



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**ΕΚΠΑΙΔΕΥΤΙΚΟΣ ΠΡΟΣΟΜΟΙΩΤΗΣ ΑΛΓΟΡΙΘΜΩΝ ΚΑΙ
ΔΟΜΩΝ ΔΕΔΟΜΕΝΩΝ**

Διπλωματική Εργασία

Βογιατζόπουλος Δημήτριος

Διαμάντης Γεώργιος

Επιβλέπων: Τσαλαπάτα Χαρίκλεια, Ε.Δι.Π, Πανεπιστήμιο Θεσσαλίας

Βόλος 2021



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**ΕΚΠΑΙΔΕΥΤΙΚΟΣ ΠΡΟΣΟΜΟΙΩΤΗΣ ΑΛΓΟΡΙΘΜΩΝ ΚΑΙ
ΔΟΜΩΝ ΔΕΔΟΜΕΝΩΝ**

Διπλωματική Εργασία

Βογιατζόπουλος Δημήτριος

Διαμάντης Γεώργιος

Επιβλέπων: Τσαλαπάτα Χαρίκλεια, Ε.Δι.Π, Πανεπιστήμιο Θεσσαλίας

Βόλος 2021



UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**A VISUALIZER FOR LEARNING DATA STRUCTURES AND
BASIC ALGORITHMS**

Diploma Thesis

Vogiatzopoulos Dimitrios

Diamantis Georgios

Supervisor: Tsalapata Hariklia, Laboratory Teaching Staff, University of
Thessaly

Volos 2021

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπουσα **Τσαλαπάτα Χαρίκλεια**
Μέλος Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Θάνος Γεώργιος**
Μέλος Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Τουσίδου Ελένη**
Μέλος Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: 23-2-2021

ΕΥΧΑΡΙΣΤΙΕΣ ή ΣΧΟΛΙΑ

Με την ολοκλήρωση της μεταπτυχιακής διπλωματικής μας εργασίας, θα θέλαμε να εκφράσουμε τις θερμές μας ευχαριστίες σε όλους όσους συνέβαλλαν στην εκπόνησή της.

Ευχαριστούμε θερμά την επιβλέπουσα καθηγήτριά μας, κυρία Τσαλαπάτα Χαρίκλεια, για την εμπιστοσύνη που μας έδειξε εξ' αρχής, αναθέτοντάς μας το συγκεκριμένο θέμα, την επιστημονική της καθοδήγηση, τις υποδείξεις της, την επιμονή της, το αμείωτο ενδιαφέρον της, τη συμπαράστασή της, τη συνεχή της υποστήριξη που έδειξε από την αρχή μέχρι το τέλος.

Επίσης, ευχαριστούμε τον καθηγητή, κύριο Θάνο Γεώργιο και την καθηγήτρια, κυρία Τουσίδου Ελένη, για τις εποικοδομητικές τους υποδείξεις και την πολύτιμη συμβολή τους στην ολοκλήρωση αυτής της εργασίας, ως μέλη της τριμελούς επιτροπής.

Τέλος, θα ήθελα εκφράσουμε την ευγνωμοσύνη μας στην οικογένειά μας για όλη τη στήριξη, τη συμπαράσταση και την κατανόησή τους, καθ' όλη τη διάρκεια των σπουδών μας.

**ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ
ΔΙΚΑΙΩΜΑΤΩΝ**

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο Δηλών

Ο Δηλών

(Υπογραφή)

Βογιατζόπουλος Δημήτριος

Ημερομηνία

(Υπογραφή)

Διαμάντης Γεώργιος

Ημερομηνία

ΠΕΡΙΛΗΨΗ

Η ραγδαία τεχνολογική εξέλιξη των τελευταίων χρόνων έχει επηρεάσει αρκετά τον τομέα της εκπαίδευσης. Ανέκαθεν ο εκσυγχρονισμός της εκπαίδευσης και η εξυγίανση της παραδοσιακής διδασκαλίας είναι ένα θέμα το οποίο ταλανίζει την πανεπιστημιακή κοινότητα. Ειδικότερα την περίοδο στην οποία διανύουμε η πανεπιστημιακή εκπαίδευση καλείτε να ανταπεξέλθει σε μεγάλες δυσκολίες και προβλήματα εισάγοντας νέους τρόπους μάθησης οι οποίοι βασίζονται στην τεχνολογία. Νέοι τρόποι διδασκαλίας όπως είναι η εξ αποστάσεως μάθηση, το ανοικτό πανεπιστήμιο, η ηλεκτρονική και κινητή μάθηση αλλάζουν τον παραδοσιακό τρόπο μάθησης[1]. Στην παρούσα διπλωματική εργασία, γίνεται ανάλυση του σχεδιασμού ενός προσομοιωτή που δημιουργήσαμε με στόχο την εκμάθηση και την κατανόηση δομών δεδομένων και βασικών αλγορίθμων. Ο προσομοιωτής αυτός επιτρέπει στους εκπαιδευτικούς να αλληλοεπιδρούν με τους φοιτητές μέσω της παιχνιδοποιημένης μάθησης και να αυξήσουν τα μαθησιακά αποτελέσματα.

ABSTRACT

The rapid technological development of recent years has greatly influenced the field of education. The modernization of education and the consolidation of traditional teaching has always been an issue that plagues the university community. Especially in this period that we are going through, the university education is calling to deal with great difficulties and problems by introducing new ways of learning that are based on technology. New teaching methods such as distance learning, open university, e-learning and mobile learning are changing the traditional way of teaching. In this diploma thesis, we analyze the design of our simulator which helps students to learn and understand data structures and basic algorithms. This simulator allows teachers to interact with students through gamified learning and in this way they can increase learning outcomes.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	viii
ABSTRACT	x
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	xii
ΚΕΦΑΛΑΙΟ 1	1
ΕΙΣΑΓΩΓΗ	1
1.1 Γενικό πλαίσιο	1
1.2 Στόχος εργασίας	1
1.3 Οργάνωση του τόμου	2
ΚΕΦΑΛΑΙΟ 2	3
ΕΚΠΑΙΔΕΥΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ	3
2.1 Η παλιά δομή της τριτοβάθμιας εκπαίδευσης και τα προβλήματά της	3
2.2 Εκπαίδευση και νέες τεχνολογίες	4
2.3 Το μάθημα των Δομών Δεδομένων και αλγορίθμων	6
ΚΕΦΑΛΑΙΟ 3	9
ΠΑΙΧΝΙΔΟΠΟΙΗΜΕΝΗ ΜΑΘΗΣΗ	9
3.1 Εισαγωγή.....	9
3.2 Η παιχνιδοποιημένη μάθηση στην εκπαίδευση.....	10
3.3 Το περιβάλλον της παιχνιδοποιημένης μάθησης.....	12
3.4 Σχετική έρευνα	15
.....	19
3.5 Σχετικές εφαρμογές.....	20
ΚΕΦΑΛΑΙΟ 4	22
ΕΠΙΣΚΟΠΗΣΗ ΕΦΑΡΜΟΓΗΣ	22
4.1 Τεχνολογική επισκόπηση.....	22
4.2 Γλώσσα προγραμματισμού C#.....	24
4.3 Λειτουργική επισκόπηση	25
4.3.1 Περιγραφή των βασικών Δομών Δεδομένων	26
4.3.2 Λίστες	26
4.3.3 Στοιβές	31
4.3.4 Ουρές	34
4.3.5 Γράφοι.....	35
4.3.6 Δέντρα.....	37

4.4 Λειτουργίες εφαρμογής.....	41
4.4.1 Λίστες	42
4.4.2 Στοιβες	49
4.4.3 Δέντρα	57
4.4.4 Ουρές	71
4.4.5 Γράφοι.....	79
4.4.6 Παιχνίδι ερωτήσεων	88
4.4.7 Γενικές πληροφορίες.....	90
4.4.8 Project ALIEN.....	91
ΚΕΦΑΛΑΙΟ 5.....	93
ΣΥΝΟΨΗ.....	93
5.1 Συμπεράσματα	93
5.2 Μελλοντικές επεκτάσεις της εφαρμογής	93
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	95

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Γενικό πλαίσιο

Η ραγδαία τεχνολογική εξέλιξη των τελευταίων δεκαετιών έχει επιφέρει τεράστιες αλλαγές σε όλες τις πτυχές της ζωής των ανθρώπων. Οι αλλαγές αυτές παρατηρούνται τόσο σε κοινωνικοπολιτικό επίπεδο όσο και στην απλή καθημερινότητα και φυσικά δεν λείπουν και από το εργασιακό περιβάλλον. Το εκπαιδευτικό σύστημα αδυνατεί να συμβαδίσει με την τεχνολογία και να ανταπεξέλθει στις ανάγκες των νέων.

Για αυτό τον λόγο, δοκιμάζονται και υιοθετούνται νέοι και καινοτόμοι τρόποι διδασκαλίας και μάθησης. Στον επαγγελματικό τομέα προκειμένου ο νέος να οδηγηθεί στην επίλυση ποικίλων προβλημάτων χρειάζεται ένα νέο σύνολο ικανοτήτων και δεξιοτήτων. Δεν επαρκεί πλέον η επαλήθευση και η απλή αναπαραγωγή επαληθευμένης γνώσης που προσφέρει το παραδοσιακό σύστημα. Απαιτείται κριτική σκέψη, προσαρμοστικότητα και ικανότητα συνεργασίας. Τα τελευταία χρόνια έχουν δημιουργηθεί πολλοί και διαφορετικοί μέθοδοι διδασκαλίας με στόχο να προετοιμάσουν τους νέους να αντιμετωπίσουν τις νέες προκλήσεις του τεχνολογικού κόσμου. Μία από αυτές τις μεθόδους είναι η ένταξη της παιχνιδοποιημένης μάθησης στη διαδικασία της μάθησης.

1.2 Στόχος εργασίας

Στην παρούσα διπλωματική εργασία σχεδιάστηκε και υλοποιήθηκε ένας προσομοιωτής δομών δεδομένων και αλγορίθμων. Απώτερος σκοπός είναι η ένταξη του προσομοιωτή στην εκπαίδευση. Δίνεται η δυνατότητα στους εκπαιδευτικούς να εντάξουν τη διαδικασία της μάθησης σε ένα ψηφιακό περιβάλλον. Ο προσομοιωτής αυτός είναι αρκετά εύχρηστος

και εξασφαλίζει την ενασχόληση των φοιτητών με το συγκεκριμένο μάθημα καθώς αυξάνεται το ενδιαφέρον τους μέσα από στοιχεία ψηφιακών παιχνιδιών.

1.3 Οργάνωση του τόμου

Στο κεφάλαιο 1 υπάρχει η εισαγωγή της διπλωματικής, όπου αναλύεται η ιδέα και ο στόχος της. Στο κεφάλαιο 2 περιγράφεται η εκπαιδευτική διαδικασία στις μέρες μας τα προβλήματά της και αναλύεται πως μπορεί η ιδέα μας να ενταχθεί στην εκπαίδευση και να βελτιώσει τη διαδικασία της μάθησης. Στο κεφάλαιο 3 αναλύεται ο όρος παιχνιδιοποιημένη μάθηση καθώς και των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Στο κεφάλαιο 4 περιγράφεται αναλυτικά η δομή και η λειτουργικότητα της εφαρμογής μας, και τέλος στο κεφάλαιο 5 βρίσκονται τα συμπεράσματα και οι μελλοντικές επεκτάσεις της διπλωματικής μας εργασίας.

ΚΕΦΑΛΑΙΟ 2

ΕΚΠΑΙΔΕΥΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

2.1 Η παλιά δομή της τριτοβάθμιας εκπαίδευσης και τα προβλήματά της

Η λέξη εκπαίδευση αναφέρεται στην παιδεία που παρέχει το εκπαιδευτικό σύστημα στις μέρες μας. Ωστόσο η εκπαίδευση αποσκοπεί στην παροχή γνώσεων, στην καλλιέργεια της νόησης, στη μόρφωση, στη διαμόρφωση προσωπικότητας καθώς και στη διάπλαση χαρακτήρα [1]. Η σημασία της παιδείας είναι αδιαμφισβήτητη διότι τα οφέλη της είναι πολλαπλά.

Πιο συγκεκριμένα οι αξίες που μπορεί να αποκομίσει ένας νέος από την τριτοβάθμια εκπαίδευση είναι αξιοσημείωτες [2]. Ειδικότερα η εμπειρία των πανεπιστημιακών σπουδών φέρνει το άτομο σε επαφή με μία διαφορετική προσέγγιση της γνώσης από αυτή που είχε γνωρίσει στο πλαίσιο της δευτεροβάθμιας εκπαίδευσης. Πλέον ο κύριος στόχος του νέου είναι η κεκτημένη γνώση μέσα από την διαδικασία της έρευνας. Εξαιτίας των τεράστιων πανεπιστημιακών εγχειριδίων ο φοιτητής πλέον συνειδητοποιεί πως η αποστήθιση είναι ανώφελη και οδηγείται στην κατανόηση και στην εμβάθυνση του αντικειμένου μελέτης του. Επιπλέον οι πανεπιστημιακές σπουδές συμβάλουν στην πληρέστερη διαμόρφωση της προσωπικότητας και των απόψεων του νέου, διότι προσφέρουν στον νέο ένα πεδίο ώριμης έκφρασης. Με αυτόν τον τρόπο ο φοιτητής αποκτά την ικανότητα του επαγγελματισμού, καθώς αντιλαμβάνεται το επίπεδο αυτοπειθαρχίας που απαιτείται προκειμένου να κατακτήσει το επίπεδο γνώσης που επιθυμεί. Τέλος οι πανεπιστημιακές σπουδές μπορούν να λειτουργήσουν ως απαραίτητο εφόδιο για την επαγγελματική αποκατάσταση του νέου και ένα να αποτελέσουν ένα πολύ σημαντικό βήμα για την επιστημονική κατάρτιση και εκπαίδευση του νέου.

Ωστόσο η τριτοβάθμια εκπαίδευση στη χώρα μας βρίσκεται αντιμέτωπη με ποικίλα προβλήματα που υπονομεύουν την ποιότητα των σπουδών και θέτουν σε αμφισβήτηση την πραγματική δυνατότητα των πανεπιστημιακών ιδρυμάτων [2]. Σε μια συνεχώς

εξελισσόμενη κοινωνία η τριτοβάθμια αναγκάζεται να εστιάζει στον τομέα της έρευνας και της καινοτομίας. Ποικίλες καινοτόμες ιδέες και προϊόντα προέρχονται από τα σημερινά πανεπιστήμια. Επίσης από τα πανεπιστήμια αναμένεται πλέον υψηλότερη εξειδίκευση στα προγράμματα σπουδών τους, προκειμένου οι απόφοιτοί τους να μπορούν να αξιοποιηθούν γρήγορα στους εργασιακούς τομείς.

Τα ζητούμενα αυτά προσκρούουν σε θεμελιώδη προβλήματα της τριτοβάθμιας εκπαίδευσης. Πρώτον και κύριων η έλλειψη οργάνωσης χαρακτηρίζει την εκπαίδευση του σήμερα με αποτέλεσμα να προκαλούνται σοβαρά προβλήματα στην ομαλή λειτουργία των ιδρυμάτων. Επιπλέον αισθητή είναι η απουσία ενδιαφέροντος από την πλευρά των φοιτητών. Το γεγονός αυτό συμβαίνει διότι οι ίδιοι οι φοιτητές δεν εισάγονται πάντα σε ότι αποτελούσε την πρώτη τους επιλογή, επειδή ποτέ δεν έλαβαν την κατάλληλη καθοδήγηση και δεν έχουν ανακαλύψει με ορθό τρόπο τις κλίσεις τους και τις δεξιότητες τους. Επίσης η διδασκαλία πολλές φορές πραγματοποιείται με παρωχημένο τρόπο. Οι εκπαιδευτικοί στηρίζουν την διδασκαλία τους πολλές φορές σε ξεπερασμένα συγγράμματα και η μάθηση είναι απλώς μια διαδικασία αναπαραγωγής επαληθευμένης γνώσης. Όλα αυτά αποθαρρύνουν τον φοιτητή να συμμετέχει ενεργά στην μάθηση μετατρέποντας τον σε έναν άβουλο και αδιάφορο άνθρωπο.

Ένας τρόπος αντιμετώπισης όλων αυτών των προβλημάτων που αντιμετωπίζει η σημερινή τριτοβάθμια εκπαίδευση είναι να εισάγει στον τρόπο διδασκαλίας νέες τεχνολογίες και νέες υποδομές οι οποίες συμβαδίζουν με την ραγδαία τεχνολογική εξέλιξη. Με αυτόν τον τρόπο ο σύγχρονος νέος θα αποκτήσει ξανά ενδιαφέρον για την μάθηση και θα συμμετέχει ενεργά στην κατάκτηση της γνώσης και στην διαδικασία της έρευνας.

2.2 Εκπαίδευση και νέες τεχνολογίες

Διανύουμε μια εποχή στην οποία τα πάντα αλλάζουν. Το γεγονός αυτό σχετίζεται άμεσα με τον αυξανόμενο ρυθμό της τεχνολογίας, τις νέες εφευρέσεις και τις νέες επινοήσεις. Τα γεγονότα αυτά δεν αφήνουν ανεπηρέαστο τον τομέα της εκπαίδευσης. Η ελληνική τριτοβάθμια εκπαίδευση θα πρέπει να εξοπλιστεί με τα κατάλληλα εφόδια προκειμένου

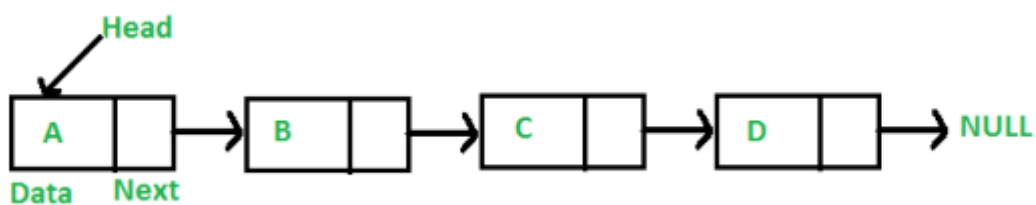
να συμβαδίσει με την ραγδαία τεχνολογική εξέλιξη και να χρησιμοποιήσει τις νέες τεχνολογίες σαν τρόπο διδασκαλίας.

Οι νέες τεχνολογίες μπορούν να επηρεάσουν ιδιαίτερα θετικά την εκπαιδευτική διαδικασία [3]. Αρχικά οι χρήση νέων τεχνολογιών στη διαδικασία του μαθήματος, όπως για παράδειγμα η χρήση εικόνων και πολυμέσων διεγείρουν το ενδιαφέρον του φοιτητή και ο ίδιος δεν αφαιρείται την ώρα της παράδοσης. Επίσης η επαφή του φοιτητή με αντικείμενα εκτός των παραδοσιακών μέσων διδασκαλίας και η επεξήγηση παραδειγμάτων μέσα από νέες τεχνολογίες τον βοηθούν να αποκτήσει μια πιο κατανοητή και ολοκληρωμένη εικόνα για το μάθημα. Έτσι δημιουργούνται πιο ευχάριστες και δημιουργικές εμπειρίες για τον φοιτητή. Επιπλέον δημιουργείται ένα περιβάλλον που πέρα από την παιδαγωγική του διάσταση παρέχει τη δυνατότητα στους φοιτητές επανεξέτασης και αναθεώρησης γενικών προβλημάτων στους χώρους μάθησης και εκπαίδευσης. Γενικότερα με την χρήση νέων τεχνολογιών αντιμετωπίζεται ένα από τα σοβαρότερα προβλήματα που χαρακτηρίζει την σημερινή εκπαίδευση, η αποστασιοποίηση και η αποξένωση των φοιτητών από τη διαδικασία της μάθησης.

Συμπερασματικά λοιπόν για την βελτίωση του εκπαιδευτικού συστήματος και την πιο ευχάριστη και δημιουργική ενασχόληση του μαθητή κατά τη διάρκεια της διδασκαλίας, θα πρέπει τεχνολογία και εκπαίδευση να συνυπάρξουν. Το εκπαιδευτικό σύστημα επιβάλλεται να αφομοιώσει όλα τα θετικά που προσφέρουν οι τεχνολογικές εφευρέσεις προκειμένου να αντιμετωπιστούν δυσμενή προβλήματα που αντιμετωπίζει η εκπαίδευση στις μέρες μας. Με αυτόν τον τρόπο θα μπορέσει να υπάρξει ένα άρτιο αποτέλεσμα στο ελληνικό εκπαιδευτικό σύστημα, το οποίο επομένως θα είναι πιο λειτουργικό και θα αποτελεί κίνητρο για τους φοιτητές [4].

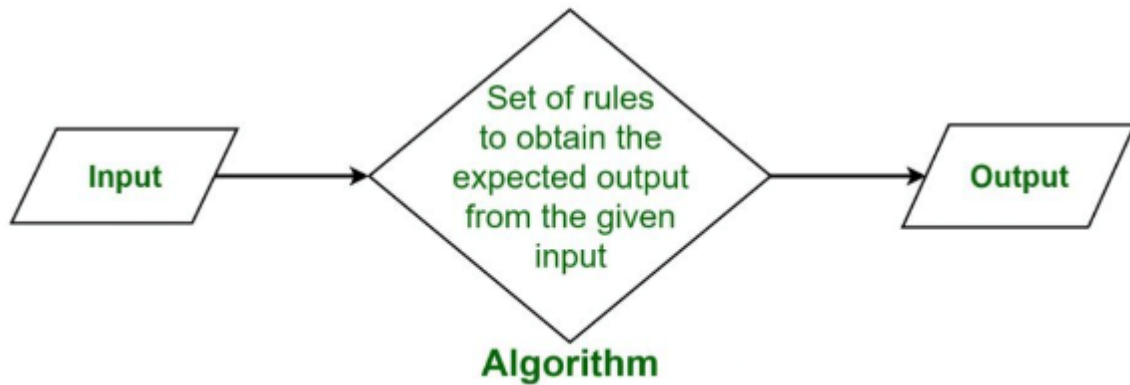
2.3 Το μάθημα των Δομών Δεδομένων και αλγορίθμων

Το μάθημα των Δομών Δεδομένων και αλγορίθμων αποτελεί ίσως και το βασικότερο μάθημα του τμήματος επιστήμης των υπολογιστών. Ο βασικός στόχος του μαθήματος είναι η μελέτη των βασικών δομών δεδομένων και αλγορίθμων. Η μελέτη περιλαμβάνει τη θεωρητική ανάλυσή τους καθώς επίσης και τις εφαρμογές της κάθε δομής. Πιο συγκεκριμένα μελετώνται πίνακες, λίστες, στοίβες, ουρές προτεραιότητας, δέντρα αναζήτησης, κατακερματισμός, Γράφοι, ταξινόμηση και άλλοι βασικοί αλγόριθμοι (Εικόνα 2.1). Οι δομές δεδομένων αποτελούν βασικά εργαλεία τα οποία χρησιμοποιούνται από αλγορίθμους για την επίλυση διαφόρων προβλημάτων (Εικόνα 2.2). Πιο συγκεκριμένα οι δομές δεδομένων μαζί με τους αλγορίθμους αποτελούν τα προγράμματα. Τα συγκεκριμένα προγράμματα στοχεύουν στην αποτελεσματική και αποδοτική οργάνωση των δεδομένων έτσι ώστε κάποιες λειτουργίες να εκτελούνται γρήγορα. Η χρήση καλύτερου hardware είναι η προφανής και όχι πάντα η καλύτερη λύση για τη μείωση του χρόνου εκτέλεσης μίας εφαρμογής ή την επίλυση ενός προβλήματος. Ωστόσο η χρήση κατάλληλης δομής και του κατάλληλου αλγορίθμου θεωρούνται πιο σημαντικά δεδομένα από την απόκτηση καλύτερου hardware και κυριότερος λόγος είναι το οικονομικό πρόβλημα. Ένα καλύτερο hardware στοιχίζει πολύ ενώ η δημιουργία ενός προγράμματος με την χρήση κατάλληλων δομών και αλγορίθμων στοιχίζει μόνο χρόνο.



Εικόνα 2.1 Παράδειγμα δομής δεδομένων [5]

What is Algorithm?

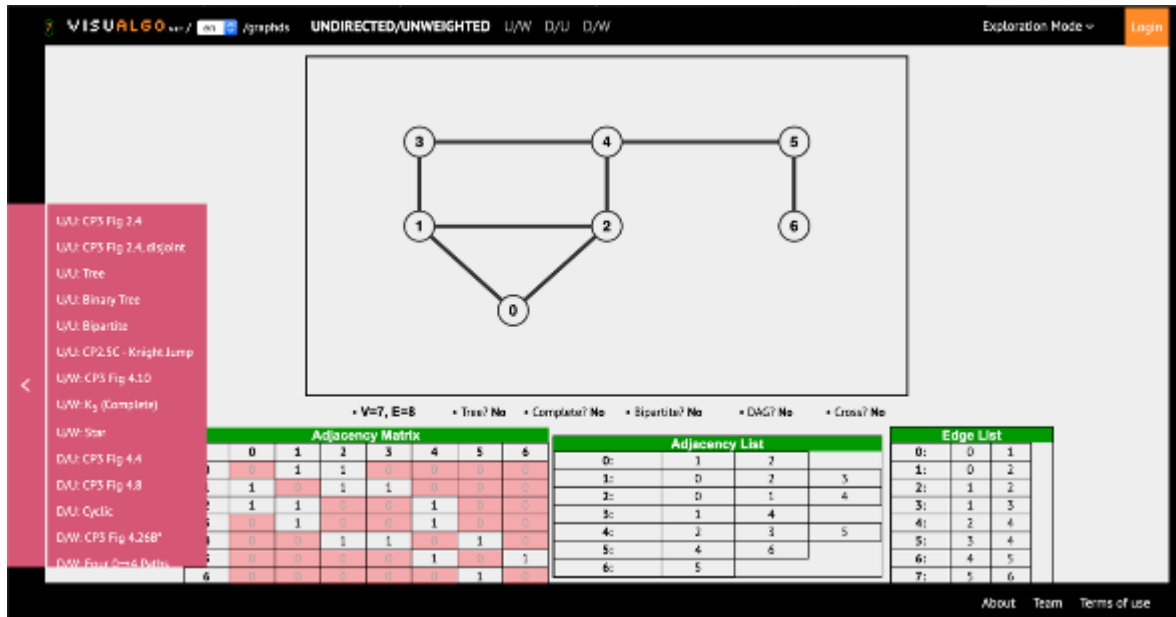


Εικόνα 2.2 Τι είναι ο αλγόριθμος [6]

Στα σημερινά πανεπιστήμια παρατηρείτε μία εξαιρετική δυσκολία στο συγκεκριμένο μάθημα. Όλο και περισσότεροι φοιτητές αδυνατούν να περάσουν επιτυχώς το μάθημα. Ο τρόπος διδασκαλίας του συγκεκριμένου μαθήματος πραγματοποιείται με παρωχημένο τρόπο καθώς οι διαλέξεις πραγματοποιούνται μέσα από κακογραμμένα συγγράμματα και μαυροπίνακες. Οι καθηγητές αρκετές φορές θεωρούνται εμπειρογνώμονες και η μάθηση καταλήγει να είναι μια διαδικασία απορρόφησης και αναπαραγωγής επαληθευμένης γνώσης. Με αυτόν τον τρόπο οι φοιτητές χάνουν το ενδιαφέρον τους και απομακρύνονται από την διαδικασία της έρευνας και την κατάκτηση της γνώσης και τελικά απογοητεύονται και αποτυγχάνουν στο μάθημα.

Προκειμένου να αντιμετωπιστεί κατάλληλα το συγκεκριμένο πρόβλημα πανεπιστήμια οφείλουν να εξοπλιστούν με νέες τεχνολογίες με απώτερο σκοπό την ριζική αλλαγή της διαδικασίας της μάθησης. Για παράδειγμα το μάθημα των δομών δεδομένων και αλγορίθμων πρέπει να σταματήσει να πραγματοποιείται με χρήση ξεπερασμένων τρόπων διδασκαλίας και να χρησιμοποιηθούν παιχνιδοποιημένες εφαρμογές, ψηφιακές εικόνες και πολυμέσα (Εικόνα 2.3). Ο συγκεκριμένος τρόπος διδασκαλίας συμβαδίζει με την

ραγδαία εξέλιξη της τεχνολογίας και είναι πιο φιλικός προς τους νέους, οι οποίοι έχουν μεγαλώσει στην λεγόμενη ψηφιακή εποχή. Με αυτόν τον τρόπο οι φοιτητές θα συμμετάσχουν ενεργά στην διαδικασία της μάθησης, θα κατακτήσουν την γνώση με έναν πιο ευχάριστο και δημιουργικό τρόπο.



Εικόνα 2.3 Παράδειγμα παιχνιδιοποιημένης εφαρμογής [7]

ΚΕΦΑΛΑΙΟ 3

ΠΑΙΧΝΙΔΟΠΟΙΗΜΕΝΗ ΜΑΘΗΣΗ

3.1 Εισαγωγή

Στις μέρες μας παρατηρείται αδιαμφισβήτητα μια ραγδαία εξέλιξη της τεχνολογίας. Διανύουμε μια ψηφιακή εποχή η οποία επηρεάζει άμεσα τον τομέα της εκπαίδευσης και ειδικότερα την εκπαίδευση στα πανεπιστήμια. Οι σημερινοί φοιτητές έχουν μεγαλώσει με πολλές τεχνολογίες της ψηφιακής εποχής [8]. Οι καθηγητές καλούνται να αντιμετωπίσουν νέες προκλήσεις και να επιλύσουν σημαντικά ζητήματα τα οποία σχετίζονται με την προσαρμογή της διδασκαλίας στις ανάγκες, τις προτιμήσεις και τις απαιτήσεις των φοιτητών, προκειμένου να είναι ενεργοί και να μην χάνουν το ενδιαφέρον τους για την αναζήτηση της γνώσης. Η παιχνιδοποιημένη μάθηση αποτελεί μία νέα μορφή διδασκαλίας κατά την οποία ο ίδιος ο φοιτητής οδηγείται στην κατάκτηση της γνώσης μέσω ψηφιακών παιχνιδιών.

Υπάρχουν αρκετές ανάγκες της εκπαίδευσης τις οποίες μπορεί να καλύψει η παιχνιδοποιημένη μάθηση [7]. Διάφορα εκπαιδευτικά προγράμματα και μαθήματα εισαγωγής μπορούν να πραγματοποιηθούν μέσα σε ένα ψηφιακό παιχνίδι. Με αυτόν τον τρόπο διάφορες εκπαιδευτικές διαδικασίες πραγματοποιούνται με έναν πιο δημιουργικό και ευχάριστο τρόπο. Με την παιχνιδοποιημένη μάθηση επίσης ενισχύονται επαγγελματικές δεξιότητες όπως είναι η επικοινωνία, η ικανότητα εργασίας υπό πίεση, η λήψη αποφάσεων, η ικανότητα προσαρμογής, η διαχείριση χρόνου, η ομαδική εργασία και η δημιουργικότητα. Όλα τα ψηφιακά παιχνίδια έχουν συγκεκριμένους κανόνες, αυτό έχει ως αποτέλεσμα τη συμμόρφωση σε κανονισμούς. Τέλος μία βασική ανάγκη της εκπαίδευσης που καλύπτεται είναι η παιχνιδοποίηση των αξιολογήσεων σε παραδοσιακά εξ' αποστάσεως μαθήματα.

Η παιχνιδοποιημένη μάθηση προσφέρει ποικίλα πλεονεκτήματα στην διαδικασία της μάθησης [9]. Πρώτον και κύριων μέσα από την διαδραστικότητα που προσφέρει ένα ψηφιακό παιχνίδι οι φοιτητές αποκτούν την δυνατότητα να απολαύσει μια ευχάριστη εμπειρία μάθησης. Μέσα από τις διάφορες προκλήσεις που υπάρχουν μέσα στο παιχνίδι τις οποίες πρέπει να αντιμετωπίσει οι χρήστες προκειμένου να κατακτήσουν μια καλή βαθμολογία και συνεπώς να έχουν μία καλή επίδοση, αυξάνεται το κίνητρό τους και με αυτόν τον τρόπο επέρχεται η εμπέδωση και η διατήρηση της γνώσης. Επίσης ο φοιτητής έχει την δυνατότητα άμεσης ανατροφοδότησης καθώς οι καθηγητές μπορούν να υποβάλουν σχόλια και βαθμούς σχετικά με την πρόοδο του μέσα στο παιχνίδι. Επιπλέον την παιχνιδοποιημένη μάθηση έχει ως βασικό στόχο την εξεύρεση λύσεων μέσω της αλλαγής συμπεριφοράς των χρηστών καθώς και την αύξηση της συμμετοχικότητας και της δέσμευσής τους. Τέλος οι χρήστες εμπλέκονται σε σενάρια που προσομοιάζουν καταστάσεις και συνθήκες ανάλογες με αυτές της καθημερινής ζωής. Με αυτόν τον τρόπο παίρνει αποφάσεις και δέχεται τις αντίστοιχες συνέπειες μέσα στο ασφαλές περιβάλλον του παιχνιδιού [10].

3.2 Η παιχνιδοποιημένη μάθηση στην εκπαίδευση

Προκειμένου η παιχνιδοποιημένη μάθηση να ενταχθεί αποτελεσματικά στην εκπαίδευση απαιτείται να αναπτυχθεί μια στρατηγική η οποία περιλαμβάνει μία βαθύτατη ανάλυση των καίριων συνθηκών και να χρησιμοποιηθούν συγκεκριμένα εργαλεία λογισμικού. Τα κύρια βήματα της στρατηγικής περιλαμβάνουν τα εξής: [11]

1. Προσδιορισμό των χαρακτηριστικών των μαθητών
2. Προσδιορισμό των μαθησιακών στόχων
3. Δημιουργία εκπαιδευτικού περιεχομένου και δραστηριοτήτων για παιχνιδοποιημένη μάθηση
4. Προσθήκη στοιχείων και μηχανισμών ψηφιακών παιχνιδιών

Προκειμένου να πραγματοποιηθούν και να εφαρμοστούν νέες προσεγγίσεις στη διαδικασία της μάθησης επιβάλλεται να προσδιοριστούν τα διάφορα «προφίλ» των φοιτητών και με αυτόν τον τρόπο θα καθοριστεί εάν οι νέες τεχνικές παιχνιδοποιημένης

μάθηση θα είναι αποτελεσματικές. Ο ανταγωνιστικός χαρακτήρας των φοιτητών αποτελεί το “κλειδί” στην όλη διαδικασία ένταξης της παιχνιδοποιημένης μάθησης στα πανεπιστήμια. Οι καθηγητές επιβάλλεται να λάβουν υπόψη τους και να σκεφτούν σοβαρά τις δραστηριότητες που πρόκειται να αναθέσουν στους φοιτητές. Οι δραστηριότητες αυτές είναι αναγκαίο να συμβαδίζουν με τις ικανότητες και δεξιότητες των φοιτητών. Εάν οι εργασίες είναι πολύ εύκολες ή πολύ δύσκολες είναι πιθανό να προκύψει ο υποβιβασμός των φοιτητών, το οποίο είναι ακριβώς το αντίθετο αποτέλεσμα από αυτό που θέλει ο καθηγητής να προσφέρει με την χρήση των gamify εφαρμογών. Το κίνητρο των φοιτητών να συμμετάσχουν ενεργά στην μάθηση εξαρτάται πλήρως από το πλαίσιο της μαθησιακής διδασκαλίας και το αποτέλεσμα που προκύπτει από τα επιτεύγματά τους [12].

Επίσης αναγκαίο είναι να προσδιοριστούν οι μαθησιακοί στόχοι. Ο σκοπός της εκπαίδευσης είναι ένας και βασικός, η επίτευξη των μαθησιακών στόχων, αλλιώς όλες οι δραστηριότητες που υλοποιούνται μέσω παιχνιδοποιημένης μάθησης θα είναι άσκοπες. Οι στόχοι καθορίζουν ποιο εκπαιδευτικό περιεχόμενο και ποιες δραστηριότητες θα περιληφθούν στην διαδικασία της μάθησης καθώς καθορίζουν και την επιλογή των κατάλληλων μηχανικών και των τεχνικών παιχνιδιών.

Επιπλέον η εκπαίδευση πρέπει να πραγματοποιείται με έναν πιο διαδραστικό και ελκυστικό τρόπο προκειμένου να τραβήξουν το ενδιαφέρον του χρήστη. Οι διαφορές εκπαιδευτικές δραστηριότητες που πραγματοποιούνται μέσω της παιχνιδοποιημένης μάθησης πρέπει να συμβαδίζουν με τους μαθησιακούς στόχους και να επιτρέπουν τα παρακάτω στοιχεία [13]. Οι μαθησιακές δραστηριότητες πρέπει να σχεδιαστούν με τέτοιο τρόπο ώστε οι φοιτητές να μπορούν να τις ολοκληρώσουν επιτυχώς και να έχουν τη δυνατότητα να τις επαναλάβουν σε περίπτωση αποτυχημένης προσπάθειας. Με αυτόν τον τρόπο οι χρήστες μέσα από τις επαναλήψεις θα βελτιώσουν αρκετές δεξιότητές τους. Επίσης οι μαθησιακές δραστηριότητες πρέπει να είναι εφικτές και προσαρμοσμένες στις δυνατότητες και τις δεξιότητες του χρήστη. Επιπλέον μια gamify εφαρμογή είναι αναγκαίο να προκαλεί τον χρήστη καθώς αυξάνεται το επίπεδο δυσκολίας στις εργασίες κατά τη διάρκεια του παιχνιδιού και τέλος πρέπει να δίνεται η δυνατότητα στους χρήστες να

επιλέξουν διαφορετικά μονοπάτια για την επίτευξη των στόχων του παιχνιδιού. Έτσι οι φοιτητές δημιουργούν τις δικές τους στρατηγικές και συμμετέχουν πιο ενεργά στη διαδικασία της μάθησης.

Τέλος με την προσθήκη μηχανισμών ψηφιακών παιχνιδιών, οι οποίοι στοχεύουν στην επίτευξη προκαθορισμένων μαθησιακών στόχων. Όλοι αυτοί οι μηχανισμοί που θα αναφερθούν παρακάτω αποτελούν το περιβάλλον της παιχνιδοποιημένης μάθησης.

3.3 Το περιβάλλον της παιχνιδοποιημένης μάθησης

Ο κύριος τρόπος ένταξης της παιχνιδοποιημένης μάθησης στην εκπαίδευση στο πανεπιστήμιο, είναι η εισαγωγή συγκεκριμένων μηχανισμών παιχνιδιού όπως πόντους, μετάλλια, πίνακες σκορ, γραφήματα απόδοσης, ιστορίες με νόημα, όμορφες εικόνες προφίλ και να υπάρχει δυνατότητα συνεργασίας με άλλους χρήστες [14]. Οι χρήστες με τους συγκεκριμένους μηχανισμούς παιχνιδιού αποκτούν ενδιαφέρον για τις διάφορες εφαρμογές παιχνιδοποιημένης μάθησης.

Οι πόντοι είναι ένα από τα βασικότερα στοιχεία που πρέπει να υπάρχουν στις gamified εφαρμογές. Οι πόντοι ανταμείβονται κάτω από συγκεκριμένες περιστάσεις, όταν για παράδειγμα ο χρήστης ολοκληρώσει ορισμένες δραστηριότητες μέσα στο περιβάλλον του παιχνιδιού. Οι πόντοι με λίγα λόγια αντιπροσωπεύουν αριθμητικά την πρόοδο του χρήστη. Υπάρχουν διάφοροι τύποι πόντων όπως πόντοι φήμης, πόντοι εξαργύρωσης, πόντοι εμπειρίας, καθένας από τους οποίους εξυπηρετεί τον δικό του σκοπό. Ο βασικότερος σκοπός της χρήσης των πόντων είναι η μέτρηση της συμπεριφοράς των χρηστών μέσα στο παιχνίδι. Με αυτόν τον τρόπο χρησιμεύουν ως συνεχή και άμεση ανατροφοδότηση, και ως ανταμοιβή.

Τα μετάλλια είναι οπτικές αναπαραστάσεις των επιτευγμάτων και συλλέγονται μέσα από το περιβάλλον του παιχνιδιού. Επιβεβαιώνουν τα επιτεύγματα των χρηστών, και αντικατοπτρίζουν την επίτευξη των επιπέδων ή των στόχων τους. Η απόκτηση ενός μεταλλίου εξαρτάται από έναν συγκεκριμένο αριθμό πόντων ή από συγκεκριμένες

δραστηριότητες εντός του παιχνιδιού. Οι χρήστες αποκτούν κίνητρο και προσπαθούν όλο και περισσότερο στην απόκτηση των μεταλλίων καθώς αποτελούν το βασικότερο στόχο του παιχνιδιού. Με τον ίδιο τρόπο όπως και οι πόντοι, παρέχουν σχόλια καθώς δείχνουν την απόδοση των παικτών. Επιπλέον έχουν την δυνατότητα να επηρεάσουν την συμπεριφορά των παικτών καθώς επιβάλλεται να επιλέξουν συγκεκριμένες διαδρομές και να αντιμετωπίσουν διάφορες προκλήσεις προκειμένου να κερδίσουν τα αντίστοιχα μετάλλια.

