



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κατασκευή διαδικτυακού ιστότοπου παιχνιδιών

Λάζαρος Μπογιατζάρας

Εισηγητής: Δρ. Κωνσταντίνος Κουκουλέτσος, Καθηγητής

**ΑΘΗΝΑ
ΝΟΕΜΒΡΗΣ 2018**

Κατασκευή Διαδικτυακού Ιστότοπου Παιχνιδιών

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κατασκευή διαδικτυακού ιστότοπου παιχνιδιών

**Λάζαρος Μπογιατζάρας
Α.Μ. 43644**

Εισηγητής:

Δρ. Κωνσταντίνος Κουκουλέτσος, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Μπογιατζάρας Λάζαρος, του Παναγιώτη, με αριθμό μητρώου 43644 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της κατασκευής ιστοσελίδων. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου Κωνσταντίνος Κουκουλέτσος, τον οποίο θα ήθελα να ευχαριστήσω.

Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου και ιδιαίτερα τη μητέρα μου που με στήριξε ψυχολογικά κατά τη διάρκεια των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει ως θέμα τη σχεδίαση και υλοποίηση ενός διαδικτυακού ιστότοπου ο οποίος περιέχει βιντεοπαιχνίδια. Αρχικά γίνεται μια ιστορική αναδρομή πάνω στην έννοια του βιντεοπαιχνιδιού και ειδικότερα των δύο που υλοποιήθηκαν στην παρούσα πτυχιακή, το φιδάκι (snake) και το arkanoid. Στη συνέχεια θα ακολουθήσει μια επεξήγηση της κάθε τεχνολογίας που χρησιμοποιήθηκε (HTML, CSS, JavaScript, PHP, MySQL), καθώς και των προγραμμάτων που βοήθησαν στην υλοποίηση της εφαρμογής. Η πτυχιακή εργασία ολοκληρώνεται με την αναφορά των συμπερασμάτων και με προτάσεις για περαιτέρω εμβάθυνση και μελλοντική βελτίωση της εφαρμογής. Με λίγα λόγια, η κύρια λειτουργία της εφαρμογής είναι η εξής: ο χρήστης θα μπορεί να εισέρχεται στον ιστότοπο με χρήση ενός username και password, θα επιλέγει ποιο παιχνίδι θέλει να παίξει και αφού χάσει ή κερδίσει οι πόντοι του θα αποθηκεύονται στη βάση δεδομένων για προβολή και σύγκριση με άλλους χρήστες.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Κατασκευή Ιστοσελίδων Διαδικτύου

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: διαδίκτυο, βιντεοπαιχνίδια, ιστότοπος, ψυχαγωγία, JavaScript

ABSTRACT

This thesis deals with the design and implementation of a web site that presents video games to users to choose and play online.

The user may enter the site using a username and password and choose which game he wants to play. When the game is over the user's score will be stored in a database for viewing and comparing with the scores of other users.

The thesis starts with a historical review of the development of the video games and in particular it focuses on the two that for practical reasons were implemented: Snake and Arkanoid. Following that an explanation of each technology employed (HTML, CSS, JavaScript, PHP, MySQL) is presented. Afterwards, the code developed in order to implement the application is explained and analyzed.

Finally, the thesis ends with the conclusions of the developed application and gives suggestions for future improvements to a professional level.

SCIENTIFIC AREA: Website Developing

KEY WORDS: web, videogames, website, entertainment, JavaScript

ΠΕΡΙΕΧΟΜΕΝΑ

| | |
|--|----|
| 1. ΕΙΣΑΓΩΓΗ..... | 15 |
| 1.1 Αντικείμενο της πτυχιακής εργασίας | 15 |
| 1.2 Δομή | 15 |
| 2. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΝΝΟΙΑ ΤΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΑΙΧΝΙΔΙΩΝ | 17 |
| 2.1 Τι είναι βιντεοπαιχνίδι | 17 |
| 2.2 Ιστορική αναδρομή | 17 |
| 2.2.1 Τα πρώτα χρόνια | 17 |
| 2.2.2 Μπαίνοντας στον κόσμο των Arcade | 19 |
| 2.2.3 Το μεγάλο «μπαμ» | 20 |
| 2.2.4 Πτώση και η στροφή στους υπολογιστές | 20 |
| 2.2.5 Οι κονσόλες ξανά στο προσκήνιο | 21 |
| 2.2.6 Βιντεοπαιχνίδια στις αρχές του 21 ^{ου} αιώνα..... | 23 |
| 2.2.7 Τα βιντεοπαιχνίδια του σήμερα | 24 |
| 2.2.8 Τι μας επιφυλάσσει το μέλλον..... | 25 |
| 3. ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΗΣ..... | 27 |
| 3.1 HTML..... | 27 |
| 3.1.1 Εισαγωγή στα βασικά στοιχεία..... | 27 |
| 3.1.2 Χρήση της HTML στην εφαρμογή | 28 |
| 3.2 CSS | 28 |
| 3.2.1 Εισαγωγή στα βασικά στοιχεία..... | 28 |
| 3.2.2 Χρήση της CSS στην εφαρμογή..... | 29 |
| 3.3 JavaScript..... | 30 |
| 3.3.1 Εισαγωγή στα βασικά στοιχεία..... | 30 |
| 3.3.2 JavaScript και διακομιστής..... | 31 |
| 3.3.3 Χρήση της JavaScript στην εφαρμογή | 32 |
| 3.4 PHP | 33 |
| 3.4.1 Εισαγωγή στα βασικά στοιχεία..... | 33 |

| | | |
|-------|---|----|
| 3.4.2 | Χρήση της PHP στην εφαρμογή..... | 34 |
| 3.5 | MySQL..... | 35 |
| 3.5.1 | Εισαγωγή στα βασικά στοιχεία..... | 35 |
| 3.5.2 | Χρήση της MySQL στην εφαρμογή..... | 35 |
| 3.6 | Προγράμματα που χρησιμοποιήθηκαν..... | 36 |
| 4. | ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΣΦΑΛΕΙΑ..... | 37 |
| 4.1 | Σχεδιασμός βάσης δεδομένων..... | 37 |
| 4.2 | Ασφάλεια εφαρμογής..... | 38 |
| 5. | ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ..... | 41 |
| 5.1 | Λίγα λόγια για τα παιχνίδια της εφαρμογής..... | 41 |
| 5.2 | Η σελίδα σύνδεσης χρήστη (login.php)..... | 42 |
| 5.3 | Η κύρια σελίδα (landing_page.php)..... | 43 |
| 5.4 | Το παιχνίδι Snake (snake_game.php)..... | 44 |
| 5.5 | Η σελίδα πόντων του παιχνιδιού Snake (snake_leaderboard.php)..... | 45 |
| 5.6 | Το παιχνίδι Arkanoid (arkanoid_game.php)..... | 46 |
| 5.7 | Η σελίδα πόντων του παιχνιδιού Arkanoid (arkanoid_game.php)..... | 47 |
| 6. | ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΕΦΑΡΜΟΓΗΣ..... | 49 |
| 6.1 | config.php..... | 49 |
| 6.2 | login.php..... | 49 |
| 6.3 | logout.php..... | 52 |
| 6.4 | landing_page.php..... | 52 |
| 6.5 | snake_game.php..... | 55 |
| 6.6 | snake.js..... | 56 |
| 6.7 | snake_leaderboard.php..... | 61 |
| 6.8 | arkanoid_game.php..... | 63 |
| 6.9 | arkanoid.js..... | 64 |
| 6.10 | arkanoid_leaderboard.php..... | 73 |
| 6.11 | savescores.php..... | 75 |
| 7. | ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ..... | 77 |

Κατασκευή Διαδικτυακού Ιστότοπου Παιχνιδιών

| | | |
|-----|------------------------------|----|
| 7.1 | Συμπεράσματα | 77 |
| 7.2 | Μελλοντικές Επεκτάσεις | 77 |
| 8. | ΕΠΙΛΟΓΟΣ..... | 79 |
| | ΒΙΒΛΙΟΓΡΑΦΙΑ..... | 81 |

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

| | |
|--|----|
| Εικόνα 2-1: Άλαν Τούρινγκ, μαθηματικός και κρυπτογράφος (Άλαν Τούρινγκ, 2018)..... | 18 |
| Εικόνα 2-2: Ο PDP-01 και ο δημιουργός του Spacewars! Steve Russell (PDP-1, 2018)..... | 19 |
| Εικόνα 2-3: Διάφορα μηχανήματα Arcade (Wikipedia, 2011) | 20 |
| Εικόνα 2-4: Η κάρτα επιτάχυνσης γραφικών Voodoo3 - 2000AGP (Wikipedia, 2006)..... | 23 |
| Εικόνα 2-5: Το προϊόν Oculus Rift CV1 (Oculus Rift, 2017) | 25 |
| Εικόνα 3-1: Παράδειγμα ιστοσελίδας με χρήση CSS (Bootstrap Example, 2015) | 29 |
| Εικόνα 3-2: Αναπαράσταση δέντρου DOM (DOM Model, 2012) | 31 |
| Εικόνα 3-3: Σχεδιάγραμμα μιας δυναμικής σελίδας με χρήση PHP (Scheme Dynamic Page, 2012)..... | 34 |
| Εικόνα 4-1: Script δημιουργίας πίνακα users | 37 |
| Εικόνα 4-2: Script δημιουργίας πίνακα scores..... | 38 |
| Εικόνα 5-1: Η σελίδα σύνδεσης (login page) | 42 |
| Εικόνα 5-2: Η κύρια σελίδα της εφαρμογής | 43 |
| Εικόνα 5-3: Η σελίδα του παιχνιδιού Snake | 44 |
| Εικόνα 5-4: Η σελίδα με τα σκορ του παιχνιδιού Snake | 45 |
| Εικόνα 5-5: Η σελίδα του παιχνιδιού Arkanoid | 46 |
| Εικόνα 5-6: Η σελίδα με τα σκορ του παιχνιδιού Arkanoid | 47 |

ΠΙΝΑΚΑΣ ΑΚΡΩΝΥΜΙΩΝ

| | |
|--------|---|
| CRT | Cathode Ray Tube |
| RAM | Random Access Memory |
| VGA | Video Graphics Adapter |
| NES | Nintendo Entertainment System |
| RTS | Real Time Strategy (games) |
| FPS | First Person Shooter |
| PS2 | PlayStation 2 |
| MMORPG | Massively Multiplayer Online Role Playing Games |
| PSP | PlayStation Portable |
| PS3 | PlayStation 3 |
| HD | High Definition |
| VR | Virtual Reality |
| AI | Artificial Intelligence |
| PC | Personal Computer |
| HTML | Hyper-Text Markup Language |
| CSS | Cascading Style Sheet |
| DOM | Document Object Model |
| W3C | World Wide Web Consortium |
| JS | JavaScript |
| ECMA | European Computer Manufacturers Association |
| API | Application Programming Interface |
| AJAX | Asynchronous JavaScript And XML |
| HTTP | Hyper-Text Transfer Protocol |
| PHP | Hypertext Preprocessor |
| CMS | Content Management System |
| RDBMS | Relational Data Base Management System |
| SQL | Structured Query Language |
| IDE | Integrated Development Environment |
| CE | Community Edition |

Κατασκευή Διαδικτυακού Ιστότοπου Παιχνιδιών

| | |
|-----|----------------------|
| PK | Primary Key |
| FK | Foreign Key |
| XSS | Cross-Site Scripting |

1. ΕΙΣΑΓΩΓΗ

Η παρούσα πτυχιακή εργασία με τίτλο «Κατασκευή Διαδικτυακού Ιστότοπου Παιχνιδιών» εκπονήθηκε κατά τη διάρκεια του ακαδημαϊκού έτους 2017-2018 στα πλαίσια της υποχρεωτικής πτυχιακής εργασίας στον κύκλο μαθημάτων του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής. Τα βιντεοπαιχνίδια (video games) έχουν αρχίσει να γίνονται το πιο διαδεδομένο μέσο ψυχαγωγίας τη σήμερον ημέρα, σε μικρές (κυρίως) αλλά και μεγάλες ηλικίες, ιδίως με την αυξημένη προσβασιμότητα στους ηλεκτρονικούς υπολογιστές. Έτσι, γεννήθηκε και το κίνητρο συγγραφής της παρούσας πτυχιακής. Σκοπός της εργασίας είναι να εισάγει τον αναγνώστη στις κύριες τεχνολογίες που χρησιμοποιούνται για ανάπτυξη βιντεοπαιχνιδιών με βάση έναν φυλλομετρητή (browser-based video games) καθώς και των κύριων μεθόδων κατά την κατασκευή τέτοιων παιχνιδιών.

1.1 Αντικείμενο της πτυχιακής εργασίας

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι να παρουσιάσει στον αναγνώστη τα βιντεοπαιχνίδια, και τις τεχνολογίες στις οποίες βασίζεται η ανάπτυξή τους. Στόχος της πτυχιακής αυτής είναι ο αναγνώστης να έχει αποκτήσει μια βαθύτερη κατανόηση της προέλευσης των βιντεοπαιχνιδιών και των τεχνολογιών που χρησιμοποιούνται για να παραχθούν.

1.2 Δομή

Η δομή της πτυχιακής εργασίας είναι η εξής:

- Στο 1^ο κεφάλαιο παρουσιάζεται η εισαγωγή της πτυχιακής εργασίας.
- Στο 2^ο κεφάλαιο γίνεται ανάλυση του όρου βιντεοπαιχνίδι και παρουσίαση της ιστορίας του.
- Στο 3^ο κεφάλαιο παρουσιάζονται όλες οι τεχνολογίες και προγράμματα που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής.

Κατασκευή Διαδικτυακού Ιστότοπου Παιχνιδιών

- Στο 4^ο κεφάλαιο γίνεται ανάλυση της δομής της βάσης δεδομένων, καθώς και αναφορά στην ασφάλεια της εφαρμογής.
- Στο 5^ο κεφάλαιο παρουσιάζεται σελίδα – σελίδα η λειτουργία της εφαρμογής.
- Στο 6^ο κεφάλαιο δίνεται ο επίλογος, τα συμπεράσματα και οι μελλοντικές προσθήκες και αναβαθμίσεις που μπορούν να γίνουν.

2. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΝΝΟΙΑ ΤΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΑΙΧΝΙΔΙΩΝ

Στο κεφάλαιο αυτό θα γίνει μια γενική παρουσίαση των βιντεοπαιχνιδιών, καθώς και μια ιστορική αναδρομή.

2.1 Τι είναι βιντεοπαιχνίδι

Όταν λέμε βιντεοπαιχνίδι εννοούμε οποιοδήποτε παιχνίδι που μπορεί να πραγματοποιηθεί με τη χρήση κάποιου ηλεκτρονικού μέσου, όπως ηλεκτρονικό υπολογιστή, κονσόλα βιντεοπαιχνιδιών (ή παιχνιδομηχανή), κινητό τηλέφωνο κ.α. Όλα τα βιντεοπαιχνίδια αποτελούνται από δύο βασικά στοιχεία: μία ή παραπάνω συσκευές εισόδου (πληκτρολόγιο, ποντίκι, τηλεχειριστήριο, οθόνη αφής, κ.α.) και μία ή παραπάνω συσκευές εξόδου (οθόνη υπολογιστή, ηχεία, κ.α.). (Βιντεοπαιχνίδι, 2017)

2.2 Ιστορική αναδρομή

2.2.1 Τα πρώτα χρόνια

Τα πρώτα μικρά βήματα για αυτό που αργότερα θα ονομαστεί βιντεοπαιχνίδι έγιναν από έναν Άγγλο μαθηματικό, καθηγητή της λογικής και κρυπτογράφο, Άλαν Τούρινγκ. (Άλαν Τούρινγκ, 2018) Μαζί με τον καθηγητή David Champenowne έφτιαξαν ένα πρόγραμμα με όνομα Turochamp, το οποίο δεν ήταν άλλο από ένα παιχνίδι σκακιού σχεδιασμένο για επίδειξη της τεχνητής νοημοσύνης στην οποία πραγματοποιούσε έρευνες ο Τούρινγκ. Το παιχνίδι αυτό όμως δεν υλοποιήθηκε ποτέ σε υπολογιστή. Το πρώτο υλοποιημένο τέτοιο παιχνίδι ήταν η ευρέως γνωστή σε όλους μας τρίλιζα (tic-tac-toe) που κατασκευάστηκε αργότερα το ίδιο έτος από τον Josef Kates και τον Rogers Majestic. (History of video games, 2018)



Εικόνα 2-1: Άλαν Τούρινγκ, μαθηματικός και κρυπτογράφος (Άλαν Τούρινγκ, 2018)

Αργότερα φτιάχτηκε και το πρώτο ηλεκτρονικό παιχνίδι που χρησιμοποιούσε την πρώτη μορφή οθόνης της εποχής, τον καθωδικό σωλήνα (cathode ray tube - CRT). Αυτό ήταν η γνωστή ντάμα (checkers) και ήταν δημιουργία του Christopher Strachey, υλοποιημένο για τον πρώτο υπολογιστή διαθέσιμο για τον καταναλωτή Ferranti Mark 1. (History of video games, 2018)

Με το πέρασμα των χρόνων οι υπολογιστές εξελίσσονταν ραγδαία. Όταν υλοποιήθηκε ο πρώτος υπολογιστής που εφαρμόζε την ιδέα της μνήμης τυχαίας προσπέλασης (random access memory - RAM), TX-0, ο δρόμος για την εξέλιξη των βιντεοπαιχνιδιών άνοιξε ακόμα περισσότερο. Ο υπολογιστής αυτός δωρίστηκε στο πανεπιστήμιο MIT το 1958 και μπορούσαν πλέον οι σπουδαστές να πειραματιστούν ελεύθερα. Μετά από 3 χρόνια (1961) το MIT έλαβε άλλη μία δωρεά, την αναβαθμισμένη έκδοση του TX-0, τον PDP-01. Πάνω σε αυτόν μία ομάδα σπουδαστών σχεδίασε το πρώτο και με τη μεγαλύτερη επιρροή στην κοινότητα βιντεοπαιχνίδι, Spacewars!. (History of video games, 2018)



Εικόνα 2-2: Ο PDP-01 και ο δημιουργός του Spacewars! Steve Russell (PDP-1, 2018)

2.2.2 Μπαίνοντας στον κόσμο των Arcade

Το 1972 ιδρύθηκε η εταιρεία Atari, η οποία στη συνέχεια έφτιαξε το πρώτο βιντεοπαιχνίδι που χρησιμοποιούσε μπάλα και ρακέτα, το λεγόμενο pong. Αφού εισήχθη στη βιομηχανία της διασκέδασης με μηχανήματα που δούλευαν με κέρματα (arcade amusements) αμέσως ξέσπασε μια τρέλα για τέτοιου είδους παιχνίδια με πολλές εταιρείες να βγάζουν διαφορετικές εκδοχές για τα επόμενα δύο χρόνια. (History of video games, 2018)

Γρήγορα αυτή η τρέλα καταλάγιασε και τα έσοδα των εταιρειών άρχισαν να μειώνονται. Έτσι οι εταιρείες οδηγήθηκαν στην ανάπτυξη νέων ειδών παιχνιδιών, όπως βιντεοπαιχνίδια αγώνων (racing games) και βολών (shooter games), για να ανακάμψουν. Σημαντικό προς αναφορά είναι ότι το 1976, η εταιρεία Atari δημιούργησε το βιντεοπαιχνίδι Breakout, που αποτελεί πρόγονο του ενός από τα δύο παιχνίδια που περιλαμβάνονται στην παρούσα πτυχιακή εργασία, του Arkanoid. Το 1975 κατασκευάστηκαν και οι πρώτες οικιακές κονσόλες. (History of video games, 2018)



Εικόνα 2-3: Διάφορα μηχανήματα Arcade (Wikipedia, 2011)

2.2.3 Το μεγάλο «μπαμ»

Το έτος 1978 είναι έτος ορόσημο για την ιστορία των βιντεοπαιχνιδιών. Ο Ιάπωνας δημιουργός βιντεοπαιχνιδιών Tomohiro Nishikado, βασιζόμενος στο παιχνίδι Breakout δημιούργησε το εικονικό παιχνίδι-ορόσημο Space Invaders. Ήταν το πρώτο βιντεοπαιχνίδι που ενσωμάτωσε τις τότε πρωτοποριακές έννοιες των ζών και το σώσιμο των πόντων του παίκτη (high score) στο μηχανήμα. Αυτό το παιχνίδι και το παιχνίδι Asteroids της Atari έκαναν πάνω από 70000 πωλήσεις, καθιερώνοντας έτσι τα arcade βιντεοπαιχνίδια ως τον κύριο πόλο έλξης στα κέντρα arcade, αφήνοντας σε δεύτερη μοίρα παιχνίδια όπως το φλιπεράκι (pinball). Η τάση αυτή μεταφέρθηκε και στην Αμερική, με κύριους πρωταγωνιστές βιντεοπαιχνίδια όπως τα πασίγνωστα Pac-man, Donkey Kong και Galaga. Τα έσοδα εκείνη την περίοδο είδαν μνημειώδη αύξηση: από 308 εκατομμύρια δολάρια το 1978, σε 968 εκατομμύρια το 1979, σε 2.8 δισεκατομμύρια το 1980 και τέλος σε 4.9 δισεκατομμύρια το 1981. (History of video games, 2018)

