

Τμήμα Μηχανικών Πληροφορικής τ.ε.

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΑΛΛΗΛΕΠΙΔΡΑΣΤΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ ΜΕ ΧΡΗΣΗ WEB ΤΕΧΝΟΛΟΓΙΩΝ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΩΝ:

Χατζίρη Έλενα, ΑΜ: 1297

Ανδρέοπουλος Κυριάκος- Μάριος, ΑΜ: 0921

ΕΠΙΒΛΕΠΩΝ: Σωτήρης Χριστοδούλου

ΑΝΤΙΡΡΙΟ 2017

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο, 2017,

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. , Υπογραφή

2. , Υπογραφή

3. , Υπογραφή

Περίληψη

Η παρούσα εργασία έχει ως αντικείμενο της την επισκόπηση, παρουσίαση, ανάλυση και κατανόηση από μέρους μας, της σημαντικής και ωφέλιμης σχέσης μεταξύ των ηλεκτρονικών παιχνιδιών και της σύγχρονης παιδαγωγικής τεχνικής. Ανατρέχοντας σε παραδοσιακές και σύγχρονες τεχνικές μάθησης, γίνεται μια προσπάθεια για να διερευνηθούν όλα τα πιθανά πλεονεκτήματα αλλά και μειονεκτήματα που μπορεί να έχει η χρήση ηλεκτρονικών παιχνιδιών και πολυμεσικών εφαρμογών στην διαδικασία της μάθησης. Γίνεται επίσης μια πιο εκτενής ανάλυση των ηλεκτρονικών παιχνιδιών και των υποομάδων που εμπεριέχονται σε αυτά με σκοπό την καλύτερη κατανόηση του καινούριου αυτού μέσου μάθησης. Στο παραπάνω πλαίσιο, πραγματοποιήθηκε ο σχεδιασμός και η υλοποίηση ενός παιχνιδιού, το οποίο ανήκει στην κατηγορία «Παιχνίδια γρίφων» (puzzle games). Αυτού του είδους τα παιχνίδια απαιτούν από τον παίχτη να λύσει διάφορους λογικούς γρίφους. Στόχος του παιχνιδιού είναι ο χρήστης να γεμίσει με 100 αριθμούς ένα πίνακα βάσει κάποιων κανόνων.

Abstract

The current projects' subject matter is the preview, presentation, analysis and overall understanding on our part, of the important and beneficial relationship between electronic video games and contemporary teaching techniques. Overviewing several traditional and contemporary learning techniques, an attempt is made to investigate all possible positive and negative consequences that stem from the usage of video games and other multimedia applications during the teaching process. A more extensive analysis on videogames and their associated sub categories is also presented with the goal of better understanding this new and important medium of learning. In the above framework, puzzle game is designed and implemented, which is in the area of "Puzzle Games". These kinds of games require the player to solve various rational puzzles. The goal of the game is to fill a table with 100 numbers based some rules.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε θερμά τον καθηγητή κ. Χριστοδούλου που μας έδωσε την δυνατότητα να πραγματοποιήσουμε την πτυχιακή μας εργασία. Για την βοήθεια που μας παρείχε από την επιλογή του θέματος αλλά και την καθοδήγηση και τις χρήσιμες συμβουλές καθ'όλη την διάρκεια της εκπόνησης της πτυχιακής αυτής εργασίας.

Πίνακας περιεχομένων

Περίληψη	iii
Abstract	iv
ΕΥΧΑΡΙΣΤΙΕΣ	v
Κεφάλαιο 1 – Ηλεκτρονικά Παιχνίδια	7
1.1 Εισαγωγή	7
1.2 Χαρακτηριστικά Ηλεκτρονικών Παιχνιδιών	7
1.3 Κατηγορίες ηλεκτρονικών παιχνιδιών	8
Κεφάλαιο 2 - Ηλεκτρονικά Παιχνίδια & Εκπαίδευση	14
2.1 Ηλεκτρονικά παιχνίδια και γνωστικές λειτουργίες	14
2.2 Επιδράσεις των ηλεκτρονικών παιχνιδιών στην εκπαίδευση	15
Κεφάλαιο 3 - Τεχνολογίες Πληροφορίας και Επικοινωνιών στην Εκπαίδευση	19
3.1 Εισαγωγή	19
3.2 Εκπαίδευση και επαγγελματικά ζητήματα	20
3.3 ΤΠΕ στην εκπαίδευση	23
Κεφάλαιο 4 - Τεχνολογίες WEB	25
4.1 HTML	25
4.2 Εισαγωγή στη Javascript	28
4.2.1. Γλώσσα προγραμματισμού Java	28
4.2.2. Από τη Java στη JavaScript	29
4.2.3. Σύγκριση Java και JavaScript	29
4.2.4. Πλεονεκτήματα JavaScript	31
Κεφάλαιο 5 - Υλοποίηση παιχνιδιού	48
5.1 Εισαγωγή	48
5.2 Κανόνες του παιχνιδιού	50
5.3 Παράρτημα – Κώδικας Υλοποίησης	56
Βιβλιογραφία	77

Κεφάλαιο 1 – Ηλεκτρονικά Παιχνίδια

1.1 Εισαγωγή

Η Κάθε μορφή διδασκαλίας, είτε αυτή διενεργείται στα πλαίσια μια συνηθισμένης τάξης είτε μέσω προηγμένων εκπαιδευτικών μεθόδων, αποσκοπεί στην ενίσχυση της μάθησης (Seelye, H. N. 1984). Το βασικότερο ερώτημα που τίθεται είναι πώς θα προσαρμόσουμε τη διδακτική πράξη με τρόπο τέτοιο ώστε το μαθησιακό αποτέλεσμα να μην είναι τυχαίο ή αναποτελεσματικό, αλλά όσο το δυνατό πιο αποδοτικό, αποτελεσματικό και ουσιαστικό. Οι σύγχρονες ψυχοπαιδαγωγικές θεωρίες αναφέρουν ως απαραίτητη προϋπόθεση για την επίτευξη των μαθησιακών στόχων τη βαθμιαία αυτονόμηση του εκπαιδευομένου από το παραδοσιακό παθητικό δασκαλοκεντρικό σύστημα μετάδοσης γνώσης και την ενσωμάτωση του σε μια ενεργητική μαθησιακή διαδικασία (Cordova, D. I., & Lepper, M. R. 1996). Μια διαδικασία κατά την οποία θα εξερευνά, θα ανακαλύπτει, θα επιλέγει πληροφορίες τις οποίες θα επεξεργάζεται και θα μετουσιώνει σε δομημένη γνώση. Η υποκίνηση αποτελεί μια από τις βασικότερες προϋποθέσεις για τη δραστηριοποίηση του εκπαιδευομένου. Είναι εκείνη η δύναμη που συγκρατεί ένα πρόσωπο μέσα στη μαθησιακή διαδικασία και το ενθαρρύνει να μάθει (Rogers, 1999). Ο υποκινούμενος εκπαιδευόμενος είναι ενθουσιώδης, εστιασμένος και «δεσμευμένος» στο στόχο του καθώς και οι δραστηριότητες στις οποίες συμμετέχει είναι ενδιαφέρουσες, αλλά και η επίτευξη των επιθυμητών αποτελεσμάτων (και των επιβραβεύσεων) θεωρείται σημαντική. Το ενδιαφέρον του για γνώση είναι έκδηλο και νιώθει ευχάριστα καθώς μαθαίνει. Η συμπεριφορά του στηρίζεται σε εσωτερικά κίνητρα και δεν είναι αποτέλεσμα εξωγενών και μόνο παραγόντων (Garris et al., 2002). Τα τελευταία χρόνια οι αναπτυσσόμενες Τεχνολογίες της Πληροφορίας και της Επικοινωνίας (ΤΠΕ), συνέβαλαν στη ραγδαία εξέλιξη της Εκπαιδευτικής Τεχνολογίας προσφέροντας στην εκπαίδευση δυνατότητες που πριν από λίγα χρόνια θα ήταν ασύλληπτες. Ως εκπαιδευτικά εργαλεία θα μπορούσαν να χρησιμοποιηθούν πολλαπλά νέα μέσα για την ανάπτυξη περιβαλλόντων που θα καθιστούν ενεργητικό τον εκπαιδευόμενο.

1.2 Χαρακτηριστικά Ηλεκτρονικών Παιχνιδιών

Μερικά βασικά στοιχεία που χαρακτηρίζουν τα ηλεκτρονικά παιχνίδια είναι η φαντασία, η πρόκληση και η περιέργεια. Για να μπορέσει ένα περιβάλλον να χαρακτηριστεί ότι δημιουργεί

πρόκληση θα πρέπει να προβάλλει στόχους των οποίων η επίτευξη είναι αμφίβολη (Malone, 1980).

Τα ηλεκτρονικά παιχνίδια είναι μέρος της κατηγορίας των ενεργητικών περιβαλλόντων που προκαλούν την ενασχόληση, η σχεδίαση των οποίων θα πρέπει να ακολουθεί συγκεκριμένους κανόνες και να εμπεριέχουν μεταξύ άλλων (Fabricatore, C. 2000):

- Δραστηριότητες με εμφανείς στόχους, οι οποίες πρέπει να ολοκληρωθούν και που παρέχουν άμεση επαναληπτικότητα
- Δραστηριότητες που απαιτούν συγκέντρωση
- Εξάσκηση και ανάπτυξη των ενεργειών του χρήστη και των ικανοτήτων του
- Ενασχόληση που να απαιτεί έναν βαθμό αφοσίωσης και να έχει ως αποτέλεσμα ο χρήστης να «ξεχνιέται» από την καθημερινότητα του και τα προβλήματα που τον απασχολούν

Τα προαναφερθέντα χαρακτηριστικά πρέπει να συνδυάζονται με τα βασικά χαρακτηριστικά των ηλεκτρονικών παιχνιδιών (Prensky, 2001) που είναι: οι κανόνες του παιχνιδιού, οι στόχοι που θέτει το παιχνίδι, τα αποτελέσματα από την προσπάθεια επίτευξης των στόχων αυτών και η επανατροφοδότηση των στόχων αυτών, η δημιουργία ενός περιβάλλοντος που γεννάει τον ανταγωνισμό μέσω της αλληλεπίδρασης των χρηστών και τέλος η αναπαράσταση κάποιου ιστορικού γεγονότος ή ύπαρξη σεναρίου.

Βάσει των παραπάνω στοιχείων, ένα ηλεκτρονικό παιχνίδι θα πρέπει να προβάλλει συγκεκριμένους κανόνες τους οποίους να μπορεί να χρησιμοποιήσει ο χρήστης για να καταφέρει να φέρει σε πέρας τους στόχους που έχουν τεθεί κατά την σχεδίαση του. Οι δραστηριότητες θα πρέπει να είναι δομημένες με τέτοιο τρόπο ώστε να μπορεί ο χρήστης να μειώσει ή να αυξήσει τη δυσκολία τους με σκοπό να μπορέσει να ανταπεξέλθει ανάλογα με τις ικανότητές του. Θα πρέπει να υπάρχουν καθαρά κριτήρια αξιολόγησης των προσπαθειών έτσι ώστε ο χρήστης να γνωρίζει πόσο καλά ή πόσο άσχημα τα πηγαίνει ανά πάσα στιγμή. Τέλος το σενάριο θα πρέπει να διαδραματίζεται με τέτοιο τρόπο που να προσελκύει το ενδιαφέρον του χρήστη και να γεννάει την επιθυμία να αλληλεπιδράσει με το περιβάλλον.

1.3 Κατηγορίες ηλεκτρονικών παιχνιδιών

Εξαιτίας όμως, της μεγάλης ποικιλίας των ηλεκτρονικών παιχνιδιών που έχουν δημιουργηθεί, των διαφόρων χαρακτηριστικών τους, των στόχων που επιδιώκουν να επιτύχουν, τον αριθμό

των παιχτών που απευθύνονται και άλλων στοιχείων τους, είναι δύσκολο να βρεθεί μία κοινή κατηγοριοποίηση, καθώς τα παιχνίδια αυξάνονται συνεχώς και πολλές φορές παρατηρούμε πως ένα παιχνίδι δεν μπορεί να ανήκει σε μία και μόνο κατηγορία. Ο Caillois το 1958 (Prensky, 2001), ταξινόμησε τα ηλεκτρονικά παιχνίδια σε τέσσερις βασικές κατηγορίες: παιχνίδια ανταγωνισμού, τύχης, προσομοίωσης και κίνησης. Ακολούθησαν διάφορες κατηγοριοποιήσεις παιχνιδιών από έρευνες που έχουν πραγματοποιηθεί σε αυτό το πεδίο, από σχεδιαστές παιχνιδιών, από ιστοσελίδες ανάλογου περιεχομένου κλπ. Παρόλα αυτά δεν υπάρχει ακόμα μια κατηγοριοποίηση που να περιλαμβάνει όλα τα ηλεκτρονικά παιχνίδια.

Παρακάτω παρουσιάζεται μια λίστα με κατηγορίες ηλεκτρονικών παιχνιδιών και μια σύντομη περιγραφή και παραδείγματα από το κάθε είδος, από την online εγκυκλοπαίδεια wordiq: (http://www.wordiq.com/definition/Computer_and_video_game_genres):

1) Μαζικά πολυχρηστικά διαδικτυακά παιχνίδια και μαζικά πολύχρηστικά διαδικτυακά παιχνίδια ρόλων (Massive(ly) multiplayer online games and massive(ly) multiplayer online role-playing games – MMOGs and MMORPGs): τα παιχνίδια αυτά βασίζονται σε εικονικούς κόσμους για χιλιάδες παίκτες που αλληλεπιδρούν μεταξύ τους. Τέτοια παιχνίδια είναι τα εξής: Meridian 59, Ultima Online, Ever Quest, Final Fantasy XI και Lineage.

2) Εκπαιδευτικά παιχνίδια (educational games): σκοπός αυτών των παιχνιδιών είναι να διδάξουν το χρήστη και απευθύνονται σε άτομα από 3 περίπου ετών έως εφήβους και ενήλικες. Υπάρχουν πολυάριθμα εκπαιδευτικά παιχνίδια και το καθένα για διαφορετικό γνωστικό πεδίο. Δημοφιλή παιχνίδια σ' αυτή την κατηγορία είναι: Carmen Sandiego, Mavis Beacon Teaches Typing και Oregon Trail.

3) Διαδραστικές ταινίες (Interactive movies): αυτά τα παιχνίδια έχουν προ – κινηματογραφηθεί και περιέχουν πλήρη κίνηση των κινουμένων σχεδίων ή ζωντανές σκηνές δράσεις, στις οποίες ο παίχτης ελέγχει κάποιες από τις κινήσεις των βασικών χαρακτήρων. Η μόνη δραστηριότητα του παίχτη είναι να επιλέξει ή να μαντέψει την κίνηση που οι σχεδιαστές του παιχνιδιού τον προόριζαν να κάνει. Αν και αρχικά ήταν πολύ δημοφιλή και εθιστικά παιχνίδια, σύντομα ξεχάστηκαν. Το πιο γνωστό παιχνίδι αυτής της κατηγορίας είναι το Dragon's Lair.

4) Παιχνίδια retro: είναι τα πρώτα παιχνίδια και χρησιμοποιούν μια πολύ απλή διεπαφή, η οποία περιλαμβάνει ένα απλό σύστημα καθοδήγησης τεσσάρων κατευθύνσεων και μια βασική εντολή. Δεν πρόκειται για μια συγκεκριμένη κατηγορία, καθώς πολλά παιχνίδια μπορούν να ενταχθούν σε αυτήν. Τέτοια παιχνίδια είναι: Pong και Pac – Man.

5) Παραδοσιακά παιχνίδια (traditional games): πρόκειται για τα πιο δημοφιλή επιτραπέζια παιχνίδια ή παιχνίδια καρτών, τα οποία έχουν μηχανογραφηθεί και συμβάλουν στη βελτίωση των ικανοτήτων του ατόμου σε παραδοσιακά παιχνίδια. Χαρακτηριστικά παιχνίδια είναι το Σκάκι, η Ντάμα και το Τάβλι.

6) Παιχνίδια πλατφόρμας (platform games): τα παιχνίδια αυτά περιορίζουν τον παίκτη να κινείται σε οριζόντιες επιφάνειες, οι οποίες αναφέρονται ως πλατφόρμες. Παραδοσιακά στοιχεία αυτών είναι το τρέξιμο, τα άλματα και το ανέβασμα σε σκάλες. Περιέχουν επίσης στοιχεία μάχης και σκοποβολής. Είναι ένα από τα πρώτα είδη παιχνιδιών, αλλά έχασαν τη δημοτικότητά τους μετά την ανάπτυξη των 3D γραφικών. Μερικά παραδείγματα αυτού του είδους είναι: Donkey Kong, Super Mario Bros, Lode Runner, Sonic the Hedgehog, Spyro the Dragon και Crash Bandicoot.

