



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ελληνική Μυθολογία – Διαδικτυακό Παιχνίδι



Του φοιτητή
Δάτση Αλέξανδρου
Αρ. Μητρώου: 154444

Επιβλέπων
Όνοματεπώνυμο: Αδαμίδης
Παναγιώτης
Βαθμίδα: Καθηγητής

Ημερομηνία Ιούνιος 2021

Τίτλος Π.Ε.: Ελληνική Μυθολογία - Διαδικτυακό Παιχνίδι

Κωδικός Π.Ε. 20145

Όνοματεπώνυμο φοιτητή/τών: Δάλτσης Αλέξανδρος

Όνοματεπώνυμο εισηγητή Αδαμίδης Παναγιώτης

Ημερομηνία ανάληψης Π.Ε. 11-09-2020

Ημερομηνία περάτωσης Π.Ε. 03-04-2021

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Αλέξανδρου Δάλτση που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Η πτυχιακή αυτή εργασία εκπονήθηκε στα πλαίσια των σπουδών μου στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος (ΔΙΠΙΑΕ). Στόχος της είναι η δημιουργία ενός διαδικτυακού παιχνιδιού, τύπου επιτραπέζιου, με θέμα την ελληνική μυθολογία. Μπορεί κανείς να παίξει το συγκεκριμένο παιχνίδι είτε με αντίπαλο τον υπολογιστή, είτε με φίλους που εκείνος θα επιλέξει. Βασικό κίνητρο στην επιλογή της πτυχιακής μου, ήταν η αγάπη και η χρόνια ενασχόλησή μου με τα video games. Η δημιουργία ενός παιχνιδιού μου κίνησε το ενδιαφέρον, καθώς πάντοτε ήθελα να γνωρίσω τις διαδικασίες που απαιτούνται για την ανάπτυξη ενός παιχνιδιού. Παράλληλα, ο συνδυασμός με την ελληνική μυθολογία μου έδωσε ένα επιπλέον έναυσμα να ασχοληθώ, καθώς μέσα από διάφορων δυσκολιών ερωτήσεις στο παιχνίδι, ο παίκτης θα παίρνει πληθώρα γνώσεων αλλά και θα οξύνει το νου. Με αυτή την πτυχιακή εργασία μου δόθηκε η ευκαιρία να διευρύνω τις γνώσεις μου σε μια νέα πλατφόρμα ανάπτυξης λογισμικού που αναπτύσσεται τα τελευταία χρόνια, τη λεγόμενη Firebase, να εφαρμόσω όλες τις γνώσεις που έλαβα από την σπουδαστική μου πορεία και τελικά να παίξω ένα παιχνίδι που ο ίδιος δημιούργησα.

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι να αναπτυχθεί ένα παιχνίδι με θέμα την ελληνική μυθολογία, το οποίο έχει τη μορφή των κλασικών επιτραπέζιων παιχνιδιών. Το παιχνίδι χρησιμοποιεί την Firebase ως πλατφόρμα ανάπτυξης για το back end μέρος της εφαρμογής, που είναι μια serverless λύση για εφαρμογές υψηλής απόδοσης. Για το front end επιλέχθηκε η Angular που αποτελεί ίσως το μεγαλύτερο framework για εφαρμογές στο διαδίκτυο. Σε αυτό το παιχνίδι οι παίκτες θα μπορούν να παίξουν είτε μόνοι τους, ανταγωνιζόμενοι με τον υπολογιστή, είτε με άλλους χρήστες, τους οποίους οι ίδιοι θα επιλέγουν. Πυρήνας του παιχνιδιού είναι η απάντηση ερωτήσεων που σχετίζονται με την ελληνική μυθολογία, σε διάφορα επίπεδα δυσκολίας. Με κάθε σωστή απάντησή τους, οι παίκτες θα προχωρούν μπροστά στο διαδικτυακό “ταμπλό” ή θα κερδίζουν πόντους έχοντας τελικό στόχο, μέσα από τη μάθηση, να προσπεράσουν τους αντιπάλους τους. Νικητής ανακηρύσσεται ο παίκτης που θα φτάσει πρώτος στον τερματισμό και θα έχει αρκετούς πόντους(ανάλογα με το πόσους έχουν ορίσει οι ίδιοι οι παίκτες στην αρχή του παιχνιδιού).

Greek Mythology - Online Game

Alexandros Daltsis

Abstract

The purpose of this dissertation is to develop a game based on Greek mythology, which will be focused on the rules of classic board games. Firebase, a serverless solution for high quality applications, was used for the back end infrastructure. On the other hand, Angular, one of the most popular frameworks for developing applications on the web, was used for the front end (presentation layer) of the application. In this game, players are able to play either alone, competing with the computer, or with other users they choose. The core of the gameplay is answering questions related to Greek mythology, at various levels of difficulty. With each correct answer, the players will move forward on the "board" or earn points aiming, through learning, to overtake their opponents and thus be the winner. The first one who reaches the finish line and has enough points, is the winner.

Ευχαριστίες

Με αφορμή την ολοκλήρωση της παρούσας πτυχιακής εργασίας, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου κύριο Αδαμίδα Παναγιώτη για τη συμβολή, το ενδιαφέρον, τις υποδείξεις και τη συνεχή καθοδήγησή του σε όλη τη διάρκεια της εργασίας μου. Επίσης, δεν μπορώ να παραλείψω να ευχαριστήσω πολύ και τα μέλη της εξεταστικής επιτροπής. Τέλος, το μεγαλύτερο ευχαριστώ οφείλω στους γονείς μου που καθ'όλη τη διάρκεια των σπουδών μου ήταν αρωγοί και υποστηρικτές μου.

Περιεχόμενα

Πρόλογος.....	iii
Περίληψη.....	iv
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα	vii
Κατάλογος Εικόνων	x
Κατάλογος Πινάκων.....	xi
Συνομογραφίες.....	xii
Κεφάλαιο 1ο: Εισαγωγή.....	1
Κεφάλαιο 2ο: Ανάπτυξη παιχνιδιών.....	2
2.1 Εισαγωγή.....	2
2.2 Ορισμός παιχνιδιού	2
2.2.1 Ηλεκτρονικά παιχνίδια	2
2.3 Ιστορική αναδρομή ηλεκτρονικών παιχνιδιών.....	3
2.3.1 Δεκαετία 1940	4
2.3.2 Δεκαετία 1950	5
2.3.3 Δεκαετία 1960	6
2.3.4 Δεκαετία 1970	6
2.3.5 Δεκαετία 1980	8
2.3.6 Δεκαετία 1990	8
2.3.7 Δεκαετία 2000	10
2.3.8 Δεκαετία 2010	12
2.4 Διαδικασία ανάπτυξης ηλεκτρονικών παιχνιδιών και ρόλοι.....	13
2.4.1 Στοιχεία που δομούν ένα ηλεκτρονικό παιχνίδι.....	15
2.4.2 Ρόλοι στη διαδικασία ανάπτυξης παιχνιδιών	15
2.4.3 Στάδια ανάπτυξης ενός ηλεκτρονικού παιχνιδιού	16
2.5 Μεταφορές επιτραπέζιων παιχνιδιών σε παιχνίδια υπολογιστών	17
2.5.1 Κατηγορίες επιτραπέζιων παιχνιδιών.....	18
2.5.2 Η μετάβαση στα ηλεκτρονικά επιτραπέζια παιχνίδια	19
2.5.3 Εκπαιδευτικά παιχνίδια	20
2.5.4 Το σκάκι	20
2.5.5 Το πόκερ.....	22

2.5.6	Το τάβλι.....	22
2.6	Multiplayer Computer Games	23
2.6.1	Game Networking	24
Κεφάλαιο 3ο:	Παιχνίδια με θέμα την ελληνική μυθολογία.....	27
3.1	Ηλεκτρονικά παιχνίδια βασισμένα στην ελληνική μυθολογία.....	29
3.2	Παιχνίδια περιηγητή ιστού βασισμένα στην ελληνική μυθολογία.....	32
Κεφάλαιο 4ο:	Ανάλυση και επιλογή τεχνολογιών	34
4.1	Τεχνολογίες Front End	34
4.1.1	HTML.....	34
4.1.2	CSS.....	37
4.1.3	JavaScript	38
4.1.4	Front End Frameworks	41
4.2	Τεχνολογίες Back End.....	44
4.2.1	PHP.....	45
4.2.2	ASP.NET	46
4.2.3	JAVA.....	46
4.2.4	Node.js.....	47
4.2.5	Back End Frameworks	47
4.3	“Πρόσθετα”	49
4.3.1	Adobe Flash Player	49
4.3.2	Java.....	50
4.3.3	Unity Web Player	50
4.4	Τεχνολογίες που επιλέχθηκαν για το παιχνίδι.....	51
4.4.1	Angular.....	51
4.4.2	Firebase	54
4.4.3	Γιατί επιλέχθηκαν αυτές οι τεχνολογίες.....	58
4.5	Πώς χρησιμοποιήθηκαν οι τεχνολογίες.....	58
Κεφάλαιο 5ο:	Περιγραφή παιχνιδιού, γραφικά και περιπτώσεις χρήσης.....	63
5.1	Περιβάλλον εφαρμογής.....	63
5.1.1	Αρχική σελίδα	64
5.1.2	Παιχνίδι με Η/Υ.....	65
5.1.3	Δωμάτια.....	65
5.1.4	Προφίλ.....	66
5.1.5	Σελίδα φίλων	66
5.2	Εκτέλεση παιχνιδιού.....	67

5.2.1	Παιχνίδι με Η/Υ ή ατομικό παιχνίδι.....	68
5.2.2	Ομαδικό παιχνίδι.....	69
5.3	Γραφικά.....	69
5.4	Περιπτώσεις χρήσης.....	69
Κεφάλαιο 6ο:	Επίλογος και Συμπεράσματα.....	77
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		79

Κατάλογος Εικόνων

Εικόνα 2.1 Υπολογιστής ENIAC	4
Εικόνα 2.2 Cathode Ray Tube Amusement Device	5
Εικόνα 2.3 Mouse in the Maze.....	6
Εικόνα 2.4 Pong	7
Εικόνα 2.5 Atari 2600	7
Εικόνα 2.6 Final Fantasy II	8
Εικόνα 2.7 Doom	9
Εικόνα 2.8 Warcraft: Orcs & Humans	10
Εικόνα 2.9 Sims	11
Εικόνα 2.10 World of Warcraft.....	12
Εικόνα 2.11 Minecraft.....	13
Εικόνα 2.12 Dungeons & Dragons.....	18
Εικόνα 2.13 Pokemon Card Game	19
Εικόνα 2.14 Magic: The Gathering	20
Εικόνα 2.15 Το κλασικό σκάκι	21
Εικόνα 2.16 Το σκάκι online.....	21
Εικόνα 2.17 Texas Hold'em	22
Εικόνα 2.18 Το τάβλι online	23
Εικόνα 2.19 Client - Server	25
Εικόνα 2.20 Peer to Peer	26
Εικόνα 2.21 Dedicated Server	26
Εικόνα 3.1 Troy Board Game	27
Εικόνα 3.2 Age of Mythology: The Board Game	28
Εικόνα 3.3 Madness and the Minotaur	30
Εικόνα 3.4 God of War	31
Εικόνα 3.5 Ikarium	33
Εικόνα 4.1 Πρόγραμμα Hello World σε HTML	36
Εικόνα 4.2 Ιδιότητες στοιχείου HTML	37
Εικόνα 4.3 Επιλογείς id και κλάση	38
Εικόνα 4.4 JavaScript σε HTML έγγραφο	40
Εικόνα 4.5 Χειρισμός DOM με τη χρήση JavaScript	40
Εικόνα 4.6 Μια κλάση σε React.....	42
Εικόνα 4.7 Ένα τυπικό Component της Angular	43
Εικόνα 4.8 Ένα component σε Vue	44
Εικόνα 4.9 Κώδικας PHP.....	45
Εικόνα 4.10 ngModel Directive	52
Εικόνα 4.11 NgIf και NgFor Directives	52
Εικόνα 4.12 Μεταβλητή περιβάλλοντος	53
Εικόνα 4.13 Assets του παιχνιδιού.....	53
Εικόνα 4.14 Custom Claims.....	55
Εικόνα 4.15 Ένα αντικείμενο στη βάση RealTime	56
Εικόνα 4.16 Cloud Functions	57
Εικόνα 4.17 Δομή βάσης.....	59
Εικόνα 4.18 Οι κλάσεις του παιχνιδιού.....	62

Εικόνα 5.1 Περιβάλλον εφαρμογής	64
Εικόνα 5.2 Παιχνίδι με Η/Υ	65
Εικόνα 5.3 Σελίδα Προφίλ	66
Εικόνα 5.4 Σελίδα Φίλων	67
Εικόνα 5.5 Παράθυρο ερώτησης.....	68

Κατάλογος Πινάκων

Πίνακας 3.1 Επιτραπέζια παιχνίδια ελληνικής μυθολογίας	28
Πίνακας 3.2 Ηλεκτρονικά παιχνίδια ελληνικής μυθολογίας.....	30
Πίνακας 3.3 Browser παιχνίδια βασισμένα στην ελληνική μυθολογία.....	32
Πίνακας 4.1 Pull Requests σε γλώσσες προγραμματισμού (GitHub)	41
Πίνακας 4.2 Δημοφιλή Back End Frameworks.....	48
Πίνακας 5.1 ΠΧ 1 - Δημιουργία λογαριασμού	69
Πίνακας 5.2 ΠΧ 2 - Σύνδεση	70
Πίνακας 5.3 ΠΧ 3 - Παιχνίδι με Η/Υ	70
Πίνακας 5.4 ΠΧ 4 - Δημιουργία Δωματίου	71
Πίνακας 5.5 ΠΧ 5 - Σύνδεση σε δωμάτιο	71
Πίνακας 5.6 ΠΧ 6 - Περιβάλλον δωματίου	72
Πίνακας 5.7 ΠΧ 7 - Ομαδικό παιχνίδι	73
Πίνακας 5.8 ΠΧ 8 - Προφίλ χρήστη	73
Πίνακας 5.9 ΠΧ 9 - Φίλοι	74
Πίνακας 5.10 ΠΧ 10 - Προσκλήσεις.....	75
Πίνακας 5.11 ΠΧ 11 - Αιτήματα φιλίας.....	75
Πίνακας 5.12 ΠΧ 12 - Αποσύνδεση.....	75

Συντομογραφίες

ΔΠΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
H/Y	Ηλεκτρονικός Υπολογιστής
Π.Ε.	Πτυχιακή Εργασία
ΠΧ	Περίπτωση Χρήσης
AJAX	Asynchronous JavaScript And XML
ASP	Active Server Pages
AWS	Amazon Web Services
CERN	European Council for Nuclear Research
CPU	Computer Processing Unit
CRT	Cathode Ray Tube
CSS	Cascading Style Sheets
DOM	Document Object Model
EA	Electronic Arts
EDSAC	Electronic Delay Storage Automatic Calculator
ENIAC	Electronic Numerical Integrator and Computer
FDT	Flash Development Toolkit
FIBS	First Internet Backgammon Server
GPU	Graphics Processing Unit
HD	High Definition
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines
IDE	Integrated Development Environment
JDK	Java Development Kit
JS	Javascript
JSON	Javascript Object Notation
JVM	Java Virtual Machine
JWT	JSON Web Token
LAN	Local Area Network

MIT	Massachusetts Institute of Technology
MMOG	Massive Multiplayer Online Game
MMORPG	Massive Multiplayer Online Role Playing Game
NATO	North Atlantic Treaty Organization
NES	Nintendo Entertainment System
NoSQL	Non Structured Query Language
PC	Personal Computer
PHP	Hyper Text Preprocessor
PS3	Playstation 3
PWA	Progressive Web App
RPG	Role Playing Games
SDK	Software Development Kit
SNES	Super Nintendo Entertainment System
SVG	Scalable Vector Graphics
UDK	Unity Development Kit
USSR	Union of Soviet Socialist Republics
VR	Virtual Reality
WebGL	Web Graphic Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language
Y2K	Year 2000 problem

Κεφάλαιο 1ο: Εισαγωγή

Τα τελευταία χρόνια, τα ηλεκτρονικά παιχνίδια έχουν εισέλθει στις ζωές εκατομμυρίων ανθρώπων, ανεξαρτήτου ηλικίας. Προσφέρουν διασκέδαση αλλά και έναν τρόπο διαφυγής από τα προβλήματα της πραγματικότητας. Επίσης, πολλά από αυτά παρέχουν και βοηθούν στην εκπαίδευση αλλά και σε διάφορα κοινωνικά και πολιτισμικά επίπεδα. Με την εμφάνιση των υπολογιστών, εντάχθηκαν και τα ηλεκτρονικά παιχνίδια στη ζωή μας και εισέβαλαν στην καθημερινότητά μας. Αν και ο αρχικός στόχος κατασκευής του υπολογιστή ήταν η επίλυση προβλημάτων, με την πάροδο του χρόνου εξελίχθηκαν και τα ηλεκτρονικά παιχνίδια.

Η πτυχιακή εργασία έχει ως σκοπό την ανάπτυξη ενός διαδικτυακού παιχνιδιού με θέμα την Ελληνική μυθολογία. Στοχεύει στην κατανόηση της διαδικασίας της ανάπτυξης και των τεχνολογιών που συμβάλλουν σε αυτήν. Παράλληλα, αποσκοπεί στη δημιουργία ενός εκπαιδευτικού παιχνιδιού, που πέρα από την γνώση προσφέρει και διασκέδαση. Οι τεχνολογίες που χρησιμοποιήθηκαν για την παρούσα εφαρμογή είναι η Firebase για το μέρος του back end, καθώς και η Angular για την ανάπτυξη του front end μέρους.

Στο δεύτερο κεφάλαιο ορίζεται η έννοια του παιχνιδιού και τα χαρακτηριστικά του. Στη συνέχεια γίνεται ανάλυση των ηλεκτρονικών παιχνιδιών, ξεκινώντας από μία ιστορική αναδρομή προκειμένου να φανεί η εξέλιξη των υπολογιστών και κατ' επέκταση των ηλεκτρονικών παιχνιδιών που είναι και το σημείο ενδιαφέροντος. Επισημαίνεται η ανάπτυξή τους και οι ρόλοι των ανθρώπων που παίρνουν μέρος σε αυτήν, καθώς επίσης και τα στάδια ανάπτυξης που ακολουθούνται προκειμένου να δημιουργηθεί ένα παιχνίδι. Στη συνέχεια, περιγράφεται η μεταφορά των επιτραπέζιων παιχνιδιών σε παιχνίδια υπολογιστή με κάποια παραδείγματα τέτοιων κλασικών παιχνιδιών που εντάχθηκαν και ψηφιακά, και δίνοντας έμφαση στα εκπαιδευτικά παιχνίδια. Παράλληλα, αναλύεται η κατηγορία των Multiplayer Computer Games και η αρχιτεκτονική τους.

Στο τρίτο κεφάλαιο, και επικεντρώνοντας στα παιχνίδια που ασχολούνται με την ελληνική μυθολογία, γίνεται αναφορά των κατηγοριών παιχνιδιών με αυτό το θέμα. Παράλληλα, περιγράφονται και οι τεχνολογίες που χρησιμοποιούνται σε αυτά τα παιχνίδια.

Στο επόμενο κεφάλαιο, περιγράφονται αναλυτικά όλες οι τεχνολογίες που υπάρχουν και μπορούν να χρησιμοποιηθούν για τη κατασκευή ενός ηλεκτρονικού παιχνιδιού. Αφού γίνει αυτή η ανάλυση, στη συνέχεια αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν στο παιχνίδι που πραγματεύεται η συγκεκριμένη πτυχιακή εργασία και περιγράφονται οι λόγοι που επιλέχθηκαν αυτές.

Στο τελευταίο τμήμα της εργασίας, περιγράφεται το παιχνίδι και παρουσιάζεται η υλοποίηση του μέσω των τεχνολογιών Firebase και Angular. Γίνεται εκτενής αναφορά στο πώς χρησιμοποιούνται αυτές οι τεχνολογίες και πώς εκμεταλλεύεται κάποιος τα εργαλεία και τις διεργασίες που του παρέχουν. Τονίζεται η δυνατότητα της Angular να οργανώνει τον κώδικα σε επιμέρους τμήματα (components) και την επεκτασιμότητα (scalability) του server μέσω των υπηρεσιών της Firebase.

Κεφάλαιο 2ο: Ανάπτυξη παιχνιδιών

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο ορίζεται η έννοια του παιχνιδιού, και ειδικότερα του ηλεκτρονικού, και η διαδικασία ανάπτυξής του. Θα περιγραφούν οι ειδικότητες που λαμβάνουν μέρος στην υλοποίηση ενός ηλεκτρονικού παιχνιδιού από την αρχή ως το τέλος και το πώς συμβάλλει η καθεμία σε αυτό.

2.2 Ορισμός παιχνιδιού

Προκειμένου να αποσαφηνιστεί ο όρος ενός ηλεκτρονικού παιχνιδιού, πρέπει πρώτα να οριστεί η λέξη παιχνίδι. Παιχνίδι είναι μια δομημένη δραστηριότητα, που έχει σκοπό την ψυχαγωγία ή και την εκπαίδευση του παίκτη. Προσελκύει έντονα την προσοχή του παίκτη και κρατάει αμείωτο το ενδιαφέρον του. Σύμφωνα με τον Roger Caillois, στο “Les jeux et les hommes”, το παιχνίδι ορίζεται ως μια μη υποχρεωτική δραστηριότητα, με δικό της χώρο και χρόνο, έχει αβέβαιο αποτέλεσμα, δεν αποφέρει υλικά πλούτη, ελέγχεται από κανόνες και περιέχει στοιχεία φανταστικά και εξωπραγματικά [1].

Τα βασικά χαρακτηριστικά ενός παιχνιδιού είναι [1]:

- **Οι κανόνες:** Καθορίζουν την έκβασή του και οδηγούν τον παίκτη να ακολουθήσει συγκεκριμένη πορεία για να φτάσει στο επιθυμητό αποτέλεσμα. Ορίζουν το δίκαιο στο παιχνίδι και προκαλούν τον ενθουσιασμό του παίκτη.
- **Οι σκοποί και οι στόχοι:** Βασικότερο κίνητρο ενός παίκτη είναι η επίτευξη των στόχων του. Οι στόχοι ορίζονται μαζί με τους κανόνες στην αρχή του παιχνιδιού.
- **Η έκβαση και η ανάδραση:** Τα περισσότερα παιχνίδια έχουν ως στόχο την νίκη. Η επιτυχία ή η αποτυχία ενός παίκτη επηρεάζει ανάλογα και τα συναισθήματα του. Η ανάδραση αποτελεί το στοιχείο που μας ενημερώνει για την έκβαση του παιχνιδιού και είναι αυτή που κάνει το παιχνίδι διαδραστικό.
- **Η σύγκρουση/ο ανταγωνισμός/η πρόκληση/η αντιπαράθεση:** Αυτά είναι τα εμπόδια που πρέπει να ξεπεράσει ένας παίκτης αλλά και τα βασικά σημεία που τραβούν το ενδιαφέρον και την προσοχή του.
- **Η διάδραση:** Είναι η δυνατότητα ενός παίκτη να συνεργάζεται ή να ανταγωνίζεται άλλους παίκτες.
- **Η αναπαράσταση ή το σενάριο:** Αφορά το θέμα με το οποίο σχετίζεται το κάθε παιχνίδι. Κοινώς, είναι η αφήγησή του.

2.2.1 Ηλεκτρονικά παιχνίδια

Με τον όρο ηλεκτρονικό παιχνίδι, εννοείται οποιοδήποτε παιχνίδι πραγματοποιείται με τη χρήση κάποιας ηλεκτρονικής συσκευής. Περιλαμβάνει την ανθρώπινη αλληλεπίδραση με μια διεπαφή χρήστη (user interface) για την παραγωγή οπτικής ανάδρασης σε μια συσκευή απεικόνισης (οθόνη υπολογιστή, κινητό, tablet κ.α.) [2].

Τα ηλεκτρονικά παιχνίδια χωρίζονται σε κατηγορίες ανάλογα με τους στόχους και τους σκοπούς τους [1],[3].

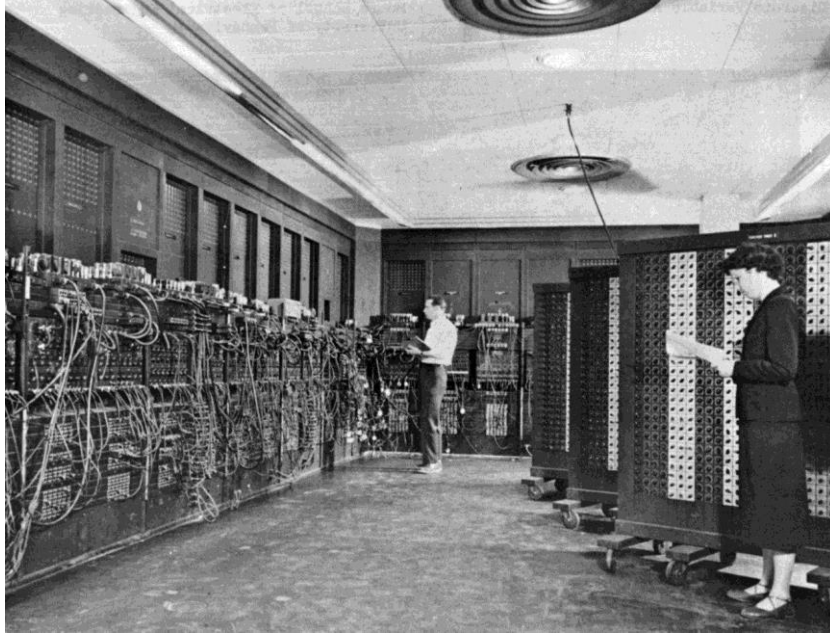
- **Παιχνίδια δράσης:** Βασικό χαρακτηριστικό της κατηγορίας αυτής είναι η απαίτηση για γρήγορα αντανακλαστικά και καλό συγχρονισμό ματιού - χεριού. Με την πάροδο του χρόνου δημιουργήθηκαν και κάποιες υποκατηγορίες των παιχνιδιών δράσης: βολών πρώτου προσώπου (fps), βολών τρίτου προσώπου (third person shooter), πάλης και πλατφόρμας.
- **Παιχνίδια περιπέτειας:** Προέκυψαν όταν εμφανίστηκαν οι προσωπικοί υπολογιστές οι οποίοι υστερούσαν σε σχέση με τις κονσόλες. Σε αυτά τα παιχνίδια, ο παίκτης πρέπει να σκεφτεί, να εξερευνήσει και να αναζητήσει λύσεις. Το δυνατό τους σημείο είναι το σενάριο και οι λεκτικές περιγραφές. Συχνά, ενώνονται με τα παιχνίδια δράσης, δημιουργώντας μια νέα κατηγορία: παιχνίδια περιπέτειας- δράσης.
- **Παιχνίδια στρατηγικής:** Ο παίκτης χειρίζεται μια ή περισσότερες ομάδες ανθρώπων, και όχι έναν ήρωα, και κατέχει κάποιου είδους δύναμη (π.χ. στρατός). Απαιτούν σύνθετο χειρισμό, για αυτό και αναπτύχθηκαν κυρίως για ηλεκτρονικούς υπολογιστές.
- **Παιχνίδια ρόλων (RPG):** Σε αυτά ο παίκτης χειρίζεται έναν ήρωα ο οποίος στην αρχή είναι αδύναμος. Στόχος του παίκτη είναι σταδιακά να αυξήσει τη δύναμη του ήρωά του, μέσα από αποστολές και κινδύνους που καλείται να ανταπεξέλθει. Τα παιχνίδια αυτά παίζονται κυρίως μέσω διαδικτύου από πολλούς παίκτες.
- **Παιχνίδια σκέψης - γρίφων:** Στόχος τους είναι η λύση μια σειράς προβλημάτων σε ορισμένο χρόνο και με συγκεκριμένο βαθμό δυσκολίας (π.χ. Tetris). Έχουν μικρές απαιτήσεις με αποτέλεσμα να εντάσσονται εύκολα σε κινητά τηλέφωνα και συσκευές με χαμηλή υπολογιστική ισχύ.
- **Παιχνίδια μάχης:** Συνήθως περιέχουν δύο ήρωες που μάχονται μεταξύ τους σε μία αρένα με σκοπό ένας από τους δύο να κερδίσει (π.χ. Tekken, Street Fighter).
- **Παιχνίδια εξομοίωσης:** Γίνεται μεταφορά του πραγματικού κόσμου στον κόσμο του παιχνιδιού. Λόγω του μεγάλου εύρους της περιλαμβάνει και τις εξής υποκατηγορίες:
 - Παιχνίδια οδηγικής εξομοίωσης: Ο παίκτης χειρίζεται εικονικά μια σειρά από μηχανήματα του πραγματικού κόσμου (π.χ. πλοία, αεροσκάφη, αυτοκίνητα).
 - Αθλητικά παιχνίδια: Είναι σχεδόν σαν τα παιχνίδια δράσης απλώς αφορούν αθλήματα. Ο παίκτης χειρίζεται έναν ή περισσότερους παίκτες ταυτόχρονα.
 - Παιχνίδια κατασκευής και management: Βασίζονται στην ανάπτυξη μιας κοινωνίας, ή και τη διοίκηση μιας επιχείρησης. Είναι ιδιαίτερα προσιτά στο γυναικείο φύλο και απαιτούν σκέψη και σχεδιασμό.
 - Μουσικά παιχνίδια: Σε αυτά ο παίκτης μπορεί να πραγματοποιήσει χορευτικές κινήσεις ή να παίζει κάποιες νότες ανάλογα με το τι προβάλλεται.

2.3 Ιστορική αναδρομή ηλεκτρονικών παιχνιδιών

Η ιστορία των video games ξεκινά επί της ουσίας στα τέλη της δεκαετίας του '40, ενώ έγιναν ευρέως αποδεκτά και εντάχθηκαν στην καθημερινή ζωή των ανθρώπων κατά τη δεκαετία '70 και '80 όπου και εισήχθησαν τα arcade games, οι κονσόλες παιχνιδιών και τα pc games. Με την πάροδο των χρόνων, η βιομηχανία των βιντεοπαιχνιδιών αναπτύχθηκε ραγδαία και αυτή τη στιγμή αποτελεί μία από τις πιο κερδοφόρες στον κόσμο. Σήμερα, τα ηλεκτρονικά παιχνίδια είναι μία δημοφιλής μορφή διασκέδασης και ψυχαγωγίας και αποτελούν μέρος του παγκόσμιου πολιτισμού.

2.3.1 Δεκαετία 1940

Αρχικά, σημαντικό είναι να αναφερθεί ότι το 1945, κατασκευάστηκε ο πρώτος μεγάλης κλίμακας επαναπρογραμματιζόμενος υπολογιστής ο ENIAC (Εικόνα 2.1). Αυτός ο υπολογιστής, ήταν το πρώτο βήμα για τη μετέπειτα δημιουργία των ηλεκτρονικών παιχνιδιών.



Εικόνα 2.1 Υπολογιστής ENIAC

Το 1947 οι Thomas T. Goldsmith Jr. και Estle Ray Mann άρχισαν να δημιουργούν μια εργασία βασισμένη στις θερμικές βαλβίδες που είχαν εφεύρει. Χρησιμοποίησαν οχτώ θερμικές βαλβίδες τις οποίες τοποθέτησαν στον υπολογιστή που τότε ήταν στα πρώτα του βήματα. Αντί για ηλεκτρονικά σχέδια χρησιμοποιούσαν αυτές τις βαλβίδες όπου το φως τους εμφανιζόταν μέσα από την οθόνη. Έτσι, δημιουργήθηκε το πρώτο ηλεκτρονικό παιχνίδι με το όνομα “Cathode Ray Tube Amusement Device” (Εικόνα 2.2). Το παιχνίδι αυτό, εμπνευσμένο από τον Β΄ Παγκόσμιο Πόλεμο, ήταν ένας εξομοιωτής πυραύλων. Χρησιμοποιούσε αναλογικό και όχι ψηφιακό κύκλωμα για τον έλεγχο δέσμης CRT και για την τοποθέτηση κουκίδας σε οθόνη [4].



Εικόνα 2.2 Cathode Ray Tube Amusement Device

2.3.2 Δεκαετία 1950

Το 1950, οι Claude Shannon και Alan Turing άρχισαν να δημιουργούν το διάσημο επιτραπέζιο παιχνίδι “Chess”, σε υπολογιστή. Αν και ήταν αρκετά δύσκολο και χρονοβόρο καθώς απαιτούσε τη δημιουργία τεχνητής νοημοσύνης για να μπορεί ο παίκτης να παίζει μόνος του, κατάφεραν να το τελειώσουν τον ίδιο χρόνο [4].

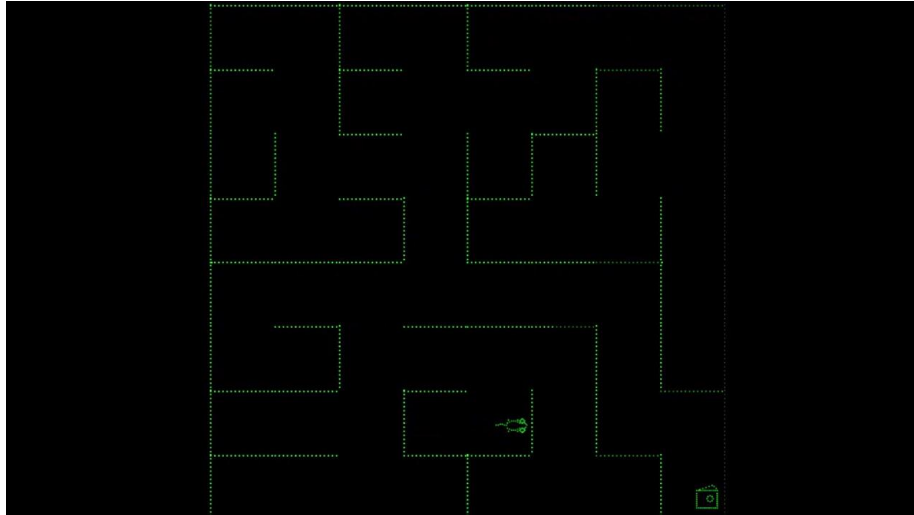
Το 1952, ο Βρετανός Καθηγητής A.S. Douglas, δημιούργησε το παιχνίδι OXO ή αλλιώς tic-tac-toe, ως μέρος της διδακτορικής του διατριβής στο Πανεπιστήμιο του Cambridge. Το παιχνίδι αυτό “έτρεχε” στον υπολογιστή EDSAC (Electronic Delay Storage Automatic Calculator) [4].