2.2.4 Πτώση και η στροφή στους υπολογιστές

Η μεγάλη δημοτικότητα του βιντεοπαιχνιδιού Space Invaders οδήγησε σε εμφάνιση πολλών νέων εταιρειών με δικές τους κονσόλες και παιχνίδια, τα οποία όμως υστερούσαν σε ποιότητα και ουσία. Έτσι φτάσαμε το 1983 στη λεγόμενη συντριβή των βιντεοπαιχνιδιών (1983 video game crash) με πολλές χρηματικές απώλειες για τις εταιρείες. (Chikhani, 2015)

Την σκυτάλη τότε πήραν οι υπολογιστές σπιτιού, όπως οι Commodore Vic-20, Commodore 64 και Apple II. Αυτά τα μηχανήματα διέθεταν ισχυρότερους επεξεργαστές και ήταν σχετικά πιο προσβάσιμα στους καταναλωτές με αξία 300 δολαρίων (περίπου 860 σημερινά δολάρια). Η γλώσσα προγραμματισμού BASIC που χρησιμοποιούταν στους υπολογιστές αυτούς βοήθησε επίσης τον κόσμο να φτιάξει και να μοιραστεί τον δικό του κώδικα παιχνιδιών. (Chikhani, 2015)

Το 1985 έγινε η αρχή για την αρχιτεκτονική των 16 bit υπολογιστών με τα μηχανήματα Atari ST και Commodore Amiga. Η εταιρεία IBM επίσης παρουσίασε τον δικό της υπολογιστή, IBM PC compatible, πιο φθηνό και ισχυρό από τους ανταγωνιστές. Ενσωμάτωσε επίσης και το πρωτόκολλο του ελεγκτή οθόνης βίντεο (VGA – Video Graphics Adapter) που έδωσε τη δυνατότητα στον υπολογιστή να απεικονίσει γραφικά σε 256 χρώματα. Η εισαγωγή των αφοσιωμένων καρτών ήχου (sound cards) από την εταιρεία Creative Labs το 1989 βελτίωσε την ποιότητα του ήχου αλλά δυστυχώς οι κάρτες αυτές είχαν μεγάλο κόστος και δεν γνώρισαν μεγάλη απήχηση μέχρι τη δεκαετία του 90. (History of video games, 2018)

2.2.5 Οι κονσόλες ξανά στο προσκήνιο

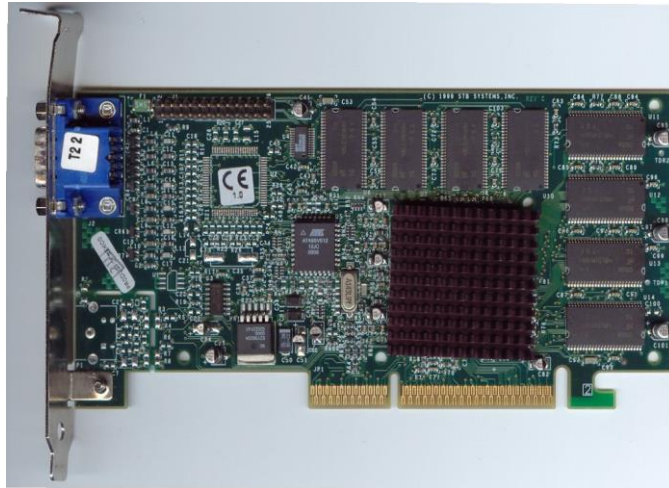
Την ίδια στιγμή, η εταιρεία Nintendo έβγαλε στην αγορά της Αμερικής την κονσόλα NES (Nintendo Entertainment System) και αναζωπύρωσε την αγορά βιντεοπαιχνιδιών. Ήταν η πρώτη κονσόλα που έφερε στο προσκήνιο την έννοια του τηλεχειριστηρίου όπως περίπου είναι τώρα, με ένα σταυρό κατεύθυνσης (D-pad) και τουλάχιστον 2 ή περισσότερα κουμπιά ενεργειών. Πάνω σε αυτήν έκαναν την αρχή τους πολλές μεγάλες σειρές βιντεοπαιχνιδιών, όπως το Legend of Zelda και το Final Fantasy. Επίσης, η σειρά Metal Gear έκανε την εμφάνισή της στον υπολογιστή MSX2 της Microsoft. (History of video games, 2018)

Το 1988 εμφανίστηκαν τα πρώτα CD-Rom drives και τα πρώτα τριών διαστάσεων γραφικά που χρησιμοποιούσαν πολύγωνα (3D graphics with polygons). Το έτος 1991 είδε την εισαγωγή δύο ιστορικών χαρακτήρων βιντεοπαιχνιδιών, του Sonic the Hedgehog της εταιρείας Sega και του Mario της εταιρείας Nintendo στα ομώνυμά τους βιντεοπαιχνίδια. Παράλληλα, εισήχθη στην αγορά και η πρώτη φορητή κονσόλα από τη Nintendo, το Gameboy, (έτος 1989)

με εικονικούς τίτλους όπως τα Pokemon Red και Blue. (History of video games, 2018)

Η δεκαετία του 90 είδε την εισαγωγή 32 και 64 bit κονσόλων, όπως το Sony PlayStation (1994) και το Nintendo 64 (1996) και το πέρασμα σε γραφικά με χαρτογράφηση (texture mapping). Την περίοδο αυτή εμφανίστηκαν και άλλα είδη βιντεοπαιχνιδιών, όπως βιντεοπαιχνίδια στρατηγικής πραγματικού χρόνου (real time strategy games - RTS) κυρίως στους υπολογιστές (όπως οι σειρές Age of Empires, Warcraft και Starcraft) και το είδος επιβίωσης φόβου (survival horror), με κύριους αντιπρόσωπους τις σειρές Silent Hill και Resident Evil. Ήταν επίσης η περίοδος που αναπτύχθηκε και η πρώτη μορφή κάρτας γραφικών, η κάρτα επιτάχυνσης 3D γραφικών (3D accelerator card) ονομαζόμενη Voodoo. Το είδος βολών πρώτου προσώπου (first person shooter - FPS) εκμεταλλεύτηκε καλύτερα από οποιονδήποτε άλλο την καινούρια αυτή αρχιτεκτονική, που αποδέσμευσε τον επεξεργαστή από ολόκληρη την επεξεργασία γραφικών. Παράλληλα, η ανάπτυξη κάποιων επεκτάσεων στους φυλλομετρητές διαδικτύου (web browser plugins), όπως Java και Adobe Flash, επέτρεψαν την εμφάνιση των πρώτων απλών βιντεοπαιχνιδιών βασισμένα στους φυλλομετρητές (browser-based games). (History of video games, 2018)

Παροδικά τα βιντεοπαιχνίδια βασίζονταν όλο και περισσότερο στα 3D γραφικά, με τα πρώτα βιντεοπαιχνίδια (όπως οι σειρές Crash Bandicoot και Spyro the Dragon για το PlayStation) να φτιάχνονται εξ' ολοκλήρου σε 3D περιβάλλοντα. Αυτή την περίοδο έκανε και την πρώτη εμφάνισή του το ένα από τα δύο παιχνίδια που πραγματεύεται η πτυχιακή αυτή, το φιδάκι, με την εταιρεία Nokia να το φέρνει στη σειρά κινητών τηλεφώνων που λάνσαρε τότε ως ένα μικρό παιχνίδι για να περνάει ο κόσμος το χρόνο του. (History of video games, 2018)



Εικόνα 2-4: Η κάρτα επιτάχυνσης γραφικών Voodoo3 - 2000AGP (Wikipedia, 2006)

2.2.6 Βιντεοπαιχνίδια στις αρχές του 21^{ου} αιώνα

Το έτος 2000 είδε την εταιρεία Sony να παρουσιάζει το διάδοχο του PlayStation, PlayStation 2 (PS2), ίσως την πιο πετυχημένη κονσόλα στην ιστορία της βιομηχανίας των παιχνιδιών. Με αυτή, η Sony κατέλαβε την πρωτιά στην αγορά, αφήνοντας πίσω τη Nintendo και τη νεοεισαχθείσα Microsoft (που λάνσαρε τη δικιά της πρώτη κονσόλα στα τέλη του 2001, Xbox). (History of video games, 2018)

Στο μέτωπο των υπολογιστών, το 2001 διατέθηκε στην αγορά το βιντεοπαιχνίδι Runescape, το πρώτο του είδους μαζικών διαδικτυακών παιχνιδιών ρόλων πολλαπλών παικτών (MMORPG – massively multiplayer online role-playing games), που επέτρεπε σε παίκτες ανά τον κόσμο να παίζουν και να επικοινωνούν μεταξύ τους με τη χρήση του διαδικτύου μέσω συζητήσεων (chat function). (Chikhani, 2015)

Το βιντεοπαιχνίδι Grand Theft Auto III που ανακοινώθηκε τότε έκανε διάσημο το είδος των παιχνιδιών ανοικτού κόσμου (open world games), παρά την κριτική που γνώρισε λόγω περιεχομένου (βία, σεξουαλικό περιεχόμενο αλλά και χλευασμό της αστυνομίας). Η Nintendo εξακολουθούσε να κατέχει την πρωτιά στο χώρο των φορητών κονσόλων με την κυκλοφορία του Game Boy Advance το 2001. Τα πράγματα κύλησαν στα ίδια νερά τα επόμενα χρόνια με τη Sony να εισέρχεται στον κόσμο των φορητών κονσόλων με το PlayStation Portable (PSP) και τη Nintendo να παρουσιάζει το διάδοχο του Game Boy Advance, το Nintendo DS, (έτος 2004) παραμένοντας η κυριότερη προτίμηση του κοινού στον τομέα αυτό.

Με το διαδίκτυο να έχει εδραιωθεί πλέον στην αγορά, οι 2 μεγαλύτερες εταιρείες στη βιομηχανία των κονσόλων Sony και Microsoft, κυκλοφόρησαν τις ανανεωμένες εκδόσεις των προϊόντων τους, PlayStation 3 (PS3) και Xbox 360 αντίστοιχα, με δυνατότητες απεικόνισης γραφικών υψηλής ευκρίνειας (HD) (high definition graphics) μέσω καλωδίου HDMI, ενσωματωμένη διαδίκτυωση (και Wi-Fi) και ενσωματωμένο αποθηκευτικό χώρο για τα βιντεοπαιχνίδια και τα σωσμένα αρχεία τους. Ήταν οι πρώτες κονσόλες που μπορούσαν να ανταγωνιστούν τους υπολογιστές από άποψη υλικού (hardware). Το διαδίκτυο άλλαξε επίσης τον τρόπο που αγοράζουμε βιντεοπαιχνίδια ριζικά, με την πλειοψηφία των βιντεοπαιχνιδιών να είναι διαθέσιμα προς αγορά ψηφιακά και όχι (μόνο) σε φυσική μορφή. Με το πάτημα ενός κουμπιού από τον υπολογιστή μας μπορούμε να επιλέξουμε το βιντεοπαιχνίδι που θέλουμε, να το αγοράσουμε και να το «κατεβάσουμε» στην κονσόλα που επιθυμούμε χωρίς κόπο. (History of video games, 2018)

2.2.7 Τα βιντεοπαιχνίδια του σήμερα

Πλησιάζοντας όλο και περισσότερο το σήμερα, οι τάσεις στα βιντεοπαιχνίδια δεν γνωρίζουν κάποια δραματική αλλαγή στην κατεύθυνσή τους, με το ευρύ κοινό να προσανατολίζεται σε βιντεοπαιχνίδια πολλών παικτών μέσω διαδικτύου (online multiplayer games). Τα μεγάλα βήματα ανάπτυξης της τεχνολογίας με το πέρασμα των χρόνων έχουν βοηθήσει αδιαμφισβήτητα την εξέλιξη των βιντεοπαιχνιδιών (κυρίως στο θέμα των γραφικών), με τους υπολογιστές μεν να έχουν το πάνω χέρι σε αυτό το θέμα αλλά με σημαντική διαφορά χρημάτων από μία κονσόλα. (History of video games, 2018)

Κάποια στατιστικά στοιχεία (ενημερωμένα μέχρι τον Απρίλιο του 2017) δείχνουν ότι το 65% των οικογενειών διαθέτουν κάποια συσκευή για να παίξει βιντεοπαιχνίδια και το 48% διαθέτει συσκευή αποκλειστικά για βιντεοπαιχνίδια. Ακόμα, η μέση ηλικία ενός παίκτη (gamer) είναι τα 35 χρόνια, ενώ ο μέσος χρόνος που ασχολούνται με βιντεοπαιχνίδια ανέρχεται στα 13 χρόνια. Ένα άλλο ενδιαφέρον στοιχείο είναι ποιες συσκευές προτιμάει ο κόσμος: 56% παίζει σε υπολογιστή, 53% σε κονσόλα, 36% σε «έξυπνα» κινητά (smartphones), 31% σε ασύρματες συσκευές και 17% σε φορητές κονσόλες. (Lofgren, 2017)

2.2.8 Τι μας επιφυλάσσει το μέλλον

Η βιομηχανία των παιχνιδιών έχει αρχίσει να στρέφεται στα πεδία της εικονικής πραγματικότητας (VR – Virtual Reality) και της τεχνητής νοημοσύνης (AI – Artificial Intelligence) προκειμένου να κάνει το βήμα παραπέρα. Προσπάθειες όσων αφορά την εικονική πραγματικότητα έχουν γίνει με την κυκλοφορία του προϊόντος Oculus Rift για το PC το 2016 αλλά η βιομηχανία αυτή είναι ακόμη νέα και δεν έχει αποφέρει αποτελέσματα. Το μέλλον φαντάζει φωτεινό για τα βιντεοπαιχνίδια με πολλές δυνατότητες και τεχνολογίες να εμφανίζονται μέρα με τη μέρα, δεν μπορούμε καν να φανταστούμε την κατάστασή τους σε 5-10 χρόνια από τώρα.



Εικόνα 2-5: Το προϊόν Oculus Rift CV1 (Oculus Rift, 2017)

3. ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΗΣ

Σε αυτό το κεφάλαιο θα παρουσιαστούν οι τεχνολογίες που χρησιμοποιήθηκαν για το στήσιμο της εφαρμογής.

3.1 HTML

3.1.1 Εισαγωγή στα βασικά στοιχεία

Η γλώσσα HTML (Hyper Text Markup Language – Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης που χρησιμοποιείται για απεικόνιση ιστοσελίδων σε κάποιο φυλλομετρητή όπως Google Chrome. Η γλώσσα αυτή μαζί με τις τεχνολογίες CSS και JavaScript (θα αναλυθούν στη συνέχεια) συνθέτουν το βασικό τρίπτυχο δομής του Παγκόσμιου Ιστού (World Wide Web). Οι προδιαγραφές της τηρούνται από την οργάνωση W3C (World Wide Web Consortium) η οποία είναι υπεύθυνη για όλες τις βασικές αρχές των τεχνολογιών που απαρτίζουν τον Παγκόσμιο Ιστό. (HTML, 2018)

Το βασικό στοιχείο της γλώσσας αυτής είναι οι ετικέτες (τα λεγόμενα tags), οι οποίες είναι λέξεις της αγγλικής και περικλείονται από τα σύμβολα < (μεγαλύτερο από) και > (μικρότερο από) (π.χ. <body>). Συνήθως οι ετικέτες λειτουργούν σε ζευγάρια, με την πρώτη ετικέτα να λέγεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης και ξεχωρίζουν με την πρόσθεση του συμβόλου / (π.χ. </body>). Η κύρια λειτουργία της HTML είναι να «λέει» στον φυλλομετρητή πως να απεικονίσει την ιστοσελίδα. Με λίγα λόγια, είναι υπεύθυνη για το περιεχόμενο και το βασικό στήσιμο μιας ιστοσελίδας. (HTML, 2018)

Η HTML παρέχει τη δυνατότητα για απεικόνιση πολλών διάφορων αντικειμένων και στοιχείων, όπως πίνακες, εικόνες, παραγράφους, φόρμες, συνδέσμους κ.α. με τη χρήση των κατάλληλων ετικετών (π.χ. για πίνακες χρησιμοποιούμε <table>). Επίσης, παρέχει και δυνατότητα μορφοποίησης του κάθε στοιχείου δίνοντάς του ιδιότητες (attributes). Κάποια παραδείγματα ιδιοτήτων είναι id, class, style κ.α.. Οι μορφοποιήσεις που μπορούν να γίνουν είναι όμως πολύ βασικές και λιγοστές, γι'

αυτό η περαιτέρω μορφοποίηση της εμφάνισης της ιστοσελίδας γίνεται μέσω CSS που παρέχει περισσότερα και πιο εξειδικευμένα εργαλεία και δυνατότητες. (HTML, 2018)

3.1.2 Χρήση της HTML στην εφαρμογή

Η εφαρμογή χρησιμοποιεί της τελευταία έκδοση της γλώσσας HTML, HTML5, και όλες οι ιστοσελίδες που απαρτίζουν την εφαρμογή είναι φτιαγμένες πάνω σε αυτή. Αποτελούνται από μία κεφαλίδα (head), κάποια στυλ απεικόνισης (style), το κυρίως σώμα (body) και συνήθως κάποια σενάρια εντολών (scripts) σε γλώσσα JavaScript. Επίσης Τα βιντεοπαιχνίδια βασισμένα σε φυλλομετρητή χρησιμοποιούν πάντα την ετικέτα <canvas> που παρέχει η HTML5. Η ετικέτα αυτή παρέχει τη δυνατότητα στον browser να απεικονίσει γραφικά σε μια ιστοσελίδα. Πρέπει να αναφερθεί ότι ο πραγματικός σχεδιασμός των γραφικών γίνεται μέσω JavaScript, το canvas απλώς θεσπίζει το χώρο στην ιστοσελίδα που θα εμφανιστούν τα γραφικά.

3.2 CSS

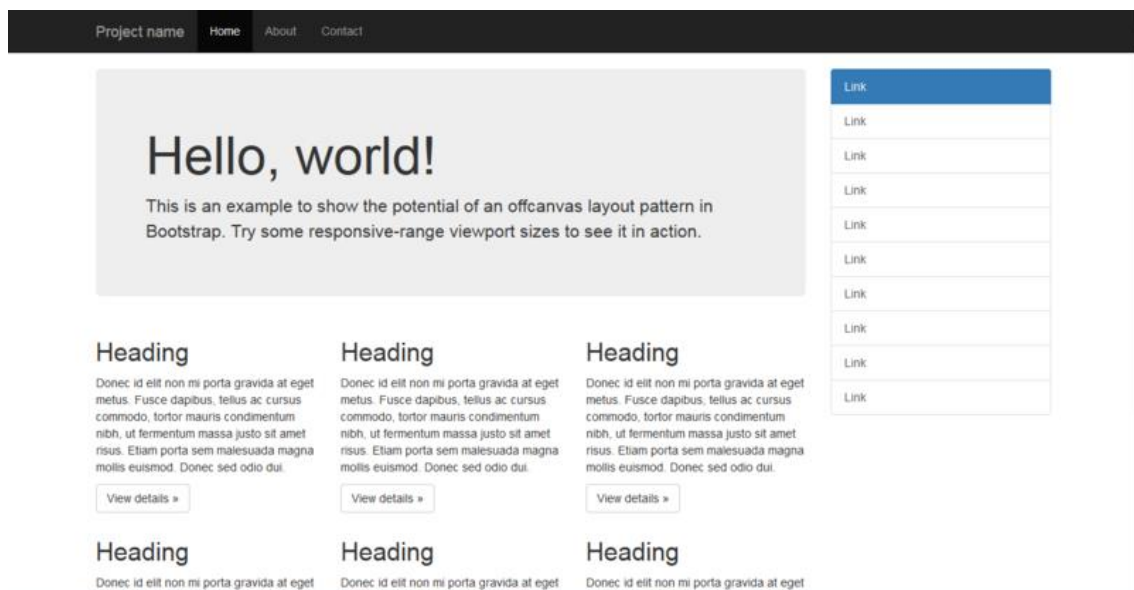
3.2.1 Εισαγωγή στα βασικά στοιχεία

Η γλώσσα CSS (Cascading Style Sheets – Διαδοχικά Φύλλα Ύφους) είναι μια γλώσσα υπολογιστή η οποία χρησιμοποιείται για να καθορίσει τη μορφή και την εμφάνιση των στοιχείων και αντικειμένων μιας ιστοσελίδας που έχει γραφτεί με μία γλώσσα σήμανσης, όπως η HTML. Στις μέρες μας η χρήση της είναι απαραίτητη αν θέλουμε να στήσουμε μια καλοσχεδιασμένη ιστοσελίδα που χτυπάει στο μάτι και προσελκύει επισκέπτες. Οι προδιαγραφές της τηρούνται και αυτές από την οργάνωση W3C. (Cascading Style Sheets, 2018)

Η γλώσσα CSS προσφέρει άπειρες δυνατότητες που με την χρήση απλής HTML είναι αδύνατο να εφαρμοστούν. Μπορούμε να αναθέσουμε ειδικές ιδιότητες σε οποιοδήποτε στοιχείο μιας ιστοσελίδας θέλουμε, ακόμα και σε πολλά μαζί, αρκεί να πούμε για ποιο στοιχείο προορίζονται οι αλλαγές που έχουμε γράψει. Για να το κάνουμε αυτό υπάρχουν πολλοί τρόποι:

- Καθορίζοντας την ετικέτα του στοιχείου που θέλουμε, π.χ. όλα τα στοιχεία με ετικέτα <h1> (όλες οι επικεφαλίδες h1).
- Καθορίζοντας ένα συγκεκριμένο attribute ενός στοιχείου, συνήθως με το id ή το class.
- Στοιχεία με βάση το DOM (Document Object Manager) (θα αναλυθεί περαιτέρω στο κεφάλαιο 3.3.1).

