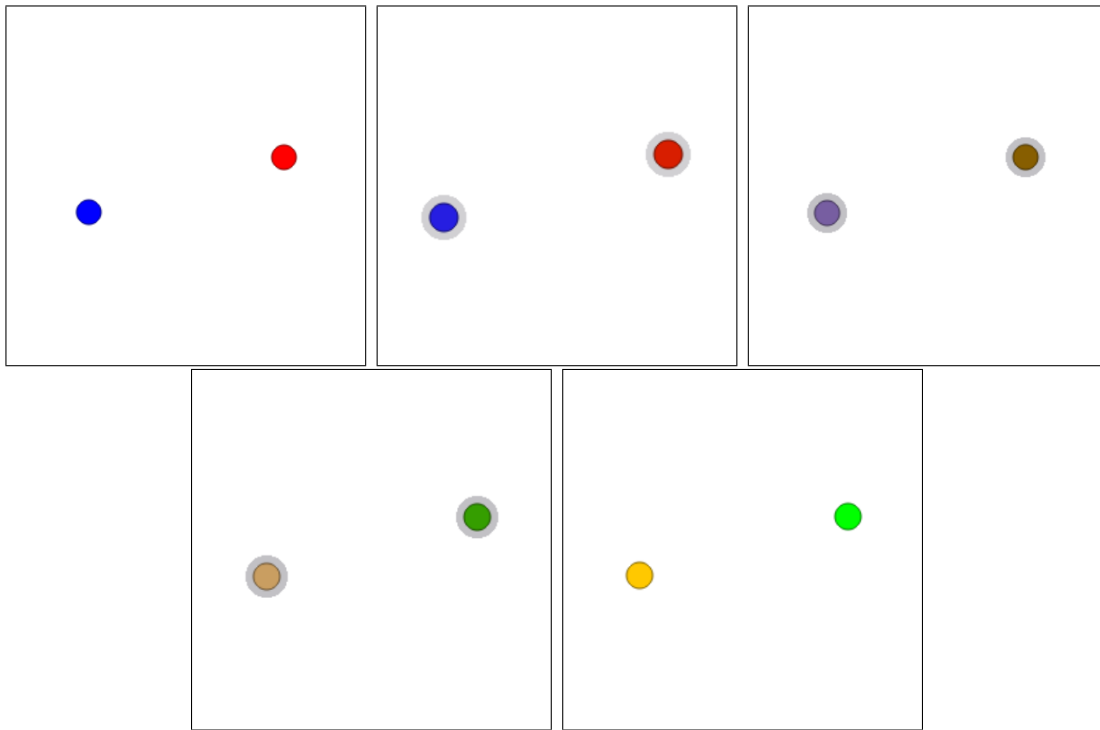


Σχήμα 22: Οι ρυθμίσεις για την ομαλή μετάβαση του γραφήματος και η παλέτα χρωμάτων που προτείνεται

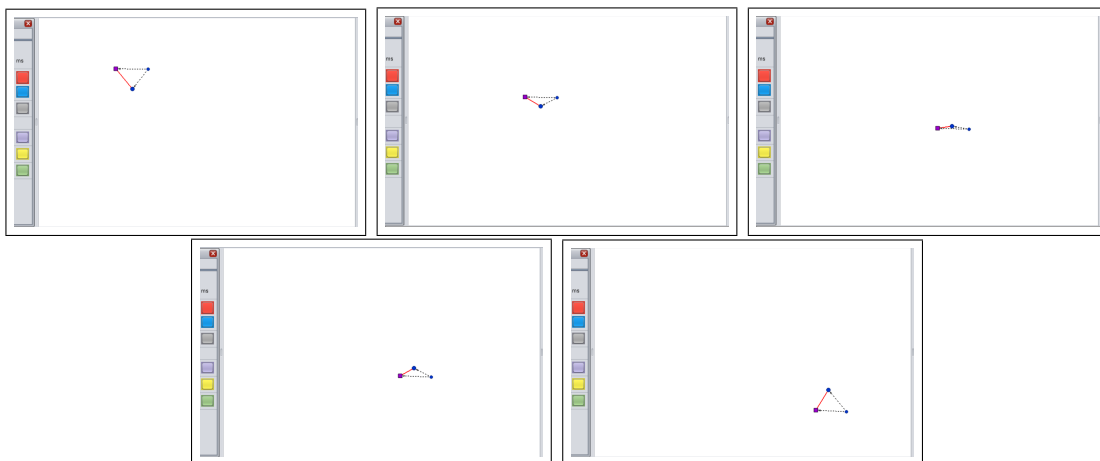
Διαγραφή- Εισαγωγή Αντικειμένων (Fade-out/Fade-in): Κατά την μετάβαση από το G_i στο G_{i+1} αφαιρούνται κάποιοι κόμβοι και ακμές και προστίθενται κάποιοι νέοι. Στην περίπτωση αυτή η καθυστέρηση μετάβασης (transition delay) χωρίζεται στη μέση. Στο πρώτο μισό οι κόμβοι και οι ακμές που αφαιρούνται από το G_i περικλείονται με ένα κόκκινο halo και αποχρωματίζονται ομαλά μέχρις ότου εξαφανιστούν. Στο δεύτερο μισό οι κόμβοι και οι ακμές που προστίθεται στο G_{i+1} περικλείονται επίσης με ένα μπλε halo και χρωματίζονται ομαλά, δηλαδή γίνονται ολοένα και πιο αδιαφανή, μέχρις ότου εμφανιστούν πλήρως στην τελική τους μορφή. Τα χρώματα των halo είναι προεπιλεγμένα, αλλά ο χρήστης μπορεί να τα αλλάξει μέσω των ρυθμίσεων.



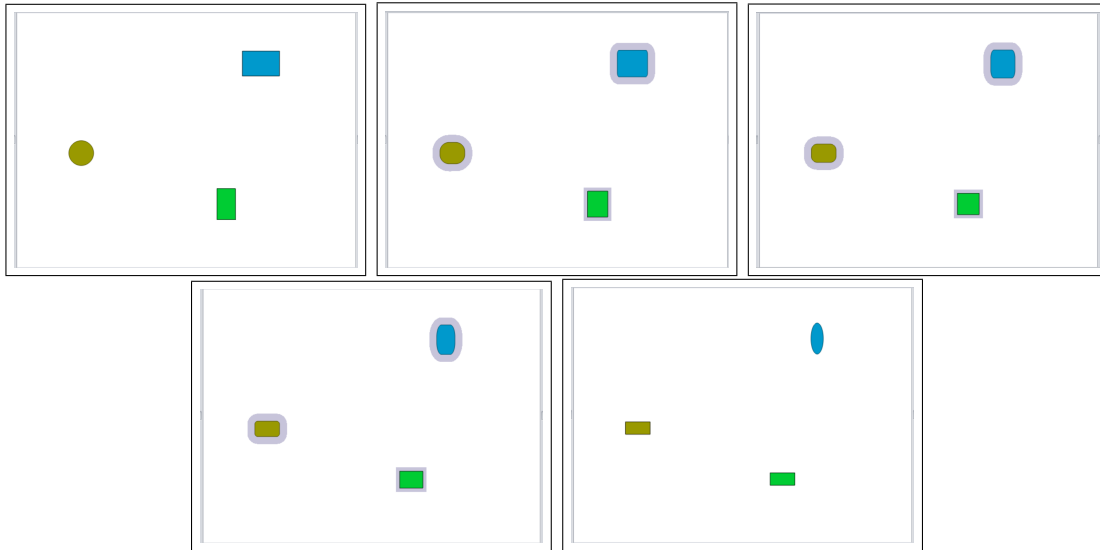
Αλλαγή χρώματος (Color): Η μετάβαση του χρώματος ενός αντικειμένου γίνεται καθ' όλη την διάρκεια του transition delay και το αντικείμενο περικλείεται με ένα γκρι halo, ενώ το χρώμα του αντικειμένου μεταβάλλεται ομαλά μέχρις ότου πάρει την τελική του μορφή.



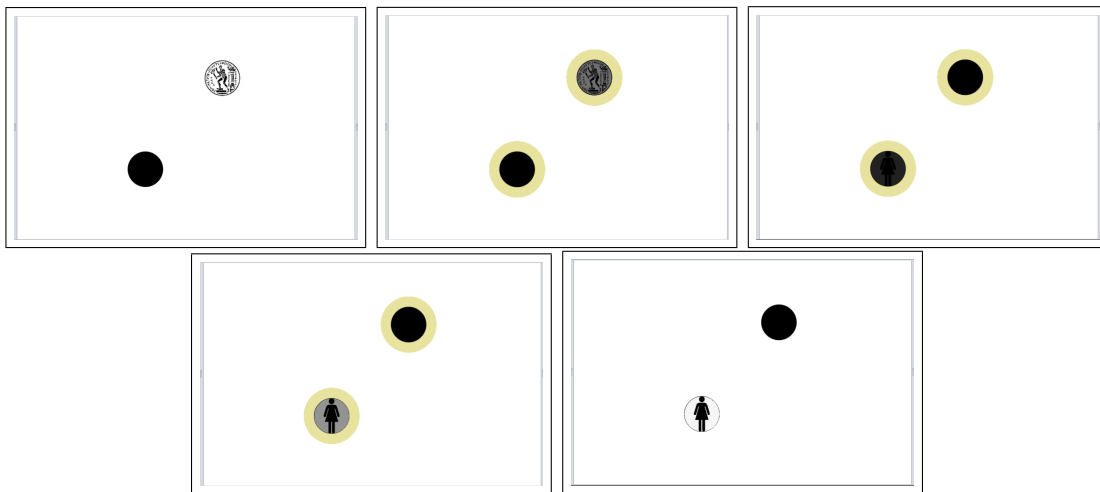
Αλλαγή θέσης κόμβου (Node Position): Όταν αλλάζει η θέση ενός κόμβου, αυτός δεν περικλείεται με halo, αλλά κινείται ομαλά από την μία θέση στην άλλη.



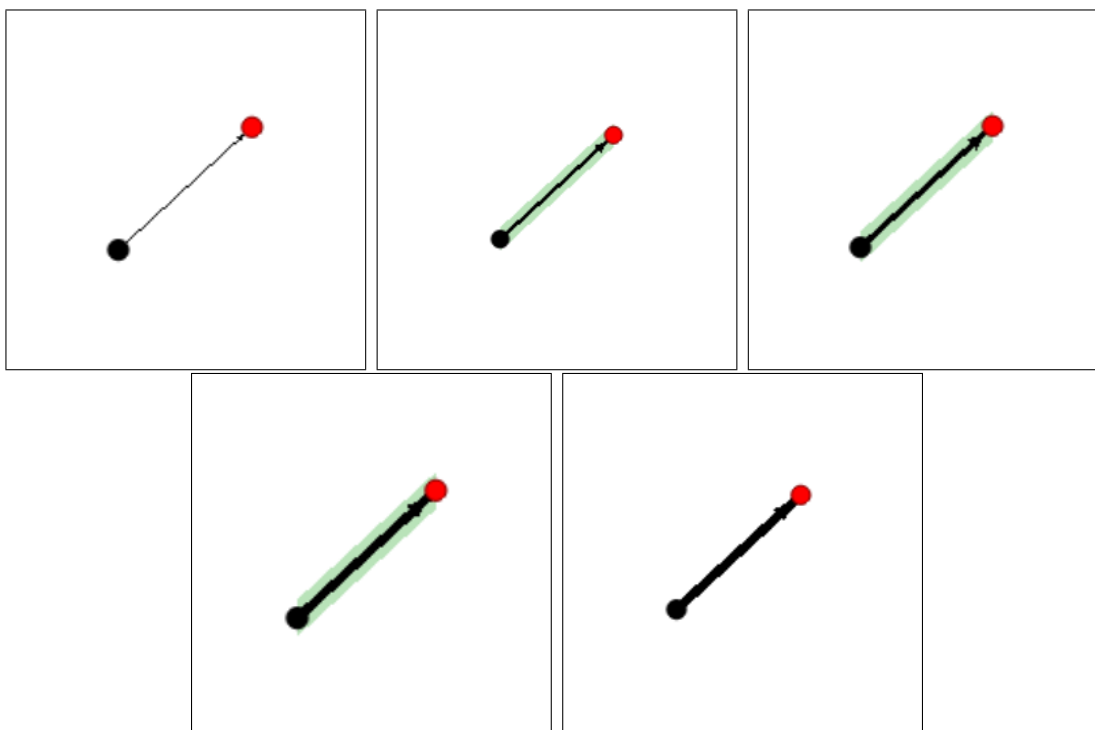
Αλλαγή σχήματος κόμβου (Node Shape): Κατά τη μετάβαση του σχήματος ενός κόμβου το αντικείμενο περικλείεται με ένα μαβ halo, ενώ το σχήμα του αντικειμένου μεταβάλλεται ομαλά μέχρις ότου πάρει την τελική του μορφή.



Αλλαγή φωτογραφίας κόμβου (Node Icon): Το halo στην συγκεκριμένη περίπτωση είναι κίτρινο και η φωτογραφία του κόμβου σβήνει ή εμφανίζεται.



Αλλαγή πάχους ακμής (Edge Width): Τέλος, όταν αυξάνεται ή μειώνεται το πάχος μιας ακμής, εμφανίζεται ένα πράσινο halo γύρω της και εκείνη αλλάζει μέγεθος ομαλά.



3.3 Πυκνό χρονικό εύρος- Zoom Panel

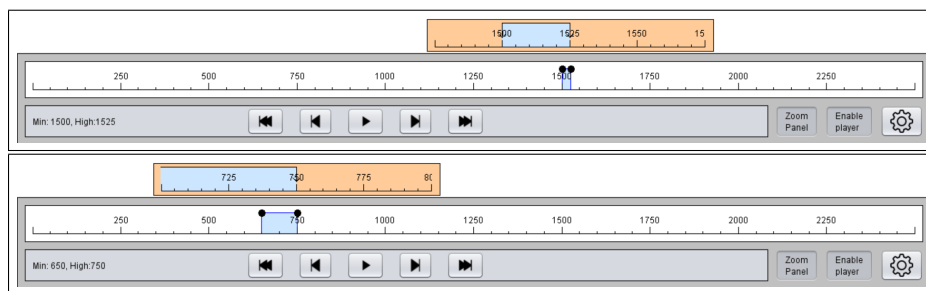
Όταν το εύρος ενός δυναμικού γραφήματος είναι πολύ πυκνό, δηλαδή περιέχει πολλές χρονικές στιγμές, τότε με την ενεργοποίηση της δυναμικής απεικόνισης ενεργοποιείται δίπλα στο κουμπί “Enable Player” ένα ακόμα κουμπί, το Zoom Panel. Με την ενεργοποίηση αυτού εμφανίζεται ακριβώς πάνω από τον δρομέα μία (πιο μικρή σε μήκος από την κανονική) μπάρα χρόνου, η οποία αποτελεί μία μεγέθυνση του (μεγάλου) βαθμονομημένου χάρακα.

Η καινούρια αυτή μπάρα του χρόνου αποτελείται από έναν βαθμονομημένο χάρακα 100 ακεραίων (ή ημερολογιακών ημερών). Στο μέσο της είναι ζωγραφισμένο ένα βέλος, το οποίο, αν η τρέχουσα χρονική στιγμή δεν είναι διάστημα, τότε «δείχνει» στην τρέχουσα χρονική στιγμή, ενώ διαφορετικά είναι το πάνω άκρο ενός ορθογώνιου παραλληλόγραμμου, το οποίο απεικονίζει το διάστημα της χρονικής στιγμής. Η διαφορά με την κύρια μπάρα είναι ότι με την πάροδο του χρόνου δεν κινείται ο δρομέας, αλλά κινείται το φόντο της. Όταν, λοιπόν, παίζει το «βίντεο» της δυναμικής απεικόνισης μένει σταθερό το βέλος στο μέσο της μπάρας και κινείται ο βαθμονομημένος χάρακας που βρίσκεται από πίσω. Μία δεύτερη κίνηση της μπάρας αυτής είναι ότι καθώς κινείται ο κύριος δρο-

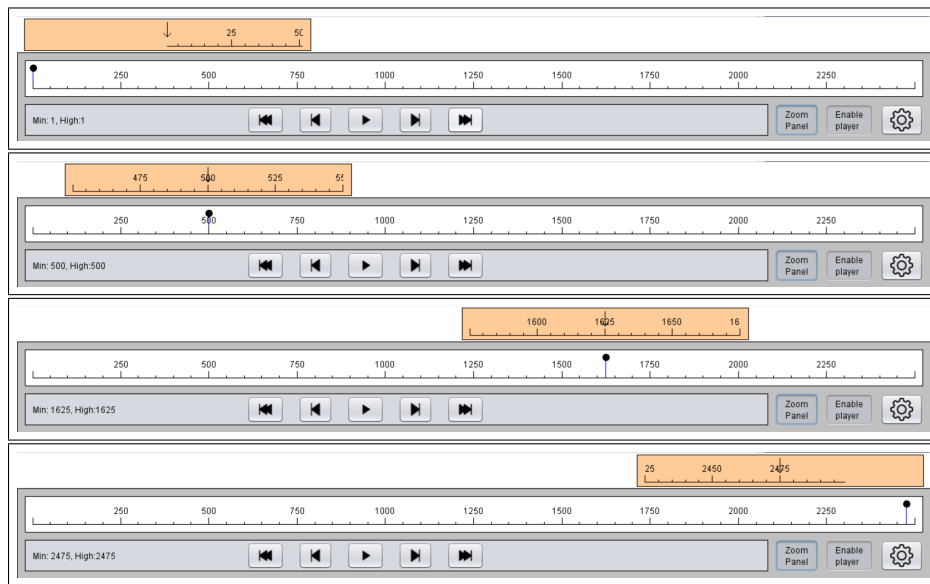
μέας, κινείται και η ίδια, ώστε το βέλος, δηλαδή το μέσο της, να βρίσκεται ακριβώς πάνω από τον δρομέα. Όταν ο δρομέας βρίσκεται πολύ αριστερά ή πολύ δεξιά, το βέλος της μπάρας είναι αδύνατο να βρίσκεται ακριβώς από πάνω του, οπότε η μπάρα του χρόνου βρίσκεται στο αριστερά ή στο δεξιά μέρος του παραθύρου.