Το 1955, ο αμερικανικός στρατός δημιούργησε ένα παιχνίδι προσομοίωσης πολέμου με όνομα “Hutspiel”, στο οποίο η κόκκινη ομάδα αντιπροσώπευε το NATO και η μπλε ομάδα την USSR [4].

Το 1956, ο Arthur Samuel ανέπτυξε ένα πρόγραμμα για το επιτραπέζιο παιχνίδι “Checkers” (Ντάμα) για τον υπολογιστή IBM 701 [4].

Το 1958, ο Αμερικανός φυσικός William Higinbotham, δημιούργησε το παιχνίδι “Tennis for Two”, το οποίο χαρακτηρίστηκε ως το πρώτο διαδραστικό παιχνίδι. Η χρήση παλμογράφου αντί της οθόνης δίχασε το κοινό για το αν αυτό ήταν τελικά ένα “βίντεοπαιχνίδι” [4 - 6].

Το 1959, μαθητές του MIT, δημιούργησαν το παιχνίδι “Mouse in the Maze” (Εικόνα 2.3). Οι παίκτες χρησιμοποιώντας ένα ειδικό στυλό με φως ως συσκευή εισόδου, τοποθετούσαν τους τοίχους τους λαβύρινθου και τις τελείες που αντιστοιχούσαν στο “τυρί” πάνω σε μια οθόνη τύπου αφής. Έπειτα, ένα ψηφιακό ποντίκι προσπαθούσε να βρει τα τυριά κινούμενο μέσα στο λαβύρινθο [4].



Εικόνα 2.3 Mouse in the Maze

2.3.3 Δεκαετία 1960

Το 1962 δημιουργήθηκε το “Spacewar”, το πρώτο computer based ηλεκτρονικό παιχνίδι, από τον Steve Russel, σπουδαστή του MIT. Στο παιχνίδι αυτό, οι δύο παίκτες προσπαθούσαν να εκτοξεύσουν τορπίλες ο ένας στον άλλο και ταυτόχρονα να αποφύγουν τη βαρύτητα του Ήλιου που τους τραβούσε προς τον “θάνατο” [4 - 6].

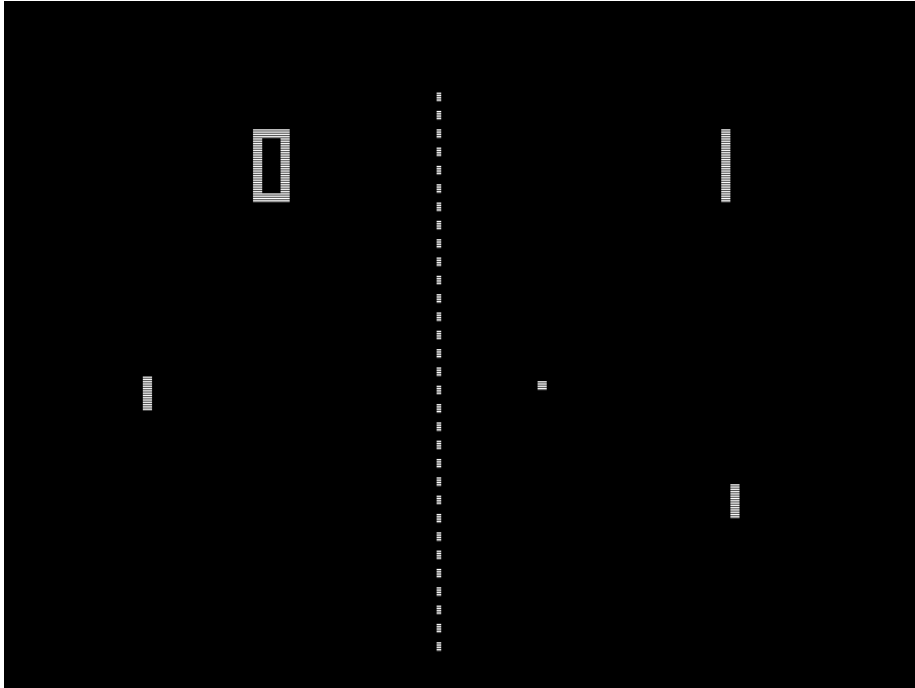
Το 1964, ο John Kemeny, πρώην βοηθός του Albert Einstein, είπε: “Ο καθένας είναι προγραμματιστής”. Συν-ανέπτυξε τη γλώσσα προγραμματισμού BASIC που επιτρέπει τη δημιουργία ηλεκτρονικών παιχνιδιών με ευκολία [4].

Το 1966-1968, οι προγραμματιστές της Sander Associates Inc. με επικεφαλής τον Ralph Baer, εφηύραν ένα πολυμεσικό σύστημα multiplayer ηλεκτρονικού παιχνιδιού, το οποίο θα μπορούσε να παιχτεί σε μια τηλεόραση. Ονομαζόταν “The Brown Box”. Χορηγήθηκε στη Magnavox, η οποία κυκλοφόρησε το σύστημα το 1972 ως Magnavox Odyssey [4 - 6].

2.3.4 Δεκαετία 1970

Το 1971, οι Nolan Bushnell και Ted Dabney, οι μετέπειτα ιδρυτές της Atari, δημιούργησαν το “Computer Space”, ένα arcade παιχνίδι, το οποίο ήταν το πρώτο εμπορικό ηλεκτρονικό παιχνίδι. Την ίδια χρονιά, τρεις φοιτητές από τη Μινεσότα, ο Don Rawitsch, ο Bill Heinemann και ο Paul Dillenberger, δημιούργησαν το “The Oregon Trail” για τον τηλετύπο. Το 1973 προσαρμόστηκε σε υπολογιστή και το 1978 διανεμήθηκε σε δισκέτες. Το σενάριο του παιχνιδιού αναφέρεται στο ταξίδι του παίκτη μαζί με την ομάδα του στο Όρεγκον και ο στόχος του είναι να προφυλάξει τα μέλη από την πείνα και τις αρρώστιες [4 - 6]. Η πιο πρόσφατη κυκλοφορία του “Oregon Trail” ήταν το 2011.

Το 1972, γεννήθηκε το Pong (Εικόνα 2.4), ο θρύλος των arcade παιχνιδιών. Οι προαναφερόμενοι ιδρυτές της Atari μαζί με τον Al Alcorn δημιούργησαν ένα επιτραπέζιο παιχνίδι πινγκ πονγκ, για arcade μηχανή. Γρήγορα έγινε πολύ δημοφιλές και ήταν ο λόγος που άρχισαν να μπαίνουν arcade μηχανές σε μπαρ και εμπορικά κέντρα [4 - 6].



Εικόνα 2.4 Pong

Το 1974, κυκλοφόρησε το πρώτο first person shooter game, το “Maze War”. Σε αυτό, ο παίκτης κινείται σε ένα λαβύρινθο, προσπαθώντας να βρει άλλους παίκτες και να πυροβολήσει [4].

Το 1977, η Atari κυκλοφόρησε το Video Computer System, γνωστό ως Atari 2600 (Εικόνα 2.5), την πρώτη κονσόλα παιχνιδιών για το σπίτι. Ήταν ένα πρωτοποριακό σύστημα με κασέτες, στις οποίες ο παίκτης μπορούσε να αποθηκεύσει πληροφορίες που αφορούσαν το παιχνίδι. Περιλάμβανε επίσης ένα joystick, χρωματική παλέτα, αυξανόμενο αριθμό παιχνιδιών που μπορούσε να “τρέξει” και τη δυνατότητα αλλαγής επιπέδων δυσκολία [4 - 6].



Εικόνα 2.5 Atari 2600

Το 1978, ο Niskikado Tomohiro, δημιούργησε το “Space Invaders”, ορόσημο των arcade games για τη δεκαετία του ‘70. Είχε πολύ μεγάλη απήχηση καθώς ήταν πρωτοπόρο στα shooting games, εισάγοντας το top-down gameplay, όπου ο παίκτης πυροβολεί εξωγήινα διαστημόπλοια που εμφανίζονται από το πάνω μέρος της οθόνης [4 - 6].

Το 1979, κυκλοφόρησε το “Asteroids” από την Atari. Σε αυτό, ο παίκτης έλεγχε ένα διαστημόπλοιο, μέσα σε ένα πεδίο αστεροειδών από το οποίο περνάνε ιπτάμενοι δίσκοι. Σκοπός του παιχνιδιού ήταν ο παίκτης να πυροβολήσει και να καταστρέψει αστεροειδείς και ιπτάμενους δίσκους χωρίς να συγκρουστεί ή να χτυπηθεί από αντίπαλα πυρά [4 - 6].

2.3.5 Δεκαετία 1980

Το 1980, ο Iwatani Toru, επηρεασμένος από μία πίτσα, σχεδίασε ένα από τα πιο διάσημα ηλεκτρονικά παιχνίδια στον κόσμο, το “Pac-man”. Επέλεξε το σχεδιασμό του με λαβύρινθο και φαντασματάκια στοχεύοντας ένα ευρύ κοινό, αγοριών και κοριτσιών, το οποίο θα κατάφερνε με τη δημιουργία ενός μη βίαιου παιχνιδιού. Κυκλοφόρησε από την Namco και κατέκτησε τις μεγαλύτερες πωλήσεις. Έγινε το πρώτο arcade παιχνίδι αλλά και το πρώτο που είχε κινούμενο και επώνυμο κύριο χαρακτήρα. Δύο χρόνια αργότερα, βγήκε και η θηλυκή έκδοσή του, το “Ms Pac-man”. Στόχος του παιχνιδιού είναι ο παίκτης να συγκεντρώσει πόντους τρώγοντας όλα τα Pac-Dots στο λαβύρινθο και έτσι να ολοκληρώσει την πίστα και να πάει στην επόμενη [4 - 6].

Το 1981, κυκλοφόρησε από την Nintendo, το “Donkey Kong”. Σηματοδότησε την αρχή της σειράς παιχνιδιών Mario [4 - 6].

Το 1983, λανσαρίστηκε το ηλεκτρονικό παιχνίδι “Star Wars” από την Atari [5]. Ήταν ένα first person space simulator παιχνίδι που προσομοίωνε την επίθεση στο Death Star από την ομώνυμη ταινία. Αποτελούνταν από 3D διανυσματικά γραφικά.

Το 1985, η Nintendo κυκλοφόρησε την κονσόλα NES (Nintendo Entertainment System) ή αλλιώς Famicom. Είχε βελτιωμένα 8-bit γραφικά, χρώματα, ήχο και gameplay από τις προηγούμενες κονσόλες, με πάνω από 800 τίτλους παιχνιδιών συμπεριλαμβανομένων και των Metroid, Megaman, Castlevania, Legend of Zelda, Ducktales, Super Mario Bros [4 - 6].

Το 1988, εμφανίστηκε το Final Fantasy (Εικόνα 2.6) από την Square Enix [6]. Είναι ένα παιχνίδι ρόλων που μέχρι και σήμερα εξελίσσεται και φέρνει καινούργιες εκδοχές και τίτλους.



Εικόνα 2.6 Final Fantasy II

Το 1989, ξεκινά η εποχή του φορητού παιχνιδιού με την κυκλοφορία του Game Boy από την Nintendo. Ξεκίνησε με το παιχνίδι Tetris. Έναν χρόνο αργότερα κυκλοφόρησε και την SNES (Super Nintendo Entertainment System), μια 16-bit κονσόλα [4 - 6].

2.3.6 Δεκαετία 1990

Το 1990, η Microsoft στα Windows 3.0 λάνσαρε το Solitaire, την ηλεκτρονική έκδοση της γνωστής πασιέντζας. Βοήθησε πολύ στην εξοικείωση των ανθρώπων με τα ηλεκτρονικά παιχνίδια και γρήγορα έγινε ένα από τα δημοφιλέστερα [4].

Το 1991, κυκλοφόρησε το “Sonic the Hedgehog”, που έγινε σήμα κατατεθέν της Sega. Την ίδια χρονιά εμφανίστηκε και το “Street Fighter II” από την Capcom, ένα fighting game που απεικόνιζε το αίμα όταν ένας χαρακτήρας χτυπούσε τον άλλο. Έτσι εισάχθηκε η βία στα παιχνίδια. Ένα παρόμοιο παιχνίδι είναι το Mortal Kombat (1992), το οποίο μαζί με το προηγούμενο αποτελούν μέχρι σήμερα τα πιο δημοφιλή fighting games [4 - 6].

Το 1993, βγήκε το “Doom” (Εικόνα 2.7). Θεωρείται ένας από τους πιο σημαντικούς τίτλους στην ιστορία των ηλεκτρονικών παιχνιδιών, καθώς πρωτοπόρησε ως first person shooter game. Η κυκλοφορία του σε συνδυασμό με το Mortal Kombat και την βιαιότητα που πρόβαλλαν, οδήγησε ένα χρόνο αργότερα, στην ίδρυση του Entertainment Software Ratings Board από την Entertainment Software Association, ενός οργανισμού για τον καθορισμό της επιτρεπόμενης ηλικίας και την αξιολόγηση του περιεχομένου των βιντεοπαιχνιδιών [4 - 6].



Εικόνα 2.7 Doom

Μέχρι τότε, στα ηλεκτρονικά παιχνίδια δε χρησιμοποιούνταν δίκτυα για την επικοινωνία δύο ή περισσότερων ανθρώπων. Με την εμφάνιση όμως των δικτύων LAN και αργότερα του Internet, εμφανίστηκαν τα πρώτα multiplayer παιχνίδια που επέτρεπαν τους παίκτες να παίζουν από απόσταση.

Το 1994, η Blizzard κυκλοφόρησε το “Warcraft: Ores & Humans” (Εικόνα 2.8), ένα από τα πρώτα παιχνίδια στρατηγικής με λεπτομερείς αποστολές. Μέχρι και σήμερα, η σειρά Warcraft προσελκύει εκατομμύρια ανθρώπους [4], [5].



Εικόνα 2.8 Warcraft: Orcs & Humans

Το 1995, με την κυκλοφορία του PlayStation ξεκινά μια νέα γενιά παιχνιδιών με τρισδιάστατα γραφικά. Τα παιχνίδια αυτά απαιτούσαν πολύ δυνατό hardware, κάτι το οποίο κατείχε το PlayStation. Διάσημοι τίτλοι αυτής της γενιάς είναι οι: Crash Bandicoot, Tomb Raider, Final Fantasy VII, Gran Turismo, Metal Gear Solid, The Legend of Zelda: Ocarina of Time, Medal of Honor, Silent Hill, Super Mario 64 και το Half Life [4 - 6].

Τέλος, δεν μπορούμε να παραλείψουμε το πιο διάσημο mobile game, το Snake, το οποίο κυκλοφόρησε το 1997 στα κινητά τηλέφωνα Nokia.

2.3.7 Δεκαετία 2000

Από το 2000 μέχρι και σήμερα έχει αυξηθεί ραγδαία η παραγωγή video games, μιας και αυτή η βιομηχανία αποφέρει τεράστια χρηματικά ποσά. Αναπτύσσονται παιχνίδια για όλα τα γούστα και τις ηλικίες με αποτέλεσμα να υπάρχει ένας ατελείωτος κατάλογος ηλεκτρονικών παιχνιδιών.

Το 2000, εμφανίστηκε πρώτη φορά το Sims (Εικόνα 2.9) από την EA. Είναι ένα simulation παιχνίδι που τράβηξε κυρίως το γυναικείο κοινό καθώς σε αυτό χτίζεις τον χαρακτήρα σου, την εμφάνισή του, το σπίτι του και γενικότερα τη ζωή του [4], [5]. Την ίδια χρονιά, κυκλοφόρησαν και άλλα αξιόλογα παιχνίδια όπως το Spiderman, Pokemon Silver Version, Counter Strike, Diablo II αλλά και το PlayStation II.



Εικόνα 2.9 Sims

Το 2001, η Microsoft κυκλοφόρησε το Xbox, το οποίο διέθετε τεχνολογία H/Y και θύρα Ethernet. Ήταν η πρώτη κονσόλα παιχνιδιών με ενσωματωμένη μονάδα σκληρού δίσκου, με σκοπό την αποθήκευση της προόδου των παιχνιδιών [4 - 6].

Το 2003, η Valve κυκλοφόρησε την ψηφιακή πλατφόρμα παιχνιδιών, το Steam, δίνοντας την δυνατότητα στους PC gamers, να "κατεβάσουν" και να παίξουν παιχνίδια στον υπολογιστή, χωρίς την ανάγκη CD ή DVD, καθώς επίσης και να κάνουν update τα παιχνίδια τους. Το Steam, μέχρι και σήμερα, αποτελεί τη μεγαλύτερη ψηφιακή πλατφόρμα παιχνιδιών, προσφέροντας πρόσβαση σε αμέτρητα βιντεοπαιχνίδια [4].

Το 2004, εισάγονται στην αγορά το Nintendo DS από τη Nintendo και το PlayStation Portable (PSP) από τη Sony, φορητές κονσόλες παιχνιδιών. Διέθεταν ενσωματωμένη οθόνη, χειριστήριο και ηχεία. Όπως τα προηγούμενα χρόνια με την κυκλοφορία του Game Boy, πρόσφεραν στον παίκτη τη δυνατότητα να παίζει ακόμα και όταν δεν βρισκόταν στο σπίτι, μιας και ήταν εύκολες στη μεταφορά, λόγω του μικρού μεγέθους [4].

Την ίδια χρονιά εμφανίστηκε το World of Warcraft (Εικόνα 2.10) από την Blizzard ένα MMORPG το οποίο μέσα σε 4 μόλις χρόνια κατάφερε να φτάσει τους 10 εκατομμύρια συνδρομητές [4]. Το παιχνίδι αυτό δημιούργησε έναν απέραντο κόσμο με αποστολές και NPCs που κρατούσαν τους παίκτες μπροστά από τους υπολογιστές τους για ατελείωτες ώρες.



Εικόνα 2.10 World of Warcraft

Το 2005 κυκλοφόρησε το Xbox 360 το οποίο έφερε μαζί του παιχνίδια σε ανάλυση HD καθώς και βελτιώσεις στα online παιχνίδια. Την επόμενη χρονιά το διαδέχθηκε το PS3 δίνοντας τη δυνατότητα στους χρήστες να παίζουν Blu-Rays, ταινίες και μουσική, καθώς και το Nintendo Wii που χαρακτηριζόταν για τους αισθητήρες κίνησης των τηλεχειριστηρίων του [4], [5].

Προς το τέλος της δεκαετίας, τα ηλεκτρονικά παιχνίδια εξαπλώθηκαν σε πλατφόρμες κοινωνικών μέσων δικτύωσης όπως το Facebook και κινητές συσκευές όπως το iPhone. Η ταχεία εξέλιξη της τεχνολογίας, η κυκλοφορία των smartphones και των app stores, εκτόξευσαν το mobile gaming. Τα παιχνίδια πλέον έχουν απήχηση σε τεράστιο εύρος κοινού, ελκύοντας ακόμη και αυτούς που δεν ασχολούνται με το gaming.

2.3.8 Δεκαετία 2010

Απο το 2010 έως και σήμερα η εξέλιξη των 3D γραφικών στα games είναι τόσο μεγάλη, που τα κάνει να φαίνονται απίστευτα ρεαλιστικά, ενώ εισάγεται και η εικονική πραγματικότητα (Virtual Reality) στα games. Παράλληλα, δίνεται η δυνατότητα για πρώτη φορά στην ιστορία των ηλεκτρονικών παιχνιδιών οι παίκτες να παίζουν μεταξύ τους από διαφορετικές κονσόλες (Cross-Platform).

Σε αυτή τη δεκαετία, πολλά παιχνίδια διάλεξαν έναν τρόπο λίγο παράδοξο για να αποφέρουν χρήματα. Αυτό το μοντέλο ονομάζεται Free to Play και το υιοθετούν πάρα πολλές εταιρίες για τα παιχνίδια τους, ειδικά τα mobile. Επιτρέπουν στον χρήστη να παίζει δωρεάν και διαθέτουν κάποιο premium subscription ή in-game-purchase με σκοπό το κέρδος.

Το 2010 εμφανίστηκε ένα indie game (παιχνίδι φτιαγμένο από ένα άτομο), με μεγάλη απήχηση στα παιδιά, το Minecraft (Εικόνα 2.11). Ο δημιουργός του Markus Persson (γνωστός ως Notch) διάλεξε έναν διαφορετικό τρόπο απεικόνισης των γραφικών με την μέθοδο Voxel (Voxel graphics). Αυτή η

μέθοδος του επέτρεψε να ασχοληθεί εκτενέστερα με την ποιότητα και τους μηχανισμούς του παιχνιδιού χωρίς όμως να δώσει έμφαση στο art design [7].



Εικόνα 2.11 Minecraft

Το 2016 κυκλοφόρησε το Oculus Rift, ένας εξοπλισμός headset εικονικής πραγματικότητας. Αποτελείται από μία συσκευή με οθόνη και ακουστικά, την οποία ο παίκτης φοράει στο κεφάλι του και απο μοχλούς που κρατάει στα χέρια με τους οποίους αλληλεπιδρά στο παιχνίδι. Έκτοτε, πολλά παιχνίδια ενσωμάτωσαν τη δυνατότητα χρήσης VR.

Η ιστορία των ηλεκτρονικών παιχνιδιών γράφεται καθημερινά με την κυκλοφορία ολοένα και περισσότερων τίτλων που απευθύνονται σε όλα τα κοινά. Κανείς δεν ξέρει πού μπορεί να φτάσει το gaming στο μέλλον, το μόνο σίγουρο είναι ότι όσο εξελίσσεται η τεχνολογία, θα αξιοποιείται από τις εταιρείες και τους developers για την δημιουργία καινοτομιών, με σκοπό την καλύτερη δυνατή εμπειρία στο gaming όσον αφορά το κομμάτι του ρεαλισμού και της διαδραστικότητας.

2.4 Διαδικασία ανάπτυξης ηλεκτρονικών παιχνιδιών και ρόλοι

Η διαδικασία ανάπτυξης ηλεκτρονικών παιχνιδιών (game development) είναι η διαδικασία δημιουργίας ενός παιχνιδιού συμπεριλαμβανομένων όλων των σταδίων που σχετίζονται με την παραγωγή, από τη σύλληψη της ιδέας μέχρι και την κυκλοφορία του τελικού προϊόντος στο κοινό. Συχνά συγχέεται με τον όρο design, που αποτελεί ένα μόνο μέρος της διαδικασίας ανάπτυξης.

Τα τελευταία χρόνια, υπήρξε ραγδαία αύξηση μικρών εταιρειών που ασχολήθηκαν με την ανάπτυξη ηλεκτρονικών παιχνιδιών. Ορμητήριο των περισσότερων, η αγάπη και το πάθος για το gaming. Ωστόσο, η ανάπτυξη είναι μια δύσκολη, χρονοβόρα και περίπλοκη διαδικασία ανεξάρτητα του επιπέδου δυσκολίας του παιχνιδιού και του πόσο εύκολο φαίνεται στο κοινό [8].

Κατά την ανάπτυξη ενός παιχνιδιού, πρωταρχικός στόχος του developer είναι να δημιουργήσει ένα διασκεδαστικό παιχνίδι. Προκειμένου να το πετύχει αυτό πρέπει να λάβει σοβαρά υπόψη του ορισμένες παραμέτρους [9] :

- **Το είδος του παιχνιδιού**

Είναι πολύ σημαντικό ο developer να γνωρίζει με ποιο είδος παιχνιδιού θα ασχοληθεί. Με βάση το είδος μπορεί να καταλάβει τον ανταγωνισμό που θα έχει αλλά και του πού περίπου θα πρέπει να κινηθεί σύμφωνα με παιχνίδια του ίδιου είδους. Μπορεί να κάνει αλλαγές ή να προσθέσει νέα στοιχεία που θα

δώσουν κάτι καινοτόμο στο παιχνίδι και θα το διαχωρίσουν από τα υπόλοιπα του είδους του χωρίς όμως να αποξενώσουν τον παίκτη από τις προσδοκίες που είχε.

- **Το αγοραστικό κοινό**

Όσο πιο διαφορετικό είναι το παιχνίδι που δημιουργείται, τόσο πιο δύσκολο είναι να εντοπιστεί και το κοινό στο οποίο απευθύνεται. Ο σχεδιαστής θα πρέπει να αναζητήσει πληροφορίες για το κοινό του είτε μέσω διαδικτύου είτε μέσα από βιβλία και περιοδικά, προκειμένου να μπορέσει να ανταποκριθεί στις προσδοκίες του.

- **Η πηγή της διασκέδασης**

Για να καταστεί ένα παιχνίδι διασκεδαστικό, πρέπει πρώτα να εντοπίσει ο ίδιος ο developer τι είναι αυτό που τον διασκεδάζει. Παίζοντας ένα παιχνίδι, πρέπει να παρατηρήσει και να εξηγήσει στον εαυτό του ποια είναι τα σημεία που τον διασκεδάζουν, προκειμένου να τα εντοπίσει και να τα εφαρμόσει και στο δικό του project.

- **Η πρωτοτυπία**

Ο developer πρέπει να εισάγει κάτι καινοτόμο στο παιχνίδι του που θα το ξεχωρίσει από τα υπόλοιπα του είδους του. Αυτό θα έχει ως αποτέλεσμα να προσελκύσει τους παίκτες και να κρατά αμείωτο το ενδιαφέρον τους.

- **Η πρώτη εντύπωση**

Πολλές φορές ο παίκτης σταματά να παίζει από τα πρώτα λεπτά διότι το παιχνίδι δε του φαίνεται αρκετά ενδιαφέρον ή δεν καταλαβαίνει τι πρέπει να κάνει. Για αυτό το λόγο, πρέπει να εισαχθεί στο παιχνίδι το στοιχείο εκείνο που θα μαγνητίσει τον παίκτη και παράλληλα θα του εξηγήσει πώς παίζεται.

- **Ο ρόλος του παίκτη**

Πρέπει να δημιουργήσει ένα παιχνίδι γεμάτο προκλήσεις, στο οποίο ο παίκτης θα αναλαμβάνει ένα ρόλο που θα τον κάνει να νιώθει δυνατός και με εξουσία. Σκοπός του θα είναι να ανταπεξέλθει στις προκλήσεις με επιτυχία.

- **Ο σεβασμός στον παίκτη**

Κερδίζεται με ποικίλους τρόπους όπως επιτρέποντας στον παίκτη να αποχωρεί από το παιχνίδι χωρίς να χάσει την πρόοδό του (save), να έχει επιλογές για το πώς θα αντιμετωπίσει τις προκλήσεις, να είναι ενημερωμένος για τους κανόνες του παιχνιδιού και να γνωρίζει τις συνέπειες σε περίπτωση αποτυχίας καθώς και να έχει άμεσα τη δυνατότητα να ξαναπροσπαθήσει.

- **Η αγάπη για το παιχνίδι**

Προκειμένου να έχει απήχηση στο κοινό το παιχνίδι, πρέπει πρώτα να το αγαπά ο ίδιος ο σχεδιαστής. Παράλληλα, πρέπει να αγαπά τη δουλειά του και να αντιμετωπίζει την ανάπτυξη του παιχνιδιού ως μια συνεχόμενη πρόκληση.

- **To design document**

Είναι πάρα πολύ σημαντικό κατά την ανάπτυξη ενός ηλεκτρονικού παιχνιδιού να υπάρχει οργάνωση. Για αυτό το λόγο, ο developer πρέπει να δημιουργήσει ένα οργανόγραμμα προκειμένου να κρατά σημειώσεις και να ξέρει ανά πάσα ώρα και στιγμή τι εργασία πρέπει να επιτελέσει.

2.4.1 Στοιχεία που δομούν ένα ηλεκτρονικό παιχνίδι

Ένα video game χαρακτηρίζεται από συγκεκριμένα στοιχεία τα οποία είναι απαραίτητα για την σωστή ανάπτυξή του [3]:

- **Ο χώρος (space):** τα μέρη του παιχνιδιού, το μέγεθος, η διάσταση και τα όρια του κόσμου που ο παίκτης κινείται.
- **Τα αντικείμενα (objects):** έχουν ένα ή περισσότερα χαρακτηριστικά που περιλαμβάνουν πληροφορίες (π.χ. χαρακτήρες, σκορ, μπάρα ζωής και οτιδήποτε μπορεί να αλληλεπιδράσει ο παίκτης)
- **Ο χειρισμός:** τα κουμπιά (**keys**) που μπορεί να χρησιμοποιήσει ο παίκτης για να χειριστεί τον χαρακτήρα του ή να ελέγξει μια κατάσταση.
- **Η δράση (action):** είναι η ενεργητική που αφορά τις βασικές κινήσεις του παίκτη και η προκύπτουσα που αφορά τις επιπτώσεις των ενεργειών του στο μέλλον.
- **Οι κανόνες (rules):** είναι το βασικότερο στοιχείο του παιχνιδιού. Καθορίζουν τον τρόπο που ο παίκτης θα αλληλεπιδράσει με τον κόσμο του παιχνιδιού, τα δικαιώματά του, τις ευθύνες του, καθώς και τον σκοπό του παιχνιδιού.
- **Τα επίπεδα και η δυσκολία:** ο παίκτης, ξεπερνώντας προκλήσεις, ανεβάζει τα επίπεδα δυσκολίας του παιχνιδιού.
- **Η δεξιότητα (skill):** το κάθε παιχνίδι απαιτεί την χρήση διαφορετικών δεξιοτήτων από τον παίκτη. Αυτές μπορεί να είναι:
 - **φυσικές:** απαιτούν από τον παίκτη αποτελεσματικό χειρισμό των πλήκτρων, ποντικιού, μοχλού κλπ.
 - **πνευματικές:** περιλαμβάνουν μνήμη, παρατηρητικότητα και λήψη αποφάσεων
 - **κοινωνικές:** περιλαμβάνουν πρόβλεψη κινήσεων και αντίδρασης αντιπάλου, αντιπερισπασμό και σωστό συντονισμό με τους συμπαίκτες.
- **Η τυχαιότητα (chance):** επηρεάζει τις αλληλεπιδράσεις αλλάζοντας την εξέλιξη του παιχνιδιού και προσφέροντας διασκέδαση.
- **Οι παράγοντες της αποτυχίας (losing factors):** ενημερώνουν τον παίκτη για τις συνθήκες κάτω από τις οποίες χάνει, καθώς επίσης τον παρακινούν να προσπαθήσει περισσότερο και να συγκεντρωθεί στο στόχο.
- **Η οθόνη φόρτωσης (loading screen):** είναι μια εικόνα που προβάλλεται στον παίκτη, κατά την διάρκεια φόρτωσης των στοιχείων του παιχνιδιού (π.χ. φωτογραφία, animation κ.α.) [10].
- **Οι επιλογές αποθήκευσης (save) και εγκατάλειψης (quit):** δίνει την δυνατότητα ο παίκτης να αποθηκεύει την πρόοδό του και να μπορεί να αποχωρήσει από το παιχνίδι όποτε επιθυμεί.

2.4.2 Ρόλοι στη διαδικασία ανάπτυξης παιχνιδιών

Ο αριθμός των εργαζομένων που θα αναλάβουν την ανάπτυξη ενός ηλεκτρονικού παιχνιδιού ποικίλει ανάλογα με το μέγεθός του και τις ανάγκες που πρέπει να καλυφθούν. Ένας άνθρωπος μπορεί να αναλάβει περισσότερους από ένα ρόλο, αλλά οι ρόλοι στην ανάπτυξη μένουν σταθεροί [8].

1. **Game Developer:** Είναι εκείνος που επεξεργάζεται τις λεπτομέρειες του σχεδιασμού ενός παιχνιδιού, και αναθεωρεί το παιχνίδι βάσει των κριτικών που λαμβάνει από το κοινό. Επιβλέπει το γενικό σχεδιασμό του και συντονίζει την ομάδα ώστε να υπάρξει το καλύτερο δυνατό αποτέλεσμα.
2. **Game Designer:** Υλοποιεί την ιδέα, τους κεντρικούς μηχανισμούς και του κανόνες του παιχνιδιού. Ασχολείται με όλες τις πτυχές του παιχνιδιού με τις οποίες αλληλεπιδρά ο παίκτης

(π.χ. χαρακτηριστικά, ζωή, όπλα, εμπόδια κλπ). Ερευνά και συλλέγει δεδομένα που μπορεί να του χρειαστούν κατά τη διαδικασία ανάπτυξης. Συχνά, αυτό είναι το άτομο που δίνει τον τίτλο και καθορίζει το θέμα του. Το game design αποτελεί το μεγαλύτερο κομμάτι στην ανάπτυξη video game για αυτό και πολλές φορές χωρίζεται σε επιμέρους ρόλους:

- a. Level Designer:** καθορίζει και συντονίζει τα επιμέρους επίπεδα του ηλεκτρονικού παιχνιδιού
 - b. User Interface Designer:** είναι υπεύθυνος για το σχεδιασμό και τη διάταξη της οθόνης σε διάφορα σημεία του video game
- 3. Programmer:** Είναι υπεύθυνος για την ενσωμάτωση όλων των στοιχείων που συλλέγει από όλη την ομάδα. Αναπτύσσει ή βελτιστοποιεί τη μηχανική του παιχνιδιού, κατασκευάζει ή χρησιμοποιεί τα εργαλεία για τη δημιουργία του, γράφει τον κώδικα που ενώνει τα κομμάτια, ασχολείται με τη φυσική και το περιβάλλον του παιχνιδιού, δημιουργεί τη βάση δεδομένων που αφορά χαρακτηριστικά των αντικειμένων του και τα στοιχεία ελέγχου καθώς επίσης προγραμματίζει πότε πρέπει να ενεργοποιηθεί ένα ηχητικό εφέ ή ένα σενάριο. Με άλλα λόγια, “δίνει ζωή” στο παιχνίδι.
- 4. Writer:** Είναι υπεύθυνος για τη συγγραφή της ιστορίας, του σεναρίου και των διαλόγων του παιχνιδιού.
- 5. Art Director:** Διαχειρίζεται την παραγωγή όλων των οπτικών γραφικών (visual art) στο ηλεκτρονικό παιχνίδι. Επιπλέον, αναλαμβάνει να σχεδιάσει το πώς θα φαίνεται οπτικά το μενού, η οθόνη φόρτωσης κα.
- 6. Audio Director:** Αναλαμβάνει την παραγωγή ακουστικού υλικού για το παιχνίδι (π.χ. μουσική, ήχοι περιβάλλοντος, διάλογοι κ.α.)
- 7. Game Tester:** Αναλαμβάνει τη δοκιμή όλων των στοιχείων του παιχνιδιού, της συμβατότητας κ.α. προκειμένου να σιγουρευτεί ότι λειτουργεί σωστά. Σκοπός του είναι να μειώσει πιθανά σφάλματα.

