ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: Ανάπτυξη εκπαιδευτικού λογισμικού με χρήση πολυμέσων για την Ψηφιακή Σχεδίαση

Φοιτητές: **Μαργαριτόπουλος Μερκούριος**
Μάστορας Κωνσταντίνος

Επιβλέπων: **Επ. καθηγητής Μανιτσάρης Αθανάσιος**

ΘΕΣΣΑΛΟΝΙΚΗ, ΙΟΥΝΙΟΣ 1997

ΠΕΡΙΕΧΟΜΕΝΑ

σελ.

ΕΙΣΑΓΩΓΗ 1

ΚΕΦΑΛΑΙΟ 1°. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΟΡΟΛΟΓΙΑ ΤΩΝ HYPERMEDIA - MULTIMEDIA 4
1.1 Διαλογικότητα - Interactivity 4
1.2 Υπερκείμενα - Hypertext 5
1.3 Υπερμέσα - Hypermedia 6
1.4 Πολυμέσα - Multimedia 7
1.5 Διαλογικά Πολυμέσα - Interactive Multimedia 8
1.6 Υπερέγγραφο - Hyperdocument 8
1.7 Interface 9
1.8 Συγκριτική παρουσίαση των εννοιών 10

ΚΕΦΑΛΑΙΟ 2°. Η ΔΟΜΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ 12
2.1 Η οργάνωση της εφαρμογής σε αρχεία 12
2.2 Η δομή του τμήματος Διδασκαλίας 13
2.3 Η δομή του τμήματος Εξάσκησης 15
2.4 Η δομή του τμήματος Αξιολόγησης 17
2.4.1Περιβάλλον συγγραφής και διόρθωσης τεστ 18
2.4.2 Περιβάλλον επίλυσης τεστ 19
2.4.3 Περιβάλλον ελέγχου αποτελεσμάτων 21
2.4.4 Η ροή πληροφοριών του τμήματος Αξιολόγησης 21

ΚΕΦΑΛΑΙΟ 3°. Η ΧΡΗΣΗ ΤΗΣ ΨΗΦΙΑΚΗΣ ΣΧΕΔΙΑΣΗΣ 26
3.1 Εισαγωγή στην εφαρμογή 26
3.2 Περιγραφή του τμήματος Διδασκαλίας 31
3.2.1 Περιγραφή του πίνακα περιεχομένων 31
3.2.2 Περιγραφή του βασικού περιβάλλοντος της θεωρίας 35
3.2.3 Περιγραφή της βοήθειας για τη χρήση των τμημάτων Διδασκαλίας και Αξιολόγησης 40
3.2.4 Χαρακτηριστικές σελίδες του τμήματος Διδασκαλίας 42
3.3 Περιγραφή του τμήματος Εξάσκησης 48
3.3.1 Περιγραφή του βασικού περιβάλλοντος των ασκήσεων 48
3.3.2 Χαρακτηριστικές ασκήσεις του τμήματος Εξάσκησης 51
3.4 Περιγραφή του τμήματος Αξιολόγησης 62
3.4.1 Κυρίως Μενού 62
3.4.2 Περιβάλλον συγγραφής και διόρθωσης τεστ 63
3.4.3 Περιβάλλον επίλυσης τεστ 69
3.4.4 Περιβάλλον ελέγχου αποτελεσμάτων 71
3.4.5 Περιβάλλον βοήθειας 75

ΚΕΦΑΛΑΙΟ 4°. Ο ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΗΣ ΨΗΦΙΑΚΗΣ ΣΧΕΔΙΑΣΗΣ 76
4.1 Μηνύματα του ΤΒ που χρησιμοποιεί η εφαρμογή 76
4.2 Αναφορές στον κώδικα της εισαγωγής στην εφαρμογή 79
4.2.1 Υλοποίηση του κινούμενου βιβλίου που αλλάζει σελίδες 79
4.2.2 Υλοποίηση του πίνακα περιεχομένων 82
4.3 Αναφορές στον κώδικα του τμήματος Διδασκαλίας 85

4.3.1 Υλοποίηση του slider πλοήγησης εντός των υποενοτήτων 85
4.3.2 Απόδοση ηχητικών κομματιών με χρήση MCI συναρτήσεων 88
4.3.3 Υλοποίηση του animation με τον αριθμό που περνάει διαδοχικά από 4 αριθμητικά συστήματα 91
4.4 Αναφορές στον κώδικα του τμήματος Εξάσκησης 93
4.4.1 Υλοποίηση του μαυροπίνακα με την κιμωλία 93
4.4.2 Υλοποίηση του χειρισμού του περιβάλλοντος Εξάσκησης 95
4.4.3 Υλοποίηση της άσκησης με τα συμπληρώματα ως προς 10 98
4.5 Αναφορές στον κώδικα του τμήματος Αξιολόγησης 99
4.5.1 Συνεχής μουσική σε όλα τα βιβλία 99
4.5.2 Μέγιστος αριθμός χαρακτήρων σε ένα πεδίο 100
4.5.3 Αυτόματη γραμμή κύλισης σε πεδίο 101
4.5.4 Συνάρτηση εύρεσης των ημερών ενός μήνα 102
4.5.5 Έλεγχος δίσεκτου έτους 103
4.5.6 Σύγκριση δύο ημερομηνιών 103
4.5.7 Υλποίηση του παραθύρου Ρολόϊ - Χρονόμετρο 104
4.5.8 Ρουτίνα πλοήγησης στην Βάση Δεδομένων 107

ΚΕΦΑΛΑΙΟ 5°. ΕΠΙΛΟΓΟΣ - ΣΥΜΠΕΡΑΣΜΑΤΑ 112

ΠΑΡΑΡΤΗΜΑ Α. ΕΙΣΑΓΩΓΗ ΣΤΟ MULTIMEDIA TOOLBOOK ΕΚΔ. 3.0 ΤΗΣ ASYMETRIX 114
Α.1 Γενικά για το Toolbook 115
Α.1.1 Εισαγωγή 115
Α.1.2 Επίπεδο Συγγραφέα και Αναγνώστη 115
Α.1.3 Αντικειμενοστραφής Προγραμματισμός 116
Α.1.4 Οδηγούμενο από συμβάντα (Events - driven) 116
Α.1.5 Ιεραρχία αντικειμένων 117
Α.2 Αντικείμενα και οι ιδιότητές τους 118
Α.2.1 Αντικείμενα που σχεδιάζει το Toolbook 118
Α.2.2 Αντικείμενα που σχεδιάζει ο χρήστης 120
Α.2.3 Ιδιότητες που ορίζει ο χρήστης (User Properties) 123
Α.3 Συγγραφή κώδικα 123
Α.3.1 Επισκόπηση σεναρίων (Scripting overview) 123
Α.3.2 Διόρθωση σεναρίων 128
Α.4 Στοιχεία της γλώσσας Openscript 129
Α.4.1 Έτοιμα μηνύματα 129
Α.4.2 Μηνύματα φτιαγμένα από το χρήστη (procedures) 131
Α.4.3 Αναφορά σε ιδιότητες αντικειμένων 131
Α.4.4 Statements και συναρτήσεις 133
Α.4.5 Μεταβλητές 136
Α.5 Ο χειρισμός των multimedia από το ΤΒ 140
Α.5.1 Γενικά 140
Α.5.2 Clips 140
Α.5.3 Multimedia πηγές 141
Α.5.4 Stages (σκηνές) 142
Α.5.5 Multimedia εντολές 143
Α.5.6 Έλεγχος multimedia συσκευών με τη χρήση MCI εντολών 144
Α.6 Ο χειρισμός των βάσεων δεδομένων από το ΤΒ 145
Α.6.1 Χρήση του Μηχανισμού της Paradox 145
Α.6.2 Κανόνες ονομασίας 147

Α.6.3 Τύποι δεδομένων 147
Α.6.4 Αρχικοποίηση του Μηχανισμού της Paradox 148
Α.6.5 Επεξεργασία πινάκων 149
Α.6.6 Μετάβαση σε εγγραφές 149
Α.6.7 Παροχή πληροφοριών για ένα πεδίο 150
Α.6.8 Index και αναζήτηση 150

ΠΑΡΑΡΤΗΜΑ Β. Ο ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ 153

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΕΙΣΑΓΩΓΗ

Τα Hypermedia - Multimedia είναι ένας τομέας της Πληροφορικής ο οποίος αν και δεν είναι καινούριος σε επίπεδο σύλληψης και προσδιορισμού των αντίστοιχων εννοιών, συγκεντρώνει μόλις πολύ τελευταία το ενδιαφέρον και την προσοχή τόσο των εταιρειών πληροφορικής, όσο και των «καταναλωτών». Η εντυπωσιακή δυναμική που έχει να παρουσιάσει - καθημερινά γίνεται όλο και περισσότερο αντιληπτή - και η ραγδαία εξελισσόμενη τεχνολογία, καθιστούν βέβαιη την προοπτική της πολύ σύντομης καθιέρωσης των Multimedia σαν το standard, τουλάχιστον στο χώρο των μικροϋπολογιστών.

Τα πολυμέσα είναι μια τεχνολογία στην οποία συνυπάρχουν διάφορες μορφές δεδομένων όπως κείμενο, γραφικά, ψηφιοποιημένη φωνή, κινούμενα σχέδια, γραφικά κ.λπ. Εφαρμογή πολυμέσων θεωρείται ένα πληροφοριακό σύστημα που αξιοποιεί ένα ή και περισσότερα από τα παραπάνω στοιχεία. Αν στα παραπάνω συμπεριλάβουμε και την αμφίδρομη συμμετοχή του χρήστη, τότε μιλάμε για αλληλεπιδρώντα πολυμέσα και για Υπερμέσα. Τα Υπερμέσα βασίζονται στην έννοια της ταυτόχρονης επεξεργασίας και παρουσίασης πληροφοριών με διαφορετική μορφή που διακινούνται και συνυπάρχουν στο ίδιο σύστημα. Αποτελούν, δηλ. έναν αρμονικό συνδυασμό της τεχνολογίας των πολυμέσων με την τεχνική της επικοινωνίας [3].

Ένας βασικός τομέας, των Υπερμέσων, είναι και η εκπαίδευση, η οποία επιτρέπει στον εκπαιδευόμενο να συμμετέχει άμεσα (ενεργά) στην εκπαιδευτική διαδικασία. Εξάλλου, η ταχύτητα αφομοίωσης αυξάνεται, αφού είναι αποδεδειγμένο ότι το άτομο απομνημονεύει το 20% όσων ακούει και βλέπει, και το 75% όσων έχει συμμετάσχει και το ίδιο, σε πρακτικό επίπεδο [8].

Παρακινούμενοι από το μέχρι τώρα διαμορφωμένο αλλά πολύ περισσότερο από το ταχύτατα εξελισσόμενο περιβάλλον στο χώρο της πληροφορικής, όπου, όπως προηγουμένως αναφέρθηκε, τα Hypermedia - Multimedia τείνουν να καθιερωθούν ως standard στο χώρο των μικροϋπολογιστών, επιλέξαμε ένα θέμα που κινείται στο πεδίο αυτό και ειδικότερα στον εκπαιδευτικό χώρο, για να το πραγματευτούμε για την πτυχιακή μας εργασία. Ειδικότερα, το αντικείμενο που επελέγη ήταν η σχεδίαση ενός εκπαιδευτικού προγράμματος hypermedia με στοιχεία multimedia που θα μπορούσε να χρησιμοποιηθεί από τους μαθητές - σπουδαστές στα πλαίσια του μαθήματος των Ψηφιακών Ηλεκτρονικών ως βοήθημα για την εκμάθηση της Ψηφιακής θεωρίας αλλά παράλληλα και την εξάσκηση σε ένα πλήθος σχετικών ασκήσεων και προβλημάτων, και επιπλέον την αξιολόγηση των μαθητών στη διδαχθείσα ύλη από τον καθηγητή του μαθήματος.

Έτσι, μετά από την ενασχόληση με το θέμα που περιγράφηκε πιο πάνω, εννέα και πλέον μηνών, προέκυψε η **Ψηφιακή Σχεδίαση**.

Η ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ είναι ένα αλληλεπιδραστικό εκπαιδευτικό πρόγραμμα που περιλαμβάνει τη διδασκαλία, εξάσκηση και αξιολόγηση του μαθητή - σπουδαστή στο μάθημα της σχεδίασης ψηφιακών ηλεκτρονικών κυκλωμάτων. Τα ψηφιακά κυκλώματα χρησιμοποιούνται στη σχεδίαση

συστημάτων όπως: ψηφιακών υπολογιστών, συστημάτων ελέγχου, μετάδοσης δεδομένων και πολλών άλλων εφαρμογών που απαιτούν ψηφιακό ηλεκτρονικό υλικό. Η ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ παρουσιάζει τα βασικά εργαλεία για τη σχεδίαση ψηφιακών κυκλωμάτων με εποπτικό τρόπο ώστε να είναι ευνόητα και ευκολότερα στην εμπέδωση από το μαθητή, με τη βοήθεια και τα μέσα που παρέχει μια hypermedia - multimedia εφαρμογή.

Ταυτόχρονα και παράλληλα με το τμήμα διδασκαλίας και εξάσκησης στην ψηφιακή θεωρία, αναπτύχθηκε και ένα εργαλείο αξιολόγησης μαθητών και σπουδαστών στην Ψηφιακή Θεωρία το οποίο δίνει τη δυνατότητα στον καθηγητή να αξιολογήσει το επίπεδο γνώσεων που αποκόμισε ο μαθητής από τη μελέτη της θεωρίας. Το εργαλείο αυτό αποτελεί ένα σύστημα δημιουργίας τεστ ερωτήσεων πολλαπλής επιλογής (multiple choice) και μπορεί να λειτουργήσει ανεξάρτητα από την Ψηφιακή Σχεδίαση, για οποιοδήποτε μάθημα.

Ασφαλώς, το συγκεκριμένο θέμα (Ψηφιακά Ηλεκτρονικά) που επελέγη, δεν αποτελεί το καταλληλότερο πεδίο για εφαρμογή multimedia μεθόδων. Παρόλα αυτά εκτιμάται ότι ο τρόπος παρουσίασης του θέματος παρέχει στο μαθητή ένα ισχυρό εργαλείο αναζήτησης, εκμάθησης και εξάσκησης της ψηφιακής θεωρίας, που είναι ταυτόχρονα εύχρηστο και ευχάριστο στη χρήση.

Η παρούσα εργασία είναι διαρθρωμένη και απαρτίζεται από πέντε κεφάλαια:

Το **πρώτο κεφάλαιο** καταπιάνεται με μια προσπάθεια επεξήγησης και αποσαφήνισης κυρίαρχων όρων του τομέα αυτού της πληροφορικής, που η χρήση τους γίνεται αδιάκριτα από τους περισσότερους. Συγκεκριμένα, διευκρινίζονται βασικές θεωρητικές διαφορές ανάμεσα στους όρους Multimedia, Interactive Multimedia, Hypermedia και Hypertext και περιγράφονται τα ιδιαίτερα χαρακτηριστικά του καθενός. Μ' αυτόν τον τρόπο επιχειρείται να δοθεί ένα ελάχιστο θεωρητικό υπόβαθρο.

Το **δεύτερο κεφάλαιο** επιχειρεί την παρουσίαση της δομής της εφαρμογής, τον τρόπο δηλ. με τον οποίο διατάσσονται και λειτουργούν ως μέρη ενός ενιαίου και αδιαίρετου συνόλου τα τμήματα από τα οποία αποτελείται η εφαρμογή, δηλ. τα τμήματα διδασκαλίας, εξάσκησης και αξιολόγησης. Επίσης, γίνεται αναφορά στους τρόπους επικοινωνίας και σύνδεσης των τμημάτων μεταξύ τους.

Στο **τρίτο κεφάλαιο** περιγράφεται η εφαρμογή της **Ψηφιακής Σχεδίασης**. Γίνεται αναλυτική περιγραφή της λειτουργίας του προγράμματος από τη σκοπιά του χρήστη. Συγκεκριμένα, γίνονται αναφορές στο τμήμα διδασκαλίας (τρόπος πλοήγησης στις σελίδες του ηλεκτρονικού βιβλίου, μέθοδοι παρουσίασης συγκεκριμένων θεμάτων της θεωρίας κ.λπ.), στο τμήμα εξάσκησης (τρόπος παρουσίασης της λύσης των ασκήσεων), καθώς, επίσης, και στο τμήμα της αξιολόγησης των μαθητών στη διδαχθείσα ύλη (συγγραφή τεστ, επίλυση τεστ κ.λπ.).

Στο **τέταρτο κεφάλαιο** αναλύεται η **Ψηφιακή Σχεδίαση** από τη σκοπιά του προγραμματιστή. Γίνονται ιδιαίτερες αναφορές σε επιλεγμένα σημεία του κώδικα που χρήζουν ιδιαίτερου σχολιασμού, όσον αφορά τις τεχνικές προγραμματισμού που χρησιμοποιήθηκαν (π.χ. υλοποίηση του slider πλοήγησης, χρήση MCI συναρτήσεων των Windows, σχεδίαση των βάσεων δεδομένων των ερωτήσεων για την αξιολόγηση κ.λπ.).

Το **πέμπτο κεφάλαιο** αποτελεί έναν επίλογο στην πτυχιακή εργασία όπου παρατίθενται τα συμπεράσματα που εξήχθησαν από την συνεχή ενασχόληση με το αντικείμενο που επιλέξαμε και αναφέρονται στην εμπειρία που αποκομίσαμε αλλά και στις προοπτικές του μέλλοντος.

Το **παράρτημα Α** παρουσιάζει μια εισαγωγή στο Multimedia Toolbook της Asymetrix το οποίο είναι ένα software πακέτο - εργαλείο συγγραφής για την ανάπτυξη πολυμεσικών εφαρμογών και ήταν αυτό που χρησιμοποιήθηκε για την ανάπτυξη της **Ψηφιακής Σχεδίασης**. Συγκεκριμένα, δίνονται τα βασικά στοιχεία για τον προγραμματισμό στο Toolbook, για τα αντικείμενα και τις ιδιότητές τους, για τη συγγραφή κώδικα, καθώς και στοιχεία της γλώσσας Openscript που χρησιμοποιεί το Toolbook. Επίσης, γίνονται αναφορές στους τρόπους με τους οποίους χειρίζεται το Toolbook τα multimedia, και τις βάσεις δεδομένων.

Τέλος, στο **παράρτημα Β** παρατίθεται αυτούσιος ολόκληρος ο κώδικας της εφαρμογής.

Στις προσπάθειές μας αυτές είχαμε την αμέριστη βοήθεια του εποπτεύοντος καθηγητή κ. Μανιτσάρη τον οποίο και ευχαριστούμε. Επίσης, πολλές ευχαριστίες ανήκουν στις Λαφαζάνη Στέλλα και Μιλλερού Σοφία που δάνεισαν τη φωνή τους στα video clips και τα ηχητικά σποτς που συνοδεύουν το πρόγραμμα. Οι ευχαριστίες, όμως, σ' αυτές, ανήκουν περισσότερο για τη συμπαράστασή τους καθόλη τη διάρκεια της φοίτησής μας στο Πανεπιστήμιο, καθώς επίσης και για την υπομονή και ανοχή τους, χωρίς την οποία η παρούσα εργασία δεν θα είχε πραγματοποιηθεί.

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΟΡΟΛΟΓΙΑ ΤΩΝ HYPERMEDIA - MULTIMEDIA

Είναι γεγονός ότι τόσο σε θεωρητικό επίπεδο, όσο και σε επίπεδο τεχνολογίας και προϊόντων υπάρχει μια αρκετά σοβαρή διάσταση απόψεων και προτάσεων σε σχέση με τους όρους Hypermedia και Multimedia αλλά και σε σχέση με άλλους συναφείς όρους όπως, διαλογικότητα (interactivity), υπερκείμενο (hypertext) κ.λπ. Στη συνέχεια θα αναφερθούν όλα τα βασικά σημεία της σχετικής ορολογίας, ώστε να τεθεί το απαραίτητο θεωρητικό υπόβαθρο που θα βοηθήσει στην κατανόηση της **Ψηφιακής Σχεδίασης**.

1.1 ΔΙΑΛΟΓΙΚΟΤΗΤΑ - INTERACTIVITY

Η απόδοση του όρου **interactivity** στα ελληνικά μπορεί να γίνει τόσο με τη χρησιμοποίηση της λέξης **διαλογικότητα**, όσο και με τη χρήση της λέξης **αλληλεπίδραση**, απόδοση που θεωρείται εξίσου αποδεκτή.

Σαν διαλογικά υπολογιστικά συστήματα *(interactive computer systems)*, θεωρούνται εκείνα τα υπολογιστικά συστήματα, στα οποία ο *τελικός χρήστης (end user)* μπορεί να ελέγχει την ανάκτηση, την επεξεργασία και την εμφάνιση της πληροφορίας. Αυτά τα υπολογιστικά συτήματα απαιτούν μια εξειδικευμένη σχεδίαση και κατασκευή όσον αφορά την *επικοινωνία με το χρήστη (user interface)*, με σκοπό να διευκολύνουν τη ροή της πληροφορίας μεταξύ του υπολογιστικού συστήματος και του χρήστη [1].

Θεμελιώδες στοιχείο στη σχεδίαση αυτού του είδους επικοινωνίας με το χρήστη, είναι η ιδέα της **επίδρασης (reactivity)**. Θεωρούμε σαν «reactive user interface» εκείνον το μηχανισμό, μέσω του οποίου ένας χρήστης υπολογιστικού συστήματος μπορεί να ενεργοποιήσει μία διεργασία, η οποία θα λάβει χώρα στο υπολογιστικό σύστημα, ανεξάρτητα από το αν αναφέρεται σε *τοπικό (local)* ή σε *απομακρυσμένο (remote)* υπολογιστικό σύστημα. Σε μια λοιπόν **επιδραστική επικοινωνία με το χρήστη (reactive user interface)** η ροή της πληροφορίας πραγματοποιείται σε μονοσήμαντη κατεύθυνση, από το χρήστη προς τον υπολογιστή.

Περισσότερο χρήσιμος τύπος επικοινωνίας με το χρήστη είναι αυτός που προσφέρει επικοινωνία χρήστη και υπολογιστή σε διπλή κατεύθυνση, κατά τρόπο ώστε η ροή της πληροφορίας πραγματοποιούμενη αμφίδρομα να σχηματίζει την έννοια του διαλόγου. Αυτού του είδους οι επικοινωνίες, είναι αυτές που αναφέρονται σαν **διαλογική επικοινωνία με το χρήστη (interactive user interface)**.

Το ουσιαστικό στοιχείο που χαρακτηρίζει τον interactive τύπο είναι το γεγονός της αποστολής πίσω στο χρήστη κάποιας *ανάδρασης (feedback)*, η οποία προέρχεται από τη διαδικασία επεξεργασίας που ενεργοποίησε ο χρήστης στο αμέσως προηγούμενο στάδιο. Τα μηνύματα που λαμβάνει ο χρήστης, υλοποιούμενη με αυτόν τον τρόπο η έννοια της ανάδρασης, μπορούν να έχουν σαν αποτέλεσμα την εμφάνιση αλλαγών στις διαθέσεις, τις σκέψεις και τις προθέσεις του χρήστη απέναντι στον υπολογιστή. Έτσι με τη

σειρά του μπορεί ο χρήστης να τροποποιήσει τις λειτουργίες που εκτελεί ο υπολογιστής, σε εκείνα, βέβαια, τα σημεία που ο σχεδιασμός του λογισμικού επιτρέπει την παρέμβασή του. Πραγματοποιούνται λοιπόν με αυτόν τον τρόπο, αμοιβαίες αλλαγές, τόσο στο υπολογιστικό, όσο και στον ανθρώπινο παράγοντα, αλλαγές που οφείλονται και υλοποιούνται μέσω των μηχανισμών της ανάδρασης. Ακριβώς αυτή η σχέση της αλληλεπίδρασης στην εναλλαγή των καταστάσεων, αποτελεί τη βάση του τομέα εκείνου της επιστήμης της Πληροφορικής που ονομάζεται Αλληλεπίδραση Ανθρώπου - Υπολογιστή (Human - Computer Interaction) [1].

Δίνεται μεγάλη σημασία στην έννοια της διαλογικότητας γιατί είναι βασικό σημείο διαφοροποίησης των εννοιών που θα ανφερθούν στη συνέχεια.

1.2 ΥΠΕΡΚΕΙΜΕΝΟ - HYPERTEXT

Ο όρος **Υπερκείμενο (Hypertext)** δημιουργήθηκε από τον Theodor Holme Nelson, το 1965, για να περιγράψει αρχεία κειμένου, οργανωμένα μη γραμμικά μη σειριακά. Αναφερόμενος σε ένα υπολογιστικό σύστημα, παρατήρησε πως ο χρόνος για να βρεθεί ένα κομμάτι πληροφορίας κειμένου, ήταν αναλογικά πολύ μεγαλύτερος από το χρόνο που απαιτείτο, στη συνέχεια, για να μεταφερθεί αυτό το κομμάτι προς επεξεργασία στην Κ.Μ.Ε.. Ανίχνευσε λοιπόν το πρόβλημα στον τρόπο οργάνωσης και προσπέλασης των δεδομένων του και η παρατήρησή του αυτή, όντας εύστοχη, τον οδήγησε στο να προτείνει μια διαφορετική για την εποχή του προσέγγιση στον τρόπο διαχείρισης των δεδομένων.

Στον τρόπο διαχείρισης που πρότεινε, οι πληροφορίες είναι αποθηκευμένες σε ένα δίκτυο από *κόμβους (nodes)*, οι δε κόμβοι επικοινωνούν μεταξύ τους με *συνδέσμους (links)*. Στην ουσία, κόμβος είναι ένα σύνολο δεδομένων, τα οποία είναι οργανωμένα γύρω από ένα κοινό θέμα.

Τα κύρια χαρακτηριστικά των Υπερκειμένων είναι:

- αποτελούνται από ένα σύνολο πληροφοριών αποκλειστικά σε μορφή κειμένου, η πρόσβαση δε στην πληροφορία μπορεί να γίνει μέσα από πολλαπλούς διαδρόμους.
- στον τρόπο παρουσίασης, χρησιμοποιείται η μέθοδος των πολλαπλών παραθύρων, κατά την οποία σε διαφορετικά παράθυρα στην οθόνη, απεικονίζονται ταυτόχρονα πολλά διαφορετικά κείμενα.
- επιτρέπουν ένα σημαντικό βαθμό ελευθερίας σε επίπεδο διαλογικότητας: παρέχουν τη δυνατότητα στο χρήστη να κάνει τις επιλογές του σχετικά με τη διαδρομή που θα ακολουθήσει για να φτάσει στην επιζητούμενη πληροφορία.

Με αυτήν την έννοια, ο όρος υπερκείμενο διαφέρει ριζικά από τον όρο *ηλεκτρονικό βιβλίο*, ο οποίος θέλει να περιγράψει την ψηφιακή αποθήκευση, επεξεργασία και απεικόνιση σε οθόνη υπολογιστή, δεδομένων κειμένου, των οποίων η ανάγνωση, όμως, μπορεί να γίνει μόνο σειριακά.

Στην αρχική περίοδο εμφάνισης και χρήσης του όρου Υπερκείμενο, οι κόμβοι περιείχαν μόνο δεδομένα κειμένου, συνθήκη η οποία έρχεται σε πλήρη συμφωνία με τα τρία χαρακτηριστικά των υπερκειμένων που δόθηκαν

παραπάνω. Στη συνέχεια και από τη στιγμή που οι κόμβοι άρχισαν να περιέχουν διαφορετικού τύπου περιεχόμενα όπως γραφικά, video, ήχο , κ.λπ., οι θεωρητικοί της ορολογίας, διαφοροποιήθηκαν και πάλι χρησιμοποιώντας άλλοι σαν διαφορετικούς και άλλοι σαν ταυτόσημους τους όρους Υπερκείμενο και Υπερμέσα (Hypermedia), που θα εξεταστούν στη συνέχεια.

Θα πρέπει να τονιστεί πως όρος Υπερκείμενο χρησιμοποιείται για να περιγράψει και να χαρακτηρίσει είτε «πακέτα» λογισμικού (Hypertext systems), είτε εφαρμογές (Hypertext applications) δημιουργημένες με τα εν λόγω «πακέτα». Στην πραγματικότητα, σε κάποιες περιπτώσεις, δεν πρόκειται για «πακέτα» ή εφαρμογές Υπερκειμένου αλλά Υπερμέσων, αφού τα δεδομένα που μπορούν και χειρίζονται τα εν λόγω προγράμματα, δεν είναι μόνο δεδομένα κειμένου, αλλά επιπλέον δεδομένα πολλών και διαφορετικών μορφών, όπως γραφικά, ήχος, προσομοίωση κίνησης (animation), και video.

1.3 ΥΠΕΡΜΕΣΑ - HYPERMEDIA

Όταν τα «στοιχεία» της πληροφορίας που ήδη ορίστηκαν στην παράγραφο 1.2 σαν κόμβοι (nodes), δεν περιέχουν μόνο δεδομένα *κειμένου*, αλλά οποιασδήποτε μορφής δεδομένα, όπως *ήχο, γραφικά, εικόνες, προσομοίωση κίνησης (animation), video* κ.λπ., τότε αναφερόμαστε σε **Υπερμέσα (Hypermedia)**. Θα μπορούσε με άλλα λόγια να ειπωθεί πως Υπερμέσα δεν είναι παρά Υπερκείμενο, το οποίο μπορεί να χειριστεί όχι μόνο δεδομένα κειμένου, αλλά και δεδομένα οποιασδήποτε μορφής.

Τα κύρια χαρακτηριστικά της φιλοσοφικής σύλληψης του Υπερκειμένου προσαρμόζονται και στο χώρο των Υπερμέσων. Ένα σύνολο πληροφοριών που ανήκουν σε διαφορετικούς τύπους το καθένα (κείμενο, ήχος, εικόνα κ.λπ.), μπορούν να παρουσιαστούν στο χρήστη μέσα από διαφορετικούς και ποικίλους δρόμους πρόσβασης, δρόμους μη καθορισμένους, που μπορεί να τους καθορίζει ο χρήστης διαφορετικά κάθε φορά, σύμφωνα με τις διαθέσεις του, τις επιθυμίες του και τα κύρια σημεία ενδιαφέροντός του.

