

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΛΟΠΟΝΗΣΟΥ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΗΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

6/7/2020

Ηλεκτρονικά και διαδικτυακά παίγνια με HTML5

Αρβανίτης Δημήτριος

Επιβλέπωντας καθηγητής:

Γεώργιος Ασημακόπουλος

Περιεχόμενα

Εισαγωγή	4
Introduction.....	5
Κεφάλαιο 1ο	6
1.1 Γλώσσα προγραμματισμού- HTML	6
1.2 Χαρακτηριστικά HTML.....	6
1.3 HTML 5	9
1.4 Γλώσσα προγραμματισμού PHP.....	10
1.5 JAVA	11
1.5.1 Χαρακτηριστικά της Java	14
1.5.2 Εργαλεία της Java και τύποι δεδομένων.....	17
Κεφάλαιο 2 ^ο	23
2.1 Ηλεκτρονικά παιχνίδια - Ιστορία	23
2.2 Αλλαγές στην κατασκευή παιχνιδιών για ηλεκτρονικούς υπολογιστές	26
2.3 Διάκριση ηλεκτρονικών παιχνιδιών.....	27
2.4 Πλεονεκτήματα ψηφιακών παιχνιδιών	29
Κεφάλαιο 3 ^ο	30
3.1 Tic Tac Toe Ιστορία.....	30
3.1.2 Παίζοντας Tic Tac Toe στο διαδίκτυο.....	30
3.2 Στρατηγική.....	30
3.3 Τεχνητή νοημοσύνη	32
3.4 Παρουσίαση εφαρμογής	34
3.4.1 Ανάλυση απαιτήσεων	38
3.5 Ανάλυση κώδικα.....	41
Βιβλιογραφία	45

Εικόνα 1 Γλώσσα HTML	6
Εικόνα 2 εκδόσεις HTML.....	9
Εικόνα 3 Γλώσσα PHP	11
Εικόνα 4 Γλώσσα Java.....	12
Εικόνα 5 Σύγκριση μεταγλώττισης προγραμμάτων C++ και Java	13
Εικόνα 6 Επιλογές με το Jar	20
Εικόνα 7 Τύποι δεδομένων στη Java	23
Εικόνα 8 DiGRA αναφορές	26
Εικόνα 9 Θέσεις νίκης	31
Εικόνα 10 Θέσεις ήττας	31
Εικόνα 11 Λογικό διάγραμμα αναπαράστασης του κώδικα.....	32
Εικόνα 12 Τμήματα του δέντρου αναζήτησης για το παιχνίδι Tic Tac Toe.....	33
Εικόνα 13 Αρχική οθόνη εφαρμογής.....	35
Εικόνα 14 Επιλογή επιπέδου παιχνιδιού.....	36
Εικόνα 15 Περίπτωση ισοπαλίας.....	37
Εικόνα 16 Περίπτωση νίκης	38

Εισαγωγή

Η συγκεκριμένη εργασία ασχολείται με το σχεδιασμό και την ανάπτυξη ενός ψηφιακού παιχνιδιού. Γίνεται μία σύντομη αναφορά στα εργαλεία που χρησιμοποιούνται για την κατασκευή παιχνιδιών ηλεκτρονικού υπολογιστή καθώς και μία αναδρομή στα παιχνίδια. Στη συνέχεια θα περιγραφεί η λειτουργία του παιχνιδιού και τμηματικά κάποια μέρη κώδικα. Καθώς το παιχνίδι που υλοποιήθηκε λειτουργεί και με αντίπαλο τον ηλεκτρονικό υπολογιστή, κρίθηκε αναγκαία η αναφορά στη λειτουργία της τεχνητής νοημοσύνης.

Η κατασκευή παιχνιδιού είναι μία διαδικασία ανάπτυξης λογισμικού μέσω της οποίας ο χρήστης μπορεί να δημιουργήσει τα δικά του παιχνίδια. Πρόκειται για έναν κλάδο, ο οποίος αναπτύσσεται μέρα με τη μέρα καθώς δημιουργούνται καινούργιες τεχνολογίες που βοηθούν στο να γίνουν πιο λεπτομερή τα παιχνίδια και πιο αποδοτικά τόσο από άποψη γραφικών όσο και από άποψη ενδιαφέροντος. Η ανάλυση του συγκεκριμένου παιχνιδιού είναι απαραίτητη για την κατανόησή του.

Το παιχνίδι που αναπτύχθηκε είναι ένα παιχνίδι τύπου τρίλιζας το Tic Tac Toe, ένα δημοφιλές παιχνίδι το οποίο μπορεί να παιχτεί είτε μεταξύ δύο παιχτών είτε ενός με αντίπαλο τον υπολογιστή. Επίσης το παιχνίδι μπορεί να παιχτεί και μέσω διαδικτύου.

Η Java, ως εργαλείο προγραμματισμού, κατάφερε να εξαλείψει πολύπλοκες διαδικασίες και να βοηθήσει στην ανάπτυξη απλών και φορητών παιχνιδιών. Για το σκοπό αυτό, επιλέχθηκε για την κατασκευή του παιχνιδιού. Άλλος ένας λόγος είναι η ευελιξία ενός ανοιχτού κώδικα παιχνιδιού, το οποίο μπορεί να τροποποιηθεί και να αλλάξει μελλοντικά.

Introduction

This paper deals with the design and development of a digital game. There is a brief reference to the tools used to make computer games as well as a review of the games. Then the game function and some parts of the code will be described. As the game is run on a computer, it was deemed necessary to refer to the operation of artificial intelligence.

Game development is a software development process through which the user can create their own games. It is an industry that is evolving day by day as new technologies are created that help to make games more detailed and more efficient both in terms of graphics and interest. The analysis of this game is necessary for its understanding.

The game developed is a Tic Tac Toe, a popular game that can be played either between two players or one with a computer opponent. The game can also be played online.

Java, as a programming tool, has managed to eliminate complex processes and help develop simple and portable games. For this purpose, it was selected for the construction of the game. Another reason is the flexibility of an open source game code, which can be modified and changed in the future.

Κεφάλαιο 1ο

1.1 Γλώσσα προγραμματισμού- HTML

Ο παγκόσμιος ιστός (World Wide Web) υλοποιείται από διάφορες ιστοσελίδες (webpages), οι οποίες οφείλουν την ανάπτυξή τους σε ένα σύνολο τεχνολογιών, στο κέντρο των οποίων τοποθετείται η HTML (Hyper Text Markup Language) που είναι γνωστή κι ως Γλώσσα Μορφοποίησης Υπερκειμένου. Σε γενικές γραμμές η HTML δύναται να παρουσιάσει το περιεχόμενο των ιστοσελίδων, να καθορίσει ουσιαστική τη μορφή με την οποία θα εμφανιστούν κείμενα και εικόνες, έχοντας μία σχετικά απλή διαμόρφωση.



Εικόνα 1 Γλώσσα HTML

Ο λόγος για τον οποίο έχουν αναπτυχθεί παραπάνω από μία τεχνολογίες για να παραχθεί ένα αποτέλεσμα, η ιστοσελίδα, είναι απλός, αν λάβουμε υπόψη ότι απαιτούνται πληθώρα διαφορετικών δεξιοτήτων. Όσον αφορά την εμφάνιση απαιτούνται γνώσεις πάνω στα γραφιστικά, ενώ για την διαμόρφωση και την αλληλεπίδραση απαιτούνται γνώσεις πάνω στον προγραμματισμό και το σχεδιασμό.

1.2 Χαρακτηριστικά HTML

Η HTML ορίζει ένα σύνολο κοινών στυλ που χρησιμοποιούνται στις Web σελίδες. Τα στυλ αυτά εμπεριέχουν τίτλους (titles) που είναι απαραίτητοι καθώς υποδεικνύουν το περιεχόμενο και το σκοπό της σελίδας, επικεφαλίδες (headings) που χρησιμοποιούνται προκειμένου οι ενότητες του κειμένου να διαχωριστούν, παραγράφους (paragraphs), λίστες (lists) καθώς και πίνακες (tables). Ορίζει επίσης, στυλ χαρακτήρων μεταξύ των

οποίων η έντονη γραφή (boldface) και οι ενότητες κώδικα. Στην HTML παρατηρούνται πέντε είδη λιστών 1) οι αριθμημένες λίστες, οι οποίες κάνουν χρήση αριθμών για τα στοιχεία τους, 2) οι λίστες κουκκιδίων, 3) οι λίστες γλωσσαρίου, οι οποίες χρησιμοποιούν όρους ή ορισμούς για κάθε τους στοιχείο, 4) οι λίστες σε μορφή μενού, που χρησιμοποιούνται για ειδικές διατάξεις και 5) οι λίστες καταλόγου, που χρησιμοποιούνται για να παρουσιάσουν σύντομα στοιχεία.

Βασίζεται στην έννοια της ετικέτας (TAG) ή μίας δεδομένης λέξης. Πρόκειται για μία λέξη κλειδί, η οποία έχει αρχή, τέλος και περιεχόμενο και για να δηλωθεί επιστρατεύονται τα σύμβολα < >. Οι ετικέτες συνιστούν βασικά δομικά στοιχεία όλων των διαδικτυακών τόπων. Βασικό χαρακτηριστικό της γλώσσας είναι η μορφοποίηση κειμένου, ωστόσο παρέχει τη δυνατότητα να ενσωματωθούν εικόνες και άλλα αντικείμενα καθώς και να δημιουργηθούν φόρμες με σκοπό να υπάρξει αλληλεπίδραση με τον χρήστη. Δίνοντας εντολές στους περιηγητές (browsers) μέσω των ετικετών και του κειμένου, εξηγεί σε αυτούς τον τρόπο που θα διαβάσουν τα έγγραφα της αλλά και τον τρόπο που θα συνθέσουν τις σελίδες που θα απεικονιστούν στις οθόνες. Ο εκάστοτε περιηγητής δύναται να εμφανίσει το ίδιο αρχείο με διαφορετικό τρόπο σύμφωνα με τα χαρακτηριστικά που έχει το πρόγραμμα περιήγησης που επιλέχθηκε από τον καθένα.

Επιπρόσθετα, πρέπει να σημειωθούν τα εξής, ότι η HTML δεν διαχωρίζει τα κεφαλαία από τα μικρά γράμματα, κάτι που συμβαίνει σε άλλες γλώσσες προγραμματισμού. Δεν εστιάζει στα κενά, αλλά τα αγνοεί. Όσον αφορά τα λάθη στο συντακτικό φαίνεται να τα «συγχωράει», αφού εν αντιθέσει με άλλες γλώσσες προγραμματισμού προσπαθεί τα λάθη αυτά να τα διορθώσει και να εμφανίσει την ιστοσελίδα. Στις υπόλοιπες γλώσσες ακόμη και ένα μικρό συντακτικό λάθος καθιστά την μεταγλώττιση αδύνατη. Ωστόσο, είναι φυσικό πως όταν τα λάθη είναι σημαντικά η εμφάνιση δεν είναι σωστή.

Η HTML περιγράφεται δημοσίως για πρώτη φορά το 1991 (Bemers-Lee, 1991). Σε κείμενο που δημοσιεύτηκε φαίνεται πως ο ίδιος ο συγγραφέας προ¹ότεινε και σχεδίασε ένα σύστημα, το οποίο οι ερευνητές του CERN στο οποίο απασχολούνταν ως φυσικός, θα έχουν τη δυνατότητα να χρησιμοποιούν αλλά και να ανταλλάσσουν κείμενα. Ήδη από το 1989 είχε γράψει την πρόταση του αναφορικά με ένα σύστημα υπερκειμένου, το οποίο θα λειτουργεί με βάση το Internet ώσπου το 1990 τυποποίησε την HTML, γράφοντας το λογισμικό για την αξιοποίηση της (ένας εξυπηρετητής και ένας πελάτης-

¹ Cern: εργαστήριο φυσική στη Γενεύη.

server-client). Συνοδοιπόρος και συνεργάτης στη προσπάθεια αυτή, υπήρξε ο μηχανικός λογισμικού Robert Cailliau, ο οποίος υποστήριξε το κομμάτι του WWW. Εκτός από το tag υπερσυνδέσμου, η HTML επηρεάστηκε σημαντικά από την SGML.guid, τρόπος με τον οποίο συνέγραφαν εγχειρίδια στο CERN και η οποία βασιζόταν στην Standard Generalized Markup Language (SGML).

Αναφορικά με την ιδέα του υπερκειμένου να σημειωθεί πως ο Αμερικανός μηχανικός εφευρέτης Vannevar Brush την ανέφερε κατά τη δεκαετία του '40 σε δημοσιευμένο άρθρο του, με τίτλο "As we May Think". (Vannevar, 1945). Η διαδικασία δημιουργίας ενός αρχείου HTML δεν είναι δύσκολη. Αρκεί η ύπαρξη ενός διορθωτή κειμένου, για παράδειγμα το Notepad++ που διαθέτουν τα Windows. Έπειτα, γράφεται ο επιθυμητός κώδικας html προς εκτέλεση και αποθηκεύεται με την κατάληξη .html ή .htm. Πιο συνηθισμένη είναι η κατάληξη .html.

Με την HTML σηματοδοτήθηκε η αρχή του WWW όπως είναι γνωστό σήμερα, με τις σελίδες που αποτελούσαν τη βάση του WWW να είναι γραμμένες στην πρώτη έκδοση της γλώσσας HTML. Το 1994 κυκλοφορεί η έκδοση HTML 2.0 από την Internet Engineering Task Force. Η έκδοση που την ακολούθησε ήταν η 3.0, την οποία δεν δέχθηκαν ούτε η Microsoft ούτε η Netscape και έτσι το 1996 αντικαταστάθηκε από την έκδοση HTML 3.2. Η έκδοση 3.2 συμπεριέλαβε πληθώρα σημάνσεων (tags) από τις προαναφερθείσες εταιρείες. Τον Ιούνιο του 1997 παρουσιάστηκε η έκδοση 4.0, ενώ η HTML 5, συνιστά μία πρόσφατη έκδοση, η προσπάθεια της οποίας ξεκίνησε το 2007.

Στον Πίνακα 1. που ακολουθεί παρουσιάζονται οι εκδόσεις τις HTML με κάποια χαρακτηριστικά τους.