Μία ακόμα δυνατότητα που μας προσφέρει η CSS είναι ότι μπορούμε να έχουμε όλες τις στυλιστικές αλλαγές μαζεμένες σε ένα μέρος του κώδικα, μέσα στην ετικέτα <style> της HTML σελίδας μας. Έτσι, καθιστούμε τον κώδικά μας ευανάγνωστο και τακτοποιημένο. Μπορούμε ακόμα και να ξεχωρίσουμε τον κώδικα που περιέχει όλες τις αλλαγές σε ένα καινούριο αρχείο με την κατάληξη .css και να δημιουργήσουμε ένα σύνδεσμο στο HTML αρχείο με την ετικέτα <link> και να προσδιορίσουμε εκεί τη διαδρομή του καινούριου αρχείου. Το 99% των εφαρμογών παγκοσμίως ακολουθούν αυτή την πρακτική για μεγαλύτερη διαχειριστικότητα του project. (Cascading Style Sheets, 2018)



Εικόνα 3-1: Παράδειγμα ιστοσελίδας με χρήση CSS (Bootstrap Example, 2015)

3.2.2 Χρήση της CSS στην εφαρμογή

Η παρούσα εφαρμογή υιοθετεί χρήση της CSS στην μορφή που επεξηγήθηκε παραπάνω, αλλά κάνει χρήση και μιας από τις δημοφιλέστερες συλλογές εργαλείων ανοικτού κώδικα (open-source framework) ονόματι Bootstrap. Το Bootstrap αποτελείται από αρχεία τύπου CSS και περιλαμβάνει διάφορους

αποτυπωμένους κανόνες και στυλ για στοιχεία όπως φόρμες, κουμπιά κ.α.. Για να λειτουργήσει πρέπει να προστεθεί ένας σύνδεσμος προς τα αρχεία της Bootstrap στο εκάστοτε αρχείο HTML και αυτά μπορεί να είναι είτε τοπικά κατεβάζοντας και έχοντας τα σε ένα φάκελο είτε από κάποιο διαδικτυακό τόπο που τα διαθέτει. Η παρούσα εφαρμογή χρησιμοποιεί την έκδοση 3.3.7 της CSS και όχι την τελευταία (4.1.3), καθώς αυτή παρουσίαζε προβλήματα συμβατότητας. Η κύρια χρήση της ήταν το στήσιμο της οθόνης πρόσβασης (login page), καθώς και όλων των κουμπιών την εφαρμογής.

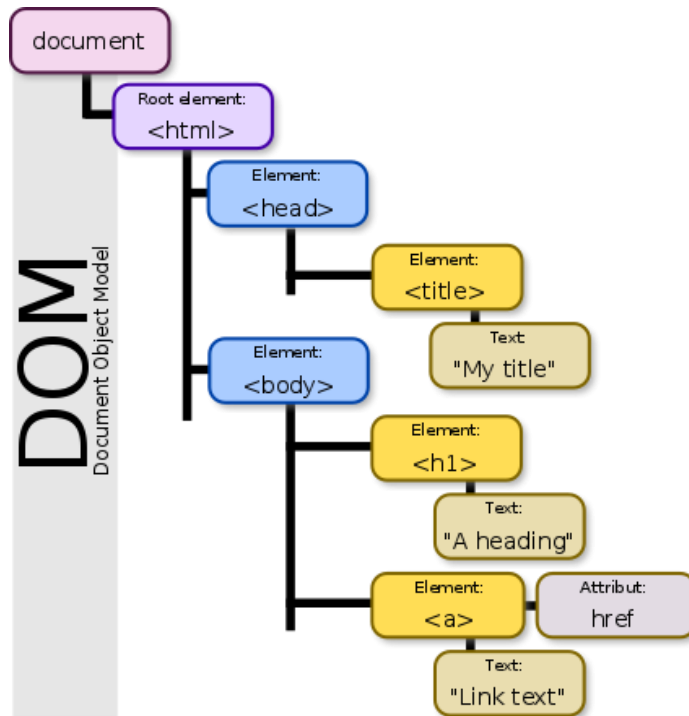
3.3 JavaScript

3.3.1 Εισαγωγή στα βασικά στοιχεία

Η γλώσσα JavaScript (ή JS σε συντομογραφία) είναι μια πολύ διαδεδομένη διερμηνευμένη γλώσσα προγραμματισμού (κοινώς σκριπτογλώσσα). Αποτελεί το τρίτο μέλος των τεχνολογιών που απαρτίζουν τον Παγκόσμιο Ιστό. Παρά το όνομα, η γλώσσα αυτή δεν παρουσιάζει καμία ομοιότητα με τη γλώσσα Java, αντιθέτως είναι επηρεασμένη συντακτικά από τη γλώσσα C. Πέρα όμως από το συναρτησιακό στυλ προγραμματισμού της C, η JavaScript υποστηρίζει και αντικειμενοστρεφές στυλ προγραμματισμού. Το 1996 η εταιρεία Netscape κατέθεσε τη JavaScript στον οργανισμό τυποποίησης γλωσσών προγραμματισμού ECMA, ο οποίος ένα χρόνο αργότερα δημοσίευσε την επίσημη τυποποίηση της γλώσσας, ECMAScript. (JavaScript, 2018)

Σκοπός της γλώσσας αυτής είναι να χειρίζεται αιτήματα (requests) και γεγονότα (events) στην πλευρά του χρήστη (client-side), ο οποίος αναλαμβάνει να μεταφράσει αυτόν τον κώδικα και να τον εκτελέσει χρησιμοποιώντας τον ενσωματωμένο διερμηνέα για JavaScript. Αυτό σημαίνει ότι ο κώδικας αυτός είναι εκτελέσιμος και όταν δεν υπάρχει σύνδεση στο διαδίκτυο. Ενσωματώνεται εύκολα στο αρχείο .html, αρκεί να γραφτεί μέσα στην ετικέτα <script> και να δηλωθεί το attribute style ως text/JavaScript. Μπορούμε επίσης να ακολουθήσουμε την ίδια τεχνική που αναφέρθηκε για τη CSS, αποθηκεύοντας τον κώδικα JavaScript σε ξεχωριστό αρχείο με την κατάληξη .js και με αναφορά στη διαδρομή του αρχείου αυτού ο κώδικάς μας θα εκτελεστεί. (JavaScript, 2018)

Μέσω της γλώσσας αυτής μπορούμε επίσης να χειριστούμε αντικείμενα του DOM. Το DOM είναι μια διεπαφή προγραμματισμού εφαρμογών (API) η οποία χειρίζεται το έγγραφο (δηλ. ιστοσελίδα) ως ένα δέντρο, με το κάθε αντικείμενό του να αποτελεί ένα παιδί (κόμβος - node). Ως κορυφή του δέντρου είναι το ίδιο το έγγραφο (document) το οποίο δημιουργείται με την πρώτη εμφάνιση/εκτέλεση της σελίδας. Με τον τρόπο αυτό καθίσταται εύκολη η πρόσβαση σε οποιοδήποτε αντικείμενο της ιστοσελίδας θέλουμε, ακολουθώντας ιεραρχικά το δέντρο.



Εικόνα 3-2: Αναπαράσταση δέντρου DOM (DOM Model, 2012)

3.3.2 JavaScript και διακομιστής

Μία άλλη δυνατότητα που μας παρέχει η JavaScript είναι επικοινωνία με τον διακομιστή (server) ασύγχρονα (δηλ. χωρίς να το βλέπει ο χρήστης, στα παρασκήνια) για αποστολή και παραλαβή στοιχείων χωρίς να χρειαστεί ανανέωση της ιστοσελίδας για εμφάνιση των αποτελεσμάτων. Αυτό επιτυγχάνεται με τη χρήση ενός σετ τεχνολογιών AJAX (Asynchronous JavaScript And XML). Μέσα στο πακέτο αυτό περιλαμβάνεται και ένα ειδικό αντικείμενο για ασύγχρονη επικοινωνία, το XMLHttpRequest, του οποίου γίνεται χρήση και στην παρούσα εφαρμογή. Το αντικείμενο αυτό κάνει χρήση του πρωτόκολλου HTTP (Hyper-Text

Transfer Protocol) για να αποστείλει ή να λάβει δεδομένα από τον server ασύγχρονα, είτε με μέθοδο GET είτε με POST. (AJAX, 2018)

Όπως προαναφέρθηκε, η JavaScript εκτελείται στον πελάτη και όχι στον διακομιστή. Έχουν αναπτυχθεί όμως και πακέτα με συναρτήσεις και κώδικα για χρήση από την πλευρά του server, όπως το πακέτο Node.js που κυκλοφόρησε το 2009. Το πακέτο αυτό καθιστά δυνατή τη χρήση JavaScript για συγγραφή κώδικα ικανό να εκτελεστεί από τον server, δημιουργώντας έτσι εφαρμογές εξ'ολοκλήρου χτισμένες σε γλώσσα JavaScript. (Node.js, 2018)

3.3.3 Χρήση της JavaScript στην εφαρμογή

Η JavaScript ήταν η κύρια γλώσσα που χρησιμοποιήθηκε στην παρούσα εφαρμογή, καθώς και τα δύο παιχνίδια είναι γραμμένα σε αυτή. Μία θεμελιώδης αρχή όσων αφορά την υλοποίηση των παιχνιδιών αυτών βασίστηκε σε μία συνάρτηση της JavaScript, `setInterval()`, και κατ' επέκταση στη συνάρτηση που τείνει να την αντικαταστήσει, `requestAnimationFrame()`. Η λογική πίσω από αυτές τις δύο συναρτήσεις είναι η εξής: κάλεσε τη συνάρτηση που θα σου καθορίσω ανά χ χιλιοστά του δευτερολέπτου. Η διαφορά τους βρίσκεται στο χρόνο, με την πρώτη να χρειάζεται να καθορίσει ο προγραμματιστής των χρόνων ενώ με τη δεύτερη ο χρόνος αφήνεται να προσδιοριστεί από τον browser. Το πλεονέκτημα της δεύτερης φαίνεται ολοένα και περισσότερο στις μέρες μας, με τις αυξημένες υπολογιστικές δυνατότητες των υπολογιστών και με τους υψηλούς ρυθμούς ανανέωσης των σύγχρονων οθονών να προκαλούν μη συγχρονισμό εφαρμογής και φυσικής ικανότητας του υπολογιστή-οθόνης να απεικονίσει την επόμενη εκτέλεση της συνάρτησης. (Understanding JavaScript's `requestAnimationFrame()` method for smooth animations, 2016)

Οτιδήποτε άλλο φαίνεται στα παιχνίδια είναι εξ'ολοκλήρου υλοποιημένο με JavaScript μέσω του αντικειμένου `canvas`, που εκπροσωπεί την περιοχή που θα εμφανιστεί το παιχνίδι. Ο χρωματισμός του φόντου, οι κινήσεις του φιδιού, της μπάλας και της ρακέτας καθώς και όλα τα σχήματα είναι σχεδιασμένα με συναρτήσεις JavaScript.

3.4 PHP

3.4.1 Εισαγωγή στα βασικά στοιχεία

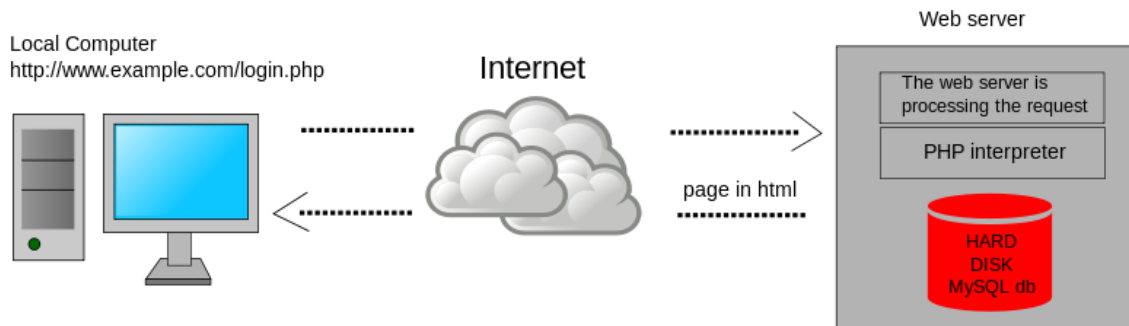
Η γλώσσα PHP (Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού με βάση τα σενάρια για εκτέλεση σε κάποιο διακομιστή (server-side scripting language). Τα αρχεία που περιέχουν κώδικα PHP πρέπει να έχουν την κατάλληλη κατάληξη (.php) έτσι ώστε ο server να μπορέσει να μεταφράσει τον κώδικα. Προφανώς, αν επιχειρήσουμε να τρέξουμε ένα script PHP εκτός server τότε ο κώδικας θα αντιμετωπιστεί ως απλό κείμενο. Ο πιο διαδεδομένος web server αυτή τη στιγμή είναι ο Apache HTTP Server. Μέχρι πρόσφατα η PHP δούλευε χωρίς κάποια αυθεντική γραμμένη τυποποίηση, με αυτή να έρχεται μόλις το 2014. Η γλώσσα PHP αποτελεί ελεύθερο λογισμικό. (PHP, 2018)

Η PHP υποστηρίζει και αντικειμενοστρεφή προγραμματισμό από την έκδοση PHP 5 και έπειτα, η οποία θεωρείται και η πρώτη μεγάλη βασική έκδοσή της. Η έκδοση 5.5 ήταν η πρώτη που έγινε διαθέσιμη στο λογισμικό Windows. Με το πέρασμα των χρόνων, η γλώσσα είχε αλλάξει πολύ εσωτερικά, με αποτέλεσμα την κυκλοφορία της νέας μεγάλης έκδοσης, PHP 7, η οποία συνεχίζει να ανανεώνεται μέχρι σήμερα.

Ότι κώδικα επιθυμούμε να μεταφραστεί από τον διερμηνέα (interpreter) της PHP πρέπει να γραφτεί μέσα σε οριοθέτες. Οι γνωστότεροι είναι ο `<?php` για άνοιγμα και `?>` για κλείσιμο της περιοχής. Οτιδήποτε εκτός αυτής της περιοχής θα θεωρηθεί ως κώδικας HTML από τον διερμηνέα και δεν θα εκτελεστεί. Θα μπορούσαμε να χαρακτηρίσουμε τη γλώσσα αυτή ως ένα φίλτρο το οποίο δέχεται ένα αρχείο που περιέχει κείμενο και/ή PHP εντολές στην είσοδό του και παράγει έναν άλλο τύπο δεδομένων στην έξοδό του. Η έξοδος συνήθως είναι κώδικας HTML αλλά μπορεί να είναι και κάτι άλλο (π.χ. JSON, XML κ.α.). (PHP, 2018)

Η κύρια χρήση της είναι να παρέχει δυναμικό περιεχόμενο από ένα web server στον πελάτη, γι' αυτό και ο κώδικάς της αναφέρεται στον server. Βέβαια η PHP μπορεί να χρησιμοποιηθεί και για εφαρμογές πέρα από client-server, καθώς είναι μια γλώσσα γενικής χρήσης. Λόγω της ευρείας χρήσης της, έχουν αναπτυχθεί πολλά διαδικτυακά πακέτα λογισμικού (web frameworks) βασισμένα στην PHP, όπως Laravel και CakePHP, που βοηθάνε τους χρήστες να χτίζουν γρήγορα εφαρμογές από την αρχή με μεγάλη ευκολία. Αυτή τη στιγμή η PHP

χρησιμοποιείται από το 83,5% των ιστοσελίδων ως η κύρια γλώσσα προγραμματισμού από την πλευρά του διακομιστή. Πολλά διαδικτυακά προγράμματα διαχείρισης περιεχομένου (web Content Management Systems - CMS), όπως Wordpress, Joomla, Drupal κ.α., είναι φτιαγμένα σε PHP γλώσσα. Επίσης, πολλές μεγάλες εταιρείες και οργανισμοί κάνουν χρήση της PHP, με κάποια παραδείγματα να είναι η Google, το Facebook και η Yahoo. (PHP, 2018)



Εικόνα 3-3: Σχεδιάγραμμα μιας δυναμικής σελίδας με χρήση PHP (Scheme Dynamic Page, 2012)

3.4.2 Χρήση της PHP στην εφαρμογή

Στην παρούσα εφαρμογή η PHP χρησιμοποιήθηκε ως μέσο επικοινωνίας του χρήστη με τον server, για δύο κύριες χρήσεις:

- Εξακρίβωση στοιχείων χρήστη κατά την είσοδό του στην εφαρμογή με αυτά στη βάση δεδομένων και αποκλεισμό του από αυτή αν υπέβαλε λανθασμένα στοιχεία.
- Αποθήκευση των πόντων που μάζεψε ανά προσπάθεια σε κάθε παιχνίδι στη βάση δεδομένων.

Για να τρέξουν τα script της PHP εγκαταστάθηκε το ελεύθερο πακέτο διαδικτυακών διακομιστών XAMPP, που περιέχει διερμηνέα PHP και τον Apache web server μεταξύ άλλων (ανάλυση του πακέτου αυτού θα γίνει σε επόμενο κεφάλαιο). Συγκεκριμένα, οι εκδόσεις τους είναι: Apache 2.4.33 και PHP 7.2.7. Η επικοινωνία με τον server γίνεται κυρίως με τη μέθοδο POST για μεγαλύτερη ασφάλεια.

3.5 MySQL

3.5.1 Εισαγωγή στα βασικά στοιχεία

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοικτού κώδικα (open-source relational database management system - RDBMS). Για τη συγγραφή της έχουν χρησιμοποιηθεί οι γλώσσες C και C++. Είναι μια πολύ διάσημη βάση δεδομένων με πάνω από 11 εκατομμύρια εγκαταστάσεις ανά τον κόσμο, όπως επίσης χρησιμοποιείται και από μεγάλες εταιρείες όπως Google, Facebook, Twitter και YouTube. Η MySQL ήταν ιδιοκτησία της σουηδικής εταιρείας MySQL AB, η οποία τώρα ανήκει στην εταιρεία Oracle. Υπάρχουν δύο εκδόσεις διαθέσιμες για το κοινό: η δωρεάν Community Edition και η επιπληρωμής Enterprise Edition, με την Enterprise να προσφέρει περισσότερες λειτουργίες από τη δωρεάν έκδοση, καθώς και ευρύτερη βοήθεια (support) στους πελάτες. (MySQL, 2018)

Η MySQL, προφανώς, δέχεται εντολές της γλώσσας SQL (Structured Query Language) και υποστηρίζει τα πιο βασικά εργαλεία και λειτουργίες που χρειάζεται μια βάση. Κάποια παραδείγματα αυτών είναι triggers, cursors, views και stored procedures (η υποστήριξη των οποίων όμως ξεκίνησε μόλις στην τελευταία έκδοση). Ακόμα, υποστηρίζει και άλλες πιο εξεζητημένες λειτουργίες όπως subqueries (εμφωλευμένα Select ερωτήματα) και events (προγραμματισμό και επαναλαμβανόμενο τρέξιμο ενός ερωτήματος SQL ανά κάποιο χρονικό διάστημα). (MySQL, 2018)

3.5.2 Χρήση της MySQL στην εφαρμογή

Η βάση δεδομένων της εφαρμογής σχεδιάστηκε με τη βοήθεια της MySQL. Ο απαιτούμενος MySQL Server είναι μέρος του πακέτου XAMPP που χρησιμοποιήθηκε σε αυτήν την εφαρμογή, έκδοση 10.1.34. Για την ευκολότερη δημιουργία και διαχείριση της βάσης χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE) MySQL Workbench CE.