2.4.3 Στάδια ανάπτυξης ενός ηλεκτρονικού παιχνιδιού

Για την ανάπτυξη ενός ηλεκτρονικού παιχνιδιού υπάρχουν ορισμένες φάσεις μέσα από τις οποίες περνάει το παιχνίδι. Οι φάσεις αυτές δεν είναι προκαθορισμένες αλλά διαφέρουν ανάλογα με την εταιρεία και το είδος του παιχνιδιού που πρόκειται να σχεδιαστεί και να αναπτυχθεί [3].

2.4.3.1 Πρόταση παιχνιδιού

Στο πρώτο στάδιο περιλαμβάνονται οι ιδέες που εκφράζονται για το παιχνίδι. Μπορεί να είναι πρωτότυπες, να αποτελούν εξέλιξη κάποιου προηγούμενου παιχνιδιού ή να βασίζονται σε κάποια ταινία. Οι ιδέες που επικρατούν μοιράζονται σε μικρές ομάδες που αποτελούνται από τουλάχιστον έναν προγραμματιστή και ένα σχεδιαστή. Η κάθε ομάδα ετοιμάζει ένα προσχέδιο με τις βασικές πληροφορίες σχετικά με το παιχνίδι, όπως το είδος του, το κοινό στο οποίο απευθύνεται, εκτιμήσεις για το κόστος ανάπτυξής του και συνολική εκτίμηση του έργου. Παράλληλα, διαμορφώνεται το τελικό καλλιτεχνικό σχέδιο, εκεί όπου αποτυπώνεται το στυλ και το ύφος του παιχνιδιού. Η ομάδα που θα κάνει την καλύτερη πρόταση παίρνει έγκριση από τον χρηματοδότη για να προχωρήσει.

2.4.3.2 Προπαρασκευή

Αυτό το στάδιο αποτελείται από δύο φάσεις. Αρχικά, παράγεται ένα λεπτομερές σχέδιο ανάπτυξης του παιχνιδιού που θα χρησιμοποιηθεί ως πρότυπο στην τελική υλοποίηση του παιχνιδιού. Σε αυτό καταγράφονται όλες οι διαδικασίες του παιχνιδιού και γίνεται υπολογισμός για τον αριθμό των ατόμων

και του εξοπλισμού που απαιτείται. Στη δεύτερη φάση, παράγεται ένα πρώτο προϊόν του παιχνιδιού στο οποίο περιέχονται όλα τα χαρακτηριστικά, η κεντρική ιδέα και οι μηχανισμοί του παιχνιδιού.

2.4.3.3 Παραγωγή παιχνιδιού

Η φάση αυτή μπορεί να διαρκέσει από 6 μήνες έως και περισσότερο από 2 χρόνια. Τα άτομα που απαρτίζουν την ομάδα ανάπτυξης του παιχνιδιού αρχίζουν να αυξάνονται και η ομάδα μεγαλώνει κατά πολύ μέχρι την τελική φάση του παιχνιδιού. Το στάδιο αυτό είναι το πιο δύσκολο γιατί εδώ αναπτύσσονται τα πιο κομβικά σημεία του παιχνιδιού.

2.4.3.4 Στάδιο Alpha

Μόλις ολοκληρωθεί η κύρια περίοδος ανάπτυξης του παιχνιδιού, η ομάδα είναι σε θέση να δείξει το παιχνίδι και σε ανθρώπους εκτός ομάδας με την πρώτη έκδοση Alpha. Για να βγει σε αυτή την έκδοση το παιχνίδι, πρέπει ο χρήστης να μπορεί να ολοκληρώσει όλα τα επίπεδά του. Από το στάδιο αυτό και έπειτα, το παιχνίδι βρίσκεται σε δοκιμή προκειμένου να βρεθούν και να λυθούν τυχόν σφάλματα. Επιπλέον, γίνονται διορθώσεις που αφορούν την ισορροπία του παιχνιδιού. Αν το παιχνίδι προορίζεται για κονσόλα, ερευνάται κατά πόσο τηρούνται τα απαραίτητα στοιχεία και προϋποθέσεις του κατασκευαστή.

2.4.3.5 Στάδιο Beta

Σε αυτό το στάδιο απαρτίζεται η πλήρης έκδοση του παιχνιδιού και η ανάπτυξη του παιχνιδιού και κάθε είδους αλλαγή σταματά. Η προσοχή της ομάδας στρέφεται στον εντοπισμό σφαλμάτων και την εξασφάλιση της καλύτερης λειτουργίας του παιχνιδιού. Τα σφάλματα που εντοπίζονται, ταξινομούνται με βάση την κρισιμότητά τους. Τα υψηλής κρισιμότητας σφάλματα διορθώνονται πρώτα γιατί με την παρουσία τους μπορεί να οδηγήσουν σε μη έκδοση του παιχνιδιού. Τα σφάλματα μικρότερης σημασίας, αν δεν παραλειφθούν, διορθώνονται μετά την έκδοση του παιχνιδιού.

2.4.3.6 Πιστοποίηση και συσκευασία παιχνιδιού

Εφόσον έχει ολοκληρωθεί το στάδιο Beta, ο κατασκευαστής πρέπει να εξετάσει το παιχνίδι και να εγκρίνει την καλή λειτουργία και ποιότητά του. Η πιστοποίηση είναι ιδιαίτερα σημαντική γιατί χωρίς αυτή δεν μπορεί να κυκλοφορήσει το παιχνίδι. Από το στάδιο αυτό και έπειτα καμία αλλαγή δεν μπορεί να πραγματοποιηθεί. Γίνεται μαζική αναπαραγωγή και το παιχνίδι συσκευάζεται για να διατεθεί στην αγορά.

2.4.3.7 Μετά την κυκλοφορία

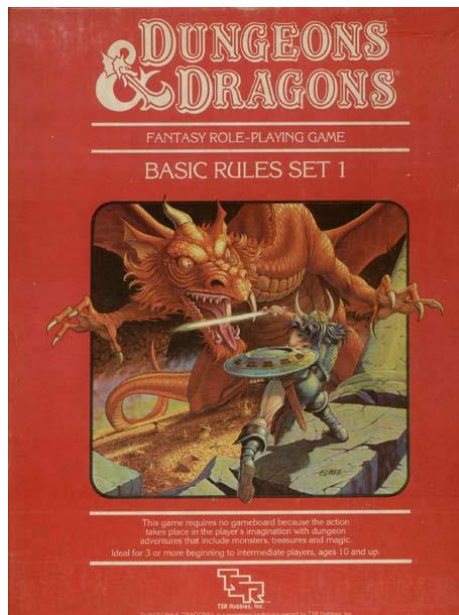
Μετά την έκδοση του παιχνιδιού, η ομάδα ανάπτυξης δε σταματά να δουλεύει. Πολλές φορές χρειάζεται να γίνουν κάποιες ακόμη διορθώσεις. Ιδιαίτερα σε περιπτώσεις που το παιχνίδι προορίζεται για προσωπικούς υπολογιστές, χρειάζεται να αντιμετωπιστεί η ασυμβατότητα του παιχνιδιού με το υλικό του υπολογιστή. Επιπλέον, παρατηρείται έντονα το φαινόμενο οι παίκτες να μπορούν να ανανεώσουν διαδικτυακά το παιχνίδι. Για να συμβεί αυτό η ομάδα ελέγχου πρέπει να εργαστεί για περισσότερο χρονικό διάστημα και μετά την έκδοση του παιχνιδιού.

2.5 Μεταφορές επιτραπέζιων παιχνιδιών σε παιχνίδια υπολογιστών

Ένα επιτραπέζιο παιχνίδι είναι ένα απλό ή πολύπλοκο σύστημα αλληλοεξαρτώμενων, αλληλεπιδρώντων και αλληλένδετων στοιχείων με βασικό κριτήριο τον αριθμό πιθανών

αποτελεσμάτων του παιχνιδιού [11]. Η δομή του αποτελείται από τα βασικά και δομικά στοιχεία, τους κανόνες και τέλος τους στόχους. Οι στόχοι του παιχνιδιού πρέπει να έχουν σαφήνεια και να δίνουν τη δυνατότητα της επιβράβευσης [12].

Η μοντέρνα βιομηχανία φανταστικού παιχνιδιού γεννήθηκε το 1973 από τον Gary Gygax και τον συνάδελφό του Dave Arneson, οι οποίοι δημιούργησαν ένα μικρό πολεμικό παιχνίδι, το γνωστό D&D (Dungeons & Dragons) (Εικόνα 2.12). Απαρτιζόταν από ιστορικές μάχες με τη χρήση μινιατούρων και δίνοντας ρόλους στους παίκτες. Σε ένα παιχνίδι όμως, πρέπει να υπάρχουν και κανόνες προκειμένου να παίρνονται αποφάσεις, να γίνονται εξερευνήσεις και δημιουργείται και ο ανταγωνισμός. Στη συνέχεια, εισήγαγαν και το ζάρι για να υπάρχει τυχαιότητα στο παιχνίδι. Το ζάρι στις επικίνδυνες καταστάσεις του παιχνιδιού ήταν αυτό που καθόριζε το αποτέλεσμα. Συνήθως διαδραματιζόταν σε μια μεγάλη ιστορία και παιζόταν σε μία ή περισσότερες συναντήσεις (game sessions) [13].



Εικόνα 2.12 Dungeons & Dragons

2.5.1 Κατηγορίες επιτραπέζιων παιχνιδιών

Τα επιτραπέζια παιχνίδια αποτελούνται από μια επιφάνεια (ταμπλό) και από ένα τουλάχιστον ζάρι, το οποίο βοηθά στον τερματισμό των παικτών. Υπάρχουν και τα παιχνίδια με κάρτες που συνήθως παίζονται με τραπουλόχαρτα (π.χ. πασιέντζα, πόκερ κλπ)

Χωρίζονται σε τρεις κατηγορίες:

- στρατηγικής (το αποτέλεσμά του κρίνεται από τις αποφάσεις του παίκτη και έχει σκοπό την κατοχή μεγάλου μέρους του ταμπλό) π.χ. σκάκι, Stratego κα
- αγώνων (ο κάθε παίκτης στοχεύει να τερματίσει πρώτος) π.χ. φιδάκι, γκρινιάρης
- συνδυασμός των δύο παραπάνω

Στη σύγχρονη εποχή τα επιτραπέζια παιχνίδια διακρίνονται σε κλασικά, όπως το σκάκι, το τάβλι και τη ντάμα, και στα σύγχρονα, όπως τη Μονόπολη, το φιδάκι, το Scrabble, το Taboo και το Trivial.

2.5.2 Η μετάβαση στα ηλεκτρονικά επιτραπέζια παιχνίδια

Αν και η μεταφορά επιτραπέζιων παιχνιδιών σε παιχνίδια υπολογιστή γίνεται όσο το δυνατόν πιο πιστά έτσι ώστε να θυμίζει ακριβώς το εκάστοτε γνωστό παιχνίδι, η ψηφιακή τους μορφή γίνεται πολλές φορές πιο ελκυστική, καθώς προσφέρει επιπλέον προκλήσεις. Τέτοιες μπορεί να είναι οι κατασκευές κτιρίων, η συντήρηση πληθυσμού ή η εξερεύνηση χαρτών.

Έτσι, για παράδειγμα, τα ηλεκτρονικά παιχνίδια στρατηγικής χωρίστηκαν σε [14]:

1. Παιχνίδια με γύρους (Turn Based Strategy) π.χ. Civilization, Total War
2. Παιχνίδια πραγματικού χρόνου (Real Time Strategy) π.χ. Age of Empires, StarCraft

Γενικά, με βάση την πολυπλοκότητά τους, τα ηλεκτρονικά επιτραπέζια παιχνίδια διακρίνονται σε [15]:

1. Roll & Move (όπου το ζάρι ελέγχει την κίνηση του παίκτη)
2. Open Movement (δεν υπάρχει καθόλου ζάρι παρά μόνο πόντοι και κίνηση)
3. Worker Placement (με πόνια σε συγκεκριμένα σημεία του ταμπλό)
4. Simultaneous Action (περιορίζει τον χρόνο αναμονής γιατί εμπλέκει τον παίκτη σε δράση - αντίδραση και λήψη αποφάσεων)
5. Role Selection (με επιλογή και ανάλυση ρόλων)
6. Cooperative Play (ιδανικό για τον εκπαιδευτικό να εμπλέξει μαθητές με διαφορετικές δεξιότητες στη δημιουργία ενός λιγότερο ανταγωνιστικού περιβάλλοντος)



Εικόνα 2.13 Pokemon Card Game

Παραδείγματα γνωστών ηλεκτρονικών παιχνιδιών με κάρτες ή που προσομοιώνουν επιτραπέζια παιχνίδια είναι: Magic: The Gathering (Εικόνα 2.14), Pokemon Card Game (Εικόνα 2.13), Hearthstone, 100% Orange Juice, Mysterium κ.α.



Εικόνα 2.14 Magic: The Gathering

2.5.3 Εκπαιδευτικά παιχνίδια

Τα εκπαιδευτικά παιχνίδια δημιουργήθηκαν κυρίως για διδακτικούς σκοπούς αλλά επίσης και για διασκέδαση και ψυχαγωγία. Διακρίνονται σε επιτραπέζια και παιχνίδια με κάρτες και σκοπό έχουν να διδάξουν, να επεκτείνουν και να αναπτύξουν τις γνώσεις των μαθητών, να βοηθήσουν στην κατανόηση ιστορικών γεγονότων και να βελτιώσουν τις δεξιότητές τους. Δίνουν κίνητρο στα παιδιά να εφαρμόσουν τις γνώσεις τους αλλά και να μάθουν επιπλέον πράγματα. Παίζονται είτε ατομικά είτε ομαδικά και στοχεύουν στο να δώσουν γνωστικές, κοινωνικές και συναισθηματικές διαστάσεις.

Τα χαρακτηριστικά ενός εκπαιδευτικού παιχνιδιού είναι να δίνει πολλά περισσότερα από την απλή διασκέδαση, να προσφέρει ευγενή άμιλλα, να εμπλέκει όλους τους μαθητές και να τους ενθαρρύνει να εστιάζουν στη μάθηση και την εξάσκηση. Είναι ένα πολύ σημαντικό εργαλείο για τον εκπαιδευτικό καθώς μέσω της χρήσης του Η/Υ, αναπτύσσεται η περιέργεια και η φαντασία στα παιδιά [16].

2.5.4 Το σκάκι

Ένα από τα πρώτα παιχνίδια που μεταφέρθηκαν στον υπολογιστή και στο οποίο χρησιμοποιήθηκε τεχνητή νοημοσύνη είναι το σκάκι. Είναι ένα παιχνίδι στρατηγικής στο οποίο παίζουν δύο παίκτες σε μια σκακιέρα 8x8 και με 32 πιόνια. Ο κάθε παίκτης έχει στην κατοχή του 16 πιόνια τα οποία έχουν διαφορετικό μέγεθος, σχήμα και κινήσεις που εκτελούν. Σκοπός του παιχνιδιού είναι να γίνει ματ στον αντίπαλο βασιλιά (Εικόνα 2.15).



Εικόνα 2.15 Το κλασικό σκάκι

Μέσω του υπολογιστή, μπορεί κανείς να παίξει σκάκι από το διαδίκτυο είτε με κάποιον φίλο του, είτε με έναν άγνωστο από την άλλη άκρη του κόσμου με ασφάλεια είτε με κάποιο σκακιστικό πρόγραμμα που χρησιμοποιεί κάποιου είδους σκακιστική μηχανή (chess engine). Έτσι, ο παίκτης μπορεί να βελτιωνεται, είτε με την εξάσκηση είτε με τη βοήθεια διαφόρων βάσεων δεδομένων. Παράλληλα, του δίνεται η δυνατότητα να λύσει διάφορα σκακιστικά προβλήματα όπως επίσης και να δει διάφορα video με επεξηγήσεις [14] (Εικόνα 2.16).

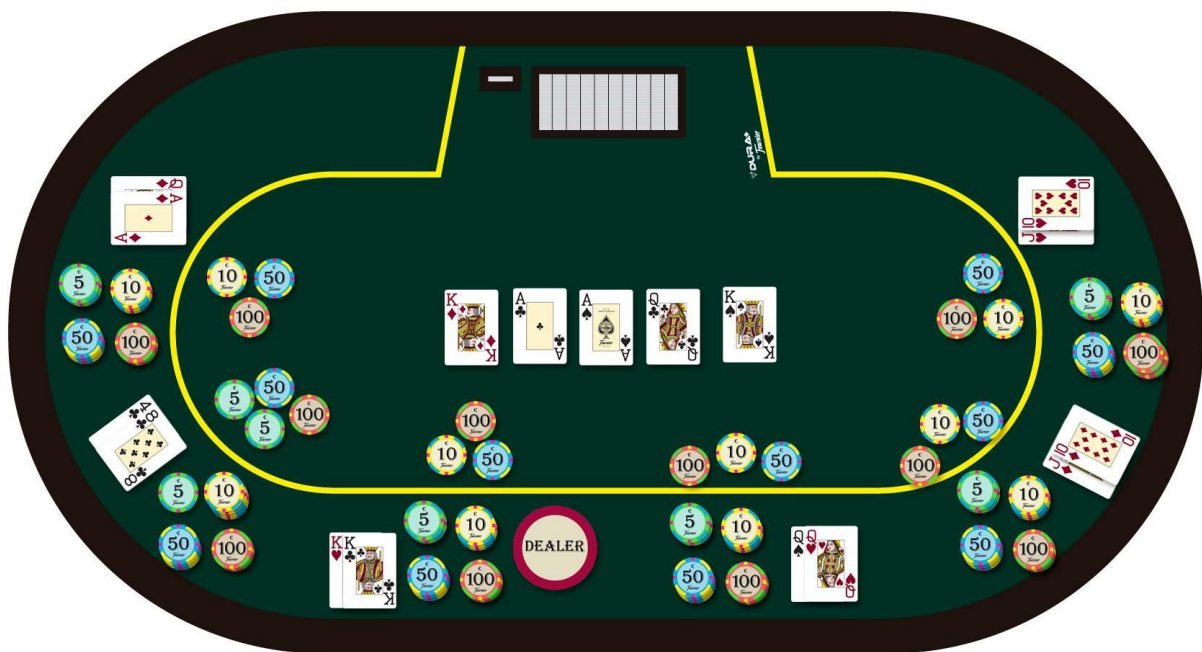


Εικόνα 2.16 Το σκάκι online

Το 1997, ο υπολογιστής κατάφερε να κερδίσει για πρώτη φορά τον καλύτερο ενεργό σκακιστή, τον Garry Kasparov, σε αγώνα έξι παρτίδων με σκορ 4-2, με τη μηχανή Deep Blue. Από τότε, οι σκακιστικές μηχανές γίνονται πιο δυνατές και με καλύτερους αλγόριθμους. Οι περισσότεροι αλγόριθμοι χρησιμοποιούν τη μέθοδο brute force για τον υπολογισμό όλων των πιθανών κινήσεων και απαιτούν αρκετή υπολογιστική ισχύ [17].

2.5.5 Το πόκερ

Το πόκερ είναι ένα παιχνίδι στο οποίο τα τελευταία χρόνια έχει εφαρμοστεί η τεχνητή νοημοσύνη. Η πιο γνωστή παραλλαγή πόκερ είναι το Texas Hold'em (Εικόνα 2.17). Η ανάπτυξη ενός παιχνιδιού πόκερ επικεντρώνεται στην ανάπτυξη αλγορίθμων για το “Opponent Modelling”, επιλέγοντας την καλύτερη στρατηγική εναντίον κάθε δεδομένου αντιπάλου. Αυτό μπορεί να βελτιώνεται συνεχώς με βάση τα συλλεχθέντα στατιστικά στοιχεία και το ιστορικό στοιχημάτων του αντιπάλου [18]. Τα χαρακτηριστικά του πόκερ, δηλαδή η ελλιπής γνώση, η διαχείριση κινδύνου, η μοντελοποίηση της προσωπικότητας των αντιπάλων και η αντιμετώπιση αναξιόπιστων πληροφοριών είναι εκείνα που το κάνουν ενδιαφέρον ως προς την τεχνητή νοημοσύνη. Η μεταφορά του στον υπολογιστή προσφέρει την δυνατότητα εκμάθησης το παιχνιδιού και των στρατηγικών του καθώς επίσης και τη δημιουργία ενός “τραπεζιού” με παίκτες από όλο τον κόσμο.



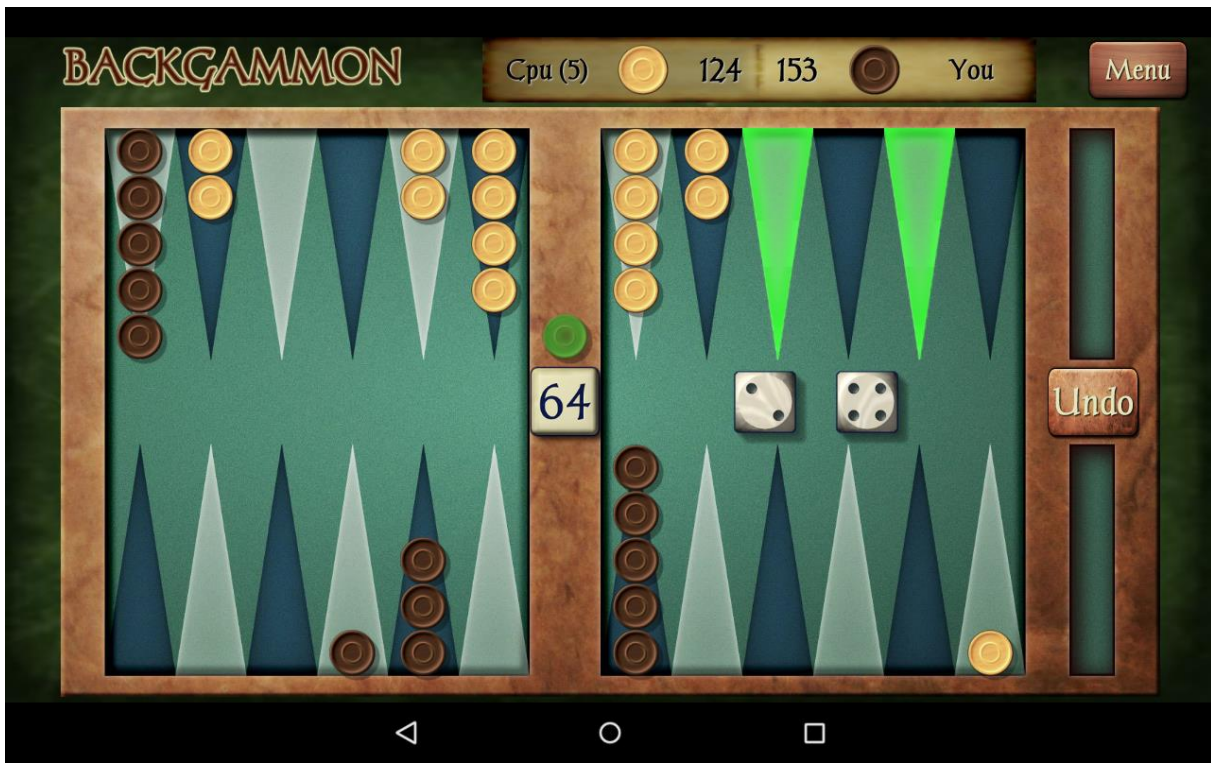
Εικόνα 2.17 Texas Hold'em

2.5.6 Το τάβλι

Το τάβλι είναι παιχνίδι στρατηγικής αλλά και τύχης. Το ταμπλό του έχει συνολικά 24 θέσεις. Κάθε παίκτης κατέχει 15 πούλια, που κινούνται σε ειδικό ταμπλό σύμφωνα με τα αποτελέσματα δύο ζαριών. Σκοπός του κάθε παίκτη είναι να μαζέψει πρώτος όλα τα πούλια από το ταμπλό. Ο παίκτης που ολοκληρώνει πρώτος το μάζεμα είναι και ο νικητής. Τα τρία βασικά παιχνίδια που παίζονται στο τάβλι είναι οι Πόρτες, το Πλακωτό και το Φεύγα.

Οι ζαριές του αποτελούνται από τυχαίες ή ψεύδο-τυχαίες γεννήτριες αριθμών. Με κάθε ζαριά, οι παίκτες πρέπει να επιλέξουν ανάμεσα στις επιλογές που τους δίνει ο υπολογιστής για την μετακίνηση των πούλιων και να αποφύγουν τυχόν κινήσεις του αντιπάλου του θα τους φέρουν σε δύσκολη θέση.

Το λογισμικό της ηλεκτρονικής προσομοίωσης του έχει αναπτυχθεί με τέτοιο τρόπο ώστε το παιχνίδι να καταστεί εύκολο μεταξύ ανθρώπων μέσω Internet όπως και για να νικήσει παίκτες παγκόσμιας κλάσης (Εικόνα 2.18). Το online παιχνίδι για το τάβλι ξεκίνησε το 1992 με τον First Internet Backgammon Server (FIBS). Ο συγκεκριμένος είναι ο μακροβιότερος server τάβλι στο Internet [19].



Εικόνα 2.18 Το τάβλι online

2.6 Multiplayer Computer Games

Τα παιχνίδια πολλών χρηστών ή αλλιώς τα multiplayer computer games είναι ηλεκτρονικά παιχνίδια όπου δύο ή περισσότεροι παίκτες, μπορούν να παίξουν ταυτόχρονα στον ίδιο χρόνο και στο ίδιο περιβάλλον με άλλους παίκτες και οι ενέργειες του ενός παίκτη επηρεάζουν και τον άλλον αλλά και το συνολικό αποτέλεσμα. Μπορούν να παιχτούν “τοπικά” ή μέσω δικτύου.

Όταν αναφέρεται ότι οι παίκτες παίζουν “τοπικά”, σημαίνει ότι όλοι οι παίκτες είναι συνδεδεμένοι στο ίδιο δίκτυο, δηλαδή οι υπολογιστές τους είναι συνδεδεμένοι στο ίδιο router (LAN) και δεν απαιτείται σύνδεση στο διαδίκτυο. Αυτός ο τρόπος δε χρησιμοποιείται ιδιαίτερα σήμερα, αλλά εξακολουθεί να χρησιμοποιείται σε ορισμένες περιπτώσεις, όπως σε ίντερνετ καφέ, ιδιωτικά τουρνουά ή απλά για φίλους που παίζουν στο ίδιο σπίτι. Από την άλλη, η δεύτερη επιλογή είναι η σύνδεση στο διαδίκτυο. Χρησιμοποιούνται τεχνολογίες δικτύωσης όπου όλοι οι παίκτες συνδέονται με έναν διακομιστή ή έναν κεντρικό υπολογιστή που είναι υπεύθυνος για τον συγχρονισμό τους και έτσι μπορούν να παίξουν το ίδιο παιχνίδι ταυτόχρονα. Τα περισσότερα multiplayer είδη παιχνιδιών, όπως παιχνίδια στρατηγικής, ρόλων, αγώνων κ.α. παρέχουν την επιλογή για πολλούς χρήστες.

Το συγκεκριμένο είδος παιχνιδιών είναι ευρέως δημοφιλές και αυτό γιατί ένας παίκτης βρίσκει πιο ελκυστικό και ανταγωνιστικό να παίζει με άλλους ανθρώπους αντί για την ύπαρξη τεχνητής νοημοσύνης. Η τεχνητή νοημοσύνη είναι ένας τομέας της βιομηχανίας των παιχνιδιών που χρήζει σημαντικής βελτίωσης ακόμα και σήμερα. Σε τέτοιου είδους παιχνίδια, όπου οι παίκτες στην ουσία παίζουν με τον υπολογιστή, αν η σχεδίασή του είναι κακή, είναι εύκολο ο παίκτης να το εκμεταλλευτεί

αυτό, να μάθει να αναγνωρίζει τις κινήσεις και να φτάσει στο σημείο να μπορεί να κερδίζει εύκολα. Αυτό είναι κάτι το οποίο δε συμβαίνει στα multiplayer παιχνίδια. Σε αυτά ο παίκτης μπορεί μεν να προβλέψει ορισμένες στρατηγικές του παιχνιδιού αλλά δε μπορεί να ξέρει τις κινήσεις του αντιπάλου του με ακρίβεια.

Οι παίκτες είναι σε θέση να αποφασίσουν για τις ενέργειές τους, να μάθουν από τα λάθη τους, να δοκιμάσουν διαφορετικές στρατηγικές και να λειτουργήσουν ομαδικά ή και ατομικά αν χρειαστεί. Για αυτό το λόγο, η τεχνητή νοημοσύνη δεν μπορεί να αντικαταστήσει αυτά τα παιχνίδια, μέχρις ότου τουλάχιστον φτάσει να έχει την ίδια δυναμική. Ένας ακόμη λόγος που αυτά τα παιχνίδια είναι τόσο δημοφιλή είναι γιατί υπάρχει η δυνατότητα εγγραφής των παικτών σε αυτά. Με αυτόν τον τρόπο, μπορούν φίλοι να παίξουν μαζί στην ίδια ομάδα, και άρα το παιχνίδι τους να γίνει ακόμη πιο διασκεδαστικό.

Επιπλέον, σε αυτά τα παιχνίδια υπάρχουν και συστήματα κατάταξης των παικτών (ranking systems), με βάση τη δημοτικότητα τους και τις ικανότητες και δεξιότητές τους. Αυτά τα συστήματα, παρακινούν τους παίκτες να ανταγωνιστούν άλλους παίκτες της ίδιας κλάσης (rank) με της δικής τους, από όλο τον κόσμο. Αυτό έχει ως αποτέλεσμα, οι παίκτες να επιθυμούν να φτάσουν στην υψηλότερη δυνατή βαθμολογία με βάση το επίπεδό τους. Τα πιο επιτυχημένα multiplayer παιχνίδια διοργανώνουν τουρνουά, όπου κορυφαίοι παίκτες και ομάδες, ανταγωνίζονται άλλους παίκτες, με σκοπό να κερδίσουν κάποιο βραβείο και αναγνώριση, ενθαρρύνοντας παράλληλα τους θαυμαστές τους να παίξουν για να φτάσουν στο δικό τους επίπεδο. Αυτό οδηγεί, στην ολοένα και μεγαλύτερη αύξηση παικτών που συμμετέχουν και στην μεγαλύτερη προσπάθεια των παλαιότερων για υψηλότερη κατάταξη.

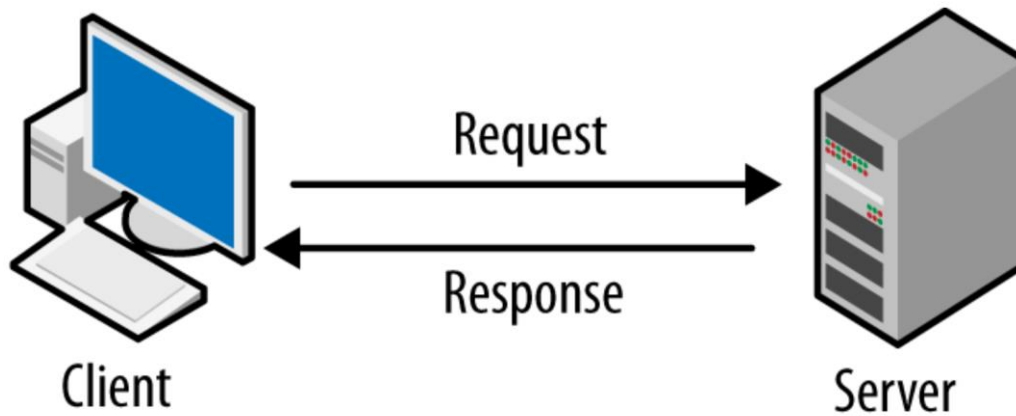
2.6.1 Game Networking

Οι αρχιτεκτονικές δικτύωσης που χρησιμοποιούνται στα συστήματα παιχνιδιών μπορούν να κατηγοριοποιηθούν σε δύο μεγάλες κατηγορίες, την αρχιτεκτονική πελάτη - εξυπηρετητή (client/server), και την peer-to-peer. Κοινός τόπος και των δύο αρχιτεκτονικών είναι οι καταναμημένες εφαρμογές και το πρόβλημα της ανακάλυψης [20].

Καταναμημένη εφαρμογή ονομάζεται εκείνη που περιέχει δύο ή περισσότερες λειτουργικές μονάδες σε διαφορετικούς υπολογιστές. Αυτές οι μονάδες αλληλεπιδρούν μεταξύ τους χρησιμοποιώντας το δίκτυο που τις συνδέει. Προκειμένου να χτιστεί μια τέτοια εφαρμογή πρέπει να είναι γνωστές κάποιες παράμετροι, όπως ο αριθμός των συμμετεχόντων, η κατανομή των λειτουργικών μονάδων στους διάφορους υπολογιστές του δικτύου και ο τρόπος που θα ανακαλύπτει κάποια μονάδα αυτή με την οποία θέλει να επικοινωνήσει. Το πρόβλημα της ανακάλυψης οριοθετείται από την ανάγκη των λειτουργικών μονάδων για επικοινωνία. Προϋπόθεση αποτελεί η γνώση για τον προορισμό των μηνυμάτων που πρέπει να σταλούν. Ο τρόπος ανάκτησης αυτής της γνώσης διαφοροποιείται ανάλογα με την αρχιτεκτονική που χρησιμοποιείται [20].

2.6.1.1 Client/Server

Σε ένα σύστημα client/server (Εικόνα 2.19) υπάρχουν δύο τύποι τερματικών, ο εξυπηρετητής (server) και ο πελάτης (client). Ο πρώτος, έχει την ευθύνη να παρέχει υπηρεσίες προς τους πελάτες, οι οποίοι δεν επικοινωνούν μεταξύ τους αλλά μόνο με τον εξυπηρετητή. Συνήθως, ο server παρέχει ολόκληρο το περιβάλλον του παιχνιδιού ή και ένα κομμάτι του. Μπορεί να χωρίζεται σε επιμέρους τμήματα τα οποία επικοινωνούν απευθείας μεταξύ τους για τον συγχρονισμό του παιχνιδιού.



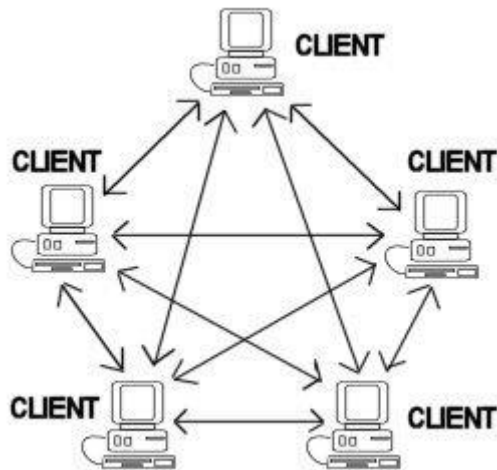
Εικόνα 2.19 Client - Server

Πιο αναλυτικά, κάθε client στέλνει και λαμβάνει ενημερώσεις μέσω ενός διακομιστή κατά τη διάρκεια του παιχνιδιού και διακομιστές επικοινωνούν μεταξύ τους μέσω peer-to-peer πρωτοκόλλων. Όταν ένας κεντρικός υπολογιστής λαμβάνει μια ενημέρωση από έναν από τους παίκτες του, πρέπει να ενημερώνει τους υπόλοιπους ενδιαφερόμενους παίκτες που είναι συνδεδεμένοι, καθώς επίσης και με τους υπόλοιπους κεντρικούς υπολογιστές.