HTML 2.0	Διαθέτει τα περισσότερα στοιχεία που έχουν αναφερθεί, ωστόσο δεν υποστήριζε πίνακες και ιδιότητες στοίχισης
HTML 3.	Ο Dave Raggett επιχείρησε την αναβάθμιση των χαρακτηριστικών και των βοηθημάτων της HTML, χωρίς όμως να ολοκληρώσει τη προσπάθεια του.
HTML 3.2	Υποστηρίζει πίνακες, εικόνες, επικεφαλίδες, ιδιότητες στοίχισης και γενικά σχολαστικές λεπτομέρειες. Κατανοητή από κάθε περιηγητή του διαδικτύου. Απουσιάζουν από αυτή επεκτάσεις για Internet Explorer και Netscape.

HTML 4.01	Υποστηρίζει επεκτάσεις (Frames, Embed, Applets) και επιπρόσθετα χαρακτηριστικά (έγγραφα σε πληθώρα γλωσσών, CSS, περισσότερους πίνακες και φόρμες, συμβατότητα με JavaScript)
HTML 5	Μειώνει την ανάγκη για ιδιόκτητα plug-ins και πλούσιες διαδικτυακές εφαρμογές (Adobe Flash, Microsoft Silverlight, Apache Pivot, Sun JavaFX)

Εικόνα 2 εκδόσεις HTML

1.3 HTML 5

Κύριο γνώρισμα της HTML 5 είναι η δημιουργία “mobile web sites” προκειμένου να προσφέρονται ταυτόχρονα πληθώρα ιδιοτήτων που υπάρχουν σε εφαρμογές, οι οποίες συναντώνται σε πλατφόρμες κινητών. Έτσι, το διαδίκτυο παρουσιάζεται και στην κινητή τηλεφωνία. Η νέα αυτή γλώσσα δημιουργήθηκε από το World Wide Web Consortium (W3C) και το Web Hypertext Application Technology Working Group (WHATWG) προκειμένου να υπάρξουν υψηλά standards και πλήρη συμβατότητα.

Με την HTML 5 προσφέρεται πάνω στο λογισμικό του κινητού GPS, εργαλείο ιδιαίτερα χρήσιμο όχι μόνο για το ηλεκτρονικό εμπόριο αλλά και για τις διαφημίσεις στο διαδίκτυο. Χαρακτηριστικό γνώρισμα της έκδοσης αυτής αποτελεί και η αναπαραγωγή βίντεο σε mobile web sites, δίχως να χρησιμοποιηθεί το Flash. Στη συνέχεια γίνεται λόγος σε κάποιους βασικούς κανόνες, οι οποίοι έχουν οριστεί για την συγκεκριμένη έκδοση:

- Τα νέα χαρακτηριστικά θέτουν ως βάση τους τα HTML, CSS (Cascading Style Sheets/ Διαδοχικά Φύλλα Στυλ), DOM (Document Object Model) και JavaScript.
- Τα plugins, δηλαδή τα συστήματα συστατικών κάποιων λογισμικών, τα οποία προσθέτουν δυνατότητες σε ένα μεγαλύτερο λογισμικό, όπως είναι η προβολή βίντεο, η ανίχνευση ιών, η επίδειξη νέων τύπων αρχείων (π.χ. Adobe Flash Player και Quick Time) ελαχιστοποιούνται.
- Τα λάθη εντοπίζονται από καλύτερες λειτουργίες.
- Για την αντικατάσταση του scripting θα πρέπει να εφαρμοστεί περισσότερο markup.
- Ύπαρξη πλήρους συμβατότητας ανεξάρτητα από την συσκευή.

1.4 Γλώσσα προγραμματισμού PHP

Αρχικά, η PHP είναι μία γλώσσα προγραμματισμού με σκοπό να δημιουργούνται ιστοσελίδες με δυναμικό περιεχόμενο. Τι σημαίνει όμως αυτό; σημαίνει πως χάρη σε αυτή μία ιστοσελίδα μπορεί και αποκτά διαδραστικότητα. Έτσι παραδείγματος χάρη, εάν ένας χρήστης θέλει να συνδεθεί στο eshop ενός καταστήματος, με την εισαγωγή ενός κωδικού και την επαλήθευση αυτού, ο χρήστης επιτυγχάνει την είσοδο/σύνδεση του σε αυτή την ιστοσελίδα. Το WordPress αλλά και το Facebook, μεγάλα συστήματα blog, χρησιμοποιούν την PHP. Όσον αφορά τα αρχεία της PHP αυτά είναι δυνατόν να περιέχουν κείμενο, κώδικα HTML, CSS, JavaScript και PHP. Ο κώδικας εκτελείται στον server, ενώ το αποτέλεσμα επιστρέφει στον browser, ως μία απλή σελίδα HTML.

Η PHP ξεκίνησε να αναπτύσσεται από τον Rasmus Lerdorf το 1994, γράφοντας μια σειρά από Common Gateway Interface (CGI) script στη γλώσσα Perl. Τα χρησιμοποίησε στη δική του σελίδα με σκοπό να εκτελούνται εργασίες, όπως η εμφάνιση βιογραφικού ή η καταγραφή της κυκλοφορίας στην ιστοσελίδα του. Έπειτα, προχώρησε στην εκ νέου συγγραφή των script στη γλώσσα C, προσθέτοντας σε αυτά δυνατότητες συνεργασίας με web φόρμες και επικοινωνίας με βάσεις δεδομένων. Της έδωσε το όνομα Personal Home Page/ Forms Interpreter (PHP/FI). Η γλώσσα PHP (HyperText PreProcessor) κυκλοφόρησε δημόσια το 1995 (8 Ιουνίου) με βασικό χαρακτηριστικό ότι κώδικας μεταγλωττίζεται στον server πρώτα και έπειτα φορτώνεται σε κάποιο άλλο έγγραφο html (Βόγκλης, & Καραβασίλης, 2014). Έτσι ο χρήστης δεν δύναται να δει τον αρχικό κώδικα.

Είναι σημαντικό να σημειωθεί πως η PHP δεν δημιουργήθηκε με σκοπό να αποτελέσει γλώσσα προγραμματισμού. Ωστόσο, μεγάλωσε σταδιακά όταν ο Lerdorf συνέχισε το επόμενο λογικό βήμα προς την ανάπτυξη. Αφού συστάθηκε μία ομάδα ανάπτυξης, η οποία προχώρησε σε δοκιμές beta για αρκετούς μήνες, η PHP/FI 2, κυκλοφόρησε επίσημα το 1997 (2 Νοεμβρίου). Χάρη στην PHP και την τεχνολογία της είναι δυνατόν να δημιουργηθούν δυναμικές ιστοσελίδες (dynamic web pages). Τεχνολογίες οι οποίες λειτουργούν ανταγωνιστικά στην PHP είναι η ASP (Active Server Pages), από την εταιρεία Microsoft και η CFML (ColdFusion Markup Language).



Εικόνα 3 Γλώσσα PHP

Η PHP βρέθηκε στο επίκεντρο πολλών επικριτών, οι οποίοι υποστήριζαν πως εξαιτίας της μη ύπαρξης ενός αρχικού σχεδιασμού, και εξαιτίας της οργανικής της ανάπτυξης, οι λειτουργίες και οι παράμετροι της ονοματοποιήθηκαν χωρίς συνέπεια. Η συντακτική ανάλυση (parser) της PHP ξαναγράφηκε το 1997 από τους Zeev Suraski και Andi Gutmans, αποτελώντας τη βάση της PHP 3. Η γλώσσα άλλαξε όνομα με το σημερινό αναδρομικό ακρωνύμιο PHP και εκδόθηκε τον Ιούνιο του 1998. Το 1999 οι ίδιοι προσπάθησαν να γράψουν ξανά τον πυρήνα της PHP, παράγοντας κατ' αυτόν τον τρόπο το Zend Engine, αποτελώντας τη βάση της PHP 4. Τον Ιούλιο του 2004 η PHP 5 κυκλοφορεί, με τροφοδότηση από το πιο νέο Zend Engine II, περιλαμβάνοντας νέα χαρακτηριστικά, μεταξύ των οποίων η ανανεωμένη υποστήριξη για τον αντικειμενοστραφή προγραμματισμό, τις επεκτάσεις αντικειμένων PHP Data (POP), καθώς και πληθώρα βελτιώσεων στις εκδόσεις. Έκτοτε η PHP 5 συνιστά τη μόνη σταθερή έκδοση, η οποία αναπτύσσεται διαρκώς, φέρνοντας μέχρι σήμερα στο προσκήνιο αρκετές υποεκδόσεις.

1.5 JAVA

Ένας web browser στοχεύει στο να διαβάσει τα έγγραφα HTML και στη συνέχεια να τα συνθέσει σε σελίδες, οι οποίες είναι δυνατόν να διαβαστούν ή να ακουστούν. Ο browser, ωστόσο δεν δύναται να εμφανίσει τις ετικέτες HTML, αλλά να τις χρησιμοποιήσει με σκοπό την ερμηνεία του περιεχομένου της σελίδας. Η HTML και τα στοιχεία της χρησιμοποιούνται για την δημιουργία ιστοτόπων, επιτρέποντας ενσωμάτωση εικόνων και διάφορων αντικειμένων σε αυτούς, ενώ παράλληλα χρησιμοποιείται για την εμφάνιση διαδραστικών φορμών. Ακόμη, η γλώσσα HTML προσφέρει μεθόδους, ικανές να δημιουργήσουν δομημένα έγγραφα, τα οποία αποτελούνται από το μεταφερόμενο περιεχόμενο και τον κώδικα μορφοποίησης του

περιεχομένου, καθορίζει τα δομικά στοιχεία, τα οποία είναι καθοριστικής σημασίας για το κείμενο (π.χ. επικεφαλίδες, παραγράφους, λίστες, συνδέσμους) και ενσωματώνει σενάρια εντολών σε άλλες γλώσσες.

Η PHP αποτελεί ίσως την πιο διαδεδομένη τεχνολογία του Παγκόσμιου Ιστού, αν ληφθούμε υπόψη ότι ένας μεγάλος αριθμός εφαρμογών και ιστοτόπων την χρησιμοποιούν. Το Σύστημα Διαχείρισης Περιεχομένου αποτελεί την πιο διάσημη εφαρμογή που χρησιμοποιεί την γλώσσα PHP.

Με την ραγδαία ανάπτυξη του Internet και του Παγκόσμιου Ιστού (WWW) δημιουργήθηκε η ανάγκη για νέους τρόπους ανάπτυξης αλλά και διανομής λογισμικού. Οι νέες αυτές απαιτήσεις οδήγησαν στον δημιουργηθεί η Java, μία γλώσσα προγραμματισμού από την εταιρεία Sun Microsystems. Ο σχεδιασμός της γλώσσας αυτής οφείλεται στη προσπάθεια να αναπτυχθούν εφαρμογές, ικανές να «τρέξουν» σε ετερογενή δικτυακά περιβάλλοντα.

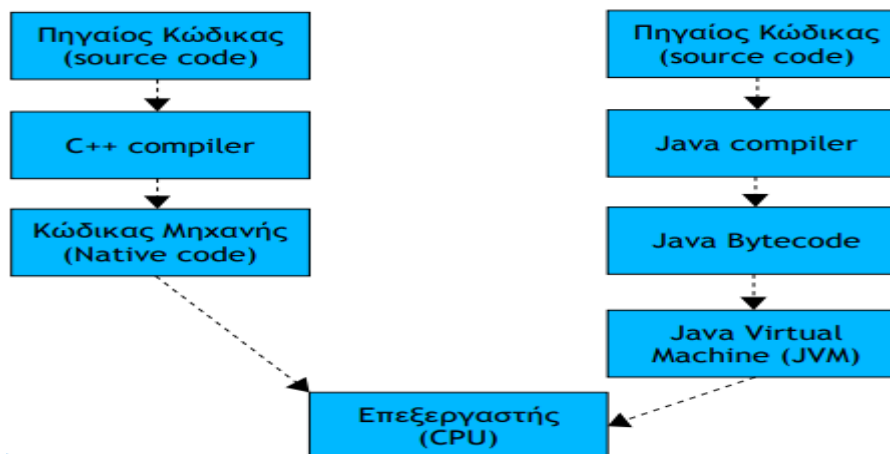
Η γλώσσα JAVA ξεκίνησε την ανάπτυξη της από την Sun Microsystems από τους James Gosling και Bil Joy, το 1991. Στην αρχή αποτέλεσε κομμάτι από ένα ερευνητικό έργο ανάπτυξης λογισμικού για ηλεκτρονικές συσκευές σε καταναλωτικό επίπεδο, όπως η τηλεόραση και το βίντεο. Έπειτα, ο τρόπος με τον οποίο χρησιμοποιήθηκε, τη μετέτρεψε σε μία ιδανική γλώσσα με σκοπό να διανείμει εκτελέσιμα προγράμματα μέσω του Παγκόσμιου Ιστού (WWW) αλλά και σε μία γλώσσα προγραμματισμού, με σκοπό να αναπτυχθούν προγράμματα, τα οποία θα μπορούν να χρησιμοποιηθούν αλλά και να μεταφερθούν εύκολα στα διάφορα λειτουργικά συστήματα.



Εικόνα 4 Γλώσσα Java

Ως γλώσσα βασίστηκε στην C++. Ουσιαστικά και στις δύο γλώσσες παρατηρείται η ίδια σύνταξη, μόνο που εντοπίζονται κάποια πρόσθετα χαρακτηριστικά και

διαφοροποιήσεις. Η Java θεωρείται μία καλά οργανωμένη και εκφρασμένη γλώσσα προγραμματισμού, η οποία εκτός από την C++ έχει δανειστεί και από άλλες γλώσσες χαρακτηριστικά, μεταξύ των οποίων η SmallTalk και η Lisp. Κράτησε δηλαδή τα χαρακτηριστικά εκείνα που της ήταν χρήσιμα, και αφαίρεσε κάθε στοιχείο από αυτές, που κατά τη χρησιμοποίησή τους, οι χρήστες θα οδηγούνταν σε σύγχυση. Στην Εικόνα που ακολουθεί απεικονίζεται η διαδικασία μεταγλώττισης σε ένα πρόγραμμα C++ και σε ένα πρόγραμμα Java.



Εικόνα 5 Σύγκριση μεταγλώττισης προγραμμάτων C++ και Java

Η Sun Microsystems τη χρησιμοποίησε σε πληθώρα εφαρμογών, έχοντας το όνομα Οακ, προκειμένου να δημιουργηθούν προϊόντα για την ηλεκτρονική αγορά. Το κοινό, ωστόσο, προσελκύνθηκε όταν η γλώσσα συνδυάστηκε με τον πρόγραμμα ανάγνωσης (browser) της HotJava. Απόρροια του συνδυασμού αυτού ήταν η στενή σύνδεση της γλώσσας με την προώθηση διαδικτυακών μικρο-εφαρμογών. Η Java απογειώθηκε όταν η Netscape ενσωμάτωσε τη HotJava και τις δυνατότητες της, με σκοπό οι μικρο-εφαρμογές αυτές να «τρέχουν» και στο δικό της πρόγραμμα οργάνωσης ιστοσελίδων.