Άλλος ένας μηχανισμός παιχνιδιού που υπάρχει στις gamify εφαρμογές είναι οι βαθμολογικοί πίνακες. Οι βαθμολογικοί πίνακες κατατάσσουν τους παίκτες με βάση ενός συγκεκριμένου κριτηρίου επιτυχίας. Ως εκ τούτου οι βαθμολογικοί πίνακες μπορούν να βοηθήσουν στον προσδιορισμό του ποιος παίκτης έχει την καλύτερη απόδοση πάνω σε μία συγκεκριμένη δραστηριότητα του παιχνιδιού, το γεγονός αυτό τους καθιστά ανταγωνιστικούς δείκτες προόδου που σχετίζονται άμεσα με την απόδοση ενός χρήστη σε σύγκριση με την απόδοση άλλων χρηστών. Η πίεση και ο υγιής ανταγωνισμός που δημιουργείται από τους πίνακες κατάταξης επιδρούν θετικά στη συμμετοχή του φοιτητή στη μάθηση. Ωστόσο αυτές οι θετικές επιπτώσεις του ανταγωνισμού είναι πιο πιθανές εφόσον οι αντίστοιχοι ανταγωνιστές είναι περίπου στο ίδιο επίπεδο απόδοσης.

Επίσης πολύ συχνά σε gamify εφαρμογές χρησιμοποιούνται γραφήματα απόδοσης, τα οποία παρέχουν πληροφορίες σχετικά με την απόδοση των παικτών σε σύγκριση με την προηγούμενη απόδοση τους κατά τη διάρκεια ενός παιχνιδιού. Έτσι σε αντίθεση με τους πίνακες βαθμολογίας, τα γραφήματα απόδοσης δεν συγκρίνουν την απόδοση του παίκτη με την απόδοση άλλων παιχτών, αλλά αξιολογούν την απόδοση του παίκτη με την πάροδο του χρόνου. Εμφανίζοντας γραφικά την απόδοση του χρήστη για μια καθορισμένη περίοδο, τα γραφήματα απόδοσης εστιάζουν σε βελτιώσεις. Παρέχουν κίνητρο στον χρήστη να βελτιώσει την απόδοση του και με αυτόν τον τρόπο ο φοιτητής συμμετέχει πιο ενεργά στη μάθηση.

Οι αφηγηματικές ιστορίες αποτελούν στοιχεία σχεδιασμού παιχνιδιών που δεν σχετίζονται με την απόδοση του παίκτη. Μια αφηγηματική ιστορία η οποία είναι ενσωματωμένη σε μία εφαρμογή προσαρμόζει τις δραστηριότητες και τους χαρακτήρες του παιχνιδιού και δίνει νόημα στους χρήστες να χρησιμοποιούν την εφαρμογή, πέρα από την απλή αναζήτηση πόντων και επιτευγμάτων. Οι ιστορίες των παιχνιδιών δεν αντικατοπτρίζουν πάντα τις πραγματικές συνθήκες της καθημερινότητας οι οποίες θεωρούνται βαρετές και ανούσιες από τους φοιτητές. Αρκετές είναι οι περιπτώσεις όπου gamify εφαρμογές βασίζονται σε ιστορίες με εξωπραγματικά και φανταστικά στοιχεία. Αυτό έχει ως αποτέλεσμα να παρακινήσει τους παίκτες και να τους εμπνεύσει, ιδίως αν η ιστορία του παιχνιδιού τους διεγείρει την φαντασία. Αναμενόμενο λοιπόν είναι οι αφηγηματικές ιστορίες να αποτελούν σημαντικό μέρος στις εφαρμογές παιχνιδιοποιημένης μάθησης, καθώς έχουν την ικανότητα να αλλάξουν το νόημα των πραγματικών δυνατοτήτων προσθέτοντας μια αφήγηση.

Οι εικόνες προφίλ αποτελούν οπτικές αναπαραστάσεις των παικτών εντός του περιβάλλοντος του παιχνιδιού. Συνήθως επιλέγονται και αρκετές φορές δημιουργούνται από τον παίκτη. Οι εικόνες προφίλ μπορούν να σχεδιαστούν πολύ απλά ως μια πολύ απλή εικόνα ή και μπορούν να είναι πολύπλοκες και τρισδιάστατες αναπαραστάσεις. Η βασική λειτουργία των εικόνων προφίλ είναι να ξεχωρίζει ο κάθε παίκτης από τους υπόλοιπους. Επιπλέον επιτρέπουν στους χρήστες να δημιουργήσουν μια ταυτότητα και να γίνουν μέλη μιας ευρύτερης ψηφιακής κοινότητας.

Τέλος στις gamify εφαρμογές υπάρχουν συγκεκριμένες ομάδες παικτών οι οποίοι συνεργάζονται για την επίτευξη ενός κοινού στόχου. Το γεγονός αυτό προκαλεί σύγκρουση ανταγωνισμό και συνεργασία, στοιχεία τα οποία απαρτίζουν την πραγματικότητα.

3.4 Σχετική έρευνα

Μία έρευνα πάνω σε ένα σύστημα πολυμέσων πραγματοποιήθηκε στο πανεπιστήμιο Aguascalientes στο Μεξικό, όπου χρησιμοποιήθηκαν τρεις ομάδες των τριάντα φοιτητών [15]. Η πρώτη ομάδα θα χρησιμοποιούσε παραδοσιακές μεθόδους διδασκαλίας, η δεύτερη ομάδα θα χρησιμοποιούσε ιστοσελίδες και η Τρίτη ομάδα ένα διαδραστικό σύστημα πολυμέσων. Το μάθημα πάνω στο οποίο πραγματοποιήθηκε η έρευνα είναι οι Δομές Δεδομένων, όπου έχει παρατηρηθεί μία σχετική δυσκολία στο συγκεκριμένο μάθημα. Ενενήντα φοιτητές του συγκεκριμένου πανεπιστημίου του τμήματος Επιστήμης Υπολογιστών κατανεμήθηκαν τυχαία στις τρεις ομάδες. Οι φοιτητές είχαν περάσει επιτυχώς τα βασικά μαθήματα πάνω στις γλώσσες προγραμματισμού. Το θέμα της σχετικής έρευνας ήταν τα δυαδικά δέντρα. Επίσης για να είναι η έρευνα αντικειμενική όλοι οι φοιτητές παρακολούθησαν διάλεξη από τον ίδιο καθηγητή, έλαβαν τις ίδιες σημειώσεις, υλικό και ασκήσεις έτσι ώστε να μην υπάρχουν διαφορές διδακτικού υλικού και οι φοιτητές είχαν την δυνατότητα ανατροφοδότησης από τον συγκεκριμένο καθηγητή. Τέλος προκειμένου να μετρηθεί εάν οι συμμετέχοντες μαθαίνουν για το θέμα, το ίδιο πείραμα εφαρμόστηκε και πριν και μετά τη διάλεξη.

Για τη συγκεκριμένη μελέτη το υλικό προσαρμόστηκε σε ισπανόφωνο κοινό. Για την παραδοσιακή ομάδα διδασκαλίας το εκπαιδευτικό υλικό που δόθηκε ήταν σε μορφή PDF με μερικές εικόνες ως παραδείγματα. Οι ιστοσελίδες οι οποίες χρησιμοποιήθηκαν από την δεύτερη ομάδα είχαν κυρίως ένα κείμενο εικόνες και παραδείγματα για την παροχή χρήσιμων πληροφοριών (Εικόνα 3.1). Ένα σημαντικό ζήτημα των ιστοσελίδων που χρησιμοποιήθηκαν ήταν η απουσία στοιχείων πολυμέσων. Το gamify σύστημα που χρησιμοποιήθηκε από την τρίτη ομάδα περιείχε πολλές εικόνες και διαδραστικές ασκήσεις (Εικόνα 3.2). Οι φοιτητές είχαν τη δυνατότητα μέσω της συγκεκριμένης εφαρμογής να απαντήσουν σε ασκήσεις μετακινώντας δεδομένα και ήχους και τους βαθμολόγησε το ίδιο το σύστημα, ενώ οι άλλες ομάδες βαθμολογήθηκαν από τον καθηγητή.

Proyecto de Investigación - Windows Internet Explorer

C:\Documents and Settings\LibethMunoz\Mis documentos\PROYECTO INVEI

Live Search

Archivo Edición Ver Favoritos Herramientas Ayuda

Proyecto de Investigación

Universidad Autónoma de Aguascalientes

Proyecto de Investigación:
Tema: Estructura de Datos: Árboles Binarios de Búsqueda



ESTRUCTURA DE DATOS

Árboles Binarios de Búsqueda: | Concepto | Operaciones Básicas | Aplicaciones |

Concepto de Árbol Binario de Búsqueda



Un árbol es una colección de nodos que puede estar vacía o no. Si no está vacía, el árbol estará formado por un nodo raíz y cero o más subárboles que están unidos a la raíz por aristas o ramas.

Mi equipo 100%

Εικόνα 3.1 Ιστοσελίδα για δυαδικά δέντρα [15]



Εικόνα 3.2 Εφαρμογή για δυαδικά δέντρα [15]

Η γραπτή εργασία η οποία ανατέθηκε στους φοιτητές αποτελούνταν από δυο μέρη. Το πρώτο μέρος περιείχε τέσσερις ερωτήσεις πολλαπλής επιλογής οι οποίες αφορούσαν την θεωρία των δυαδικών δέντρων και τις διάφορες διεργασίες που μπορούν να εκτελεστούν πάνω στα δυαδικά δέντρα. Το δεύτερο μέρος περιείχε τρεις ασκήσεις που αφορούσαν την εισαγωγή, διαγραφή και αναζήτηση κόμβου. Η βαθμολογία για κάθε ερώτηση πολλαπλής επιλογής ήταν ένας πόντος ενώ για κάθε άσκηση ήταν δύο πόντοι. Προκειμένου η έρευνα να είναι αντικειμενική, κάθε ομάδα έλαβε την ίδια διάλεξη για το θέμα των δυαδικών δέντρων, τα ίδια παραδείγματα και τις ίδιες ασκήσεις. Όλες οι ομάδες είχαν ελεύθερο χρόνο να μελετήσουν και να χρησιμοποιήσουν το αντίστοιχο εκπαιδευτικό υλικό που έλαβαν. Επίσης διέθεταν χρόνο για ερωτήσεις και είχαν την δυνατότητα να λάβουν ανατροφοδότηση από τον καθηγητή.

Ο παρακάτω πίνακας (Πίνακας 3.1) προβάλλει τα στατιστικά στοιχεία από τα αποτελέσματα της έρευνας πριν και μετά την διάλεξη. Σε όλες της ομάδες ο μέσος όρος αυξήθηκε και παρατηρείτε πως οποιαδήποτε μέθοδος διδασκαλίας συνέβαλλε στη γνώση. Η βαθμολόγηση πραγματοποιήθηκε σε μία κλίμακα από το μηδέν έως το δέκα. Ο μέσος όρος στην ομάδα που χρησιμοποιήθηκε παραδοσιακός τρόπος διδασκαλίας αυξήθηκε κατά είκοσι τοις εκατό, στην ομάδα που χρησιμοποιήθηκε ιστοσελίδα αυξήθηκε κατά δέκα τοις εκατό ενώ στην τελευταία ομάδα αυξήθηκε κατά είκοσι τρία τοις εκατό.

Πίνακας 3.1 Στατιστικά των δοκιμών [15]

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	BEFORE_TG AFTER_TG	4.679 6.583	30 30	3.1041 1.4025	.5667 .2561
Pair 2	BEFORE_WG AFTER_WG	5.666 6.533	30 30	2.6238 1.5365	.4790 .2805
Pair 3	BEFORE_MIG AFTER_MIG	5.001 7.353	30 30	2.7015 1.0925	.4932 .1995

Επίσης πραγματοποιήθηκε και στις τρεις ομάδες μία τυπική εξέταση ANOVA πριν την διάλεξη προκειμένου να ελεγχθεί εάν οι φοιτητές και των τριών ομάδων είχαν παρόμοιες γνώσεις από προηγούμενα μαθήματα. Τα αποτελέσματα (Πίνακας 3.2) έδειξαν ότι δεν υπάρχουν σημαντικές διαφορές και επομένως οι γνώσεις των φοιτητών δεν επηρέασαν το αποτέλεσμα του πειράματος.

Πίνακας 3.2 ANOVA εξέταση πριν την διάλεξη [15]

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	15.210	2	7.605	.958	.388
Within Groups	690.719	87	7.939		
Total	705.929	89			

Επιπλέον μία ANOVA εξέταση πραγματοποιήθηκε και μετά την διάλεξη προκειμένου να ελεγχθεί κατά πόσο η κάθε προσέγγιση διδασκαλίας επηρέασε την

απόδοση των φοιτητών. Το αποτέλεσμα (Πίνακας 3.3) δείχνει σημαντική διαφορά στις τρεις ομάδες.

Πίνακας 3.3 ANOVA εξέταση μετά την διάλεξη [15]

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	12.678	2	6.339	3.444	.036
Within Groups	160.123	87	1.840		
Total	172.801	89			

Τέλος ο παρακάτω πίνακας (Πίνακας 3.4) αναπαριστά αναλυτικά τα αποτελέσματα της έρευνας και δείχνει ξεκάθαρα ότι ο τρόπος διδασκαλίας με την gamify εφαρμογή είναι πιο αποτελεσματικός.

Πίνακας 3.4 Αναλυτικά αποτελέσματα της έρευνας [15]

	(I) Group	(J) Group	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
						Lower Bound	Upper Bound
LSD	Traditional group (TG).	Web page group (WG).	.0500	.3503	.887	-.646	.746
		Multimedia-Interactive group (MIG).	-.7700*	.3503	.031	-1.466	-.074
	Web page group (WG).	Traditional group (TG).	-.0500	.3503	.887	-.746	.646
		Multimedia-Interactive group (MIG).	-.8200*	.3503	.022	-1.516	-.124
	Multimedia-Interactive group (MIG).	Traditional group (TG).	.7700*	.3503	.031	.074	1.466
		Web page group (WG).	.8200*	.3503	.022	.124	1.516
Dunnnett t (2-sided)	Traditional group (TG).	Multimedia-Interactive group (MIG).	-.7700	.3503	.056	-1.558	.018
	Web page group (WG).	Multimedia-Interactive group (MIG).	-.8200*	.3503	.040	-1.608	-.032

Τα αποτελέσματα της παραπάνω μελέτης αποδεικνύουν ότι ο παραδοσιακός τρόπος διδασκαλίας και η χρήση ιστοσελίδων στην διαδικασία της εκπαίδευσης δεν έδειξαν σημαντικές διαφορές στην απόδοση των φοιτητών. Ωστόσο η χρήση gamify εφαρμογών και διαδραστικών πολυμέσων κάνουν πιο αποτελεσματικό τον τρόπο διδασκαλίας λόγω του συνδυασμού πολυμέσων και της αλληλεπίδρασης που δέχτηκαν οι φοιτητές από τις συνθήκες της συγκεκριμένης μελέτης.

3.5 Σχετικές εφαρμογές

Το Data Structure Visualizations [16] (Εικόνα 3.3) είναι μια διαδικτυακή εφαρμογή η οποία απευθύνεται και σε επαγγελματίες και στην εκπαίδευση. Δημιουργήθηκε από τον καθηγητή David Galles και την ομάδα του του τμήματος επιστήμης υπολογιστών του πανεπιστημίου του Σαν Φρανσίσκο. Η διαδικτυακή εφαρμογή μπορεί να χρησιμοποιηθεί και από υπολογιστή αλλά και από κινητή συσκευή και αποσκοπεί στην εκμάθηση βασικών δομών δεδομένων και αλγορίθμων. Οι χρήστες έχουν την δυνατότητα να καταλάβουν τους αλγόριθμους μέσα από τη χρήση πολυμέσων και τη χρήση εικόνων.

Data Structure Visualizations

About
Algorithms
F.A.Q
Known Bugs /
Feature Requests
Java Version
Flash Version
Create Your Own /
Source Code
Contact

David Galles
Computer Science
University of San
Francisco

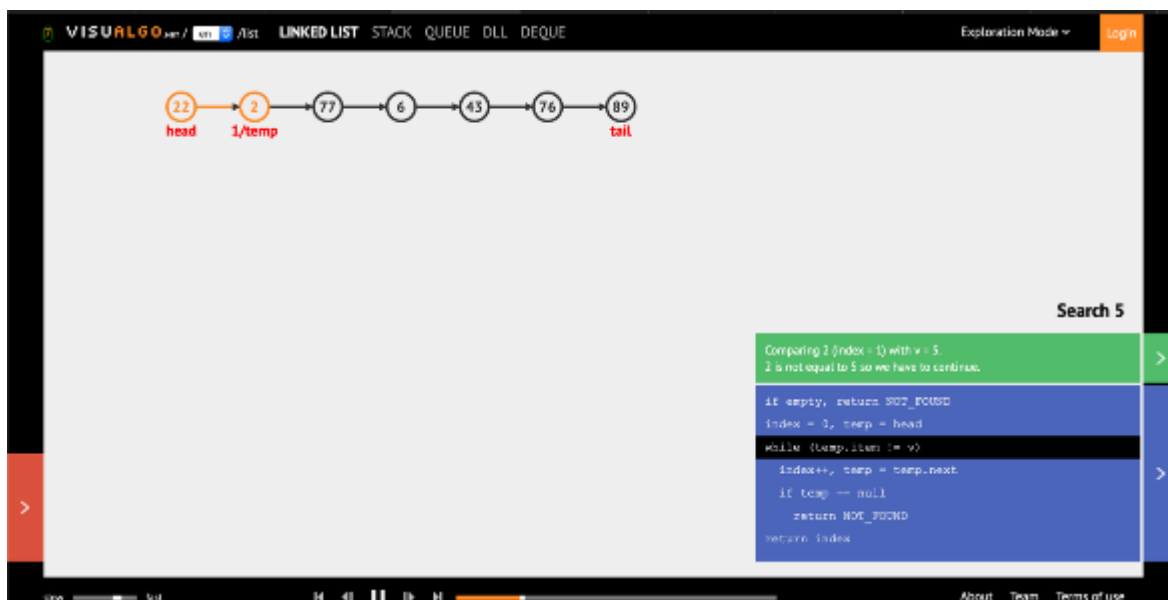
Currently, we have visualizations for the following data structures and algorithms:

- **Basics**
 - Stack: Array Implementation
 - Stack: Linked List Implementation
 - Queues: Array Implementation
 - Queues: Linked List Implementation
 - Lists: Array Implementation (available in java version)
 - Lists: Linked List Implementation (available in java version)
- **Recursion**
 - Factorial
 - Reversing a String
 - N-Queens Problem
- **Indexing**
 - Binary and Linear Search (of sorted list)
 - Binary Search Trees
 - AVL Trees (Balanced binary search trees)
 - Red-Black Trees
 - Splay Trees
 - Open Hash Tables (Closed Addressing)
 - Closed Hash Tables (Open Addressing)
 - Closed Hash Tables, using buckets
 - Trie (Prefix Tree, 26-ary Tree)
 - Radix Tree (Compact Trie)
 - **Ternary Search Tree (Tree with BST of children)**
 - B Trees
 - B+ Trees
- **Sorting**
 - Comparison Sorting
 - Bubble Sort
 - Selection Sort
 - Insertion Sort
 - Shell Sort
 - Merge Sort
 - Quick Sort
 - Bucket Sort
 - Counting Sort
 - Radix Sort
 - Heap Sort
- **Heap-like Data Structures**
 - Heaps
 - Binomial Queues
 - Fibonacci Heaps
 - Leftist Heaps
 - Skew Heaps
- **Graph Algorithms**
 - Breadth-First Search
 - Depth-First Search
 - Connected Components
 - Dijkstra's Shortest Path
 - Prim's Minimum Cost Spanning Tree
 - Topological Sort (Using Indegree array)
 - Topological Sort (Using DFS)
 - Floyd-Warshall (all pairs shortest paths)
 - Kruskal Minimum Cost Spanning Tree Algorithm
- **Dynamic Programming**
 - Calculating nth Fibonacci number
 - Making Change
 - Longest Common Subsequence
- **Geometric Algorithms**
 - 2D Rotation and Scale Matrices
 - 2D Rotation and Translation Matrices
 - 2D Changing Coordinate Systems
 - 3D Rotation and Scale Matrices
 - 3D Changing Coordinate Systems
- **Others ...**
 - Disjoint Sets
 - Huffman Coding (available in java version)

Sp Adobe Spark

Εικόνα 3.3 Το εργαλείο Data Structure Visualizations [16]

Επίσης μία άλλη παρόμοια εφαρμογή είναι η VISUALGO [17], όπου και αυτή απευθύνεται και σε επαγγελματίες και στην εκπαίδευση. Η VISUALGO είναι μία διαδικτυακή εφαρμογή όπου με την χρήση εικόνων πολυμέσων και κινούμενων σχεδίων αναπαριστά βασικούς αλγόριθμους και δομές δεδομένων. Η συγκεκριμένη εκτός από την αναπαράσταση των δομών δεδομένων και αλγορίθμων παρέχει στον χρήστη την δυνατότητα να παρακολουθήσει τον αντίστοιχο κώδικα της διαδικασίας που επιθυμεί να πραγματοποιήσει. Στην παρακάτω εικόνα (Εικόνα 3.6) αναπαρίσταται ένα παράδειγμα αναζήτησης στοιχείου σε μία απλή λίστα.



Εικόνα 3.6 Αναζήτηση σε απλή λίστα της εφαρμογής VISUALGO[17]

Όπως παρατηρούμε υπάρχουν αρκετές πλατφόρμες πολυμέσων για την εκμάθηση δομών δεδομένων και αλγορίθμων. Ωστόσο εμείς δημιουργήσαμε μία gamify εφαρμογή η οποία αποσκοπεί στην καλύτερη κατανόηση των δομών δεδομένων και των αλγορίθμων αυτών. Η εμπειρία του χρήστη κατά την χρήση της εφαρμογής είναι ευχάριστη, έχει την δυνατότητα να δοκιμάσει αλγόριθμους πάνω σε δικά του παραδείγματα. Τέλος ο χρήστης έχει την δυνατότητα να παίξει ένα απλό παιχνίδι ερωτήσεων με σκοπό να δει το ποσοστό κατανόησης των αλγορίθμων αυτών.

ΚΕΦΑΛΑΙΟ 4

ΕΠΙΣΚΟΠΗΣΗ ΕΦΑΡΜΟΓΗΣ

4.1 Τεχνολογική επισκόπηση



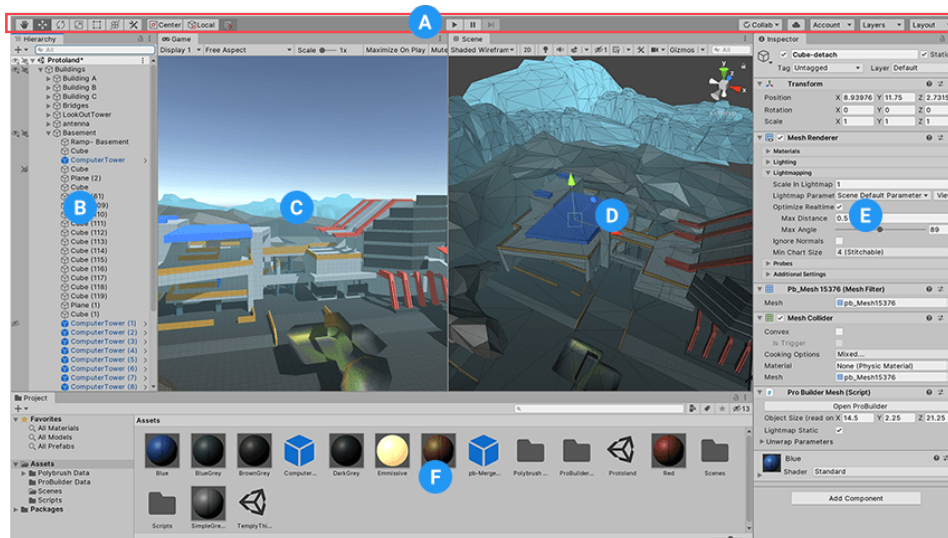
Εικόνα 4.1. Το λογότυπο της παιχνιδομηχανής [18]

Το Unity είναι μια ανεξαρτήτου πλατφόρμας παιχνιδομηχανή (game engine), υλοποιημένο από την εταιρεία Unity Technologies. Κυκλοφόρησε για πρώτη φορά τον Ιούνιο του 2005 και ήταν αποκλειστικά για ανάπτυξη παιχνιδιών σε συσκευές με λειτουργικό σύστημα Mac OS [18]. Από το 2018 και μετά όμως υποστηρίζει άλλες 25 πλατφόρμες μερικές εκ των οποίων είναι τα Windows, Linux, WebGL, PlayStation 4, Xbox One. Το Unity χρησιμοποιείται κυρίως για ανάπτυξη δισδιάστατων και τρισδιάστατων παιχνιδιών, παιχνιδιών εικονικής και επαυξημένης πραγματικότητας καθώς και για ανάπτυξη προσομοιώσεων και άλλων παρόμοιων εφαρμογών. Το Unity χρησιμοποιείται επίσης και από βιομηχανίες όπως αυτή των ταινιών, της μηχανικής και της αρχιτεκτονικής. Στην παρακάτω εικόνα (εικόνα 4.1.1.2.) φαίνεται ο Unity Editor. Ο Unity Editor αποτελείται από τα εξής κομμάτια:

- Κομμάτι A: Το Toolbar, το οποίο στα αριστερά περιέχει κάποια εργαλεία χειρισμού του Scene View (κομμάτι D) και των αντικειμένων (Game Objects) που βρίσκονται μέσα σε αυτό. Τα κουμπιά που βρίσκονται από τα δεξιά δίνουν πρόσβαση στα Unity Collaborate, Cloud Services και Unity Account. Επίσης υπάρχουν και άλλα δύο

κουμπιά τα οποία έχουν να κάνουν με Layer Visibility και Editor Layout. Το Editor Layout έχει να κάνει με το πως είναι στοιβαγμένα τα επιμέρους κομμάτια.

- Κομμάτι B: Το Hierarchy. Είναι μια ιεραρχική επισκόπηση όλων των αντικειμένων που βρίσκονται στο Scene View.
- Κομμάτι C: Το Game View. Το Game View είναι εκεί που γίνεται η προσομοίωση του παιχνιδιού. Η προσομοίωση αυτή ξεκινάει πατώντας το κουμπί Play.
- Κομμάτι D: Το Scene View. Εκεί είναι που γίνεται ο γραφικός σχεδιασμός του παιχνιδιού. Ότι αντικείμενο υπάρχει στο Scene View ονομάζεται Game Object. Όλα αυτά τα αντικείμενα επίσης εμφανίζονται και στο Hierarchy. Το Scene View έχει 2D και 3D perspective ανάλογα με το παιχνίδι που αναπτύσσεται.
- Κομμάτι E: Ο Inspector. Ο Inspector δίνει πληροφορίες για το κάθε αντικείμενο που υπάρχει στο Scene View. Από τον Inspector επίσης μπορούμε να αλλάξουμε/επεξεργαστούμε εάν αντικείμενο, για παράδειγμα να του αλλάξουμε εικόνα, χρώμα, να του δώσουμε μια συγκεκριμένη συμπεριφορά (script) και πολλά άλλα.
- Κομμάτι F: Το Project Window. Το Project Window μας δείχνει όλα τα assets τα οποία έχουμε στην διάθεσή μας για την ανάπτυξη του παιχνιδιού, για παράδειγμα εικόνες, γραμματοσειρές, textures, ήχοι, scripts που έχουμε δημιουργήσει και άλλα.



Εικόνα 4.2 Unity Editor [19]

4.2 Γλώσσα προγραμματισμού C#



Εικόνα 4.3 Το λογότυπο της γλώσσας C# [20]

Η C# είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού η οποία κυκλοφόρησε για πρώτη φορά το 2000 και είναι υλοποιημένη από την Microsoft και σχεδιασμένη από τον [Anders Hejlsberg](#). Η C# έχει πολλές ομοιότητες με την γλώσσα Java, η οποία είναι και αυτή αντικειμενοστραφής. Έχει σχεδιαστεί για να είναι οικονομική όσον αφορά την μνήμη που χρησιμοποιεί (χρησιμοποιώντας Garbage Collector) όμως δεν συγκρίνεται με τις επιδόσεις της C ή τις assembly [20].

Με το που δημιουργείται ένα C# script στο Unity γράφονται αυτομάτως οι παρακάτω γραμμές κώδικα στο αρχείο:

```
using UnityEngine;
using System.Collections;

public class MainPlayer : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

}
```

Εικόνα 4.4 C# script template

Αυτός ο κώδικας για να εκτελεστεί, θα πρέπει να έχει τοποθετηθεί σε ένα αντικείμενο του Scene View. Κάθε κώδικας που τοποθετείται σε ένα αντικείμενο κληρονομεί από την διεπαφή MonoBehaviour. Αν ο κώδικας δεν κληρονομεί από αυτήν την διεπαφή τότε δεν μπορεί να τοποθετηθεί σε κανένα αντικείμενο. Η συνάρτηση Start() εκτελείται μια φορά στην αρχή της προσομοίωσης ενώ η Update() εκτελείται σε κάθε frame. Η συμπεριφορά του αντικειμένου δημιουργείται φτιάχνοντας δικές μας συναρτήσεις χρησιμοποιώντας το scripting API του Unity (Unity Engine). Επίσης σχεδόν πάντα είναι απαραίτητη και η χρήση συγκεκριμένων βιβλιοθηκών της C# (για παράδειγμα System.Collections).

4.3 Λειτουργική επισκόπηση

Όπως έχει αναφερθεί ο σκοπός αυτού του παιχνιδιού/προσομοιωτή είναι να γίνει η εκμάθηση των δομών δεδομένων και κάποιων αλγορίθμων τους με έναν πιο διαδραστικό και αποτελεσματικό τρόπο. Οι δομές δεδομένων είναι πολύ σημαντικές για έναν προγραμματιστή αφού έχουν να κάνουν με το πώς οργανώνονται και αποθηκεύονται τα δεδομένα στην μνήμη [21]. Οι βασικές δομές δεδομένων οι οποίες έχουν επιλεγεί είναι

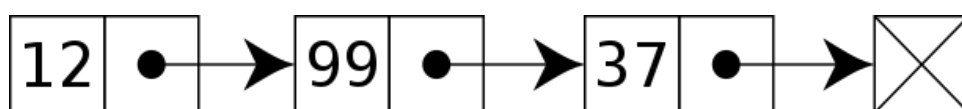
οι εξής: Λίστες (απλές λίστες, διπλές λίστες, απλές κυκλικές λίστες, διπλές κυκλικές λίστες), Στοίβες, Ουρές και Γραφήματα (μη κατευθυνόμενα γραφήματα, κατευθυνόμενα γραφήματα, δυαδικά και ισοζυγισμένα δέντρα). Η κάθε εφαρμογή, ανάλογα με την λειτουργία της, απαιτεί την υλοποίηση μιας από τον παραπάνω ή ακόμα και συνδυασμό αυτών. Η μη σωστή επιλογή της κατάλληλης δομής μπορεί να επηρεάσει αρνητικά τις επιδόσεις μιας εφαρμογής στην ταχύτητα, ειδικά όταν αυτή η εφαρμογή διαχειρίζεται μεγάλο όγκο δεδομένων.

4.3.1 Περιγραφή των βασικών Δομών Δεδομένων

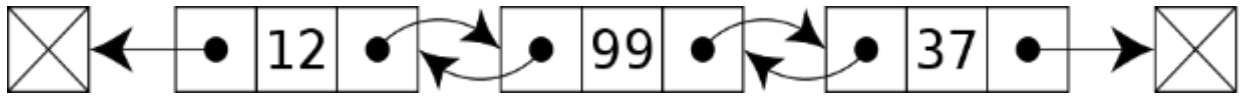
Οι δομές δεδομένων είναι ένας τρόπος οργάνωσης των δεδομένων τον υπολογιστή έτσι ώστε να μπορέσουν να χρησιμοποιηθούν όσο το δυνατόν πιο αποτελεσματικά. Οι βασικές δομές δεδομένων που θα δούμε είναι οι λίστες, στοίβες, ουρές, γράφοι και δέντρα. Οι στοίβες και οι ουρές συνήθως υλοποιούνται με την βοήθεια μιας λίστας ενώ τα δέντρα είναι ένας συγκεκριμένος τύπος γράφου.

4.3.2 Λίστες

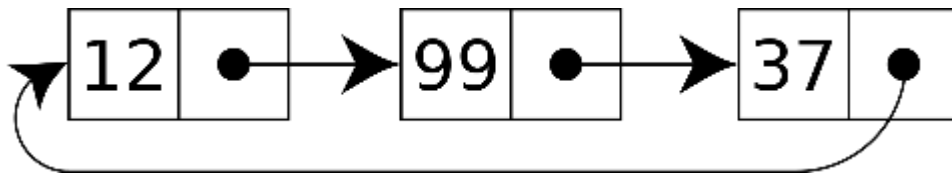
Στην επιστήμη των υπολογιστών, οι συνδεδεμένες λίστες είναι μια γραμμική συλλογή δεδομένων. Ο κάθε κόμβος της λίστας συνδέεται με τον επόμενο στην περίπτωση της απλά συνδεδεμένης λίστας ενώ στην περίπτωση της διπλά συνδεδεμένης λίστας ο κάθε κόμβος συνδέεται και με τον προηγούμενο του. Ο κάθε κόμβος στην πιο απλή του μορφή περιέχει δύο πεδία: μια πληροφορία και τη διεύθυνση του επόμενου κόμβου στην μνήμη. Στην περίπτωση της μη κυκλικής λίστας (Εικόνα 4.5) (Εικόνα 4.6) ο τελευταίος κόμβος δείχνει στην διεύθυνση null, που στην ουσία σημαίνει ότι δεν δείχνει πουθενά ενώ στην περίπτωση της κυκλικής (Εικόνα 4.7) δείχνει στο πρώτο στοιχείο (κεφαλή λίστας). Οι λίστες επίσης χρησιμοποιούνται πολύ συχνά και για την υλοποίηση άλλων δομών δεδομένων όπως οι στοίβες και οι ουρές [19].



Εικόνα 4.5 C# script Απλά συνδεδεμένη λίστα [22]



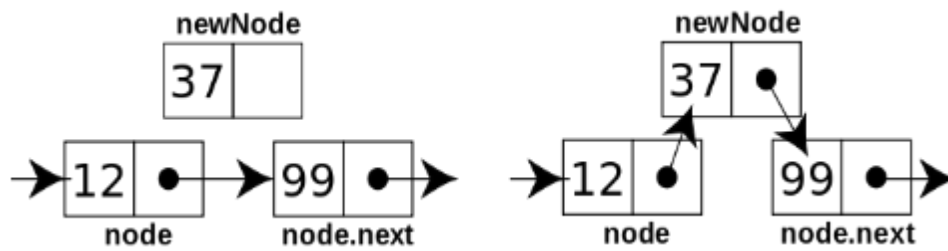
Εικόνα 4.6 C# Διπλά συνδεδεμένη λίστα [22]



Εικόνα 4.7 Απλά συνδεδεμένη κυκλική λίστα [22]

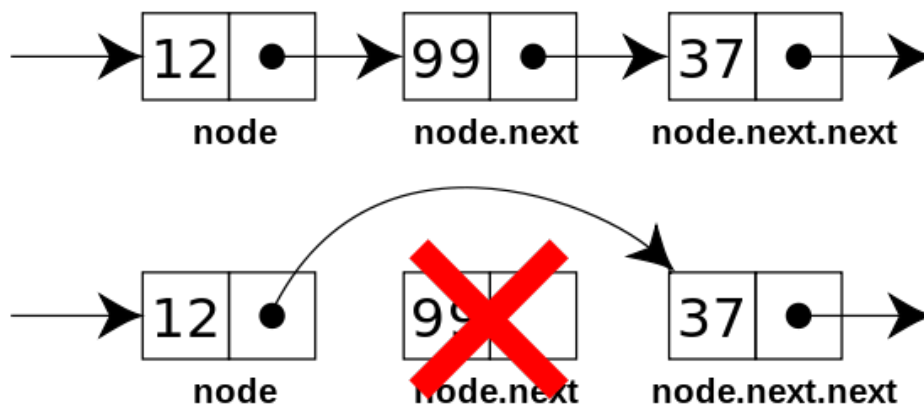
Για την υλοποίηση των λιστών διατηρούνται κυρίως δύο μεταβλητές, η κεφαλή της λίστας (το πρώτο στοιχείο) και η ουρά της λίστας (το τελευταίο στοιχείο).

Η πρόσθεση ενός κόμβου σε μια λίστα χωρίζεται στην ουσία σε τρεις υποκατηγορίες: πρόσθεση κόμβου στην αρχή της λίστας, πρόσθεση κόμβου στο τέλος της λίστας και πρόσθεση κόμβου σε ένα άλλο σημείο ενδιάμεσα της λίστας. Στην πρώτη περίπτωση, δημιουργούμε τον κόμβο, τον συνδέουμε με την κεφαλή της λίστας και στην συνέχεια τον ορίζουμε ως την νέα κεφαλή της λίστας. Στην περίπτωση της πρόσθεσης κόμβου στο τέλος η διαδικασία είναι παρόμοια, συνδέουμε τον τελευταίο κόμβο της λίστας με τον νέο κόμβο, και ορίζουμε τον νέο κόμβο ως την νέα ουρά της λίστας. Την τελευταία περίπτωση θα χρειαστεί να την περιγράψουμε με ένα απλό παράδειγμα. Έστω ότι θέλουμε να προσθέσουμε έναν κόμβο στην νιοστή θέση της λίστας. Ξεκινάμε την διαπέραση της λίστας από την αρχή και μετράμε τον αριθμό των κόμβων που έχουμε διαπεράσει. Μόλις αυτός ο αριθμός γίνει ίσος με $n-1$ τότε γίνεται η πρόσθεση του κόμβου. Τοποθετούμε τον κόμβο ανάμεσα από τον κόμβο που βρισκόμαστε αυτή την στιγμή και τον επόμενο του. Αν η θέση πρόσθεσης είναι μηδέν, τότε κάνουμε πρόσθεση κόμβου στην αρχή της λίστας, αν είναι η τελευταία θέση τότε κάνουμε πρόσθεση στο τέλος (Εικόνα 4.8).



Εικόνα 4.8 Πρόσθεση κόμβου σε ένα σημείο ενδιάμεσα της λίστας [22]

Για την διαγραφή κόμβου ξεκινάμε την διαπέραση της λίστας από την αρχή, διατηρώντας κάθε φορά τον προηγούμενο κόμβο. Μόλις βρούμε τον κόμβο που θέλουμε να διαγράψουμε, τότε ενώνουμε τον προηγούμενο του με τον επόμενο, και στη συνέχεια τον διαγράφουμε (Εικόνα 4.9).



Εικόνα 4.9 Διαγραφή κόμβου [22]

```

void add_front(struct Node** head_ref, int new_data)
{
    struct Node* new_node = (struct Node*) malloc(sizeof(struct Node));
    new_node->data = new_data;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}
void insert_after(struct Node* prev_node, int new_data)
{
    if (prev_node == NULL)
    {
        printf("the given previous node cannot be NULL");
        return;
    }
    struct Node* new_node =(struct Node*) malloc(sizeof(struct Node));
    new_node->data = new_data;
    new_node->next = prev_node->next;
    prev_node->next = new_node;
}
void add_end(struct Node** head_ref, int new_data)
{
    struct Node* new_node = (struct Node*) malloc(sizeof(struct Node));
    struct Node *last = *head_ref;
    new_node->data = new_data;
    new_node->next = NULL;
    if (*head_ref == NULL)
    {
        *head_ref = new_node;
        return;
    }
    while (last->next != NULL)
        last = last->next;
    last->next = new_node;
    return;
}

```

Εικόνα 4.10 Υλοποίηση ένθεσης κόμβου σε απλή λίστα στη γλώσσα C

```

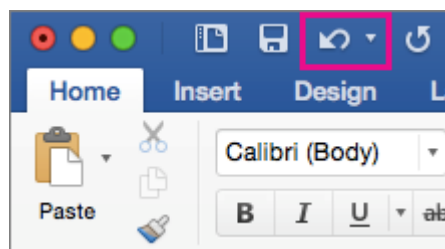
void delete(struct Node **head_ref, int key)
{
    struct Node* temp = *head_ref, *prev;
    if (temp != NULL && temp->data == key)
    {
        *head_ref = temp->next;
        free(temp);
        return;
    }
    while (temp != NULL && temp->data != key)
    {
        prev = temp;
        temp = temp->next;
    }
    if (temp == NULL) return;
    prev->next = temp->next;
    free(temp);
}

```

Εικόνα 4.11 Υλοποίηση διαγραφής κόμβου σε απλή λίστα στη γλώσσα C

Για την αναζήτηση κόμβου ξεκινάμε την διαπέραση της λίστας από την αρχή μέχρι να βρούμε τον κόμβο που ψάχνουμε. Οι παραπάνω λειτουργίες υλοποιούνται παρόμοια και στους άλλους τύπους λίστας αλλάζοντας κάποιες λεπτομέρειες.

Οι λίστες χρησιμοποιούνται για την υλοποίηση άλλων δομών δεδομένων όπως οι στοίβες και οι ουρές. Επίσης λίστες θα μπορούσαν να χρησιμοποιηθούν για την υλοποίηση λιστών αναπαραγωγής μουσικής (playlist). Χρησιμοποιούνται επίσης για την αναπαράσταση πολυωνύμων αλλά και για την υλοποίηση της λειτουργίας Undo, την επαναφορά δηλαδή ενός αρχείου σε μια πρόσφατα προηγούμενη κατάσταση.



