



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ

ΡΟΗ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΑΔΡΑΣΤΙΚΗ ΟΠΤΙΚΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΓΡΑΦΩΝ ΣΕ ΠΕΡΙΒΑΛΛΟΝ ΠΑΙΧΝΙΔΙΟΥ

Του φοιτητή

ΠΑΝΑΓΟΣ ΝΙΚΟΛΑΟΣ

Αριθμός Μητρώου : **71347461**

Επιβλέπων Καθηγητής

Γεώργιος Μπαρδής, Επίκουρος Καθηγητής

Αιγάλεω, Οκτώβριος 2023



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ
ΡΟΗ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΑΔΡΑΣΤΙΚΗ ΟΠΤΙΚΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΓΡΑΦΩΝ ΣΕ ΠΕΡΙΒΑΛΛΟΝ ΠΑΙΧΝΙΔΙΟΥ

Ον/μο: Νικόλαος Πανάγος

Αριθμός Μητρώου : 71347461

Επιβλέπων καθηγητής: Γεώργιος Μπαρδής, Επίκουρος Καθηγητής

Ημερομηνία εξέτασης : 11/10/2023

Εγκρίθηκε από 3 μέλη επιτροπή των:

--	--	--

Γεώργιος Μπαρδής,
Επίκουρος Καθηγητής

Χρήστος Τρούσσας,
Επίκουρος Καθηγητής

Παναγιώτα Τσελέντη,
Ε.ΔΙ.Π.

Αιγάλεω, Οκτώβριος 2023



UNIVERSITY OF WEST ATTICA
SCHOOL OF ENGINEERING
DEPARTMENT OF INFORMATICS AND COMPUTER
ENGINEERING
SOFTWARE AND INFORMATION SYSTEMS FLOW

INTERACTIVE VISUALIZATION OF GRAPH ALGORITHMS IN A GAME ENVIROMENT

Full Name: PANAGOS NIKOLAOS

ID NUMBER: 71347461

Supervisor: George Bardis, Assistant Professor

Egaleo, October 2023

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Πανάγος Νικόλαος του Ιωάννη, με αριθμό μητρώου 71347461 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



Περίληψη

Οι γράφοι αποτελούν ένα ισχυρό εργαλείο μοντελοποίησης προβλημάτων. Έχει αναπτυχθεί ένας ιδιαίτερος τομέας της πληροφορικής για τη μελέτη τους. Το γεγονός αυτό αποτελεί το κίνητρο για την ανάπτυξη σχετικών προγραμμάτων εκπαίδευσης. Η οπτικοποίηση του εκπαιδευτικού υλικού αποτελεί σημαντικό παράγοντα για την αύξηση της αποδοτικότητας της διδασκαλίας. Τα εκπαιδευτικά παιχνίδια είναι χαρακτηριστική έκφραση της οπτικοποίησης της εκπαίδευσης. Στην παρούσα εργασία επιχειρήθηκε η οπτικοποίηση δύο βασικών αλγορίθμων γράφων: της αναζήτησης πρώτα κατά βάθος (Depth first Search – DFS) και της αναζήτησης πρώτα κατά πλάτος (Breadth first Search – BFS). Χρησιμοποιήθηκε η πλατφόρμα ανάπτυξης παιχνιδιών Unity. Μέσα από τη διαδικασία ανάπτυξης φάνηκε ότι το εκπαιδευτικό παιχνίδι μπορεί να συμβάλει στην παραστατική οπτικοποίηση των μαθημάτων.

Λέξεις - Κλειδιά: Γράφοι, Μοντελοποίηση προβλημάτων, Πληροφορική, Εκπαίδευση, Οπτικοποίηση, Εκπαιδευτικά προγράμματα, Εκπαιδευτικά παιχνίδια, Αναζήτηση πρώτα κατά βάθος (DFS), Αναζήτηση πρώτα κατά πλάτος (BFS), Unity, Παραστατική οπτικοποίηση, Κίνητρο, Διδασκαλία.

Abstract

Graphs are a powerful problem modeling tool. A special field of computer science has been developed for their study. This fact is the motivation for the development of relevant training programs. The visualization of educational material is an important factor in increasing the efficiency of teaching. Educational games are a typical expression of the visualization of education. In this work, the visualization of two basic graph algorithms was attempted: Depth first Search (DFS) and Breadth first Search (BFS). Unity game development platform was used. Through the development process, it was seen that the educational game can contribute to the practical visualization of the lessons.

Keywords: Graphs, Problem modeling, Computer science, Training programs, Visualization, Educational material, Efficiency of teaching, Educational games, Depth First Search (DFS), Breadth First Search (BFS), Unity game development platform, Practical visualization, Lessons.

Πίνακας Περιεχομένων

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	6
Περίληψη	7
Abstract.....	8
Πίνακας Εικόνων	12
1 Εισαγωγή	13
2 Εκπαιδευτικό Λογισμικό	16
2.1 Τεχνολογική Πρόοδος και Εκπαίδευση	16
2.2 Εκπαιδευτικό Λογισμικό	18
2.3 Κατηγορίες εκπαιδευτικού λογισμικού	19
2.4 Εκπαιδευτικά Παιχνίδια.....	21
3 Αλγόριθμοι Γράφων	24
3.1 Ορισμός – Χαρακτηριστικά – Είδη Γράφων	24
3.2 Βασικοί αλγόριθμοι γράφων	26
3.2.1 Αλγόριθμος αναζήτησης πρώτα κατά βάθος (Depth-First Search - DFS)	27
3.2.2 Αλγόριθμος αναζήτησης πρώτα κατά πλάτος (Breadth-First Search - BFS)	27
3.2.3 Αλγόριθμος Dijkstra.....	27
3.2.4 Αλγόριθμος Bellman Ford	27
3.2.5 Αλγόριθμος Floyd Warshal	28
3.2.6 Αλγόριθμοι Prim και Kruskal	28
3.3 Αλγόριθμοι αναζήτησης σε γράφους	28
3.3.1 Αναζήτηση πρώτα κατά βάθος.....	28

3.3.2	Αναζήτηση πρώτα κατά πλάτος	32
3.4	Διδασκαλία των αλγορίθμων γράφων.....	35
4	Η πλατφόρμα ανάπτυξης παιχνιδιών Unity.....	38
4.1	Γενική Περιγραφή	38
4.2	Το Περιβάλλον Ανάπτυξης.....	39
5	Εφαρμογή οπτικοποίησης αλγορίθμων γράφων.....	43
5.1	Σχεδίαση του παιχνιδιού	43
5.2	Περιγραφή του παιχνιδιού	43
5.3	Λειτουργίες του Παιχνιδιού	44
5.4	Καταστάσεις παιχνιδιού.....	45
5.5	Χρήση του παιχνιδιού	46
5.5.1	Αρχική οθόνη	46
5.5.2	Βασική οθόνη του παιχνιδιού.....	47
5.6	Υλοποίηση	55
5.6.1	Δημιουργία συνεκτικών γράφων.....	56
5.6.2	Δημιουργία Κόμβων.....	56
5.6.3	Προβολή χάρτη	57
5.6.4	Εναλλαγή προβολών	57
5.6.5	Προσέγγιση κόμβου	57
5.6.6	Προβολή βοήθειας	58
5.6.7	Ολοκλήρωση παιχνιδιού	58
5.6.8	Αλλαγή σε παρακολούθηση αλγορίθμου	58
5.6.9	Έλεγχος για κίνηση πάνω στην ακμή.....	59
5.6.10	Χειριστήρια κίνησης του χαρακτήρα.....	60
5.6.11	Κίνηση στα όρια του πεδίου	60

6	Συμπεράσματα	61
	Βιβλιογραφία	65
7	Παράρτημα Α: Δημιουργία μονοπατιού Hamilton.....	67
8	Παράρτημα Β: Δημιουργία ακμών του γράφου	69
9	Παράρτημα Γ: Δημιουργία κόμβων	71
10	Παράρτημα Δ: Εναλλαγή προβολής	73
11	Παράρτημα Ε: Έλεγχοι για την προσέγγιση κόμβου.....	74
12	Παράρτημα Ζ: Προβολή βοήθειας.....	77
13	Παράρτημα Η: Ολοκλήρωση του παιχνιδιού	79
14	Παράρτημα Θ: Υλοποίηση του αλγορίθμου BFS.....	80
15	Παράρτημα Στ: Κώδικας ελέγχου κίνησης πάνω σε ακμή	84
16	Παράρτημα Ι: Υλοποίηση του αλγορίθμου DFS	85
16.1.1	Παράρτημα ΙΑ: Χειρισμός κινούμενου χαρακτήρα.....	86
16.1.2	Παράρτημα ΙΒ: Έλεγχος παραμονής στο πεδίο	87

Πίνακας Εικόνων

Εικόνα 1: Παράδειγμα γράφου	24
Εικόνα 2: Παράδειγμα αναπαράστασης γράφου με λίστες γειτνίασης	26
Εικόνα 3: Παράδειγμα αναπαράστασης γράφου με πίνακα γειτνίασης	26
Εικόνα 4: Εξέλιξη της αναδρομικής εκδοχής του DFS	30
Εικόνα 5: Η εξέλιξη του DFS με χρήση στοίβας.....	31
Εικόνα 6: Ο ψευδοκώδικας του αλγορίθμου BFS.....	34
Εικόνα 7: Σχηματική αναπαράσταση της εξέλιξης του BFS.....	35
Εικόνα 8: Οθόνη διαμόρφωσης γραφικών στοιχείων εφαρμογών	40
Εικόνα 9: Παράδειγμα κώδικα Unity	41
Εικόνα 10: Παραγωγή εκτελέσιμου αρχείου	42
Εικόνα 11: Διάγραμμα δραστηριοτήτων εξέλιξης παιχνιδιού	45
Εικόνα 12: Αρχική οθόνη επιλογής παραμέτρων παιχνιδιού	47
Εικόνα 13: Πλήκτρα αλλαγής τρόπου κίνησης	48
Εικόνα 14: Προβολή της ακμής που πρέπει να κινηθεί ο κινούμενος χαρακτήρας.	49
Εικόνα 15: Επισήμανση ακμής για επιστροφή	50
Εικόνα 16: Επισήμανση ακμής που έχει ήδη διανυθεί	50
Εικόνα 17: Επισήμανση επίσκεψης κόμβου.....	51
Εικόνα 18: Επισήμανση κόμβων που έχουν ελεγχθεί σε προβολή χάρτη	51
Εικόνα 19: Προβολή βοήθειας	52
Εικόνα 21: Έλεγχος κίνησης πάνω σε ακμή.....	53
Εικόνα 22: Προσέγγιση κόμβου (προβολή χάρτη).....	54
Εικόνα 23: Προσέγγιση κόμβου (ρεαλιστική προβολή).....	54
Εικόνα 24: Τέλος του παιχνιδιού σε ρεαλιστική προβολή	55

1 Εισαγωγή

Αλγόριθμος ονομάζεται το σύνολο των σταδίων από τα οποία περνάει η επίλυση ενός προβλήματος. Τα στάδια αυτά θα πρέπει να είναι πεπερασμένα σε πλήθος, σαφώς καθορισμένα. Το αποτέλεσμα που παρέχει ο αλγόριθμος ως λύση θα πρέπει να είναι σωστό. Από έναν αλγόριθμο είναι επίσης ζητούμενο να ολοκληρώνεται σε εύλογο χρονικό διάστημα. Πριν υλοποιηθεί ένας αλγόριθμος με τη χρήση κάποιας γλώσσας προγραμματισμού, εξετάζεται το κατά πόσο είναι αποδοτικός, την κατάταξη του σε μία κλάση πολυπλοκότητας. Οι κλάσεις πολυπλοκότητας είναι ένας τρόπος για την κατάταξη των αλγορίθμων ως προς της ποιότητα τους. Συχνά για ένα πρόβλημα προτείνονται διάφοροι αλγόριθμοι για την επίλυση τους ώστε τελικά να επιλεγεί ο πλέον αποδοτικός. Σε πολλές επίσης περιπτώσεις, η απόδοση των αλγορίθμων επηρεάζεται τόσο από τον τρόπο που θα μοντελοποιηθεί το πρόβλημα, όσο και από τις δομές δεδομένων που θα χρησιμοποιηθούν (Papadimitriou, 2003).

Οι γράφοι είναι ένα τρόπος μοντελοποίησης των προβλημάτων. Έχουν την δυνατότητα να αναπαραστούν με λίγα και απλά γραφικά σύμβολα, πολύπλοκα ως προς την περιγραφή τους προβλήματα με αποτέλεσμα τα ζητούμενα τους αλλά και η εξέλιξη των αλγορίθμων επίλυσης τους, να είναι ευκολότερα κατανοητή από τον άνθρωπο. Παράλληλα, οι δομές δεδομένων που αναπαρίστανται με γράφους, μπορούν εύκολα να αποθηκευτούν σε ηλεκτρονικούς υπολογιστές.

Μία ακόμα πρόκληση που χρειάζεται να αντιμετωπίζεται με επιτυχία, είναι η εύρεση τρόπων εκμάθησης των αλγορίθμων επίλυσης προβλημάτων. Η φύση της μελέτης των αλγορίθμων (στην οποία εμπλέκεται και η γνώση μαθηματικών σχέσεων), προκαλεί δυσκολίες στην κατανόηση τους από μεγάλη μερίδα ανθρώπων. Είναι ένα από τα βασικά ζητούμενα κατά τη διδασκαλία τους, η ανεύρεση τρόπων περιγραφής τους, που να είναι απολύτως κατανοητή από τον άνθρωπο. Οι εικόνες είναι το αποτελεσματικό εργαλείο για την υποβοήθηση της απομνημόνευσης μεθόδων επίλυσης προβλημάτων. Από την άλλη μεριά, οι ίδιοι οι γράφοι είναι στην πραγματικότητα απεικονίσεις δομών δεδομένων και στατικών συμβολισμών λειτουργιών, με αποτέλεσμα εύκολα να μπορούν να αποδοθούν σε εικόνα. Τα πλεονεκτήματα των απεικονίσεων, σε συνδυασμό με την παιχνιδιοποίηση της εκπαίδευσης και τη διαδραστικότητα, μπορούν να βελτιστοποιήσουν την απόδοση της εκπαίδευσης με αντικείμενο τους αλγορίθμους.

Σήμερα διατίθενται ένας μεγάλος αριθμός εργαλείων για την ταχεία ανάπτυξη εκπαιδευτικού λογισμικού, ικανού να ικανοποιήσει μία ευρεία γκάμα απαιτήσεων. Οι προγραμματιστές, χρησιμοποιώντας τα εργαλεία αυτά, αποκτούν τη δυνατότητα να

δημιουργούν εκπαιδευτικές εφαρμογές, προσαρμοσμένες απόλυτα στις εκπαιδευτικές λειτουργικές απαιτήσεις τους, που ικανοποιούν τις μη λειτουργικές τους απαιτήσεις και με υψηλό βαθμό προσαρμοστικότητας σε μεταβολές των απαιτήσεων. Ένα τέτοιο εργαλείο είναι το περιβάλλον ανάπτυξης παιχνιδιών Unity.

Στην παρούσα εργασία παρουσιάζεται ο τρόπος με τον οποίο η ανάπτυξη μίας διαδραστικής εφαρμογής για την υποστήριξη της διδασκαλίας αλγορίθμων, μπορεί να αναβαθμίσει την αποτελεσματικότητα της εκπαιδευτικής διαδικασίας. Η στόχευση της παραγωγής συμπερασμάτων δεν περιορίζεται αποκλειστικά στην αποδοτικότητα της εκπαίδευσης. Επεκτείνεται επίσης στο βαθμό δυσκολίας ανάπτυξης τέτοιου είδους εφαρμογών, όπως επίσης και στο οικονομικό κόστος που απαιτείται για την ολοκλήρωσή τους.

Το υπόλοιπο του παρόντος κειμένου είναι διαρθρωμένο ως εξής:

- Κεφάλαιο 1^ο: Το κεφάλαιο αυτό είναι αφιερωμένο στον τρόπο και τον βαθμό που το εκπαιδευτικό λογισμικό μπορεί να συμβάλει στην αναβάθμιση της εκπαιδευτικής διαδικασίας. Παρουσιάζονται συνοπτικά οι κατηγορίες του εκπαιδευτικού λογισμικού και πως η κάθε μία επηρεάζει θετικά την εκπαίδευση, με έμφαση στα εκπαιδευτικά παιχνίδια και τις προσομοιώσεις. Στο τέλος του κεφαλαίου, περιλαμβάνεται μία συνοπτική περιγραφή του Unity.
- Κεφάλαιο 2^ο: Στο κεφάλαιο αυτό περιγράφονται τα στοιχεία που θα αποτελέσουν το αντικείμενο της μελέτης περίπτωσης. Γίνονται αναφορές στους γράφους, σε αλγορίθμους που χειρίζονται γράφους για την επίλυση προβλημάτων. Παρουσιάζονται οι αλγόριθμοι αναζήτησης πρώτα κατά βάθος και πρώτα κατά πλάτος καθώς και τα προβλήματα του πραγματικού κόσμου, στον οποίων την επίλυση χρησιμοποιούνται.
- Κεφάλαιο 3^ο: Μετά τη θεωρητική προσέγγιση του αντικειμένου της εργασίας, περιγράφεται συνοπτικά το περιβάλλον ανάπτυξης παιχνιδιών Unity, που χρησιμοποιήθηκε για την ανάπτυξη του παιχνιδιού. Καταγράφονται οι δυνατότητες τους, οι προϋποθέσεις λειτουργίας του καθώς και η δυναμική του.
- Κεφάλαιο 4^ο: Στο κεφάλαιο αυτό περιγράφεται η εφαρμογή που αναπτύχθηκε. Αρχικά παρουσιάζεται η διαδικασία σχεδίασης της με φυσική γλώσσα και με τη χρήση διαγραμμάτων. Στην συνέχεια παρουσιάζεται ο τρόπος χρήσης της εφαρμογής.
- Κεφάλαιο 5^ο: Στο τελευταίο κεφάλαιο της εργασίας παρέχονται τα συμπεράσματα που προκύπτουν τόσο από τη γενικότερη συνεισφορά του εκπαιδευτικού λογισμικού στις εκπαιδευτικές διαδικασίες όσο και το βαθμό

που η χρήση διαδραστικών εφαρμογών κάνει για τον άνθρωπο, περισσότερο κατανοητές έννοιες και διαδικασίες των αλγορίθμων επίλυσης προβλημάτων. Επίσης παρουσιάζονται και συμπεράσματα που αναφέρονται στην θεώρηση της ανάπτυξης των εκπαιδευτικών εφαρμογών, από τη μεριά των προγραμματιστών.

2 Εκπαιδευτικό Λογισμικό

2.1 Τεχνολογική Πρόοδος και Εκπαίδευση

Η τεχνολογική πρόοδος στην Πληροφορική και τις τηλεπικοινωνίες, έχει βρει πολλές και σημαντικές εφαρμογές στην εκπαίδευση. Αναπτύχθηκαν νέες μεθοδολογίες για την πιο αποδοτική διδασκαλία μίας ποικιλίας αντικειμένων. Αν και οι παραδοσιακές μορφές διδασκαλίας δεν έχουν εγκαταλειφθεί πλήρως, μεγάλων μέρος των εκπαιδευτικών διαδικασιών βασίζονται πλέον στους ηλεκτρονικούς υπολογιστές και το διαδίκτυο. Βασικό χαρακτηριστικό των νέων αυτών μεθοδολογιών είναι η ευρεία χρήση πολυμέσων και υπερκειμένων. Ειδικότερα η ανάπτυξη του διαδικτύου, ενίσχυσε τις δυνατότητες για την διεξαγωγή εκπαίδευσης από απόσταση. Για να μπορούν χρησιμοποιηθούν οι τεχνολογικές δυνατότητες στην εκπαίδευση, θα πρέπει το εκπαιδευτικό υλικό να ψηφιοποιηθεί. Οι πρώτες εκτεταμένες προσπάθειες ανάπτυξης εκπαιδευτικού λογισμικού βασίστηκαν στη χρήση πολυμέσων. Μέσω κυρίως της εικόνας, του βίντεο και του ήχου, αυξήθηκε η αποδοτικότητα της διδασκαλίας. Αυτό επιτεύχθηκε κυρίως λόγω του ότι η κατανόηση των εκπαιδευτικών αντικειμένων έγινε κατά πολύ ευκολότερη μέσα από την οπτική τους αναπαράσταση. Από την άποψη του εκπαιδευτικού, το εξειδικευμένο λογισμικό αποτελεί ένα ισχυρό εργαλείο για την εξέλιξη των εκπαιδευτικών διαδικασιών, καθώς η γνώση μεταλαμπαδεύεται με μεγαλύτερο ρεαλισμό και παραστατικό – πρακτικό τρόπο, ακόμα και όταν αυτές αφορούν θεωρητικά αντικείμενα (Κόκκας, 2008).

Τα πολυμέσα μπορεί να εξυπηρετούν μία ποικιλία στόχων, όταν χρησιμοποιούνται στις εκπαιδευτικές διαδικασίες. Ανάλογα με το είδος των στόχων αυτών, διακρίνονται δύο κυρίαρχες τάσεις στην εκπαίδευση της σύγχρονης εποχής. Αυτές είναι:

- **Τεχνοκρατική Τάση:** Η τάση αυτή καθορίζει σαν βασικό στόχο της χρήσης των πολυμέσων στην εκπαίδευση την αύξηση της ελκυστικότητας του εκπαιδευτικού αντικειμένου. Σε πολλές περιπτώσεις, το αντικείμενο της διδασκαλίας δεν ανταποκρίνεται στα ενδιαφέροντα του εκπαιδευομένου. Αυτό δημιουργεί την αποστροφή προς τη διαδικασία και τελικά την έλλειψη ενσυνείδητης συμμετοχής του. Η εφαρμογή καινοτόμων μεθόδων παρουσίασης των εκπαιδευτικών αντικειμένων που βασίζονται στην εκτεταμένη χρήση πολυμέσων, την ενεργή συμμετοχή του εκπαιδευομένου στην εξέλιξη της διδασκαλίας, την παραστατική παρουσίαση της πληροφορίας, συμβάλλει αποφασιστικά στην μεγιστοποίηση της συμμετοχικότητας των εκπαιδευομένων.

- Μαθητοκεντρική Τάση: Κεντρικό στοιχείο της προσέγγισης αυτής είναι ο εκπαιδευόμενος. Η διδασκαλία προσαρμόζεται στις ιδιαίτερες απαιτήσεις της αύξησης του βαθμού κατά τον οποίο η γνώση μεταδίδεται και συντηρείται μέσω της εμπέδωσης της. Κομβικά στοιχεία αναγνωρίζονται:
 - η ανάκληση προηγούμενης γνώσης και εμπειρίας
 - η βελτίωση της ψυχολογικής και φυσικής κατάστασης των συμμετεχόντων
 - η εκμετάλλευση υπαρκτών δεξιοτήτων και δυνατοτήτων
 - η προσαρμογή του τρόπου που εξελίσσεται η εκπαιδευτική διαδικασία στις ιδιαίτερες απαιτήσεις των εκπαιδευομένων (Σπαντιδάκης, 2008).

Η εφαρμογή των πολυμέσων λειτούργησε σαν έναυσμα για το πέρασμα της εκπαίδευσης σε περισσότερο πρακτικές, εποπτικές και ρεαλιστικές προσεγγίσεις. Η ανάπτυξη των τεχνολογιών του διαδικτύου ήταν ο καταλύτης για την μετάβαση της εκπαίδευσης προς αυτόν τον προσανατολισμό. Προστέθηκαν προσιτά και ισχυρά εκπαιδευτικά εργαλεία στη φαρέτρα των εκπαιδευτικών προκειμένου να καταστήσουν την εκπαίδευση ακόμα περισσότερο παραστατική, ευχάριστη, ρεαλιστική. Ο εκπαιδευτικός μπορεί πλέον να επιλέγει από μία μεγάλη ποικιλία μεθοδολογιών, που βασίζονται στην οπτικοποίηση της πληροφορίας, προκειμένου να παρέχουν την γνώση στους διδασκομένους με τρόπους τέτοιους, που να γίνεται αμεσότερα και με μεγαλύτερη ακρίβεια κατανοητή από αυτούς. Σημαντικό ρόλο στη ραγδαία εξάπλωση των διαδικτυακών ευκολιών στην εκπαίδευση, διαδραμάτισε και το γεγονός ότι, σχετικά γρήγορα, ακόμα και οι προηγμένες τεχνολογίες του έγιναν προσιτές για το μεγαλύτερο μέρος του παγκοσμίου πληθυσμού¹ (Chien, 2012).

Επιπλέον, η ανάπτυξη της τεχνολογίας και σε επίπεδο υλικού, κατέστησε εφικτή τη χρήση των διαδικτυακών υπηρεσιών από μία ποικιλία συσκευών, πέρα των ηλεκτρονικών υπολογιστών, όπως είναι οι έξυπνες κινητές συσκευές ή οι συσκευές του διαδικτύου των πραγμάτων (Internet of Things – IoT). Η διεύρυνση του διαδικτύου σε πολλαπλά επίπεδα (σε επίπεδο εφαρμογών, μέσω πρόσβασης, προσανατολισμού χρήσης και προφίλ χρηστών), πολλαπλασίασε τον αριθμό των ανθρώπων, που πλέον είναι δεκτικοί με την εκτεταμένη χρήση των ηλεκτρονικών υπολογιστών σε κάθε πτυχή της δραστηριότητάς τους – και στην εκπαίδευση (Rahmani, Bayramov, & Kalejahi, 2022).