Τα τελευταία χρόνια η Java χρησιμοποιείται, όπως αναφέρθηκε, για να αναπτυχθούν μικρο-εφαρμογές. Αυτό, όμως, δε σημαίνει ότι κρίνεται ακατάλληλη για τη δημιουργία ολοκληρωμένων εφαρμογών. Απόδειξη αυτού, το γεγονός ότι ένας μεγάλος αριθμός εργαλείων έχουν γραφτεί με την γλώσσα Java. Η θεωρία ανάπτυξης μεταγλωττιστών υποστηρίζει πως για να «ενηλικιωθεί» μία γλώσσα θα πρέπει οι μεταγλωττιστές να

μπορούν από την ίδια γλώσσα να γράψουν. Λαμβάνοντας αυτό υπόψη, η java, ως γλώσσα προγραμματισμού έχει «ενηλικιωθεί».

1.5.1 Χαρακτηριστικά της Java

Η Java έχει χαρακτηριστεί από τους δημιουργούς της ως μία «απλή, αντικειμενοστρεφής, διαμοιραζόμενη, εύρωστη, ασφαλής, έχοντας ουδέτερη αρχιτεκτονική, εύκολα μεταφερόμενη, με υψηλή απόδοση και πολυνηματική». Έχει στη διάθεση της αρκετά χαρακτηριστικά, τα οποία δεν τα συναντά κανείς σε άλλη γλώσσα. Για να αντιληφθούν και να κατανοηθούν οι παραπάνω ορισμοί θα πρέπει να μελετηθούν ξεχωριστά.

➤ Αρχεία JAR (Java Archive Files):

Η Java έχει προκύψει από την C++, με την αφαίρεση κάποιων χαρακτηριστικών. Άρα, θεωρείται πιο εύκολη από αυτήν, ειδικά αν ληφθεί υπόψη ότι η πρώτη γλώσσα που μαθαίνει κάθε προγραμματιστής είναι η C++. Οι βασικοί τύποι δεδομένων που περιλαμβάνονται στη γλώσσα Java είναι τρεις, οι αριθμοί, οι λογικές μεταβλητές (Boolean) και οι πίνακες. Κάθε τι άλλο αποτελεί μία κλάση. Παραδείγματος χάρη, τα αλφαριθμητικά αποτελούν αντικείμενα και όχι έναν απλό πίνακα χαρακτήρων.

Μεταξύ των χαρακτηριστικών που έχουν αφαιρεθεί από την C++ είναι η δήλωση goto, η υπερφόρτωση των τελεστών, κάτι που σύγχυζε τους προγραμματιστές, οι δομές, τα ενωτικά, οι δηλώσεις #define και Typedef. Πρόκειται για χαρακτηριστικά, που ο προγραμματιστής τα χρειάζεται στη C++ για τη σωστή μεταγλώττιση του κώδικα που υπάρχει ήδη στη C, και ο οποίος σχετίζεται άμεσα με τα αυτά τα χαρακτηριστικά. Η Java παρέλειψε από την C++ το στοιχείο της άμεσης διαχείρισης της μνήμης, μέσω της χρησιμοποίησης δεικτών, τους οποίους ο χρήστης είναι πιθανό να ορίσει κατά λάθος σε μία λάθος μεταβλητή, με αποτέλεσμα το πρόγραμμα να μην λειτουργεί σωστά, με πιθανότητα το σύστημα να καταρρεύσει. Επιπλέον, με τους δείκτες αποθηκεύεται η κατανεμημένη μνήμη, η οποία σε περίπτωση που δεν ελευθερωθεί το πρόγραμμα σταδιακά, θα δεσμεύσει μνήμη μέχρι να μην υπάρχει στη διάθεση μας καθόλου μνήμη. Τα Bounds Checker, είναι προϊόντα, τα οποία δημιουργήθηκαν με σκοπό την επίλυση προβλημάτων από τους προγραμματιστές, προβλήματα που οφείλονται στους δείκτες.

Ωστόσο, με την Java τέτοιου είδους προβλήματα απλοποιούνται, με την αφαίρεση της έννοιας των δεικτών από τη γλώσσα προγραμματισμού.

Εκτός από την αφαίρεση στοιχείων παρατηρείται και πρόσθεση στοιχείων, καθιστώντας τη όχι έναν φτωχό μεταγλωττιστή αλλά μία νέα γλώσσα προγραμματισμού. Το στοιχείο της αυτόματης διαχείρισης της μνήμης, το επονομαζόμενο «μονάδα συλλογής σκουπιδιών (garbage collector)» είναι ένα από αυτά. Έτσι, ο εκάστοτε προγραμματιστής, δεν είναι υπεύθυνος για την απελευθέρωση της μνήμης που κατανέμει, αφού η εικονική μηχανή της Java (Virtual Machine ή VMJ) είναι υπεύθυνη γι' αυτό.

Επιπρόσθετα, με την Java υποστηρίζεται η δημιουργία πολυνηματικών προγραμμάτων, τα οποία είναι σχεδιασμένα με τέτοιο τρόπο που είναι σε θέση να εκτελούν πολλαπλές δουλειές την ίδια στιγμή.

➤ Αντικειμενοστραφής:

Σχετικά με τα αντικείμενα στη γλώσσα Java, να σημειωθεί ότι στο μεγαλύτερο βαθμό ακολουθεί τα τη γλώσσα SmallTalk. Τα αντικείμενα συνιστούν κάθε τι εκτός από τους τύπους των δεδομένων, με τη κάθε συνάρτηση να καλείται μέσα από ένα αντικείμενο. Βασικό της χαρακτηριστικό είναι πως δεν προχωρά στην υιοθέτηση χαρακτηριστικών πολλαπλής κληρονομικότητας. Κάθε κλάση αποτελείται από μεθόδους και μεταβλητές, όπου μέθοδοι είναι οι συναρτήσεις όπου απευθύνεται το αντικείμενο μιας κλάσης, και μεταβλητές είναι τα δεδομένα, όπου καθορίζουν τη κατάσταση του αντικειμένου.

➤ Διαμοιραζόμενη:

Με την Java επιτρέπεται να δημιουργηθούν διαμοιραζόμενες εφαρμογές από μία σειρά κλάσεων, οι οποίες συναντώνται σε εφαρμογές στο διαδίκτυο. Όταν χρησιμοποιείται η URL κλάση της Java, η εκάστοτε εφαρμογή είναι σε θέση να προσπελάσει σχετικά εύκολα κάποιος διακομιστή, ο οποίος είναι απομακρυσμένος.

➤ Εύρωστη:

Ο σχεδιασμός της Java είναι τέτοιος, ώστε η γλώσσα να χρησιμοποιείται για την επίλυση πολύπλοκων προγραμματιστικών προβλημάτων. Το να δημιουργηθεί ένα πρόγραμμα διαμοιραζόμενο και πολυνηματικό, το οποίο θα είναι ικανό να «τρέξει» σε πληθώρα λειτουργικών συστημάτων, όπου υπάρχει διαφοροποίηση στους επεξεργαστές, δεν είναι εύκολα εγχείρημα. Αυτός είναι και ο λόγος άλλωστε που

δημιουργήθηκε αυτή η δυνατή γλώσσα. Ο εμπλουτισμός της διαχείρισης της μνήμης έχει γίνει με δύο τρόπους. Από τη μία, οι δείκτες δεν χρησιμοποιούνται, και άρα το πρόγραμμα της Java δεν δύναται να καταστρέψει δεδομένα ακόμα και να γράψει πάνω σε κάποια άλλα. Από την άλλη, ακολουθώντας τις γλώσσες Lisp και SmallTalk, απελευθερώνει αυτόματα τη μνήμη, η οποία δεν χρησιμοποιείται πια και είναι καταμερισμένη.

➤ Ασφάλεια:

Η Java διαθέτει ισχυρούς μηχανισμούς ασφάλειας. Έχει σχεδιαστεί ώστε να είναι ασφαλής, με τον μεταγλωττιστή να διασφαλίζει πως κανένας κανόνας ασφάλειας δεν θα παραβιάζεται από τον πηγαίο κώδικα. Έπειτα, ο κώδικας που εκτελείται από το περιβάλλον χρόνου εκτέλεσης είναι γραμμένος με τη μορφή bytecode, και ελέγχεται με στόχο τη διασφάλιση της υπακοής στους κανόνες. Οι κλάσεις δεν παραβιάζουν τους ισχύοντες περιορισμούς όταν φορτώνονται στο σύστημα και αυτό διασφαλίζεται από την ίδια τη μονάδα φόρτωσης κλάσεων. Αναφορικά με το API, με την ασφάλεια στην Java οι μίνι-εφαρμογές δεν μπορούν να επέμβουν και να προβούν σε καταστροφικές ενέργειες για το σύστημα.

➤ Ουδέτερη- Αρχιτεκτονική:

Στη δεκαετία του '80 στους ηλεκτρονικούς υπολογιστές υπήρχε μεγάλη ποικιλία, αφού στην αγορά υπήρχαν εταιρείες, όπως η Apple, Commodore, Radio Shack, Atari και IBM. Στην κάθε εταιρία, άρα και στον κάθε υπολογιστή, χρησιμοποιούνταν διαφορετικό λειτουργικό σύστημα. Καθώς η ανάπτυξη λογισμικού ήταν χρονοβόρα, ένα μικρό μέρος από αυτό που αναπτυσσόταν για να χρησιμοποιηθεί από κάποιον Η/Υ έβρισκε εφαρμογή και σε άλλα μηχανήματα. Με τα Windows, τα Macintosh από την Apple και το Unix, το πρόβλημα αυτό λύθηκε εν μέρει. Παρ' όλα αυτά ακόμη και σήμερα δεν είναι εύκολο να αναπτυχθεί ένα λογισμικό, ικανό να χρησιμοποιηθεί ταυτοχρόνως και από τα Windows, Unix και Macintosh. Πρόκειται για πρόβλημα, το οποίο η Java έχει καταφέρει να λύσει. Ο μεταγλωττιστής της Java έχει καταφέρει να δημιουργήσει ένα κώδικα, γραμμένο σε byte, ο οποίος μεταγλωττίζεται από τον αυτόν, προσδίδει μία ουδέτερη αρχιτεκτονική. Συνεπώς, από διαφορετικά μηχανήματα, ανεξαρτήτου λογισμικού, τα προγράμματα της Java είναι δυνατόν να εκτελεστούν.

➤ Μεταφέρεται εύκολα:

Πρωταρχικός στόχος της Java ήταν να δημιουργήσει εφαρμογές φορητές, και όσο νέες αρχιτεκτονικές θα εμφανίζονται εξαιτίας λειτουργικών συστημάτων ή ίδιων μηχανημάτων, η Java και το περιβάλλον της θα μπορούν να λειτουργήσουν εντός αυτών. Τα δεδομένα έχουν τους εξής τύπους, 1) ακέραιοι, 2) κινητής υποδιαστολής, και 3) κινητής υποδιαστολής διπλής ακριβείας, το μέγεθος των οποίων καθορίζεται ανεξάρτητα από το μηχάνημα ή το λειτουργικό σύστημα όπου το πρόγραμμα «τρέχει». Σε αυτό το σημείο εντοπίζεται και μία διαφορά από την γλώσσα C++, της οποίας το μέγεθος των τύπων δεδομένων έχει άμεση εξάρτηση από τον μεταγλωττιστή.

➤ Υψηλή απόδοση:

Οι εφαρμογές που αναπτύσσονται με την Java δεν είναι αποδοτικές στο σημείο που είναι οι εφαρμογές που αναπτύσσονται με την C++. Βέβαια για ένα μεγάλο αριθμό εφαρμογών, οι οποίες διαθέτουν γραφικά που συναντώνται στο διαδίκτυο, η Java φαίνεται να επαρκεί, σε σημείο που κάποιες εφαρμογές δεν διαφέρουν σημαντικά όσον αφορά την απόδοση από εκείνη της C++. Εν κατακλείδι, η Java θεωρείται αρκετά γρήγορη, επιτρέποντας στον προγραμματιστή να προβεί σε ενέργειες, που με την C++ δεν θα μπορούσε.

➤ Πολυνηματική:

Κάθε σύγχρονη εφαρμογή καλείται να εκτελέσει παραπάνω από μία λειτουργίες κάθε φορά. Τα νήματα που επιστρατεύει η Java συμβάλλουν στην ανάπτυξη τέτοιων εφαρμογών. Τα νήματα συνιστούν μία ροή εκτέλεσης μέσα σε μία εφαρμογή, με την εκτέλεση του καθενός να γίνεται ανεξάρτητα σε μία εφαρμογή ή με ταυτόχρονη εκτέλεση από πολλά νήματα. Το να χρησιμοποιήσει ένας προγραμματιστής συγχρονισμένα νήματα, συνιστά χρήσιμο βήμα προς την δημιουργία διαμοιραζόμενων, δικτυακών εφαρμογών. Παραδείγματος χάρι όταν θέλουμε να επικοινωνήσουμε με ένα διακομιστή απομακρυσμένο αυτό επιτυγχάνεται με τη χρήση νήματος, ενώ παράλληλα με τη χρήση ενός άλλου νήματος σημειώνεται αλληλεπίδραση με τον χρήστη.

1.5.2 Εργαλεία της Java και τύποι δεδομένων

Για το γράψιμο μιας εφαρμογής στη γλώσσα Java, απαιτούνται όχι μόνο η ίδια γλώσσα αλλά και διάφορα εργαλεία με στόχο το γράψιμο, τη δοκιμή και την

μεταγλώττιση του προγράμματος. Για να εκτελεστούν τα προγράμματα της Java έχουν αναπτυχθεί και συνεχίζουν να αναπτύσσονται μέχρι σήμερα πολλά περιβάλλοντα (Παπαδοπούλου, 2012). Αυτό το εργαλείο, όμως, που έχει επικρατήσει, είναι της Sun Microsystems, λέγεται JDK, και διατίθεται στην ηλεκτρονική διεύθυνση <http://www.java.cun.com>

➤ Διερμηνευτής χρόνου εκτέλεσης (Interpreter): πρόκειται για εργαλείο που χρησιμοποιείται με στόχο να εκτελεστούν προγράμματα και εφαρμογές της Java. Για να εμφανιστούν οι μινι-εφαρμογές της γλώσσας προγραμματισμού απαιτείται υποστήριξη από ένα Web πρόγραμμα περιήγησης. Ο διερμηνευτής ενδέχεται να είναι κάποιο τμήμα του λογισμικού που λειτουργεί αυτόνομα, ενδέχεται όμως να είναι και μέρος από μία μεγάλη εφαρμογή (π.χ. Netscape). Χάρη σε αυτό ο χρήστης μπορεί να φορτώσει αρχεία κλάσεων, και να ερμηνεύσει δυαδικό κώδικα, αρχεία τα οποία έχουν τη μορφή bytecode. Αρκετές φορές μπορεί να λειτουργήσει και ως μεταγλωττιστής, μεταγλωττίζοντας τον δυαδικό κώδικα Java σε κώδικα μηχανής.