7) Παιχνίδια περιπέτειας (adventure games): ο παίκτης είναι ο πρωταγωνιστής μιας ιστορίας και απαιτείται να λύσει διάφορους γρίφους ή να βρει αντικείμενα προκειμένου να συνεχίσει την πορεία του. Τα πρώτα παιχνίδια περιπέτειας περιείχαν κείμενο, ενώ στη συνέχεια εμπλουτίστηκαν και με οπτικά εφέ. Χάρη στην τεχνολογική πρόοδο, τα παιχνίδια αυτά απέκτησαν μεγαλύτερη διεπαφή χρήστη – υπολογιστή, χρησιμοποιώντας απλά μια συσκευή κατάδειξης. Η παραγωγή αυτών των παιχνιδιών ξεκίνησε το 1970 με το παιχνίδι Colossal Cave Adventure. Ακολούθησαν τα Zork, Day of the Tentacle, King’s Quest, το Monkey Island.

8) Παιχνίδια γρίφων (puzzle games): αυτού του είδους τα παιχνίδια απαιτούν από τον παίκτη να λύσει διάφορους λογικούς γρίφους ή να πλοηγηθεί σε περίπλοκες περιοχές, όπως ένας λαβύρινθος. Αυτό το είδος παιχνιδιών διασταυρώνεται συχνά με παιχνίδια περιπέτειας και

εκπαιδευτικά παιχνίδια. Δημοφιλέστερα αυτού του είδους είναι τα εξής: ο Ναρκαλιευτής (Minesweeper), Q*Bert και Tetris.

9) Παιχνίδια μάχης (fighting games): πρόκειται για παιχνίδια που επικεντρώνονται στις πολεμικές τέχνες και δίνουν έμφαση στη μάχη ένας προς ένα, ανάμεσα σε δύο παίκτες, ένας εκ των οποίων μπορεί να είναι και ο ίδιος ο υπολογιστής. Γνωστά παιχνίδια αυτής της κατηγορίας είναι: King of Fighters, Mortal Kombat, Street Fighter, Soul Edge, Soul Calibur, Tekken και Virtua Fighter

10) First – person shooter παιχνίδια: τα παιχνίδια αυτά «τοποθετούν» τον παίκτη πίσω από ένα όπλο και δίνουν έμφαση στην καταπολέμηση του εχθρού από μία συγκεκριμένη προοπτική. Αυτή η προοπτική αποσκοπεί στο να δώσει στον παίκτη την αίσθηση του ότι «είναι εκεί». Τα περισσότερα first – person shooter παιχνίδια είναι πολύ βίαια, γρήγορα και απαιτούν γρήγορα αντανακλαστικά. Μερικά από τα πιο δημοφιλή παιχνίδια σήμερα είναι: Doom, Half – Life, Halo, Quake και Unreal.

11) Third – person shooter παιχνίδια: χρησιμοποιούν συγκεκριμένη προοπτική για τον παίκτη, ο οποίος είναι συνήθως πίσω από τον χαρακτήρα του εκάστοτε παιχνιδιού. Πολλά από αυτά τα παιχνίδια κατατάσσονται σε άλλες κατηγορίες. Μερικά παραδείγματα της κατηγορίας αυτής είναι το Grand Theft Auto, Heretic II, Mafia, Oni κ.α.

12) Fix shooter παιχνίδια: αυτά τα παιχνίδια χρησιμοποιούν συνήθως μια σταθερή προοπτική και ένα απλό σύστημα ελέγχου. Αποσκοπούν στο να «πετύχει» ο χρήστης τον εχθρό, ο οποίος μπορεί να είναι άνθρωπος, εξωγήινος ή έντομο. Αν και η δημοτικότητα αυτών των παιχνιδιών έχει πέσει, μερικά από τα πιο γνωστά παιχνίδια είναι: Space Invaders, Asteroids, Missile Command, Galada και Robotron: 2084.

13) Shoot 'Em Up παιχνίδια: γνωστά και ως scrolling shooters, τονίζουν τον γρήγορο ρυθμό πυροβολισμών σε εάν κυλιόμενο γήπεδο – πεδίο, το οποίο μπορεί να είναι είτε οριζόντιο, είτε κατακόρυφο. Η διάκρισή τους από τα άλλα shooter games δεν είναι ξεκάθαρη, αλλά είναι μια από τις πρώτες κατηγορίες παιχνιδιών. Το πρώτο παιχνίδι δημιουργήθηκε το 1980 και ήταν το

Defender. Το 1982 ακολούθησε το Xevious. Άλλα γνωστά παιχνίδια είναι το Gradius, το Darius και το R-Type.

14) Παιχνίδια λαθραίων πράξεων (stealth games): πρόκειται για μια σχετικά πρόσφατη κατηγορία και συχνά αναφέρονται ως «όπουλα» παιχνίδια. Έχουν πολλά κοινά με τις κατηγορίες first – person shooter και third – person shooter παιχνίδια και δίνουν έμφαση στα τεχνάσματα και την ακρίβεια των χτυπημάτων. Τέτοια παιχνίδια είναι τα: Thief, Metal Gear Solid, Splinter Cell και Beyond Good and Evil.

15) Παιχνίδια επιβίωσης τρόμου (survival horror games): τα παιχνίδια αυτά επικεντρώνονται στο φόβο και προσπαθούν να «τρομάξουν» τον παίχτη, χρησιμοποιώντας στοιχεία της μυθοπλασίας τρόμου, όπως το αίμα, ο θάνατος, οι αθάνατοι κλπ. Και αυτή η κατηγορία περιέχει πολλά στοιχεία από τα first –person shooter παιχνίδια. Χαρακτηριστικά παιχνίδια της κατηγορίας αυτής είναι: Resident Evil, Alone in the Dark, System Shock, Silent Hill και Fatal Frame (Project Zero).

16) Μουσικά παιχνίδια (music games): σε αυτού του είδους τα παιχνίδια η πρόκληση που έχει να αντιμετωπίσει ο παίχτης είναι να ακολουθήσει κάποιες μουσικές ακολουθίες ή να αναπτύξει συγκεκριμένους ρυθμούς. Ορισμένα από αυτά απαιτούν από το χρήστη να συνθέσει ρυθμούς, χρησιμοποιώντας κάποιο χειριστήριο ή το πληκτρολόγιο, ενώ άλλα απαιτούν να χορεύει πραγματικά σε συγχρονισμό με τη μουσική. Ένα από τα πιο γνωστά παιχνίδια της κατηγορίας αυτής είναι το Dance Revolution.

17) Αγωνιστικά παιχνίδια (racing games): πρόκειται για μια από τις πιο παραδοσιακές κατηγορίες παιχνιδιών. Εδώ ο παίχτης «κάθεται» τυπικά στο κάθισμα του οδηγού ενός οχήματος υψηλών επιδόσεων και καλείται να ανταγωνιστεί άλλους οδηγούς ή να ολοκληρώσει μια διαδρομή σε συγκεκριμένο χρόνο. Τα παιχνίδια αυτά πρωτοεμφανίστηκαν το 1980 και εξακολουθούν να είναι πολύ δημοφιλή. Αξιοσημείωτα αγωνιστικά παιχνίδια είναι: Out Run, Gran Turismo και Mario Kart.

18) Αθλητικά παιχνίδια (sports games): τα παιχνίδια αυτά «μιμούνται» τα παραδοσιακά αθλήματα όπως το ποδόσφαιρο, το γκολφ, το χόκεϋ, το μπάσκετ, το τένις κλπ. Άλλα παιχνίδια δίνουν έμφαση στον πραγματικό τρόπο με τον οποίο παίζεται το εκάστοτε άθλημα και άλλα στη στρατηγική πίσω από αυτό. Αν και η κατηγορία αυτή εμφανίστηκε νωρίς στην ιστορία των ηλεκτρονικών παιχνιδιών παραμένει αρκετά δημοφιλής και ανταγωνιστική ακόμα και σήμερα. Τα πιο γνωστά παιχνίδια της κατηγορίας αυτής είναι το Championship Manager, το FIFA, το Pro evolution soccer και το Arch Rivals.

19) Παιχνίδια στρατηγικής (strategy games): τα παιχνίδια αυτά επικεντρώνονται στον προσεκτικό σχεδιασμό και στην επιδέξια διαχείριση των πόρων προκειμένου να επιτευχθεί η νίκη. Απευθύνονται σε εφήβους και σε ένα γενικότερα πιο ώριμο κοινό και χαρακτηρίζονται ως παιχνίδια σκέψης. Περιλαμβάνουν δύο υποκατηγορίες, αυτή των turn - based και αυτή των realtime. Η συντριπτική πλειοψηφία των παιχνιδιών στρατηγικής θα μπορούσε να χαρακτηριστεί και ως «παιχνίδια στρατηγικής πολέμου». Μερικά από τα πιο γνωστά παιχνίδια είναι το Warcraft, το Star Craft, το Sid Meier's Civilization και το Heroes of Might.

20) Παιχνίδια ρόλων (computer role – playing games): τα παιχνίδια ρόλων περιέχουν στοιχεία από τα παιχνίδια στρατηγικής και τοποθετούν τον παίκτη σε μια «φανταστική» θέση. Τα περισσότερα από αυτά είναι παρόμοια με τα παραδοσιακά παιχνίδια ρόλων που παίζονται με χαρτί και μολύβι, με μόνη διαφορά ότι ο υπολογιστής αναλαμβάνει την τήρηση των αρχείων. Ο παίκτης έχει το ρόλο ενός τυχοδιώκτη που ειδικεύεται σε ένα συγκεκριμένο σύνολο δεξιοτήτων, οι οποίες ονομάζονται «κλάσεις». Επίσης, έχει τη δυνατότητα να χρησιμοποιήσει έναν ή περισσότερους χαρακτήρες με πολλές και διαφορετικές δεξιότητες. Το πιο γνωστό παιχνίδι αυτής της κατηγορίας, με το οποίο έγινε και γνωστή, είναι το Diablo.

21) Σοβαρά παιχνίδια (serious games) και παιχνίδια προσομοίωσης (simulation games): πρόκειται για μια πολύ νέα κατηγορία παιχνιδιών που απευθύνεται σε ένα ευρύ κοινό και αποσκοπεί στο να διδάξει έννοιες του πραγματικού κόσμου μέσα από παιχνίδια που προσομοιώνουν μια συγκεκριμένη δραστηριότητα, όσο γίνεται πιο ρεαλιστικά. Χαρακτηριστικά παιχνίδια αυτού του είδους είναι: Full Spectrum Warrior, The Sims, Sim City και Sim Earth.

Κεφάλαιο 2 - Ηλεκτρονικά Παιχνίδια & Εκπαίδευση

2.1 Ηλεκτρονικά παιχνίδια και γνωστικές λειτουργίες

Η κύρια αιτία που παρουσιάζεται τέτοιο ενδιαφέρον για την διερεύνηση της σχέσης μεταξύ των ηλεκτρονικών παιχνιδιών και της εκπαίδευσης είναι η δύναμη που έχουν τα ηλεκτρονικά παιχνίδια να παρακινούν τους νέους να ασχολούνται με αυτά με έναν μοναδικό τρόπο, κάτι που δεν έχει τη δύναμη να κάνει ο τυπικός τρόπος εκπαίδευσης. Οι νέοι, και όχι μόνο αυτοί, με δική τους πρωτοβουλία και επιθυμία εμπλέκονται αρκετές ώρες με το παιχνίδι διαφόρων συχνά αρκετά περίπλοκων ηλεκτρονικών παιχνιδιών έξω από το χώρο του σχολείου. Τα ηλεκτρονικά παιχνίδια έχουν μία δυναμική στο να προκαλούν τους νέους να ασχοληθούν με αυτά κεντρίζοντας τους το ενδιαφέρον. Ως εκ τούτου έχουμε την υποχρέωση να ερευνήσουμε τα πιθανά σενάρια με τα οποία θα μπορούσαμε να χρησιμοποιήσουμε αυτή τη δυναμική για να αναζωπυρώσουμε την επιθυμία των μαθητών για μάθηση.

Τα ηλεκτρονικά παιχνίδια έχουν τη δυνατότητα, πέραν της προαναφερθείσας δυναμικής, να “επαναπροσδιορίζουν” τη σκέψη των ενασχολούμενων με αυτά (Prensky, 2001) καθώς αυτοί αναπτύσσουν νέες ικανότητες και γνωστικές λειτουργίες όπως:

- όξυνση των αντανακλαστικών
- αντίληψη πληροφοριών μέσω γραφικών
- δυνατότητα εκτέλεσης ταυτόχρονων λειτουργιών
- Τυχαία προσπέλαση
- Ενεργητικότητα
- Ανάπτυξη επικοινωνίας
- Φιλική αντιμετώπιση της τεχνολογίας

Όλα τα παραπάνω αναπτύσσονται παράλληλα με την ικανότητα των χρηστών να επεξεργάζονται τις πληροφορίες οι οποίες τους παρουσιάζονται πολύ γρήγορα καθώς πρέπει να προσδιορίσουν τι είναι σχετικό και αναγκαίο κατά τη διαδικασία, αναπτύσσουν την ικανότητα να επεξεργάζονται την πληροφορία παράλληλα την ίδια στιγμή από διάφορες πηγές και με τυχαία σειρά, αναπτύσσουν την ικανότητα να αναγνωρίζουν την πληροφορία πρώτα μέσω των εικόνων και των γραφικών και στη συνέχεια να χρησιμοποιούν τυχόν κείμενο για να

εξακριβώνουν, να αναπτύσσουν και να εξερευνούν. Επίσης αναπτύσσεται η επικοινωνία των χρηστών οι οποίοι επικοινωνούν μεταξύ τους είτε για να παίξουν μεταξύ τους είτε για να αναζητήσουν περισσότερες πληροφορίες και μεθόδους επίλυσης των προβλημάτων που τους παρουσιάζονται μέσα στο παιχνίδι. Οι χρήστες βρίσκουν το παιχνίδι σαν μία φυσική δραστηριότητα που τους οδηγεί στην θεώρηση του υπολογιστή σαν ένα εργαλείο για παιχνίδια, περιμένουν την αναγνώριση των προσπαθειών τους μέσα από το παιχνίδι, κάτι που τους δίνεται σαν ανατροφοδότηση και τους παρακινεί να συνεχίσουν σε ακόμα πιο δύσκολες διαδικασίες. Μέσω αυτής της διαδικασίας αναπτύσσεται το μοντέλο “κάνω για να μάθω” σε αντίθεση με το μοντέλο “μαθαίνω για να κάνω”. Ακόμη η φαντασία των χρηστών αναπτύσσεται μέσα από το περιβάλλον του παιχνιδιού και της εμπειρίας που αποκτούν από την αλληλεπίδραση με αυτό, αποτέλεσμα της οποίας είναι να γεννούνται οι κατάλληλες συνθήκες που θα κάνουν έναν χρήστη να εκτιμήσει την τεχνολογία καθώς μεγαλώνουν με αυτή.

2.2 Επιδράσεις των ηλεκτρονικών παιχνιδιών στην εκπαίδευση

Στο τρέχον κεφάλαιο παρουσιάζονται μια σειρά από ερευνητικές αναφορές, μέσω των οποίων οι ερευνητές προσπαθούν να αναδείξουν του εκπαιδευτικού χαρακτήρα των ηλεκτρονικών παιχνιδιών και της σχέσης τους με σύγχρονες θεωρίες μάθησης. Πιο συγκεκριμένα, εξετάζεται η εκπαιδευτική δυναμική αυτών μέσα από τη διερεύνηση των ιδιοτήτων της υποκίνησης, της διασκέδασης και της πρόκλησης ενδιαφέροντος οι οποίες θεωρείται ότι έχουν την δυνατότητα να οδηγήσουν σε θετικά μαθησιακά αποτελέσματα. Προηγούμενες έρευνες συνηγορούν στο ότι τα ηλεκτρονικά παιχνίδια προσελκύουν τους νέους ανθρώπους και φαίνεται να τους υποκινούν με ένα περισσότερο εποικοδομητικό τρόπο από αυτόν που υιοθετεί έως τώρα η συμβατική εκπαίδευση (Klawe, 1999; et al). Συγκεκριμένα, έρευνες πάνω σε ηλεκτρονικά παιχνίδια ανέδειξαν ότι μερικά χαρακτηριστικά προκαλούν στους χρήστες εσωτερική υποκίνηση (Malone, 1981; Cordova and Lepper, 1996). Μερικά από αυτά τα χαρακτηριστικά είναι

- Η φαντασία
- Ο έλεγχος
- Η πρόκληση
- Η περιέργεια
- Ο ανταγωνισμός

Υπογραμμίζεται επίσης ότι η αποτελεσματικότητα της εσωτερικής υποκίνησης κρύβεται στο γεγονός ότι το άτομο που εκπαιδεύεται συμμετέχει σε μαθησιακές δραστηριότητες χωρίς να έχει την απαίτηση να ανταμειφθεί.