3.6 Προγράμματα που χρησιμοποιήθηκαν

Κινητήριος μοχλός της εφαρμογής ήταν το πακέτο προγραμμάτων ΧΑΜΡΡ. Τα αρχικά του σημαίνουν με τη σειρά:

- Χ αναφέρεται στην ιδιότητα cross-platform, δηλαδή τη δυνατότητα του λογισμικού να τρέχει σε όλα τα λειτουργικά συστήματα
- Α από τον διακομιστή Apache http server
- Μ από το MySQL (στις νεότερες εκδόσεις αναφέρεται στο MariaDB)
- Ρ από τη γλώσσα PHP
- Ρ από τη γλώσσα PERL

Αυτό το πακέτο μας παρείχε με τους πολυπόθητους server για τη βάση δεδομένων MySQL και για τη χρήση της γλώσσας PHP (Apache http server και διερμηνέα της γλώσσας PHP). Χρησιμοποιήθηκε η νεότερη έκδοση του πακέτου ΧΑΜΡΡ που υπήρξε διαθέσιμη τον καιρό της ανάπτυξης της εφαρμογής, 7.2.7. Αυτή περιείχε μέσα την έκδοση 7.2.7 της PHP και την έκδοση 10.1.34 της MySQL.

Για τη διαχείριση της βάσης δεδομένων χρησιμοποιήθηκε το πρόγραμμα MySQL Workbench Community Edition, έκδοση 8.0, ένα IDE που διαθέτει η ίδια η Oracle ελεύθερα στο κοινό. Το πρόγραμμα αυτό χρησιμοποιήθηκε αντί του IDE PhpMyAdmin που προσφέρει το πακέτο ΧΑΜΡΡ ως κύριο λόγο την προσωπική διευκόλυνση του συγγραφέα στην εκτέλεση των queries (ερωτημάτων) προς τη βάση κατά τον έλεγχο της εφαρμογής.

Επίσης, με το μόνο γνώμονα επιλογής να είναι η εξοικείωση του συγγραφέα, επιλέχθηκαν τα προγράμματα Notepad++ ως επεξεργαστής κειμένου και οι browsers Mozilla Firefox, Firefox Developer Edition, Google Chrome και Internet Explorer ως περιβάλλοντα απεικόνισης των ιστοσελίδων.

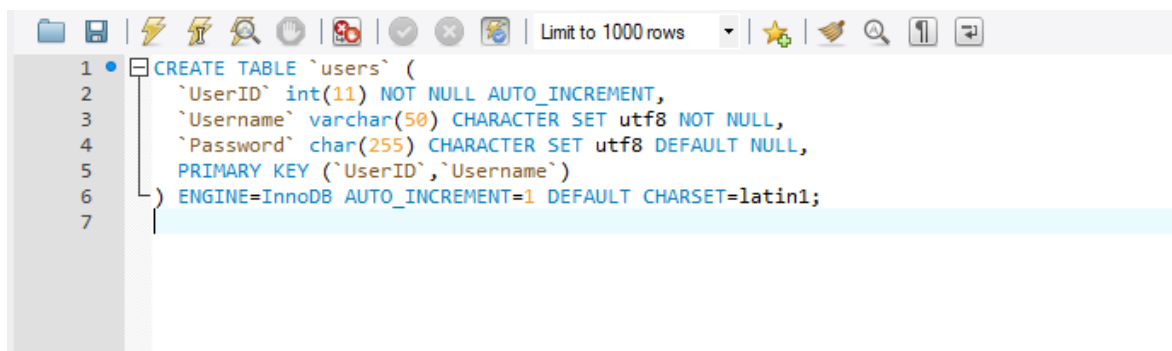
4. ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΣΦΑΛΕΙΑ

Σε αυτό το κεφάλαιο θα γίνει μια παρουσίαση της βάσης δεδομένων της εφαρμογής, καθώς και ανάλυση των ενεργειών που πραγματοποιήθηκαν για να αυξηθεί η ασφάλειά της.

4.1 Σχεδιασμός βάσης δεδομένων

Πρώτο βήμα που πραγματοποιήθηκε στην ανάπτυξη της εφαρμογής ήταν ο προσδιορισμός του μοντέλου της βάσης δεδομένων. Το πόρισμα ήταν η δημιουργία δύο πινάκων: ένας πίνακας θα κράταγε τα στοιχεία των χρηστών της εφαρμογής και ο άλλος πίνακας θα κράταγε τους πόντους των χρηστών ανά προσπάθεια στα διαθέσιμα παιχνίδια. Οι δύο αυτοί πίνακες καλύπτουν επαρκώς την εφαρμογή όπως είναι την παρούσα στιγμή. Ακολουθεί αναλυτική επεξήγηση των δύο πινάκων:

- Πίνακας users



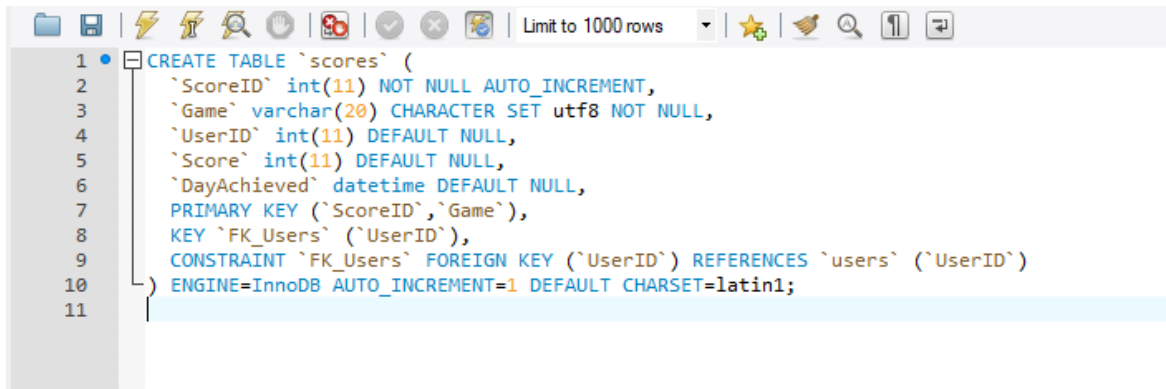
```
1 CREATE TABLE `users` (  
2   `UserID` int(11) NOT NULL AUTO_INCREMENT,  
3   `Username` varchar(50) CHARACTER SET utf8 NOT NULL,  
4   `Password` char(255) CHARACTER SET utf8 DEFAULT NULL,  
5   PRIMARY KEY (`UserID`,`Username`)  
6 )  
7 ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

Εικόνα 4-1: Script δημιουργίας πίνακα users

Ο πίνακας αποτελείται από 3 πεδία όπως φαίνεται από την εικόνα, ένα πεδίο UserID τύπου Int(11) το οποίο αυξάνεται αυτόματα με κάθε νέα εγγραφή (auto increment), ένα πεδίο Username τύπου Varchar(50) χαρακτήρων (αλφαριθμητικό) που κρατάει το όνομα του χρήστη στην εφαρμογή και τέλος ένα πεδίο Password τύπου Char(255) που κρατάει τον κωδικοποιημένο κωδικό του χρήστη. Το πρωτεύον κλειδί (PK - primary key) αποτελείται από τα πρώτα δύο πεδία του

πίνακα, UserID και Username, έτσι ώστε να υπάρχει μοναδικότητα στο όνομα χρήστη.

- Πίνακας scores