Εικόνα 4.12 Λειτουργία Undo

Μειονεκτήματα των λιστών:

- Χρησιμοποιούν περισσότερη μνήμη από τους απλούς πίνακες διότι εκτός από την πληροφορία πρέπει να αποθηκεύουν και την διεύθυνση του επόμενου κόμβου, αν είναι απλά συνδεδεμένη ενώ αν είναι διπλά συνδεδεμένη αποθηκεύει και του προηγούμενου.
- Οι κόμβοι σε μια συνδεδεμένη λίστα προσπελούνται με την σειρά από την αρχή της.
- Λόγο του ότι η προσπέλαση γίνεται γραμμικά μπορεί να υπάρξουν χαμηλές επιδόσεις όσον αφορά την ταχύτητα, ειδικά όταν η λίστα αποτελείται από πολλούς κόμβους.
- Στην περίπτωση της απλά συνδεδεμένης λίστας είναι πολύ δύσκολο να γίνει προσπέλαση της λίστας από το τέλος λόγω του ότι ο κάθε κόμβος αποθηκεύει την διεύθυνση μόνο του επόμενου κόμβου. Στην περίπτωση της διπλά συνδεδεμένης λίστας αυτή η προσπέλαση είναι πολύ εύκολη όμως χρησιμοποιείται πολύ περισσότερη μνήμη αφού ο κάθε κόμβος αποθηκεύει την διεύθυνση όχι μόνο του επόμενου αλλά του προηγούμενου κόμβου.

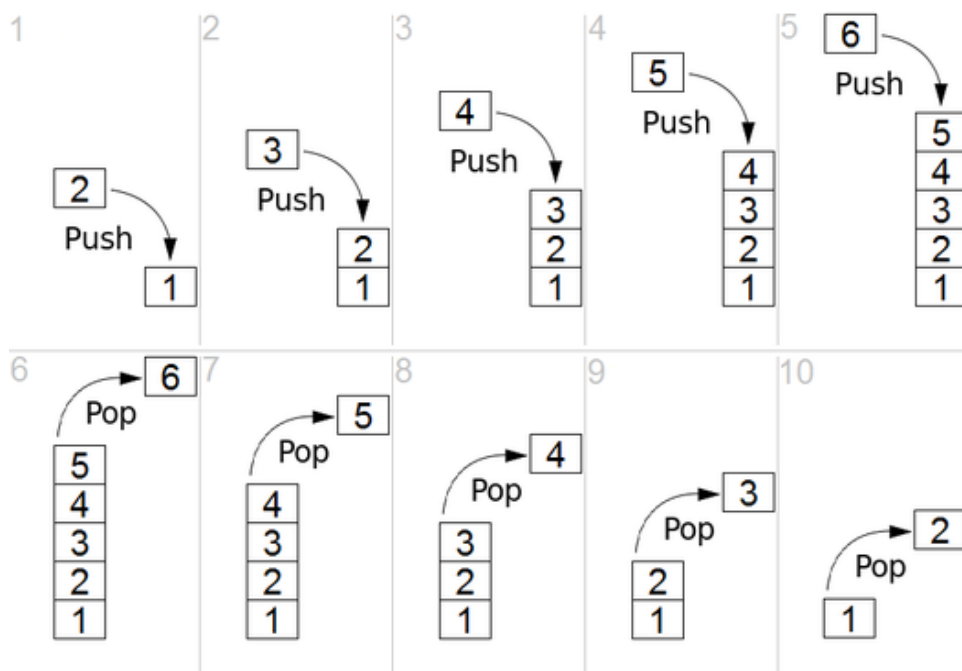
4.3.3 Στοίβες

Οι στοίβες είναι μια συλλογή από δεδομένα η οποία υποστηρίζει δύο κύριες λειτουργίες: την ώθηση (push) και την απόσπαση (pop). Πρωτοεμφανίστηκαν στην επιστήμη των υπολογιστών το 1946 από τον Alan M. Turing για να περιγράψει την διαδικασία των υπορουτινών (subroutines), οι οποίες είχαν ήδη υλοποιηθεί το 1945 από τον Konrad Zuse [23]. Η ώθηση και η απόσπαση γίνονται πάντα από το τέλος της συλλογής, οι στοίβες χρησιμοποιούν δηλαδή την τακτική LIFO (first in, first out). Οι στοίβες συνήθως υλοποιούνται με την χρήση διπλά συνδεδεμένης λίστας, αλλά μπορούν επίσης να υλοποιηθούν χρησιμοποιώντας έναν πίνακα σταθερού μήκους. Όταν η στοίβα είναι γεμάτη και δεν είναι δυνατόν να ωθήσουμε ένα καινούργιο στοιχείο λέμε ότι βρίσκεται σε

καθεστώς υπερχειλίσσης (overflow) ενώ όταν είναι άδεια και δεν μπορούμε να εξάγουμε κάποιο στοιχείο τότε λέμε ότι βρίσκεται σε καθεστώς υποχειλίσσης (underflow). Οι στοίβες πολύ συχνά παρομοιάζονται με μια στοίβα από πιάτα. Όταν προσθέτουμε ένα πιάτο στην στοίβα το τοποθετούμε επάνω στα ήδη υπάρχοντα, κάνουμε δηλαδή ώθηση (push) στο τέλος, ενώ όταν θέλουμε να αφαιρέσουμε ένα πιάτο το αφαιρούμε και αυτό από το τέλος (pop).

Μια στοίβα μπορεί να υλοποιηθεί χρησιμοποιώντας πίνακα συγκεκριμένου μεγέθους με τον εξής τρόπο. Το πρώτο στοιχείο τοποθετείται στην θέση 0 και συχνά αναφέρεται ως bottom. Το πρόγραμμα θα πρέπει να διατηρεί μια μεταβλητή η οποία κάθε φορά που γίνεται ώθηση και απόσπαση θα ανανεώνεται αναλόγως. Στην περίπτωση της ώθησης θα αυξάνεται κατά μια μονάδα και στην περίπτωση της απόσπασης θα μειώνεται κατά μια μονάδα. Αυτή η μεταβλητή δηλαδή χρησιμοποιείται για να μετράει πόσα στοιχεία βρίσκονται ανά πάσα στιγμή στην στοίβα.

Η υλοποίηση με λίστα γίνεται ως εξής: η ώθηση ενός στοιχείου γίνεται πρόσθεση ενός νέου κόμβου στο τέλος της λίστας, ενώ η απόσπαση γίνεται με απλή διαγραφή του τελευταίου κόμβου. Σε αυτήν την υλοποίηση δεν χρειάζεται να μετράμε τον αριθμό των στοιχείων της στοίβας αφού δεν υπάρχει κάποιο συγκεκριμένο μέγεθος. Υπερχειλίσση μπορεί μόνο να συμβεί στην περίπτωση που έχει εξαντληθεί όλη η μνήμη (Εικόνα 4.13).



Εικόνα 4.13 Στοίβα υλοποιημένη με λίστα και οι βασικές λειτουργίες της [23]

```
class Stack:
    def __init__(self):
        self.items = []
    def isEmpty(self):
        return self.items == []
    def push(self, item):
        self.items.append(item)
    def pop(self):
        return self.items.pop()
    def peek(self):
        return self.items[len(self.items)-1]
    def size(self):
        return len(self.items)
```

Εικόνα 4.14 Υλοποίηση στοίβας στην γλώσσα Python

Η μετατροπή μιας έκφρασης σε μορφή prefix, postfix από infix γίνεται με την βοήθεια μιας στοίβας (Εικόνα 4.15). Επίσης πολλοί compilers χρησιμοποιούν στοίβες για parenthesis matching ή αλλιώς bracket matching (Εικόνα 4.16).

Valid	Invalid
()	(
[]	[[]
{ }	(]
(([]))))
{ { { } } }	())
([] { () [] }) [{ }]	([] { () [] } [{ }]

Εικόνα 4.15 Parenthesis matching [24]



Εικόνα 4.16 Από infix σε postfix και prefix [25]

Επίσης οι στοίβες χρησιμοποιούνται σε backtracking αλγόριθμους, ένας από αυτούς είναι και ο αλγόριθμος της αναζήτησης κατά βάθος, τον οποίο θα αναφέρουμε στο κομμάτι των γραφημάτων.

4.3.4 Ουρές

Η βασική λειτουργικότητα των ουρών είναι η εισαγωγή στοιχείων στην πίσω θέση και η εξαγωγή από την μπροστινή θέση. Οι ουρές δηλαδή ακολουθούν την τακτική FIFO (first in, first out). Το πρώτο στοιχείο που θα προστεθεί θα είναι και το πρώτο που θα εξαχθεί/εξυπηρετηθεί. Οι ουρές είναι ο αφηρημένος τύπος δεδομένων μιας ουράς αναμονής στην καθημερινή ζωή (για παράδειγμα μια ουρά εξυπηρέτησης πελατών σε εάν ταμείο) [26]. Συνήθεις χρήσεις των ουρών είναι σε κυκλικές προσωρινές μνήμες και στις διασυνδεδεμένες λίστες. Ένα άλλο παράδειγμα χρήσης των ουρών είναι στην λειτουργία της προσωρινής μνήμης (buffer) [27].

Η υλοποίηση μιας ουράς γίνεται ακριβώς με το ίδιο σκεπτικό που γίνεται και η υλοποίηση μιας στοίβας. Έχουμε δηλαδή υλοποίηση με λίστα και υλοποίηση με πίνακα. Η μόνη διαφορά με την στοίβα είναι ότι στην ουρά η πρόσθεση ενός στοιχείου γίνεται πάντα στο τέλος, σε αντίθεση με την στοίβα που γίνεται στην αρχή (Εικόνα 4.17).

```

class Queue:
    def __init__(self, items = None):
        if items is None:
            items = []
        self.__queue = items
    def isEmpty(self):
        return len(self.__queue) == 0
    def enqueue(self, obj):
        self.__queue.append(obj)
    def dequeue(self):
        return self.__queue.pop(0)
    def peek(self):
        return self.__queue[0]

```

Εικόνα 4.17 Παράδειγμα υλοποίησης ουράς στην γλώσσα Python

Οι ουρές συνήθως χρησιμοποιούνται σε εφαρμογές στις οποίες χρειάζεται να τηρηθεί κάποια σειρά. Για παράδειγμα σε έναν κοινόχρηστο εκτυπωτή και σε τηλεφωνικά κέντρα τα οποία βάζουν τους πελάτες σε μια ουρά αναμονής μέχρι να βρεθεί κάποιος διαθέσιμος υπάλληλος [27].

4.3.5 Γράφοι

Στα διακριτά μαθηματικά, ένας Γράφος ή γράφημα είναι μια αφηρημένη αναπαράσταση ενός συνόλου στοιχείων, όπου μερικά ζευγάρια στοιχείων συνδέονται μεταξύ τους με ακμές. Τα στοιχεία αυτά ονομάζονται κορυφές [28]. Οι κορυφές καλούνται επίσης, και οι ακμές ονομάζονται επίσης γραμμές. Η λέξη «γράφημα» χρησιμοποιήθηκε για πρώτη φορά με αυτή την έννοια από τον James Joseph Sylvester το 1878.

Ένα γράφημα είναι ένα διατεταγμένο ζεύγος $G = (V, E)$ αποτελούμενο από ένα σύνολο V κόμβων μαζί με E σύνολο από ακμές, οι οποίες είναι υποσύνολα δύο στοιχείων V . Οι κορυφές που ανήκουν σε μια ακμή ονομάζονται τελικά σημεία ή τελικές κορυφές της ακμής. Μια κορυφή μπορεί να υπάρχει σε ένα γράφημα και να μην ανήκει σε ακμή. Η σειρά ενός γραφήματος είναι ο αριθμός των κορυφών. Το μέγεθος ενός γραφήματος είναι ο αριθμός των ακμών. Ο βαθμός ενός κόμβου είναι ο αριθμός των ακμών που

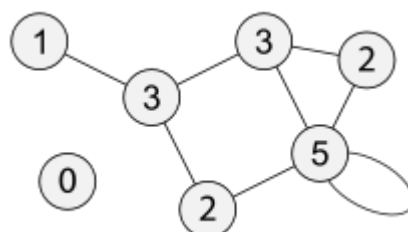
συνδέονται με αυτόν, όπου μια ακμή η οποία συνδέεται με την κορυφή και στα δύο άκρα (ένας βρόχος) συνυπολογίζεται δύο φορές.

Μερικοί τύποι γραφημάτων:

- Μη κατευθυνόμενο γράφημα: ένα μη κατευθυνόμενο γράφημα είναι εκείνο στο οποίο οι ακμές δεν έχουν προσανατολισμό.
- Κατευθυνόμενο γράφημα: ένα κατευθυνόμενο γράφημα ή διγράφημα είναι ένα διατεταγμένο ζεύγος $D = (V, A)$ όπου V , είναι ένα σύνολο των κόμβων και A , είναι μια σειρά από διατεταγμένα ζεύγη κόμβων, τα οποία ονομάζονται βέλη. Ένα βέλος $a = (x, y)$ κατευθύνεται από το x στο y . Το τόξο (y, x) ονομάζεται το τόξο (x, y) αναστραμμένο.
- Μικτό γράφημα: ένα μικτό γράφημα G είναι ένα γράφημα στο οποίο μερικές ακμές μπορεί να είναι κατευθυνόμενες και κάποιες μπορεί να είναι μη κατευθυνόμενες.

Σημαντικές κατηγορίες γραφημάτων:

- Τυπικό γράφημα: ένα τυπικό γράφημα είναι ένα γράφημα όπου κάθε κορυφή έχει τον ίδιο αριθμό γειτόνων, δηλαδή κάθε κορυφή έχει τον ίδιο βαθμό.
- Πλήρες γράφημα: σε ένα πλήρες γράφημα το κάθε ζευγάρι κορυφών έχει μια ακμή που να τους συνδέει.
- Πεπερασμένα και άπειρα γραφήματα: ένα πεπερασμένο γράφημα είναι ένα γράφημα $G = (V, E)$ έτσι ώστε V και E να είναι πεπερασμένα σύνολα. Ένα άπειρο γράφημα είναι ένα γράφημα με ένα άπειρο σύνολο κορυφών ή ακμών ή και τα δύο (Εικόνα 4.18).

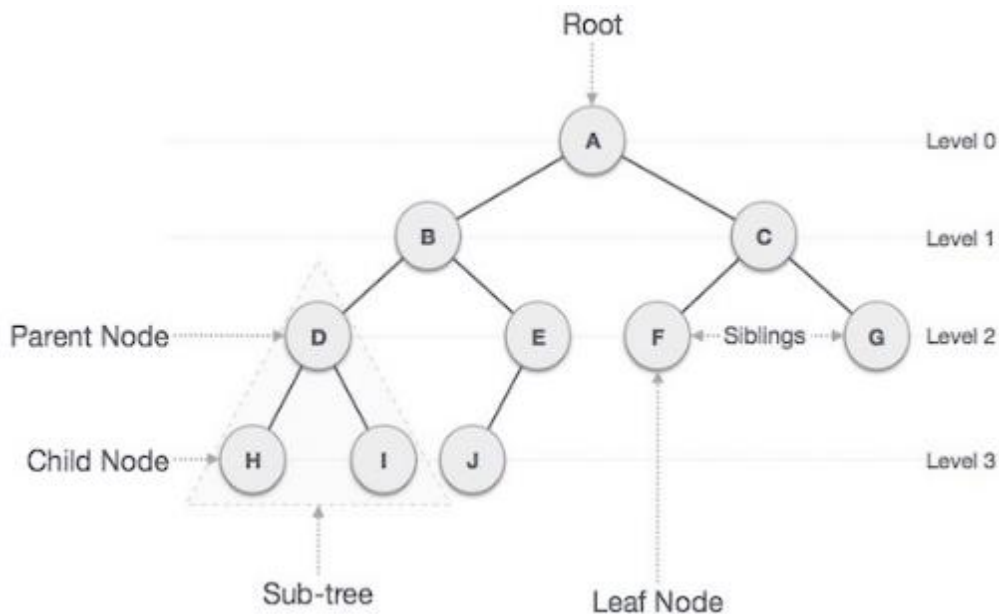


Εικόνα 4.18 Μη-Κατευθυνόμενος Γράφος [28]

4.3.6 Δέντρα

Ένα δέντρο στη Θεωρία Γράφων, είναι ένας μη κατευθυνόμενος Γράφος, στον οποίο οποιεσδήποτε δύο κορυφές συνδέονται με ένα και μόνο ακμή. Οι κόμβοι του δέντρου συνδέονται μεταξύ τους με τη βοήθεια ακμών με βάση τους εξής κανόνες: (Εικόνα 4.19)

- υπάρχει ένας και μόνον ένας κόμβος στον οποίο δεν καταλήγει καμία ακμή (η ρίζα του δέντρου).
- Σε όλους τους υπόλοιπους κόμβους καταλήγει υποχρεωτικά μόνο μία ακμή.



Εικόνα 4.19 Παράδειγμα 2-αδικού δέντρου [29]

Οι κόμβοι ενός δέντρου μπορεί να αντιπροσωπεύουν μια τιμή, μια συνθήκη ή ακόμα και μια άλλη δομή δεδομένων. Κάθε κόμβος μπορεί να έχει κόμβους-παιδιά (child nodes). Όταν ένας κόμβος έχει ένα παιδί τότε το παιδί λέμε ότι έχει πατέρα (parent node) τον κόμβο αυτό. Οι κόμβοι παιδιά που έχουν τον ίδιο πατέρα ονομάζονται αδελφοί κόμβοι (sibling nodes).

Εσωτερικός κόμβος ονομάζεται κάθε κόμβος του δέντρου ο οποίος έχει ένα ή περισσότερα παιδιά. Εξωτερικός κόμβος ή φύλλο είναι κάθε κόμβος του δέντρου ο οποίος δεν έχει

κανένα παιδί. Ύψος ενός κόμβου είναι το μήκος του μονοπατιού από τον κόμβο αυτόν μέχρι έναν εξωτερικό κόμβο. Το ύψος της ρίζας είναι και το ύψος του δέντρου.

Υποδέντρο ενός δέντρου A είναι το δέντρο το οποίο αποτελείται από έναν κόμβο του A και όλους τους απογόνους του.

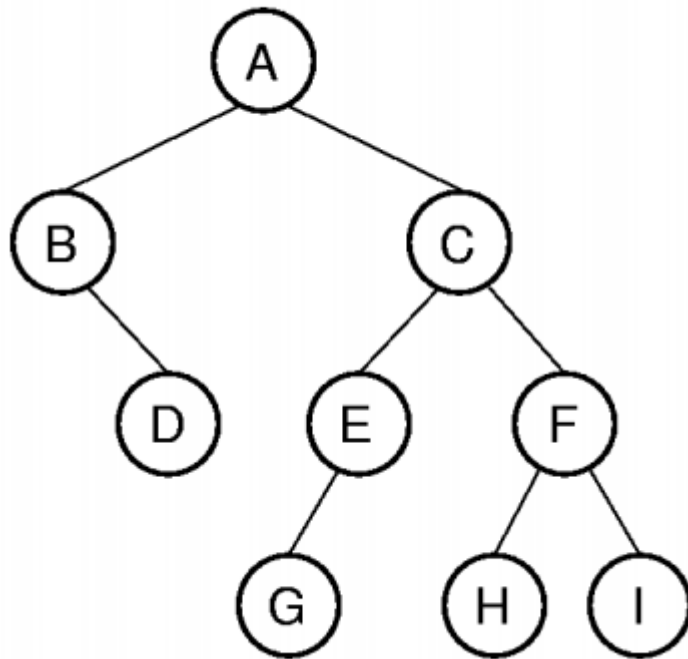
Υπάρχουν πολλοί τύποι δέντρων, εμείς θα ασχοληθούμε με τα απλά δυαδικά δέντρα και με τα ισοζυγισμένα δυαδικά δέντρα αναζήτησης (AVL trees).

Στα δυαδικά δέντρα, σε αντίθεση με τα δέντρα αναζήτησης, οι κόμβοι δεν μπαίνουν με κάποια συγκεκριμένη σειρά. Για αυτόν τον λόγο, τα δυαδικά δέντρα σε αυτήν την εφαρμογή χρησιμοποιούνται μόνο για την επίδειξη των αλγορίθμων διαπέρασης των δέντρων (traversal algorithms). Υπάρχουν τέσσερις τρόποι να διαπεράσουμε ένα δέντρο και είναι οι εξής:

- Προδιατεταγμένη διάσχιση (preorder): Για κάθε κόμβο, επισκεπτόμαστε πρώτα τον ίδιο τον κόμβο, έπειτα τους κόμβους του αριστερού του υποδέντρου και στη συνέχεια τους κόμβους του δεξιού του υποδέντρου.
- Μεταδιατεταγμένη διάσχιση (postorder): Για κάθε κόμβο, επισκεπτόμαστε πρώτα τους κόμβους του αριστερού του υποδέντρου, έπειτα τους κόμβους του δεξιού του υποδέντρου και στη συνέχεια τον ίδιο τον κόμβο.
- Ενδοδιατεταγμένη διάσχιση (inorder): Για κάθε κόμβο, επισκεπτόμαστε πρώτα τους κόμβους του αριστερού του υποδέντρου, έπειτα τον ίδιο τον κόμβο και στη συνέχεια τους κόμβους του δεξιού του υποδέντρου.
- Κατά σειρά επιπέδων (level order): Επισκεπτόμαστε τους κόμβους κατά επίπεδα, από πάνω (τη ρίζα) προς τα κάτω, και μέσα σε ένα επίπεδο από αριστερά προς τα δεξιά.

Για παράδειγμα για το δέντρο της εικόνας 4.20:

- Προδιατεταγμένη διαπέραση: A, B, D, C, E, G, F, H, I
- Μεταδιατεταγμένη διαπέραση: D, B, G, E, H, I, F, C, A
- Ενδοδιατεταγμένη διαπέραση: B, D, A, G, E, C, H, F, I
- Κατά σειρά επιπέδων διαπέραση: A, B, C, D, E, F, G, H, I



Εικόνα 4.20 Παράδειγμα δυαδικού δέντρου [30]

Until all nodes are traversed -
Step 1 - Recursively traverse left subtree.
Step 2 - Visit root node.
Step 3 - Recursively traverse right subtree.

Εικόνα 4.21 Αλγόριθμος Ενδοδιατεταγμένης διαπέρασης [31]

Until all nodes are traversed -
Step 1 - Visit root node.
Step 2 - Recursively traverse left subtree.
Step 3 - Recursively traverse right subtree.

Εικόνα 4.22 Αλγόριθμος Προδιατεταγμένης διαπέρασης [31]

Until all nodes are traversed -
Step 1 - Recursively traverse left subtree.
Step 2 - Recursively traverse right subtree.
Step 3 - Visit root node.

Εικόνα 4.23 Αλγόριθμος Μεταδιατεταγμένης διαπέρασης [31]

```

levelorder(root)
  q ← empty queue
  q.enqueue(root)
  while not q.isEmpty() do
    node ← q.dequeue()
    visit(node)
    if node.left ≠ null then
      q.enqueue(node.left)
    if node.right ≠ null then
      q.enqueue(node.right)

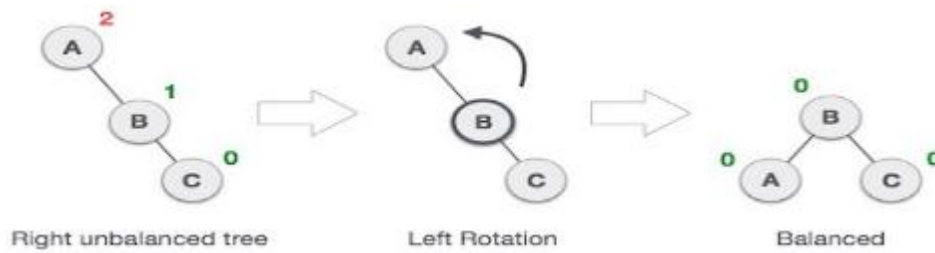
```

Εικόνα 4.24 Αλγόριθμος Κατά σειρά επιπέδων διαπέρασης [31]

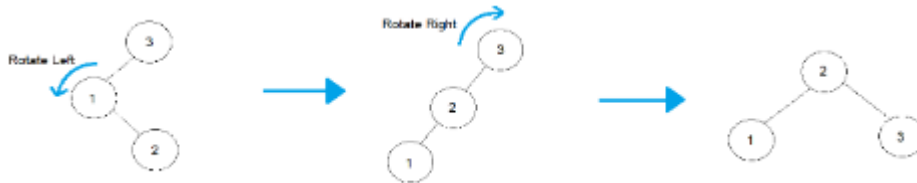
Τα AVL trees εφόσον είναι δέντρα αναζητήσεως ακολουθούν τον ακόλουθο βασικό κανόνα: το αριστερό υποδέντρο ενός κόμβου περιέχει μόνο κόμβους οι οποίοι έχουν κλειδιά μικρότερα από αυτόν και το δεξί υποδέντρο ενός κόμβου περιέχει μόνο κόμβους οι οποίοι έχουν κλειδιά μεγαλύτερα από αυτόν. Η βασική διαφορά των AVL δέντρων με τα απλά δέντρα αναζητήσεως είναι ότι τα AVL αναπροσαρμόζονται έτσι ώστε να είναι ισορροπημένα. Πιο συγκεκριμένα πρέπει για κάθε κόμβο u τα ύψη των παιδιών του να διαφέρουν το πολύ κατά ένα (υποθέτουμε πως το κενό δένδρο έχει ύψος μείον ένα). Οι αναπροσαρμογές ονομάζονται περιστροφές (rotations) και είναι τέσσερις:

- double left rotation
- double right rotation
- left-right rotation
- right-left rotation

Στις εικόνες 4.25, 4.26 φαίνονται σχηματικά οι αναπροσαρμογές double left rotation και left-right rotation. Οι αναπροσαρμογές double right rotation και right-left rotation γίνονται αντίστοιχα.



Εικόνα 4.25 Double left rotation [32]



Εικόνα 4.26 Left-right rotation [32]

4.4 Λειτουργίες εφαρμογής

Η εφαρμογή μας αποτελείται όπως παρουσιάζεται και στην εικόνα 4.27 από ένα άσπρο σκηνικό όπου πάνω εμφανίζονται όλες οι δομές που μπορεί να επιλέξει ο χρήστης. Κάτω αριστερά ο χρήστης έχει την δυνατότητα να επιλέξει το κουμπί (Quiz Game) και να ελέγξει τις γνώσεις του με τη βοήθεια ενός παιχνιδιού ερωτήσεων. Επίσης δίπλα από αυτό το κουμπί υπάρχει ένα κουμπί ελέγχου της ταχύτητας εκτέλεσης των αλγορίθμων. Η ταχύτητα παίρνει τις τιμές από μηδέν έως ένα. Ακόμα άλλη μία δυνατότητα δίνεται στον χρήστη μέσω ενός κουμπιού παύσης (Stop Algorithm) όπου ο χρήστης μπορεί να σταματήσει την εκτέλεση των αλγορίθμων και τέλος υπάρχει ακόμα ένα κουμπί (Run step by step) όπου ο χρήστης μπορεί να διατρέξει τους αντίστοιχους αλγόριθμους βήμα-βήμα.



4.4.1 Λίστες

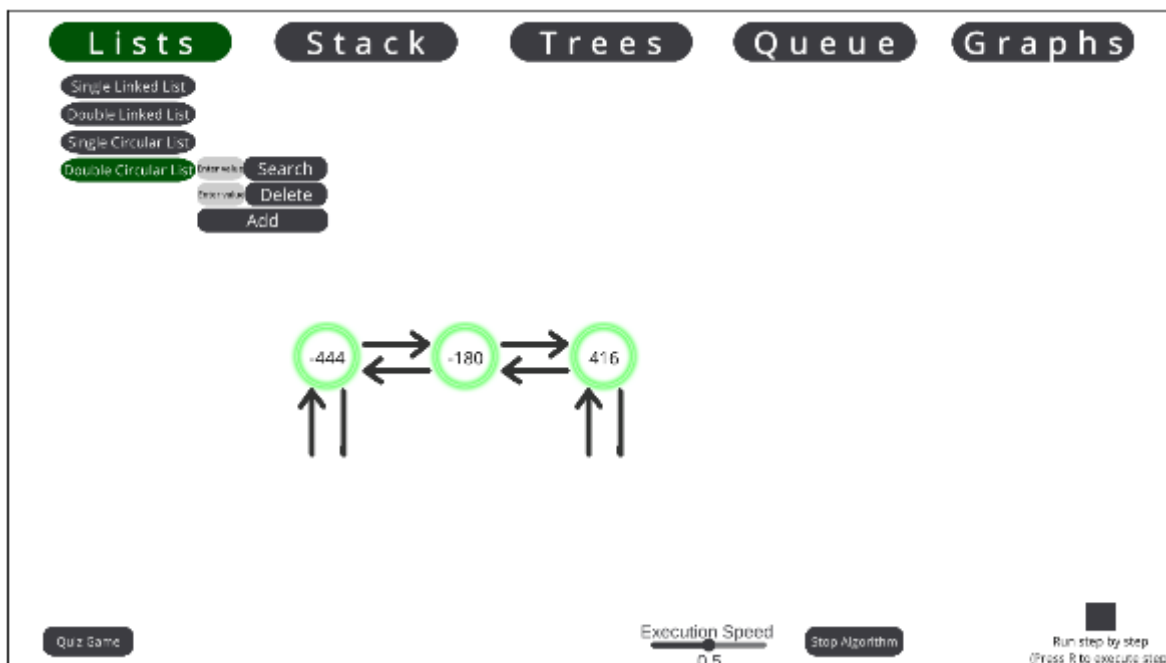
Εικόνα 4.27 Ο προσομοιωτής

Πατώντας την επιλογή λίστες (Lists) της εφαρμογής αναγράφονται οι τέσσερις διαφορετικοί τύποι λιστών (Εικόνα 4.28). Απλά συνδεδεμένη (Single Linked List), διπλά συνδεδεμένη (Double Linked List), κυκλική απλά συνδεδεμένη (Single Circular List) και κυκλική διπλά συνδεδεμένη λίστα (Double Circular List).



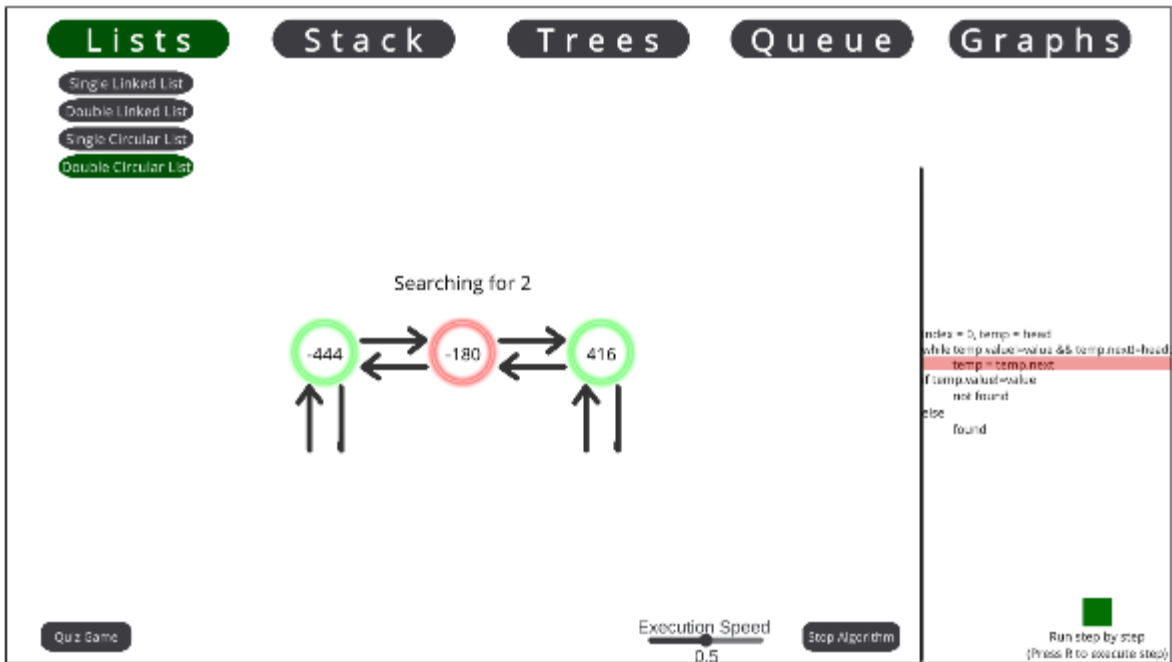
Εικόνα 4.28 Τύποι λιστών

Επιλέγοντας οποιοδήποτε είδος λίστας εμφανίζονται οι επιλογές της ένθεσης κόμβου (Add), διαγραφής κόμβου (Delete) και αναζήτησης κόμβου (Search) (Εικόνα 4.29). Κατά την διαγραφή και την αναζήτηση κόμβου ο χρήστης πρέπει να πληκτρολογήσει τον κόμβο που επιθυμεί να διαγράψει ή να αναζητήσει.



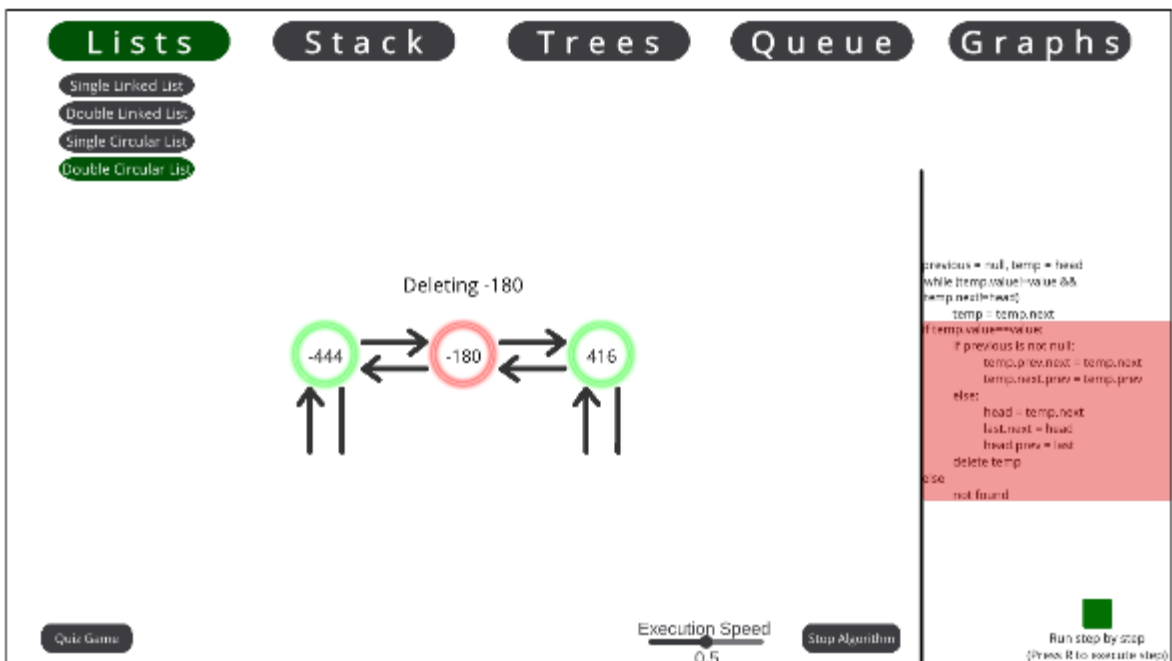
Εικόνα 4.29 Αλγόριθμοι λιστών

Κατά την αναζήτηση για παράδειγμα του κόμβου δύο, ο χρήστης πρέπει να πληκτρολογήσει τον αριθμό δύο και να επιλέξει το κουμπί αναζήτησης (search). Ο αλγόριθμος ξεκινά να διατρέχει την λίστα από την αρχή μέχρι να τον συναντήσει (Εικόνα 4.30).

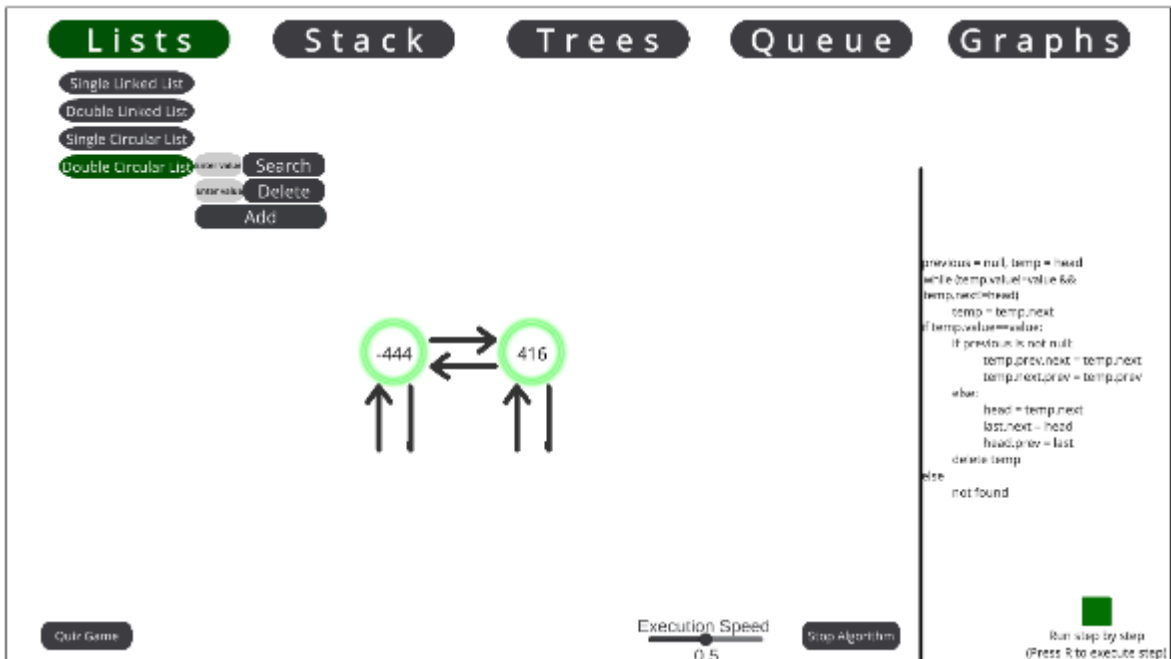


Εικόνα 4.30 Αναζήτηση του κόμβου δύο

Κατά την λειτουργία της διαγραφής, ο χρήστης πρέπει πάλι να πληκτρολογήσει τον κόμβο τον οποίο επιθυμεί να διαγράψει. Για παράδειγμα κατά την διαγραφή του κόμβου (-180) ο αλγόριθμος διατρέχει την λίστα από την αρχή μέχρι να συναντήσει τον αντίστοιχο κόμβο (Εικόνα 4.30). Εάν δεν τον βρει εμφανίζεται μήνυμα αποτυχίας, ειδάλλως τον διαγράφει επιτυχώς (Εικόνα 4.31).

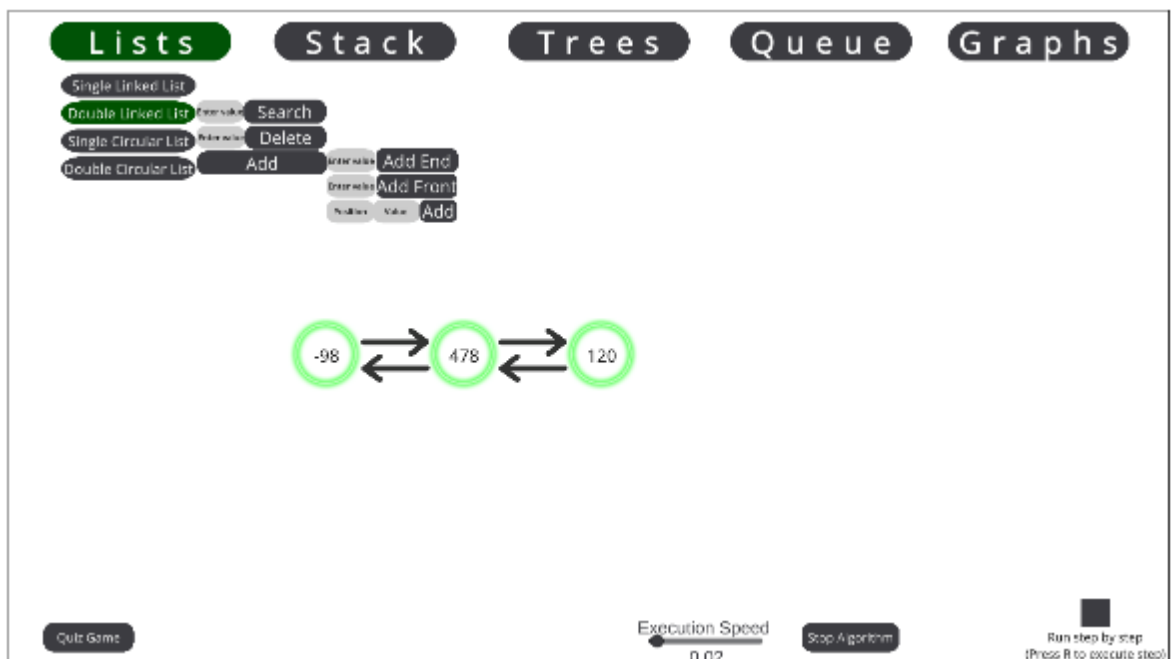


Εικόνα 4.31 Αναζήτηση κόμβου -180 για διαγραφή



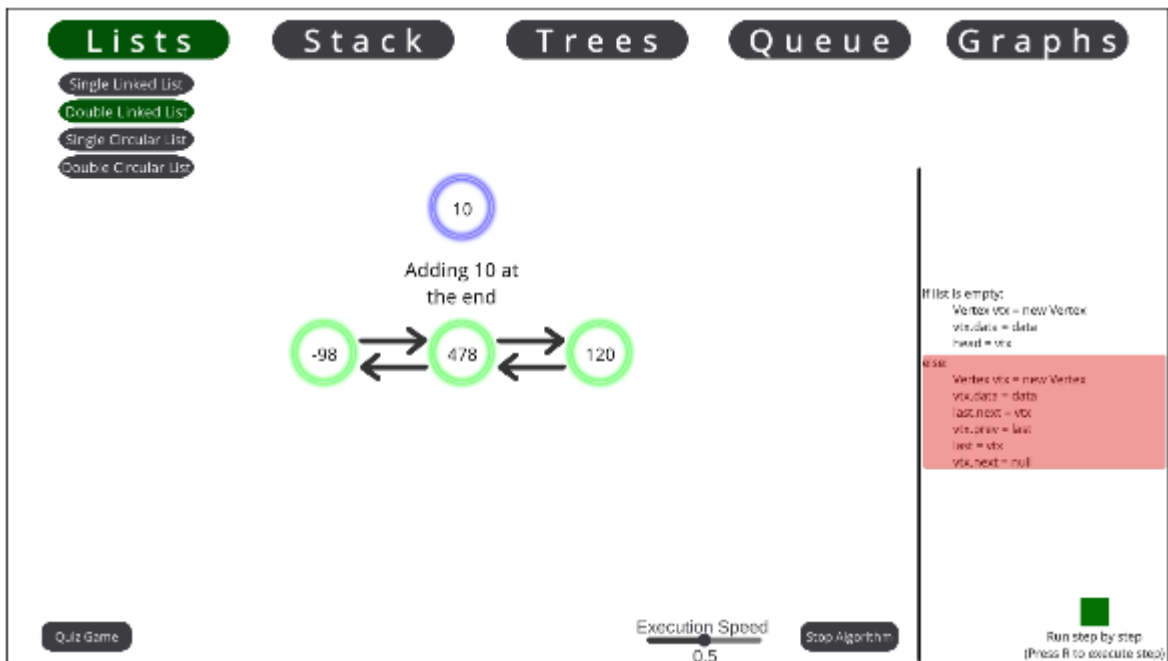
Εικόνα 4.32 Επιτυχής διαγραφή του κόμβου -180

Στην επιλογή της ένθεσης κόμβου, υπάρχουν οι εξής δυνατότητες, ένθεση στο τέλος της λίστας (Add End), ένθεση στην αρχή της λίστας (Add Front) και ένθεση σε θέση που προσδιορίζει ο χρήστης (Add) (Εικόνα 4.33).