¹ Η προσέγγιση του υπολογιστικού νέφους κατέστησε την πρόσβαση σε επεξεργαστικούς, υπολογιστικούς και αποθηκευτικούς πόρους, υψηλής ποιότητας και ποσότητας, εφικτή και μέσα από φιλικές διεπαφές για ανθρώπους με μέση εξοικείωση με τη χρήση ηλεκτρονικών υπολογιστών. Εκτός αυτού απαιτούμενος εξοπλισμός έγινε ευρύτερα διαθέσιμος, η πρόσβαση στις απαραίτητες γνώσεις έγινε ευκολότερη και το οικονομικό κόστος σύνδεσης χαμηλότερο. Η πρόσβαση στο διαδίκτυο γίνεται αποδοτικά μέσω ενσύρματων ή ασύρματων ευρυζωνικών συνδέσεων.

Σε υψηλότερο επίπεδο, οι περισσότερες κυβερνήσεις και οι διεθνείς φορείς στάθηκαν αρωγοί σε κάθε προσπάθεια εφαρμογής της ευρυζωνικότητας και γενικότερα ενθάρρυνσης της πρόσβασης σε διαδικτυακές εφαρμογές. Γίνονται συνεχείς προσπάθειες σε όλο τον κόσμο για την διάδοση της χρήσης των ηλεκτρονικών υπολογιστών και των διαδικτυακών εφαρμογών και την εξοικείωση όλο και περισσότερων ανθρώπων με αυτές. Στον τομέα αυτό ιδιαίτερα σημαντική ήταν και η συμβολή των μέτρων περιορισμού της διασποράς του ιού Covid-19, καθώς πολλοί άνθρωποι αναγκάστηκαν, θέλοντας ή μη να εμπλακούν με τη χρήση διαδικτυακών εφαρμογών. Στον εκπαιδευτικό τομέα, την περίοδο αυτή πραγματοποιήθηκε μεγάλος αριθμός διαδικτυακών εκπαιδευτικών συνόδων σε μια μεγάλη ποικιλία εκπαιδευτικών ιδρυμάτων (Masalimova, et al., 2022).

2.2 Εκπαιδευτικό Λογισμικό

Η τεχνολογική πρόοδος συνέβαλλε στην εμφάνιση και εδραίωση μίας κατηγορίας λογισμικού, του εκπαιδευτικού. Το γεγονός ότι οι υπολογιστές εξασφαλίζουν την ενεργό συμμετοχή των μαθητών στην εκπαίδευση λαμβάνοντας υπόψη τις εξειδικευμένες απαιτήσεις τους από το εκπαιδευτικό περιβάλλον, έφερε στο προσκήνιο τη χρήση των ηλεκτρονικών υπολογιστών στην εκπαίδευση. Με τη χρήση τους, αναπτύσσονται μεθοδολογίες εκπαίδευσης που συνδυάζουν τις αρχές της αυτό-μάθησης με την τεχνολογία της πληροφορικής και των τηλεπικοινωνιών, που λαμβάνουν χώρα στο εκπαιδευτικό περιβάλλον. Η διδακτική διαδικασία αναβαθμίζεται μέσα από την διεύρυνση των κινήτρων των μαθητών, κυρίως ως προς την ταχύτητα της κατανόησης των εκπαιδευτικών αντικειμένων. Η εκπαίδευση με τη βοήθεια υπολογιστή μπορεί επίσης να περιγραφεί ως δραστηριότητες στις οποίες οι μαθητές αλληλεπιδρούν με προγραμματισμένα μαθήματα, ενώ ο δάσκαλος λειτουργεί ως οδηγός (ο υπολογιστής συνθέτει το εκπαιδευτικό περιβάλλον). Ο ηλεκτρονικός υπολογιστής χρησιμοποιείται ως εργαλείο διδασκαλίας που βοηθά τον δάσκαλο αναλαμβάνοντας ορισμένα από τα καθήκοντα και επίσης ως περιβάλλον όπου λαμβάνει χώρα η μάθηση. Όταν χρησιμοποιείται με αυτόν τον τρόπο, ο υπολογιστής δεν μπορεί να αντικαταστήσει τα άλλα στοιχεία του εκπαιδευτικού συστήματος όπως φυσικό εκπαιδευτικό υλικό (πχ βιβλία), συμμαθητές και δασκάλους.

Το εκπαιδευτικό λογισμικό που χρησιμοποιείται στην εκπαίδευση, έχει κάνει τη διδασκαλία πιο διασκεδαστική. Αποτελείται από στοιχεία που επιτρέπουν την ανάπτυξη περιβαλλόντων πολλαπλής μάθησης και οδηγούν τους μαθητές στη μάθηση (Şahin, 2021).

Σύμφωνα με ότι έχει αναφερθεί παραπάνω, ο όρος εκπαιδευτικό λογισμικό αναφέρεται σε οποιονδήποτε είδος λογισμικού σχεδιάστηκε ειδικά για εκπαιδευτικούς σκοπούς ή χρησιμοποιείται για τέτοιους σκοπούς (ακόμα και αν είναι γενικής χρήσης). Σκοπός του είναι να παρέχει εκπαιδευτικό περιεχόμενο, να υποστηρίζει τη διαδικασία της μάθησης και να παρέχει εργαλεία για τη βελτίωση της εκπαιδευτικής διαδικασίας (Bulman, 2016). Αποτελεί μια εναλλακτική μορφή προγράμματος σπουδών, που παρέχεται με τη βοήθεια της ψηφιακής τεχνολογίας. Αναφέρεται σε οποιαδήποτε εφαρμογή υπολογιστή που βελτιώνει τη μάθηση. Περιλαμβάνει λογισμικό διαχείρισης τάξης, συστήματα πληροφοριών μαθητών, λογισμικό γλώσσας, λογισμικό αναφοράς και άλλα. Αυτό σημαίνει ότι δεν περιορίζεται στην εκπαιδευτική δραστηριότητα αλλά και σε κάθε πτυχή της εκπαίδευσης (λειτουργική ή υποστηρικτική). Ως προς τη διδασκαλία, με τη βοήθεια της προηγμένης τεχνολογίας, καταλήγει να απλοποιεί τη δραστηριότητα. Κάνει τη μάθηση διαδραστική και καθηλωτική για τους μαθητές, καθώς και πολύ πιο διαχειρίσιμη και συνεργατική για τους εκπαιδευτικούς.

Η χρησιμότητα του εκπαιδευτικού λογισμικού έχει γίνει αντιληπτή και εκτιμηθεί από μεγάλη μερίδα των επαγγελματιών της εκπαίδευσης και της τεχνολογίας. Πολλές εταιρείες σε όλο τον κόσμο που επικεντρώνονται στην ανάπτυξη προσαρμοσμένου εκπαιδευτικού λογισμικού για σχολεία ή μαθητές ενός συγκεκριμένου κύκλου σπουδών ή για πιο γενικές σκοπιμότητες. Τα χαρακτηριστικά και τα επίπεδα πολυπλοκότητας διαφέρουν από λογισμικό σε λογισμικό και από προσανατολισμό σε προσανατολισμό.

2.3 Κατηγορίες εκπαιδευτικού λογισμικού

Ανάλογα με τον τρόπο που προσεγγίζουν το εκπαιδευτικό αντικείμενο, τα μέσα που χρησιμοποιούνται και την μέθοδο με την οποία συμμετέχουν εκπαιδευτικοί και εκπαιδευόμενοι, το εκπαιδευτικό λογισμικό μπορεί να διακριθεί στις ακόλουθες κατηγορίες (Şahin, 2021):

- **Διαδραστικά παιχνίδια:** Πρόκειται για λογισμικό, που μέσα από τη διαδραστική συμμετοχή του εκπαιδευόμενου σε ένα σενάριο παιχνιδιού, παρουσιάζουν σε αυτόν το εκπαιδευτικό αντικείμενο. Συνδυάζουν την απόκτηση γνώσεων και δεξιοτήτων με τη διασκέδαση και την αλληλεπίδραση με τον χρήστη. Ο συνδυασμός αυτός ενισχύει τα μαθησιακά αποτελέσματα κυρίως μέσα από την ενσυνείδητη συμμετοχή του εκπαιδευόμενου στην εκπαιδευτική διαδικασία.
- **Εκπαιδευτικά βίντεο:** Τα εκπαιδευτικά βίντεο, δημιουργούνται ειδικά για εκπαιδευτικούς σκοπούς και περιέχουν πλούσιο περιεχόμενο γνώσεων και

πληροφοριών, ενταγμένα σε ένα οπτικοποιημένο σενάριο. Η πλοκή τους σκηνοθετείται με τρόπο τέτοιο ώστε να εμπλέκει νοερά τον εκπαιδευόμενο στην διδασκαλία. Προβάλλονται είτε σε ομάδες εκπαιδευομένων είτε ατομικά και σε ορισμένες περιπτώσεις μπορεί να συνοδεύονται από τη συμμετοχή του διδάσκοντα.

- Πολυμεσικά βιβλία: Αποτελούν εξέλιξη των συμβατικών βιβλίων. Είναι ψηφιακές εκδόσεις που περιλαμβάνουν πολυμέσα, υπερκείμενο και διαδραστικά στοιχεία. Η μορφή τους είναι τέτοια που επιτρέπει στους εκπαιδευόμενους να έχουν μια πιο αποτελεσματική εμπειρία μάθησης μέσα από την οπτικοποίηση της πληροφορίας και την αλληλεπίδραση με το εκπαιδευτικό περιεχόμενο.
- Εικονική πραγματικότητα: Τα εκπαιδευτικά σενάρια, δίνουν την ευκαιρία στους εκπαιδευμένους να έχουν μια ρεαλιστική εμπειρία με το εκπαιδευτικό περιεχόμενο. Μέσα από την αλληλεπίδραση με εικονικά περιβάλλοντα και αντικείμενα, έχουν τη δυνατότητα να συμμετέχουν σε μία ρεαλιστική
- παρουσίαση του εκπαιδευτικού αντικειμένου.
- Πλατφόρμες ηλεκτρονικής μάθησης: Συνήθως είναι διαδικτυακές πλατφόρμες που παρέχουν ένα ολοκληρωμένο περιβάλλον μάθησης από απόσταση. Τα εκπαιδευτικά αντικείμενα παρέχονται σε μορφές, που μπορούν να υποστηριχθούν από το διαδίκτυο (υπερκείμενο, βίντεο). Χρησιμοποιούνται τόσο για σύγχρονη όσο για ασύγχρονη τηλεκπαίδευση. Έχουν την δυνατότητα επίσης να υποστηρίζουν τη συνολική εκπαιδευτική διαδικασία.
- Εικονικά εργαστήρια: Είναι λογισμικό που παρέχει τη δυνατότητα εκτέλεσης εργαστηριακών δραστηριοτήτων, που πραγματοποιούνται σε ηλεκτρονικό υπολογιστή, τις περισσότερες φορές στο πλαίσιο τηλεκπαίδευσης. Πλέον αυτού σε ορισμένες περιπτώσεις παρέχεται η δυνατότητα για χρήση πραγματικών συσκευών από απόσταση, μέσω του διαδικτύου. Τα αποτελέσματα των δραστηριοτήτων γνωστοποιούνται από διαδικτυακές διεπαφές, είτε σε ομάδες εκπαιδευομένων είτε ατομικά. Με τον τρόπο αυτό οι εκπαιδευόμενοι αποκτούν τη δυνατότητα να απολαμβάνουν την εμπειρία του εργαστηρίου με χαμηλό κόστος.
- Λογισμικό Επαυξημένης Πραγματικότητας: Με το λογισμικό αυτού του είδους παρέχεται η δυνατότητα στους εκπαιδευόμενους να χειρίζονται και να αλληλεπιδρούν με τρισδιάστατα (3-D) εικονικά αντικείμενα. Το λογισμικό αυτό παράγει τρισδιάστατα εικονικά αντικείμενα, δημιουργώντας ένα διαδραστικό περιβάλλον ,στο οποίο οι εκπαιδευόμενοι αλληλεπιδρούν με

εικονικά αντικείμενα μέσα στον πραγματικό κόσμο. Η εμπειρία των εκπαιδευομένων με την διάδραση αυτή, ενισχύει την αληθοφάνεια της εκπαίδευσης.

- Εργαλεία Αξιολόγησης: Είναι εφαρμογές που χρησιμοποιούνται για την αξιολόγηση των εκπαιδευομένων. Σκοπός τους είναι είτε να αξιολογηθούν οι ίδιοι οι εκπαιδευόμενοι, είτε η αποτελεσματικότητα των εκπαιδευτικών διαδικασιών, είτε και τα δύο.

2.4 Εκπαιδευτικά Παιχνίδια

Υπάρχουν πολυάριθμοι ορισμοί των παιχνιδιών υπολογιστή. Υπάρχουν επίσης πολλές διαφορετικές εκτιμήσεις για την αξία τους και το ρόλο τους στην ανάπτυξη της ανθρώπινης προσωπικότητας. Σε γενικές γραμμές, ένα παιχνίδι είναι ένα σύστημα στο οποίο οι παίκτες εμπλέκονται σε έναν τεχνητό ανταγωνισμό, που ορίζεται από κανόνες. Το αποτέλεσμα του παιχνιδιού είναι μετρήσιμο και συχνά χρησιμοποιείται για την αξιολόγηση των συμμετεχόντων. Βασικά χαρακτηριστικά στοιχεία των παιχνιδιών είναι:

- Οι κανόνες που διέπουν την εξέλιξη τους και τον τρόπο συμμετοχής των παικτών τους.
- Οι στόχοι που τίθενται ως προς το αποτέλεσμα τους. Τους στόχους αυτούς καλούνται να προσεγγίσουν οι εκπαιδευόμενοι και σχετίζονται άμεσα ή έμμεσα με το εκπαιδευτικό αντικείμενο.
- Το αποτέλεσμα τους που προκύπτει από τη συμμετοχή των εκπαιδευομένων.
- Οι μεθοδολογίες ανατροφοδότησης των παικτών τους.
- Ο τρόπος με τον οποίο συγκρούονται οι συμμετέχοντες (μεταξύ τους ή με τη ροή του παιχνιδιού).
- Το βαθμό του ανταγωνισμού μεταξύ των συμμετεχόντων.
- Η πρόκληση που ελκύει τους συμμετέχοντες.
- Ο τρόπος με τον οποίο αλληλεπιδρούν και επικοινωνούν οι συμμετέχοντες.
- Το σενάριο και ο τρόπος με τον οποίο αναπαρίσταται.

Τα τελευταία χρόνια αναδείχθηκε ο όρος σοβαρά παιχνίδια (serious games). Τα σοβαρά παιχνίδια, περιλαμβάνουν διαφορετικούς τύπους εκπαιδευτικών παιχνιδιών αλλά δεν περιορίζονται αποκλειστικά σε αυτόν τον προσανατολισμό. Τα serious games μπορεί να χρησιμοποιηθούν για μία ποικιλία σκοπιμοτήτων όπως:

- Ενίσχυση των προωθητικών ενεργειών μέσω της δυνατότητας τους να ελκύουν το καταναλωτικό κοινό.
- Κοινωνική βελτίωση ως μέσο ανάπτυξης της συνεργασίας μεταξύ των ανθρώπων.

Ένας ορισμός αναφέρει ότι πρόκειται για διανοητικό διαγωνισμό, που πραγματοποιείται με ηλεκτρονικό υπολογιστή σύμφωνα με συγκεκριμένους κανόνες. Χρησιμοποιεί την προτίμηση του ανθρώπου στην ψυχαγωγία για να προωθήσει την εκπαίδευση αλλά και άλλες επιδιώξεις και στρατηγικούς επικοινωνιακούς στόχους. Σε πολλές περιπτώσεις αποτελούν μια συνέχεια μεταξύ παιχνιδιών και βιωματικών περιβαλλόντων για έναν κοινό σκοπό². Σε κάθε περίπτωση, τα σοβαρά παιχνίδια προσελκύουν τον χρήστη και συμβάλλουν στην επίτευξη ενός καθορισμένου σκοπού εκτός της καθαρής ψυχαγωγίας. Ο χρήστης μπορεί να γνωρίζει τον εκπαιδευτικό σκοπό και χαρακτήρα του παιχνιδιού ή και όχι. Ο σκοπός ενός παιχνιδιού μπορεί να διατυπωθεί είτε από τον ίδιο τον χρήστη ή από τον σχεδιαστή του, πράγμα που σημαίνει ότι επίσης ένα εμπορικό παιχνίδι, που χρησιμοποιείται για μη ψυχαγωγικούς σκοπούς, μπορεί να θεωρηθεί σοβαρό παιχνίδι. Αξίζει να σημειωθεί ότι η χρήση τεχνολογιών παιχνιδιών και οπτικοποίησης, προσομοιώσεων και εικονικών κόσμων για σκοπούς πέρα από την ψυχαγωγία μπορεί να συμπεριληφθεί και σε αυτόν τον ορισμό. Ενώ αυτός ο ορισμός περιλαμβάνει μη ψηφιακά παιχνίδια, οι περισσότερες αναφορές σε σοβαρά παιχνίδια είναι στην πραγματικότητα σε ψηφιακά (κυρίως διαδικτυακά) παιχνίδια.

Η επιτυχία των serious games, έχει εντείνει την έρευνα, αναζητώντας την εξέλιξη σε δύο κατευθύνσεις: την διεύρυνσή του προσανατολισμού τους και την αναβάθμιση της αποτελεσματικότητάς τους. Αυτό που είναι δύσκολο να επιτευχθεί σε αυτού του είδους τα παιχνίδια είναι η ισορροπία μεταξύ της ψυχαγωγικής τους φύσης και του εκπαιδευτικού τους χαρακτήρα. Η υπερβολή στη ενσωμάτωση ψυχαγωγικών στοιχείων μπορεί να αποβούν εις βάρος της εκπαιδευτικής ικανότητας των παιχνιδιών. Υπάρχει έλλειψη γνώσης σχετικά με τις επιπτώσεις της χρήσης παιχνιδιών σε εκπαιδευτικά πλαίσια, καθώς ο συγκεκριμένος τομέας έχει τα μοναδικά του προβλήματα όσον αφορά τις μεθόδους, την εστίαση και τα σχετικά ερευνητικά ερωτήματα. Ο σχεδιασμός των serious games μπορεί να εξυπηρετούν διαφορετικές προσεγγίσεις μάθησης (συμπεριφορισμός, γνωστικισμός, εποικοδομισμός και κοινωνικοπολιτισμικές). Αν και η αποτελεσματικότητα των παιχνιδιών είναι γενικά μεγάλη, δεν είναι πάντα η καταλληλότερη μέθοδος. Για το λόγο αυτό κατά τη φάση της σχεδίασης της χρειάζεται να γίνεται ενδελεχής μελέτη της σκοπιμότητας της εκπαίδευσης και του τρόπο με

² Για παράδειγμα, πολλές σοβαρές εφαρμογές παιχνιδιών χρησιμοποιούν τις τεχνολογίες που συνήθως σχετίζονται με τα παιχνίδια υπολογιστή και όχι το στοιχείο παιχνιδιού. Αυτές οι εφαρμογές αναφέρονται ως εικονικά περιβάλλοντα και ψηφιακά μέσα χωρίς παραδοσιακά χαρακτηριστικά παιχνιδιού.

τον οποίο το παιχνίδι θα μπορέσει να την ικανοποιήσει (Backlund & Hendrix, 2013). Κατά τη σχεδίαση και ανάπτυξη των serious games εξετάζονται δύο βασικά στοιχεία, ώστε η μαθησιακή διδασκαλία να είναι αρκούντως αποδοτική:

- Η κατάλληλη διδασκαλική προσέγγιση
- Ο δυναμισμός στη ροή του σεναρίου.

Τα στοιχεία αυτά, μαζί με την εργασία που πρέπει να συνοδεύει τη διδασκαλία, καθορίζουν την επιτυχία του εκπαιδευτικού παιχνιδιού. Η εργασία αυτή θα πρέπει να περιλαμβάνει τα ακόλουθα βασικά σημεία:

- Καθορισμός των στόχων της χρήσης τους
- Ανάπτυξη ενός διδακτικού οδηγού στον οποίο να προσδιορίζεται ποιες έννοιες πρέπει να ενισχυθούν και ποιες ικανότητες μπορούν να χρησιμοποιηθούν για την προώθηση.
- Καθορισμός πλαισίου που πρέπει να ακολουθήσει ο δάσκαλος και του μοντέλου αξιολόγησης του βαθμού επίτευξης των στόχων. Υπό αυτή την έννοια, πρέπει να υπάρχει συνεχής ανατροφοδότηση για την αντιμετώπιση τυχόν σημείων που μπορεί να αποθαρρύνουν τον χρήστη (δεν μπορεί να συνεχίσει το παιχνίδι, έχει δυσκολίες ή έχει ξεκινήσει άσχημα).
- Δημιουργία ενός τελικού εργαλείου ανατροφοδότησης για ολόκληρη την ομάδα για να συγκρίνει τα αποτελέσματα του κάθε χρήστη με αυτά της υπόλοιπης ομάδας.
- Αξιολόγηση των παρεχόμενων ευκαιριών για αλληλεπίδραση. Πώς και πότε το παιχνίδι πυροδοτεί στιγμές προβληματισμού στην εμπειρία του χρήστη και ποιους μηχανισμούς αξιολόγησης μπορεί να χειριστεί για τη μέτρηση της γνώσης και της κατανόησης που αποκτά ο εκπαιδευόμενος.
- Η χρήση σοβαρών παιχνιδιών ως μεθόδου μάθησης στην αρχική εκπαίδευση πρέπει να λαμβάνει υπόψη τα προφίλ των μαθητών και ιδιαίτερα την προηγούμενη εμπειρία τους στον τομέα που μελετήθηκε (Noem & Máximo, 2014).

Οι δεξιότητες και οι ικανότητες που συνήθως αναπτύσσουν αυτά τα παιχνίδια είναι η βιωσιμότητα, η ομαδική εργασία, η αλληλεγγύη, η καινοτομία, η δημιουργικότητα, η επίλυση προβλημάτων, η συνεχής βελτίωση, η ενεργειακή απόδοση, η μαθηματική ακρίβεια, η πρωτοβουλία, η επίτευξη στόχων, ο προσανατολισμός στο αποτέλεσμα, η ευελιξία και η εργασία με το περιβάλλον.

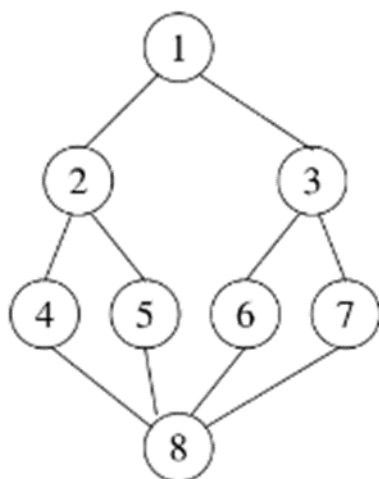
3 Αλγόριθμοι Γράφων

3.1 Ορισμός – Χαρακτηριστικά – Είδη Γράφων

Οι γράφοι είναι οπτικές αναπαραστάσεις μαθηματικών μοντέλων που αντιστοιχούν σε σχέσεις μεταξύ αντικειμένων. Αποτελούνται από:

- Ένα σύνολο κόμβων (συνήθως συμβολίζεται με V), που αντιστοιχούν στα αντικείμενα τα οποία συσχετίζονται.
- Ένα σύνολο συνδέσεων μεταξύ των κόμβων (συνήθως συμβολίζεται με E) που αντιστοιχούν στις συσχετίσεις των προαναφερομένων αντικειμένων.

Ένας γράφος μπορεί να χρησιμοποιηθεί για να μοντελοποιήσει ποικίλες δομές που εμφανίζονται στον πραγματικό κόσμο, κυρίως διαφόρων ειδών δίκτυα (οδικά, κοινωνικά κ.λπ.), γενεαλογικά δέντρα, δένδρα συγκρίσεων ή αποφάσεων κα. Η μοντέλοποίηση συσχετίσεων με γράφους χρησιμοποιείται στην επίλυση διαφόρων προβλημάτων. Η ικανότητα τους αυτή να μοντελοποιούν την περιγραφή προβλημάτων, τους κάνουν να έχουν πολλές εφαρμογές σε διάφορους τομείς όπως ο έλεγχος των κοινωνικών δικτύων, η ανάλυση δεδομένων, η βιολογία, η ρομποτική, η τηλεπικοινωνία, η αναζήτηση πληροφορίας, η διαχείριση δικτύων κα. Ένα παράδειγμα γράφου φαίνεται στην παρακάτω εικόνα.



Εικόνα 1: Παράδειγμα γράφου

Στον παραπάνω γράφο το σύνολο των κόμβων είναι $V = \{1,2,3,4,5,6,7,8\}$, ενώ το σύνολο των ακμών είναι $E = \{(1,2),(1,3),(2,4),(2,5),(3,6),(3,7),(4,8),(5,8),(6,8),(7,8)\}$.