Το γεγονός πως ο χρήστης μπορεί να «ταξιδεύει» μέσα σε αυτό το σύνολο των διαφορετικής μορφής πληροφοριών από ένα κομμάτι πληροφορίας (node) σε ένα άλλο, μέσω συνδετικών κρίκων (links), ή με άλλα λόγια η ύπαρξη συνδεδεμένων στοιχείων πληροφορίας (linked elements), στις οποίες ο χρήστης μπορεί να έχει πρόσβαση, αποτελεί το κύριο γνώρισμα των εφαρμογών Υπερμέσων.

Σημαντικότατο, πρωταρχικό χαρακτηριστικό και των Υπερμέσων είναι ο μη γραμμικός, μη σειριακός τρόπος οργάνωσης των δεδομένων, ακριβώς όπως συμβαίνει και με το Υπερκείμενο. Ο τρόπος αυτός οργάνωσης σχετίζεται δε άμεσα με την έννοια της διαλογικότητας, έννοια η οποία είναι παρούσα και είναι βασικό χαρακτηριστικό γνώρισμα των εφαρμογών Υπερμέσων. Αυτό εξάλλου το χαρακτηριστικό, είναι το σημείο διαφοροποίησης Υπερμέσων και Πολυμέσων (Multimedia) τα οποία αναφέρονται στη συνέχεια.

Ο όρος Υπερμέσα χρησιμοποιείται, όπως άλλωστε και ο όρος Υπερκείμενο, για να περιγράψει τόσο «πακέτα» λογισμικού, όσο και εφαρμογές. Μια εφαρμογή Υπερμέσων είναι μια εφαρμογή στηριγμένη σε μια

φιλοσοφία μη γραμμικής ανάπτυξης των προγραμμάτων της, γεγονός που έχει σαν αποτέλεσμα αυτές τις ποικίλες δυνατότητες πλοήγησης του χρήστη μέσα στο σύνολο των διαφορετικής μορφής πληροφοριών που ήδη αναφέρθηκε παραπάνω. Και, τέλος, «πακέτα» Υπερμέσων είναι εκείνα τα προγράμματα συγγραφής (authoring tools), που είναι δομημένα έτσι ώστε να μπορούν, αφενός να χειρίζονται όλους αυτούς τους διαφορετικούς τύπους δεδομένων, αφετέρου να παρέχουν τρόπους μη γραμμικής οργάνωσης των δεδομένων, και τα οποία χρησιμοποιούνται για τη δημιουργία εφαρμογών Υπερμέσων [1].

1.4 ΠΟΛΥΜΕΣΑ - MULTIMEDIA

Multimedia είναι όρος που κατεξοχήν, ιδιαίτερα στη χώρα μας, ακούγεται και χρησιμοποιείται για να περιγράψει και να δηλώσει οτιδήποτε κινείται και αναφέρεται στο γενικότερο χώρο των Υπερμέσων. Μια πολύ μεγάλη μερίδα ανθρώπων της Πληροφορικής, δεν κάνει σαφή διάκριση ανάμεσα στους όρους **Πολυμέσα (Multimedia)** και Υπερμέσα (Hypermedia). Εν τούτοις, η διαφορά ανάμεσα στους δύο όρους υπάρχει και είναι σαφέστατη.

Πριν αναφερθούμε σ' αυτήν, θα πρέπει να σημειωθεί πως, κατ' αναλογία με το τι συμβαίνει και με τους άλλους παρεμφερείς όρους, έτσι και ο όρος Πολυμέσα, εμφανίζεται σαν επιθετικός προσδιορισμός στις εκφράσεις «εφαρμογές Πολυμέσων (Multimedia applications)», «πακέτα Πολυμέσων», και Multimedia ηλεκτρονικοί υπολογιστές ή συνηθέστερα MPCs (Multimedia Personal Computers). Οι σχετικές διευκρινίσεις των όρων με τον επιθετικό προσδιορισμό Υπερμέσα που δόθηκαν στην προηγούμενη παράγραφο, αντιστοιχίζονται πλήρως και για την περίπτωση που σαν επιθετικός προσδιορισμός χρησιμοποιείται ο όρος Πολυμέσα.

Μια εφαρμογή Πολυμέσων χαρακτηρίζεται από τη συλλογή και χρησιμοποίηση ενός συνδυασμού δεδομένων κειμένου, γραφικών, ήχου, κινούμενων σχεδίων και video ή κάποιων από αυτά [5]. Ο χρήστης της εφαρμογής Πολυμέσων δεν μπορεί να παρέμβει κατά τη διάρκεια του «τρεξίματος» της εφαρμογής (πέρα από το να τη σταματήσει), δεν έχει κανενός είδους έλεγχο πάνω στη ροή της εφαρμογής, δεν μπορεί να επιλέξει τι και πότε θα δεί (ή ακούσει) και δεν μπορεί παρά να μείνει απαθής θεατής των όσων εξελίσσονται μπροστά στα μάτια του (και στα αυτιά του), μέσα από τον υπολογιστή.

Αυτή ακριβώς η έλλειψη ελέγχου από την πλευρά του χρήστη πάνω στην εφαρμογή, είναι που χαρακτηρίζει μία εφαρμογή Πολυμέσων, πέρα, βέβαια, από το γεγονός, της χρησιμοποίησης όλων αυτών των ποικίλων μορφών δεδομένων που έχουμε αναφέρει αρκετές φορές μέχρι τώρα. Από τη στιγμή που αυτή η έλλειψη ελέγχου παύει να ισχύει, τότε πλέον πρέπει να μιλάμε ανάλογα με την αρχιτεκτονική και τη δομή της εφαρμογής, είτε για εφαρμογή Διαλογικών Πολυμέσων (Interactive Multimedia application), είτε για εφαρμογή Υπερμέσων (Hypermedia application).

Η διαφοροποίηση αυτή ανάμεσα σε Πολυμέσα, Διαλογικά Πολυμέσα και Υπερμέσα, θα πρέπει να σημειωθεί πως υφίσταται τόσο σε επίπεδο εφαρμογών, όσο και σε επίπεδο των «πακέτων» ανάπτυξής τους. Κάποια

εργαλεία συγγραφής εφαρμογών υποστηρίζουν τη μη σειριακή ανάπτυξη των δεδομένων της εφαρμογής και τη δημιουργία κόμβων και συνδέσμων, κάποια άλλα τη δυνατότητα παροχής ελέγχου του χρήστη πάνω στην εφαρμογή και κάποια τρίτα τη σειριακή και μόνο συγγραφή διαδοχικών γεγονότων που εξελίσσονται ανεξάρτητα από την παρουσία του χρήστη υλοποιώντας κατ΄ αυτόν τον τρόπο την κατηγοριοποίηση των εφαρμογών σε Υπερμέσων, Διαλεκτικών Πολυμέσων και Πολυμέσων. Όσον αφορά το υλικό, ένας υπολογιστής Multimedia ή MPC μπορεί να χρησιμοποιηθεί το ίδιο καλά για το «τρέξιμο» μίας εφαρμογής Πολυμέσων ή Διαλογικών Πολυμέσων ή Υπερμέσων.

1.5 ΔΙΑΛΟΓΙΚΑ ΠΟΛΥΜΕΣΑ - INTERACTIVE MULTIMEDIA

Όλα τα γνωρίσματα των Πολυμέσων, εμπλουτισμένα με ένα ακόμη επιπλέον χαρακτηριστικό, μορφοποιούν την κατηγορία των **Διαλογικών Πολυμέσων (Interactive Multimedia)**. Μιλώντας λοιπόν για μια εφαρμογή Διαλογικών Πολυμέσων (Interactive Multimedia application), αναφερόμαστε σε μια εφαρμογή στην οποία μπορούν να χρησιμοποιούνται κατά συνθήκη όλες οι γνωστές μορφές δεδομένων (κείμενο, ήχος γραφικά, εικόνες, video), και που έχει σαν επιπρόσθετο χαρακτηριστικό τη δυνατότητα του χρήστη να επεμβαίνει πάνω στην εξέλιξη της εφαρμογής και να καθορίζει το τι και πότε θα δει ή θα ακούσει.

Αυτή ακριβώς η δυυνατότητα αλληλεπίδρασης προγράμματος και χρήστη είναι που διαφοροποιεί μια εφαρμογή Διαλογικών Πολυμέσων από μια εφαρμογή Πολυμέσων. Από την άλλη όμως πλευρά, οι δυνατότητες του χρήστη είναι περιορισμένες και οι επιλογές του περιλαμβάνονται σε ένα αυστηρά καθορισμένο πλαίσιο. Αυτός ο περιορισμός, βέβαια, έχει να κάνει με τον τρόπο δόμησης του προγράμματος και την οργάνωση των δεδομένων της εφαρμογής.

Η αρχιτεκτονική δόμησης του προγράμματος που ακολουθείται είναι συνήθως δενδροειδούς μορφής, αρχιτεκτονική που δεν σχετίζεται με αυτή που ακολουθείται στην περίπτωση μιας εφαρμογής Υπερμέσων, όπου χαρακτηριστικό στοιχείο είναι η μη σειριακή οργάνωση των δεδομένων και θεμελιώδεις έννοιες οι κόμβοι και οι σύνδεσμοι.

1.6 ΥΠΕΡΕΓΓΡΑΦΟ - HYPERDOCUMENT

Ο όρος **Υπερέγγραφο (Hyperdocument)** αποτελεί νεολογισμό που πρωτοπαρουσιάστηκε πολύ πρόσφατα και θέλει να προσδιορίσει και να αναφερθεί στο σύνολο των δεδομένων που χρησιμοποιούνται σε μια εφαρμογή Υπερμέσων.

Συνεχίζοντας τη σημαντική αυτή διαφοροποίηση που παρουσιάζεται σε θέματα ορολογίας, σχεδόν ταυτόχρονα εμφανίζονται οι όροι *Υπερβάση*

(Hyperbase) και *Υπεραντικείμενο Hyperobject)*. Συνολικά, οι τρεις αυτοί όροι αποτελούν έννοιες ταυτόσημες μεταξύ τους.

1.7 INTERFACE

Στο εγκυρότατο λεξικό Ηλεκτρικών και Ηλεκτρονικών όρων, IEEE Standard Directory of Electrical and Electronic Terms, η έννοια του όρου **interface** αποδίδεται με τη φράση «**shared boundary**» που μεταφράζεται «**κοινό ή διαμοιραζόμενο όριο ή σύνορο**». Ο όρος λοιπόν **user interface** θέλει να δείξει το σημείο επαφής χρήστη και μηχανής, τη διαχωριστική γραμμή πίσω από την οποία, από τη μία μεριά βρίσκεται ο υπολογιστής και από την άλλη ο άνθρωπος. Με άλλα λόγια ο όρος σημαίνει ένα σύνολο από, κύρια οπτικές, αλλά και ακουστικές παραστάσεις, που προβάλλει ο υπολογιστής μέσω του εκάστοτε εκτελούμενου προγράμματος προς τον χρήστη με σκοπό την καλύτερη επικοινωνία και συνεργασία μεταξύ ανθρώπου και μηχανής.

Επίσης, για την απόδοση του όρου interface χρησιμοποιούνται οι λέξεις «διασύνδεση», «επικοινωνία», «αντιμετώπιση» και «προσαρμογή» διαμορφώνοντας έτσι την ερμηνεία του όρου user interface σε «επικοινωνία με το χρήστη», «αντιμετώπιση του χρήστη» και «προσαρμογή του χρήστη», αντίστοιχα.

Η χρησιμότητα ή μη ενός προϊόντος εξαρτάται πολλές φορές από την ικανότητα του καταναλωτή να το χειριστεί με επιτυχία. Εάν ένας χρήστης δεν μπορεί να εξακριβώσει πώς λειτουργεί ένα προϊόν, τότε η επικοινωνία του προϊόντος με το χρήστη θεωρείται αυτόματα αποτυχημένη. Μέσα στον όρο «επικοινωνία με το χρήστη» περιλαμβάνεται η έννοια της αλληλεπίδρασης του χρήστη με το προϊόν. Τι μπορεί να κάνει ο χρήστης με το προϊόν, τι και τι μπορεί να κάνει το προϊόν για το χρήστη, είναι όλα μέρη της επικοινωνίας του χρήστη με το προϊόν, με λίγα λόγια μία συνομιλία πρόσωπο με πρόσωπο [1].

Παντού, κάθε χρονική στιγμή στην καθημερινή μας ζωή, μας περιβάλλουν καλά η άσχημα σχεδιασμένοι τρόποι επικοινωνίας του χρήστη με τα διάφορα προϊόντα. Αλλά όσο τα προϊόντα γίνονται πιο σύνθετα, ειδικά όταν πρόκειται για ηλεκτρικές και ηλεκτρονικές συσκευές, τότε όλο και πιο συχνά ο χρήστης απογοητεύεται από τη δυσκολία που παρουσιάζει ο τρόπος επικοινωνίας μαζί του και πολύ συχνά δεν έχει τη διάθεση να κάνει καμία προσπάθεια για να τα κατανοήσει και να τα χειριστεί. Περιμένει τα πράγματα να είναι απλά στην κατανόηση και τη χρήση και όταν δεν είναι έτσι, εγκαταλείπει το προϊόν σε μια άκρη, αρνούμενος να προσπαθήσει να βρει τρόπο επικοινωνίας μαζί του.

Στην κορυφή όλων αυτών των «δύστροπων» συσκευών που έχουν μπει στην καθημερινή ζωή, πάρα πολύ συχνά βρίσκεται ο ηλεκτρονικός υπολογιστής. Ο φυσικός έλεγχος του υλικού (πληκτρολόγιο, ποντίκι κ.λπ.) είναι λογικός και συγκεκριμένος, αλλά τα προβλήματα αρχίζουν από τις πολλές ποικιλίες λογισμικού, που έχουν σαν αποτέλεσμα τη συχνότατη εναλλαγή εικόνων στην οθόνη του υπολογιστή.

Μερικοί κατασκευαστές ηλεκτρονικών υπολογιστών προσπάθησαν να ξεπεράσουν αυτό το πρόβλημα θέτοντας κάποιες σταθερές στη σχεδίαση του

τρόπου επικοινωνίας χρήστη και υπολογιστή. Η πιο επιτυχής από αυτές τις προσπάθειες ήταν αυτή της Apple. Η Apple σχεδίασε κάποιες τέτοιου είδους σταθερές πριν αναπτύξει το λογισμικό της. Στη συνέχεια ζήτησε από όλους τους προγραμματιστές που θα ανέπτυσσαν λογισμικό για τους υπολογιστές της να ακολουθήσουν αυτές τις σταθερές. Επιπλέον ήταν αυτή που πρώτη εισήγαγε τη χρήση εικονιδίων και γραφημάτων στις διαδικασίες επικοινωνίας χρήστη και υπολογιστή, δημιουργώντας την έννοια της «**επικοινωνίας με το χρήστη με γραφικά**» (**graphical user interface**). Αυτή η μορφή επικοινωνίας χρήστη και υπολογιστή αυξάνει τη λειτουργικότητα, την ευκολία στη χρήση, την αποτελεσματικότητα και την ταχύτητα του διαλόγου μεταξύ ανθρώπου και μηχανής.

1.8 ΣΥΓΚΡΙΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΕΝΝΟΙΩΝ

Τέσσερις είναι οι βασικές έννοιες που παρουσιάστηκαν σε αυτό το κεφάλαιο και στην προσπάθειά μας για τη σαφή κατανόηση και εμπέδωσή τους θα επιχειρήσουμε και με τη βοήθεια του παρακάτω πίνακα να επανέλθουμε διαχωρίζοντας πλήρως τους όρους Υπερκείμενο (Hypertext), Υπερμέσα (Hypermedia), Διαλογικά Πολυμέσα (Interactive Multimedia) και Πολυμέσα (Multimedia).

	Μορφή Δεδομένων	**Αλληλεπίδραση με χρήστη**	**Μη γραμμική, μη σειριακή οργάνωση πληροφοριών**
Hypertext	κείμενο	ναι	ναι
Hypermedia	όλα	ναι	ναι
Interactive Multimedia	όλα	ναι	όχι
Multimedia	όλα	όχι	όχι

* όλα = κείμενο, ήχος, γραφικά, εικόνα, animation, video

Για να μπορέσουν να γίνουν ορατές οι διαφορές, αλλά και οι ομοιότητες μεταξύ των τεσσάρων αυτών εννοιών, θα χρησιμοποιηθούν τρεις παράμετροι: το είδος των δεδομένων που χρησιμοποιούνται στο αντίστοιχο είδος των εφαρμογών, ο τρόπος οργάνωσης των πληροφοριών των εν λόγω εφαρμογών και η δυνατότητα αλληλεπίδρασης των εφαρμογών με το χρήστη.

Αποσαφηνίζονται ορισμένα βασικά σημεία που σχετίζονται με τον τρόπο οργάνωσης των πληροφοριών. Όταν αναφέρεται μη γραμμική, μη σειριακή οργάνωση πληροφοριών, εννοείται μια θεωρητικά απεριόριστη δυνατότητα ποικιλότροπης πρόσβασης του χρήστη στο σύνολο των πληροφοριών που χειρίζεται η εφαρμογή. Κάθε κομμάτι πληροφορίας που αποτελεί έναν κόμβο (node), συνδέεται με άλλα θεματολογικά συσχετιζόμενα, κομμάτια

πληροφοριών, μέσω ειδικών διασυνδετικών κρίκων, των συνδέσμων (links). Ο χρήστης έχει τη δυνατότητα μέσω αυτών των συνδέσμων να «ταξιδεύει» μέσα στο σύνολο των πληροφοριών, ορίζοντας ο ίδιος το «δρομολόγιο» που κάθε φορά θέλει να ακολουθήσει. Σε *εννοιολογικό επίπεδο (conceptual level)* οι εφαρμογές Υπερκειμένου και Υπερμέσων, χαρακτηρίζονται από μη γραμμική, μη σειριακή οργάνωση των πληροφοριακών στοιχείων της εφαρμογής.

Οι εφαρμογές Υπερκειμένου διαφοροποιούνται από όλες τις άλλες από το γεγονός πως τα δεδομένα που χρησιμοποιούνται σε μια τέτοια εφαρμογή είναι μόνο δεδομένα κειμένου, αντίθετα δηλ. με το τι συμβαίνει στις άλλες τρεις περιπτώσεις όπου τα δεδομένα μπορεί να είναι όλων των μορφών (κείμενο, ήχος, γραφικά, προσομοίωση κίνησης, video). Από την άλλη πάλι πλευρά, οι εφαρμογές Υπερκειμένου συμπίπτουν ως προς τα άλλα δύο χαρακτηριστικά στοιχεία, τον τρόπο οργάνωσης των πληροφοριών και τη δυνατότητα αλληλεπίδρασης με το χρήστη, με τις εφαρμογές Υπερμέσων.

Οι εφαρμογές Υπερμέσων και Διαλογικών Πολυμέσων, συμπίπτουν ως προς το είδος των δεδομένων που χρησιμοποιούν και την ύπαρξη ελέγχου του χρήστη πάνω στην εφαρμογή, διαφοροποιούνται δε στον τρόπο οργάνωσης των πληροφοριών.

Δυνατότητα χρησιμοποίησης κάθε μορφής δεδομένων έχουμε στις εφαρμογές τύπου Υπερμέσων, Διαλογικών Πολυμέσων και Πολυμέσων ενώ δυνατότητα αλληλεπίδρασης εφαρμογής και χρήστη παρουσιάζεται στις περιπτώσεις των εφαρμογών Διαλογικών Πολυμέσων, Υπερμέσων και Υπερκειμένου. Τέλος, μη σειριακή οργάνωση των δεδομένων, με ταυτόχρονη ύπαρξη συνδέσμων και κόμβων, έχουμε στις περιπτώσεις των εφαρμογών Υπερμέσων και Υπερκειμένου.

Θα πρέπει για μια ακόμη φορά να τονιστεί πως η διαφοροποίηση στην ορολογία αφορά το λογισμικό μέρος, είτε η αναφορά γίνεται σε «πακέτα» ανάπτυξης εφαρμογών είτε στις ίδιες τις εφαρμογές, αφού οι ηλεκτρονικοί υπολογιστές με τα περιφερειακά τους που απαιτούνται για το «τρέξιμο» αυτού του είδους των εφαρμογών, είναι κοινοί σε όλες τις περιπτώσεις. Εξαίρεση αποτελεί η περίπτωση των εφαρμογών Υπερκειμένου, όπου οι απαιτήσεις σε υλικό είναι ακόμα μικρότερες σε σχέση με τις τρεις άλλες περιπτώσεις, αφού δεν χειρίζονται αυτού του είδους οι εφαρμογές παρά μόνο δεδομένα κειμένου και κατά συνέπεια ειδικές διατάξεις, όπως κάρτες video και ήχου, δεν απαιτούνται.

ΚΕΦΑΛΑΙΟ 2: Η ΔΟΜΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

2.1 Η ΟΡΓΑΝΩΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΣΕ ΑΡΧΕΙΑ

Η Ψηφιακή Σχεδίαση είναι μια εφαρμογή που αποτελείται από ένα πλήθος αρχείων του Toolbook. Η διαίρεση αυτή σε αρχεία ακολουθεί την εννοιολογική διάρθρωση της εφαρμογής σε τμήματα (διδασκαλία - εξάσκηση - αξιολόγηση) καθώς και σε κεφάλαια.

Έτσι, το βιβλίο CONTENTS.TBK περιλαμβάνει την εισαγωγή στην εφαρμογή, τη σελίδα επιλογής του τμήματος της εφαρμογής, καθώς και τον πίνακα περιεχομένων.

Τα βιβλία UNIT1.TBK - UNIT7.TBK περιλαμβάνουν τη θεωρία καθενός από τα 7 κεφάλαια από τα αποτελείται η Ψηφιακή Σχεδίαση.

Τα βιβλία EXER1.TBK - EXER7.TBK περιλαμβάνουν τις ασκήσεις που αναφέρονται σε κάθε ένα από τα 7 κεφάλαια της θεωρίας.

Το βιβλίο HELPD.TBK περιέχει τη βοήθεια για τη χρήση των τμημάτων της διδασκαλίας και της εξάσκησης.

Το τμήμα της αξιολόγησης απαρτίζεται από τα εξής βιβλία:

Το βιβλίο CONT1.TBK που περιέχει το περιβάλλον του κυρίως μενού.

Το βιβλίο QUEST1.TBK που περιέχει το περιβάλλον συγγραφής και διόρθωσης τεστ.

Το βιβλίο ANSWER1.TBK που περιέχει το περιβάλλον επίλυσης τεστ.

Το βιβλίο REPORT1.TBK που περιέχει το περιβάλλον ελέγχου των αποτελεσμάτων.

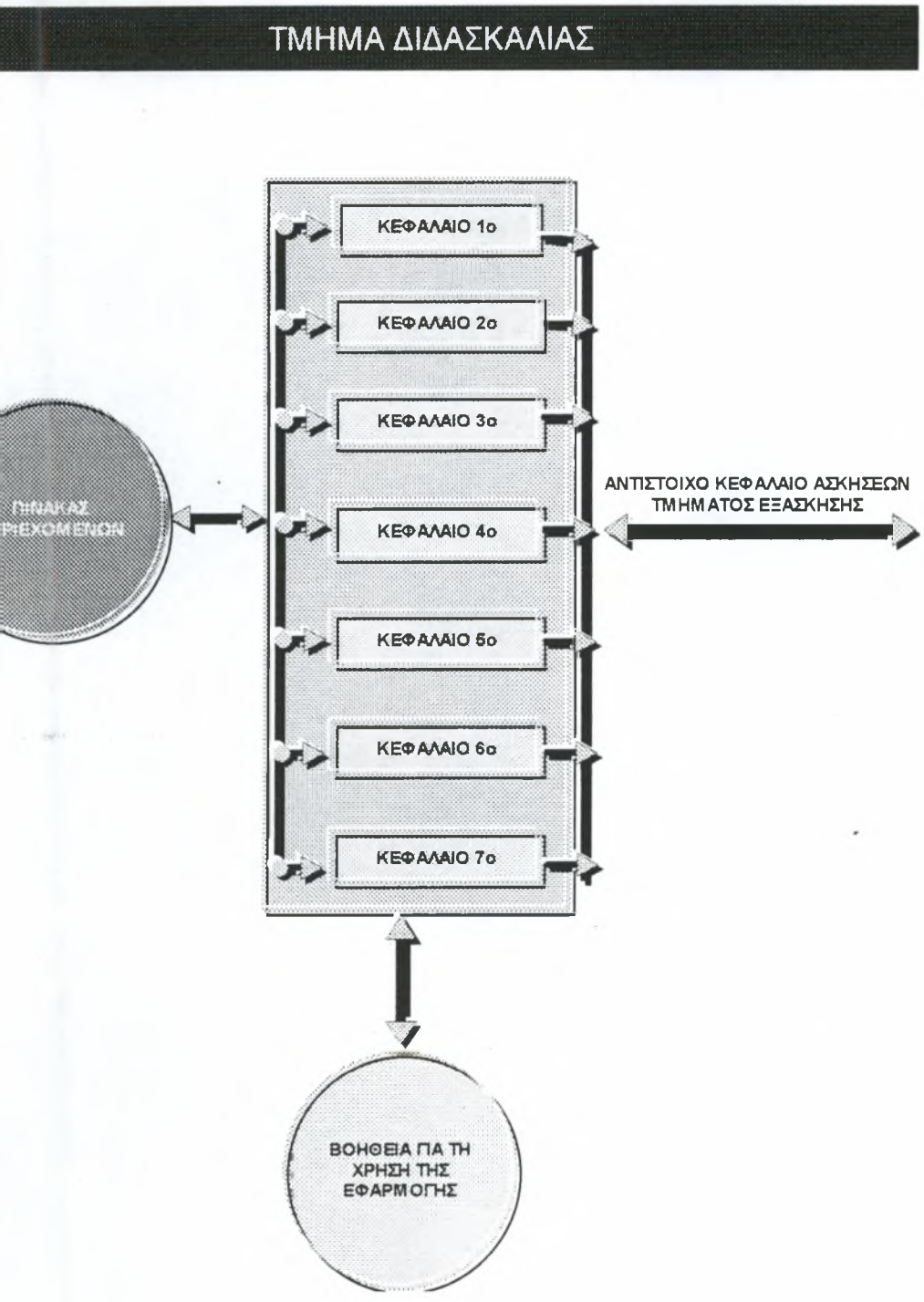
Το βιβλίο AVIHELP1.TBK που περιέχει το περιβάλλον της βοήθειας για το τμήμα της αξιολόγησης.

Πέρα από τα αρχεία του Toolbook, υπάρχει ένα πλήθος άλλων αρχείων υποστήριξης της εφαρμογής, όπως τα αρχεία βάσεων δεδομένων που χρησιμοποιεί το τμήμα αξιολόγησης και τα οποία αναλύονται διεξοδικά στην παράγραφο παρουσίασης της δομής του τμήματος αυτού, και τα αρχεία ήχου που περιέχουν ακουστικές πληροφορίες για συγκεκριμένα τμήματα της θεωρίας. Τα αρχεία αυτά είναι τα EK1.WAV-EK12.WAV καθώς και το BLKBRD.WAV που περιέχει τον ήχο της κιμωλίας που γράφει σε πίνακα (βλ. Κεφάλαιο 3 «Η χρήση της Ψηφιακής Σχεδίασης»).

Επίσης, τα αρχεία video που είναι τα UNIT1.AVI - UNIT7.AVI που περιλαμβάνουν τη γενική περιγραφή των περιεχομένων καθενός κεφαλαίου της θεωρίας, αλλά και τα αρχεία video του περιβάλλοντος βοήθειας του τμήματος αξιολόγησης τα οποία είναι τα εξής: AUTOTEST.AVI, CREATE.AVI, FIND.AVI, GOTO.AVI, LESSON.AVI, MAINMENU.AVI, MANUALTEST.AVI, NTYPE.AVI, OPEN.AVI, QUESTION.AVI, RENAME.AVI, STOIXEIA.AVI, και TEST.AVI.

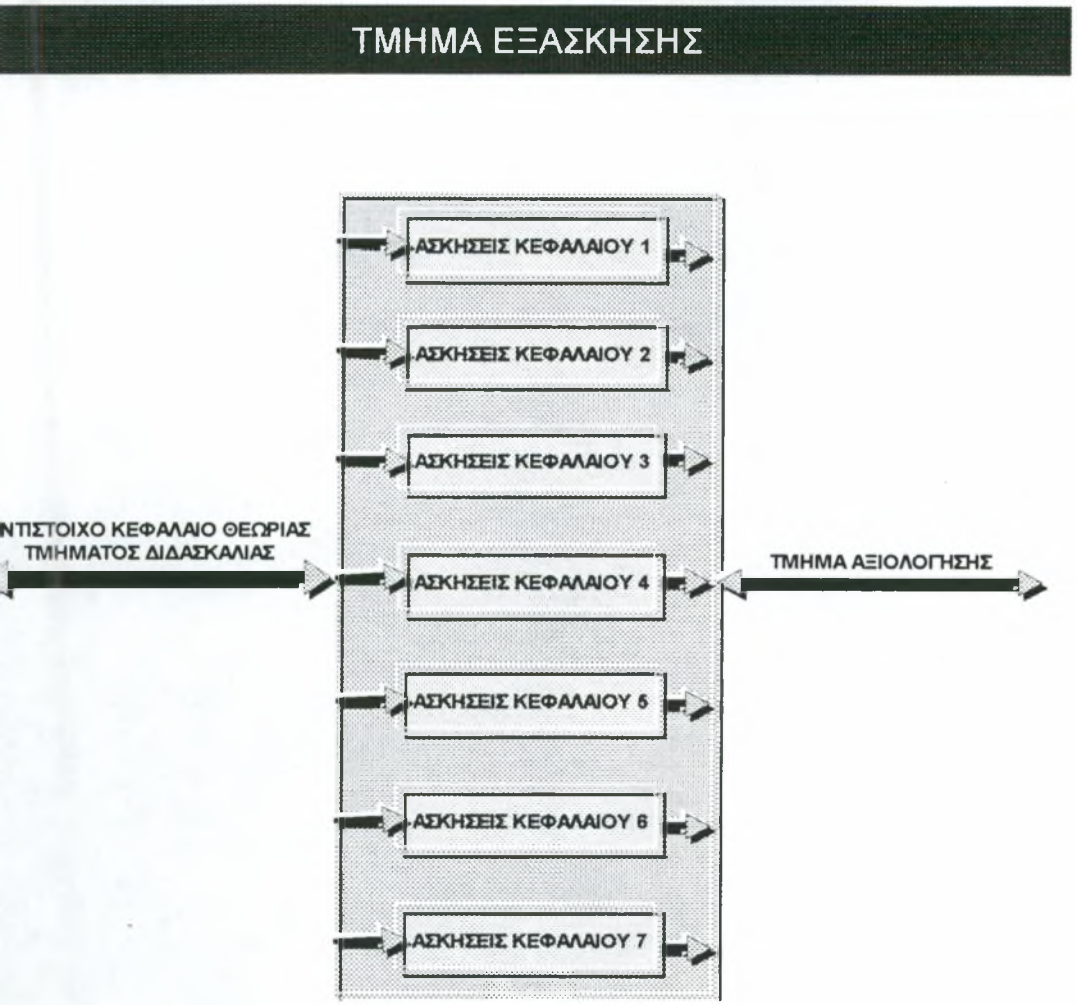
Τέλος, την εφαρμογή συνδράμουν και 35 αρχεία MIDI με ορχηστρική μουσική η οποία ακούγεται χαμηλόφωνα στο παρασκήνιο καθόλη τη διάρκεια «τρεξίματος» του τμήματος αξιολόγησης. Τα αρχεία αυτά είναι τα TT01.MID - TT35.MID.