Μέσω των πρώτων κιάλας ερευνών που πραγματοποιήθηκαν για την χρήση των παιχνιδιών στην εκπαίδευση (Gordon, 1970) διαπιστώθηκε ότι αποτελούν μία πηγή κινήτρου για τους χρήστες προκειμένου να δοκιμάσουν, να αναπτύξουν και να εφαρμόσουν τις γνώσεις τους, καθώς και να μάθουν πράγματα που δεν γνωρίζουν, ενώ ταυτόχρονα διασκεδάζουν (Malone, 1980). Ο Prensky (2002) αναφέρει ότι μάθηση και διασκέδαση δεν είναι ασυμβίβαστες έννοιες, ενώ οι Lepper & Cordova (1992) θεωρούν ότι η μάθηση που είναι διασκεδαστική μπορεί να είναι και αποτελεσματική. Επιπρόσθετα, οι Sandford, R., & Williamson, B. (2005)..επισημαίνουν ότι τα παιχνίδια είναι δυνατόν να προσδώσουν ένα ελκυστικό και ευχάριστο χαρακτήρα στην μάθηση, προσφέροντας μια ισχυρή «φόρμα» για τον σχεδιασμό αποτελεσματικών εκπαιδευτικών περιβαλλόντων. Χαρακτηριστικά όπως

- η χρήση των πολυμέσων
 - οι ελκυστικές ιστορίες που παρουσιάζουν πραγματικούς ή φανταστικούς στόχους
 - οι «πράκτορες» που συνοδεύουν τους χρήστες κατά τη διάρκεια του παιχνιδιού
- αποτελούν ελκυστικά στοιχεία που ενισχύουν την μαθησιακή επίτευξη (Klawe, 1999).

Κατά τον Papert (1993) τα παιχνίδια μέσω υπολογιστή εμπλουτίζουν την διδασκαλία με ένα γρήγορο και κατά συνέπεια ενδιαφέρον ρυθμό στη, σε αντίθεση με τα συμβατικά διδακτικά μέσα τα οποία καθιστούν αργή και βαρετή την εκπαιδευτική διαδικασία Gee, J. P. (2003). Τέτοιου είδους εκπαιδευτικά περιβάλλοντα μπορούν τόσο να στηρίζουν τις ήδη υπάρχουσες μαθησιακές αρχές Kirriemuir, J., & McFarlane, A. (2004), όσο και να αποτελέσουν μια νέα μορφή ηλεκτρονικής μάθησης Fudenberg, D., & Levine, D. K. (1998). Οι σημερινοί μαθητές είναι πιθανόν να απολαμβάνουν περισσότερο μια εμπειρία μάθησης ενσωματωμένη σε ηλεκτρονικό παιχνίδι. Αυτό συμβαίνει διότι τα ηλεκτρονικά παιχνίδια βασίζονται στην αρχέγονη μορφή μάθησης «παίζω και μαθαίνω» απ' όπου κι αντλούν τα πλεονεκτήματα τους ως εκπαιδευτικό μέσο. Επίσης, βασίζονται σε σύγχρονες θεωρίες μάθησης και καινοτόμα μαθησιακά μοντέλα, όπως

- η «βιωματική μάθηση»

- η «ανακαλυπτική μάθηση»
- η «μάθηση μέσω συμμετοχής σε κοινότητες» (Facer, 2003; Shaffer et al., 2004).

Υποστηρίζονται τέλος από το ανεπτυγμένο και σύγχρονο περιβάλλον μάθησης που προσφέρουν οι υπολογιστές, κάτι που είναι άμεσα αποδεκτό από την πλειοψηφία των εκπαιδευομένων. Στον αντίποδα είναι πολύ πιθανόν οι εκπαιδευόμενοι να παρουσιάζουν έλλειψη προθυμίας να δεσμευθούν σε μια διαδικασία που χρησιμοποιεί παραδοσιακούς τρόπους μάθησης και μέσων, που συνήθως γίνονται αντιληπτά από αυτούς ως επίπονα και κουραστικά (BECTA, 2002; Prensky, 2002).

Παρακάτω παρουσιάζονται τα βασικότερα πλεονεκτήματα αλλά και τα μειονεκτήματα που μπορούν να προσφέρουν τα ηλεκτρονικά παιχνίδια στον χώρο της εκπαίδευσης Kafai, Y. B. (2006). Μερικά από τα πλεονεκτήματα είναι τα εξής:

- 1) Χρήση παιχνιδιών προσομοίωσης θα μπορούσε να γίνει για την προετοιμασία των μαθητών στον χώρο εργασίας.
- 2) Ανάπτυξη βασικών δεξιοτήτων μάθησης, όπως: γνωστική επεξεργασία, λογική σκέψη και ανεξάρτητη λήψη αποφάσεων.
- 3) Ενθάρρυνση διαπροσωπικών σχέσεων, με αποτέλεσμα τη συλλογική και ανταγωνιστική συμπεριφορά σε ένα στρατηγικό πλαίσιο
- 4) Δυνατότητα ενσάρκωσης διαφορετικών χαρακτήρων, βοηθώντας έτσι να αναπαραχθούν συμπεριφορές ανοχής και κατανόησης.
- 5) Εμπλοκή σε δραστηριότητες που θα ήταν πολύ δαπανηρές ή δύσκολες να εφαρμοστούν μέσα στην τάξη.
- 6) Επιθετικά παιχνίδια μπορούν να βοηθήσουν τους παίκτες να χαλαρώσουν και να ισορροπήσουν την επιθετικότητα τους.
- 7) Ανάπτυξη σημαντικών φυσικών ιδιοτήτων, όπως οι κινητικές δεξιότητες.
- 8) Συμβολή στην επίλυση προβλημάτων, τη λήψη αποφάσεων και τη βραχυπρόθεσμη και μακροπρόθεσμη μνήμη.
- 9) Σύνδεση μεταξύ των δραστηριοτήτων μέσα στην τάξη και τη ζωή έξω από το σχολείο. Μια τέτοια σύνδεση θα συμβάλει στην ενίσχυση της μάθησης και μπορεί να ενθαρρύνει τους μαθητές να συνεχίσουν να αναπτύσσουν τις δεξιότητές τους έξω από το περιβάλλον της τάξης.

10) Παροχή ενός ασφαλούς τεχνητού περιβάλλοντος στο οποίο οι μαθητές με χαμηλή αυτοεκτίμηση μπορούν να αισθάνονται περισσότερο διατεθειμένοι να εξερευνήσουν, να διερευνήσουν και να εκφραστούν.

Μερικά από τα μειονεκτήματα είναι τα εξής

- 1) Δυνατότητες που δεν προϋπάρχουν είδη στο άτομο σε κάποιο βαθμό είναι αδυνατούν να εξελιχθούν
- 2) Ελκυστικά παιχνίδια μπορούν να οδηγήσουν το άτομο σε εθισμό με άσχημα αποτελέσματα για το άτομο σε κοινωνικό
- 3) Η βιαιότητα των χαρακτήρων και του περιβάλλοντος πολλών παιχνιδιών μπορούν να ενθαρρύνουν το άτομο να προβεί σε αντικοινωνική και επιθετική συμπεριφορά
- 4) Απόσπαση της προσοχής και από την μελέτη. Όταν το άτομο παύει να βλέπει το παιχνίδι ως εργαλείο μάθησης
- 5) Αρνητικό αντίκτυπο σε ευαίσθητα μικρά παιδιά που είναι ιδιαίτερα επιρρεπή σε συμπεριφορές της αντιγραφής και μπορεί να δυσκολευτούν στη διάκριση ανάμεσα στην πραγματικότητα και ένα εικονικό περιβάλλον ηλεκτρονικού υπολογιστή.
- 6) Ο εθισμός από την υπερβολική χρήση μπορεί να οδηγήσει το άτομο στο να διαπράξει παράνομες πράξεις προκειμένου να μπορέσει να συνεχίσει να παίζει, επίσης αισθητή είναι και η αρνητική επίδραση στην αυτοεκτίμηση του χρήστη
- 7) Η υπερβολική ενασχόληση μπορεί να δημιουργήσει πληθώρα σωματικών προβλημάτων όπως για παράδειγμα μυϊκούς πόνους
- 8) Η πλειοψηφία των ηλεκτρονικών παιχνιδιών προσανατολίζεται σε συγκεκριμένο φύλο. Η συντριπτική πλειοψηφία των παιχνιδιών παίζεται από άντρες και αυτό μπορεί να αποξενώσει τις γυναίκες – παίκτριες.

Συνοψίζοντας, καταλήγουμε στο συμπέρασμα πως τα ηλεκτρονικά παιχνίδια, με την κατάλληλη αξιοποίησή τους, έχουν τη δυνατότητα να βοηθήσουν τους παίκτες – μαθητές να αναπτύξουν δεξιότητες στρατηγικής και αναλυτικής σκέψης, ικανότητες επίλυσης προβλημάτων, σχεδιασμού και λήψης αποφάσεων (Gee, 2005; Hickmott, 2006). Οι χρήστες μπορούν να γίνουν παραγωγοί, να αναλάβουν κινδύνους, να διερευνήσουν, να αποτύχουν και

να συνεχίσουν. Να αποκτήσουν κίνητρα για μάθηση και χρησιμοποιώντας όλα τα «έξυπνα εργαλεία» που έχουν στη διάθεσή τους, να «παίζουν» το δικό τους παιχνίδι (Gee 2003; 2005).

Κεφάλαιο 3 - Τεχνολογίες Πληροφορίας και Επικοινωνιών στην Εκπαίδευση

3.1 Εισαγωγή

Το σύνολο των επαγγελματικών χώρων οι οποίοι σχετίζονται με τη μελέτη, ανάπτυξη, σχεδίαση, υλοποίηση, συντήρηση και διαχείριση υπολογιστικών πληροφοριακών συστημάτων ονομάζεται τεχνολογία πληροφοριών ή αλλιώς (ΤΠΕ), και αφορά κυρίως εφαρμογές λογισμικού και υλικού υπολογιστών. Τα επαγγέλματα ΤΠΕ βασίζονται στην ανάπτυξη, εγκατάσταση και συντήρηση προϊόντων πληροφορικής και τηλεπικοινωνιών, με στόχο την παραγωγή, αποθήκευση, διαχείριση και μετάδοση πληροφοριών κάθε τύπου. Στις ΤΠΕ συγκαταλέγεται και η βιομηχανία ανάπτυξης λογισμικού, ως διακριτό υποσύνολο. Υπό τον όρο αυτό μπορούν επίσης να υπάγονται και τμήματα τεχνικής υποστήριξης σε οργανισμούς και επιχειρήσεις, αλλά και έργα που αφορούν προϊόντα πληροφορικής και τηλεπικοινωνιών είτε αυτά είναι δημόσια είτε είναι ιδιωτικά.

Ως ΤΠΕ ορίζουμε εν ολίγοις οτιδήποτε αφορά τη συγκέντρωση και ηλεκτρονική κωδικοποίηση, ταξινόμηση, επεξεργασία, επιλεκτική και συνδυαστική ανάσυρση, διακίνηση και διάχυση, γνωστοποίηση και μελέτη της όποιας πληροφορίας σε κάθε μορφή (κείμενο, αριθμοί, ήχος, εικόνα. Βίντεο, γράφημα κτλ.), είτε αυτό γίνεται μέσω κάποιας μεθόδου είτε μέσω της χρήσης εφαρμογών προϊόντων της τεχνολογίας. Η Ευρωπαϊκή Ένωση από τις αρχές το 1990 αναγνωρίζει την “αναγκαιότητα” εισαγωγής και ενσωμάτωσης των ΤΠΕ στην εκπαίδευση και θέτει σε εφαρμογή προγράμματα και δράσεις, που υποστηρίζουν και προωθούν τέτοια θέματα όπως το Socrates, Grunding. Η ΕΕ προσπαθεί να καινοτομήσει στην ποιοτική αλλαγή των δομών της εκπαίδευσης, στη διευκόλυνση της πρόσβασης στην εκπαίδευση και στην εξειδίκευση και επέκταση των τρόπων μάθησης.

Βασικός σκοπός είναι να καταστεί δυνατή η πρόσβαση στα συστήματα κατάρτισης και εκπαίδευσης για όλους τους πολίτες και κατ' επέκταση η δυνατότητα μετακίνησης από την συστηματική μάθηση στο πλαίσιο της δια βίου μάθησης. Σε μία προσπάθεια απλοποίησης και ενοποίησης ενός ιδιαίτερου χώρου εκπαίδευσης-κατάρτισης-επιμόρφωσης που προϋποθέτει και την υπέρβαση εμποδίων, όπως εθνικές, γλωσσικές, πολιτισμικές διαφορές, ώστε να

συνδεθούν με την επονομαζόμενη Κοινωνία της Πληροφορίας, συνδυάζοντας την αγορά και την κοινωνική παιδεία όλης της Ευρωπαϊκής Ένωσης.

3.2 Εκπαίδευση και επαγγελματικά ζητήματα

Από το 1993 κι έπειτα στη δευτεροβάθμια εκπαίδευση, διδάσκονται καθ' όλη τη διάρκεια του Γυμνασίου βασικές έννοιες της πληροφορικής σε βασικό επίπεδο σε συνδυασμό με την εκμάθηση απλών δεξιοτήτων χρήσης ηλεκτρονικών υπολογιστών και ΤΠΕ. Στο Γενικό Λύκειο διδάσκεται επιπρόσθετα το περισσότερο επιστημονικά προσανατολισμένο και πανελλαδικά εξεταζόμενο μάθημα «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον» σε μαθητές αντίστοιχης κατεύθυνσης, εισάγοντάς τους σε βασικές έννοιες σχεδίασης και ανάλυσης αλγορίθμων και προγραμματισμού υπολογιστών. Στα τεχνικά λύκεια λειτουργεί κύκλος μαθημάτων πληροφορικής, για μαθητές οι οποίοι τον επιλέγουν, ο οποίος εμπεριέχει μία αναλυτικότερη συγκριτικά με το Γυμνάσιο εισαγωγή σε κάποιες βασικές έννοιες της πληροφορικής από κοινού με τεχνικές γνώσεις περί ΤΠΕ, που αποσκοπούν στην κατάρτιση μελλοντικών τεχνικών και επαγγελματιών για βοηθητικές θέσεις εργασίας.

Στο τομέα της τριτοβάθμιας εκπαίδευσης, παρόμοιο πρόγραμμα σπουδών έχουν και αρκετά πανεπιστημιακά τμήματα (Επιστήμης Υπολογιστών, Πληροφορικής, Μηχανικών Πληροφορικής κλπ.), μπορεί όμως να εντάσσονται σε διαφορετικές σχολές συνήθως Θετικών και Οικονομικών Σπουδών ή Πολυτεχνεία. Ορισμένα από τα ανωτέρω τμήματα έχουν περισσότερο διακλαδικό χαρακτήρα, όπως π.χ. τα τμήματα «Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής» τα οποία εστιάζουν στην τομή πληροφορικής και ηλεκτρονικής μηχανικής, τα τμήματα «Πληροφορικής και Τηλεπικοινωνιών» ή «Μηχανικών Πληροφορικής και Τηλεπικοινωνιών» τα οποία εστιάζουν στην τομή πληροφορικής και τηλεπικοινωνιών κοκ. Τα τμήματα σε σχολές Οικονομικών συνήθως εστιάζουν κυρίως στα πληροφοριακά συστήματα, επικεντρώνοντας στην ανάπτυξη και εφαρμογή προϊόντων πληροφορικής για την αντιμετώπιση προβλημάτων και αναγκών σε επιχειρηματικό επίπεδο, βασιζόμενα σε ένα αμάλγαμα μαθημάτων οικονομικής και πληροφορικής.

Στην Ελλάδα τα πρώτα πανεπιστημιακά τμήματα Πληροφορικής εμφανιστήκαν το 1980 με την ίδρυση του Τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών και, στη συνέχεια, το 1984, με το Τμήμα Επιστήμης Υπολογιστών της Σχολής Θετικών Επιστημών του Πανεπιστημίου Κρήτης.

Τμήματα με παρόμοιους ή πανομοιότυπους τίτλους και κυρίως πρακτικό / τεχνολογικό προσανατολισμό υπάρχουν και σε ΑΤΕΙ, σε Σχολές Τεχνολογικών Εφαρμογών τετραετούς φοίτησης, ενώ μία λίστα όλων των ανάλογων τμημάτων σε ΑΕΙ και ΤΕΙ μπορεί να βρεθεί στην κατάλληλη ενότητα του άρθρου κατάλογος ελληνικών τμημάτων ανώτατης εκπαίδευσης (γνωστικό πεδίο: Πληροφορική). Η πλειοψηφία από τα παραπάνω πανεπιστημιακά τμήματα προσφέρουν και μεταπτυχιακό πρόγραμμα σπουδών, οδηγώντας στην απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης (Master of Science) ακόμη και Διδακτορικού Διπλώματος (PhD).