Επιπλέον, είναι δυνατόν να εκτελεστούν όχι μόνο προγράμματα τα οποία υποστηρίζονται με κείμενο, αλλά και εφαρμογές Java, οι οποίες υποστηρίζονται από ποικίλα γραφικά, όχι όμως μινι-εφαρμογές. Ο διερμηνευτής χρόνου εκτέλεσης για να χρησιμοποιηθεί συντάσσεται ως έχει: `java Επιλογές Όνομα_κλάσης Ορίσματα`. Με το όρισμα `Όνομα_κλάσης` προσδιορίζεται το όνομα της κλάσης, την οποία θέλει ο χρήστης να εκτελέσει. Στη περίπτωση που η κλάση βρίσκεται σε ένα πακέτο, τότε είναι υποχρεωτικό να γραφεί ολόκληρο το όνομα της. Παραδείγματος χάρη κατά την εκτέλεση μίας κλάσης με όνομα `Solvent`, όταν βρίσκεται σε ένα πακέτο για παράδειγμα με όνομα `Equations`, στον διερμηνευτή καλείται ως: `java Equations.Solvent`

Κατά την εκτέλεση μιας κλάσης ο διερμηνευτής της Java στην ουσία εκτελεί την μέθοδο `main()` της κλάσης, με τον τερματισμού του διερμηνευτεί να παρατηρείται όταν η μέθοδος `main()` και κάθε άλλου δημιουργημένο νήμα, ολοκληρώνουν την εκτέλεση τους. Θα ήταν σημαντική παράλειψη να μην αναφερθεί πως όταν ο διερμηνευτής χρόνου εκτέλεσης της Java εκτελεί την μέθοδο `main()` κατά την εκτέλεση μιας κλάσης, υποδηλώνει έναν από τους λόγους, όπου δεν είναι εφικτό να «τρέξουν» μίνι εφαρμογές με την χρήση του διερμηνευτή χρόνου εκτέλεσης (Παπαδοπούλου, 2012).

➤ Μεταγλωττιστής: με τον μεταγλωττιστή της Java (`javac`) μεταγλωττίζονται αρχεία του πηγαίου κώδικα σε εκτελέσιμες κλάσεις, λαμβάνοντας τη μορφή bytecode. Όσον αφορά τα αρχεία πηγαίου κώδικα, πρόκειται για αρχεία που έχουν επέκταση `java`

και είναι αρχεία ASCII, και τα οποία μοιάζουν με αρχεία πηγαίου κώδικα που έχουν άλλες γλώσσες προγραμματισμού. Όσον αφορά τα εκτελέσιμα αρχεία, πρόκειται για εκείνα που έχουν επέκταση .class και τα οποία αντιπροσωπεύουν και μία κλάση της Java στην χρησιμοποιήσιμη μορφή της.

Τα bytecode της Java συνιστούν ειδικές εντολές μηχανής, των οποίων αποδέκτες δεν αποτελούν αποκλειστικά συγκεκριμένοι επεξεργαστές ή συστήματα υπολογιστών. Ο συγκεκριμένος ως προς το σύστημα διερμηνευτής ονομάζεται εικονική μηχανή της java, και εκτελεί τις όποιες οδηγίες των bytecode. Από τον μεταγλωττιστή επιτρέπεται η ύπαρξη μίας μόνο public κλάσης, με την κοινή ονομασία του αρχείου και της κλάσης να είναι υποχρεωτική. Η εντολή συντάσσεται ως εξής: javac Επιλογές Όνομα Αρχείου, όπου Όνομα Αρχείου, το όνομα του αρχείου με τον πηγαίο κώδικα προς μεταγλώττιση. Ο μεταγλωττιστής είναι υπόχρεος για την παραγωγή αρχείων κλάσεων σε μορφή bytecode κάθε όποια κλάση ορίζεται εντός του αρχείου αυτού. Στη περίπτωση που μία κλάση A μεταγλωττίζεται, και παράγεται από μία κλάση B, η οποία όμως δεν έχει μεταγλωττιστεί ακόμη, ο μεταγλωττιστής εντοπίζει αυτή την εξάρτηση, μεταγλωττίζοντας και τις δύο κλάσεις. Για παράδειγμα, όταν εκτελείται η εντολή javac BigBird, δημιουργείται μία κλάση, έχοντας όνομα BigBird.class, και της οποίας η αποθήκευση γίνεται σε ένα κοινό κατάλογο με το πηγαίο αρχείο.

Επιπλέον, οι επιλογές που αφορούν τον τρόπο δημιουργία των εκτελέσιμων αρχείων κλάσεων της Java από τον μεταγλωττιστή, καθορίζονται από το όρισμα Επιλογές του μεταγλωττιστή. Ενώ ο αρχικός κατάλογος, όπου αποθηκεύονται οι μεταγλωττισμένες κλάσεις καθορίζεται από την επιλογή -d. Με την επιλογή αυτή δημιουργείται η δομή καταλόγων κάτω από τον κατάλογο, κάτι ιδιαίτερα σημαντικό, αφού δεν είναι λίγες οι φορές, όπου οι κλάσεις οργανώνονται σε ιεραρχική δομή καταλόγων. Ο μεταγλωττιστής αναγκάζεται να παράγει πίνακες αποσφαλμάτωσης για τις κλάσεις της Java με την επιλογή -g. Οι πίνακες αυτοί προορίζονται να χρησιμοποιηθούν από τον αποσφαλματωτή της java, εμπεριέχοντας πληροφορίες (τοπικές μεταβλητές, αριθμούς γραμμών). Για να μπορέσει η μεταγλωττιστής να εκτελέσει τις παραπάνω εντολές, θα πρέπει έστω οι μεταγλωττισμένες εκδοχές των κλάσεων να είναι στη διάθεση του. Δεν απαιτείται η ύπαρξη ενός κώδικα για κάθε αντικείμενο, ενώ οι κλάσεις διαθέτουν κάθε πληροφορία η οποία διατίθεται στα αρχεία του κώδικα.

➤ Αρχεία JAR (Java Archive Files): τα αρχεία JAR μπορούν να χρησιμοποιηθούν για την συμπίεση και την τοποθέτηση ενός υπέρογκου αριθμού αρχείων, με σκοπό να φορτωθούν όλα μαζί σε ένα μίνι πακέτο και, έτσι η διαδικασία φόρτωσης να

επιταχύνεται. Για τη δημιουργία ενός αρχείου JAR χρησιμοποιείται το αρχείο jar.exe. Εάν παραδείγματος χάρη έχουμε μία μίνι-εφαρμογή canvasser.java, στην οποία δημιουργούνται δύο αρχεία κλάσης κατά τη μεταγλώττιση canvasser.class και BoxCanvas.class, τότε είναι δυνατόν να συμπιεστούν (Zip) και τα δύο αρχεία, με την τοποθέτηση τους σε ένα αρχείο JAR, για την από κοινού φόρτωση. Για να γίνει αυτό θα πρέπει να χρησιμοποιηθεί το Jar.exe και να δημιουργηθεί το canvasser.jar έτσι: C:\java-1\canvasser>jar cvf canvasser.jar*. Το τελευταίο σύμβολο που χρησιμοποιήθηκε «*» δηλώνει τη συμπίεση όλων των αρχείων που εμπεριέχονται στον κατάλογο μέσα στο αρχείο JAR. Το «cvf» δηλώνει τις επιλογές που μπορεί ο χρήστης να χρησιμοποιήσει με το αρχείο jar.exe.

Επιλογές	Λειτουργίες
c	Δημιουργία μίας νέας ή άδεια αρχειοθέτησης πάνω στη βασική έξοδο
t	Παράθεση του πίνακα περιεχομένων από τη βασική έξοδο
x (αρχείο)	Εξαγωγή όλων των αρχείων ή των αρχείων που έχουν όνομα από τη βασική έξοδο. Στη περίπτωση ένα αρχείο αφαιρείται, εξάγονται όλα τα αρχεία, σε διαφορετική περίπτωση εξάγεται μόνο το ή τα αρχεία που έχουν καθοριστεί.
f [jar -αρχείο]	Το δεύτερο όρισμα, καθορίζει προς επεξεργασία ένα αρχείο. Κατά τη δημιουργία το συγκεκριμένο όρισμα αναφέρεται στο όνομα του αρχείου JAR που επρόκειτο να δημιουργηθεί (στη θέση της βασικής εξόδου). Στη περίπτωση του πίνακα, το δεύτερο όρισμα προσδιορίζει το αρχείο JAR, που επρόκειτο να παρατεθεί σε κάποια λίστα ή να εξαχθεί.
v	Δημιουργία μιας εξόδου (φορτωμένη αρκετά) στο stderr.

Εικόνα 6 Επιλογές με το Jar

➤ Μονοπάτι κλάσης: προκειμένου η Java να είναι σε θέση να χρησιμοποιήσει μία κλάση, θα πρέπει να μπορεί να βρει τη κλάση αυτή στο σύστημα αρχείων. Ειδάλλως, θα εμφανίζεται μήνυμα πως η κλάση δεν υπάρχει ή δεν είναι διαθέσιμη. Για την εύρεση των κλάσεων μπορεί να χρησιμοποιηθεί είτε το όνομα του ίδιου του πακέτου είτε των καταλόγων, οι οποίοι αναφέρονται στην μεταβλητή CLASSPATH. Η ονομασία των πακέτων αντιστοιχεί στην ονομασία των καταλόγων του συστήματος αρχείων και άρα η κλάση στη περίπτωση της java.applet.Applet θα βρίσκεται στον κατάλογο java. Για τον κάθε κατάλογο η java αναζητά με την σειρά, μέσα στους καταλόγους της όσο

αρχεία αναφέρονται στην μεταβλητή CLASSPATH. Στην περίπτωση που η java αναζητά μία κλάση, η οποία έχει αναφερθεί στον πηγαίο κώδικα του χρήστη, αναζητά εκείνο το όνομα του πακέτου και της κλάσης σε κάθε της κατάλογο, επιστρέφοντας λάθος στη περίπτωση που αυτό το αρχείο της κλάσης δεν εντοπιστεί.

➤ Βοήθημα εμφάνισης μίνι- εφαρμογών (Applet Viewer): για να «τρέξει» μία μίνι-εφαρμογή χωρίς να χρησιμοποιηθεί ένα Web πρόγραμμα περιήγησης οι δημιουργοί της Java επιστρατεύουν το εργαλείο αυτό. Χάρη σε αυτό εξυπηρετούνται καταστάσεις, όπως θα συνέβαινε με κάποιον άλλον στοιχειώδη μηχανισμό ελέγχου των μίνι-εφαρμογών. Στην ουσία το βοήθημα αυτό ισοδυναμεί με ένα Web πρόγραμμα περιήγησης, ωστόσο διαφέρει ως προς τη λειτουργία του. Αρχικά. Λειτουργεί πάνω σε έγγραφα της HTML, ψάχνοντας, όμως, για ετικέτες <Applet> οι οποίες έχουν ενσωματωθεί. Έπειτα, αγνοεί τον υπόλοιπο κώδικα που υπάρχει στο έγγραφο. Όταν το Applet Viewer συναντά μία ετικέτα <Applet> μέσα σε κάποιο έγγραφο HTML, τότε προχωρά στο άνοιγμα ενός ξεχωριστού παραθύρου, στο οποίο εμπεριέχεται η αντίστοιχη μίνι-εφαρμογή.

Το βοήθημα αυτό είναι ιδιαίτερα χρήσιμο, με μοναδικό του ελάττωμα ότι δεν παρουσιάζει τον τρόπο όπου η μίνι- εφαρμογή θα «τρέξει» όταν θα βρεθεί περιορισμένη στα όρια μιας πραγματικής ιστοσελίδας. Το βοήθημα αυτό συντάσσεται ως: appletviewer Επιλογές URL. Αναφορικά με το όρισμα URL χρησιμοποιείται για τον καθορισμό της URL του εγγράφου, που περιέχει μία HTML σελίδα, σε μία μίνι-εφαρμογή της Java που είναι ενσωματωμένη.

➤ Αποσφαλματωτής (jdb): με το εργαλείο αυτό τα προγράμματα, που είναι γραμμένα με την java μπορούν να αποσφαλματωθούν. Μεταξύ των λειτουργιών αποσφαλμάτωσης συγκαταλέγονται ο ορισμός των σημείων διακοπής και η βηματική εκτέλεση του κώδικα. Όταν γίνεται λόγος για τον ορισμό των σημείων διακοπής εννοείται ο καθορισμός της γραμμής του κώδικα, όπου το πρόγραμμα σταματά να εκτελείται. Όταν γίνεται λόγος στην βηματική εκτέλεση του κώδικα, σημαίνει ότι ο κώδικας εκτελείται γραμμή προς γραμμή. Πριν τη χρήση του αποσφαλματωτή ο κώδικας επιβάλλεται να μεταγλωττιστεί. Το εργαλείο αυτό συντάσσεται ως: jdb Επιλογές <Όνομα Κλάσης>, το όρισμα Όνομα Κλάση δεν είναι υποχρεωτικό, καθορίζει, όμως, το όνομα της κλάσης η οποία επρόκειτο να εκτελεστεί.

➤ Γεννήτρια τεκμηρίωσης (JavaDoc): πρόκειται για εργαλείο, το οποίο χρησιμοποιείται προκειμένου να παραχθεί απευθείας τεκμηρίωση από τον πηγαίο κώδικα. Μέσω αυτής τα πηγαία αρχεία αναλύονται και παράγονται HTML σελίδες,

στηριζόμενη τόσο στις δηλώσεις όσο και στα σχόλια. Η σύνταξη του εργαλείου αυτού είναι: Javadoc Επιλογές Όνομα Αρχείου, όπου το Όνομα Αρχείου καθορίζει άλλοτε το πακέτο και άλλοτε το αρχείο πηγαίου κώδικα της Java.