2.2 Η ΔΟΜΗ ΤΟΥ ΤΜΗΜΑΤΟΣ ΔΙΔΑΣΚΑΛΙΑΣ



Διάγραμμα 1

2.3 ΔΟΜΗ ΤΟΥ ΤΜΗΜΑΤΟΣ ΕΞΑΣΚΗΣΗΣ



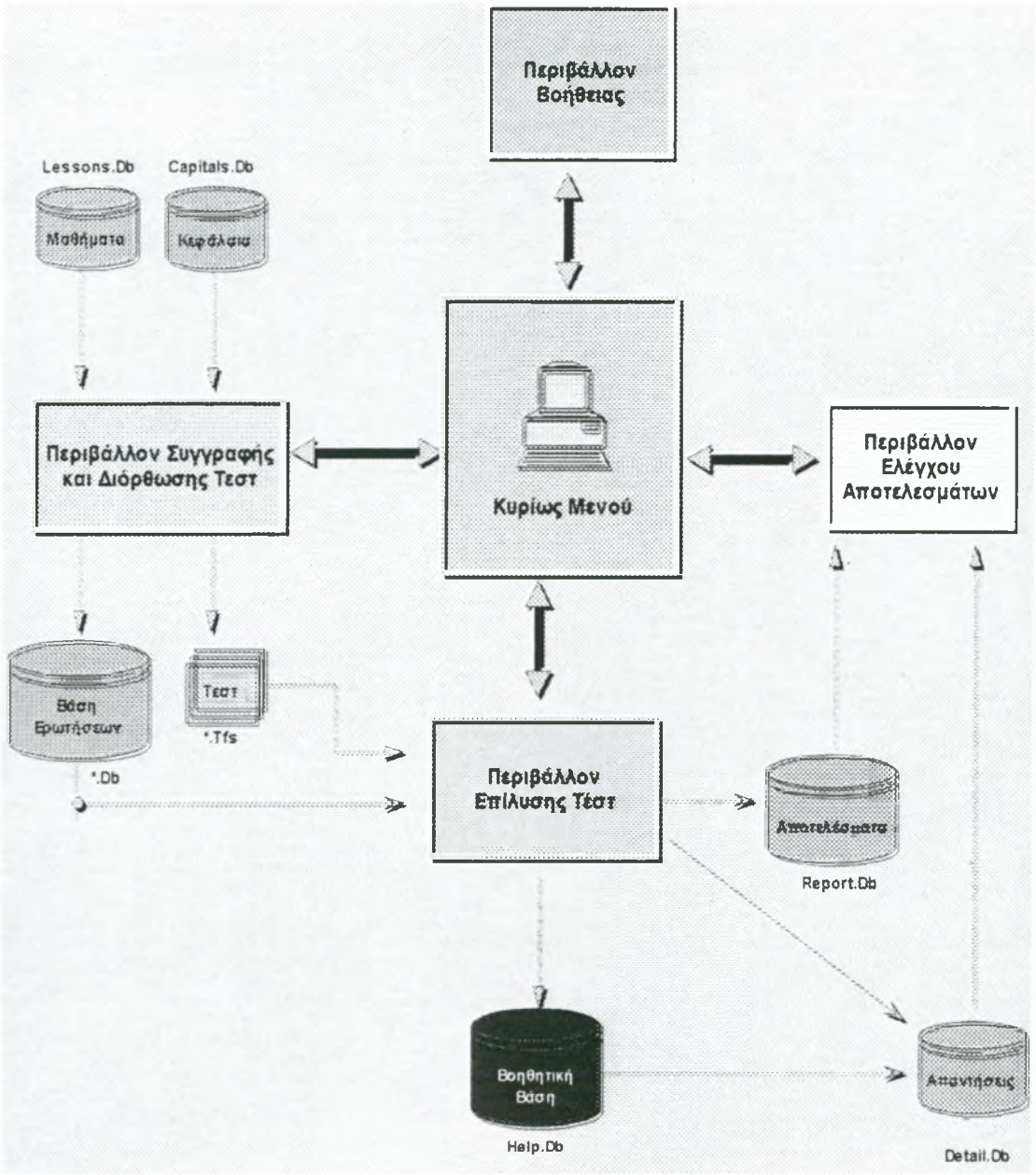
Διάγραμμα 2

Το τμήμα εξάσκησης (βλ. Διάγραμμα 2) στο μάθημα των Ψηφιακών εκτρονικών πάνω στη διδαχθείσα ύλη του τμήματος διδασκαλίας ριλαμβάνει και αυτό 7 κεφάλαια με ασκήσεις, που αντιστοιχούν το καθένα ένα βιβλίο - αρχείο του Toolbook. Όπως και το τμήμα διδασκαλίας, έτσι και τμήμα εξάσκησης, επιχειρεί να αναπτύξει τις δεξιότητες των ταιδευομένων στην επίλυση ασκήσεων επί των θεμάτων που διδάχθηκαν η θεωρία, κεντρίζοντας το ενδιαφέρον τους και αξιοποιώντας τις νατότητες των multimedia στοιχείων που χρησιμοποιεί (κυρίως animation).

Κάθε άσκηση καταλαμβάνει το χώρο μίας σελίδας του Toolbook (βλ. ράρτημα) και δίνει τη δυνατότητα στον εκπαιδευόμενο να παρακολουθήσει λύση της προτεινόμενης άσκησης κατά βήματα, ρίχνοντας το βάρος της ρουσίασης στον τρόπο δόμησης και στα διαδοχικά βήματα επίλυσης μέχρι τελικό αποτέλεσμα.

Οι ασκήσεις κάθε κεφαλαίου δομούνται σε κατηγορίες ανάλογα με το ματικό περιεχόμενο και το τμήμα της ύλης στο οποίο αναφέρονται. Οι

2.4 Η ΔΟΜΗ ΤΟΥ ΤΜΗΜΑΤΟΣ ΑΞΙΟΛΟΓΗΣΗΣ



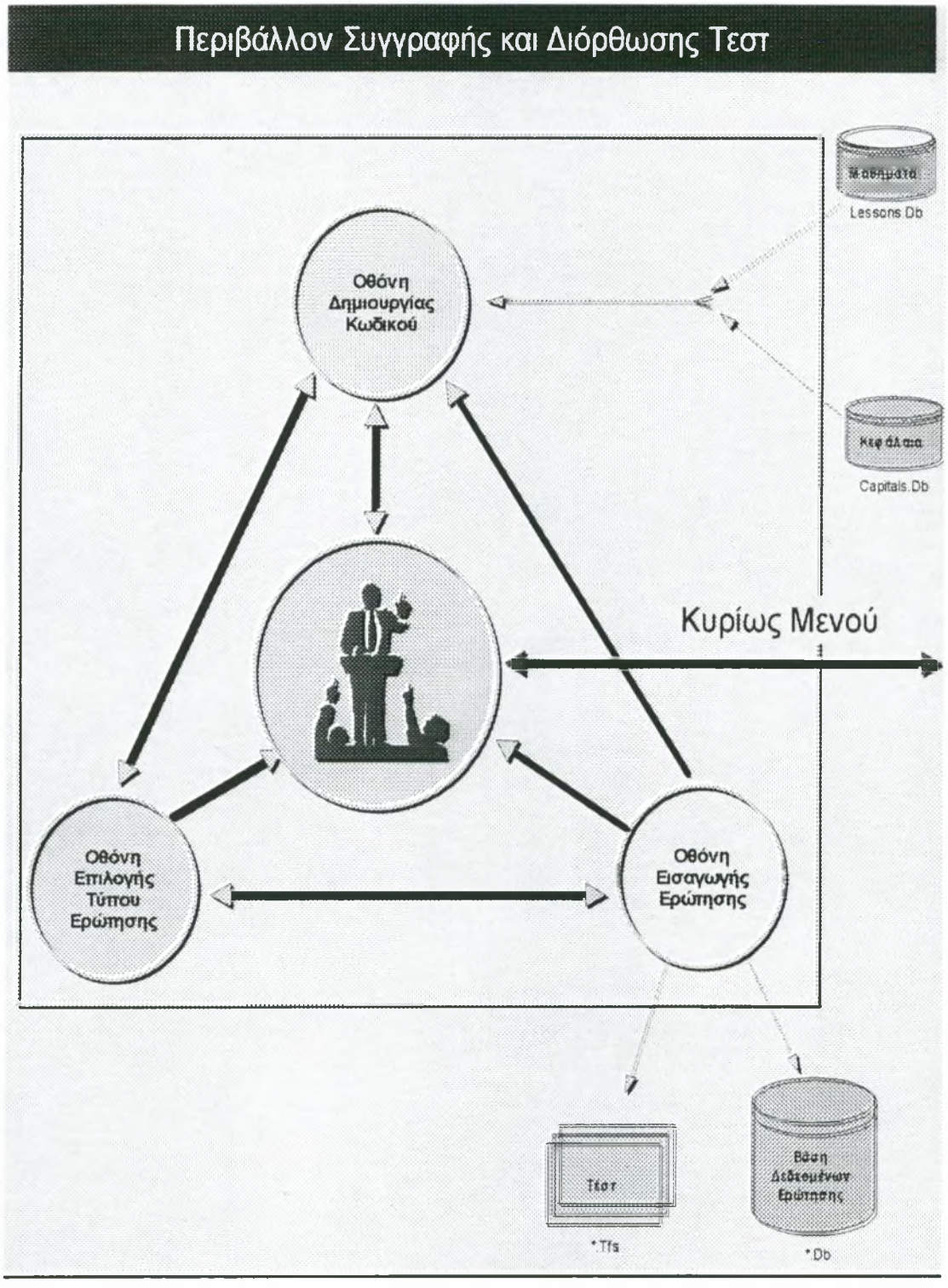
Διάγραμμα 3

Στο Διάγραμμα 3 φαίνεται η δομή του Τμήματος Αξιολόγησης [3]. Εκτός από το Κυρίως Μενού και τα τέσσερα Περιβάλλοντα στα οποία έχει πρόσβαση ο χρήστης (Καθηγητής ή Μαθητής), στο διάγραμμα φαίνονται και όλες οι Βάσεις Δεδομένων και τα Αρχεία που χρησιμοποιούνται για είσοδο και έξοδο.

Επίσης, στο ίδιο διάγραμμα παρουσιάζεται και ο τρόπος σύνδεσης των στοιχείων, και η ροή πληροφοριών και δεδομένων μεταξύ του Τμήματος και των Βάσεων Δεδομένων.

2.4.1 Περιβάλλον Συγγραφής και Διόρθωσης Τεστ

Στο παρακάτω διάγραμμα (Διάγραμμα 4) μπορούμε να δούμε την εσωτερική δομή του Περιβάλλοντος Συγγραφής και Διόρθωσης Τεστ.



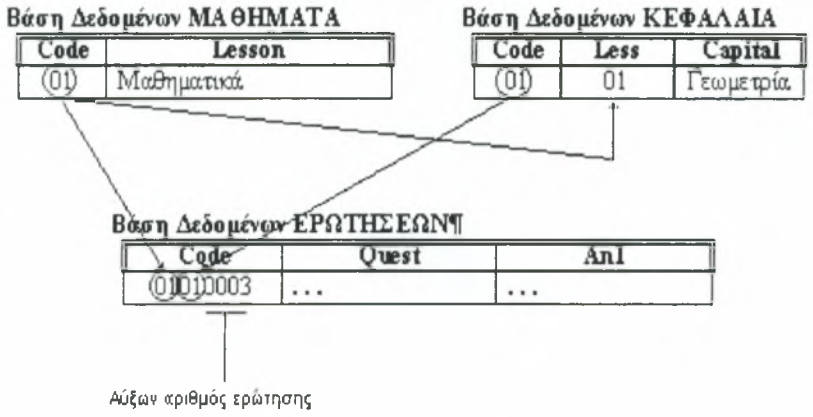
Διάγραμμα 4

Στο Περιβάλλον αυτό ο καθηγητής δημιουργεί μία Βάση Ερωτήσεων με σκοπό την δημιουργία ενός Τεστ στο οποίο θα εξεταστεί ο Μαθητής.

Δημιουργία Κωδικού ερώτησης

Οι κωδικοί του μαθήματος και του κεφαλαίου που επιλέγει ο καθηγητής αποτελούν τα 4 πρώτα ψηφία του Κωδικού της ερώτησης (πεδίο Code της Βάσης Δεδομένων ΕΡΩΤΗΣΕΩΝ) (βλέπε Διάγραμμα 5)

- μεταβαίνει στην **Οθόνη Επιλογής Τύπου Ερώτησης**



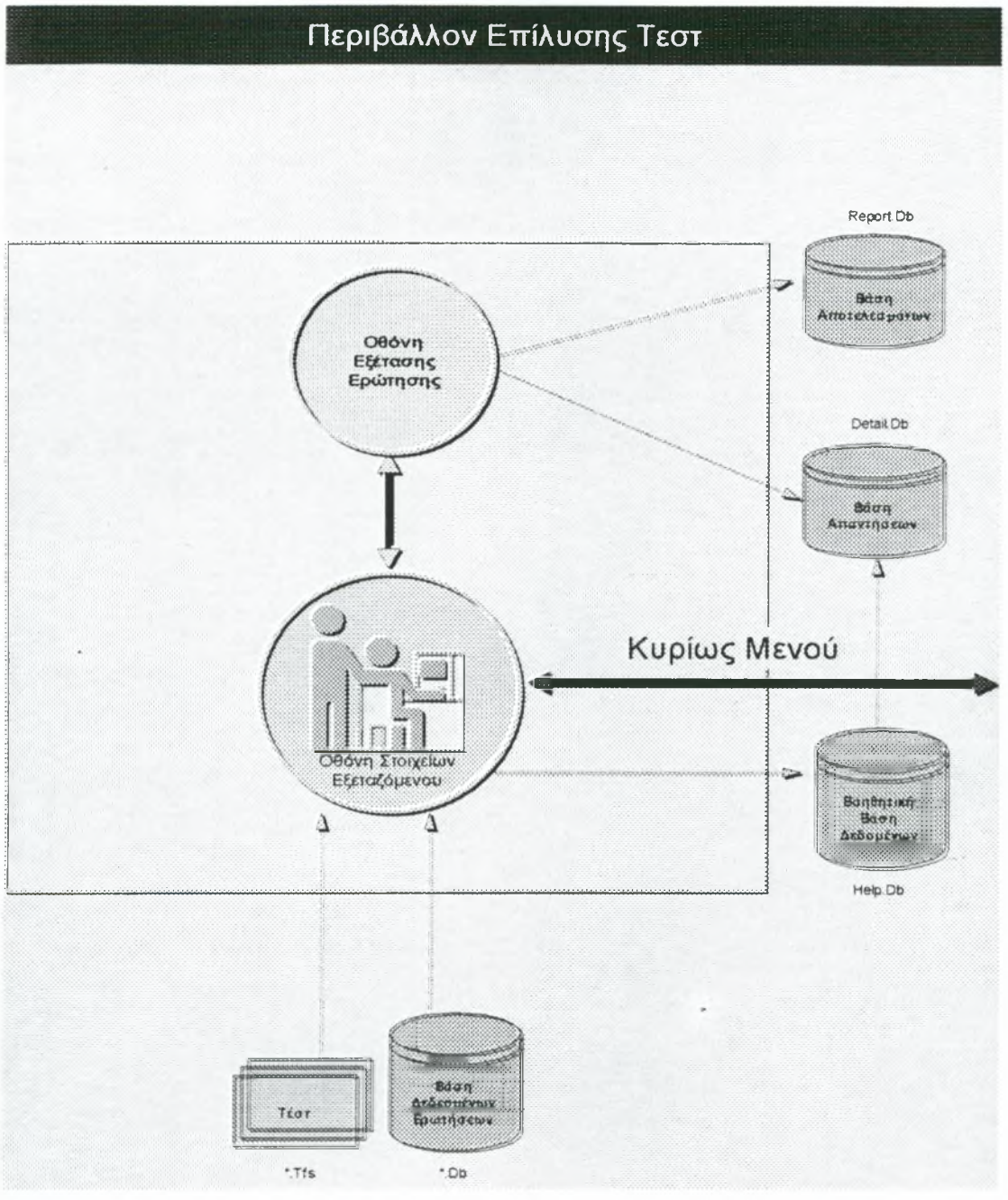
Διάγραμμα 5

Επιλογή Τύπου Ερώτησης

Ανάλογα με τον τύπο της ερώτησης που επιλέγει ο καθηγητής το πεδίο Type της Βάσης Δεδομένων ΕΡΩΤΗΣΕΩΝ λαμβάνει την τιμή 1, 2 ή 3.

2.4.2 Περιβάλλον Επίλυσης Τεστ

Στο ακόλουθο διάγραμμα (Διάγραμμα 6) φαίνεται η εσωτερική δομή του Περιβάλλοντος Επίλυσης Τεστ.



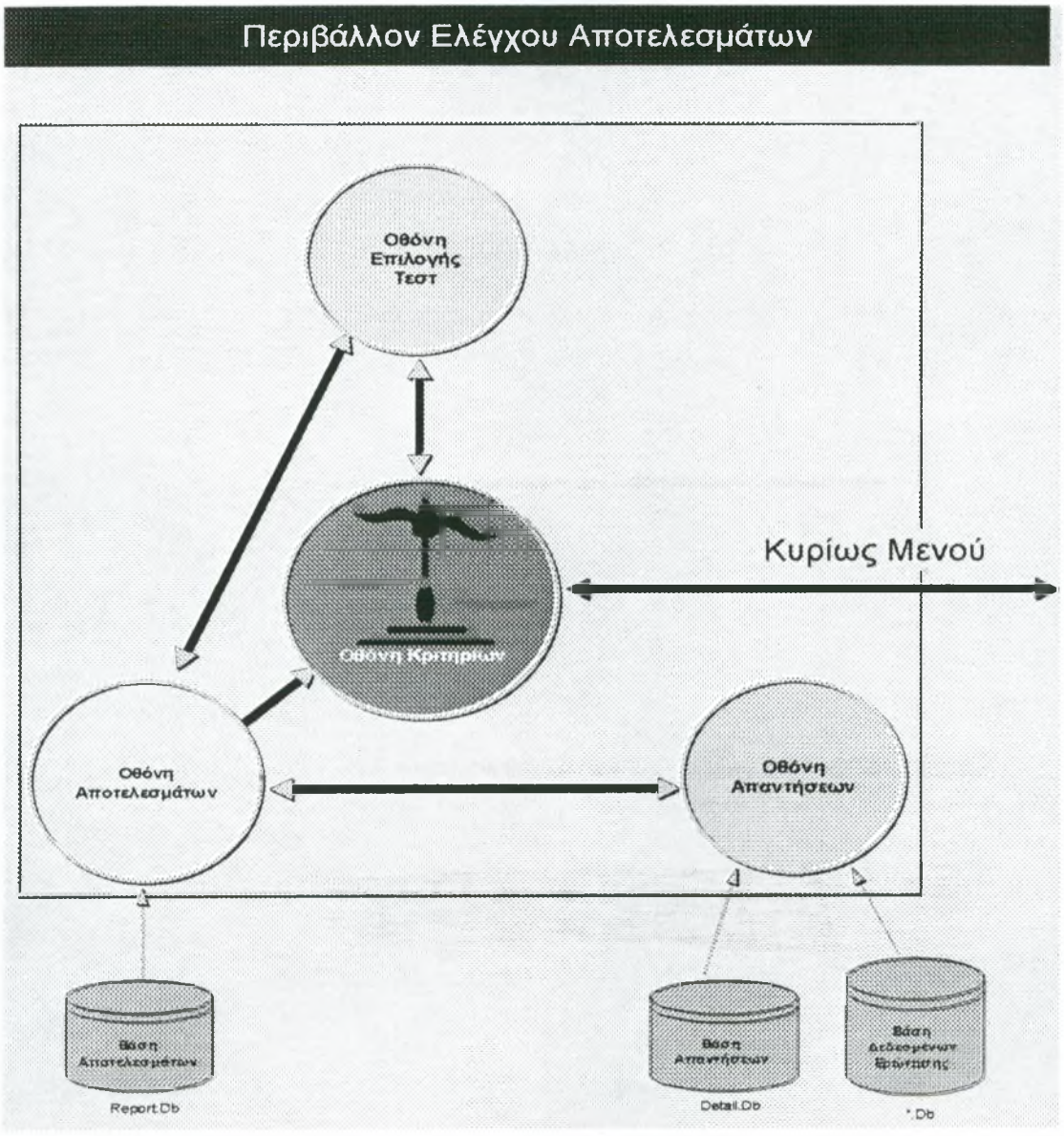
Διάγραμμα 6

Με το πέρας της εξέτασης του Τεστ ενημερώνονται άμεσα δύο Βάσεις Δεδομένων, η Βάση Δεδομένων ΑΠΟΤΕΛΕΣΜΑΤΑ, και η Βάση Δεδομένων ΑΠΑΝΤΗΣΕΙΣ. Τα στοιχεία που αποθηκεύονται στην πρώτη δεν είναι παρά τα στοιχεία του εξεταζόμενου, τα στοιχεία του Τεστ, και ο βαθμός επιτυχίας του Μαθητή.

Η ενημέρωση της δεύτερης Βάσης Δεδομένων είναι πιο χρονοβόρα. Ουσιαστικά δηλαδή η Βάση Δεδομένων ΑΠΑΝΤΗΣΕΙΣ είναι ένα μερικό αντίγραφο της ΒΟΗΘΗΤΙΚΗΣ Βάσης Δεδομένων (βλέπε Διάγραμμα 8).

2.4.3 Περιβάλλον Ελέγχου Αποτελεσμάτων

Στο ακόλουθο διάγραμμα (Διάγραμμα 7) φαίνεται η εσωτερική δομή του Περιβάλλοντος Ελέγχου Αποτελεσμάτων. Στο περιβάλλον αυτό ο καθηγητής ελέγχει την απόδοση του εξεταζόμενου.



Διάγραμμα 7

2.4.4 Η Ροή Πληροφοριών του Τμήματος Αξιολόγησης

Οι Βάσεις Δεδομένων και τα αντίστοιχα αρχεία που χρησιμοποιούνται είναι:

Η Βάση Δεδομένων ΜΑΘΗΜΑΤΑ (αρχείο LESSONS.DB)

Χρησιμοποιείται κατά την αναζήτηση κάποιας ερώτησης από τον Καθηγητή. Κάθε ερώτηση που δημιουργείται πρέπει οπωσδήποτε να ανήκει σε κάποιο Μάθημα. Τα πεδία (fields) που έχει η Βάση Δεδομένων είναι:
Code: Διψήφιος αριθμός που δηλώνει τον κωδικό του κάθε μαθήματος.
Lesson: Το όνομα του κάθε μαθήματος (μέγεθος: 30 χαρακτήρες)

Η Βάση Δεδομένων ΚΕΦΑΛΑΙΑ (αρχείο CAPITALS.DB)

Η Βάση αυτή περιέχει τα Κεφάλαια του κάθε Μαθήματος. Κάθε ερώτηση που δημιουργείται πρέπει οπωσδήποτε να ανήκει και σε κάποιο Κεφάλαιο (πέρα από το Μάθημα). Τα πεδία που περιέχει η Βάση Δεδομένων είναι:
Code: Διψήφιος αριθμός. Είναι ο κωδικός του κάθε Κεφαλαίου.
Less: Διψήφιος αριθμός. Είναι ο κωδικός του Μαθήματος στο οποίο ανήκει το Κεφάλαιο
Capital: Είναι το όνομα του Κεφαλαίου (μέγεθος: 30 χαρακτήρες)

Η Βάση Δεδομένων των ΕΡΩΤΗΣΕΩΝ (αρχείο *.DB)

Είναι ουσιαστικά η κυριώτερη Βάση Δεδομένων μιας και χρησιμοποιείται απ' όλους τους χρήστες και όλα τα περιβάλλοντα. Συγκεκριμένα:

- Η βάση δημιουργείται και διορθώνεται από τον Καθηγητή στο Περιβάλλον Συγγραφής και Διόρθωσης Τεστ.
- Ο Μαθητής εξετάζεται σε ένα τμήμα της στο Περιβάλλον Επίλυσης Τεστ.
- Η βάση χρησιμοποιείται για την εξαγωγή συμπερασμάτων, από τον Καθηγητή όσον αφορά το επίπεδο γνώσης του Μαθητή, στο Περιβάλλον Ελέγχου Αποτελεσμάτων.

Στη βάση αυτή περιέχονται όλες οι ερωτήσεις που έχουν εισαχθεί κατά καιρούς από τον Καθηγητή. Για κάθε ερώτηση υπάρχουν πέντε πιθανές απαντήσεις που παρουσιάζονται στον Μαθητή ώστε αυτός να διαλέξει την σωστή (ή τις σωστές). Τα πεδία που περιέχει είναι:
Code: 8ψήφιος αριθμός. Είναι ο κωδικός της ερώτησης με τον οποίο γίνονται και όλες οι αναζητήσεις από τον Καθηγητή.
Quest: Περιέχει το κείμενο της ερώτησης (μέγεθος: 100 χαρακτήρες)
An1: Το κείμενο της πρώτης πιθανής απάντησης στη συγκεκριμένη ερώτηση (μέγεθος: 100 χαρακτήρες).
An2: Το κείμενο της δεύτερης πιθανής απάντησης στη συγκεκριμένη ερώτηση (μέγεθος: 100 χαρακτήρες).
An3: Το κείμενο της τρίτης πιθανής απάντησης στη συγκεκριμένη ερώτηση (μέγεθος: 100 χαρακτήρες).
An4: Το κείμενο της τέταρτης πιθανής απάντησης στη συγκεκριμένη ερώτηση (μέγεθος: 100 χαρακτήρες).
An5: Το κείμενο της πέμτπης πιθανής απάντησης στη συγκεκριμένη ερώτηση (μέγεθος: 100 χαρακτήρες).
Type: Διψήφιος αριθμός. Δηλώνει τον τύπο της ερώτησης.
Dif: Είναι ο βαθμός δυσκολίας της κάθε ερώτησης (μέγεθος: 3 χαρακτήρες). Παίρνει τιμές από 10-100.

Στο πεδίο της κάθε πιθανής απάντησης αποθηκεύεται και ο χαρακτηρισμός της, δηλαδή αν είναι σωστή (true) ή λάθος (false). Αυτό

επιτυγχάνεται χρησιμοποιώντας σαν πρώτο χαρακτήρα του πεδίου το γράμμα T (True) ή το γράμμα F (False).

Η ΒΟΗΘΗΤΙΚΗ Βάση Δεδομένων (αρχείο HELP.DB)

Πρόκειται για ένα μερικό αντίγραφο της Βάσης Δεδομένων των ΕΡΩΤΗΣΕΩΝ. Κάθε φορά δηλαδή που κάποιος μαθητής λύνει κάποιο τεστ στο Περιβάλλον Επίλυσης Τεστ, αντιγράφονται από την Βάση των ΕΡΩΤΗΣΕΩΝ στη ΒΟΗΘΗΤΙΚΗ Βάση μόνο εκείνες οι ερωτήσεις που πρέπει να απαντήσει ο Μαθητής. Μετά την αντιγραφή συμπληρώνονται τα πεδία των πιθανών απαντήσεων (An1, An2, ...An5) με τον χαρακτήρα F (false) στο τέλος τους. Τα πεδία που έχει αυτή η Βάση Δεδομένων είναι ακριβώς τα ίδια με την Βάση των ΕΡΩΤΗΣΕΩΝ (βλέπε Διάγραμμα 8).

Η Βάση Δεδομένων ΑΠΟΤΕΛΕΣΜΑΤΑ (αρχείο REPORT.DB)

Μετά την εξέταση του Μαθητή στο Περιβάλλον Επίλυσης του Τεστ ενημερώνεται η Βάση ΑΠΟΤΕΛΕΣΜΑΤΑ με την επίδοση του Μαθητή στο συγκεκριμένο Τεστ. Πέρα από τα στοιχεία του Μαθητή και τον Βαθμό Επιτυχίας του στο Τεστ, σ' αυτή την Βαση αποθηκεύονται και άλλα στοιχεία όπως το όνομα της Βασης Ερωτήσεων, το όνομα του Τεστ ή Ημερομηνία Εξέτασης κ.λ.π. Όλα αυτά τα στοιχεία μπορούν να ελεγχθούν στο Περιβάλλον Ελέγχου Αποτελεσμάτων από τον Καθηγητή. Τα πεδία που περιέχει αυτή η Βάση είναι:

IdTest: 4ψήφιος αριθμός. Αποτελεί τον αύξοντα αριθμό του Τεστ. Δίνεται στο Τεστ τη στιγμή που δημιουργείται. Συνδέει την συγκεκριμένη Βάση με την Βάση ΑΠΑΝΤΗΣΕΙΣ.

Lname: Το επώνυμο του Εξεταζόμενου (μέγεθος: 30 χαρακτήρες).

Fname: Το μικρό όνομα του Εξεταζόμενου (μέγεθος: 30 χαρακτήρες).

Code: Ο κωδικός του Εξεταζόμενου (μέγεθος: 12 χαρακτήρες).

TestName: Το όνομα (του αρχείου με το πλήρες μονοπάτι του) του Τεστ (μέγεθος: 30 χαρακτήρες).

DbName: Το όνομα της Βάσης ΕΡΩΤΗΣΕΩΝ (του αρχείου με το πλήρες μονοπάτι του (μέγεθος: 30 χαρακτήρες)

DateExam: Η ημερομηνία εξέτασης (με μορφή dd/mm/yyyy) (μέγεθος: 10 χαρακτήρες).