Σύμφωνα με την ισχύουσα νομοθεσία στην Ελλάδα, οι απόφοιτοι των ΑΕΙ με τίτλο «Πληροφορικής» ή παρόμοιο, αναγνωρίζεται πως έχουν την ικανότητα να ασχοληθούν ενδεικτικά με δραστηριότητες όπως :

- μελέτη
- σχεδίαση
- ανάλυση
- υλοποίηση
- εγκατάσταση
- επίβλεψη
- λειτουργία
- αξιολόγηση

αλλά και διενέργεια πραγματογνωμοσύνης και πιστοποίησης στους παρακάτω επιστημονικούς τομείς:

- της πληροφορικής
- του υλικού και λογισμικού των ηλεκτρονικών υπολογιστών
- των συστημάτων και εφαρμογών γραφικών, επεξεργασίας σημάτων, επεξεργασίας εικόνας και επεξεργασίας ομιλίας
- των συστημάτων και δικτύων επικοινωνιών, τηλεπικοινωνιακών υπηρεσιών και εφαρμογών Διαδικτύου.

Επιπρόσθετα, εκ των ανωτέρω απόφοιτων οι διπλωματούχοι (απόφοιτοι τμημάτων πενταετούς φοίτησης) καθώς και οι πτυχιούχοι των (τετραετούς φοίτησης) Τμημάτων «Επιστήμης και Τεχνολογίας Τηλεπικοινωνιών» και «Πληροφορικής και Τηλεπικοινωνιών», μπορούν επίσης να εργασθούν και στον επιστημονικό τομέα των τηλεπικοινωνιακών συστημάτων και δικτύων,

με βάση τον ίδιο κρατικό νόμο. Ωστόσο, σύμφωνα με παλαιότερο νόμο, όλοι οι πληροφορικοί και όλοι οι ηλεκτρονικοί μηχανικοί, απόφοιτοι είτε ΑΕΙ είτε ΤΕΙ, μπορούν με τον νομό να στελεχώσουν ως τεχνικοί διάφορους φορείς τηλεπικοινωνιών.

Τη δυνατότητα επίσης να απασχοληθούν επαγγελματικά σε ιδιωτικά ή δημόσια ερευνητικά ιδρύματα έχουν όλοι οι ανωτέρω απόφοιτοι, καθώς και στη διδασκαλία σε δευτεροβάθμια ή τριτοβάθμια εκπαιδευτικά ιδρύματα, καθώς και στην παροχή υπηρεσιών σε μονάδες πληροφορικής, δικτύων υπολογιστών, μηχανοργάνωσης και τεχνικών υπηρεσιών στον δημόσιο και στον ιδιωτικό τομέα. Η απόκτηση πιστοποίησης παιδαγωγικής επάρκειας είναι ωστόσο απαραίτητη για την διδασκαλία στη Δευτεροβάθμια Εκπαίδευση.

Οτιδήποτε έχει ειπωθεί έως εδώ ισχύει κατά γράμμα και για τους διπλωματούχους ηλεκτρολόγους ή ηλεκτρονικούς μηχανικούς, καθώς η πληροφορική και η επιστήμη ηλεκτρονικού μηχανικού έχουν καλυφθεί σε σημαντικό βαθμό ενώ οι τηλεπικοινωνίες είναι σημαντικό πεδίο μελέτης για την τελευταία. Το κύριο σημείο της εν λόγω επικάλυψης, όχι όμως το μοναδικό, είναι ο κλάδος της μηχανικής υπολογιστών, εφόσον ένας ηλεκτρονικός υπολογιστής μπορεί να ιδωθεί τόσο ως ηλεκτρονικό όσο και ως λογικό κύκλωμα. Ως εκ τούτου, όλα τα πανεπιστημιακά τμήματα (Ηλεκτρολόγων Μηχανικών και Ηλεκτρονικών Μηχανικών) στην Ελλάδα έχουν αφομοιώσει και τη μηχανική υπολογιστών στο πρόγραμμα σπουδών άλλα και στον τίτλο των σχολών.

Από τα ΤΕΙ οι απόφοιτοι με τίτλο «Πληροφορικής», καθώς κατέχουν τίτλο (πτυχιούχων) μηχανικών πληροφορικής, σύμφωνα με τον νόμο έχουν τη δυνατότητα να απασχοληθούν επαγγελματικά στον δημόσιο ή στον ιδιωτικό τομέα, αυτοδύναμα ή σε συνεργασία με άλλους επιστήμονες υπολογιστών, ως προγραμματιστές ή αναλυτές εφαρμογών λογισμικού ή υπολογιστικών συστημάτων. Μπορούν επίσης να απασχοληθούν ως ερευνητές ή ως εκπαιδευτικοί στα εν λόγω πεδία. Ακόμα, οι απόφοιτοι των ΤΕΙ με τίτλους όπως

- «Βιομηχανικής Πληροφορικής»,
- «Ηλεκτρονικών Υπολογιστικών Συστημάτων»

και παρόμοιων σχολών, έχουν τη δυνατότητα να απασχοληθούν επαγγελματικά, στον δημόσιο ή στον ιδιωτικό τομέα, αυτοδύναμα ή σε συνεργασία με άλλους επιστήμονες, σε τομείς όπως

- της σχεδίασης, συντήρησης
- εγκατάστασης, διαχείρισης και υλοποίησης

- υπολογιστικών ή τηλεπικοινωνιακών δικτύων
- της σχεδίασης, ανάπτυξης και προγραμματισμού λογισμικού
- καθώς και της σχεδίασης, ανάπτυξης, συντήρησης και εγκατάστασης υπολογιστικού υλικού ή συσκευών

Από τότε που δόθηκε η δυνατότητα στα ΑΤΕΙ, το 2013, να έχουν κατευθύνσεις σπουδών στα τμήματα τους, πλέον οι πτυχιούχοι μηχανικοί πληροφορικής εξειδικεύονται μέσω κάποιας κατεύθυνσης σε διάφορους τομείς (μηχανικοί δικτύων, μηχανικοί υπολογιστών, μηχανικοί λογισμικού).

Στην Ελλάδα δεν υπάρχει κάποιο αυστηρό νομοθετικό πλαίσιο ρύθμισης των τεχνολογιών πληροφοριών και επικοινωνίας εκτός φυσικά από τους εν λόγω κανονισμούς, ούτε και κάποιο επαγγελματικό επιμελητήριο ειδικά για τους επαγγελματίες του κλάδου. Ισχύει μεν ότι οι απόφοιτοι πανεπιστημιακών τμημάτων πληροφορικής από Οικονομικές Σχολές μπορούν να εγγραφούν στο Οικονομικό Επιμελητήριο ως οικονομολόγοι και οι απόφοιτοι τμημάτων πληροφορικής από Πολυτεχνεία στο Τεχνικό Επιμελητήριο Ελλάδος ως μηχανικοί, όμως οι εν λόγω ρυθμίσεις δεν σχετίζονται με την όποια επαγγελματική τους ιδιότητα περί ΤΠΕ. Υπάρχουν όμως παρ' όλα αυτά διάφορες επιστημονικές ενώσεις όπως

- Ένωση Πληροφορικών Ελλάδας
- Ένωση Μηχανικών Πληροφορικής και Επικοινωνιών Ελλάδας
- Σύλλογος Τεχνολογικής Εκπαίδευσης Μηχανικών Πληροφορικής.

3.3 ΤΠΕ στην εκπαίδευση

Εξαιτίας της ραγδαίας εξέλιξης της τεχνολογίας κρίθηκε απαραίτητη η εισαγωγή και η ενσωμάτωση των ΤΠΕ στο εκπαιδευτικό σύστημα. Στην ουσία γίνεται ένας ανασχηματισμός των εκπαιδευτικών συστημάτων, καθώς η χρήση του υπολογιστή και των ΤΠΕ επιφέρει μία σειρά από αλλαγές στα αναλυτικά προγράμματα, στον τρόπο διδασκαλίας και στο ρόλο του εκπαιδευτικού στην τάξη. Οι παρουσιάσεις με την χρήση πολυμέσων ελκύουν και διατηρούν το ενδιαφέρον του εκπαιδευομένου κυρίως λόγω των πολλαπλών αναπαραστάσεων της πληροφορίας που παρέχουν, με αποτέλεσμα να συμμετέχει το άτομο στην διαδικασία της μάθησης χρησιμοποιώντας πολλές από τις αισθήσεις του.

Τα γραφικά και γενικά οι εικόνες έχουν την δυνατότητα να μεταδίδουν μηνύματα και αποτελούν τα καλύτερα μέσα παρουσίασης κατά την διάρκεια της διδασκαλίας. Το βίντεο αποτελεί πολύ σημαντικό παράγοντα στη δυναμική των πολυμέσων. Σύμφωνα με εμπειρικές μελέτες, η χρήση λογισμικού παρουσιάσεων κατά την διδακτική διαδικασία συγκριτικά πάντα με τον συμβατικό τρόπο διδασκαλίας ενισχύει την αποτελεσματικότητα των μαθητών και τους κινητοποιεί περισσότερο στο να μάθουν.

Τα οφέλη μιας νέας τεχνολογίας είναι πολλά. Αφενός καταφέρνει να τραβήξει την προσοχή των παιδιών και να κάνει την εκπαίδευση πιο ενδιαφέρουσα μετατρέποντας την σε παιχνίδι. Αυξάνει την αποδοτικότητα και είναι μια ανεξάντλητη πηγή γνώσης. Είναι εύκολη στην χρήση και διευκολύνει πολύ την ζωή όλων μας, ενώ ταυτόχρονα ανοίγει νέους ορίζοντες. Μπορεί να απευθύνεται σε όλους και ο καθένας μπορεί να βρει οτιδήποτε τον ενδιαφέρει αμέσως είτε για να καλύψει διδακτικά κενά είτε για να καθοδηγηθεί καλύτερα. Εκμηδενίζει τις αποστάσεις και φέρνει σε επαφή ανθρώπους που βρίσκονται χιλιόμετρα μακριά ενώ η έλλειψη χρονικών περιορισμών κάνουν την ασύγχρονη επικοινωνία (e-mail, World Wide Web κ.τ.λ.) ευκολότερη.

Η ανάγκη που έχουν τα παιδιά για την εκμάθηση της διαμόρφωσης και διαχείρισης διαπροσωπικών σχέσεων είναι το ίδιο, ίσως και πιο σημαντική, από την ανάγκη για μόρφωση. Με την νέα τεχνολογία η επικοινωνία έχει αλλάξει σελίδα. Μέσω του Διαδικτύου (Internet) μπορούν οι μαθητές να μην περιορίζουν τις συναναστροφές τους σε άτομα της συνοικίας ή της πόλης τους, αλλά μπορούν να συνομιλούν με άτομα που βρίσκονται χιλιάδες χιλιόμετρα μακριά, σε οποιοδήποτε άλλο έθνος. Οι αποστάσεις μεταξύ των ανθρώπων μικραίνουν αισθητά καθώς το μέγεθος του κόσμου φαντάζει μικρότερο και άνθρωποι από όλες τις γωνίες του κόσμου έχουν την δυνατότητα να γνωριστούν και να μάθουν ο ένας από τον άλλον.

Το όλο εγχείρημα είναι ωφέλιμο και για τους εκπαιδευτικούς που ξεφεύγουν από έναν απομονωμένο «κόσμο». Πλησιάζουν τους μαθητές και γίνονται πιο ενεργητικοί και η βοήθεια τους πιο ουσιαστική, αφού αφήνουν τις διαλέξεις και περνάνε στην πράξη, καθοδηγώντας και βοηθώντας τους μαθητές χωρίς να τους δίνουν έτοιμες λύσεις. Πολλοί νέοι εκπαιδευτικοί ίσως ταυτιστούν περισσότερο με τον νέο αυτό ρόλο τους και ως αποτέλεσμα να αντλήσουν μεγαλύτερη ευχαρίστηση από το επάγγελμά τους.

Κεφάλαιο 4 - Τεχνολογίες WEB

4.1 HTML

Η HTML (αρχικοποίηση του αγγλικού HyperText Markup Language, ελλ. Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

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

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της

HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990. Τον ίδιο χρόνο, ο Μπέρνερς Λι και ο μηχανικός συστημάτων πληροφορικής του CERN Robert Cailliau συνεργάστηκαν σε μια κοινή προσπάθεια εύρεσης χρηματοδότησης, αλλά το έργο δεν υιοθετήθηκε ποτέ επίσημα από το CERN. Στις προσωπικές του σημειώσεις από το 1990, ο Μπέρνερς Λι αριθμεί «μερικές από τις πολλές χρήσεις του υπερκειμένου», και αναφέρει πρώτα από όλες μια εγκυκλοπαίδεια.

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο με το όνομα Ετικέτες HTML, το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991. Περιέγραφε τα 20 στοιχεία τα οποία αποτελούσαν τον αρχικό και σχετικά απλό σχεδιασμό της HTML. Εκτός από την ετικέτα υπερσυνδέσμου, οι υπόλοιπες ήταν έντονα επηρεασμένες από την SGMLguid, μια μορφή δημιουργίας τεκμηρίωσης, φτιαγμένη στο CERN και βασισμένη στην SGML. Δεκατρία από εκείνα τα αρχικά στοιχεία υπάρχουν ακόμα σήμερα στην HTML 4.

Το ίδιο το πρότυπο SGML αναπαράγει μερικές από τις τεχνικές των τυπογράφων, αλλά εκτός από απλή μίμηση της τυπογραφίας προσθέτει γενικευμένη σήμανση βασισμένη σε στοιχεία, τα οποία μπορούν να εμφωλεύονται το ένα μέσα στο άλλο και να φέρουν ιδιότητες. Ακόμα, το SGML διαχωρίζει τη δομή από το περιεχόμενο, κατεύθυνση προς την οποία αργότερα κινήθηκε και η HTML, με τα CSS. Πολλά από τα στοιχεία κειμένου προέρχονται από την τεχνική αναφορά ISO TR 9537, Techniques for using SGML (τεχνικές χρήσης της SGML), η οποία με τη σειρά της καλύπτει τα χαρακτηριστικά των πρώιμων γλωσσών μορφοποίησης κειμένου που χρησιμοποιούνταν από τα TYPSET και RUNOFF, και είχαν αναπτυχθεί στις αρχές της δεκαετίας του 1960 για το λειτουργικό σύστημα CTSS.

Ο Μπέρνερς Λι θεώρησε την HTML ως μια υλοποίηση του SGML. Αυτό ορίστηκε και επίσημα από το Internet Engineering Task Force (IETF) με τη δημοσίευση της πρώτης πρότασης για μια προδιαγραφή της HTML, στα μέσα του 1993, η οποία περιλάμβανε και έναν Ορισμό τύπου εγγράφου (DTD, Document Type Definition) της SGML, ο οποίος όριζε την γραμματική. Αυτό το πρόχειρο έληξε μετά την πάροδο έξι μηνών, αλλά περιέχει κάτι αξιοσημείωτο: την αναγνώριση της ετικέτας του NCSA Mosaic για την ενσωμάτωση εικόνων μέσα στο κείμενο, η οποία αντικατοπτρίζει την φιλοσοφία του IETF για ενσωμάτωση επιτυχημένων πρωτότυπων μέσα στα πρότυπα. Κάτι παρόμοιο περιείχε και το ανταγωνιστικό πρόχειρο του Dave Raggett, «HTML+ (Hypertext Markup Format)», από τα τέλη του, το οποίο πρότεινε την προτυποποίηση μερικών ήδη υλοποιημένων δυνατοτήτων, όπως οι πίνακες και οι φόρμες.

Μετά που τα πρόχειρα HTML και HTML+ έληξαν, στις αρχές του 1994, το IETF δημιούργησε την Ομάδα Εργασίας για την HTML, η οποία το 1995 ολοκλήρωσε την «HTML 2.0», με την πρόθεση να αποτελέσει την πρώτη προδιαγραφή πάνω στην οποία θα βασίζονταν οι μελλοντικές υλοποιήσεις. Η HTML 2.0 δημοσιεύτηκε ως RFC 1866, και περιείχε ιδέες από τα πρόχειρα HTML και HTML+. Η αρίθμηση 2.0 σκόπευε απλά να ξεχωρίσει την νέα έκδοση από τα πρόχειρα που προηγήθηκαν. designation was intended to distinguish the new edition from previous drafts.

Η περαιτέρω ανάπτυξη κάτω από την επίβλεψη του IETF καθυστέρησε λόγω σύγκρουσης ενδιαφερόντων. Από το 1996 και μετά, οι προδιαγραφές της HTML τηρούνται, μαζί με ανάδραση από τους δημιουργούς λογισμικού, από το World Wide Web Consortium (W3C). Ωστόσο, το 2000 η HTML έγινε επίσης παγκόσμιο πρότυπο (ISO/IEC 15445:2000). Η τελευταία προδιαγραφή της HTML, η HTML 4.01 δημοσιεύτηκε από το W3C το 1999, και το 2001 δημοσιεύτηκαν επίσης και τα λάθη και οι παραλείψεις της (errata).