Η κλάση που είναι υπεύθυνη για την καινούρια μπάρα του χρόνου είναι η `ZoomPanel`. Ο βαθμονομημένος χάρακας είναι αντικείμενο της κλάσης `TimelineDrawer`, όπως και ο χάρακας της κύριας μπάρας. Η κλάση `TimelineDrawer` δημιουργεί μια εικόνα με έναν βαθμονομημένο χάρακα με $100 + step$ ακεραίους, όπου `step` το βήμα της απεικόνισης. Όταν το `play` δεν είναι ενεργοποιημένο, δηλαδή το «βίντεο» δεν παίζει, τότε η `ZoomPanel` απεικονίζει μόνο τους πρώτους 100 ακεραίους, με την τρέχουσα στιγμή να είναι στο μέσο της. Όταν το «βίντεο» ξεκινήσει να παίζει, τότε για μία μετάβαση του δυναμικού γραφήματος, η `ZoomPanel` κόβει αυτή την εικόνα σε επόμενα pixel και την απεικονίζει. Αν, για παράδειγμα, το μήκος του `Zoom Panel` είναι 400 pixel και το μήκος της εικόνας με τους $100 + step$ ακεραίους είναι 420 pixel, τότε η κλάση `ZoomPanel` θα κόψει την φωτογραφία πρώτα στο [1,401], μετά [2, 402], μετά [3,403], ..., [20,420] και θα απεικονίσει διαδοχικά κάθε κομμάτι της εικόνας στην μικρή μπάρα. Όταν η μετάβαση ολοκληρωθεί, στην μπάρα θα απεικονίζεται το τελευταίο κομμάτι της εικόνας και η κλάση `TimelineDrawer` θα επιστρέψει την καινούρια εικόνα με τους επόμενους $100 + step$ ακεραίους.

Μέσω του `zoom Panel`, ο χρήστης μπορεί να πλοηγηθεί στις χρονικές στιγμές του γραφήματος μετακινώντας τον χάρακα αριστερά- δεξιά. Επίσης, αν μετακινήσει τον κύριο δρομέα, τότε αυτόματα μετακινείται και το `zoom Panel`, οπότε αν ο χρήστης θελήσει να δει το δυναμικό γράφημα για κάποια χρονική στιγμή, αρκεί να σύρει τον κύριο δρομέα σ' αυτήν την χρονική στιγμή και μετά να «πετύχει» ακριβώς την στιγμή σέρνοντας τον χάρακα του `zoom Panel`.



Σχήμα 23: Το `Zoom Panel` όταν η χρονική στιγμή αφορά διάστημα μήκους 25 και 100



Σχήμα 24: Η θέση του Zoom Panel σε σχέση με τον δρομέα

3.4 Συγχρονισμός κινήσεων- Multithreading

Μέχρι στιγμής, λοιπόν, έχουμε περιγράψει κάποιες κινήσεις, οι οποίες πραγματοποιούνται στο UI και απαιτούνται συγχρονισμό μεταξύ τους. Οι κινήσεις αυτές είναι:

- η κίνηση του δρομέα, ο οποίος πρέπει να κινείται ομαλά,
- η εναλλαγή των γραφήματων στον καμβά, η οποία πρέπει να συγχρονίζεται με την άφιξη του δρομέα στο pixel της τρέχουσας χρονικής στιγμής,
- η μετάβαση του γραφήματος από μια χρονική στιγμή στην επόμενη που περιλαμβάνει το fade-in/fade-out και την ομαλή μεταβολή των αντικειμένων και
- η κίνηση του zoom Panel, σε κάποιες περιπτώσεις.

Ο συγχρονισμός αυτός επιτυγχάνεται μέσω του Multithreading. Για τον συγχρονισμό όλων των Threads που εκτελούν τις παραπάνω διεργασίες υπεύθυνη είναι η κλάση Player. Μεταξύ όλων των Threads που θα αναφέρουμε παρακάτω, «ενεργό» είναι και το main Thread της Java, όπου, κατά κανόνα, όλες οι εντολές που εφαρμόζουν κάποια αλλαγή στο UI πρέπει να καλούνται από αυτό. Η απεικόνιση του δυναμικού γραφήματος, λοιπόν, εκτελείται από τα παρακάτω Threads:

1. **Calculator:** Για την απεικόνιση του δυναμικού γραφήματος πρέπει να υπολογίζονται οι προβολές του, G_i , έτσι ώστε να απεικονιστούν στον καμβά. Υπεύθυνο για τον υπολογισμό αυτό είναι το Thread Calculator, το οποίο γεμίζει μια ουρά προτεραιότητας με τις προβολές που θα απεικονιστούν στη συνέχεια. Η ουρά αυτή είναι 20 θέσεων και κάθε φορά που «αδειάζει» μια θέση, δηλαδή όποτε απεικονίζεται το πρώτο G_i της ουράς, καλείται το Thread, ώστε να υπολογίσει ένα νέο και να το τοποθετήσει στο τέλος της ουράς.
2. **Transition Calculator:** Όταν είναι ενεργοποιημένη η ομαλή μετάβαση του γραφήματος για κάποιες ιδιότητες των αντικειμένων, τότε πρέπει να υπολογιστούν οι κόμβοι και οι ακμές για τους οποίους υπάρχει μεταβολή στις συγκεκριμένες ιδιότητες. Για τον υπολογισμό αυτό υπεύθυνο είναι το Thread Transition Calculator, το οποίο για κάθε μία ιδιότητα «γεμίζει» μια ουρά προτεραιότητας 20 θέσεων, όπου η ουρά προτεραιότητας περιέχει λίστες από αντικείμενα. Έτσι, η ουρά προτεραιότητας για μια ιδιότητα περιέχει για τις επόμενες 20 χρονικές στιγμές τα αντικείμενα, των οποίων μεταβάλλεται η συγκεκριμένη ιδιότητα. Το Thread αυτό χρησιμοποιεί για τους υπολογισμούς του το Calculator Thread, δηλαδή χρησιμοποιεί τις προβολές του δυναμικού γραφήματος για τις επόμενες 20 χρονικές στιγμές. Όταν το Calculator Thread υπολογίζει ένα νέο γράφημα, τότε ειδοποιεί το Transition Calculator Thread, ώστε να υπολογίσει τις νέες λίστες με αντικείμενα και να τις εισάγει στις ουρές προτεραιότητας.
3. **Fade-out/in painter:** Το Thread αυτό είναι υπεύθυνο για την απεικόνιση των αντικειμένων που κάνουν fade-out και fade-in. Χρησιμοποιεί την πρώτη λίστα από την αντίστοιχη ουρά προτεραιότητας που έχει υπολογιστεί από το Transition Calculator Thread και είναι αυτό που κάνει τα αντικείμενα πιο διαφανή ή πιο έντονα και που δημιουργεί τα halo γύρω από τα αντικείμενα. Όταν πρόκειται για το fade-out, το πρώτο 20% του χρόνου κάνει fade-in το halo, για το επόμενο 60% του χρόνου κάνει fade-out το αντικείμενο με το halo, χωρίς να εξαφανίζεται πλήρως το halo, και για το υπόλοιπο 20% εξαφανίζεται το halo. Όμοια συμβαίνει και με το fade-in. Αυτό συμβαίνει ώστε να μην είναι ούτε η εμφάνιση και η εξαφάνιση των halos έντονη για το μάτι και η μετάβαση να γίνεται εξ' ολοκλήρου ομαλά.
4. **Transition painter:** Οι μεταβολές των υπόλοιπων ιδιοτήτων των αντικειμένων ρυθμίζονται από αυτό το Thread. Για κάθε ιδιότητα που είναι ενεργοποιημένη, χρησιμοποιείται η αντίστοιχη λίστα αντικειμένων από την ουρά προτεραιότητας. Έτσι, οι ιδιότητες των αντικειμένων μεταβάλλονται σιγά-σιγά μέχρι να γίνει η πλήρης μετάβαση.

5. **Slider:** Αυτό το Thread είναι υπεύθυνο για την κίνηση του δρομέα, για την απεικόνιση του γραφήματος, αλλά και για τον συγχρονισμό όλων των υπόλοιπων Threads, οπότε είναι και το πιο σημαντικό. Αποτελείται από ένα loop, όπου η κάθε επανάληψη είναι η κίνηση του δρομέα για μία μετάβαση του δυναμικού γραφήματος.

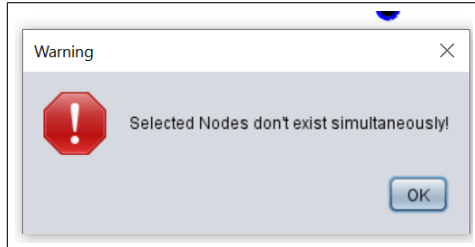
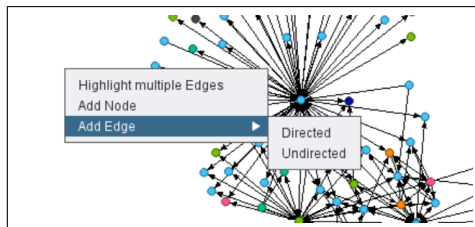
- Αν η λειτουργία της ομαλής μετάβασης δεν είναι ενεργοποιημένη, τότε ο δρομέας κινείται από το αρχικό σημείο στο τελικό, όπου ολοκληρώνεται η μετάβαση και, μέσω του main Thread, απεικονίζεται το «νέο» γράφημα, δηλαδή η προβολή του δυναμικού γραφήματος τη νέα χρονική στιγμή. Όταν απεικονίζεται το νέο γράφημα, αφαιρείται από την ουρά προτεραιότητας ένα στοιχείο, επομένως καλείται το αντίστοιχο Thread για τον υπολογισμό. Το loop ξεκινάει από την αρχή για την καινούρια, τώρα χρονική στιγμή.
- Από την άλλη, αν η λειτουργία της ομαλής μετάβασης είναι ενεργοποιημένη, το Thread εκτελεί τις εξής εργασίες:
 - (α') ο δρομέας κινείται για την απόσταση που αναλογεί στο play delay σε σχέση με την ολική καθυστέρηση
 - (β') καλούνται τα δύο Threads που είναι υπεύθυνα για την απεικόνιση του fade-out και των μεταβάσεων των ιδιοτήτων των αντικειμένων
 - (γ') ο δρομέας συνεχίζει την κίνηση του για την απόσταση που αναλογεί στο μισό του transition delay σε σχέση με την ολική καθυστέρηση
 - (δ') καλείται και πάλι το Thread για την απεικόνιση του fade-in
 - (ε') ο δρομέας ολοκληρώνει την απόσταση ανάμεσα στα δύο σημεία.Τα (β')-(γ') και (δ')-(ε') εκτελούνται ταυτόχρονα.

6. **Zoom Panel:** Οι δύο κινήσεις του Zoom Panel που περιγράφηκαν στο κεφάλαιο 3.3 εκτελούνται από ένα καινούριο Thread. Το Thread αυτό είναι υπεύθυνο για την αντικατάσταση του φόντου της μπάρας του χρόνου, έτσι ώστε η κίνηση του χάρακα να φαίνεται ομαλή. Επίσης, είναι υπεύθυνο για την τοποθέτηση της μπάρας του χρόνου πάνω από τον κύριο δρομέα, έτσι ώστε η κίνηση της να ταυτίζεται με την κίνηση του δρομέα.

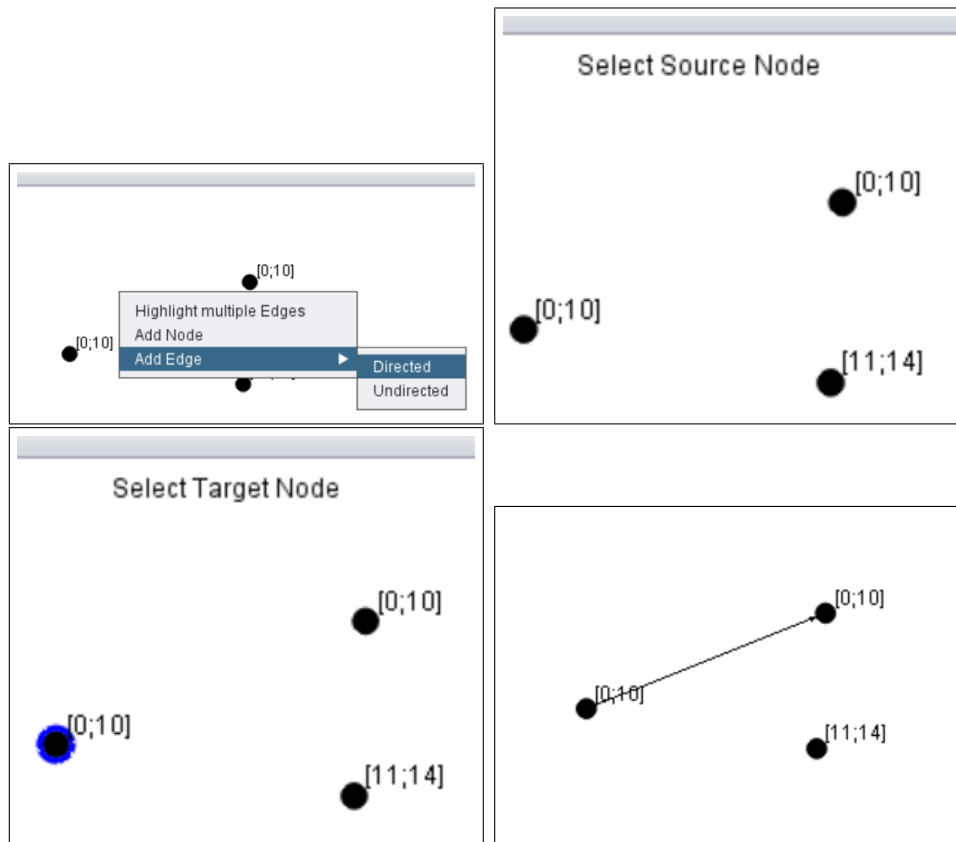
4 Δημιουργία δυναμικών γραφημάτων (Dynamic Graph Editing)

Σ' αυτό το κεφάλαιο θα περιγράψουμε τον τρόπο με τον οποίο μπορεί κάποιος να εισάγει και να επεξεργαστεί κόμβους και ακμές. Όταν εισάγεται ένα αντικείμενο στο δυναμικό γράφημα, τότε κάθε ιδιότητα του αποτελείται από μία τιμή, η οποία ισχύει για όλο το εύρος ζωής του. Ο χρήστης μπορεί να επιλέξει ποια τιμή θα πάρει η κάθε ιδιότητα του αντικειμένου κατά την εισαγωγή, αλλά και να επεξεργαστεί το εύρος ζωής του, το πότε γίνεται ορατό και πότε όχι, αλλά και τις ιδιότητες που αφορούν την απεικόνιση του, όπως είναι το χρώμα, το σχήμα, το πάχος κλπ.

4.1 Προσθήκη Κόμβου και Ακμής



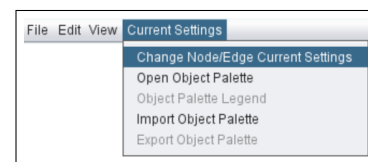
Η προσθήκη κόμβου και ακμής γίνεται με δεξί κλικ πάνω στον καμβά. Και τα δύο αντικείμενα προστίθενται στο γράφημα με τις προεπιλεγμένες ιδιότητες, τις οποίες ο χρήστης μπορεί να επεξεργαστεί μέσω των αντίστοιχων ρυθμίσεων (Κεφάλαιο 4.1.1). Η εισαγωγή κόμβου γίνεται στη σημείο που γίνεται το δεξί κλικ, ενώ για την προσθήκη ακμής στο δυναμικό γράφημα, ο χρήστης καλείται να επιλέξει αν η ακμή θα είναι κατευθυνόμενη ή όχι. Όπως περιγράψαμε στο κεφάλαιο 2.2 η ιδιότητα αυτή της ακμής δεν είναι δυναμική και δεν αλλάζει με το χρόνο. Στην συνέχεια ο χρήστης καλείται να επιλέξει την αφετηρία και το τέλος τις ακμής, δηλαδή τους δύο κόμβους που αποτελούν άκρα της ακμής. Κατά την προσθήκη μια ακμής στο δυναμικό γράφημα, πρέπει οι δύο κόμβοι (source και target) να ζουν ταυτόχρονα, δηλαδή να υπάρχει τουλάχιστον μια κοινή χρονική στιγμή στο εύρος ζωής των κόμβων. Αν αυτό δεν ισχύει, τότε η ακμή δεν προστίθεται στο δυναμικό γράφημα και εμφανίζεται προειδοποιητικό μήνυμα.