Degree: Ο Βαθμός επιτυχίας του Εξεταζόμενου στο Τεστ. Παίρνει τιμές από 0 - 100% (μέγεθος: 6 χαρακτήρες).

Dif: Ο μέσος Βαθμός Δυσκολία του Τεστ. Παίρνει τιμές από 0 - 100 (μέγεθος: 6 χαρακτήρες).

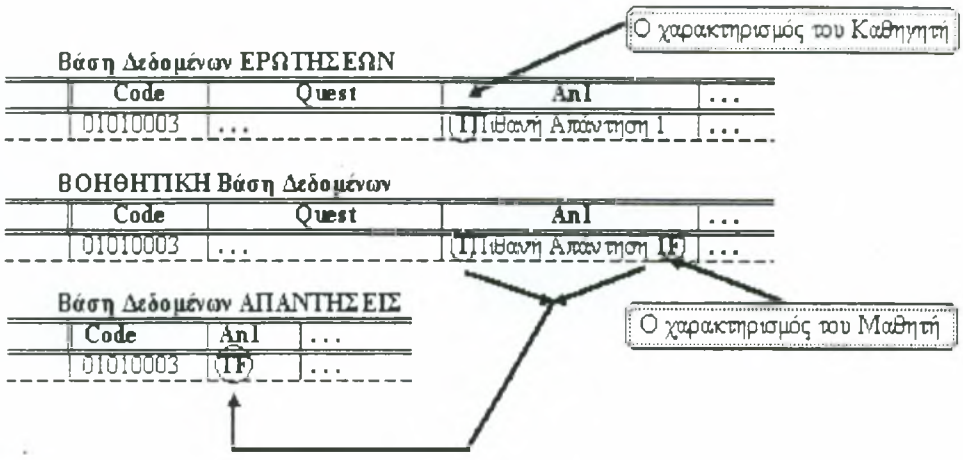
TotalTime: Ο συνολικός χρόνος που ορίστηκε για την επίλυση του Τεστ (μέγεθος: 4 χαρακτήρες).

SpendTime: Ο χρόνος που ξοδεύτηκε από τον Εξεταζόμενο για το Τεστ (μέγεθος: 4 χαρακτήρες).

Οι Βάση Δεδομένων ΑΠΑΝΤΗΣΕΙΣ (αρχείο DETAIL.DB)

Δημιουργείται στο Περιβάλλον Επίλυσης του Τεστ με σκοπό τον έλεγχο των αδυναμιών του Μαθητή από τον Καθηγητή. Σ' αυτήν αποθηκεύονται όλες οι ερωτήσεις (με την μορφή κωδικού) του Τεστ και οι χαρακτηρισμοί (T ή F) όλων των πιθανών απαντήσεων. Αποθηκεύονται, και οι χαρακτηρισμοί που

έχει δώσει ο Καθηγητής αλλά και αυτοί που έχει δώσει ο Μαθητής, με σκοπό την περαιτέρω σύγκριση. Με άλλα λόγια, μετά το πέρας του Τεστ, αντιγράφονται στα πεδία των απαντήσεων της Βάσης ΑΠΑΝΤΗΣΕΙΣ ο πρώτος χαρακτήρας (Τ ή F) (που έχει ορίσει ο Καθηγητής) και ο τελευταίος (Τ ή F των απαντήσεων του Μαθητή) των πεδίων της ΒΟΗΘΗΤΙΚΗΣ Βάσης (βλέπε Διάγραμμα 8).



Διάγραμμα 8

Τα πεδία που περιέχει η Βάση Δεδομένων ΑΠΑΝΤΗΣΕΙΣ είναι:

IdTest: Πρόκειται για το ίδιο πεδίο της Βάσης Δεδομένων Αποτελέσματα, δηλαδή ο αύξων αριθμός του Τεστ. (μέγεθος: 4 χαρακτήρες).

Code: Ο κωδικός της κάθε ερώτησης (μέγεθος: 8 χαρακτήρες).

An1: Ο χαρακτηρισμός της πρώτης πιθανής απάντησης (Τ ή F) από τον Καθηγητή και ο χαρακτηρισμός της πρώτης πιθανής απάντησης (Τ ή F) από τον Μαθητή (μέγεθος: 2 χαρακτήρες) (βλέπε Διάγραμμα 8).

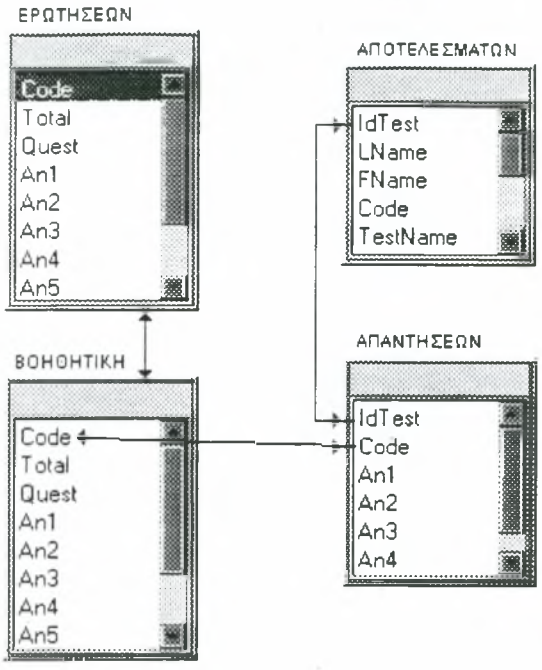
An2: όμοια με το πεδίο An1.

An3: όμοια με το πεδίο An1.

An4: όμοια με το πεδίο An1.

An5: όμοια με το πεδίο An1.

Στο Διάγραμμα 9 που ακολουθεί βλέπουμε τις σχέσεις των τεσσάρων κυριοτέρων Βάσεων Δεδομένων που χρησιμοποιεί το τμήμα αξιολόγησης.



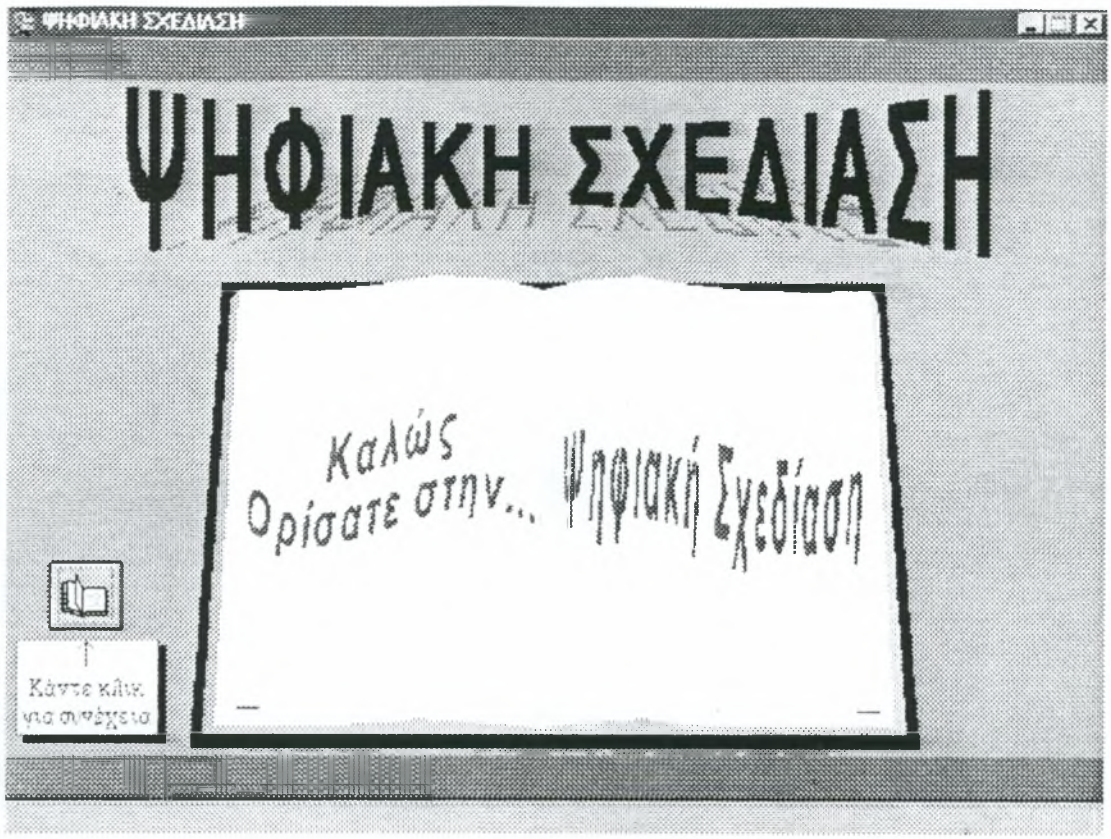
Διάγραμμα 9

<u>Το αρχείο του Τεστ (αρχείο με επέκταση *.TFS)</u>

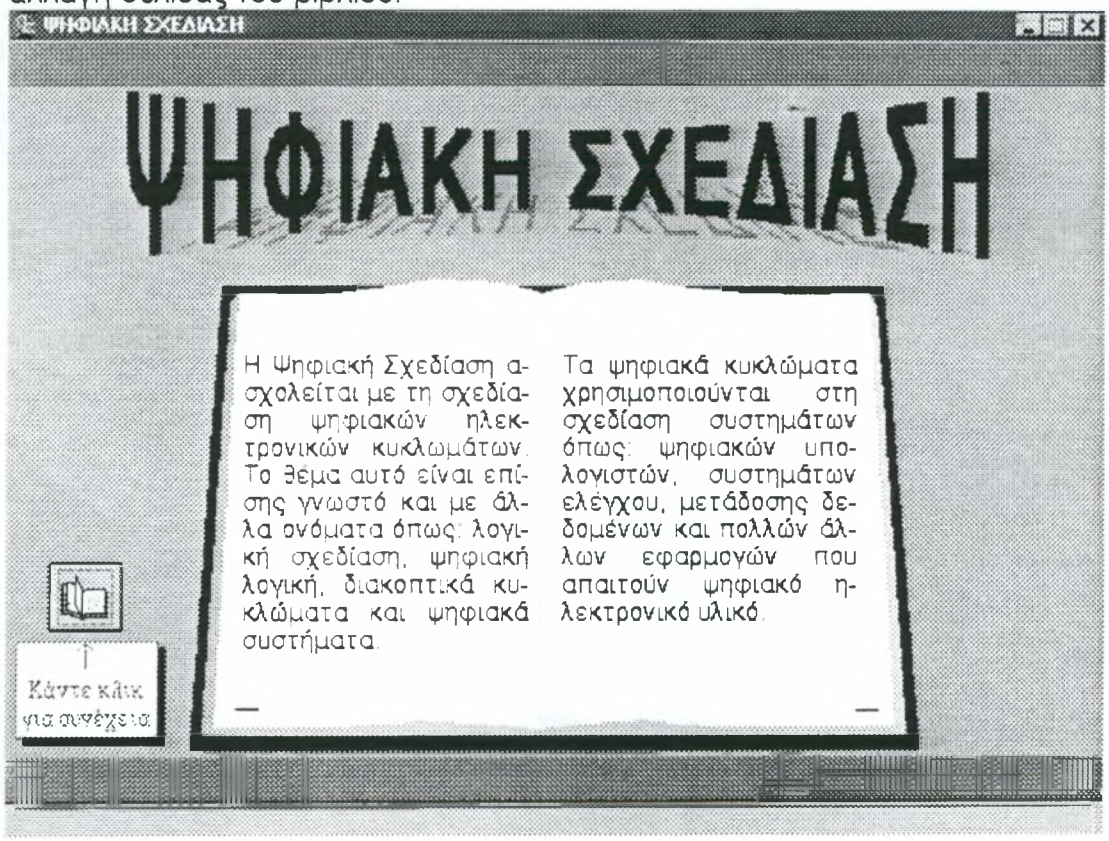
Αν και έχει τη δική του επέκταση (*.tfs) το αρχείο του τεστ δεν είναι παρά ένα αρχείο text (*.txt). Τα στοιχεία που περιέχει είναι:

- Το σύνολο των ερωτήσεων που έχουν επιλεγεί είτε τυχαία (από το πρόγραμμα) είτε χειροκίνητα (από τον καθηγητή) στις οποίες θα εξεταστεί ο μαθητής. Στο αρχείο του τεστ οι ερωτήσεις αποθηκεύονται με τη βοήθεια του κωδικού τους (των οκταψήφιο αυτό αριθμό της Βάσης Δεδομένων ΕΡΩΤΗΣΕΩΝ).
- Το όνομα της Βάσης (με το πλήρες μονοπάτι του αρχείου της Βάσης Δεδομένων ΕΡΩΤΗΣΕΩΝ).
- Τον Συνολικό Χρόνο Επίλυσης. Είναι ο χρόνος που έχει οριστεί από τον Καθηγητή σαν τον μέγιστο χρόνο Επίλυσης του Τεστ.

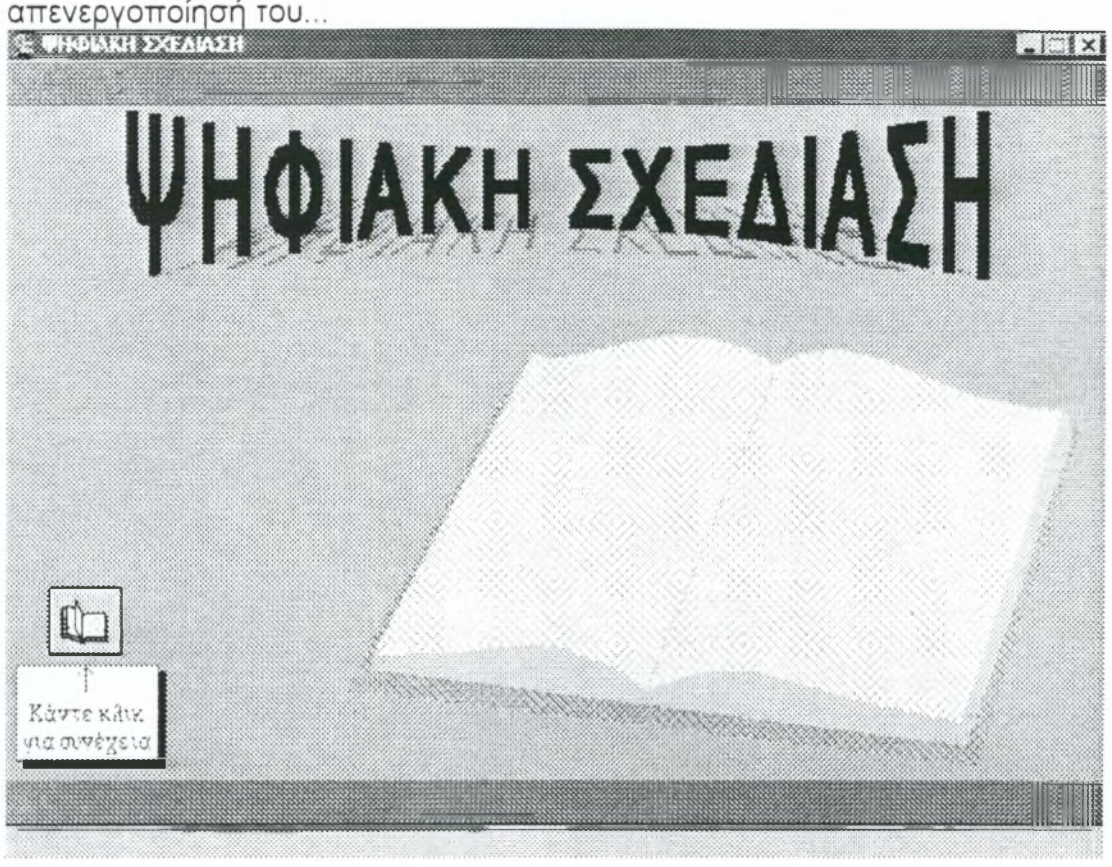
Τα στοιχεία στο αρχείο του Τεστ είναι τοποθετημένα στη σειρά και χωρίζονται με ένα κόμμα «,». Το αρχείο μπορεί να τροποποιηθεί από οποιοδήποτε επεξεργαστή κειμένου.



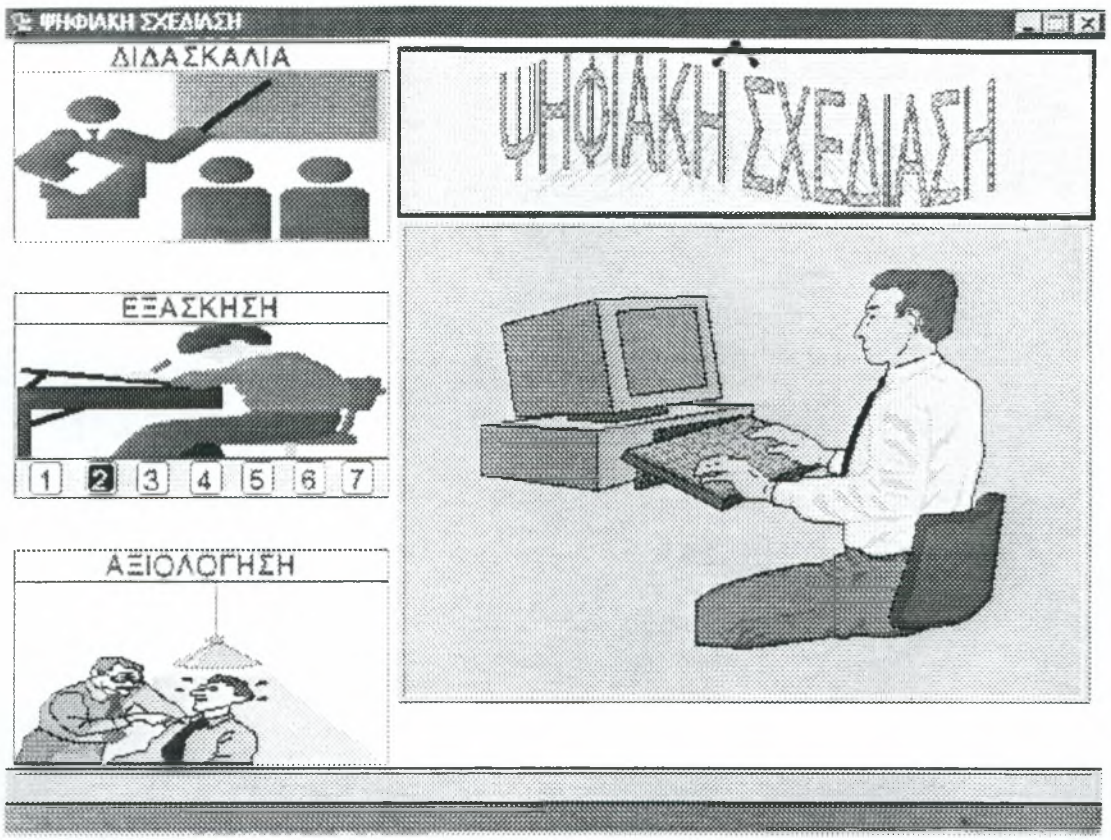
Αφού γίνει το καλωσόρισμα του χρήστη από την εφαρμογή, ο χρήστης προτρέπεται να κάνει κλικ στο ειδικό πλήκτρο για να «ξεφυλλίσει» και τις επόμενες σελίδες του βιβλίου. Με τη βοήθεια animation προσομοιώνεται η αλλαγή σελίδας του βιβλίου:



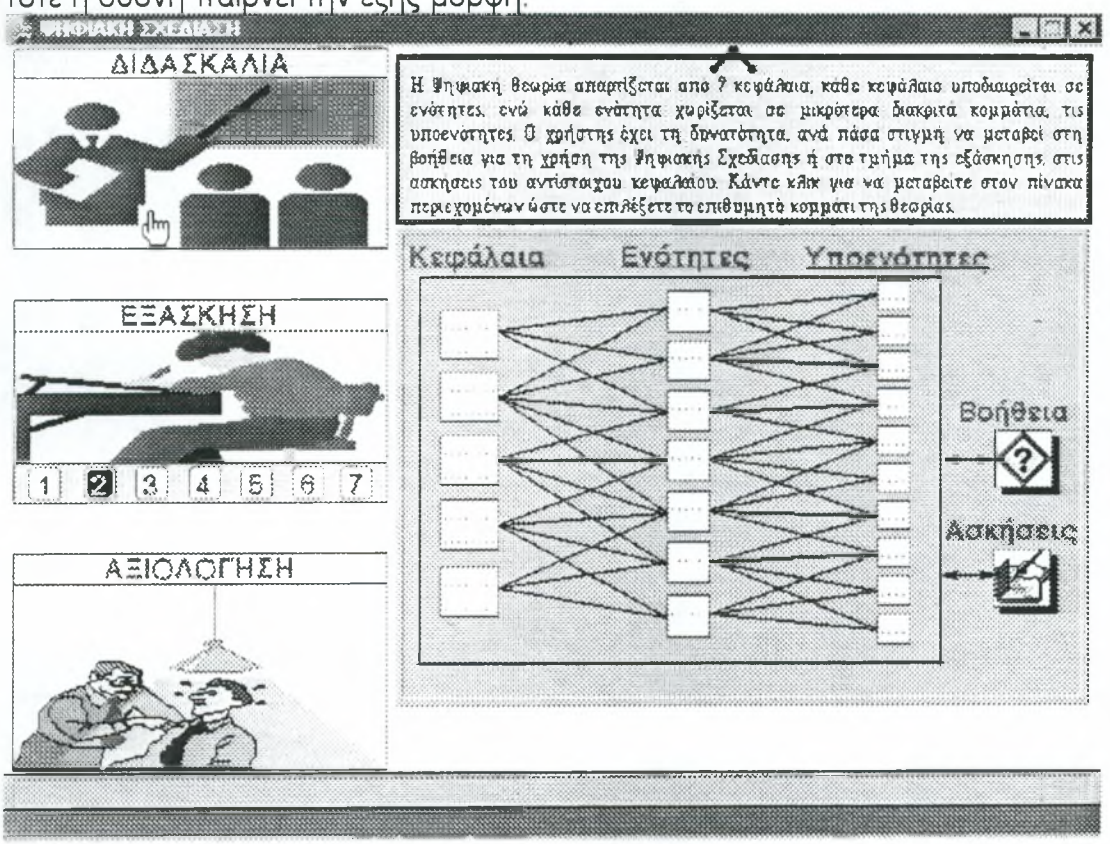
Τελικά, μετά την τοποθέτηση του βιβλίου στο παρασκήνιο και την απενεργοποίησή του...



ο χρήστης εισάγεται στην οθόνη επιλογής του τμήματος της εφαρμογής που επιθυμεί. Η οθόνη αυτή περιέχει στο αριστερό τμήμα της τρεις εικόνες που αντιστοιχούν η καθεμία στη διδασκαλία, την εξάσκηση και την αξιολόγηση δηλ. στα τρία τμήματα της Ψηφιακής Σχεδίασης.

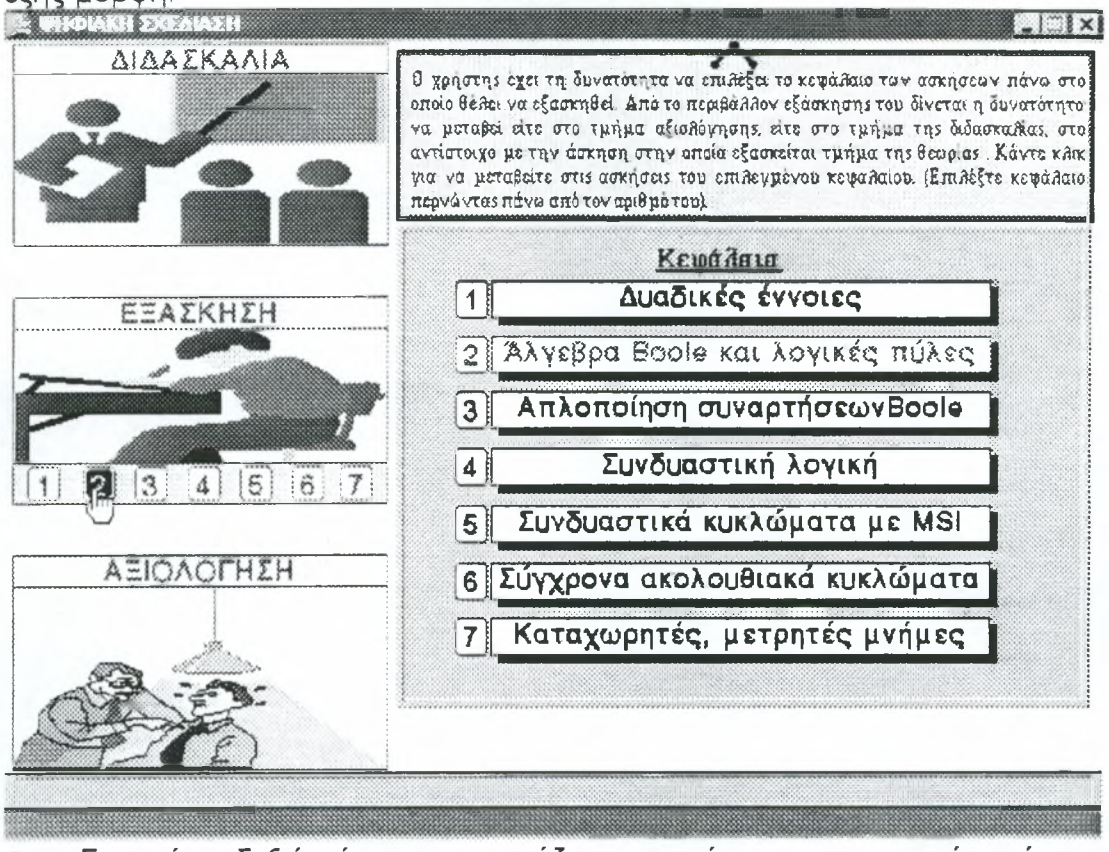


Μόλις το ποντίκι εισέλθει σε καθεμία από τις παραπάνω εικόνες του αριστερού τμήματος, τότε στο δεξιό τμήμα εμφανίζονται σχετικές παραστάσεις αντίστοιχα. Συγκεκριμένα, αν το ποντίκι εισέλθει στην εικόνα «ΔΙΔΑΣΚΑΛΙΑ» τότε η οθόνη παίρνει την εξής μορφή:



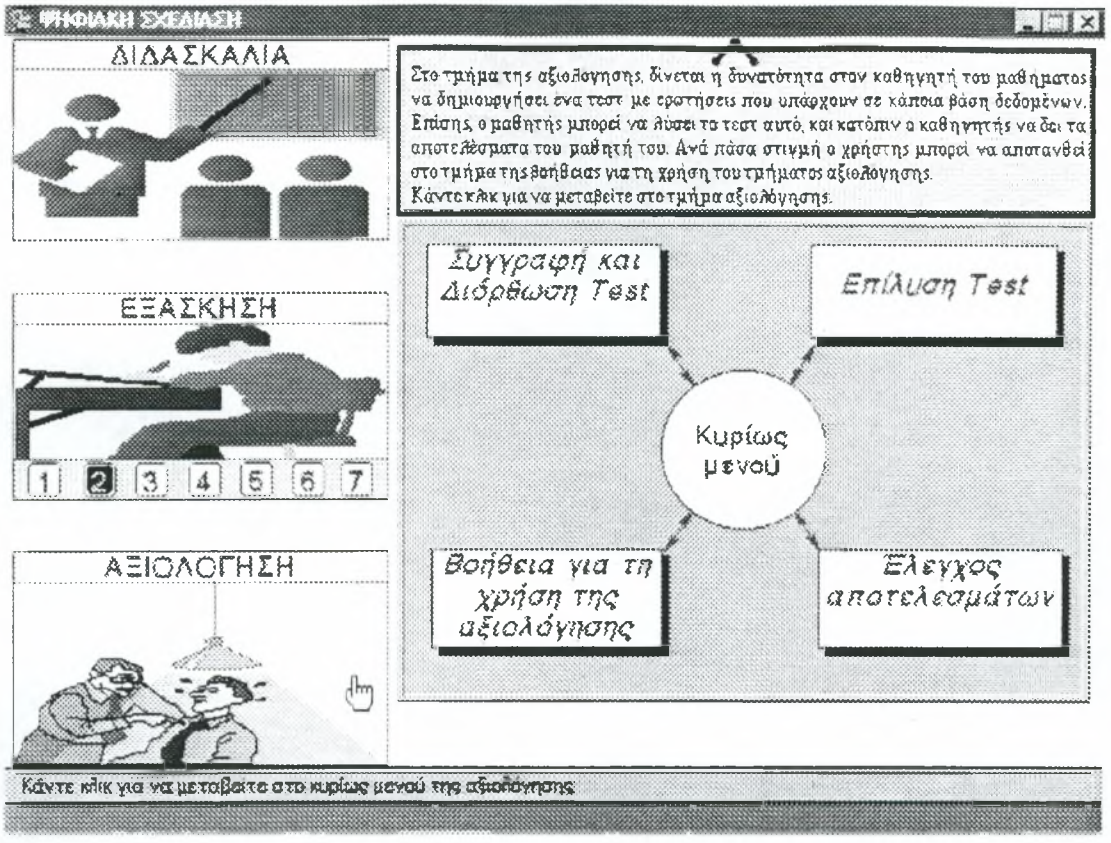
Στο πάνω δεξιό μέρος παρουσιάζεται μία σχετική περιγραφή, ενώ στο κάτω εμφανίζεται η διάρθρωση της θεωρίας σε κεφάλαια, ενότητες και υποενότητες. Κάνοντας κλικ, ο χρήστης μεταβαίνει στον πίνακα περιεχομένων όπου μπορεί να επιλέξει το επιθυμητό τμήμα της θεωρίας.

Αν το ποντίκι εισέλθει στην εικόνα «ΕΞΑΣΚΗΣΗ», η οθόνη λαμβάνει την εξής μορφή:



Στο πάνω δεξιό μέρος παρουσιάζεται μια σύντομη περιγραφή, ενώ στο κάτω τμήμα φαίνονται τα κεφάλαια της Ψηφιακής θεωρίας. Ο χρήστης μπορεί να επιλέξει το κεφάλαιο των ασκήσεων στο οποίο θέλει να εξασκηθεί περνώντας το ποντίκι πάνω από τον αριθμό του αντίστοιχου κεφαλαίου και κατόπιν, κάνοντας κλικ να μεταβεί στις ασκήσεις αυτού του κεφαλαίου.