```
1 CREATE TABLE `scores` (  
2   `ScoreID` int(11) NOT NULL AUTO_INCREMENT,  
3   `Game` varchar(20) CHARACTER SET utf8 NOT NULL,  
4   `UserID` int(11) DEFAULT NULL,  
5   `Score` int(11) DEFAULT NULL,  
6   `DayAchieved` datetime DEFAULT NULL,  
7   PRIMARY KEY (`ScoreID`,`Game`),  
8   KEY `FK_Users` (`UserID`),  
9   CONSTRAINT `FK_Users` FOREIGN KEY (`UserID`) REFERENCES `users` (`UserID`)  
10  ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;  
11
```

Εικόνα 4-2: Script δημιουργίας πίνακα scores

Ο πίνακας αποτελείται από 5 πεδία, όπως φαίνεται από την εικόνα. Το πρώτο πεδίο είναι το ScoreID τύπου Int(11) το οποίο αυξάνει αυτόματα με κάθε εγγραφή. Το δεύτερο πεδίο είναι το Game τύπου Varchar(20) το οποίο καθορίζει σε ποιο παιχνίδι αναφέρεται η κάθε εγγραφή, με τις τιμές να είναι είτε snake είτε arkanoid. Το τρίτο πεδίο είναι το UserID τύπου Int(11) το οποίο καθορίζει τον χρήστη που μάζεψε τους αναφερόμενους πόντους. Το τέταρτο πεδίο είναι το Score τύπου Int(11) το οποίο κρατάει τους πόντους και τέλος το πέμπτο πεδίο είναι το DayAchieved τύπου Datetime το οποίο κρατάει την ημερομηνία που πραγματοποιήθηκε η εγγραφή στη βάση. Το πρωτεύον κλειδί το αποτελούν τα δύο πρώτα πεδία, ScoreID και Game, ενώ υπάρχει και διασύνδεση των δύο πινάκων τις εφαρμογής μέσω του ξένου κλειδιού (FK - foreign key) UserID.

4.2 Ασφάλεια εφαρμογής

Η ασφάλεια είναι πάντα ένα πολύ σημαντικό κομμάτι μιας εφαρμογής ανεξαρτήτως μεγέθους και δεδομένων και πρέπει πάντα τα δεδομένα των χρηστών να προστατεύονται. Παρόλο που η εφαρμογή δεν πραγματεύεται ευαίσθητα δεδομένα, έχουν εφαρμοστεί αποδεδειγμένοι αλγόριθμοι και τεχνικές ασφάλειας. Οι κωδικοί των χρηστών κωδικοποιούνται πριν αποθηκευτούν στη

Κατασκευή Διαδικτυακού Ιστότοπου Παιχνιδιών

βάση δεδομένων με αλγόριθμο κωδικοποίησης ο οποίος μετατρέπει τον κωδικό σε μια ακολουθία 60 τυχαίων χαρακτήρων. Επίσης, κατά τη μετάβαση στις διάφορες σελίδες γίνεται πρώτα έλεγχος αν ο χρήστης έχει πραγματοποιήσει είσοδο στη εφαρμογή, αλλιώς τον ανακατευθύνει στην σελίδα εισόδου.

5.ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Το κεφάλαιο αυτό ξεκινά με την επεξήγηση των στόχων και των κανόνων των παιχνιδιών που αναπτύχθηκαν σε αυτή την εφαρμογή. Έπειτα γίνεται επεξήγηση του συνόλου της εφαρμογής, αναλύοντας κάθε ιστοσελίδα ξεχωριστά και με τη λογική σειρά που παρουσιάζονται στον χρήστη.

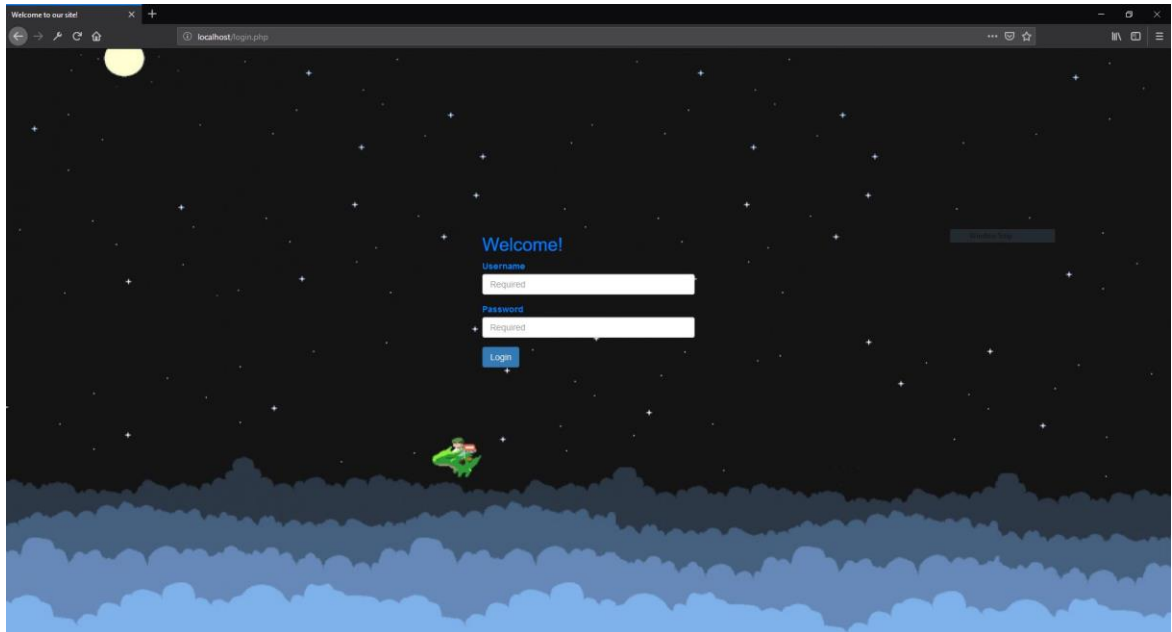
5.1 Λίγα λόγια για τα παιχνίδια της εφαρμογής

Το ένα από τα δύο παιχνίδια που είναι διαθέσιμα για τον χρήστη να παίξει είναι το γνωστό «φιδάκι» (snake). Σκοπός του παιχνιδιού είναι να οδηγήσεις το φιδάκι ώστε να «φάει» το φαγητό που παρουσιάζεται στην οθόνη. Καθώς το φιδάκι τρώει το φαγητό, το σώμα του μεγαλώνει, αυξάνοντας έτσι τη δυσκολία του παιχνιδιού. Αυτό γίνεται γιατί αν το φιδάκι χτυπήσει μέρος του σώματός του ή το τείχος (δηλαδή την άκρη της οθόνης – τα περιθώρια του παιχνιδιού) τότε ο παίκτης έχει χάσει και το παιχνίδι τελειώνει. Παραλλαγές του παιχνιδιού έχουν αναπτυχθεί ανά τα χρόνια, με προσθήκες όπως ειδικές δυνάμεις (power-ups) (το φιδάκι να πηγαίνει πιο αργά, μείωση μήκους φιδιού κ.α.), εμποδίων κ.α., αλλά ο βασικός τρόπος παιχνιδιού έχει παραμείνει ο ίδιος. Ο κώδικας για τα δύο παιχνίδια έχει παρθεί από το site ελεύθερου λογισμικού Github (Github, 2018) και έχουν γίνει κατάλληλες επεμβάσεις και βελτιώσεις για τις ανάγκες της πτυχιακής.

Το άλλο διαθέσιμο παιχνίδι είναι το λεγόμενο Arkanoid. Σκοπός του παιχνιδιού είναι να χτυπήσεις όλα τα τούβλα (bricks) στο πάνω μέρος της οθόνης με την μπάλα, χρησιμοποιώντας ένα κομμάτι ξύλου (paddle) για να εμποδίσεις την μπάλα να φύγει από το χώρο του παιχνιδιού και να την ανακατευθύνεις ξανά προς τα τούβλα. Χτυπώντας τα τούβλα, αυτά εξαφανίζονται και όταν εξαφανιστούν όλα τότε ο παίκτης προχωρά στην επόμενη πίστα η οποία περιέχει διαφορετικό σχεδιασμό τούβλων. Υπάρχουν πολλά διαφορετικά είδη τούβλων που το καθένα έχει διαφορετική αντοχή, που σημαίνει ότι χρειάζεται περισσότερα χτυπήματα με την μπάλα για σπάσει. Ο παίκτης κερδίζει όταν φτάσει στην 4^η και τελευταία πίστα και σπάσει όλα τα τούβλα εκεί, αλλιώς αν χάσει και τις 3 ζωές που του δίνονται

κατά την πορεία του παιχνιδιού, τότε χάνει και πρέπει να ξεκινήσει πάλι από την αρχή. Υπάρχουν βέβαια πολλές παραλλαγές του παιχνιδιού αυτού που περιλαμβάνουν εμπόδια, power-ups (όπως επιμήκυνση του ξύλου, ενδυνάμωση της μπάλας κ.α.), καθώς και την παρουσία ενός «αφεντικού» (boss) ως τελευταία πίστα (δύσκολος και επιβλητικός εχθρός χειριζόμενος από τον υπολογιστή).

5.2 Η σελίδα σύνδεσης χρήστη (login.php)

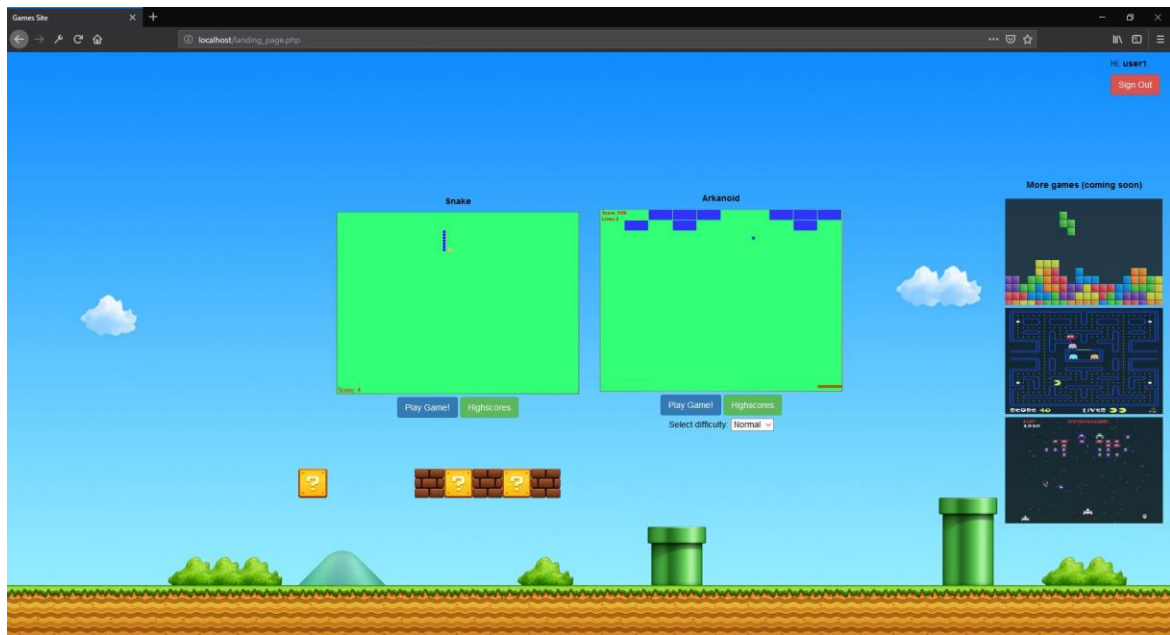


Εικόνα 5-1: Η σελίδα σύνδεσης (login page)

Η πρώτη σελίδα που βλέπει ο χρήστης είναι αυτή της σύνδεσης. Αποτελείται από μια φόρμα με τα δύο στοιχεία που χρειάζεται ο χρήστης για να συνδεθεί, Όνομα χρήστη και Κωδικό πρόσβασης. Και τα δύο αυτά στοιχεία είναι υποχρεωτικά και σε περίπτωση που δεν συμπληρωθούν εμφανίζεται ενημερωτικό μήνυμα με κόκκινα γράμματα.

Αν ο χρήστης έχει συμπληρώσει και τα δύο στοιχεία, τότε η σελίδα στέλνει αίτημα στον server για διασταύρωση των στοιχείων με τη βάση δεδομένων. Αν κάποιο από τα δύο στοιχεία δεν συμφωνεί με τα δεδομένα που αντλήθηκαν από τη βάση δεδομένων, τότε εμφανίζεται το αντίστοιχο μήνυμα ειδοποίησης. Αν τα στοιχεία ταυτοποίησης είναι έγκυρα, τότε ο χρήστης ανακατευθύνεται επιτυχώς στην κύρια σελίδα της εφαρμογής.

5.3 Η κύρια σελίδα (landing_page.php)



Εικόνα 5-2: Η κύρια σελίδα της εφαρμογής

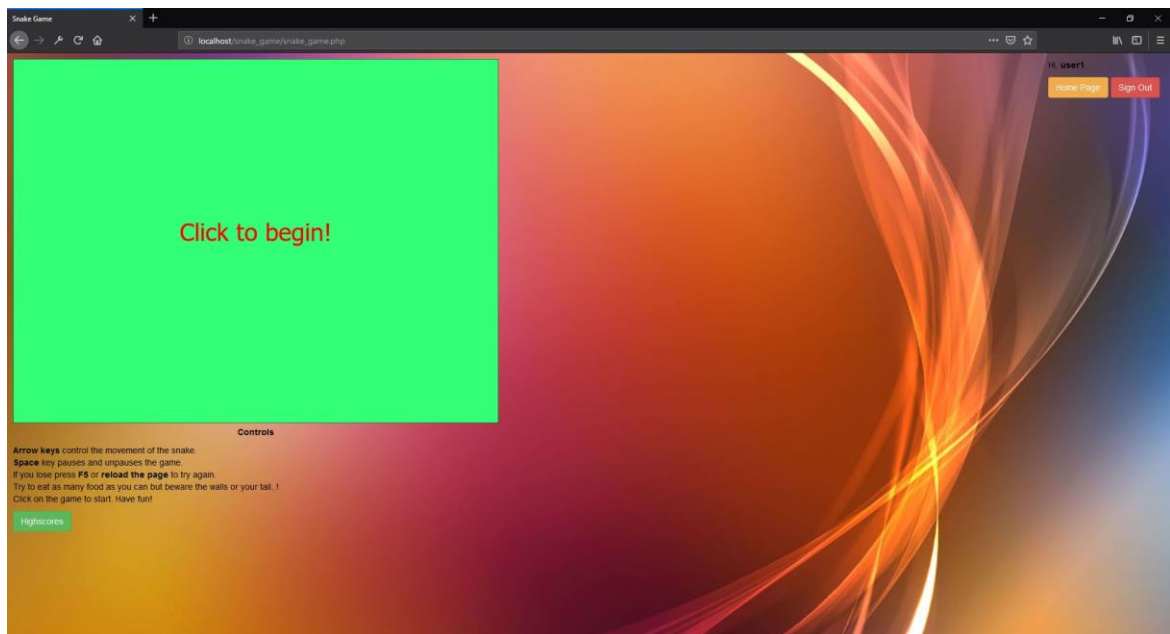
Η επόμενη σελίδα που εμφανίζεται στον χρήστη είναι η κύρια σελίδα της εφαρμογής. Η σελίδα αυτή αποτελεί τη μητρική σελίδα της εφαρμογής και περιλαμβάνει μια πληθώρα επιλογών:

- Στο πάνω δεξιό μέρος την οθόνης εμφανίζεται ένα μήνυμα υποδοχής του χρήστη μαζί με την επιλογή για αποσύνδεση από την εφαρμογή και επιστροφή στην οθόνη σύνδεσης. Αυτή η λειτουργία επαναλαμβάνεται σε όλες τις υπόλοιπες σελίδες για να μπορεί ο χρήστης να αποσυνδεθεί οποιαδήποτε στιγμή θελήσει.
- Στο κέντρο της οθόνης εμφανίζονται οι προεπισκοπήσεις των παιχνιδιών (στιγμιότυπα) που είναι διαθέσιμα, Snake (αριστερά) και Arkanoid (δεξιά).
- Ακριβώς από κάτω από κάθε παιχνίδι υπάρχουν δύο κουμπιά, το κουμπί Play και το κουμπί Highscores. Το Play μεταφέρει τον χρήστη στη σελίδα με το αντίστοιχο παιχνίδι, ενώ το κουμπί Highscores μεταφέρει τον χρήστη στη σελίδα με τον πίνακα των υψηλότερων βαθμών (score) για το εκάστοτε παιχνίδι.
- Στα δεξιά της οθόνης τοποθετήθηκαν κάποιες εικόνες από άλλα παιχνίδια που δεν συμπεριλαμβάνονται στην παρούσα πτυχιακή, κυρίως ως μια πρόταση για το πως θα μπορούσαμε να περιλαμβάνουμε στο μέλλον

περισσότερα παιχνίδια στην εφαρμογή. Σύροντας τον κέρσορα πάνω στις εικόνες τις σελίδας εμφανίζονται τα ονόματα των παιχνιδιών.

- Τέλος, ειδικά για το παιχνίδι Arkanoid έχει προστεθεί η λειτουργία της δυσκολίας με τρία επίπεδα προς επιλογή: Easy (Εύκολο), Medium (Κανονικό) και Hard (Δύσκολο). Η δυσκολία ουσιαστικά επηρεάζει το μέγεθος της μπάλας, κάνοντας δηλαδή ευκολότερο ή δυσκολότερο να την πετύχεις με το ξύλο.

5.4 Το παιχνίδι Snake (snake_game.php)



Εικόνα 5-3: Η σελίδα του παιχνιδιού Snake

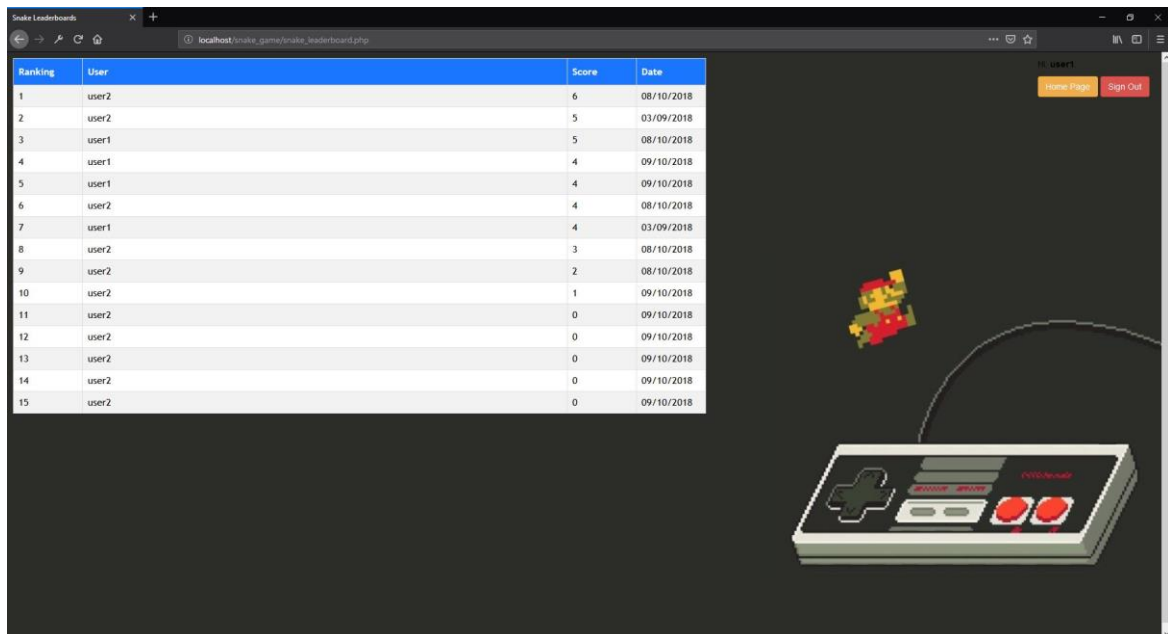
Μπορούμε να μετακινηθούμε σε αυτή τη σελίδα πατώντας το κουμπί Play κάτω από την εικόνα-προεπισκόπηση του παιχνιδιού Snake στην κύρια οθόνη της εφαρμογής.

- Στο πάνω δεξιά μέρος της σελίδας βρίσκουμε πάλι το μικρό μενού του χρήστη. Σε αυτό έχει προστεθεί η λειτουργία επιστροφής στην αρχική σελίδα της εφαρμογής, πατώντας το κουμπί Home Page. Το όνομα του χρήστη και η επιλογή για αποσύνδεση είναι πάντα διαθέσιμα.
- Στην αριστερή πλευρά της σελίδας βρίσκεται το πλαίσιο που αποτελεί τον χώρο του παιχνιδιού (το πλαίσιο με το πράσινο φόντο). Σε αυτήν την περιοχή λαμβάνει μέρος το παιχνίδι.

Κατασκευή Διαδικτυακού Ιστότοπου Παιχνιδιών

- Ακριβώς κάτω από το πλαίσιο αυτό υπάρχουν κάποιες βασικές οδηγίες. Αναφέρονται τα πλήκτρα χειρισμού του παιχνιδιού, πως να το εκκινήσετε, καθώς και το σκοπό του.
- Τέλος, το κουμπί Highscores οδηγεί στη σελίδα με τον πίνακα υψηλότερων βαθμών σε αυτό το παιχνίδι για σύγκριση με άλλους χρήστες.

5.5 Η σελίδα πόντων του παιχνιδιού Snake (snake_leaderboard.php)



| Ranking | User | Score | Date |
|---------|-------|-------|------------|
| 1 | user2 | 6 | 08/10/2018 |
| 2 | user2 | 5 | 03/09/2018 |
| 3 | user1 | 5 | 08/10/2018 |
| 4 | user1 | 4 | 09/10/2018 |
| 5 | user1 | 4 | 09/10/2018 |
| 6 | user2 | 4 | 08/10/2018 |
| 7 | user1 | 4 | 03/09/2018 |
| 8 | user2 | 3 | 08/10/2018 |
| 9 | user2 | 2 | 08/10/2018 |
| 10 | user2 | 1 | 09/10/2018 |
| 11 | user2 | 0 | 09/10/2018 |
| 12 | user2 | 0 | 09/10/2018 |
| 13 | user2 | 0 | 09/10/2018 |
| 14 | user2 | 0 | 09/10/2018 |
| 15 | user2 | 0 | 09/10/2018 |

Εικόνα 5-4: Η σελίδα με τα σκορ του παιχνιδιού Snake

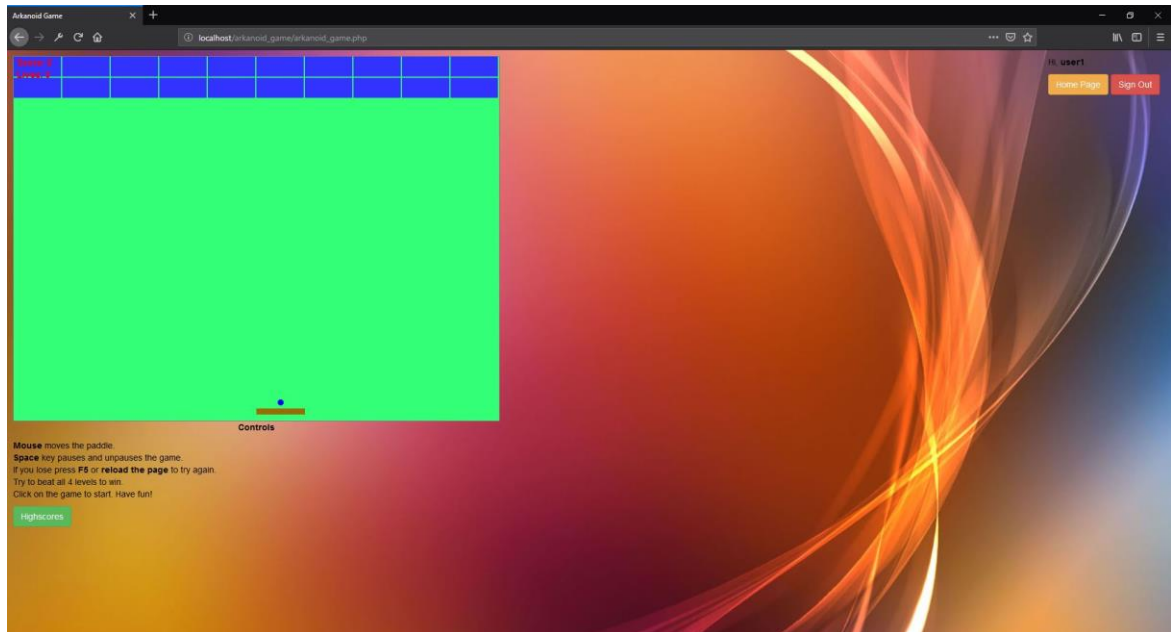
Σε αυτή τη σελίδα παρουσιάζονται οι καλύτερες προσπάθειες που έχουν επιτευχθεί στο παιχνίδι αυτό, για να μπορεί ο χρήστης να συγκρίνει την προσπάθειά του με τις προσπάθειες των υπολοίπων χρηστών. Μπορούμε να μετακινηθούμε σε αυτή είτε από την κύρια σελίδα είτε από τη σελίδα του παιχνιδιού, πατώντας το κουμπί Highscores.

- Στο πάνω δεξιά μέρος της σελίδας βρίσκουμε πάλι το μικρό μενού του χρήστη, όπως ακριβώς στην προηγούμενη σελίδα. Οι λειτουργίες παραμένουν οι ίδιες.
- Στο αριστερό μέρος της σελίδας βρίσκουμε τον πίνακα με τους πόντους των προσπαθειών των χρηστών. Απεικονίζονται οι 15 καλύτερες προσπάθειες (δηλαδή με τους περισσότερους πόντους) σε φθίνουσα σειρά. Ο πίνακας αυτός δεν είναι παρά μια απεικόνιση του πίνακα scores

Κατασκευή Διαδικτυακού Ιστότοπου Παιχνιδιών

της βάσης δεδομένων για το παιχνίδι Snake. Περιλαμβάνει όνομα χρήστη, τους πόντους, καθώς και την ημερομηνία που πραγματοποιήθηκε η προσπάθεια.

5.6 Το παιχνίδι Arkanoid (arkanoid_game.php)

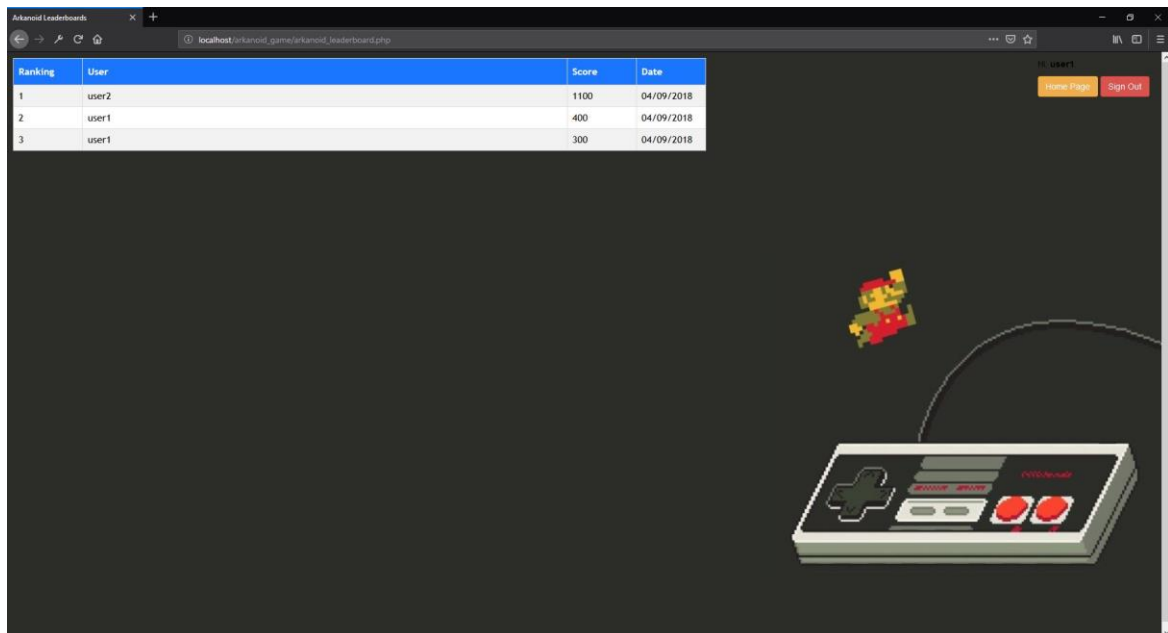


Εικόνα 5-5: Η σελίδα του παιχνιδιού Arkanoid

Μπορούμε να μετακινηθούμε σε αυτή τη σελίδα πατώντας το κουμπί Play κάτω από την εικόνα-προεπισκόπηση του παιχνιδιού Arkanoid στην κύρια οθόνη της εφαρμογής. Η σελίδα αυτή είναι παρόμοια της σελίδας του παιχνιδιού Snake και μοιράζεται τις ίδιες λειτουργίες.

- Το μενού χρήστη στο πάνω δεξιά μέρος της σελίδας.
- Την περιοχή του παιχνιδιού στο δεξί μέρος της σελίδας.
- Τα κουμπιά χειρισμού του παιχνιδιού, όπως και ο σκοπός του.
- Το κουμπί Highscores για προβολή της σελίδας με τα υψηλότερα σκορ.

5.7 Η σελίδα πόντων του παιχνιδιού Arkanoid (arkanoid_game.php)



Εικόνα 5-6: Η σελίδα με τα σκορ του παιχνιδιού Arkanoid

Μπορούμε να μετακινηθούμε σε αυτή είτε από την κύρια σελίδα είτε από τη σελίδα του παιχνιδιού, πατώντας το κουμπί Highscores. Η σελίδα αυτή είναι παρόμοια της σελίδας με τα σκορ του παιχνιδιού Snake και μοιράζεται τις ίδιες λειτουργίες και εμφάνιση.

- Το μενού χρήστη στο πάνω δεξιά μέρος της σελίδας.
- Τον πίνακα με τα υψηλότερα σκορ του παιχνιδιού Arkanoid στο αριστερό μέρος της σελίδας.

6.ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΕΦΑΡΜΟΓΗΣ

Στο κεφάλαιο αυτό παρουσιάζεται ο κώδικας που συντάχθηκε για την υλοποίηση της εφαρμογής. Συνοδεύεται από επεξηγηματικά σχόλια μέσα στον κώδικα, καθώς και μία μικρή περίληψη των βασικών στοιχείων του κάθε script που χρησιμοποιείται στην εφαρμογή.

6.1 config.php

```

1. <?php
2.     define('DB_SERVER', 'localhost');
3.     define('DB_USERNAME', 'secret');
4.     define('DB_PASSWORD', 'secret');
5.     define('DB_DATABASE', 'games_site');
6.     $db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
7.
8.     // Check connection
9.     if($db === false){
10.         die("ERROR: Could not connect. " . mysqli_connect_error());
11.     }
12. ?>

```

Το script αυτό αναλαμβάνει τη σύνδεση της εφαρμογής με τον server και καλείται από τις σελίδες που διακινούν δεδομένα από και προς αυτόν. Κύρια παραδείγματα είναι η σελίδα σύνδεσης χρήστη και οι δύο σελίδες με τους πίνακες των σκορ.

6.2 login.php

```

1. // Define variables and initialize with empty values
2. $username = $password = "";
3. $username_err = $password_err = $log_err = "";
4.
5. // Processing form data when form is submitted
6. if($_SERVER["REQUEST_METHOD"] == "POST"){
7.
8.     // Check if username is empty
9.     if(empty(trim($_POST["username"]))){
10.         $username_err = 'Please enter your username.';
11.     } else{
12.         $username = mysqli_real_escape_string($db,$_POST['username']);
13.     }
14.
15.     // Check if password is empty
16.     if(empty(trim($_POST['password']))){
17.         $password_err = 'Please enter your password.';

```

```

18.     } else{
19.         $password = mysqli_real_escape_string($db,$_POST['password']);
20.     }
21.
22.     // Validate credentials
23.     if(empty($username_err) && empty($password_err)){
24.         // Prepare a select statement
25.         $sql = "SELECT UserID, Username, Password FROM users WHERE Username = ?";
26.
27.         if($stmt = mysqli_prepare($db, $sql)){
28.             // Bind variables to the prepared statement as parameters
29.             mysqli_stmt_bind_param($stmt, "s", $param_username);
30.
31.             // Set parameters
32.             $param_username = $username;
33.             //$param_password = $password;
34.
35.             // Attempt to execute the prepared statement
36.             if(mysqli_stmt_execute($stmt)){
37.                 // Store result
38.                 mysqli_stmt_store_result($stmt);
39.
40.                 // Check if username exists, if yes then verify password
41.                 if(mysqli_stmt_num_rows($stmt) == 1){
42.                     // Bind result variables
43.                     mysqli_stmt_bind_result($stmt, $userid, $username, $userpass)
44.                 };
45.                 if(mysqli_stmt_fetch($stmt)){
46.                     if(password_verify($password, $userpass)){
47.                         session_start();
48.                         $_SESSION['username'] = $username;
49.                         $_SESSION['userid'] = $userid;
50.                         header("Location: landing_page.php");
51.                     } else{
52.                         $log_err = "The password you entered was not valid. Please
53.                         try again.";
54.                     }
55.                 } else{
56.                     $log_err = "The username you entered was not valid. Please tr
57.                     y again.";
58.                 }
59.             }
60.             // Close statement
61.             mysqli_stmt_close($stmt);
62.         }
63.
64.         // Close connection
65.         mysqli_close($db);
66.     }
67. }?>
68.
69.
70.
71. <!DOCTYPE html>
72. <html lang="en">
73. <head>
74.     <meta charset="UTF-8">
75.     <title>Welcome to our site!</title>
76.     <link rel="stylesheet" href="css/bootstrap.css">
77.     <style type="text/css">
78.         body{

```