Σε αντίθεση με την εξ ολοκλήρου διαχείριση από έναν κεντρικό υπολογιστή, μπορεί να γίνει κατανομή του φόρτου εργασίας για τον χειρισμό των πελατών με τη διαίρεση του σε επιμέρους διακομιστές. Με αυτόν τον τρόπο, το σύστημα μπορεί να υποστηρίξει ολοένα και αυξανόμενο αριθμό πελατών αρκεί να προστεθεί ένας ακόμα διακομιστής [21].

2.6.1.2 Peer-to-Peer

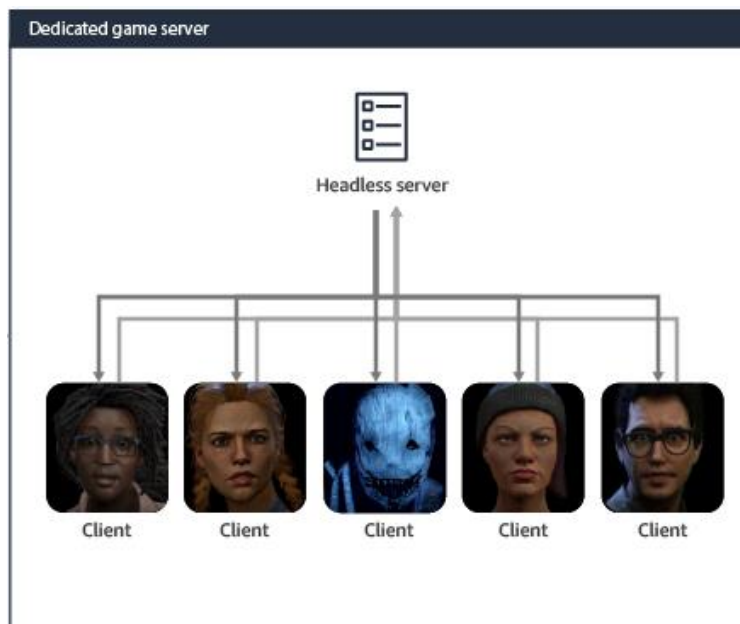
Σε αυτού του είδους την αρχιτεκτονική, ο κάθε παίκτης, στη θέση εργασίας του, μπορεί να επικοινωνήσει άμεσα με τον σταθμό εργασίας οποιουδήποτε άλλου παίκτη. Δεν ελέγχεται από έναν κεντρικό υπολογιστή, αλλά έχει την ευθύνη της σωστής εφαρμογής του παιχνιδιού και των κανόνων που το διέπουν. Τα πλεονεκτήματα της peer-to-peer (Εικόνα 2.20) είναι ότι δεν υπάρχει φόρτος εργασίας σε έναν κεντρικό υπολογιστή αλλά διανέμεται μεταξύ των clients. Έτσι, δίνεται η δυνατότητα μεγάλης επεκτασιμότητας με πολύ μικρό κόστος. Βασικό μειονέκτημα αυτής της αρχιτεκτονικής όμως, είναι ότι δεν υπάρχει απόλυτος έλεγχος από τον server [21].



Εικόνα 2.20 Peer to Peer

2.6.1.3 Dedicated Server

Σε αυτήν την αρχιτεκτονική, τον ρόλο του server τον αναλαμβάνει ένας από τους πελάτες (Εικόνα 2.21). Αυτό έχει ως αποτέλεσμα, να μην υπάρχει φόρτος εργασίας σε κεντρικό υπολογιστή της εκάστοτε εταιρείας κατασκευής του παιχνιδιού, αλλά σε έναν από τους παίκτες. Μεγάλο της μειονέκτημα είναι, ότι αν ο φόρτος εργασίας είναι μεγάλος, ή αποσυνδεθεί ο παίκτης που έχει τον ρόλο του server, τότε ο υπολογιστής του δε μπορεί να τον διαχειριστεί και δημιουργούνται προβλήματα σε όλους του παίκτες.

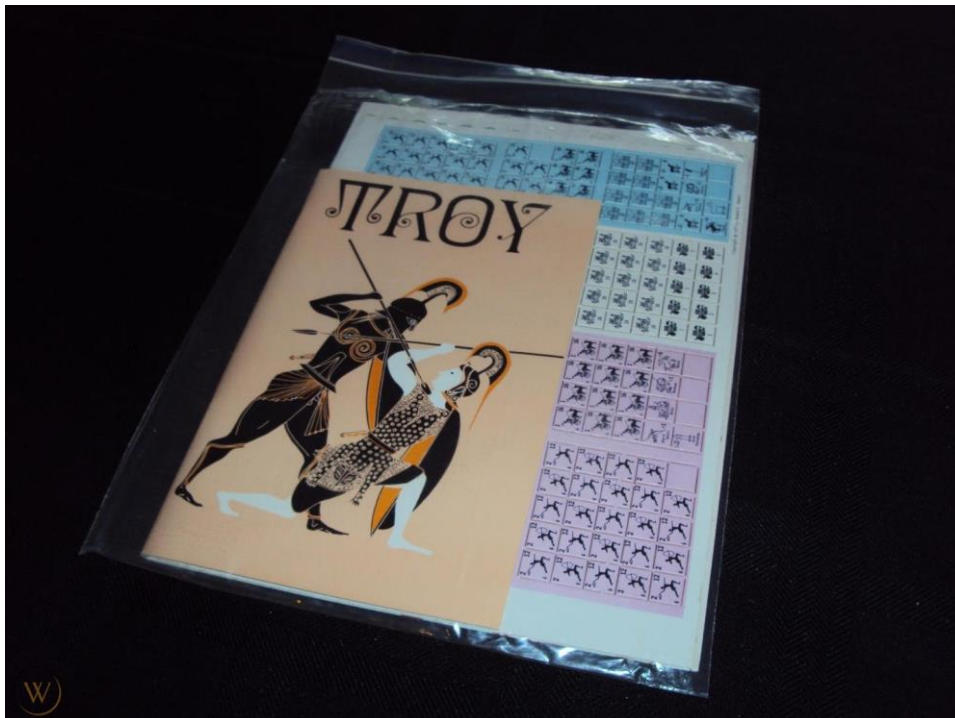


Εικόνα 2.21 Dedicated Server

Κεφάλαιο 3ο: Παιχνίδια με θέμα την ελληνική μυθολογία

Ως ελληνική μυθολογία ορίζεται η αφήγηση των μυθικών ιστορημάτων που δημιουργήθηκαν από τους Αρχαίους Έλληνες και αφορούσαν τους θεούς του Ολύμπου και τους ήρωες, τη φύση του κόσμου και τις τελετουργικές πρακτικές της λατρείας τους. Συνίσταται σε μια πλούσια συλλογή αφηγημάτων που αναφέρονται στην προέλευση του κόσμου και εξιστορούν τη ζωή και τις περιπέτειες μιας ευρείας ποικιλίας θεών, ηρώων, ηρωίδων και άλλων μυθολογικών πλασμάτων [22].

Η ελληνική μυθολογία έχει ουσιώδη επιρροή στο δυτικό πολιτισμό γενικότερα, στη φιλοσοφία του, στην ιστορία του, στην πολιτική του, στις τέχνες και στη λογοτεχνία του και θεωρείται βασικό στοιχείο της δυτικής κληρονομιάς. Αποτελεί δε, μέρος της εκπαίδευσης, από νεαρή ηλικία, σε πολλές δυτικές χώρες με σημαντική ελληνομάθεια [22]



Εικόνα 3.1 Troy Board Game

Αυτή της η επιρροή, είχε φυσικό επακόλουθο την έμπνευση για δημιουργία παιχνιδιών που η ιστορία τους είναι βασισμένη στους μυθικούς ήρωες. Τέτοιου είδους παιχνίδια προκαλούν το ενδιαφέρον του κοινού και πολλοί είναι εκείνοι οι παίκτες που σπεύδουν να προμηθευτούν παιχνίδια που από τη μια πλευρά είναι εκπαιδευτικά, μιας και παρέχουν γνώσεις της ελληνικής μυθολογίας, και από την άλλη, κατακλύζονται από μάχες μυθικών ηρώων και τεράτων.

Η πρώτη απόπειρα δημιουργίας παιχνιδιών αυτού του θέματος, ήταν στην κατηγορία των επιτραπέζιων παιχνιδιών. Είναι εμπνευσμένα από εντυπωσιακά γεγονότα της μυθολογίας και μυθικούς ήρωες. Αποτελούνται από ένα ταμπλό με διάφορες μινιατούρες ή φιγούρες, ζάρια και κάποιες φορές από κάρτες ή και νομίσματα.



Εικόνα 3.2 Age of Mythology: The Board Game

Κάποια από τα πιο γνωστά επιτραπέζια παιχνίδια με χρονολογική σειρά καταγράφονται στον πίνακα [23] (Πίνακας 3.1)

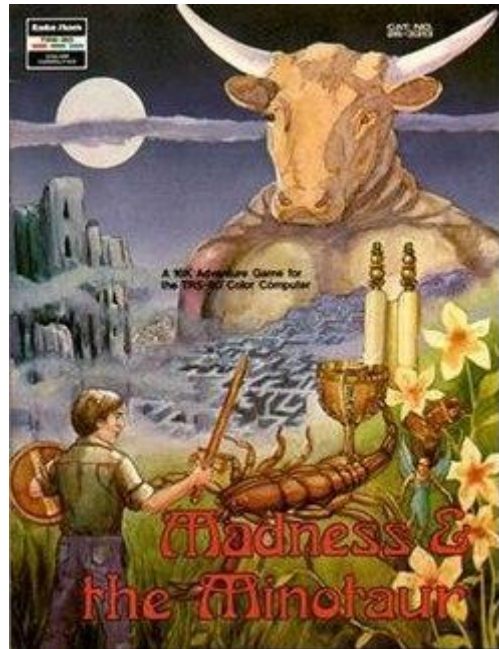
Πίνακας 3.1 Επιτραπέζια παιχνίδια ελληνικής μυθολογίας

Τίτλος	Χρονολογία	Διάρκεια (min)	Αριθμός Παικτών
Troy (Εικόνα 3.1)	1977	90	2
Iliad: The Siege of Troy	1978	180	2
Odyssey: The Gods Clash	1980	120-240	2
Mythology: A Game of Adventure in the Age of Heroes	1980	120-240	2-10
Colossal Arena	1997	40-60	2-5
Hera and Zeus	2000	30	2
Battle Line	2000	30	2
Ulysses	2001	45	3-5
Odysseus	2001	45	3-6
Hellas	2002	45-60	2

Age of Mythology: The Board Game (Εικόνα 3.2)	2003	120	2-4
Attika	2003	45-60	2-4
Santorini	2004	20	2-3
Risk: Godstorm	2004	120	2-5
Iliad	2006	45	2-6
Amyitis	2007	60-120	2-4
Battlefields of Olympus	2008	30	2
Deukalion	2008	60	2-4
Odyssey: Wrath of Poseidon	2015	30	2-5
Elysium	2015	60	2-4
Immortal	2017	30-90	2-4

3.1 Ηλεκτρονικά παιχνίδια βασισμένα στην ελληνική μυθολογία

Με την αυξανόμενη δημοτικότητα των επιτραπέζιων παιχνιδιών με θέμα την ελληνική μυθολογία, καθώς και με το ενδιαφέρον που προκάλεσαν στο κοινό, δε γινόταν να μη μεταφερθούν τόσο και στους υπολογιστές όσο και σε εξειδικευμένες κονσόλες. Από τους πρώτους μόλις προσωπικούς υπολογιστές άρχισαν να εμφανίζονται προσομοιώσεις επιτραπέζιων παιχνιδιών που αφορούσαν μυθικούς ήρωες. Μέχρι και σήμερα βγαίνουν τίτλοι που συναρπάζουν εκατομμύρια παίκτες ανά τον κόσμο (Πίνακας 3.2).

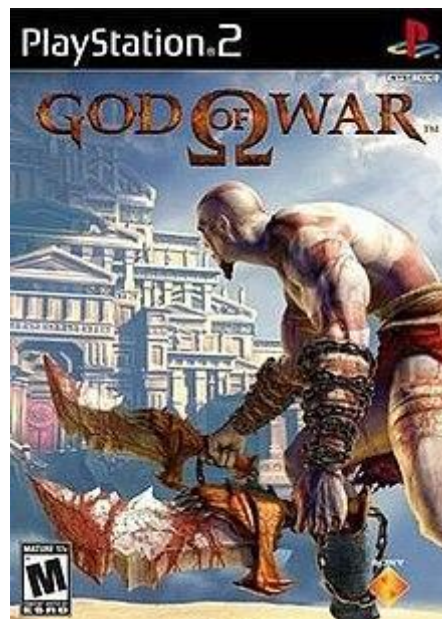


Εικόνα 3.3 Madness and the Minotaur

Πίνακας 3.2 Ηλεκτρονικά παιχνίδια ελληνικής μυθολογίας

Τίτλος	Χρονολογία	Είδος	Πλατφόρμα	Εταιρεία
Labyrinth	1980	Adventure	TRS-80	Med System Software
Ulysses and the Golden Fleece	1981	Adventure	Apple II	Sierra On-Line
Madness and the Minotaur (Εικόνα 3.3)	1981	Interactive Fiction	TRS-80	Radio Shark
Perseus and Andromeda	1983	Interactive Fiction	Atari 8-bit, Commodore 64	Digital Fantasia
Hercules	1984	Platformer	Commodore 64	Interdisc
Trojan	1986	Hack and Slash	DOS, NES	Capcom
The Battle of Olympus	1988	Action-adventure	NES, Game Boy	Infinity, Radical Entertainment
Phelios	1989	Vertical - scrolling shooter	Namco System 2	Namco
Populous II: Trials of the Olympian Gods	1991	Strategy	Amiga	Electronic Arts

Wrath of the Gods	1994	Adventure	Windows, MAC OS	Luminaria
Disney Hercules	1997	Platformer	Game Boy, Windows, PS2	Virgin Interactive, Sony, Disney Interactive, THQ
Zeus: Master of Olympus	2000	Strategy	Windows	Sierra Entertainment
Age of Mythology	2002	Strategy	Windows, OSX	Ensemble Studios
God of War (Εικόνα 3.4)	2005	Hack and Slash	PS2	Santa Monica Studio
Rise of the Argonauts	2008	ARPG	Windows, PS3, Xbox360	Liquid Entertainment
Assassin's Creed Odyssey	2018	ARPG	Windows, PS4, Xbox One, Nintendo Switch	Ubisoft
Hades	2020	Roguelike	macOS, Windows, Nintendo Switch	Supergiant Games



Εικόνα 3.4 God of War

3.2 Παιχνίδια περιηγητή ιστού βασισμένα στην ελληνική μυθολογία

Με την εμφάνιση του διαδικτύου (κατά τη δεκαετία του 1990), αναπτύχθηκαν προγράμματα που έδιναν την δυνατότητα περιήγησης στον παγκόσμιο ιστό (world wide web), τα λεγόμενα browsers (περιηγητές ιστού). Με την τεχνολογική εξέλιξη των browsers και την πάροδο του χρόνου, άρχισαν να κατασκευάζονται και να αναπτύσσονται διαδικτυακά παιχνίδια (Πίνακας 3.3). Αυτό πρόσφερε μεγαλύτερη δυνατότητα επιλογής τους από το κοινό καθώς ήταν εύκολα προσβάσιμα από κάθε υπολογιστή με ένα μόνο κλικ.

Μερικές από τις κύριες τεχνολογίες που χρησιμοποιούνται μέχρι και σήμερα για την ανάπτυξη παιχνιδιών περιηγητή ιστού είναι:

- HTML
- CSS
- JavaScript

Τα παιχνίδια που επέτρεπαν να παίζουν πολλοί χρήστες ταυτόχρονα (multiplayer games), χρησιμοποιούσαν κάποιον διακομιστή. Συνήθως αυτός βασιζόταν στις τεχνολογίες:

- Java
- PHP
- ASP .NET
- Flash

Κυρίως στο παρελθόν, αλλά και σήμερα, για να παίξει κανείς ένα πιο σύνθετο παιχνίδι περιηγητή ιστού, απαιτούνταν η χρήση κάποιων “πρόσθετων” (Ενότητα 4.3).

Πίνακας 3.3 Browser παιχνίδια βασισμένα στην ελληνική μυθολογία

Τίτλος	Χρονολογία	Είδος	Εταιρεία
Ikariam	2008	Strategy	GameForge
Achilles	2008	Hack and Slash	Armor Games
Mythopolis	2009	Strategy	Koram Game
Grepolis	2009	Strategy	InnoGames
Alexander - Dawn of an Empire	2010	Strategy	Armor Games
Myths and Mortals	2011	Fantasy	Golf Games

Το Ikariam, ένα από τα γνωστότερα παιχνίδια φυλλομετρητή, είναι ένα παιχνίδι πολλαπλών χρηστών που παίζεται στο διαδίκτυο με τον όρο Massive Multiple Online Game (MMOG), γραμμένο στην γλώσσα προγραμματισμού PHP για τη διαχείριση των servers. Διαδραματίζεται στην εποχή της κλασικής Ελλάδας, σε ένα αρχιπέλαγος, με σκοπό οι παίκτες να χτίσουν τις πόλεις τους, να μαζέψουν υλικά αγαθά και να συνθλίψουν τους αντιπάλους τους. Το κύριο χαρακτηριστικό του παιχνιδιού είναι ότι είναι εύκολα προσβάσιμο για μικρούς και μεγάλους με αποτέλεσμα να έχει ένα τεράστιο κοινό. Το

παιχνίδι αυτό αποτελείται από πολλούς servers, εκ των οποίων κάποιοι είναι καθαρά ελληνικοί (Εικόνα 3.5).



Εικόνα 3.5 Ikarium

Παρόμοιο παιχνίδι είναι και το Grepolis. Το συγκεκριμένο εστιάζει στη συνεργασία και τη στρατηγική. Σκοπός του παιχνιδιού είναι να μετατρέψεις μια μικρή πόλη σε τεράστια μητρόπολη. Ο παίκτης, φτιάχνει στρατό και στόλο για να κερδίζει στις μυθικές μάχες και παράλληλα, συνάπτει συμμαχίες για την εξόντωση του κοινού εχθρού. Όλα αυτά, παίρνοντας την εύνοια των θεών.

Κεφάλαιο 4ο: Ανάλυση και επιλογή τεχνολογιών

Για να αναλυθούν οι τεχνολογίες που χρησιμοποιούνται σε παιχνίδια περιηγητή ιστού, πρέπει πρώτα να οριστεί η έννοια του περιηγητή ιστού (web browser). Είναι ένα λογισμικό που επιτρέπει στον χρήστη να προβάλλει και να αλληλεπιδρά με κείμενα, εικόνες, βίντεο, μουσική, παιχνίδια και άλλες πληροφορίες που συνήθως αναρτώνται σε μια ιστοσελίδα ενός ιστοτόπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Με τον browser επιτρέπεται στον χρήστη η γρήγορη πρόσβαση σε πληροφορίες που βρίσκονται σε διάφορες ιστοσελίδες και ιστότοπους εναλλάσσοντας τις ιστοσελίδες μέσω των υπερσυνδέσμων.

Οι περιηγητές ιστού χρησιμοποιούν τη γλώσσα μορφοποίησης HTML για την εμφάνιση στοιχείων (elements) [24]. Επιπρόσθετα, χρησιμοποιείται η CSS για τη διαμόρφωση του “στυλ” αυτών των στοιχείων και η JavaScript για να δημιουργήσει διεπαφές χρήστη και δυναμικές ιστοσελίδες γενικότερα. Για κάθε browser διατίθενται και αρκετά πρόσθετα στοιχεία “add-ons” ή “plug-ins”, με στόχο την επαύξηση των δυνατοτήτων τους, τη βελτίωση της χρηστικότητάς τους και την προστασία του χρήστη σε θέματα ασφαλείας. Οι περισσότεροι χρησιμοποιούμενοι browsers είναι οι:

- Google Chrome
- Windows Internet Explorer
- Microsoft Edge
- Mozilla Firefox
- Apple Safari
- Opera

Τα παιχνίδια περιηγητή ιστού είναι εύκολα προσβάσιμα από όλους τους χρήστες, καθώς μπορούν να εκτελεστούν σε πολλούς διαφορετικούς τύπους υπολογιστών. Μοναδική προϋπόθεση για τη χρήση τους είναι η σύνδεση στο διαδίκτυο. Δημιουργούν την άνεση του να παίζεις χωρίς να χρειάζεται να “κατεβάσεις” το παιχνίδι στον υπολογιστή ή το κινητό σου, κάτι το οποίο τα κάνει να πλεονεκτούν από τα υπόλοιπα είδη παιχνιδιών. Υπάρχουν δύο σύνολα διαδικασιών που ακολουθούνται κατά τη δημιουργία τέτοιου είδους παιχνιδιών: το frontend και το backend. Καθένα από αυτά τα δύο σύνολα συγκαταλέγεται από κάποιες τεχνολογίες οι οποίες θα αναλυθούν.

4.1 Τεχνολογίες Front End

Η κατασκευή των ιστοσελίδων προϋποθέτει την χρήση κάποιων βασικών τεχνολογιών (HTML, CSS, JS). Οι πρώτοι περιηγητές ιστού χρησιμοποιούσαν στατικές σελίδες και υπερσυνδέσμους μέσω HTML και CSS. Γρήγορα δημιουργήθηκε η ανάγκη για κάτι επιπλέον και έτσι, αναπτύχθηκε μια απλή γλώσσα προγραμματισμού για να αυξήσει τη δυναμικότητα των ιστοσελίδων (JavaScript). Με την ανάπτυξη της JS, εμφανίστηκαν και διάφορα frameworks που επικεντρώνονταν στην ανάπτυξη ιστοσελίδων αλλά και παιχνιδιών.

4.1.1 HTML

Η HTML (Hyper Text Markup Language), είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Γράφεται υπό μορφή στοιχείων HTML, τα οποία αποτελούνται από ετικέτες (tags), για παράδειγμα `<html>`, μέσα στο περιεχόμενο της ιστοσελίδας. Συνήθως λειτουργούν ανά ζεύγη (π.χ. `<body></body>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (opening and closing brackets). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες και άλλα HTML στοιχεία.

Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να παρουσιάσει το περιεχόμενο της σελίδας. Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα.

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990. Τον ίδιο χρόνο, ο Μπέρνερς Λι και ο μηχανικός συστημάτων πληροφορικής του CERN Robert Cailliau συνεργάστηκαν σε μια κοινή προσπάθεια εύρεσης χρηματοδότησης, αλλά το έργο δεν υιοθετήθηκε ποτέ επίσημα από το CERN.

Το 1995, δημοσιεύτηκε η HTML 2.0, προσθέτοντας στην γλώσσα τις εξής δυνατότητες: ανέβασμα αρχείου από φόρμα, πίνακες, χάρτες εικόνων, διεθνοποίηση και τοπικοποίηση (Localization). Το 1997, αρχικά δημοσιεύτηκε ως σύσταση η έκδοση HTML 3.2 από το W3C. Σε αυτήν αφαιρέθηκαν εντελώς οι μαθηματικοί τύποι και οι υπερκαλύψεις που προέρχονταν από επεκτάσεις του προτύπου από τρίτους φορείς. Στη συνέχεια όμως, δημοσιεύθηκε και η HTML 4.0. Υιοθέτησε πολλά στοιχεία και ιδιότητες που προηγουμένως χρησιμοποιούνταν μόνο από συγκεκριμένους browsers, αλλά ταυτόχρονα προσέβλεπε στην σταδιακή κατάργηση των δυνατοτήτων οπτικής διαμόρφωσης των ξεπερασμένων πια περιηγητών ιστού, προτείνοντας τα CSS στη θέση τους.

Το 2008 δημοσιεύτηκε η τελευταία μέχρι στιγμής έκδοσή της, η HTML5 [25]. Η συγκεκριμένη, δίνει τη δυνατότητα στους προγραμματιστές να δημιουργήσουν mobile web sites τα οποία έχουν ταυτόχρονα πολλές από τις ιδιότητες των εφαρμογών που συναντώνται στις διάφορες πλατφόρμες κινητών, ανοίγοντας έτσι νέους ορίζοντες στην παρουσία του διαδικτύου στην όλη εμπειρία με τα κινητά τηλέφωνα [26]. Σήμερα, με την εμφάνιση της HTML5, η εμπειρία στις εφαρμογές των κινητών έχει αναπτυχθεί κατά πολύ. Αυτό συμβαίνει διότι η εφαρμογή εγκαθίστανται πάνω στο λειτουργικό του κινητού και αξιοποιεί στο έπακρο τις τεχνικές δυνατότητές του.

Για την ανάπτυξη και τον χειρισμό πολυμέσων και γραφικού περιεχομένου προστέθηκαν καινούργια στοιχεία όπως:

- <video>
- <audio>
- <canvas>, κ.α

Επιπλέον, εισάχθηκε η υποστήριξη των SVG (Scalable Vector Graphics) που δίνει την δυνατότητα αλλαγής του μεγέθους και του χρώματος εικονιδίων.

Για τον εμπλουτισμό του σημασιολογικού περιεχομένου των εγγράφων, προστίθενται νέα στοιχεία δομής σελίδας όπως:

- <main>
- <section>
- <article>

- <header>
- <footer>, κ.α

Η σήμανση HTML αποτελείται από μερικά βασικά συστατικά, συμπεριλαμβανομένων των στοιχείων (και των ιδιοτήτων τους), τους βασισμένους σε χαρακτήρες τύπους δεδομένων, τις αναφορές χαρακτήρων και τις αναφορές οντοτήτων. Ένα ξεχωριστό σημαντικό συστατικό είναι η δήλωση τύπου εγγράφου (document type declaration), η οποία ορίζει στο πρόγραμμα περιήγησης (browser) τον τρόπο εμφάνισης της σελίδας.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

Εικόνα 4.1 Πρόγραμμα Hello World σε HTML

Το κείμενο (Εικόνα 4.1) ανάμεσα στο <html> και το </html> περιγράφει την ιστοσελίδα, και το κείμενο μεταξύ του <body> και του </body> είναι το ορατό μέρος της. Το σημασμένο κείμενο <title>Hello HTML</title> καθορίζει τον τίτλο που θα εμφανίζεται στην μπάρα τίτλου του browser.

Οι περισσότερες ιδιότητες των στοιχείων είναι ζεύγη ονομάτων και τιμών, τα οποία διαχωρίζονται με ένα «=» και γράφονται μέσα στην ετικέτα εκκίνησης ενός στοιχείου, μετά το όνομα του στοιχείου (Εικόνα 4.2). Υπάρχουν και μερικές κοινές ιδιότητες οι οποίες εμφανίζονται σε πολλά στοιχεία:

- ❑ Η ιδιότητα id παρέχει ένα αναγνωριστικό για ένα στοιχείο το οποίο είναι μοναδικό σε ολόκληρο το έγγραφο.
- ❑ Η ιδιότητα class παρέχει τη δυνατότητα ταξινόμησης παρόμοιων αντικειμένων στην ίδια κλάση. Μπορεί να χρησιμοποιηθεί για να αποδώσει κάποια σημασία στο στοιχείο, ή για σκοπούς εμφάνισης.
- ❑ Η ιδιότητα style εφαρμόζει στυλ εμφάνισης σε συγκεκριμένα στοιχεία. Θεωρείται καλύτερη τακτική να χρησιμοποιούνται οι ιδιότητες id ή class ώστε να επιλέγεται το στοιχείο μέσα σε ένα CSS, αλλά μερικές φορές μπορεί να είναι πιο απλό να ανατεθούν style κατευθείαν στο στοιχείο.
- ❑ Η ιδιότητα title προσθέτει μια εξήγηση στο στοιχείο στο οποίο εφαρμόζεται. Στους περισσότερους φυλλομετρητές αυτή η ιδιότητα εμφανίζεται ως αναδυόμενο παράθυρο βοήθειας (tooltip).
- ❑ Η ιδιότητα lang ταυτοποιεί την φυσική γλώσσα των περιεχομένων του στοιχείου, η οποία μπορεί να είναι διαφορετική από το υπόλοιπο έγγραφο.

```

<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  <span style="color: red;" class="myClass" id="text1">
    Κείμενο
  </span>
</body>
</html>

```

Εικόνα 4.2 Ιδιότητες στοιχείου HTML

4.1.2 CSS

Η CSS προτάθηκε για πρώτη φορά από τον Håkon Wium Lie στις 10 Οκτωβρίου 1994. Εκείνη την εποχή, ο Lie συνεργαζόταν με τον Tim Berners-Lee στο CERN. Αρκετές άλλες γλώσσες φύλλων στυλ για τον Ιστό προτάθηκαν περίπου την ίδια στιγμή, και οι συζητήσεις σχετικά με τις δημόσιες λίστες αλληλογραφίας και εντός της Κοινοπραξίας World Wide Web οδήγησαν στην κυκλοφορία της πρώτης σύστασης W3C CSS (CSS1) το 1996 [27].

Η CSS (Cascading Style Sheets) είναι μια τεχνολογία που ανήκει στην κατηγορία γλωσσών φύλλων στυλ που χρησιμοποιείται για να ελέγχει την εμφάνιση ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Προορίζεται για την ανάπτυξη του στυλ μιας ιστοσελίδας και γενικότερα ενός ιστότοπου, δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και περισσότερες δυνατότητες σε σχέση με την HTML. Ουσιαστικά, η CSS συμπληρώνει την HTML με τις επιλογές μορφοποίησης που παρέχει και απλουστεύει τις πληροφορίες των εγγράφων της.

Η σύνταξη που χρησιμοποιεί αποτελείται από λέξεις κλειδιά που καθορίζουν τα ονόματα διαφόρων ιδιοτήτων στυλ (style properties). Ένα αρχείο CSS προέρχεται από μια λίστα κανόνων. Κάθε κανόνας ή σύνολο κανόνων αποτελείται από έναν ή περισσότερους επιλογείς (selectors) και ένα μπλοκ δήλωσης.

Οι επιλογείς δηλώνουν το τμήμα της σήμανσης στο οποίο εφαρμόζεται ένα στυλ, αντιστοιχίζοντας ετικέτες και χαρακτηριστικά στην ίδια τη σήμανση. Έτσι, οι επιλογείς διακρίνονται σε:

- Όλα τα στοιχεία ενός συγκεκριμένου τύπου (π.χ κεφαλίδα h2)
- Στοιχεία που χαρακτηρίζονται από τις ιδιότητες: “id”, “class”
- Στοιχεία που βασίζονται στο πώς τοποθετούνται σε σχέση με άλλα στοιχεία.

Οι κλάσεις (CSS classes) και τα ids είναι ευαίσθητες ως προς τα πεζά και κεφαλαία γράμματα όπως στις περισσότερες γλώσσες προγραμματισμού. Μπορούν να περιλαμβάνουν αλφαριθμητικούς χαρακτήρες, παύλες και κάτω παύλες. Μια κλάση μπορεί να χρησιμοποιηθεί σε περισσότερα από ένα στοιχεία σε αντίθεση από το id που μπορεί να εφαρμοστεί μόνο σε ένα στοιχείο.

Το μπλοκ δήλωσης ή αλλιώς “σώμα” αποτελείται από μία λίστα δηλώσεων με αγκύλες. Κάθε δήλωση αποτελείται από μια ιδιότητα, μια τιμή και ενδιάμεσα μια άνω και κάτω τελεία (:). Εάν υπάρχουν πολλές

δηλώσεις σε ένα μπλοκ, τότε πρέπει να εισαχθεί ένα ελληνικό ερωτηματικό, το οποίο θα διαχωρίζει κάθε δήλωση.

Οι ιδιότητες καθορίζονται στο πρότυπο CSS (Εικόνα 4.3). Ορισμένες από αυτές μπορούν να επηρεάσουν οποιονδήποτε τύπο στοιχείου και άλλες ισχύουν μόνο για συγκεκριμένες ομάδες στοιχείων. Κάθε ιδιότητα έχει ένα σύνολο πιθανών τιμών. Αυτές μπορεί να είναι λέξεις-κλειδιά (π.χ 'center') ή αριθμητικές τιμές. Οι τιμές χρώματος μπορούν να χαρακτηριστούν με λέξεις-κλειδιά (π.χ 'red'), καθώς και δεκαεξαδικές τιμές (π.χ #ff00ff).

```
3
4 #text1 {
5     position: absolute;
6     color: #ff00aa;
7     display: block;
8 }
9
10 .myClass {
11     font-size: 14px;
12 }
13
```

Εικόνα 4.3 Επιλογείς id και κλάση

Ενα κύριο χαρακτηριστικό της CSS είναι η κληρονομικότητα (inheritance), η οποία εμφανίζεται σχεδόν σε όλες τις γλώσσες προγραμματισμού. Είναι ο μηχανισμός με τον οποίο οι ιδιότητες εφαρμόζονται όχι μόνο σε ένα καθορισμένο στοιχείο, αλλά και στους απογόνους του. Βασίζεται στην ιεραρχία των στοιχείων της HTML σε μια σελίδα που βασίζεται στην ένθεση. Τα στοιχεία-παιδιά κληρονομούν κάποιες τιμές ιδιοτήτων (π.χ χρώμα, γραμματοσειρά, ορατότητα, κ.α) των προγόνων που τα περικλείουν. Η κληρονομικότητα μπορεί να χρησιμοποιηθεί για να αποφευχθεί η δήλωση ορισμένων ιδιοτήτων κατ' επανάληψη, με αποτέλεσμα να υπάρχουν πολύ λιγότερες γραμμές.

4.1.3 JavaScript

Η HTML μαζί με την CSS μπορούν να δημιουργήσουν πανέμορφες στατικές ιστοσελίδες, με την χρήση της JavaScript όμως, αυτό μπορεί να αλλάξει. Πρόκειται για μια διερμηνευμένη γλώσσα προγραμματισμού (Interpreted Programming language) που είναι επηρεασμένη από γλώσσα C. Πολλοί μπερδεύονται από την ονομασία και θεωρούν ότι είναι το ίδιο με την γλώσσα προγραμματισμού Java.