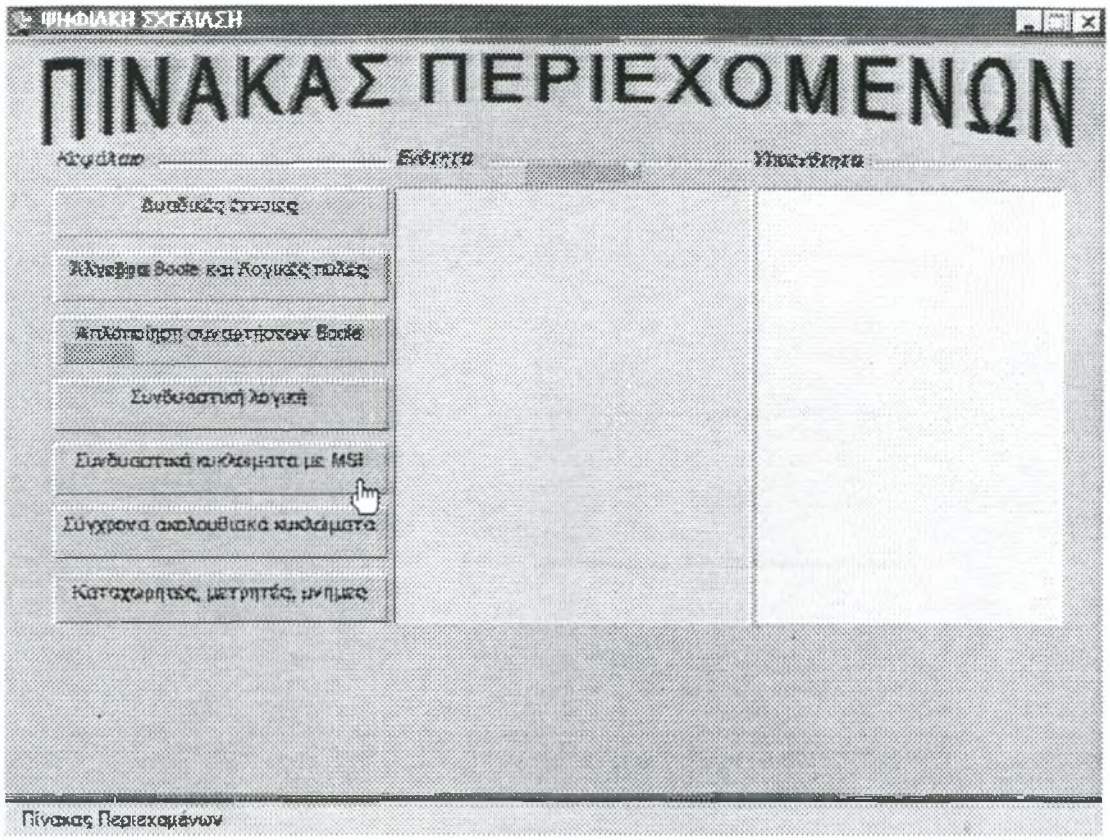
Αν το ποντίκι εισέλθει στην εικόνα «ΑΞΙΟΛΟΓΗΣΗ» τότε η οθόνη εμφανίζει στο δεξιό της τμήμα μια σύντομη περιγραφή και τα επιμέρους τμήματα από τα οποία αποτελείται το κομμάτι της αξιολόγησης. Κάνοντας κλικ, ο χρήστης μεταβαίνει στο τμήμα αυτό.



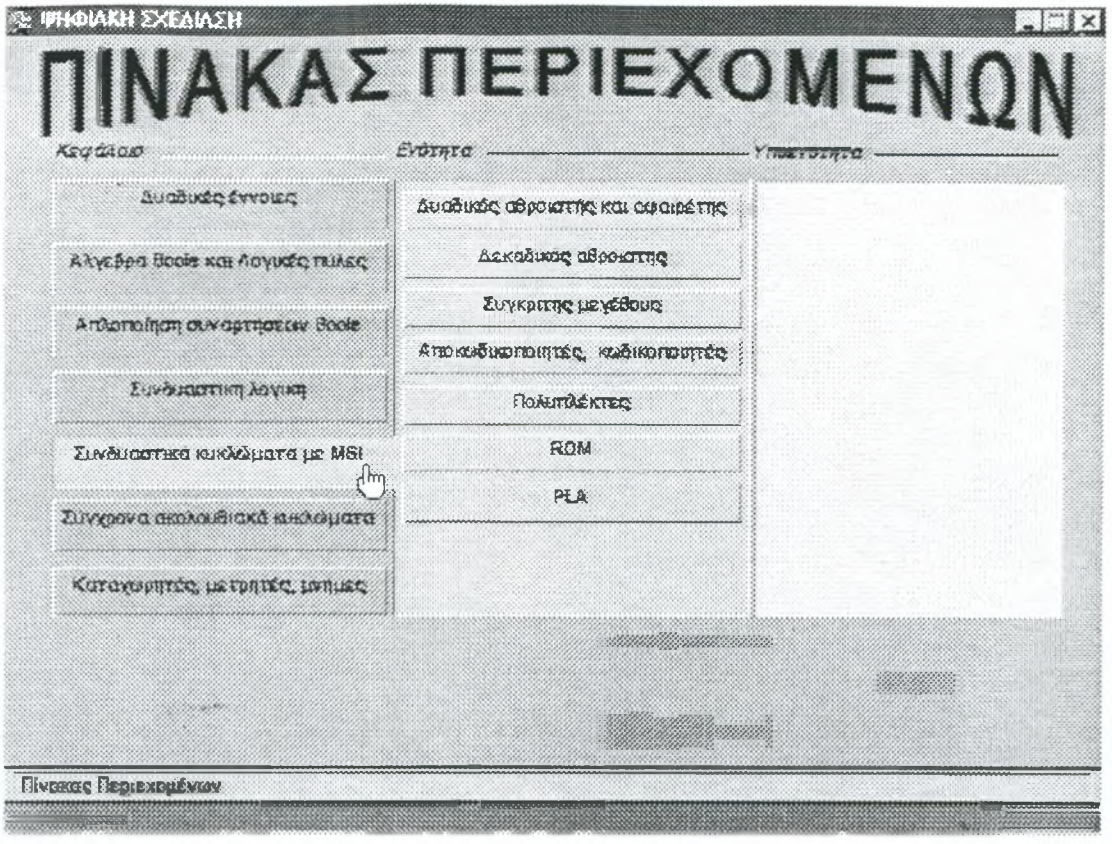
3.2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΤΜΗΜΑΤΟΣ ΔΙΔΑΣΚΑΛΙΑΣ

3.2.1 Περιγραφή του πίνακα περιεχομένων

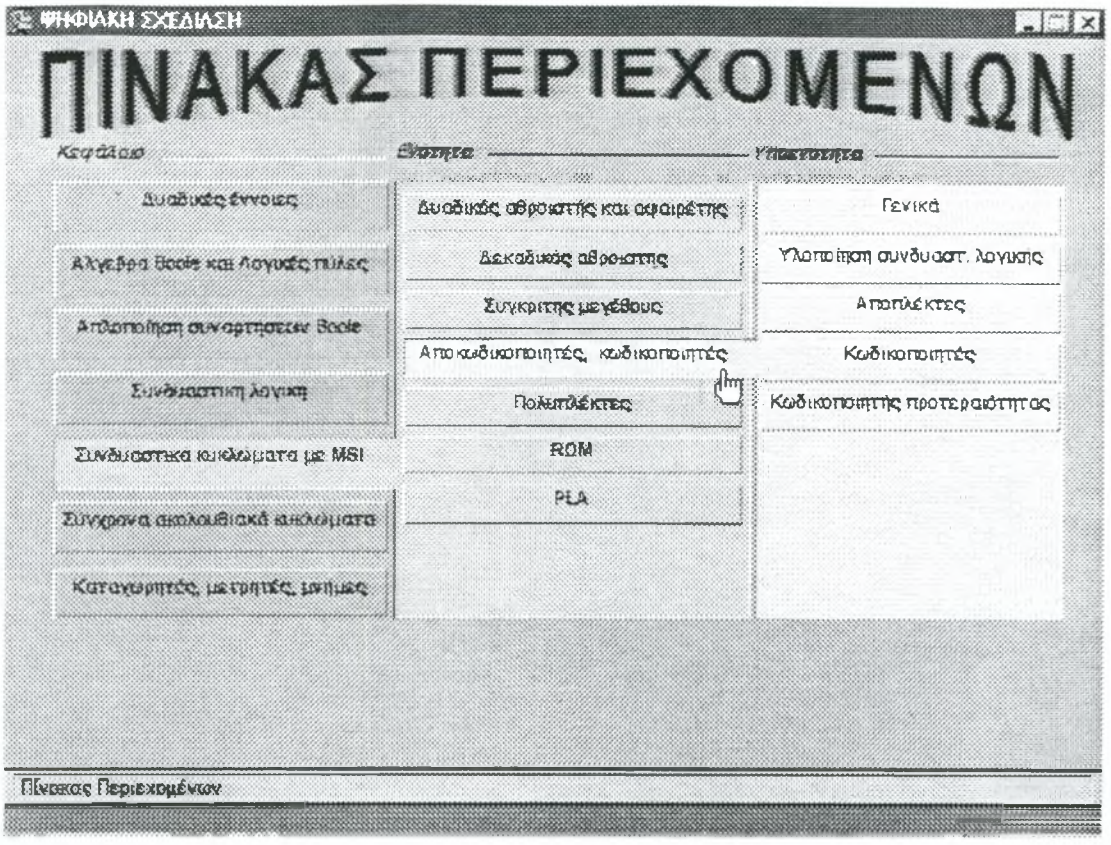
Ο πίνακας περιεχομένων παρουσιάζει τα περιεχόμενα των κεφαλαίων της Ψηφιακής θεωρίας, δηλαδή τις ενότητες μαζί με τις υποενότητες από τις οποίες αποτελείται κάθε κεφάλαιο. Η αρχική εμφάνιση του πίνακα περιεχομένων είναι η εξής:



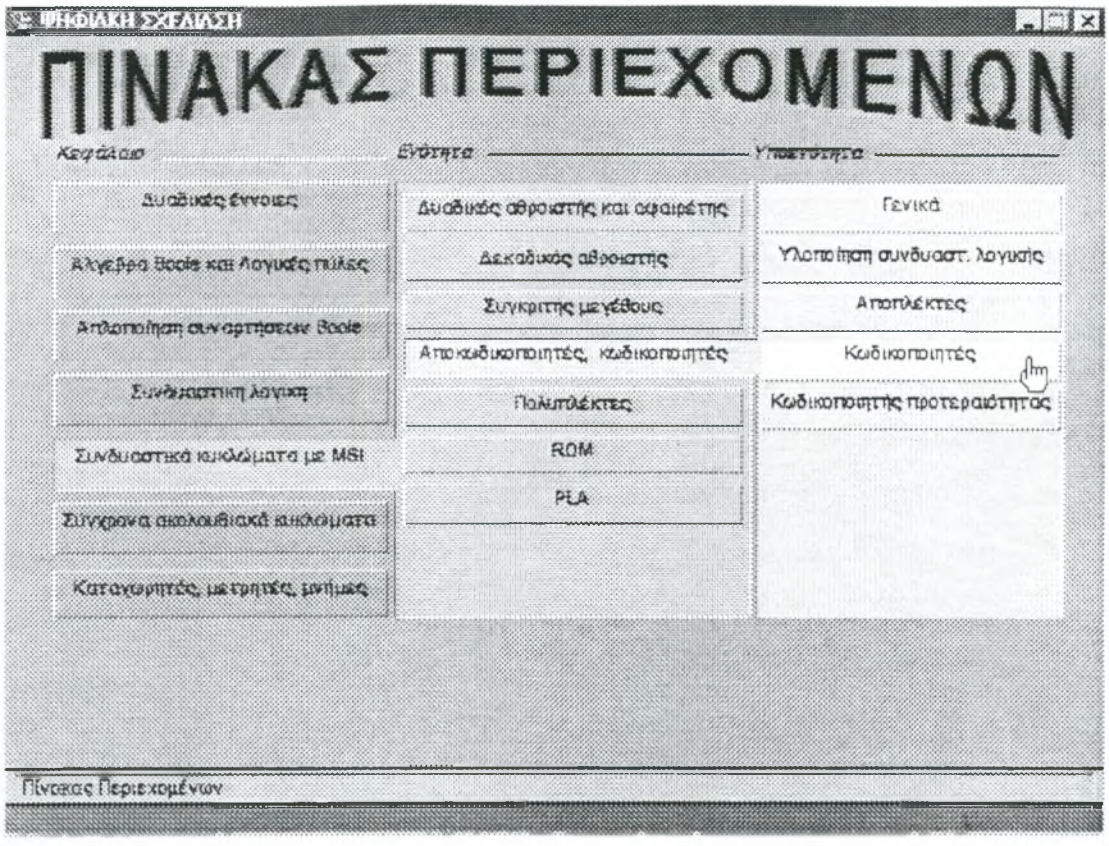
Ο χρήστης έχει τη δυνατότητα να επιλέξει ένα από τα 7 κεφάλαια, οι τίτλοι των οποίων παρουσιάζονται κάτω από την επισήμανση «Κεφάλαιο», κάνοντας κλικ στην αντίστοιχη μπάρα (ο δείκτης του ποντικιού μετατρέπεται σε χέρι πάνω από κάθε μπάρα). Αμέσως, αλλάζει το στυλ της μπάρας που πατήθηκε (γίνεται inset) και έρχεται στο ίδιο επίπεδο με το φάτνωμα που περιέχει τους τίτλους των ενοτήτων του συγκεκριμένου κεφαλαίου. Οι ενότητες αυτές εμφανίζονται αυτόματα με την επιλογή του κεφαλαίου. Π.χ. αν επιλεγεί το κεφάλαιο «Συνδυαστικά κυκλώματα με MSI», τότε ο πίνακας περιεχομένων έχει την εξής μορφή:



Η επιλογή της ενότητας γίνεται ακριβώς με τον ίδιο τρόπο με τον οποίο γίνεται και η επιλογή του κεφαλαίου. Ο χρήστης κάνει κλικ στην μπάρα η οποία περιέχει τον τίτλο της επιθυμητής ενότητας (ο δείκτης του ποντικιού μετατρέπεται σε χέρι πάνω από κάθε μπάρα) και, ομοίως με την επιλογή κεφαλαίου, η επιλεγείσα μπάρα αλλάζει στυλ (γίνεται inset ώστε να έλθει στο ίδιο επίπεδο με το φάτνωμα που περιέχει τους τίτλους των υποενοτήτων της επιλεγείσας ενότητας) ενώ ταυτόχρονα εμφανίζονται οι τίτλοι των υποενοτήτων κάτω από την επισήμανση «Υποενότητες». Αν π.χ. επιλεγεί η ενότητα «Αποκωδικοποιητές, κωδικοποιητές», τότε ο πίνακας περιεχομένων λαμβάνει την εξής μορφή:



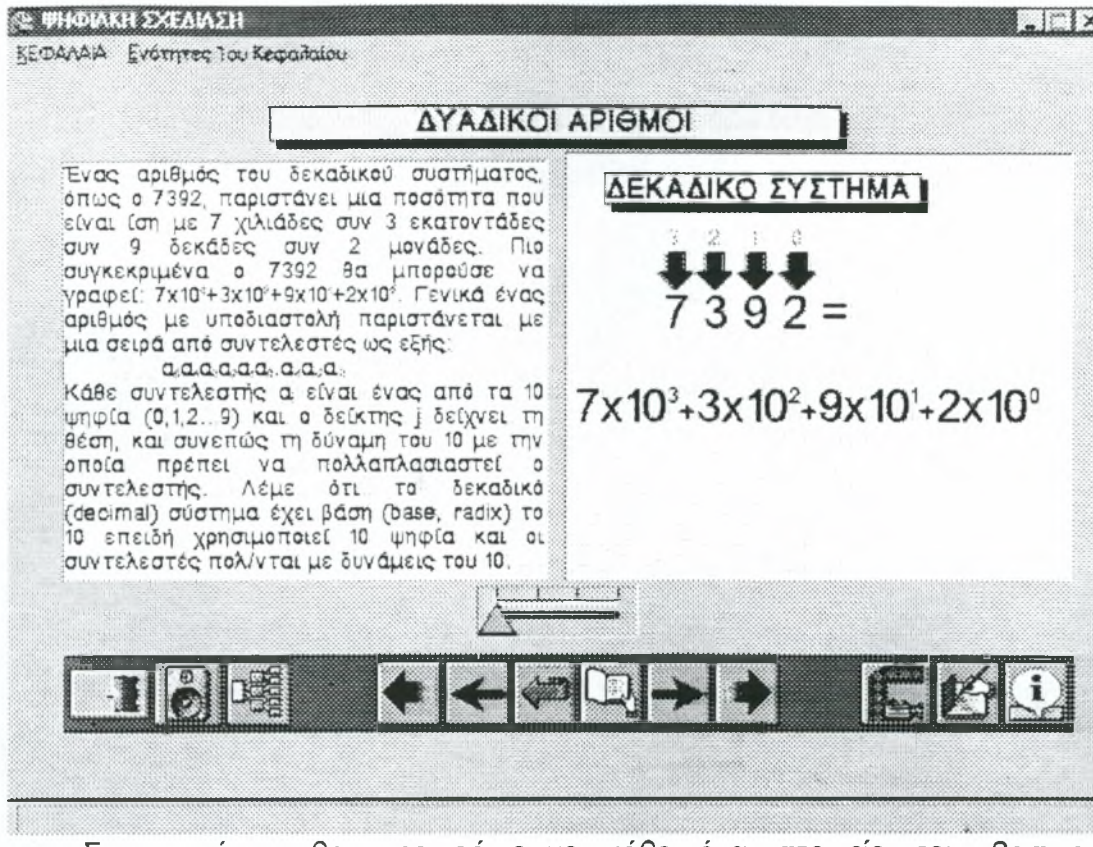
Η επιλογή της επιθυμητής υποενότητας γίνεται και πάλι κάνοντας κλικ στην αντίστοιχη μπάρα. Κάθε φορά που περνά ο δείκτης του ποντικιού πάνω από κάθε μπάρα, μετατρέπεται σε χέρι ενώ αλλάζει ταυτόχρονα και το χρώμα της μπάρας (γίνεται πιο ανοιχτό γκρι). Κάνοντας κλικ σε κάποια μπάρα που περιέχει τον τίτλο υποενότητας μεταφερόμαστε στο αντίστοιχο τμήμα της θεωρίας. Η περιγραφή της θεωρίας θα γίνει στην επόμενη παράγραφο. Κατά την επιλογή μιας υποενότητας, ο πίνακας περιεχομένων έχει την εξής μορφή:



3.2.2 Περιγραφή του βασικού περιβάλλοντος της θεωρίας

Αφού γίνει η επιλογή της επιθυμητής υποενότητας στον πίνακα περιεχομένων, ο χρήστης μεταφέρεται στο βασικό περιβάλλον διδασκαλίας της Ψηφιακής θεωρίας. Το περιβάλλον αυτό αποτελεί τη διασύνδεση του χρήστη με την εφαρμογή - το user interface. Όπως είδαμε στο **Κεφάλαιο 1** το user interface είναι ένα σύνολο από, κύρια οπτικές, αλλά και ακουστικές παραστάσεις, που προβάλλει ο υπολογιστής μέσω του εκάστοτε εκτελούμενου προγράμματος προς τον χρήστη με σκοπό την καλύτερη επικοινωνία και συνεργασία μεταξύ ανθρώπου και μηχανής.

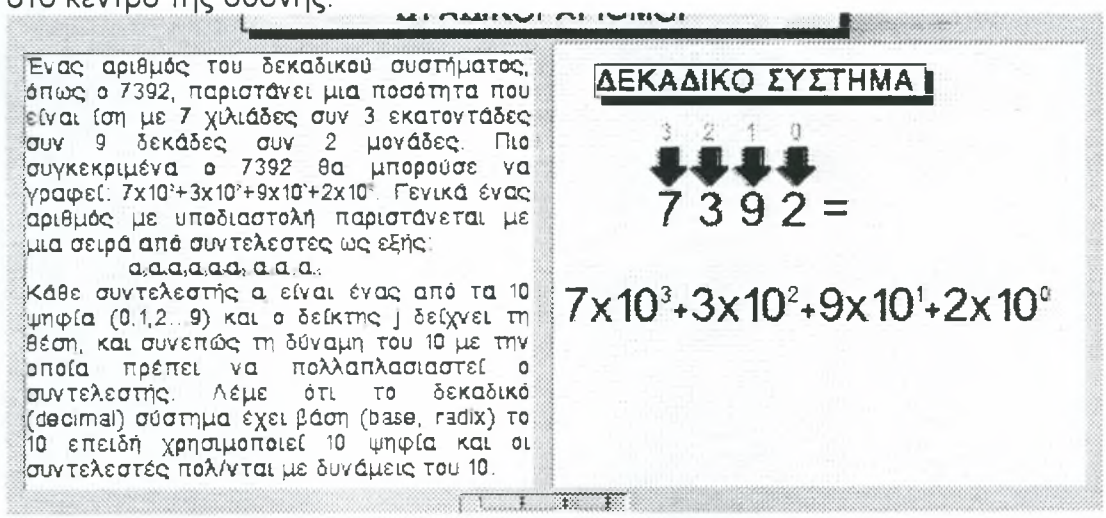
Το κύριο λοιπόν περιβάλλον διδασκαλίας της Ψηφιακής θεωρίας, εμφανίζεται ως εξής: (παρουσιάζεται ενδεικτικά η πρώτη σελίδα).



Στη συνέχεια θα περιγράψουμε κάθε ένα στοιχείο του βασικού περιβάλλοντος, δίνοντας έμφαση στον τρόπο με τον οποίο ο χρήστης αλληλεπιδρά με αυτό το περιβάλλον.

Τα κύρια παράθυρα

Το μεγαλύτερο μέρος της κύριας οθόνης της διδασκαλίας της Ψηφιακής θεωρίας καταλαμβάνεται από δύο παραπλεύρως τοποθετημένα «παράθυρα» στο κέντρο της οθόνης.

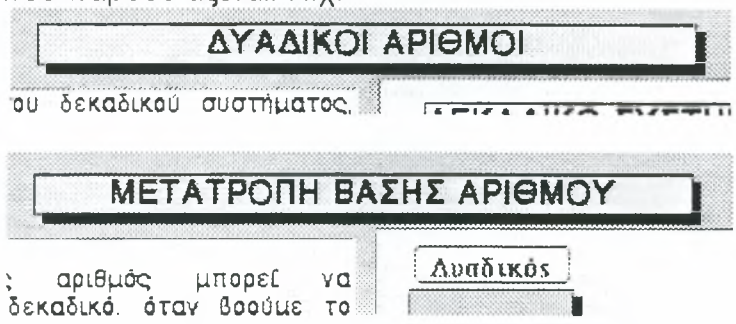


Το αριστερό παράθυρο περιέχει μια λεπτομερειακή ανάλυση ενός τμήματος της θεωρίας, ενώ το δεξιό παράθυρο εμφανίζει είτε κάποια χαρακτηριστικά παραδείγματα του τμήματος θεωρίας του αριστερού παραθύρου, είτε κάποιο εποπτικό κανόνα κατανόησης της θεωρίας, είτε κάποιο πινακοποιημένο παράδειγμα, είτε κάποιο σχόλιο, είτε κάποιας μορφής animation που βοηθάει στην καλύτερη εμπέδωση της θεωρίας. Στην

παραπάνω οθόνη, στο αριστερό παράθυρο παρουσιάζεται η θεωρητική θεμελίωση του δεκαδικού συστήματος αρίθμησης, ενώ στο δεξιό παράθυρο δίνεται μέσω ενός παραδείγματος, η βαρύτητα κάθε ψηφίου στο σχηματισμό του αριθμού ανάλογα με τη θέση του σ' αυτόν (από το λιγότερο σημαντικό ως το περισσότερο σημαντικό).

Ο τίτλος της υποενότητας

Πάνω από τα δύο κύρια παράθυρα εμφανίζεται εντός σκιασμένου πλαισίου ο τίτλος της υποενότητας στην οποία ανήκει το συγκεκριμένο τμήμα της θεωρίας που παρουσιάζεται. Π.χ.



Το slider πλοήγησης

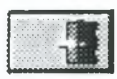
Το slider πλοήγησης αποτελεί μία πρωτοτυπία της εφαρμογής της Ψηφιακής Σχεδίασης και είναι ένα ισχυρό και ιδιαίτερα ελκυστικό χαρακτηριστικό του interface.

Το slider αυτό παρέχει στο χρήστη τη δυνατότητα να πλοηγηθεί εντός των σελίδων της υποενότητας. Συγκεκριμένα, ο χρήστης μπορεί να μετακινηθεί σε οποιαδήποτε σελίδα της υποενότητας επιθυμεί, κάνοντας κλικ με το ποντίκι στο αντίστοιχο tickmark του slider. Το slider περιέχει τόσα tickmarks όσες και οι σελίδες της υποενότητας που εμφανίζεται στην οθόνη. (Αν μια υποενότητα περιέχει μόνο μία σελίδα, το slider δεν εμφανίζεται). Επιπλέον, παρέχεται η δυνατότητα στο χρήστη, πέρα από το κλικ στο αντίστοιχο tickmark, να σύρει το slider διαδοχικά από tickmark σε tickmark, και έτσι να περάσει διαδοχικά από όλες τις σελίδες τις υποενότητας εν συντομία.

Μπάρα πλοήγησης

Η μπάρα πλοήγησης που βρίσκεται κάτω από το slider περέχει όλα εκείνα τα πλήκτρα που δίνουν τη δυνατότητα στο χρήστη να πλοηγηθεί εντός της εφαρμογής ανάλογα με τις επιθυμίες του.

Αναλυτικά, κάθε πλήκτρο επιτελεί την εξής λειτουργία:



Για έξοδο από την εφαρμογή και επιστροφή στα Windows.

Για την απόδοση των σχετικών ηχητικών κομματιών που αναφέρονται στο παρουσιαζόμενο τμήμα της θεωρίας.

Για μετακίνηση στη σελίδα επιλογής διδασκαλίας - εξάσκησης - αξιολόγησης.

Για μετακίνηση στην προηγούμενη ενότητα.

Για μετακίνηση στην προηγούμενη υποενότητα.

Για μετακίνηση στην προηγουμένως εμφανισθείσα σελίδα (το γνωστό back).

Για μετακίνηση στον πίνακα περιεχομένων.

Για μετακίνηση στην επόμενη υποενότητα.

Για μετακίνηση στην επόμενη ενότητα.

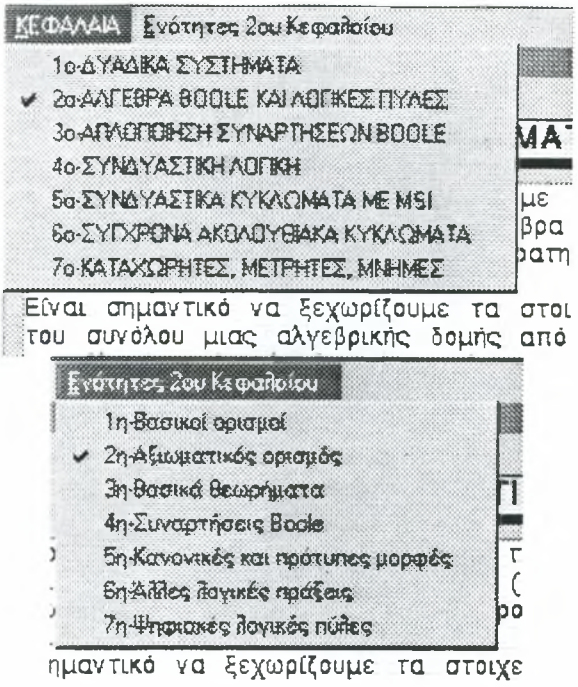
Για την προβολή του σχετικού video-clip που αναφέρεται στην ύλη που περιέχεται σε κάθε κεφάλαιο.

Για μετακίνηση στο τμήμα των ασκήσεων του συγκεκριμένου κεφαλαίου.

Για εμφάνιση της βοήθειας που αναφέρεται στη χρήση της Ψηφιακής Σχεδίασης.