Σχήμα 25: Η προσθήκη ακμής ανάμεσα σε δύο κόμβους που ζουν στο διάστημα $[0;10]$

4.1.1 Αλλαγή προεπιλεγμένων ρυθμίσεων

Στο Μενού της εφαρμογής μπορεί να βρει κάποιος την επιλογή `Change Node\Edge Current Settings`. Μ' αυτή την επιλογή ανοίγει ένα παράθυρο στο αριστερό μέρος της εφαρμογής, όπου κάποιος μπορεί να αλλάξει τις προεπιλεγμένες ιδιότητες των αντικειμένων που προστίθενται. Οι ιδιότητες αυτές ισχύουν για όλο το διάστημα ζωής των αντικειμένων και ο χρήστης μπορεί στη συνέχεια να επεξεργαστεί τα αντικείμενα (βλ. κεφ. 4.2). Για τη ζωή των αντικειμένων υπάρχουν τρεις επιλογές:



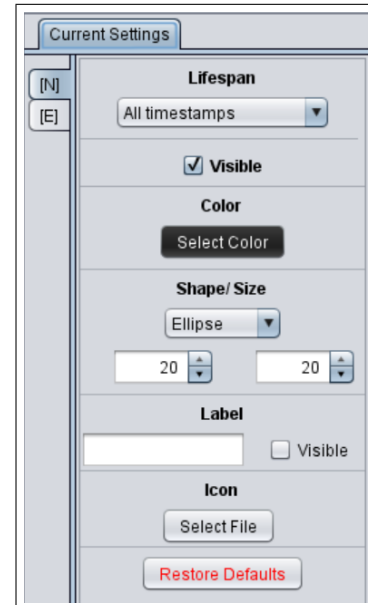
1. All Timestamps: Οι κόμβοι προστίθενται σε όλο το χρονικό εύρος του δυναμικού γραφήματος και οι ακμές προστίθενται στο μεγαλύτερο δυνατό διάστημα για το οποίο ζουν και οι δύο κόμβοι.

2. From this point forward: Οι κόμβοι προστίθενται από την τρέχουσα χρονική στιγμή μέχρι το τέλος του χρονικού εύρους του δυναμικού γραφήματος και οι ακμές προστίθενται από την τρέχουσα χρονική στιγμή μέχρι το τέλος του μεγαλύτερου δυνατού διαστήματος που ζουν οι δύο κόμβοι.
3. One timestamp: Οι κόμβοι και οι ακμές προστίθενται μόνο την τρέχουσα χρονική στιγμή.

(*) Αν η τρέχουσα χρονική στιγμή είναι διάστημα, τότε η προσθήκη του αντικειμένου γίνεται στο πάνω άκρο του διαστήματος.

(*) Αν η δυναμική απεικόνιση δεν είναι ενεργοποιημένη, τότε σαν τρέχουσα χρονική στιγμή θεωρείται ο πρώτος ακέραιος ή η πρώτη ημερολογιακή ημέρα του χρονικού εύρους του δυναμικού γραφήματος.

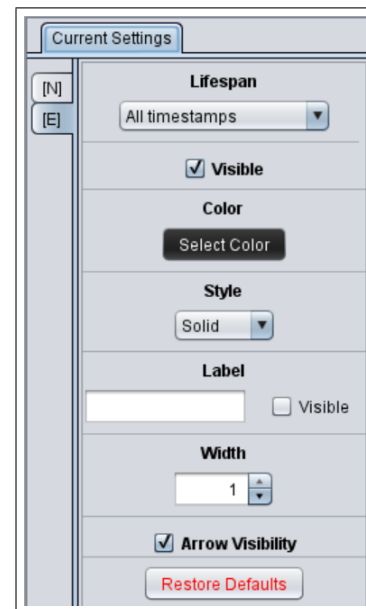
Για τους κόμβους, ο χρήστης μπορεί να αλλάξει το εύρος ζωής τους, το αν θα είναι ορατοί, το χρώμα τους, το σχήμα τους (έλλειψη ή ορθογώνιο παραλληλόγραμμο), το πλάτος και το ύψος τους, το όνομα τους και τη φωτογραφία τους. Στις ρυθμίσεις δεν υπάρχει κάποιο πεδίο για τη θέση του κόμβου, αφού ο κόμβος προστίθεται στο σημείο που βρίσκεται το ποντίκι τη στιγμή της προσθήκης. Επίσης, δεν επιλέγεται το id του κόμβου από το χρήστη, αλλά ο κόμβος παίρνει σαν id τον πρώτο (μεγαλύτερο ή ίσο του μηδενός) ακέραιο που είναι «διαθέσιμος» στο δυναμικό γράφημα, δηλαδή τον πρώτο ακέραιο που δεν υπάρχει σαν id άλλου κόμβου. Οι επιλογές που θα κάνει ο χρήστης καθορίζουν τις ιδιότητες με τις οποίες θα προστεθεί ένας κόμβος στη συνέχεια. Σύμφωνα με τις αρχικές προεπιλεγμένες ρυθμίσεις ένας κόμβος ζει με χρονικό εύρος “All Timestamps”, είναι ορατός μαύρου χρώματος, είναι κύκλος ακτίνας 10, χωρίς όνομα και χωρίς φωτογραφία. Οι ρυθμίσεις αυτές επανέρχονται με το κόκκινο κουμπί στο κάτω μέρος του παραθύρου με την ένδειξη “Restore Defaults”.





Σχήμα 26: Η προσθήκη ενός κόμβου στο δυναμικό γράφημα.

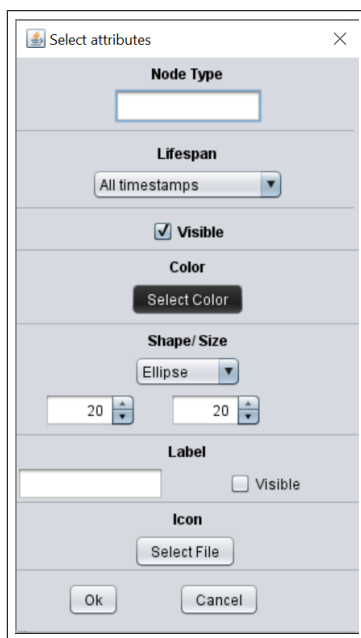
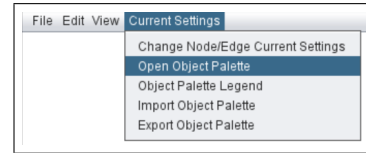
Όσον αφορά τις ακμές, ο χρήστης μπορεί να αλλάξει το εύρος ζωής τους, το αν θα είναι ορατές, το χρώμα τους, το είδος τους (συνεχείς, διακεκομμένες με τελείες ή διακεκομμένες με παύλες), το όνομα τους, το πάχος τους και το αν θα είναι ορατό το βέλος στις κατευθυνόμενες ακμές. Όμοια με τους κόμβους, για τις ακμές δεν επιλεγείται κάποιο id, αλλά δίνεται από το ίδιο το πρόγραμμα. Σύμφωνα με τις αρχικές προεπιλεγμένες ρυθμίσεις μια ακμή ζει με χρονικό εύρος “All Timestamps”, είναι ορατή, μαύρου χρώματος, συνεχής και παχους 1, δεν έχει όνομα και το βέλος είναι ορατό όταν η ακμή είναι κατευθυνόμενη. Οι ρυθμίσεις αυτές επανέρχονται με το κόκκινο κουμπί στο κάτω μέρος του παραθύρου με την ένδειξη “Restore Defaults”.



4.1.2 Τύποι Αντικειμένων

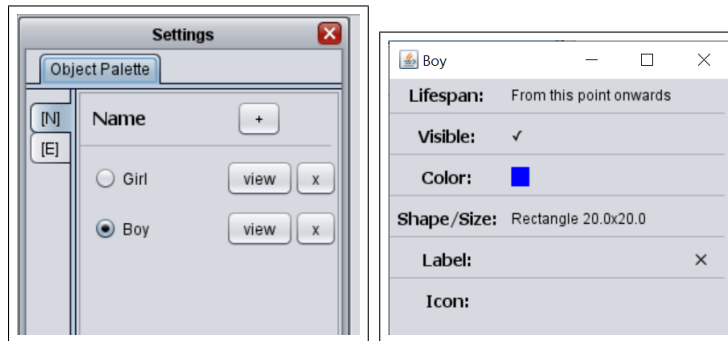
Για την γρήγορη εισαγωγή ενός αντικειμένου στο δυναμικό γράφημα υπάρχει η επιλογή να αποθηκεύονται στο πρόγραμμα κάποιοι τύποι αντικειμένων. Ο τύπος αντικειμένου είναι κάποιες αποθηκευμένες ρυθμίσεις, που όταν επιλεγθούν, εφαρμόζονται στις προεπιλεγμένες ρυθμίσεις και όλα τα αντικείμενα που εισάγονται στη συνέχεια έχουν αυτά τα χαρακτηριστικά. Είναι, δηλαδή, ένας πιο γρήγορος τρόπος να αλλάζουν οι προεπιλεγμένες ρυθμίσεις όταν ο χρήστης επιθυμεί να εισάγει αντικείμενα τα οποία έχουν όμοιες

ιδιότητες. Για παράδειγμα, αν τα δεδομένα αφορούν ανθρώπους, μπορεί ο χρήστης να συμβολίσει τις ηλικιακές ομάδες με διαφορετικό τύπο κόμβων, και να εισάγει τους κόμβους πιο γρήγορα χωρίς να χρειάζεται να αλλάζει όλες τις προεπιλεγμένες ρυθμίσεις για κάθε προσθήκη κόμβου που αφορά κάποιον άνθρωπο σε διαφορετική ηλικιακή ομάδα. Οι κόμβοι και οι ακμές έχουν διαφορετικούς τύπους, υπάρχουν, δηλαδή, δύο «κατηγορίες» τύπων. Στο λογισμικό, οι τύποι αυτοί αναφέρονται ως «παλέτα» αντικειμένων και ο χρήστης μπορεί από το μενού της εφαρμογής να ανοίξει το παράθυρο, μέσω του οποίου μπορεί να εισάγει έναν τύπο ή να διαλέξει από τους ήδη αποθηκευμένους.



Με την επιλογή αυτή ανοίγει ένα παράθυρο στο αριστερό μέρος του παραθύρου, το οποίο αποτελείται από δύο καρτέλες, μία για τις ακμές και μία για τους κόμβους (Σχήμα 27). Κάθε καρτέλα αποτελείται από ένα κουμπί «+» και με όλους τους, ήδη, αποθηκευμένους τύπους αντικειμένων. Ο χρήστης μπορεί να εισάγει έναν τύπο πατώντας το «+», όπου εμφανίζεται ένα καινούριο παράθυρο, στο οποίο καλείται να εισάγει όλες τις επιθυμητές ιδιότητες του τύπου, όπως το όνομα του και τα χαρακτηριστικά του (χρώμα, σχήμα, πάχος κλπ.). Το όνομα του τύπου πρέπει να είναι μοναδικό και αν εισαχθεί ένας τύπος με το ίδιο όνομα, τότε ο παλιός θα αντικατασταθεί με τον καινούριο. Για κάθε τύπο αντικειμένου, ο χρήστης μπορεί είτε να δει πληροφορίες για αυτόν με το κουμπί “view” είτε να τον διαγράψει από την παλέτα με το κουμπί “x”.

Κάθε φορά που επιλέγεται ένας τύπος από τη λίστα, ανανεώνονται όλες οι ιδιότητες στο παράθυρο των προεπιλεγμένων ρυθμίσεων (Current Settings), ώστε να ταιριάζουν με τις ιδιότητες του επιλεγμένου τύπου, και κάθε φορά που αλλάζουν οι προεπιλεγμένες ρυθμίσεις απειλούνται τυχόν τύποι που είχαν επιλεγθεί. Η παλέτα των αντικειμένων είναι αποθηκευμένη στο παράθυρο και όχι στο γράφημα, οπότε οι τύποι που έχουν αποθηκευτεί «χάνονται» μόνο αν ο χρήστης κλείσει το παράθυρο και όχι αν δημιουργήσει ένα καινούριο γράφημα ή αν εισάγει δεδομένα.



Σχήμα 27: Το παράθυρο με τους τύπους κόμβου και οι πληροφορίες για έναν τύπο κόμβου

Ο χρήστης μπορεί να εξάγει τους αποθηκευμένους τύπους σε αρχείο .grpal. Το αρχείο αυτό δημιουργήθηκε συγκεκριμένα για τις πληροφορίες της παλέτας, περιέχει πληροφορίες για τους τύπους κόμβων και ακμών, είναι γραμμένο σε XML και μοιάζει με μια απλοποιημένη μορφή του αρχείου .grdyn. Η δομή του περιγράφεται παρακάτω.

Όλες οι πληροφορίες για τους τύπους αντικειμένων βρίσκονται μέσα στο block palette.

```
<palette>
  ...
</palette>
```

Μέσα σ' αυτό το block βρίσκονται δύο μικρότερα blocks, ένα για τους κόμβους και ένα για τις ακμές.

```
<palette>
  <nodes>
    ...
  </nodes>
  <edges>
    ...
  </edges>
</palette>
```

Μέσα στο block για τους τύπους των κόμβων βρίσκονται πληροφορίες για κάθε τύπο ξεχωριστά, οι οποίες ξεκινάνε με το tag <nnode> και τελειώνουν με το </nnode>. Όλες οι πληροφορίες για τις ιδιότητες του κάθε τύπου, λοιπόν, συμπεριλαμβάνονται μέσα σε

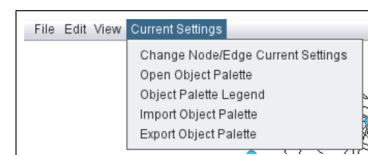
ένα block node. Το tag lifespan παίρνει τις τιμές 0,1 και 2, όπου το 0 αντιστοιχεί στο “All timestamps”, το 1 στο “From this point forward” και το 2 στο “One timestamp”, όπως αυτά περιγράφηκαν παραπάνω στο κεφάλαιο 4.1.1. Ένας κόμβος έχει τη μορφή:

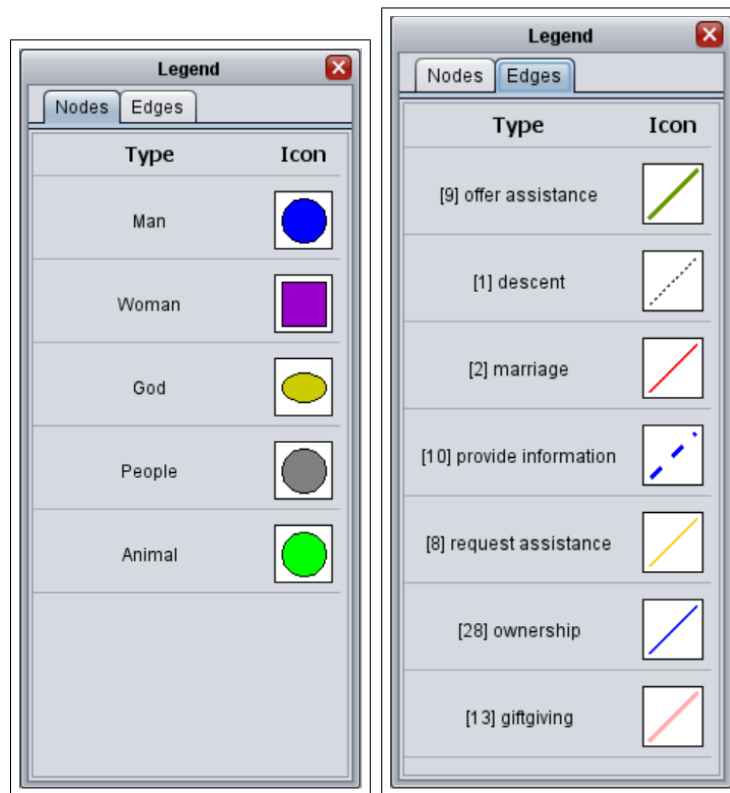
```
<node type="Girl">
  <lifespan value="0"/>
  <visible value="true"/>
  <label value="" visible="false"/>
  <color value="(255;0;0)"/>
  <shape value="ellipse" width="20.0" height="20.0"/>
  <icon value="C:\Users\30693\Documents\woman.jpg"/>
</node>
```

Από την άλλη, με το tag <edge> ξεκινάει ένας καινούριος τύπος ακμής και με το </edge> τελειώνει. Μέσα σε κάθε block ενός τύπου ακμής, λοιπόν, συμπεριλαμβάνονται όλες οι πληροφορίες για τις ιδιότητες του τύπου. Μια ακμή έχει τη μορφή:

```
<edge type="relationship">
  <lifespan value="0"/>
  <visible value="true"/>
  <label value="" visible="false"/>
  <color value="(255;0;0)"/>
  <width value="3.0"/>
  <style value="dotted"/>
  <arrow value="true"/>
</edge>
```

Ο χρήστης μπορεί να εισάγει δεδομένα από αρχείο .grpal στο λογισμικό ή να εξάγει την ήδη υπάρχουσα παλέτα. Οι επιλογές αυτές βρίσκονται στο μενού της εφαρμογής. Επίσης, με την επιλογή “Open Palette Legend” ανοίγει στο δεξιά μέρος της εφαρμογής ένα παράθυρο, το οποίο αποτελεί υπόμνημα της τρέχουσας παλέτας αντικειμένων. Στο υπόμνημα υπάρχουν δύο καρτέλες, μία για τους κόμβους και μία για τις ακμές, και για κάθε τύπο αντικειμένου παρουσιάζεται ένα εικονίδιο με τις αντίστοιχες ιδιότητες. Το υπόμνημα αυτό ανανεώνεται κάθε φορά που προστίθεται ή διαγράφεται ένας τύπος κόμβου ή ακμής.