4.2 Εισαγωγή στη Javascript

4.2.1. Γλώσσα προγραμματισμού Java

Η Java είναι μια γλώσσα προγραμματισμού, παρόμοια κατά κάποιο τρόπο με τη C++, αλλά με την πρόθεση να είναι πιο αξιόπιστη για εφαρμογές Internet. Η Java είναι λοιπόν για να φτιάχνετε προγράμματα, αλλά χρειάζεστε και ένα λειτουργικό σύστημα για να τρέξει. Ο προγραμματιστής μπορεί να δημιουργήσει μία και μόνο έκδοση του προγράμματος, που να τρέχει μέσα από ένα μεταφραστή της Java, ένα πρόγραμμα όπως το HotJava ή ο Netscape Navigator. Υπάρχουν διαφορετικοί διερμηνευτές για διαφορετικούς υπολογιστές και λειτουργικά συστήματα (ο Netscape Navigator που τρέχει σε Macintosh, μια έκδοσή του που τρέχει σε Windows 3.1, μια για Windows 95 κ.λ.π.). Έτσι, ένα και μόνο πρόγραμμα Java μπορεί να τρέξει σε πολλά και διαφορετικά λειτουργικά συστήματα, εφόσον ο χρήστης έχει έναν διερμηνευτή που να τρέχει στο σύστημά του. Οι διερμηνευτές της Java δεν είναι αληθινά λειτουργικά συστήματα αν και βασίζονται λίγο-πολύ στις ίδιες αρχές. Ένα λειτουργικό σύστημα είναι η σύζευξη μεταξύ ενός προγράμματος και του υπολογιστή. Αντί για ένα πρόγραμμα που μιλάει κατευθείαν στα περιφερειακά του υπολογιστή, ο προγραμματιστής μπορεί απλά να γράψει ένα πρόγραμμα που μιλάει στο λειτουργικό σύστημα και να βάλει το λειτουργικό σύστημα στον κόπο να λειτουργήσει τα περιφερειακά. Ο διερμηνευτής της Java λειτουργεί με παρόμοιο τρόπο. Βρίσκεται ανάμεσα στο λειτουργικό σύστημα και το Java πρόγραμμα. Αντί να γράφει διαφορετικές εκδοχές του ίδιου προγράμματος, γράφει ένα και μόνο πρόγραμμα που επικοινωνεί με οποιονδήποτε διερμηνευτή της Java σε οποιοδήποτε σύστημα υπολογιστή. Ο διερμηνευτής μεσολαβεί ανάμεσα στο Java πρόγραμμα και το λειτουργικό σύστημα, μεταφράζοντας ό,τι λέει το ένα σε κάτι που μπορεί να καταλάβει το άλλο.

Πρόκειται για ριζοσπαστική τεχνολογία, υποστηρίζουν οι υπερασπιστές της Java, γιατί γκρεμίζει τους περιορισμούς που μπαίνουν στο λογισμικό εξαιτίας των λειτουργικών συστημάτων. Η Java προσφέρει μια πολυδύναμη γλώσσα προγραμματισμού με την οποία μπορούν να δημιουργηθούν καλομελετημένα προγράμματα που μπορούν να τρέξουν σε πολλούς διαφορετικούς υπολογιστές.

4.2.2. Από τη Java στη JavaScript

Ένα σύστημα σαν τη Java είναι χρήσιμο μόνο όταν είναι πανταχού παρόν ή βρίσκεται κοντά σε αυτό. Πρέπει να βρίσκεται παντού. Ένα κλασσικό παράδειγμα συστήματος που απέτυχε γιατί δεν ήταν έτσι, είναι το OS/2. Πολλοί χρήστες του OS/2 θα παραπονεθούν ότι το σύστημα αυτό της IBM είναι κατά πολύ ανώτερο από οποιαδήποτε έκδοση των Microsoft Windows. Μπορεί να συμβαίνει μπορεί και όχι. Ένα πράγμα όμως είναι βέβαιο, ότι σχετικά λίγοι χρήστες χρησιμοποιούν OS/2. Αυτό έχει ως συνέπεια να γράφονται πολύ λίγα προγράμματα για το OS/2.

Η Sun αντιλαμβάνεται αυτή τη μικρή ζωτική λεπτομέρεια και θέλει να βεβαιωθεί ότι η Java θα καταλήξει να είναι το λειτουργικό σύστημα για το Internet (το DOS του Internet). Αλλά υπάρχει ένα πρόβλημα: η Java είναι μια γλώσσα προγραμματισμού. Δεν είναι εύκολο να φτιάξεις μια εφαρμογή της Java, εκτός αν είσαι προγραμματιστής που καταλαβαίνει πως να γράψει στη Java. Η Sun λοιπόν σχεδιάζει να προσφέρει ειδικά εργαλεία για να διευκολύνει κάθε μη προγραμματιστή να φτιάξει προγράμματα Java. Δεν έχει γίνει ακόμα κάτι τέτοιο, αλλά η JavaScript αποτελεί το πρώτο βήμα στην απλοποίηση της Java. Παρόλο που η Netscape ξεκίνησε την ανάπτυξη της JavaScript αυτόνομα, γρήγορα συνεργάστηκαν με τη Sun, συμφωνώντας να κάνουν τη JavaScript ένα υποσύνολο της Java. Η JavaScript μοιάζει κατά κάποιο τρόπο με τη Java αλλά είναι πολύ πιο απλή στη χρήση. Δεν υπάρχει ανάγκη βοηθητικών εργαλείων, ούτε χρειάζεται μεταγλώττιση εφαρμογών της Java. Το μόνο που χρειάζεται είναι να εισάγετε ένα σενάριο στις ηλεκτρονικές σελίδες σας και όταν ένας browser που υποστηρίζει την JavaScript φτάνει στη σελίδα, διαβάζει το σενάριο και ακολουθεί τις οδηγίες.

4.2.3. Σύγκριση Java και JavaScript

Υπάρχουν κάποιες σημαντικές διαφορές ανάμεσα στην Java και την JavaScript. Παρακάτω συγκρίνουμε τα δύο αυτά συστήματα.

Java	JavaScript
Περίπλοκη στη χρήση	Σχετικά εύκολη στη χρήση
Θα χρειαστείτε το JDK (Java Developers Kit) που προσφέρεται δωρεάν από τη Sun Microsystems.	Δεν χρειάζεστε τίποτε άλλο από Πληροφορίες για το πως να γράψετε σενάρια .
Τα προγράμματα Μεταγλωττίζονται σε εκτελέσιμα έγγραφα.	Τα προγράμματα ενσωματώνονται στη σελίδα με τη μορφή σεναρίου.
Στην παρούσα κατάσταση του Internet οι εφαρμογές της Java τείνουν να γίνουν λίγο αργές.	Η JavaScript τρέχει πολύ γρήγορα.
Η Java έχει περισσότερες Δυνατότητες.	Η JavaScript είναι πιο κατάλληλη για σχετικά απλές εφαρμογές.
Η Java είναι προσανατολισμένη στο αντικείμενο.	Η JavaScript είναι βασισμένη στο αντικείμενο.
Η Java έχει αυστηρούς κανόνες γύρω από τη χρήση μεταβλητών.	Η JavaScript είναι πιο χαλαρή στη χρήση μεταβλητών.
Η Java χρησιμοποιεί στατική	Η JavaScript χρησιμοποιεί

<p>σύνδεση. Οι αναφορές στα αντικείμενα πρέπει να υπάρχουν όταν μεταγλωττίζεται το πρόγραμμα.</p>	<p>δυναμική σύνδεση. Οι αναφορές στα αντικείμενα ελέγχονται όταν το σενάριο εκτελείται.</p>
---	---

Αν θέλετε να δημιουργήσετε προγράμματα σε C ++ ή Visual Basic χρειάζεστε ένα περιβάλλον ανάπτυξης λογισμικού και έναν μεταγλωττιστή. Δεν υπάρχει όμως περιβάλλον ανάπτυξης λογισμικού για την JavaScript και θα χρειαστεί ποτέ να μεταγλωττίσετε τα σενάρια σας. Αυτό που θα χρειαστείτε είναι ένας συμβατός με την JavaScript browser. Δηλαδή ένας World Wide Web browser που μπορεί να διαβάσει, να μεταφράσει και να τρέξει τα σενάρια που δημιουργείτε. Αυτό ισχύει για Netscape Navigator 4.0 ή Internet Explorer 4.0 ή νεότερες εκδόσεις.

4.2.4. Πλεονεκτήματα JavaScript

Με τη JavaScript μπορείτε να φτιάξετε σενάρια που να εκτελούν αυτόματες εργασίες, π.χ. όταν μια σελίδα του Web ανοίγει ή κλείνει. Επίσης μπορείτε να κάνετε την JavaScript να εκτελεί ενέργειες ανταποκρινόμενη σε ένα συγκεκριμένο γεγονός. Για παράδειγμα όταν ο χρήστης επιλέγει ένα κουμπί ή ένα σύνδεσμο, όταν εστιάζει από ένα στοιχείο μιας φόρμας σε ένα άλλο στοιχείο της κ.ο.κ.

Οι ενέργειες αυτές μπορεί να είναι απλές. Τα σενάρια μπορεί να ανοίγουν νέα παράθυρα στον browser και να εμφανίζουν συγκεκριμένα HTML έγγραφα ή να παρουσιάζουν μια σελίδα επιλεγμένη από τον κατάλογο ιστορικού του browser. Μπορεί επίσης να είναι και περίπλοκες δηλαδή ένα σενάριο μπορεί να ελέγχει τα περιεχόμενα μιας φόρμας που θέλει να υποβάλει ένας χρήστης και στη συνέχεια να προειδοποιεί τον χρήστη αν τα δεδομένα είναι λάθος. Το σενάριο μπορεί να ψάξει για πληροφορίες σε μια μικρή βάση δεδομένων ή να κάνει πολύπλοκους υπολογισμούς οικονομικών στοιχείων.

Με την Java Script μπορεί ο χρήστης να κάνει τα εξής:

- Πολυμερή έγγραφα με πλαίσια
- Επαναφόρτωση μέρους του παραθύρου
- Δημιουργώντας έγγραφα με αλληλεπίδραση
- Περισσότερος έλεγχος στην αλληλεπίδραση με το χρήστη

- Έγγραφα με μνήμη
- Ζωντανά έγγραφα
- Μηνύματα που ολισθαίνουν
- Ρολόγια
- Χρονικός μηχανισμός αντίστροφης μέτρησης
- Έγγραφα με αυτόματη ενημέρωση

Πολυμερή έγγραφα με πλαίσια

Μπορείτε να δημιουργήσετε έγγραφα που διαιρούν το παράθυρο του φυλλομετρητή σε επιμέρους κομμάτια. Αυτά τα κομμάτια καλούνται πλαίσια και μεγάλο μέρος της δύναμης της Java Script προέρχεται από τη διαχείριση των συγκεκριμένων πλαισίων.

Τα πλαίσια δίνουν περισσότερο έλεγχο πάνω από την εμφάνιση του εγγράφου από ότι η συμβατική HTML και δίνουν τη δυνατότητα να κρατήσετε σταθερό στην οθόνη ένα μέρος του εγγράφου ενώ κάποια άλλα αλλάζουν. Για παράδειγμα στο ένα έγγραφο μπορείτε να τοποθετήσετε το λογότυπο της εταιρείας, πληροφορίες δικαιωμάτων δημιουργού (copyright) και σε ένα άλλο να περιγράφετε μερικές ιδιαίτερα ενδιαφέρουσες πληροφορίες σχετικά με την εταιρεία. Έτσι το πλαίσιο που περιλαμβάνει το λογότυπο θα παραμένει σταθερό και ορατό ενώ οι πληροφορίες στο άλλο πλαίσιο θα αλλάζουν.

Ο κώδικας Java Script μέσα στο έγγραφο ενός πλαισίου μπορεί να αδειάσει άλλο πλαίσιο ώστε να γράψει νέα HTML ή άλλο κώδικα Java Script μέσα σε άλλο πλαίσιο. Προτού την Java Script ήταν τρομερά πολύπλοκη η δημιουργία, κατά τη λειτουργία, μιας νέας σελίδας φτιαγμένης σύμφωνα με τις οδηγίες του χρήστη. Μια τυπική ιστοσελίδα υλοποιείται από ένα στοιχείο HTML που περιέχει ένα HEAD και ένα BODY στοιχείο. Ένα πλαίσιο εγγράφου συνήθως υλοποιείται από ένα HTML στοιχείο που περιέχει ένα HEAD και ένα FRAMESET στοιχείο. Τα στοιχεία FRAME περιλαμβάνονται μέσα σε FRAMESET στοιχεία και τα FRAMESET μπορούν να περιλαμβάνουν άλλα FRAMESET στοιχεία δίνοντας τη δυνατότητα να διαιρέσετε και να υποδιαιρέσετε το παράθυρο του φυλλομετρητή. Με αυτή την ιδιότητα πρέπει να είσαστε ιδιαίτερα προσεκτικοί γιατί μπορεί να υποδιαιρέσετε το παράθυρο του φυλλομετρητή έως το σημείο που καμία τιμή δεν είναι ορατή. Ένα έγγραφο πλαισίων απαιτεί πολλαπλά αυτόνομα έγγραφα.

Εδώ είναι η λίστα του HTML αρχείου για το έγγραφο πλαισίου. Ακολουθεί το αρχείο main.htm


```

<HTML>
<HEAD>
<TITLE>Sample Frame Document</TITLE>
</HEAD>
<FRAMESET ROWS="2?%,*">
<FRAME SRC="welcome.htm" NAME="Welcome" SCROLLING="auto"
MARGINWIDTH=1 MARGINHEIGHT=1 NORESIZE>
<FRAMESET COLS="2?%,8?%">
<FRAME SRC="dir.htm" NAME="Directory" SCROLLING="auto"
MARGINWIDTH=1 MARGINHEIGHT=1 NORESIZE>
<FRAME SRC="contents.htm" NAME="Contents" SCROLLING="auto"
MARGINWIDTH=1 MARGINHEIGHT=1 NORESIZE>
</FRAMESET>
</FRAMESET>
</HTML>

```

Το στοιχείο FRAMESET χωρίζει το παράθυρο σε γραμμές. Στην ανώτατη γραμμή δίνεται (σε ποσοστό) 20% της οθόνης και στην κατώτερη δίνεται το υπόλοιπο 80%. Η ανώτατη γραμμή θα περιλαμβάνει το έγγραφο welcome.htm. Η κατώτατη γραμμή θα περιλαμβάνει ένα στοιχείο FRAMESET που διαιρεί τη γραμμή σε δύο στήλες. Η αριστερή στήλη αποκτά (σε ποσοστό) 20% από το εύρος της γραμμής και περιλαμβάνει το dir.htm. Η δεξιά στήλη αποκτά το υπόλοιπο 80% από το εύρος της γραμμής και περιλαμβάνει το contains.htm. Η επόμενη κατάταξη λίστας είναι η welcome.htm. Εμφανίζει μια απλή γραμμή την "Welcome To My Frame Document". Η γραμμή είναι κεντρικά στοιχισμένη και το κείμενο είναι πράσινο σε μαύρο φόντο.

```

<HTML>
<HEAD>
<TITLE>Welcome</TITLE>
</HEAD>
<BODY BGCOLOR="?????" TEXT="??8???">
<P ALIGN=CENTER>Welcome to My Frame Document</P>
</BODY>
</HTML>

```

Ακολουθεί το dir.htm. Περιέχει μια λίστα θεμάτων με λίστες ένθετες σε άλλες λίστες. Όπως το welcome.htm χρησιμοποιεί πράσινα γράμματα σε μαύρο φόντο.

```
<HTML>
<HEAD>
<TITLE>Directory</TITLE>
</HEAD>
<BODY BGCOLOR="?????" TEXT="??8???">
<UL>
<LI>Multi-Part Documents
<UL>
<LI>Reload Part of the Window
<LI>Interactive Documents
</UL>
<LI>More Control over User Interaction
<LI>Documents with Memory
<LI>Live Documents
<UL>
<LI>Scrolling Messages
<LI>Clocks
<LI>Countdown Timers
<LI>Self-Updating Documents
</UL>
</UL>
</BODY>
</HTML>
```

Τέλος είναι το contents.htm. Εμφανίζει ένα από τα θέματα γι'αυτή τη σελίδα με πράσινα γράμματα σε μαύρο φόντο.

```
<HTML>
<HEAD>
<TITLE>Contents</TITLE >
</HEAD>
<BODY BGCOLOR="?????" TEXT="??8???">
```

<P>

JavaScript offers you <I>much</I> more expressive power than HTML alone. This chapter will touch on a few of the things you can do with JavaScript, and is by no means exhaustive.