Ανάλογα με τη μορφή τους, οι γράφοι διακρίνονται σε:

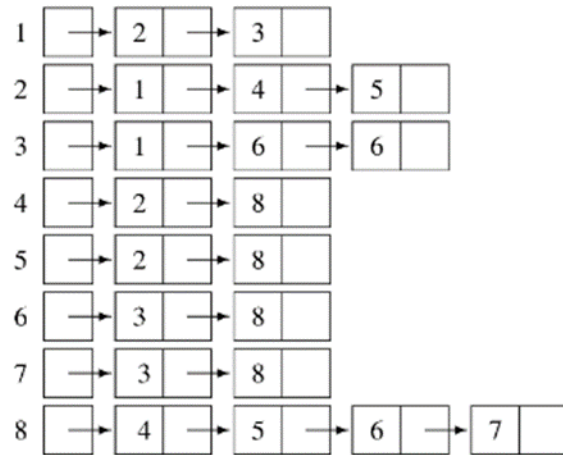
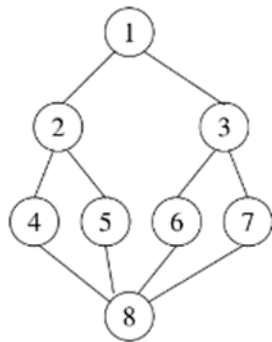
- Μη κατευθυνόμενος γράφος: Στους γράφους αυτού του είδους δεν προσδιορίζεται κατεύθυνση για την ακμή.
- Κατευθυνόμενος γράφος: Οι γράφοι αυτοί περιλαμβάνουν ακμές στις οποίες καταγράφεται η κατεύθυνση από τον κόμβο αφετηρία στον κόμβο προορισμού.
- Απλός γράφος: Οι γράφοι αυτού του είδους δεν περιλαμβάνουν πολλαπλές ακμές μεταξύ των ίδιων κόμβων και δεν περιλαμβάνουν και βρόχους (δηλαδή ακμές που και τα δύο άκρα τους τέμνουν τον ίδιο κόμβο).
- Πολλαπλοί γράφοι: Σε αυτούς τους γράφους, μπορεί να υπάρχουν πολλαπλές ακμές μεταξύ των ίδιων δύο κόμβων.
- Γράφος με βάρη: Οι γράφοι αυτού του είδους περιλαμβάνουν ακμές στις οποίες αντιστοιχίζονται βάρη³.
- Δέντρο: Ένα δέντρο είναι ένας γράφος χωρίς κύκλους. Σε έναν τέτοιο γράφο κάθε κόμβος έχει ακριβώς έναν γονέα, εκτός από τον κόμβο στην κορυφή του δέντρου που δεν έχει γονέα και ο οποίος καλείται ρίζα.

Οι γράφοι μπορούν να αναπαρασταθούν με διάφορους τρόπους, ανάλογα με τις απαιτήσεις της εφαρμογής για την οποία χρησιμοποιούνται. Οι βασικότεροι τρόποι αναπαράστασης είναι οι εξής:

- Λίστα γειτνίασης: Με την αναπαράσταση αυτή, κάθε κόμβος του γράφου αντιστοιχίζεται σε μια λίστα που περιέχει τους γείτονές του. Οι λίστες γειτνίασης είναι κατάλληλες όταν ζητούμενο είναι να αποφεύγεται η σπατάλη αποθηκευτικού χώρου (ειδικότερα στην αποθήκευση αραιών γράφων⁴). Στην επόμενη εικόνα φαίνεται ένα παράδειγμα αναπαράστασης γράφου με λίστες γειτνίασης.

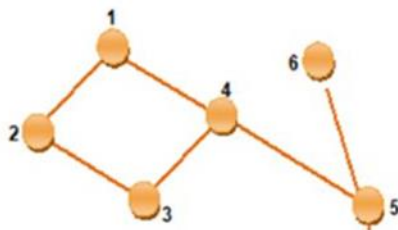
³ Τα βάρη μπορεί να αντιστοιχούν σε διάφορα μεγέθη όπως κόστος, κέρδος, απόσταση, πιθανότητα, ένταση κτλ.

⁴ Αραιοί είναι οι γράφοι οι οποίοι περιλαμβάνουν πολύ λιγότερες ακμές σε σχέση με τον μέγιστο αριθμό ακμών που θα μπορούσαν να περιέχουν (ο οποίος είναι n^2 αν το πλήθος των κόμβων είναι n).



Εικόνα 2: Παράδειγμα αναπαράστασης γράφου με λίστες γειτνίασης

- Πίνακας γειτνίασης: Με αυτού του είδους την αναπαράσταση, χρησιμοποιείται ένας διδιάστατος πίνακας, όπου οι γραμμές και οι στήλες αντιστοιχούν στους κόμβους του γράφου. Οι τιμές στα κελιά του πίνακα δηλώνουν την ύπαρξη (αν είναι μη μηδενικές ή μη απειρίζουσες) ή μη ύπαρξη ακμής (σε διαφορετική περίπτωση) μεταξύ δύο κόμβων. Αυτή η αναπαράσταση είναι κατάλληλη όταν ζητούμενο είναι η ταχεία λήψη της πληροφορίας του αν δύο κόμβοι συνδέονται. Στην επόμενη εικόνα φαίνεται ένα παράδειγμα αναπαράστασης γράφου με πίνακα γειτνίασης.



	1	2	3	4	5	6
1	0	1	0	1	0	0
2	1	0	1	0	0	0
3	0	1	0	1	0	0
4	1	0	1	0	1	0
5	0	0	0	1	0	1
6	0	0	0	0	1	0

Εικόνα 3: Παράδειγμα αναπαράστασης γράφου με πίνακα γειτνίασης

3.2 Βασικοί αλγόριθμοι γράφων

Οι γράφοι χρησιμοποιούνται ευρέως στην μοντελοποίηση προβλημάτων τα οποία στην συνέχεια επιλύονται με κατάλληλους αλγορίθμους. Συνοπτικά, οι γνωστότεροι αλγόριθμοι γράφων και τα προβλήματα που επιλύουν έχουν όπως αναφέρονται στις παρακάτω παραγράφους.

3.2.1 Αλγόριθμος αναζήτησης πρώτα κατά βάθος (Depth-First Search - DFS)

Πρόκειται για έναν αλγόριθμο που εξετάζει το σύνολο των κόμβων ενός γράφου. Η κεντρική του ιδέα είναι να ξεκινάει από έναν κόμβο αφετηρία και να μετακινείται από κόμβο σε κόμβο, απομακρυνόμενος όσο μακρύτερα είναι δυνατό. Οι πιο δημοφιλείς εφαρμογές του είναι η εύρεση συνεκτικών συνιστωσών ενός γράφου⁵, η δημιουργία τοπολογικής ταξινόμησης⁶ ή η αναζήτηση κύκλων⁷.

3.2.2 Αλγόριθμος αναζήτησης πρώτα κατά πλάτος (Breadth-First Search - BFS)

Όπως ο DFS, έτσι και ο BFS χρησιμοποιείται για την εξέταση του συνόλου των κόμβων ενός γράφου. Ξεκινώντας από έναν κόμβο αφετηρία, επισκέπτεται κάθε κόμβο ενός γράφου σε επίπεδα. Για κάθε κόμβο εξαντλεί τις επισκέψεις στους γείτονες του κόμβου, πριν προχωρήσει στο επόμενο επίπεδο. Χρησιμοποιείται συχνότερα για την εύρεση συντομότερων μονοπατιών.

3.2.3 Αλγόριθμος Dijkstra

Ο αλγόριθμος Dijkstra χρησιμοποιείται για την εύρεση του συντομότερου μονοπατιού από έναν κόμβο προς οποιοδήποτε άλλο κόμβο του, σε έναν γράφο με μη αρνητικού βάρους ακμών. Συνήθως χρησιμοποιείται σε προβλήματα δρομολόγησης, μεταφοράς και γενικότερα όποτε απαιτείται η εύρεση του βέλτιστου μονοπατιού από άποψης κόστους. Η πολυπλοκότητα του εξαρτάται από τις δομές δεδομένων που χρησιμοποιούνται και είναι της τάξης του $O(n^2)$, όπου n το πλήθος των γράφων.

3.2.4 Αλγόριθμος Bellman Ford

Ο αλγόριθμος αυτός επίσης χρησιμοποιείται για την εύρεση των μονοπατιών ελαχίστου κόστους από μία αφετηρία προς όλους τους υπολοίπους κόμβους του γράφου. Χαρακτηριστικό του είναι ότι λειτουργεί ακόμα και αν οι ακμές έχουν αρνητικά βάρη. Χρησιμοποιείται για τους ίδιους σκοπούς που χρησιμοποιείται και ο Dijkstra. Ο αλγόριθμος

⁵ Συνεκτική συνιστώσα ενός γράφου είναι ένας υπόγράφος που περιλαμβάνει κόμβους από τους οποίους υπάρχουν μονοπάτια προς όλους τους υπόλοιπους κόμβους του υπογράφου.

⁶ Μία τοπολογική ταξινόμηση των κόμβων ενός γράφου είναι μία γραμμική διάταξη τους με τρόπο τέτοιο που όλες οι ακμές του γράφου να έχουν κοινή φορά.

⁷ Κύκλος σε ένα γράφο είναι ένα μονοπάτι με κοινή αφετηρία και τερματισμό.

εξελίσσεται εξετάζοντας μονοπάτια πρώτα μήκους ένα, μετά μήκους δύο και εν τέλει μήκους $n-1$. Η πολυπλοκότητα του αλγορίθμου είναι $O(nm)$, όπου n είναι το πλήθος των κόμβων και m είναι ο πλήθος των ακμών.

3.2.5 Αλγόριθμος Floyd Warshal

Με τον αλγόριθμο αυτό εντοπίζονται οι διαδρομές ελαχίστους κόστους μεταξύ όλων των ζευγών κόμβων του γράφου. Πρόκειται για αλγόριθμο δυναμικού προγραμματισμού, όπου τα αποτελέσματα σε κάθε βήμα του αλγορίθμου κρατούνται σε έναν πίνακα, προκειμένου να χρησιμοποιηθούν σε επόμενο στάδιο. Ο αλγόριθμος εξελίσσεται εξετάζοντας μέσω ποιου κόμβου τα μονοπάτια είναι χαμηλότερου κόστους. Η πολυπλοκότητα του αλγορίθμου είναι $O(n^3)$.

3.2.6 Αλγόριθμοι Prim και Kruskal

Οι αλγόριθμοι αυτοί χρησιμοποιούνται για την εύρεση του ελάχιστου συνδετικού δέντρου ενός γράφου. Με την ολοκλήρωση των αλγορίθμων αυτών, σχηματίζεται ένα δένδρο που προκύπτει από τον γράφο στον οποίο εφαρμόζονται. Το άθροισμα των βαρών των ακμών του δένδρου είναι το ελάχιστο δυνατό. Ο αλγόριθμός Prim έχει πολυπλοκότητα $O(n^2)$, ενώ ο αλγόριθμος Kruskal έχει πολυπλοκότητα $O(m \log n)$, όπου m το πλήθος των ακμών και n το πλήθος των κόμβων.

3.3 Αλγόριθμοι αναζήτησης σε γράφους

3.3.1 Αναζήτηση πρώτα κατά βάθος

Ο αλγόριθμος αναζήτησης πρώτα κατά βάθος (DFS) είναι ένας από τους πιο βασικούς αλγορίθμους γράφων. Ο σκοπός του αλγορίθμου DFS είναι να επισκεφθεί όλους τους κόμβους ενός γράφου. Η εξέλιξη του μπορεί να περιγράψει από μία αναδρομική διαδικασία ή από μία διαδικασία που χρησιμοποιεί μία στοίβα.

Κατά την αναδρομική εκδοχή του αλγορίθμου, όταν εξετάζεται ένας κόμβος, εκκινεί μία διαδικασία αναζήτησης από κάθε ένα γείτονα του, αν δεν έχει δεχθεί επίσκεψη μέχρι εκείνη τη φάση. Ο αλγόριθμος ολοκληρώνεται όταν ολοκληρώνονται οι αναδρομικές του κλήσεις και έχουν ελεγχθεί όλοι οι κόμβοι του γράφου. Ο έλεγχος για τον έλεγχο όλων των κόμβων του γράφου πραγματοποιείται με τη χρήση ενός μονοδιάστατου πίνακα, μήκους όσο το πλήθος των

κόμβων του, στα κελία του οποίο τίθεται False αν ο αντίστοιχος κόμβος δεν έχει ελεγχθεί ή True διαφορετικά. Η εξέλιξη της αναδρομικής εκδοχής περιγράφεται από τον παρακάτω ψευδοκώδικα (Louridas, 2019):

Algorithm 1: Recursive DFS

Data: G : The graph stored in an adjacency list

root: The starting node

Result: Prints all nodes inside the graph in the *DFS* order

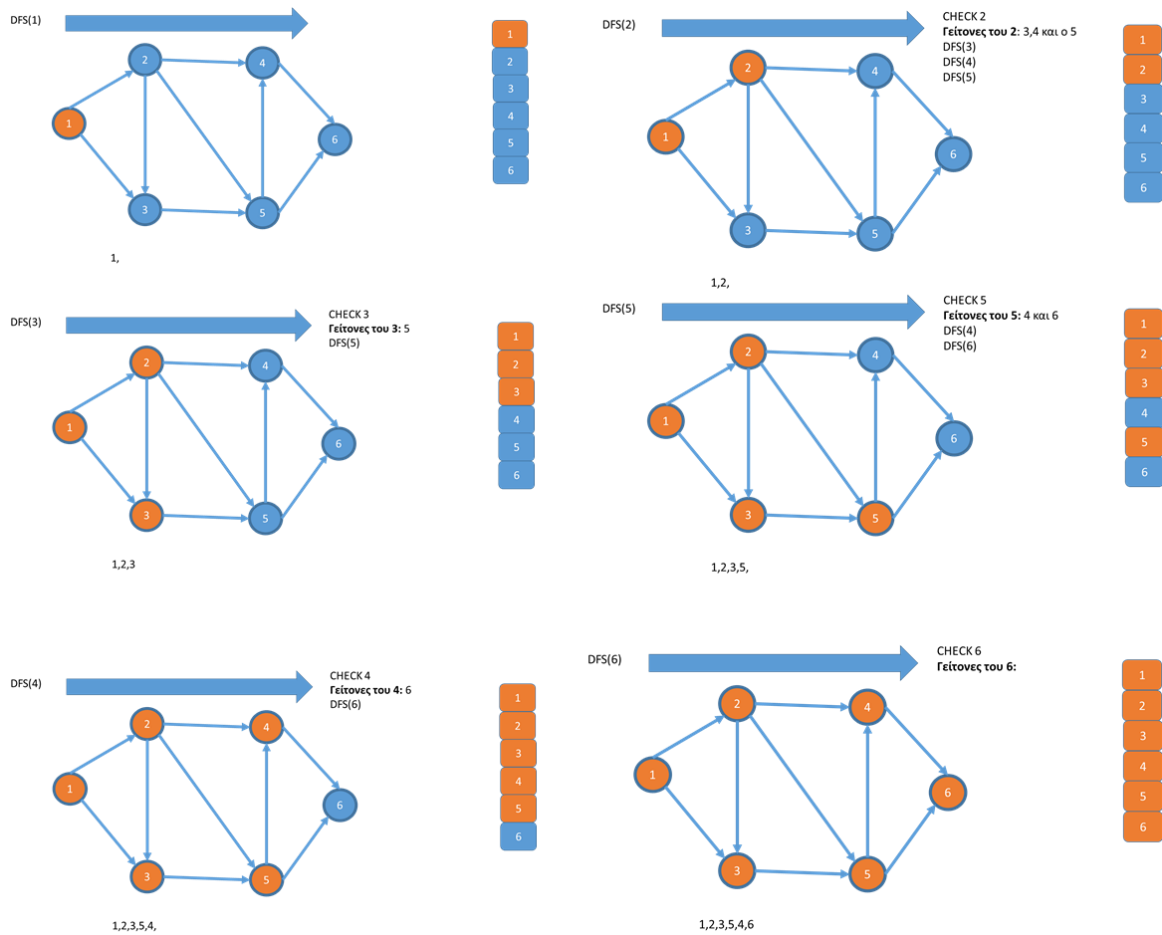
$visited \leftarrow \{false\}$;

$DFS(root)$;

Function $DFS(u)$:

```
    if  $visited[u] = true$  then
    |   return;
    end
    print( $u$ );
     $visited[u] \leftarrow true$ ;
    for  $v \in G[u].neighbors()$  do
    |    $DFS(v)$ ;
    end
end
```

Στο παρακάτω σχήμα παρουσιάζεται σχηματικά η εξέλιξη του αλγορίθμου με στην αναδρομική εκδοχή.



Εικόνα 4: Εξέλιξη της αναδρομικής εκδοχής του DFS

Ο αλγόριθμος μπορεί να εξελιχθεί και σε μία άλλη εκδοχή, κατά την οποία χρησιμοποιείται μία στοίβα. Η εξέλιξη αυτή έχει όπως παρακάτω:

- Επιλογή αρχικού κόμβου: Επιλέγεται ο κόμβος – αφετηρία της εξερεύνησης του γράφου. Αυτός ο κόμβος μπαίνει στην αρχή μιας στοίβας (stack).
- Επανάληψη: Όσο η στοίβα δεν είναι κενή, επαναλαμβάνουμε τα εξής βήματα:
- Λαμβάνεται ο κόμβος που βρίσκεται στην κορυφή της στοίβας
- Ελέγχεται αν ο κόμβος έχει ήδη ελεγχθεί. Σε θετική περίπτωση γίνεται μετάβαση στην επόμενη επανάληψη.
- Σημειώνεται ο κόμβος σαν ελεγμένος (χρησιμοποιείται και σε αυτή την περίπτωση μονοδιάστατος πίνακας).
- Εξερευνώνται οι γείτονες του τρέχοντος κόμβου. Κάθε γείτονα που δεν έχει ελεγχθεί, τοποθετείται στη στοίβα.
- Αν η στοίβα είναι κενή, ο αλγόριθμος ολοκληρώνεται (Sryheni, 2023).

Ο ψευδοκώδικας της εκδοχής αυτής του αλγορίθμου φαίνεται παρακάτω (Louridas, 2019):

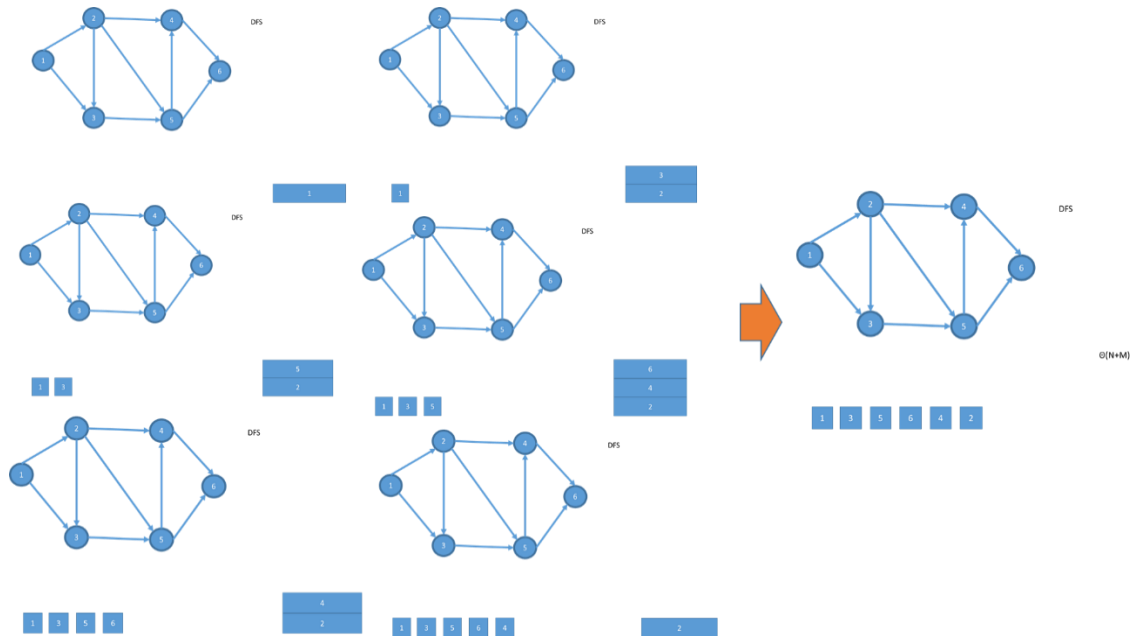
Algorithm 2: Iterative DFS

Data: G : The graph stored in an adjacency list
root: The starting node

Result: Prints all nodes inside the graph in the *DFS* order

```
visited ← {false};
stack ← {};
stack.push(root);
while ¬stack.empty() do
  u ← stack.top();
  stack.pop();
  if visited[u] = true then
    continue;
  end
  print(u);
  visited[u] ← true;
  for v ∈ G[u] do
    if visited[v] = false then
      DFS(v);
    end
  end
end
end
```

Σχηματικά η εξέλιξη της εκδοχής αυτής του αλγορίθμου φαίνεται στο παρακάτω σχήμα.



Εικόνα 5: Η εξέλιξη του DFS με χρήση στοίβας

Ο αλγόριθμος DFS χρησιμοποιείται για διάφορους σκοπούς, μεταξύ των οποίων:

- Εύρεση συνδεδεμένων συνιστωσών: Μπορεί να χρησιμοποιηθεί για τον προσδιορισμό των συνδεδεμένων συνιστωσών ενός γράφου. Αυτό είναι

χρήσιμο για τον έλεγχο της συνδεσιμότητας του γράφου και για την εύρεση διαχωρισμένων τμημάτων στο γράφο.

- Αναζήτηση μονοπατιών: Ο DFS μπορεί να χρησιμοποιηθεί για την αναζήτηση μονοπατιών μεταξύ δύο κόμβων. Εξερευνά το γράφο με βάση τις ακμές και επισκέπτεται πρώτα τα βαθύτερα μονοπάτια προτού προχωρήσει στα επόμενα.
- Αναζήτηση λύσης: Ο DFS μπορεί να χρησιμοποιηθεί για την αναζήτηση λύσης σε προβλήματα που μπορούν να αναπαρασταθούν ως γράφοι. Επισκέπτεται τους κόμβους του γράφου με έναν τακτικό τρόπο και εντοπίζει μια λύση αν υπάρχει.
- Δημιουργία τοπολογικής ταξινόμησης: Αφορά τη γραμμική τοποθέτηση των κόμβων του γράφου με τρόπο τέτοιο που οι ακμές του γράφου να είναι προσανατολισμένοι σε κοινή κατεύθυνση. Η τοπολογική ταξινόμηση χρησιμοποιείται στην αναζήτηση των κρίσιμων διαδικασιών στην ολοκλήρωση ενός έργου, τον υπολογισμό βέλτιστων διαδρομών κα.
- Ανακάλυψη κύκλων σε ένα γράφο (Brookshear, 2012).

Συνοψίζοντας, ο αλγόριθμος αναζήτησης πρώτα κατά βάθος είναι ένας απλός και αποτελεσματικός τρόπος εξερεύνησης ενός γράφου. Χρησιμοποιεί τη στοίβα για να αποθηκεύσει τους κόμβους που πρέπει να επισκεφθεί, και ακολουθεί μια προτεραιότητα "βάθους" κατά την εξερεύνηση. Η πολυπλοκότητα του αλγορίθμου αναζήτησης πρώτα κατά βάθος (DFS) εξαρτάται από το μέγεθος του γράφου και την αναπαράστασή του. Για ένα γράφο με N κόμβους και E ακμές, η πολυπλοκότητα χρόνου του DFS είναι $O(N + E)$, δηλαδή γραμμική ως προς τον αριθμό των κόμβων και των ακμών.

3.3.2 Αναζήτηση πρώτα κατά πλάτος

Ο αλγόριθμος αναζήτησης πρώτα κατά πλάτος (BFS - Breadth-First Search) είναι ένας αλγόριθμος αναζήτησης που χρησιμοποιείται για την εξερεύνηση ενός γράφου. Ο αλγόριθμος επισκέπτεται όλους τους γείτονες ενός κόμβου πριν προχωρήσει στους γείτονες των γειτόνων του. Αυτό τον κάνει να εξελίσσεται κατά επίπεδα του γράφου. Ειδικότερα εξελίσσεται στις ακόλουθες φάσεις:

- Αρχικοποίηση: Δημιουργία μιας ουράς (queue) για να αποθηκεύονται οι κόμβοι που πρέπει να εξερευνηθούν. Επιπλέον δημιουργούνται δύο γραμμικές δομές:

- Μονοδιάστατος πίνακας μεγέθους ίσου με το πλήθος των κόμβων που περιλαμβάνει ο γράφος. Κάθε κελί έχει αποθηκευμένη μία λογική τιμή, ανάλογα αν ο κόμβος έχει ελεγχθεί ή όχι.
- Μονοδιάστατος πίνακας μεγέθους ίσου με το πλήθος των κόμβων που περιλαμβάνει ο γράφος. Κάθε κελί έχει αποθηκευμένη μία λογική τιμή, ανάλογα αν ο κόμβος βρίσκεται εντός της ουράς ή όχι.
- Εισαγωγή του κόμβου εκκίνησης στην ουρά.
- Σημείωση του αρχικού κόμβου ως ελεγμένου και εντός της ουράς στους αντίστοιχους πίνακες.
- Επανάληψη μέχρι η ουρά να γίνει άδεια:
- Ανάκτηση του πρώτου κόμβου από την ουρά.
- Ενημέρωση του πίνακα με τις παρουσίες στην ουρά (για την αφαίρεση του τρέχοντος κόμβου).
- Επίσκεψη του κόμβου (εκτέλεση ενεργειών πάνω στον κόμβο) και ενημέρωση του αντίστοιχου πίνακα.
- Προσθήκη στην όλων των γειτόνων του κόμβου που δεν έχουν επισκεφθεί ακόμη και δεν βρίσκονται ήδη σε αυτή.
- Επανάληψη των παραπάνω βημάτων μέχρι η ουρά να αδειάσει.