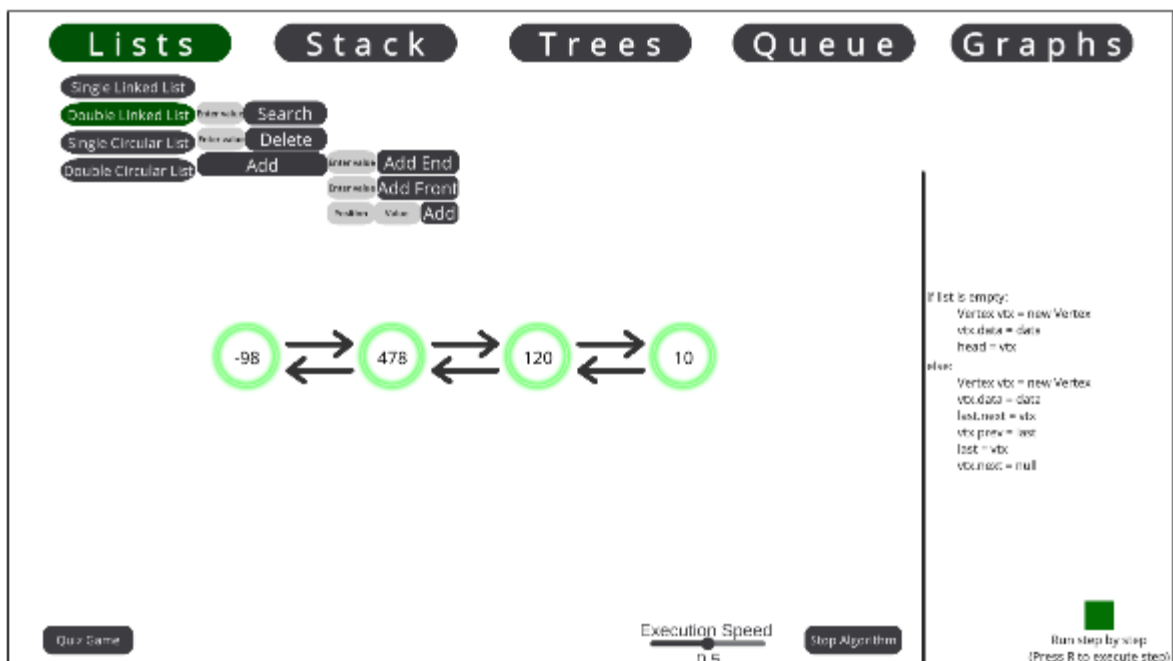


Εικόνα 4.33 Ένθεση κόμβου

Κατά την ένθεση κόμβου στο τέλος, ο χρήστης πρέπει να πληκτρολογήσει την τιμή του κόμβου που επιθυμεί να προσθέσει. Ο αλγόριθμος διατηρώντας έναν δείκτη στο τέλος της λίστας (Εικόνα 4.34), ενθέτει τον αντίστοιχο κόμβο (Εικόνα 4.35).

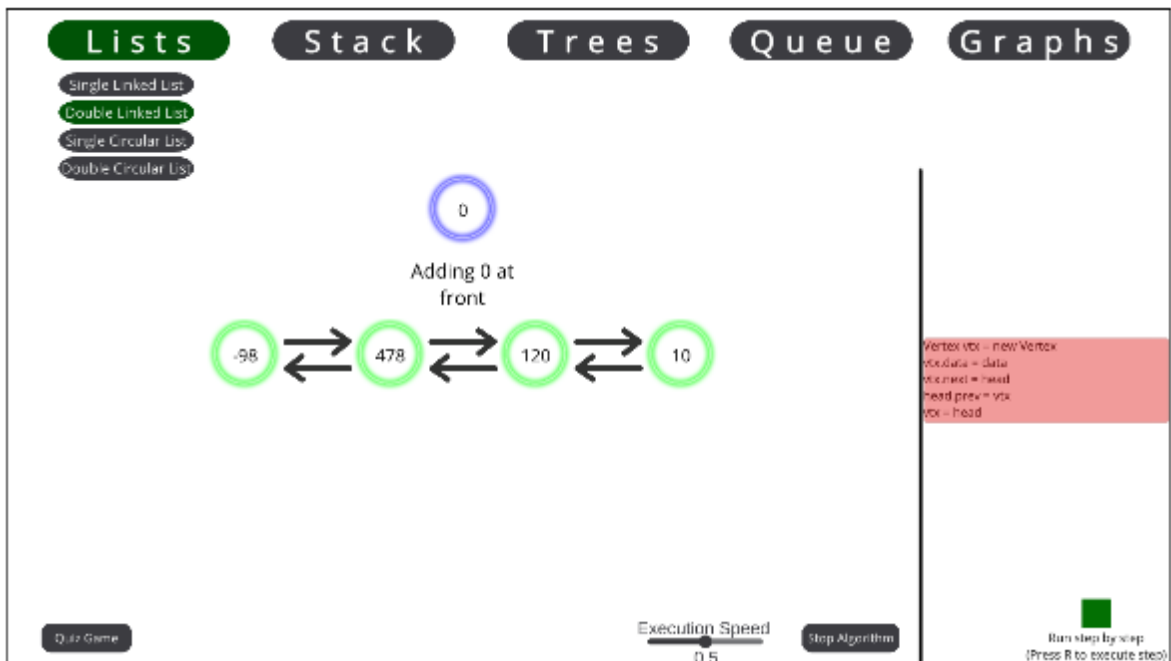


Εικόνα 4.34 Αλγόριθμος της ένθεσης στο τέλος της λίστας

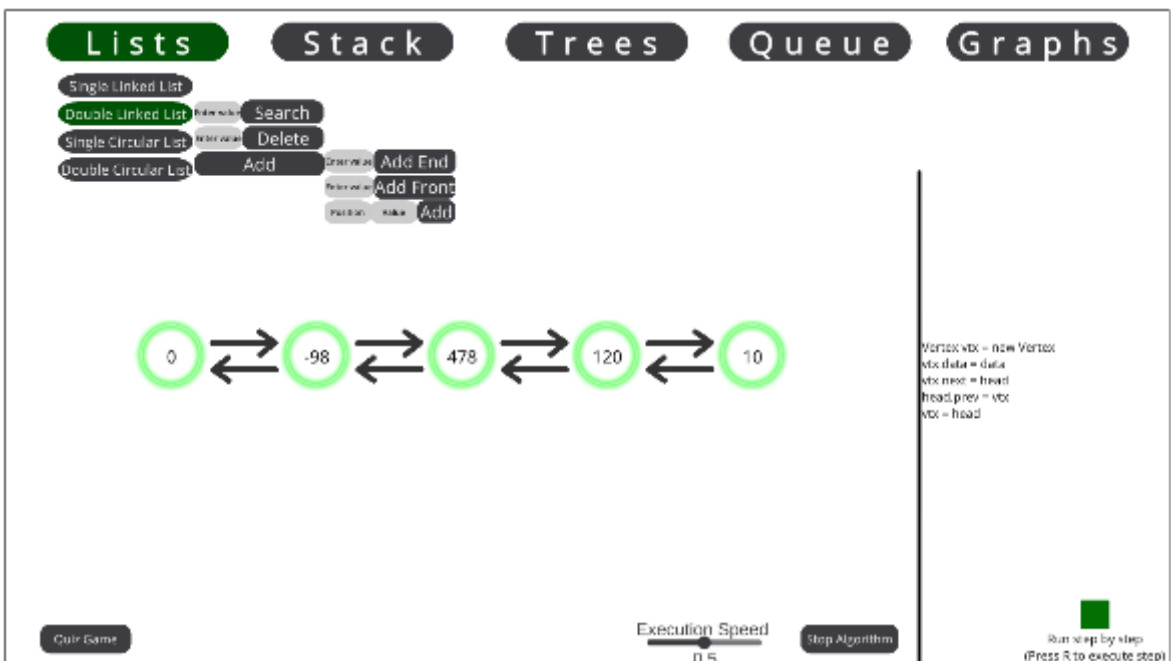


Εικόνα 4.35 Ένθεση του κόμβου 10 στο τέλος της λίστας

Κατά την ένθεση κόμβου στην αρχή, ο χρήστης πρέπει να πληκτρολογήσει την τιμή του κόμβου που επιθυμεί να προσθέσει. Ο αλγόριθμος διατηρώντας έναν δείκτη στην αρχή της λίστας (Εικόνα 4.36), ενθέτει τον αντίστοιχο κόμβο (Εικόνα 4.37).

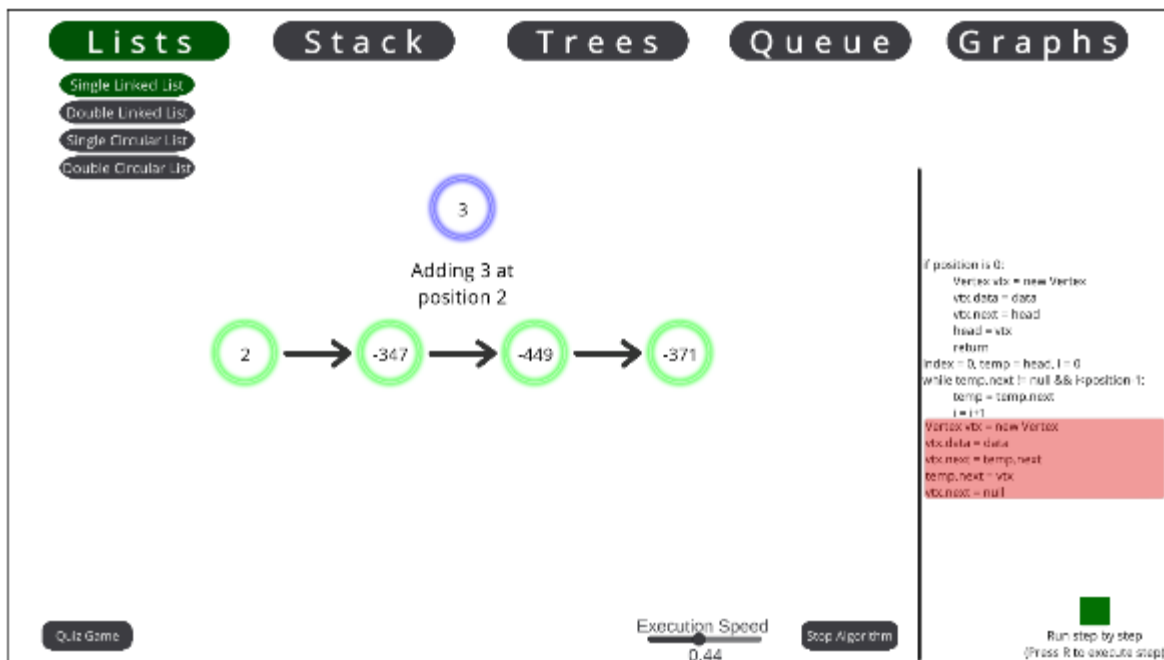


Εικόνα 4.36 Αλγόριθμος ένθεσης στην αρχή της λίστας

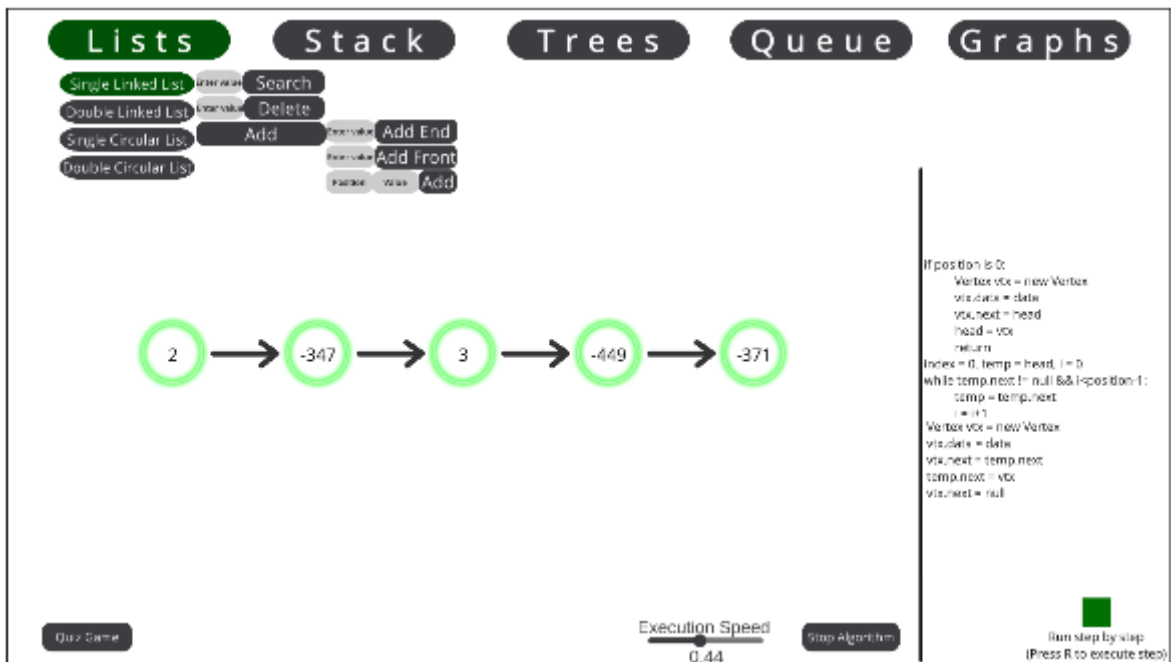


Εικόνα 4.37 Ένθεση του κόμβου 0 στην αρχή της λίστας

Κατά την ένθεση κόμβου σε συγκεκριμένη θέση της λίστας, ο χρήστης πρέπει να πληκτρολογήσει την τιμή του κόμβου που επιθυμεί να προσθέσει και την αντίστοιχη θέση. Ο αλγόριθμος διατρέχει τη λίστα μέχρι να βρει την συγκεκριμένη (Εικόνα 4.38) θέση και ενθέτει τον κόμβο (Εικόνα 4.39).



Εικόνα 4.38 Αλγόριθμος ένθεσης σε συγκεκριμένη θέση της λίστας



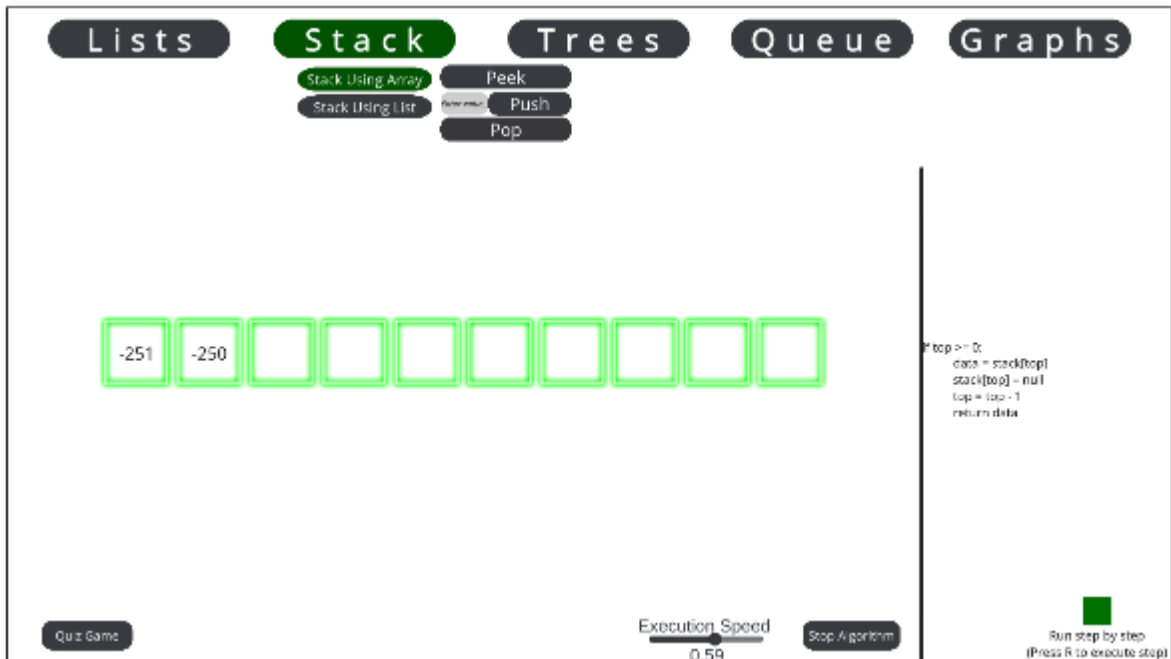
Εικόνα 4.39 Ένθεση του κόμβου 3 στην θέση 2

4.4.2 Στοιβες

Πατώντας την επιλογή στοιβες (Stacks) της εφαρμογής αναγράφονται δύο διαφορετικά είδη στοιβών (Εικόνα 4.40). Στοιβες με πίνακα (Stacks Using Array) (Εικόνα 4.41) και στοιβες με λίστες (Stacks using Lists) (Εικόνα 4.42). Επιλέγοντας οποιοδήποτε τύπο στοιβας εμφανίζονται οι τρεις αλγόριθμοι, peek , push and pop.



Εικόνα 4.40 Τύποι στοιβών

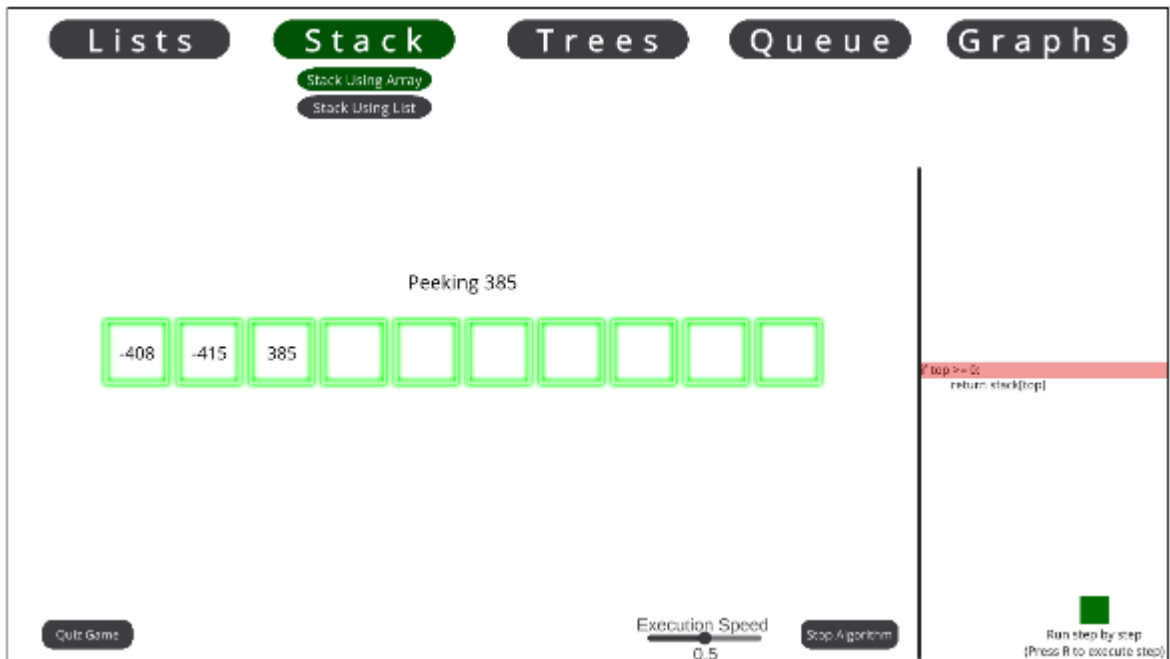


Εικόνα 4.41 Αλγόριθμοι Στοιβών με πίνακες

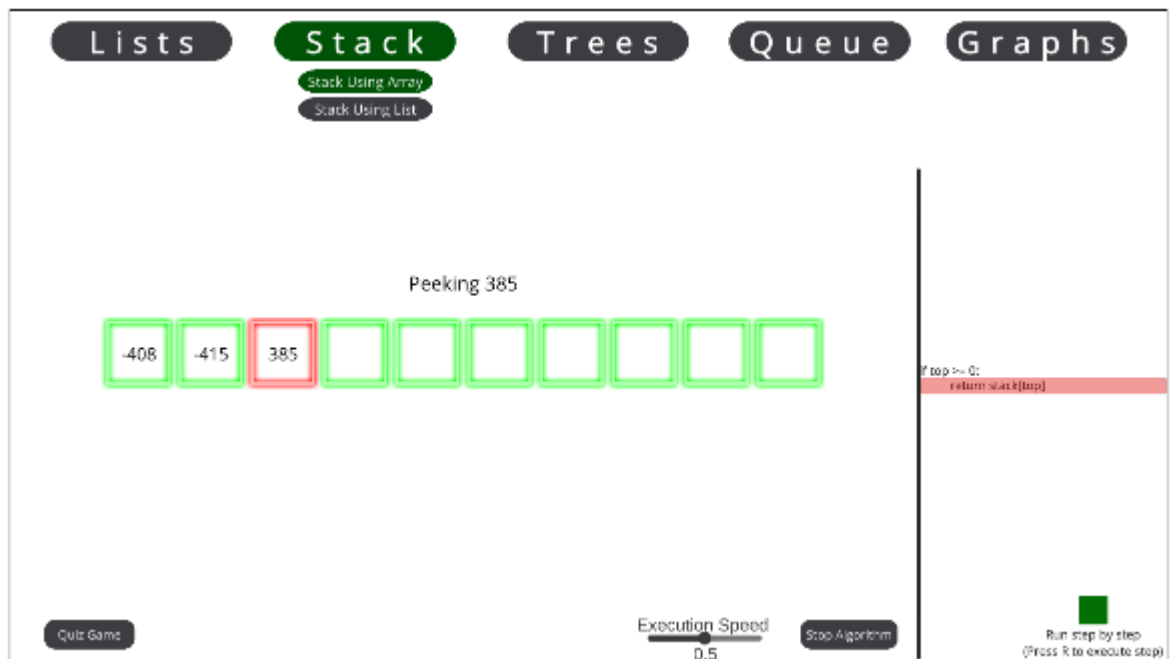


Εικόνα 4.42 Αλγόριθμοι Στοιβών με λίστες

Στις στοίβες με πίνακες κατά τον αλγόριθμο Peek επιστρέφεται η τιμή η οποία πρόκειται να γίνει pop (Εικόνα 4.43), (Εικόνα 4.44).

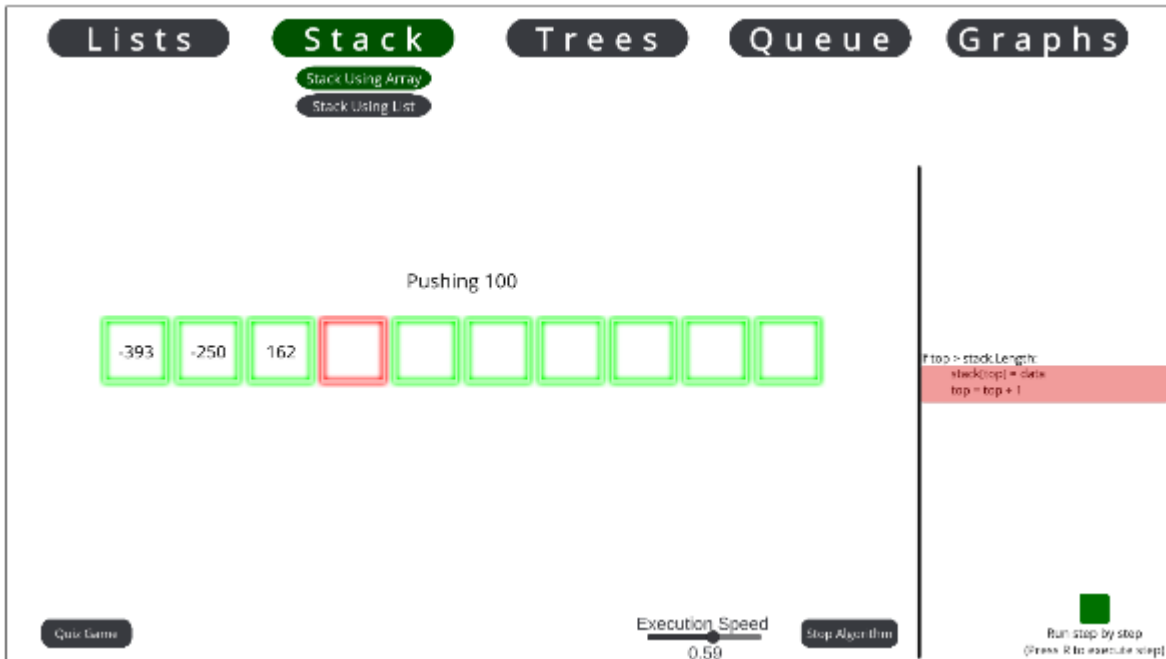


Εικόνα 4.43 Αλγόριθμός Peek σε στοίβες με πίνακες

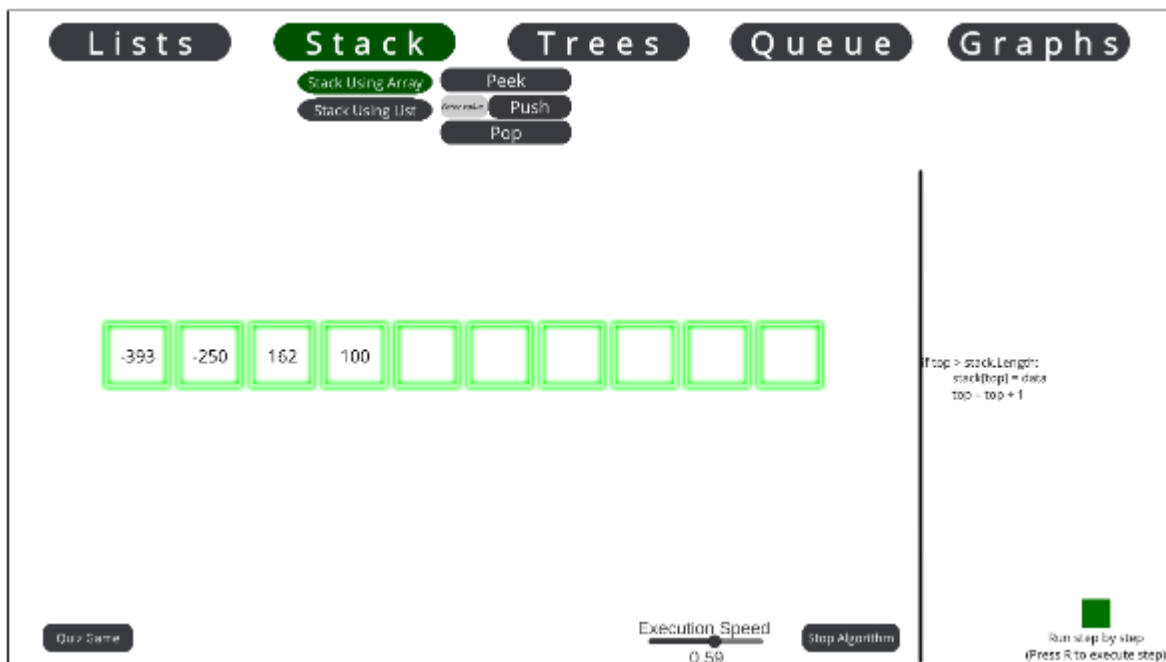


Εικόνα 4.44 Peek 385 σε στοίβες με πίνακες

Κατά τον αλγόριθμο push βάζει στην στοίβα με πίνακες την τιμή που θα πληκτρολογήσει ο χρήστης (Εικόνα 4.45), (Εικόνα 4.46).

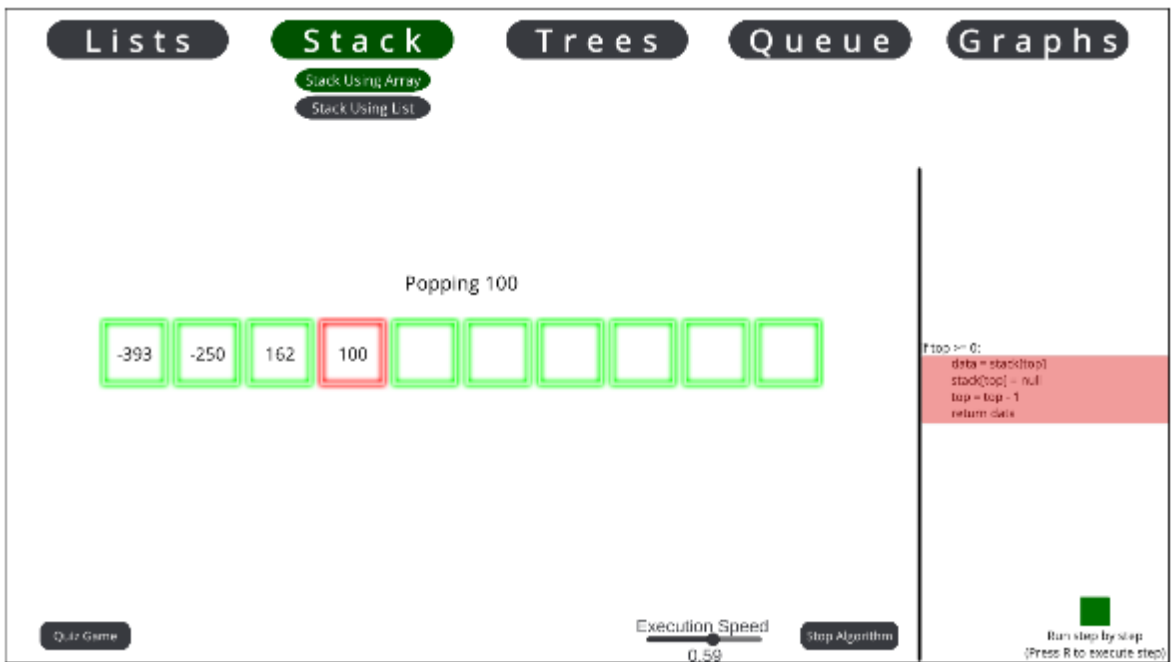


Εικόνα 4.45 Αλγόριθμος Push σε στοίβα με πίνακα

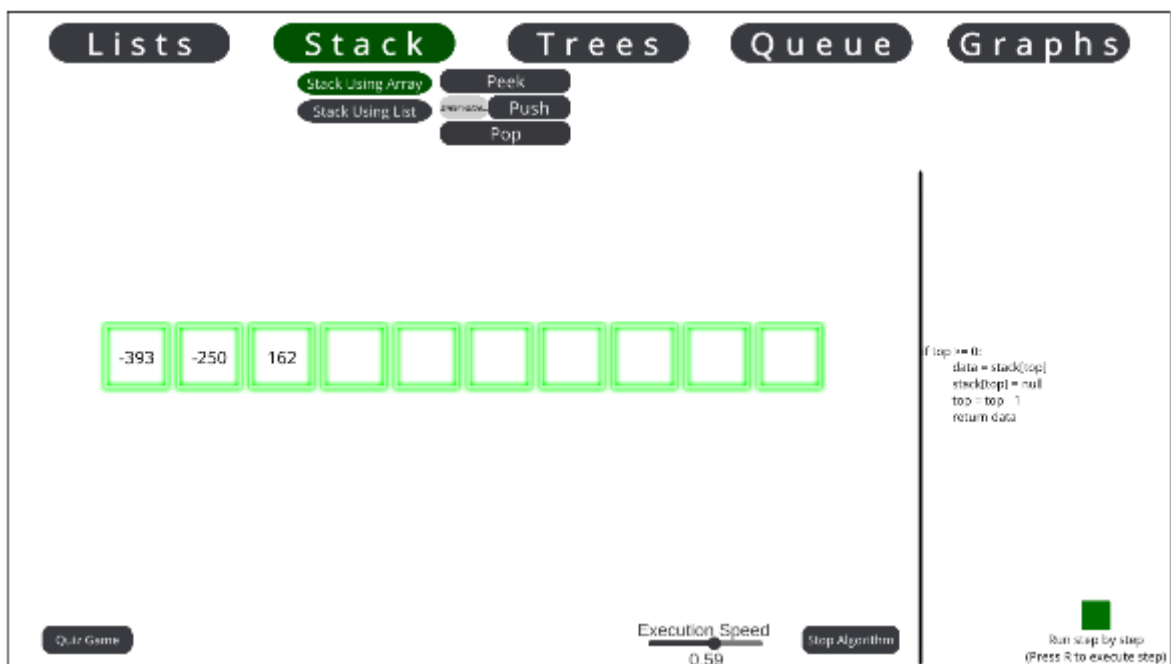


Εικόνα 4.46 Push του 100 σε στοίβα με πίνακα

Κατά τον αλγόριθμο pop βγάξει το τελευταίο στοιχείο από την στοίβα με πίνακες (Εικόνα 4.47), (Εικόνα 4.48).

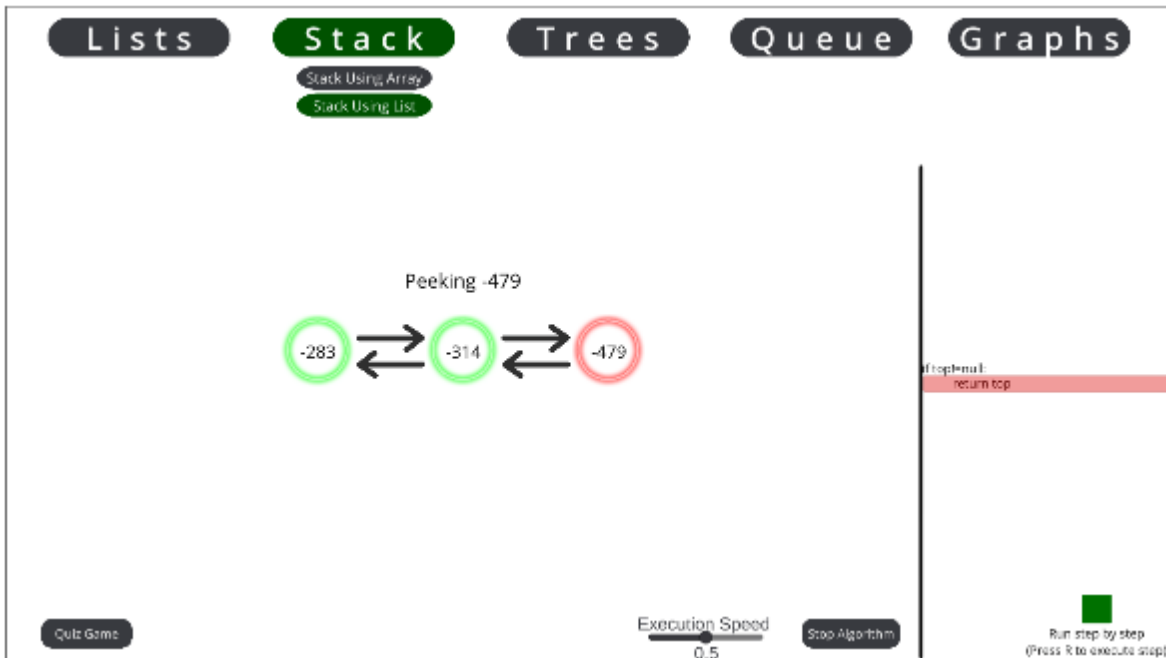


Εικόνα 4.47 Αλγόριθμος Pop σε στοίβα με πίνακες

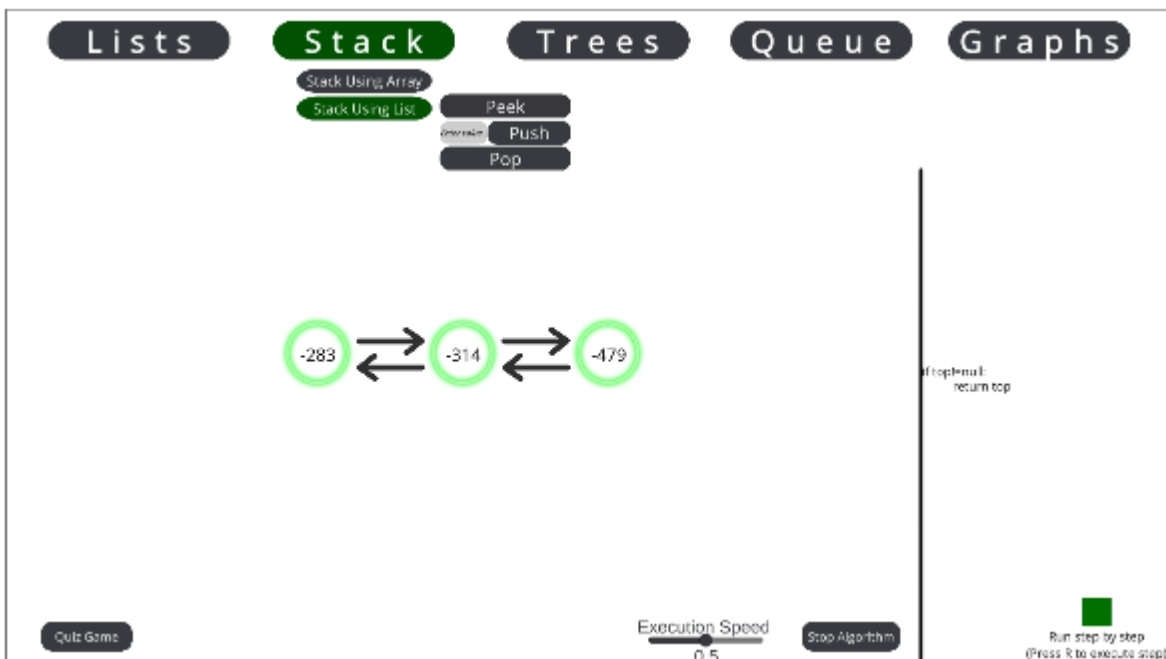


Εικόνα 4.48 Pop του 100

Στις στοίβες με λίστες κατά τον αλγόριθμο Peek επιστρέφεται η τιμή η οποία πρόκειται να γίνει pop (Εικόνα 4.49), (Εικόνα 4.50).