</P>

</BODY>

</HTML>

Επαναφόρτωση μέρους του παραθύρου

Και τι μπορείτε να κάνετε από τη στιγμή που έχετε υποδιαιρέσει το παράθυρο του φυλλομετρητή; Μπορείτε να ενημερώσετε ένα πλαίσιο φορτώνοντας το με ένα νέο έγγραφο ενώ τα άλλα πλαίσια παραμένουν σταθερά. Στο έγγραφο που υλοποιείται από το αρχείο main.htm το ανώτατο πλαίσιο περιέχει πληροφορίες σχετικά με το συνολικό σκοπό της ιστοσελίδας της επιχείρησης. Το χαμηλότερο αριστερά πλαίσιο περιέχει ένα κατάλογο σελίδων που αναφέρονται στο σκοπό που εκφράστηκε ανώτατο πλαίσιο έγγραφο. Το χαμηλότερο δεξιά πλαίσιο περιέχει μια σελίδα που ο χρήστης έχει επιλέξει από τον κατάλογο στο χαμηλότερο αριστερά πλαίσιο. Το ανώτατο και το χαμηλότερο αριστερά πλαίσιο δεν αλλάζουν ποτέ - μένουν σταθερά- αλλά ο χρήστης μπορεί να αλλάξει σελίδες στο χαμηλότερο δεξιά πλαίσιο.

Δημιουργώντας έγγραφα με αλληλεπίδραση

Μπορείτε να γράψετε κώδικα JavaScript που θα επαναδιατυπώνει τα περιεχόμενα ενός πλαισίου σε ανταπόκριση των ενεργειών του χρήστη. Δεν μπορείτε να τροποποιήσετε τα περιεχόμενα ενός πλαισίου αλλά μπορείτε να ξαναγράψετε τα περιεχόμενα του πλαισίου από την αρχή. Ακολουθούν τρία έγγραφα πλαισίων που επιδεικνύουν αυτή την ικανότητα.

Το main2.htm απλά διαιρεί την οθόνη σε δύο γραμμές.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Self-Modifying Document</TITLE>
```

```
</HEAD>
```

```
<FRAMESET ROWS="6?%,*">
```

```
<FRAME SRC="input.htm" NAME="input" SCROLLING="auto"
```

```

MARGINWIDTH=1 MARGINHEIGHT=1 NORESIZE>
<FRAMESET ROWS="78%">
<FRAME SRC="output.htm" NAME="output" SCROLLING="auto"
MARGINWIDTH=1 MARGINHEIGHT=1 NORESIZE>
</FRAMESET>
</FRAMESET>
</HTML>

```

Αυτό είναι το input.htm :

```

<HTML>
<HEAD>
<TITLE>Input</TITLE>
<SCRIPT>
<!-- hide script
function sayHello(form)
{
parent.output.document.open();
parent.output.document.open();
var gt = unescape("%3E");
parent.output.document.write("<HTML" + gt);
parent.output.document.write("<BODY BGCOLOR='??4???' TEXT='FFFFFF'" + gt);
parent.output.document.write("Hello " + form.firstName.value + " " +
form.middleName.value + " " + form.lastName.value);
parent.output.document.write("</BODY" + gt);
parent.output.document.write("</HTML" + gt);
parent.output.document.close();
}
// -->
</SCRIPT>
</HEAD>
<BODY BGCOLOR="??4???" TEXT="FFFFFF">
<FORM NAME="form">
<INPUT TYPE=TEXT SIZE=2? NAME="firstName"> First Name

```

```

<BR>
<INPUT TYPE=TEXT SIZE=2? NAME="middleName"> Middle Name
<BR>
<INPUT TYPE=TEXT SIZE=2? NAME="lastName"> Last Name
<BR>
<INPUT TYPE=BUTTON VALUE="Say Hello"
ONCLICK="sayHello(this.form)">
</FORM>
</BODY>
</HTML>

```

Το input.htm δημιουργεί μια φόρμα μέσα στην οποία μπορείτε να γράψετε το μικρό όνομα το μεσαίο και το επίθετο. Αφού ολοκληρωθεί η διαδικασία εισαγωγής κλικάρετε πάνω στο κουμπί με την επιγραφή "Say Hello" και η συνάρτηση "Say Hello" της JavaScript καλείται με μια αναφορά για τη φόρμα σαν παράμετρο. Η συνάρτηση "Say Hello " ανοίγει το έγγραφο στο παραγόμενο στην έξοδο - αυτό που φαίνεται εξωτερικά-πλαίσιο δημιουργεί και εμφανίζει μια σελίδα σ'αυτό το πλαίσιο. Χρησιμοποιεί τη φόρμα για να πάρει τα ονόματα που έχουμε εισάγει και να τα ενσωματώσει μέσα στη νέα σελίδα. Η υλοποίηση της συγκεκριμένης εργασίας χωρίς την JavaScript θα απαιτούσε την σύνταξη ενός προγράμματος που θα έτρεχε σ'έναν εξυπηρετητή (server) μια αρκετά πιο δύσκολη εργασία από το τοποθετήσετε μαζί αυτά τα μικρά κομμάτια της JavaScript.

Ακολουθεί το input.htm :

```

<HTML>
<BODY BGCOLOR="??4???" TEXT="FFFFFF">
</BODY>
</HTML>

```

Αυτός ο κώδικας κάνει το φόντο του κάθε πλαισίου να ταιριάζει με τον φόντο άλλων πλαισίων.

Περισσότερος έλεγχος στην αλληλεπίδραση με το χρήστη

Η JavaScript αναγνωρίζει αρκετά γεγονότα (events) που ο χρήστης μπορεί να προξενήσει μέσα σ'ένα έγγραφο. Μπορείτε να δημιουργήσετε κώδικα JavaScript που θα αντιδρά σ'αυτά τα γεγονότα, παρέχοντας αλληλεπίδραση με το χρήστη. Μέσα σ'ένα στοιχείο FORM, υπάρχουν

τα στοιχεία SELECT, INPUT και TEXTAREA που δρουν ως πεδία εισαγωγής. Ο χρήστης μπορεί τυπικά να μετακινεί το δρομέα από το ένα πεδίο εισαγωγής στο άλλο χρησιμοποιώντας το πλήκτρο TAB ή κλικάροντας το ποντίκι σ'ένα πεδίο εισαγωγής. Όταν ο χρήστης μετακινεί το δρομέα από το ένα πεδίο εισαγωγής (πηγή) σ'ένα άλλο (προορισμός) το πεδίο του προορισμού αποκτά την εστίαση και ο χρήστης μπορεί να τροποποιήσει τα περιεχόμενα του συγκεκριμένου πεδίου. Την ίδια στιγμή το πεδίο εισαγωγής χάνει την εστίαση. Στην ορολογία του Netscape το πεδίο προορισμού υπέστη ένα γεγονός εστίασης και το πεδίο πηγής ένα γεγονός συσκότισης.

Ένα άλλο είδος γεγονότων παίρνει μέρος όταν ο χρήστης υπερτονίζει κείμενο σ'ένα πλαίσιο. Αυτό ονομάζεται γεγονός επιλογής. Ένα γεγονός αλλαγής παίρνει χώρα όταν κείμενο μέσα σ'ένα πεδίο αλλάζει και η εστίαση μεταφέρεται σ'ένα άλλο πεδίο. Μπορούμε να γράψουμε JavaScript δηλώσεις ή συναρτήσεις όταν οποιαδήποτε από τα γεγονότα συμβαίνουν - βρίσκονται σ'ένα πεδίο.

Ο παρακάτω κώδικας παρουσιάζει πώς αυτά τα γεγονότα και οι χειριστές γεγονότων δουλεύουν. Κάθε ένα από τα πεδία εισαγωγής έχει χειριστές γεγονότων που χειρίζονται εστίαση(focus), συσκότιση(blur), επιλογή(select), αλλαγή(change) ή click γεγονότα. Κάθε διαχειριστής γεγονότος καλεί τη αγγέλουσα συνάρτηση περνώντας το όνομα του πεδίου εισαγωγής και το όνομα του διαχειριστή. Η αγγέλουσα συνάρτηση εμφανίζει την πληροφορία στη περιοχή του παραθύρου κατάστασης.

```
<HTML>
<HEAD>
<TITLE>Focus, Blur, Select, Change, Click Test</TITLE>
<SCRIPT>
<!--
function announce(widgetName, eventType)
{
status = "The " + widgetName + " field just experienced a " +
eventType + " event";
return true;
}
//-->
</SCRIPT>
```

```

</HEAD>
<BODY>
<FORM ONSUBMIT="return false">
<INPUT TYPE=BUTTON NAME="button" VALUE="button"
ONCLICK="announce('button','click')">
<INPUT TYPE=CHECKBOX NAME="checkbox"
ONCLICK="announce('checkbox','click')">A
<INPUT TYPE=CHECKBOX NAME="checkbox"
ONCLICK="announce('checkbox','click')">B
<INPUT TYPE=CHECKBOX NAME="checkbox"
ONCLICK="announce('checkbox','click')">C
<INPUT TYPE=RADIO NAME='radio'
ONCLICK="announce('radio','click')">AM
<INPUT TYPE=RADIO NAME='radio'
ONCLICK="announce('radio','click')">FM
<BR>
<INPUT TYPE=RESET NAME="reset"
ONCLICK="announce('reset','click')">
<BR>
<INPUT TYPE=SUBMIT NAME="submit"
ONCLICK="announce('submit','click')">
<BR>
<INPUT TYPE=TEXT NAME="text" SIZE=1?
ONBLUR="announce('text','blur')"
ONCHANGE="announce('text','change')"
ONFOCUS="announce('text','focus')"
ONSELECT="announce('text','select')">
<BR>
<TEXTAREA NAME="textArea" COLS="2?" ROWS="2"
ONBLUR="announce('textarea','blur')"
ONCHANGE="announce('textarea','change')"
ONFOCUS="announce('textarea','focus')"
ONSELECT="announce('textarea','select')"
Sample Text

```

```

</TEXTAREA>
<BR>
<SELECT NAME="select"
ONBLUR="announce('select','blur')"
ONCHANGE="announce('select','change')"
ONFOCUS="announce('select','focus')">
<OPTION>Option 1
<OPTION>Option 2
<OPTION>Option 3
</SELECT>
</FORM>
</BODY>
</HTML>

```

Αναδυόμενα (Pop- up) παράθυρα που δημιουργήθηκαν χρησιμοποιώντας μια από τις alert, confirm ή prompt μεθόδους μπορούν να αλληλεπιδράσουν άσχημα όταν δημιουργούνται από ένα χειριστή γεγονόςτος εστίασης. Σαν παράδειγμα, ο παρακάτω κώδικας pops- up ένα alert παράθυρο όταν το πεδίο κειμένου λαμβάνει προσοχή.

```

<HTML>
<HEAD>
<TITLE>Endless Loop</TITLE>
</HEAD>
<BODY>
<FORM>
<INPUT TYPE=TEXT NAME="text" SIZE=1?
ONFOCUS="alert('Here we go again...')">
</FORM>
</BODY>
</HTML>

```

Αν το alert παράθυρο τύχει να εμφανισθεί πάνω από το πεδίο κειμένου τότε θα έχετε πρόβλημα. Παρακάτω βλέπετε ακριβώς τι θα συμβεί.

1. Το πεδίο κειμένου αποκτά την εστίαση.
2. Ο διαχειριστής κειμένου onFocus ξεκινά την εκτέλεση.
3. Το παράθυρο alert, αναδύεται αποκτώντας την εστίαση από το πεδίο κειμένου.
4. Ο χρήστης αποδεσμεύει το παράθυρο alert.
5. Το παράθυρο alert διαγράφεται. Η εστίαση επιστρέφει στο πεδίο κειμένου.
6. Ο διαχειριστής κειμένου onFocus ξεκινά την εκτέλεση.
7. Το παράθυρο alert, pops -up αποκτώντας την εστίαση ...

Για να αποφύγετε αυτού του είδους τους ατελείωτους βρόχους, τοποθετήστε μηνύματα σ'ένα πεδίο εισόδου κειμένου, στη περιοχή κατάστασης του παραθύρου, ή σ'ένα άλλο παράθυρο ή πλαίσιο καθ'ολοκληρεία. Μέσα σ'ένα στοιχείο FORM, ένα άλλο είδος γεγονότος, το γεγονός υποβολής (submit), συμβαίνει όταν ο χρήστης κλικάρει πάνω σ'ένα κουμπί υποβολής. Πριν την εμφάνιση της JavaScript η επιλογή ενός κουμπιού υποβολής έστειλε τη φόρμα δεδομένων σε μία διαδικασία CGI σ'ένα απομακρυσμένο εξυπηρετητή. Η διαδικασία CGI επεξεργαζόταν τα δεδομένα και έστειλε πίσω μια νέα σελίδα με δεδομένα. Με την JavaScript μπορείτε να γράψετε έναν διαχειριστή γεγονότων για το γεγονός υποβολής και να κάνετε αυτό που θέλετε. Τα περισσότερα από τα περιεχόμενα του πεδίου εισόδου είναι προσιτά στον δικό σας κώδικα JavaScript. Τα πεδία κωδικών - συνθημάτων αποτελούν μια εξαίρεση γιατί μπορείτε να τροποποιήσετε τα δεδομένα και να αποφασίσετε να μην τα διαγράψετε όλα. Μπορείτε να δημιουργήσετε ένα έγγραφο και να το εμφανίσετε σ'ένα άλλο πλαίσιο ή παράθυρο. Και ακόμη μπορείτε να το στείλετε σε μια διαδικασία CGI.

Έγγραφα με μνήμη

Με τη χρήση ενός ιδιαίτερου χαρακτηριστικού που καλείται cookie τα έγγραφα μπορούν να μοιράσουν πληροφορίες μεταξύ τους. Μπορούμε να γράψουμε κώδικα JavaScript για τη δημιουργία, τροποποίηση και διαγραφή των cookies. Τα cookies είναι μικρά αντικείμενα δεδομένων που ανήκουν στο σύστημα του χρήστη. Η δύναμή τους είναι ότι παραμένουν στη μνήμη. Όταν το έγγραφο που δημιούργησε ή τροποποίησε ένα cookie δεν είναι φορτωμένο, τα δεδομένα του cookie είναι ακόμη εκεί. Άλλα έγγραφα που γνωρίζουν για το cookie μπορούν να προσπελάσουν και να τροποποιήσουν τα δεδομένα του, ώστε αυτά να μοιραστούν μεταξύ των εγγράφων.

Μια δυνατή χρήση των cookies είναι σε on-line καταλόγους. Η αποθήκευση ενός εκτεταμένου καταλόγου προϊόντων ή θεμάτων δεν θα απαιτούσε την τοποθέτηση ολόκληρου του καταλόγου σ'ένα μόνο έγγραφο. Είναι καλύτερο να διασπάσετε τον κατάλογο σε εύχρηστα κομμάτια με

κάθε ξεχωριστό έγγραφο να συγκεντρώνεται σε μία συγκεκριμένη κατηγορία θέματος-εμπορεύματος.

Ζωντανά έγγραφα

Τα ζωντανά έγγραφα είναι ιστοσελίδες που αλλάζουν καθώς περνά ο χρόνος. Μπορείτε να δημιουργήσετε χρονομέτρες στον κώδικά σας. Όταν ο χρονομέτρης μετρά αντίστροφα μια δήλωση της JavaScript θα εκτελείτε. Μπορείτε να κάνετε αρκετά πράγματα με χρονομέτρες όπως μηνύματα που ολισθαίνουν στην οθόνη ή να φορτώσετε ένα έγγραφο όταν ο χρονομέτρης μετρά αντίστροφα.

Μηνύματα που ολισθαίνουν

Πιθανόν θα έχετε δει τα μικρά έξυπνα μηνύματα που ολισθαίνουν κατά μήκος της μπάρας κατάστασης του φυλλομετρητή. Το μήνυμα ξεκινά με αυθαίρετο αριθμό κενών (το παράδειγμα που ακολουθεί χρησιμοποιεί 200). Το μήνυμα, με τα καθοδηγητικά κενά, γράφονται στην μπάρα κατάστασης. Ένας χρονοδιακόπτης ξεκινά, when timed out, τη διαδικασία από την αρχή αλλά με ένα λιγότερο κενό από τη προηγούμενη επανάληψη. Όταν ο αριθμός των κενών, προτού το μήνυμα, γίνεται μηδενικός η στρατηγική αλλάζει. Αντί το μήνυμα να προσαρτάται σ'ένα αλφαριθμητικό κενών, ένα υποαλφαριθμητικό του μηνύματος εμφανίζεται. Με κάθε επανάληψη το σημείο εκκίνησης του υποαλφαριθμητικού μετακινείται ένα χαρακτήρα προς τα δεξιά κάνοντας το μήνυμα να εμφανίζεται πως μετακινείται προς τα αριστερά. Όταν το μήνυμα δεν φαίνεται πια, συνήθως ολόκληρος ο κύκλος ξεκινά από την αρχή. Ακολουθεί ένα παράδειγμα για το πώς θα δημιουργήσετε ένα μήνυμα που ολισθαίνει.