Η JS είναι η μοναδική γλώσσα προγραμματισμού που μπορούν να “τρέξουν” οι περιηγητές ιστού. Χρησιμοποιείται για να δώσει διάδραση στις ιστοσελίδες και τις εφαρμογές στο web. Όλοι οι φυλλομετρητές έχουν μία μηχανή που εκτελεί την JavaScript.

Χρησιμοποιείται κυρίως σε φυλλομετρητές αλλά πλέον εντάχθηκε και σε άλλες πλατφόρμες. Βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης (δηλαδή θεωρεί τις συναρτήσεις ως αντικείμενα). Υποστηρίζει αντικειμενοστραφές, προστατικό και συναρτησιακό στυλ προγραμματισμού.

Δημιουργήθηκε αρχικά, σε μόλις 10 ημέρες, από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, η Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. Η εταιρεία πίσω από την JavaScript ήθελε να χρησιμοποιήσει την λέξη java στο όνομα της γλώσσας της, για να

προσελκύσει τους προγραμματιστές. LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε beta έκδοση με το πρόγραμμα περιήγησης στο Web, Netscape Navigator, τον Σεπτέμβριο του 1995.

Απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφαρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. JScript πρόσθεσε νέες μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, οι οποίες βασίστηκαν στην java.util.Date τάξη της Java. Η JScript περιλήφθηκε στο πρόγραμμα Internet Explorer εκδοχή 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996.

Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International (μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript.

Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές. Με τη χρήση της τεχνολογίας AJAX (Asynchronous Javascript And Xml), η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web [28].

Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες (π.χ. server-side).

Ο κώδικας JavaScript μιας σελίδας μπορεί να εισαχθεί μέσω των tags `<script type="text/javascript"></script>`. Το οποίο μπαίνει συνήθως στο τέλος του head tag ή στο αμέσως μετά το κλείσιμο του body. Η χρήση του ερωτηματικού μετά από κάθε εντολή δεν είναι υποχρεωτική, αλλά συνιστάται.

```

<!DOCTYPE html>
<html>
<head>
  <title></title>
  <script>
    alert('Όλα πήγαν καλά!');
    var x = 5;
    console.log(x);
  </script>
</head>
<body>
  <span style="color: red;" class="myClass" id="text1">
    Κείμενο
  </span>
</body>
</html>

```

Εικόνα 4.4 JavaScript σε HTML έγγραφο

Για να αναλυθεί με μεγαλύτερη λεπτομέρεια η JavaScript δεν πρέπει να παραληφθεί και η αναφορά στο DOM (Document Object Model). Το DOM είναι μια δενδρική δομή που αποτελείται από κόμβους. Ο κάθε κόμβος είναι ένα αντικείμενο που αντιπροσωπεύει ένα στοιχείο του εγγράφου. Μέσω της JS δίνεται η δυνατότητα επεξεργασίας του DOM (Εικόνα 4.5). Κάθε αντικείμενο έχει event listeners που “ακούν” τα γεγονότα που συμβαίνουν στην ιστοσελίδα (π.χ φόρτωμα στοιχείων, κλικ ποντικιού ή πλήκτρων). Μόλις ενεργοποιηθεί ένα γεγονός, εκτελείται ο event handler που χειρίζεται την διαδικασία του γεγονότος.

```

<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  <button id="myButton">Button1</button>
</body>
<script>
  var buttonReference = document.getElementById('myButton');
  buttonReference.addEventListener('click', function() {
    alert('afk');
  })
</script>
</html>

```

Εικόνα 4.5 Χειρισμός DOM με τη χρήση JavaScript

Πρέπει να σημειωθεί ότι η JavaScript έχει εξελιχθεί από την αρχική της έκδοση σε μια γλώσσα προγραμματισμού που χρησιμοποιείται παγκοσμίως. Είναι η πιο εξεζητημένη γλώσσα στον κόσμο (Πίνακας 4.1) και συνεχίζει να αναπτύσσεται με ραγδαίους ρυθμούς [29].

Πίνακας 4.1 Pull Requests σε γλώσσες προγραμματισμού (GitHub)

Ranking	Programming Language	Percentage
1	JavaScript	18.72%
2	Python	16.48%
3	Java	11.55%
4	Go	8.13%
5	C++	7%
6	Ruby	6.95%
7	TypeScript	6.65%
8	PHP	5.58%
9	C#	3.67%
10	C	3.13%

4.1.4 Front End Frameworks

Η χρήση της JavaScript βοηθάει στην ανάπτυξη διαδραστικών εφαρμογών, αλλά για μεγάλα projects συνιστάται η χρήση ενός Framework. Framework είναι ένα σύνολο από βιβλιοθήκες και εργαλεία γραμμένα σε JS που απλοποιούν την διαδικασία ανάπτυξης λογισμικού, διότι δίνουν την δυνατότητα να χρησιμοποιηθεί έτοιμος κώδικας για διάφορες διαδικασίες χωρίς να ξαναγραφτεί από την αρχή (Code Reusability-DRY).

Επιπλέον, κάποια frameworks δίνουν την δυνατότητα οργάνωσης κώδικα σε επιμέρους ενότητες (modules). Αυτό έχει ως αποτέλεσμα να μπορεί να αναπτυχθεί λογισμικό μεγάλης κλίμακας. Ακόμα και οι πιο γνωστές ιστοσελίδες χρησιμοποιούν κάποιο framework (π.χ Google, Facebook, Twitter, κ.α).

Τα 3 πιο δημοφιλή front end frameworks μέχρι σήμερα, είναι η Angular, η React και η Vue [30]. Η React είναι ένα από τα πιο εύκολα framework για να μάθει κανείς, αναπτύχθηκε από το Facebook για να λυθούν αρκετά προβλήματα συντήρησης κώδικα, λόγω της συνεχούς προσθήκης λειτουργιών στην εφαρμογή. Ένα αξιοσημείωτο χαρακτηριστικό της είναι η χρήση ενός εικονικού μοντέλου αντικειμένου εγγράφου ή εικονικού DOM. Το React δημιουργεί μια προσωρινή μνήμη δομής δεδομένων, υπολογίζει τις προκύπτουσες διαφορές και, στη συνέχεια, ενημερώνει αποτελεσματικά το εμφανιζόμενο DOM του προγράμματος περιήγησης [31]. Αυτή η διαδικασία ονομάζεται συμφιλίωση (Reconciliation). Αυτό επιτρέπει στον προγραμματιστή να γράφει κώδικα σαν να αποδίδεται ολόκληρη η σελίδα σε κάθε αλλαγή, ενώ οι βιβλιοθήκες του React αποδίδουν μόνο υποσυστήματα που αλλάζουν πραγματικά, με αποτέλεσμα να βελτιώνεται κατά πολύ η απόδοση. Εξοικονομεί την προσπάθεια επανυπολογισμού του στυλ CSS, την διάταξη της σελίδας και γενικότερα την απόδοση (Rendering) για ολόκληρη τη σελίδα.

```

class ParentComponent extends React.Component {
  state = { color: 'green' };
  render() {
    return (
      <ChildComponent color={this.state.color} />
    );
  }
}

```

Εικόνα 4.6 Μια κλάση σε React

Η Angular έχει αρκετές διαφορές σε σχέση με την React, αλλά η σημαντικότερη είναι ότι χρησιμοποιεί ένα υπερσύνολο της JavaScript, το λεγόμενο TypeScript. Η TypeScript είναι μια γλώσσα προγραμματισμού που αναπτύχθηκε και συντηρήθηκε από τη Microsoft [32]. Πρόκειται για ένα αυστηρό συντακτικό υπερσύνολο της JavaScript και προσθέτει προαιρετική στατική πληκτρολόγηση στη γλώσσα. Η TypeScript έχει σχεδιαστεί για την ανάπτυξη μεγάλων εφαρμογών και μετασυμπίεση (Transcompile) σε JavaScript. Καθώς είναι ένα υπερσύνολο της JavaScript, τα υπάρχοντα προγράμματα JavaScript είναι επίσης έγκυρα προγράμματα TypeScript.

Μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών JavaScript για εκτέλεση από την πλευρά του πελάτη και του διακομιστή (όπως με το Node.js ή το Deno). Υπάρχουν πολλές διαθέσιμες επιλογές για μετασυμπίεση. Μπορεί να χρησιμοποιηθεί ο προεπιλεγμένος ελεγκτής TypeScript, ή ο μεταγλωττιστής Babel μπορεί να κληθεί να μετατρέψει TypeScript σε JavaScript.

Επίσημα ξεκίνησε το 2016, η Angular αναπτύχθηκε από την Google για να γεφυρώσει το χάσμα μεταξύ των αυξανόμενων απαιτήσεων τεχνολογίας και συμβατικών εννοιών που έδειξαν αποτελέσματα. Σε αντίθεση με την React, δίνει την δυνατότητα αμφίδρομης σύνδεσης δεδομένων (2 way data binding) [30]. Αυτό σημαίνει ότι υπάρχει συγχρονισμός σε πραγματικό χρόνο μεταξύ του μοντέλου και της προβολής, όπου οποιαδήποτε αλλαγή στο μοντέλο αντικατοπτρίζει άμεσα την προβολή και το αντίστροφο. Χρησιμοποιείται για πολλές και μεγάλες εφαρμογές, καθώς και για προοδευτικές εφαρμογές ιστού (PWA-Progressive Web Apps). Δίνει την δυνατότητα οργάνωσης κώδικα στα λεγόμενα modules και components (Εικόνα 4.7). Ξεχωρίζει την λογική από τα στοιχεία της σελίδας. Σε σύγκριση με την React, η Angular είναι αρκετά δύσκολη. Αν και υπάρχει διαθέσιμη αναρίθμητη τεκμηρίωση που είναι είτε πολύ περίπλοκη είτε προκαλεί σύγχυση.

```

import { Component, OnInit } from '@angular/core';
import { Hero } from '../hero';

@Component({
  selector: 'app-heroes',
  templateUrl: './heroes.component.html',
  styleUrls: ['./heroes.component.css']
})
export class HeroesComponent implements OnInit {
  hero: Hero = {
    id: 1,
    name: 'Windstorm'
  };

  constructor() { }

  ngOnInit() {
  }
}

<h2>{{hero.name | uppercase}} Details</h2>
<div><span>id: </span>{{hero.id}}</div>
<div>
  <label for='name'>Hero name: </label>
  <input id='name' [(ngModel)]="hero.name" placeholder="name">
</div>

```

Εικόνα 4.7 Ένα τυπικό Component της Angular

Ένα από τα πιο δημοφιλή front-end frameworks σήμερα, το Vue είναι σχετικά απλό. Καταργεί τις πολυπλοκότητες που αντιμετωπίζουν οι προγραμματιστές της Angular. Είναι μικρότερο σε μέγεθος και προσφέρει δύο μεγάλα πλεονεκτήματα - οπτικό DOM και βασισμένο σε στοιχεία (Εικόνα 4.8). Χρησιμοποιεί και αυτό αμφίδρομη σύνδεση δεδομένων.

Το Vue είναι ευέλικτο και βοηθάει με πολλών ειδών εργασίες. Από τη δημιουργία εφαρμογών ιστού και εφαρμογών για κινητές συσκευές έως τις προοδευτικές εφαρμογές ιστού, μπορεί να χειριστεί εύκολα και απλές και δυναμικές διαδικασίες [31], [33].

Αν και έχει σχεδιαστεί για να βελτιστοποιεί την απόδοση των εφαρμογών και να αντιμετωπίζει τις πολυπλοκότητες, δεν είναι ευρέως δημοφιλής στους γίγαντες της αγοράς. Ωστόσο, οι Alibaba, 9gag, Reuters, Xiaomi είναι χρήστες αυτού του framework.

```

<template>
  <div id="tuto">
    <button-clicked v-bind:initial-count="0"></button-clicked>
  </div>
</template>

<script>
Vue.component('button-clicked', {
  props: ['initialCount'],
  data: () => ({
    count: 0,
  }),
  template: '<button v-on:click="onClick">Clicked {{ count }} times</button>',
  computed: {
    countTimesTwo() {
      return this.count * 2;
    }
  },
  watch: {
    count(newValue, oldValue) {
      console.log(`The value of count is changed from ${oldValue} to ${newValue}.`);
    }
  },
  methods: {
    onClick() {
      this.count += 1;
    }
  },
  mounted() {
    this.count = this.initialCount;
  }
});

new Vue({
  el: '#tuto',
});
</script>

```

Εικόνα 4.8 Ένα component σε Vue

4.2 Τεχνολογίες Back End

Στον κόσμο των υπολογιστών, το "backend" αναφέρεται σε οποιοδήποτε μέρος ενός ιστότοπου ή ενός προγράμματος λογισμικού που οι χρήστες δεν βλέπουν. Σε αντίθεση με το frontend, το οποίο αναφέρεται στη διεπαφή χρήστη ενός προγράμματος ή ιστότοπου. Στην ορολογία προγραμματισμού, το backend είναι το "επίπεδο πρόσβασης δεδομένων", ενώ το frontend είναι το "επίπεδο παρουσίασης".

Οι περισσότεροι σύγχρονοι ιστότοποι είναι δυναμικοί, πράγμα που σημαίνει ότι το περιεχόμενο της ιστοσελίδας δημιουργείται εν κινήσει. Μια δυναμική σελίδα περιέχει ένα ή περισσότερα σενάρια που εκτελούνται στον διακομιστή ιστού κάθε φορά που γίνεται πρόσβαση στη σελίδα. Αυτά τα "σενάρια" (scripts) δημιουργούν το περιεχόμενο της σελίδας, το οποίο αποστέλλεται στο πρόγραμμα περιήγησης ιστού του χρήστη. Ό, τι συμβαίνει πριν από την εμφάνιση της σελίδας σε ένα πρόγραμμα περιήγησης ιστού, είναι μέρος του συστήματος υποστήριξης.

Παραδείγματα διαδικασιών υποστήριξης περιλαμβάνουν:

- επεξεργασία μιας εισερχόμενης αίτησης ιστοσελίδας (page requesting)
- εκτέλεση script (PHP, ASP, JSP κ.λπ.) για τη δημιουργία HTML
- πρόσβαση σε δεδομένα, όπως ένα άρθρο, από μια βάση δεδομένων
- αποθήκευση ή ενημέρωση εγγραφών σε μια βάση δεδομένων
- κρυπτογράφηση και αποκρυπτογράφηση δεδομένων
- χειρισμός μεταφορτώσεων και λήψεων αρχείων

- επεξεργασία εισόδου χρήστη (user input) μέσω JavaScript

Αυτά τα παραδείγματα, εκτός από το τελευταίο, είναι διαδικασίες που εκτελούνται στον διακομιστή. Η JavaScript “τρέχει” στην πλευρά του πελάτη, που σημαίνει ότι εκτελείται στο πρόγραμμα περιήγησης ιστού. Η JavaScript μπορεί να θεωρηθεί διαδικασία backend ή frontend, ανάλογα με το εάν ο κώδικας επηρεάζει τη διεπαφή χρήστη ή όχι.

Το backend και το frontend συνεργάζονται για να δημιουργήσουν την πλήρη εμπειρία χρήστη. Τα δεδομένα που δημιουργούνται στο backend μεταφέρονται στο frontend και παρουσιάζονται στον χρήστη. Ενώ ορισμένοι οργανισμοί έχουν ξεχωριστές ομάδες ανάπτυξης backend και frontend, η γραμμή μεταξύ των δύο επιπέδων είναι σπάνια ασπρόμαυρη. Επομένως, πολλοί προγραμματιστές γράφουν κώδικα για το backend και το frontend. Αυτό είναι γνωστό ως ανάπτυξη πλήρους στοίβας (full stack development) [34].

Οι βασικές γλώσσες προγραμματισμού που χρησιμοποιούνται στο back end είναι:

- PHP
- ASP.NET
- JAVA
- Node.js

4.2.1 PHP

Η PHP (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργασθεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script (Εικόνα 4.9).

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη [35].

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

Εικόνα 4.9 Κώδικας PHP

Η PHP αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ιστότοπων. Διάσημες εφαρμογές που κάνουν εκτενή χρήση της PHP είναι το γνωστό Σύστημα Διαχείρισης Περιεχομένου (Content Management System, WordPress και το Drupal).

4.2.2 ASP.NET

Υπάρχουν πολλοί καλοί λόγοι για να χρησιμοποιήσει κανείς το ASP.NET κατά την ανάπτυξη ενός ιστότοπου ή μιας εφαρμογής. Η υψηλή ταχύτητα, το χαμηλό κόστος και η τεράστια υποστήριξη γλώσσας είναι από τα πιο σημαντικά οφέλη. Το ASP.NET είναι ενσωματωμένο στο οικείο περιβάλλον διακομιστή των Windows, που απαιτεί λιγότερη ρύθμιση και διαμόρφωση από άλλες πλατφόρμες ανάπτυξης ιστού που πρέπει να εγκατασταθούν και να διαμορφωθούν ξεχωριστά. Η δημοτικότητα του ASP.NET καθιστά εύκολη την εύρεση διαδικτυακών πόρων και ειδικευμένων προγραμματιστών [36].

Οι ιστότοποι και οι εφαρμογές που έχουν δημιουργηθεί με ASP.NET μπορούν να είναι γρηγορότεροι και πιο αποτελεσματικοί από έναν ιστότοπο που δημιουργείται με PHP, για παράδειγμα. Οι εφαρμογές ASP.NET συντάσσονται, πράγμα που σημαίνει ότι ο κώδικας μεταφράζεται σε κώδικα αντικειμένου, ο οποίος στη συνέχεια εκτελείται. Αυτή η διαδικασία κατάρτισης διαρκεί λίγο χρόνο, αλλά συμβαίνει μόνο μία φορά. Μετά τη συλλογή, ο κώδικας μπορεί να εκτελεστεί ξανά και ξανά από την πλατφόρμα .Net πολύ γρήγορα.

Ο ερμηνευμένος κώδικας δεν εκτελείται απευθείας από το μηχάνημα, αλλά πρέπει να διαβάζεται και να ερμηνεύεται κάθε φορά πριν εκτελεστεί. Ο μεταγλωττισμένος κώδικας είναι συνήθως ταχύτερος και πιο επεκτάσιμος από τον ερμηνευμένο κώδικα και μπορεί να κάνει οτιδήποτε μπορεί να κάνει ο ερμηνευμένος κώδικας. Παραδείγματα ερμηνευμένων γλωσσών περιλαμβάνουν PHP, JavaScript και Ruby.

Η διαδικασία κατάρτισης παρέχει επίσης επικύρωση ότι όλοι οι κώδικες είναι συνεπείς. Για παράδειγμα, εάν μια μέθοδος που ονομάζεται GetUser μετονομάζεται σε GetEmployee ως μέρος ορισμένων ενημερώσεων κώδικα, οποιαδήποτε αναφορά στο GetUser σε όλη την υπόλοιπη εφαρμογή θα προκαλέσει σφάλμα κατά τη μεταγλώττιση, καθιστώντας εύκολο τον εντοπισμό και τη διόρθωση. Οι ερμηνευμένες γλώσσες δεν θα αναγνώριζαν αυτό το σφάλμα έως ότου εκτελεστεί και δοκιμαστεί ο κώδικας. Σε μια μεγάλη εφαρμογή, είναι πολύ χρονοβόρο να δοκιμάζεται με μη αυτόματο τρόπο κάθε script ή να γράφεται και να διατηρείται επιπλέον κώδικας που ελέγχει κάθε σενάριο, όποτε αλλάζει ο κώδικας.

4.2.3 JAVA

Η γλώσσα προγραμματισμού JAVA χρησιμοποιείται για την ανάπτυξη διαφόρων τύπων λογισμικού. Μια γνωστή χρήση της είναι η ανάπτυξη κώδικα για back end server. Μερικά από τα πλεονεκτήματά της είναι [37]:

- **Απλότητα:** Η Java χρησιμοποιείται από προγραμματιστές για περισσότερα από 20 χρόνια και θεωρείται μια από τις απλούστερες γλώσσες που μπορούν να μάθουν λόγω της ευκολότερης συντακτικής ορολογίας που προέρχεται από το C ++. Όποιος γνωρίζει τις θεμελιώδεις αρχές του προγραμματισμού, μπορεί να μάθει γρήγορα και να αρχίσει να χρησιμοποιεί την Java αμέσως.
- **Cross-platform:** Όντας αντικειμενοστρεφής, μεταγλωτιζόμενη γλώσσα, η Java σας επιτρέπει να γράφετε τον κώδικα μία φορά και να τον εκτελείτε οπουδήποτε σε οποιαδήποτε πλατφόρμα

διαθέτει την εικονική μηχανή της Java (JVM), (Windows, Mac OS και Linux), καθιστώντας την ιδανική επιλογή για ανάπτυξη εφαρμογών για κινητά, ανάπτυξη εφαρμογών ιστού, σύνδεση βάσης δεδομένων, δικτύωση, και πολλά άλλα.

- **Multi-Threading:** Η Java χρησιμοποιεί έναν διακομιστή ιστού πολλαπλών νημάτων που επεξεργάζεται κάθε αίτημα σε ξεχωριστό νήμα (Thread). Η επεξεργασία πολλαπλών νημάτων καθιστά την Java ένα εξαιρετικό εργαλείο για εφαρμογές υψηλής έντασης CPU, όπως ροή βίντεο, επεξεργασία εικόνας κ.λπ.
- **Ασφάλεια:** Η Java λέγεται ότι είναι πιο ασφαλής από οποιαδήποτε άλλη γλώσσα προγραμματισμού. Η ασφάλειά του λαμβάνεται από ένα μεγάλο σύνολο API, εργαλείων και υλοποιήσεων αλγορίθμων ασφαλείας, μηχανισμών και πρωτοκόλλων που χρησιμοποιούνται συνήθως. Τα API της Java περιλαμβάνουν κρυπτογραφία, υποδομή δημόσιου κλειδιού, ασφαλή επικοινωνία, έλεγχο ταυτότητας και έλεγχο πρόσβασης. Το JDK (Java-Development-Kit) αναπτύχθηκε αρχικά με μεγάλη έμφαση στην ασφάλεια που επιτρέπει τη δημιουργία ασφαλών εφαρμογών από την αρχή.
- **Βιβλιοθήκες ανοιχτού κώδικα:** Υπάρχει ένας μεγάλος αριθμός βιβλιοθηκών Java διαφορετικών σκοπών: καταγραφή, ανάλυση JSON, Unit Testing, βιβλιοθήκες ανάλυσης XML και HTML, ανταλλαγή μηνυμάτων, PDF και Excel Reading, κρυπτογραφικές βιβλιοθήκες και πολλά άλλα. Οι περισσότερες βιβλιοθήκες Java είναι ανοιχτού κώδικα και υποστηρίζονται από έμπειρους προγραμματιστές Java. Η χρήση τέτοιων βιβλιοθηκών επιταχύνει σημαντικά τον προγραμματισμό εφαρμογών ιστού.

4.2.4 Node.js

Το Node.js είναι ένα περιβάλλον εκτέλεσης JavaScript ανοιχτού κώδικα, πολλαπλών πλατφορμών, back-end που τρέχει στη μηχανή V8 (η μηχανή εκτέλεσης κώδικα του Google Chrome) και εκτελεί κώδικα JavaScript εκτός προγράμματος περιήγησης ιστού. Το Node.js επιτρέπει στους προγραμματιστές να χρησιμοποιούν JavaScript για να γράφουν εργαλεία γραμμής εντολών και για δέσμες ενεργειών από την πλευρά του διακομιστή με αποτέλεσμα να παράγουν δυναμικό περιεχόμενο ιστοσελίδας πριν από την αποστολή της σελίδας στο πρόγραμμα περιήγησης ιστού του χρήστη. Κατά συνέπεια, το Node.js αντιπροσωπεύει ένα παράδειγμα "JavaScript παντού", ενοποιώντας την ανάπτυξη εφαρμογών ιστού γύρω από μία γλώσσα προγραμματισμού, αντί για διαφορετικές γλώσσες για σενάρια διακομιστή και πελάτη. Αυτό επιτρέπει τους προγραμματιστές να χρησιμοποιούν την ίδια γλώσσα προγραμματισμού στο front και στο back end, γεγονός που απλουστεύει κατά πολύ την διαδικασία ανάπτυξης του server [38].

Το Node.js φέρνει προγραμματισμό βάσει συμβάντων (Event-Driven) σε διακομιστές ιστού, επιτρέποντας την ανάπτυξη γρήγορων διακομιστών ιστού σε JavaScript. Οι προγραμματιστές μπορούν να δημιουργήσουν διακομιστές με δυνατότητα κλιμάκωσης χωρίς να χρησιμοποιούν νήματα, χρησιμοποιώντας ένα απλοποιημένο μοντέλο προγραμματισμού βάσει συμβάντων που χρησιμοποιεί επιστροφές κλήσεων (callbacks) για να σηματοδοτήσει την ολοκλήρωση μιας εργασίας.

4.2.5 Back End Frameworks

Στην ανάπτυξη λογισμικού για server υπάρχει ένα μεγάλο μέρος του κώδικα που είναι σχεδόν πάντα ίδιο. Για να μην χρειάζεται οι προγραμματιστές να ασχολούνται με κάποιες βασικές λειτουργίες του server (π.χ authentication, connections, κ.α), χρησιμοποιούν κάποιο framework. Έτσι, δίνεται η δυνατότητα να ασχοληθούν περισσότερο με την λογική.

Κεφάλαιο 4

Τα Backend frameworks (Πίνακας 4.2) είναι βασικά για την ανάπτυξη εφαρμογών για αμέτρητες επιχειρήσεις σε όλο τον κόσμο σήμερα. Η εύρεση του κατάλληλου framework, μπορεί να είναι απολύτως απαραίτητη για τους προγραμματιστές για να διασφαλίσουν τη βέλτιστη απόδοση και επεκτασιμότητα. Με τόσες πολλές διαθέσιμες επιλογές σήμερα, η επιλογή ενός μπορεί να είναι μια ταλαιπωρία.

Μερικά από τα πλεονεκτήματα της χρήσης ενός framework είναι:

- Εξοικονόμηση χρόνου
- Επεκτασιμότητα
- Robustness
- Ασφάλεια

Πίνακας 4.2 Δημοφιλή Back End Frameworks

Framework	Γλώσσα προγραμματισμού	Δημοφιλείς χρήσεις
Django	Python	Instagram Pinterest Coursera
Laravel	PHP	Deltanet Travel Neighborhood Lender MyRank
Ruby on Rails	Ruby	ZendDesk Shopify GitHub
ExpressJS	NodeJS	MySpace GeekList Storify
CakePHP	PHP	Mapme Educationunlimited Followmy Tv
Flask	Python	Red Hat Rackspace Reddit
Asp .NET	C#	Microsoft Godaddy Ancestry
Spring Boot	Java	Trivago Via Varejo Intuit
Phoenix	Elixir	Finanical Times Fox 10 ABC15

4.3 “Πρόσθετα”

Στην πληροφορική, ένα πρόσθετο (plug-in) είναι ένα στοιχείο λογισμικού που προσθέτει μια συγκεκριμένη δυνατότητα σε ένα υπάρχον πρόγραμμα υπολογιστή. Όταν ένα πρόγραμμα υποστηρίζει τα plug-ins, τότε επιτρέπει και την προσαρμογή. Τα “πρόσθετα” συνήθως χρησιμοποιούνται σε προγράμματα περιήγησης ιστού για παραγωγή βίντεο, ανίχνευση ιών, επίδειξη νέων τύπων αρχείων, αλλαγή εμφάνισης, και εκτέλεση παιχνιδιών. Δημοφιλέστερα “πρόσθετα” για την εκτέλεση παιχνιδιών είναι το Flash, Java και το Unity web player.

4.3.1 Adobe Flash Player

Το Adobe Flash είναι μια πλατφόρμα δημιουργίας λογισμικού που χρησιμοποιείται για την παραγωγή κινούμενων σχεδίων, εφαρμογών πολυμέσων, ιστοσελίδων, εφαρμογών και παιχνιδιών για κινητά κ.α. Το Flash εμφανίζει κείμενο, διανυσματικά και raster γραφικά για να παρέχει κινούμενα σχέδια, βίντεο, παιχνίδια και εφαρμογές. Επιτρέπει την αναπαραγωγή ήχου και βίντεο, και μπορεί να χειριστεί το ποντίκι, το πληκτρολόγιο, το μικρόφωνο, την κάμερα καθώς και όλους τους σένσορες μιας κινητής συσκευής.

Οι χρήστες μπορούν να δουν το περιεχόμενο Flash μέσω του Flash Player (web browsers), AIR (για desktop ή mobile apps) ή χρησιμοποιώντας players τρίτων κατασκευαστών όπως Scaleform (για video games). Το Adobe Flash Player, το οποίο υποστηρίζεται από τα Microsoft Windows, το macOS και το Linux, επιτρέπει στους τελικούς χρήστες να δουν περιεχόμενο Flash χρησιμοποιώντας προγράμματα περιήγησης στο web, κάτι που υποστηριζόταν μέχρι το τέλος του 2020. Στα κινητά (Android και iOS) το Flash κάνει export σε apk ή ipa ενσωματώνοντας το Adobe AIR, δίνοντας τη δυνατότητα στους developers να φτιάξουν εφαρμογές και να τις δημοσιεύσουν στο GooglePlay και στο AppStore.

Η γλώσσα προγραμματισμού ActionScript επιτρέπει την ανάπτυξη διαδραστικών κινουμένων σχεδίων, βίντεο, παιχνιδιών, εφαρμογών web, εφαρμογών desktop και εφαρμογών για κινητά. Οι προγραμματιστές μπορούν να υλοποιήσουν λογισμικό Flash χρησιμοποιώντας ένα IDE όπως το Adobe Animate, Adobe Flash Builder, Adobe Director, FlashDevelop και Powerflasher FDT. Το Adobe AIR επιτρέπει την ανάπτυξη εφαρμογών desktop και mobile που μπορούν να αναπτυχθούν με το Flash και να δημοσιευτούν σε Windows, macOS, Android και iOS.

Αν και το Flash ήταν μια κυρίαρχη online πλατφόρμα στο web, σταδιακά η χρήση του επικεντρώθηκε στη δημιουργία mobile εφαρμογών και animations. Ο βασικός λόγος της εγκατάλειψης των browsers ήταν η διαμάχη με την Apple, καθώς η τελευταία απαγόρευσε τη χρήση του στο λειτουργικό της υποστηρίζοντας ότι δεν ήταν ασφαλές, ήταν αργό κ.α. Σύμφωνα όμως με πολλούς, η πραγματική αιτία ήταν ότι θα έδινε στους χρήστες τη δυνατότητα να έχουν πρόσβαση σε δωρεάν εφαρμογές και παιχνίδια μέσω browser χωρίς να χρειάζεται να αγοράσουν οτιδήποτε από το AppStore.

Τα Flash παιχνίδια ήταν πολύ δημοφιλή στο Διαδίκτυο, με ιστοσελίδες όπως το Newgrounds να είναι αφιερωμένες στη φιλοξενία Flash παιχνιδιών. Δημοφιλή παιχνίδια που αναπτύχθηκαν με το Flash περιλαμβάνουν τα Angry Birds, Clash of Clans, το FarmVille, AdventureQuest και Machinarium.

Η Adobe εισήγαγε διάφορες τεχνολογίες για να βοηθήσει στην δημιουργία video games, όπως το Adobe AIR (για να δημοσιεύονται τα παιχνίδια σε desktop ή mobile πλατφόρμες όπως το Android και το iOS), Adobe Scout (για τη βελτίωση των επιδόσεων), CrossBridge (για τη μετατροπή C++ παιχνιδιών σε Flash), και Stage3D (για την υποστήριξη GPU-accelerated video games). 3D frameworks όπως Away3D και Flare3D απλοποιούν τη δημιουργία 3D περιεχομένου για το Flash.

Το Flash επίσης χρησιμοποιείται για την κατασκευή interfaces και Head Up Displays για 3D παιχνίδια χρησιμοποιώντας το Scaleform GFX, μια τεχνολογία που χρησιμοποιεί Flash περιεχόμενο σε μη-Flash παιχνίδια. Το Scaleform υποστηρίζεται από περισσότερες από 10 μεγάλες video game engines, συμπεριλαμβανομένων των Unreal Engine, UDK, CryEngine και PhyreEngine, και έχει χρησιμοποιηθεί για να παρέχει 3D interfaces για περισσότερα από 150 σημαντικούς τίτλους βιντεοπαιχνιδιών από την έναρξή του το 2003 [39].

4.3.2 Java

Οι μικροεφαρμογές Java ήταν μικρές εφαρμογές γραμμένες στη γλώσσα προγραμματισμού Java ή άλλη γλώσσα προγραμματισμού που μεταγλωττίζεται στο Java bytecode. Ο χρήστης ξεκινάει την εφαρμογή Java από μια ιστοσελίδα και η εφαρμογή εκτελείται στη συνέχεια σε μια εικονική μηχανή Java (JVM) σε μια διαδικασία ξεχωριστή από το ίδιο το πρόγραμμα περιήγησης ιστού. Μια εφαρμογή Java θα μπορούσε να εμφανιστεί σε ένα πλαίσιο της ιστοσελίδας, σε ένα νέο παράθυρο εφαρμογής, στο Sun's AppletViewer ή σε αυτόνομο εργαλείο για τη δοκιμή μικροεφαρμογών.

Τα applet Java εισήχθησαν στην πρώτη έκδοση της γλώσσας Java, η οποία κυκλοφόρησε το 1995. Αρχίζοντας το 2013, τα μεγάλα προγράμματα περιήγησης στο Web άρχισαν να καταργούν σταδιακά την υποστήριξη για τις υποκείμενες μικροεφαρμογές τεχνολογίας που χρησιμοποιούνταν, με τις εφαρμογές να μην μπορούν να εκτελεστούν έως το 2015–2017. Τα applet Java καταργήθηκαν από το Java 9 το 2017 και αφαιρέθηκαν από το Java SE 11, το οποίο κυκλοφόρησε τον Σεπτέμβριο του 2018.

Οι μικροεφαρμογές Java γράφονταν συνήθως σε Java, αλλά και σε άλλες γλώσσες όπως Jython, JRuby, Pascal, Scala ή Eiffel (μέσω SmartEiffel). Λειτουργούσαν με πολύ γρήγορες ταχύτητες και μέχρι το 2011, ήταν πολλές φορές πιο γρήγορες από τη JavaScript. Σε αντίθεση με τη JavaScript, οι μικροεφαρμογές Java είχαν πρόσβαση σε τρισδιάστατη επιτάχυνση υλικού, καθιστώντας τις κατάλληλες για μη ασήμαντες, εντατικές υπολογιστικές απεικονίσεις. Καθώς τα προγράμματα περιήγησης έχουν αποκτήσει υποστήριξη για γραφικά που επιταχύνουν το υλικό χάρη στην τεχνολογία καμβά (ή συγκεκριμένα το WebGL στην περίπτωση των τρισδιάστατων γραφικών), οι διαφορές σε απόδοση με τη Java έχουν ελαχιστοποιηθεί.