Ο ψευδοκώδικας του αλγορίθμου έχει όπως παρακάτω:

Algorithm: Graph breadth-first search

$\text{BFS}(G, \text{node}) \rightarrow \text{visited}$

Input: $G = (V, E)$, a graph

node , the starting vertex in G

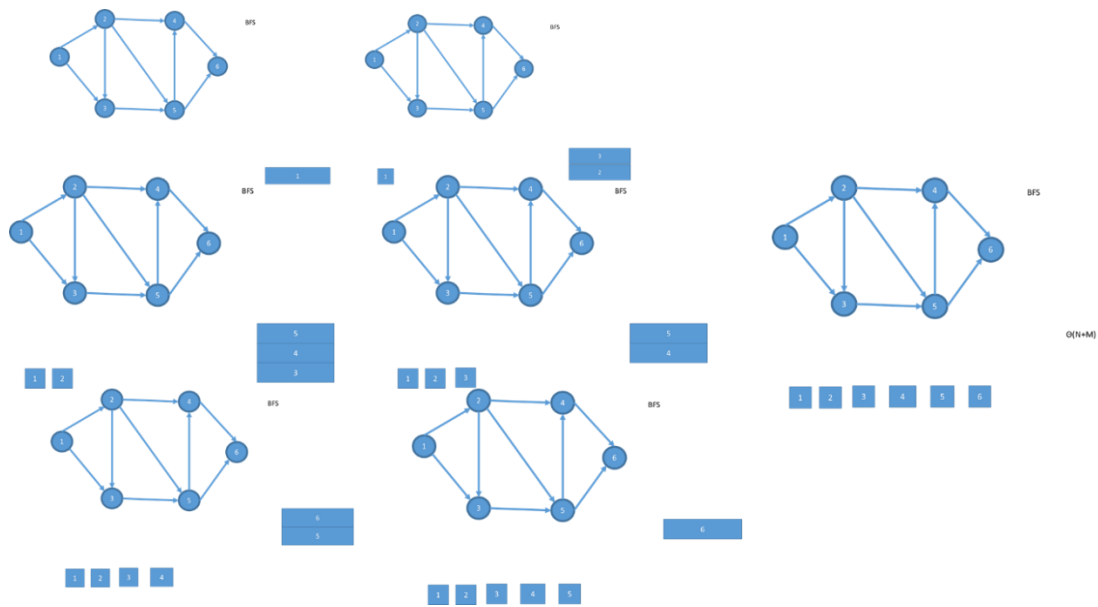
Output: visited , an array of size $|V|$ such that $\text{visited}[i]$ is TRUE if we have visited node i , FALSE otherwise

```
1  $Q \leftarrow \text{CreateQueue}()$ 
2  $\text{visited} \leftarrow \text{CreateArray}(|V|)$ 
3  $\text{inqueue} \leftarrow \text{CreateArray}(|V|)$ 
4 for  $i \leftarrow 0$  to  $|V|$  do
5      $\text{visited}[i] \leftarrow \text{FALSE}$ 
6      $\text{inqueue}[i] \leftarrow \text{FALSE}$ 
7  $\text{Enqueue}(Q, \text{node})$ 
8  $\text{inqueue}[\text{node}] \leftarrow \text{TRUE}$ 
9 while not  $\text{IsQueueEmpty}(Q)$  do
10      $c \leftarrow \text{Dequeue}(Q)$ 
11      $\text{inqueue}[c] \leftarrow \text{FALSE}$ 
12      $\text{visited}[c] \leftarrow \text{TRUE}$ 
13     foreach  $v$  in  $\text{AdjacencyList}(G, c)$  do
14         if not  $\text{visited}[v]$  and not  $\text{inqueue}[v]$  then
15              $\text{Enqueue}(Q, v)$ 
16              $\text{inqueue}[v] \leftarrow \text{TRUE}$ 
17 return  $\text{visited}$ 
```

Εικόνα 6: Ο ψευδοκώδικας του αλγορίθμου BFS

Είναι αξιοσημείωτη η ομοιότητα του ψευδοκώδικα αυτού με τον αντίστοιχο του DFS. Μοναδική διαφορά τους αποτελεί το γεγονός ότι ο DFS χρησιμοποιεί στοιβά και ο BFS ουρά.

Σχηματικά η εξέλιξη του αλγορίθμου φαίνεται στο παρακάτω σχήμα.



Εικόνα 7: Σχηματική αναπαράσταση της εξέλιξης του BFS

Ο αλγόριθμος BFS εξερευνά το γράφο σε επίπεδα. Αρχικά, εξερευνά τον αρχικό κόμβο και στη συνέχεια μεταβαίνει στους γείτονές του. Έπειτα, εξερευνά τους γείτονες των γειτόνων του αρχικού κόμβου, και έτσι συνεχίζει σταδιακά σε κάθε επίπεδο του γράφου. Εξασφαλίζει ότι οι κόμβοι εξερευνώνται σύμφωνα με την απόστασή τους από τον αρχικό κόμβο. Αυτό σημαίνει ότι πρώτα εξερευνώνται οι κόμβοι που είναι σε απόσταση 1 από τον αρχικό κόμβο, έπειτα οι κόμβοι που είναι σε απόσταση 2, και ούτω καθεξής.

Ο αλγόριθμος BFS είναι ευρέως χρησιμοποιούμενος και έχει πολλές εφαρμογές, όπως:

- Εύρεση συντομότερου μονοπατιού μεταξύ δύο κόμβων.
- Εύρεση όλων των κόμβων που απέχουν ένα βήμα από έναν αρχικό κόμβο.
- Εύρεση όλων των συνδεδεμένων συνιστωσών ενός γράφου.
- Επίλυση προβλημάτων όπως το πρόβλημα εύρεσης των νησιών σε έναν χάρτη το πρόβλημα του εμποδίου λαβυρίνθου κα.

Η πολυπλοκότητα του αλγορίθμου BFS είναι $O(N + E)$, όπου N είναι ο αριθμός των κόμβων και E είναι ο αριθμός των ακμών του γράφου. Αυτό σημαίνει ότι ο αλγόριθμος επισκέπτεται κάθε κόμβο και ακμή μία φορά. Επιπλέον, η πολυπλοκότητα χώρου του BFS είναι $O(N)$, καθώς απαιτείται χώρος για την αποθήκευση των κόμβων στην ουρά.

3.4 Διδασκαλία των αλγορίθμων γράφων

Η ικανότητα των γράφων να μοντελοποιούν μία ευρεία γκάμα κατηγοριών προγραμμάτων, τους καθιστά ένα χρήσιμο εργαλείο για τη διδασκαλία πολλών και ποικίλων

θεμάτων. Πολλές φορές για να γίνει κατανοητή η χρήση τους στη μοντελοποίηση προβλημάτων, είναι απαραίτητη η κατανόηση των χαρακτηριστικών τους. Στις περιπτώσεις αυτές επιβάλλεται η διδασκαλία τους.

Η διδασκαλία των γράφων μπορεί να γίνει με διάφορους τρόπους. Κάθε φορά επιλέγεται εκείνος που ταιριάζει καλύτερα με το γνωστικό υπόβαθρο των εκπαιδευομένων και τις απαιτήσεις τους ως προς το εύρος της εμβάθυνσης στην απόκτηση γνώσεων. Σε γενικές γραμμές, η διδασκαλία τους απαιτεί τα παρακάτω:

- Κατανόηση της θεωρητικής βάσης της έννοιας των γράφων: Οι εκπαιδευόμενοι θα πρέπει να είναι σε θέση να κατανοούν τη σημασία των βασικών δομικών στοιχείων των γράφων (οι κόμβοι και οι ακμές) και τις βασικές έννοιες των μεταξύ τους σχέσεων (γείτονες, βαθμός κόμβου, κατευθύνσεις). Στη συνέχεια θα πρέπει να μπορούν να διακρίνουν τις κύριες ιδιότητες των γράφων, όπως είναι η απλότητα, η συνεκτικότητα, οι κύκλοι κα. Είναι σημαντική η παράθεση παραδειγμάτων για την κατανόηση κάθε έννοιας.
- Οπτικοποίηση: Η φύση των γράφων είναι τέτοια όπου η καταλληλότερη μεθοδολογία για την περιγραφή τους είναι η εικόνα. Επομένως είναι σημαντικό στη διδασκαλία να χρησιμοποιείται μεγάλος αριθμός από γραφικές αναπαραστάσεις για την οπτικοποίηση των γράφων και της σημασιολογίας τους. Αυτό μπορεί να γίνει με τη χρήση διαγραμμάτων, σχεδίων ή καθοδήγησης των μαθητών στην κατασκευή γράφων με φυσικά αντικείμενα. Μπορεί επίσης να πραγματοποιηθεί με τη χρήση κατάλληλου εκπαιδευτικού λογισμικού. Η οπτικοποίηση βοηθά στην κατανόηση της δομής των γράφων και των σχέσεων μεταξύ των κόμβων τους.
- Εφαρμογές και προβλήματα: Η παρουσίαση πραγματικών προβλημάτων που μπορούν να μοντελοποιηθούν και να επιλυθούν με τη χρήση γράφων, μπορεί να δώσει στους μαθητές δυνατότητες για επεξηγηματικούς συνειρμούς. Τέτοια ζητήματα είναι η μελέτη των κοινωνικών δικτύων, των δικτύων οδικών μεταφορών ή την ανάλυση δικτύων δεδομένων.
- Αλγοριθμική σκέψη: Θα πρέπει οι μαθητές να επιχειρούν να χρησιμοποιούν τους αλγορίθμους γράφων για την επίλυση πραγματικών προβλημάτων. Και σε αυτή την εκπαιδευτική απαίτηση, σημαντικό ρόλο μπορεί να παίζει η οπτικοποίηση της εξέλιξης των αλγορίθμων με εικόνα, βίντεο ή γραφικά. Έμφαση χρειάζεται να δίνεται στην επιλογή των καλύτερων αλγορίθμων με βάση τις πολυπλοκότητες τους.

- Εργαλεία λογισμικού: Χρησιμοποιήστε λογισμικά και εργαλεία που επιτρέπουν την οπτικοποίηση και την επεξεργασία γράφων, όπως γραφικά περιβάλλοντα προγραμματισμού ή ειδικά λογισμικά πακέτα για την ανάλυση γράφων. Αυτά τα εργαλεία μπορούν να βοηθήσουν τους μαθητές να εξερευνήσουν, να αναλύσουν και να επιλύσουν προβλήματα γράφων με πιο αποτελεσματικό τρόπο.
- Εφαρμογή σε πρακτικά προβλήματα: Οι εκπαιδευόμενοι θα πρέπει να ενθαρρύνονται να χρησιμοποιούν τους αλγορίθμους γράφων για την επίλυση μίας ποικιλίας προβλημάτων. Η πρακτική εμπλοκή με την υλοποίηση των αλγορίθμων θα τους βοηθήσει στη βαθιά κατανόηση τους.
- Ασκήσεις και προβλήματα: Ασκήσεις και προβλήματα που αφορούν γράφους, συμβάλλουν στην ανάπτυξη δεξιοτήτων στην ανάλυση και την επίλυση προβλημάτων γράφων.

Η διδασκαλία των γράφων πρέπει να συνδυάζει θεωρητική κατανόηση, οπτικοποίηση, αλγοριθμική σκέψη, εφαρμογές σε πρακτικά προβλήματα και ασκήσεις. Με αυτόν τον τρόπο, οι μαθητές θα αποκτήσουν μια πλήρη εικόνα των γράφων και θα μπορούν να εφαρμόσουν τις γνώσεις τους σε πραγματικά και θεωρητικά προβλήματα. Είναι σημαντικό να υποστηρίζεται η δημιουργική σκέψη και η συνεργατική εργασία κατά τη διδασκαλία των γράφων (όπως ισχύει και για κάθε εκπαιδευτική διαδικασία). Οι μαθητές μπορούν να ενθαρρυνθούν να σκέφτονται εκτεταμένα, να προτείνουν νέες ιδέες και να συνεργάζονται για την επίλυση προβλημάτων με χρήση γράφων. Οι εκπαιδευτικοί μπορούν επίσης να παρέχουν προκλήσεις και πρακτικά παραδείγματα που εμπνέουν τους μαθητές να ανακαλύψουν την ισχύ των γράφων και την εφαρμογή τους σε σενάρια του πραγματικού κόσμου.

Τέλος, η διδασκαλία των γράφων μπορεί να ενθαρρύνει την ανάπτυξη των δεξιοτήτων προγραμματισμού και την κατανόηση των αλγορίθμων. Οι μαθητές μπορούν να μάθουν πώς να αναπτύξουν κώδικα που να εκτελεί αλγορίθμους γράφων, να λύνει προβλήματα και να αξιοποιεί τις δυνατότητες των γράφων για την ανάλυση δεδομένων και τη λήψη αποφάσεων. Συνολικά, η διδασκαλία των γράφων μπορεί να ενθαρρύνει τη δημιουργικότητα, την ανάπτυξη κριτικής σκέψης, την ανταλλαγή ιδεών και την ανάπτυξη δεξιοτήτων προγραμματισμού.

4 Η πλατφόρμα ανάπτυξης παιχνιδιών Unity

4.1 Γενική Περιγραφή

Η επέκταση της χρηστικότητας των ηλεκτρονικών παιχνιδιών πέρα του ψυχαγωγικού προσανατολισμού, έχει τονώσει το ενδιαφέρον της κοινότητας των προγραμματιστών για την ανάπτυξη τους. Συνέπεια αυτού ήταν να προταθούν διάφορα περιβάλλοντα ανάπτυξης παιχνιδιών προορισμένων για μία ποικιλία λειτουργικών συστημάτων και συσκευών. Ένα από τα περισσότερο δημοφιλή περιβάλλοντα αυτού του είδους είναι το UNITY. Πρόκειται για ένα ολοκληρωμένο περιβάλλον ανάπτυξης παιχνιδιών, εφαρμογών εικονικής πραγματικότητας (Virtual Reality - VR) και επαυξημένης πραγματικότητας (Augmented Reality - AR). Αναπτύσσεται και διανέμεται δωρεάν από την Unity Technologies.

Βασικό χαρακτηριστικό στοιχείο της πλατφόρμας είναι το γεγονός ότι αποτελεί ένα ολοκληρωμένο πλαίσιο εργασίας, που επιτρέπει στους προγραμματιστές να δημιουργούν διαδραστικά παιχνίδια, εφαρμογές και περιβάλλοντα εικονικής πραγματικότητας και επαυξημένης πραγματικότητας, παρέχοντας σημαντικές διευκολύνσεις για ταχεία ανάπτυξη. Με τη χρήση του UNITY, είναι δυνατή η δημιουργία παιχνιδιών πλούσια σε γραφικά και εφαρμογών που μπορούν να τρέξουν σε μία μεγάλη ποικιλία περιβαλλόντων, όπως ηλεκτρονικούς υπολογιστές, κονσόλες παιχνιδιών, κινητές συσκευές. Αυτό ακριβώς το χαρακτηριστικό του, το έχει καταστήσει εξαιρετικά δημοφιλές ανάμεσα στους προγραμματιστές.

Το UNITY περιλαμβάνει ένα γραφικό περιβάλλον ανάπτυξης (IDE), όπου οι προγραμματιστές μπορούν να αναπτύξουν τον κώδικά των έργων τους, να διαχειριστούν τα γραφικά τους, τον ήχο, τις φυσικές αλληλεπιδράσεις μεταξύ των αντικειμένων, που έχουν ρόλο στα σενάρια των παιχνιδιών και να προσαρμόσουν τις ρυθμίσεις για την αναπαραγωγή στην επιθυμητή πλατφόρμα. Επιπλέον, το UNITY παρέχει ένα ισχυρό σύστημα φυσικής μοντελοποίησης, υποστήριξη πολυμέσων (π.χ. γραφικά, ήχος, βίντεο), δυνατότητες ανίχνευσης κίνησης και ανίχνευσης χειρονομιών, καθώς και δικτύωσης για παιχνίδια στα οποία μπορεί να συμμετέχουν περισσότεροι του ενός χρηστών από διαφορετικές γεωγραφικές τοποθεσίες. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν το UNITY για να δημιουργήσουν εικονικά περιβάλλοντα, να προσθέσουν αληθοφανή γραφικά, να δημιουργήσουν προηγμένες φυσικές αλληλεπιδράσεις, να προγραμματίσουν λογική συμπεριφορά, να διαχειριστούν ήχο και να αξιοποιήσουν διάφορες άλλες δυνατότητες για να δημιουργήσουν εντυπωσιακές εμπειρίες για τους χρήστες.

Η ανάπτυξη των προγραμμάτων που ρυθμίζουν την εξέλιξη των παιχνιδιών μπορεί αν γίνει σε διάφορες γλώσσες προγραμματισμού, όπως C#, JavaScript και Boo. Επίσης, υπάρχουν

εκτεταμένες δυνατότητες προσαρμογής και επέκτασης μέσω πρόσθετων (add-ons) και πακέτων (packages) που παρέχονται από την κοινότητα του UNITY. Μεγάλος αριθμός αυτών παρέχονται δωρεάν. Τα παραπάνω στοιχεία του, το τοποθετούν στις κορυφαίες θέσεις των προτιμήσεων στη βιομηχανία ανάπτυξης εφαρμογών παιχνιδιών, εικονικής πραγματικότητας, επαυξημένης πραγματικότητας, εκπαιδευτικών εφαρμογών, αρχιτεκτονικής σχεδίασης και αναπαράστασης προϊόντων και υπηρεσιών. Ενισχυτικό της δεκτικότητας, που επιδεικνύει η προγραμματιστική κοινότητα στο Unity είναι το γεγονός ότι ο κώδικας που αναπτύσσεται για την υλοποίηση ενός έργου, μπορεί να εξαχθεί σε ποικίλες μορφές για χρήση σε αντίστοιχες πλατφόρμες.

Η υποστήριξη που παρέχεται στους προγραμματιστές για τη χρήση της πλατφόρμας είναι πλήρης και ικανή να καλύψει κάθε απαίτηση. Βασική πηγή υποστήριξης είναι η ομάδα ανάπτυξης του μέσω της επίσης ιστοσελίδας του UNITY⁸. Σε αυτή παρέχονται:

- Πλούσιο εκπαιδευτικό υλικό σε ποικίλες μορφές
- Οργανωμένα ηλεκτρονικά μαθήματα
- Οδηγίες ανάπτυξης συγκεκριμένων στοιχείων
- Παραδείγματα εφαρμογών
- Πόροι που μπορούν να ρυθμιστούν κατάλληλα και να ενσωματωθούν σε έργα
- Το ίδιο το λογισμικό του UNITY σε διάφορες εκδόσεις

Στην ίδια ιστοσελίδα ο επισκέπτης μπορεί να ενημερωθεί για όλες τις νέες εξελίξεις στον τομέα της ανάπτυξης εφαρμογών με UNITY. Εκτός από την επίσημη, υποστήριξη για την ανάπτυξη και συντήρηση εφαρμογών παρέχεται και ανεπίσημα από την ευρεία διαδικτυακή κοινότητα που έχει αναπτυχθεί γύρω από το UNITY. Κάθε ερώτημα που υποβάλλεται στην κοινότητα, τυγχάνει ταχείας απάντησης. Παράλληλα, όλο και περισσότεροι εκπαιδευτικοί οργανισμοί εντάσσουν στα προγράμματα σπουδών τους το UNITY.

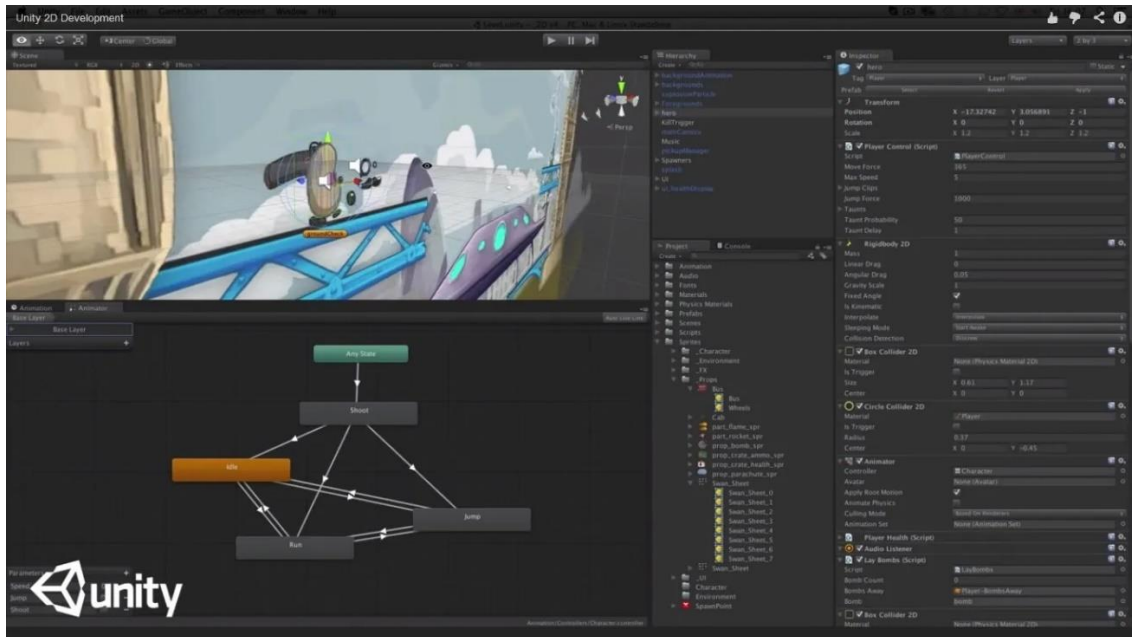
4.2 Το Περιβάλλον Ανάπτυξης

Το περιβάλλον ανάπτυξης του UNITY IDE διακρίνεται σε δύο κύριους τομείς:

- Το γραφικό περιβάλλον διαμόρφωσης των γραφικών στοιχείων
- Το περιβάλλον ανάπτυξης κώδικα για τον προγραμματισμό της συμπεριφοράς των αντικειμένων που συμμετέχουν στην εξέλιξη του παιχνιδιού.

⁸ <https://unity.com/>

Στην παρακάτω εικόνα παρουσιάζεται το περιβάλλον διαμόρφωσης των γραφικών στοιχείων των εφαρμογών.



Εικόνα 8: Οθόνη διαμόρφωσης γραφικών στοιχείων εφαρμογών

Ο προγραμματιστής μπορεί να επιλέγει τα γραφικά στοιχεία που θέλει να εντάξει στην εφαρμογή του, να τα παραμετροποιήσει και να ρυθμίσει τη συμπεριφορά τους, ανάλογα με το σενάριο του παιχνιδιού. Τα στοιχεία αυτά μπορεί:

- Να τα δημιουργήσει ο ίδιος χρησιμοποιώντας και διαμορφώνοντας βασικά αντικείμενα που παρέχει η πλατφόρμα.
- Να τα ανακτήσει από το διαδίκτυο (είτε από την επίσημη ιστοσελίδα του UNITY είτε από τρίτες πηγές) και είτε να τα ενσωματώσει αυτούσια στα έργα του είτε αφού να διαμορφώσει σύμφωνα με τις επιθυμίες του.

Όλα τα στοιχεία που ενσωματώνονται στο έργο, φαίνονται σε λίστα στη διεπαφή της εφαρμογής. Ο προγραμματιστής μπορεί να χρησιμοποιεί κατάλληλες φόρμες (που εμφανίζονται με κλικ πάνω στο αντίστοιχο στοιχείο του παιχνιδιού) για να προσαρμόσει τη μορφή και τη συμπεριφορά του. Ο προγραμματιστής μπορεί να επέμβει στην αναπαραγωγή ήχων, την εμφάνιση εικόνων και γραφικών, στην προσαρμογή του μοντέλου κίνησης των μοντέλων που χρησιμοποιούνται. Μπορεί επίσης να αλλάξει τις ρυθμίσεις των αντικειμένων κατά την ώρα εκτέλεσης της εφαρμογής, προκειμένου να έχει σαφή αντίληψη του πώς οι αλλαγές επιδρούν στην εξέλιξη της. Το γεγονός ότι όλα αυτά μπορούν να γίνουν με τη χρήση φιλικών διεπαφών κάνει τον προγραμματιστή ταχύτερο στην ανάπτυξη των εφαρμογών και ακριβέστερο στην παραγωγή των αποτελεσμάτων τους.