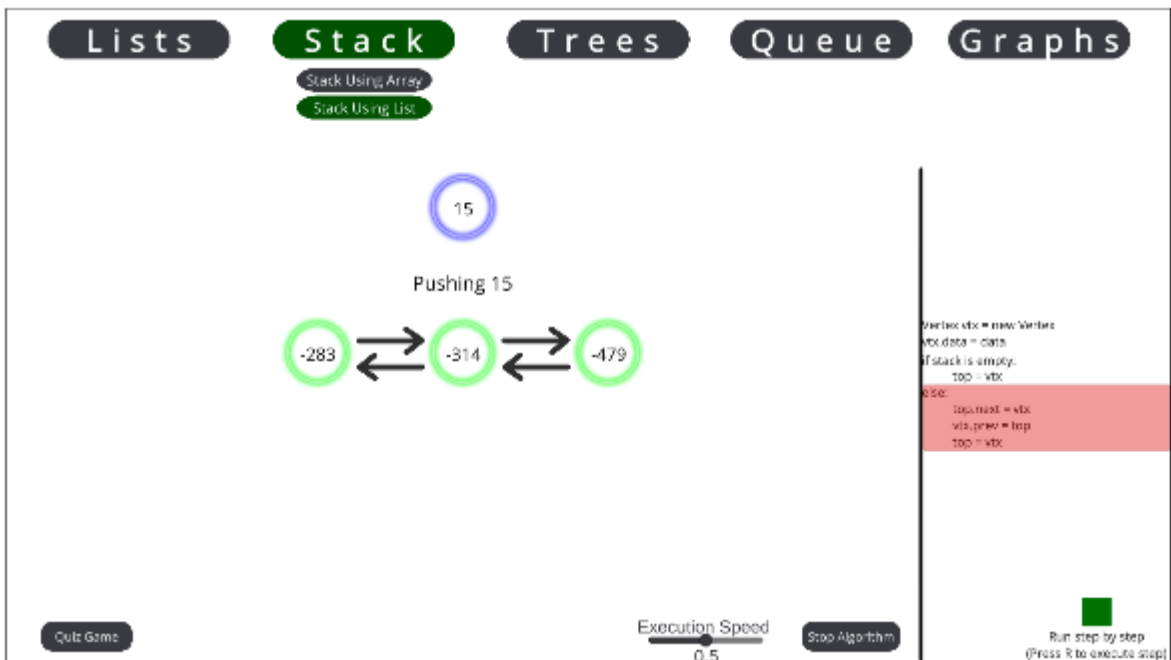


Εικόνα 4.49 Αλγόριθμος Peek σε στοίβες με λίστες

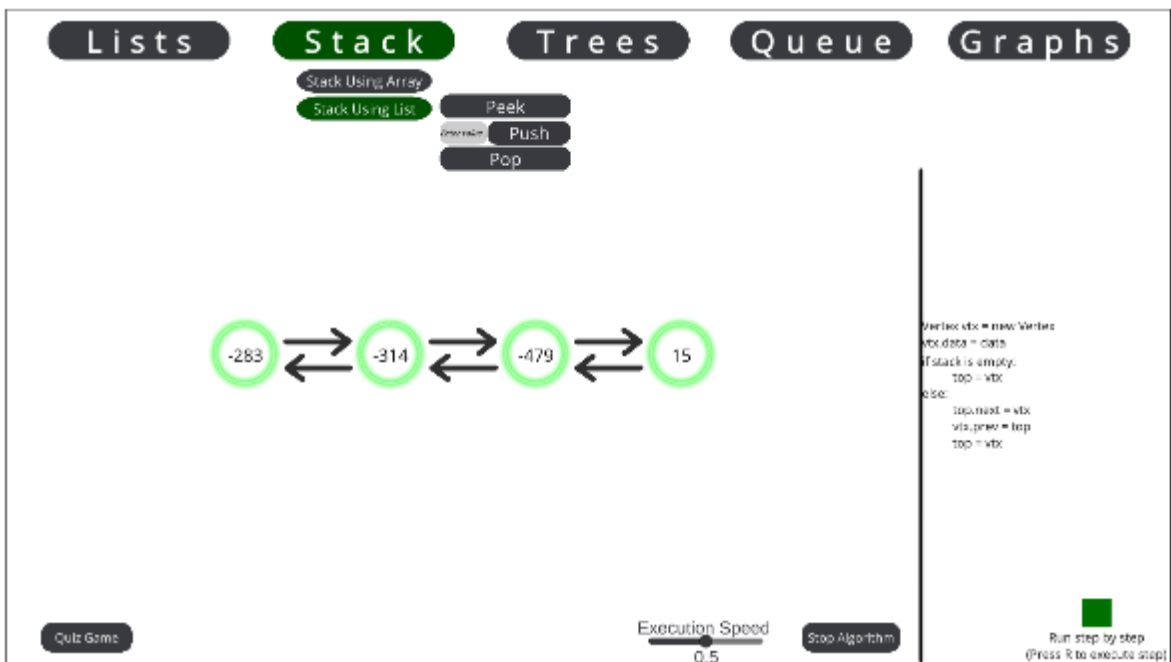


Εικόνα 4.50 Peek του -479 σε στοίβες με λίστες

Κατά τον αλγόριθμο push εισέρχεται στην στοίβα η τιμή που θα πληκτρολογήσει ο χρήστης (Εικόνα 4.51), (Εικόνα 4.52).



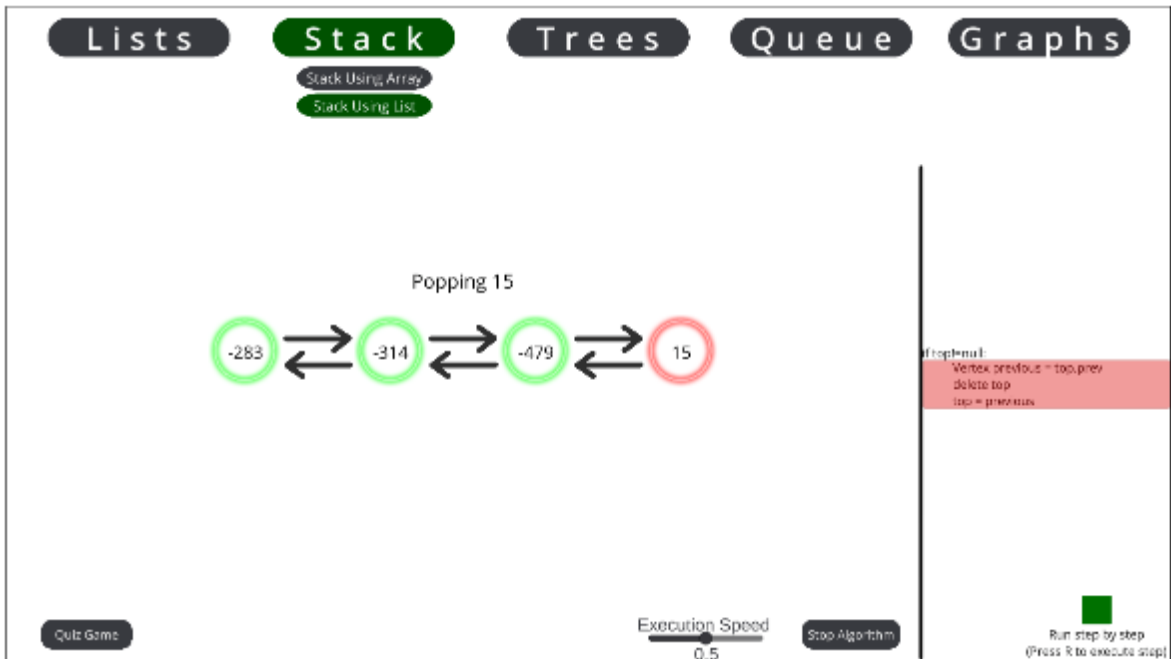
Εικόνα 4.51 Αλγόριθμος Push σε στοίβες με λίστες



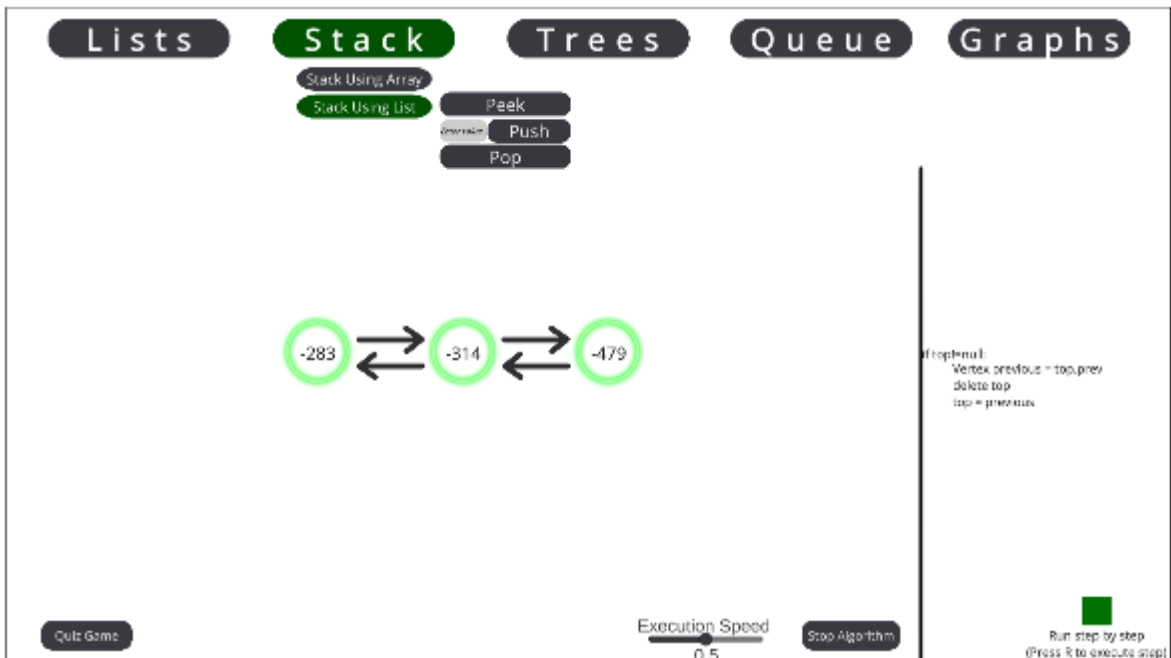
Εικόνα 4.52 Push του 15 σε στοίβες με λίστες

Κατά τον αλγόριθμο pop εξέρχεται το τελευταίο στοιχείο από την στοίβα με λίστα (Εικόνα 4.53), (Εικόνα 4.54). Είναι εμφανές ότι οι αλγόριθμοι λειτουργούν ακριβώς το ίδιο και στα δύο είδη στοίβας παρόλο που αποτελούνται από διαφορετικές δομές δεδομένων. Για την διευκόλυνση του χρήστη σε όλους τους αλγόριθμους που αναλύθηκαν παραπάνω, η τιμή

η οποία γίνεται Peek, Pop και Push εμφανίζεται κάθε φορά πάνω από τη στοίβα και στα δεξιά της εφαρμογής περιγράφεται ο κώδικας της κάθε λειτουργίας.



Εικόνα 4.53 Αλγόριθμος Pop σε στοίβες με λίστες



Εικόνα 4.54 Pop του 15 σε στοίβες με λίστες

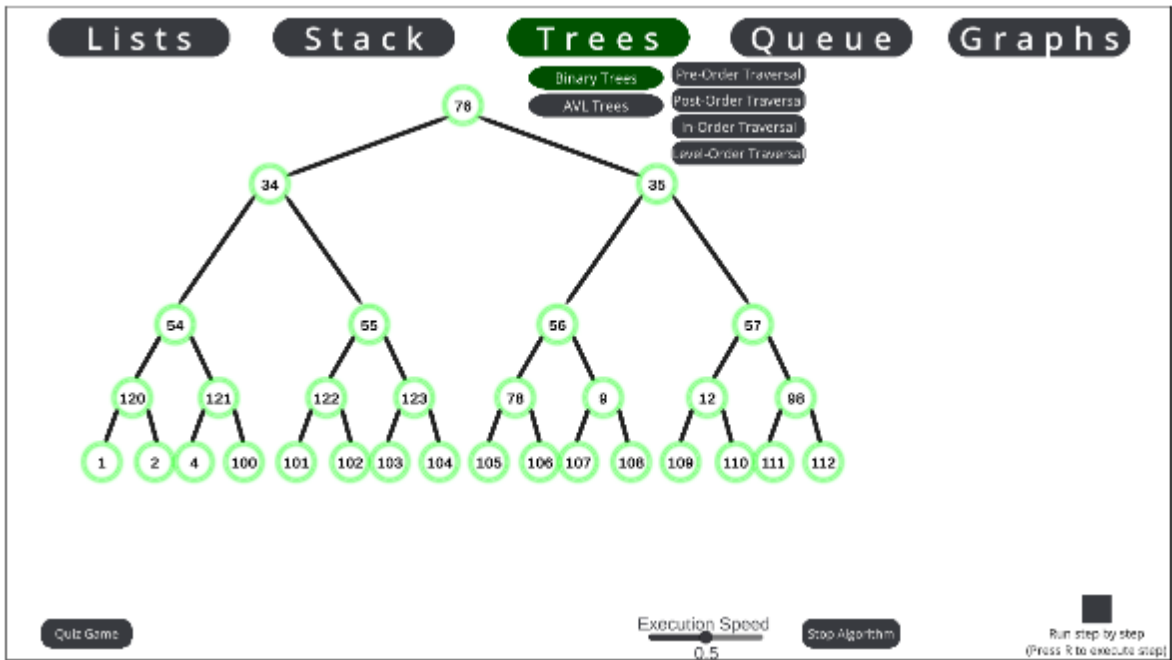
4.4.3 Δέντρα

Πατώντας το κουμπί δέντρα (Trees), εμφανίζονται οι εξής δύο επιλογές, δυαδικά δέντρα (Binary Trees) και δέντρα AVL (AVL Trees) (Εικόνα 4.55).



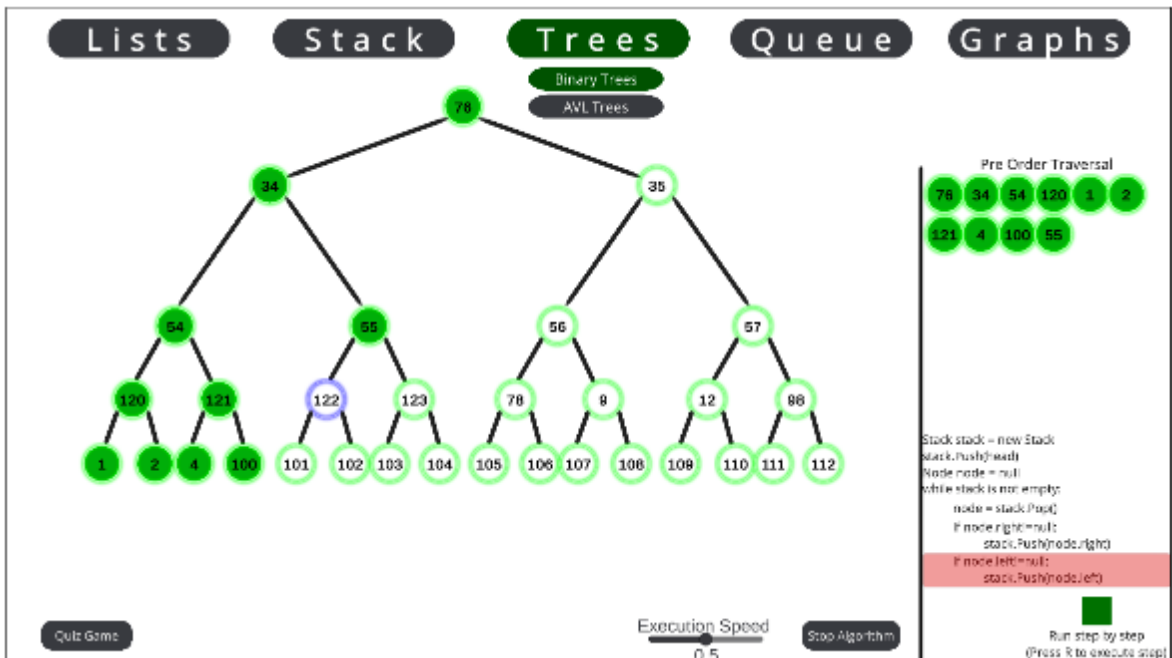
Εικόνα 4.55 Κατηγορίες δέντρων

Πατώντας την επιλογή δυαδικά δέντρα (Binary Trees), εμφανίζεται ένα δυαδικό δέντρο πέντε επιπέδων και οι τέσσερις αλγόριθμοι διαπέρασης, προδιατεταγμένη διαπέραση (Pre-Order Traversal), μεταδιατεταγμένη διαπέραση (Post-Order Traversal), ενδοδιατεταγμένη διαπέραση (In-Order Traversal) και κατά σειρά επιπέδων διαπέραση (Level-Order Traversal) (Εικόνα 4.56).



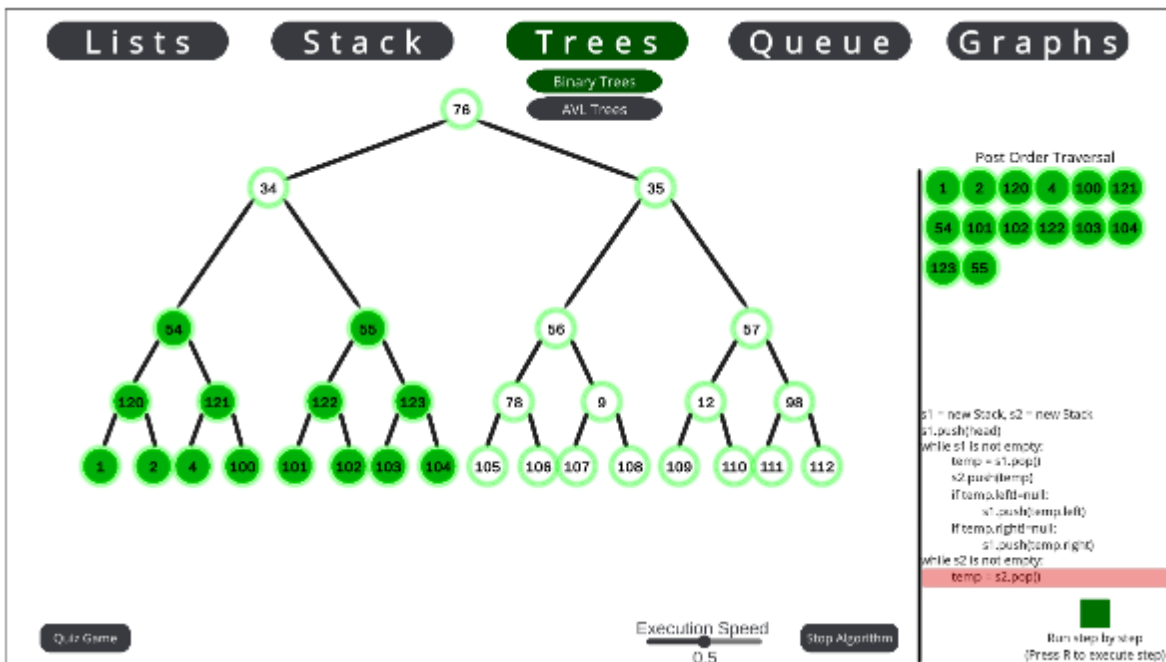
Εικόνα 4.56 Δυαδικά δέντρα

Κατά την προδιατεταγμένη διαπέραση δυαδικού δέντρου (Pre-Order Traversal), ένας κόμβος του δέντρου θεωρείται πως προηγείται των παιδιών του (Εικόνα 4.57)



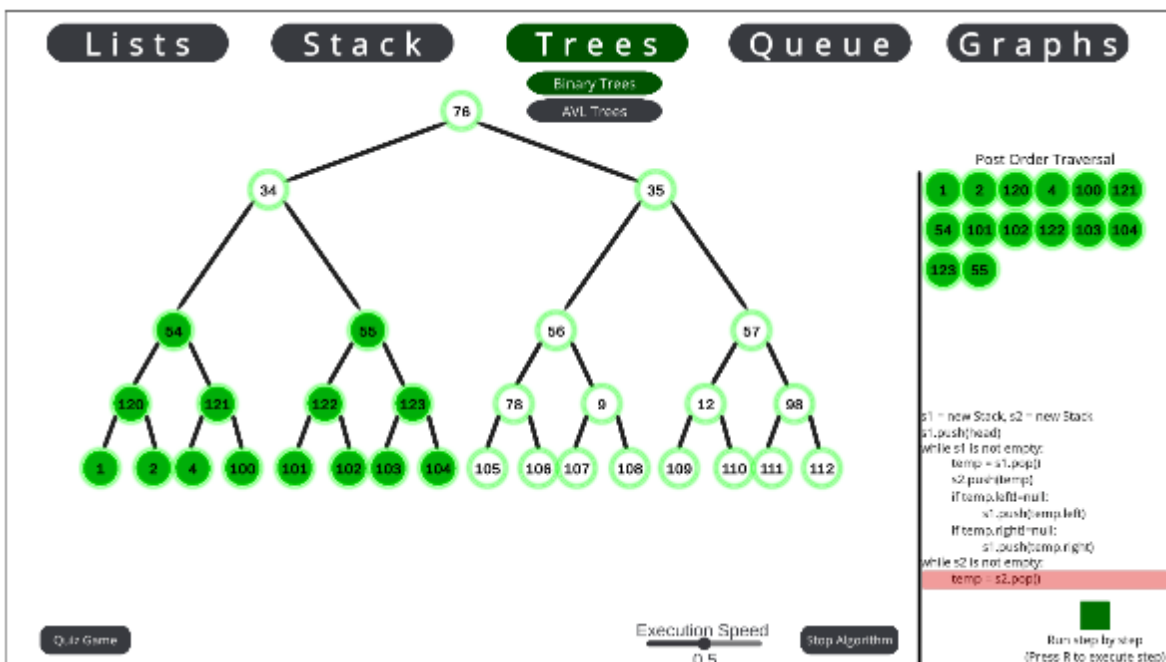
Εικόνα 4.57 Αλγόριθμος προδιατεταγμένης διαπέρασης Pre-Order Traversal

Κατά τον αλγόριθμο της μεταδιατεταγμένης διαπέρασης δυαδικού δέντρου (Post-Order Traversal), ένας κόμβος έπεται των παιδιών του (Εικόνα 4.58).



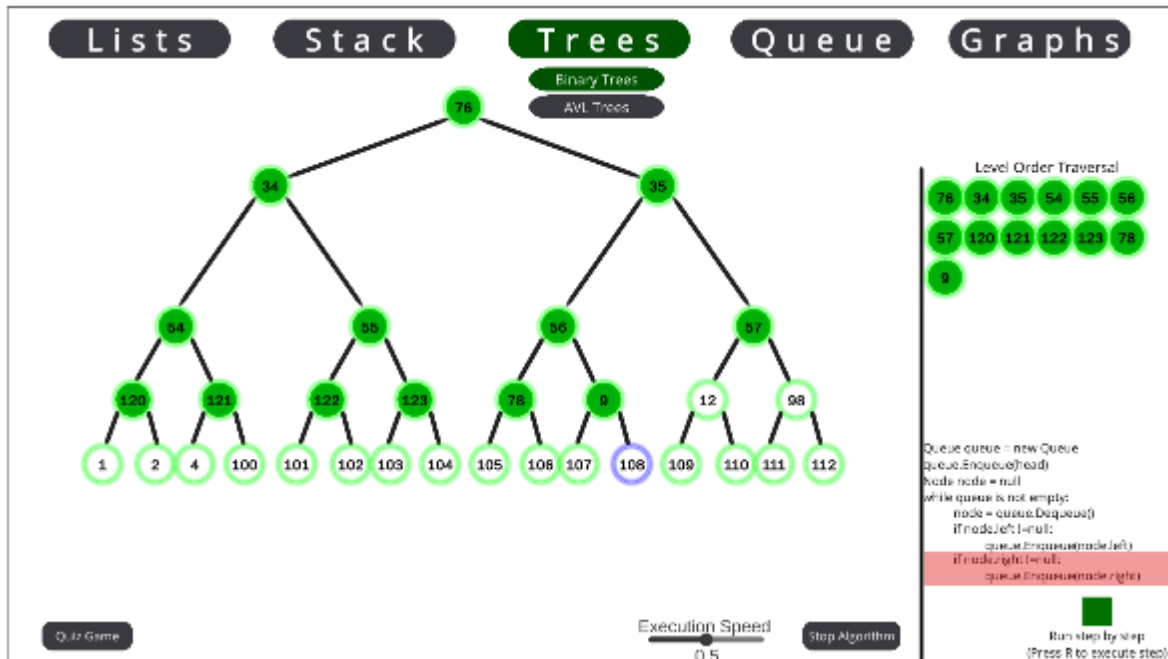
Εικόνα 4.58 Αλγόριθμος μεταδιατεταγμένης διαπέρασης Post-Order Traversal

Κατά τον αλγόριθμο της ενδοδιατεταγμένης διαπέρασης δυαδικού δέντρου (In-Order Traversal), ορίζει ότι ένας κόμβος έπεται του αριστερού παιδιού, αλλά προηγείται του δεξιού παιδιού (Εικόνα 4.59)



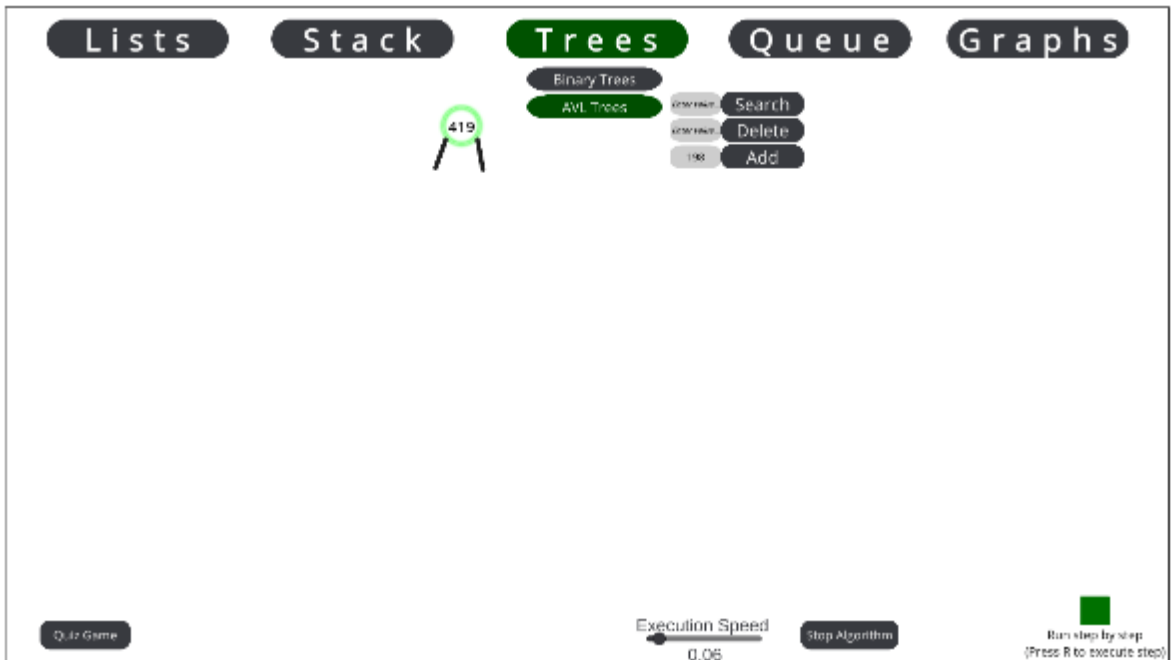
Εικόνα 4.59 Αλγόριθμος ενδοδιατεταγμένης διαπέρασης In-Order traversal

Κατά τον αλγόριθμο της κατά σειράς επιπέδου διαπέρασης διαδικού δέντρου (Level-Order Traversal), επισκέπτεται τους κόμβους επίπεδο προς επίπεδο από πάνω προς τα κάτω και από αριστερά προς τα δεξιά (Εικόνα 4.60).



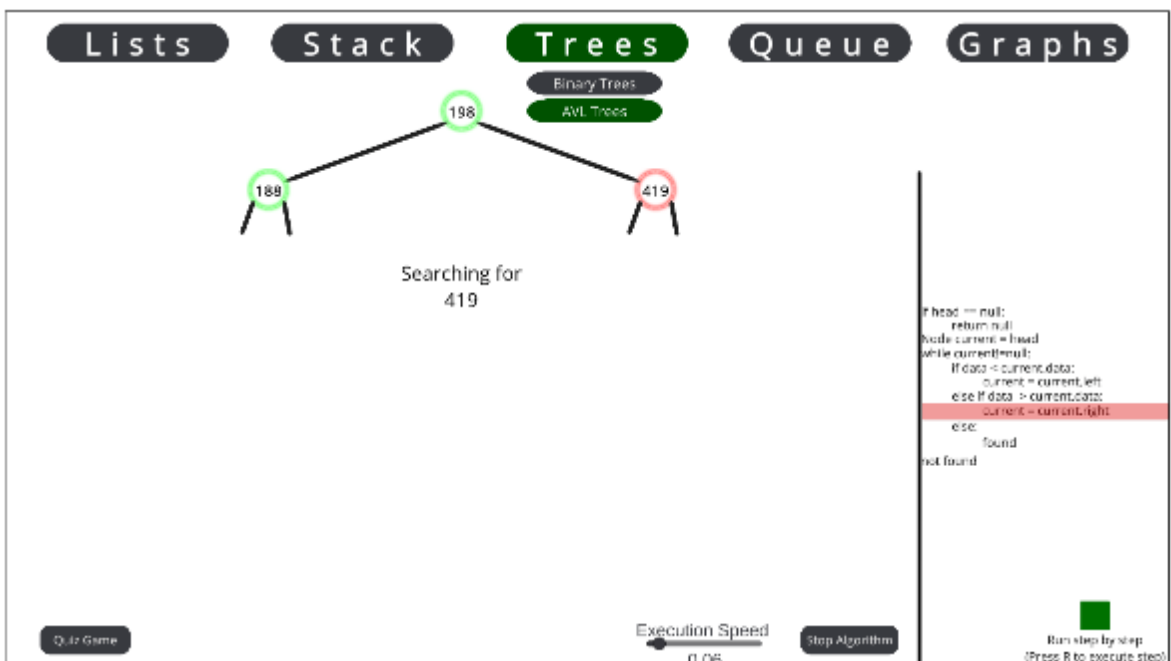
Εικόνα 4.60 Αλγόριθμος διαπέρασης κατά σειρά επιπέδου Level-Order traversal

Ο χρήστης πατώντας την επιλογή για δέντρα AVL, εμφανίζεται η ρίζα του δέντρου με μία τυχαία τιμή όπου ο χρήστης μπορεί να ενθέσει οποιαδήποτε τιμή επιθυμεί κάνοντας κλικ πάνω στον κόμβο. Επίσης εμφανίζονται οι εξής αλγόριθμοι, αναζήτηση κόμβου (Search), διαγραφή κόμβου (delete) και ένθεση κόμβου (Add) (Εικόνα 4.61).



Εικόνα 4.61 AVL δέντρα

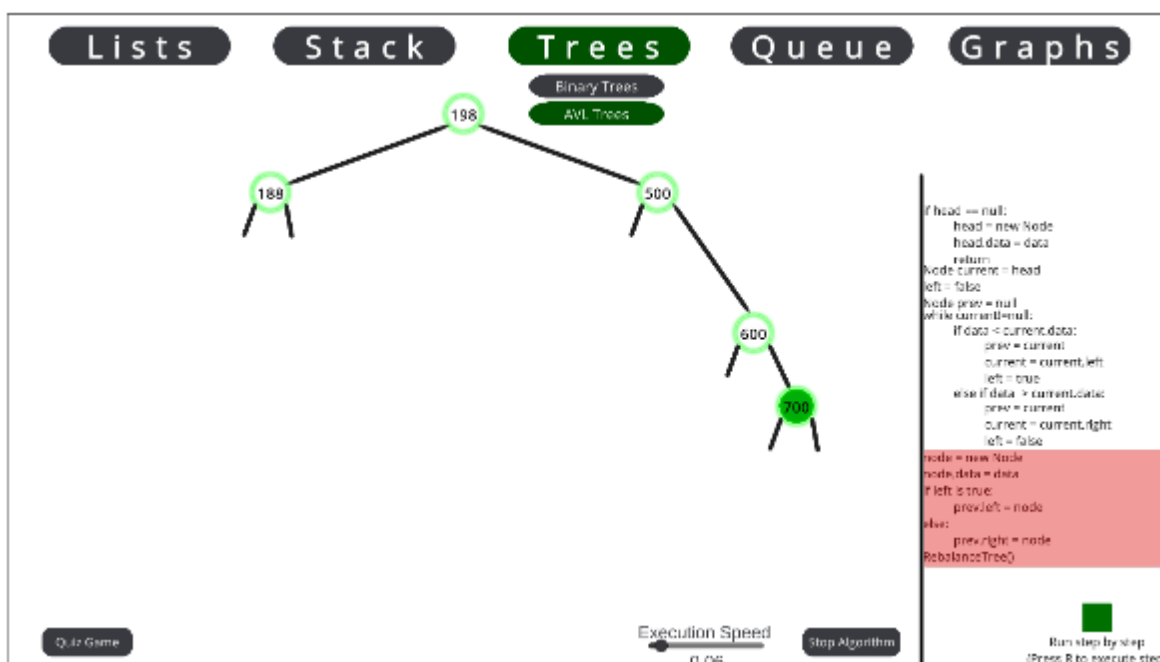
Κατά τον αλγόριθμο αναζήτησης (Search), ο χρήστης πληκτρολογεί την τιμή του κόμβου που επιθυμεί να αναζητήσει και ο αλγόριθμος εφαρμόζει δυαδική αναζήτηση μέχρι να βρει τον συγκεκριμένο κόμβο. Συγκρίνεται το κλειδί του τρέχοντος κόμβου με το κλειδί του κόμβου προς αναζήτηση. Εάν είναι ίσα, τότε ο κόμβος έχει εντοπιστεί. Διαφορετικά κινείται είτε αριστερά εάν το κλειδί είναι μικρότερο είτε δεξιά εάν το κλειδί είναι μεγαλύτερο. (Εικόνα 4.62).



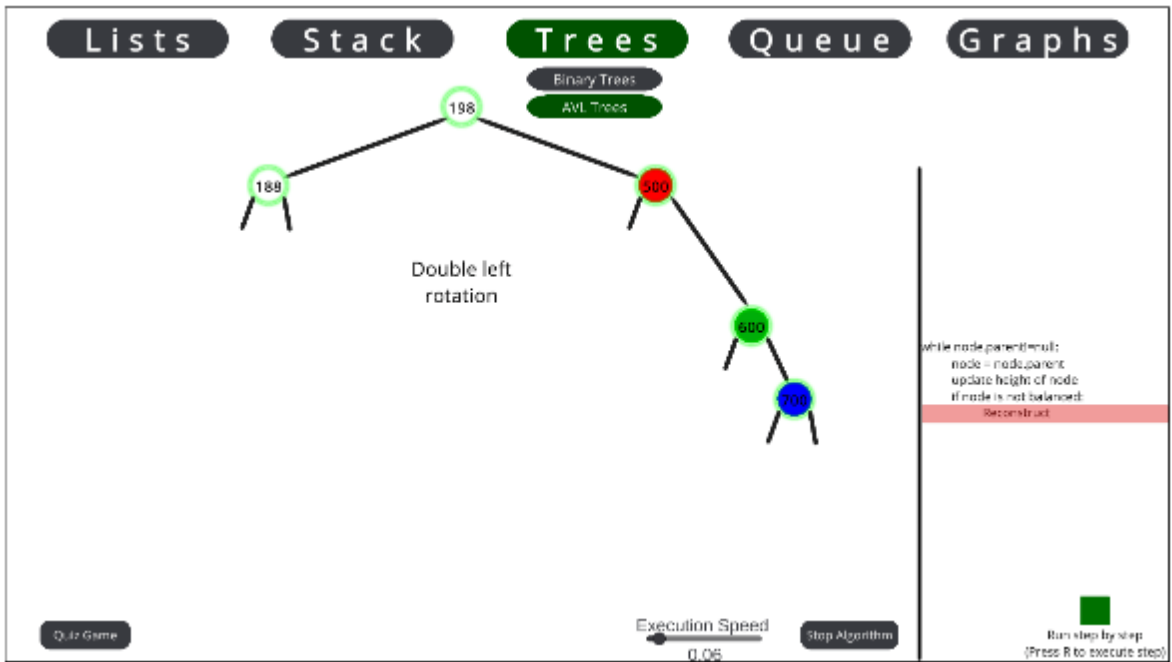
Εικόνα 4.62 Αλγόριθμος αναζήτησης δυαδικού δέντρου

Κατά τον αλγόριθμο ένθεσης (Add), ο χρήστης πληκτρολογεί την τιμή του κόμβου που επιθυμεί να ενθέσει. Κατά την ένθεση κόμβου συγκρίνεται το κλειδί του τρέχοντος κόμβου με το κλειδί του κόμβου ένθεσης, κινείται είτε αριστερά εάν το κλειδί είναι μικρότερο είτε δεξιά εάν το κλειδί είναι μεγαλύτερο. Κρίσιμο σημείο στην πράξη της ένθεσης αποτελούν οι επαναζυγιστικές πράξεις (rebalancing operation), οι οποίες προσπαθούν να διατηρήσουν τη διαφορά ύψους αδελφών κόμβων.

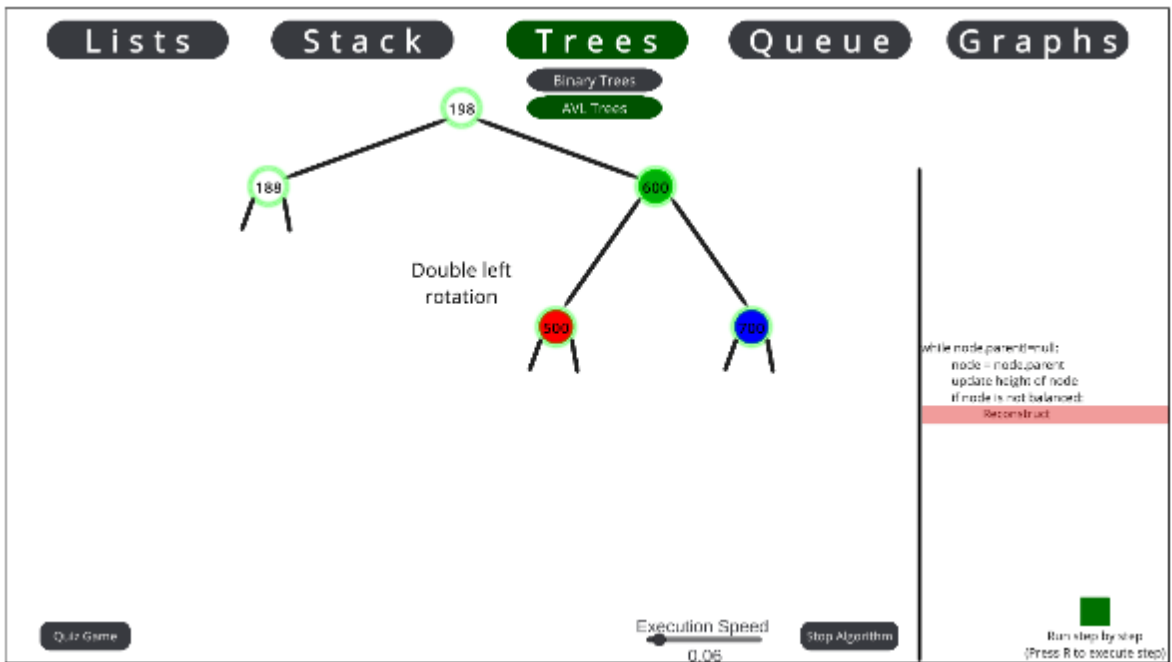
Εάν οι κόμβοι σχηματίζουν δεξιό ή αριστερό "ευθές" μονοπάτι παραβιάζεται η συνθήκη ισοζύγησης λόγω της αύξησης του ύψους από $h+2$ σε $h+3$ (Εικόνα 4.63), (Εικόνα 4.64), (Εικόνα 4.66), (Εικόνα 4.67). Τότε αρκεί μία, αριστερή απλή περιστροφή (Εικόνα 4.65) ή δεξιά απλή περιστροφή (Εικόνα 4.68).