Ακόμη, το JavaDoc με στόχο να προσθέσει επιπρόσθετες πληροφορίες στα HTML έγγραφα που παράγονται υποστηρίζει ειδικές ετικέτες. Κάθε ετικέτα ξεκινά με το σύμβολο “ @ ”, η εμφάνιση του οποίου είναι απαραίτητη στην αρχή κάθε γραμμής. Στη συνέχεια καταγράφεται ένα παράδειγμα πηγαίου κώδικα, το οποίο χρησιμοποιεί ετικέτες τεκμηρίωσης κλάσεων:

```
π.χ.:  /**
        * @see Object
        * @see gemology.Rock
        * @version 2.0 Dec 5, 1996
        * @author Brett Weir
```

*/

➤ **Diassembler** αρχείων κλάσεων: πρόκειται για εργαλείο, το οποίο χρησιμοποιείται για να «αποσυναρμολογηθεί» ένα αρχείο κλάσης. Έτσι, το αρχείο κλάσης που εκτελείται αναλύεται σε μία λίστα από δημόσια δεδομένα, μεθόδους ή οδηγίες σε μορφή bytecode. Η χρησιμότητα του diassembler αρχείων κλάσεων γίνεται αντιληπτή όταν για μία κλάση απουσιάζει ο πηγαίος κώδικας, και παρά το γεγονός αυτό ο χρήστης θέλει να μάθει σχετικά με τον τρόπο που υλοποιήθηκε (Παπαδοπούλου, 2012).

Τα δεδομένα στην Java έχουν σχεδόν τους ίδιους τύπους με της C και της C++. Ωστόσο, υπάρχουν κάποιες διαφοροποιήσεις, με την πιο σημαντική ότι η Java δεν διαθέτει pointers. Στη συνέχεια (Πίνακας 5.) αναφέρονται οι τύποι δεδομένων που χρησιμοποιεί η γλώσσα Java.

Όνομα	Μέγεθος (bytes)	Όρια
byte	1 (8bits)	-128 μέχρι 127
short	2 (16bits)	-32768 μέχρι 32767

int	4 (32bits)	-2147473648 μέχρι -2147473647
long	8 (64bits)	-9223372036854775808 μέχρι -9223372036854775807
float	4 (32bits)	$1.4 * 10^{-45}$ μέχρι $3.4 * 10^{38}$
double	8 (64bits)	$4.9 * 10^{-324}$ μέχρι $1.8 * 10^{308}$
bool	1	true/false
char	2	-
string	μεταβάλλεται	-

Εικόνα 7 Τύποι δεδομένων στη Java

Τα byte, short, int και long χρησιμοποιούνται για τους ακέραιους αριθμούς, ενώ τα float και double για τους δεκαδικούς αριθμούς. Τα δεδομένα με τον τύπο bool παίρνουν δύο τιμές (false και true), ενώ για τα δεδομένα τύπου char το μέγεθος ισοδυναμεί με 2 bytes, καθώς ο σχεδιασμός του περιλαμβάνει χαρακτήρες Unicode (UTF-16) (Σφέτσος, & Στάμελος, 2003).

Κεφάλαιο 2^ο

2.1 Ηλεκτρονικά παιχνίδια - Ιστορία

Τα παιχνίδια μέσω του ηλεκτρονικού υπολογιστή μπορεί να είναι τρισδιάστατα ή και δισδιάστατα ενώ μπορεί να απαιτούν τη χρήση του διαδικτύου. Οι παίχτες μπορούν παράλληλα να αλληλοεπιδρούν με άλλους ή να παίζουν εναντίον του υπολογιστή. Παράλληλα, Τα διαδικτυακά παιχνίδια, λειτουργούν ως ένας μηχανισμός σύνδεσης παικτών.

Τα ηλεκτρονικά παιχνίδια, αποτελούν ουσιαστικά την πρώτη διαφοροποιημένη έκδοση παιχνιδιού, παρατηρώντας την έκλειψη των παραδοσιακών. Τη δεκαετία του 1960 κάνουν την εμφάνισή τους στις βιομηχανικές χώρες, ενώ διαδίδονται στα τέλη της δεκαετίας του '70 αρχικά στα καταστήματα ηλεκτρονικών παιχνιδιών ενώ ακολουθούν

και στις οικίες (Τάσση, 2006). Η γρήγορη εξάπλωσή τους οφείλεται κατά ένα μεγάλο μέρος στην ραγδαία εξέλιξη της τεχνολογίας.

Οι ηλεκτρονικοί υπολογιστές προσφέρουν νέους τρόπους παρουσίασης υλικού βελτιώνοντας τα επίπεδα ψηφιακής αντίληψης. Το 1950 δημιουργείται από φοιτητές του MIT το πρώτο ψηφιακό παιχνίδι με το όνομα Spacewar. Απαιτούνταν δύο χρήστες οι οποίοι μέσω του ίδιου υπολογιστή χειρίζονταν ένα διαστημόπλοιο. Στη συνέχεια, η DEC ενσωμάτωσε το παιχνίδι στους υπολογιστές της και έτσι ο κάθε αγοραστής το είχε στη διάθεσή του προεγκατεστημένο (ΠΙΚ, 2020) Με σκοπό μία άκρως ρεαλιστική εμπειρία για το χρήστη, το 1970 κάνει την εμφάνισή του το παιχνίδι Pong. Το Pong κυκλοφόρησε τον Ιούνιο του 1970 από την εταιρεία Atari Inc και αποτέλεσε το πρώτο εμπορικά επιτυχημένο βιντεοπαιχνίδι που χρησιμοποιούσε δισδιάστατη γραφική αναπαράσταση για την προσομοίωση ενός παιχνιδιού τένις. Οι παίκτες χειρίζονταν τις κινήσεις τους μέσω κουμπιών με στόχο την κατεύθυνση μίας ασπρόμαυρης μπάλας.

Το Pong ο Atari PONG κυκλοφόρησε τον Ιούνιο του 1972 και είναι το πρώτο εμπορικά επιτυχημένο βιντεοπαιχνίδι και βασίζεται σε μια απλή δισδιάστατη γραφική αναπαράσταση ενός παιχνιδιού που μοιάζει με τένις. Οι παίκτες χρησιμοποιούν κουπιά για να χτυπήσουν μια μπάλα μπρος-πίσω σε ασπρόμαυρη οθόνη.

Το Pong ελεγχόταν εξ ολοκλήρου από υπολογιστή φέρνοντας έτσι μία νέα εποχή, την εποχή των Computer Control Games δηλαδή τα Ελεγχόμενα παιχνίδια από υπολογιστή. Επιπλέον δεν υπάρχουν κινούμενα μέρη πέρα από το μοχλό (joystick) που δύναται να χρησιμοποιήσει ο χρήστης. Το παιχνίδι, σύντομα διαδίδεται στην παγκόσμια αγορά με την κοινή αποδοχή του καθώς αντικατόπτριζε την ευγενή άμιλλα ενός αθλήματος, του τένις (Χρήστου, 2007).

Τα πρώτα εμπορικά ψηφιακά παιχνίδια που ακολούθησαν, ήταν γνωστά σαν μηχανικά και ηλεκτρομηχανικά παιχνίδια σε ηλεκτρονικό μέσο. Έτσι περιορίζονταν τυχόν εννοιολογικά εμπόδια σχετικά με τα ηλεκτρονικά παιχνίδια για τους καταναλωτές. Οι προγραμματιστές παιχνιδιών, ωστόσο, εργάζονταν πέρα από τα όρια της τότε τεχνολογίας για τη δημιουργία νέων τύπων παιχνιδιών και εμπειριών. Ωστόσο, μόνο τα πιο επιτυχημένα παιχνίδια μπόρεσαν να ενσωματώσουν και τις δύο προσεγγίσεις συνδυάζοντας το παραδοσιακό με τα νέα μέσα.

Σήμερα, το Διαδίκτυο είναι ίσως η πιο σημαντική πηγή για την αναδρομή της ιστορίας των ψηφιακών παιχνιδιών και όχι μόνο. Το 1981 κυκλοφορούν πλέον στην αγορά

πεντακόσιοι τίτλοι παιχνιδιών με πάνω από 25000 αντίτυπα για τα πιο επιτυχημένα. Λόγω της μεγάλης επιτυχίας του παιχνιδιού που κυκλοφόρησε από την Atari, ακολουθούν κι άλλες εταιρίες την κατασκευή παιχνιδιών για ηλεκτρονικούς υπολογιστές όπως, η Mattel το 1980 παρουσίασε το IntelliVision ενώ το 1982 εμφανίζεται το ColecoVision.

Στον παρακάτω πίνακα βλέπουμε τις βιβλιογραφικές αναφορές στα ψηφιακά παιχνίδια από το 2002 έως το 2015 (DiGRA, 2020).

Έτος	Θεματολογία	Συνολικές Αναφορές	Ιστορικό
2002	Computer games and digital cultures conference proceedings	27	1
2003	Level up conference proceedings	73	6
2005	Changing views: Worlds in play	142	8
2007	Situated play	113	11
2009	Breaking new ground: Innovation in games, play, practice and theory	139	11
2010	Proceedings of DiGRA Nordic 2010: Experiencing	13	0

	games: Games, play, and players		
2011	Think design play: The fifth international conference of the Digital Research Association (DiGRA)	91	9
2012	Proceedings of 2012 DiGRA Nordic	23	4
2013	DiGRA 2013— DeFragging game studies	72	13
2014	DiGRA 2014	16	1
2015	DiGRA 2015	38	1

Εικόνα 8 DiGRA αναφορές

Από το 2003 και μετά παρατηρείται μία ιδιαίτερη αύξηση των επιστημονικών αναφορών, δεδομένου πως από το 2005 κάνει την εμφάνισή του ο επεξεργαστής φυσικής PhysX όπου οι φυσικές ιδιότητες των αντικειμένων προσομοιώνονται προσφέροντας στο χρήστη νέα επίπεδα ρεαλισμού.

2.2 Αλλαγές στην κατασκευή παιχνιδιών για ηλεκτρονικούς υπολογιστές

Οι αλλαγές που αφορούν τη δημιουργία ψηφιακών παιχνιδιών τα τελευταία πενήντα χρόνια παγκοσμίως, μπορούν να ταξινομηθούν σε διάφορα στάδια (Crawford, 2003):

Αλλαγές στο υλικό: Οι αλλαγές ως προς το υλικό (hardware) επέδρασαν ιδιαίτερα στα παιχνίδια μέσω ηλεκτρονικού υπολογιστή, επιτρέποντας στους προγραμματιστές να τα

εξελίξουν. Τέτοιες αλλαγές ήταν η αύξηση της υπολογιστικής ισχύς, της μνήμης RAM, της χωρητικότητας των αποθηκευτικών μέσων, η αναβάθμιση των καρτών γραφικών και η ύπαρξη σύνδεσης στο διαδίκτυο με αυξημένες ταχύτητες.

Αλλαγές στις συσκευές αλληλεπίδρασης: Με την πάροδο των ετών υπήρξαν αλλαγές στις συσκευές αλληλεπίδρασης για τον χειρισμό των παιχνιδιών. Οι πρώτες κονσόλες παιχνιδιών είχαν ειδικά περιστρεφόμενα κουμπιά ή απλά χειριστήρια. Σήμερα υπάρχουν χειριστήρια αφής, προσομοιώσεις χειριστηρίων (όπλα, ρακέτες) αλλά και χειριστήρια με αισθητήρες κίνησης όπως οι ελεγκτές Wii του Nintendo ή το σύστημα Kinect της Microsoft.

Αλλαγές στα εργαλεία λογισμικού: Σήμερα σε αντίθεση με το παρελθόν υπάρχουν εργαλεία για τη συγγραφή κώδικα καθώς επίσης και αυτοματοποιημένες εφαρμογές δημιουργίας παιχνιδιών που δεν απαιτούν τη συγγραφή του, όπως η Unity.

Αλλαγές στα δημογραφικά στοιχεία παιχτών: Ενώ αρχικά τα πρώτα παιχνίδια παίζονταν από άντρες σήμερα ο αριθμός των παιχτών δεν περιορίζεται ως προς το φύλλο ούτε την ηλικία, γεγονός που οδήγησε στη δημιουργία νέων θεματικών παιχνιδιών.

2.3 Διάκριση ηλεκτρονικών παιχνιδιών

Τα ηλεκτρονικά παιχνίδια αποτελούν ένα κοινωνικό φαινόμενο το οποίο έχει πολλές προεκτάσεις. Οι εξελίξεις των τελευταίων χρόνων στις ψηφιακές τεχνολογίες κατέστησαν εφικτή τη δημιουργία ποικίλων ειδών παιχνιδιών με αποτέλεσμα την κατηγοριοποίησή του (Greenfield, 1984):

Παιχνίδια περιπέτειας (adventure games): στα παιχνίδια αυτά, ο παίχτης είναι και ο πρωταγωνιστής του παιχνιδιού όπου απαιτείται να λύσει γρίφους ή να εντοπίσει αντικείμενα ώστε να συνεχίσει την πορεία του. Στο αρχικό τους στάδιο, τα παιχνίδια περιπέτειας ενσωμάτωναν κείμενο ενώ στην πορεία προστέθηκαν οπτικά εφέ. Η παραγωγή τους χρονολογείται στο 1970 με πρώτο το παιχνίδι Colossar Cane Adventure.

Εκπαιδευτικά παιχνίδια (educational games): Τα παιχνίδια αυτά κατασκευάστηκαν προκειμένου να αυξήσουν τις γνωστικές δεξιότητες των παιχτών για κάποιο θεματικό

αντικείμενο. Δημοφιλή παιχνίδια αυτής της κατηγορίας, αποτελούν το Carmen Sandiego, το Oregon Trail και το Mavis Beacon Teaches Typing.

Παιχνίδια μάχης (fighting games): Τα παιχνίδια αυτά επικεντρώνονται στις πολεμικές τέχνες και στις μάχες μεταξύ δύο ή περισσότερων χαρακτήρων και παικτών. Από τα πιο γνωστά παιχνίδια της κατηγορίας είναι το Mortal Kombat.

Παιχνίδια στόχου (First person shooter): Τα παιχνίδια αυτά οδηγούν το πρωταγωνιστή και παίχτη στην καταπολέμηση ενός εχθρού μέσω όπλου. Συνήθως ο πρωταγωνιστής έχει την αίσθηση ότι βρίσκεται στο χώρο. Τέτοιο παιχνίδι αποτελεί το Doom.