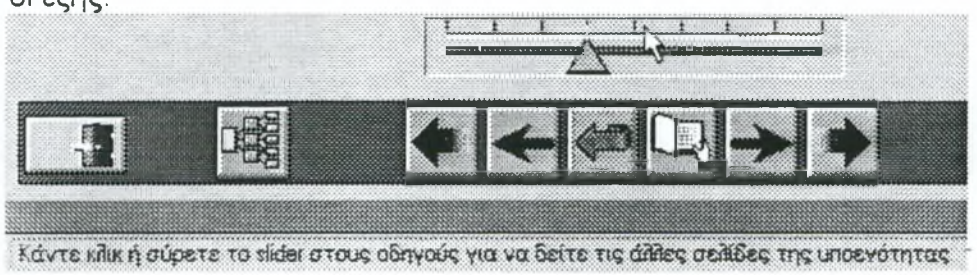
<u>Η μπάρα των μενού</u>

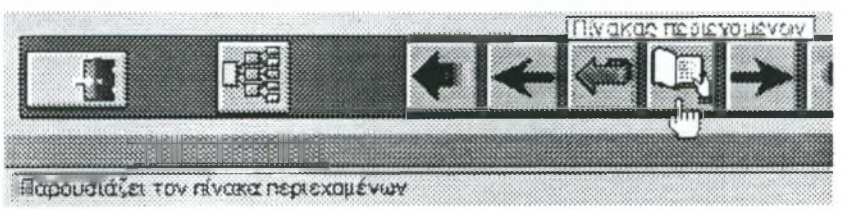
Στην κορυφή της οθόνης του βασικού περιβάλλοντος της θεωρίας υπάρχουν δύο drop-down menus με τη βοήθεια των οποίων ο χρήστης μπορεί άμεσα (χωρίς να μεταβεί στον πίνακα περιεχομένων) να μετακινηθεί στην πρώτη σελίδα κάθε κεφαλαίου ή στην πρώτη σελίδα κάθε ενότητας του τρέχοντος κεφαλαίου. Η μορφή αυτών των drop-down menus είναι η εξής:



Το statusbar

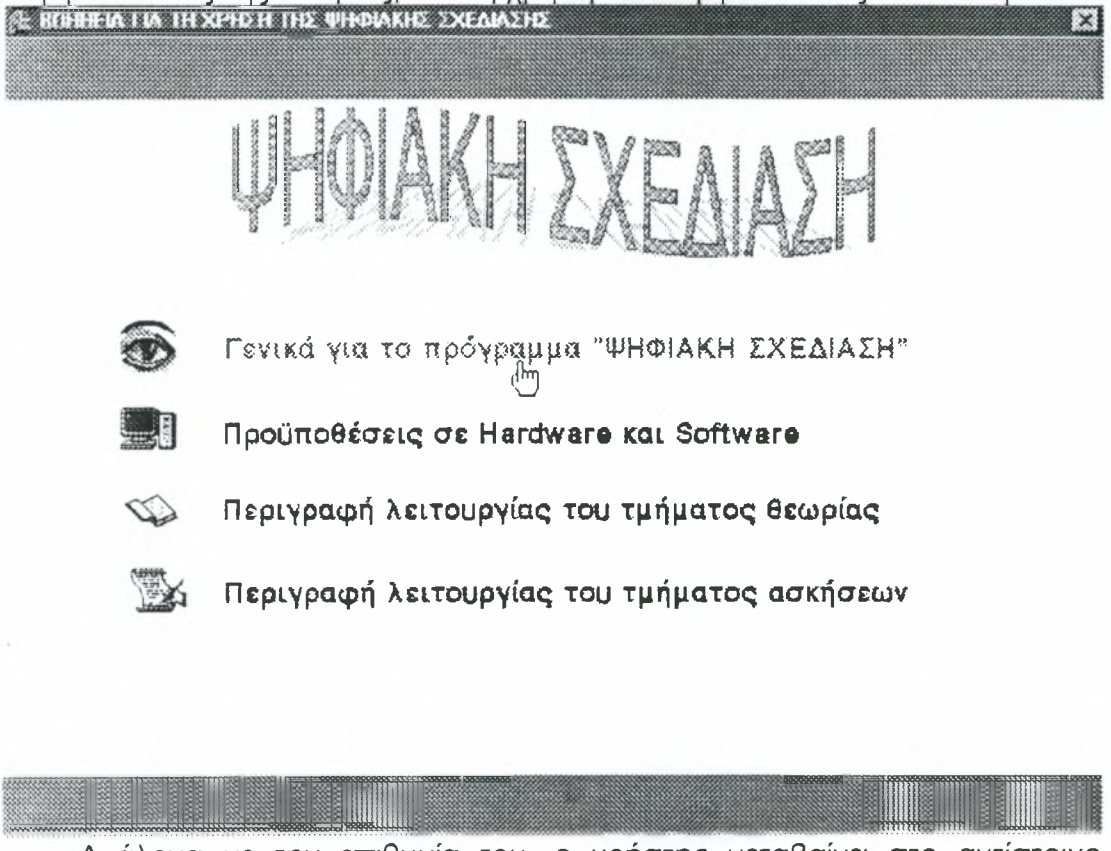
Στη βάση της οθόνης του βασικού περιβάλλοντος της θεωρίας βρίσκεται το statusbar στο οποίο εμφανίζονται ανά πάσα στιγμή πληροφορίες που αφορούν οποιοδήποτε ενεργό τμήμα της οθόνης πάνω από το οποίο περνά το ποντίκι. Όταν το ποντίκι δεν βρίσκεται πάνω σε ενεργό τμήμα της οθόνης (δηλ. πάνω σε πλήκτρο ή μενού ή το slider), το statusbar εμφανίζει τον τίτλο του κεφαλαίου και της ενότητας που ανήκει το τμήμα της θεωρίας που παρουσιάζεται στην οθόνη. Κάποιες χαρακτηριστικές εμφανίσεις του statusbar είναι οι εξής:





3.2.3 Περιγραφή της βοήθειας για τη χρήση των τμημάτων διδασκαλίας (θεωρίας) και εξάσκησης

Η βοήθεια για τη χρήση των τμημάτων διδασκαλίας (θεωρίας) και εξάσκησης καλείται με το πάτημα του πλήκτρου που βρίσκεται στη μπάρα πλοήγησης του βασικού περιβάλλοντος της θεωρίας. Εμφανίζεται λοιπόν το κυρίως μενού της βοήθειας με το οποίο ο χρήστης μπορεί να επιλέξει είτε να δει γενικές πληροφορίες για το πρόγραμμα, είτε να δει τις προϋποθέσεις σε hardware και software, είτε τη χρήση του βασικού περιβάλλοντος της θεωρίας, είτε τη χρήση του περιβάλλοντος των ασκήσεων.

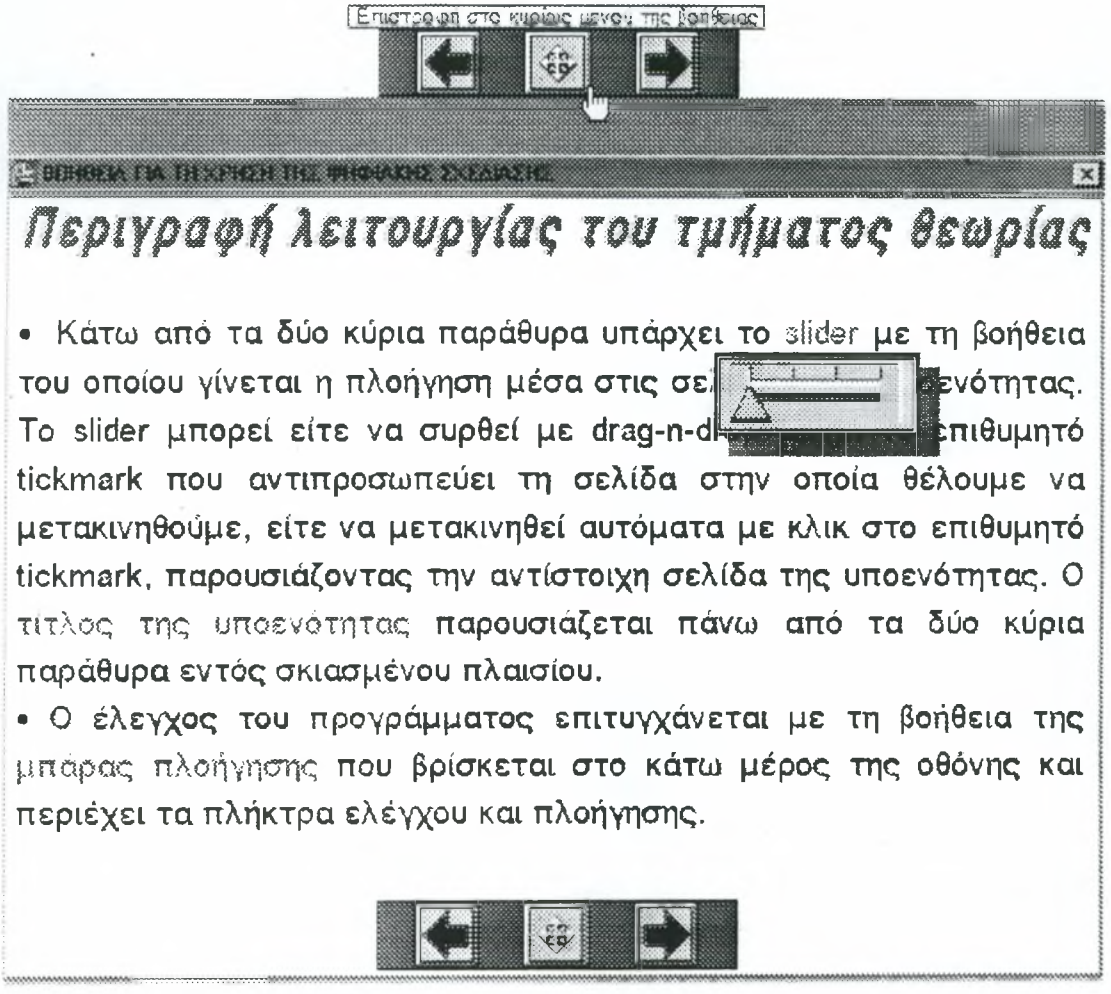


Ανάλογα με την επιθυμία του, ο χρήστης μεταβαίνει στο αντίστοιχο τμήμα της βοήθειας που τον ενδιαφέρει. Παρακάτω παρουσιάζονται κάποιες χαρακτηριστικές οθόνες της βοήθειας:

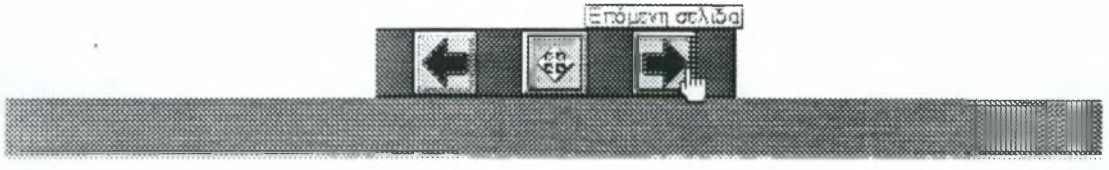
Προϋποθέσεις σε Hardware και Software

Το πρόγραμμα έχει αναπτυχθεί στο Multimedia Toolbook 3.0 της Asymetrix σ' ένα PC με επεξεργαστή Pentium στα 75MHz στο περιβάλλον των Windows 95. Προϋπόθεση για τη σωστή λειτουργία της ΨΗΦΙΑΚΗΣ ΣΧΕΔΙΑΣΗΣ είναι η ύπαρξη ενός PC με Windows έκδοση 3.1 ή ανώτερη και μνήμη τουλάχιστον 8Mb.

Άλλες απαιτήσεις σε hardware και software είναι:



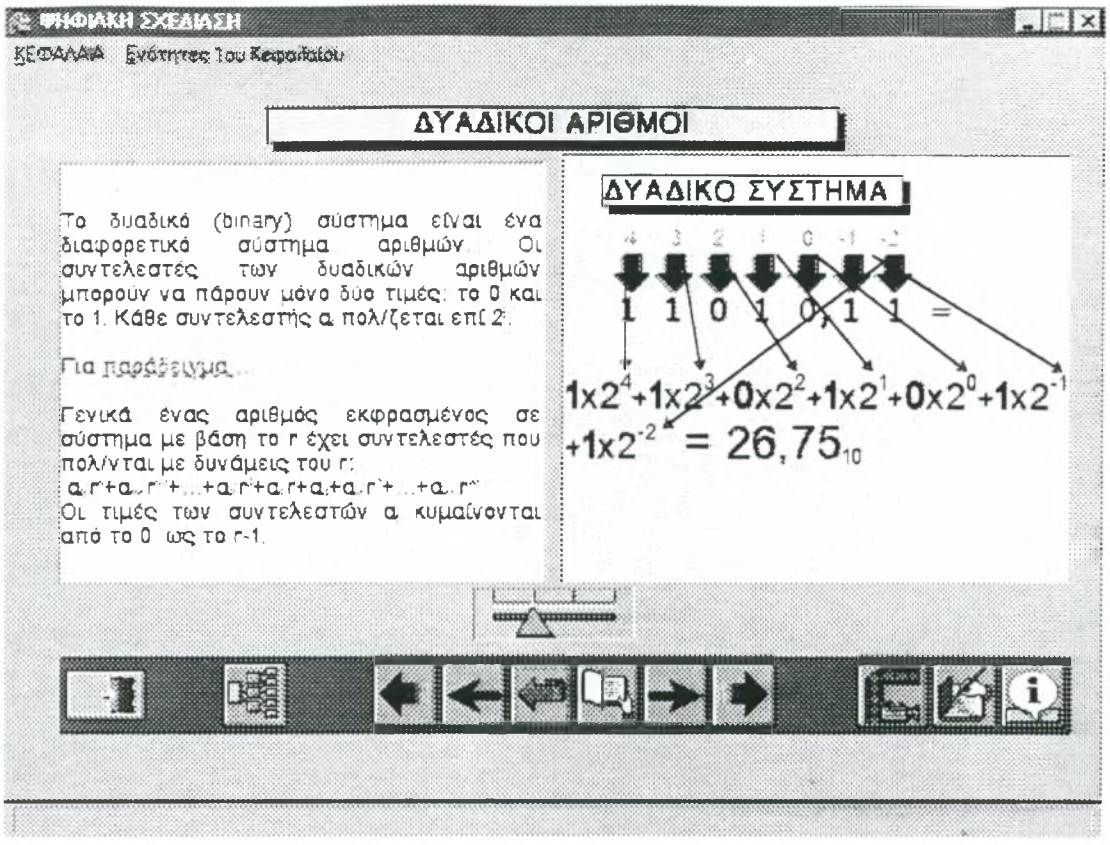
Η παρουσίαση των ασκήσεων γίνεται με τον κλασσικό τρόπο όπου στο πάνω μέρος της οθόνης εμφανίζεται η εκφώνηση και ο μαθητής προτρέπεται να κάνει κλικ στο πλήκτρο της λύσης για να παρουσιαστεί η λύση της προτεινόμενης άσκησης. Κατά την παρουσίαση της λύσης ιδιαίτερη σημασία δίνεται στον τρόπο σχηματισμού αυτής και στην αλληλουχία των βημάτων επίλυσης. Στην κορυφή της οθόνης είναι τοποθετημένα τα ακόλουθα πλήκτρα ελέγχου:



3.2.4 Χαρακτηριστικές σελίδες του τμήματος διδασκαλίας

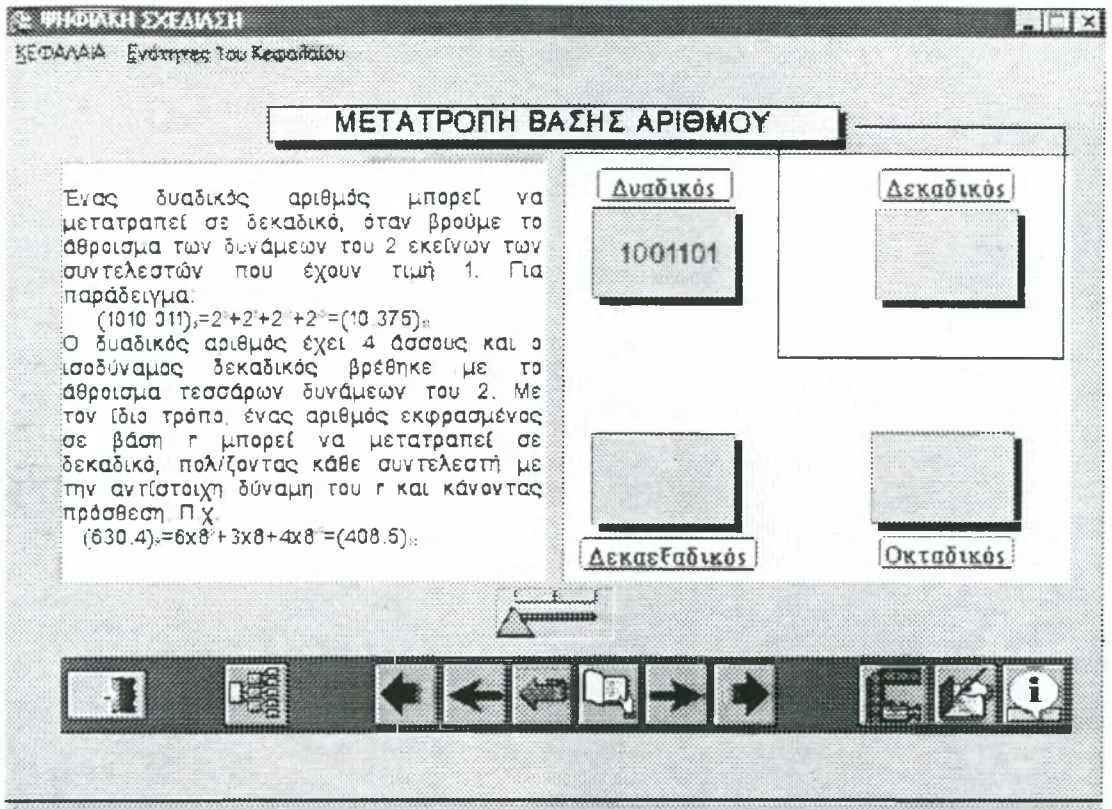
Αριθμοί στο δυαδικό σύστημα

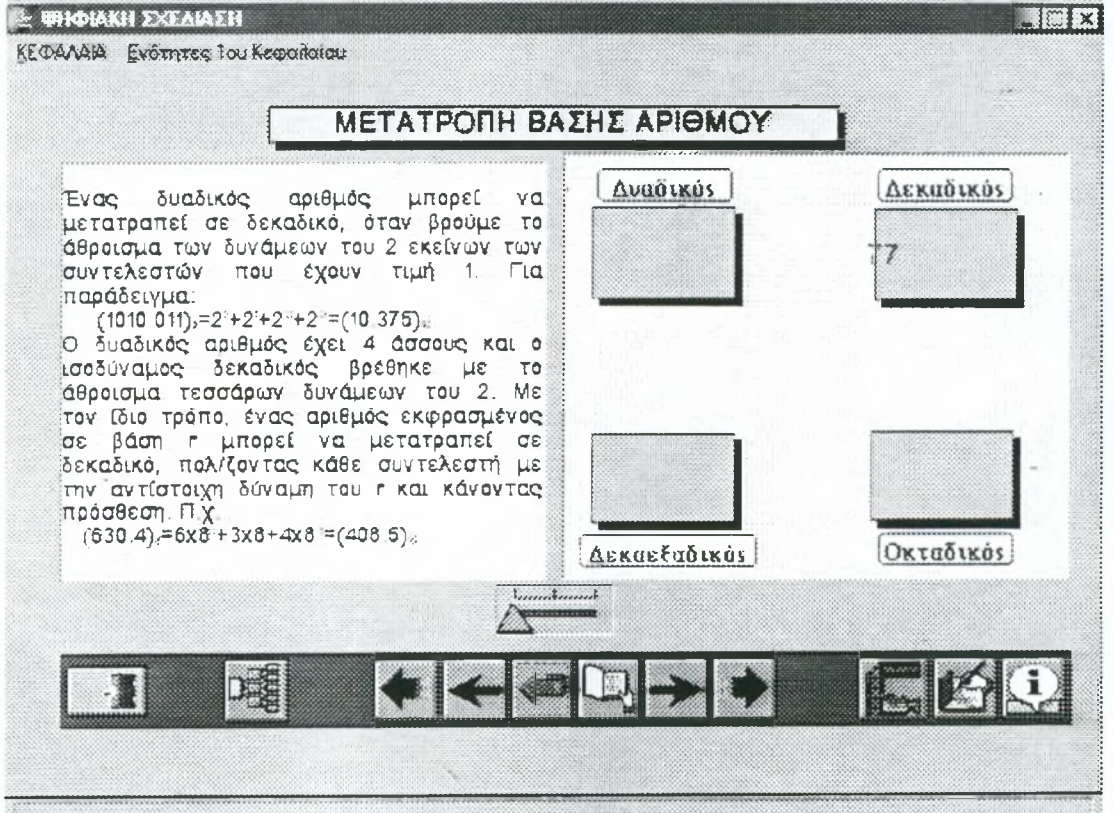
Στην ενότητα «δυαδικοί αριθμοί» του κεφαλαίου «δυαδικές έννοιες» ο μαθητής εισάγεται στους δυαδικούς αριθμούς. Στο αριστερό κύριο παράθυρο δίνεται ο ορισμός ενός αριθμού με βάση το 2 και πώς αυτός μετατρέπεται στο δεκαδικό σύστημα, ενώ στο αριστερό κύριο παράθυρο εμφανίζεται ο δυαδικός 11010.11 και η διαδικασία μετατροπής του σε δεκαδικό ανάλογα με το βάρος κάθε ψηφίου του. Τα βάρη των ψηφίων, ανάλογα με τη θέση κάθε ψηφίου στον αριθμό, μετακινούνται και γίνονται οι εκθέτες στην αριθμητική παράσταση της μετατροπής. Η κίνηση αυτή των αριθμών των βαρών των ψηφίων στην αρχική και τελική τους θέση γίνεται καθόλη τη διάρκεια μη ύπαρξης δραστηριότητας από την πλευρά του χρήστη (συμβάν idle - βλ. Κεφάλαιο 2 «Εισαγωγή στο Toolbook»). Η οθόνη που παρουσιάζει αυτό το εφφέ είναι η εξής:

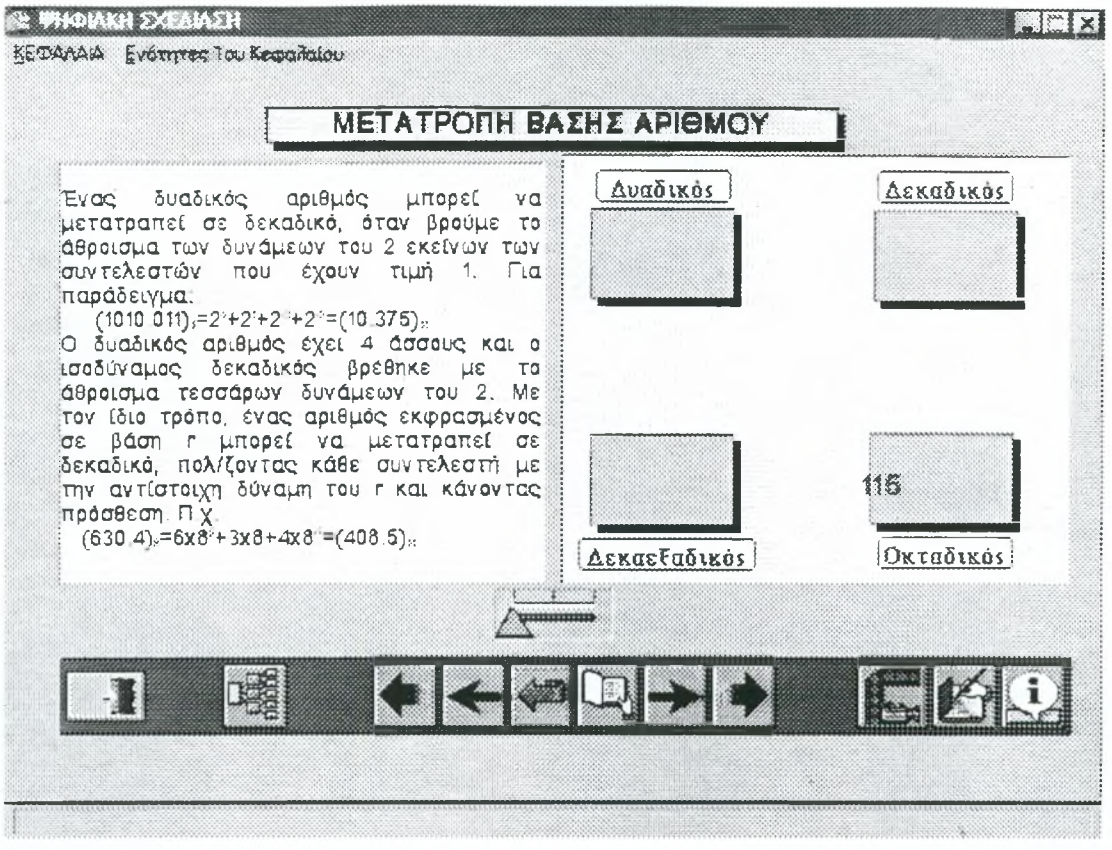


Μετατροπή βάσης αριθμού

Στην ενότητα «μετατροπή βάσης αριθμού» του κεφαλαίου «δυαδικές έννοιες» ο μαθητής διδάσκεται τη διαδικασία μετατροπής ενός αριθμού από βάση σε βάση. Στο αριστερό κύριο παράθυρο κάθε σελίδας της ενότητας αυτής παρουσιάζεται η σχετική διαδικασία, ενώ στο δεξιό κύριο παράθυρο της πρώτης σελίδας, ο μαθητής βλέπει έναν αριθμό ο οποίος κινείται κυκλικά, περνά μέσα από διαφορετικά αριθμητικά συστήματα και αλλάζει ανάλογα.

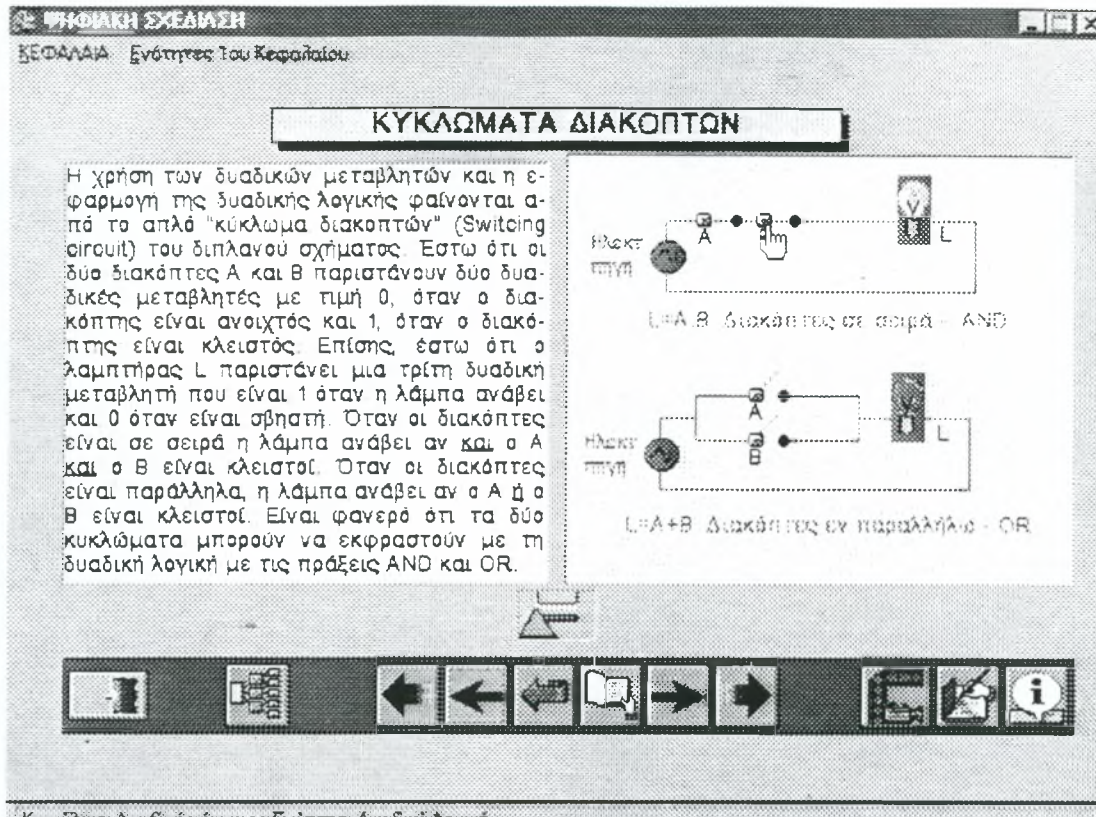

ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ
ΚΕΦΑΛΑΙΑ Ενότητες Του Κεφαλαίου
ΜΕΤΑΤΡΟΠΗ ΒΑΣΗΣ ΑΡΙΘΜΟΥ
Ένας δυαδικός αριθμός μπορεί να μετατραπεί σε δεκαδικό, όταν βρούμε το άθροισμα των δυνάμεων του 2 εκείνων των συντελεστών που έχουν τιμή 1. Για παράδειγμα:
(1010 011)₂=2⁶+2⁴+2¹+2⁰=(10 375)₁₀
Ο δυαδικός αριθμός έχει 4 άσσους και ο ισοδύναμος δεκαδικός βρέθηκε με το άθροισμα τεσσάρων δυνάμεων του 2. Με τον ίδιο τρόπο, ένας αριθμός εκφρασμένος σε βάση r μπορεί να μετατραπεί σε δεκαδικό, πολ/ζοντας κάθε συντελεστή με την αντίστοιχη δύναμη του r και κάνοντας πρόσθεση. Π.χ.
(630.4)₈=6x8²+3x8+4x8⁻¹=(408.5)₁₀
Δυαδικός
1001101
Δεκαδικός
Δεκαεξαδικός
Οκταδικός

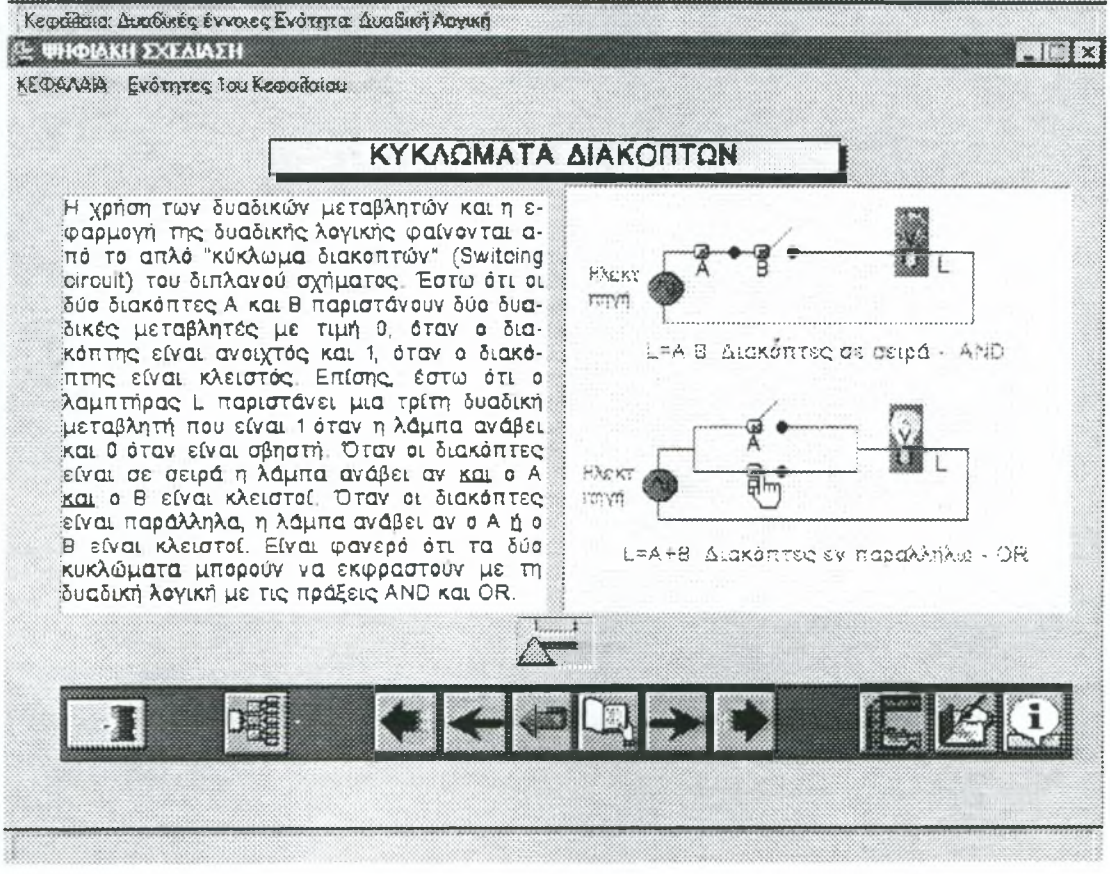

ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ
ΚΕΦΑΛΑΙΑ Ενότητες Του Κεφαλαίου
ΜΕΤΑΤΡΟΠΗ ΒΑΣΗΣ ΑΡΙΘΜΟΥ
Ένας δυαδικός αριθμός μπορεί να μετατραπεί σε δεκαδικό, όταν βρούμε το άθροισμα των δυνάμεων του 2 εκείνων των συντελεστών που έχουν τιμή 1. Για παράδειγμα:
(1010 011)₂=2⁶+2⁴+2¹+2⁰=(10 375)₁₀
Ο δυαδικός αριθμός έχει 4 άσσους και ο ισοδύναμος δεκαδικός βρέθηκε με το άθροισμα τεσσάρων δυνάμεων του 2. Με τον ίδιο τρόπο, ένας αριθμός εκφρασμένος σε βάση r μπορεί να μετατραπεί σε δεκαδικό, πολ/ζοντας κάθε συντελεστή με την αντίστοιχη δύναμη του r και κάνοντας πρόσθεση. Π.χ.
(630.4)₈=6x8²+3x8+4x8⁻¹=(408.5)₁₀
Δυαδικός
Δεκαδικός
77
Δεκαεξαδικός
Οκταδικός



Κυκλώματα διακοπτών

Στην ενότητα «δυαδική λογική» του κεφαλαίου «δυαδικές έννοιες» ο μαθητής διδάσκεται την εφαρμογή της δυαδικής λογικής μέσω των κυκλωμάτων διακοπτών. Στο αριστερό κύριο παράθυρο καλύπτεται το θεωρητικό κομμάτι, ενώ στο δεξιό κύριο παράθυρο παρουσιάζονται τα δύο απλά κυκλώματα διακοπτών που αντιστοιχούν στις λογικές πράξεις AND και OR. Ο μαθητής έχει τη δυνατότητα να δει τη λειτουργία των κυκλωμάτων (πότε ανάβει και πότε σβήνει ο λαμπτήρας) ανοίγοντας και κλείνοντας τους διακόπτες, κάνοντας κλικ πάνω τους.