Εικόνα 4.63 Ένθεση στοιχείου σε δέντρο AVL



Εικόνα 4.64 Εντοπισμός παραβίασης συνθήκης



Εικόνα 4.65 Αριστερή περιστροφή

Lists Stack **Trees** Queue Graphs

Binary Trees
AVL Trees

```

if head == null:
    head = new Node
    head.data = data
    return
Node current = head
left = false
Node prev = null
while current != null:
    if data < current.data:
        prev = current
        current = current.left
        left = true
    else if data > current.data:
        prev = current
        current = current.right
        left = false
node = new Node
node.data = data
if left is true:
    prev.left = node
else:
    prev.right = node
RebalanceTree()

```

Execution Speed 0.06 Step Algorithm

Runs step by step (Press R to execute step)

Εικόνα 4.66 Ένθεση στοιχείου σε δέντρο AVL

Lists Stack **Trees** Queue Graphs

Binary Trees
AVL Trees

Double right rotation

```

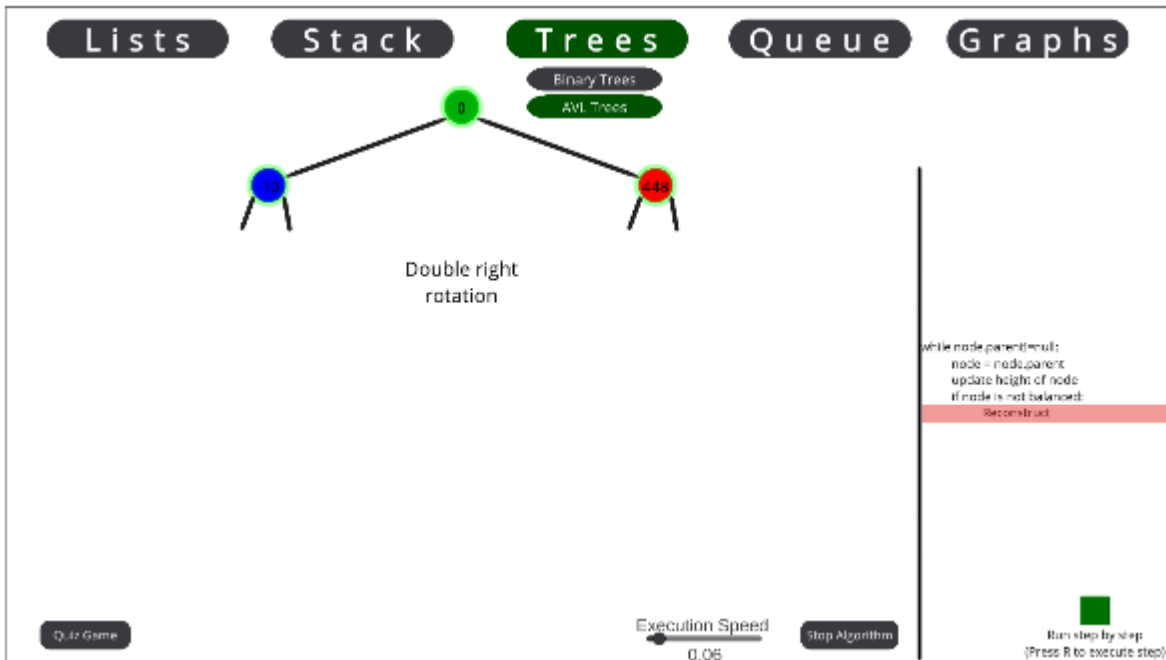
while node.parent != null:
    node = node.parent
    update height of node
    if node is not balanced:
        Reconstruct

```

Execution Speed 0.06 Step Algorithm

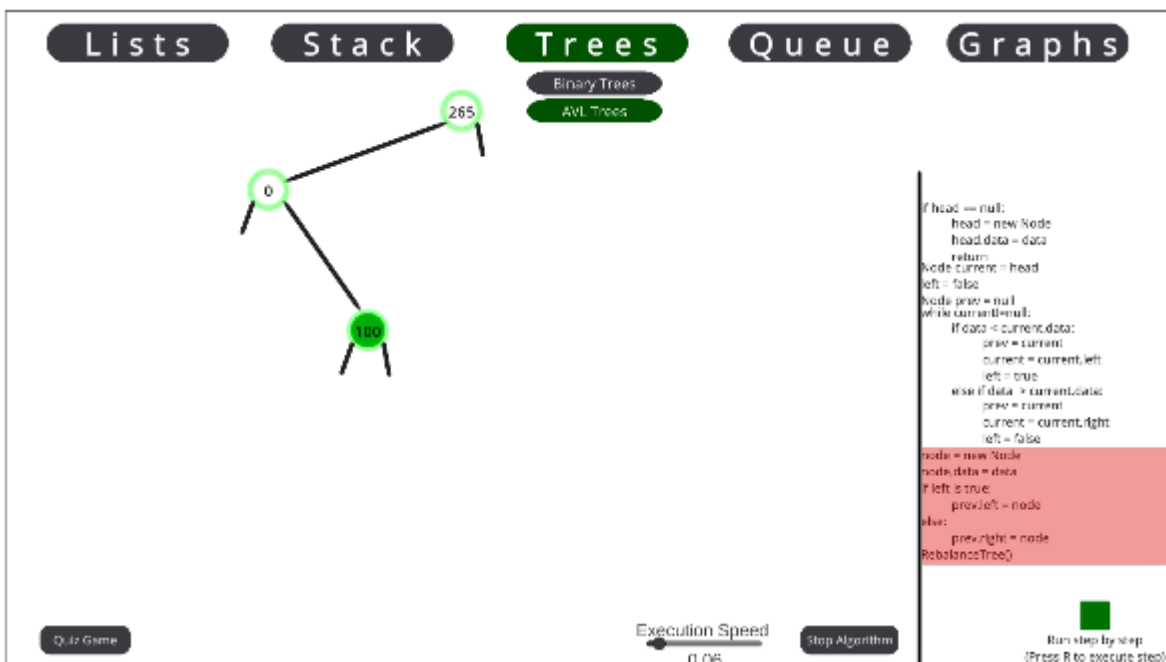
Runs step by step (Press R to execute step)

Εικόνα 4.67 Εντοπισμός παραβίασης συνθήκης

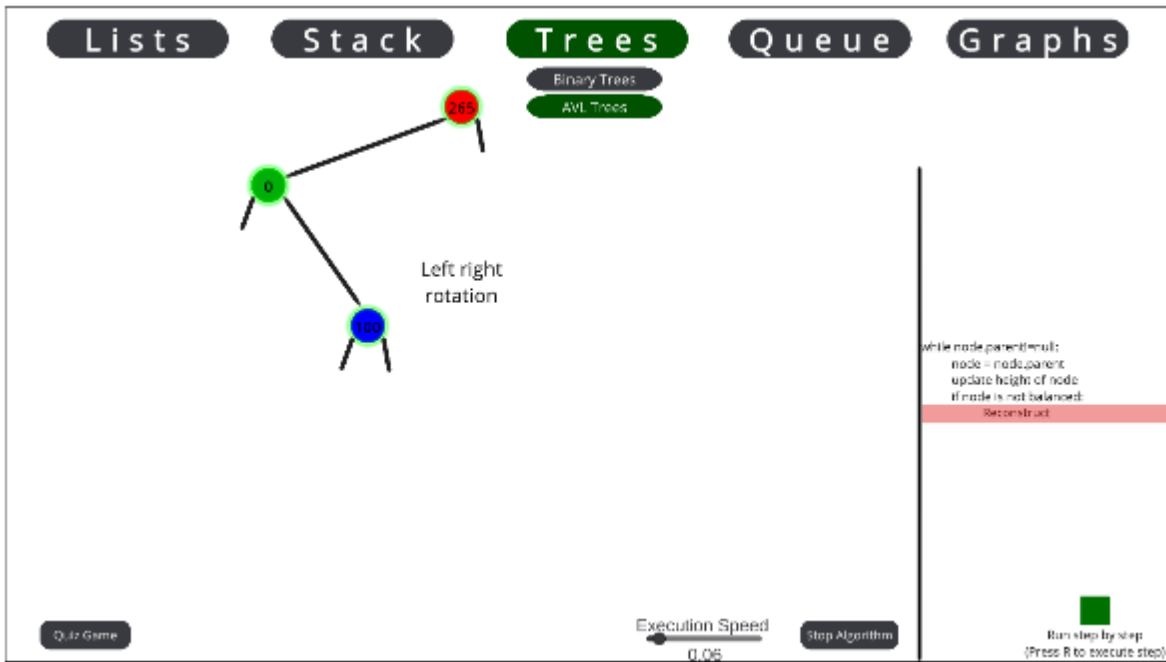


Εικόνα 4.68 Δεξιά περιστροφή

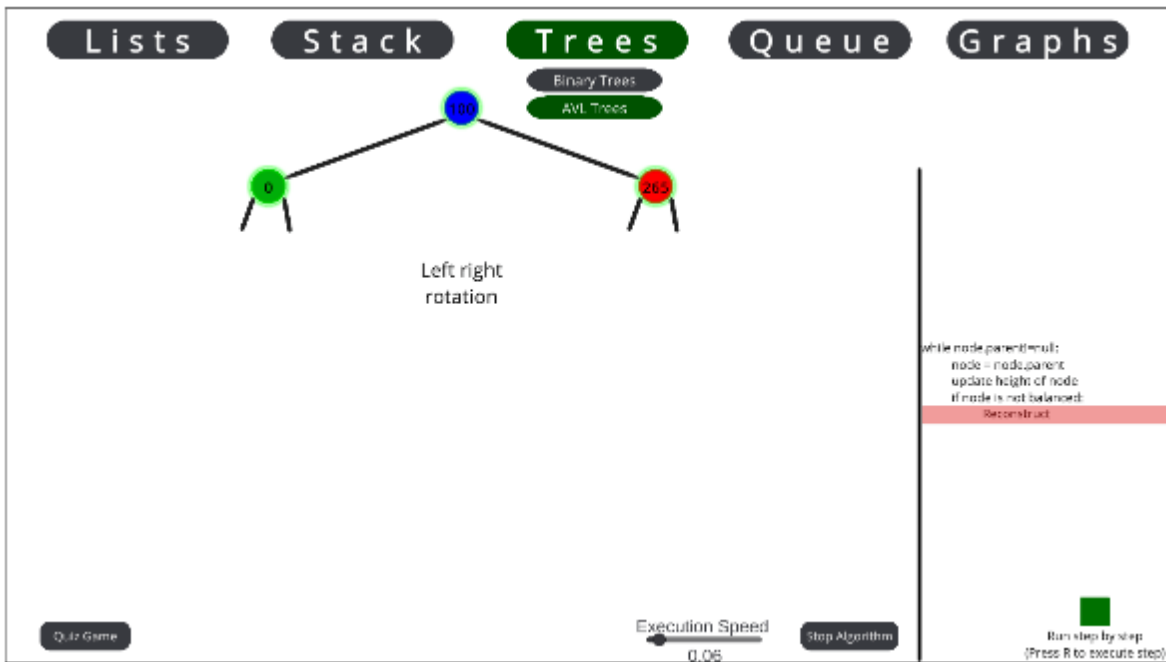
Εάν οι κόμβοι σχηματίζουν δεξιά ή αριστερή γωνία τότε ο νέος κόμβος παραβιάζει την συνθήκη λόγω αύξησης κατά μίας μονάδας του ύψους σε $h+3$ (Εικόνα 4.69), (Εικόνα 4.70), (Εικόνα 4.72), (Εικόνα 4.73). Τότε αρκεί μία αριστερή-δεξιά διπλή περιστροφή (Εικόνα 4.71) ή μία δεξιά-αριστερή διπλή περιστροφή (Εικόνα 4.74).



Εικόνα 4.69 Ένθεση στοιχείου σε δέντρο AVL



Εικόνα 4.70 Εντοπισμός παραβίασης συνθήκης



Εικόνα 4.71 Αριστερή-δεξιά περιστροφή

The screenshot shows an AVL tree simulation. At the top, there are navigation tabs: Lists, Stack, Trees (selected), Queue, and Graphs. Under 'Trees', there are sub-tabs for Binary Trees and AVL Trees. The tree structure is as follows:

- Root node: 50
- Left child of 50: 23
- Right child of 50: 35

 The code editor on the right contains the following code:


```

    if head == null:
        head = new Node
        head.data = data
    return
    Node current = head
    left = false
    Node prev = null
    while current != null:
        if data < current.data:
            prev = current
            current = current.left
            left = true
        else if data > current.data:
            prev = current
            current = current.right
            left = false
    node = new Node
    node.data = data
    if left is true:
        prev.left = node
    else:
        prev.right = node
    RebalanceTree()
    
```

 At the bottom, there is an 'Execution Speed' slider set to 0.05 and a 'Step Algorithm' button. A green square icon indicates 'Run step by step (Press R to execute step)'.

Εικόνα 4.72 Ένθεση στοιχείου σε δέντρο AVL

This screenshot shows the same AVL tree simulation after a rotation. The tree structure remains:

- Root node: 50
- Left child of 50: 23
- Right child of 50: 35

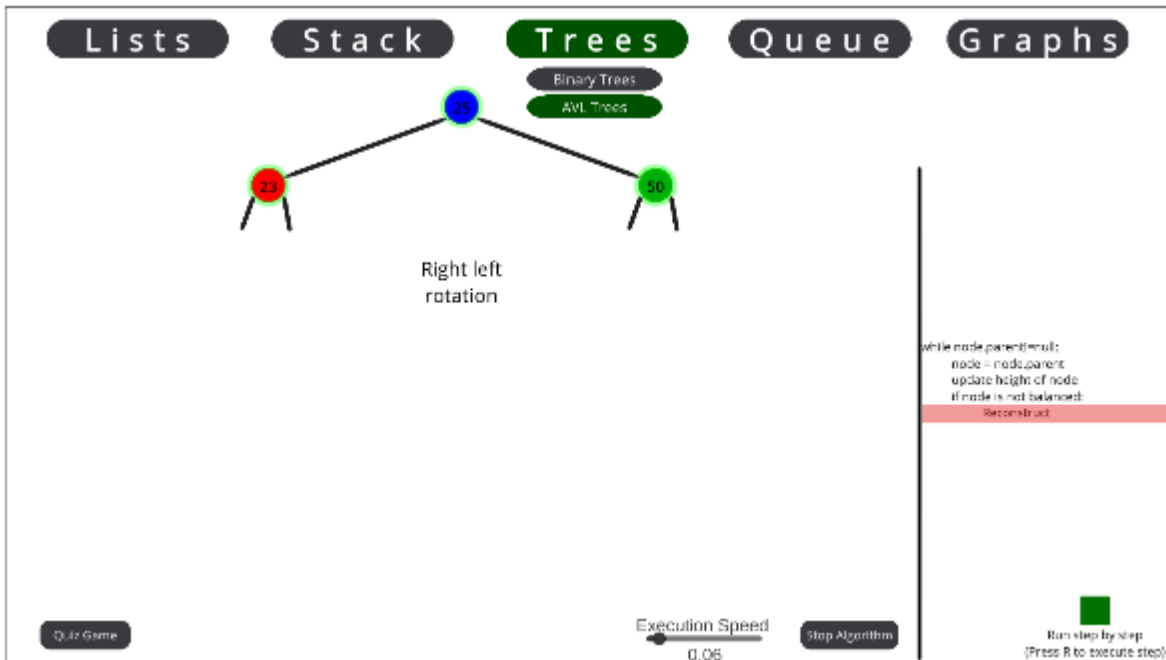
 However, node 23 is now highlighted in red, and node 35 is highlighted in blue. The text 'Right left rotation' is displayed in the center of the tree. The code editor on the right shows the following code:


```

    while node.parent != null:
        node = node.parent
        update height of node
        if node is not balanced:
            Reconstruct
    
```

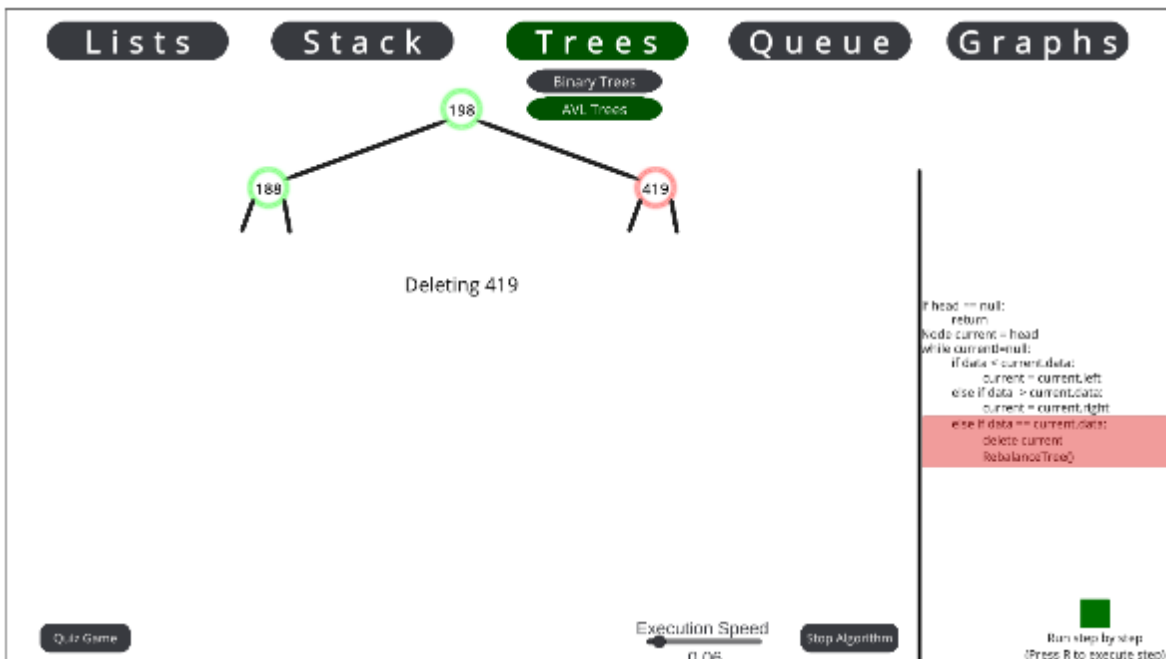
 The 'Reconstruct' line is highlighted in red. At the bottom, there is an 'Execution Speed' slider set to 0.05 and a 'Step Algorithm' button. A green square icon indicates 'Run step by step (Press R to execute step)'.

Εικόνα 4.73 Εντοπισμός παραβίασης συνθήκης

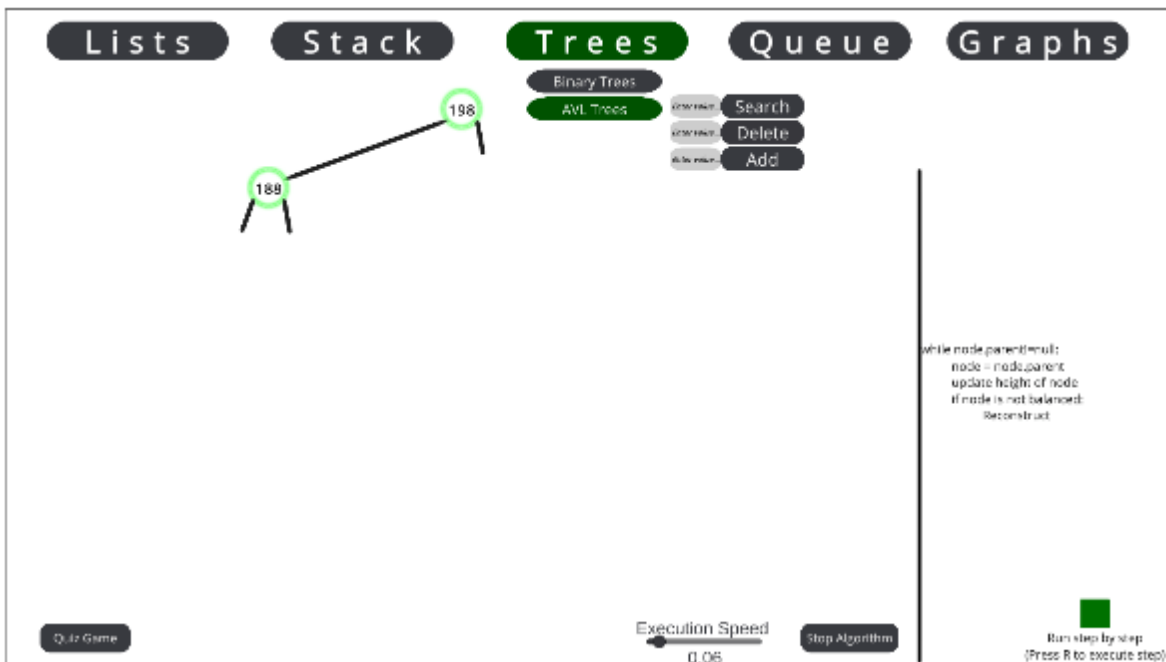


Εικόνα 4.74 Δεξιά-Αριστερή περιστροφή

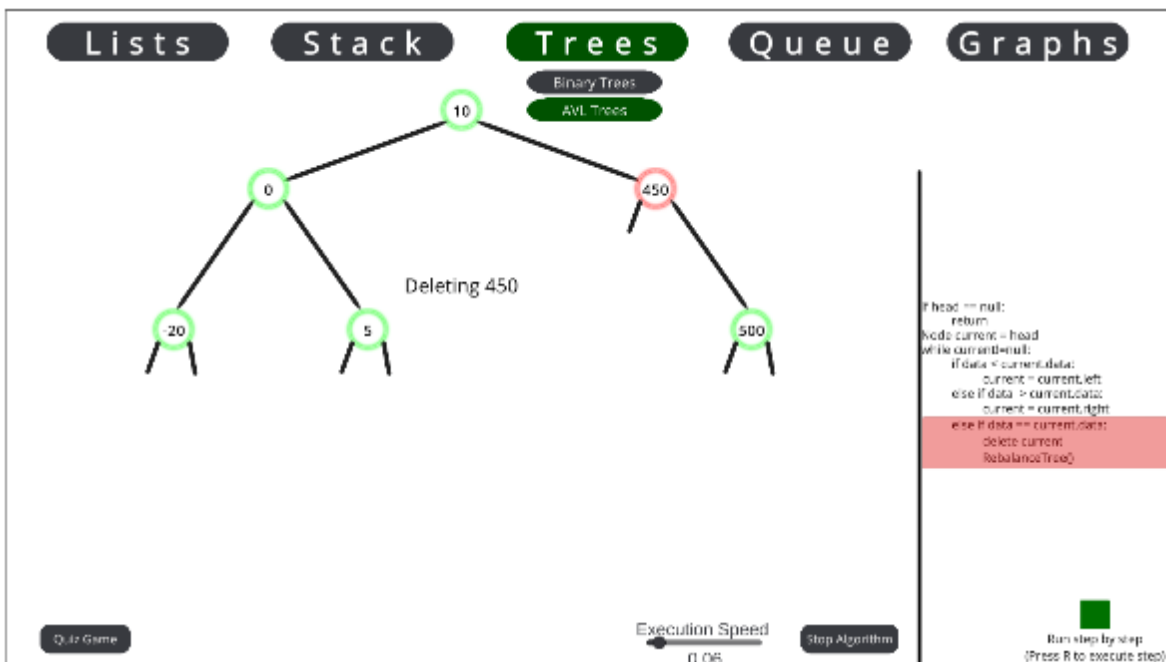
Κατά την διαδικασία της διαγραφής εφαρμόζεται μία διαδικασία αναζήτησης του κόμβου προς διαγραφή (Εικόνα 4.75), (Εικόνα 4.76). Εάν το δέντρο δεν διαθέτει τον κόμβο προς διαγραφή τότε ο αλγόριθμος σταματά. Εάν ο κόμβος προς διαγραφή διαθέτει τουλάχιστον ένα κενό φύλλο, καταργείται ο κόμβος προς διαγραφή και το μη κενό παιδί του συνδέεται με τον πατέρα του (Εικόνα 4.77), (Εικόνα 4.78).



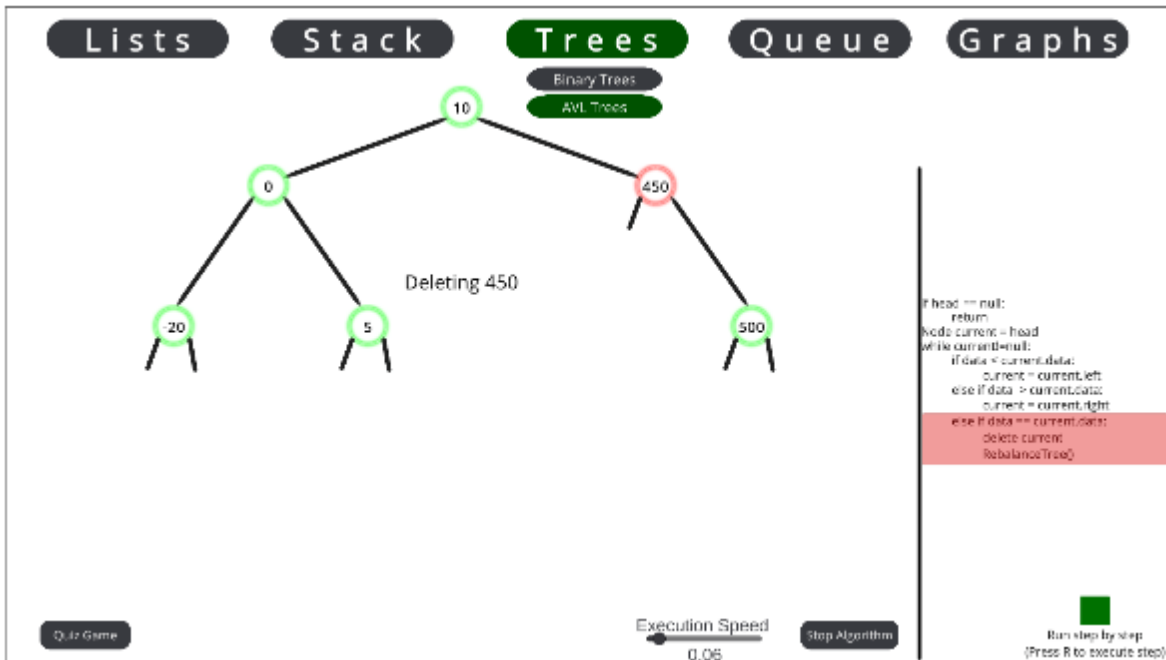
Εικόνα 4.75 Διαγραφή κόμβου χωρίς παιδιά



Εικόνα 4.76 Διαγραφή κόμβου

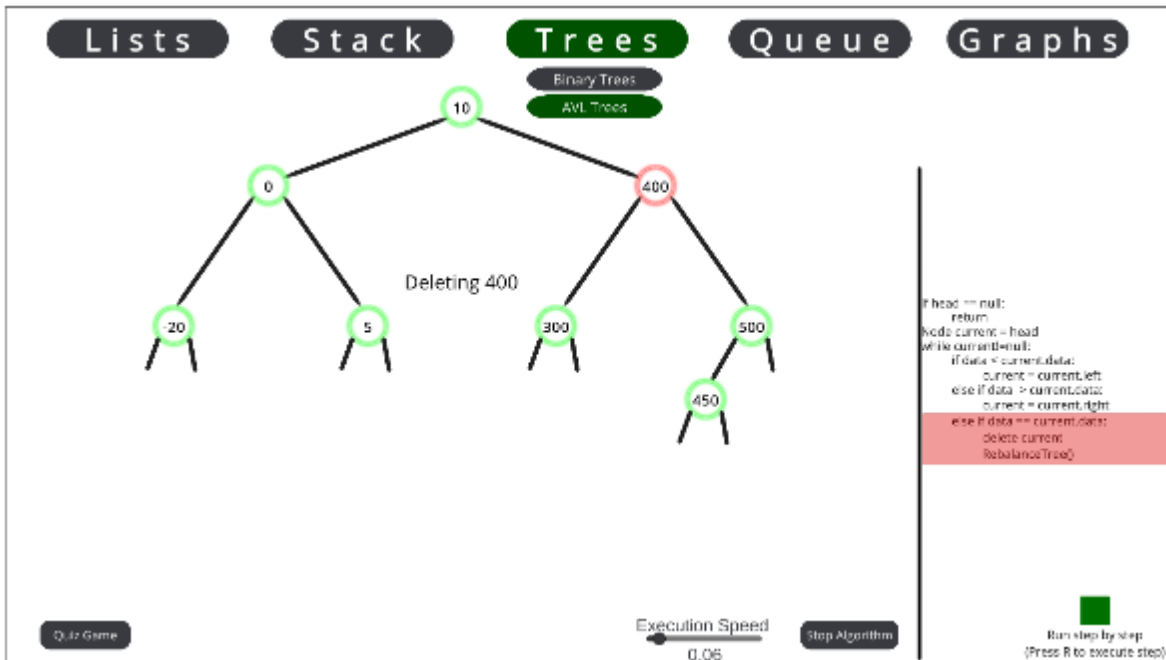


Εικόνα 4.77 Διαγραφή κόμβου με ένα παιδί

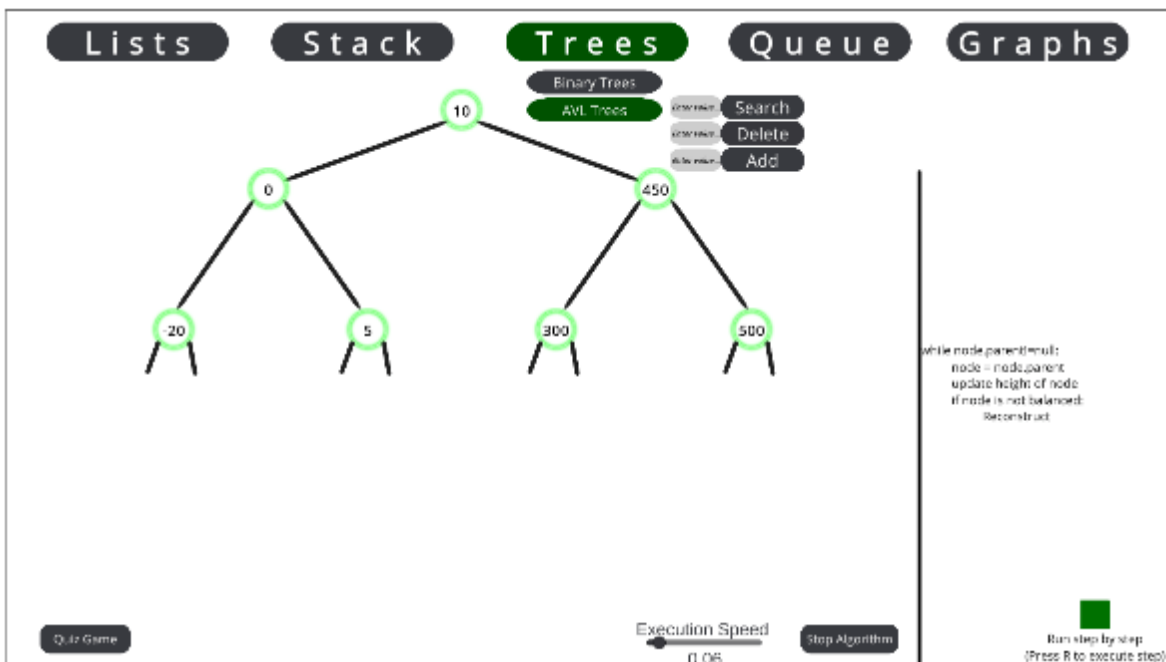


Εικόνα 4.78 Αντικατάσταση κόμβου με το παιδί

Διαφορετικά, εάν ο κόμβος προς διαγραφή διαθέτει μη κενό δεξιό γιο, ανταλλάσσεται το στοιχείο του κόμβου με αυτό του βαθύτερου, αριστερότερου μη κενού απογόνου του δεξιού υποδέντρου (Εικόνα 4.79). Με αυτόν τον τρόπο οι κόμβοι άλλαξαν θέσεις και στην συνέχεια ο κόμβος διαγράφεται (Εικόνα 4.80). Κατά την διαδικασία διαγραφής εφαρμόζονται πάλι όπου χρειάζεται πάλι οι επαναζυγιστικές πράξεις που είναι ίδιες με αυτές που αναλύθηκαν πιο πάνω.



Εικόνα 4.79 Διαγραφή κόμβου με παιδιά



Εικόνα 4.80 Αντικατάσταση του κόμβου με το αριστερότερο παιδί του δεξιού υποδέντρου

4.4.4 Ουρές

Πατώντας την επιλογή ουρές (Queue) της εφαρμογής αναγράφονται δύο διαφορετικά είδη ουρών (Εικόνα 4.81). Ουρές με πίνακα (Queues Using Array) (Εικόνα 4.82) και ουρές

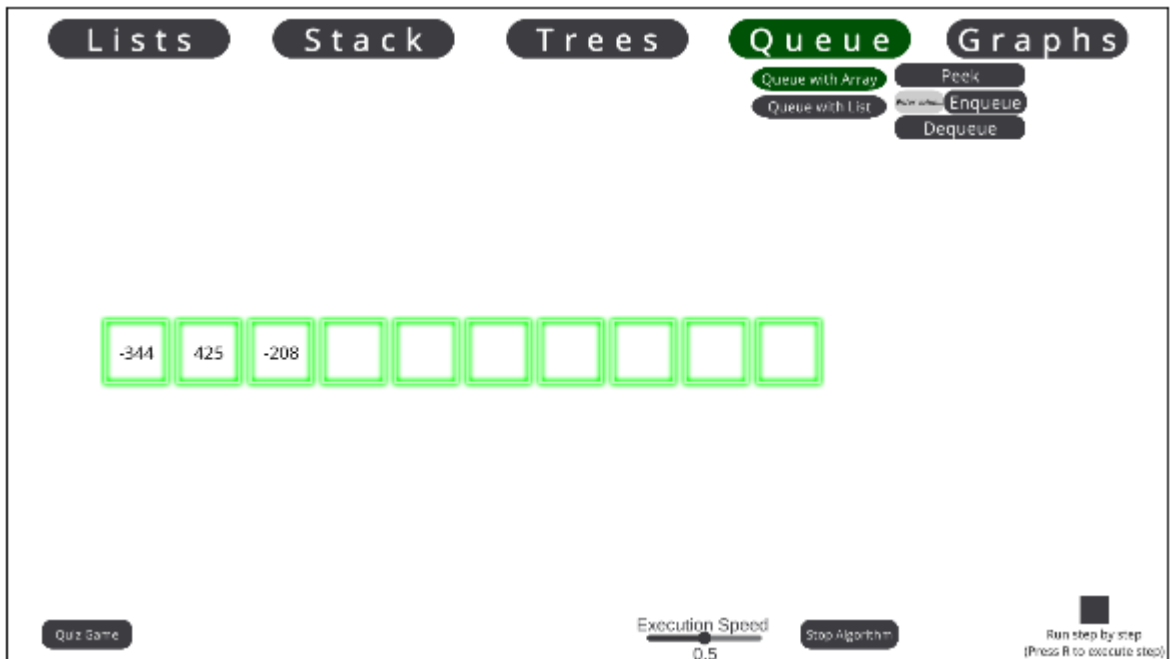
με λίστες (Queues using Lists) (Εικόνα 4.83). Επιλέγοντας οποιοδήποτε τύπο ουράς εμφανίζονται οι τρεις αλγόριθμοι, Peek, Enqueue και Dequeue.



Εικόνα 4.81 Τύποι ουρών

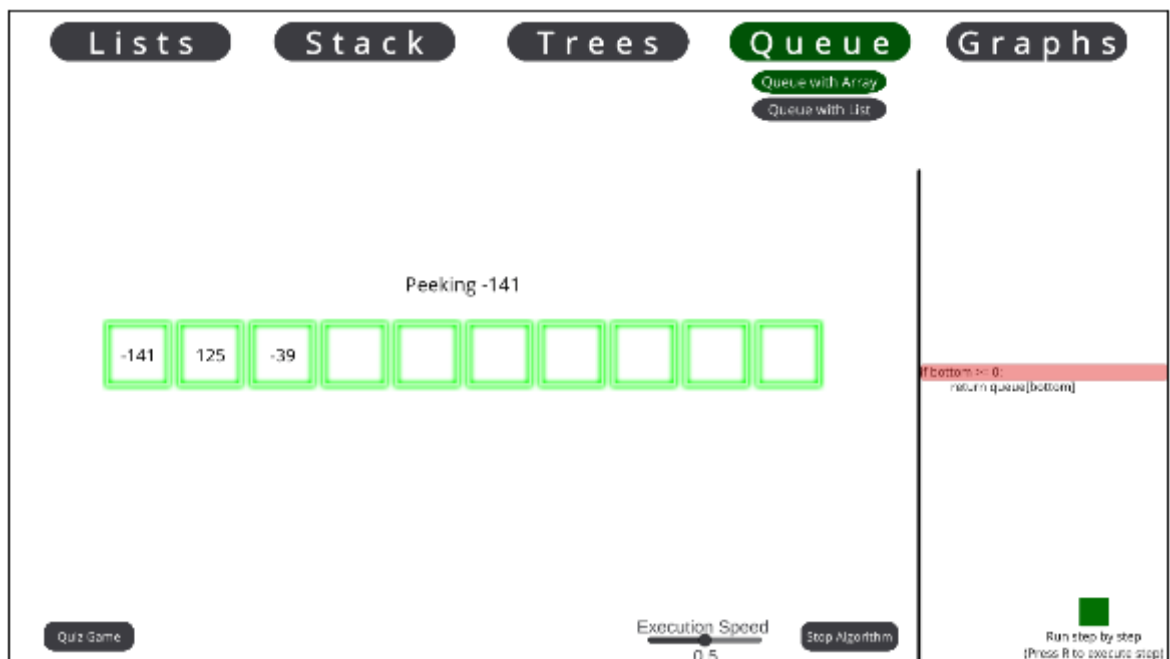


Εικόνα 4.82 Αλγόριθμοι Ουρών με πίνακες

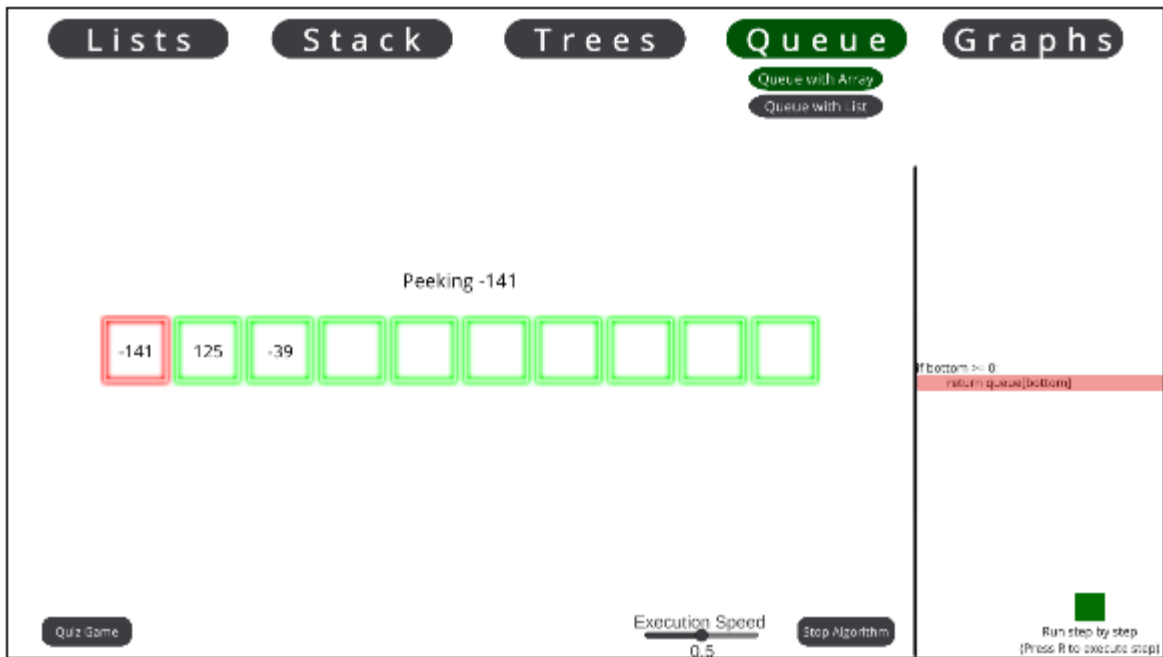


Εικόνα 4.83 Αλγόριθμοι Στοιβών με λίστες

Στις ουρές με πίνακες κατά τον αλγόριθμο Peek επιστρέφεται η τιμή η οποία πρόκειται να γίνει dequeue (Εικόνα 4.84), (Εικόνα 4.85).