Αγωνιστικά παιχνίδια (racing games): Η κατηγορία αυτή είναι μία από τις παραδοσιακότερες όπου ο παίκτης καλείται να ανταγωνιστεί άλλους οδηγούς εξομοιώνοντας το εκάστοτε όχημα (αεροπλάνο, λεωφορείο, ράλι). Η κατασκευή αυτή χρονολογείται στο 1980 με πρώτο το Gran Trak της Atari. Σήμερα υπάρχει μία πληθώρα αγωνιστικών παιχνιδιών, ανάμεσά τους το Gran Turismo και το Need for Speed.

Παιχνίδια γρίφων (puzzle games): Τα παιχνίδια αυτά, απαιτούν την επίλυση διαφόρων γρίφων και την πλοήγηση των παιχτών σε πολύπλοκες περιοχές (λαβύρινθος). Στην εποχή μας, συχνά ενσωματώνονται σε εκπαιδευτικά παιχνίδια και παιχνίδια περιπέτειας.

Παιχνίδια στρατηγικής (strategy games): τα παιχνίδια αυτά στοχεύουν στον σχεδιασμό και στην επιδέξια διαχείριση πόρων ώστε να επιτευχθεί η νίκη. Αποτελούνται από δύο κατηγορίες, τα turn - based και τα real time. Κάποια από τα δημοφιλέστερα παιχνίδια είναι το Warcraft, το Star Craft, το SidMeier's Civilization και το Heroes of Might.

Αθλητικά παιχνίδια (sports games): τα παιχνίδια προσομοιώνουν τα παραδοσιακά αθλήματα όπως το ποδόσφαιρο, το μπάσκετ, το γκολφ, το χόκεϋ, και το τένις. Πολλά από αυτά εστιάζουν στον ρεαλιστικό τρόπο παιχνιδιού του εκάστοτε αθλήματος ενώ άλλα δίνουν έμφαση στη στρατηγική του. Τα πιο γνωστά παιχνίδια της κατηγορίας αυτής είναι το Championship Manager, το FIFA, το Pro evolution soccer και το Arch Rivals.

Παραδοσιακά παιχνίδια (traditional games): Από τις πιο δημοφιλείς κατηγορίες επιτραπέζιων παιχνιδιών αποτελεί η κατηγορία των παραδοσιακών. Τα παιχνίδια αυτά βοηθούν στο να βελτιωθούν οι ικανότητες του ατόμου καθώς απαιτούν οξυδέρκεια και στρατηγική. Τέτοιου είδους παιχνίδια είναι το Σκάκι, η Τρίλιζα, και το Τάβλι.

2.4 Πλεονεκτήματα ψηφιακών παιχνιδιών

Όσον αφορά τα παιδιά, τα ψηφιακά παιχνίδια είναι μπορούν να βοηθήσουν ακόμη και την υγεία. Γενικότερα τα παιχνίδια απασχολούν εξίσου τα δυο φύλλα, καθώς σύμφωνα με έρευνες το 97% των παιδιών ηλικίας 10 έως 19 ετών παίζουν παιχνίδια υπολογιστών (Parisod et al., 2014).

Οι απαιτήσεις της εποχής καθώς και η εξέλιξη του διαδικτύου έχουν δημιουργήσει νέες ανάγκες τόσο για τους ενήλικες όσο και για τα παιδιά. Οι άνθρωποι βρίσκουν ελκυστικό αλλά και διασκεδαστικό το να παίζουν ψηφιακά παιχνίδια καθώς αυτά είναι εντυπωσιακά και τους οδηγούν σε φανταστικούς κόσμους με αληθοφανές εικόνες, εντυπωσιακά γραφικά και ήχο. Ακόμη τα άτομα μπορούν να αναπτύξουν δεξιότητες βελτιώνοντας την πνευματική τους ικανότητα τις γνώσεις τους, τη συγκέντρωση προσοχής, τη δεξιότητα, την ικανότητα επίλυσης προβλημάτων, το συντονισμό και να εξοικειωθούν με την τεχνολογία.

Επιπλέον, αποτελούν δυναμικά περιβάλλοντα μάθησης, μέσω της δημιουργίας εικονικών κόσμων για τις δράσεις παιχτών και χρηστών, ενώ μπορεί να βοηθήσει στην ανάπτυξη ευεργετικών κοινωνικών πρακτικών, δυναμικών ταυτοτήτων και όμοιων αξιών και τρόπων σκέψης. Τα παιχνίδια μέσω ηλεκτρονικού υπολογιστή μπορούν να χαρακτηριστούν δυναμικά περιβάλλοντα μάθησης επειδή καθιστούν εφικτή τη δημιουργία εικονικών κόσμων και την απόκτηση εμπειριών με το φυσικό κόσμο. Τέλος μπορούν να βοηθήσουν στην απαλλαγή από το άγχος και στην εκτόνωση του στρες επιτρέποντας τον πειραματισμό και τη βίωση διαφόρων συναισθημάτων (Ανθούλιας, 1985).

Κεφάλαιο 3^ο

3.1 Tic Tac Toe Ιστορία

Το Tic-Tac-Toe είναι ένα από τα παραδοσιακά παιχνίδια που παιζόταν με χαρτί και μολύβι. Αυτό το παιχνίδι απαιτεί ένα ταμπλώ σε πλέγμα 3x3 με τον Παίκτη 1 να λειτουργεί ως "Ο" και ο Παίκτης 2 ενεργεί ως "Χ" ή το αντίστροφο. Ο στόχος αυτού του παιχνιδιού είναι να επιτευχθεί μία συνεχής σειρά σύνδεσης με οριζόντιο, κατακόρυφο ή διαγώνιο τρόπο από τον ίδιο παίκτη.

Αυτό το παιχνίδι παρουσιάστηκε για πρώτη φορά στην αρχαία εποχή, ωστόσο δεν υπάρχουν στοιχεία που να το κατατάσσουν χρονολογικά. Πολλοί πιστεύουν ότι το παιχνίδι εφευρέθηκε στην Αρχαία Αίγυπτο και στη συνέχεια η Ρωμαϊκή Αυτοκρατορία το ονόμασε «Terni Lapili», καθώς το σχέδιο πλέγματος για το παιχνίδι βρέθηκε χαραγμένο με κιμωλία σε όλα τα ερείπια της αρχαίας πόλης.

Το παιχνίδι έκτοτε εμφανίστηκε ξανά στην Αγγλία με το όνομα "Naught and Crosses" το 1864. Αυτή η εκ νέου εμφάνιση αποτελεί τη μοντέρνα εκδοχή του παιχνιδιού Tic-Tac-Toe που γνωρίζουν οι άνθρωποι μέχρι σήμερα (Wolf, 2012).

Το 1952, ο Alexander S. Douglas στο Πανεπιστήμιο του Cambridge ανέπτυξε ένα ψηφιακό παιχνίδι Tic-Tac-Toe που ονομάστηκε «OXO» για τον υπολογιστή EDSAC. Αυτό ήταν το πρώτο βιντεοπαιχνίδι του Tic-Tac-Toe και ενσωμάτωνε AI, όπου ο άνθρωπος θα μπορούσε να παίζει με αντίπαλο του υπολογιστή.

3.1.2 Παίζοντας Tic Tac Toe στο διαδίκτυο

3.2 Στρατηγική

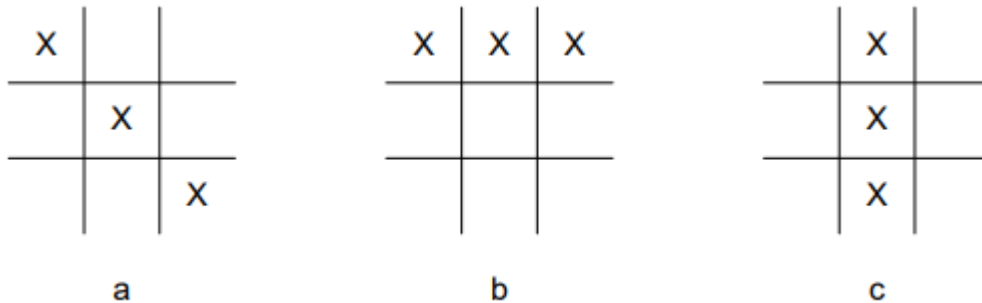
Μπορούν να χρησιμοποιηθούν πολλές στρατηγικές για το παιχνίδι. Το βασικό σημείο της στρατηγικής είναι ότι οι παίκτες πρέπει να μπλοκάρουν το αντίπαλο πιόνι, οριζόντια, κάθετα ή διαγώνια, ώστε να αποτρέψουν την επίτευξη μίας σειράς από τρία πιόνια επιτυγχάνοντας παράλληλα τη δική τους τρίλιζα.

Συνδυαστικά και αν υποθέσουμε ότι το «πιόνι» Χ κινείται πρώτα:

- Υπάρχουν 91 διαφορετικές θέσεις που μπορεί να κερδίσει το Χ.
- 44 διαφορετικές θέσεις που μπορεί να κερδίσει το Ο.

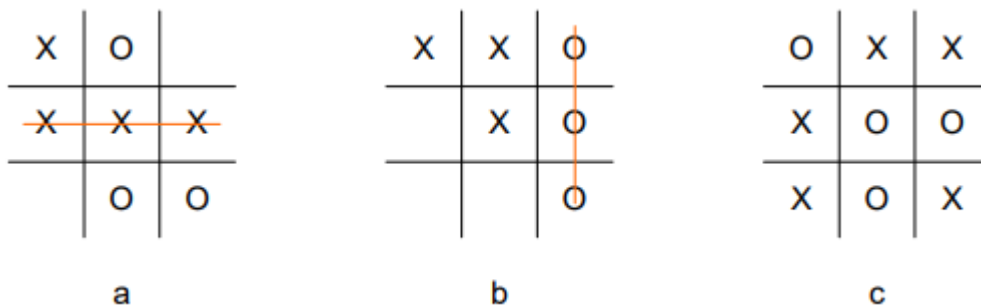
- 3 διαφορετικές θέσεις για ισοπαλία.

Μελετώντας το πλέγμα (grid) 3 X 3 που αποτελείται από εννέα θέσεις διαδοχικά μπορούν να υπάρχουν οι ακόλουθες συνθήκες όπως φαίνονται παρακάτω:



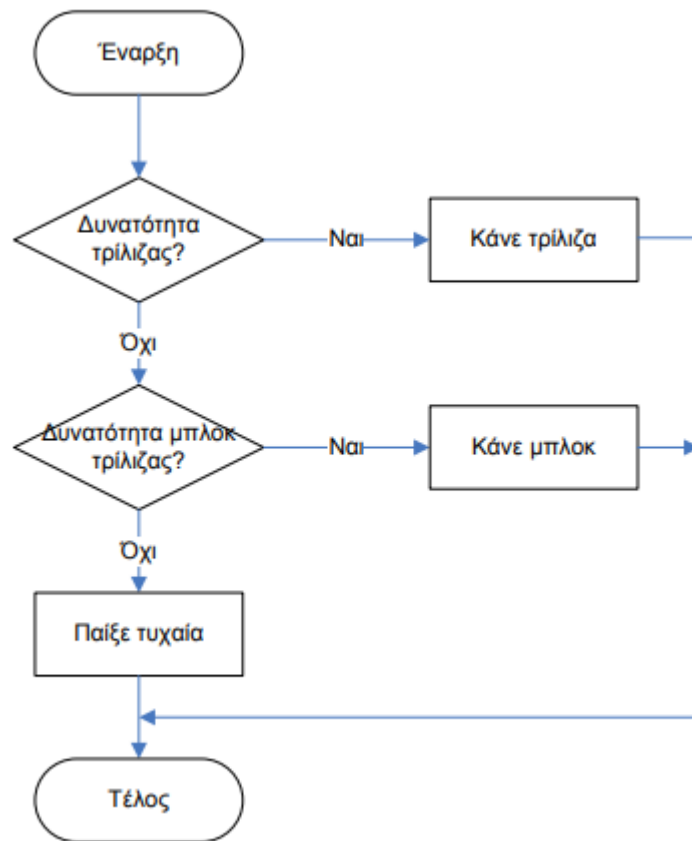
Εικόνα 9 Θέσεις νίκης

Επιπλέον στην πιο κάτω εικόνα βλέπουμε παραδείγματα ήττας.



Εικόνα 10 Θέσεις ήττας

Καθώς το παιχνίδι στα ελληνικά μεταφράζεται ως τρίλιζα, ο υπολογιστής εκτελεί βασικές κωδικοποιημένες κινήσεις από το πρόγραμμα για να αντιμετωπίσει τον παίχτη – χρήστη. Το πρόγραμμα, θέτει ως προτεραιότητα τη νίκη, ακολούθως την αποφυγή ήττας ενώ τελευταία ακολουθεί την τυχαία κίνηση. Το λογικό διάγραμμα του κώδικα αναπαριστά αυτές ακριβώς τις προτεραιότητες κατά σειρά.