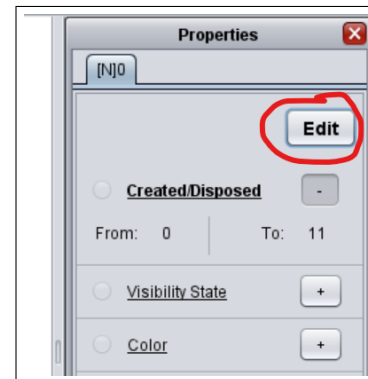




Σχήμα 28: Το υπόμνημα για τους κόμβους και για τις ακμές

4.2 Επεξεργασία Κόμβου και Ακμής

Για να επεξεργαστεί κάποιος τις ιδιότητες ενός κόμβου ή μιας ακμής αρκεί να πατήσει το κουμπί “Edit” στο παράθυρο “Properties” (βλ. κεφ. 2.5.2), το οποίο περιέχει πληροφορίες για όλες τις ιδιότητες ενός αντικειμένου. Με το κουμπί “Edit”, λοιπόν, ενεργοποιείται η λειτουργία της επεξεργασίας και ο χρήστης μπορεί να επεξεργαστεί τις δυναμικές ιδιότητες ενός αντικειμένου.

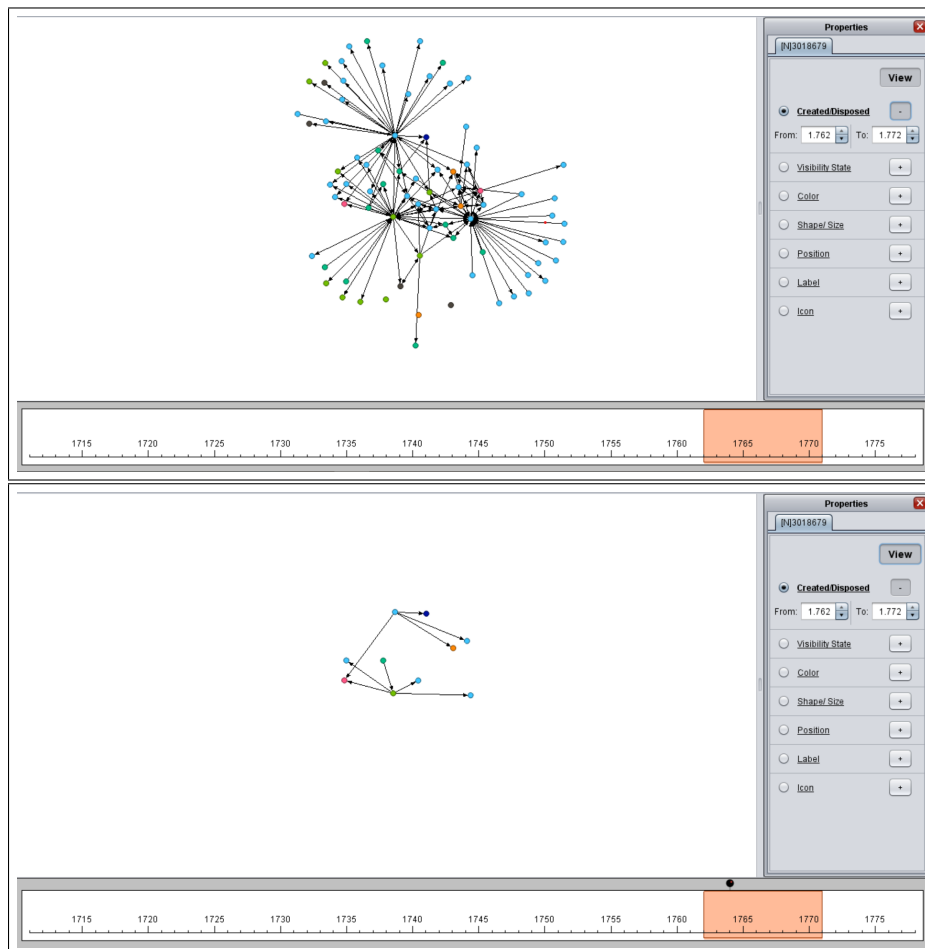


Με την ενεργοποίηση της επεξεργασίας εξαφανίζονται τα κουμπιά για τις λειτουργίες βίντεο και την δυναμική απεικόνιση και η μπάρα του χρόνου καταλαμβάνει όλο το κάτω μέρος του UI. Στο παράθυρο “Properties” ενεργοποιείται για κάθε ιδιότητα ένα κουμπί, το οποίο καθορίζει την ιδι-

ότητα που βρίσκεται σε επεξεργασία. Η ιδιότητα, που τίθεται σε επεξεργασία αυτόματα, είναι αυτή του εύρους ζωής ενός αντικειμένου, όπου εμφανίζεται πάνω στην μπάρα του χρόνου ένα πορτοκαλί διαφανές ορθογώνιο παραλληλόγραμμο, το οποίο περιέχει όλες τις χρονικές στιγμές για τις οποίες ζει το αντικείμενο. Αν ο χρήστης σύρει το πάνω ή το κάτω άκρο του ορθογωνίου, τότε μπορεί να μεγαλώσει ή να μικρύνει το εύρος ζωής του αντικειμένου. Ένας άλλος τρόπος να επεξεργαστεί κάποιος το εύρος είναι από το παράθυρο Properties, όπου μπορεί να μεταβάλει τα πεδία created και disposed.

- * Αν, κατά την επεξεργασία του εύρους ζωής ενός κόμβου, το πεδίο created γίνει μικρότερο ή το πεδίο disposed γίνει μεγαλύτερο από το χρονικό εύρος του δυναμικού γραφήματος, τότε η μπάρα του χρόνου επεκτείνεται ώστε να περιλαμβάνει το καινούριο χρονικό εύρος του δυναμικού γραφήματος, δηλαδή τις καινούριες χρονικές στιγμές.
- * Το εύρος ζωής μιας ακμής πρέπει, όπως προαναφέρθηκε, να βρίσκεται ανάμεσα στο εύρος ζωής των δύο κόμβων, δηλαδή πρέπει το πεδίο created να είναι μεγαλύτερο ή ίσο και το πεδίο disposed να είναι μικρότερο ή ίσο από τα αντίστοιχα πεδία των κόμβων. Έτσι, βεβαιώνεται ότι η ακμή δεν μπορεί να ζει σε χρονική στιγμή που δεν ζουν και οι δύο κόμβοι. Έτσι, τα πεδία created και disposed της ακμής δεν μπορούν και μέσω του UI να πάρουν τιμές που δεν επιτρέπεται.
- * Όταν μικραίνει το εύρος ζωής ενός αντικειμένου, διαγράφονται οι τιμές των ιδιοτήτων τους για τις οποίες το αντικείμενο δεν ζει πια. Όταν μεγαλώνει το εύρος ζωής τους, οι ιδιότητες τους παίρνουν την «πιο κοντινή» τιμή, δηλαδή αν μεγαλώσει το διάστημα προς τα αριστερά, οι ιδιότητες θα πάρουν την πρώτη τιμή, ενώ αν μεγαλώσει προς τα δεξιά, οι ιδιότητες θα πάρουν την τελευταία τιμή. Όταν μειώνεται το εύρος ζωής ενός κόμβου, διαγράφονται από το γράφημα οι ακμές οι οποίες ζουν μόνο στις χρονικές στιγμές που ο κόμβος ζούσε, αλλά δεν ζει πια.
- * Όταν η λειτουργία της δυναμικής απεικόνισης είναι ενεργοποιημένη, τότε κατά την επεξεργασία ενός αντικειμένου εμφανίζεται πάνω από τη μπάρα του χρόνου μία πινέζα, η οποία «δείχνει» την τρέχουσα χρονική στιγμή (ή το πάνω άκρο αν πρόκειται για διάστημα). Την πινέζα αυτή μπορεί κάποιος να την μετακινήσει αραστερά- δεξιά μέσα στο εύρος ζωής του αντικειμένου, ώστε να εμφανιστεί στον καμβά η προβολή του δυναμικού γραφήματος για κάποια άλλη χρονική στιγμή.

Στο σχήμα 29 φαίνονται στιγμιότυπα του UI κατά την επεξεργασία ενός αντικειμένου.



Σχήμα 29: Το UI κατά την επεξεργασία του εύρους ζωής ενός κόμβου

Αν στο παράθυρο “Properties” επιλέξει ο χρήστης κάποια άλλη ιδιότητα για να επεξεργαστεί, τότε τη θέση του βαθμονομημένου χάρακα στο κάτω μέρος παίρνει ένας καινούριος βαθμονομημένος χάρακας, ο οποίος περιέχει μόνο τις χρονικές στιγμές που ζει το εκάστοτε αντικείμενο. Έτσι, ο χρήστης μπορεί να επεξεργαστεί την κάθε ιδιότητα μόνο σε χρονικές στιγμές που το αντικείμενο ζει. Οι επιλογές για τους κόμβους και τις ακμές είναι:

- Visibility State (πότε είναι ορατό ένα αντικείμενο και πότε όχι),
- Color (χρώμα) και
- Label (όνομα).

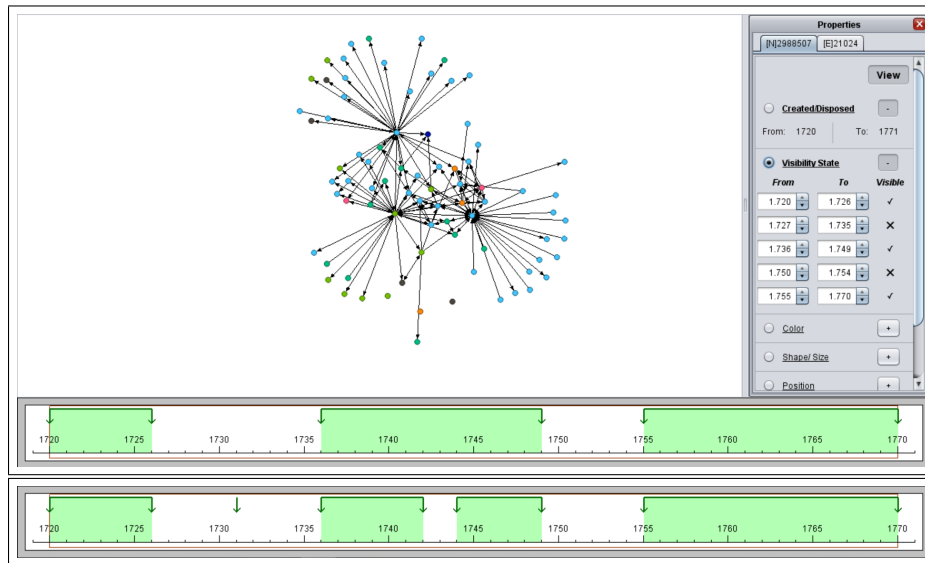
Για τους κόμβους οι έξτρα επιλογές είναι:

- Shape/ Size (σχήμα και μέγεθος),
- Position (θέση) και
- Icon (φωτογραφία),

ενώ για τις ακμές:

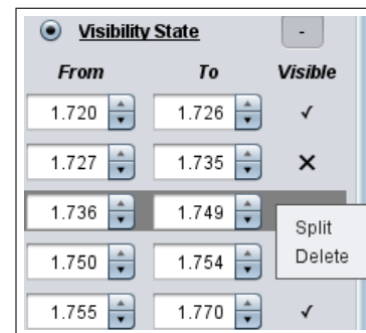
- Width (πλάτος),
- Style (είδος) και
- Arrow Visibility (ορατότητα του βέλους μόνο για τις κατευθυνόμενες ακμές).

Visibility State: Κατά την επεξεργασία του πότε είναι ορατό ένα αντικείμενο ή όχι εμφανίζονται πάνω στη μπάρα του χρόνου πράσινα διαφανή ορθογώνια παραλληλόγραμμα, ένα για κάθε διάστημα που το αντικείμενο είναι ορατό. Ο χρήστης μπορεί να επεκτείνει, να μικρύνει ή να ενώσει δύο διαστήματα, σέρνοντας το πάνω ή το κάτω άκρο τους. Επίσης, πατώντας δεξί κλικ πάνω στη μπάρα του χρόνου σε κάποια χρονική στιγμή εισάγεται μόνο για την συγκεκριμένη χρονική στιγμή η αντίθετη τιμή από την ήδη υπάρχουσα. Αν σε ένα διάστημα $[t_i, t_j]$ το αντικείμενο είναι ορατό και πατηθεί δεξί κλικ σε μια χρονική στιγμή $t_k, t_i < t_k < t_j$, τότε εισάγεται για την χρονική στιγμή t_k η τιμή false, οπότε το τελικό αποτέλεσμα είναι τρία διαστήματα: $[t_i, t_k - 1]$, όπου το αντικείμενο είναι ορατό, $[t_k, t_k]$, όπου το αντικείμενο δεν είναι ορατό και $[t_k + 1, t_j]$, όπου το αντικείμενο είναι ορατό. Απ' την άλλη, αν πατηθεί δεξί κλικ σε άκρο διαστήματος, π.χ. στο t_i , τότε απλά μικραίνει το διάστημα κατά μία χρονική στιγμή, αφού το τελικό διάστημα, όπου ο κόμβος είναι ορατός, θα είναι το $[t_i + 1, t_j]$. Όμοια σε διάστημα που το αντικείμενο δεν είναι ορατό, εισάγεται η τιμή true μόνο για την επιλεγμένη χρονική στιγμή (Σχήμα 30).

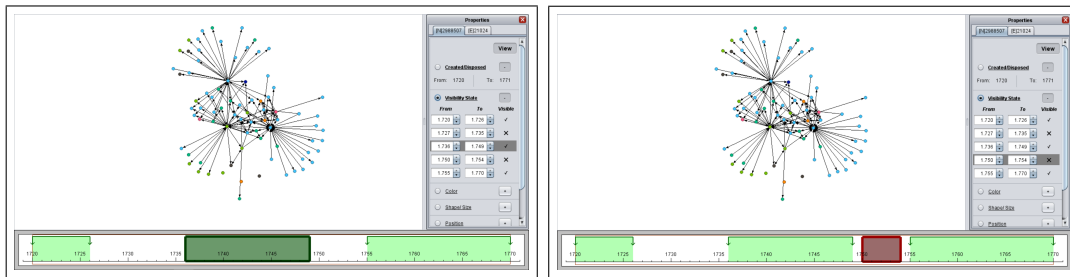


Σχήμα 30: Το UI κατά την επεξεργασία του τότε είναι ορατό ένα αντικείμενο και η μπάρα του χρόνου μετά από δεξί κλικ τη χρονική στιγμή 1731 και 1743

Η επεξεργασία μπορεί να γίνει και από το παράθυρο “Properties”. Αλλάζοντας τα αντίστοιχα πεδία From και To (Σχήμα 30) μπορεί ο χρήστης να επεκτείνει ή μικρύνει ένα διάστημα, αλλά όχι να δημιουργήσει ή να ενώσει διαστήματα. Αυτές οι λειτουργίες γίνονται με δεξί κλικ, όπου εμφανίζονται οι επιλογές Split και Delete. Με την επιλογή Split το διάστημα χωρίζεται στα δύο, σαν να πατάει κάποιος δεξί κλικ στο μέσον του διαστήματος (το αποτέλεσμα είναι 3 διαστήματα). Με την επιλογή Delete το αντίστοιχο διάστημα παίρνει την «αντίθετη» τιμή, δηλαδή το αντικείμενο από ορατό γίνεται μη ορατό και αντίστροφα.



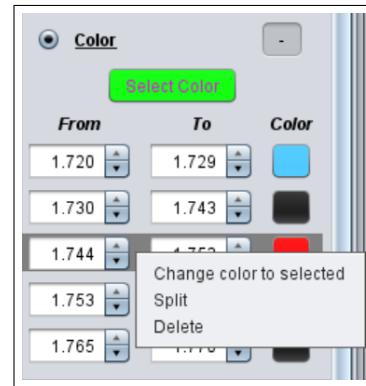
Ο χρήστης μπορεί να επιλέξει ένα διάστημα είτε από το παράθυρο “Properties” είτε από την μπάρα του χρόνου. Άν για το επιλεγμένο διάστημα το αντικείμενο είναι ορατό, τότε στην μπάρα του χρόνου το αντίστοιχο διάστημα γίνεται σκούρο πράσινο και στο παράθυρο “Properties” η αντίστοιχη σειρά γίνεται γκρι. Αν, απ’ την άλλη, για το επιλεγμένο διάστημα το αντικείμενο δεν είναι ορατό, τότε στην μπάρα του χρόνου το αντίστοιχο διάστημα γίνεται σκούρο κόκκινο και στο παράθυρο “Properties” η αντίστοιχη σειρά γίνεται, και πάλι, γκρι.