```

79.         font: 14px sans-serif;
80.         height: 100%;
81.         color: #0080ff;
82.     }
83.     html{
84.         height: 100%
85.     }
86.     .bg {
87.         background-image: url("img/bg8.jpg");
88.         height: 100%;
89.         background-position: center;
90.         background-repeat: no-repeat;
91.         background-size: cover;
92.     }
93.     .wrapper{
94.         width: 350px;
95.         height: 200px;
96.         position: absolute;
97.         left: 50%;
98.         top: 40%;
99.         margin: -100px 0 0 -175px;
100.    }
101.    </style>
102. </head>
103. <body class="bg">
104.     <div class="wrapper">
105.         <h2>Welcome!</h2>
106.         <!-- <p>Please fill in your credentials to login.</p> -->
107.         <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);
108.         ?>" method="post">
109.             <div class="form-
110.             group <?php echo (!empty($username_err)) ? 'has-error' : ''; ?>">
111.                 <label>Username</label>
112.                 <input type="text" name="username" class="form-
113.                 control" placeholder="Required">
114.                 <span class="help-
115.                 block"><?php echo $username_err; ?></span>
116.             </div>
117.             <div class="form-
118.             group <?php echo (!empty($password_err)) ? 'has-error' : ''; ?>">
119.                 <label>Password</label>
120.                 <input type="password" name="password" class="form-
121.                 control" placeholder="Required">
122.                 <span class="help-
123.                 block"><?php echo $password_err; ?></span>
124.             </div>
125.             <div class="form-group">
126.                 <input type="submit" class="btn btn-
127.                 primary" value="Login">
128.                 <span class="help-
129.                 block" style="color: red"><?php echo $log_err; ?></span>
130.             </div>
131.         </form>
132.     </div>
133. </body>
134. </html>

```

Ο κώδικας αυτός εκτελεί πολλές λειτουργίες. Αναλαμβάνει την όλη διαδικασία της σύνδεσης καθώς και την εμφάνιση της ιστοσελίδας. Λαμβάνει τα δεδομένα που πληκτρολόγησε ο χρήστης και παραθέτει ερωτήματα προς τη βάση

δεδομένων για την εγκυρότητά τους. Αν επιτύχει η σύγκριση, τότε δημιουργεί το session του χρήστη. Παράλληλα, χτίζει την ιστοσελίδα και διαχειρίζεται τα μηνύματα ελέγχου που διακρίνουμε στις ετικέτες .

6.3 logout.php

```
1. <?php
2. // Initialize the session
3. session_start();
4.
5. // Unset all of the session variables
6. $_SESSION = array();
7.
8. // Destroy the session.
9. session_destroy();
10.
11. // Redirect to login page
12. header("location: login.php");
13. exit;
14. ?>
```

Το script αυτό ενεργοποιείται κατά την αποσύνδεση του χρήστη από την εφαρμογή και στην ουσία καταστρέφει το δημιουργημένο session και ανακατευθύνει τον browser πίσω στην οθόνη σύνδεσης.

6.4 landing_page.php

```
1. <?php
2. // Initialize the session
3. session_start();
4.
5. // If session variable is not set it will redirect to login page
6. if(!isset($_SESSION['userid']) || empty($_SESSION['userid'])){
7.     header("location: login.php");
8.     exit;
9. }
10. ?>
```

Αυτό το κομμάτι κώδικα επαναλαμβάνεται σε όλες τις ιστοσελίδες της εφαρμογής από εδώ και στο εξής. Είναι υπεύθυνο για τον έλεγχο δυνατότητας πρόσβασης στην κάθε ιστοσελίδα, αφού κοιτάει πρώτα αν υπάρχει δημιουργημένο session, δηλαδή αν ο χρήστης έχει περάσει επιτυχώς από το

login. Αν όχι, τότε δεν του επιτρέπεται να δει την ιστοσελίδα και ανακατευθύνεται στη σελίδα login.

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <meta charset="UTF-8">
5.   <title>Games Site</title>
6.   <link rel="stylesheet" href="/css/bootstrap.css">
7.   <style type="text/css">
8.     body{
9.       font: 14px sans-serif;
10.      color: black;
11.      height: 100%;
12.    }
13.    html{
14.      height: 100%
15.    }
16.    .bg {
17.      background-image: url("img/bg7.jpg");
18.      height: 100%;
19.      background-position: center;
20.      background-repeat: repeat-y;
21.      background-size: cover;
22.    }
23.    .logoutPos{
24.      position:fixed;
25.      right:20px;
26.      top:10px;
27.    }
28.    #container{
29.      position: absolute;
30.      top: 45%;
31.      left: 50%;
32.      transform: translate(-50%, -50%);
33.      text-align: center;
34.    }
35.    .gameCont{
36.      display: inline-block;
37.      width: 400px;
38.      height: 400px;
39.      margin: 15px;
40.    }
41.    img{
42.      max-width: 100%;
43.      max-height: 100%;
44.    }
45.    /* The grid: Four equal rows that float below one another */
46.    .column {
47.      float: right;
48.      width: 220px;
49.      padding: 10px;
50.      margin-top: 200px;
51.      margin-right: 20px;
52.    }
53.    /* Style the images inside the grid */
54.    .row img {
55.      opacity: 0.8;
56.      cursor: pointer;
57.      width: 260px;
58.      height: 175px;
59.      margin-top: 5px;
```

```

60.     }
61.     .column img:hover {
62.         opacity: 1;
63.     }
64. </style>
65. </head>
66. <body class="bg">
67.     <div class="logoutPos">
68.         <p>Hi, <b><?php echo htmlspecialchars($_SESSION['username']); ?></b>.</p>
69.         <p><a href="logout.php" class="btn btn-danger">Sign Out</a></p>
70.     </div>
71.
72.     <div class="column">
73.         <p style="text-align: center;"><b>More games (coming soon)</b></p>
74.         <div class="row">
75.             
76.         </div>
77.         <div class="row">
78.             
79.         </div>
80.         <div class="row">
81.             
82.         </div>
83.     </div>
84.
85.     <div id="container">
86.         <div class="gameCont">
87.             <p><b>Snake</b></p>
88.             
89.             <input type="button" onclick="window.location.href='snake_game\\snake
90. _game.php';"
91.                 class="btn btn-primary" value="Play Game!" style="margin-
92. top:5px">
93.             <input type="button" onclick="window.location.href='snake_game\\snake
94. _leaderboard.php';"
95.                 class="btn btn-success" value="Highscores" style="margin-
96. top:5px">
97.             <br>
98.             <input type="button" style="visibility:hidden;">
99.         </div>
100.        <div class="gameCont">
101.            <p><b>Arkanoid</b></p>
102.            
103.            <input type="button" onclick="window.location.href='arkanoid_game\\ar
104. kanoid_game.php';"
105.                class="btn btn-primary" value="Play Game!" style="margin-
106. top:5px">
107.            <input type="button" onclick="window.location.href='arkanoid_
108. game\\arkanoid_leaderboard.php';"
109.                class="btn btn-success" value="Highscores" style="margin-
110. top:5px">
111.            <br>
112.            Select difficulty:
113.            <select id="difficulty" style="margin-
114. top:5px" onchange="setDifficulty();"
115.                onfocus="this.selectedIndex = -
116. 1;" onload="this.selectedIndex = 2">
117.                <option value="Easy">Easy</option>
118.                <option value="Normal" selected>Normal</option>
119.                <option value="Hard">Hard</option>
120.            </select>
121.        </div>
122.    </div>
123.

```



```
114.     <script type="text/javascript">
115.         window.onload = function()
116.         {
117.             sessionStorage.setItem("difficulty", "Normal");
118.         }
119.
120.         function setDifficulty()
121.         {
122.             var sel = document.getElementById("difficulty");
123.             var dif = sel.options[sel.selectedIndex].value;
124.             sessionStorage.setItem("difficulty", dif);
125.         }
126.     </script>
127.
128. </body>
129. </html>
```

Αυτό είναι το υπόλοιπο script που δημιουργεί τη σελίδα. Μέσα στην ετικέτα `<style>` διακρίνουμε όλες τις εντολές CSS που έχουν χρησιμοποιηθεί για τα διάφορα μέρη της σελίδας, όπως τη μορφοποίηση των εικόνων για τα μελλοντικά παιχνίδια. Επίσης, μπορούμε να διακρίνουμε στις τελευταίες γραμμές τις εντολές που χρησιμοποιήθηκαν για τη λειτουργία του επιπέδου δυσκολίας για το παιχνίδι Arkanoid.

6.5 snake_game.php

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <meta charset="UTF-8">
5.         <title>Snake Game</title>
6.         <link rel="stylesheet" href="../css/bootstrap.css">
7.         <style type="text/css">
8.             .logoutPos{
9.                 position:fixed;
10.                 right:20px;
11.                 top:10px;
12.             }
13.             body{
14.                 color: black;
15.                 height: 100%;
16.             }
17.             html{
18.                 height: 100%
19.             }
20.             .bg {
21.                 background-image: url("../img/bg5.jpg");
22.                 background-position: center;
23.                 background-repeat: repeat-y;
24.                 background-size: cover;
25.             }
26.         </style>
27.     </head>
28.     <body class="bg">
```

```

29.     <script src="snake.js" type="text/javascript"></script>
30.
31.     <div class="logoutPos">
32.         <p>Hi, <b><?php echo htmlspecialchars($_SESSION['username']); ?></b>.
    </p>
33.         <p>
34.             <a href="..\landing_page.php" class="btn btn-
warning">Home Page</a>
35.             <a href="..\logout.php" class="btn btn-danger">Sign Out</a>
36.         </p>
37.     </div>
38.     <div id="container">
39.         <div id="divCanvas" style="display: inline-block; margin-
left: 10px; margin-top: 10px;">
40.             <canvas id="canvas" width="800" height="600"></canvas>
41.             <p style="text-align: center" > <b>Controls</b> </p>
42.             <p>
43.                 <b>Arrow keys</b> control the movement of the snake.<br>
44.                 <b>Space</b> key pauses and unpauses the game.<br>
45.                 If you lose press <b>F5</b> or <b>reload the page</b> to try
again.<br>
46.                 Try to eat as many food as you can but beware the walls or yo
ur tail...!<br>
47.                 Click on the game to start. Have fun!
48.             </p>
49.             <p><input type="button" onclick="window.location.href='snake_lead
erboard.php';"
50.                 class="btn btn-success" value="Highscores"></p>
51.         </div>
52.     </div>
53. </body>
54. </html>

```

Εδώ παρουσιάζεται ο κώδικας που δημιουργεί τη σελίδα με το παιχνίδι Snake. Μπορούμε να διακρίνουμε τις εντολές CSS που χρησιμοποιήθηκαν, την επικέτα `<canvas>` (γραμμή 40) που αποτελεί το χώρο του παιχνιδιού, καθώς και το κείμενο με τις οδηγίες ακριβώς από κάτω.

6.6 snake.js

```

1. window.onload = function(){
2.
3.     //Canvas
4.     canvas = document.getElementById('canvas');
5.     var ctx = canvas.getContext('2d');
6.     var w = canvas.getAttribute('width');
7.     var h = canvas.getAttribute('height');
8.
9.     const CW = 10; //The cell width constant
10.    const FPS = 60; //Frames per second for our Interval timer
11.    const SNAKE_LENGTH = 3; //Length of the snake
12.    var d;
13.    var nd;
14.    var food;
15.    var score;
16.    var game_over = false;

```

```

17.   var game_paused = false;
18.
19.   //Lets create the snake now
20.   var snake_array; //an array of cells to make up the snake
21.
22.   // start screen
23.   function start_screen()
24.   {
25.       paint_canvas();
26.       ctx.textAlign = "center";
27.       ctx.font = "40px Tahoma"
28.       ctx.fillStyle = "red"
29.       ctx.fillText("Click to begin!",w/2,h/2);
30.       ctx.textAlign = "left"; //reset text align
31.
32.       canvas.addEventListener('click', mouse_click)
33.   }
34.
35.   function init()
36.   {
37.       d = "right"; //default direction
38.       nd = [];
39.       create_snake();
40.       create_food();
41.       score = 0;
42.       game_over = false;
43.       game_paused = false;
44.
45.       //Lets move the snake now using a timer which will trigger
46.       //the render function according to our FPS constant
47.       if(typeof game_loop != "undefined") clearInterval(game_loop);
48.       game_loop = setInterval(render, FPS);
49.   }
50.
51.   function render()
52.   {
53.       ctx.clearRect(0, 0, canvas.width, canvas.height);
54.       update();
55.       paint();
56.   }
57.
58.   function create_snake()
59.   {
60.       snake_array = []; //Empty array to start with
61.       for(var i = SNAKE_LENGTH-1; i>=0; i--)
62.       {
63.           //This will create a horizontal snake starting from the top left
64.           snake_array.push({x: i, y:0});
65.       }
66.   }
67.
68.   function create_food()
69.   {
70.       var food_x = Math.round(Math.random()*(w-CW)/CW);
71.       var food_y = Math.round(Math.random()*(h-CW)/CW);
72.
73.       //Check if the created food coordinates are inside the snake
74.       //if they are, create another set of coordinates and check again
75.       while (check_collision(food_x, food_y, snake_array)){
76.           food_x = Math.round(Math.random()*(w-CW)/CW);
77.           food_y = Math.round(Math.random()*(h-CW)/CW);
78.       }
79.
80.       food = {
81.           x: food_x,

```

```

82.         y: food_y,
83.     });
84.     //This will create a cell with x/y between width and height
85. }
86.
87. function paint_canvas()
88. {
89.     ctx.fillStyle = "#33ff77";
90.     ctx.fillRect(0, 0, w, h);
91.     ctx.strokeStyle = "black";
92.     ctx.strokeRect(0, 0, w, h);
93. }
94.
95. //Secondary function of game: does all the painting
96. function paint()
97. {
98.     //To avoid the snake trail we need to paint the BG on every frame
99.     paint_canvas();
100.
101.     for(var i = 0; i < snake_array.length; i++)
102.     {
103.         var c = snake_array[i];
104.         paint_cell(c.x, c.y, "blue"); //paint snake
105.     }
106.
107.     //Paint the food
108.     paint_cell(food.x, food.y, "orange");
109.
110.     if (game_paused && !game_over) {
111.         ctx.fillStyle = 'rgb(255,0,0)';
112.         ctx.font = 'bold 20px Arial';
113.         ctx.fillText('Pause', canvas.width / 2 -
114. 30, canvas.height / 2);
115.     }
116.
117.     if (game_over) {
118.         ctx.fillStyle = 'rgb(255,0,0)';
119.         ctx.font = 'bold 20px Arial';
120.         ctx.fillText('Game Over', canvas.width / 2 -
121. 40, canvas.height / 2);
122.         clearInterval(game_loop);
123.         saveScores();
124.     }
125.
126.     //Paint the score
127.     var score_text = "Score: " + score;
128.     ctx.font = "20px Tahoma"
129.     ctx.fillStyle = "red"
130.     ctx.fillText(score_text, 5, h-5);
131. }
132.
133. //Primary function of game: controls all the logic
134. function update()
135. {
136.     if (game_over || game_paused) return;
137.
138.     if (nd.length) {
139.         d = nd.shift();
140.     }
141.
142.     //Logic for snake movement: Pop out the tail cell and place it in
143.     //front of the head cell
144.     var nx = snake_array[0].x;
145.     var ny = snake_array[0].y; //These are the position of the head c
146.     ell.

```

```

143.
144.           //We will increment them to get the new head position based on di
rection
145.           if(d == "right") nx++;
146.           else if(d == "left") nx--;
147.           else if(d == "up") ny--;
148.           else if(d == "down") ny++;
149.
150.
151.           //Game over clauses: snake hits wall or its own body
152.           if(nx == -1 || nx == w/CW || ny == -
1 || ny == h/CW || check_collision(nx, ny, snake_array))
153.           {
154.               game_over = true;
155.           }
156.
157.           //Code for snake eating food
158.           //Logic for eating food: If the new head position matches with th
at of the food,
159.           //create a new head instead of moving the tail
160.           if(nx == food.x && ny == food.y)
161.           {
162.               var tail = {x: nx, y: ny};
163.               score++;
164.               //Create new food
165.               create_food();
166.           }
167.           else
168.           {
169.               var tail = snake_array.pop(); //pops out the last cell
170.               tail.x = nx; tail.y = ny;
171.           }
172.
173.           snake_array.unshift(tail); //puts back the tail as the first cell
174.       }
175.
176.       //Generic function to paint cells
177.       function paint_cell(x, y, color)
178.       {
179.           ctx.fillStyle = color;
180.           ctx.fillRect(x*CW, y*CW, CW, CW);
181.           ctx.strokeStyle = "white";
182.           ctx.strokeRect(x*CW, y*CW, CW, CW);
183.       }
184.
185.       //Function for collision checking
186.       function check_collision(x, y, array)
187.       {
188.           //This function will check if the provided x/y coordinates exist
189.           //in an array of cells or not
190.           for(var i = 0; i < array.length; i++)
191.           {
192.               if(array[i].x == x && array[i].y == y)
193.                   return true;
194.           }
195.           return false;
196.       }
197.
198.       //Keyboard controls for moving the snake
199.       document.onkeydown = function(evt)
200.       {
201.           if (game_over)
202.               return;

```

```

203.         // code to prevent quick entry mixup of directions (update:28/07
still not working)
204.         if (nd.length) {
205.             var td = nd[nd.length - 1];
206.         } else {
207.             td = d;
208.         }
209.
210.         if(evt.keyCode == 37 && td != "right") d = "left";
211.         else if(evt.keyCode == 38 && td != "down") d = "up";
212.         else if(evt.keyCode == 39 && td != "left") d = "right";
213.         else if(evt.keyCode == 40 && td != "up") d = "down";
214.         else if(evt.keyCode == 32) game_paused = !game_paused;
215.     }
216.
217.     //Function to start game if left mouse button is clicked
218.     function mouse_click(evt)
219.     {
220.         evt = evt || window.event;
221.         if (evt.which == 1)
222.             canvas.removeEventListener('click', mouse_click);
223.             init();
224.     }
225.
226.     function saveScores()
227.     {
228.         if (window.XMLHttpRequest) {
229.             // code for modern browsers
230.             xmlhttp = new XMLHttpRequest();
231.         } else {
232.             // code for old IE browsers
233.             xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
234.         }
235.
236.         xmlhttp.open("GET", "../savescores.php?varScore=" + score + "&varGa
me=snake", true);
237.         xmlhttp.send();
238.     }
239. }

```

Εδώ βλέπουμε το αρχείο JavaScript που διαχειρίζεται τη λειτουργία του παιχνιδιού Snake. Είναι εκτενές και περίπλοκο αλλά υπάρχουν αρκετά σχόλια με διευκρινήσεις. Κυριότερο σημείο είναι η function `init()` (γραμμή 35) η οποία αρχικοποιεί τα αντικείμενά μας και καλεί τη βασική συνάρτηση που έχουμε αναφέρει, `setInterval()`. Αυτή καλεί τη συνάρτηση `render()` συνεχόμενα για να «ζωγραφιστούν» τα γραφικά πάνω στην οθόνη. Σημαντική προς αναφορά είναι και η function `saveScores()` στο τέλος, η οποία είναι υπεύθυνη για τη μεταφορά του σκορ στο script `savescores.php` (θα αναλυθεί παρακάτω) προς αποθήκευση στη βάση δεδομένων.

6.7 snake_leaderboard.php