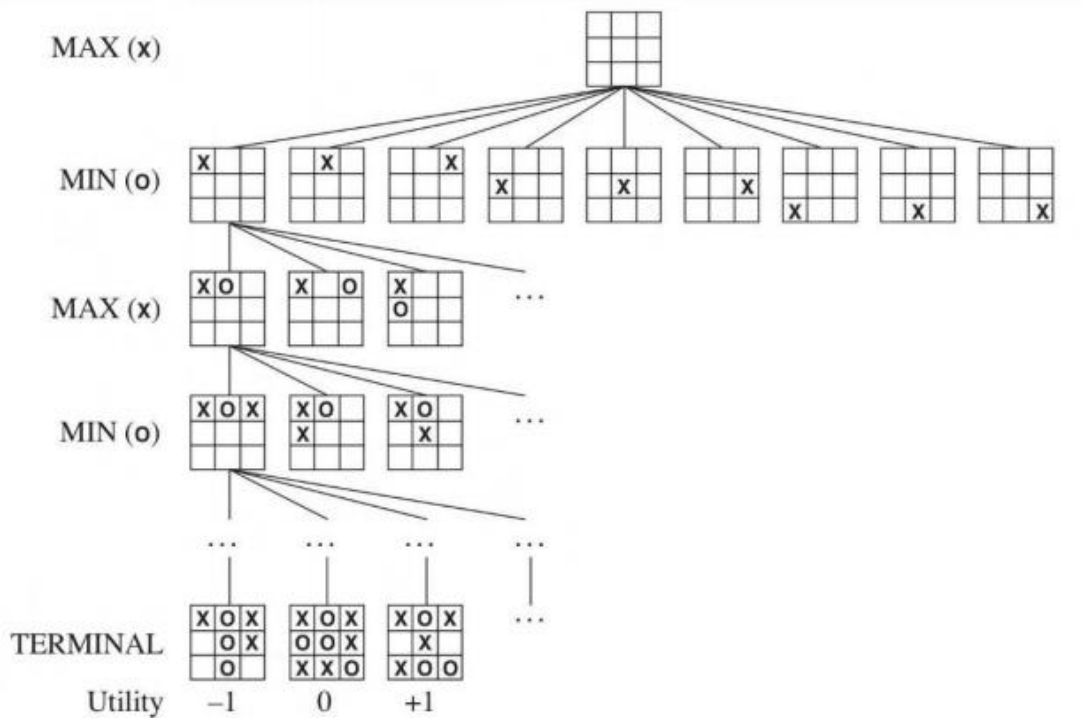


Εικόνα 11 Λογικό διάγραμμα αναπαράστασης του κώδικα

Διάγραμμα υπορουτίνας

3.3 Τεχνητή νοημοσύνη

Η τεχνητή νοημοσύνη μπορεί να βρει εφαρμογή στο παιχνίδι αυτό στη λειτουργία ανθρώπου εναντίον υπολογιστή. Υπάρχουν πολλά δείγματα αυτού του παιχνιδιού με τεχνητή νοημοσύνη στο Διαδίκτυο και κάθε παράδειγμα έχει τον δικό του αλγόριθμο ο οποίος εξαρτάται από τον προγραμματιστή. Ο δημοφιλέστερος αλγόριθμος που μπορεί να χρησιμοποιηθεί για το Tic-Tac-Toe είναι ο αλγόριθμος Minimax. Στην εικόνα βλέπουμε τα τμήματα του δέντρου αναζήτησης για το παιχνίδι Tic Tac Toe.



Εικόνα 12 Τμήματα του δέντρου αναζήτησης για το παιχνίδι Tic Tac Toe

Πρωτού εξηγήσουμε τη λειτουργία του αλγόριθμου Minimax ας δούμε ένα παράδειγμα του παιχνιδιού Tic-Tac-Toe. Το πρόγραμμα είναι γραμμένο σε Java και εμπεριέχει τον αλγόριθμο Minimax τεχνητής νοημοσύνης. Επίσης αποτελείται από δύο στρατηγικές που βασίζονται στην έκδοση παιχνιδιού των Newell και Simon, το 1972:

- Κλείσιμο του κέντρου
- Αποκλεισμός του αντιπάλου

Το κύριο μέρος αυτού του αλγορίθμου βρίσκεται στην μεταβλητή με το όνομα μοτίβο (pattern). Αυτή η μεταβλητή ελέγχει τον αλγόριθμο. Σαν αρχική τιμή, η τιμή αυτής της μεταβλητής είναι:

```
int pattern [ ] [ ] = { {10, 1, 2, 3, 11},
                        {10, 1, 4, 7, 11},
                        {10, 1, 5, 9, 11},
                        {10, 2, 5, 8, 11},
                        {10, 3, 5, 7, 11},
```

{10, 3, 6, 9, 11},

{10, 4, 5, 6, 11},

{10, 7, 8, 9, 11}};

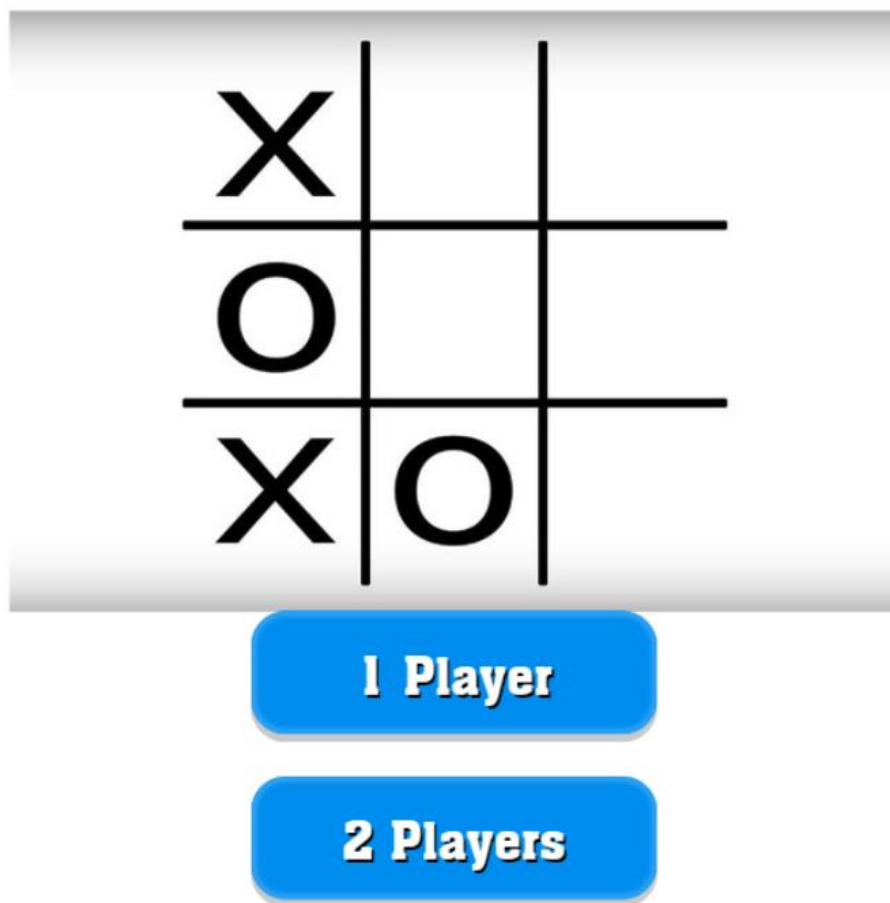
Αυτή η μεταβλητή αποτελείται από:

- 1) Flag (10, ..., ..., 11)
- 2) Προϋποθέσεις για να κερδίσει το X:
 - a. 1,2,3
 - b. 1,4,7
 - c. 1,5,9
 - d. 2,5,8
 - e. 3,5,7
 - f. 3,6,9
 - g. 4,5,6
 - h. 7,8,9

Ο αλγόριθμος minimax σε μία τρέχουσα κατάσταση παιχνιδιού, υπολογίζει την απόφαση minimax, πραγματοποιώντας αναζήτηση σε βάθος στο δέντρο του παιχνιδιού. Η πολυπλοκότητά του ως προς το χρόνο είναι $O(b^m)$. Η χωρική του πολυπλοκότητα είναι $O(bm)$. Το b είναι ο μέσος παράγοντας διακλάδωσης.

3.4 Παρουσίαση εφαρμογής

Στην αρχική οθόνη παρατηρούμε την οθόνη καλωσορίσματος του παιχνιδιού. Παρατηρούμε πως ο χρήστης μπορεί να επιλέξει παιχνίδι με αντίπαλο τον υπολογιστή επιλέγοντας το κουμπί 1 player και παιχνίδι με άλλον αντίπαλο μέσω του κουμπιού 2 players. Για την τελευταία επιλογή απαιτείται ο δεύτερος χρήστης να ενεργεί όταν έρθει η σειρά του.



Εικόνα 13 Αρχική οθόνη εφαρμογής

Επιπλέον, στο πάνω δεξιό τμήμα του παράθυρο, υπάρχει το κουμπί ήχου. Στην περίπτωση που ο χρήστης επιθυμεί να τον απενεργοποιήσει, αρκεί να πατήσει πάνω του.

Αφού ο χρήστης επιλέξει τον αντίπαλο του, στη συνέχεια καλείται να διαλέξει και το μέγεθος του ταμπλό του παιχνιδιού. Υπάρχουν τρεις επιλογές:

- 3 X 3 δηλαδή εννέα συνολικά τετράγωνα,

- 5 X 5 δηλαδή εικοσιπέντε συνολικά τετράγωνα
- 7 X 7 δηλαδή 49 τετράγωνα στο σύνολο

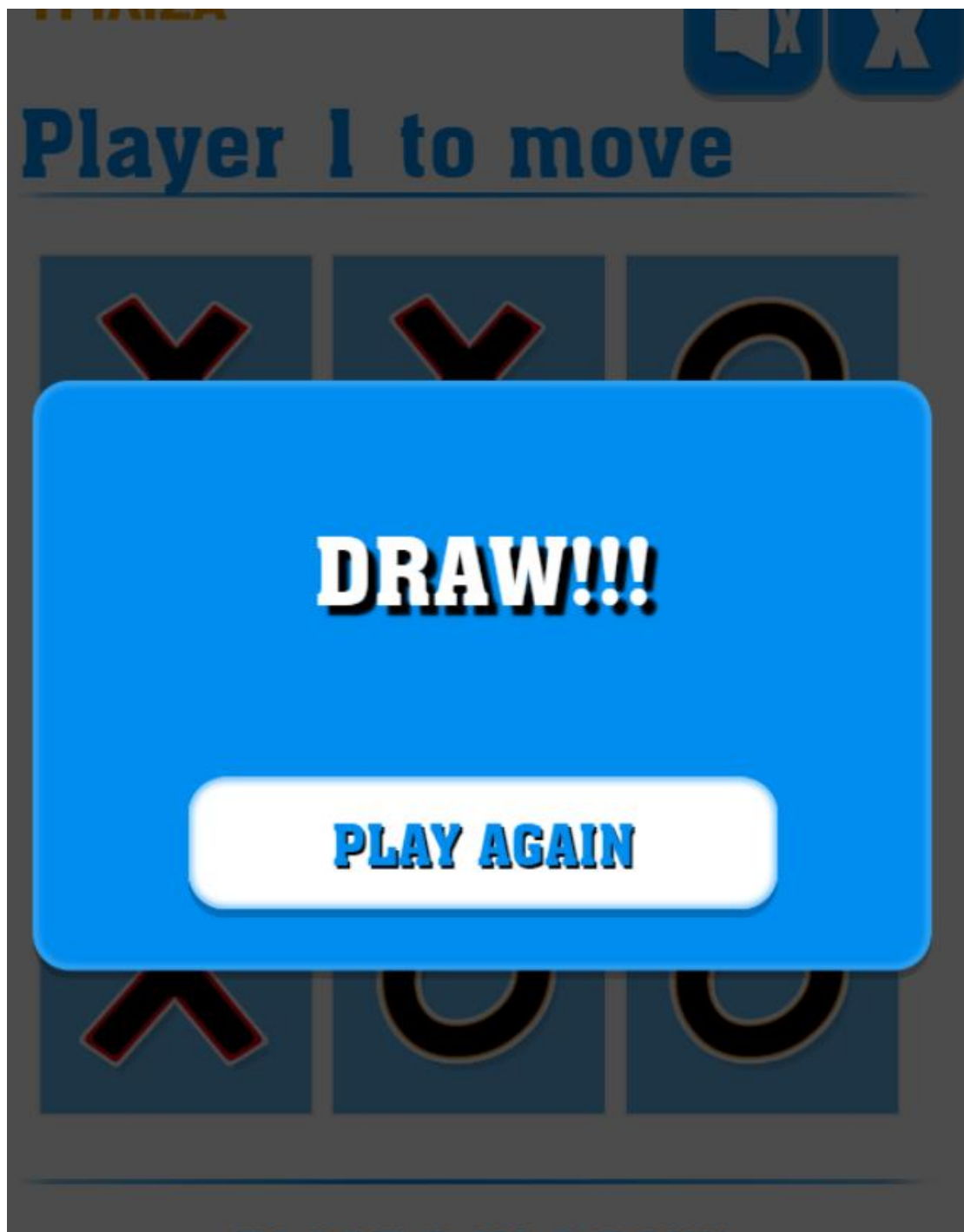
Ο αριθμός των πλαισίων αντικατοπτρίζει το επίπεδο δυσκολίας θα μπορούσαμε να ισχυριστούμε σε εύκολο μέτριο και δύσκολο αντίστοιχα.

ΕΠΙΛΟΓΗ ΕΠΙΠΕΔΟΥ



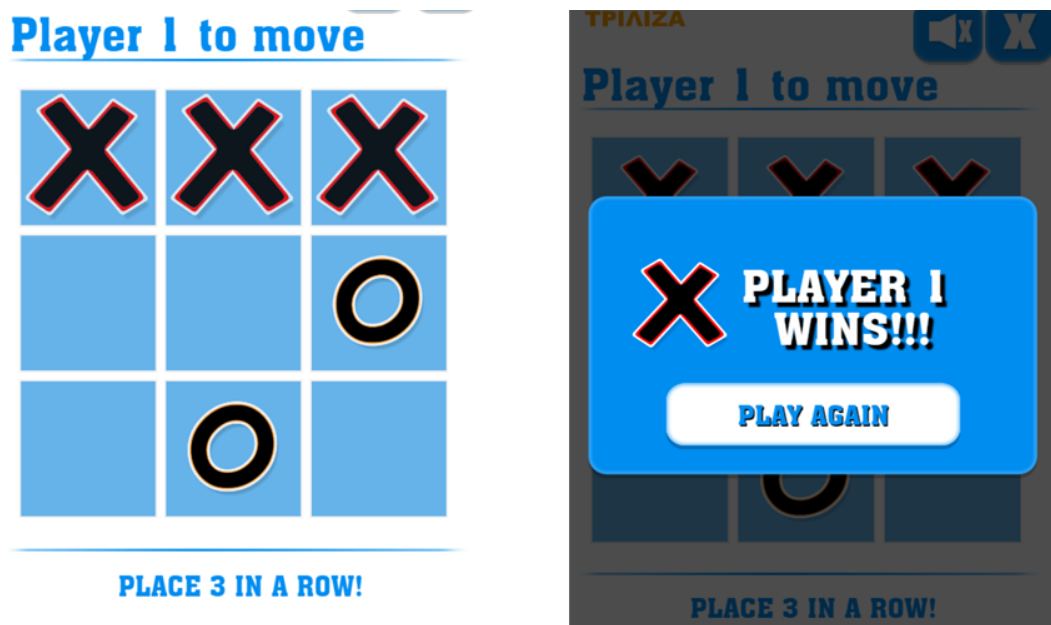
Εικόνα 14 Επιλογή επιπέδου παιχνιδιού

Στο επίπεδο 3 X 3 ο κάθε παίχτης πρέπει να επιτύχει την τοποθέτηση του πιονιού του σε τρία συνεχόμενα κάθετα, οριζόντια ή διαγώνια. Σε κάθε περίπτωση ο πρώτος παίχτης έχει το πιόνι X ενώ κάθε φορά που έρχεται η σειρά του άλλου, εμφανίζεται το αντίστοιχο μήνυμα. Στην περίπτωση της ισοπαλίας, στην οθόνη θα εμφανιστεί το αντίστοιχο μήνυμα με το κουμπί για επανάληψη παιχνιδιού (play again).