```
<HTML>
<HEAD>
<TITLE>Scrolling Message</TITLE>
<SCRIPT>
<!--
var winLength = 2??; // guess at how many spaces wide the status bar is
var speed = 1??; // number of milliseconds between updates
function scroll(count)
{
```

```

var msg = "Hi! This is my scrolling message in the status bar.";
var out = " ";
var cmd = "scroll(";
if (count <= winLength && ? < count)
{
var c = ?;
for (c = ? ; c < count ; c++)
{
out += " ";
}
out += msg;
}
else if (count <= ?)
{
if (-count < msg.length)
{
out += msg.substring(-count,msg.length);
}
else
{
count = winLength + 1;
}
}
window.status = out;
count--;
cmd += count + ")";
window.setTimeout(cmd,speed);
}
//-->
</SCRIPT>
</HEAD>
<BODY ONLOAD="window.setTimeout('scroll(winLength)',speed);">
</BODY>
</HTML>

```

Τα μηνύματα που ολισθαίνουν δεν είναι απαραίτητο να φαίνονται στην περιοχή status. Μπορείτε να δημιουργήσετε μια φόρμα και να τοποθετήσετε το μήνυμα σ'ένα πεδίο κειμένου μέσα στη φόρμα.

Ρολόγια

Τα ρολόγια αποτελούν μια παραλλαγή των μηνυμάτων που ολισθαίνουν. Απλά είναι σταθερά προσηλωμένα- εγκατεστημένα και λένε την ώρα όπως το ρολόι στην οθόνη του υπολογιστή. Η JavaScript αναγνωρίζει ημέρες και ώρες και η δημιουργία ενός απλού ρολογιού είναι μια αρκετά απλή διαδικασία όπως φαίνεται από τον παρακάτω κώδικα.

```
<HEAD>
<TITLE>Clock</TITLE>
<SCRIPT>
<!--
function updateTime()
{
var now = new Date();
var time = "" + now.getHours() + ":";
var minute = now.getMinutes();
if (minute < 1?)
{
time += "?";
}
time += minute + ":";
var second = now.getSeconds();
if (second < 1?)
{
time += "?";
}
time += second;
window.status = time;
window.setTimeout("updateTime()",1??);
}
//--
```

```

</SCRIPT>
</HEAD>
<BODY ONLOAD="window.setTimeout('updateTime()',1???)">
</BODY>
</HTML>

```

Αυτός ο κώδικας δέχεται την τρέχουσα ημερομηνία δημιουργώντας ένα νέο αντικείμενο Date. Αποσπά την τρέχουσα ώρα τα λεπτά και τα δευτερόλεπτα και τα εμφανίζει στην μπάρα κατάστασης. Στην περίπτωση των λεπτών και των δευτερολέπτων ελέγχει αν κάποια τιμή είναι μονό ψηφίο. Αν μια από αυτές τις τιμές είναι, προσθέτει το αρχικό μηδενικό (π.χ. μια ώρα δώδεκα και πέντε θα εμφανιζόταν ακανόνιστα - παράξενα ως "12:5:0"). Αφού εμφανισθεί η ώρα ένας νέος χρονοδιακόπτης δημιουργείται που θα τερματίσει ακριβώς 1,000 χιλιοστό του δευτερολέπτου. Όταν ο χρονοδιακόπτης τερματίζει εμφανίζει την ώρα και θέτει έναν άλλο χρονοδιακόπτη.

Χρονικός μηχανισμός αντίστροφης μέτρησης

Μια άλλη παραλλαγή στο θέμα του χρονοδιακόπτη είναι και ο χρονικός μηχανισμός αντίστροφης μέτρησης. Μπορείτε να χρησιμοποιήσετε έναν τέτοιο μετρητή ώστε να αφήσετε το χρήστη να γνωρίζει ότι κάτι πρόκειται να συμβεί και πότε. Είναι εύκολο να δημιουργήσετε χρονικούς μηχανισμούς αντίστροφης μέτρησης στην JavaScript όπως φαίνεται παρακάτω :

```

<HTML>
<HEAD>
<TITLE>Count Down</TITLE>
<SCRIPT>
<!--
function countdown(tick)
{
if (tick == ?)
{
window.status = "We have liftoff...";
return;
}
var time = "T minus ";

```

```

var minute = Math.floor(tick / 60);
if (minute < 10)
{
time += "0";
}
time += minute + ":";
var second = tick % 60;
if (second < 10)
{
time += "0";
}
time += second;
window.status = time;
--tick;
var command = "countDown(" + tick + ")";
window.setTimeout(command,1000);
}
//-->
</SCRIPT>
</HEAD>
<BODY ONLOAD="window.setTimeout('countDown(120)',1000);">
</BODY>
</HTML>

```

Εδώ ένας ακέραιος κρατά την ώρα σε δευτερόλεπτα. Η ώρα διαιρείται με 60 για να πάρουμε τα λεπτά (η `math.floor()` χρησιμοποιείται ώστε να είμαστε σίγουροι ότι το πηλίκο είναι ένας ακέραιος) και το ακέραιο υπόλοιπο χρησιμοποιείται για να πάρουμε τα δευτερόλεπτα. Τα λεπτά και τα δευτερόλεπτα εμφανίζονται στην μπάρα κατάστασης. Στο τέλος του κώδικα του χρονοδιακόπτη ένα νέο `TIME-OUT` δημιουργείτε που επαναλαμβάνει τον κώδικα μ'ένα λιγότερο δευτερόλεπτο, ένα δευτερόλεπτο αργότερα. Όταν τα δευτερόλεπτα μηδενίζονται, η μπάρα κατάστασης αλλάζει και δεν δημιουργούνται άλλα `time-outs`.

Έγγραφο με αυτόματη ενημέρωση

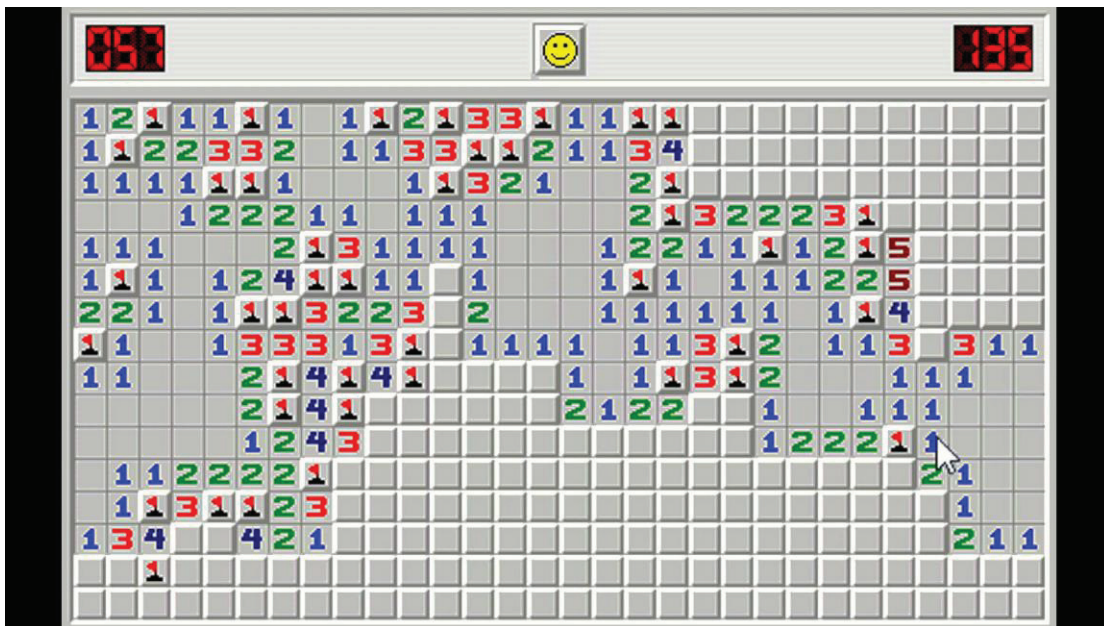
Ένα έγγραφο μπορεί να αυτοενημερωθεί. Για παράδειγμα κάθε πέντε λεπτά 'ένα χρηματιστηριακό - μεσιτικό γραφείο μπορεί να δημιουργήσει μια εικόνα GIF που θα εμφανίζει ένα γράφημα με την άνοδο και κάθοδο του Dow -Jones κατά τη διάρκεια ενός 5λεπτου διαστήματος. Μπορείτε να δημιουργήσετε ένα έγγραφο που περιέχει το GIF αρχείο σαν εσωτερική εικόνα , και το έγγραφο μπορεί να αυτοενημερώνεται κάθε πέντε λεπτά. Το έγγραφο καλεί την history.go() κάθε 300 δεύτερα (5 λεπτά). Το νέο κάλεσμα της history.go() δρα σαν τη χρήση του Reload.

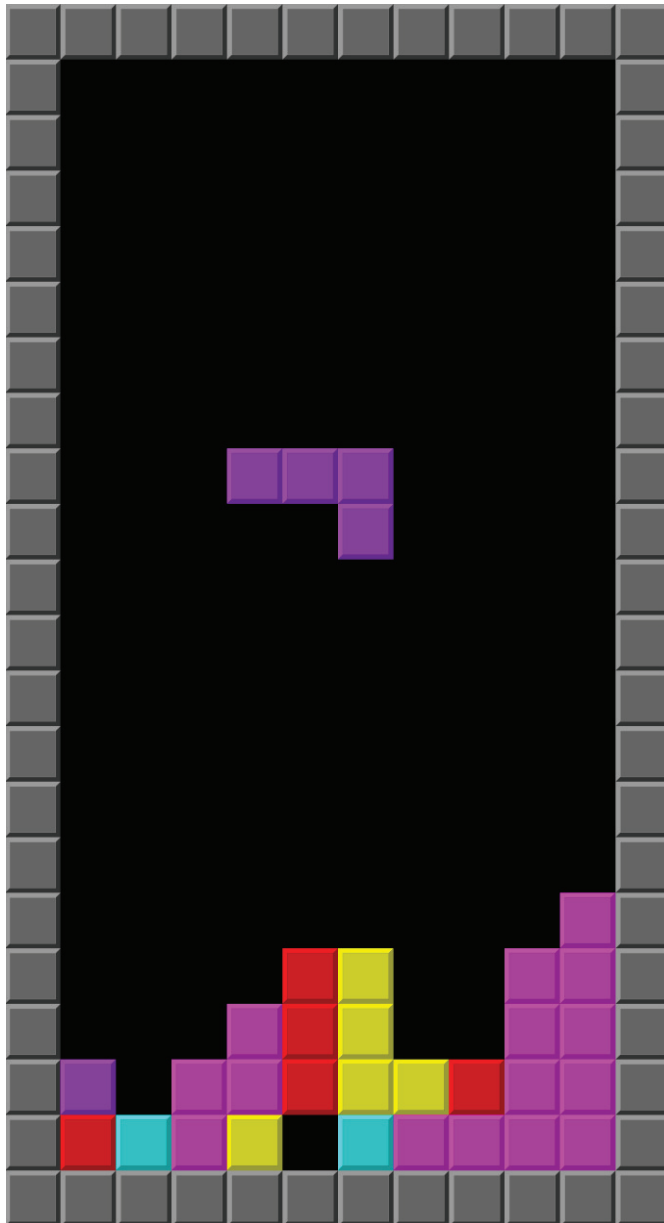
```
<HTML>
<HEAD>
<TITLE>The                Dow                Jones</TITLE>
</HEAD>
<BODY>
<IMG SRC="http://www.mythical_brokerage_house.com/DJ.GIF"
ALT="Picture it..." HEIGHT="5??" WIDTH="5??">
</BODY ONLOAD="window.setTimeout('history.go(?)',3?????)">
</HTML>
```

Κεφάλαιο 5 - Υλοποίηση παιχνιδιού

5.1 Εισαγωγή

Το παιχνίδι που υλοποιήσαμε ανήκει στην κατηγορία «Παιχνίδια γρίφων» (puzzle games). Αυτού του είδους τα παιχνίδια απαιτούν από τον παίκτη να λύσει διάφορους λογικούς γρίφους ή να πλοηγηθεί σε περίπλοκες περιοχές, όπως ένας λαβύρινθος. Αυτό το είδος παιχνιδιών διασταυρώνεται συχνά με παιχνίδια περιπέτειας και εκπαιδευτικά παιχνίδια. Δημοφιλέστερα αυτού του είδους είναι τα εξής: ο Ναρκαλιευτής (Minesweeper) και Tetris.





Το παιχνίδι μας είναι ένας συνδυασμός HTML και JAVASCRIPT. Η JavaScript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού ειδικά για υλοποίηση σε browsers ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Ουσιαστικά η γλώσσα JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Το συντακτικό της γλώσσας είναι παρόμοιο με της JAVA αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme.

Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

5.2 Κανόνες του παιχνιδιού

Η βασική εικόνα του παιχνιδιού είναι η παρακάτω:

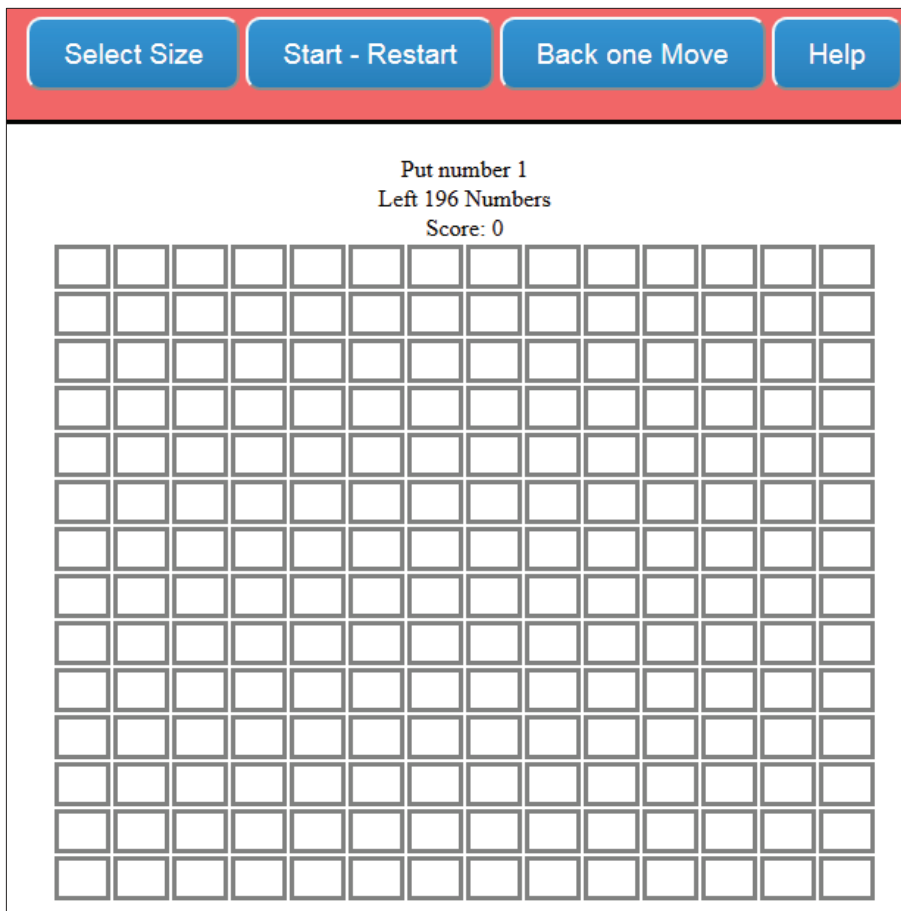


Select Size Start - Restart Back one Move Help

Put number 1
Left 64 Numbers
Score: 0

Select Size Start - Restart Back one Move Help

Put number 1
Left 100 Numbers
Score: 0



Στόχος είναι ο χρήστης να γεμίσει με 64 ή 100 ή 196 αριθμούς ένα πίνακα 8x8, 10x10, 14x14 που ουσιαστικά είναι ο πίνακας του παιχνιδιού. Όμως οι αριθμοί μπαίνουν με τους παρακάτω κανόνες.

- Αν ο χρήστης έχει τοποθετήσει ένα αριθμό σε ένα τετράγωνο ο επόμενος αριθμός πρέπει να απέχει 2 τετράγωνα οριζόντια ή κάθετα από το υπάρχον τετράγωνο ή 1 τετράγωνο διαγώνια.
- Έτσι ο παίκτης μπορεί να τοποθετήσει τον επόμενο αριθμό μόνο σε πράσινο τετράγωνο.
- Φυσικά δεν μπορεί να ξαναβάλει σε ένα κατειλημμένο τετράγωνο που έχει ήδη αριθμό άλλο αριθμό.
- Με βάση τους παραπάνω κανόνες στόχος είναι ο χρήστης να βάλει όσο γίνεται περισσότερα τετράγωνα στο πίνακα του παιχνιδιού.
- Όταν δεν υπάρχουν άλλες θέσεις τότε έχουμε GAME OVER.

Η παρακάτω εικόνα δείχνει τις επιτρεπτές θέσεις όταν έχουμε τοποθετήσει τον αριθμό 1 στο κέντρο του πίνακα του παιχνιδιού

Μια εικόνα ενός συμπληρωμένου παιχνιδιού με σχετικά μεγάλο σκορ είναι η παρακάτω:

Start - Restart Back Help

Put number 92
Left 9Numbers
Score: 91

1	43	22	2	44	23	3	45	24	4
32	55	13	33	54	14	34	61	15	35
21	68	77	72	69	78	73	70	79	46
12	42	53	88	75	62	89	86	25	5
31	56				71		60	16	36
20	67	76			87	74	63	80	47
11	41	52	91	84		90	85	26	6
30	57	19		58	18		59	17	37
51	66	83	50	65	82	49	64	81	48
10	40	29	9	39	28	8	38	27	7

Τέλος, όταν επιλέγει ο χρήστης ένα λάθος κουτάκι, τότε εμφανίζεται μήνυμα λάθους.

Start - Restart

Back

Help

Put number 2
Left 99Numbers
Score: 1

Error! Not Correct Place !

OK

5.3 Παράρτημα – Κώδικας Υλοποίησης