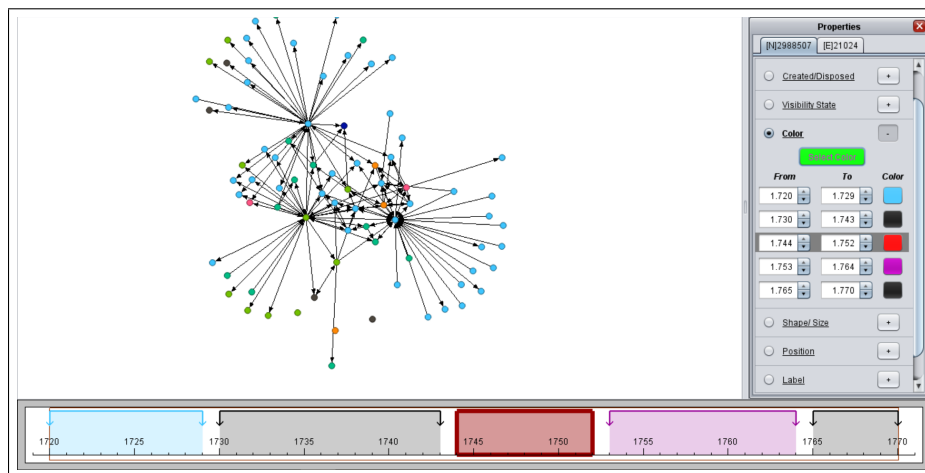
Σχήμα 31: Η μπάρα του χρόνου και το παράθυρο “Properties” όταν επιλέγεται ένα διάστημα

Όσον αφορά το χρώμα και το όνομα των αντικειμένων, το σχήμα, τη θέση και τη φωτογραφία των κόμβων, αλλά και το πάχος και το είδος των ακμών η επεξεργασία ακολουθεί για όλες τις ιδιότητες ένα συγκεκριμένο μοτίβο. Κατά την επεξεργασία τους εμφανίζονται στην μπάρα του χρόνου διαφανή ορθογώνια παραλληλόγραμμα, όπου το κάθε ένα αντιστοιχεί σε μία εναλλαγή της τιμής της ιδιότητας. Κάθε διάστημα, δηλαδή, όπου ισχύει μια διαφορετική τιμή, συμβολίζεται στην μπάρα του χρόνου με ένα παραλληλόγραμμο διαφορετικού χρώματος. Ο χρήστης μπορεί να επεκτείνει, να μικρύνει ή να διαγράψει τα διαστήματα αυτά σέρνοντας το κάτω ή το πάνω άκρο τους αριστερά- δεξιά. Για κάθε ιδιότητα από τις παραπάνω, εμφανίζονται στο παράθυρο “Properties” κάποια πεδία, όπως κουμπιά ή πεδία κειμένου, στα οποία ο χρήστης μπορεί να εισάγει μια νέα τιμή για την κάθε ιδιότητα που επιθυμεί να «καταχωρήσει». Πατώντας δεξί κλικ πάνω στην μπάρα του χρόνου σε κάποια χρονική στιγμή, το αντίστοιχο διάστημα «σπάει» στα δύο στο σημείο που βρίσκεται το ποντίκι και στο δεύτερο μέρος του διαστήματος εισάγεται η τιμή που αναγράφεται στα αντίστοιχα πεδία εισαγωγής. Αν, για παράδειγμα, ένας κόμβος είναι στο διάστημα [10; 20] κόκκινος, στο αντίστοιχο κουμπί έχει επιλεγεί το πράσινο χρώμα και ο χρήστης πατήσει δεξί κλικ τη χρονική στιγμή 16, τότε το χρώμα του κόμβου θα διαμορφωθεί ως κόκκινο στο διάστημα [10; 15] και πράσινο στο διάστημα [16; 20]. Αν ο χρήστης επιλέξει ένα από τα ορθογώνια παραλληλόγραμμα, τότε αυτό χρωματίζεται με πιο σκούρο χρώμα στην μπάρα του χρόνου, και η αντίστοιχη γραμμή στο παράθυρο “Properties” χρωματίζεται με γκρι χρώμα. Τέλος, όσον αφορά την επεξεργασία μιας ιδιότητας από το παράθυρο “Properties”, αν ο χρήστης πατήσει δεξί κλικ πάνω σε μια γραμμή, δηλαδή πάνω σε ένα διάστημα όπου ισχύει μία τιμή της ιδιότητας, εμφανίζονται τρεις επιλογές, όπου η πρώτη αντικαθιστά την υπάρχουσα τιμή με την τιμή του πεδίου εισαγωγής σε όλο το διάστημα, η δεύτερη «κόβει» το διάστημα στα δύο και το δεύτερο μέρος παίρνει την τιμή του πεδίου εισαγωγής και η τρίτη διαγράφει την υπάρχουσα τιμή και η ιδιότητα παίρνει την τιμή του προηγούμενου διαστήματος (ή του επόμενου αν πρόκειται για την πρώτη γραμμή ή την προεπιλεγμένη τιμή αν πρόκειται για την μοναδική γραμμή).

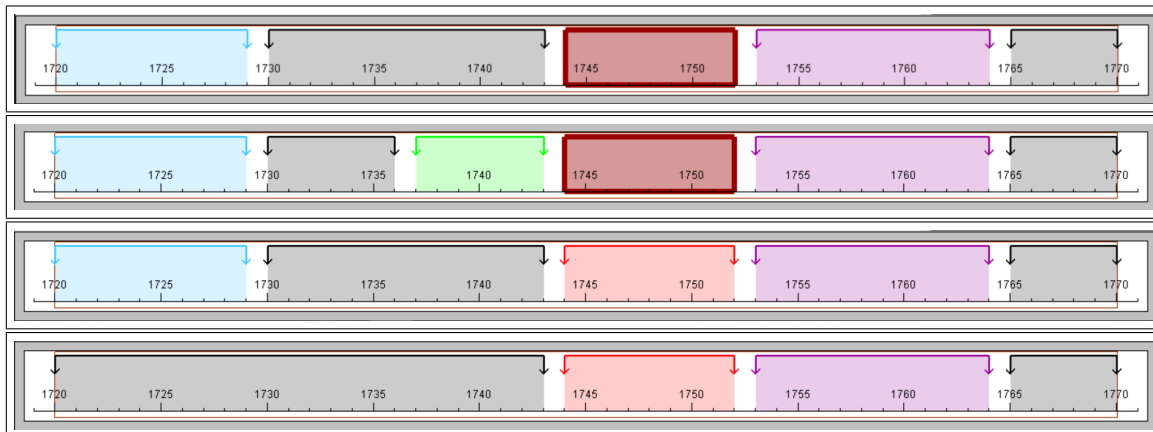
Color: Το πεδίο εισαγωγής για το χρώμα είναι ένα κουμπί με το όνομα “Select color” με το οποίο ανοίγει μια παλέτα και ο χρήστης μπορεί να διαλέξει το χρώμα που επιθυμεί να προσθέσει. Το φόντο του έχει πάντα το χρώμα που είναι επιλεγμένο. Κατά την επεξεργασία του χρώματος ενός αντικειμένου τα διαφανή ορθογώνια παραλληλόγραμμα στη μπάρα του χρόνου έχουν το χρώμα που έχει το αντικείμενο στο αντίστοιχο διάστημα. Αντίστοιχα με παραπάνω, ο χρήστης μπορεί να σύρει το κάτω ή το πάνω άκρο ενός διαστήματος, αλλά και να πατήσει δεξί κλικ σε κάποια χρονική στιγμή. Στο δεξί κλικ το αντίστοιχο διάστημα «σπάει» στα δύο και το δεύτερο μέρος του διαστήματος παίρνει το χρώμα που είναι επιλεγμένο, δηλαδή το χρώμα που έχει ως φόντο το κουμπί “Select Color” (Σχήμα 32).



Με δεξί κλικ σε κάποιο διάστημα στο παράθυρο “Properties” ανοίγουν οι επιλογές Change Color to Selected, Split και Delete. Με την πρώτη επιλογή αλλάζει το χρώμα σε όλο το επιλεγμένο διάστημα και παίρνει το χρώμα του κουμπιού, με την δεύτερη «σπάει» το διάστημα σε δύο κομμάτια και το δεύτερο παίρνει το χρώμα του κουμπιού και με την τρίτη όλο το διάστημα διαγράφεται και παίρνει το χρώμα που έχει το αντικείμενο στο προηγούμενο διάστημα (ή στο επόμενο διάστημα αν πρόκειται για το πρώτο διάστημα ή μαύρο αν πρόκειται για το μοναδικό διάστημα) (Σχήμα 33).

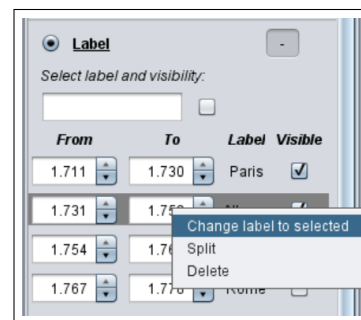


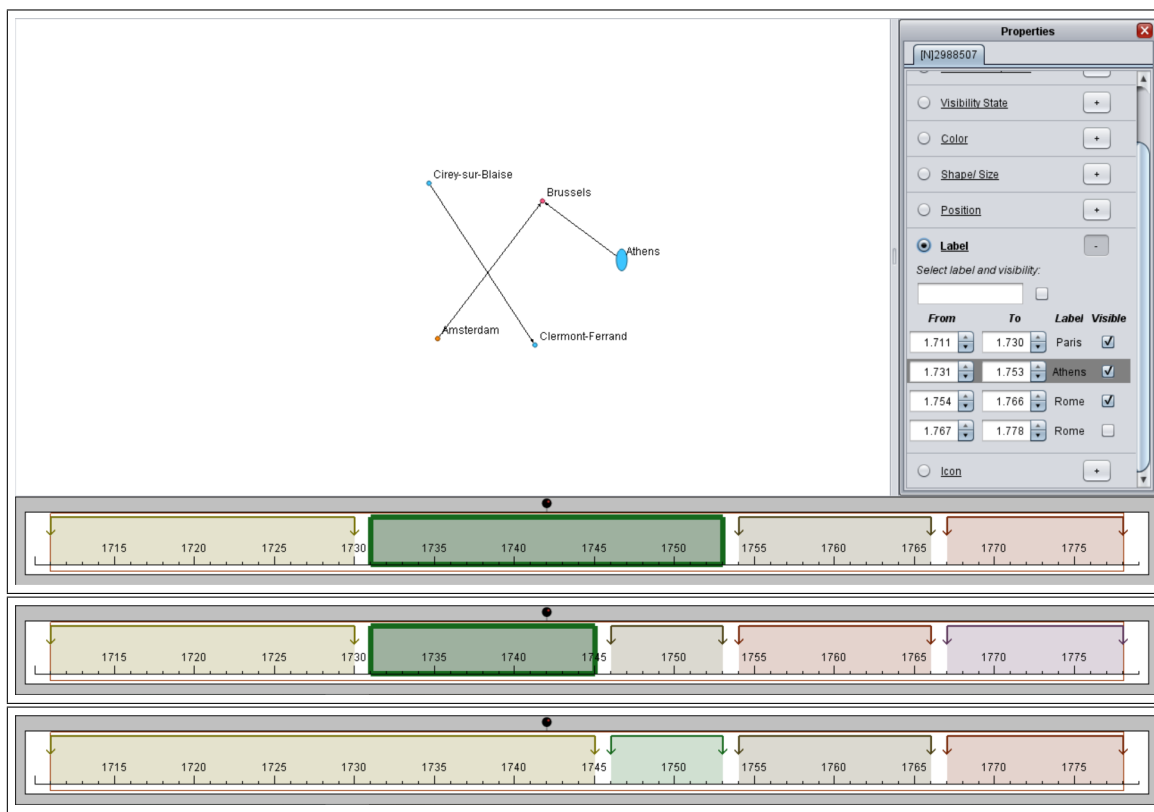
Σχήμα 32: Το UI κατά την επεξεργασία χρώματος ενός αντικειμένου.



Σχήμα 33: Η μπάρα του χρόνου κατά την επεξεργασία χρώματος ενός αντικειμένου, μετά από δεξί κλικ στη χρονική στιγμή 1737, μετά από διαγραφή (delete) του πράσινου διαστήματος και μετά από διαγραφή (delete) του γαλάζιου διαστήματος.

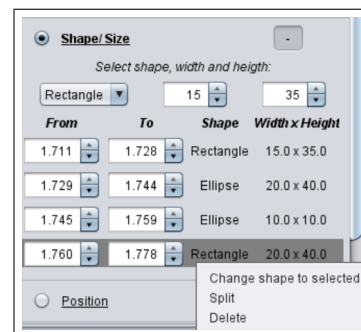
Label: Όταν ο χρήστης επιλέξει να επεξεργαστεί το όνομα του αντικειμένου, εμφανίζονται δύο πεδία, όπου στο ένα μπορεί να εισάγει μια αλφαριθμητική τιμή για το όνομα και στο άλλο μπορεί να επιλέξει αν επιθυμεί αν είναι ορατή αυτή η τιμή ή όχι. Αντίστοιχα και για το όνομα, κατά την επεξεργασία του εμφανίζονται πάνω στη μπάρα του χρόνου διαφανή ορθογώνια παραλληλόγραμμα που υποδεικνύουν το χρονικό διάστημα για το οποίο ισχύει η κάθε τιμή της ιδιότητας. Όμοια με τις παραπάνω ιδιότητες, ένα διάστημα μπορεί να επεκταθεί, να μικρύνει ή να διαγραφεί αν ο χρήστης σύρει το κάτω ή το πάνω άκρο ενός διαστήματος και να «σπάσει» στα δύο αν ο χρήστης πατήσει δεξί κλικ πάνω σε μια χρονική στιγμή. Επίσης, οι επιλογές από το παράθυρο για ένα διάστημα είναι οι Change label to selected, Split και Delete, όπου οι λειτουργίες είναι όμοιες μ' αυτές που περιγράφηκαν παραπάνω για τις συγκεκριμένες επιλογές.

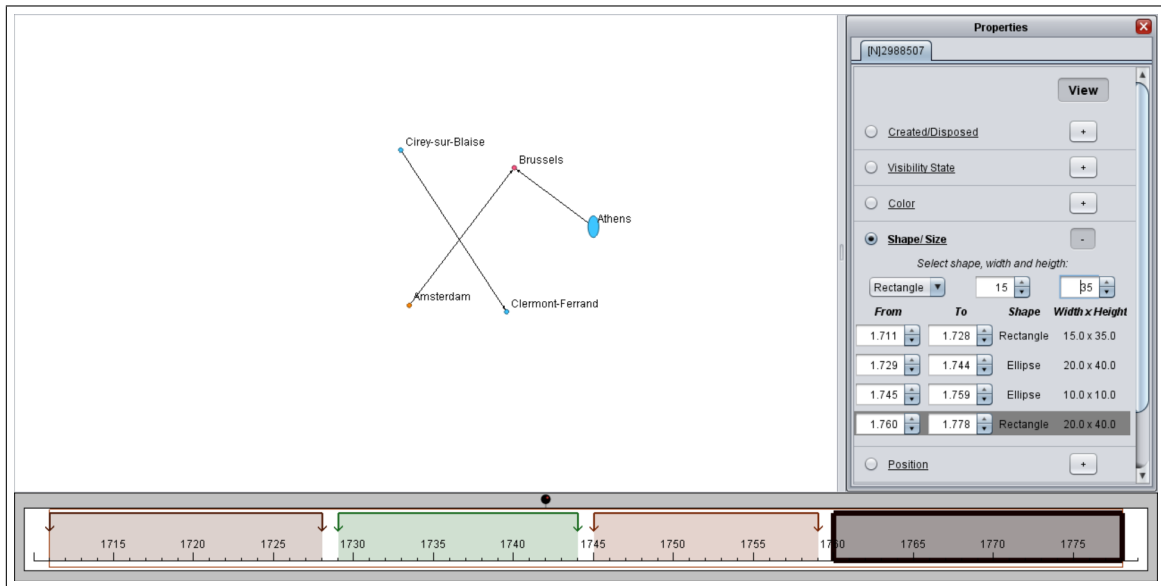




Σχήμα 34: Το UI κατά την επεξεργασία ονόματος ενός αντικειμένου και η μπάρα του χρόνου μετά από δεξί κλικ τη χρονική στιγμή 1746 και μετά από διαγραφή του πρώτου διαστήματος.

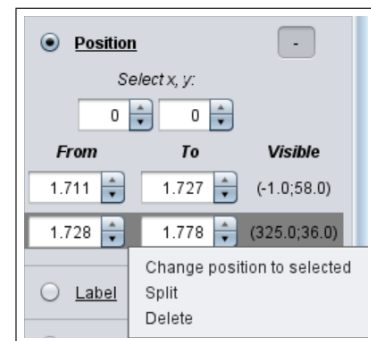
Shape/ Size: Για την επεξεργασία του σχήματος και του μεγέθους ενός κόμβου υπάρχουν τρία πεδία εισαγωγής, όπου στο ένα επιλεγεται το σχήμα (έλλειψη ή ορθογώνιο παραλληλόγραμμο) και στα άλλα δύο επιλέγονται το πλάτος και το ύψος του σχήματος. Τόσο οι λειτουργίες επεξεργασίας στην μπάρα του χρόνου, όσο και οι λειτουργίες επεξεργασίας στο παράθυρο “Properties” είναι διαθέσιμες και για την επεξεργασία του σχήματος ενός κόμβου.

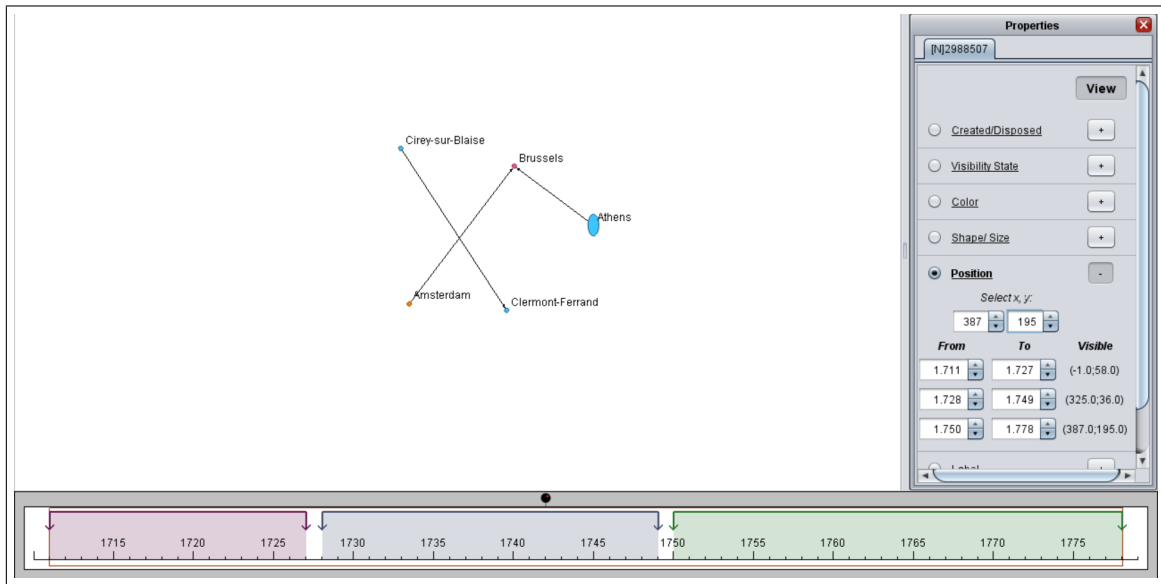




Σχήμα 35: Το UI κατά την επεξεργασία του σχήματος ενός κόμβου.