Δεδομένου ότι το Java bytecode είναι cross-platform (ή ανεξάρτητο από την πλατφόρμα), τα applet Java μπορούν να εκτελεστούν από προγράμματα περιήγησης (ή άλλους πελάτες) για πολλές πλατφόρμες, συμπεριλαμβανομένων των Microsoft Windows, FreeBSD, Unix, macOS και Linux. Δεν μπορούν να εκτελεστούν σε σύγχρονες κινητές συσκευές, οι οποίες δεν υποστηρίζουν Java [40].

4.3.3 Unity Web Player

Το Unity web player, είναι ένα “πρόσθετο” το οποίο δίνει τη δυνατότητα στο χρήστη να “τρέξει” παιχνίδια που είναι φτιαγμένα με τη μηχανή ανάπτυξης παιχνιδιών Unity [41].

Η Unity είναι μια μηχανή ανάπτυξης παιχνιδιών πολλαπλών πλατφορμών που αναπτύχθηκε από την Unity Technologies, η οποία ανακοινώθηκε και κυκλοφόρησε για πρώτη φορά τον Ιούνιο του 2005 στο Worldwide Developers Conference της Apple Inc. ως αποκλειστική μηχανή παιχνιδιών Mac OS X. Από το 2018, η μηχανή είχε επεκταθεί για να υποστηρίξει περισσότερες από 25 πλατφόρμες. Μπορεί να χρησιμοποιηθεί για τη δημιουργία τρισδιάστατων, δισδιάστατων, εικονικής πραγματικότητας και παιχνιδιών επαυξημένης πραγματικότητας, καθώς και προσομοιώσεων και άλλων εμπειριών. Έχει υιοθετηθεί από βιομηχανίες εκτός βιντεοπαιχνιδιών, όπως φιλμ, αυτοκίνητα, αρχιτεκτονική, μηχανική και κατασκευές. Αρκετές σημαντικές εκδόσεις του Unity έχουν κυκλοφορήσει μέχρι σήμερα. Η τελευταία σταθερή έκδοση, 2020.2.2, κυκλοφόρησε τον Ιανουάριο του 2021.

4.4 Τεχνολογίες που επιλέχθηκαν για το παιχνίδι

Πολλές είναι οι τεχνολογίες που μπορούν να χρησιμοποιηθούν για την ανάπτυξη του παιχνιδιού που πραγματεύεται η συγκεκριμένη πτυχιακή εργασία. Αυτές που τελικά επιλέχθηκαν είναι η Angular, για το front end μέρος του παιχνιδιού και μια "serverless" λύση για το back end (back end as a service) που ονομάζεται Firebase. Η γλώσσα προγραμματισμού που χρησιμοποιείται και στις δύο είναι η TypeScript. Έτσι, διευκολύνεται η διαδικασία ανάπτυξης αφού χρησιμοποιείται η ίδια γλώσσα για το front end αλλά και για το back end.

Αυτές οι δύο τεχνολογίες, χρησιμοποιούνται πολύ συχνά σε συνδυασμό για ανάπτυξη εφαρμογών και όχι τόσο παιχνιδιών. Παρόλα αυτά, στο συγκεκριμένο project χρησιμοποιούνται για να δείξουν ότι είναι εξίσου άξιες με άλλες τεχνολογίες για την ανάπτυξη παιχνιδιού.

4.4.1 Angular

Η Angular είναι μία τεχνολογία που παρέχει τη δυνατότητα οργάνωσης του κώδικα σε επιμέρους τμήματα. Το καθένα από αυτά εκτελεί συγκεκριμένες λειτουργίες και όλα μαζί σε συνδυασμό απαρτίζουν την εφαρμογή στο σύνολό της. Μέσω της TypeScript, χρησιμοποιούμε αντικειμενοστραφή μέθοδο προγραμματισμού, κάτι το οποίο ταιριάζει απόλυτα στην δημιουργία ενός παιχνιδιού. Αυτό συμβαίνει διότι δίνεται η δυνατότητα περιγραφής των αντικειμένων του παιχνιδιού μέσω κλάσεων.

Τα τμήματα στα οποία μπορεί να οργανωθεί και να καταμεριστεί ο κώδικας είναι ουσιαστικά κλάσεις οι οποίες "βαφτίζονται" ανάλογα με την λειτουργία που επιτελούν ως εξής:

- Components
- Modules
- Services
- Directives

Ένα Component συνήθως χωρίζεται σε ένα αρχείο TypeScript, στο οποίο βρίσκεται η λογική, και σε ένα αρχείο HTML, όπου βρίσκονται τα στοιχεία (elements) που το απαρτίζουν. Υπάρχει πάντα ένα αρχικό component από το οποίο ξεκινάει η εφαρμογή. Μέσα σε αυτό, μπορούν να τοποθετηθούν επιμέρους components, σε δενδρική μορφή, με αποτέλεσμα τη δημιουργία ιεραρχίας. Οι μεταβλητές και οι μέθοδοι που ορίζονται στο αρχείο TypeScript, μπορούν να χρησιμοποιηθούν στο template (αρχείο HTML) με αποτέλεσμα να κάνουν δυναμικό το περιεχόμενο. Με τη χρήση του data binding η δυναμικότητα αυτή ενισχύεται. Συνήθως, τα components αποκαλούνται "views" γιατί ουσιαστικά είναι υπεύθυνα για οτιδήποτε εμφανίζεται στην οθόνη. Είναι σημαντικό να αναφερθεί ότι για να μοιραστούν δεδομένα δύο components, πρέπει είτε να συνδέονται με σχέση γονέα-παιδιού είτε να χρησιμοποιηθεί κάποιο service. Ένα component έχει ένα κύκλο ζωής (lifecycle) που περνάει από κάποια στάδια, μερικά από τα οποία είναι το **ngOnInit** (όταν δημιουργείται), **ngOnDestroy** (όταν καταστρέφεται), **ngAfterContentChecked** (όταν το περιεχόμενο έχει ελεγχθεί από τον compiler της Angular).

Τα Modules δίνουν τη δυνατότητα της καλύτερης οργάνωσης των components. Συνήθως χρησιμοποιούνται για να εισάγουν βιβλιοθήκες και άλλα modules της Angular, καθώς και εξωτερικές βιβλιοθήκες. Σε ένα module δηλώνονται τα components, services και modules που χρειάζονται. Κάθε εφαρμογή της Angular πρέπει να έχει ένα τουλάχιστον module (root module) αλλά συνιστάται η τμηματοποίηση της εφαρμογής και κάθε feature ξεχωριστά να γίνεται module. Με αυτό τον τρόπο μπορεί το module να ξαναχρησιμοποιηθεί σε κάποια άλλη εφαρμογή.

Τα Services, σε αντίθεση με τα components, δεν χρησιμοποιούνται για να "δείξουν" στον χρήστη, αλλά για να εκτελούν την λογική της εφαρμογής. Πιο αναλυτικά, αν υπάρχει κομμάτι κώδικα που μπορεί να

χρησιμοποιηθεί σε περισσότερα από ένα components, τότε γράφεται σε κάποιο service. Ένα service καλείται απο components κάνοντας **Inject** στον constructor του component. Επιπλέον, ένα service μπορεί να καλεί περισσότερα services κάνοντας έτσι ευκολότερη και πιο κατανοητή την δομή του κώδικα. Τα services αποτελούν τον μοναδικό τρόπο με τον οποίο μοιράζονται δεδομένα τα components που δεν έχουν σχέση μεταξύ τους.

Τα Directives είναι από τις πιο σύνθετες επιλογές κλάσεων της Angular. Δίνουν την δυνατότητα παραμετροποίησης των στοιχείων της HTML ενός component διαμορφώνοντας εσωτερικά το DOM. Η Angular από μόνη της παρέχει μερικά attribute directives όπως:

- NgClass (δίνει την δυνατότητα αλλαγής των css κλάσεων αν ικανοποιείται μια συνθήκη)
- NgStyle (δίνει την δυνατότητα αλλαγής συγκεκριμένων στύλ ανάλογα με τη συνθήκη)
- NgModel (δίνει την δυνατότητα για two-way data-binding) (Εικόνα 4.10).

```
<input type="text" [(ngModel)]="searchTerm">
<button (click)="searchPlayer()">Εύρεση παίκτη</button>
```

Εικόνα 4.10 ngModel Directive

Επιπλέον, υπάρχουν και κάποια directives για την δομή του template (structural directives) (Εικόνα 4.11). Κάποια από αυτά είναι:

- ngIf (εμφανίζει το element μόνο αν ικανοποιείται η συνθήκη που δίνεται)
- ngFor (χρησιμοποιείται για να εμφανίσει πολλαπλά elements από έναν πίνακα ή λίστα)
- ngSwitch (πραγματοποιεί μια switch case και ανάλογα με το αποτέλεσμα εμφανίζει το κατάλληλο element).

```
<div *ngIf="resultUser">
  {{resultUser.displayName}}
  <button (click)="addFriend()">Προσθήκη φίλου</button>
</div>
<hr>
<br>
<div *ngFor="let f of friends">
  {{f.displayName}} <span>{{f.active ? 'Online' : 'Offline'}}</span>
</div>
```

Εικόνα 4.11 NgIf και NgFor Directives

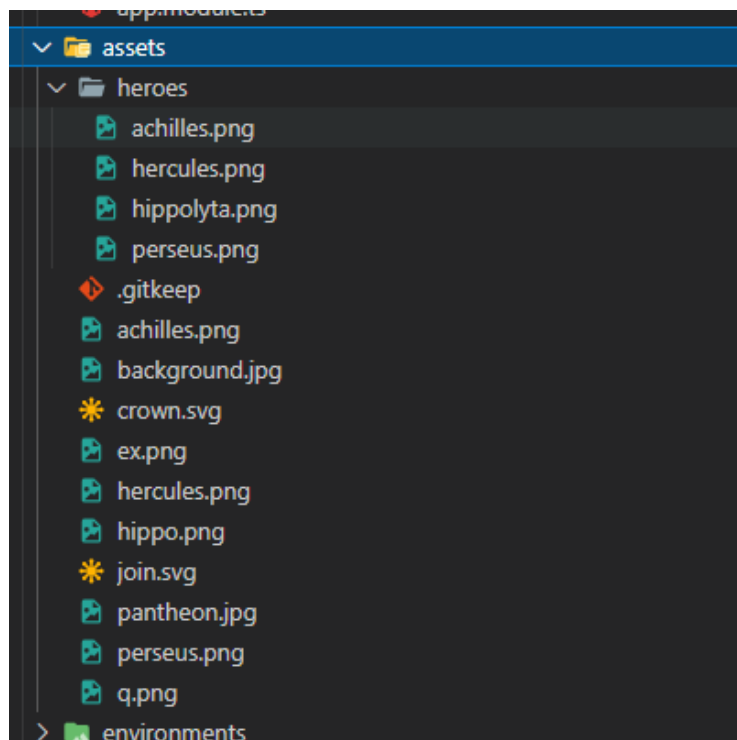
Η διαχείριση σελίδων της εφαρμογής γίνεται μέσω του συστήματος Router. Ένας router αποτελείται από μονοπάτια (paths) τα οποία αντιστοιχούν σε components ή modules. Αυτό έχει ως αποτέλεσμα μεγαλύτερη απόδοση, διότι τα modules δεν φορτώνονται όλα μαζί αλλά μόνο όταν ο χρήστης μεταβεί στην υποσελίδα που θέλει (lazy loading). Τα paths μπορούν να προστατευτούν μέσω Guards. Τα guards έχουν πολλές λειτουργίες για να επιτρέπουν την πρόσβαση ή όχι σε συγκεκριμένες σελίδες ή υποσελίδες κ.α. Μέσω κάποιων build-in λειτουργιών της Angular όπως **canActivate** είναι πολύ εύκολο να επιτρέπεται η πρόσβαση σε σελίδες της εφαρμογής μόνο από εξουσιοδοτημένους χρήστες.

Τα περισσότερα δεδομένα μιας Angular εφαρμογής βρίσκονται στα components και τα services ή “κατεβάζονται” από κάποια βάση δεδομένων. Όμως, υπάρχει η ανάγκη για μεταβλητές περιβάλλοντος που δεν αλλάζουν κατά την εκτέλεση της εφαρμογής. Η Angular προσφέρει λύση για αυτό το πρόβλημα μέσω των environment (Εικόνα 4.12). Συνήθως υπάρχει ένα αρχείο (ή περισσότερα) **environment.ts** που κρατάει τέτοιες μεταβλητές περιβάλλοντος (π.χ στοιχεία για την σύνδεση με βάση δεδομένων).

```
export const environment = {
  production: false,
  firebaseConfig: {
    apiKey: "AIzaSyDA2GUtPGxCws3oXf-TfVtWzQgnigL6Gwk",
    authDomain: "greek-mythos.firebaseio.com",
    databaseURL: "https://greek-mythos.firebaseio.com",
    projectId: "greek-mythos",
    storageBucket: "greek-mythos.appspot.com",
  }
};
```

Εικόνα 4.12 Μεταβλητή περιβάλλοντος

Ακόμα ένα σημαντικό πρόβλημα που εντοπίζεται στην ανάπτυξη μιας εφαρμογής ή παιχνιδιού είναι η αποθήκευση και η διαχείριση των **assets** (αρχεία φωτογραφιών, μουσικά κλιπ, βίντεο, κ.α). Η Angular λύνει αυτό το πρόβλημα, ορίζοντας φακέλους που χρησιμοποιούνται για την αποθήκευση αυτών των αρχείων (Εικόνα 4.13).



Εικόνα 4.13 Assets του παιχνιδιού

Μέσω της TypeScript ή καλύτερα του compiler της, δίνεται η δυνατότητα να οριστούν σε ποιες εκδόσεις της JavaScript θα λειτουργεί η εφαρμογή. Αυτό είναι πολύ σημαντικό αν υπάρχει η ανάγκη για υποστήριξη παλαιότερων περιγητών ιστού.

4.4.2 Firebase

Υπάρχουν πολλά συστήματα που μπορούν να χρησιμοποιηθούν στο back end. Η Firebase, αν και σχετικά καινούργια, δίνει την δυνατότητα ενός ολοκληρωμένου συστήματος για το back end. Αποτελεί μια λύση με σύστημα νέφους (cloud based system), με αποτέλεσμα να υπάρχει μεγάλη επεκτασιμότητα. Είναι πολύ δημοφιλής στις εφαρμογές για κινητά τηλέφωνα αλλά και για εφαρμογές περιγητή ιστού.

Αρχικά, η Firebase δημιουργήθηκε από μία ξεχωριστή εταιρεία το 2011. Από το 2014 και μετά, ανήκει στην Google και ανταγωνίζεται με το cloud based solution της Amazon το AWS. Τον Μάιο του 2016, στο Google I / O, το ετήσιο συνέδριο προγραμματιστών της εταιρείας, η ομάδα της Firebase παρουσίασε το Firebase Analytics και ανακοίνωσε ότι επεκτείνει τις υπηρεσίες του για να γίνει μια ενοποιημένη πλατφόρμα backend-as-a-service (BaaS) για προγραμματιστές κινητών. Η Firebase ενοποιείται πλέον με διάφορες άλλες υπηρεσίες Google, όπως το Google Cloud Platform, το AdMob και το Google Ads, για να προσφέρουν ευρύτερα προϊόντα και κλίμακα για προγραμματιστές. Το Google Cloud Messaging, η υπηρεσία Google για την αποστολή ειδοποιήσεων push σε συσκευές Android, αντικαταστάθηκε από ένα προϊόν Firebase, το Firebase Cloud Messaging, το οποίο πρόσθεσε τη λειτουργικότητα για την παροχή ειδοποιήσεων push σε συσκευές iOS και ιστού. Τον Ιανουάριο του 2017, η Google απέκτησε το Fabric και το Crashlytics από το Twitter για να προσθέσει αυτές τις υπηρεσίες στο Firebase [42].

Η Firebase απαρτίζεται από πολλά προϊόντα, τα περισσότερα από τα οποία είναι δωρεάν (κατά ένα μέρος) αλλά υπάρχουν και κάποια που απαιτούν πληρωμή. Αυτά χωρίζονται σε τέσσερις κατηγορίες:

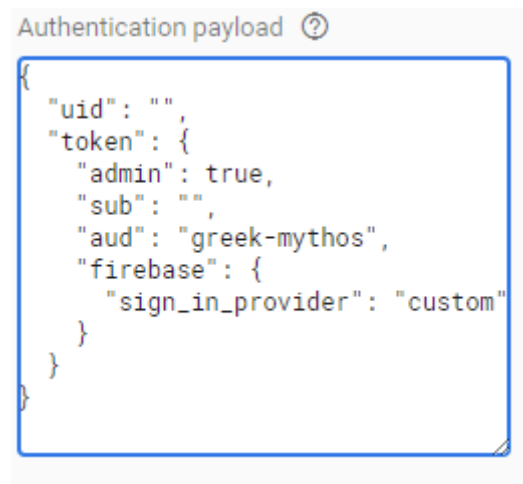
- **Build**
 - **Authentication**
 - Cloud Firestore
 - **Realtime Database**
 - **Storage**
 - **Hosting**
 - **Functions**
 - Machine Learning
- **Release & Monitor**
 - Crashlytics
 - Performance
 - Test Lab
 - App Distribution
- **Analytics**
 - Dashboard
 - Realtime
 - Events
 - Conversions
 - Audiences
 - Funnels

- Custom Definitions
- Latest Release
- Retention
- DebugView
- Engage
 - Predictions
 - A/B Testing
 - Cloud Messaging
 - Remote Config
 - Dynamic Links
 - AdMob

Η πλατφόρμα δίνει την δυνατότητα να χρησιμοποιηθούν κάποιες από τις λειτουργίες αυτές όπου χρειάζεται. Στα πλαίσια της πτυχιακής εργασίας χρησιμοποιήθηκαν μόνο τα Authentication, Realtime Database, Storage, Hosting, και Functions.

Το **Authentication** περιέχει μεθόδους αυθεντικοποίησης χρηστών χρησιμοποιώντας τις τελευταίες τεχνολογίες κρυπτογράφησης. Δίνεται η δυνατότητα εγγραφής στην εφαρμογή μέσω τρίτων π.χ Facebook, Twitter, Google κ.α , καθώς και η αυθεντικοποίηση ανώνυμου χρήστη. Επιπλέον, μπορούν να δημιουργηθούν λογαριασμοί μέσω email και κωδικού ακόμα και με αριθμό τηλεφώνου. Είναι σημαντικό να σημειωθεί ότι οι κωδικοί των χρηστών είναι κρυπτογραφημένοι και **μη** προσβάσιμοι από τους διαχειριστές της εκάστοτε εφαρμογής.

Πολλές φορές στις εφαρμογές υπάρχει η ανάγκη ρόλων των χρηστών π.χ διαχειριστής, moderator, απλός χρήστης κ.α. Η Firebase δίνει λύση σε αυτό το πρόβλημα μέσω των “custom claims” (Εικόνα 4.14). Επειδή το Authentication χρησιμοποιεί JWT (Json Web Token), δίνεται η δυνατότητα να εισαχθούν μέσα στο token, δεδομένα από τον server π.χ **“admin”: true**. Αυτό διευκολύνει την διαδικασία ανάθεσης ρόλων στους χρήστες μιας εφαρμογής.



Εικόνα 4.14 Custom Claims

Για την διαχείριση δεδομένων της εφαρμογής συνήθως χρειάζεται μια βάση δεδομένων. Η Firebase δίνει δύο προϊόντα ως βάσεις δεδομένων. Η **Firestore** είναι η καινούργια βάση που προσφέρεται και υποστηρίζει ευέλικτες, ιεραρχικές δομές δεδομένων. Τα δεδομένα οργανώνονται σε συλλογές που περιέχουν έγγραφα. Τα έγγραφα μπορούν να περιέχουν πολύπλοκα ένθετα αντικείμενα ή/και υπο-συλλογές.

Η δεύτερη λύση είναι η **Realtime Database**, η οποία χρησιμοποιείται και στα πλαίσια της πτυχιακής εργασίας. Προσφέρει μια NoSQL βάση δεδομένων χωρίς αυστηρούς τύπους, δηλαδή δεν χρειάζεται να οριστούν οι τύποι των δεδομένων για κανένα σημείο της βάσης (Εικόνα 4.15). Η δομή της είναι δενδρική ξεκινώντας από μια ρίζα (root) και φτάνοντας μέχρι 32 επίπεδα. Η Firebase προσφέρει ένα SDK (Software Development Kit) που δίνει την δυνατότητα στους προγραμματιστές να χρησιμοποιήσουν τα προϊόντα της με ευκολία. Επιπλέον, μέσω του SDK μπορούν να χρησιμοποιηθούν ακροατές (listeners) σε σημεία της βάσης, έχοντας ως αποτέλεσμα αν αλλάξουν τα δεδομένα στην βάση να ενημερωθεί ο χρήστης αυτόματα. Η ασφάλεια της βάσης είναι αρκετά εύκολη στην υλοποίηση, με τη βοήθεια των κανόνων (rules). Οι κανόνες επιτρέπουν ή όχι την ανάγνωση και εγγραφή στην βάση σε όλα τα σημεία. Το γεγονός ότι δεν χρειάζεται να οριστεί η δομή των δεδομένων μέσα στην βάση, κάνει πολύ πιο απλή την διαδικασία ανάπτυξης των εφαρμογών. Βέβαια, μεταγενέστερες αλλαγές που επηρεάζουν την δομή σε μεγάλο βαθμό μπορεί να προκαλέσουν προβλήματα αν δεν γίνει σωστή διαχείριση.



Εικόνα 4.15 Ένα αντικείμενο στη βάση RealTime

Αν και τα περισσότερα δεδομένα αποθηκεύονται στην βάση δεδομένων, υπάρχουν κάποια που έχουν μεγάλο μέγεθος όπως φωτογραφίες, βίντεο, μεγάλα κείμενα κ.α. Αυτά συνήθως αποθηκεύονται σε κάποιο άλλο σημείο και όχι στις βάσεις δεδομένων. Η Firebase έχει ένα συγκεκριμένο τρόπο αποθήκευσης τέτοιων δεδομένων που ονομάζεται **Storage**. Χρησιμοποιεί σύστημα με bucket (δηλαδή έναν χώρο που αποθηκεύονται μεγάλα αρχεία). Για την οργάνωση χρησιμοποιείται το σύστημα φακέλων/αρχείων που όλοι γνωρίζουν. Για να μην έχει οποιοσδήποτε πρόσβαση σε όλα τα αρχεία, χρησιμοποιούνται κανόνες όπως και με την Realtime Database.

Η Firebase προσφέρει την δυνατότητα στους προγραμματιστές να “ανεβάσουν” την ιστοσελίδα/εφαρμογή στους servers της Google. Μέσω της υπηρεσίας **Hosting**, η εφαρμογή είναι εύκολα προσβάσιμη από ένα σύνδεσμο του τύπου <όνομα εφαρμογής>.web.app. Επίσης, δίνεται η δυνατότητα για επιστροφή σε παλαιότερη έκδοση της ιστοσελίδας.

Αν και δεν υπάρχει πρόσβαση στον server που εκτελεί αυτές τις υπηρεσίες, η Firebase έδωσε την δυνατότητα αλληλεπίδρασης μαζί του μέσω των **Cloud Functions**. Αποτελούνται από μεθόδους που εκτελούνται στους servers της Firebase σε συγκεκριμένους χρόνους. Πιο αναλυτικά, χωρίζονται σε ορισμένες κατηγορίες ανάλογα με τον τρόπο χρήσης τους:

- Firestore triggers
- Database triggers
- Auth triggers
- Google Analytics triggers
- Crashlytics triggers
- Storage triggers
- Remote Config triggers
- Cloud Pub/Sub triggers
- Http callable

```
export const rollDice = dbfs.https.onCall(rollDiceHandler);

export const joinRoom = dbfs.https.onCall(joinRoomHandler);

export const onStartGame = dbfs.database.ref(`rooms/{roomId}/start`).onUpdate(startGameHandler);

export const checkGameActivity = dbfs.https.onCall(checkGameActivityHandler);
```

Εικόνα 4.16 Cloud Functions

Τα Firestore και Database triggers, εκτελούνται όταν υπάρχει κάποια αλλαγή στην βάση δεδομένων firestore ή Realtime αντίστοιχα. Τέτοιες αλλαγές μπορεί να είναι στις περιπτώσεις που κάτι δημιουργήθηκε, άλλαξε ή διαγράφηκε από την βάση. Τα Auth Triggers, εκτελούνται όταν ένας χρήστης δημιουργείται ή διαγράφεται. Όσον αφορά τα Google Analytics triggers, εκτελούνται όταν οι χρήστες κάνουν κάποιο “event” που έχει οριστεί στα Analytics. Τα Crashlytics triggers, ενεργοποιούνται όταν υπάρχει κάποιο πρόβλημα με την εφαρμογή. Τα Storage triggers, εκτελούνται όταν “ανεβαίνει”, ενημερώνεται ή διαγράφεται κάποιος φάκελος ή αρχείο. Τα Remote Config triggers, εκτελούνται όταν έχει γίνει κάποιο γεγονός όπως η αλλαγή έκδοσης της εφαρμογής. Στα Cloud Pub/Sub triggers, ο προγραμματιστής ορίζει ένα χρονοδιάγραμμα εκτέλεσης των μεθόδων. Δηλαδή, διευκρινίζει κάθε πότε θα εκτελείται η μέθοδος, για παράδειγμα “κάθε μια ώρα”, “κάθε Πέμπτη στις 5” κ.α. Τέλος, οι Http callable μέθοδοι καλούνται μέσω http requests και μπορούν να δεχτούν οποιαδήποτε δεδομένα.

Κατά την εκτέλεση των μεθόδων, μπορεί να διαβαστεί η βάση δεδομένων, δίνοντας έτσι την δυνατότητα στους προγραμματιστές να ελέγξουν, να διαγράψουν και γενικότερα να εκτελέσουν την λογική που θέλουν κατευθείαν στον server. Οι Cloud Functions εκτελούνται σε περιβάλλον Node.js και υπάρχει πληθώρα τρίτων βιβλιοθηκών που μπορούν να χρησιμοποιηθούν για να εμπλουτίσουν τις λειτουργίες της εφαρμογής.

Είναι σημαντικό να σημειωθεί ότι αυτές οι μέθοδοι εκτελούν τον κώδικα που έχει οριστεί και μετά σταματάνε. Πρέπει να ξανά ενεργοποιηθεί κάποιο trigger ή event για να επανεκτελεστεί ο κώδικας. Αν αργήσει μια μέθοδος να τελειώσει την εκτέλεση της είτε από λάθος είτε από άλλο λόγο, τότε υπάρχει ένας μέγιστος χρόνος που αν ξεπεραστεί η μέθοδος σταματάει.

4.4.3 Γιατί επιλέχθηκαν αυτές οι τεχνολογίες

Οι δύο αυτές τεχνολογίες συνεργάζονται καλά μεταξύ τους διότι υπάρχει το SDK της Firebase που διευκολύνει κατά πολύ την επικοινωνία του front end με το back end. Εκτός από αυτό, η Angular, η οποία αναπτύχθηκε από την Google, αποτελεί μια από τις πιο δημοφιλείς τεχνολογίες για front end σήμερα. Επιτρέπει την επεκτασιμότητα της εφαρμογής καθώς κρατάει οργανωμένο το περιβάλλον ανάπτυξης. Επιπλέον, λόγω της TypeScript ο εντοπισμός των προβλημάτων γίνεται ευκολότερα.

Η επιλογή της Firebase έγινε για να ενισχύσει την επεκτασιμότητα της εφαρμογής, αφού είναι ένα σύστημα νέφους εξαιρετικά επεκτάσιμο. Επιπροσθέτως, η επιλογή της έδωσε την δυνατότητα για ένα έτοιμο σύστημα αυθεντικοποίησης χρηστών. Η επιλογή της βάσης δεδομένων έγινε για την απλότητα και ταχύτητα που προσφέρει μια NoSQL βάση, κάτι πολύ σημαντικό για τα παιχνίδια. Ένας ακόμα σημαντικός λόγος της επιλογής είναι η διαχείριση των φωτογραφιών προφίλ των παικτών, που με το Storage της Firebase λύνεται αμέσως. Ο κυριότερος λόγος που επιλέχθηκαν αυτές οι δύο τεχνολογίες είναι ότι χρησιμοποιούν την ίδια γλώσσα προγραμματισμού.

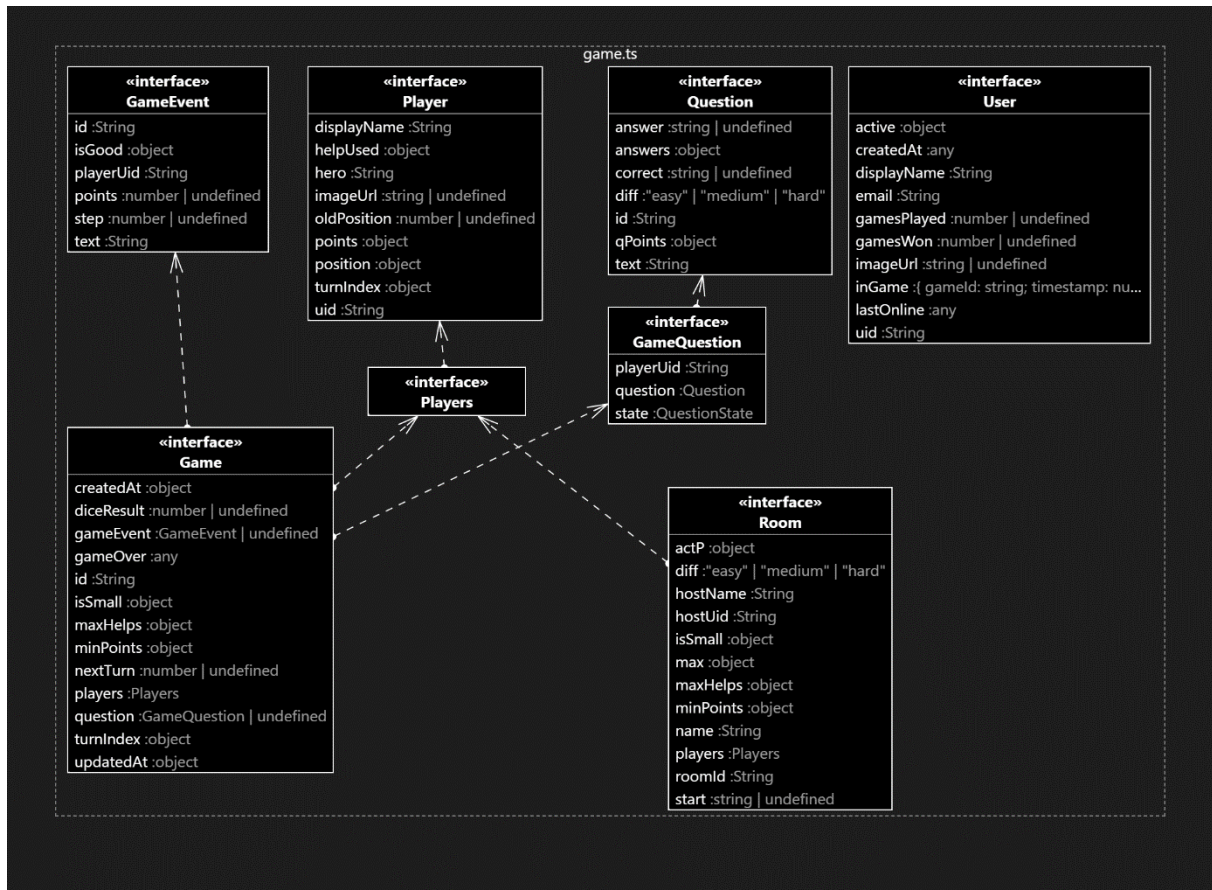
4.5 Πώς χρησιμοποιήθηκαν οι τεχνολογίες

Για την βάση δεδομένων χρησιμοποιήθηκε η υπηρεσία RealTime Database της Firebase. Η συγκεκριμένη αποτελεί μια NoSQL (χωρίς αυστηρή δομή και σχέσεις μεταξύ των δεδομένων) βάση, με υψηλές ταχύτητες για εγγραφή και ανάγνωση. Λόγω της δομής της, επιτρέπει την εύκολη τροποποίηση των δομών για τα δεδομένα που αποθηκεύονται. Ο τρόπος που λειτουργεί η Firebase είναι μέσω web sockets. Τα Web Sockets δίνουν την δυνατότητα για ενημερώσεις σε πραγματικό χρόνο, κάτι που είναι απαραίτητο για την ανάπτυξη παιχνιδιών.

Ένα ακόμα πολύ σημαντικό πλεονέκτημα της RealTime είναι η δυνατότητα της να κρατάει την κατάσταση της σύνδεσης με τους συνδεδεμένους χρήστες. Αυτό έχει ως αποτέλεσμα να πραγματοποιείται εύκολα το σύστημα των φίλων στην εφαρμογή, δηλαδή, να μπορεί να δει ένας χρήστης ποιοι φίλοι του είναι ενεργοί ανά πάσα στιγμή. Αυτό υλοποιείται με μια μέθοδο που μας δίνει το SDK (Software Development Kit) της Firebase, όπου ορίζεται ένα αντικείμενο το οποίο θα γραφτεί στην βάση όταν ο χρήστης θα αποσυνδεθεί.

Φυσικά, κανένα εργαλείο δεν είναι τέλειο. Έτσι και η RealTime Database έχει και κάποιους περιορισμούς. Αρχικά, υπάρχει μέγιστο όριο στις ταυτόχρονες συνδέσεις (μέχρι 100.000), κάτι που μπορεί να ξεπεραστεί με δημιουργία καινούργιων βάσεων. Πολύ σημαντικό είναι να σημειωθεί ότι δεν μπορούν να γίνουν περίπλοκα queries (ερωτήματα στην βάση για επιστροφή δεδομένων μέσω διαφόρων παραμέτρων), διότι δεν υποστηρίζεται από την βάση.

Τα δεδομένα στην βάση αναπαρίστανται από διαδρομές που ξεκινούν από μία ρίζα (root '/'). Υπάρχει ένα όριο στα παιδιά κάτω από την ρίζα (32). Αντίθετα από μια SQL βάση όπου υπάρχουν σχέσεις στα δεδομένα, στην RealTime για να δημιουργηθούν σχέσεις στα αντικείμενα, συνήθως χρησιμοποιούνται αναγνωριστικά (μοναδικά ids) τα οποία βρίσκονται σε πολλές θέσεις στη βάση.