Για να ρυθμίσει τη συμπεριφορά των αντικειμένων του παιχνιδιού αλλά και για να καθορίσει τον τρόπο εξέλιξης του παιχνιδιού, ο προγραμματιστής καθορίζει τις αντιδράσεις της εφαρμογής σε συμβάντα. Προγραμματίζει δηλαδή τα συμβάντα που πραγματοποιούνται κατά την εξέλιξη των εφαρμογών, γράφοντας μικρά (τις περισσότερες φορές) σενάρια σε γλώσσα C# ή Javascript ή Boo. Η ανάπτυξη των σεναρίων αυτών ακολουθεί το αντικειμενοστραφές μοντέλο. Ο προγραμματιστής έχει την ελευθερία να επιλέξει τον editor που τον εξυπηρετεί. Τα πιο συνηθισμένα συμβάντα που προγραμματίζονται είναι:

- Εκκίνηση (Start): Αφορά την εκκίνηση της εφαρμογής ή μίας φάσης αυτής.
- Εξέλιξη (Update): Αφορά την ολοκλήρωση ενός πλαισίου (frame) της εξέλιξης της εφαρμογής.
- Σύγκρουση (Collision): Αφορά την σύγκρουση αντικειμένων που συμμετέχουν στην εξέλιξη του παιχνιδιού.

Παράλληλα, ο προγραμματιστής μπορεί να ενσωματώσει στις εφαρμογές του πλήρεις γραφικές διεπαφές χρήστη για επέμβαση του τελικού χρήστη στην εξέλιξη του ή για την ενημέρωση του.

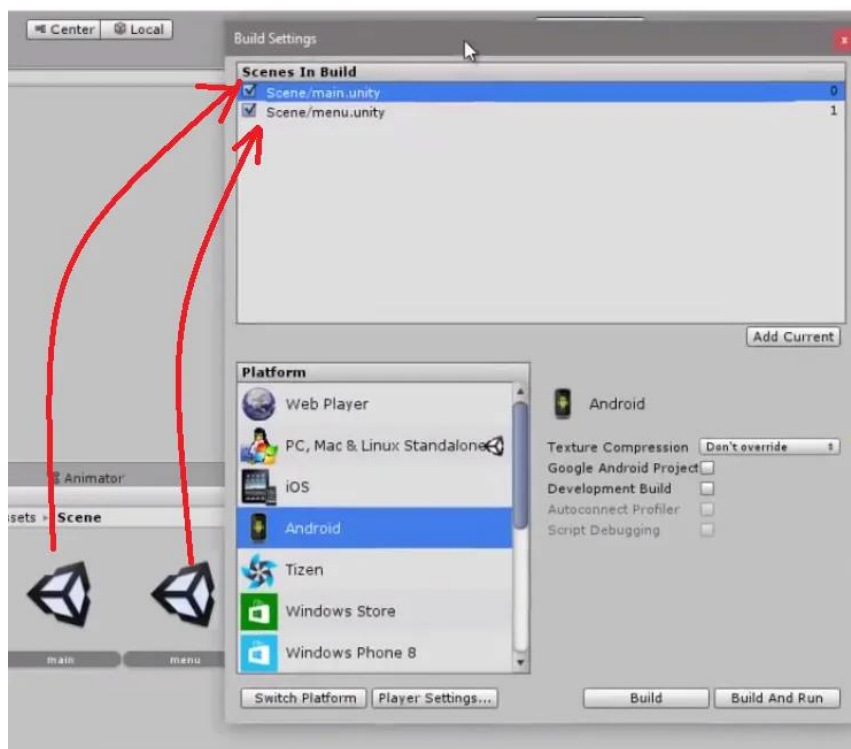
Στην παρακάτω εικόνα φαίνεται ένα παράδειγμα κώδικα προγραμματισμού συμβάντων σε περιβάλλον UNITY.

```
21 {
22     if (other.transform == player)
23     {
24         m_IsPlayerInRange = false;
25     }
26 }
27
28 void Update ()
29 {
30     if (m_IsPlayerInRange)
31     {
32         Vector3 direction = player.position - transform.position + Vector3.up;
33         Ray ray = new Ray(transform.position, direction);
34         RaycastHit raycastHit;
35
36         ray.Cast(raycastHit, out raycastHit)
37
38         if (raycastHit.collider.transform == player)
39         {
40             gameEnding.CaughtPlayer ();
41         }
42     }
43 }
44 }
45
46
```

Εικόνα 9: Παράδειγμα κώδικα Unity

Ένα ακόμα σημαντικό στοιχείο του UNITY είναι ο Unity Recorder. Πρόκειται για ένα εργαλείο που χρησιμοποιείται για την εγγραφή σκηνών και gameplay από το περιβάλλον του Unity. Επιτρέπει τη δημιουργία βίντεο ή εικόνων από την εκτέλεση ενός παιχνιδιού ή μιας εφαρμογής στο Unity. Παρέχει δυνατότητα για διάφορες επιλογές για την εγγραφή σκηνών, συμπεριλαμβανομένης της εγγραφής προβολής της κάμερας, των κινήσεων των χαρακτήρων, των αλληλεπιδράσεων με τα αντικείμενα και άλλων στοιχείων της σκηνής. Ο προγραμματιστής ρυθμίζει την ποιότητα των γραφικών, τον τύπο του αρχείου που θα αποθηκευτεί το αρχείο που παράγεται και άλλες παραμέτρους κατά την εγγραφή. Μπορεί να χρησιμοποιηθεί για διάφορους σκοπούς (όπως η δημιουργία gameplay trailers, η δημιουργία εκπαιδευτικού υλικού, η καταγραφή bugs ή προβλημάτων κατά την εκτέλεση ενός παιχνιδιού, η παρουσίαση έργων). Με το Unity Recorder, μπορεί να καταγραφεί η εμπειρία του παιχνιδιού ή της εφαρμογής τους προκειμένου να παρουσιαστεί εύκολα σε τρίτους. Μπορεί να ενεργοποιηθεί και να ρυθμιστεί μέσω του περιβάλλοντος εργασίας του Unity και μπορεί να προσαρμοστεί στις ανάγκες και τις προτιμήσεις του κάθε χρήστη.

Η εξαγωγή των έργων σε εκτελέσιμα αρχεία γίνεται μέσα από μία απλή διαδικασία κατά την οποία επιλέγεται σε πρώτο χρόνο το περιβάλλον στο οποίο θα τρέξει και σε δεύτερο χρόνο οι εξειδικευμένοι παράμετροι του. Στην επόμενη εικόνα παρουσιάζεται η οθόνη παραγωγής εκτελέσιμου αρχείου.



Εικόνα 10: Παραγωγή εκτελέσιμου αρχείου

5 Εφαρμογή οπτικοποίησης αλγορίθμων γράφων

5.1 Σχεδίαση του παιχνιδιού

Το παιχνίδι σχεδιάστηκε για να εξελιχθεί σε δύο οθόνες. Στην πρώτη ο χρήστης μπορεί να επιλέξει τις παραμέτρους του παιχνιδιού και στη δεύτερη παρουσιάζεται ο καμβάς της εξέλιξης του παιχνιδιού.

5.2 Περιγραφή του παιχνιδιού

Στο πλαίσιο της παρούσας εργασίας αναπτύχθηκε ένα εκπαιδευτικό παιχνίδι, με σκοπό της καλύτερη κατανόηση των αλγορίθμων αναζήτησης πρώτα κατά βάθος και πρώτα κατά πλάτος, μέσω της οπτικοποίησης και της παιχνιδοποίησης. Εκτιμήθηκε ότι μέσω της εμπλοκής με ένα παιχνίδι, θα ήταν περισσότερο κατανοητή η λειτουργία των δύο αλγορίθμων.

Το σενάριο του παιχνιδιού προβλέπει την κίνηση ενός ανθρώπου στην έρημο. Στην έρημο υπάρχουν περιοχές με νερό (όαση). Σε μερικές από αυτές υπάρχουν και πολύτιμα αντικείμενα. Σκοπός του είναι να συλλέξει όλα αυτά τα πολύτιμα αντικείμενα. Ζητούμενο είναι να καθοριστεί η σειρά μετάβασης από όαση σε όαση. Ξεκινάει την προσπάθεια του με ελεύθερη κίνηση. Κάθε φορά που προσεγγίζει μία όαση. Μπορεί να επιλέξει να κινηθεί με βάση το σχεδιασμό που μπορεί να παράξει ο αλγόριθμος DFS ή ο αλγόριθμος BFS. Στην περίπτωση αυτή, συγκεκριμένες διαδρομές από όαση σε όαση είναι επιτρεπτές. Κάθε όαση αντιστοιχεί σε έναν κόμβο γράφου, ενώ οι επιτρεπτές συνδέσεις μεταξύ κόμβων αντιστοιχούν στις ακμές του. Ο άνθρωπος θα πρέπει να περάσει από όλες τις οάσεις με βάση τη σειρά που του καθορίζει ο αλγόριθμος που χρησιμοποιεί (αναζήτηση πρώτα κατά πλάτος ή κατά βάθος), μέχρι να εντοπίσει όλα τα πολύτιμα αντικείμενα.

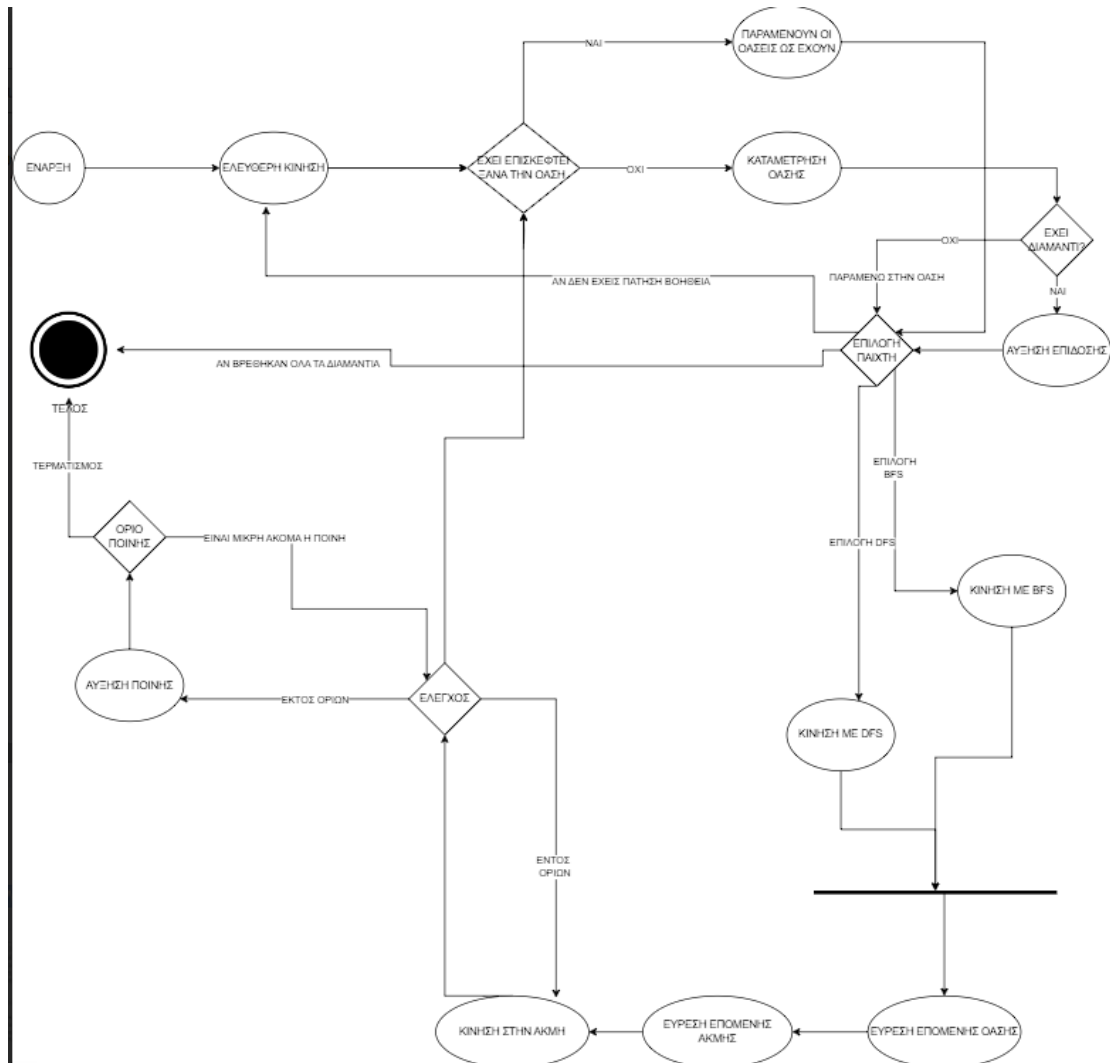
Η ύπαρξη πολύτιμων αντικειμένων, εκτιμήθηκε ότι θα κάνει το παιχνίδι πιο ελκυστικό. Μέσα από τις μεταβάσεις από κόμβο σε κόμβο, που παρουσιάζονται οπτικοποιημένα (είτε σε μορφή χάρτη είτε με πιο ρεαλιστικό τρόπο παρακολουθώντας τον άνθρωπο), ο εκπαιδευόμενος αντιλαμβάνεται τις διαφορές στην εκτέλεση των δύο αλγορίθμων. Κατά την εκκίνηση του παιχνιδιού ο χρήστης έχει τη δυνατότητα να κινηθεί ελεύθερα, χωρίς να περιορίζεται από κάποιον αλγόριθμο. Αυτό μπορεί να του δώσει την ευκαιρία να συγκρίνει την χρήση κάποιου αλγορίθμου και την ολοκληρωτική απουσία τους.

5.3 Λειτουργίες του Παιχνιδιού

Οι λειτουργίες που πραγματοποιούνται κατά τη ροή εκτέλεσης του παιχνιδιού είναι οι εξής:

- Ελεύθερη Κίνηση: Ο άνθρωπος κινείται με βάση τις επιλογές που κάνει ο χρήστης. Με την προσέγγιση σε κάποια όαση μπορεί να αλλάξει σε άλλη κατάσταση κίνησης.
- Καταμέτρηση όασης: Όταν ο άνθρωπος είναι κοντά σε όαση, ελέγχεται αν την έχει επισκεφθεί ξανά. Αν όχι, τότε προσμετράτε στις ελεγμένες.
- Αύξηση επίδοσης: Αν η όαση που έχει προσεγγίσει ο άνθρωπος περιλαμβάνει πολύτιμο αντικείμενο, αυξάνεται το σκορ του παίκτη. Αν έχουν βρεθεί όλα τα πολύτιμα αντικείμενα, το παιχνίδι ολοκληρώνεται.
- Κίνηση με βάση το DFS: Αν ο παίκτης επιλέξει να κινηθεί με τον DFS, καθορίζεται το δρομολόγιο που πρέπει να ακολουθήσει. Η αλληλουχία των οάσεων, που πρέπει να ακολουθήσει, παράγεται με τον αλγόριθμο DFS.
- Κίνηση με βάση το BFS: Αν ο παίκτης επιλέξει να κινηθεί με τον BFS, καθορίζεται το δρομολόγιο που πρέπει να ακολουθήσει. Η αλληλουχία των οάσεων, που πρέπει να ακολουθήσει, παράγεται με τον αλγόριθμο BFS.
- Εύρεση επόμενου κόμβου: Από την αλληλουχία των κόμβων που πρέπει να επισκεφθεί ο άνθρωπος, εντοπίζεται ο επόμενος προς επίσκεψη κόμβος.
- Εύρεση επόμενης ακμής: Με βάση το σημείο στάσης και τον επόμενο κόμβο, επιλέγεται η ακμή που πρέπει να ακολουθηθεί.
- Κίνηση στην ακμή: Ελέγχεται αν ο άνθρωπος κινείται πάνω στην ακμή (ή κοντά σε αυτή)
- Αύξηση βαθμών ποινής: Αν ο άνθρωπος δεν κινείται πάνω στην ακμή, αυξάνονται οι βαθμοί ποινής. Αν οι βαθμοί ποινής φθάσουν το ανώτερο επιτρεπτό όριο, το παιχνίδι τερματίζεται.

Στο παρακάτω διάγραμμα δραστηριοτήτων φαίνονται οι λειτουργίες του παιχνιδιού και οι μεταξύ τους σχέσεις.



Εικόνα 11: Διάγραμμα δραστηριοτήτων εξέλιξης παιχνιδιού

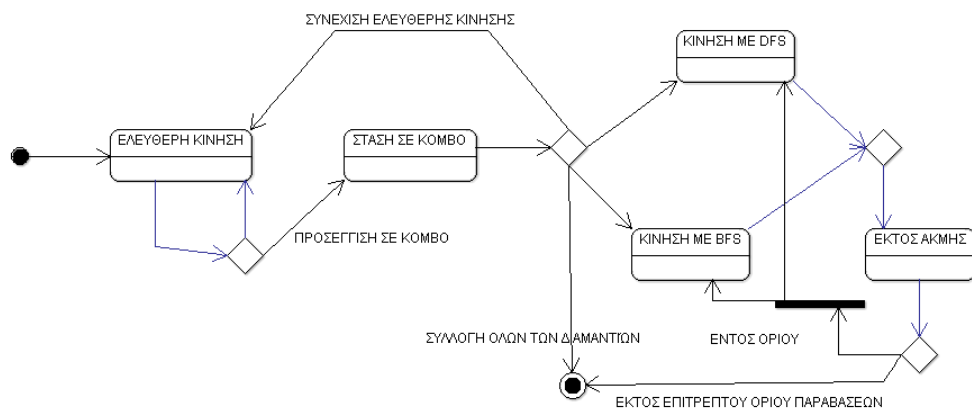
5.4 Καταστάσεις παιχνιδιού

Οι καταστάσεις από τις οποίες διέρχεται ο παίκτης κατά τη ροή του παιχνιδιού είναι οι ακόλουθες:

- Ελεύθερη κίνηση: Στην κατάσταση αυτή ο άνθρωπος που χειρίζεται ο χρήστης κινείται ελεύθερα με βάση μόνο το χειρισμό του χρήστη και χωρίς περιορισμούς.
- Στάση σε κόμβο: Στην κατάσταση αυτή, ο άνθρωπος είναι κοντά σε κόμβο. Ο χρήστης μπορεί να επιλέξει να γίνει ο σχεδιασμός κίνησης με βάση κάποιον από τους αλγορίθμους.

- Κίνηση με DFS: Ο άνθρωπος κινείται με βάση την αλληλουχία κόμβων που βασίζεται στον αλγόριθμο DFS.
- Κίνηση με BFS: Ο άνθρωπος κινείται με βάση την αλληλουχία κόμβων που βασίζεται στον αλγόριθμο BFS.
- Εκτός ακμής: Ο άνθρωπος κινείται εκτός της ακμής που υποδεικνύει ο αλγόριθμος που επέλεξε.

Στο επόμενο διάγραμμα καταστάσεων φαίνονται οι καταστάσεις από τις οποίες περνάει η ροή του παιχνιδιού καθώς και οι πιθανές μεταβάσεις.



5.5 Χρήση του παιχνιδιού

Οι διεπαφές και τα χειριστήρια του παιχνιδιού σχεδιάστηκαν και υλοποιήθηκαν με τρόπο τέτοιο, που ο τρόπος χρήσης τους να είναι προφανής. Στις παρακάτω παραγράφους περιγράφεται ο τρόπος χρήσης του.

5.5.1 Αρχική οθόνη

Με την είσοδο του χρήστη στο παιχνίδι, του δίνεται η δυνατότητα να επιλέξει τις παραμέτρους του. Οι επιλογές που πρέπει να κάνει αφορούν:

- Το αν θα παρέχεται στο χρήστη η βοήθεια του χάρτη ή όχι
- Τον αριθμό των πολύτιμων αντικειμένων που θα πρέπει να τοποθετηθούν στις οάσεις.
- Ο αριθμός των κόμβων που περιλαμβάνονται στον γράφο: Καλείται να καταχωρήσει σε ένα πλαίσιο κειμένου, τον αριθμό των κόμβων που επιθυμεί να έχει ο γράφος.



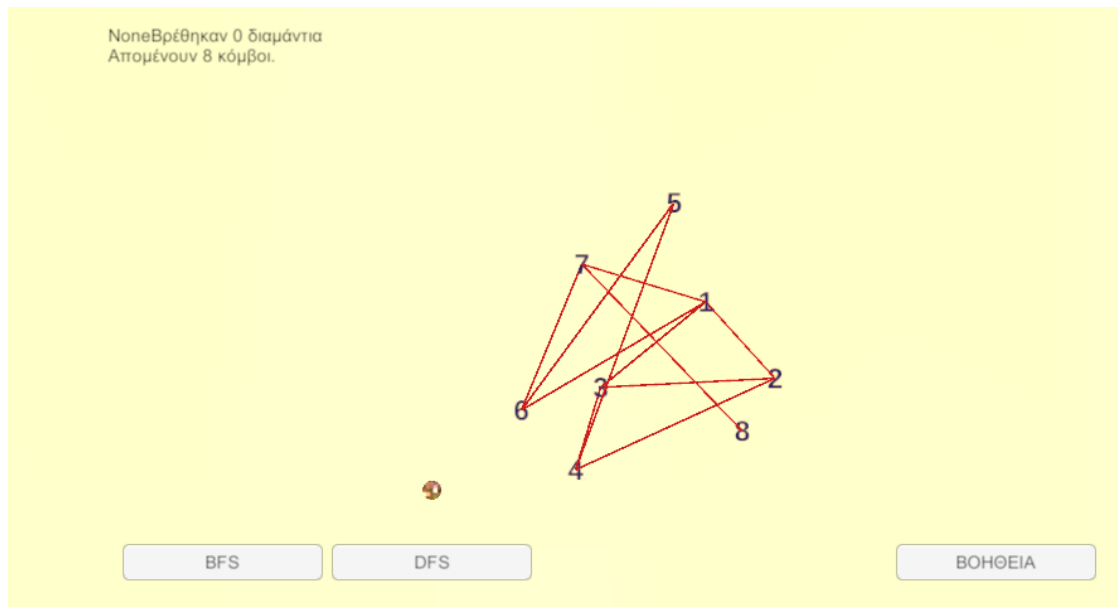
Εικόνα 12: Αρχική οθόνη επιλογής παραμέτρων παιχνιδιού

Αφού ο χρήστης ολοκληρώσει τις επιλογές των παραμέτρων, κάνει κλικ στο πλήκτρο «ΕΝΑΡΞΗ» για να οδηγηθεί στην βασική οθόνη της εξέλιξης του παιχνιδιού.

5.5.2 Βασική οθόνη του παιχνιδιού

5.5.2.1 Αρχικοποίηση του παιχνιδιού

Στον χρήστη, με την είσοδο του στη βασική οθόνη του παιχνιδιού, προβάλλεται ο χάρτης του γράφου – αν έχει επιλέξει να έχει διαθέσιμη τη βοήθεια του χάρτη, ή ο άνθρωπος που χειρίζεται – αν έχει επιλέξει να παίζει χωρίς βοήθεια. Σε κάθε περίπτωση υπάρχουν σχεδιασμένες οι ακμές του γράφου. Στο χάρτη οι κόμβοι παριστάνονται με την αρίθμηση τους. Στο χάρτη επιπλέον αποτυπώνεται η αρχική θέση του ανθρώπου (πολύχρωμος κύκλος). Με την έναρξη του παιχνιδιού ο χρήστης μπορεί να κινηθεί στο χώρο ελεύθερα. Ωστόσο σε οποιαδήποτε στιγμή που βρίσκεται σε κάποιον κόμβο, μπορεί να επιλέξει να κινηθεί με βάση κάποιον από τους αλγορίθμους BFS ή DFS, κάνοντας κλικ στο αντίστοιχο πλήκτρο.



Εικόνα 13: Πλήκτρα αλλαγής τρόπου κίνησης

Παρουσιάζονται επίσης, σε μορφή κειμένου στην πάνω αριστερά πλευρά της οθόνης, πληροφορίες για το παιχνίδι. Αυτές οι πληροφορίες περιλαμβάνουν:

- Το πλήθος των διαμαντιών που έχει συλλέξει
- Το πλήθος των κόμβων που πρέπει να επισκεφθεί

Πέραν αυτών των πληροφοριών, στο πάνω δεξιά μέρος της οθόνης και σε μορφή κειμένου, αναφέρεται το αν ο άνθρωπος βαδίζει πάνω σε κάποια ακμή του γράφου ή έχει μετατοπιστεί μακριά από αυτή (σε περίπτωση που έχει επιλέξει να κινηθεί βάση κάποιου από τους αλγορίθμους).