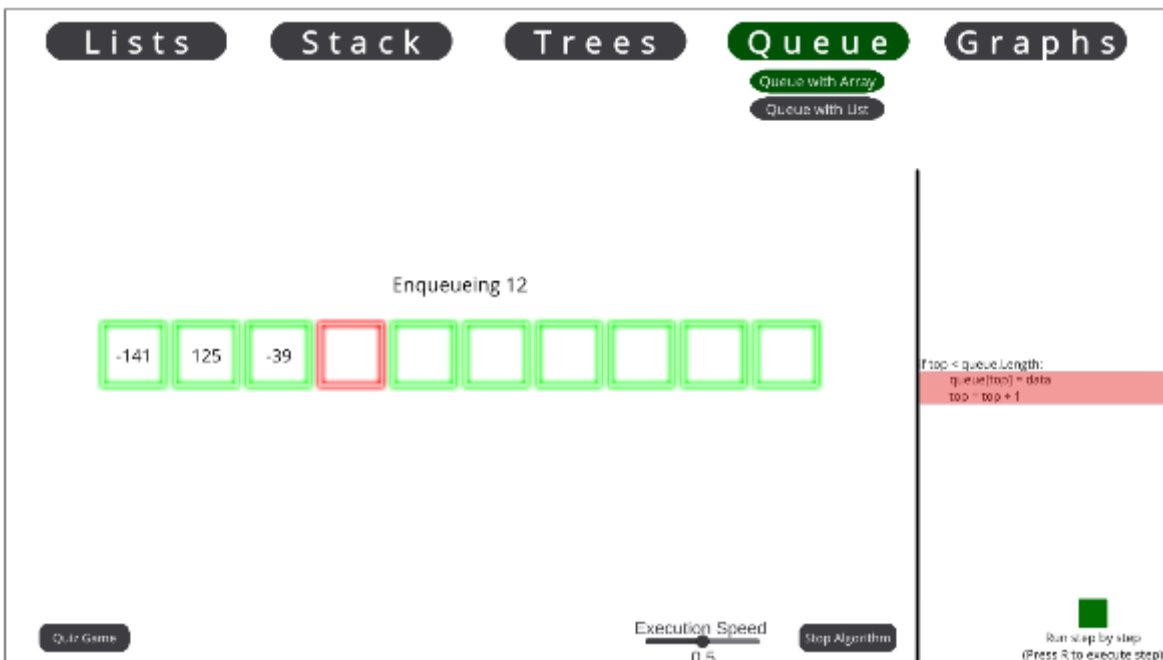


Εικόνα 4.84 Αλγόριθμος peek σε ουρές με πίνακες

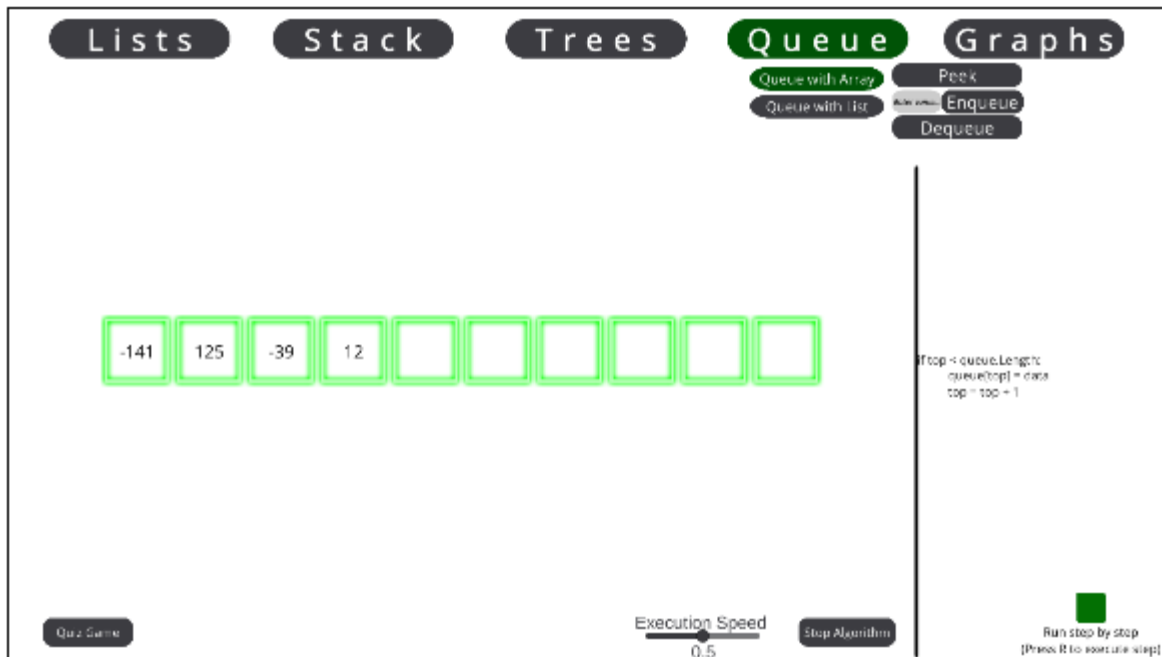


Εικόνα 4.85 Peek του -141 σε ουρά με πίνακες

Κατά τον αλγόριθμο enqueue βάζει στην ουρά με πίνακες την τιμή που θα πληκτρολογήσει ο χρήστης (Εικόνα 4.86), (Εικόνα 4.87).

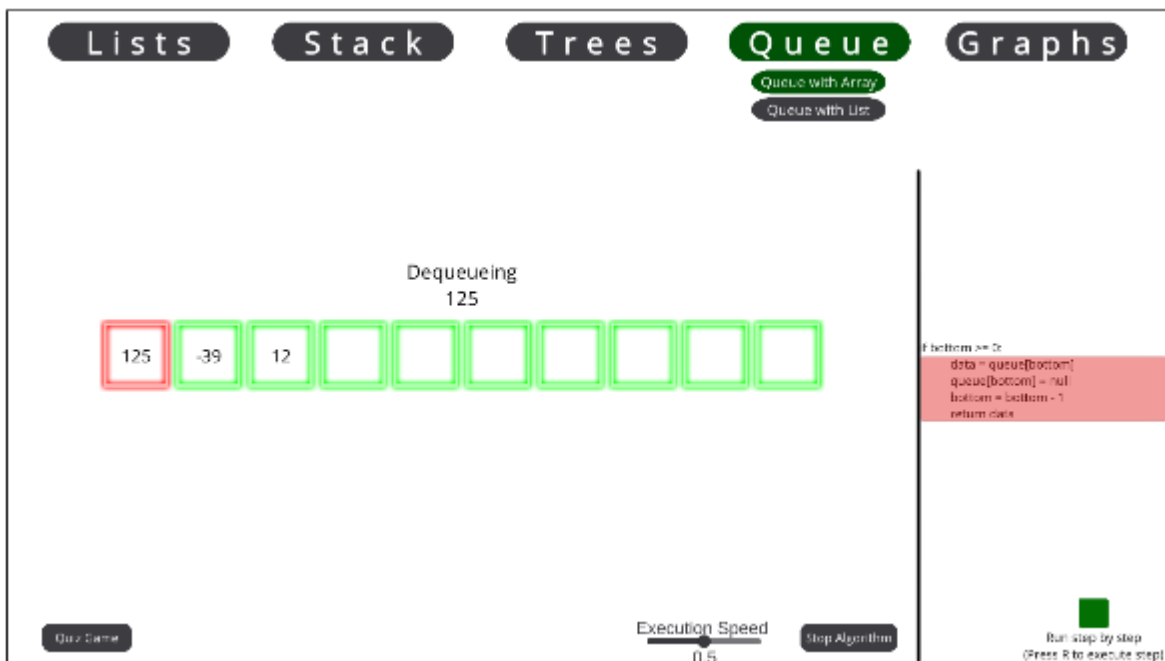


Εικόνα 4.86 Αλγόριθμος Enqueue σε ουρά με πίνακες

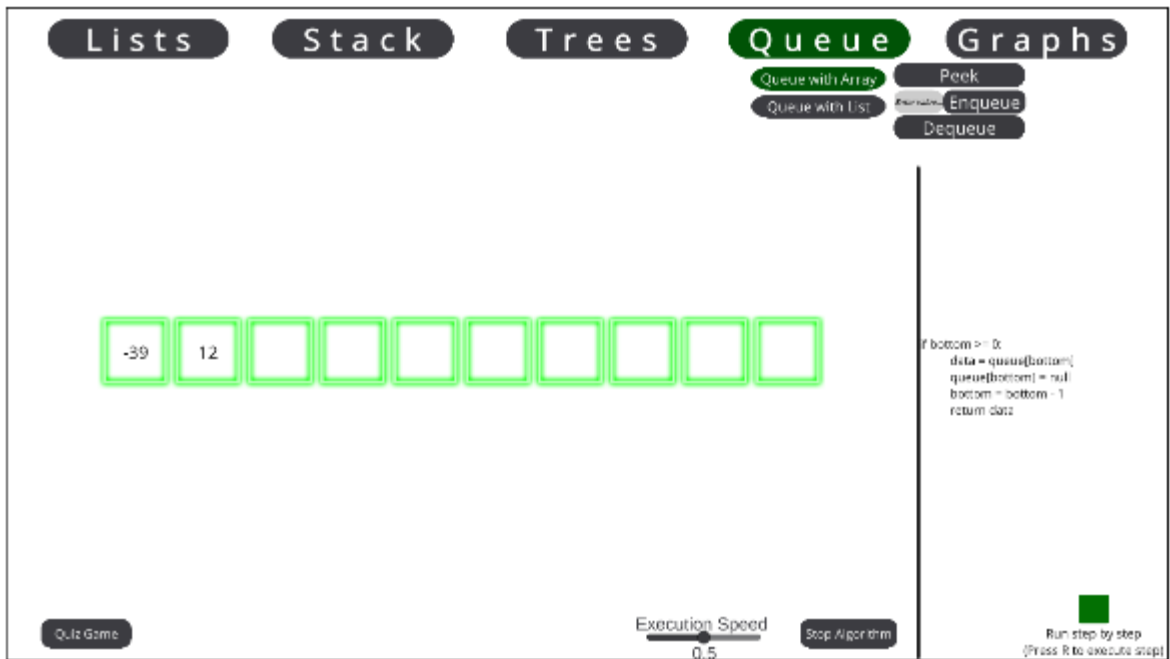


Εικόνα 4.87 Ενημεύω 12 σε ουρά με πίνακες

Κατά τον αλγόριθμο dequeue βγάζει το πρώτο στοιχείο από την ουρά με πίνακες (Εικόνα 4.88), (Εικόνα 4.89).

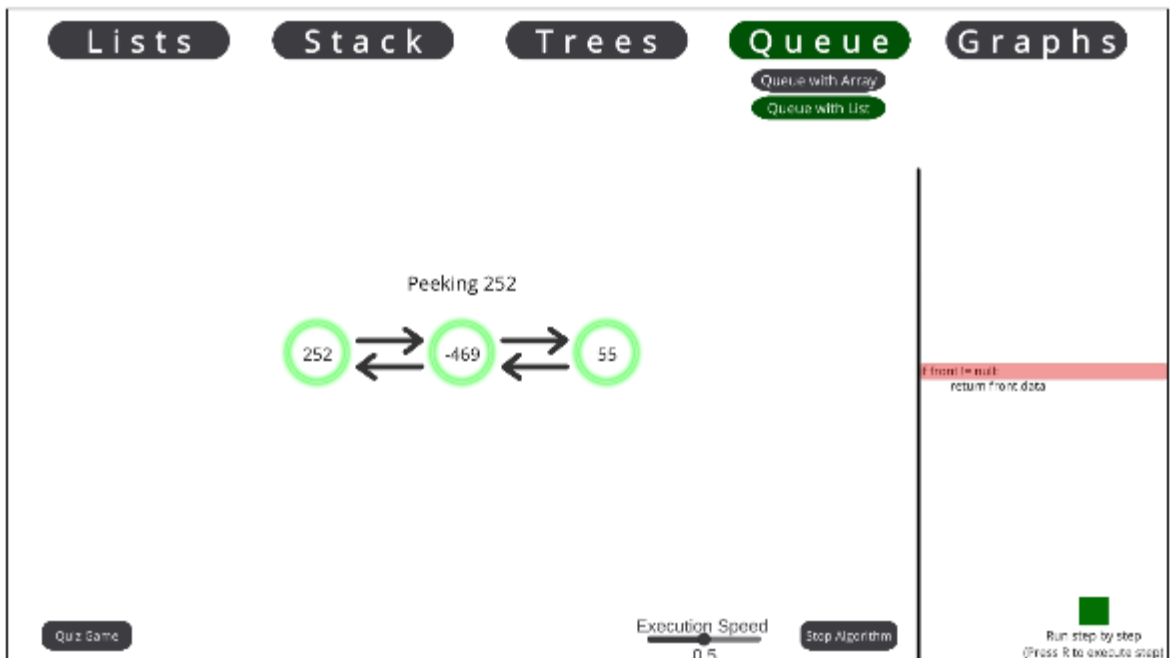


Εικόνα 4.88 Αλγόριθμος Dequeue σε ουρά με πίνακες

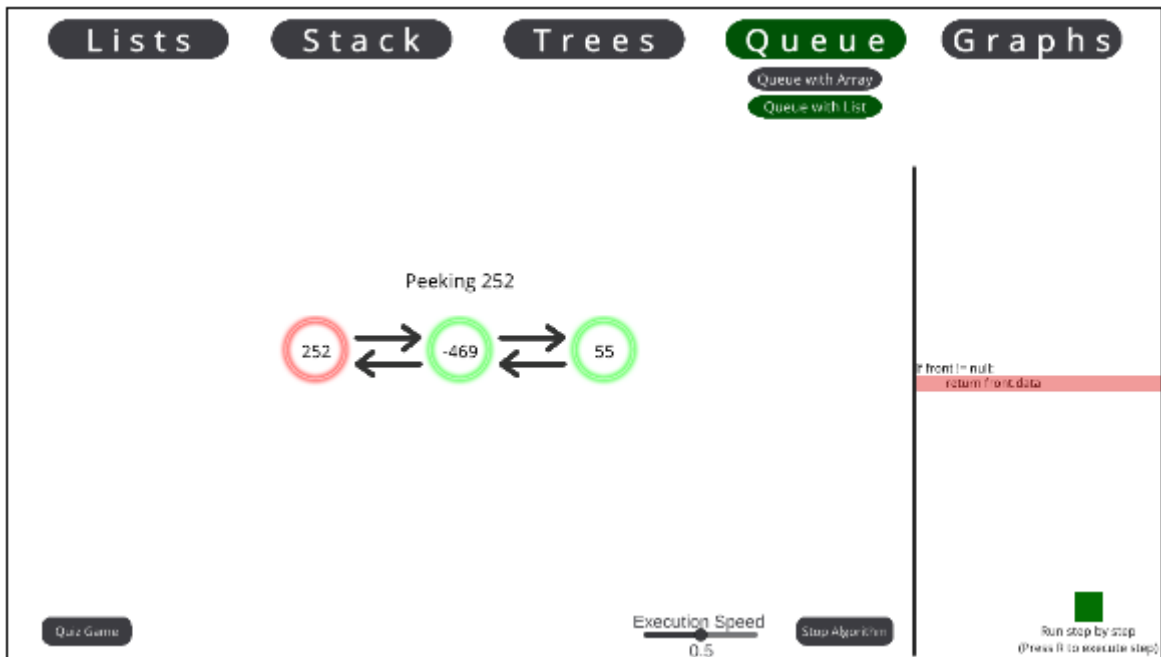


Εικόνα 4.89 Dequeue 125 σε ουρά με πίνακες

Στις ουρές με λίστες κατά τον αλγόριθμο Peek επιστρέφεται η τιμή η οποία πρόκειται να γίνει dequeue (Εικόνα 4.90), (Εικόνα 4.91).

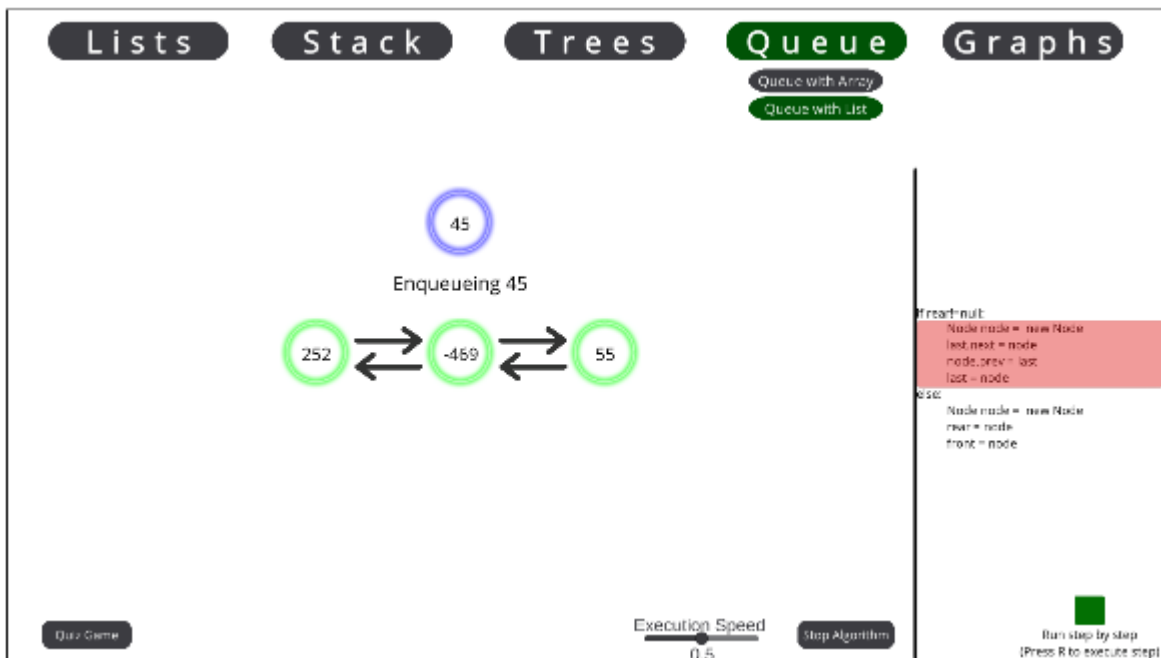


Εικόνα 4.90 Αλγόριθμος Peek σε ουρά με λίστες

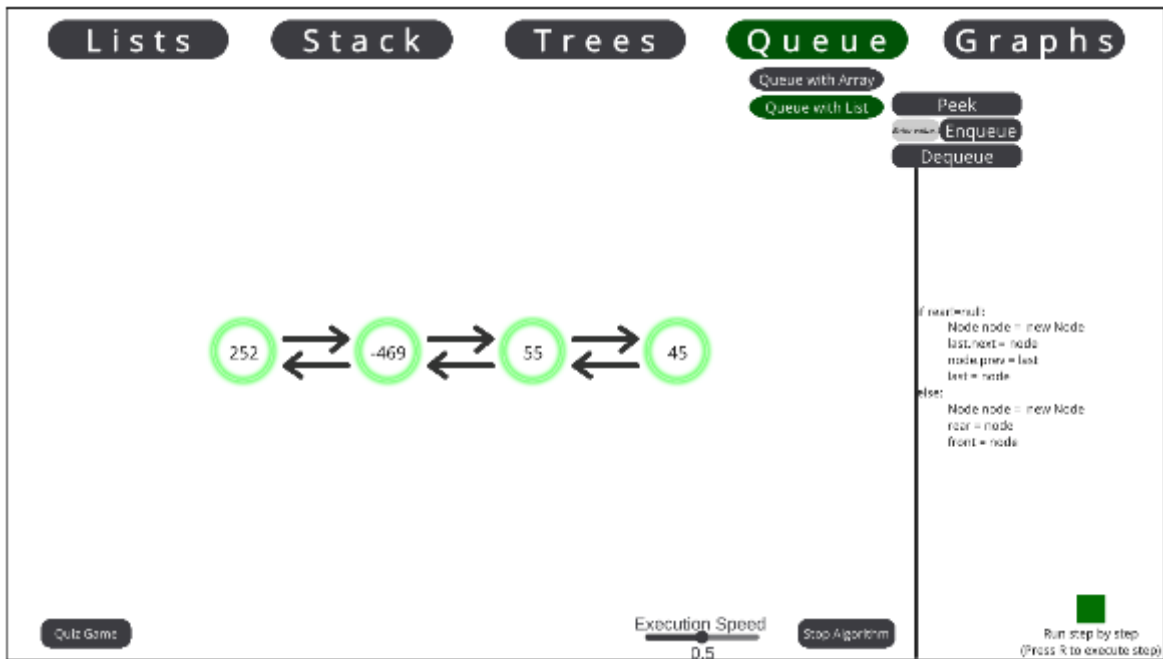


Εικόνα 4.91 Peek 252 σε ουρά με λίστες

Κατά τον αλγόριθμο enqueue βάζει στην ουρά με λίστες την τιμή που θα πληκτρολογήσει ο χρήστης (Εικόνα 4.92), (Εικόνα 4.93).

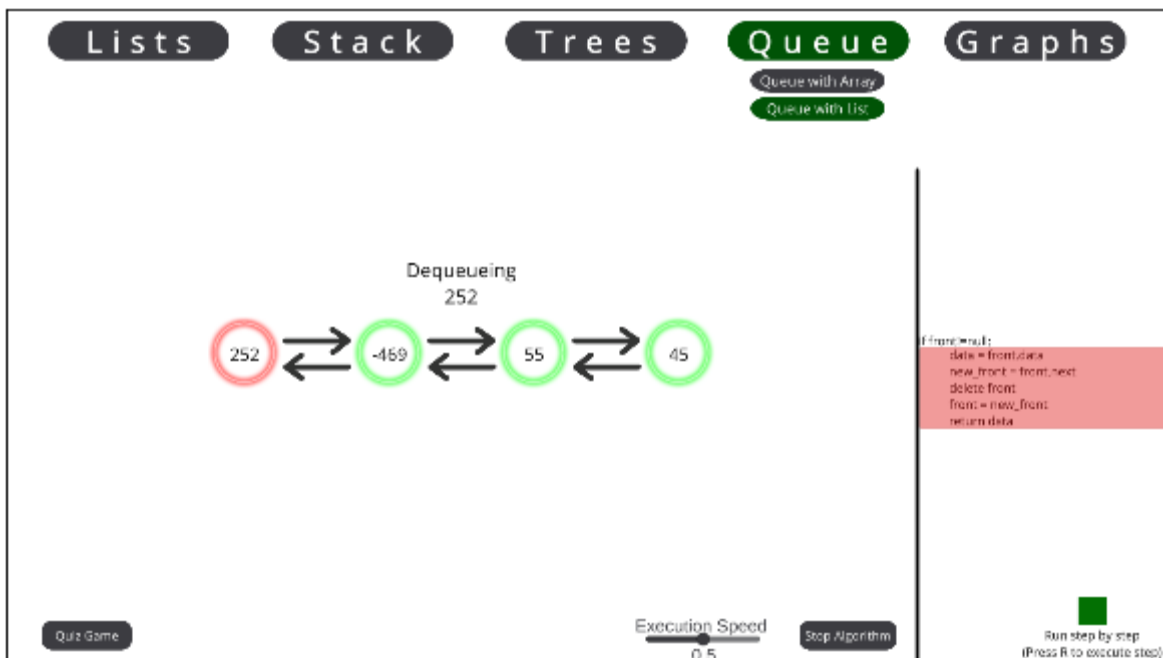


Εικόνα 4.92 Αλγόριθμος Enqueue σε ουρά με λίστες



Εικόνα 4.93 Ενημευση 45 σε ουρά με λίστες

Κατά τον αλγόριθμο dequeue βγάζει το πρώτο στοιχείο από την ουρά με λίστες (Εικόνα 4.94), (Εικόνα 4.95).



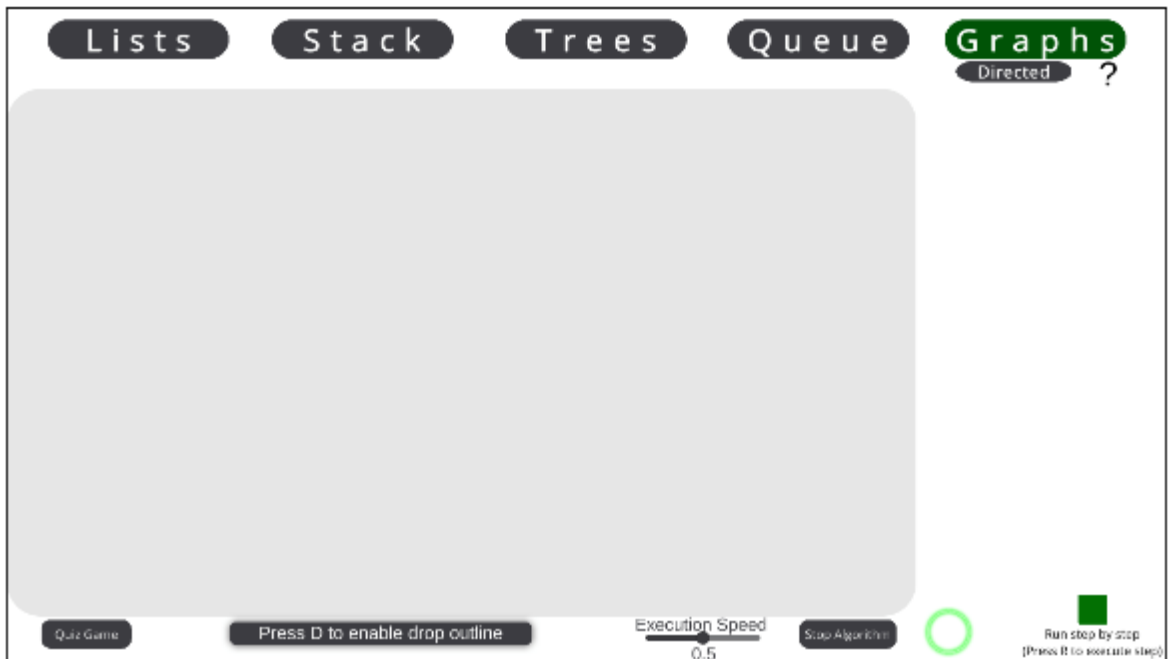
Εικόνα 4.94 Αλγόριθμος Dequeue σε ουρά με λίστες



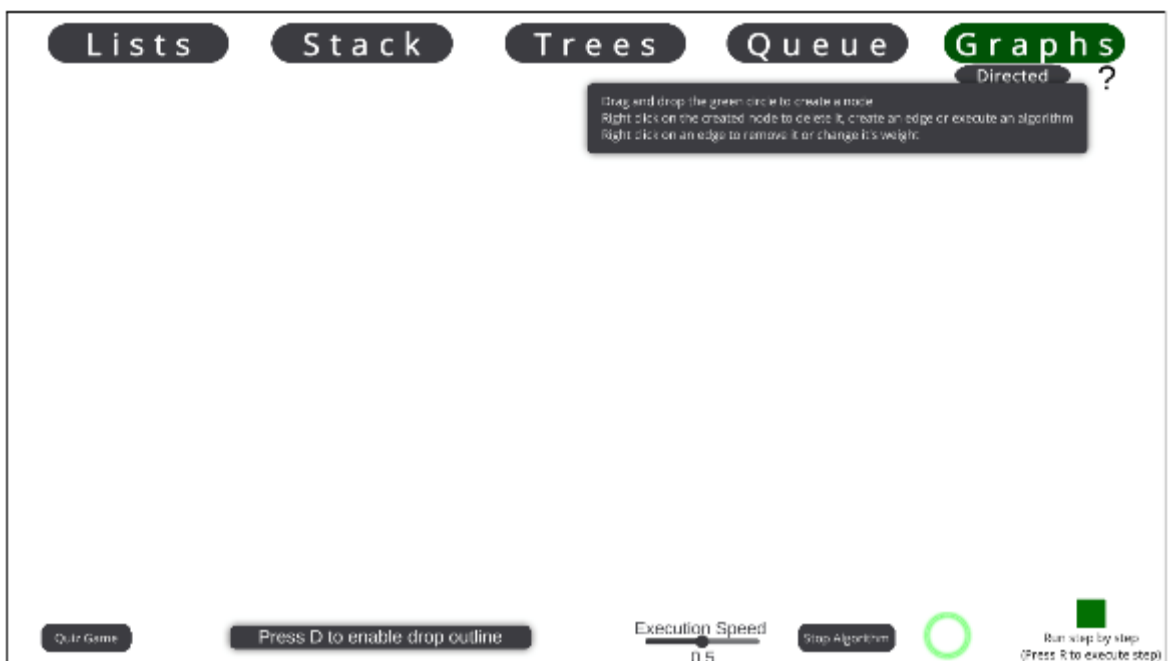
Εικόνα 4.95 Dequeue 252 σε ουρά με λίστες

4.4.5 Γράφοι

Εάν ο χρήστης επιλέξει την επιλογή Γράφους (Graphs), εμφανίζεται ο κόμβος στο κάτω μέρος της εφαρμογής όπου ο χρήστης πρέπει να σύρει κάθε κόμβο στη σκιασμένη περιοχή την οποία μπορεί να δει με το κουμπί d (Press D to enable drop outline) και ο κάθε κόμβος παίρνει τυχαίες τιμές (Εικόνα 4.96). Επίσης επιλέγοντας τους Γράφους (Graphs), εμφανίζεται η επιλογή κατευθυνόμενοι Γράφοι (directed) και δίπλα ένα ερωτηματικό στο οποίο εμφανίζονται πληροφορίες σχετικά με τον τρόπο λειτουργίας των γράφων (Εικόνα 4.97).



Εικόνα 4.96 Γράφοι



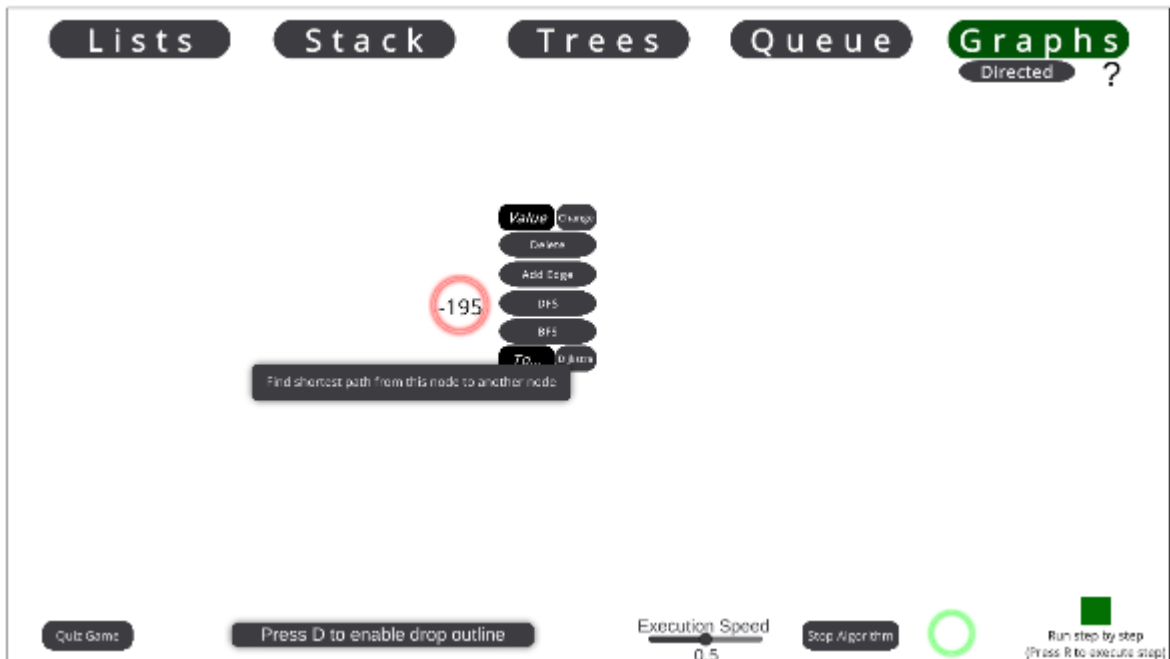
Εικόνα 4.97 Πληροφορίες γράφων

Σε κάθε κόμβο πατώντας δεξί κλικ ο χρήστης έχει την δυνατότητα να επιλέξει κάποια λειτουργία πάνω στους Γράφους (Εικόνα 4.98). Οι επιλογές είναι οι εξής:

- Αλλαγή τιμής του κάθε κόμβου (Change Value)
- Διαγραφή κόμβου (Delete)

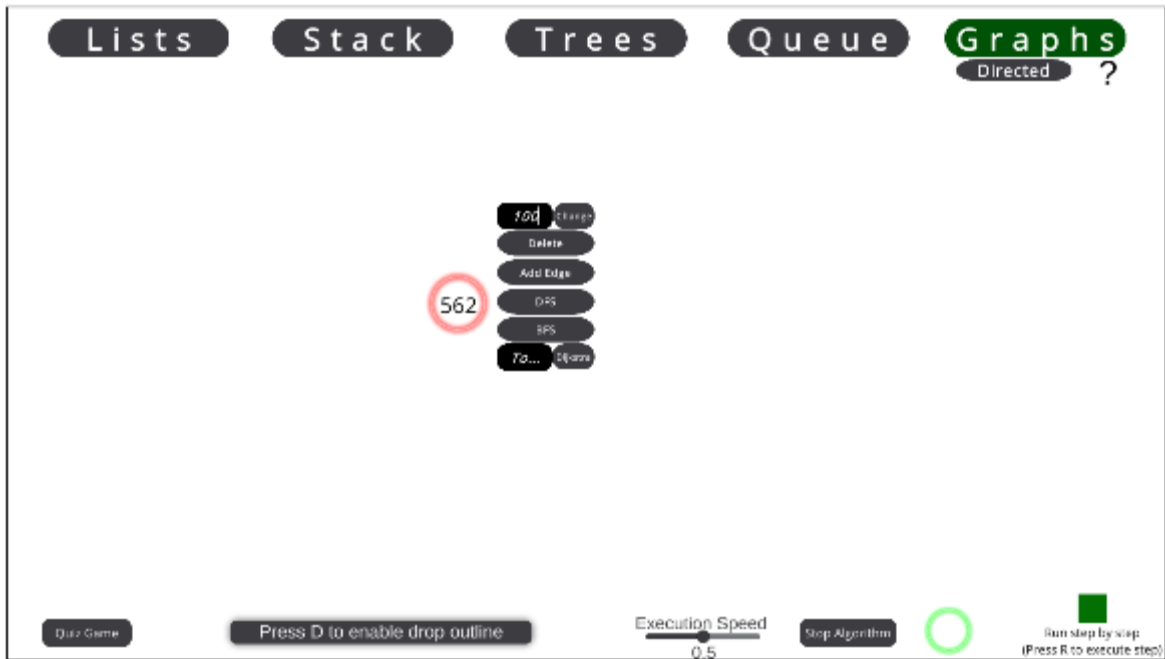
- Προσθήκη ακμής (Add Edge)
- Αναζήτηση κατά βάθος (DFS)
- Αναζήτηση κατά πλάτος (BFS)
- Ο αλγόριθμος Dijkstra

Επίσης πάνω από κάθε επιλογή εμφανίζονται πληροφορίες προς διευκόλυνση του χρήστη.

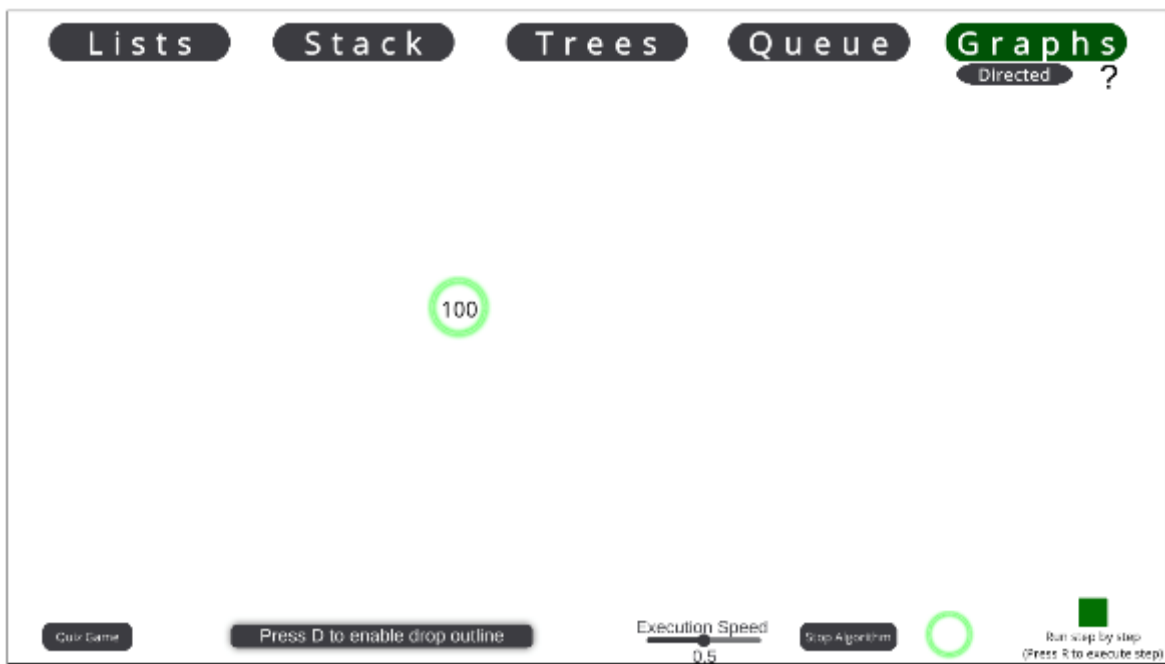


Εικόνα 4.98 Επιλογές Γράφων

Κατά την επιλογή αλλαγή τιμής (Change Value) ο χρήστης πληκτρολογεί την τιμή που επιθυμεί να έχει ο κόμβος (Εικόνα 4.99), (Εικόνα 4.100).



Εικόνα 4.99 Αλλαγή τιμής σε κόμβο

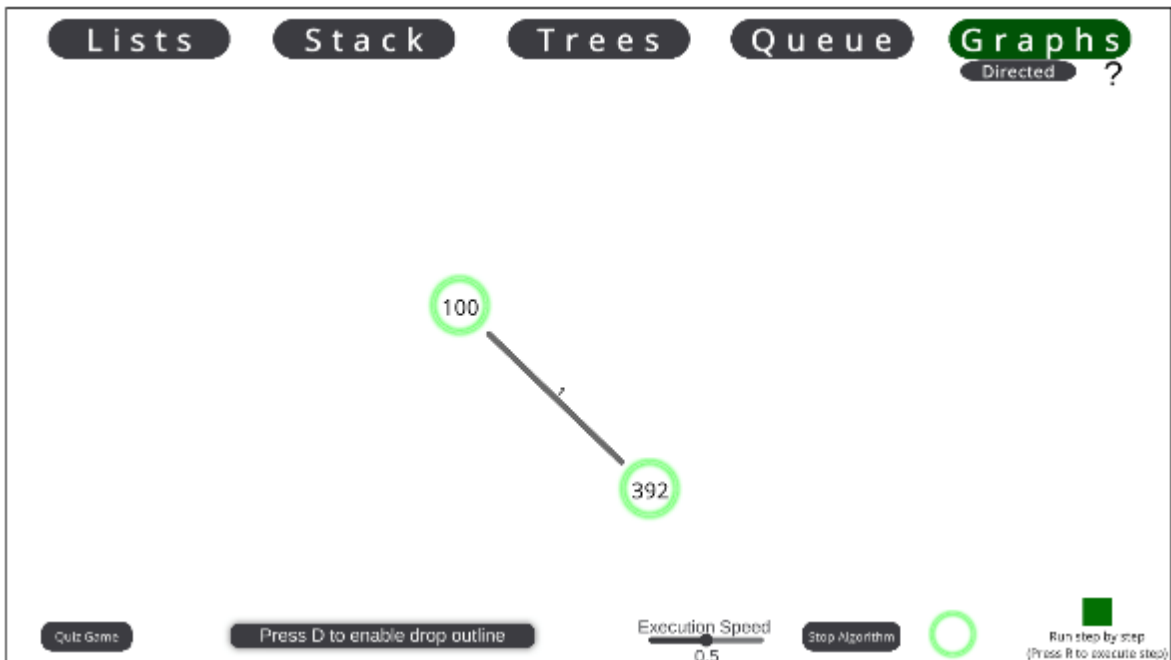


Εικόνα 4.100 Αλλαγή της τιμής του κόμβου σε 100

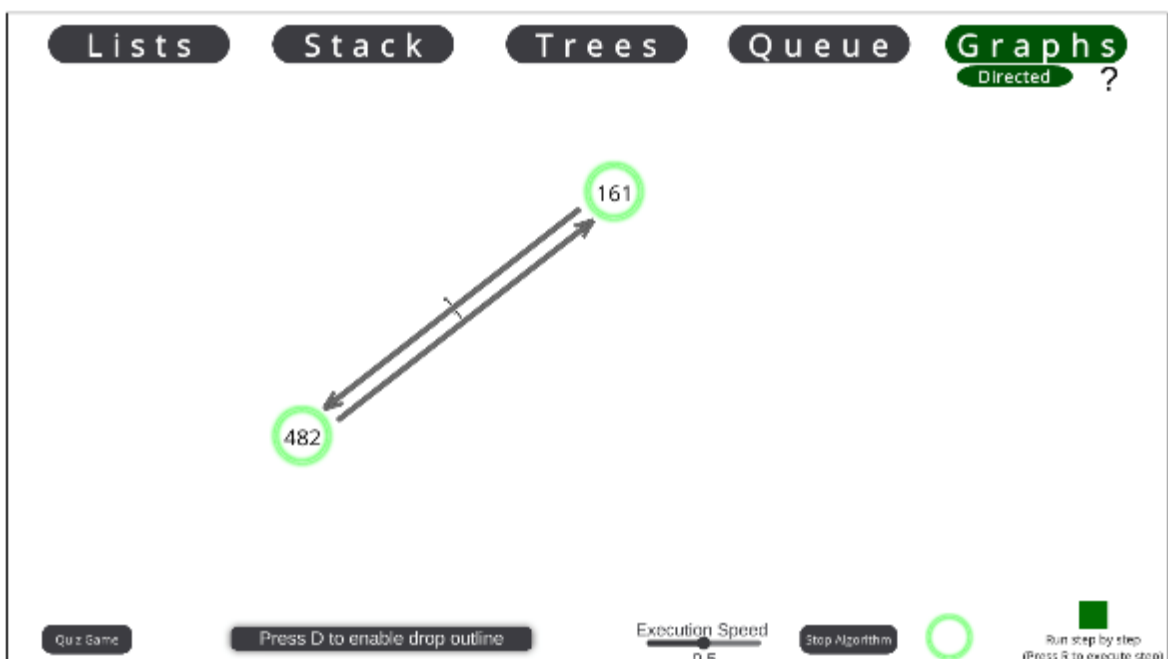
Κατά την επιλογή της διαγραφή ο χρήστης διαγράφει τον κόμβο που επιθυμεί. Κατά την διαγραφή κόμβου διαγράφονται και οι αντίστοιχες ακμές.