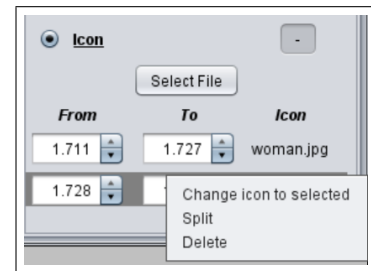
Position: Για την επεξεργασία της θέσης ενός κόμβου υπάρχουν δύο πεδία εισαγωγής, όπου στο ένα εισάγεται η x συντεταγμένη και στην άλλη η y συντεταγμένη. Ο χρήστης μπορεί σ' αυτά τα πεδία να πληκτρολογήσει τους δύο αριθμούς που επιθυμεί ή μπορεί να πατήσει διπλό κλικ οπουδήποτε στον καμβά και στα πεδία θα εισαχθούν αυτόματα οι συντεταγμένες όπου έγινε το διπλό κλικ.

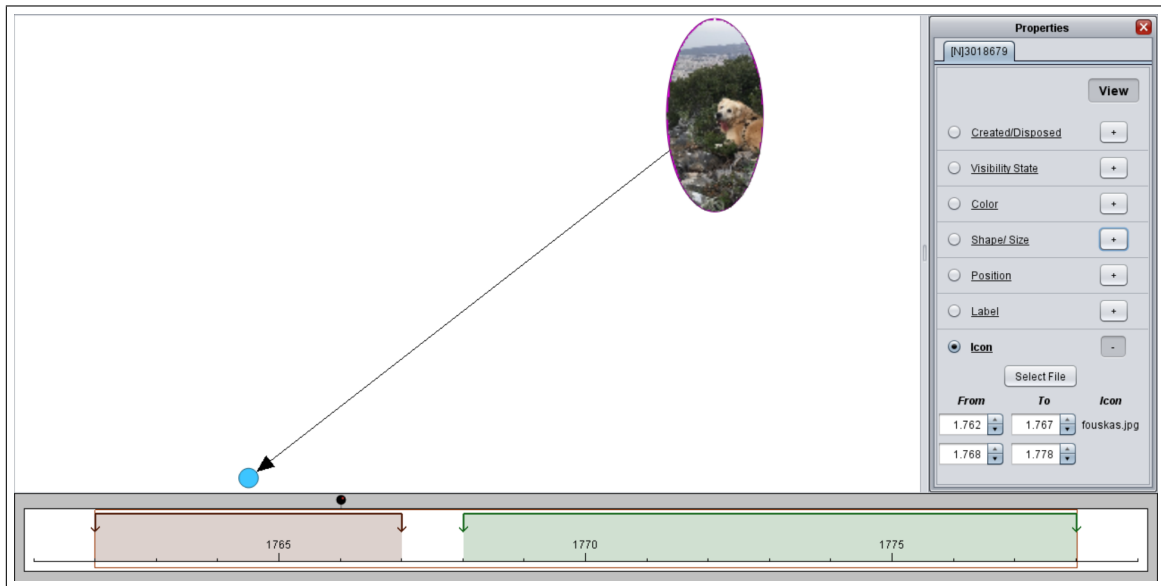




Σχήμα 36: Το UI κατά την επεξεργασία της θέσης ενός κόμβου.

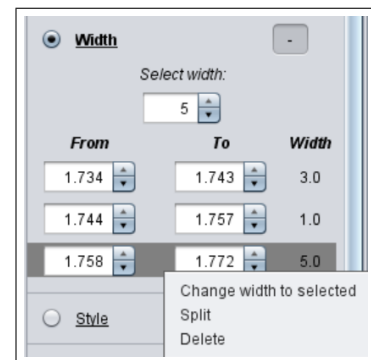
Icon: Για την επεξεργασία της φωτογραφίας ενός κόμβου εμφανίζεται ένα κουμπί “Select File”, με το οποίο ο χρήστης μπορεί να επιλέξει από τον υπολογιστή του τη φωτογραφία που θέλει να εισάγει. Το κουμπί αυτό στη συνέχεια παίρνει το όνομα του αρχείου που επιλέχθηκε. Σαν διακριτή τιμή στη φωτογραφία θεωρείται και το κενό, δηλαδή για κάθε διάστημα που ο κόμβος είτε έχει διακριτή φωτογραφία είτε δεν έχει καθόλου εμφανίζεται στην μπάρα του χρόνου ένα διαφορετικό ορθογώνιο παραλληλόγραμμο. Επίσης, για να επιλέξει κάποιος να εισάγει ένα διάστημα όπου ο κόμβος δεν θα έχει φωτογραφία αρκεί στην αναζήτηση αρχείου να επιλέξει «Ακύρωση», αρκεί, δηλαδή, στο κουμπί αναζήτησης να αναγράφεται “Select File” και όχι κάποιο όνομα αρχείου. Ο χρήστης μπορεί στη συνέχεια να επεξεργαστεί αυτή την ιδιότητα όπως περιγράφηκε παραπάνω.

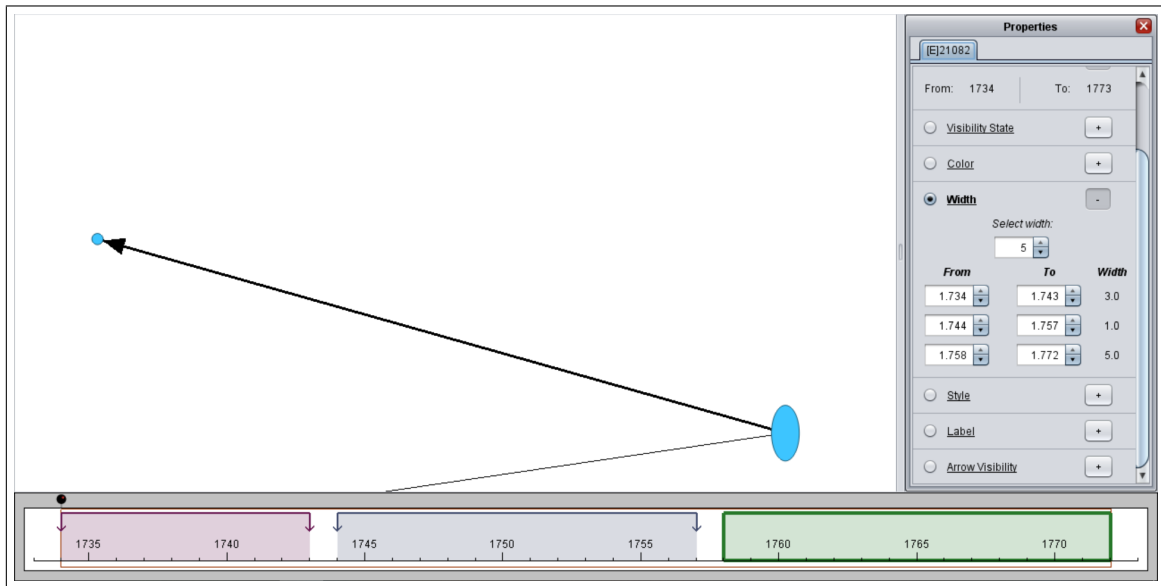




Σχήμα 37: Το UI κατά την επεξεργασία της φωτογραφίας ενός κόμβου.

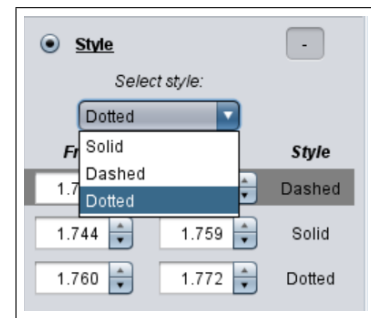
Width: Για την επεξεργασία του πάχους μιας ακμής εμφανίζεται ένα πεδίο, όπου ο χρήστης μπορεί να εισάγει τον αριθμό που επιθυμεί για το πάχος. Τα χρώματα που παίρνουν τα ορθογώνια παραλληλόγραμμα στην μπάρα του χρόνου δεν έχουν κάποιο σημασιολογικό χαρακτήρα, αλλά κάθε εναλλαγή τιμής για το πάχος της ακμής αντιστοιχεί σε ένα ορθογώνιο παραλληλόγραμμο διαφορετικού χρώματος. Όσον αφορά την επεξεργασία, ο χρήστης μπορεί και για το πάχος της ακμής να επεκτείνει, να μικρύνει ή να διαγράψει ένα διάστημα, ενώ με δεξί κλικ μπορεί να «κόψει» στα δύο ένα διάστημα όπως περιγράφηκε παραπάνω. Για την επεξεργασία του πλάτους από το παράθυρο “Properties” μπορεί και πάλι ο χρήστης να επεκτείνει ή να μικρύνει ένα διάστημα αλλάζοντας τις τιμές From και To, δηλαδή τα άκρα του κάθε διαστήματος, αλλά και να αλλάξει την τιμή σε όλο το διάστημα, να το «κόψει» σε δύο ή να το διαγράψει (οι επιλογές αυτές εμφανίζονται με δεξί κλικ πάνω σε ένα διάστημα). Κατά τη διαγραφή, το εκάστοτε διάστημα παίρνει την τιμή του επόμενου διαστήματος (ή του προηγούμενου αν πρόκειται για το πρώτο διάστημα ή την τιμή 1 αν πρόκειται για το μοναδικό διάστημα).

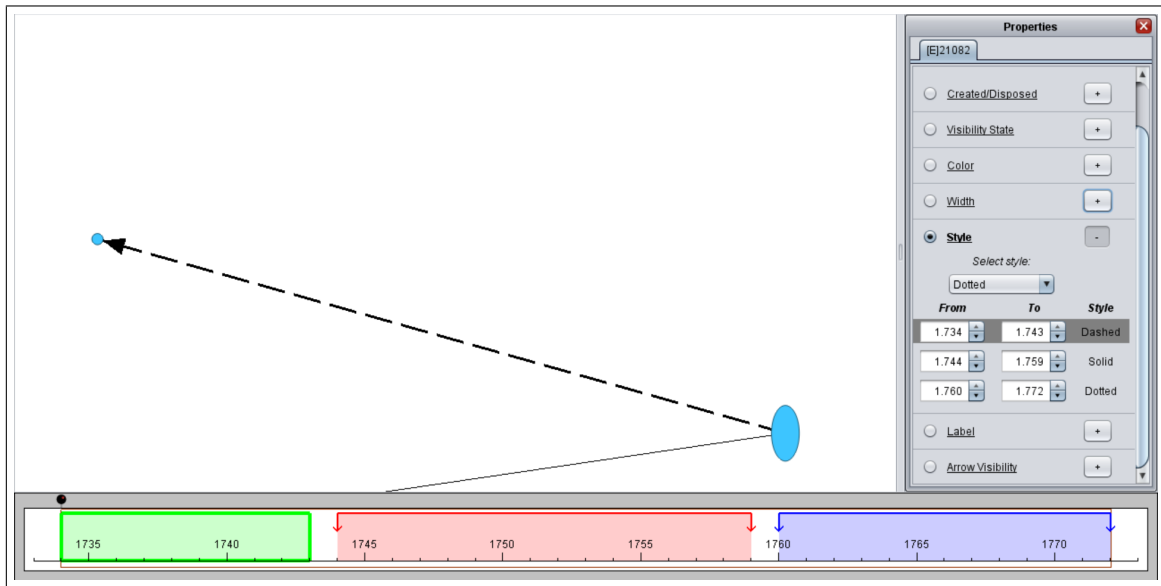




Σχήμα 38: Το UI κατά την επεξεργασία του πάχους μια ακμής.

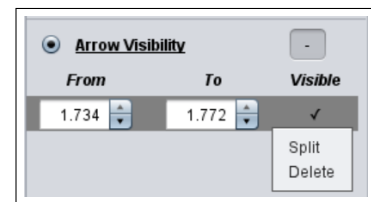
Style: Για την επεξεργασία του είδους μιας ακμής το πεδίο εισαγωγής είναι μια λίστα επιλογής και δίνει τις τρεις επιλογές Solid, Dashed και Dotted, δηλαδή συνεχές ευθύγραμμο τμήμα, διακεκομμένο με παύλες και διακεκομμένο με τελείες. Επειδή για το είδος της ακμής οι τιμές που μπορεί να πάρει η ιδιότητα είναι περιορισμένες, τα χρώματα που έχουν τα ορθογώνια παραλληλόγραμμα στην μπάρα του χρόνου είναι συγκεκριμένα και έχουν σημασιολογικό χαρακτήρα. Αυτά είναι: πράσινο για διακεκομμένο με παύλες, κόκκινο για συνεχές και μπλε για διακεκομμένο με τελείες. Το δεξί κλικ, λοιπόν, στη μπάρα του χρόνου και οι επιλογές Change Style to Selected και Split εισάγουν το είδος της ακμής που είναι επιλεγμένο στη λίστα επιλογής. Το προεπιλεγμένο είδος για τις ακμές είναι το συνεχές ευθύγραμμο τμήμα.

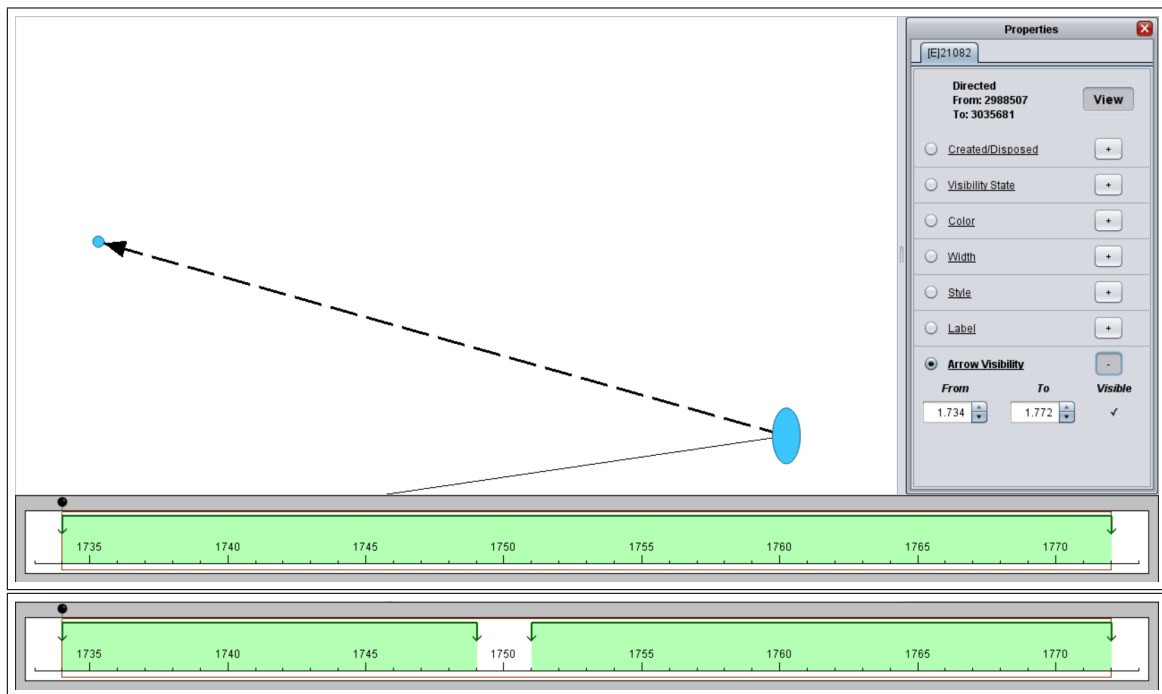




Σχήμα 39: Το UI κατά την επεξεργασία του είδους μια ακμής.

Arrow Visibility: Τέλος, για τις κατευθυνόμενες ακμές ο χρήστης μπορεί να επεξεργαστεί το αν θα είναι ορατό το βέλος της ακμής. Η επεξεργασία αυτή ακολουθεί ακριβώς το μοτίβο της ιδιότητας Visibility, δηλαδή του πότε είναι ορατό ένα αντικείμενο και πότε όχι. Στην μπάρα του χρόνου εμφανίζονται μόνο πράσινα ορθογώνια παραλληλόγραμμα, τα οποία αντιστοιχούν στα διαστήματα όπου το βέλος είναι ορατό. Αν ο χρήστης κάνει δεξί κλικ στην μπάρα του χρόνου σε κάποια χρονική στιγμή, τότε εισάγεται η «αντίθετη» τιμή από την ήδη υπάρχουσα μόνο στην συγκεκριμένη χρονική στιγμή. Για να ενώσει κάποιος δύο διαστήματα αρκεί να σύρει τα άκρα τους, ώστε να ενωθούν μεταξύ τους. Για την επεξεργασία από το παράθυρο “Properties” υπάρχουν δύο επιλογές, η επιλογή Split και η επιλογή Delete. Με την πρώτη επιλογή, γίνεται μια «τρύπα» στο μέσον του επιλεγμένου διαστήματος, δηλαδή εισάγεται η αντίθετη τιμή της ιδιότητας από την υπάρχουσα, σαν να πατούσε κάποιος δεξί κλικ στο μέσον του διαστήματος. Με την επιλογή Delete όλο το διάστημα παίρνει την αντίθετη τιμή, δηλαδή αν το βέλος είναι ορατό γίνεται μη ορατό και το αντίθετο.