```

1. <?php
2.     // Initialize the session
3.     session_start();
4.
5.     // If session variable is not set it will redirect to login page
6.     if(!isset($_SESSION['userid']) || empty($_SESSION['userid'])){
7.         header("location: ../login.php");
8.         exit;
9.     }
10.
11.    // Include config file
12.    require_once '../config.php';
13. ?>
14.
15. <html>
16. <head>
17.     <meta charset="UTF-8">
18.     <title>Snake Leaderboards</title>
19.     <link rel="stylesheet" href="../css/bootstrap.css">
20.     <style>
21.         #scores {
22.             font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
23.             border-collapse: collapse;
24.             width: 60%;
25.             margin-top: 10px;
26.             margin-left: 10px;
27.         }
28.
29.         #scores td, #scores th {
30.             border: 1px solid #ddd;
31.             padding: 8px;
32.         }
33.
34.         #scores tr:nth-child(even){background-color: #f2f2f2;}
35.         #scores tr:nth-child(odd){background-color: white;}
36.
37.         #scores tr:hover {background-color: #ddd;}
38.
39.         #scores th {
40.             padding-top: 12px;
41.             padding-bottom: 12px;
42.             text-align: left;
43.             background-color: #1A75FF;
44.             color: white;
45.         }
46.
47.         .logoutPos{
48.             position:fixed;
49.             right:20px;
50.             top:10px;
51.         }
52.         body{
53.             color: black;
54.             height: 100%;
55.         }
56.         html{
57.             height: 100%
58.         }
59.         .bg {
60.             background-image: url("../img/bg9.jpg");
61.             height: 100%;
62.             background-position: center;

```

```

63.         background-repeat: repeat-y;
64.         background-size: cover;
65.     }
66. </style>
67. </head>
68. <body class="bg">
69.
70.     <div class="logoutPos">
71.         <p>Hi, <b><?php echo htmlspecialchars($_SESSION['username']); ?></b>.
</p>
72.         <p>
73.             <a href="..\landing_page.php" class="btn btn-
warning">Home Page</a>
74.             <a href="..\logout.php" class="btn btn-danger">Sign Out</a>
75.         </p>
76.     </div>
77.     <table id="scores">
78.         <tr>
79.             <th style="width: 10%;">Ranking</th>
80.             <th style="width: 70%;">User</th>
81.             <th style="width: 10%;">Score</th>
82.             <th style="width: 10%;">Date</th>
83.         </tr>
84.
85.         <?php
86.             if (mysqli_num_rows($result) > 0)
87.             {
88.                 while($row = mysqli_fetch_assoc($result))
89.                 {
90.                     $var = $row["UserID"];
91.                     $query1 = "SELECT Username FROM users WHERE UserID = '$va
r'";
92.                     $result1 = mysqli_query($db, $query1);
93.                     $username = mysqli_fetch_object($result1);
94.                     $aa++;
95.                     $datetime = strtotime($row["DayAchieved"]);
96.                     $formattedDate = date("d/m/Y", $datetime);
97.
98.                     echo "<tr>";
99.                     echo "<td>".$aa."</td>";
100.                    echo "<td>".$username->Username."</td>";
101.                    echo "<td>".$row["Score"]."</td>";
102.                    echo "<td>".$formattedDate."</td>";
103.                    echo "</tr>";
104.                }
105.            }
106.            mysqli_close($db);
107.        ?>
108.    </table>
109. </body>
110. </html>

```

Εδώ βλέπουμε τον κώδικα που δημιουργεί τη σελίδα με τον πίνακα των σκορ του παιχνιδιού Snake. Για το γέμισμα του πίνακα με δεδομένα έχει χρησιμοποιηθεί επαναληπτικός βρόχος while με εντολές γλώσσας PHP και queries προς τη βάση δεδομένων.

6.8 arkanoid_game.php

```

1. <?php
2. // Initialize the session
3. session_start();
4.
5. // If session variable is not set it will redirect to login page
6. if(!isset($_SESSION['userid']) || empty($_SESSION['userid'])){
7.     header("location: ../login.php");
8.     exit;
9. }
10. ?>
11.
12.
13. <html>
14.     <head>
15.         <meta charset="UTF-8">
16.         <title>Arkanoid Game</title>
17.         <link rel="stylesheet" href="../css/bootstrap.css">
18.         <style type="text/css">
19.             .logoutPos{
20.                 position:fixed;
21.                 right:20px;
22.                 top:10px;
23.             }
24.             body{
25.                 color: black;
26.                 height: 100%;
27.             }
28.             html{
29.                 height: 100%
30.             }
31.             .bg {
32.                 background-image: url("../img/bg5.jpg");
33.                 height: 100%;
34.                 background-position: center;
35.                 background-repeat: repeat-y;
36.                 background-size: cover;
37.             }
38.         </style>
39.     </head>
40.     <body onload="checkCanvasIsSupported();" class="bg">
41.
42.         <div class="logoutPos">
43.             <p>Hi, <b><?php echo htmlspecialchars($_SESSION['username']); ?></b>.
44.         </p>
45.         <p>
46.             <a href="..\landing_page.php" class="btn btn-
47. warning">Home Page</a>
48.             <a href="..\logout.php" class="btn btn-danger">Sign Out</a>
49.         </p>
50.     </div>
51.     <div id="container">
52.         <div id="divCanvas" style="display: inline-block; margin-
53. left: 10px; margin-top: 10px;">
54.             <canvas id="gameCanvas" style="border: solid 1px gray;"></canvas>
55.         <br/>
56.         <p style="text-align: center" > <b>Controls</b> </p>
57.         <p>
58.             <b>Mouse</b> moves the paddle.<br>
59.             <b>Space</b> key pauses and unpauses the game.<br>
60.             If you lose press <b>F5</b> or <b>reload the page</b> to try
61. again.<br>
62.             Try to beat all 4 levels to win.<br>

```

```
58.         Click on the game to start. Have fun!
59.     </p>
60.     <p><input type="button" onclick="window.location.href='arkanoid_1
        eaderboard.php';" class="btn btn-success" value="Highscores"></p>
61.     </div>
62. </div>
63. </body>
64. </html>
```

Εδώ παρουσιάζεται ο κώδικας που δημιουργεί τη σελίδα με το παιχνίδι Arkanoid. Μπορούμε να διακρίνουμε τις εντολές CSS που χρησιμοποιήθηκαν, την ετικέτα `<canvas>` (γραμμή 51) που αποτελεί το χώρο του παιχνιδιού, καθώς και το κείμενο με τις οδηγίες ακριβώς από κάτω.

6.9 arkanoid.js

```
1. //Make method requestAnimationFrame run on most browsers and if not fallback to u
   sing setTimeout method
2.     window.requestAnimationFrame = window.requestAnimationFrame
3.                                     || window.mozRequestAnimationFrame
4.                                     || window.webkitRequestAnimationFrame
5.                                     || window.msRequestAnimationFrame
6.                                     || function(f){return setTimeout(f, 1000/60)}
7.
8.     window.cancelAnimationFrame = window.cancelAnimationFrame
9.                                     || window.mozCancelAnimationFrame
10.                                    || function(requestID){clearTimeout(requestID)}
11.
12. function Paddle(x, y, width, height) {
13.     this.x = x;
14.     this.y = y;
15.     this.width = width;
16.     this.height = height;
17. };
18.
19. var BallDirs = {
20.     NONE : 0,
21.     LEFT : 1,
22.     RIGHT : 2,
23.     UP : 4,
24.     DOWN : 8
25. };
26.
27. function Ball(x, y, radius, dir, speed) {
28.     this.x = x;
29.     this.y = y;
30.     this.radius = radius;
31.     this.dir = BallDirs.NONE;
32.     this.speed = speed;
33. }
34.
35. var BricksTypes = {
36.     DEFAULT : 1,
37.     ICE : 1,
```

```

38.     WOOD : 2,
39.     STONE : 3,
40.     IRON : 4,
41.     STEEL : 5
42. };
43.
44. function Brick(x, y, width, height, type) {
45.     this.x = x;
46.     this.y = y;
47.     this.width = width;
48.     this.height = height;
49.     this.lives = type;
50. }
51.
52. function Bricks(hor_num, vert_num, brick_width, brick_height) {
53.     this.bricks = new Array();
54.     for (var i = 0; i < vert_num; i++) {
55.         this.bricks[i] = new Array();
56.         for (var j = 0; j < hor_num; j++) {
57.             this.bricks[i][j] = new Brick(j * brick_width, i * brick_height, brick_width, brick_height, BricksTypes.DEFAULT);
58.         }
59.     }
60. }
61.
62. var Game_Over = false;
63.
64. //-----
65. // Arkanoid Game class
66. //-----
67. function ArkanoidGame(canvas, context) {
68.
69.     var PADDLE_SPEED = 1;
70.     var BALL_DEFAULT_SPEED = 3;
71.     var BALL_MAX_SPEED = 6;
72.     var BRICK_SCORE = 100;
73.
74.     var dif = sessionStorage.getItem("difficulty");
75.     if (dif == "Easy") {
76.         BALL_RADIUS = 8;
77.     } else if (dif == "Normal") {
78.         BALL_RADIUS = 5;
79.     } else if (dif == "Hard") {
80.         BALL_RADIUS = 3;
81.     }
82.
83.     this.level = 0;
84.     this.lives = 3;
85.     this.score = 0;
86.     this.paddle = new Paddle(canvas.width / 2 - PADDLE_WIDTH / 2, canvas.height - 20, PADDLE_WIDTH, PADDLE_HEIGHT);
87.     this.ball = new Ball(canvas.width / 2, canvas.height / 2, BALL_RADIUS, BallDirections.NONE, BALL_DEFAULT_SPEED);
88.     this.gameOver = false;
89.     this.gameWin = false;
90.     this.gamePaused = false;
91.     this.bricks = new Bricks(10, 2, BRICK_WIDTH, BRICK_HEIGHT);
92.
93.     this.init = function() {
94.         this.level = 0;
95.         this.lives = 3;
96.         this.score = 0;
97.         this.gameOver = false;

```

```

98.     this.gameWin = false;
99.     this.gamePaused = false;
100.    this.ball.dir = BallDirs.NONE;
101.    this.initLevel(this.level);
102.    }
103.
104.    this.initLevel = function(level) {
105.        switch (level) {
106.            case 0:
107.                this.bricks = new Bricks(10, 2, BRICK_WIDTH, BRICK_HEIGHT
108.                );
109.                for (var i = 0; i < this.bricks.bricks.length; i++) {
110.                    for (var j = 0; j < this.bricks.bricks[i].length; j++)
111.                        this.bricks.bricks[i][j].lives = BricksTypes.DEFA
112.                        ULT;
113.                    }
114.                }
115.                break;
116.            case 1:
117.                this.bricks = new Bricks(10, 2, BRICK_WIDTH, BRICK_HEIGHT
118.                );
119.                for (var i = 0; i < this.bricks.bricks.length; i++) {
120.                    for (var j = 0; j < this.bricks.bricks[i].length; j++)
121.                        this.bricks.bricks[i][j].lives = BricksTypes.DEFA
122.                        ULT + i;
123.                    }
124.                }
125.                break;
126.            case 2:
127.                this.bricks = new Bricks(10, 4, BRICK_WIDTH, BRICK_HEIGHT
128.                );
129.                for (var i = 0; i < this.bricks.bricks.length; i++) {
130.                    for (var j = 0; j < this.bricks.bricks[i].length; j++)
131.                        this.bricks.bricks[i][j].lives = BricksTypes.DEFA
132.                        ULT + i;
133.                    }
134.                }
135.                break;
136.            case 3:
137.                this.bricks = new Bricks(10, 5, BRICK_WIDTH, BRICK_HEIGHT
138.                );
139.                for (var i = 0; i < this.bricks.bricks.length; i++) {
140.                    for (var j = 0; j < this.bricks.bricks[i].length; j++)
141.                        this.bricks.bricks[i][j].lives = BricksTypes.DEFA
142.                        ULT + i;
143.                    }
144.                }
145.                break;
146.            default:
147.                break;
148.        }
149.    }
150.
151.    this.drawBall = function() {
152.        context.beginPath();
153.        context.arc(this.ball.x, this.ball.y, this.ball.radius, 0, 2 * Ma
154.        th.PI, false);

```

```

150.         context.fillStyle = 'blue';
151.         context.fill();
152.     }
153.
154.     this.drawBricks = function() {
155.         for (var i = 0; i < this.bricks.bricks.length; i++) {
156.             for (var j = 0; j < this.bricks.bricks[i].length; j++) {
157.                 if (this.bricks.bricks[i][j].lives > 0) {
158.                     switch (this.bricks.bricks[i][j].lives) {
159.                         case BricksTypes.ICE: context.fillStyle = 'rgb(51
160. , 51, 255)'; break;
161.                         case BricksTypes.WOOD: context.fillStyle = 'rgb(1
162. 15, 77, 38)'; break;
163.                         case BricksTypes.STONE: context.fillStyle = 'rgb(
164. 217, 217, 217)'; break;
165.                         case BricksTypes.IRON: context.fillStyle = 'rgb(7
166. 7, 77, 77)'; break;
167.                         case BricksTypes.STEEL: context.fillStyle = 'rgb(
168. 179, 179, 204)'; break;
169.                         case BricksTypes.DEFAULT: context.fillStyle = 'rg
170. ba(220,220,255,0.9)'; break;
171.                         default: context.fillStyle = 'rgb(255,0,0)';
172.                     }
173.                     context.fillRect(this.bricks.bricks[i][j].x, this.bri
174. cks.bricks[i][j].y, this.bricks.bricks[i][j].width -
175. 2, this.bricks.bricks[i][j].height - 2);
176.                 }
177.             }
178.         }
179.     }
180.
181.     this.draw = function() {
182.         context.fillStyle = 'rgb(51, 255, 119)';
183.         context.fillRect(0, 0, canvas.width, canvas.height);
184.
185.         this.drawBall();
186.
187.         // draw paddle
188.         context.fillStyle = 'rgb(155,110,5)';
189.         context.fillRect(this.paddle.x, this.paddle.y, this.paddle.width,
190. this.paddle.height);
191.
192.         this.drawBricks();
193.
194.         if (this.gamePaused && !this.gameWin && !this.gameOver) {
195.             context.fillStyle = 'rgb(255,0,0)';
196.             context.font = 'bold 20px Arial';
197.             context.fillText('Pause', canvas.width / 2 -
198. 30, canvas.height / 2);
199.         }
200.
201.         if (this.gameOver) {
202.             context.fillStyle = 'rgb(255,0,0)';
203.             context.font = 'bold 20px Arial';
204.             context.fillText('Game Over', canvas.width / 2 -
205. 40, canvas.height / 2);
206.             Game_Over = true;
207.             saveScores(this.score);
208.         }
209.
210.         if (this.gameWin) {
211.             context.fillStyle = 'rgb(255,0,0)';
212.             context.font = 'bold 20px Arial';
213.             context.fillText('You Win', canvas.width / 2 -
214. 40, canvas.height / 2);

```

```

203.         }
204.
205.         context.fillStyle = 'rgb(255,0,0)';
206.         context.font = 'bold 15px Arial';
207.         context.fillText('Score: ' + this.score, 5, 15);
208.
209.         context.fillStyle = 'rgb(255,0,0)';
210.         context.font = 'bold 15px Arial';
211.         context.fillText('Lives: ' + this.lives, 5, 35);
212.     }
213.
214.     this.update = function() {
215.         if (this.gamePaused || this.gameWin || this.gameOver) return;
216.
217.         // update ball pos
218.         if (this.ball.dir & BallDirs.RIGHT) this.ball.x += this.ball.speed;
219.         else if (this.ball.dir & BallDirs.LEFT) this.ball.x -
= this.ball.speed;
220.         if (this.ball.dir & BallDirs.UP) this.ball.y -
= this.ball.speed;
221.         else if (this.ball.dir & BallDirs.DOWN) this.ball.y += this.ball.
speed;
222.
223.         // ball bounce from paddle
224.         if ((this.ball.x + this.ball.radius > this.paddle.x && this.ball.
x - this.ball.radius < this.paddle.x + this.paddle.width) &&
225.             (this.ball.y + this.ball.radius > this.paddle.y)) {
226.             if (this.ball.speed < BALL_MAX_SPEED) this.ball.speed += 0.5;
227.
228.             if (this.ball.dir & BallDirs.DOWN) {
229.                 this.ball.dir = this.ball.dir -
BallDirs.DOWN + BallDirs.UP;
230.             } else if (this.ball.dir & BallDirs.UP) {
231.                 this.ball.dir = this.ball.dir -
BallDirs.UP + BallDirs.DOWN;
232.             }
233.
234.             // update ball
235.             if (this.ball.x - this.ball.radius < 0) {
236.                 this.ball.x = this.ball.radius;
237.                 this.ball.dir = this.ball.dir -
BallDirs.LEFT + BallDirs.RIGHT;
238.             }
239.             if (this.ball.x + this.ball.radius > canvas.width) {
240.                 this.ball.x = canvas.width - this.ball.radius;
241.                 this.ball.dir = this.ball.dir -
BallDirs.RIGHT + BallDirs.LEFT;
242.             }
243.             if (this.ball.y - this.ball.radius < 0) {
244.                 this.ball.y = this.ball.radius;
245.                 this.ball.dir = this.ball.dir -
BallDirs.UP + BallDirs.DOWN;
246.             }
247.
248.             if (this.ball.y + this.ball.radius > canvas.height) {
249.                 // lost one life
250.                 this.lives--;
251.                 this.ball.speed = BALL_DEFAULT_SPEED;
252.                 if (this.lives == 0) {
253.                     this.gameOver = true;
254.                 } else {
255.                     this.ball.x = canvas.width / 2;
256.                     this.ball.y = canvas.height / 2;

```

```

257.         this.ball.dir = BallDirs.NONE;
258.     }
259. }
260.
261.     if (this.ball.dir == BallDirs.NONE) {
262.         this.ball.x = this.paddle.x + this.paddle.width / 2;
263.         this.ball.y = this.paddle.y - this.ball.radius * 2;
264.     }
265.
266.     // bounces
267.     for (var i = 0; i < this.bricks.bricks.length; i++) {
268.         for (var j = 0; j < this.bricks.bricks[i].length; j++) {
269.             if (this.bricks.bricks[i][j].lives > 0) {
270.                 if (this.ball.dir == BallDirs.LEFT + BallDirs.UP) {
271.                     if (this.isPointInRect(this.ball.x -
272. this.ball.speed, this.ball.y -
273. 0, this.bricks.bricks[i][j].x, this.bricks.bricks[i][j].y, this.bricks.bricks[i]
274. [j].width, this.bricks.bricks[i][j].height)) {
275.                         this.ball.x = this.bricks.bricks[i][j].x + th
276. is.bricks.bricks[i][j].width + this.ball.speed;
277.                         this.ball.dir = this.ball.dir -
278. BallDirs.LEFT + BallDirs.RIGHT;
279.                         this.bricks.bricks[i][j].lives--;
280.                         this.score += BRICK_SCORE;
281.                         return;
282.                     }
283.                     if (this.isPointInRect(this.ball.x -
284. 0, this.ball.y -
285. this.ball.speed, this.bricks.bricks[i][j].x, this.bricks.bricks[i][j].y, this.br
286. icks.bricks[i][j].width, this.bricks.bricks[i][j].height)) {
287.                         this.ball.y = this.bricks.bricks[i][j].y + th
288. is.bricks.bricks[i][j].height + this.ball.speed;
289.                         this.ball.dir = this.ball.dir -
290. BallDirs.UP + BallDirs.DOWN;
291.                         this.bricks.bricks[i][j].lives--;
292.                         this.score += BRICK_SCORE;
293.                         return;
294.                     }
295.                 } else if (this.ball.dir == BallDirs.LEFT + BallDirs.
296. DOWN) {
297.                     if (this.isPointInRect(this.ball.x -
298. this.ball.speed, this.ball.y + 0, this.bricks.bricks[i][j].x, this.bricks.bricks
299. [i][j].y, this.bricks.bricks[i][j].width, this.bricks.bricks[i][j].height)) {
300.                         this.ball.x = this.bricks.bricks[i][j].x + th
301. is.bricks.bricks[i][j].width + this.ball.speed;
302.                         this.ball.dir = this.ball.dir -
303. BallDirs.LEFT + BallDirs.RIGHT;
304.                         this.bricks.bricks[i][j].lives--;
305.                         this.score += BRICK_SCORE;
306.                         return;
307.                     }
308.                     if (this.isPointInRect(this.ball.x -
309. 0, this.ball.y + this.ball.speed, this.bricks.bricks[i][j].x, this.bricks.bricks
310. [i][j].y, this.bricks.bricks[i][j].width, this.bricks.bricks[i][j].height)) {
311.                         this.ball.y = this.bricks.bricks[i][j].y -
312. this.ball.speed;
313.                         this.ball.dir = this.ball.dir -
314. BallDirs.DOWN + BallDirs.UP;
315.                         this.bricks.bricks[i][j].lives--;
316.                         this.score += BRICK_SCORE;
317.                         return;
318.                     }
319.                 } else if (this.ball.dir == BallDirs.RIGHT + BallDirs
320. .UP) {
321.                     if (this.isPointInRect(this.ball.x + this.ball.sp

```