Στην επιλογή προσθήκη επιλογής ο χρήστης επιλέγει με κλικ τον αντίστοιχο κόμβο που επιθυμεί να συνδέσει (Εικόνα 4.101), (Εικόνα 4.102). Επίσης σε κάθε ακμή εμφανίζονται με δεξί κλικ οι εξής επιλογές:

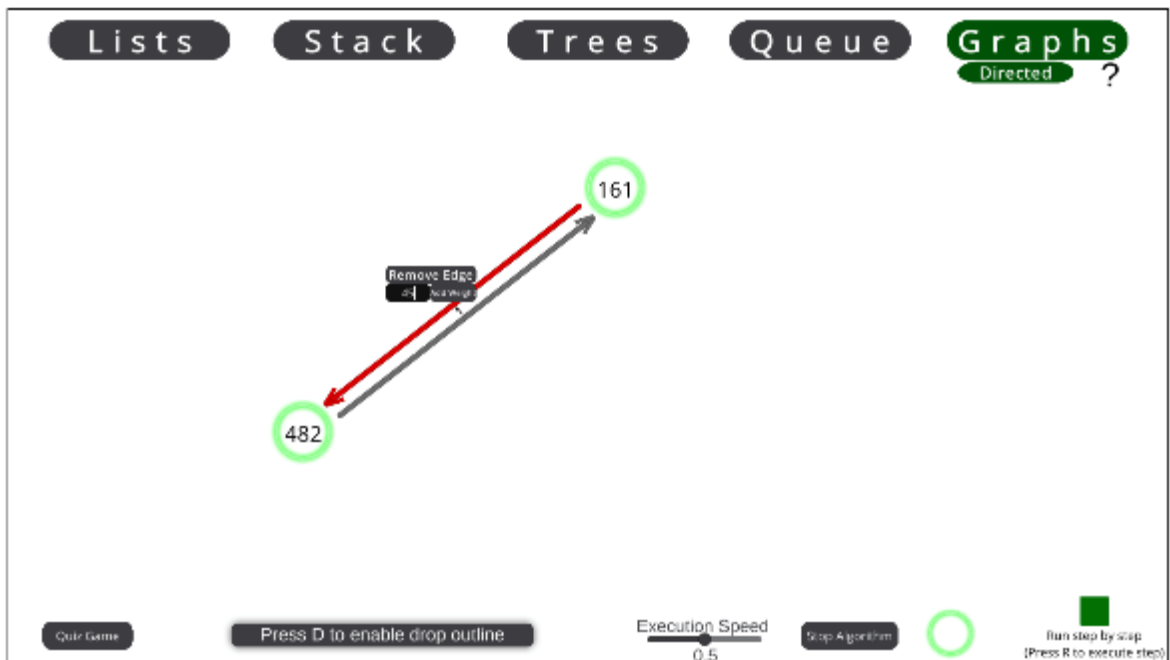
- Διαγραφή ακμής (Remove Edge), όπου διαγράφεται η αντίστοιχη ακμή
- Προσθήκη βάρους ακμής (Add weight) (Εικόνα 4.103), (Εικόνα 4.104)



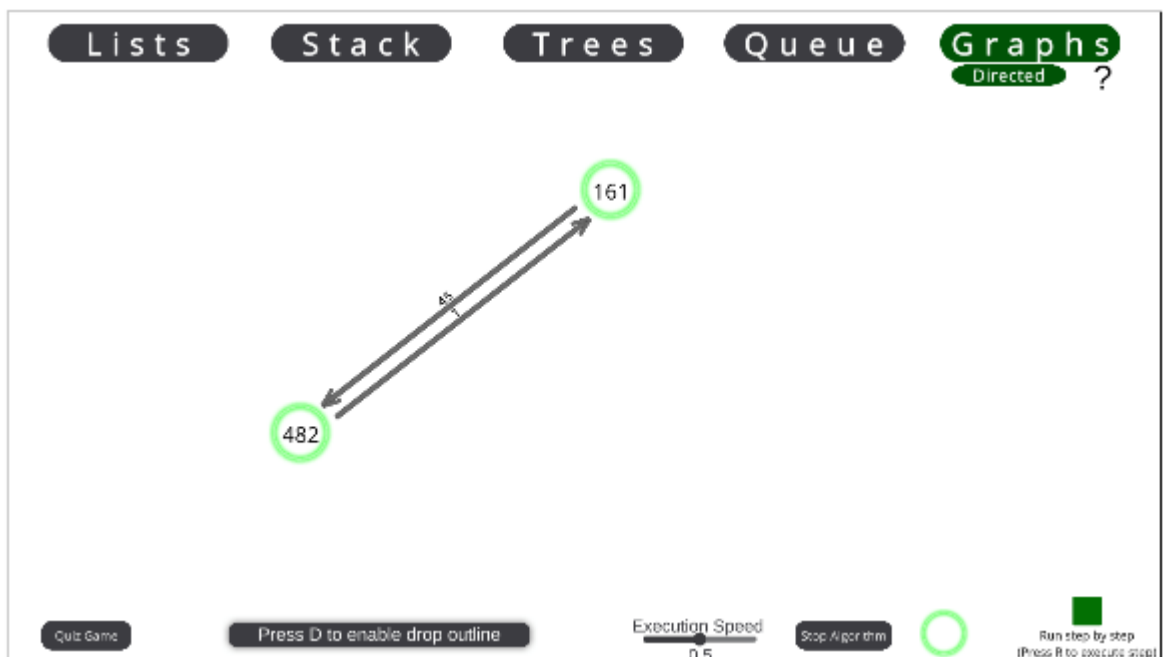
Εικόνα 4.101 Προσθήκη ακμής



Εικόνα 4.102 Προσθήκη ακμής σε κατευθυνόμενο γράφο



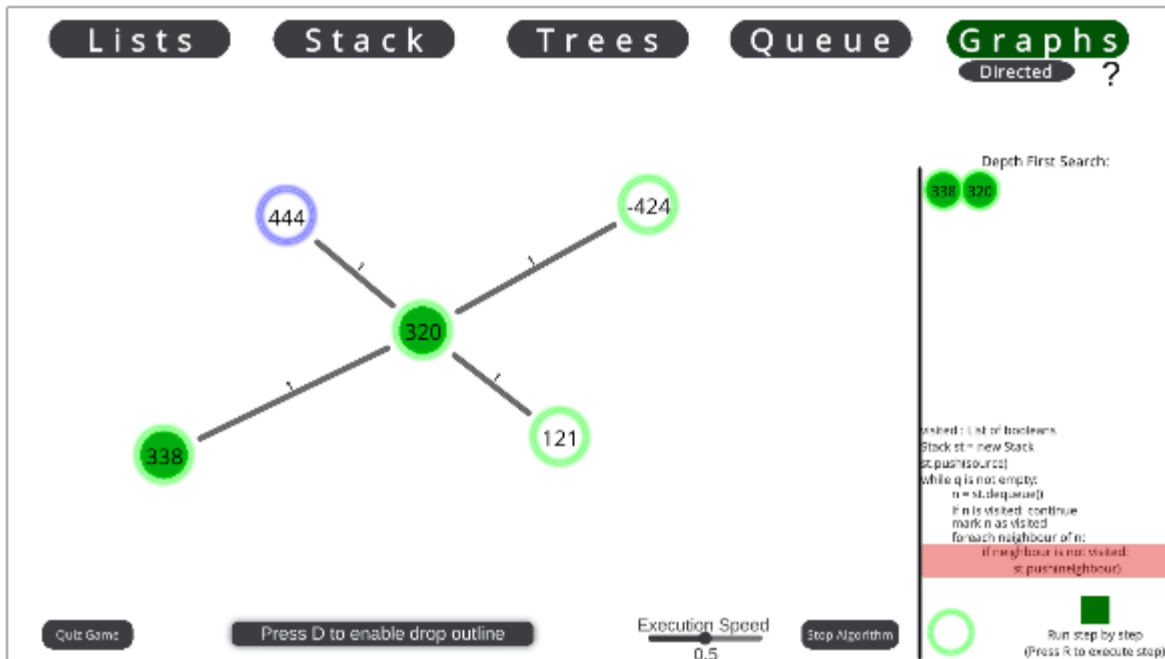
Εικόνα 4.103 Προσθήκη βάρους σε ακμή



Εικόνα 4.104 Προσθήκη βάρους σε ακμή ίσο με 45

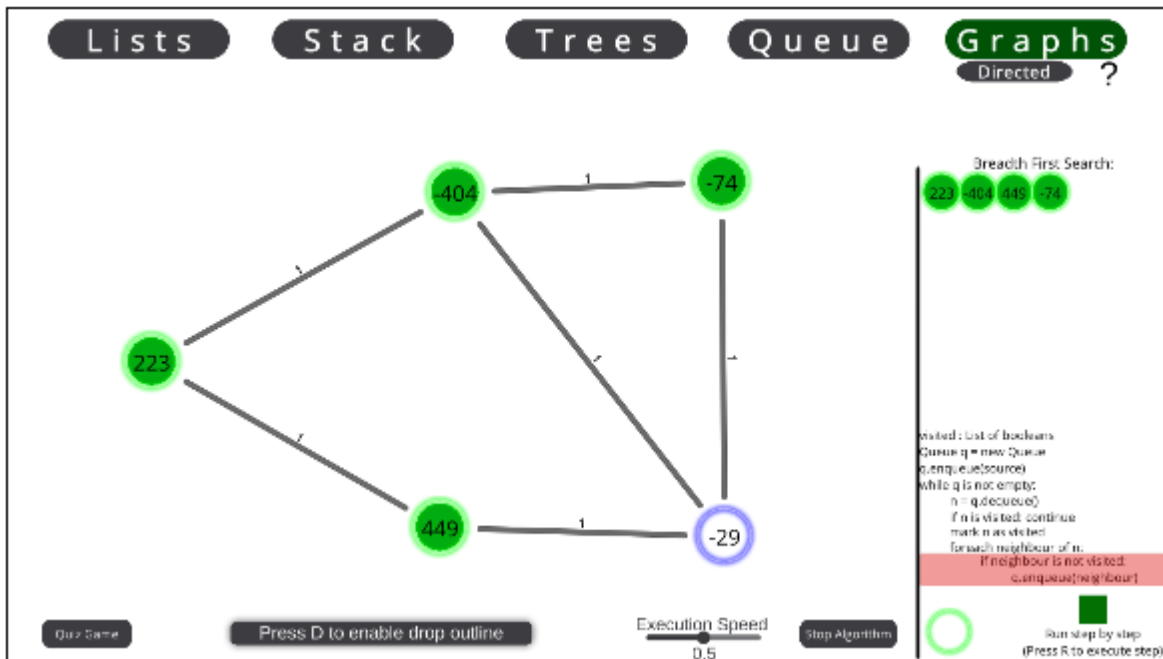
Στην επιλογή αναζήτησης κατά βάθος επιτυγχάνεται διάσχιση ή αναζήτηση σε δέντρο ή γράφημα. Η διάσχιση ξεκινά από έναν κόμβο και εξερευνά όσο το δυνατόν περισσότερο κατά μήκος γράφου μέχρι να φτάσει σε αδιέξοδο. Ο αλγόριθμος DFS πραγματοποιεί αναζήτηση που ξεκινά με την επέκταση του πρώτου κόμβου αναζήτησης που εμφανίζεται

και συνεχίζει σε βάθος του βάθος μέχρι να φτάσει σε κόμβο που δεν έχει παιδιά. Στη συνέχεια, η αναζήτηση συνεχίζεται στον πιο πρόσφατο ανεξερευνήτο κόμβο. Σε μια μη-αναδρομική υλοποίηση, όλοι οι νέοι ανεξερευνήτοι κόμβοι τοποθετούνται στην κορυφή μιας στοίβας και η αναζήτηση συνεχίζεται με έναν από αυτούς (Εικόνα 4.105).



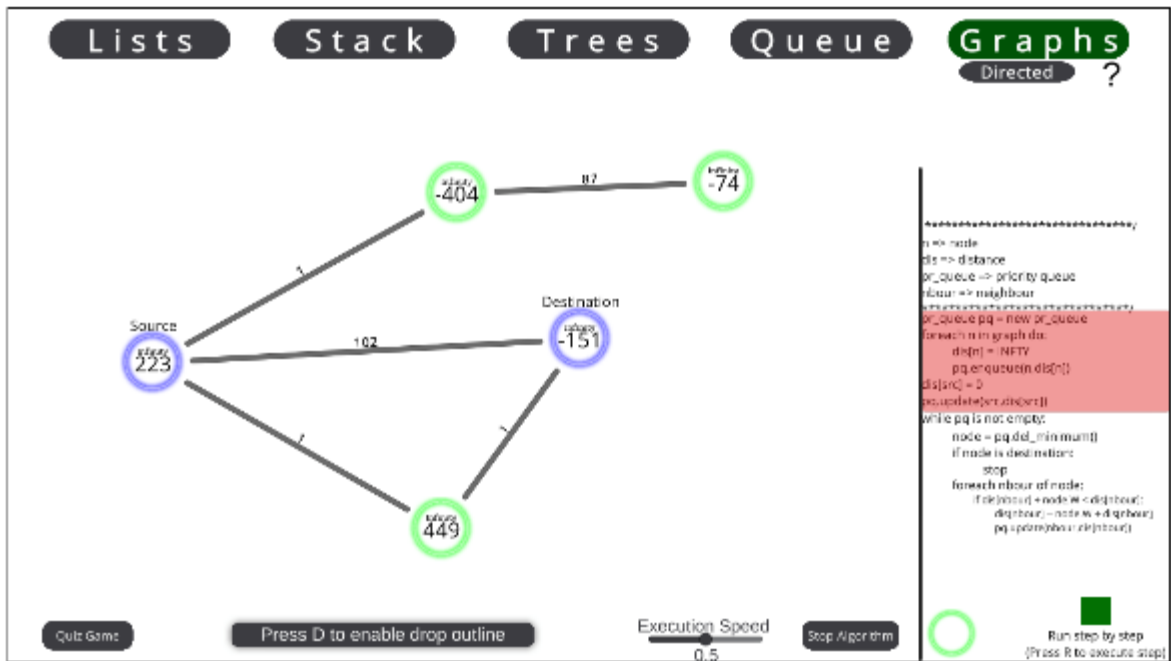
Εικόνα 4.105 Αλγόριθμος αναζήτησης κατά βάθος

Στην επιλογή αναζήτηση κατά πλάτος η αναζήτηση από κάποιο αυθαίρετο κόμβο του γραφήματος και διερευνά πρώτα τους γειτονικούς κόμβους, προτού μεταβεί στους γείτονες του επόμενου επίπεδο (Εικόνα 4.106).

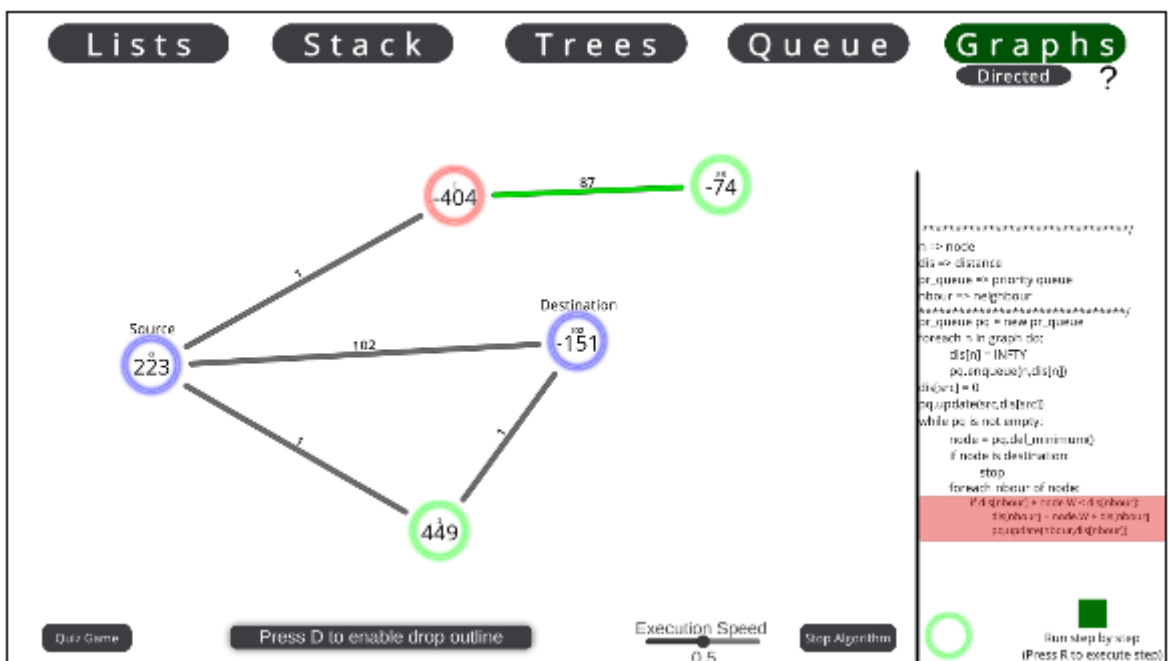


Εικόνα 4.106 Αλγόριθμος αναζήτησης κατά πλάτος

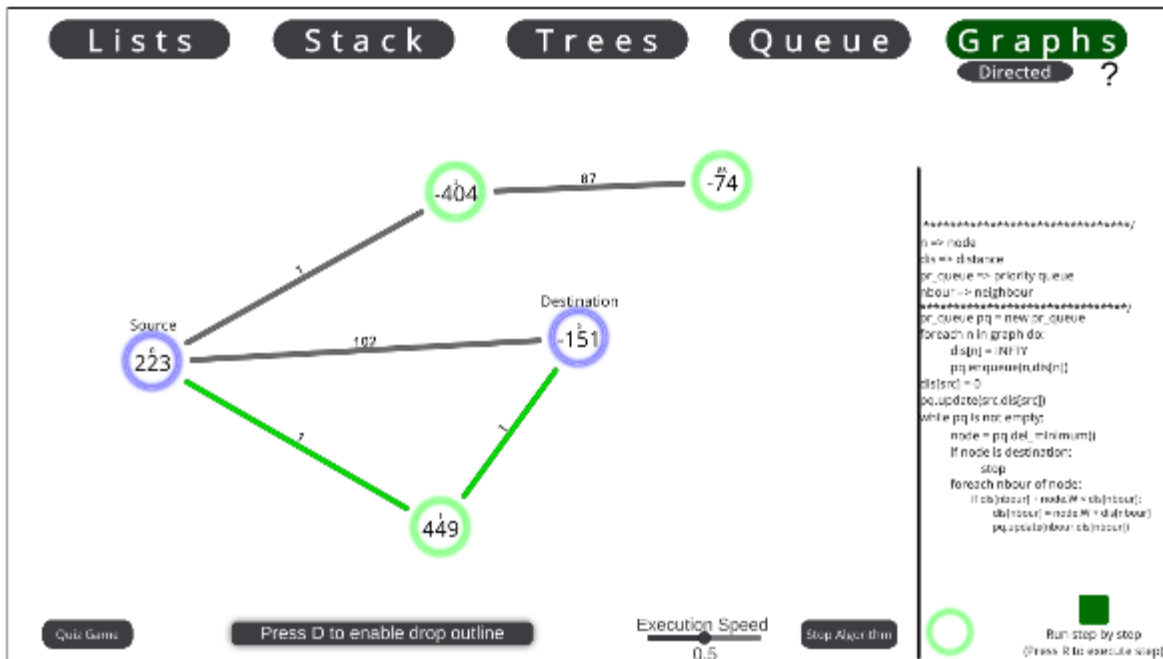
Τέλος στην τελευταία επιλογή Dijkstra πρόκειται για έναν αλγόριθμο εύρεσης συντομότερων διαδρομών (single-source shortest path problem) από κοινή αφετηρία σε έναν (κατευθυνόμενο ή μη) Γράφο με μη αρνητικά βάρη στις ακμές. Ο αλγόριθμος του Dijkstra είναι άπληστος. Δηλαδή, σε κάθε βήμα επιλέγει την τοπικά βέλτιστη λύση, ώσπου στο τελευταίο βήμα συνθέτει μια συνολικά βέλτιστη λύση. Αν ο Γράφος περιέχει αρνητικά βάρη, ο αλγόριθμος του Dijkstra δεν δίνει σωστό αποτέλεσμα. Στις εικόνες 4.107, 4.108, 4.109 φαίνεται η αφετηρία και ο προορισμός κατά την εφαρμογή του αλγόριθμου και στο τέλος εμφανίζεται το συντομότερο μονοπάτι.



Εικόνα 4.107 Αφαιρία και προορισμός κατά τον αλγόριθμο Dijkstra



Εικόνα 4.108 Ενδιάμεσο στάδιο Dijkstra



Εικόνα 4.109 Συντομότερο μονοπάτι

4.4.6 Παιχνίδι ερωτήσεων

Ένα από τα σημαντικότερα στοιχεία της εφαρμογής είναι το παιχνίδι ερωτήσεων που διαθέτει. Ο τρόπος διδασκαλίας αλλάζει διαρκώς και σήμερα η εκπαίδευση προσπαθεί να χρησιμοποιεί νέους τρόπους μάθησης οι οποίοι αποδεικνύονται πιο αποτελεσματικοί καθώς ενισχύουν τις δεξιότητες των νέων. Μία τέτοια μέθοδος αποτελούν τα παιχνίδια ερωτήσεων [33].

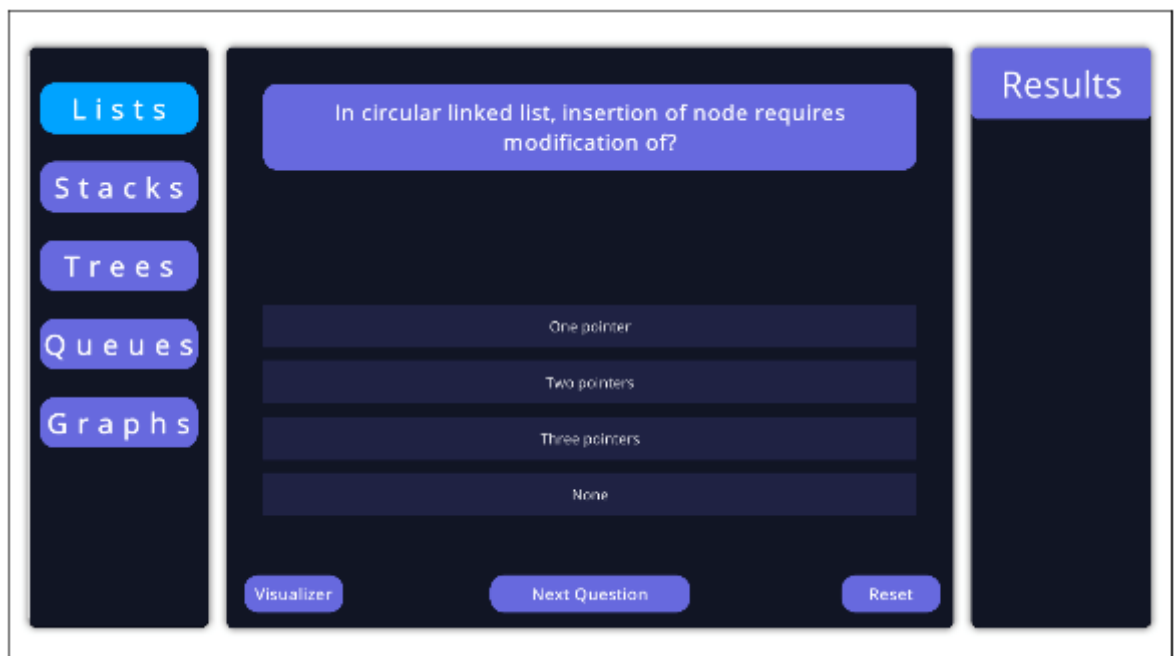
Τα παιχνίδια ερωτήσεων εκτός του ότι είναι αρκετά διασκεδαστικά για τους νέους, αποτελούν μια πιο δημιουργική μορφή μάθησης καθώς οι νέοι απομακρύνονται από τον παραδοσιακό και ανιαρό τρόπο διδασκαλίας. Επιπλέον συμβάλουν σημαντικά στην εξάσκηση την γνώσης και στην τόνωση του ενδιαφέροντος για μάθηση.

Ο χρήστης επιλέγοντας το κουμπί (Quiz Game) μέσα στην εφαρμογή, έχει την δυνατότητα να παίξει ένα απλό παιχνίδι ερωτήσεων και να τεστάρει τις γνώσεις του πάνω σε βασικές δομές δεδομένων και αλγορίθμων. Το παιχνίδι ερωτήσεων αποτελείται από τις εξής πέντε κατηγορίες:

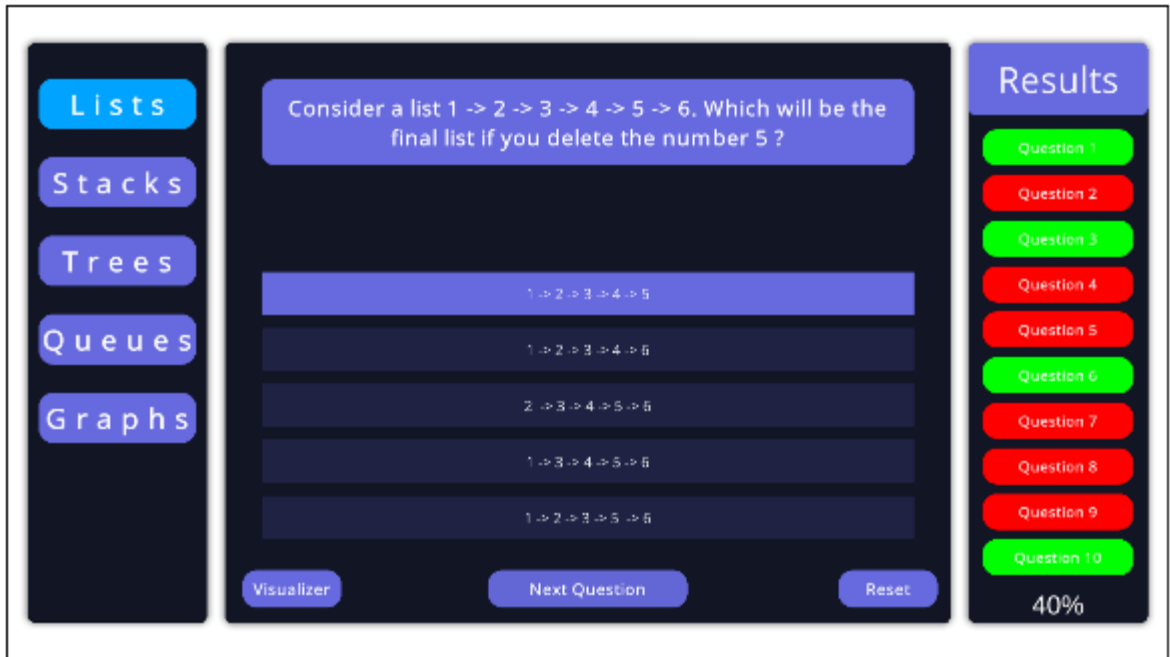
- Λίστες (Lists)
- Στοίβες (Stacks)

- Δέντρα (Trees)
- Ουρές (Queues)
- Γράφους (Graphs)

Κάθε κατηγορία περιέχει δέκα ερωτήσεις όπου ο χρήστης καλείται να απαντήσει. Στο τέλος του παιχνιδιού εμφανίζονται ποιες ερωτήσεις απαντήθηκαν σωστά, ποιες λανθασμένες καθώς και το ποσοστό επιτυχίας. Επίσης ο παίχτης έχει την δυνατότητα να επαναλάβει το παιχνίδι ερωτήσεων οποιαδήποτε στιγμή επιθυμεί (Εικόνα 4.110), (Εικόνα 4.111)



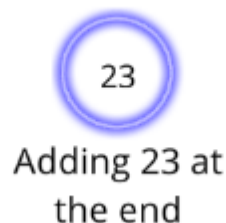
Εικόνα 4.110 Παιχνίδι ερωτήσεων



Εικόνα 4.111 Ποσοστό επιτυχίας παιχνιδιού ερωτήσεων

4.4.7 Γενικές πληροφορίες

Τέλος σε όλους τους αλγόριθμους της εφαρμογής για την διευκόλυνση του χρήστη εμφανίζονται ορισμένες πληροφορίες όπως για παράδειγμα ο κόμβος ο οποίος αναζητείται (Εικόνα 4.112), και στα δεξιά της εφαρμογής εμφανίζεται ο αντίστοιχος κώδικας της κάθε λειτουργίας. Με αυτόν τον τρόπο ο χρήστης έχει την δυνατότητα να κατανοήσει τον τρόπο λειτουργίας των αντίστοιχων αλγορίθμων (Εικόνα 4.113). Επιπλέον σε περίπτωση αποτυχίας κάποιου αλγορίθμου ή κάποιου λάθους του χρήστη εμφανίζεται μήνυμα αποτυχίας (Εικόνα 4.114). Όλες αυτές οι συγκεκριμένες λειτουργίες κάνουν το περιβάλλον της εφαρμογής πιο φιλικό και πιο απλοποιημένο προς το χρήστη.



Εικόνα 4.112 Πληροφορίες εφαρμογής στην ένθεση κόμβου

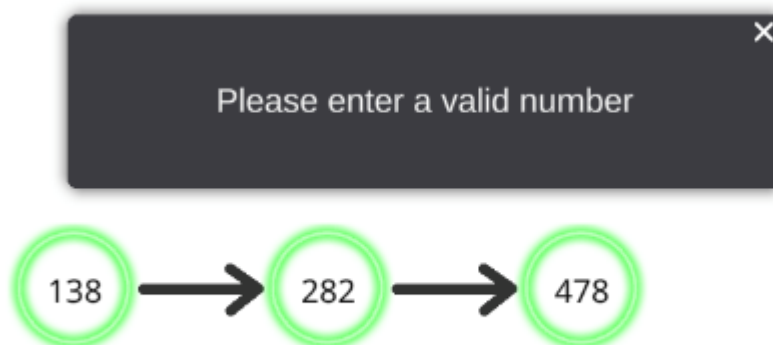
Level Order Traversal

1 2 3

```
Queue queue = new Queue
queue.Enqueue(head)
Node node = null
while queue is not empty:
  node = queue.Dequeue()
  if node.left != null:
    queue.Enqueue(node.left)
  if node.right != null:
    queue.Enqueue(node.right)
```

Run step by step
(Press R to execute step)

Εικόνα 4.113 Ψευδοκώδικας αλγορίθμων



Εικόνα 4.114 Μήνυμα αποτυχίας

4.4.8 Project ALIEN

Η παρούσα διπλωματική εργασία αναπτύχθηκε στα πλαίσια του project ALIEN το οποίο στοχεύει στη βελτίωση της ποιότητας της τριτοβάθμιας εκπαίδευσης παρέχοντας πιο

ενθαρρυντικά και αποτελεσματικά μαθησιακά πλαίσια που βασίζονται στην ενεργή μάθηση και στη μάθηση με βάση τα διάφορα προβλήματα που προετοιμάζουν τους φοιτητές για την επαγγελματική τους ζωή μέσω της ανάπτυξης των επιθυμητών ικανοτήτων της βιομηχανίας. Επίσης στοχεύει στην καθιέρωση της ενεργητικής μάθησης για την επίτευξη καλύτερης εκπαίδευσης στον τομέα της μηχανικής [34].

Τα πλεονεκτήματα του project ALIEN μέσω της ενεργητικής και βασισμένης σε προβλήματα μάθησης είναι πολλά [35]:

- Εκσυγχρονίζει τη μάθηση καθώς προωθεί την ενεργό μάθηση μέσα από ψηφιακές τεχνολογίες.
- Χτίζει βασικές γνώσεις μέσω των εργαστηρίων ALIEN τα οποία προωθούν την ενεργητική ανάπτυξη της θεμελιώδους γνώσης στις αρχές της μηχανικής.
- Δημιουργεί δεξιότητες όπως είναι η κριτική σκέψη, η αναλυτική σκέψη και η ικανότητα συνεργασίας
- Προωθεί την μεταλαμπάδευση της γνώσης. Η ενεργή μάθηση με τη μορφή ψηφιακών παιχνιδιών και προσομοιώσεων προωθεί την χρήση νέων γνώσεων με τρόπο που προσομοιώνει πρακτικές του πραγματικού κόσμου.

Το project ALIEN χρηματοδοτείται από το πρόγραμμα Erasmus+ και εφαρμόζεται σε πέντε Ευρωπαϊκές χώρες και πέντε Ασιατικές χώρες με την συμμετοχή 17 πανεπιστημίων από την Ελλάδα, την Πορτογαλία, το Ηνωμένο Βασίλειο, την Εσθονία, τη Βουλγαρία, τη Μαλαισία, το Βιετνάμ, την Καμπότζη, το Νεπάλ και το Πακιστάν [34].

ΚΕΦΑΛΑΙΟ 5

ΣΥΝΟΨΗ

5.1 Συμπεράσματα

Το εκπαιδευτικό σύστημα προσπαθώντας να ανταπεξέλθει και να συμβαδίσει με τη ραγδαία τεχνολογική εξέλιξη και να καλύψει τις όλο και περισσότερο αυξανόμενες ανάγκες των νέων προσπαθεί να υιοθετήσει καινοτόμους τρόπους διδασκαλίας. Ένας από αυτούς είναι η παιχνιδοποιημένη μάθηση, όπου τα στοιχεία της είναι αποδεδειγμένο ότι καθιστούν πιο αποτελεσματική την διαδικασία της μάθησης.

Στην παρούσα διπλωματική εργασία ασχοληθήκαμε με τη δημιουργία ενός προσομοιωτή δομών δεδομένων και αλγορίθμων, με στόχο την διευκόλυνση των νέων στην κατανόηση του συγκεκριμένου μαθήματος με γνώμονα την παιχνιδοποιημένη μάθηση. Το εργαλείο αυτό κεντρίζει το ενδιαφέρον των φοιτητών και τους οδηγεί στην συστηματική ενασχόληση και έρευνα. Επίσης, προσφέρει στους καθηγητές την δυνατότητα να εισάγουν καινοτόμες μαθησιακές μεθόδους στην εκπαίδευση μέσω αναδυόμενων τεχνολογιών.

Το κύριο στοιχείο της εφαρμογής είναι η ενσωμάτωση της γνώσης στα πλαίσια ενός ψηφιακού περιβάλλοντος. Με αυτόν τον τρόπο η εκπαίδευση απομακρύνεται από τους παραδοσιακούς και ανιαρούς τρόπους διδασκαλίας και προσφέρει στους νέους ποικίλα ερεθίσματα για την κατάκτηση της γνώσης.

5.2 Μελλοντικές επεκτάσεις της εφαρμογής

Η πλατφόρμα είναι απόλυτα λειτουργική και η σχεδίαση της είναι αποτέλεσμα μιας επαναληπτικής διαδικασίας βασισμένη στην ανατροφοδότηση των χρηστών και των εκπαιδευτικών. Παρ' όλα αυτά υπάρχουν περιθώρια βελτίωσης στα παρακάτω σημεία:

- Σχεδίαση διεπαφής χρήστη ώστε να είναι όσο το δυνατό περισσότερο φιλική προς τους χρήστες
- Βελτίωση απεικόνισης της εφαρμογής σε κινητές συσκευές
- Ένταξη περισσότερων στοιχείων παιχνιδοποιημένης μάθησης σε όλη τη ροή της εφαρμογής
- Δημιουργία και απεικόνιση ακόμα περισσότερων δομών δεδομένων και αλγορίθμων
- Χρήση περισσότερων πολυμέσων στην περιγραφή των αλγορίθμων

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] «Παιδεία - Εκπαίδευση» [Ηλεκτρονικό]. Available:
https://latistor.blogspot.com/2015/07/blog-post_19.html.
[Πρόσβαση 4/1/2021].
- [2] «Ανώτατη Εκπαίδευση» [Ηλεκτρονικό]. Available:
http://5lyk-kallith.att.sch.gr/images/Ανώτατη_Εκπαίδευση.pdf
[Πρόσβαση 4/1/2021].
- [3] Δημήτρης Χρυσόπουλος, Οι νέες τεχνολογίες στην εκπαίδευση, 2018
- [4] Δήμητρα Θεολόγου, Η τεχνολογία στην εκπαίδευση? Ωφελεί ή βλάπτει;, 2017
- [5] "Linked List Data Structure - GeeksforGeeks", *GeeksforGeeks*, 2021. [Online].
Available: <https://www.geeksforgeeks.org/data-structures/linked-list/>.
[Πρόσβαση 4/1/2021].
- [6] "Introduction to Algorithms - GeeksforGeeks", *GeeksforGeeks*, 2021.
[Ηλεκτρονικό].
Available: <https://www.geeksforgeeks.org/introduction-to-algorithms/>.
[Πρόσβαση 2/1/2021].
- [7] "VisuAlgo - visualising data structures and algorithms through animation", *Visualgo.net*, 2021.
[Online]. Available: <https://visualgo.net/en>. [Πρόσβαση 2/1/2021].
- [8] Καπραβέλου, Αναστασία – Αλέξανδρος, Ενσωμάτωση της ηλεκτρονικής μάθησης στην πανεπιστημιακή διδασκαλία, 2006
- [9] Gabriela Kiryakova , Nadezhda Angelova , Lina Yordanova, Gamification in education, 2014

- [10] «Gamification: έννοια και οφέλη» [Ηλεκτρονικό]. Available: <https://www.flexlearn.com/blog/articles/gamification-ennoia-kai-ofeli/>
[Πρόσβαση 26/12/20].
- [11] «Gamification και e-learning» [Ηλεκτρονικό]. Available: <https://www.sqllearn.gr/gamification-kai-e-learning/> [Πρόσβαση 17/12/20]
- [12] Wendy Hsin-Yuan Huang and Dilip Soman, A Practitioner's Guide To Gamification of Education, 2003
- [13] Simoes, J., R. Diaz Redondo, A. Fernandez Vilas, A social gamification framework for a K-6 learning platform. Computers in Human Behavior, 2013
- [14] «Gamification»[Ηλεκτρονικό].Available: <https://en.wikipedia.org/wiki/Gamification>
[Πρόσβαση 27/12/20].
- [15] Estela Lizbeth Munoz Andrade, Carlos Argelio Mercado, Juan Manuel Gomez Reynoso, Learning Data Structures Using Multimedia-Interactive Systems, 2015
- [16] «Data Structure Visualizations» [Ηλεκτρονικό]. Available: <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>
[Πρόσβαση 2/1/2021].
- [17] «VISUALGO» [Ηλεκτρονικό]. Available: <https://visualgo.net/en>
[Πρόσβαση 2/1/2021].
- [18] «Unity Engine»[Ηλεκτρονικό].Available: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
[Πρόσβαση: 13/2/2021].
- [19] U. Technologies, "Unity - Manual: Unity's interface", *Docs.unity3d.com*, 2021. [Ηλεκτρονικό].Available:

- <https://docs.unity3d.com/Manual/UsingTheEditor.html>.
- [Πρόσβαση: 13/2/2021].
- [20] «C Sharp (programming language)»[Ηλεκτρονικό].Available:
[https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))
- [21] «Data Structure»[Ηλεκτρονικό].Available:
https://en.wikipedia.org/wiki/Data_structure. [Πρόσβαση: 13/2/2021].
- [22] «Linked list»[Ηλεκτρονικό].Available:
https://en.wikipedia.org/wiki/Linked_list. [Πρόσβαση: 13/2/2021].
- [23] « Stack (abstract data type) »[Ηλεκτρονικό].Available:
[https://en.wikipedia.org/wiki/Stack_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type)).
[Πρόσβαση: 13/2/2021].
- [24] "How to solve the valid parentheses problem in Java?", *Memoirs of a Software developer*,2021.[Ηλεκτρονικό].Available:
<https://ajayiyengar.com/2020/08/16/how-to-solve-the-valid-parentheses-problem-in-java/>. [Πρόσβαση: 13/2/2021].
- [25] "infix, prefix & postfix notations (Polish Notations) – Ritambhara Technologies", *Ritambhara.in*,2021.[Ηλεκτρονικό].Available:
<https://www.ritambhara.in/infix-prefix-postfix-notations-polish-notations/>. [Πρόσβαση: 13/2/2021].
- [26] « Queue (abstract data type) »[Ηλεκτρονικό].Available:
[https://en.wikipedia.org/wiki/Queue_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Queue_(abstract_data_type)).
[Πρόσβαση: 13/2/2021].
- [27] « What is a Queue Data Structure?»[Ηλεκτρονικό].Available:

<https://www.studytonight.com/data-structures/queue-data-structure>

[Πρόσβαση: 13/2/2021].

[28] « Γράφος »[Ηλεκτρονικό], Available:

[29] "Data Structure and Algorithms - Tree - Tutorialspoint", *Tutorialspoint.com*, 2021.

[Ηλεκτρονικό].Available:

https://www.tutorialspoint.com/data_structures_algorithms/tree_data_structure.htm. [Πρόσβαση: 13/2/2021].

[30] "Binary tree", *En.wikipedia.org*, 2021. [Ηλεκτρονικό].

Available: https://en.wikipedia.org/wiki/Binary_tree.

[Πρόσβαση: 13/2/2021].

[31] "Tree traversal", *En.wikipedia.org*, 2021. [Ηλεκτρονικό].

Available:

https://en.wikipedia.org/wiki/Tree_traversal. [Πρόσβαση: 13/2/2021].

[32] "Data Structure and Algorithms - AVL Trees - Tutorialspoint", *Tutorialspoint.com*,

2021.[Ηλεκτρονικό].Available:

https://www.tutorialspoint.com/data_structures_algorithms/avl_tree_algorithm.htm. [Πρόσβαση: 13/2/2021].

[33] «THE BENEFITS OF QUIZES IN THE CLASSROOM» [Ηλεκτρονικό], Available:

<https://www.ricpublications.com.au/blog/post/the-benefits-of-quizzes-in-the-classroom/>.

[Πρόσβαση: 13/2/2021].

[34] «ALIEN Active Learning in Engineering Education» [Ηλεκτρονικό], Available:

<http://projectalien.eu>.

[Πρόσβαση 14/2/2021].

[35] «ALIEN Active Learning in Engineering Education Benefits» [Ηλεκτρονικό],
Available:

<http://projectalien.eu/index.php/objectives/>.

[Πρόσβαση 14/2/2021].