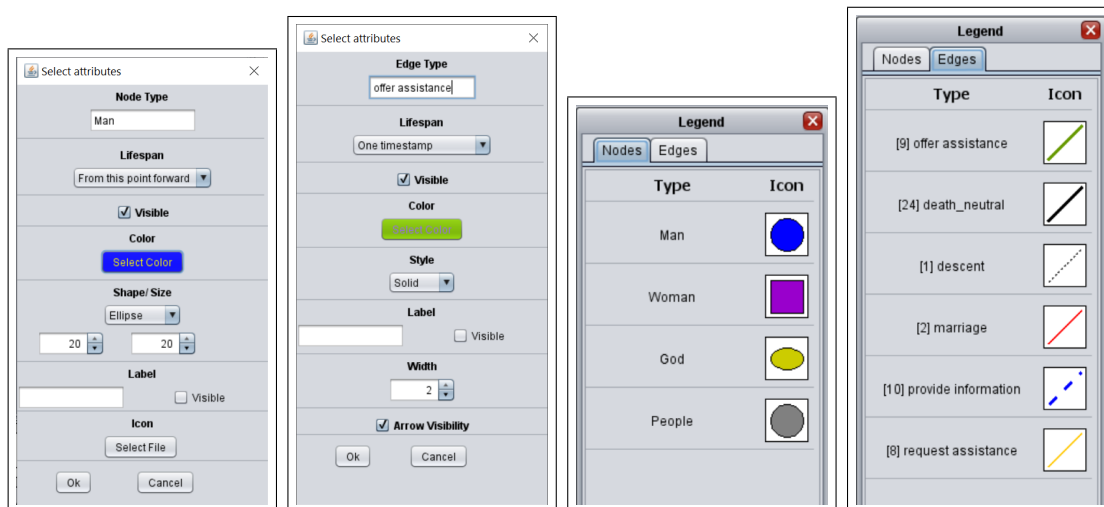


Σχήμα 40: Το UI κατά την επεξεργασία του βέλους μια ακμής και μετά από δεξί κλικ τη χρονική στιγμή 1750.

5 Εφαρμογή: Hrafnkels Saga

Στην συνέχεια θα δημιουργήσουμε ένα δυναμικό γράφημα, το οποίο θα αντιστοιχεί σε ένα μικρό μέρος του ισλανδικού μύθου Hrafnkels Saga[17]. Ο μύθος αυτός αποτέλεσε αντικείμενο του διαγωνισμού Graph Drawing and Network Visualization, ο οποίος το 2020 έλαβε χώρα στο πανεπιστήμιο της British Columbia [9]. Στα πλαίσια του διαγωνισμού αυτού δόθηκαν σε αρχείο Excel τα δεδομένα της ιστορίας κωδικοποιημένα, τα οποία θα χρησιμοποιήσουμε για την απεικόνιση του δυναμικού γραφήματος, ενώ παράλληλα θα δοθεί μια ελεύθερη μετάφραση των πρώτων δύο κεφαλαίων της ιστορίας.

Αρχικά, θα δημιουργήσουμε την παλέτα τύπων, η οποία για τους κόμβους θα αποτελείται από τους τύπους Man, Woman, God και People και για τις ακμές θα αποτελείται από τους τύπους offer assistance, death-neutral, descent, marriage, provide information και request assistance. Με την βοήθεια της παλέτας δημιουργούμε το δυναμικό γράφημα που φαίνεται στο σχήμα 42. Ενδεικτικά για την πρώτη χρονική στιγμή: προσθέτουμε τους κόμβους Hallfred και Hrafnkell επιλέγοντας τον τύπο Man, ενώ για τους κόμβους Hallfred's Wife και Arnthrud θα επιλέξουμε τον τύπο Woman. Οι ακμές είναι: 1) τύπου descent από τον Hrafnkell στον Hallfred και 2) από τον Hrafnkell στην σύζυγο του Hallfred, 3) τύπου marriage από τον Hallfred στην σύζυγο του και 4) τύπου offer assistance από την Arnthrud στον Hallfred.

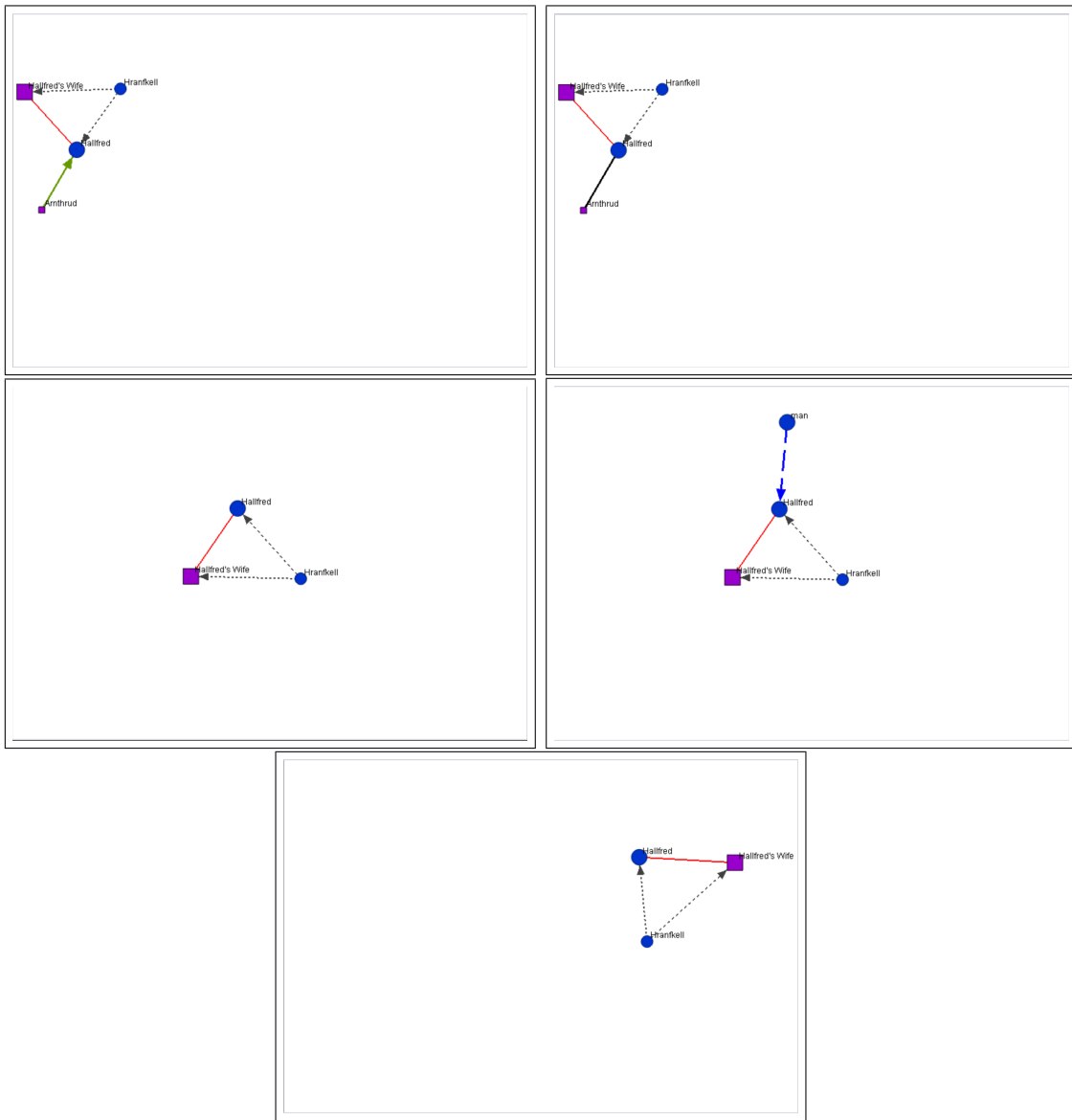


Σχήμα 41: Η προσθήκη του τύπου κόμβου Man και του τύπου ακμής offer assistance και το υπόμνημα για τους κόμβους και τις ακμές

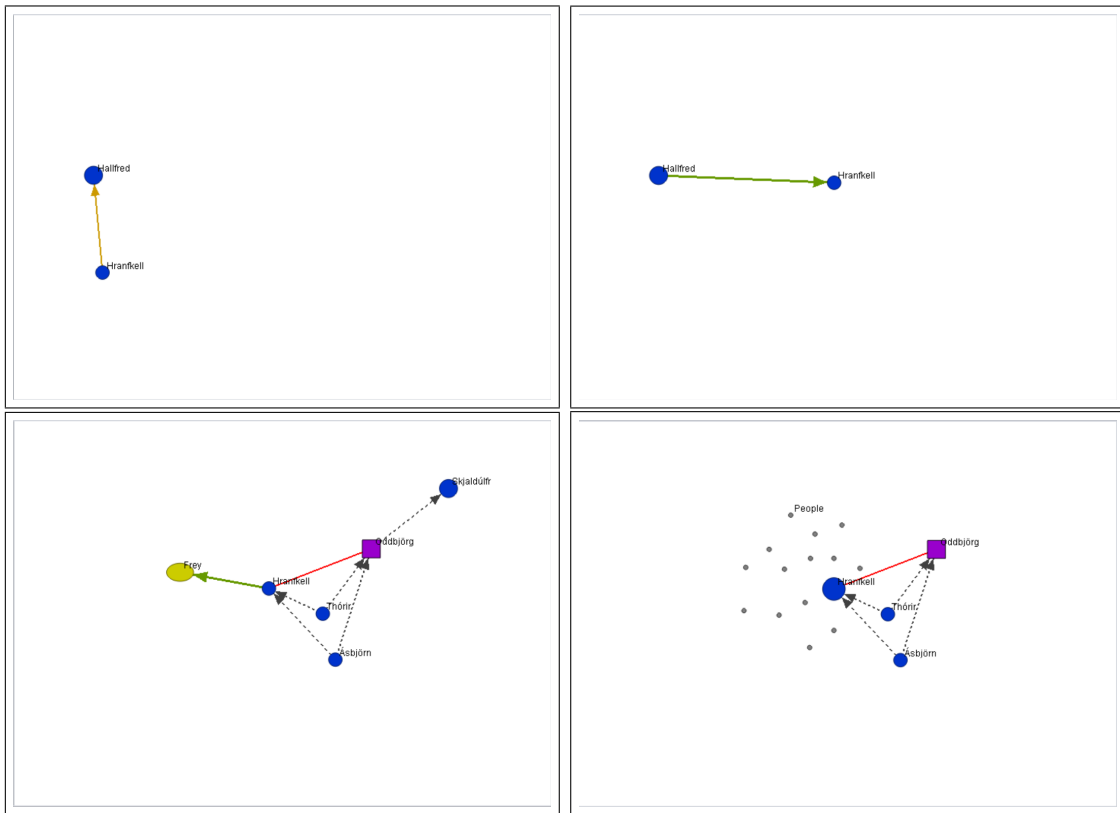
Κεφάλαιο 1: Ήταν στις μέρες του Βασιλιά Harold Fairhair που ένας άντρας έφερε

το πλοίο του στην Ισλανδία στο Breidal, με το όνομά του να είναι Hallfred. Το Breidal είναι μια εξοχή κάτω από αυτήν του Fljótsdalr. Στο πλοίο του ήταν η σύζυγος και ο γιος του, με το όνομα Hrafnkell, ο οποίος ήταν τότε δεκαπέντε χειμώνες, φιλόδοξος και καλός. Ο Hallfred δημιούργησε ένα σπίτι. Κατά τη διάρκεια του χειμώνα πέθανε μια υπηρέτρια, το όνομα της οποίας ήταν Arnthrádr, εξ ου και το όνομα του τόπου, Arnthrádr-stadir. Την άνοιξη ο Hallfred μετακίνησε το σπίτι του προς τα βόρεια και δημιούργησε ένα σπίτι σε ένα μέρος που ονομάζεται Geitdalr. Ένα βράδυ ονειρευόταν ότι ήρθε ένας άντρας σε αυτόν, και τον συμβούλευσε να μετακινηθεί μακρινά δυτικά απέναντι από το Lagarfljót. Στη συνέχεια ξύπνησε και μάζεψε τα υπάρχοντα του για ένα μέρος, το οποίο έκτοτε ονομάστηκε Hallfred-stadir, και εκεί έζησε ως τα γηρατεία.

Κεφάλαιο 2: Ο Hrafnkell στις περιηγήσεις του ανακάλυψε μια συγκεκριμένη κενή κοιλάδα πέρα από το Jokuldalr, η οποία του φάνηκε να είναι ένας καλύτερος οικισμός από άλλες κοιλάδες που είχε ήδη δει. Και όταν επέστρεψε στο σπίτι, ζήτησε από τον πατέρα του βοήθεια, λέγοντας, ότι είχε στο μυαλό να δημιουργήσει ένα καινούριο σπίτι σ' αυτήν την κοιλάδα. Αυτό του έδωσε ο πατέρας του, και στην κοιλάδα που είχε βρει, έκανε μια κατοικία για τον εαυτό του, την οποία ονόμασε Adalból. Ο Hrafnkell πήρε για σύζυγό του την Oddbjorg, κόρη του Skjaldúlfir, από το Laxárdalr, με την οποία γέννησε δύο γιους, τον μεγαλύτερο Thórir και τον νεότερο Ásbjorn. Ο Hrafnkell πραγματοποίησε μια μεγάλη γιορτή θυσίας, και επίσης έχτισε έναν μεγάλο ναό. Ο Hrafnkell δεν αγαπούσε κανέναν άλλο θεό πέρα από τον Frey, και σε αυτόν έκανε θυσίες όλων των καλύτερων πραγμάτων που είχε. Ο Hrafnkell εγκαταστήθηκε στην κοιλάδα, προσφέροντας έδαφη σε άλλους ανθρώπους, υπό την προϋπόθεση ότι θα ήταν ο αρχηγός τους. Έτσι, το όνομα του μεγάλωσε και ονομάστηκε Freysgodi.



Σχήμα 42: Το δυναμικό γράφημα για τις χρονικές στιγμές 0 έως 4 (Κεφάλαιο 1)

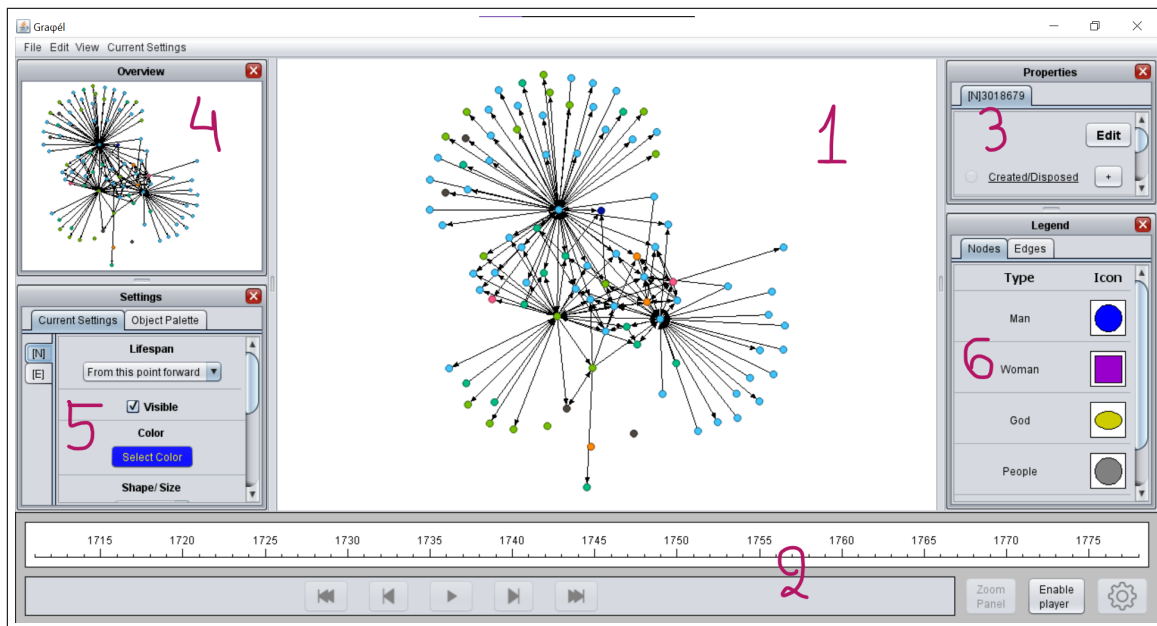


Σχήμα 43: Το δυναμικό γράφημα για τις χρονικές στιγμές 5 έως 8 (Κεφάλαιο 2)