```

    eed, this.ball.y -
    0, this.bricks.bricks[i][j].x, this.bricks.bricks[i][j].y, this.bricks.bricks[i]
    [j].width, this.bricks.bricks[i][j].height)) {
302.         this.ball.x = this.bricks.bricks[i][j].x -
    this.ball.speed;
303.         this.ball.dir = this.ball.dir -
    BallDirs.RIGHT + BallDirs.LEFT;
304.         this.bricks.bricks[i][j].lives--;
305.         this.score += BRICK_SCORE;
306.         return;
307.     }
308.     if (this.isPointInRect(this.ball.x + 0, this.ball
    .y -
    this.ball.speed, this.bricks.bricks[i][j].x, this.bricks.bricks[i][j].y, this.br
    icks.bricks[i][j].width, this.bricks.bricks[i][j].height)) {
309.         this.ball.y = this.bricks.bricks[i][j].y + th
    is.bricks.bricks[i][j].height + this.ball.speed;
310.         this.ball.dir = this.ball.dir -
    BallDirs.UP + BallDirs.DOWN;
311.         this.bricks.bricks[i][j].lives--;
312.         this.score += BRICK_SCORE;
313.         return;
314.     }
315.     } else if (this.ball.dir == BallDirs.RIGHT + BallDirs
    .DOWN) {
316.         if (this.isPointInRect(this.ball.x + this.ball.sp
    eed, this.ball.y + 0, this.bricks.bricks[i][j].x, this.bricks.bricks[i][j].y, thi
    s.bricks.bricks[i][j].width, this.bricks.bricks[i][j].height)) {
317.             this.ball.x = this.bricks.bricks[i][j].x -
    this.ball.speed;
318.             this.ball.dir = this.ball.dir -
    BallDirs.RIGHT + BallDirs.LEFT;
319.             this.bricks.bricks[i][j].lives--;
320.             this.score += BRICK_SCORE;
321.             return;
322.         }
323.         if (this.isPointInRect(this.ball.x + 0, this.ball
    .y + this.ball.speed, this.bricks.bricks[i][j].x, this.bricks.bricks[i][j].y, thi
    s.bricks.bricks[i][j].width, this.bricks.bricks[i][j].height)) {
324.             this.ball.y = this.bricks.bricks[i][j].y -
    this.ball.speed;
325.             this.ball.dir = this.ball.dir -
    BallDirs.DOWN + BallDirs.UP;
326.             this.bricks.bricks[i][j].lives--;
327.             this.score += BRICK_SCORE;
328.             return;
329.         }
330.     }
331. }
332. }
333. }
334.
335.     full_level_life = 0;
336.     for (var i = 0; i < this.bricks.bricks.length; i++) {
337.         for (var j = 0; j < this.bricks.bricks[i].length; j++) {
338.             full_level_life += this.bricks.bricks[i][j].lives
339.         }
340.     }
341.
342.     if (full_level_life == 0) {
343.         if (this.level < 3) {
344.             this.ball.dir = BallDirs.NONE;
345.             this.level++;
346.             this.initLevel(this.level);
347.         } else this.gameWin = true;

```



```

348.     }
349.     }
350.
351.     this.isPointInRect = function(x, y, rect_x, rect_y, rect_width, rect_
height) {
352.         if ((x > rect_x && x < rect_x + rect_width) &&
353.             (y > rect_y && y < rect_y + rect_height))
354.             return true;
355.         return false;
356.     }
357.
358.     this.render = function() {
359.         context.clearRect(0, 0, canvas.width, canvas.height);
360.         this.update();
361.         this.draw();
362.     }
363.
364.     this.togglePause = function() {
365.         this.gamePaused = !this.gamePaused;
366.     }
367.
368.     this.setPaddlePos = function(x) {
369.         if (this.gamePaused || this.gameWin || this.gameOver) return;
370.         if (x < 0) x = 0;
371.         if (x > canvas.width - this.paddle.width) x = canvas.width -
this.paddle.width;
372.         this.paddle.x = x;
373.     }
374.
375.     this.startGame = function() {
376.         if (this.gamePaused) return;
377.         if (this.ball.dir == BallDirs.NONE) {
378.             this.ball.dir = BallDirs.RIGHT + BallDirs.UP;
379.         }
380.     }
381. };
382.
383.
384. var arkanoidGame;
385.
386. function render() {
387.     arkanoidGame.render();
388.
389.     //Use requestAnimationFrame to call render function again in a loop i
f it's not game over
390.     id = requestAnimationFrame(render);
391.     if (Game_Over) {
392.         cancelAnimationFrame(id);
393.     }
394. }
395.
396. function checkCanvasIsSupported() {
397.     canvas = document.getElementById("gameCanvas");
398.     canvas.width = 800;
399.     canvas.height = 600;
400.     canvas.style.cursor = "none";
401.     canvas.onclick = function() { arkanoidGame.startGame(); }
402.     if (canvas.getContext) {
403.         context = canvas.getContext('2d');
404.
405.         arkanoidGame = new ArkanoidGame(canvas, context);
406.         arkanoidGame.init();
407.
408.         render();
409.     } else {

```

```

410.         alert("Sorry, but your browser doesn't support a canvas.");
411.     }
412. }
413.
414.     var KeyCodes = {
415.         SPACE : 32
416.     };
417.
418.     document.onkeydown = function(event) {
419.         var keyCode;
420.         if (event == null) {
421.             keyCode = window.event.keyCode;
422.         } else {
423.             keyCode = event.keyCode;
424.         }
425.         switch (keyCode) {
426.             case KeyCodes.SPACE:
427.                 arkanoidGame.togglePause();
428.                 break;
429.             default:
430.                 break;
431.         }
432.     }
433.
434.     document.onmousemove = function(event) {
435.         arkanoidGame.setPaddlePos(event.pageX);
436.     }
437.
438.     function saveScores(score) {
439.         if (window.XMLHttpRequest) {
440.             // code for modern browsers
441.             xhttp = new XMLHttpRequest();
442.         } else {
443.             // code for old IE browsers
444.             xhttp = new ActiveXObject("Microsoft.XMLHTTP");
445.         }
446.
447.         //alert(score);
448.
449.         xhttp.open("GET", "../savescores.php?varScore=" + score + "&varGame=a
rkanoid", true);
450.         xhttp.send();
451.     }

```

Παραπάνω βλέπουμε τον κώδικα που απαρτίζει το παιχνίδι Arkanoid. Για το παιχνίδι αυτό χρησιμοποιήθηκε μια διαφορετική προσέγγιση από το Snake. Συγκεκριμένα, έγινε χρήση αντικειμενοστρεφές στυλ προγραμματισμού για την ανάπτυξή του προς ανάδειξη της δυνατότητας της JavaScript να χειριστεί και διαδικαστικό στυλ καθώς και αντικειμενοστρεφές. Χρησιμοποιούνται function «κατασκευαστές» (constructors) για όλα τα αντικείμενα, όπως η μπάλα, τα τούβλα κτλπ. Επίσης, γίνεται χρήση και της βασικής λειτουργίας requestAnimationFrame() που έχουμε αναφέρει (γραμμή 390). Ο κώδικας συνοδεύεται από σαφή σχόλια για την εύκολη διαχείρισή του.

6.10 arkanoid_leaderboard.php

```

1. <?php
2.     // Initialize the session
3.     session_start();
4.
5.     // If session variable is not set it will redirect to login page
6.     if(!isset($_SESSION['userid']) || empty($_SESSION['userid'])){
7.         header("location: ../login.php");
8.         exit;
9.     }
10.
11.    // Include config file
12.    require_once '../config.php';
13. ?>
14.
15. <html>
16. <head>
17.     <meta charset="UTF-8">
18.     <title>Arkanoid Leaderboards</title>
19.     <link rel="stylesheet" href="../css/bootstrap.css">
20. <style>
21.     #scores {
22.         font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
23.         border-collapse: collapse;
24.         width: 60%;
25.         margin-top: 10px;
26.         margin-left: 10px;
27.     }
28.
29.     #scores td, #scores th {
30.         border: 1px solid #ddd;
31.         padding: 8px;
32.     }
33.
34.     #scores tr:nth-child(even){background-color: #f2f2f2;}
35.     #scores tr:nth-child(odd){background-color: white;}
36.
37.     #scores tr:hover {background-color: #ddd;}
38.
39.     #scores th {
40.         padding-top: 12px;
41.         padding-bottom: 12px;
42.         text-align: left;
43.         background-color: #1A75FF;
44.         color: white;
45.     }
46.
47.     .logoutPos{
48.         position:fixed;
49.         right:20px;
50.         top:10px;
51.     }
52.     body{
53.         color: black;
54.         height: 100%;
55.     }
56.     html{
57.         height: 100%
58.     }
59.     .bg {
60.         background-image: url("../img/bg9.jpg");
61.         height: 100%;
62.         background-position: center;

```

```

63.         background-repeat: repeat-y;
64.         background-size: cover;
65.     }
66. </style>
67. </head>
68. <body class="bg">
69.
70.     <div class="logoutPos">
71.         <p>Hi, <b><?php echo htmlspecialchars($_SESSION['username']); ?></b>.
</p>
72.         <p>
73.             <a href="..\landing_page.php" class="btn btn-
warning">Home Page</a>
74.             <a href="..\logout.php" class="btn btn-danger">Sign Out</a>
75.         </p>
76.     </div>
77.     <table id="scores">
78.         <tr>
79.             <th style="width: 10%;>Ranking</th>
80.             <th style="width: 70%;>User</th>
81.             <th style="width: 10%;>Score</th>
82.             <th style="width: 10%;>Date</th>
83.         </tr>
84.
85.         <?php
86.             if (mysqli_num_rows($result) > 0)
87.             {
88.                 while($row = mysqli_fetch_assoc($result))
89.                 {
90.                     $var = $row["UserID"];
91.                     $query1 = "SELECT Username FROM users WHERE UserID = '$va
r'";
92.                     $result1 = mysqli_query($db, $query1);
93.                     $username = mysqli_fetch_object($result1);
94.                     $aa++;
95.                     $datetime = strtotime($row["DayAchieved"]);
96.                     $formattedDate = date("d/m/Y", $datetime);
97.
98.                     echo "<tr>";
99.                     echo "<td>".$aa."</td>";
100.                    echo "<td>".$username->Username."</td>";
101.                    echo "<td>".$row["Score"]."</td>";
102.                    echo "<td>".$formattedDate."</td>";
103.                    echo "</tr>";
104.                }
105.            }
106.            mysqli_close($db);
107.        ?>
108.    </table>
109. </body>
110. </html>

```

Εδώ βλέπουμε τον κώδικα που δημιουργεί τη σελίδα με τον πίνακα των σκορ του παιχνιδιού Arkanoid. Για το γέμισμα του πίνακα με δεδομένα έχει χρησιμοποιηθεί επαναληπτικός βρόχος while με εντολές γλώσσας PHP και queries προς τη βάση δεδομένων.

6.11 savescores.php

```
1. <?php
2.     // Initialize the session
3.     session_start();
4.
5.     // If session variable is not set it will redirect to login page
6.     if(!isset($_SESSION['userid']) || empty($_SESSION['userid'])){
7.         header("location: login.php");
8.         exit;
9.     }
10.
11.    // Include config file
12.    require_once 'config.php';
13.
14.    $game = $_GET['varGame'];
15.    $userID = $_SESSION['userid'];
16.    $score = mysqli_real_escape_string($db, $_GET['varScore']);
17.
18.    $query = "INSERT INTO scores (Game, UserID, Score, DayAchieved)
19.            VALUES ('$game', '$userID', '$score', CURDATE())";
20.
21.    if($stmt = mysqli_prepare($db, $query))
22.    {
23.        if(mysqli_stmt_execute($stmt))
24.        {
25.
26.        }
27.    }
28.    mysqli_close($db);
29. ?>
```

Τέλος, βλέπουμε το script κώδικα που αναλαμβάνει να σώσει στη βάση δεδομένων την προσπάθεια του παίκτη σε κάποιο από τα δύο παιχνίδια. Λαμβάνει από το session το id του χρήστη, από τη σελίδα του παιχνιδιού το σκορ καθώς και ποιο παιχνίδι ήταν και τα αποθηκεύει στον πίνακα scores της βάσης.

7. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

7.1 Συμπεράσματα

Στην παρούσα πτυχιακή είδαμε μια εκδοχή του πως μπορεί να στηθεί ένα σύνολο ιστοσελίδων με θέμα την ψυχαγωγία μέσω των βιντεοπαιχνιδιών. Ανάμεσα στην πληθώρα των τεχνολογιών και προσεγγίσεων σε ένα τέτοιο θέμα, η προσπάθεια αυτή μπορεί να θεωρηθεί ως η πιο προσβάσιμη και ευκατανόητη προσέγγιση για κάποιον που προσπαθεί να κατανοήσει τη λειτουργία του διαδικτύου, των βιντεοπαιχνιδιών σε browsers και τη σύμπτυξη των δύο αυτών κομματιών σε μία ενιαία λειτουργική εφαρμογή.

Με τη χρήση και εφαρμογή των βασικών τεχνολογιών διαδικτύου (HTML, CSS, JavaScript) καθώς και άλλων βοηθητικών καταφέραμε να παρουσιάσουμε με επιτυχία έναν τέτοιο ιστότοπο. Αυτό μας οδήγησε σε μεγαλύτερη κατανόηση αυτών των τεχνολογιών και μία εμπάθυση των γνώσεων που είχαμε λάβει ήδη μέσω του κύκλου μαθημάτων του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών. Με την ολοκλήρωσή της, μπορούμε να πούμε ότι έχουμε κατανοήσει σε ικανοποιητικό επίπεδο την απαιτούμενη διαδικασία που χρειάζεται για να χτίσουμε σωστά μία ιστοσελίδα, όπως και επίσης τις θεμελιώδεις έννοιες πάνω στις οποίες στηρίζεται ένα βιντεοπαιχνίδι που έχει ως στόχο να γίνει ευχάριστο και φιλικό για το χρήστη.

7.2 Μελλοντικές Επεκτάσεις

Η εφαρμογή απεικονίζει μία βασική εκδοχή του πως θα μπορούσαμε να υλοποιήσουμε έναν ιστότοπο με παιχνίδια. Για αυτό το λόγο, υπάρχουν πολλά σημεία υποψήφια προς βελτιστοποίηση, όπως επίσης και πολλές λειτουργίες που δεν έχουν εφαρμοστεί για χάρη χρόνου και ποιότητας. Κάποια από τα σημαντικότερα στοιχεία παρουσιάζονται παρακάτω:

Κατασκευή Διαδικτυακού Ιστότοπου Παιχνιδιών

- Ανασχεδίαση του συστήματος χρηστών με την προσθήκη ενός διαχειριστή, όπως και τη δυνατότητα δημιουργίας χρήστη-λογαριασμού.
- Βελτίωση της ασφάλειας της εφαρμογής (π.χ. προστασία από cross-site scripting επιθέσεις (XSS attacks)).
- Προσθήκη κάποιου είδους chat για συνομιλία μεταξύ χρηστών.
- Προσθήκη σχολίων (comments) για κάθε παιχνίδι.
- Προσθήκη νέων παιχνιδιών.
- Περισσότερες επιλογές εξατομίκευσης του λογαριασμού του χρήστη (π.χ. προσθήκη εικόνας κ.α.)

8.ΕΠΙΛΟΓΟΣ

Η ψυχαγωγία στις μέρες μας έχει άπειρες μορφές. Από τη διασκέδαση με συνανθρώπους έξω μέχρι και την τηλεόραση, η ανάγκη μας για ψυχαγωγία και διασκέδαση γίνεται ολοένα και πιο επιτακτική. Μεγάλο μερίδιο της ψυχαγωγίας και διασκέδασης κατέχουν πια και τα βιντεοπαιχνίδια, σε όποια μορφή και αν αυτά βρίσκονται. Πολλά βιντεοπαιχνίδια αποτελούν γνήσια και επιμορφωτική εμπειρία, ειδικά στις νεότερες ηλικίες αλλά όχι μόνο. Αναπτύσσουν τη δημιουργικότητα και τονώνουν την αυτοπεποίθηση αφού έχουν τον έλεγχο του χαρακτήρα του παιχνιδιού. Επίσης, βελτιώνουν τα αντανακλαστικά και το συντονισμό χεριού ματιού, καθώς και αποτελούν ένα πολύ καλό μέσο εξοικείωσης με την τεχνολογία. Βέβαια, καθώς είναι ένα από τα μέσα που διαμορφώνουν τον χαρακτήρα και τη συμπεριφορά ενός ανθρώπου, πρέπει να αντιμετωπίζονται με σύνεση και προσοχή.

Τα βιντεοπαιχνίδια είναι αυστηρώς συνδεδεμένα με την τεχνολογία. Όσο η τεχνολογία γνωρίζει ανάπτυξη, τόσο θα βλέπουμε και καινοτομίες στο χώρο των βιντεοπαιχνιδιών. Ήδη έχουν υπάρξει προσπάθειες σε νέα πεδία, όπως αυτά της τεχνητής νοημοσύνης και εικονικής πραγματικότητας. Ριζικές αλλαγές μπορούν να γίνουν και σε ζητήματα άμεσα συνδεδεμένα με τα βιντεοπαιχνίδια, όπως ο τρόπος που τα προμηθευόμαστε και μας είναι διαθέσιμα. Ακόμα, όσο περνάει ο καιρός όλο και περισσότερα εργαλεία ανάπτυξης βιντεοπαιχνιδιών (όπως Unity) γίνονται ελεύθερα προς το κοινό και δίνεται η δυνατότητα σε περισσότερους ανθρώπους να ασχοληθούν με τη δημιουργία των δικών τους βιντεοπαιχνιδιών, αυξάνοντας την ποικιλία και βελτιώνοντας την ποιότητα των βιντεοπαιχνιδιών ως σύνολο. Εν κατακλείδι, οι δυνατότητες είναι άπειρες και το μέλλον των βιντεοπαιχνιδιών φαντάζει άγνωστο και απρόβλεπτο την παρούσα στιγμή.

ΒΙΒΛΙΟΓΡΑΦΙΑ

AJAX. (2018). Ανάκτηση από Wikipedia:

[https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))

Bootstrap Example. (2015). Ανάκτηση από Wikimedia:

https://commons.wikimedia.org/wiki/File:Twitter_Bootstrap_Under_Firefox_32.png

Cascading Style Sheets. (2018). Ανάκτηση από Wikipedia:

https://en.wikipedia.org/wiki/Cascading_Style_Sheets

Chikhani, R. (2015). *Techcrunch*. Ανάκτηση από www.techcrunch.com:

<https://techcrunch.com/2015/10/31/the-history-of-gaming-an-evolving-community/>

DOM Model. (2012). Ανάκτηση από Wikimedia:

<https://commons.wikimedia.org/wiki/File:DOM-model.svg>

Github. (2018). Ανάκτηση από Github: <https://www.github.com/>

History of video games. (2018). Ανάκτηση από www.en.wikipedia.org:

https://en.wikipedia.org/wiki/History_of_video_games

HTML. (2018). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/HTML>

JavaScript. (2018). Ανάκτηση από Wikipedia:

<https://en.wikipedia.org/wiki/JavaScript>

Lofgren, K. (2017). *Bigfishgames*. Ανάκτηση από www.bigfishgames.com:

<https://www.bigfishgames.com/blog/2017-video-game-trends-and-statistics-whos-playing-what-and-why/>

MySQL. (2018). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/MySQL>

Node.js. (2018). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/Node.js>

Oculus Rift. (2017). Ανάκτηση από www.commons.wikimedia.org:

<https://commons.wikimedia.org/wiki/File:Oculus-Rift-CV1-Headset-Front.jpg>

PDP-1. (2018). Ανάκτηση από www.en.wikipedia.org:

<https://en.wikipedia.org/wiki/PDP-1>

PHP. (2018). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/PHP>

Scheme Dynamic Page. (2012). Ανάκτηση από Wikimedia:

https://commons.wikimedia.org/wiki/File:Scheme_dynamic_page_en.svg

Understanding JavaScript's requestAnimationFrame() method for smooth animations. (2016). Ανάκτηση από Javascriptkit:

<http://www.javascriptkit.com/javatutors/requestanimationframe.shtml>

Wikipedia. (2006). *Wikimedia*. Ανάκτηση από www.upload.wikimedia.org:

<https://commons.wikimedia.org/wiki/File:Voodoo3-2000AGP.jpg>

Wikipedia. (2011). *Wikimedia*. Ανάκτηση από www.commons.wikimedia.org:

https://commons.wikimedia.org/wiki/File:Four_Arcade_Games.jpg

Άλαν Τούρινγκ. (2018). Ανάκτηση από www.el.wikipedia.org:

https://el.wikipedia.org/wiki/Άλαν_Τούρινγκ

Βιντεοπαιχνίδι. (2017). Ανάκτηση από www.el.wikipedia.org:

<https://el.wikipedia.org/wiki/Βιντεοπαιχνίδι>