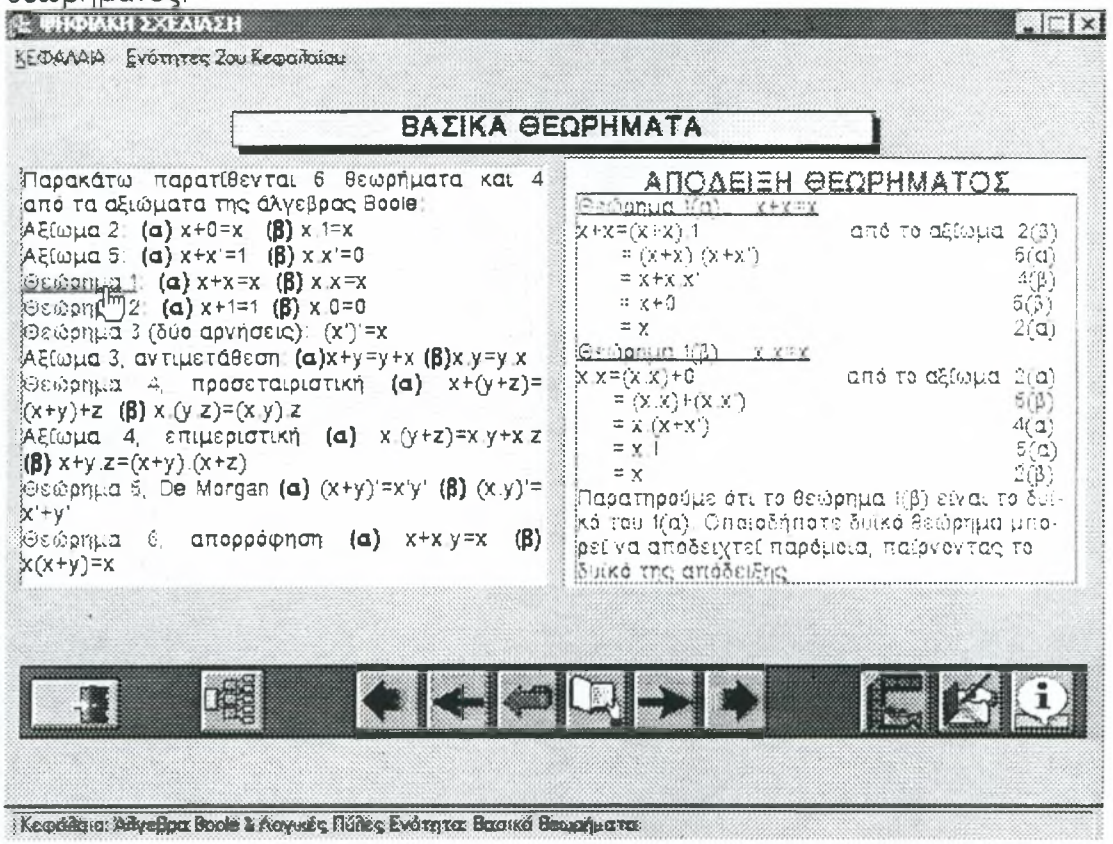




Βασικά θεωρήματα της άλγεβρας Boole

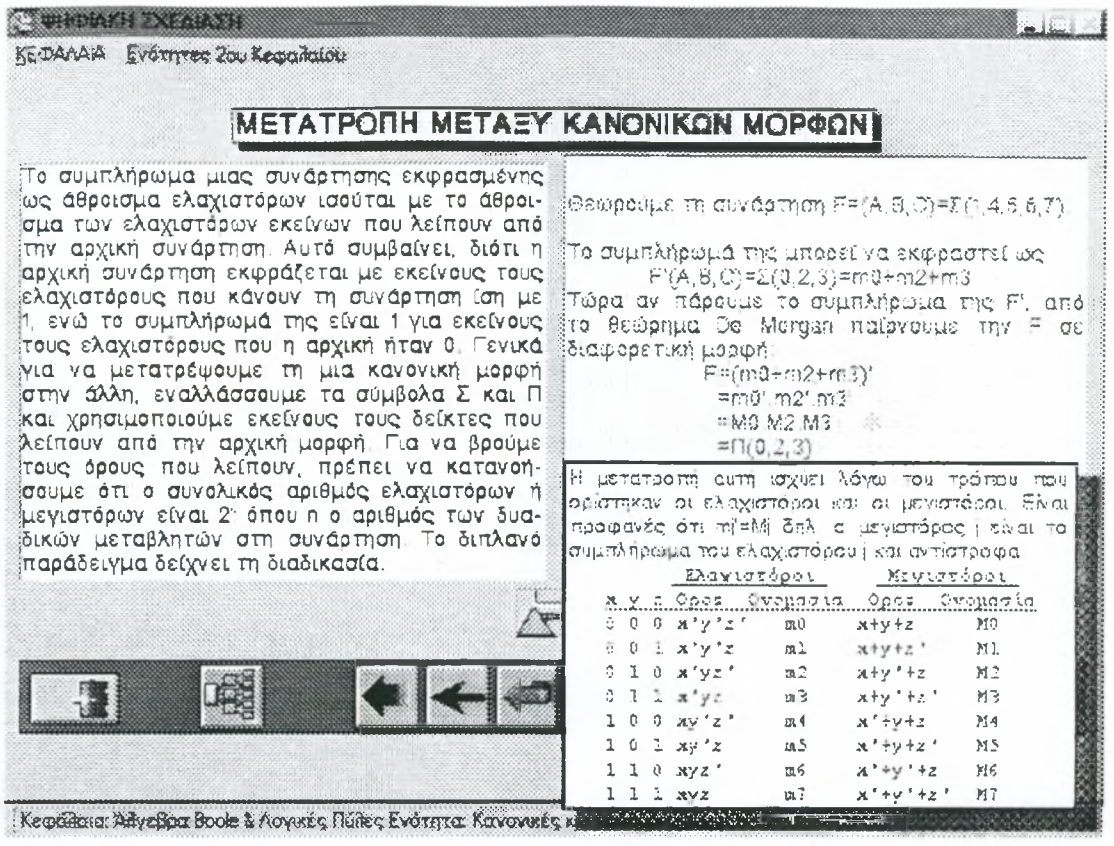
Στην ενότητα «βασικά θεωρήματα» του κεφαλαίου «Άλγεβρα Boole και λογικές πύλες» ο μαθητής διδάσκεται τα βασικά θεωρήματα της άλγεβρας Boole. Στο αριστερό κύριο παράθυρο παρουσιάζονται τα αξιώματα και τα

βασικά θεωρήματα, ενώ ο μαθητής κάνοντας κλικ πάνω σε κάθε θεώρημα, έχει τη δυνατότητα να δει στο δεξιό κύριο παράθυρο την απόδειξη του θεωρήματος.



Μετατροπή μεταξύ κανονικών μορφών

Στην ενότητα «κανονικές και πρότυπες μορφές» του κεφαλαίου «Άλγεβρα Boole και λογικές πύλες» ο μαθητής διδάσκεται τη διαδικασία μετατροπής μεταξύ των κανονικών μορφών (άθροισμα ελαχιστόρων και γινόμενο μεγιστόρων). Στο αριστερό κύριο παράθυρο παρουσιάζεται το θεωρητικό μέρος, ενώ στο δεξιό παρουσιάζεται ένα παράδειγμα. Ο μαθητής προτρέπεται να κάνει κλικ σ' έναν αστερίσκο που αλλάζει χρώμα για να δει μέσω ενός pop-up παραθύρου τη δικαιολόγηση της μετατροπής που γίνεται στο παράδειγμα από ελαχιστόρους σε μεγιστόρους.

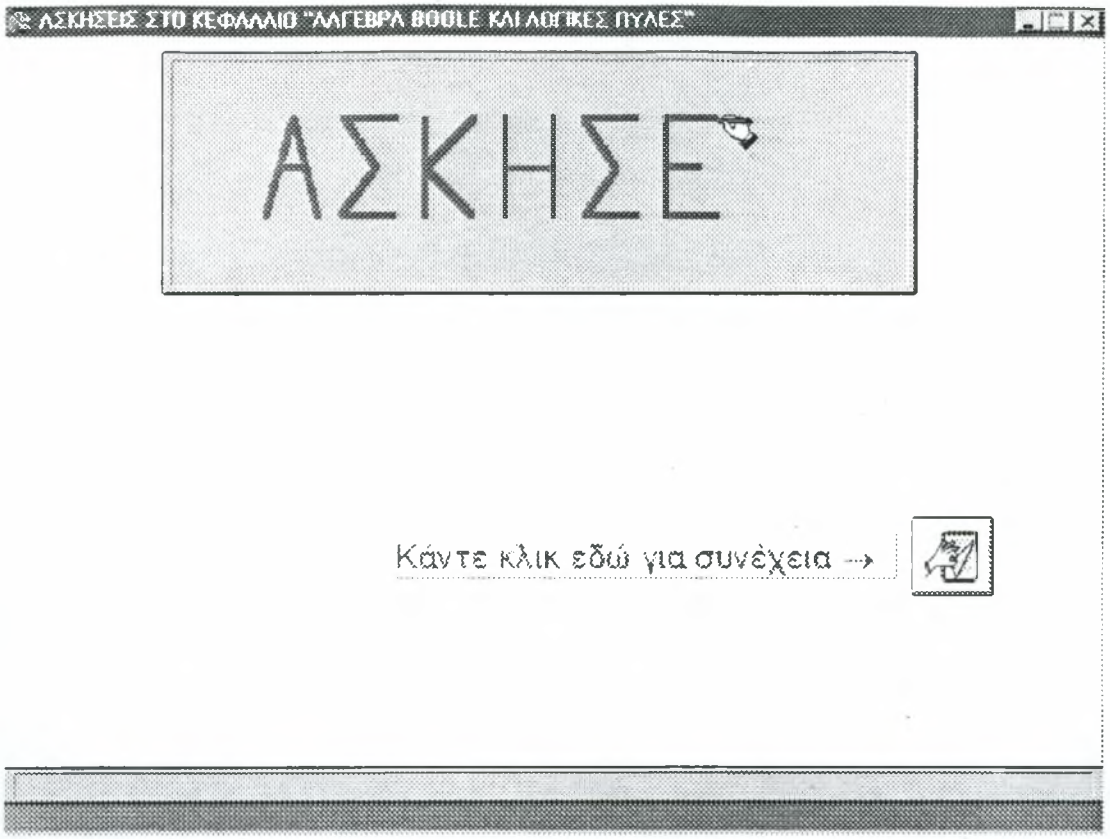


3.3 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΤΜΗΜΑΤΟΣ ΕΞΑΣΚΗΣΗΣ

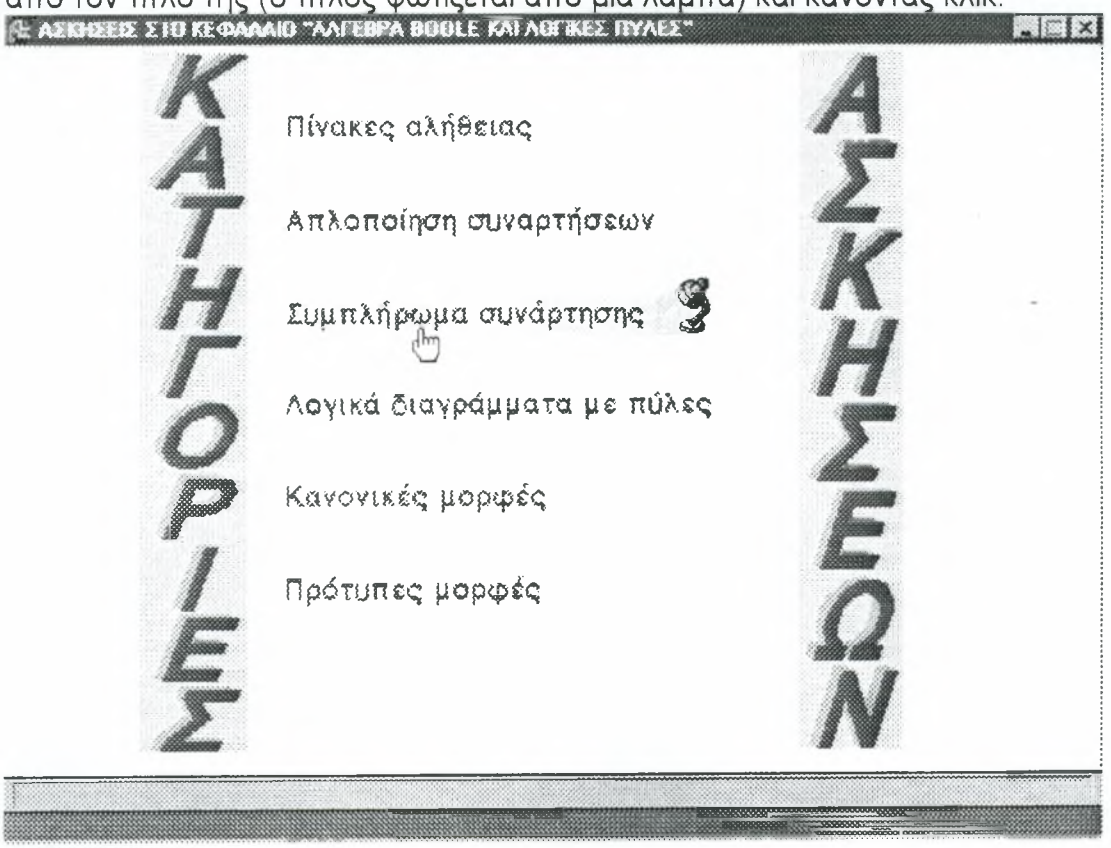
3.3.1 Περιγραφή του βασικού περιβάλλοντος των ασκήσεων

Το τμήμα της εξάσκησης είναι το κομμάτι εκείνο της Ψηφιακής Σχεδίασης μέσω του οποίου ο μαθητής έχει τη δυνατότητα να εφαρμόσει τα όσα έμαθε στο τμήμα της διδασκαλίας της Ψηφιακής θεωρίας πάνω σε μια πληθώρα σχετικών ασκήσεων που αντιστοιχούν σε κάθε ένα κεφάλαιο της θεωρίας. Η βασική ιδέα είναι να παρουσιαστεί στο μαθητή η εκφώνηση της άσκησης, να του δοθεί ο χρόνος να τη λύσει (με το χέρι), και κατόπιν να δει τη λύση ελέγχοντας ο ίδιος τα αποτελέσματα. Κατά την παρουσίαση της λύσης ιδιαίτερη σημασία δίνεται στον τρόπο σχηματισμού αυτής, καθώς και στα διαδοχικά βήματα που απαιτούνται ώστε να φτάσουμε στο τελικό αποτέλεσμα.

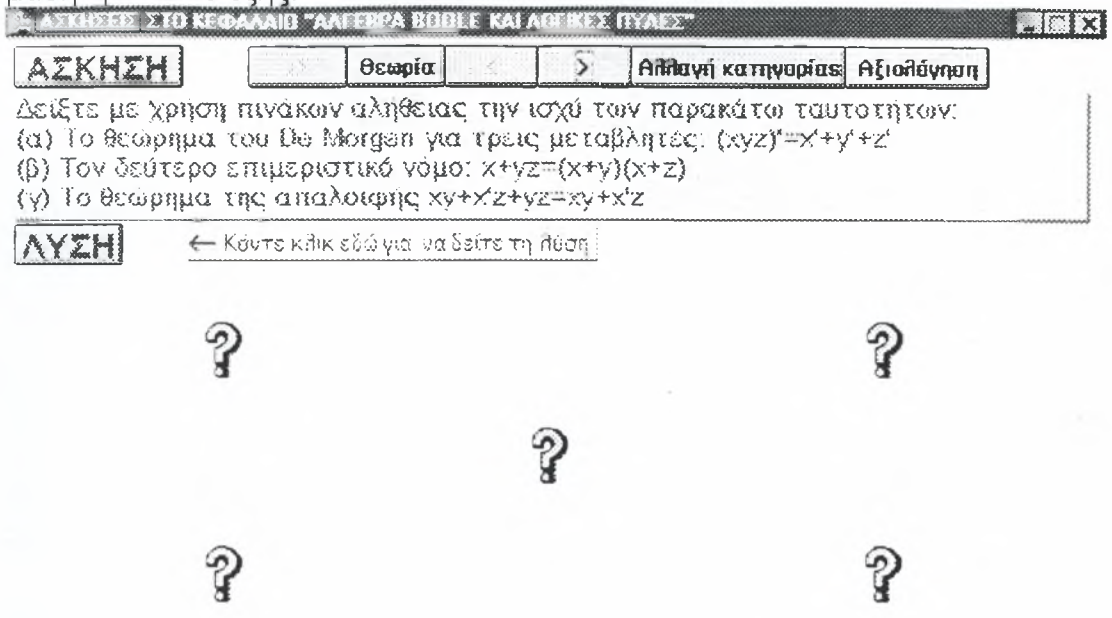
Η εισαγωγή του μαθητή στις ασκήσεις γίνεται εντυπωσιακά με την αρχική σελίδα στην οποία παρουσιάζεται ένας πίνακας διδασκαλίας και ένα χέρι με μια κιμωλία με την οποία γράφει στον πίνακα τη λέξη «ΑΣΚΗΣΕΙΣ».



Κατόπιν, κάνοντας κλικ ο μαθητής στο σχετικό πλήκτρο, μεταβαίνει στη σελίδα επιλογής κατηγορίας ασκήσεων. Στη σελίδα αυτή παρουσιάζονται οι κατηγορίες των ασκήσεων για το συγκεκριμένο κεφάλαιο και ο μαθητής μπορεί να επιλέξει την κατηγορία που επιθυμεί πηγαίνοντας το ποντίκι πάνω από τον τίτλο της (ο τίτλος φωτίζεται από μια λάμπα) και κάνοντας κλικ.



Το βασικό περιβάλλον με το οποίο παρουσιάζονται οι ασκήσεις στο μαθητή είναι το εξής:



Στο πάνω μέρος της οθόνης εμφανίζεται η εκφώνηση της άσκησης και μέχρι να κάνει ο μαθητής κλικ στο πλήκτρο «ΛΥΣΗ» ώστε να δει τη λύση της προτεινόμενης άσκησης, στην οθόνη αναβοσβήνουν πέντε ερωτηματικά. Τα πλήκτρα με τα οποία ο μαθητής χειρίζεται το περιβάλλον των ασκήσεων είναι τα εξής:

Το πλήκτρο ΑΣΚΗΣΗ

Με το πλήκτρο αυτό ο μαθητής μεταβαίνει στην πρώτη άσκηση κάποιας κατηγορίας, με τυχαίο τρόπο.

Το πλήκτρο >>

Με το πλήκτρο αυτό ο μαθητής μεταβαίνει στην επόμενη άσκηση της ίδιας κατηγορίας. Όταν δεν υπάρχουν άλλες ασκήσεις στην ίδια κατηγορία, το πλήκτρο αυτό εμφανίζεται απενεργοποιημένο.

Το πλήκτρο Θεωρία

Με το πλήκτρο αυτό ο μαθητής μεταβαίνει στο τμήμα εκείνο της θεωρίας το οποίο είναι σχετικό με την προτεινόμενη άσκηση, ώστε αν έχει δυσκολία στη λύση της να συμβουλευτεί το αντίστοιχο κομμάτι της θεωρίας.

Το πλήκτρο <

Με το πλήκτρο αυτό ο μαθητής μεταβαίνει στην προηγούμενη άσκηση σειριακά (με τη σειρά που είναι τοποθετημένες οι ασκήσεις στο βιβλίο).

Το πλήκτρο [>]

Με το πλήκτρο αυτό ο μαθητής μεταβαίνει στην επόμενη άσκηση σειριακά (με τη σειρά που είναι τοποθετημένες οι ασκήσεις στο βιβλίο).

Το πλήκτρο [Αλλαγή κατηγορίας]

Με το πλήκτρο αυτό ο μαθητής μεταβαίνει στη σελίδα επιλογής κατηγορίας ασκήσεων.

Το πλήκτρο [Αξιολόγηση]

Με το πλήκτρο αυτό ο μαθητής μεταβαίνει στο τμήμα της αξιολόγησης.

Το πλήκτρο [ΛΥΣΗ]

Με το πλήκτρο αυτό εμφανίζεται η λύση στην προτεινόμενη άσκηση.

Πρέπει να σημειωθεί, όπως επισημάνθηκε και παραπάνω, ότι κάνοντας κλικ ο μαθητής στο πλήκτρο της λύσης δεν παρουσιάζεται μονομιάς η λύση της άσκησης (εκτός από εκείνες τις περιπτώσεις που η φύση της λύσης είναι τέτοια), αλλά ο μαθητής συμμετέχει στο «χτίσιμο» της λύσης. Στην επόμενη υποπαράγραφο παρουσιάζονται κάποιες χαρακτηριστικές ασκήσεις όπου φαίνεται ο ιδιαίτερα πρωτότυπος τρόπος δημιουργίας της λύσης.

3.3.2 Χαρακτηριστικές ασκήσεις του τμήματος εξάσκησης

Μετατροπή δυαδικού αριθμού σε δεκαδικό

Στις ασκήσεις του 1ου Κεφαλαίου «Δυαδικές έννοιες» υπάρχουν ασκήσεις μετατροπής δυαδικών αριθμών σε δεκαδικούς. Έτσι, για τη μετατροπή του 101110 στο δεκαδικό σύστημα η λύση παρουσιάζεται ως εξής: Εμφανίζεται στο αριστερό μέρος ο δυαδικός αριθμός και ένα βέλος δείχνει διαδοχικά τα ψηφία του ξεκινώντας από το περισσότερο σημαντικό. Ταυτόχρονα, μετά το ίσον (=), σχηματίζεται διαδοχικά η αριθμητική παράσταση της μετατροπής στο δεκαδικό για κάθε ένα από τα ψηφία του δυαδικού (για το ψηφίο που υποδεικνύεται από το βέλος). Τέλος, μετά τη μετατροπή και του λιγότερου σημαντικού ψηφίου, εμφανίζεται το τελικό αποτέλεσμα. Οι σχετικές οθόνες φαίνονται παρακάτω:

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΔΥΑΔΙΚΕΣ ΕΝΝΟΙΕΣ"

ΑΣΚΗΣΗ | >> | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Μετατρέψτε τον επόμενο δυαδικό αριθμό σε δεκαδικό: 101110.

ΛΥΣΗ

$(101110)_2 = 1x2^5 +$

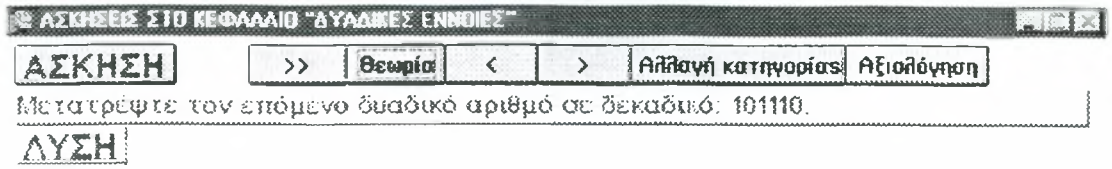
ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΔΥΑΔΙΚΕΣ ΕΝΝΟΙΕΣ"

ΑΣΚΗΣΗ | >> | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Μετατρέψτε τον επόμενο δυαδικό αριθμό σε δεκαδικό: 101110.

ΛΥΣΗ

$(101110)_2 = 1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 +$



ΛΥΣΗ

$$(101110)_2 = 1x2^5+0x2^4+1x2^3+1x2^2+1x2^1+0x2^0 = 32+0+8+4+2+0 = (46)_{10}$$

Μετατροπή δεκαδικού σε δυαδικό

Για τη μετατροπή δεκαδικού σε δυαδικό, στο 1° Κεφάλαιο των ασκήσεων προτείνεται η μετατροπή του 1231 στο δυαδικό σύστημα. Αφού κάνει κλικ ο μαθητής στο πλήκτρο «ΛΥΣΗ», αρχίζει η διαδικασία της διαδοχικής διαίρεσης του αριθμού με τη βάση του συστήματος στο οποίο θέλουμε να τον μετατρέψουμε (το 2).

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΔΥΑΔΙΚΕΣ ΕΝΝΟΙΕΣ"

ΑΣΚΗΣΗ | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Μετατρέψτε τον δεκαδικό αριθμό 1231 σε δυαδικό.

ΛΥΣΗ

	Πηλίκο	υπόλοιπο
1231/2	615	1
615/2	307	1

$(1231)_{10} = (\qquad\qquad)_2$

Αφού ολοκληρωθεί η διαδικασία της διαίρεσης, ο φωτεινός σηματοδότης γίνεται πράσινος και ο μαθητής προτρέπεται να κάνει κλικ στο πλήκτρο αυτό προκειμένου να δει το σχηματισμό του δυαδικού αριθμού από τα υπόλοιπα των διαιρέσεων (τα οποία λαμβάνονται κατά την αντίθετη φορά).

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΔΥΑΔΙΚΕΣ ΕΝΝΟΙΕΣ"

ΑΣΚΗΣΗ | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Μετατρέψτε τον δεκαδικό αριθμό 1231 σε δυαδικό.

ΛΥΣΗ

	Πηλίκο	υπόλοιπο
1231/2	615	1
615/2	307	1
307/2	153	1
153/2	76	1
76/2	38	0
38/2	19	0
19/2	9	1
9/2	4	1
4/2	2	0
2/2	1	0
1/2	0	1

$(1231)_{10} = (\qquad\qquad)_2$

Πατήστε για να δείτε το αποτέλεσμα

Κάνοντας κλικ ο μαθητής στο φωτεινό σηματοδότη αρχίζει ο σχηματισμός του αποτελέσματος.

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΔΥΑΔΙΚΕΣ ΕΝΝΟΙΕΣ"

ΑΣΚΗΣΗ | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Μετατρέψτε τον δεκαδικό αριθμό 1231 σε δυαδικό.

ΛΥΣΗ

	Πηλίκο	υπόλοιπο
1231/2	615	1
615/2	307	1
307/2	153	1
153/2	76	1
76/2	38	0
38/2	19	0
19/2	9	1
9/2	4	1
4/2	2	0
2/2	1	0
1/2	0	1

$(1231)_{10} = (\quad 111)_2$

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΔΥΑΔΙΚΕΣ ΕΝΝΟΙΕΣ"

ΑΣΚΗΣΗ | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Μετατρέψτε τον δεκαδικό αριθμό 1231 σε δυαδικό.

ΛΥΣΗ

	Πηλίκο	υπόλοιπο
1231/2	615	1
615/2	307	1
307/2	153	1
153/2	76	1
76/2	38	0
38/2	19	0
19/2	9	1
9/2	4	1
4/2	2	0
2/2	1	0
1/2	0	1

$(1231)_{10} = (0011001111)_2$

Μετατροπή δεκαεξαδικού σε δυαδικό και οκταδικό

Για τη μετατροπή ενός δεκαεξαδικού σε δυαδικό και οκταδικό προτείνεται στο μαθητή, στις ασκήσεις του 1ου Κεφαλαίου, η μετατροπή του F3A7C2. Κάνοντας κλικ στο πλήκτρο «ΛΥΣΗ», εμφανίζεται η αντιστοιχία κάθε ψηφίου του δεκαεξαδικού στο δυαδικό, και ο μαθητής προτρέπεται να κάνει κλικ στο φωτεινό σηματοδότη για να δει το σχηματισμό του αποτελέσματος. Ο δυαδικός σχηματίζεται από την παράθεση των αντίστοιχων στα δεκαεξαδικά ψηφία δυαδικών αριθμών, ενώ ο οκταδικός σχηματίζεται από τη μετατροπή σε οκταδικά ψηφία του δυαδικού αριθμού, παίρνοντας τα ψηφία του ανά τρία από το τέλος.