6 Επίλογος

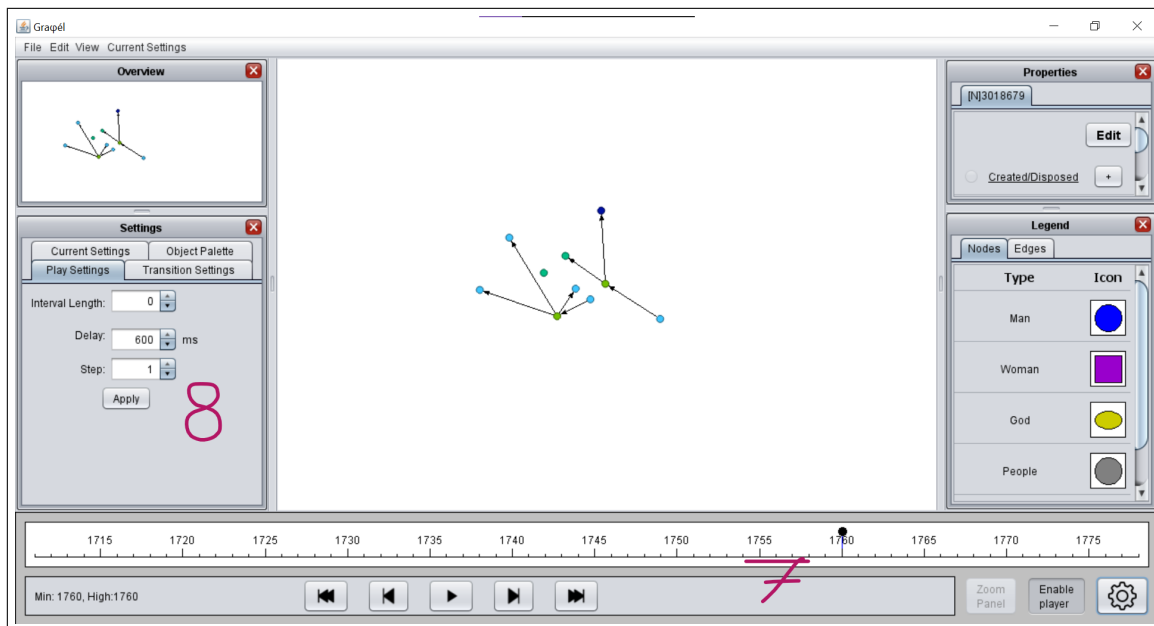
6.1 Σύνοψη

Η διπλωματική εργασία είχε ως αντικείμενο την υλοποίηση ενός λογισμικού για την απεικόνιση και την επεξεργασία δυναμικών γραφημάτων. Τα μέρη του λογισμικού περιγράφηκαν στα κεφάλαια 2, 3 και 4. Στο δεύτερο κεφάλαιο ασχοληθήκαμε με τα στοιχεία του λογισμικού που δεν έχουν σχέση με την απεικόνιση ή την επεξεργασία ενός δυναμικού γραφήματος, αλλά είναι βασικά στοιχεία της εφαρμογής. Στο τρίτο κεφάλαιο ασχοληθήκαμε με την απεικόνιση ενός δυναμικού γραφήματος και με την μετάβαση του από μια χρονική στιγμή στην επόμενη. Στο τέταρτο κεφάλαιο περιγράψαμε τον τρόπο που μπορεί κάποιος να προσθέσει αντικείμενα στο δυναμικό γράφημα, αλλά και να τα επεξεργαστεί. Συνοψίζοντας, η εφαρμογή αποτελείται από τα εξής μέρη:



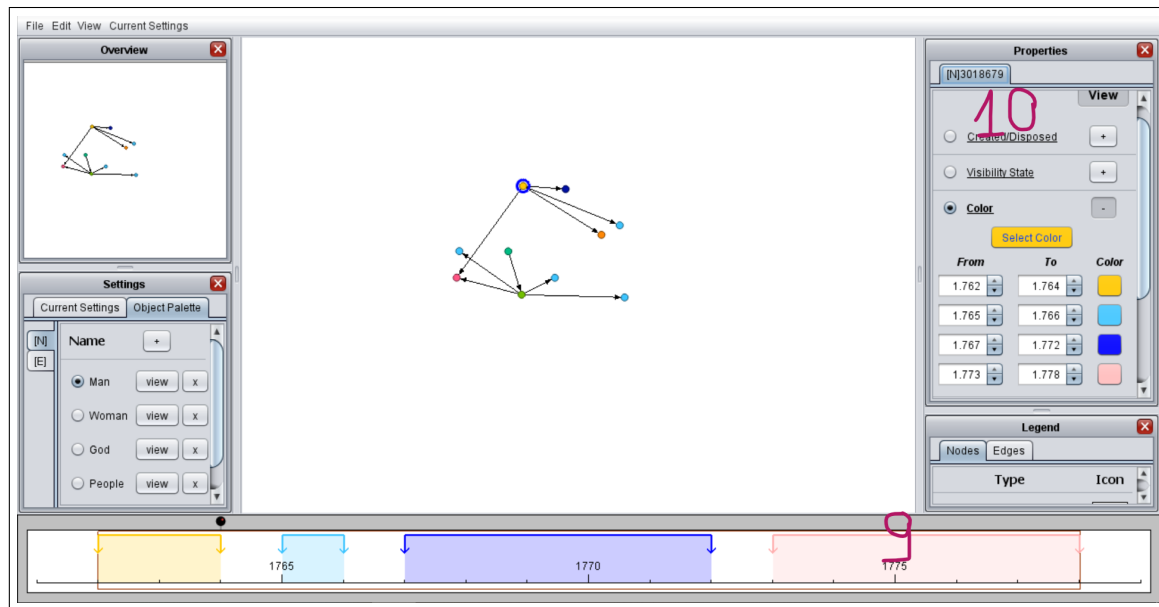
1. Τον καμβά, στον οποίον απεικονίζεται το δυναμικό γράφημα, είτε ολόκληρο είτε για κάποια συγκεκριμένη χρονική στιγμή (Κεφάλαιο 2.5.1).
2. Την μπάρα του χρόνου, όπου ένας βαθμονομημένος χάρακας δείχνει σε κλίμακα όλο το χρονικό εύρος του δυναμικού γραφήματος (Κεφάλαιο 2.1).
3. Το παράθυρο “Properties”, το οποίο αποτελείται από καρτέλες, όπου σε κάθε μία αναγράφονται όλες οι πληροφορίες για έναν κόμβο ή μια ακμή (Κεφάλαιο 2.5.2).

4. Το παράθυρο “Overview”, το οποίο δείχνει ανά πάσα στιγμή όλο το δυναμικό γράφημα, αλλά και πιο κομμάτι του είναι ορατό στον καμβά (Κεφάλαιο 2.5.3).
5. Το παράθυρο με τις ρυθμίσεις, όπου ανοίγουν καρτέλες με όλες τις δυνατές ρυθμίσεις της εφαρμογής, όπως:
 - τις προεπιλεγμένες ιδιότητες για τα αντικείμενα που προστίθενται (“Current Settings”) (Κεφάλαιο 4.1.1) και
 - την παλέτα των τύπων αντικειμένων, όπου ο χρήστης μπορεί να επιλέξει έναν τύπο για να εισάγει, αλλά και να προσθέσει κάποιον καινούριο (Κεφάλαιο 4.1.2).
6. Το υπόμνημα της παλέτας τύπων, όπου φαίνονται όλοι οι αποθηκευμένοι τύποι αντικειμένων με τα εικονίδια τους (Κεφάλαιο 4.1.2).

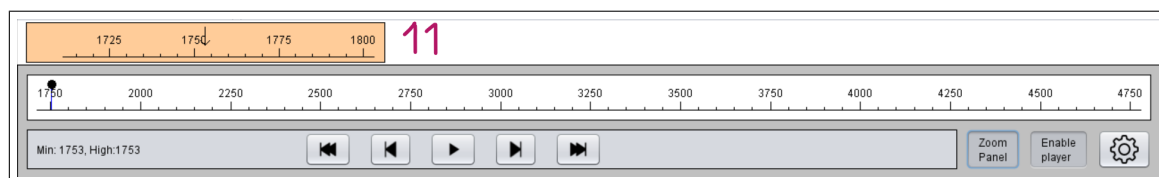


7. Τον δρομέα που δείχνει την χρονική στιγμή στην οποία βρίσκεται η απεικόνιση του δυναμικού γραφήματος, τα κουμπιά για τις λειτουργίες βίντεο και το κουμπί με το γρανάζι με το οποίο μπορεί ο χρήστης να ανοίξει ρυθμίσεις που αφορούν την δυναμική απεικόνιση (Κεφάλαιο 3.1).
8. Τις καρτέλες με τις ρυθμίσεις που αφορούν τις ρυθμίσεις για την δυναμική απεικόνιση, οι οποίες όταν απενεργοποιηθεί η δυναμική απεικόνιση κλείνουν αυτόματα και είναι:

- οι ρυθμίσεις που αφορούν την ταχύτητα του «βίντεο» (Κεφάλαιο 3.1) και
- οι ρυθμίσεις που αφορούν την ταχύτητα της ομαλής μετάβασης από μία χρονική στιγμή σε επόμενη (Κεφάλαιο 3.2).



9. Τα ορθογώνια παραλληλόγραμμα που δείχνουν τις διαφορετικές τιμές που παίρνει μια ιδιότητα του αντικειμένου προς επεξεργασία και την πινέζα που «δείχνει» την χρονική στιγμή που βρίσκεται η απεικόνιση (Κεφάλαιο 4.2).
10. Το παράθυρο “Properties”, στο οποίο κατά την επεξεργασία ενός αντικειμένου μπορεί ο χρήστης να προσθέσει μια διαφορετική τιμή ή να επεξεργαστεί τις ήδη υπάρχουσες τιμές της ιδιότητας (Κεφάλαιο 4.2).



11. Την μικρή μπάρα του χρόνου που εμφανίζεται όταν το χρονικό εύρος του δυναμικού γραφήματος είναι πολύ πυκνό (Κεφάλαιο 3.3).

6.2 Συμπεράσματα

Όσα περιγράφηκαν στην παρούσα Διπλωματική Εργασία αφορούν την ανάπτυξη ενός λογισμικού για την απεικόνιση και την επεξεργασία δυναμικών γραφημάτων.

Αρχικός στόχος ήταν, κυρίως, η μετάβαση ενός δυναμικού γραφήματος από τη μία χρονική στιγμή στην επόμενη να είναι ομαλή και ο χρήστης να μπορεί να αντιληφθεί τις μεταβολές που γίνονται στο δυναμικό γράφημα εύκολα. Όσον αφορά την υλοποίηση, η τεχνική του Multithreading υπήρξε ένα πολύ πολύτιμο εργαλείο για τον συγχρονισμό των κινήσεων, ώστε το λογισμικό να μην «κολλάει», όλες οι μεταβολές που πραγματοποιούνται στο δυναμικό γράφημα να φαίνονται ομαλές για τον χρήστη και το «βίντεο» της δυναμικής απεικόνισης να μην περιέχει απότομες αλλαγές, οι οποίες θα αποσπούν το μάτι.

Στη πορεία της υλοποίησης προέκυψε η ανάγκη για την δημιουργία και την επεξεργασία ενός δυναμικού γραφήματος. Έτσι, δημιουργήθηκε ένας δεύτερος στόχος για την παρούσα διπλωματική, ο οποίος ήταν η εισαγωγή και η επεξεργασία αντικειμένων να γίνεται γραφικά, δηλαδή να γίνεται με εύκολο και εύχρηστο τρόπο. Ως αποτέλεσμα, ένα μεγάλο μέρος του λογισμικού αφορά λειτουργίες που διευκολύνουν την εισαγωγή ή την επεξεργασία κόμβων και ακμών.

6.3 Μελλοντικές Επεκτάσεις

Πιθανές μελλοντικές επεκτάσεις του λογισμικού που περιγράφεται στην παρούσα διπλωματική εργασία είναι:

- Η ταχύτητα του βίντεο να είναι λογαριθμική σε σχέση με το πλήθος συμβάντων ανά χρονική στιγμή. Σε χρονικές στιγμές, δηλαδή, όπου στο δυναμικό γράφημα δεν συμβαίνουν πολλές μεταβολές το βίντεο θα έχει μεγαλύτερη ταχύτητα, ενώ σε αντίθετη περίπτωση, δηλαδή σε χρονικές στιγμές που γίνονται πολλές αλλαγές στο δυναμικό γράφημα το βίντεο θα «τρέχει» πιο αργά.
- Ο χρήστης να μπορεί να εισάγει και να επεξεργαστεί καινούριες ιδιότητες για τους κόμβους και τις ακμές, όπως αυτές περιγράφονται και στο format `.grdyn` (Κεφάλαιο 2.4).
- Η παράμετρος `step` της δυναμικής απεικόνισης να μην είναι σταθερή, αλλά να μεταβάλλεται. Η απεικόνιση του δυναμικού γραφήματος, δηλαδή, να μην αφορά όλες τις χρονικές στιγμές, αλλά να μπορεί ο χρήστης να επιλέξει ένα σύνολο

αυτών. Η απεικόνιση θα λαμβάνει, στη συνέχεια, υπ' όψιν της μόνο τις επιλεγμένες χρονικές στιγμές και η μετάβαση του δυναμικού γραφήματος θα γίνεται από τη μία χρονική στιγμή του συνόλου στην επόμενη.

- Να προστεθούν αλγόριθμοι για την ανάλυση του δυναμικού γραφήματος, όπως εύρεση συνεκτικών συνιστωσών, εύρεση ισχύρα συνεκτικών συνιστωσών, μεταβολή ιδιοτήτων σε βάθος χρόνου, εύρεση μοτίβων στις ιδιότητες των αντικειμένων κλπ. Επίσης, να απεικονίζονται τα clusters (συστάδες) από κόμβους είτε μέσω κάποιου αλγορίθμου είτε μέσω των δεδομένων του χρήστη.
- Στην μπάρα του χρόνου, να απεικονίζεται στο φόντο μια συνάρτηση, η οποία θα αφορά κάποια ιδιότητα του δυναμικού γραφήματος συναρτήσει των χρονικών στιγμών. Μια τέτοια συνάρτηση θα μπορούσε να είναι το πλήθος των αντικειμένων, ο μέσος όρος βαθμού των κόμβων ή το πλήθος των συνεκτικών συνιστωσών ανά χρονική στιγμή.

Βιβλιογραφία

- [1] Jae-wook Ahn, Catherine Plaisant, and Ben Shneiderman. “A Task Taxonomy for Network Evolution Analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 20 (Mar. 2014), pp. 365–. DOI: 10.1109/TVCG.2013.238.
- [2] Petra Mutzel et al. “A Library of Algorithms for Graph Drawing”. In: vol. 1547. Jan. 1998, pp. 456–457. DOI: 10.1007/3-540-37623-2_44.
- [3] Natalia Andrienko and Gennady Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Jan. 2006. ISBN: 978-3-540-25994-7. DOI: 10.1007/3-540-31190-4.
- [4] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. “GraphDiaries: Animated transitions and temporal navigation for dynamic networks.”. In: *IEEE Transactions on Visualization and Computer Graphics, Institute of Electrical and Electronics Engineers* 20 (5) (2014), pp. 740–754. DOI: 10.1109/TVCG.2013.254.
- [5] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. “Gephi: An Open Source Software for Exploring and Manipulating Networks”. In: *Proc. ICWSM* (2009). URL: <http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154>.
- [6] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. “A Taxonomy and Survey of Dynamic Graph Visualization”. In: *Computer Graphics* 00 (2016), pp. 1–27. DOI: 10.1111/cgf.12791.
- [7] Stefan Gladisch, Ulrike Kister, Christian Tominski, Raimund Dachsel, and H. Schumann. *Mapping Tasks to Interactions for Graph Exploration and Graph Editing on Interactive Surfaces*. Apr. 2015.
- [8] yWorks GmbH. *yEd*. Version 3.19. Mar. 19, 2019. URL: <https://www.yworks.com/products/yed>.
- [9] *Graph Drawing Creative Topics*. URL: <http://mozart.diei.unipg.it/gdcontest/contest2020/topics.html>.
- [10] Michael Jünger and Petra Mutzel. *Graph Drawing Softwares*. Jan. 2003. DOI: 10.1007/978-3-642-18638-7.
- [11] Natalie Kerracher, Jessie Kennedy, and Kevin Chalmers. “A Task Taxonomy for Temporal Graph Visualisation”. In: *IEEE Transactions on Visualization and Computer Graphics* 21 (Oct. 2015), pp. 1–1. DOI: 10.1109/TVCG.2015.2424889.

- [12] Corey L. Lanum. *Visualizing Graph Data*. Manning Publications, 2016. ISBN: 9781617293078.
- [13] Alexandra Lee, Daniel Archambault, and Miguel Nacenta. “Dynamic Network Plaid: A Tool for the Analysis of Dynamic Networks”. In: Apr. 2019, pp. 1–14. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300360.
- [14] Petra Mutzel, Michael Jünger, and Sebastian Leipert. *Graph Drawing, 9th International Symposium, GD 2001 Vienna, Austria, September 23-26, 2001, Revised Papers*. Jan. 2002. DOI: 10.1007/3-540-45848-4.
- [15] *Palladio*. *Visualize complex historical data with ease*. URL: <https://hdlab.stanford.edu/palladio/>.
- [16] Roberto Tamassia. “Handbook of Graph Drawing and Visualization”. In: *AMC* 10 (Jan. 2014).
- [17] *The Story of Hrafnkell, Frey’s Priest*. URL: https://sagadb.org/hrafnkells_saga_freysgoda.en.
- [18] Norman Walsh. *A Technical Introduction to XML*. October 3, 1998. URL: <https://www.xml.com/pub/a/98/10/guide0.html#AEN58>.
- [19] Roland Wiese, Markus Eiglsperger, and Michael Kaufmann. “yFiles: Visualization and Automatic Layout of Graphs”. In: vol. 2265. Sept. 2001. ISBN: 978-3-540-43309-5. DOI: 10.1007/3-540-45848-4_42.