Εικόνα 4.17 Δομή βάσης

Παράλληλα, χρησιμοποιήθηκε η υπηρεσία hosting της Firebase για να φιλοξενηθεί η ιστοσελίδα στους servers της Google. Αυτή η υπηρεσία διευκολύνει την διαδικασία εύρεσης κάποιου server με σκοπό τη συντήρησή του αλλά και τη φιλοξενία της ιστοσελίδας. Η υπηρεσία αυτή δίνει την δυνατότητα επιστροφής σε προηγούμενη έκδοση της ιστοσελίδας με ένα κλικ.

Επιπροσθέτως, χρησιμοποιήθηκε η υπηρεσία storage της Firebase για την αποθήκευση των φωτογραφιών των χρηστών. Με την βοήθεια των storage rules, δίνεται η δυνατότητα περιορισμού των αρχείων που ανεβάζουν οι χρήστες (να γίνει αποφυγή μεταφόρτωσης μεγάλων αρχείων και αρχείων που δεν είναι φωτογραφίες). Αν και το storage είναι μια αρκετά αργή μέθοδος για ανάγνωση και καταγραφή δεδομένων, είναι πρακτικά η μόνη λύση για αποθήκευση μεγάλου όγκου δεδομένων όπως φωτογραφίες, βίντεο κ.α.

Για την διαχείριση των χρηστών χρησιμοποιήθηκε το Authentication της Firebase. Αυτή η υπηρεσία, αποτελεί ένα πολύ ισχυρό εργαλείο για αυτή την εργασία, το οποίο μπορεί να χρησιμοποιηθεί εύκολα από άλλες υπηρεσίες. Στο συγκεκριμένο παιχνίδι χρησιμοποιήθηκε η αυθεντικοποίηση μέσω email και κωδικού. Ωστόσο, υπάρχουν και άλλες μέθοδοι όπως μέσω:

- τηλεφώνου
- Google Account
- Facebook
- Twitter
- Microsoft Account
- Apple Account (κ.α).

Για να είναι ακόμη πιο ασφαλής, η εφαρμογή δεν μπορεί να χρησιμοποιηθεί από χρήστες που δεν έχουν επιβεβαιώσει το email τους. Η συγκεκριμένη λειτουργία υλοποιήθηκε ως ένας συνδυασμός RealTime Database rules και μεθόδων στο front end για την απόκρυψη σελίδων.

Η επικοινωνία με τον server μέσω των cloud functions γίνεται μέσω του Admin SDK της firebase το οποίο διαφέρει από το client SDK που χρησιμοποιήθηκε από την Angular για την ανάπτυξη της εφαρμογής. Το SDK δίνει πρόσβαση σε μεθόδους διαχείρισης όλων των προϊόντων της Firebase. Συγκεκριμένα, στην υλοποίηση του παιχνιδιού δημιουργήθηκαν οι εξής μέθοδοι:

- **AcceptOrDeclineFriend** (χρησιμοποιείται κάθε φορά που ένας παίκτης κάνει αποδοχή ή απόρριψη ενός αιτήματος φιλίας)
- **AnswerQuestion** (χρησιμοποιείται κάθε φορά που δίνεται μια απάντηση σε ερώτηση)
- **CheckForAfkPlayers** (καλείται από τους παίκτες, κατά την περίοδο αναμονής για ένα συγκεκριμένο χρονικό διάστημα, έτσι ώστε να γνωρίζει το σύστημα αν κάποιος δεν έχει ρίξει το ζάρι ενώ είναι η σειρά του)
- **CleanUpGames** (εκτελείται κάθε 30 λεπτά και «καθαρίζει» παιχνίδια που είναι ανενεργά, δηλαδή έχουν να ενημερωθούν πολύ ώρα)
- **CleanUpRoom** (εκτελείται όταν ένα δωμάτιο καταστρέφεται για να διαγράψει τους πόρους που χρησιμοποιούνταν)
- **DeleteAccount** (διαγράφει τον λογαριασμό του χρήστη και ό,τι πόρους χρησιμοποιούσε όπως: φωτογραφίες στο storage, εγγραφές στη βάση, φίλους κ.α)
- **GetReward** (καλείται από τον παίκτη που επιλέγει ανταμοιβή ή τιμωρία μετά την απάντηση μιας ερώτησης)
- **JoinRoom** (καλείται από τον παίκτη όταν θέλει να συνδεθεί σε ένα δωμάτιο)
- **OnStartGame** (καλείται από τον παίκτη που δημιούργησε ένα δωμάτιο και θέλει να ξεκινήσει ένα παιχνίδι)
- **RemoveFriend** (καλείται από έναν παίκτη όταν θέλει να διαγράψει έναν φίλο του)
- **RollDice** (εκτελείται όταν ένας παίκτης ρίχνει το ζάρι)
- **SearchUser** (καλείται από τον παίκτη με σκοπό να αναζητήσει άλλους παίκτες)

Αν και τα Cloud Functions καλύπτουν σχεδόν τις ανάγκες της εφαρμογής, υπήρξε μια μεγάλη δυσκολία στην ανάπτυξη του συστήματος ελέγχου της συνεδρίας για τα παιχνίδια. Πιο αναλυτικά, επειδή η Firebase δεν δίνει άμεση πρόσβαση στον server που αποτελεί το back end, δεν υπάρχει τρόπος να προγραμματιστούν γεγονότα μετά από κάποια ώρα. Ιδανικά, θα θέλαμε να μπορούμε σε κάθε γύρο (κάθε φορά που πρέπει ένας παίκτης να ρίξει το ζάρι) να ξεκινήσουμε μια αντίστροφη μέτρηση και όταν ολοκληρωθεί να εκτελεστεί μια λειτουργία. Αυτό, δεν γίνεται με την χρήση των cloud functions και ήταν ένα σοβαρό πρόβλημα που αντιμετώπισα. Λύθηκε με την μέθοδο checkForAfkPlayers η οποία καλείται από τους παίκτες και ελέγχει αν ο παίκτης που είναι η σειρά του δεν έχει ρίξει το ζάρι για κάποιο χρονικό διάστημα (4 λεπτά). Μόλις περάσει αυτό το διάστημα, ο παίκτης βγαίνει απότομα από το παιχνίδι και σημειώνεται ως ήττα.

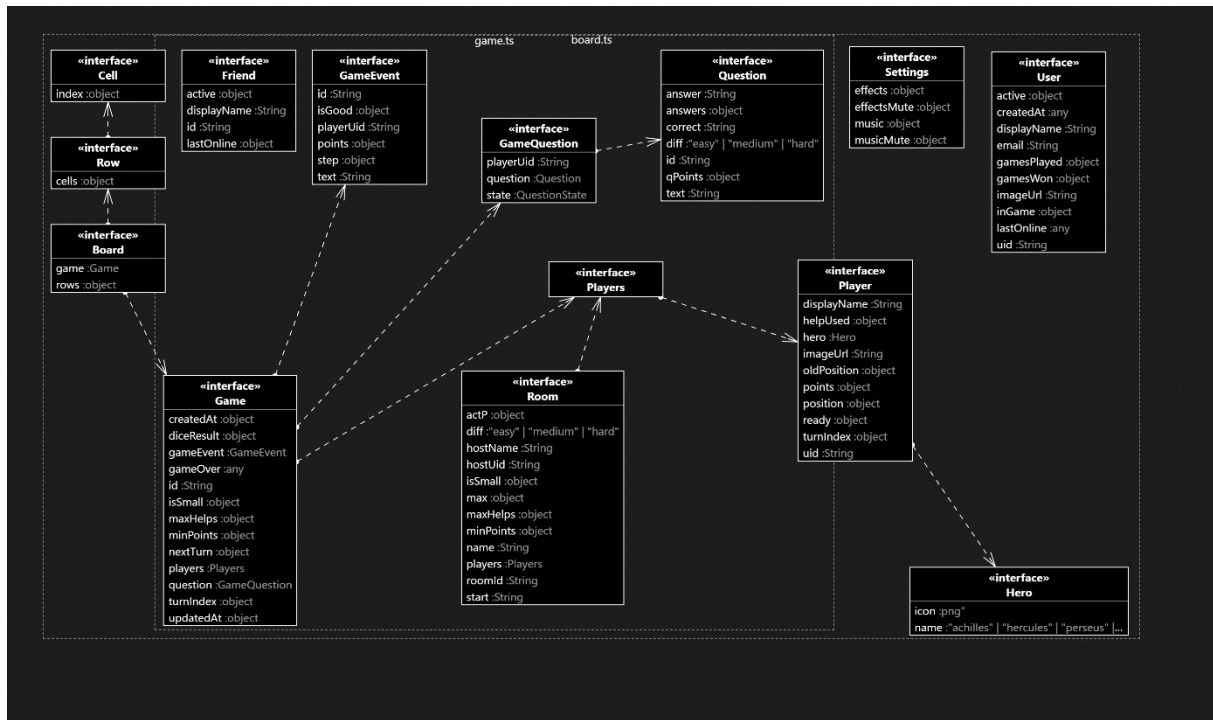
Η δομή των αρχείων για τις cloud functions υπάρχει στον φάκελο /functions και γίνεται προώθηση των αλλαγών στον server χρησιμοποιώντας το CLI (Command Line Interface) της Firebase. Η δομή έχει ως εξής: Υπάρχει ένα αρχείο index.ts που ορίζονται οι functions και για κάθε μια function υπάρχει ο χειριστής της (handler) στον υποφάκελο handlers. Επιπλέον, υπάρχουν και βοηθητικές function που εκτελούν διάφορες λειτουργίες.

Όσον αφορά το Front End, η χρήση της Angular έδωσε την δυνατότητα τμηματοποίησης του κώδικα σε ξεχωριστά components σύμφωνα με την λειτουργία τους. Τα components είναι:

- **DiceComponent** (εμφανίζει το ζάρι όταν ο παίκτης πατάει το κουμπί με το ζάρι)
- **FriendsComponent** (η σελίδα με τους φίλους)
- **GameComponent** (εδώ υπάρχει όλη η σελίδα που εμφανίζεται κατά την εκτέλεση ενός παιχνιδιού)
- **HomeComponent** (η αρχική σελίδα)
- **LoginComponent** (αυτή η σελίδα εμφανίζεται όταν ένας χρήστης δεν έχει συνδεθεί ή δημιουργεί λογαριασμό)
- **MainNavComponent** (αποτελεί το μενού στα αριστερά της οθόνης)
- **Modals**
 - **AlertModalComponent** (εμφανίζει ένα παράθυρο με ένα μήνυμα)
 - **ConfirmModalComponent** (εμφανίζει ένα παράθυρο με ένα μήνυμα και κουμπιά για επιβεβαίωση)
 - **DeleteAccountComponent** (εμφανίζει ένα παράθυρο για να βάλει ο χρήστης τα στοιχεία του για διαγράψει τον λογαριασμό του)
 - **FriendModalComponent** (εμφανίζει μια λίστα με τους φίλους για να στείλει πρόσκληση ο χρήστης)
 - **PlayersHudComponent** (εμφανίζει τους ενεργούς παίκτες σε ένα παιχνίδι, το σκορ τους και ποιον ήρωα έχουν)
 - **ProfileComponent** (η σελίδα του προφίλ χρήστη)
 - **QuestionModalComponent** (εμφανίζει το παράθυρο για την απάντηση ερώτησης, επιλογή ανταμοιβής ή τιμωρίας)
 - **RoomComponent** (εμφανίζεται όταν ο χρήστης βρίσκεται σε ένα δωμάτιο)
 - **RoomsComponent** (εμφανίζει τα δωμάτια που υπάρχουν στην εφαρμογή και κάποιες πληροφορίες τους, καθώς και ένα κουμπί για δημιουργία δωματίου)
 - **SinglePlayerComponent** (η σελίδα που ο χρήστης παίζει παιχνίδι με H/Y)
 - **VerifyComponent** (αυτή η σελίδα εμφανίζεται μόνο όταν ο χρήστης δεν έχει επιβεβαιώσει το email του)
 - **ProfileImgComponent** (ένα γενικό component για να εμφανίζει την εικόνα του χρήστη, χρησιμοποιείται σε διάφορα σημεία)

Για την λογική που υπάρχει στην εφαρμογή, χρησιμοποιήθηκαν Services της Angular για διάφορους σκοπούς. Αναλυτικότερα, τα Services της εφαρμογής είναι:

- **AudioService** (για τη διαχείριση ήχων του παιχνιδιού)
- **AuthService** (για σύνδεση, αποσύνδεση, εγγραφή, διαγραφή λογαριασμού καθώς και διαχείριση της εικόνας προφίλ)
- **FriendsService** (για την διαχείριση φίλων (διαγραφή, αποστολή αιτήματος φιλίας κ.α) και αναζήτηση παικτών)
- **GameService** (για την εκτέλεση του παιχνιδιού)
- **ModalsService** (για εμφάνιση παραθύρων (modals) για όλους τους σκοπούς)
- **PresenceService** (για την απόκτηση πληροφοριών της κατάστασης του χρήστη (ενεργός, ανενεργός))
- **QuestionService** (για τη διαχείριση των ερωτήσεων)
- **RoomService** (για την διαχείριση του δωματίου)
- **SinglePlayerService** (παρέχει την λογική για το SinglePlayerComponent)
- **StorageService** (για την μεταφόρτωση εικόνων)
- **TimerService** (για αντίστροφη μέτρηση με επιστροφή τιμών για διάφορους σκοπούς στην εκτέλεση του παιχνιδιού)



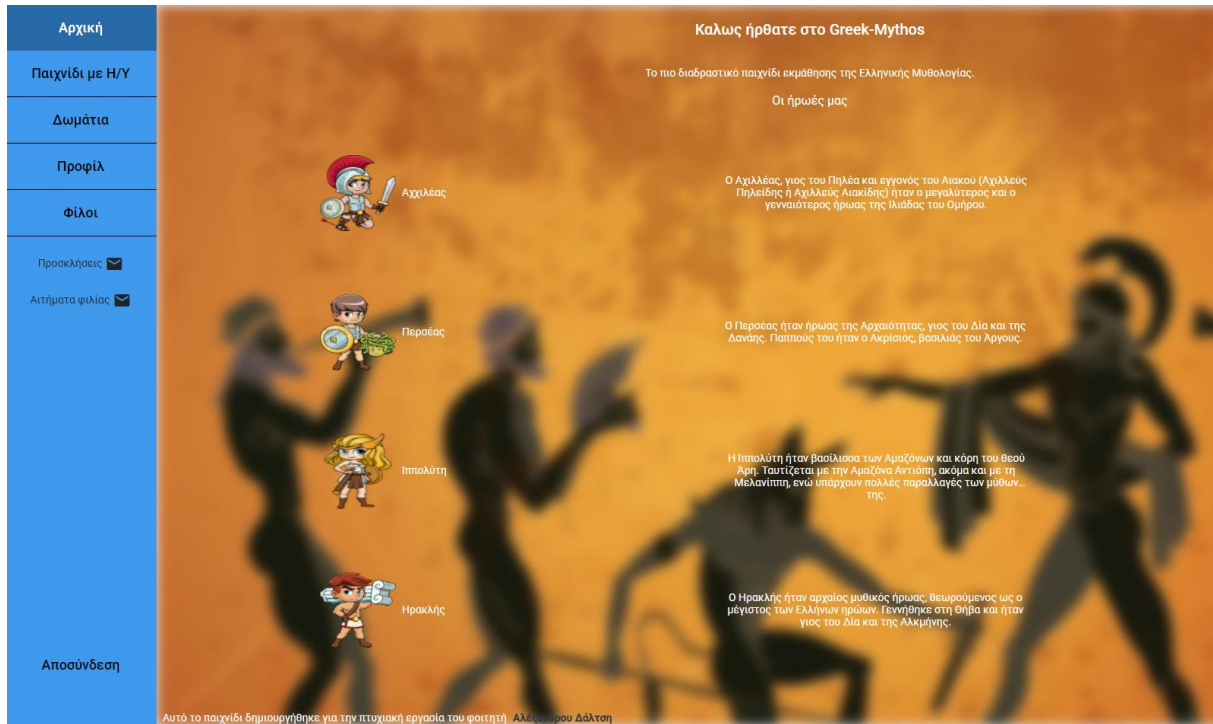
Εικόνα 4.18 Οι κλάσεις του παιχνιδιού

Κεφάλαιο 5ο: Περιγραφή παιχνιδιού, γραφικά και περιπτώσεις χρήσης

Το παιχνίδι με ονομασία “Greek Mythos” που δημιουργήθηκε στα πλαίσια της πτυχιακής εργασίας είναι ένα παιχνίδι γνώσεων και τύχης βασισμένο στην ελληνική μυθολογία. Δίνεται η δυνατότητα στον παίκτη να παίξει είτε σε ατομικό επίπεδο, έχοντας ως αντίπαλο τον Η/Υ, είτε ομαδικά με κάθε παιχνίδι να αποτελείται από δύο έως τέσσερις παίκτες. Παράλληλα, είναι διαθέσιμο σε δύο εκδοχές, τη μικρή με 49 θέσεις, και τη μεγάλη με 100. Το παιχνίδι αποτελείται από ένα διαδικτυακό ταμπλό το οποίο αποτελείται από ερωτήσεις και γεγονότα στα οποία πρέπει να απαντήσει ο εκάστοτε παίκτης με σκοπό να φτάσει στον τερματισμό. Σε κάθε ερώτηση, δίνεται η δυνατότητα χρήσης πάσου ή βοήθειας 50-50, η οποία αποκλείει τις δύο από τις τέσσερις πιθανές απαντήσεις. Αυτές οι βοήθειες δίνονται με ένα όριο το οποίο έχει οριστεί από τον ίδιο τον παίκτη κατά τη διαδικασία δημιουργίας του παιχνιδιού και η χρήση τους απαιτεί την απώλεια πόντων. Οι πόντοι, είναι εξίσου σημαντικοί με τον τερματισμό. Με κάθε σωστή απάντηση, οι παίκτες μαζεύουν πόντους τους οποίους μπορούν να χρησιμοποιήσουν για να λάβουν βοήθειες αλλά και για να μπορέσουν να τερματίσουν στο τέλος επιτυχώς. Σκοπός του παιχνιδιού δεν είναι μόνο ο τερματισμός αλλά και η συλλογή των πόντων που έχουν οριστεί στην αρχή του παιχνιδιού. Αν ένας παίκτης φτάσει στον τερματισμό και δεν έχει αρκετούς πόντους για να κερδίσει, τότε επιστρέφει κάποιες θέσεις πίσω έτσι ώστε να έχει τη δυνατότητα συλλογής πόντων.

5.1 Περιβάλλον εφαρμογής

Κατά την είσοδο στην εφαρμογή, το πρώτο πράγμα που εμφανίζεται είναι ένα παράθυρο που ζητάει από τον χρήστη να συνδεθεί. Για τη σύνδεση ενός χρήστη απαιτείται το email του καθώς και ο κωδικός πρόσβασης. Σε περίπτωση όμως που ο χρήστης επισκέπτεται πρώτη φορά την εφαρμογή, τότε θα πρέπει να δημιουργήσει ένα λογαριασμό πατώντας “Εγγραφή”. Εκεί εμφανίζεται μια φόρμα στην οποία πρέπει να συμπληρωθεί το email, το όνομα χρήστη, ο κωδικός πρόσβασης και η επιβεβαίωσή του.



Εικόνα 5.1 Περιβάλλον εφαρμογής

Αφού γίνει η εγγραφή ή η σύνδεση, ο χρήστης μεταφέρεται αυτόματα στο κύριο περιβάλλον της εφαρμογής και συγκεκριμένα στην Αρχική Σελίδα (Εικόνα 5.1). Αριστερά της σελίδας υπάρχει ένα μενού, το λεγόμενο μενού πλοήγησης. Σε αυτό βρίσκονται οι εξής ενότητες:

- Αρχική Σελίδα
- Παιχνίδι με Η/Υ
- Δωμάτια
- Προφίλ
- Φίλοι

Επιπλέον, ακριβώς από κάτω βρίσκονται και οι ειδοποιήσεις που αφορούν:

- Προσκλησεις
- Αιτήματα Φιλίας

Στο κάτω μέρος του μενού πλοήγησης υπάρχει το κουμπί της “Αποσύνδεσης”. Ο χρήστης, πατώντας το, μπορεί να βγει εντελώς από την εφαρμογή.

5.1.1 Αρχική σελίδα

Στην αρχική σελίδα, ο χρήστης καλωσορίζεται στο παιχνίδι. Παρουσιάζονται οι παίκτες (avatars) που μπορούν να χρησιμοποιηθούν ως “πίονια” καθώς και μια σύντομη περιγραφή για τον ρόλο τους στην ελληνική μυθολογία.

5.1.2 Παιχνίδι με Η/Υ

Αυτή η ενότητα αφορά το ατομικό παιχνίδι. Εδώ ο παίκτης, που θα παίξει με αντίπαλο τον Η/Υ, δύναται να επιλέξει ποια εκδοχή του παιχνιδιού θέλει να παίξει (μικρό ή μεγάλο), ποιος θα είναι ο αριθμός των πόντων που θα πρέπει να συλλεχθούν για να τερματιστεί το παιχνίδι καθώς επίσης και το όριο της βοήθειας 50-50 και του πάσο (Εικόνα 5.2).

The screenshot shows a settings screen for a game. The title is "Το παιχνίδι σου". Below the title, there are four sections:

- Μέγεθος παιχνιδιού:** Two radio buttons, "Μικρό" (unselected) and "Μεγάλο" (selected).
- Πόντοι-Στόχος *:** A text input field containing the number "200".
- Μέγιστος αριθμός βοηθειών:** A text input field containing the number "3".
- Βοήθεια πάσο *:** A text input field containing the number "2".

At the bottom center, there is a button labeled "Παίξε".

Εικόνα 5.2 Παιχνίδι με Η/Υ

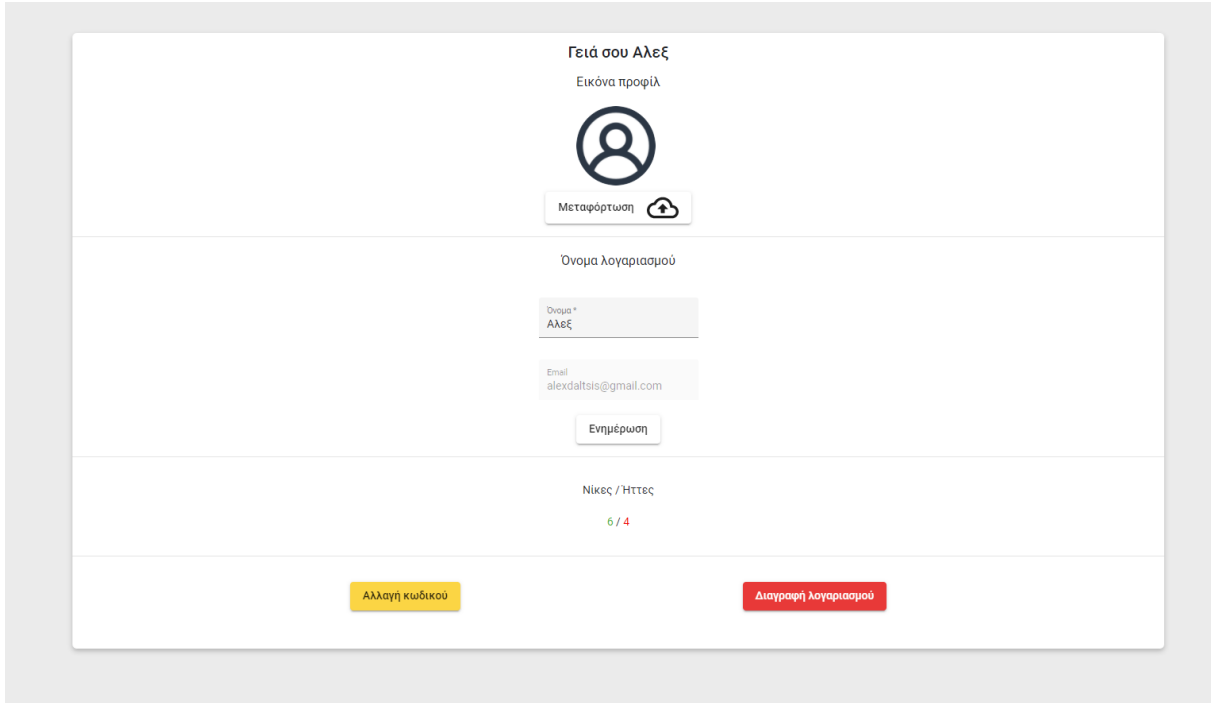
5.1.3 Δωμάτια

Επιλέγοντας την ενότητα “Δωμάτια”, ο παίκτης επιλέγει να παίξει ένα ομαδικό παιχνίδι. Εμφανίζονται όλα τα διαθέσιμα δωμάτια στα οποία μπορεί να εισέλθει και να παίξει. Για κάθε δωμάτιο υπάρχουν όλες οι πληροφορίες που πρέπει να γνωρίζει ένας χρήστης (πόντοι-στόχος, βοήθειες, μέγεθος παιχνιδιού) προκειμένου να μπορεί να αποφασίσει αν θέλει να παίξει με αυτούς τους παίκτες. Υπάρχει επίσης και η δυνατότητα δημιουργίας δωματίου. Φτιάχνοντας ένα δωμάτιο, ο χρήστης ως διαχειριστής, ορίζει το όνομα του δωματίου, το μέγεθός του καθώς και τον αριθμό πόντων και βοηθειών, όπως και σε ένα ατομικό παιχνίδι. Σε αυτό που διαφέρει όμως είναι στην επιλογή αριθμού παικτών. Σε κάθε ομαδικό παιχνίδι το όριο παικτών κυμαίνεται μεταξύ δύο και τεσσάρων παικτών.

Μόλις δημιουργηθεί το δωμάτιο, εμφανίζεται ένα νέο παράθυρο που το απεικονίζει και στο οποίο ο διαχειριστής μπορεί να προσκαλέσει φίλους του από το παιχνίδι για να παίξουν μαζί. Αν δε θέλει να το κάνει αυτό, τότε περιμένει να εισέλθουν στο δωμάτιο παίκτες. Όποιον παίκτη δεν επιθυμεί να έχει στο δωμάτιό του, μπορεί να τον διώξει. Από τη στιγμή που θα μπουν όλοι οι παίκτες πρέπει να πατήσουν το κουμπί “Ετοιμος” και ο διαχειριστής το “Ξεκίνησε” για να αρχίσει το παιχνίδι.

5.1.4 Προφίλ

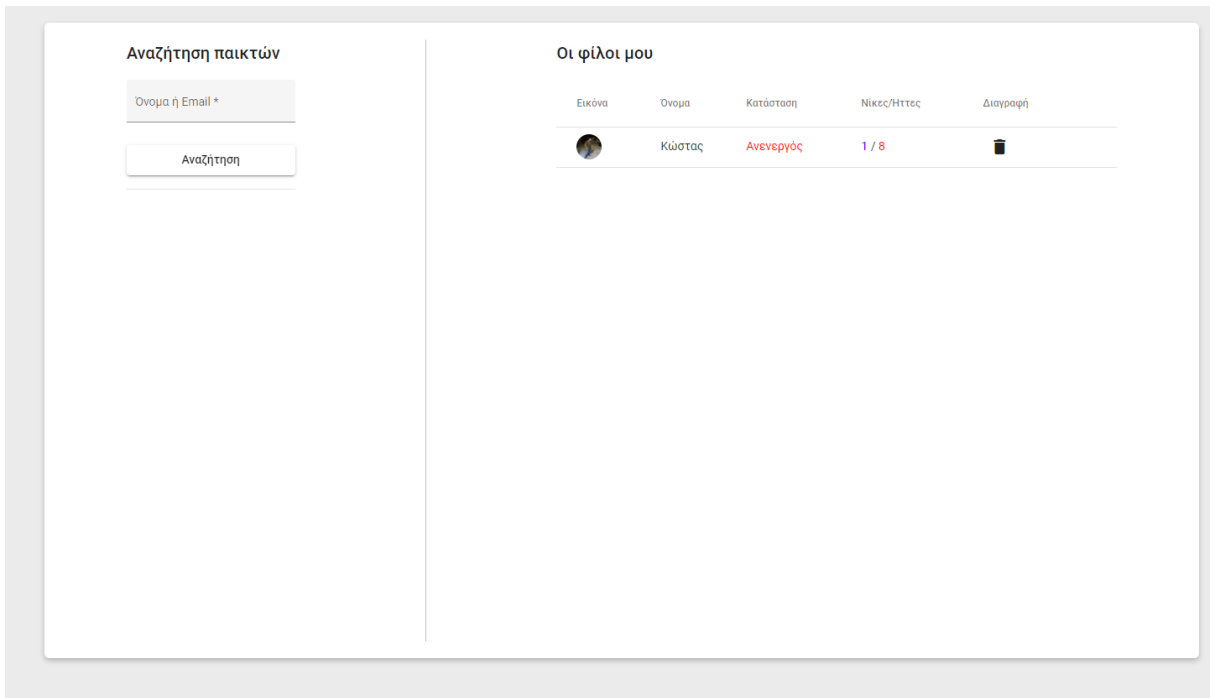
Εδώ εμφανίζονται όλα τα στοιχεία του χρήστη. Ο παίκτης μπορεί να ανεβάσει ή να αλλάξει τη φωτογραφία προφίλ του καθώς επίσης και το όνομα χρήστη. Επίσης, εμφανίζονται τα στατιστικά του στο παιχνίδι, δηλαδή πόσα παιχνίδια έχει κερδίσει και πόσα έχει χάσει.



Εικόνα 5.3 Σελίδα Προφίλ

5.1.5 Σελίδα φίλων

Στην τελευταία ενότητα του μενού πλοήγησης, ο χρήστης μπορεί να βρει τους φίλους του. Υπάρχει η δυνατότητα αναζήτησης φίλων μέσω email ή όνομα χρήστη. Δίπλα ακριβώς υπάρχει η λίστα όλων των φίλων του και η δυνατότητα διαγραφής τους οποιαδήποτε στιγμή. Ακριβώς κάτω από τις ενότητες στο μενού υπάρχουν και δύο ειδών ειδοποιήσεις. Οι πρώτες αφορούν τις προσκλήσεις σε κάποιο δωμάτιο και οι δεύτερες αιτήματα φιλίας τα οποία μπορεί να αποδεχτεί ή να απορρίψει (Εικόνα 5.4).



Εικόνα 5.4 Σελίδα Φίλων

5.2 Εκτέλεση παιχνιδιού

Αφού ο παίκτης έχει επιλέξει αν θα παίξει ατομικά ή ομαδικά, το παιχνίδι μπορεί να ξεκινήσει. Εμφανίζεται ένα ζάρι και ένα διαδικτυακό ταμπλό το οποίο αποτελείται από 49 ή 100 θέσεις (ανάλογα το μέγεθος που έχει οριστεί). Κάποιες από τις θέσεις αυτές έχουν την επιγραφή **?** και παραπέμπουν σε ερώτηση (Εικόνα 5.5), ενώ κάποιες άλλες έχουν την επιγραφή **!** και παραπέμπουν σε γεγονότα. Οι ερωτήσεις είναι χαμηλού, μέτριου και υψηλού βαθμού δυσκολίας σε όλες τις παρτίδες. Αυτό γίνεται για να προσαρμόζεται σε κάθε παίκτη και τις ανάγκες του και να μπορεί να τον διευκολύνει αλλά και να τον δυσκολέψει ταυτόχρονα κάνοντας το παιχνίδι πιο διασκεδαστικό.

Όπως προαναφέρθηκε, υπάρχει η δυνατότητα του πάσο, με όριο που έχει οριστεί, και το οποίο δίνει την δυνατότητα στον παίκτη αν έχει αρκετούς πόντους, να “πληρώσει” ώστε να πάρει πάσο και να προχωρήσει σε περίπτωση που δε γνωρίζει την απάντηση της ερώτησης που του έτυχε. Με τον ίδιο τρόπο παρέχεται στον παίκτη και η βοήθεια 50-50, η οποία αποκλείει τις δύο από τις τέσσερις πιθανές απαντήσεις της ερώτησης. Για την απάντηση κάθε ερώτησης υπάρχει το χρονικό όριο των δεκαπέντε δευτερολέπτων.



Εικόνα 5.5 Παράθυρο ερώτησης

Αν η ερώτηση απαντηθεί σωστά, τότε ο παίκτης επιλέγει αν θα προχωρήσει μπροστά ή θα πάρει πόντους. Ανάλογα με το πόσο δύσκολη ήταν η ερώτηση (εύκολη - μέτρια - δύσκολη) προχωρά ένα έως τρία βήματα μπροστά ή παίρνει ανάλογους πόντους. Αν όμως απαντήσει λάθος, τότε και πάλι επιλέγει αν θα χάσει πόντους ή θα πάει πίσω στη θέση που βρισκόταν πριν ρίξει το ζάρι.

Τα γεγονότα από την άλλη, είναι καταστάσεις στις οποίες μπαίνει ο ήρωας (παίκτης) και οι οποίες έχουν είτε καλή είτε κακή έκβαση. Ανάλογα με το μήνυμα που θα εμφανιστεί, ο παίκτης μπορεί να κερδίσει ή να χάσει πόντους όπως επίσης και να προχωρήσει μπροστά στο ταμπλό ή να πάει πίσω.

5.2.1 Παιχνίδι με Η/Υ ή ατομικό παιχνίδι

Στο ατομικό παιχνίδι, ο παίκτης παίζει με αντίπαλο τον υπολογιστή. Η ροή του παιχνιδιού στο ταμπλό είναι η ίδια με το ομαδικό παιχνίδι με τη μόνη διαφορά να αφορά τις ερωτήσεις. Προκειμένου να υπάρχει διαφάνεια, οι απαντήσεις που θα δίνονται στις ερωτήσεις από τον υπολογιστή θα είναι τυχαίες μέσα από ένα σύστημα τυχαιότητας απάντησης, έτσι ώστε να μην απαντάει πάντα ο υπολογιστής σωστά και κερδίζει το παιχνίδι. Το ίδιο θα συμβαίνει και στην επιλογή πόντων ή κίνησης όταν απαντάει είτε σωστά είτε λάθος. Επιπρόσθετα, ο υπολογιστής δε θα έχει τη δυνατότητα χρήσης βοήθειας παρά μόνο ο παίκτης.