```
<html>
<head>
<title>
MyGame
</title>

<style>

body {
    margin: 0px;
}

.a1 {
width:40px; height: 40px;
border-style: solid;
text-align:center;
font-size:12pt;
}

#menu {
text-align: center;
border-bottom-style: solid;
background-color: #f66;
padding: 20px;
}

#help {
```



```
text-align: center;
border-style: solid;
background-color: #eee;
display:none;
position: absolute;
top: 50px;
left: 25%;

width:50%;
}

.btn {
background: #3498db;
background-image: -webkit-linear-gradient(top, #3498db, #2980b9);
background-image: -moz-linear-gradient(top, #3498db, #2980b9);
background-image: -ms-linear-gradient(top, #3498db, #2980b9);
background-image: -o-linear-gradient(top, #3498db, #2980b9);
background-image: linear-gradient(to bottom, #3498db, #2980b9);
-webkit-border-radius: 10;
-moz-border-radius: 10;
border-radius: 10px;
font-family: Arial;
color: #ffffff;
font-size: 19px;
padding: 10px 20px 10px 20px;
text-decoration: none;
}
```

```
.btn:hover {
  background: #fc3c83;
  background-image: -webkit-linear-gradient(top, #fc3c83, #d93434);
  background-image: -moz-linear-gradient(top, #fc3c83, #d93434);
  background-image: -ms-linear-gradient(top, #fc3c83, #d93434);
  background-image: -o-linear-gradient(top, #fc3c83, #d93434);
  background-image: linear-gradient(to bottom, #fc3c83, #d93434);
  text-decoration: none;
}

#main {
  padding: 20px;
  text-align: center;
}

</style>

<script>
var k=1;
var vback=0;
var movex=new Array(100);
var movey=new Array(100);

function back()
```

```
{
    if (k>1)
    {
        vback=1;
        i=movex[k-2];
        j=movey[k-2];

        document.getElementById("i"+i+j).innerHTML="";
        document.getElementById("i"+i+j).style.backgroundColor="#fff";
        k=k-2;
        i=movex[k-1];
        j=movey[k-1];
        document.getElementById("i"+i+j).innerHTML="";
        document.getElementById("i"+i+j).style.backgroundColor="#fff";
        if(k>1) func(i,j);
        else reset();

    }
    else
    {
        reset();
    }
}

function reset()
```

```

    {
        for (i=0;i<8;i++)
        for (j=0;j<8;j++)
        {
            document.getElementById("i"+i+j).innerHTML="";
            document.getElementById("i"+i+j).style.backgroundColor="#fff";
        }
        k=1;

        for (i=0;i<64;i++) {movex[i]=0; movey[i]=0;}
        document.getElementById("inf").innerHTML="Put number 1<br> Left 64
Numbers<br>Score: 0";

    }

function check(i,j)
{
    d=0;
    s=0;
    if(i-3>=0)
    {
        s++;
        if (document.getElementById("i"+(i-3)+j).innerHTML!="")
d++;
    }
    if(j-3>=0)
    {
        s++;
        if (document.getElementById("i"+i+(j-3)).innerHTML!="")
d++;
    }
}

```

```

        }
        if(i+3<=7)
        {
            s++;
            if (document.getElementById("i"+(i+3)+j).innerHTML!="")
d++;
        }
        if(j+3<=7)
        {
            s++;
            if (document.getElementById("i"+i+(j+3)).innerHTML!="")
d++;
        }
        if(i+2<=7 && j+2<=7 )
        {
            s++;
            if
(document.getElementById("i"+(i+2)+(j+2)).innerHTML!="") d++;
        }
        if(i+2<=7 && j-2>=0 )
        {
            s++;
            if (document.getElementById("i"+(i+2)+(j-
2)).innerHTML!="") d++;
        }
        if(i-2>=0 && j+2<=7 )
        {
            s++;
            if (document.getElementById("i"+(i-
2)+(j+2)).innerHTML!="") d++;
        }
    }

```

```

        if(i-2>=0 && j-2>=0 )
        {
            s++;
            if (document.getElementById("i"+(i-2)+(j-2)).innerHTML!="")
d++;
        }

        if(s==d ) alert('Game Over !!!');
    }

function clearcells()
{
    for (i=0;i<8;i++)
        for (j=0;j<8;j++)
        {
            document.getElementById("i"+i+j).style.backgroundColor="#fff";
            if (document.getElementById("i"+i+j).innerHTML!="")
            {
                document.getElementById("i"+i+j).style.backgroundColor="#eee";
            }
        }
}

function func(i,j)
{
    play=1;
    if(k>1 && vback==0)
    {

```

```

                if(document.getElementById("i"+i+j).style.backgroundColor=="rgb(0,
255, 0)")
                    play=1;
                else
                {
                    alert("Error! Not Correct Place !");
                    play=0;
                }
            }
            if(play==1)
            {
                vback=0;
                clearcells();
                if (document.getElementById("i"+i+j).innerHTML!="")
                {
                    alert("Error. You select an occupied box!");
                }
                else
                {
                    x=i; y=j;
                    document.getElementById("i"+i+j).innerHTML=k;
                    movex[k-1]=i;
                    movey[k-1]=j;
                    k++;

                document.getElementById("i"+i+j).style.backgroundColor="#eee";

                if(i-3>=0)
                {

```

```

        document.getElementById("i"+(i-
3)+j).style.backgroundColor="#0f0";
    }
    if(j-3>=0)
    {
        document.getElementById("i"+i+(j-
3)).style.backgroundColor="#0f0";
    }
    if(i+3<=7)
    {
        document.getElementById("i"+(i+3)+j).style.backgroundColor="#0f0";
    }
    if(j+3<=7)
    {
        document.getElementById("i"+i+(j+3)).style.backgroundColor="#0f0";
    }
    if(i+2<=7 && j+2<=7 )
    {
        document.getElementById("i"+(i+2)+(j+2)).style.backgroundColor="#0f0";
    }
    if(i+2<=7 && j-2>=0 )
    {
        document.getElementById("i"+(i+2)+(j-
2)).style.backgroundColor="#0f0";
    }
    if(i-2>=0 && j+2<=7 )
    {
        document.getElementById("i"+(i-
2)+(j+2)).style.backgroundColor="#0f0";
    }

```



```

        }
        if(i-2>=0 && j-2>=0 )
        {
            document.getElementById("i"+(i-2)+(j-
2)).style.backgroundColor="#0f0";
        }

        for (i=0;i<8;i++)
        for (j=0;j<8;j++)
        {
            if (document.getElementById("i"+i+j).innerHTML!="")
            {
                document.getElementById("i"+i+j).style.backgroundColor="#eee";
            }
        }
        check(x,y);
    }
}
document.getElementById("inf").innerHTML="Put number "+k+"<br> Left
"+(64-k+1)+"Numbers<br>Score: "+(k-1);
}

function help()
{
    document.getElementById('help').style.display='block';
}

```

```
</script>

</head>

<body>
<div id=menu>
<a href='select.html'><button class=btn onclick='reset();'>Select Size</button></a>
<button class=btn onclick='reset();'>Start - Restart</button>
<button class=btn onclick='back();'>Back one Move</button>
<button class=btn onclick='help();'>Help</button>
</div>

<div id=main>

<center>
<div id=inf>Put number 1<br> Left 64 Numbers<br>Score: 0</div>
<table id=tb>

<tr>
  <td class=a1 id=i00 onclick='func(0,0)'>
  </td>
  <td class=a1 id=i01 onclick='func(0,1)'>
  </td>
  <td class=a1 id=i02 onclick='func(0,2)'>
  </td>
  <td class=a1 id=i03 onclick='func(0,3)'>
  </td>

```

```
<td class=a1 id=i04 onclick='func(0,4)'>
</td>
<td class=a1 id=i05 onclick='func(0,5)'>
</td>
<td class=a1 id=i06 onclick='func(0,6)'>
</td>
<td class=a1 id=i07 onclick='func(0,7)'>
</td>
</tr>
<tr>
<td class=a1 id=i10 onclick='func(1,0)'>
</td>
<td class=a1 id=i11 onclick='func(1,1)'>
</td>
<td class=a1 id=i12 onclick='func(1,2)'>
</td>
<td class=a1 id=i13 onclick='func(1,3)'>
</td>
<td class=a1 id=i14 onclick='func(1,4)'>
</td>
<td class=a1 id=i15 onclick='func(1,5)'>
</td>
<td class=a1 id=i16 onclick='func(1,6)'>
</td>
<td class=a1 id=i17 onclick='func(1,7)'>
</td>
```

```
</tr>

<tr>
  <td class=a1 id=i20 onclick='func(2,0)'>
  </td>
  <td class=a1 id=i21 onclick='func(2,1)'>
  </td>
  <td class=a1 id=i22 onclick='func(2,2)'>
  </td>
  <td class=a1 id=i23 onclick='func(2,3)'>
  </td>
  <td class=a1 id=i24 onclick='func(2,4)'>
  </td>
  <td class=a1 id=i25 onclick='func(2,5)'>
  </td>
  <td class=a1 id=i26 onclick='func(2,6)'>
  </td>
  <td class=a1 id=i27 onclick='func(2,7)'>
  </td>
</tr>

<tr>
  <td class=a1 id=i30 onclick='func(3,0)'>
  </td>
  <td class=a1 id=i31 onclick='func(3,1)'>
  </td>
  <td class=a1 id=i32 onclick='func(3,2)'>
  </td>

```

```
<td class=a1 id=i33 onclick='func(3,3)'>
</td>
<td class=a1 id=i34 onclick='func(3,4)'>
</td>
<td class=a1 id=i35 onclick='func(3,5)'>
</td>
<td class=a1 id=i36 onclick='func(3,6)'>
</td>
<td class=a1 id=i37 onclick='func(3,7)'>
</td>
</tr>
<tr>
<td class=a1 id=i40 onclick='func(4,0)'>
</td>
<td class=a1 id=i41 onclick='func(4,1)'>
</td>
<td class=a1 id=i42 onclick='func(4,2)'>
</td>
<td class=a1 id=i43 onclick='func(4,3)'>
</td>
<td class=a1 id=i44 onclick='func(4,4)'>
</td>
<td class=a1 id=i45 onclick='func(4,5)'>
</td>
<td class=a1 id=i46 onclick='func(4,6)'>
</td>
<td class=a1 id=i47 onclick='func(4,7)'>
```

```
</td>

</tr>

<tr>
  <td class=a1 id=i50 onclick='func(5,0)'>
    </td>
  <td class=a1 id=i51 onclick='func(5,1)'>
    </td>
  <td class=a1 id=i52 onclick='func(5,2)'>
    </td>
  <td class=a1 id=i53 onclick='func(5,3)'>
    </td>
  <td class=a1 id=i54 onclick='func(5,4)'>
    </td>
  <td class=a1 id=i55 onclick='func(5,5)'>
    </td>
  <td class=a1 id=i56 onclick='func(5,6)'>
    </td>
  <td class=a1 id=i57 onclick='func(5,7)'>
    </td>
</tr>

<tr>
  <td class=a1 id=i60 onclick='func(6,0)'>
    </td>
  <td class=a1 id=i61 onclick='func(6,1)'>
    </td>
```

```
<td class=a1 id=i62 onclick='func(6,2)'>
</td>
<td class=a1 id=i63 onclick='func(6,3)'>
</td>
<td class=a1 id=i64 onclick='func(6,4)'>
</td>
<td class=a1 id=i65 onclick='func(6,5)'>
</td>
<td class=a1 id=i66 onclick='func(6,6)'>
</td>
<td class=a1 id=i67 onclick='func(6,7)'>
</td>
</tr>
<tr>
<td class=a1 id=i70 onclick='func(7,0)'>
</td>
<td class=a1 id=i71 onclick='func(7,1)'>
</td>
<td class=a1 id=i72 onclick='func(7,2)'>
</td>
<td class=a1 id=i73 onclick='func(7,3)'>
</td>
<td class=a1 id=i74 onclick='func(7,4)'>
</td>
<td class=a1 id=i75 onclick='func(7,5)'>
</td>
<td class=a1 id=i76 onclick='func(7,6)'>
```

```

        </td>
        <td class=a1 id=i77 onclick='func(7,7)'>
        </td>
</tr>

</table>
<script>
reset();
</script>
</center>
</div>

<div id='help'>
<h3>Instructions</h3>

<p>The aim of the game is to put 64 numbers on the board of the game. </p>
<p>Each number is placed by clicking on a cell in the table. </p>
<p>The next click is allowed in cells following the following rule:</p>
<p>Horizontally and vertically 2 cells away from the current cell</p>
<p>Diagonal 1 cell away from the current cell.</p>
<p>When there is not any moves then "Game Over"</p>
<p> The maximum score is 64S </p>
<br>
<button class='btn'
onclick="document.getElementById('help').style.display='none';">Close</button><br><br>

</div>

```



```
</body>
</html>

<html>
<head>
<title>
MyGame
</title>

<style>

body {
    margin: 0px;
}

.a1 {
width:40px; height: 40px;
border-style: solid;
text-align:center;
font-size:12pt;
}

#menu {
text-align: center;
border-bottom-style: solid;
background-color: #f66;
padding: 20px;
```

```
}

#help {
    text-align: center;
    border-style: solid;
    background-color: #eee;
    display:none;
    position: absolute;
    top: 50px;
    left: 25%;

    width:50%;
}

.btn {
    background: #3498db;
    background-image: -webkit-linear-gradient(top, #3498db, #2980b9);
    background-image: -moz-linear-gradient(top, #3498db, #2980b9);
    background-image: -ms-linear-gradient(top, #3498db, #2980b9);
    background-image: -o-linear-gradient(top, #3498db, #2980b9);
    background-image: linear-gradient(to bottom, #3498db, #2980b9);
    -webkit-border-radius: 10;
    -moz-border-radius: 10;
    border-radius: 10px;
    font-family: Arial;
    color: #ffffff;
    font-size: 19px;
```

```
padding: 10px 20px 10px 20px;
text-decoration: none;
}

.btn:hover {
background: #fc3c83;
background-image: -webkit-linear-gradient(top, #fc3c83, #d93434);
background-image: -moz-linear-gradient(top, #fc3c83, #d93434);
background-image: -ms-linear-gradient(top, #fc3c83, #d93434);
background-image: -o-linear-gradient(top, #fc3c83, #d93434);
background-image: linear-gradient(to bottom, #fc3c83, #d93434);
text-decoration: none;
}

#main {
padding: 20px;
text-align: center;
}

</style>

</head>

<body>
<div id=menu>
<h1>Choose Table Size:</h1> <br>
```

```
<a href='game1.html'><button class=btn onclick='reset();'>8 x 8</button></a>
<a href='game2.html'><button class=btn onclick='reset();'>10 x 10</button></a>
<a href='game3.html'><button class=btn onclick='reset();'>14 x 14</button></a>

</div>

</body>

</html>
```

Βιβλιογραφία

1. Cordova, D. I., & Lepper, M. R. (1996). Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of educational psychology*, 88(4), 715.
2. Facer, K. (2003). *Screenplay: Children and computing in the home*. Psychology Press
3. Fabricatore, C. (2000). Learning and videogames: An unexploited synergy.
4. Fudenberg, D., & Levine, D. K. (1998). *The theory of learning in games (Vol. 2)*. MIT press.
5. Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. *Simulation & gaming*, 33(4), 441-467.
6. Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1), 20-20
7. Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and culture*, 1(1), 36-40.
8. Klawe, M. (1999, September). Computer games, education and interfaces: The E-GEMS project. In *Graphics interface* (pp. 36-39).
9. Kirriemuir, J., & McFarlane, A. (2004). Literature review in games and learning.
10. Malone, T. W. (1980, September). What makes things fun to learn? Heuristics for designing instructional computer games. In *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems* (pp. 162-169). ACM.
11. Malone, T. (1981). What makes computer games fun? (Vol. 13, No. 2-3, p. 143). ACM.
12. Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. BasicBooks, 10 East 53rd St., New York, NY 10022-5299.
13. Prensky, M. (2001). Fun, play and games: What makes games engaging. *Digital game-based learning*, 5, 1-05.
14. Rogers (1999), *Learning for Adults*.
15. Seelye, H. N. (1984). *Teaching Culture. Strategies for Intercultural Communication*.
16. Sandford, R., & Williamson, B. (2005). *Games and learning. A handbook*. Bristol, UK: FutureLab.
17. Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS quarterly*, 425-478.