Ο χρήστης μπορεί να μετακινήσει τον άνθρωπο ως εξής:

- Στροφή του μετώπου του με χρήση του ποντικιού αριστερά ή δεξιά. Εναλλακτικά μπορεί να τον στρέψει χρησιμοποιήσει τα πλήκτρα – βέλη για την αντίστοιχη φορά στροφής.
- Κίνηση εμπρός: Η κίνηση εμπρός γίνεται με το πάνω – βέλος του πληκτρολογίου
- Κίνηση πίσω: Η κίνηση πίσω γίνεται με το κάτω – βέλος του πληκτρολογίου

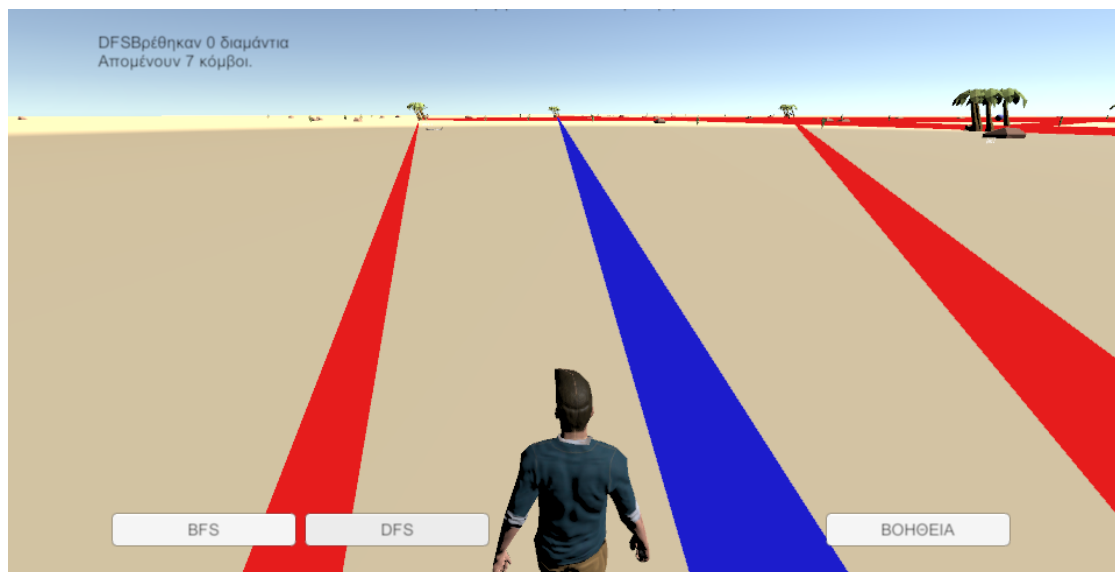
Αν ο χρήστης είναι σε άλλη προβολή και επιθυμεί να περάσει σε προβολή χάρτη, θα πρέπει να κάνει κλικ στο πλήκτρο «C» (αν έχει επιλέξει να παίζει χωρίς βοήθεια τότε η επιλογή αυτή είναι ανενεργή). Αν θέλει να περάσει σε προβολή ελέγχου του κινούμενου χαρακτήρα, θα πρέπει να κάνει κλικ στο πλήκτρο «R». Σε αυτού του τύπου την προβολή, φαίνεται η

φигούρα του κινούμενου ανθρώπου. Ο χρήστης από την οθόνη αυτή μπορεί να δει τον άνθρωπο και να χειριστεί την κίνηση του όπως και στην προβολή του χάρτη. Επιπλέον προβάλλεται το περιβάλλον της ερήμου και οι οάσεις. Σε κάθε οάση φαίνεται η ονομασία της. Στο έδαφος είναι χαραγμένες οι ακμές του γράφου.

Κατά τη ροή του παιχνιδιού και όταν ο χρήστης προσεγγίσει ένα στόχο, έχει τη δυνατότητα να αλλάξει τον τρόπο με τον οποίο κινείται. Οι επιλογές του είναι:

- Να επιλέγει τον επόμενο κόμβο που θα προσεγγίσει ακολουθώντας τον αλγόριθμο DFS.
- Να επιλέγει τον επόμενο κόμβο που θα προσεγγίσει ακολουθώντας τον αλγόριθμο BFS.

Μετά την επιλογή που θα κάνει ο χρήστης, εμφανίζεται η ακμή που οδηγεί στον επόμενο κόμβο (επισημαίνεται με μπλε χρώμα).



Εικόνα 14: Προβολή της ακμής που πρέπει να κινηθεί ο κινούμενος χαρακτήρας.

Αν η ακμή δείχνει το τρόπο επιστροφή σε έναν κόμβο, χρωματίζεται με πράσινο χρώμα. Τέλος, ακμές από τις οποίες ο άνθρωπος έχει διέλθει, χρωματίζονται με κίτρινο χρώμα.



Εικόνα 15: Επισήμανση ακμής για επιστροφή



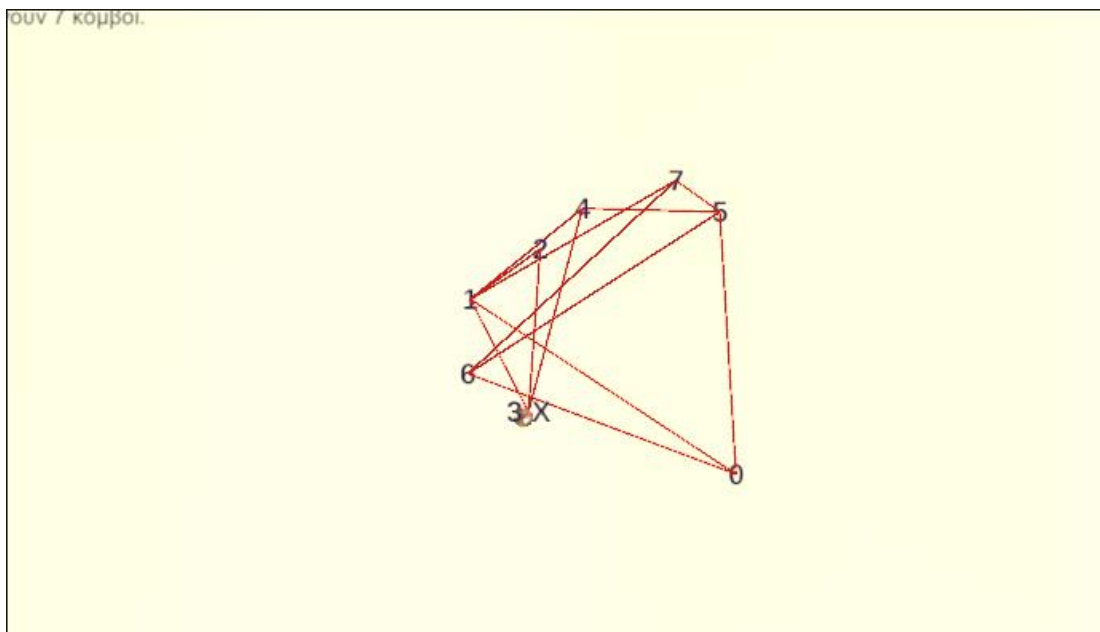
Εικόνα 16: Επισήμανση ακμής που έχει ήδη διανυθεί

Οι οάσεις που έχει ήδη επισκεφθεί ο χρήστης επισημαίνονται με ένα κόκκινο «X».



Εικόνα 17: Επισήμανση επίσκεψης κόμβου

Σε προβολή χάρτη επισημαίνονται με «X».



Εικόνα 18: Επισήμανση κόμβων που έχουν ελεγχθεί σε προβολή χάρτη

Αντίστοιχα οι κόμβοι που έχουν ελεγχθεί και περιείχαν πολύτιμα αντικείμενα, επισημαίνονται με «D».

Επιπλέον, με κλικ στο πλήκτρο βοήθεια, προβάλλονται τα επόμενα βήματα του αλγορίθμου που χρησιμοποιείται.



Εικόνα 19: Προβολή βοήθειας

Με κλικ στο πλήκτρο «ΕΠΙΣΤΡΟΦΗ», εμφανίζεται η σκηνή κίνησης του χρήστη μέσα στην έρημο.

Κατά την κίνηση του ο χρήστης – αν έχει επιλέξει να ακολουθεί κάποιον αλγόριθμο – θα πρέπει να ακολουθεί την αντίστοιχη ακμή. Κάθε φορά που απομακρύνεται από την ακμή, ειδοποιείται με μήνυμα που προβάλλεται στην οθόνη. Αν επαναλάβει την απομάκρυνση από την ακμή τόσες φορές όσο το πλήθος των ακμών, το παιχνίδι ολοκληρώνεται και ο χρήστης θεωρείται ότι έχασε.



Εικόνα 20: Έλεγχος κίνησης πάνω σε ακμή

Ένα ποσοστό των κόμβων θεωρείται ότι περιέχουν διαμάντια. Κάθε φορά που φτάνει σε έναν κόμβο με διαμάντια, τότε προστίθεται ένας βαθμός στη βαθμολογία του.

Σε κάθε φάση της κίνησης του που βρίσκεται σε έναν κόμβο, μπορεί να επιλέξει τον αλγόριθμο τον οποίο θα ακολουθήσει. Στην περίπτωση αυτή, υπολογίζεται η σειρά με την οποία πρέπει να επισκεφθεί τους κόμβους, με εκκίνηση τον τρέχοντα. Αν κατά τη διάρκεια της κίνησης του, επισκέπτεται κάποιον κόμβο που έχει ήδη επισκεφθεί, δεν προσμετράται στο σκορ, αν περιέχει διαμάντι.

Αν ο χρήστης καταφέρει να περάσει από όλους τους κόμβους του γράφου, τότε ειδοποιείται με αντίστοιχο μήνυμα και επιστρέφει στην αρχική οθόνη.

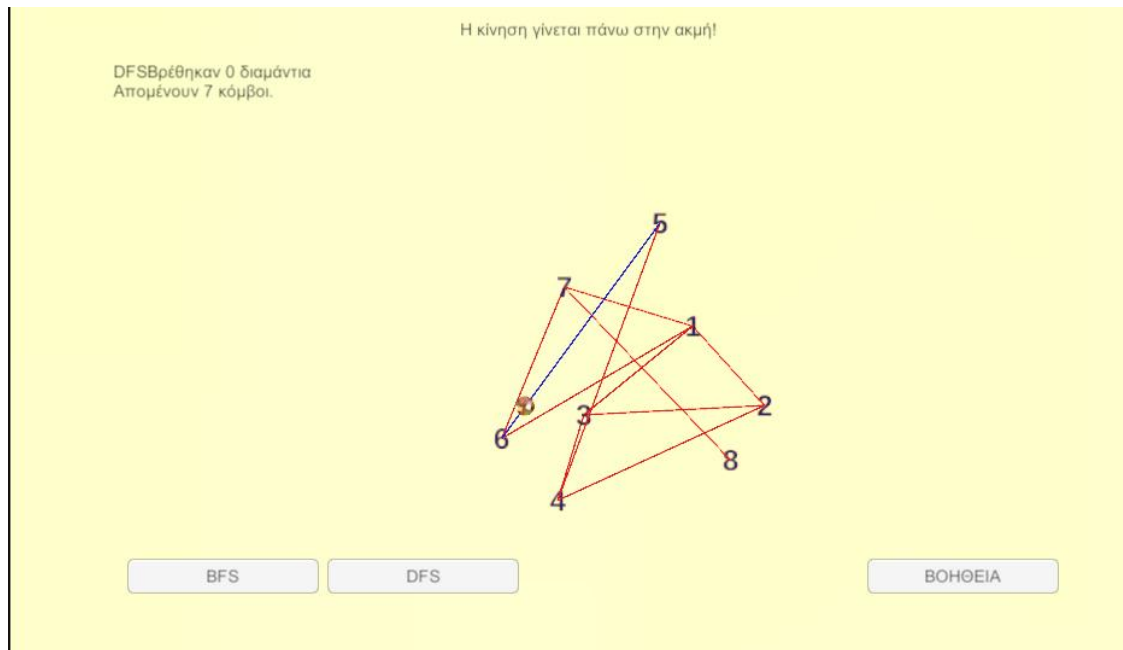
5.5.2.2 Προσέγγιση σε όαση

Όταν ο άνθρωπος προσεγγίζει σε μία όαση, πιο κοντά από 10 μονάδες, θεωρείται ότι έχει φθάσει στον κόμβο. Στην περίπτωση αυτή, η ακμή στη οποία μόλις βάδισε χρωματίζεται με μπλε χρώμα, τόσο πάνω στο έδαφος (σε ρεαλιστική προβολή) όσο και πάνω στο χάρτη (προβολή χάρτη). Στην συνέχεια καθορίζεται ο επόμενος κόμβος που θα πρέπει να επισκεφθεί ο χρήστης, οπότε:

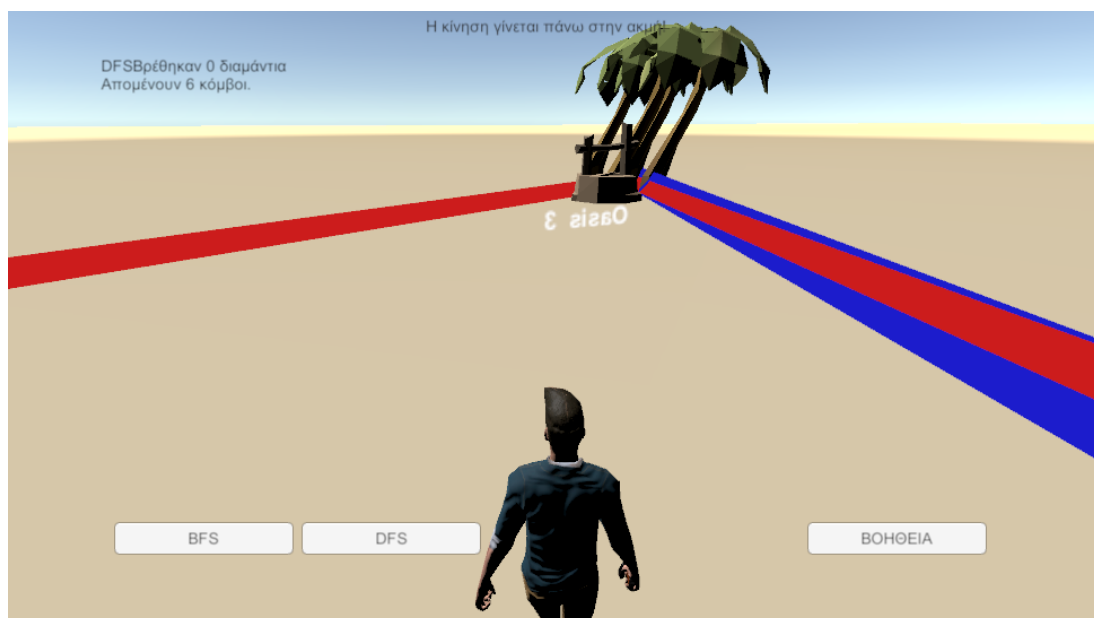
- Μειώνονται κατά ένα οι κόμβοι που πρέπει να επισκεφθεί ο άνθρωπος
- Ενημερώνεται η θέση του επόμενου προς επίσκεψη κόμβου

- Αν ο κόμβος που μόλις επισκεφθηκε ο χρήστης είχε διαμάντι, αυξάνεται το πλήθος τους.

Με την προσέγγιση ακούγεται και χαρακτηριστικός ήχος. Επίσης τονίζεται ακμή την οποία θα πρέπει να ακολουθήσει ο άνθρωπος για να φθάσει στον επόμενο κόμβο.



Εικόνα 21: Προσέγγιση κόμβου (προβολή χάρτη)



Εικόνα 22: Προσέγγιση κόμβου (ρεαλιστική προβολή)

5.5.2.3 Τέλος παιχνιδιού

Όταν ο άνθρωπος προσεγγίσει τον τελευταίο κόμβο, ολοκληρώνεται το παιχνίδι και εμφανίζεται η ένδειξη «ΤΕΛΟΣ». Το τέλος του παιχνιδιού μπορεί να προκύψει νωρίτερα, αν ο άνθρωπος βγει εκτός ακμής για την μετακίνηση σε κόμβο, δέκα φορές.

Σε κάθε φάση της εξέλιξης του παιχνιδιού ο χρήστης μπορεί να το τερματίσει με κλικ στο πλήκτρο «E».



Εικόνα 23: Τέλος του παιχνιδιού σε ρεαλιστική προβολή

5.6 Υλοποίηση

Κατά την υλοποίηση της εφαρμογής προέκυψαν δυσκολίες. Η υποστήριξη που παρέχεται από την ομάδα ανάπτυξης του Unity και από τη διαδικτυακή κοινότητα που ασχολείται με την ανάπτυξη παιχνιδιών, ήταν καταλυτική για την παράκαμψή τους. Οι σημαντικότερες από αυτές καθώς και ο τρόπος με τον οποίο αντιμετωπίστηκαν, αναφέρονται στις παρακάτω παραγράφους.

5.6.1 Δημιουργία συνεκτικών γράφων

Κατά την εκκίνηση του παιχνιδιού, ο χρήστης εισάγει τον αριθμό των κόμβων που θα έχει ο γράφος. Επιλέχθηκε ο αριθμός των ακμών να είναι το 60% του μέγιστου αριθμού που θα μπορούσαν να είναι. Δηλαδή αν ο χρήστης επιλέξει να δημιουργηθεί ένας γράφος με 10 κόμβους, θα τοποθετηθούν μεταξύ αυτών $0.6X(10X9)/2 = 27$ ακμές⁹. Αν και οι δύο αλγόριθμοι μπορούν να λειτουργήσουν και σε μη συνεκτικούς γράφους, προκειμένου η μετακίνηση μεταξύ των οάσεων – κόμβων να είναι ρεαλιστική, επιλέχθηκε να εξασφαλιστεί η δημιουργία συνεκτικών γράφων. Για το σκοπό αυτό, αν ο γράφος περιλαμβάνει n κόμβους – και επομένως $0.6XnX(n-1)/2$ ακμές – οι $n-1$ πρώτες ακμές δημιουργούν ένα μονοπάτι Hamilton¹⁰ (δηλαδή ένα μονοπάτι το οποίο περιλαμβάνει όλους τους κόμβους του γράφου και έχει μήκος $n-1$). Η σειρά των κόμβων πάνω στο μονοπάτι επιλέγεται τυχαία. Μετά τη δημιουργία του μονοπατιού Hamilton, επιλέγονται τυχαία ζεύγη κόμβων τα οποία αποτελούν τα άκρα των υπολοίπων ακμών¹¹. Για κάθε μία από τις ακμές χαράσσονται δύο γραμμές: μία πάνω στο έδαφος και μία πάνω στο χάρτη. Στο τέλος της διαδικασίας της αρχικοποίησης του γράφου, ενημερώνεται ο αριθμός των ακμών ανάλογα με το πόσες δημιουργήθηκαν στην πράξη.

Οι θέσεις των κόμβων του γράφου καταγράφονται σε πίνακα με τις ακόλουθες μορφές:

- Σαν διανύσματα για τη θέση τους πάνω στο έδαφος
- Σαν διανύσματα για τη θέση τους στο χάρτη
- Σαν ονομασίες
- Σαν GameObjects που αναπαριστούν τις οάσεις

5.6.2 Δημιουργία Κόμβων

Στην αρχική οθόνη της εφαρμογής ο χρήστης επιλέγει το πλήθος των κόμβων¹² και των πλήθος των διαμαντιών που θα τοποθετηθούν σε αυτούς. Κάθε κόμβος που δημιουργείται, επιλέγεται με τυχαίο τρόπο το που θα τοποθετηθεί. Ο χρήστης έχει επιλέξει το πόσοι κόμβοι θα περιέχουν διαμάντια (έστω $wNodes$). Οι πρώτοι $wNodes$ κόμβοι που θα δημιουργηθούν θα περιλαμβάνουν διαμάντια (ωστόσο αφού η θέση τους επιλέγεται τυχαία, θα είναι

⁹ Αν ένας μη προσανατολισμένος γράφος έχει n κόμβους, αν είναι πλήρης θα έχει $n(n-1)/2$ ακμές.

¹⁰ Ο κώδικας της δημιουργίας του μονοπατιού Hamilton φαίνεται στο Παράρτημα Α

¹¹ Ο κώδικας της δημιουργίας των υπολοίπων ακμών του γράφου φαίνεται στο Παράρτημα Β

¹² Ο κώδικας της δημιουργίας των κόμβων φαίνεται στο Παράρτημα Γ

διασκορπισμένοι σε τυχαίες περιοχές). Ανάλογα με το αν ο κόμβος περιλαμβάνει διαμάντι ή όχι, επιλέγεται το κατάλληλο prefab για τη δημιουργία του.

5.6.3 Προβολή χάρτη

Για να υλοποιηθεί η προβολή του χάρτη, δημιουργήθηκε μία προβολή του εδάφους σε μεγαλύτερο ύψος. Σε αυτή δημιουργήθηκαν οι γραμμές που απεικονίζουν τις ακμές του γράφου, επίσης σε παράλληλη προβολή τους σε ύψος λίγο μεγαλύτερο από την προβολή του εδάφους. Στην προβολή αυτή τοποθετήθηκαν λευκές κουκκίδες ακριβώς πάνω από τους κόμβους του γράφου (και λίγο πιο πάνω από την προβολή του εδάφους). Ομοίως πάνω από τον κινούμενο χαρακτήρα τοποθετείται μία κουκίδα, η οποία κινείται μαζί του, αλλά στο ύψος της προβολής του εδάφους. Στα παραρτήματα Α και Β φαίνεται η δημιουργία των ζευγών των ακμών (μία στο έδαφος και μία στην προβολή του).

5.6.4 Εναλλαγή προβολών

Για την εναλλαγή των προβολών (χάρτη και εδάφους) χρησιμοποιήθηκαν δύο κάμερες: η μία είναι προσαρμοσμένη στο έδαφος και ακολουθεί τον κινούμενο χαρακτήρα και η άλλη είναι προσαρμοσμένη στην προβολή του εδάφους. Με κλικ στα αντίστοιχα πλήκτρα του πληκτρολογίου, η μία από αυτές απενεργοποιείται και ενεργοποιείται η άλλη¹³.

5.6.5 Προσέγγιση κόμβου

Ο έλεγχος για την προσέγγιση κόμβου γίνεται με δύο τρόπους:

- Αν δεν έχει επιλεχθεί κάποιος από τους αλγόριθμους DFS ή BFS τότε γίνεται έλεγχος αν απέχει λιγότερο από 10 μονάδες από κάθε ένα από τους κόμβους του γράφου (για το σκοπό αυτό διατρέχεται ο πίνακας με τις θέσεις των κόμβων – διανύσματα – και υπολογίζεται η ευκλείδεια απόσταση από την τρέχουσα θέση του χαρακτήρα).
- Αν έχει επιλεχθεί ένας αλγόριθμος για την επιλογή του επόμενου προς επίσκεψη κόμβου, εξετάζεται αν η απόσταση του χαρακτήρα είναι μικρότερη από 10 μονάδες από τον συγκεκριμένο κόμβο¹⁴.

¹³ Ο κώδικας με τον οποίο πραγματοποιείται η εναλλαγή της κάμερας φαίνεται στο Παράρτημα Δ

¹⁴ Στο Παράρτημα Ε φαίνεται ο κώδικας για την προσέγγιση κόμβου από τον κινούμενο χαρακτήρα.

Όταν ανιχνευθεί η προσέγγιση, ενημερώνονται τα στοιχεία του παιχνιδιού (μειώνεται ο αριθμός των ακμών που απομένουν, αυξάνεται ο αριθμός των διαμαντιών που βρέθηκαν αν ο κόμβος περιείχε κάποιο). Επιπλέον, εξετάζεται να ο συγκεκριμένος κόμβος έχει προσεγγιστεί πιο νωρίς. Αυτό γίνεται με τη βοήθεια ενός πίνακα ακεραίων που έχει τόσα κελιά όσοι είναι οι κόμβοι του γράφου. Κάθε φορά που προσεγγίζεται ένας κόμβος, η τιμή του αντίστοιχου κελιού γίνεται 1.

5.6.6 Προβολή βοήθειας

Για την προβολή βοήθειας δημιουργήθηκε ένας καμβάς στον οποίο τοποθετήθηκε ένα πλαίσιο κειμένου. Στο πλαίσιο αυτό φορτώνεται κείμενο το οποίο περιλαμβάνει τις οδηγίες κίνησης για το τρέχον και το επόμενο στάδιο του αλγορίθμου που έχει επιλεγεί. Παράλληλα η κάμερα του χάρτη, αλλάζει ύψος προκειμένου να προβάλλει το background του καμβά της διεπαφής της βοήθειας. Όταν ο χρήστης κάνει κλικ στο πλήκτρο «ΕΠΙΣΤΡΟΦΗ» πραγματοποιείται η αντίστροφη διαδικασία: η κάμερα έρχεται στο ύψος της προβολής χάρτη, απενεργοποιείται ο καμβάς της βοήθειας και ενεργοποιείται ο καμβάς της κύριας προβολής¹⁵.

5.6.7 Ολοκλήρωση παιχνιδιού

Η ολοκλήρωση του παιχνιδιού συμβαίνει όταν ο αριθμός των κόμβων που απομένουν για επίσκεψη μηδενίζονται. Ο έλεγχος γίνεται κάθε φορά που προσεγγίζεται ένας κόμβος. Αφού μηδενιστούν οι εναπομείναντες κόμβοι, τότε εμφανίζεται κατάλληλο μήνυμα και ανοίγει εκ νέου η αρχική οθόνη της εφαρμογής¹⁶.