Εικόνα 15 Περίπτωση ισοπαλίας

Στην περίπτωση που κάποιος παίχτης κερδίσει, τότε εμφανίζεται το αντίστοιχο μήνυμα. Ο στόχος για την επίτευξη της νίκης στο επίπεδο 5 X 5 και 7 X 7 είναι να τοποθετηθούν τέσσερα πούλια κατά σειρά και όχι τρία όπως στο επίπεδο 3 X 3.



Εικόνα 16 Περίπτωση νίκης

3.4.1 Ανάλυση απαιτήσεων

- Να είναι εύχρηστο, με φιλικό περιβάλλον και κατάλληλα γραφικά στοιχεία
- Να μπορεί να παρέχει στο χρήστη πρόσβαση από διαφορετικές συσκευές
- Η εφαρμογή να χρησιμοποιεί το ποντίκι.
- Το παιχνίδι να υποστηρίζει πάνελ, κουμπιά, μενού και γραφικά.
- Το παιχνίδι πρέπει να έχει ένα κύριο μενού με τις εξής επιλογές: ◦ επιλογή παίχτη (επιπέδου) ◦ επιλογή ήχου ◦ επιλογή επιπέδου

- Το παιχνίδι φορτώνεται με τις ρυθμίσεις που επέλεξε ο χρήστης στο κύριο μενού.
- Το παιχνίδι υποστηρίζει διάφορους τύπους αρχείων ήχου.
- Υπάρχει κουμπί επανάληψης παιχνιδιού
- Οποιαδήποτε στιγμή ο χρήστης μπορεί να τερματίσει την εφαρμογή με το κουμπί

Μη λειτουργικές απαιτήσεις

Επίδοση

- Το σύστημα θα πρέπει να είναι διαθέσιμο 24 ώρες τη μέρα 365 μέρες το χρόνο
- Το σύστημα θα πρέπει να είναι ευέλικτο και να επιτρέπει τη διόρθωση σφαλμάτων
- Επίσης θα πρέπει να είναι κατασκευασμένο έτσι ώστε να μπορούν να πραγματοποιούνται μελλοντικές βελτιώσεις
- Σωστά καταρτισμένη πολιτική ασφαλείας και γνωστοποιημένη σε όλους τους
- Εφαρμογή μεθόδων ασφαλείας σε κάθε επίπεδο του συστήματος

Απόδοση

Ως προς την απόδοση του, όταν αναφερόμαστε σε ένα παιχνίδι αυτό δεν θα πρέπει να (Γιακουμάκης, 1994) είναι μονοδιάστατο αλλά να διαθέτει τόσο δικτυακό όσο και αποθηκευτικό εξοπλισμό, καθώς επίσης συστήματα και εφαρμογές που καθιστούν ικανοποιητική τη συνολική του απόδοση.

Ταυτόχρονα, το παιχνίδι θα πρέπει να προσφέρει στο χρήστη ένα ιδιαίτερα αποδοτικό περιβάλλον αξιοποίησής του, χωρίς να απαιτούνται πρόσθετα βήματα με συνέπεια καθυστερήσεις.

Ευχρηστία

- Το σύστημα θα πρέπει να είναι εύκολο στη χρήση του καθώς απευθύνεται σε μία κλίμακα χρηστών.

Ευελιξία – Επεκτασιμότητα

Για τη διάρκεια ζωής του παιχνιδιού αλλά και κάθε εφαρμογής εμπλέκονται πολλοί και σημαντικοί παράγοντες. Σε αυτούς, μπορεί να οφείλεται και η επιτυχία ή η αποτυχία του. Το παιχνίδι, όπως και κάθε παιχνίδι, θα πρέπει να προσαρμόζεται στις ανάγκες των επιμέρους χρηστών και στις τυχόν μεταβολές που θα προκύψουν σε αυτούς. Ακόμα, πολύ σημαντικό ζήτημα, αποτελεί και η επεκτασιμότητά του, ώστε να μπορεί να ανταπεξέλθει σε νέες ανάγκες και δεδομένα που θα προκύψουν μελλοντικά. Οι δύο αυτοί παράγοντες, η ευελιξία και η επεκτασιμότητα, είναι στενά συνδεδεμένοι, καθώς μία ορθή πρόβλεψη σε συνδυασμό με τη δυνατότητα παραμετροποίησης θα μειώσει πιθανές ανάγκες επέκτασης της εφαρμογής. Θα πρέπει να ληφθεί υπ' όψιν, πως η πρόβλεψη του συνόλου μεταβολών δεν είναι εφικτή για ένα παιχνίδι όπως επίσης και για κάποιο άλλο σύστημα. Για το λόγο αυτό, θα πρέπει να είναι έτσι δομημένο, ώστε να επιτρέπεται η μελλοντική του επέκταση.

Η σχεδίαση με παραμέτρους καθώς και η αξιοποίηση του συνόλου των στοιχείων τα οποία δεν συνιστούν φυσικές σταθερές της λειτουργικής περιοχής, συγκαταλέγονται στον κύριο παράγοντα για μεγιστοποίηση της προσαρμοστικότητας. Είναι δεδομένη η ύπαρξη δυνατότητας αλλαγής των παραμέτρων τόσο από τους διαχειριστές όσο και από τους χρήστες, σύμφωνα με το εκάστοτε επίπεδο εξουσιοδότησης. Σχετικά με την προσαρμοστικότητα και την επεκτασιμότητα, το παιχνίδι θα πρέπει να καλύπτει τις παρακάτω αξιώσεις:

- Ανάπτυξη η οποία βασίζεται σε συνιστώσες (components).
- Κάθε συνιστώσα, θα πρέπει να περιέχει βασικές πληροφορίες χρήσης.

Ακεραιότητα

Μιλώντας για ακεραιότητα, μας απασχολεί ουσιαστικά η ασφάλεια. Η τελευταία, αποτελεί έναν ιδιαίτερα σημαντικό παράγοντα για κάθε είδους λογισμικό μεταξύ των οποίων και τα παιχνίδια, τα οποία υπάρχουν στο διαδίκτυο και δεν είναι ανεκτικά σε σφάλματα. Για να υπάρχει αποτελεσματική διαχείριση της ασφάλειας, είναι απαραίτητη η εγκατάσταση πυλών ασφαλείας (Proxy Servers, Firewalls) καθώς και λύσεις για ανίχνευση τυχών διαδικτυακών επιθέσεων. Θα πρέπει να υιοθετούνται, σε κάθε περίπτωση, τεχνολογίες και επιχειρηματικές ευκαιρίες που επιτρέπουν μεταβολές στις δικτυακές υποδομές και στη διαχείριση του τμήματος της ασφάλειας, η οποία θα πρέπει να βελτιώνεται συνεχώς.

3.5 Ανάλυση κώδικα

Τα αρχεία της CSS για το οπτικό περιβάλλον του παιχνιδιού υπάρχουν στον ομώνυμο φάκελο CSS. Εκεί υπάρχει και η γραμματοσειρά του παιχνιδιού Aachen η οποία επιλέχθηκε επειδή «διαβάζεται» εύκολα από όλους τους browsers.

Οι ήχοι περιέχονται στο φάκελο Sounds και γίνεται κλήση τους στον κώδικα. Ο φάκελος Sprites περιέχει τα εικονοστοιχεία του παιχνιδιού.

Από τα αρχεία Java καθορίζονται οι λειτουργίες του παιχνιδιού. Για παράδειγμα ο παρακάτω κώδικας τοποθετεί το εικονίδιο του ηχείου στην οθόνη ενώ στο ίδιο αρχείο CMenu.js καθορίζεται η ενεργοποίηση και απενεργοποίηση του ήχου.

```
_pStartPosBut1 = {x:(CANVAS_WIDTH/2),y:CANVAS_HEIGHT -340};  
  
var oSprite = s_oSpriteLibrary.getSprite('but_box_1');  
  
_oBut1Player = new  
CTextButton(_pStartPosBut1.x,_pStartPosBut1.y,oSprite,TEXT_BUT1PLAYE  
R,FONT_GAME,'#ffffff',60,s_oStage);  
  
_oBut1Player.addEventListener(ON_MOUSE_UP, this._onBut1Player,  
this);
```

Παρακάτω, μέσω του αρχείου CMenu.js καθορίζεται η επιλογή των παικτών

```
this._onBut1Player = function(){  
    this.unload();  
  
    $(s_oMain).trigger("start_session");  
  
    s_oMain.gotoGameModeMenu(false);  
  
};  
  
this._onBut2Players = function(){
```

```
this.unload();  
$(s_oMain).trigger("start_session");  
  
s_oMain.gotoGameModeMenu(true);
```

Η εφαρμογή μέσω κώδικα javascript δύναται να προσαρμόζει την οθόνη ανάλογα με το browser και τη συσκευή.

```
function isIOS() {  
  
  var iDevices = [  
  
    'iPad Simulator',  
  
    'iPhone Simulator',  
  
    'iPod Simulator',  
  
    'iPad',  
  
    'iPhone',  
  
    'iPod'  
  
  ];  
  
  while (iDevices.length) {  
  
    if (navigator.platform === iDevices.pop()){  
  
      s_bIsIphone = true;  
  
      return true;  
  
    }  
  
  }  
  
  s_bIsIphone = false;  
  
  return false;
```

```
}
```

```
window.addEventListener("orientationchange", onOrientationChange );
```

```
function onOrientationChange(){
```

```
    if (window.matchMedia("(orientation: portrait)").matches) {
```

```
        //Προβολή πορτραίτου
```

```
        sizeHandler();
```

```
    }
```

```
    if (window.matchMedia("(orientation: landscape)").matches) {
```

```
        // Προβολή LANDSCAPE
```

```
        sizeHandler();
```

Άλλη μία παράθεση ενός αποσπάσματος του κώδικα αφορά την εμφάνιση μηνυμάτων κατά την έναρξη και τη λήξη του παιχνιδιού, το μήνυμα σε περίπτωση ισοπαλίας και τον παίχτη που έχει σειρά.

```
TEXT_GAMEOVER = "GAME OVER";
```

```
TEXT_PLAY     = "PLAY";
```

```
TEXT_PLAYAGAIN = "PLAY AGAIN";
```

```
TEXT_PL1_VICTORY = "PLAYER 1\n WINS!!!!";
```

```
TEXT_PL2_VICTORY = "PLAYER 2\n WINS!!!!";
```

```
TEXT_DRAW = "DRAW!!!!";
```

TEXT_BUT1PLAYER = "1 Player";

TEXT_BUT2PLAYERS = "2 Players";

TEXT_ACTIVEPLAYER = {pt1: "Player ", pt2: " to move"};

TEXT_AIACTIVEPLAYER = {pt1: "AI ", pt2: " to move"};

TEXT_THREE_IN_ROW = "PLACE 3 IN A ROW!";

TEXT_FOUR_IN_ROW = "PLACE 4 IN A ROW!";

Συμπεράσματα

Συνολικά το παιχνίδι ικανοποιεί τους στόχους που τέθηκαν: πρόκειται για μια απλή, εύχρηστη εφαρμογή για το παιχνίδι της τρίλιζας. Από άποψη λειτουργικότητας καλύπτει επαρκώς τα δύο βασικότερα σενάρια χρήσης (single player και multiplayer), με χαρακτηριστικό την επιλογή επιπέδου δυσκολίας για τον εκάστοτε παίχτη. Βέβαια θα μπορούσαν να υπάρξουν προτάσεις βελτίωσης σε κάποιους τομείς όπως για παράδειγμα την απομνημόνευση των νικών της ισοπαλίας.

Βιβλιογραφία

- [1] Tim Berners-Lee, (1991). Αρχαιοθετημένη επικοινωνία στην οποία γίνεται η αναφορά στο πρώτο κείμενο που περιγράφει την γλώσσα HTML. World Wide Web Consortium, 29 Οκτωβρίου, 1991, Πηγή από το Διαδίκτυο. Διαθέσιμο στο: <http://lists.w3.org/Archives/Public/www-talk/1991SepOct/0003.html>.
- [2] Chris Crawford. «On Game Design», (2003) New Riders Publishing, ISBN 0-13-146099-4.
- [3] Greenfield, P. (1984). Mind and media: the effects of television, video games and computers, Harvard University Press
- [4] H Parisod, M Aromaa, L Kauhanen, K Kimppa, C Laaksonen, V Leppänen (2014) «The advantages and limitations of digital games in children’s health promotion» Finnish Journal of eHealth and eWelfare 6 (4), 164-173
- [5] Γιακουμάκης Εμ. (1994), Τεχνολογία Λογισμικού, εκδ. Σταμούλη, Αθήνα
- [6] M. J. P. Wolf, (2012) Encyclopedia of Video Games: The Culture, Technology, and Art of Gaming, Greenwood Publishing Group
- [7] Vannevar, B., (1945). *As We May Think*. *The Atlantic Monthly*, July 1945, Πηγή από το Διαδίκτυο. Διαθέσιμο στο <http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/>
- [8] Ανθουλιάς, Τ. (1985). Παιδιά και computers. Αθήνα. Gutenberg
- [9] Βόγκλης, Α., & Καραβασίλης, Β., (2014). *Εισαγωγή στο Διαδικτυακό Προγραμματισμό (HTML, CSS, PHP)*. Ιδιωτική έκδοση. Πηγή από το

Διαδίκτυο. Διαθέσιμο στο: <https://www.ebooks4greeks.gr/eisagwgh-sto-diadiktyako-programmatismo-html-css-php>

- [10] Παιδαγωγικό Ινστιτούτο Κύπρου, (2020), «Ιστορία των Ψηφιακών παιχνιδιών», ανακτήθηκε από <https://internetsafety.pi.ac.cy/parents-games-history>
- [11] Τάσση , Μ. (2006). Το παιδί και το παιχνίδι. Χίος
- [12] DiGRA (2020), History in DiGRA Papers ανακτήθηκε από DiGRA Digital Library <http://www.digra.org/digital-library/>
- [13] Χρήστου, Ι. (2007). Παιδί και Ηλεκτρονικό Παιχνίδι. Αθήνα. Ταξιδευτής