5.2.2 Ομαδικό παιχνίδι

Το ομαδικό παιχνίδι ξεκινά όταν έχει μπει ο απαιτούμενος αριθμός παικτών (δύο έως τέσσερις) σε ένα δωμάτιο. Η ροή είναι η ίδια και εδώ. Κάθε παίκτης και πάλι έχει όριο δεκαπέντε δευτερόλεπτα για να απαντήσει στην ερώτηση που το έτυχε. Σε περίπτωση που ένας παίκτης δεν κάνει καμία κίνηση μέσα στο παιχνίδι, όπως να ρίξει το ζάρι, να απαντήσει σε κάποια ερώτηση ή να χρησιμοποιήσει κάποια βοήθεια, για πάνω από δύο λεπτά, τότε αυτόματα βγαίνει από το παιχνίδι και σημειώνεται ως ήττα. Στην περίπτωση που φύγουν από ένα παιχνίδι όλοι οι παίκτες και μένει μόνο ένας, τότε το παιχνίδι ακυρώνεται.

5.3 Γραφικά

Τα γραφικά που χρησιμοποιήθηκαν στο συγκεκριμένο παιχνίδι αποτελούνται από απλά στοιχεία της HTML και η διαμόρφωσή τους από στυλ της CSS. Οι εικόνες των ηρώων που χρησιμοποιούν οι παίκτες ως “πιόνια” στο ταμπλό προέρχονται από το www.friendlystock.com. Επίσης κάποια άλλα εικονίδια που χρησιμοποιήθηκαν (SVGs) πάρθηκαν από το <https://freeicons.io/>. Η μουσική του παιχνιδιού είναι δικής μου δημιουργίας.

5.4 Περιπτώσεις χρήσης

Οι περιπτώσεις χρήσεις είναι ένα σύνολο σεναρίων που συνδέονται με έναν συγκεκριμένο σκοπό του χρήστη. Έχουν ως στόχο:

- να καθορίσουν και να περιγράψουν τις λειτουργικές απαιτήσεις του συστήματος
- να δώσουν μια σαφή και συνεπή περιγραφή για το τι θα πρέπει να κάνει το σύστημα
- να παρέχουν την κατάλληλη βάση για να γίνονται έλεγχοι για επαλήθευση του συστήματος
- να παρέχουν την ικανότητα να εντοπίζονται οι λειτουργικές απαιτήσεις μέσα στις κλάσεις και τις λειτουργίες του συστήματος.

Πίνακας 5.1 PX 1 - Δημιουργία λογαριασμού

Όνομα: PX1 - Δημιουργία Λογαριασμού

Πρωταγωνιστής: Χρήστης

Στόχος: Εγγραφή χρήστη

Προσυνθήκες: -

Βήματα:

1. Εμφάνιση φόρμας δημιουργίας λογαριασμού
2. Συμπλήρωση στοιχείων
3. Ο χρήστης πατάει το κουμπί “Εγγραφή”
4. Δημιουργία λογαριασμού
5. Αυτόματη σύνδεση
6. Μετάβαση στην αρχική σελίδα

Εναλλακτικές:

1. Ο χρήστης συμπλήρωσε λάθος email
2. Εμφάνιση μηνύματος “Λάθος μορφή email”
3. Ο χρήστης συμπλήρωσε λάθος όνομα
4. Εμφάνιση μηνύματος “Το όνομα πρέπει να είναι μεταξύ 3 και 40 χαρακτήρων”
5. Ο χρήστης συμπλήρωσε μικρό κωδικό
6. Εμφάνιση μηνύματος “Ο κωδικός πρέπει να είναι μεγαλύτερος από 8 χαρακτήρες”
7. Ο χρήστης πληκτρολόγησε δύο διαφορετικούς κωδικούς

8. Εμφάνιση μηνύματος “Οι δύο κωδικοί πρέπει να είναι ίδιοι”
9. Ο χρήστης πληκτρολόγησε λάθος στοιχεία
10. Εμφάνιση μηνύματος “Λάθος στοιχεία”
11. Ο χρήστης δεν έχει επιβεβαιώσει τον λογαριασμό του
12. Εμφάνιση σελίδας με μήνυμα “Ο λογαριασμός σας δεν είναι ενεργοποιημένος”

Πίνακας 5.2 ΠΧ 2 - Σύνδεση

Όνομα: ΠΧ2 - Σύνδεση

Πρωταγωνιστής: Χρήστης

Στόχος: Σύνδεση χρήστη

Προσυνθήκες: ΠΧ1

Βήματα:

1. Εμφάνιση φόρμας σύνδεσης
2. Συμπλήρωση στοιχείων
3. Ο χρήστης πατάει το κουμπί “Σύνδεση”
4. Ολοκλήρωση σύνδεσης
5. Μετάβαση στην αρχική σελίδα

Εναλλακτικές:

6. Ο χρήστης συμπλήρωσε λάθος email, κωδικό ή δε συμπλήρωσε όλα τα πεδία
7. Εμφάνιση μηνύματος “Λάθος στοιχεία”
8. Ο χρήστης δεν έχει επιβεβαιώσει τον λογαριασμό του
9. Εμφάνιση σελίδας με μήνυμα “Ο λογαριασμός σας δεν είναι ενεργοποιημένος”

Πίνακας 5.3 ΠΧ 3 - Παιχνίδι με Η/Υ

Όνομα: ΠΧ3 - Παιχνίδι με Η/Υ

Πρωταγωνιστής: Χρήστης

Στόχος: Ατομικό παιχνίδι

Προσυνθήκες: ΠΧ2

Βήματα:

1. Ο χρήστης πατάει το κουμπί “Παιχνίδι με Η/Υ” στο μενού πλοήγησης
2. Εμφάνιση φόρμας συμπλήρωσης ρυθμίσεων παιχνιδιού
3. Συμπλήρωση ρυθμίσεων παιχνιδιού
4. Ο χρήστης πατάει το κουμπί “Παίξε”
5. Εκκίνηση παιχνιδιού
6. Εμφάνιση ταμπλό και γραφικών παιχνιδιού
7. Ρίψη ζαριού

8. Εμφάνιση αποτελέσματος ζαριού
9. Κίνηση παίκτη
10. Εμφάνιση ερώτησης ανάλογα τη ζαριά
 - a. Απάντηση ερώτησης εντός 15 δευτερολέπτων
 - b. Ο χρήστης απάντησε σωστά
 - c. Ο χρήστης επιλέγει ανταμοιβή
 - d. Ο χρήστης προχωράει στο ταμπλό ή κερδίζει πόντους
11. Εμφάνιση γεγονότος ανάλογα τη ζαριά
 - a. Ο χρήστης περιμένει 5 δευτερόλεπτα για να κλείσει το παράθυρο γεγονότος
 - b. Ο χρήστης κινείται, κερδίζει ή χάνει πόντους ανάλογα με το γεγονός
12. Ο χρήστης περιμένει να παίξει ο υπολογιστής
13. Επανάληψη βημάτων 7 έως 12
14. Ο παίκτης που φτάνει πρώτος στον τερματισμό με τους πόντους που χρειάζονται, κερδίζει.
15. Σταματάει η ροή της περίπτωσης χρήσης.

Εναλλακτικές:

16. Ο χρήστης δε συμπλήρωσε όλα τα πεδία
 17. Εμφάνιση μηνύματος “Δεν έχεις συμπληρώσει όλα τα στοιχεία”
 18. Αποτυχία απάντησης εντός 15 δευτερολέπτων ή λάθος απάντηση
 19. Εμφάνιση σωστής απάντησης
 20. Ο χρήστης επιλέγει την τιμωρία του μεταξύ απώλειας πόντων ή πίσω βημάτων στο ταμπλό
 21. Ο παίκτης φτάνει στον τερματισμό χωρίς τους πόντους που χρειάζονται
- Ο παίκτης επιστρέφει κάποια βήματα πίσω για να συλλέξει πόντους

Πίνακας 5.4 ΠΧ 4 - Δημιουργία Δωματίου

Όνομα: ΠΧ4 - Δημιουργία Δωματίου

Πρωταγωνιστής: Χρήστης

Στόχος: Δημιουργία Δωματίου

Προσυνθήκες: ΠΧ2

Βήματα:

1. Ο χρήστης πατάει το κουμπί “Δωμάτια” από το μενού πλοήγησης
2. Εμφάνιση καρτών δωματίων και δημιουργίας νέου
3. Ο χρήστης πατάει το κουμπί “+” για να δημιουργήσει ένα δωμάτιο
4. Εμφάνιση φόρμας συμπλήρωσης στοιχείων και ρυθμίσεων δωματίου
5. Ο χρήστης πατάει το κουμπί “Δημιουργία”
6. Μετάβαση στο δωμάτιο (ΠΧ6)

Εναλλακτικές:

1. Ο χρήστης πατάει το κουμπί “Δημιουργία” ενώ βρίσκεται ήδη σε παιχνίδι
2. Εμφάνιση μηνύματος “Βρίσκεσαι ήδη σε παιχνίδι”
3. Ο χρήστης δεν έχει συμπληρώσει όλα τα στοιχεία
4. Εμφάνιση μηνύματος “Δεν έχεις συμπληρώσει όλα τα στοιχεία”

Πίνακας 5.5 ΠΧ 5 - Σύνδεση σε δωμάτιο

Όνομα: ΠΧ5 - Σύνδεση σε δωμάτιο

Πρωταγωνιστής: Χρήστης

Στόχος: Σύνδεση χρήστη σε δωμάτιο

Προσυνθήκες: ΠΧ2

Βήματα:

1. Ο χρήστης πατάει το κουμπί “Δωμάτια” από το μενού πλοήγησης
2. Εμφάνιση καρτών δωματίων και δημιουργίας νέου
3. Ο χρήστης επιλέγει δωμάτιο και πατάει το κουμπί “Σύνδεση”
4. Εμφάνιση μηνύματος “Σύνδεση...”
5. Μετάβαση στο δωμάτιο (ΠΧ6)

Εναλλακτικές:

1. Υπάρχει σφάλμα.
2. Εμφάνιση μηνύματος “Κάτι πήγε στραβά!”
3. Δεν υπάρχει χώρος στο δωμάτιο
4. Εμφάνιση μηνύματος “Δεν υπάρχει χώρος στο δωμάτιο!”
5. Ο χρήστης βρίσκεται ήδη σε παιχνίδι
6. Εμφάνιση μηνύματος “Βρίσκεσαι ήδη σε παιχνίδι”
7. Ο χρήστης έχει ήδη συνδεθεί στο δωμάτιο
8. Μετάβαση στο δωμάτιο (ΠΧ6)

Πίνακας 5.6 ΠΧ 6 - Περιβάλλον δωματίου

Όνομα: ΠΧ6 - Περιβάλλον Δωματίου

Πρωταγωνιστής: Χρήστης και υπόλοιποι παίκτες

Στόχος: Έναρξη παιχνιδιού

Προσυνθήκες: ΠΧ2,ΠΧ4 ή ΠΧ5

Βήματα:

1. Ο χρήστης πατάει το κουμπί “Πρόσκληση”
2. Εμφάνιση λίστας φίλων
3. Ο χρήστης επιλέγει φίλο και πατάει το εικονίδιο της αποστολής
4. Αν ο χρήστης είναι διαχειριστής του δωματίου, πατάει το κουμπί “Διώξε” για να διώξει κάποιον παίκτη αν επιθυμεί
5. Οι παίκτες πατούν το κουμπί “Ετοιμος”
6. Ο διαχειριστής πατάει το κουμπί “Ξεκίνησε”
7. Μετάβαση στο παιχνίδι (ΠΧ7)

Εναλλακτικές:

1. Ο χρήστης πατάει το κουμπί “Πρόσκληση” ενώ δεν έχει φίλους
2. Εμφάνιση μηνύματος “Δεν έχεις φίλους να προσκαλέσεις”
3. Ο χρήστης πατάει το κουμπί “Πρόσκληση” ενώ το δωμάτιο είναι γεμάτο
4. Εμφάνιση μηνύματος “Δεν μπορείς να προσκαλέσεις! Το δωμάτιο είναι γεμάτο.”
5. Ο διαχειριστής πατάει το κουμπί “Ξεκίνησε” ενώ δεν έχει μπει άλλος παίκτης στο δωμάτιο
6. Εμφάνιση μηνύματος “Δεν μπορείς να ξεκινήσεις το παιχνίδι με 1 άτομο”
7. Ο διαχειριστής πατάει το κουμπί “Ξεκίνησε” ενώ δεν είναι όλοι οι παίκτες έτοιμοι
8. Εμφάνιση μηνύματος “Δεν μπορείς να ξεκινήσεις αν δεν είναι όλοι έτοιμοι”
9. Ο χρήστης πατάει το κουμπί “Εξοδος”
10. Εμφάνιση παραθύρου για επιβεβαίωση
11. Ο χρήστης επιλέγει “Όχι”
12. Μετάβαση στο δωμάτιο
13. Ο χρήστης επιλέγει “Ναι”
14. Μετάβαση στη λίστα δωματίων
15. Ο διαχειριστής πατάει το κουμπί “Εξοδος”
16. Επανάληψη των βημάτων 9 έως 10 ή 11 έως 12
17. Διαγραφή δωματίου
18. Μετάβαση στη λίστα δωματίων.

Πίνακας 5.7 ΠΧ 7 - Ομαδικό παιχνίδι

Όνομα: ΠΧ7 - Ομαδικό παιχνίδι

Πρωταγωνιστής: Χρήστης

Στόχος: Εκτέλεση παιχνιδιού

Προσυνθήκες: ΠΧ2, ΠΧ6

Βήματα:

1. Εκκίνηση παιχνιδιού
2. Εμφάνιση ταμπλό και γραφικών παιχνιδιού
3. Ρίψη ζαριού
4. Εμφάνιση αποτελέσματος ζαριού
5. Κίνηση χρήστη
6. Εμφάνιση ερώτησης ανάλογα τη ζαριά
 - a. Απάντηση ερώτησης εντός 15 δευτερολέπτων
 - b. Ο χρήστης απάντησε σωστά
 - c. Ο χρήστης επιλέγει ανταμοιβή
 - d. Ο χρήστης προχωράει στο ταμπλό ή κερδίζει πόντους
7. Εμφάνιση γεγονότος ανάλογα τη ζαριά
 - a. Ο χρήστης περιμένει 5 δευτερόλεπτα για να κλείσει το παράθυρο γεγονότος
 - b. Ο χρήστης κινείται, κερδίζει ή χάνει πόντους ανάλογα με το γεγονός
8. Ο χρήστης περιμένει να παίξουν οι υπόλοιποι παίκτες
9. Επανάληψη βημάτων 3 έως 7
10. Ο παίκτης που φτάνει πρώτος στον τερματισμό με τους πόντους που χρειάζονται, κερδίζει.
11. Εμφάνιση μηνύματος “Ο ... κέρδισε!”
12. Ο χρήστης πατάει το κουμπί “Εξοδος”
13. Σταματάει η ροή της ΠΧ

Εναλλακτικές:

1. Αποτυχία απάντησης εντός 15 δευτερολέπτων ή λάθος απάντηση
2. Εμφάνιση σωστής απάντησης
3. Ο χρήστης επιλέγει την τιμωρία του μεταξύ απώλειας πόντων ή πίσω βημάτων στο ταμπλό
4. Ο χρήστης φτάνει στον τερματισμό χωρίς τους πόντους που χρειάζονται
5. Ο χρήστης επιστρέφει κάποια βήματα πίσω για να συλλέξει πόντους
6. Ο χρήστης δεν κάνει κάποια κίνηση για χρονικό διάστημα μεγαλύτερο των 2 λεπτών
7. Ο χρήστης αποβάλλεται από το παιχνίδι
8. Ο χρήστης θεωρείται ηττημένος
9. Όλοι οι χρήστες πλην ενός, αποχωρούν από το παιχνίδι
10. Το παιχνίδι ακυρώνεται, διακόπτεται η περίπτωση χρήσης

Πίνακας 5.8 ΠΧ 8 - Προφίλ χρήστη

Όνομα: ΠΧ8 - Προφίλ χρήστη

Πρωταγωνιστής: Χρήστης

Στόχος: Επεξεργασία προφίλ χρήστη

Προσυνθήκες: ΠΧ2

Βήματα:

1. Ο χρήστης πατάει το κουμπί “Προφίλ” από το μενού πλοήγησης
2. Εμφάνιση κάρτας προφίλ χρήστη
3. Ο χρήστης πατάει το κουμπί “Μεταφόρτωση” για να ανεβάσει ή να αλλάξει εικόνα προφίλ
4. Ο χρήστης αλλάζει το όνομα λογαριασμού

5. Εμφανίζονται τα στατιστικά του χρήστη αναφορικά με τα παιχνίδια που έχει παίξει
6. Ο χρήστης πατάει το κουμπί “Ενημέρωση”
7. Επιστροφή στο ανανεωμένο προφιλ
8. Ο χρήστης πατάει το κουμπί “Αλλαγή κωδικού”
9. Αποστέλλεται ένα email στον χρήστη για αλλαγή κωδικού

Εναλλακτικές

1. Ο χρήστης δεν έχει συμπληρώσει ή έχει συμπληρώσει λάθος όνομα χρήστη
2. Εμφάνιση μηνύματος “Το όνομα πρέπει να είναι μεταξύ 3 και 40 χαρακτήρων”
3. Ο χρήστης πατάει το κουμπί “Διαγραφή λογαριασμού”
4. Διαγράφεται το προφιλ χρήστη
5. Σταματάει η περίπτωση χρήσης

Πίνακας 5.9 ΠΧ 9 - Φίλοι

Όνομα: ΠΧ9 - Φίλοι

Πρωταγωνιστής: Χρήστης

Στόχος: Αναζήτηση φίλων

Προσυνθήκες: ΠΧ2

Βήματα:

1. Ο χρήστης πατάει το κουμπί “Φίλοι” από το μενού πλοήγησης
2. Εμφάνιση λίστας φίλων του χρήστη και αναζήτησης παικτών
3. Ο χρήστης πληκτρολογεί όνομα ή email
4. Ο χρήστης πατάει το κουμπί “Αναζήτηση”
5. Ο χρήστης πατάει το κουμπί της προσθήκης φίλου
6. Εμφάνιση μηνύματος “Στάλθηκε η πρόσκληση”
7. Ο χρήστης ελέγχει αν είναι ενεργοί ή όχι οι φίλοι του από τη λίστα φίλων
8. Ο χρήστης ελέγχει αν οι φίλοι βρίσκονται σε παιχνίδι
9. Ο χρήστης ελέγχει τις νίκες και τις ήττες των φίλων του
10. Ο χρήστης πατάει το κουμπί “Διαγραφή” για ένα συγκεκριμένο φίλο
11. Εμφάνιση μηνύματος “Η διαγραφή πέτυχε”
12. Επιστροφή στη λίστα φίλων

Εναλλακτικές:

1. Ο χρήστης πληκτρολογεί μικρό όνομα ή email στην αναζήτηση
2. Εμφάνιση μηνύματος “Εισάγετε από 2 μέχρι 40 χαρακτήρες”
3. Ο χρήστης πληκτρολογεί λάθος όνομα ή email στην αναζήτηση
4. Εμφάνιση μηνύματος “Δεν βρέθηκαν χρήστες”
5. Σφάλμα κατά την αναζήτηση
6. Εμφάνιση μηνύματος “Πρόβλημα στην αναζήτηση”
7. Εντοπίστηκε σφάλμα
8. Εμφάνιση μηνύματος “Κάτι πήγε στραβά”
9. Σφάλμα κατά την πρόσκληση φίλου
10. Εμφάνιση μηνύματος “Κάτι πήγε στραβά”
11. Ο χρήστης αναζήτησε έναν ήδη υπάρχον φίλο του
12. Εμφάνιση μηνύματος “Είστε ήδη φίλοι”
13. Ο χρήστης ελέγχει ότι κάποιος φίλος του δεν έχει παίξει κανένα παιχνίδι
14. Εμφάνιση μηνύματος “Δεν υπάρχουν παιχνίδια”
15. Σφάλμα κατά τη διαγραφή
16. Εμφάνιση μηνύματος “Κάτι πήγε στραβά”

Πίνακας 5.10 ΠΧ 10 - Προσκλήσεις

<p>Όνομα: ΠΧ10 - Προσκλήσεις Πρωταγωνιστής: Χρήστης Στόχος: Απάντηση σε προσκλήσεις φίλων Προσυνθήκες: ΠΧ2 Βήματα:</p> <ol style="list-style-type: none">1. Ο χρήστης πατάει το κουμπί “Προσκλήσεις” από το μενού πλοήγησης2. Εμφάνιση σελίδας προσκλήσεων3. Ο χρήστης αποδέχεται την πρόσκληση πατώντας το κουμπί “Αποδοχή” <p>Εναλλακτικές:</p> <ol style="list-style-type: none">1. Ο χρήστης απορρίπτει την πρόσκληση2. Εμφάνιση μηνύματος “Απορρίφθηκε”3. Ο χρήστης δεν έχει προσκλήσεις4. Εμφάνιση μηνύματος “Δεν υπάρχουν προσκλήσεις”
--

Πίνακας 5.11 ΠΧ 11 - Αιτήματα φιλίας

<p>Όνομα: ΠΧ11 - Αιτήματα φιλίας Πρωταγωνιστής: Χρήστης Στόχος: Απάντηση σε αιτήματα φιλίας Προσυνθήκες: ΠΧ2 Βήματα:</p> <ol style="list-style-type: none">1. Ο χρήστης πατάει το κουμπί “Αιτήματα φιλίας” από το μενού πλοήγησης2. Εμφάνιση σελίδας αιτημάτων φιλίας3. Εμφάνιση μηνύματος “Ο .. σου έστειλε αίτημα φιλίας”4. Ο χρήστης πατάει “Αποδοχή”5. Εμφάνιση μηνύματος “Έγινες φίλος με τον...”6. Επιστροφή στα αιτήματα φιλίας7. Ο χρήστης πατάει “Επιστροφή”8. Επιστροφή στα αιτήματα φιλίας <p>Εναλλακτικές:</p> <ol style="list-style-type: none">1. Σφάλμα κατά την αποδοχή αιτήματος φιλίας2. Εμφάνιση μηνύματος “Κάτι πήγε στραβά”3. Ο χρήστης πατάει το κουμπί “Απόρριψη”4. Το αίτημα φιλίας απορρίπτεται5. Σφάλμα κατά την απόρριψη αιτήματος φιλίας6. Εμφάνιση μηνύματος “Κάτι πήγε στραβά”
--

Πίνακας 5.12 ΠΧ 12 - Αποσύνδεση

<p>Όνομα: ΠΧ12 - Αποσύνδεση Πρωταγωνιστής: Χρήστης Στόχος: Αποσύνδεση από το παιχνίδι Προσυνθήκες: ΠΧ2 Βήματα:</p> <ol style="list-style-type: none">1. Ο χρήστης πατάει το κουμπί “Αποσύνδεση” από το μενού πλοήγησης2. Αποσύνδεση από το παιχνίδι

3. Μετάβαση σε σελίδα σύνδεσης στην εφαρμογή

Εναλλακτικές:

1. Σφάλμα κατά την αποσύνδεση
2. Εμφάνιση μηνύματος “Κάτι πήγε στραβά”

Κεφάλαιο 6ο: Επίλογος και Συμπεράσματα

Σε αυτή την πτυχιακή εργασία παρουσιάζεται η δημιουργία ενός διαδικτυακού παιχνιδιού με θέμα την ελληνική μυθολογία. Γίνεται προφανές πως η δημιουργία ενός διαδικτυακού παιχνιδιού και στην προκειμένη περίπτωση ενός διαδικτυακού παιχνιδιού περιηγητή ιστού είναι μια αρκετά περίπλοκη διαδικασία. Ξεκινώντας από το αντικείμενο με το οποίο θέλει κανείς να ασχοληθεί στην εφαρμογή που θα δημιουργήσει, μέχρι και το σημαντικότερο όλων που είναι η επιλογή τεχνολογιών, απαιτείται η εκτενής μελέτη και έρευνα. Σήμερα υπάρχει πληθώρα τεχνολογιών, πρόσθετων αλλά και frameworks που μπορεί να χρησιμοποιήσει ο προγραμματιστής ανάλογα με το τι τον βοηθά καλύτερα να υλοποιήσει τους στόχους του.

Στην παρούσα πτυχιακή εργασία, επιλέχθηκαν η Angular και η Firebase για την ανάπτυξη του παιχνιδιού. Οι δύο αυτές τεχνολογίες επιλέχθηκαν εξαιτίας της πολύ καλής τους συνεργασίας, η οποία διευκολύνει τον προγραμματιστή στην ανάπτυξη μιας εφαρμογής. Από τη μία πλευρά, η Angular είναι από τις πιο δημοφιλείς τεχνολογίες front end, και από την άλλη, η Firebase αποτελεί ένα εξαιρετικά επεκτάσιμο σύστημα νέφους. Κοινός τόπος των δύο αυτών τεχνολογιών, και σημαντικότερο κριτήριο επιλογής και συνδυασμού τους, είναι η χρήση της ίδιας γλώσσας προγραμματισμού.

Επικεντρώνοντας στο παιχνίδι αυτό καθαυτό, η πτυχιακή αυτή εργασία πραγματεύεται ένα διαδικτυακό παιχνίδι περιηγητή ιστού, σε μορφή επιτραπέζιου παιχνιδιού, με θέμα την ελληνική μυθολογία. Σκοπός του παιχνιδιού είναι η παροχή αλλά και διεύρυνση γνώσεων που αφορούν στον τομέα της ελληνικής μυθολογίας, μέσα από το πρίσμα της διασκέδασης.

Μία εφαρμογή όμως, όσο καλά δομημένη και ανεπτυγμένη και να είναι, πάντα θα χρήζει βελτίωσης. Με την πάροδο του χρόνου δε, κάθε προγραμματιστής αναζητά αλλαγές που θα κάνει στον κώδικά του προκειμένου να προσελκύσει περισσότερο κοινό αλλά και να ικανοποιήσει περισσότερο το ήδη υπάρχον. Στον παρόν παιχνίδι, πολλές είναι οι βελτιώσεις και οι αλλαγές που θα μπορούσαν να γίνουν σε βάθος χρόνου. Το “Greek - Mythos” λοιπόν θα μπορούσε σίγουρα να εμπλουτιστεί με περισσότερες ερωτήσεις. Αυτό θα μπορούσε να γίνει και με τη δημιουργία ενοτήτων που θα αφορούν διάφορα θέματα της ελληνικής μυθολογίας ώστε ο παίκτης να μπορεί να επιλέξει με ποιο θέμα θα ήθελε να ασχοληθεί, αλλά και με τη δυνατότητα υποβολής ερωτήσεων από τους ίδιους τους χρήστες της εφαρμογής. Παράλληλα, θα ήταν θεμιτή η ανάπτυξη και η παροχή στους χρήστες ενός συστήματος καταγραφής των παιχνιδιών και των κινήσεων των παικτών, έτσι ώστε ο κάθε παίκτης να διατηρεί κάποιο αρχείο των παιχνιδιών του. Άλλη μια αλλαγή που θα μπορούσε να υπάρξει είναι η δημιουργία διαφόρων τύπων ερωτήσεων εκτός των τεσσάρων επιλογών απάντησης που υπάρχει τώρα. Θα μπορούσε να υπάρχει αντιστοίχιση, αναγνώριση ήχων, ακόμη και σχεδίαση ή συμπλήρωση κενών σε προτάσεις.

Αναφορικά με το περιβάλλον της εφαρμογής, η Αρχική σελίδα, θα μπορούσε να εμπλουτιστεί με διάφορα βίντεο παρουσίασης του παιχνιδιού αλλά και βίντεο περαιτέρω εκμάθησης. Η δημιουργία προφίλ θα μπορούσε να έχει ένα φίλτρο ονομάτων λογαριασμών, έτσι ώστε να μην επιτρέπονται οποιουδήποτε είδους βωμολοχίες. Θα μπορούσε να υπάρχει τέλος, ένα online chat, όπου θα μπορούσαν να μιλήσουν παίκτες από όλο τον κόσμο αλλά και φίλοι μεταξύ τους.

Ο κατάλογος των προτάσεων για αλλαγή και βελτίωση ίσως είναι ατελείωτος καθώς οι ανάγκες και οι απαιτήσεις των χρηστών συνεχώς αλλάζουν. Επιπρόσθετα, η μαγεία του προγραμματισμού είναι στη δημιουργία. Όσο μεγαλύτερη είναι η μελέτη και η εμπειρία, τόσο μεγαλύτερη και η εξέλιξη της

Κεφάλαιο 6

εφαρμογής. Ωστόσο, το παιχνίδι “Greek - Mythos” είναι εδώ για να προσφέρει μια αξέχαστη εμπειρία εκμάθησης της ελληνικής μυθολογίας και μύησης σε αυτήν, με διασκεδαστικό τρόπο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Prensky M. (2007), Μάθηση βασισμένη στο Ψηφιακό Παιχνίδι, (επιμέλεια: Μ. Μεϊμάρης, Μετάφραση: Κ. Παπασταύρου, Ν. Παπασταύρου)
- [2] Wikipedia (2021). Βιντεοπαιχνίδι. Ανακτήθηκε από <https://el.wikipedia.org/wiki/Βιντεοπαιχνίδι>
- [3] Αναγνώστου Κ. 2009 Βιντεοπαιχνίδια - Βιομηχανία και ανάπτυξη. Αθήνα: Κλειδάριθμος
- [4] Museum of play. Video Game History.
Ανακτήθηκε από <https://www.museumofplay.org/about/icheg/video-game-history/timeline>
- [5] The Video Game Explosion: A History from PONG to Playstation and Beyond
- [6] The Ultimate History of Video Games: From Pong to Pokemon and Beyond ...
- [7] Wikipedia (2021). Minecraft. Ανακτήθηκε από <https://el.wikipedia.org/wiki/Minecraft>
- [8] International Student – What is Video Game Development.
Ανακτήθηκε από: <https://www.internationalstudent.com/study-video-game-development/what-is-video-game-development/>
- [9] Game Design Foudnations – Roger E. Pedersen. Ανακτήθηκε από:
https://books.google.gr/books?id=DvYmFr_uaWwC&printsec=frontcover&dq=inauthor:%22Roger+E.+Pedersen%22&hl=el&sa=X&redir_esc=y#v=onepage&q&f=false
- [10] Wikipedia (2021). Loading Screen.
Ανακτήθηκε από: https://en.wikipedia.org/wiki/Loading_screen
- [11] Fullerton , T. , Swain C. , Hoffman S. S. (2008) Game Design Workshop A Playcentric Approach To Creating Innovative Games. USA: Elsevier Inc.
- [12] Shell, J. (2008) The Art of Game Design. A Book of Lenses. Carnegie Mellon University. USA: Elsevier Inc.
- [13] Carlton Books (2006) The Ultimate Encyclopedia of Fantasy.
- [14] What is a strategy game?
Ανακτήθηκε από: <https://www.kathleenmercury.com/what-is-a-strategy-game.html>
- [15] Mayer Brian & Harris Christopher. (2010) Libraries Got Game: Aligned Learning Through Modern Board Games. The American Library Association, USA
- [16] Ghosoon Nafeth Qteefan (2012). Thesis: The Effectiveness of Using Educational Computer Games on Developing Palestinian Fifth Graders' Achievement in English Language in Gaza Governorate. The Islamic University of Gaza - Deanery of Graduate Studies - Faculty of Education - Curriculum and English Teaching Methods Department
- [17] M. Mcphee. (2015, May) Deep Blue IBM's supercomputer defeats chess champion Garry Kasparov in 1997.

- [18] Dinis Felix, “Artificial Intelligence Techniques in Games with Incomplete Information: Opponent Modelling in Texas Hold’em”.
- [19] Wikipedia (2021). Τάβλι. Ανακτήθηκε από: <https://el.wikipedia.org/wiki/Τάβλι>
- [20] Dinesh, C. Verma [2004]. Legitimate Applications of Peer-to-Peer Networks. ISBN 0-471-46369-8
- [21] Roger Delano Paul McFarlane “Network Software Architectures for Real-Time Massively-Multiplayer Online Games”
- [22] Wikipedia (2021). Ελληνική μυθολογία.
Ανακτήθηκε από: https://el.wikipedia.org/wiki/Ελληνική_μυθολογία
- [23] Games with a Greek Mythology theme.
Ανακτήθηκε από: <https://boardgamegeek.com/geeklist/51956/games-greek-mythology-theme/page/1>
- [24] Μαρκατσέλας, Μ., & Ξαρχάκος, Κ. (2012). Μαρκατσέλας Μανώλης, Ξαρχάκος Κωνσταντίνος Ι., Μαθαίνετε εύκολα Joomla! 2.5, Αθήνα: Εκδόσεις Ξαρχάκος.
- [25] Wikipedia (2021). HTML. Ανακτήθηκε από: <https://el.wikipedia.org/wiki/HTML>
- [26] HTML5 Introduction. (2016, 2). Ανακτήθηκε από:
http://www.w3schools.com/html/html5_intro.asp
- [27] Wikipedia (2021). CSS. Ανακτήθηκε από: <https://en.wikipedia.org/wiki/CSS>
- [28] Wikipedia (2021). JavaScript. Ανακτήθηκε από: <https://el.wikipedia.org/wiki/JavaScript>
- [29] Pull Requests σε διάφορες γλώσσες προγραμματισμού στο GitHub. Ανακτήθηκε από:
https://madnight.github.io/github/#/pull_requests/2020/4
- [30] Best Frontend Frameworks of 2021 for Web Development.
Ανακτήθηκε από: <https://www.simform.com/best-frontend-frameworks/>
- [31] Wikipedia (2021). React (JavaScript library).
Ανακτήθηκε από: [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))
- [32] Wikipedia (2021). TypeScript. Ανακτήθηκε από: <https://en.wikipedia.org/wiki/TypeScript>
- [33] Wikipedia (2021). Vue. Ανακτήθηκε από: <https://en.wikipedia.org/wiki/Vue.js>
- [34] TechTerms. Backend. Ανακτήθηκε από: <https://techterms.com/definition/backend>
- [35] Wikipedia (2021). PHP. Ανακτήθηκε από: <https://el.wikipedia.org/wiki/PHP>
- [36] Wakefly. What is asp.net and why should I use it?
Ανακτήθηκε από: <https://www.wakefly.com/blog/what-is-asp-net-and-why-should-i-use-it/>
- [37] SCAND. Why Use Java for Back-end Development?
Ανακτήθηκε από: <https://scand.com/company/blog/why-use-java-for-back-end-development/>
- [38] Wikipedia (2021). Node.js Ανακτήθηκε από: <https://en.wikipedia.org/wiki/Node.js>
- [39] Wikipedia (2021). Adobe Flash. Ανακτήθηκε από: https://el.wikipedia.org/wiki/Adobe_Flash

[40] Wikipedia (2021). Java applet Ανακτήθηκε από: https://en.wikipedia.org/wiki/Java_applet

[41] Wikipedia (2021). Unity (game engine).

Ανακτήθηκε από: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))

[42] Wikipedia (2021). Firebase. Ανακτήθηκε από: <https://en.wikipedia.org/wiki/Firebase>