5.6.8 Αλλαγή σε παρακολούθηση αλγορίθμου

Με την εκκίνηση της εφαρμογής, δεν έχει ακολουθείται κάποιος αλγόριθμος για την επιλογή του επόμενου κόμβου, ο χαρακτήρας κινείται ελεύθερα. Όταν ο χρήστης κάνει κλικ σε κάποιο από τα πλήκτρα «DFS» ή «BFS», η μεταβλητή method λαμβάνει την αντίστοιχη τιμή. Ανάλογα με την τιμή αυτή πραγματοποιούνται οι παρακάτω ενέργειες:

- Επιλογή BFS: Δημιουργείται μία αλληλουχία επίσκεψης των κόμβων, ξεκινώντας από τον τρέχοντα κόμβο. Εκτελείται ο αλγόριθμος BFS για την

¹⁵ Ο κώδικας για την προβολή βοήθειας παρουσιάζεται στο Παράρτημα Ζ.

¹⁶ Στο Παράρτημα Η παρουσιάζεται ο κώδικας της ολοκλήρωσης του παιχνιδιού.

δημιουργία της αλληλουχίας αυτής. Η φύση του αλγορίθμου BFS επιβάλλει μετά την επίσκεψη κάθε κόμβου, την επιστροφή του στον προηγούμενο του. Αυτό απαιτεί σε κάθε επίσκεψη κόμβου, τη διατήρηση του προηγούμενου σε μια μεταβλητή. Αφού δημιουργηθεί η αλληλουχία, ο δεύτερος κόμβος της τίθεται σαν ο επόμενος για επίσκεψη. Σε κάθε επίσκεψη κόμβου, ο επόμενος προς επίσκεψη γίνεται αυτός που έπεται στην αλληλουχία των κόμβων¹⁷.

- Επιλογή DFS: Και στην περίπτωση αυτή δημιουργείται η αλληλουχία των κόμβων που θα επισκεφθεί ο χαρακτήρας. Η δημιουργία της αλληλουχίας γίνεται με χρήση της αναδρομικής εκδοχής του αλγορίθμου DFS. Σε κάθε αναδρομική κλήση, ο κόμβος της παραμέτρου της, τοποθετείται στην αλληλουχία και καλείται εκ νέου ο αλγόριθμος για κάθε γείτονα του. Στο τέλος των αναδρομών, προστίθεται στην αλληλουχία και πάλι ο κόμβος της παραμέτρου¹⁸.

Η μετάβαση σε κίνηση με βάση αλγόριθμο γίνεται από τον τρέχοντα κόμβο. Ο τρέχων κόμβος γίνεται ο πρώτος στην αλληλουχία. Δημιουργείται ένας προσωρινός πίνακας για να διατηρεί τους κόμβους, που έχουν ήδη τοποθετηθεί στην αλληλουχία. Ο πίνακας που διατηρεί τους κόμβους που έχει ήδη επισκεφθεί ο χαρακτήρας παραμένει ανέπαφος (σε διαφορετική περίπτωση υπήρχε η πιθανότητα ο γράφος να έπαυε να είναι συνεκτικός).

5.6.9 Έλεγχος για κίνηση πάνω στην ακμή

Ο χρήστης είναι υποχρεωμένος να κινηθεί πάνω στην ακμή που του υποδεικνύεται, σε περίπτωση που έχει επιλέξει να εκτελέσει την κίνηση του με βάση κάποιον από τους αλγορίθμους. Για το σκοπό αυτό σε κάθε frame υπολογίζεται η απόσταση (ευκλείδεια) του σημείου στάσεως, από τη γραμμή που αντιστοιχεί στην τρέχουσα ακμή. Αν η απόσταση είναι μεγαλύτερη από 20 μονάδες, ένας μετρητής αυξάνεται κατά 1. Ωστόσο δεν αυξάνεται συνέχεια όσο είναι εκτός δρομολογίου αλλά μόνο τη στιγμή που εξέρχεται. Όταν ο μετρητής φθάσει να έχει την ίδια τιμή με το πλήθος των ακμών, τότε το παιχνίδι ολοκληρώνεται αμέσως¹⁹.

¹⁷ Υλοποιείται με την μέθοδο BFS η οποία παρουσιάζεται στο Παράρτημα Στ

¹⁸ Υλοποιείται με την μέθοδο DFS η οποία παρουσιάζεται στο Παράρτημα Θ

¹⁹ Ο κώδικας υλοποίησης του φαίνεται στο παράρτημα Ι.

5.6.10 Χειριστήρια κίνησης του χαρακτήρα

Ο χρήστης καθορίζει την κίνηση του χαρακτήρα. Μπορεί να τον οδηγεί μπροστά ή πίσω κάνοντας κλικ στα αντίστοιχα βελάκια του πληκτρολογίου. Για να μπορέσει να κινηθεί προς άλλες κατευθύνσεις θα πρέπει να στρέψει το μέτωπο του προς τον προσανατολισμό που πρέπει να κινηθεί και να προχωρήσει μπροστά. Η περιστροφή γίνεται με κίνηση του ποντικιού²⁰.

5.6.11 Κίνηση στα όρια του πεδίου

Σε κάθε frame γίνεται έλεγχος να μην ξεπερνάει ο χαρακτήρας τα όρια του πεδίου κίνησης. Ελέγχονται οι συντεταγμένες του στις δύο διαστάσεις να μην ξεπερνούν τα όρια που τίθενται²¹.

²⁰ Ο κώδικας υλοποίησης του φαίνεται στο Παράρτημα ΙΑ.

²¹ Ο κώδικας υλοποίησης του φαίνεται στο Παράρτημα ΙΒ

6 Συμπεράσματα

Οι γράφοι είναι πολύτιμα εργαλεία για τη μοντελοποίηση προβλημάτων και για την περιγραφή αλγορίθμων για την επίλυση τους. Υπάρχουν πολλά προβλήματα του πραγματικού κόσμου, για τα οποία οι γράφοι χρησιμοποιούνται για την περιγραφή τους. Αυτό καθιστά τη μελέτη τους ενδιαφέρουσα για πολλές διαφορετικές δραστηριότητες του ανθρώπου. Η διδασκαλία των ιδιοτήτων και των αλγορίθμων γράφων αποτελεί θεμελιώδες αντικείμενο για τα προγράμματα σπουδών πληροφορικής και τηλεπικοινωνιών. Αποτελεί βασικό προαπαιτούμενο για τους εκπαιδευόμενους να κατανοούν τη δομή και τις λειτουργίες που σχετίζονται με τους γράφους προκειμένου να μπορούν να αντιλαμβάνονται:

- Σύνθετες δομές δεδομένων
- Προβλήματα που έχουν να κάνουν με συσχετίσεις αντικειμένων
- Τη μελέτη βασικών αλγορίθμων

Η σημασία των γράφων, καθορίζει και την υψηλή αξία της διδασκαλίας τους. Οι διδάσκοντες αναζητούν τρόπους για να καταστήσουν την εκμάθηση εκπαιδευτικών αντικειμένων που σχετίζονται με τους γράφους, ευχάριστη και αποδοτική. Η αναζήτηση αυτή τα τελευταία χρόνια έχει αποτελέσματα τα οποία εν πολλοίς οφείλονται στην βαθιά ενσωμάτωση των νέων τεχνολογιών της πληροφορικής και των τηλεπικοινωνιών στην εκπαίδευση.

Μία από τις σημαντικότερες υπηρεσίες που προσέφερε η τεχνολογία στην εκπαίδευση, είναι η δυνατότητα της οπτικοποίησης των περιεχομένων των εκπαιδευτικών διαδικασιών. Οι γράφοι προσφέρονται για οπτικοποίηση καθώς είναι εύκολο να αναπαρασταθούν με απλά γραφικά στοιχεία οι κόμβοι και οι ακμές. Παράλληλα είναι σχετικά εύκολη και η κωδικοποίηση της δομής και των λειτουργιών των γράφων με γλώσσες προγραμματισμού. Ειδικότερα με τον αντικειμενοστραφή προγραμματισμό η μοντελοποίηση των γράφων γίνεται περισσότερο προφανής.

Η μελέτη των γράφων είναι σχετικά ώριμη. Αποτέλεσμα αυτού είναι να έχουν παγιωθεί αρκετοί αλγόριθμοι για την επίλυση προβλημάτων, που μοντελοποιούνται με γράφους. Αυτό το γεγονός έχει συμβάλει στην αρτιότερη κατάρτιση των εκπαιδευτών και την ικανότητα τους να αναπτύσσουν αποτελεσματικές μεθοδολογίες διδασκαλίας τους. Η άριστη γνώση του εκπαιδευτικού αντικειμένου αυξάνει την πιθανότητα για τον διδάσκοντα να επιλέξει τα κατάλληλα εργαλεία και μεθοδολογίες για την διδασκαλία του. Καθώς η αποτελεσματικότητα της οπτικοποίησης των γράφων είναι σχεδόν προφανής, οι διδάσκοντες αναζητούν ευφάνταστους τρόπους για την οπτικοποίηση τους.

Τα εκπαιδευτικά παιχνίδια είναι ένας από τους αποτελεσματικότερους τρόπους διδασκαλίας των εκπαιδευτικών αντικειμένων γενικότερα. Όταν συνδυάζονται με την

οπτικοποίηση του, η αποτελεσματικότητά τους πολλαπλασιάζεται. Κατά συνέπεια ένα ευχάριστο ηλεκτρονικό παιχνίδι το οποίο οπτικοποιεί τη δομή των γράφων και τους σχετικούς με αυτούς αλγορίθμους, θα συμβάλει στη γρηγορότερη αφομοίωση τους από τους μαθητές. Γενικότερα, η επιτυχία τους οφείλεται σε μεγάλο βαθμό στον ψυχαγωγικό τους χαρακτήρα. Καθώς η ψυχαγωγία είναι μία δραστηριότητα που είναι ευχάριστη για τον άνθρωπο, τον κάνει να συμμετέχει ενσυνείδητα και εκούσια. Συμμετέχοντας με τέτοιο τρόπο ο εκπαιδευόμενος αφομοιώνει το εκπαιδευτικό αντικείμενο πιο εύκολα, γρηγορότερα και με μεγαλύτερη ακρίβεια. Αυτό το χαρακτηριστικό τους είναι που κάνει τους διδάσκοντες να έχουν ιδιαίτερη προτίμηση σε αυτά. Η ευρεία χρήση των παιχνιδιών για σκοπούς διαφορετικούς από ψυχαγωγικούς, ανέπτυξε ένα νέο όρο για αυτά, *serious games*, δείγμα του ότι οι σχετικές μελέτες θα συνεχιστούν τα επόμενα χρόνια. Τα τελευταία χρόνια παρατηρείται έντονο ενδιαφέρον τόσο σε ερευνητικό επίπεδο όσο και στην αγορά που σχετίζεται με τις νέες τεχνολογίες πληροφορικής και τηλεπικοινωνιών.

Η ανάπτυξη των σοβαρών παιχνιδιών αναπτύσσεται παράλληλα με τα ηλεκτρονικά παιχνίδια. Οι άνθρωποι σήμερα μπορούν να χρησιμοποιούν διαφορετικές συσκευές – πέρα από τους ηλεκτρονικούς υπολογιστές - για να χρησιμοποιούν εφαρμογές παιχνιδιών (κυρίως σε έξυπνες κινητές συσκευές). Αποτέλεσμα αυτού ήταν να διευρυνθεί το κοινό στο οποίο απευθύνονται (καθώς σχεδόν όλοι οι άνθρωποι κάθε ηλικίας έχουν στην κατοχή τους τουλάχιστον ένα έξυπνο κινητό τηλέφωνο. Αυτοί οι άνθρωποι είναι δυνητικοί χρήστες των παιχνιδιών. Η διεύρυνση του κοινού αποτελεί κίνητρο για τις εταιρείες ανάπτυξης εφαρμογών ώστε να διατηρούν ενεργή την έρευνα για καινοτόμες δημιουργίες. Ο κύκλος που δημιουργείται: έρευνα – ανάπτυξη – κερδοφορία – ανάπτυξη κινήτρων για έρευνα – νέα έρευνα, είναι η εγγύηση για τη διατήρηση της πορείας αναβάθμισης των τεχνολογιών των *serious games* στο μέλλον.

Ένας βασικός παράγοντας που συνέβαλε στην επιτυχία των έξυπνων παιχνιδιών είναι η ικανότητα τους να διατηρούν το απαιτούμενο κόστος της εκπαίδευσης σε χαμηλά επίπεδα. Έχουν αναπτυχθεί εργαλεία πληροφορικής για την ανάπτυξη εφαρμογών, τα οποία διατίθενται δωρεάν και είναι ικανά να υποστηρίξουν τη σχεδίαση και την ανάπτυξη παιχνιδιών κάθε κλίμακα. Ένα τέτοιο εργαλείο είναι και το Unity που χρησιμοποιήθηκε στην παρούσα εφαρμογή. Το Unity, πέρα από το μηδενικό κόστος εγκατάστασης και χρήσης του, έχει ως βασικό του πλεονέκτημα ότι παρέχει φιλικές γραφικές διεπαφές χρήστη για την ανάπτυξη των εφαρμογών. Επιπλέον, υποστηρίζει τη χρήση δημοφιλών γλωσσών προγραμματισμού (C#, Javascript) για τις περιπτώσεις που απαιτείται. Αυτό ενθαρρύνει μεγάλη μερίδα των προγραμματιστών να ασχοληθούν με την εκμάθηση και τη χρήση του.

Από την ενασχόληση με την ανάπτυξη της εφαρμογής, διαπιστώθηκε η πλήρης υποστήριξη που παρέχουν στο Unity, τόσο η ομάδα ανάπτυξης του, όσο και η ευρύτερη κοινότητα των προγραμματιστών. Οποιαδήποτε πρόκληση εμφανίστηκε κατά την ανάπτυξη της εφαρμογής, αντιμετωπίστηκε επιτυχώς με αναζήτηση στο διαδίκτυο και εύρεση ενημερωτικού και εκπαιδευτικού υλικού σε διάφορες μορφές (εικόνα, κείμενο, video).

Η διαδικασία ανάπτυξης επιταχύνθηκε λόγω της ύπαρξης έτοιμων γραφικών μοντέλων, τα οποία μπορούσαν να χρησιμοποιηθούν στο έργο μετά από κατάλληλες παραμετροποιήσεις. Με τον τρόπο αυτό, ο προγραμματιστής δεν είναι υποχρεωμένος να ασχοληθεί με τη σχεδίαση και την ανάπτυξη τους, εξασφαλίζοντας χρόνο για τη σχεδίαση και υλοποίηση της λειτουργικότητας της εφαρμογής.

Στα πλαίσια της παρούσας μελέτης σχεδιάστηκε και υλοποιήθηκε ένα σοβαρό παιχνίδι για την διδασκαλία αντικειμένου με προσανατολισμό τον προγραμματισμό. Κατά την σχεδίαση διαπιστώθηκε ότι η διαδικασία αυτή δεν διαφέρει από την αντίστοιχη διαδικασία που ακολουθείται για κάθε είδους λογισμικό. Ωστόσο χρειάζεται να εξεταστούν και ζητήματα τα οποία κινούνται και σε δύο ειδικές συνιστώσες: την εξέταση ζητημάτων που αφορούν της οργάνωση της εκπαιδευτικής διαδικασίας και την μέριμνα για την αποδοτικότερη χρήση των στοιχείων του παιχνιδιού. Για τον λόγο αυτό η σωστή αρχική σχεδίαση του εκπαιδευτικού παιχνιδιού είναι σημαντική και απαιτεί περισσότερο χρόνο και σχολαστικότητα σε σχέση με την γενικότερη σχεδίαση λογισμικού.

Η αποδοτικότητα ενός σοβαρού παιχνιδιού αποτιμάται από το κατά πόσο:

- Έχει τύχει αποδοχής από το κοινό στο οποίο απευθύνεται (διδάσκοντες και διδασκόμενοι)
- Αν εξυπηρετεί τον σκοπό για τον οποίο δημιουργήθηκε.
- Ο βαθμός κατά τον οποίο μπορεί να προσαρμοστεί σε διαφορετικά εκπαιδευτικά περιβάλλοντα.

Για να μπορέσει να απαντήσει στις απαιτήσεις αυτές ο σχεδιαστής της εφαρμογής θα πρέπει να μελετήσει τα χαρακτηριστικά των δυνητικών του χρηστών: βασικά κοινά στοιχεία του χαρακτήρα και της προσωπικότητας τους (καθώς προορίζεται για ανθρώπους με διαφορετικά αντίστοιχα χαρακτηριστικά με τα δικά του). Συχνά χρειάζεται να καταφύγει σε αντικειμενικά πλαίσια αξιολόγησης τους.

Συνοψίζοντας τα σοβαρά παιχνίδια μπορούν να συμβάλλουν στην βελτίωση της αποδοτικότητας της διδασκαλίας αντικειμένων της θεωρίας των γράφων, μέσα από τη δυνατότητα τους για εκτεταμένη οπτικοποίηση τους. Η δυναμική τους μεγαλώνει αν τεθεί υπ’

όψη η ενεργή συμμετοχή του εκπαιδευομένου στην εξέλιξη τους. Η διαδραστικότητα τους βοηθάει τους εκπαιδευόμενους να αντιλαμβάνονται με πρακτικό τρόπο:

- Τη δομή των γράφων
- Το τρόπο εξέλιξης των αλγορίθμων που λύνουν προβλήματα που μοντελοποιούνται με αυτούς.

Η βαθιά ωριμότητα των ηλεκτρονικών παιχνιδιών στη σύγχρονη επιτρέπει την ενσωμάτωση των περισσότερων επιτευγμάτων της τεχνολογίας της πληροφορικής και των τηλεπικοινωνιών στις αντίστοιχες εκπαιδευτικές δραστηριότητες. Με τον τρόπο αυτό αναβαθμίζεται η αποτελεσματικότητα της εκπαίδευσης. Το γεγονός ότι έννοιες και διαδικασίες μπορούν να οπτικοποιηθούν με ευχάριστα γραφικά στοιχεία επιτρέπει τη συστηματική διδασκαλία της θεωρίας των γράφων, ακόμα και σε μαθητές μικρής ηλικίας. Τα πλεονεκτήματα την διαδραστικής οπτικοποίησης γενικότερα στην εκπαίδευση, εκτιμάται ότι θα συνεχίσουν να αποτελούν το βασικό κίνητρο για τη συνέχεια της έρευνας και στο προσεχές μέλλον.

Βιβλιογραφία

- Backlund, P., & Hendrix, M. (2013). *Educational games - are they worth the effort? A literature survey of the effectiveness of serious games*. Ανάκτηση από ac: <https://core.ac.uk/download/pdf/228142915.pdf>
- Brookshear, G. (2012). *Η επιστήμη των υπολογιστών*. Αθήνα: Κλειδάριθμος.
- Bulman, G. (2016). *Technology education*. Ανάκτηση από nber: https://www.nber.org/system/files/working_papers/w22237/w22237.pdf
- Chien, J. (2012). *How digital media and Internet transforming education*. Ανάκτηση από researchgate: https://www.researchgate.net/publication/235901330_How_digital_media_and_Internet_transforming_education
- Louridas, P. (2019). *Real-World Algorithms*. MIT.
- Masalimova, A., Khvatova, M., Chikileva, L., Zvyagintseva, E., Stepanova, V., & Melnik5†, M. (2022). *Distance Learning in Higher Education During Covid-19*. Ανάκτηση από frontiersin: <https://www.frontiersin.org/articles/10.3389/feduc.2022.822958/full>
- Noem, P.-M., & Máximo, S. H. (2014). *Educational Games for Learning*. Ανάκτηση από ed: <https://files.eric.ed.gov/fulltext/EJ1053979.pdf>
- Papadimitriou, C. H. (2003, January 1). *dl.acm.org*. Ανάκτηση από In Encyclopedia of computer science: <https://dl.acm.org/doi/pdf/10.5555/1074100.107>
- Rahmani, A. M., Bayramov, S., & Kalejahi, B. K. (2022). *Internet of Things Applications: Opportunities and Threats*. Ανάκτηση από springer: <https://link.springer.com/article/10.1007/s11277-021-08907-0>
- Şahin, A. (2021). *The Use of Educational Software in Teaching Initial Reading and Writing*. Ανάκτηση από eric: <https://files.eric.ed.gov/fulltext/EJ1308656.pdf>
- Sryheni, S. (2023). *Introduction to Depth First Search Algorithm (DFS)*. Ανάκτηση από baeldung: <https://www.baeldung.com/cs/depth-first-search-intro>
- Κόκκας, Δ. (2008). *Μεθοδολογία και προδιαγραφές για την επιλογή κατάλληλων προτύπων δομικών στοιχείων Πολυμεσικού Εκπαιδευτικού υλικού*. Αθήνα: Γεωπονικό Πανεπιστήμιο Αθηνών.
- Σπαντιδάκης, Γ. (2008). *Τα εκπαιδευτικά πολυμέσα στην υπηρεσία της μάθησης και της διδασκαλία*. Ανάκτηση από Πανεπιστήμιο Κρήτης:

http://www.edc.uoc.gr/~odysseas/webs/epimorfosh/EAR2008/Spadidakis_Lecture/Spadidakis_Lecture.pdf.

7 Παράρτημα Α: Δημιουργία μονοπατιού Hamilton

```
//ΔΗΜΙΟΥΡΓΕΙΤΑΙ ΕΝΑ ΜΟΝΟΠΑΤΙ ΠΟΥ ΣΥΝΔΕΕΙ ΟΛΟΥΣ ΤΟΥΣ ΚΟΜΒΟΥΣ
//ΣΕ ΜΙΑ ΣΕΙΡΑ ΩΣΤΕ Ο ΓΡΑΦΟΣ ΝΑ ΕΙΝΑΙ ΣΥΝΕΚΤΙΚΟΣ
int[] nodesChecked = new int[this.numberofOasis];//ΟΙ ΚΟΜΒΟΙ ΠΟΥ
ΜΠΗΚΑΝ ΣΤΟ ΑΡΧΙΚΟ ΜΟΝΟΠΑΤΙ ΕΧΟΥΝ 1
for (i = 0; i < this.numberofOasis; i++)
{
    nodesChecked[i] = 0;
}
nodesChecked[0] = 1;
//ΤΟ ΜΟΝΟΠΑΤΙ ΑΡΧΙΖΕΙ ΑΠΟ ΤΟΝ ΚΟΜΒΟ 0
int _s_prev = 0;
//ΕΠΙΛΕΓΕΤΑΙ ΤΥΧΑΙΑ Ο ΕΠΙΟΜΕΝΟΣ ΚΟΜΒΙΟΣ ΣΤΟ ΜΟΝΟΠΑΤΙ
for (i = 1; i < this.numberofOasis; i++)
{
    int _s = Random.Range(0, this.numberofOasis);
    //ΑΝ Ο ΕΠΙΛΕΓΜΕΝΟΣ ΕΧΕΙ ΗΔΗ ΜΠΕΙ ΣΤΟ ΜΟΝΟΠΑΤΙ,
ΕΠΙΛΕΓΕΤΑΙ ΑΛΛΟΣ
    while (nodesChecked[_s] == 1)
    {
        _s = (_s + 1) % this.numberofOasis;
    }
    //ΒΡΕΘΗΚΕ Ο ΕΠΙΟΜΕΝΟΣ ΟΠΟΤΕ ΔΗΜΙΟΥΡΓΕΙΤΑΙ Η ΑΚΜΗ
    positionA[edge] = nodePositions[_s_prev];
    positionB[edge] = nodePositions[_s];
    posA[edge] = _s_prev;
    posB[edge] = _s;
    //ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ ΓΡΑΜΜΩΝ
    go = new GameObject();
    LineRenderer Line = go.AddComponent<LineRenderer>();
    //ΚΟΚΚΙΝΟ (ΣΤΟ ΕΔΑΦΟΣ)
    Color c1 = new Color(0.9f, 0.1f, 0.1f, 1);
    Line.SetColors(c1, c1);
    Line.startWidth = 0.05f;
    Line.endWidth = 0.05f;
    Line.positionCount = 2;
```

```

Line.material = new Material(Shader.Find("Sprites/Default"));
Line.SetPosition(0, nodes[_s_prev].transform.position + new
Vector3(0, 0.7f, 0));

Line.SetPosition(1, nodes[_s].transform.position + new Vector3(0,
0.7f, 0));

this.gameLines[edge] = Line;
//ΔΗΜΙΟΥΡΓΙΑ ΓΡΑΜΜΩΝ ΣΤΟ ΧΑΡΤΗ
GameObject goSky = new GameObject();
LineRenderer LineSky = goSky.AddComponent<LineRenderer>();

c1 = new Color(0.9f, 0.1f, 0.1f, 1);
LineSky.SetColors(c1, c1);
LineSky.startWidth = 3.0f;
LineSky.endWidth = 3.0f;
LineSky.positionCount = 2;
LineSky.material = new Material(Shader.Find("Sprites/Default"));
LineSky.SetPosition(0, nodes[_s_prev].transform.position + new
Vector3(0, 501.6f, 0));

LineSky.SetPosition(1, nodes[_s].transform.position + new
Vector3(0, 501.6f, 0));

this.gameLinesSky[edge] = LineSky;
//ΑΥΞΑΝΕΤΑΙ Ο ΑΡΙΘΜΟΣ ΤΩΝ ΑΚΜΩΝ
edge++;
noa++;
//Ο ΤΡΕΧΩΝ ΚΟΜΒΟΣ ΣΤΟ ΜΟΝΟΠΑΤΙ ΓΙΝΕΤΑΙ
ΠΡΟΗΓΟΥΜΕΝΟΣ
_s_prev = _s;
}

```

8 Παράρτημα Β: Δημιουργία ακμών του γράφου

```
//ΤΕΛΕΙΩΣΕ ΤΟ ΑΡΧΙΚΟ ΜΟΝΟΠΑΤΙ ΚΑΙ ΔΗΜΙΟΥΡΓΟΥΝΤΑΙ ΟΙ ΥΠΟΛΟΙΠΕΣ  
ΑΚΜΕΣ
```

```
    //ΔΗΜΙΟΥΡΓΟΥΝΤΑΙ ΟΙ ΥΠΟΛΟΙΠΕΣ ΑΚΜΕΣ  
    for (i = 0; i < this.numberOfOasis - 1; i++)  
    {  
        for (j = i + 1; j < this.numberOfOasis; j++)  
        {  
            float p = Random.Range(0.0f, 100.0f);  
            //ΔΗΜΙΟΥΡΓΙΑ ΑΚΜΗΣ ΜΕ ΠΙΘΑΝΟΤΗΤΑ 0.40  
            if (p <= 40)  
            {  
                noa++;  
                positionA[edge] = nodePositions[i];  
                positionB[edge] = nodePositions[j];  
                posA[edge] = i;  
                posB[edge] = j;  
                //ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ ΓΡΑΜΜΩΝ  
                go = new GameObject();  
                LineRenderer Line =  
go.AddComponent<LineRenderer>();  
                //ΚΟΚΚΙΝΟ (ΣΤΟ ΕΛΑΦΟΣ)  
                Color c1 = new Color(0.9f, 0.1f, 0.1f, 1);  
                Line.SetColors(c1, c1);  
                Line.startWidth = 0.05f;  
                Line.endWidth = 0.05f;  
                Line.positionCount = 2;  
                Line.material = new  
Material(Shader.Find("Sprites/Default"));  
                Line.SetPosition(0, nodes[i].transform.position +  
new Vector3(0, 0.7f, 0));  
                Line.SetPosition(1, nodes[j].transform.position +  
new Vector3(0, 0.7f, 0));  
                this.gameLines[edge] = Line;  
                //ΔΗΜΙΟΥΡΓΙΑ ΓΡΑΜΜΩΝ ΣΤΟ ΧΑΡΤΗ  
                GameObject goSky = new GameObject();
```

```

LineRenderer LineSky =
goSky.AddComponent<LineRenderer>();

c1 = new Color(0.9f, 0.1f, 0.1f, 1);
LineSky.SetColors(c1, c1);
LineSky.startWidth = 3.0f;
LineSky.endWidth = 3.0f;
LineSky.positionCount = 2;
LineSky.material = new
Material(Shader.Find("Sprites/Default"));
LineSky.SetPosition(0, nodes[i].transform.position +
new Vector3(0, 501.6f, 0));
LineSky.SetPosition(1, nodes[j].transform.position +
new Vector3(0, 501.6f, 0));
this.gameLinesSky[edge] = LineSky;

edge++;
}
if (edge == numberOfEdges)
{
break;
}
}
}

```

9 Παράρτημα Γ: Δημιουργία κόμβων

```
int k = 0;
    for (i = 0; i < this.numberOfOasis; i++)
    {
        //ΔΕΝ ΕΧΕΙ ΔΕΧΘΕΙ ΕΠΙΣΚΕΨΗ
        this.visited[i] = 0;
        //ΕΠΙΛΕΓΕΤΑΙ ΤΥΧΑΙΑ Η ΘΕΣΗ ΤΟΥ
        this.nodePositions[i] = new Vector3(Random.Range(10.0f,
480.0f), 0, Random.Range(10.0f, 480.0f));
        Quaternion rot = Quaternion.Euler(0, 0, 0);

        if ((wNodes > k))
        {
            nodes[i] = Instantiate(oasisWin, this.nodePositions[i],
rot);

            this.winningNodes[k] = i;
            k++;
        }
        else
        {
            nodes[i] = Instantiate(oasis, this.nodePositions[i],
rot);

        }
        //ΜΠΙΑΙΝΕΙ ΠΙΝΑΚΙΔΑ ΣΤΗΝ ΟΑΣΗ
        oasisNames[i] = string.Format("{0} {1}", "Oasis ", i);
        nodes[i].GetComponent<TextMesh>().text = oasisNames[i];
        //ΤΟΠΟΘΕΤΕΙΤΑΙ Ο ΚΟΜΒΟΣ ΣΤΟΝ ΧΑΡΤΗ
        Instantiate(oasisSky, this.nodePositions[i] + new
Vector3(0.0f, 501.7f, 0.0f), rot);

        //ΤΟΠΟΘΕΤΟΥΝΤΑΙ ΜΕΡΙΚΕΣ ΠΙΕΤΡΕΣ ΚΑΙ ΚΑΚΤΟΙ
        Instantiate(cactus, new Vector3(Random.Range(30.0f,
480.0f), 0.2f, Random.Range(20.0f, 480.0f)), rot);
        Instantiate(cactus1, new Vector3(Random.Range(10.0f,
490.0f), 0.2f, Random.Range(10.0f, 490.0f)), rot);
```

```

        Instantiate(cactus2, new Vector3(Random.Range(10.0f,
490.0f), 0.2f, Random.Range(10.0f, 490.0f)), rot);
        Instantiate(cactus3, new Vector3(Random.Range(30.0f,
480.0f), 0.2f, Random.Range(20.0f, 480.0f)), rot);
        Instantiate(cactus4, new Vector3(Random.Range(10.0f,
490.0f), 0.2f, Random.Range(10.0f, 490.0f)), rot);

        Instantiate(rock, new Vector3(Random.Range(30.0f, 480.0f),
0.2f, Random.Range(20.0f, 480.0f)), rot);
        Instantiate(rock1, new Vector3(Random.Range(30.0f,
480.0f), 0.2f, Random.Range(20.0f, 480.0f)), rot);
        Instantiate(rock2, new Vector3(Random.Range(30.0f,
480.0f), 0.2f, Random.Range(20.0f, 480.0f)), rot);
        Instantiate(rock3, new Vector3(Random.Range(30.0f,
480.0f), 0.2f, Random.Range(20.0f, 480.0f)), rot);
        Instantiate(rock4, new Vector3(Random.Range(10.0f,
490.0f), 0.2f, Random.Range(10.0f, 490.0f)), rot);

        Instantiate(bones1, new Vector3(Random.Range(30.0f,
480.0f), 0.2f, Random.Range(20.0f, 480.0f)), rot);

    }

```


10 Παράρτημα Δ: Εναλλαγή προβολής

```
//ΕΝΑΛΛΑΓΗ ΚΑΜΕΡΑΣ ΠΡΟΒΟΛΗ ΕΔΑΦΟΥΣ Η ΧΑΡΤΗ
```

```
    if (Input.GetKey(KeyCode.C))  
    {  
  
        this.human.enabled = false;  
        this.sky.enabled = true;  
    }  
  
    if (Input.GetKey(KeyCode.R))  
    {  
  
        this.human.enabled = true;  
        this.sky.enabled = false;  
    }  
}
```

11 Παράρτημα Ε: Έλεγχοι για την προσέγγιση κόμβου

Έλεγχος αν έχει προσεγγιστεί κάποιος από όλους τους κόμβους

```
//ΕΛΕΓΧΟΝΤΑΙ ΟΛΟΙ ΟΙ ΚΟΙΜΒΟΙ ΓΙΑ ΝΑ ΔΙΑΠΙΣΤΩΘΕΙ ΑΝ ΚΑΠΟΙΟΣ
ΒΡΙΣΚΕΤΑΙ
//ΚΟΝΤΑ ΣΤΟΝ ΠΑΙΚΤΗ
public int isClosedToNextNode()
{
    int i;
    if (method.Equals("None") == false)
    {
        float dist = Vector3.Distance(nodeToGo, transform.position);
        if (dist < 10)
        {
            return 1;
        }
        return 0;
    }
    else
    {
        for (i = 0; i < this.numberOfOasis; i++)
        {
            if (this.visited[i] == 0)
            {
                float dist = Vector3.Distance(nodePositions[i],
transform.position);
                if (dist < 10)
                {
                    //ΟΤΑΝ ΕΧΟΥΜΕ ΗΔΗ ΠΡΟΣΕΓΓΙΣΕΙ
ΤΟΝ ΕΠΟΜΕΝΟ ΚΟΜΒΟ ΤΟΤΕ
                    //ΑΥΤΟΣ Ο ΚΟΜΒΟΣ ΓΙΝΕΤΑΙ Ο
ΕΠΟΜΕΝΟΣ
                    currentNode = i;
                    return 1;
                }
            }
        }
    }
}
```

```

        }
    }
    return 0;
}
}

```

Έλεγχος αν ο χαρακτήρας είναι κοντά στον επόμενο προς επίσκεψη κόμβο

```

public int isClosedToNextNode()
{
    int i;
    if (method.Equals("None") == false)
    {
        float dist = Vector3.Distance(nodeToGo, transform.position);
        if (dist < 10)
        {
            return 1;
        }
        return 0;
    }
    else
    {
        for (i = 0; i < this.numberOfOasis; i++)
        {
            if (this.visited[i] == 0)
            {
                float dist = Vector3.Distance(nodePositions[i],
transform.position);
                if (dist < 10)
                {
                    //ΟΤΑΝ ΕΧΟΥΜΕ ΗΔΗ ΠΡΟΣΕΓΓΙΣΕΙ
TON ΕΠΙΟΜΕΝΟ ΚΟΜΒΟ ΤΟΤΕ
                    //ΑΥΤΟΣ Ο ΚΟΜΒΟΣ ΓΙΝΕΤΑΙ Ο
ΕΠΙΟΜΕΝΟΣ
                    currentNode = i;

```

```
        return 1;
    }
}
return 0;
}
```

12 Παράρτημα Z: Προβολή βοήθειας

Επιστροφή σε κανονική προβολή

```
void TaskBack()
{
    this.human.enabled = true;
    this.sky.enabled = false;
    this.sky.transform.position = this.sky.transform.position + new Vector3(0, -520, 0);
    canvas.SetActive(true);
    canvasHelp.SetActive(false);
}
```

Προβολή βοήθειας

```
void TaskHelp()
{
    string text;
    if (method == "BFS")
    {
        text = textFileBFS.text;
    }
    else if (method == "DFS")
    {
        text = textFileDFS.text;
    }
    else
    {
        text = "Δεν επιλέχθηκε μέθοδος";
    }
    helpText.text = text;

    this.human.enabled = false;
    this.sky.enabled = true;
    this.sky.transform.position = this.sky.transform.position + new Vector3(0,
520, 0);
```

```
canvas.SetActive(false);  
canvasHelp.SetActive(true);  
}
```

13 Παράρτημα Η: Ολοκλήρωση του παιχνιδιού

else

```
{  
    myText.text = "ΤΕΛΟΣ!!!";  
    StartCoroutine(sleep(50f));  
    myText.text = "ΕΠΙΣΤΡΟΦΗ ΣΤΗΝ ΑΡΧΙΚΗ ΟΘΟΝΗ ΣΕ 5";  
    StartCoroutine(sleep(50f));  
    SceneManager.LoadScene("MainMenu");  
}
```

14 Παράρτημα Θ: Υλοποίηση του αλγορίθμου BFS

```
public void BFS(int start)
{
    int i;
    int s = 0;
    //ΑΡΧΙΚΟΠΟΙΕΙΤΑΙ Ο ΠΙΝΑΚΑΣ ΠΟΥ ΔΕΙΧΝΕΙ ΑΝ ΕΝΑΣ ΚΟΜΒΟΣ
    ΕΧΕΙ ΕΞΕΤΑΣΤΕΙ
    int[] nodes_inserted = new int[this.numberOfOasis];
    for (i = 0; i < this.numberOfOasis; i++)
    {
        nodes_inserted[i] = 0;
    }
    //ΑΡΧΙΚΟΠΟΙΕΙΤΑΙ ΕΝΑΣ ΠΙΝΑΚΑΣ ΜΕ ΤΟΥΣ
    ΠΡΟΗΓΟΥΜΕΝΟΥΣ ΤΩΝ ΚΟΜΒΩΝ ΚΑΤΑ
    //ΤΗ ΣΕΙΡΑ ΕΠΙΣΚΕΨΗΣ
    int[] prev_nodes_Sequence = new int[this.numberOfOasis];
    for (i = 0; i < this.numberOfOasis; i++)
    {
        prev_nodes_Sequence[i] = -1;
    }
    this.Queue_int1 = new int[this.numberOfOasis * this.numberOfOasis];
    //ΑΡΧΙΚΟΠΟΙΕΙΤΑ Ο ΠΙΝΑΚΑΣ ΜΕ ΤΗ ΣΕΙΡΑ ΕΞΕΤΑΣΗΣ ΤΩΝ
    ΚΟΜΒΩΝ
    this.nodesSequence = new int[this.numberOfOasis * this.numberOfOasis];
    for (i = 0; i < this.numberOfOasis * this.numberOfOasis; i++)
    {
        this.nodesSequence[i] = -1;
    }
    //Ο ΤΡΕΧΩΝ ΓΙΝΕΤΑΙ Η ΑΦΕΤΗΡΙΑ
    int current = start;
    int currentPrev = -1;
    //ΤΟΝ ΒΑΖΟΥΜΕ ΣΤΗΝ ΟΥΡΑ
    this.queueSize0 = 0;
    this.queueBack = 0;
    this.queueFront = 0;
    this.queueBack0 = 0;
```



```

this.queueFront0 = 0;
this.queueSize = 0;
this.Queue_int1[this.queueBack0] = start;
//Ο ΠΡΟΗΓΟΥΜΕΝΟΣ ΚΟΜΒΟΣ ΕΙΝΑΙ ΚΕΝΟΣ
prev_nodes_Sequence[this.queueBack0] = -1;

//ΑΥΞΑΝΕΤΑΙ ΤΟ ΜΕΓΕΘΟΣ ΤΗΣ ΟΥΡΑΣ
this.queueSize0++;
//ΑΥΞΑΝΕΤΑΙ ΤΟ ΠΙΣΩ ΜΕΡΟΣ ΤΗΣ ΟΥΡΑΣ
this.queueBack0++;

//ΟΣΟ Η ΟΥΡΑ ΔΕΝ ΕΙΝΑΙ ΚΕΝΗ
while (this.queueSize0 > 0)
{
    //ΒΓΑΙΝΕΙ Ο ΠΡΩΤΟΣ ΚΟΜΒΟΣ ΑΠΟ ΤΗΝ ΟΥΡΑ
    current = this.Queue_int1[this.queueFront0];
    //ΒΓΑΙΝΕΙ ΚΑΙ Ο ΠΡΟΗΓΟΥΜΕΝΟΣ ΤΗΣ
    currentPrev = prev_nodes_Sequence[current];
    if (currentPrev != -1)
    {
        //ΑΝ ΔΕΝ ΕΠΑΝΑΛΑΜΒΑΝΕΤΑΙ
        if ((s > 0) && (nodesSequence[s - 1] != currentPrev) &&
(currentPrev != -1))
        {
            //ΜΠΙΝΕΙ Ο ΠΡΟΗΓΟΥΜΕΝΟΣ ΤΟΥ ΣΤΗΝ
            ΑΚΟΛΟΥΘΙΑ

            nodesSequence[s] = currentPrev;
            s++;
        }
    }
    //ΤΟ ΜΠΡΟΣΤΑ ΜΕΡΟΣ ΤΗΣ ΟΥΡΑΣ ΜΕΤΑΚΙΝΕΙΤΑΙ
    this.queueFront0++;
    //ΤΟ ΜΕΓΕΘΟΣ ΤΗΣ ΟΥΡΑΣ ΜΕΙΩΝΕΤΑΙ ΚΑΤΑ 1
    this.queueSize0--;
    //ΣΗΜΑΔΕΥΕΤΑΙ Ο ΤΡΕΧΩΝ
    nodes_inserted[current] = 1;
}

```

```

//ΜΠΙΑΙΝΕΙ ΣΤΗΝ ΑΚΟΛΟΥΘΙΑ
nodesSequence[s] = current;
s++;
//ΒΡΙΣΚΟΥΜΕ ΤΟΥΣ ΓΕΙΤΟΝΕΣ ΤΟΥ current ΚΑΙ ΤΟΥΣ
ΒΑΖΟΥΜΕ ΣΤΗΝ ΟΥΡΑ
int[] neighPointers = getConnectedToNode(current, posA, posB);
for (i = 0; i < this.numberOfOasis; i++)
{
    //ΑΝ ΔΕΝ ΈΧΟΥΜΕ ΑΛΛΟΥΣ ΓΕΙΤΟΝΕΣ
    if (neighPointers[i] == -1)
    {
        break;
    }
    //ΑΝ ΤΟΝ ΓΕΙΤΟΝΑ ΔΕΝ ΤΟΝ ΕΧΟΥΜΕ ΕΠΙΣΚΕΥΘΕΙ
    ΚΑΙ ΔΕΝ ΕΙΝΑΙ ΣΤΗΝ ΟΥΡΑ
    if ((nodes_inserted[neighPointers[i]] == 0) &&
(inQueue(neighPointers[i], this.Queue_int1, this.queueFront0, this.queueBack0) == 0))
    {

        prev_nodes_Sequence[neighPointers[i]] = current;
        //ΜΠΙΑΙΝΕΙ Ο ΤΡΕΧΩΝ ΤΟΥ ΣΤΗΝ
ΑΚΟΛΟΥΘΙΑ
nodesSequence[s] = current;
s++;
//ΜΠΙΑΙΝΕΙ Ο ΓΕΙΤΟΝΑΣ ΤΟΥ ΣΤΗΝ
ΑΚΟΛΟΥΘΙΑ
nodesSequence[s] = neighPointers[i];
s++;
//ΜΠΙΑΙΝΕΙ Ο ΤΡΕΧΩΝ ΤΟΥ ΣΤΗΝ
ΑΚΟΛΟΥΘΙΑ
nodesSequence[s] = current;
s++;
//ΒΑΖΟΥΜΕ ΤΟΝ ΓΕΙΤΟΝΑ ΣΤΗΝ ΟΥΡΑ
this.Queue_int1[this.queueBack0] =
neighPointers[i];

```

```

//ΣΤΗΝ ΠΑΡΑΛΛΗΛΗ ΟΥΡΑ ΒΑΖΟΥΜΕ ΤΟΝ
ΤΡΕΧΟΝΤΑ ΚΟΜΒΟ

//this.Queue_int2[this.queueBack0] = current;
//ΑΥΞΑΝΕΤΑΙ ΤΟ ΜΕΓΕΘΟΣ ΤΗΣ ΟΥΡΑΣ
this.queueSize0++;
//ΑΥΞΑΝΕΤΑΙ ΤΟ ΠΙΣΩ ΜΕΡΟΣ ΤΗΣ ΟΥΡΑΣ
this.queueBack0++;
    }
}
Debug.Log("S = " + s);
for (i = 0; i < s; i++)
{
    Debug.Log(i + ". " + nodesSequence[i]+"
"+this.numberOfOasis);
}
}
}

```

15 Παράρτημα Στ: Κώδικας ελέγχου κίνησης πάνω σε ακμή

```
//Ο ΕΛΕΓΧΟΣ ΓΙΑ ΚΙΝΗΣΗ ΣΤΑ ΟΡΙΑ ΠΟΥ ΟΡΙΖΟΥΝ ΟΙ BFS / DFS
//ΑΝ Η ΚΙΝΗΣΗ ΕΙΝΑΙ ΕΛΕΥΘΕΡΗ ΤΟΤΕ ΔΕΝ ΓΙΝΕΤΑΙ ΑΥΤΟΣ Ο
ΕΛΕΓΧΟΣ
if (method.Equals("None") == false)
{
//Ο ΧΡΗΣΤΗΣ ΑΠΟΜΑΚΡΥΝΕΤΑΙ ΑΠΟ ΤΗΝ ΑΚΜΗ ΠΟΥ
ΠΡΕΠΕΙ ΝΑ ΑΚΟΛΟΥΘΗΣΕΙ
if (this.distancePointToLine(transform.position, prevNode,
nodeToGo) > 20)
{
//ΕΜΦΑΝΙΖΕΤΑΙ ΜΗΝΥΜΑ
if (errorStatus == 0)
{
errorOnMoving++;
errorStatus = 1;
}
messages.text = "Η κίνηση γίνεται μακριά από την ακμή του
γράφου!!! ΠΡΟΣΟΧΗ " + errorOnMoving + " ΠΑΡΑΒΑΣΕΙΣ";
//ΑΝ ΕΧΟΥΝ ΓΙΝΕΙ 10 ΠΑΡΑΒΑΣΕΙΣ ΤΟ ΠΑΙΧΝΙΔΙ
ΤΕΛΕΙΩΝΕΙ
if (errorOnMoving >= this.numberOfEdges)
{
oasisLeft = -1;
messages.text = "Έγιναν πολλές παραβάσεις. Τέλος
παιχνιδιού!";
}
}
else
{
messages.text = "Η κίνηση γίνεται πάνω στην ακμή!";
errorStatus = 0;
}
}
}
```

16 Παράρτημα I: Υλοποίηση του αλγορίθμου DFS

```
public void DDFS(int start, int[] nvisited)
{
    //ΑΝ ΔΕΝ ΕΧΟΥΜΕ ΕΠΙΣΚΕΦΘΕΙ ΤΟΝ start
    if (nvisited[start] == 0)
    {
        //ΜΠΙΑΙΝΕΙ ΣΤΗ ΣΕΙΡΑ Ο start
        this.nodesSequence[this.currentInSequence] = start;
        this.currentInSequence++;
        nvisited[start] = 1;
        int[] neighbors = this.getConnectedToNode(start, this.posA,
this.posB);

        //ΓΙΑ ΚΑΘΕ ΓΕΙΤΟΝΑ ΤΟΥ
        for (int i = 0; i < this.numberOfOasis; i++)
        {
            //ΑΝ ΔΕΝ ΕΧΟΥΜΕ ΑΛΛΟΥΣ ΓΕΙΤΟΝΕΣ
            if (neighbors[i] == -1)
            {
                break;
            }
            DDFS(neighbors[i], nvisited);
        }
        //ΜΠΙΑΙΝΕΙ ΣΤΗ ΣΕΙΡΑ Ο start
        this.nodesSequence[this.currentInSequence] = start;
        this.currentInSequence++;
    }
}
```

16.1.1 Παράρτημα ΙΑ: Χειρισμός κινούμενου χαρακτήρα

```
if (Input.GetKey(KeyCode.UpArrow))
{
    transform.position = transform.position + transform.forward*0.5f;
    steps.PlayOneShot(steps.clip, 10.0f);
}

else if (Input.GetKey(KeyCode.DownArrow))
{
    transform.position = transform.position - transform.forward * 0.5f;
    steps.PlayOneShot(steps.clip, 1.0f);
}

else if (Input.GetKey(KeyCode.RightArrow))
{
    //Rotate the sprite about the Y axis in the positive direction
    transform.Rotate(new Vector3(0, 1, 0) * Time.deltaTime * m_Speed, Space.World);
    steps.PlayOneShot(steps.clip, 1.0f);
}

else if (Input.GetKey(KeyCode.LeftArrow))
{
    //Rotate the sprite about the Y axis in the negative direction
    transform.Rotate(new Vector3(0, -1, 0) * Time.deltaTime * m_Speed, Space.World);
    steps.PlayOneShot(steps.clip, 1.0f);
}

else
{
    steps.Stop();
}

float h = 15 * Input.GetAxis("Mouse X");
float v = 15 * Input.GetAxis("Mouse Y");
transform.Rotate(0, h, 0);
```

16.1.2 Παράρτημα IB: Έλεγχος παραμονής στο πεδίο

```
//ΑΝ ΔΕΝ ΕΧΟΥΜΕ ΕΠΙΣΚΕΥΤΕΙ ΟΛΟΥΣ ΤΟΥΣ ΚΟΜΒΟΥΣ
```

```
    if (this.oasisLeft > 0)
    {
        //ΚΡΑΤΑΜΕ ΤΟΝ ΑΝΘΡΩΠΟ ΜΕΣΑ ΣΤΑ ΟΡΙΑ ΤΗΣ ΠΙΣΤΑΣ
        if (transform.position.x <= 10.0f)
        {
            transform.position = new Vector3(10.0f,
transform.position.y, transform.position.z);
        }
        if (transform.position.z <= 10.0f)
        {
            transform.position = new Vector3(transform.position.x,
transform.position.y, 10.0f);
        }
        if (transform.position.y <= 0.8f)
        {
            transform.position = new Vector3(transform.position.x, 0.8f,
transform.position.z);
        }
        if (transform.position.x >= 495.0)
        {
            transform.position = new Vector3(500.0f,
transform.position.y, transform.position.z);
        }
        if (transform.position.z >= 495.0)
        {
            transform.position = new Vector3(transform.position.x,
transform.position.y, 500.0f);
        }
        if (transform.position.y >= 0.8)
        {
```

```
transform.position = new Vector3(transform.position.x, 0.8f,  
transform.position.z);  
  
}
```