ΑΣΚΗΣΕΙΣ Σ ΤΟ ΚΕΦΑΛΑΙΟ "ΔΥΑΔΙΚΕΣ ΕΝΝΟΙΕΣ"

ΑΣΚΗΣΗ | >> | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Μετατρέψτε τον δεκαεξαδικό αριθμό F3A7C2 σε δυαδικό και οκταδικό.

ΛΥΣΗ

Μετατρέπουμε κάθε ψηφίο του δεκαεξαδικού αριθμού στον αντίστοιχο δυαδικό του και στη συνέχεια στον αντίστοιχο οκταδικό του παίρνοντας ανά τρία τα ψηφία του δυαδικού από το τέλος.

Δεκαεξαδικό ψηφίο	Δυαδικός αντίστοιχος
F	1111
3	0011
A	1010
7	0111
C	1100
2	0010

Επομένως έχουμε:

( ) $_{16}$ = ( ) $_{2}$ = ( ) $_{8}$

Κάντε κλίκ για να δείτε τη συνέχεια της λύσης

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΔΥΑΔΙΚΕΣ ΕΝΝΟΙΕΣ"

ΑΣΚΗΣΗ | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Μετατρέψτε τον δεκαεξαδικό αριθμό F3A7C2 σε δυαδικό και οκταδικό.

ΛΥΣΗ

Μετατρέπουμε κάθε ψηφίο του δεκαεξαδικού αριθμού στον αντίστοιχο δυαδικό του και στη συνέχεια στον αντίστοιχο οκταδικό του παίρνοντας ανά τρία τα ψηφία του δυαδικού από το τέλος.

Δεκαεξαδικό ψηφίο	Δυαδικός αντίστοιχος
F	1111
3	0011
A	1010
7	0111
C	1100
2	0010

Επομένως έχουμε:

(F3A7C2)$_{16}$=(111100111010011111000010)$_2$=(74723702)$_8$

7 4 7 2 3 7 0 2

Συμπλήρωμα ως προς βάση

Για την εύρεση συμπληρώματος ως προς βάση, προτείνεται στο μαθητή η εύρεση του συμπληρώματος ως προς 10 τεσσάρων δεκαδικών αριθμών. Αφού κάνει κλικ στο πλήκτρο «ΛΥΣΗ», παρουσιάζονται στο αριστερό μέρος της οθόνης οι δεκαδικοί αριθμοί και στο δεξιό τα συμπληρώματά τους ως προς 10. Ο μαθητής προτρέπεται να περάσει το ποντίκι πάνω από τα ψηφία των δεκαδικών για να δει με ποιο τρόπο σχηματίζονται αντίστοιχα τα ψηφία στα συμπληρώματά τους. Έτσι, αν περάσει το ποντίκι πάνω από το ψηφίο 7 του δεύτερου αριθμού, ανάβει το 3 στο συμπλήρωμά του και ταυτόχρονα ο κανόνας με τον οποίο υπολογίστηκε το 3.

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΔΥΑΔΙΚΕΣ ΕΝΝΟΙΕΣ"

ΑΣΚΗΣΗ | >> | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Βρείτε το συμπλήρωμα ως προς 10 των ακόλουθων εξαψήφιων δεκαδικών αριθμών: 123900, 090657, 100000, και 000000.

ΛΥΣΗ

Το συμπλήρωμα κάθε δεκαδικού ως προς 10 προκύπτει ως εξής:
(α) αφήνουμε όλα τα λιγότερο σημαντικά 0 αμετάβλητα,
(β) αφαιρούμε το πρώτο μη μηδενικό λιγότερο σημαντικό ψηφίο από το 10
(γ) αφαιρούμε όλα τα υπόλοιπα ψηφία από το 9.
(το συμπλήρωμα του 0 είναι το 0 εξ ορισμού).
Έτσι έχουμε: (Μετακινήστε το ποντίκι πάνω στα ψηφία των αριθμών)

Αριθμός	Συμπλήρωμα
123900	876100
090657	909343
100000	900000
000000	000000

Αν περάσει το ποντίκι πάνω από το ψηφίο 5 του δεύτερου αριθμού, ανάβει το 4 στο συμπλήρωμά του και ταυτόχρονα ο κανόνας βάσει του οποίου υπολογίστηκε το 4.

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΔΥΑΔΙΚΕΣ ΕΝΝΟΙΕΣ"

ΑΣΚΗΣΗ | >> | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Βρείτε το συμπλήρωμα ως προς 10 των ακόλουθων εξαψήφιων δεκαδικών αριθμών: 123900, 090657, 100000, και 000000.

ΛΥΣΗ

Το συμπλήρωμα κάθε δεκαδικού ως προς 10 προκύπτει ως εξής:
(α) αφήνουμε όλα τα λιγότερο σημαντικά 0 αμετάβλητα,
(β) αφαιρούμε το πρώτο μη μηδενικό λιγότερο σημαντικό ψηφίο από το 10
(γ) αφαιρούμε όλα τα υπόλοιπα ψηφία από το 9.
(το συμπλήρωμα του 0 είναι το 0 εξ ορισμού).
Έτσι έχουμε: (Μετακινήστε το ποντίκι πάνω στα ψηφία των αριθμών)

Αριθμός	Συμπλήρωμα
123900	876100
090657	909343
100000	900000
000000	000000

Ομοίως, αυτό μπορεί να γίνει για τα ψηφία και των τεσσάρων αριθμών.

Απλοποίηση συναρτήσεων Boole

Στις ασκήσεις του 2ου Κεφαλαίου «Άλγεβρα Boole και λογικές πύλες» προτείνονται διάφορες ασκήσεις για απλοποίηση συναρτήσεων Boole μέσω αλγεβρικών μετασχηματισμών. Η λύση των ασκήσεων αυτών παρουσιάζεται βηματικά. Ο μαθητής δηλ. προτρέπεται να κάνει κλικ στο ίσον (=) προκειμένου να δει τον επόμενο μετασχηματισμό, ενώ ταυτόχρονα πάνω από το ίσον εμφανίζεται το θεώρημα ή το αξίωμα βάσει του οποίου υλοποιείται ο μετασχηματισμός αυτός. Έτσι, έχουμε:

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΑΛΓΕΒΡΑ BOOLE ΚΑΙ ΛΟΓΙΚΕΣ ΠΥΛΕΣ"

ΑΣΚΗΣΗ | >> | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Απλοποιείστε τις παρακάτω συναρτήσεις Boole σε έναν ελάχιστο αριθμό παραγόντων.
(α) $x'y'+x'y+xy$ (β) $(x+y)(x+y')$ (γ) $x'y+xy'+xy+x'y'$

ΛΥΣΗ

Χρησιμοποιώντας τα αξιώματα και τα βασικά θεωρήματα έχουμε:

Επιμεριστική ιδιότητα

(α) $x'y'+x'y+xy=x'(y'+y)+xy=$

(β) $(x+y)(x+y')=$

(γ) $x'y+xy'+xy+x'y'=$

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΑΛΓΕΒΡΑ BOOLE ΚΑΙ ΛΟΓΙΚΕΣ ΠΥΛΕΣ"

ΑΣΚΗΣΗ | >> | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Απλοποιείστε τις παρακάτω συναρτήσεις Boole σε έναν ελάχιστο αριθμό παραγόντων:
(α) x'y'+x'y+xy (β) (x+y)(x+y') (γ) x'y+xy'+xy+x'y'

ΛΥΣΗ

Χρησιμοποιώντας τα αξιώματα και τα βασικά θεωρήματα έχουμε:

Ουδέτερο στοιχείο

(α) x'y'+x'y+xy=x'(y'+y)+xy=x'+xy=(x'+x)(x'+y)=1(x'+y)=x'+y

(β) (x+y)(x+y')=

(γ) x'y+xy'+xy+x'y'=

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΑΛΓΕΒΡΑ BOOLE ΚΑΙ ΛΟΓΙΚΕΣ ΠΥΛΕΣ"

ΑΣΚΗΣΗ | >> | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

Απλοποιείστε τις παρακάτω συναρτήσεις Boole σε έναν ελάχιστο αριθμό παραγόντων:
(α) x'y'+x'y+xy (β) (x+y)(x+y') (γ) x'y+xy'+xy+x'y'

ΛΥΣΗ

Χρησιμοποιώντας τα αξιώματα και τα βασικά θεωρήματα έχουμε:

(α) x'y'+x'y+xy=x'(y'+y)+xy=x'+xy=(x'+x)(x'+y)=1(x'+y)=x'+y

(β) (x+y)(x+y')=

Ουδέτερο στοιχείο

(γ) x'y+xy'+xy+x'y'=y(x+x')+y'(x+x')=y.1+y'.1=y+y'=

Λογικά διαγράμματα με πύλες

Στις ασκήσεις του 2ου Κεφαλαίου προτείνονται στο μαθητή ασκήσεις για το μετασχηματισμό αλγεβρικών εκφράσεων σε λογικά διαγράμματα με πύλες.

Σε μια τέτοια άσκηση, αφού ο μαθητής κάνει κλικ στο πλήκτρο «ΛΥΣΗ» παρουσιάζεται η ζητούμενη τελική μορφή της αλγεβρικής εκφράσεως και ο μαθητής προτρέπεται να κάνει κλικ στο σχετικό πλήκτρο για να δει το λογικό διάγραμμα των πυλών.

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΑΛΓΕΒΡΑ BOOLE ΚΑΙ ΛΟΓΙΚΕΣ ΠΥΛΕΣ"

ΑΣΚΗΣΗ | >> | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

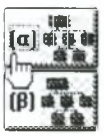
Χρησιμοποιώντας το θεώρημα De Morgan, μετατρέψτε τις παρακάτω συναρτήσεις Boole σε ισοδύναμες συναρτήσεις που να περιέχουν μόνο πράξεις OR και συμπληρώματος. Δείξτε ότι οι συναρτήσεις μπορούν να εκφραστούν με λογικά διαγράμματα που θα έχουν μόνο πύλες OR και αντιστροφείς:
(α) F=x'y'+x'z+y'z (β) F=(y+z')(x+y)(y'+z)

ΛΥΣΗ

Εφαρμόζοντας το θεώρημα De Morgan έχουμε:

(α) F=x'y'+x'z+y'z=(x+y)'+(x+z')'+(y+z')'

(β) F=(y+z')(x+y)(y'+z)=((y+z')'+(x+y)'+(y+z')')'



Εκφράσαμε δηλ. τις συναρτήσεις μόνο με πράξεις OR και συμπληρώματος. Το λογικό διάγραμμα για κάθε συνάρτηση με πύλες OR και αντιστροφείς προκύπτει πολύ εύκολα...

Κάντε κλίκ για να δείτε το λογικό διάγραμμα

ΑΣΚΗΣΕΙΣ ΣΤΟ ΚΕΦΑΛΑΙΟ "ΑΛΓΕΒΡΑ BOOLE ΚΑΙ ΛΟΓΙΚΕΣ ΠΥΛΕΣ"

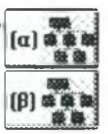
ΑΣΚΗΣΗ | >> | Θεωρία | < | > | Αλλαγή κατηγορίας | Αξιολόγηση

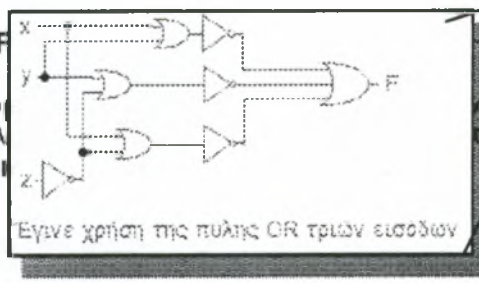
Χρησιμοποιώντας το θεώρημα De Morgan, μετατρέψτε τις παρακάτω συναρτήσεις Boole σε ισοδύναμες συναρτήσεις που να περιέχουν μόνο πράξεις OR και συμπληρώματος. Δείξτε ότι οι συναρτήσεις μπορούν να εκφραστούν με λογικά διαγράμματα που θα έχουν μόνο πύλες OR και αντιστροφείς:
(α) F=x'y'+x'z+y'z (β) F=(y+z')(x+y)(y'+z)

ΛΥΣΗ

Εφαρμόζοντας το θεώρημα De Morgan έχουμε:

(α) F=x'y'+x'z+y'z=(x+y)'+(x+z')'+(y+z')'





(β) F

+(y+z')'

Εκφ

ο με πράξεις OR και συμπληρώματος.

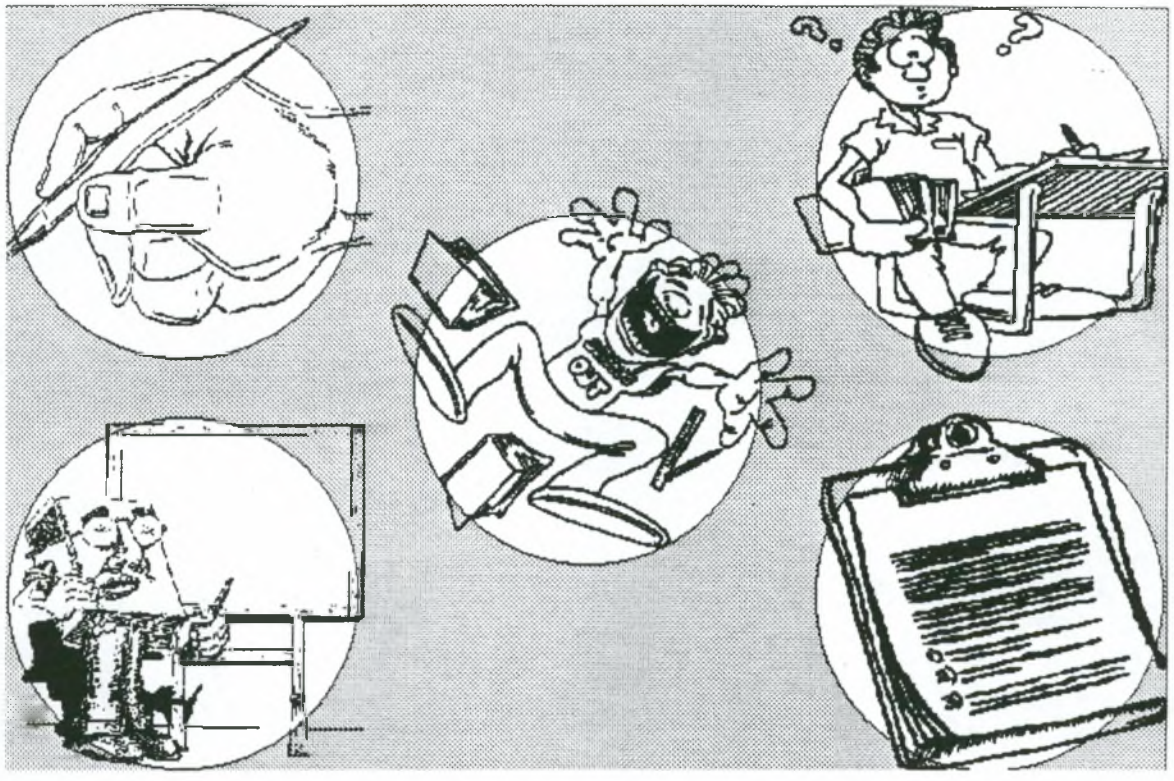
Το λ

ηση με πύλες OR και αντιστροφείς

προι

3.4 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΤΜΗΜΑΤΟΣ ΑΞΙΟΛΟΓΗΣΗΣ

3.4.1 Κυρίως Μενού



Στο Κυρίως Μενού διακρίνουμε 5 επιλογές:

Μετάβαση στο **Περιβάλλον Συγγραφής και Διόρθωσης Τεστ**. Ο καθηγητής στο περιβάλλον αυτό δημιουργεί τη δικιά του Βάση Δεδομένων με ερωτήσεις και στη συνέχεια επιλέγει τις ερωτήσεις που θα αποτελούν το Τεστ. Επίσης, έχει τη δυνατότητα να επιλέξει κάποιες ερωτήσεις από μία υπάρχουσα Βάση για να δημιουργήσει ένα τεστ.

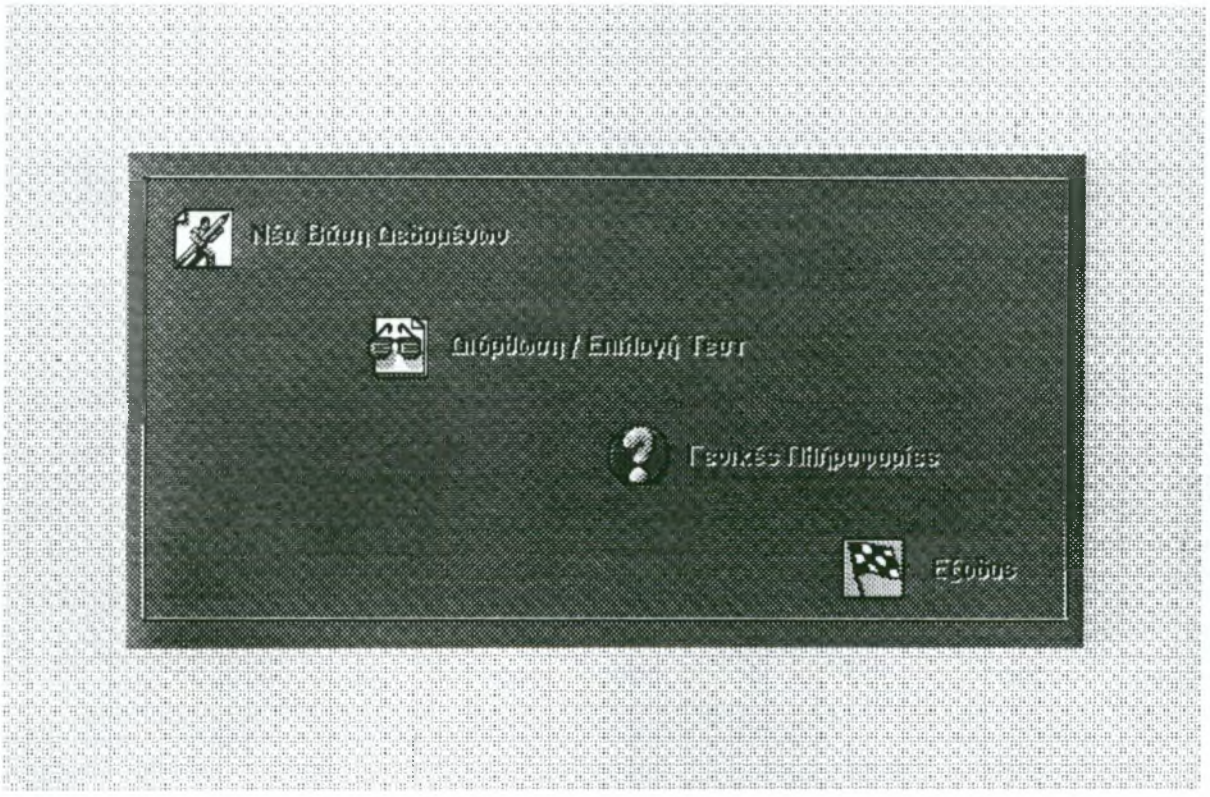
Μετάβαση στο **Περιβάλλον Επίλυσης Τεστ**. Ο μαθητής στο περιβάλλον αυτό αφού επιλέξει το τεστ που επιθυμεί, εξετάζεται σ' αυτό και αξιολογείται. Τα αποτελέσματα κάθε μαθητή αποθηκεύονται σε μία Βάση Δεδομένων για περαιτέρω επεξεργασία και έλεγχο από τον καθηγητή.

Μετάβαση στο **Περιβάλλον Ελέγχου Αποτελεσμάτων**. Ο καθηγητής ελέγχει τα αποτελέσματα των μαθητών στα Τεστ στα οποία έχουν εξεταστεί.

Μετάβαση στο **Περιβάλλον Βοήθειας**. Ο χρήστης (καθηγητής ή μαθητής) μαθαίνει να χειρίζεται το πρόγραμμα με τη βοήθεια video clips.

3.4.2 Περιβάλλον Συγγραφής και Διόρθωσης Τεστ

Οθόνη Περιεχομένων



Περιλαμβάνει τις εξής επιλογές:

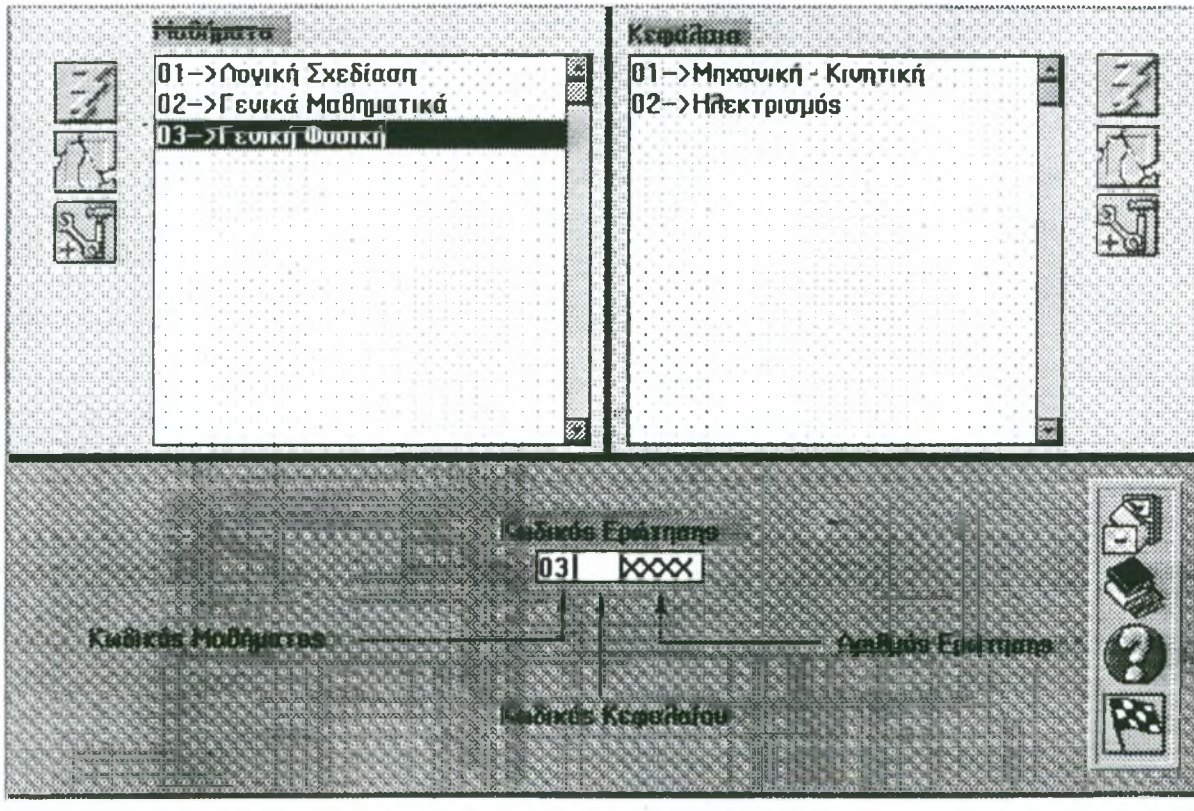
Νέα Βάση Δεδομένων. Ο καθηγητής δημιουργεί μία νέα βάση δεδομένων με ερωτήσεις. Αν η Βάση υπάρχει ήδη, δίνεται η δυνατότητα στον καθηγητή να προσθέσει νέες ερωτήσεις ή να διαγράψει κάποιες από τις υπάρχουσες.

Διόρθωση / Επιλογή Τέστ. Ο καθηγητής ανοίγει μία υπάρχουσα βάση δεδομένων και επιλέγει απ' αυτήν τις ερωτήσεις που θέλει να αποτελούν το Τεστ. Επίσης, αν επιθυμεί μπορεί να διορθώσει τις ερωτήσεις και τις απαντήσεις που υπάρχουν σ' αυτήν. Στην κατάσταση αυτή δεν υπάρχει η δυνατότητα πρόσθεσης νέων ερωτήσεων.

Γενικές Πληροφορίες. Εμφανίζει σε αναδυόμενο παράθυρο τις λειτουργίες κάθε πλήκτρου που εμφανίζεται στην τρέχουσα οθόνη.

Έξοδος. Κλείνει το τρέχον περιβάλλον και επιστρέφει στο Κυρίως Μενού

Οθόνη Δημιουργίας Κωδικού:



Για να δημιουργηθεί μία καινούργια ερώτηση ο καθηγητής πρέπει να καθορίσει το μάθημα και το κεφάλαιο στο οποίο θα ανήκει η ερώτηση. Η επιλογή του αντίστοιχου μαθήματος και κεφαλαίου από τις αντίστοιχες λίστες είναι χρήσιμη μόνο στην περίπτωση της αναζήτησης. Ο κωδικός του μαθήματος (διψήφιος αριθμός) και ο κωδικός του κεφαλαίου (διψήφιος αριθμός) αποτελούν τα τέσσερα πρώτα ψηφία του κωδικού κάθε ερώτησης. Τα υπόλοιπα τέσσερα είναι ο αύξων αριθμός της ερώτησης. Π.χ. η ερώτηση του μαθήματος Γενική Φυσική (με κωδικό **03**) και του κεφαλαίου Ηλεκτρισμός (με κωδικό **02**) και αύξοντα αριθμό **12** θα έχει τελικό κωδικό το **03020012**. Τον αριθμό αυτόν δεν τον βλέπει ο εξεταζόμενος αλλά βηθάει τον καθηγητή στην αναζήτηση της συγκεκριμένης ερώτησης.

Τα στοιχεία αυτής της οθόνης είναι οι δύο λίστες, αντίστοιχα με μαθήματα και κεφάλαια και τα εξής πλήκτρα:

Δημιουργία νέου μαθήματος (ή νέου κεφαλαίου)

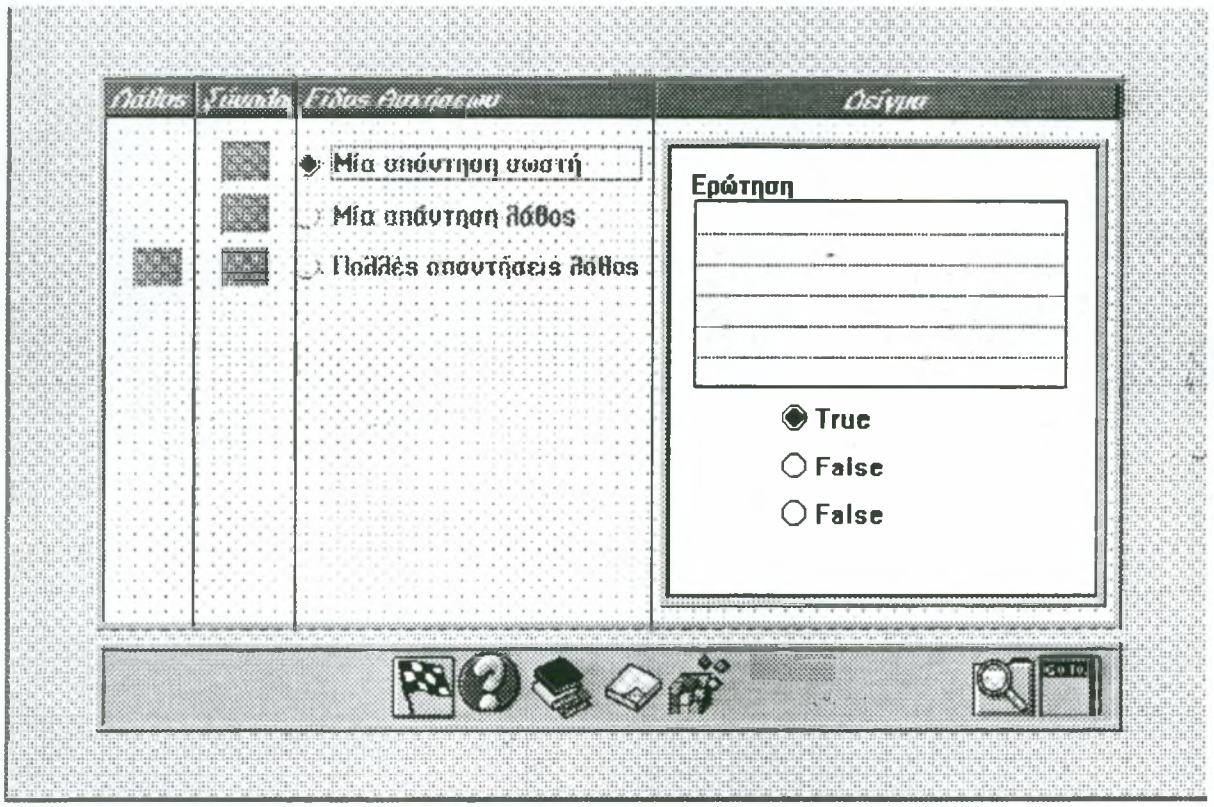
Διαγραφή του επιλεγμένου μαθήματος (ή κεφαλαίου)

Μετονομασία του επιλεγμένου μαθήματος (ή κεφαλαίου)

Μετάβαση στην **Οθόνη Επιλογής Τύπου Ερώτησης**. Για να γίνει η μετάβαση σ'αυτή την οθόνη πρέπει ο καθηγητής να έχει δημιουργήσει σωστά τα τέσσερα πρώτα ψηφία του κωδικού. Με άλλα λόγια θα πρέπει να έχει επιλέξει κάποιο μάθημα και κάποιο κεφάλαιο.

Επιστροφή στην **Οθόνη Περιεχομένων**.

<u>Οθόνη Επιλογής Τύπου Ερώτησης:</u>



Στην οθόνη αυτή ο καθηγητής μπορεί να επιλέξει το είδος της ερώτησης που επιθυμεί. Αν δηλαδή στις 5 πιθανές απαντήσεις θα περιέχονται:

- μία απάντηση Αληθής (true) και οι υπόλοιπες Ψευδείς (false)
- μία απάντηση Ψευδής (false) και οι υπόλοιπες Αληθείς (true)

- πολλές απαντήσεις Αληθείς (true) και οι υπόλοιπες Ψευδείς (false)

Στην τρίτη περίπτωση μπορεί ο καθηγητής να δηλώσει και τον αριθμό των ψευδών απαντήσεων.

Στο δεξί μέρος της οθόνης υπάρχει το αντίστοιχο δείγμα για τον κάθε τύπο ερώτησης. Τα πλήκτρα που βρίσκονται στη βάση της οθόνης κάνουν τις εξής λειτουργίες:

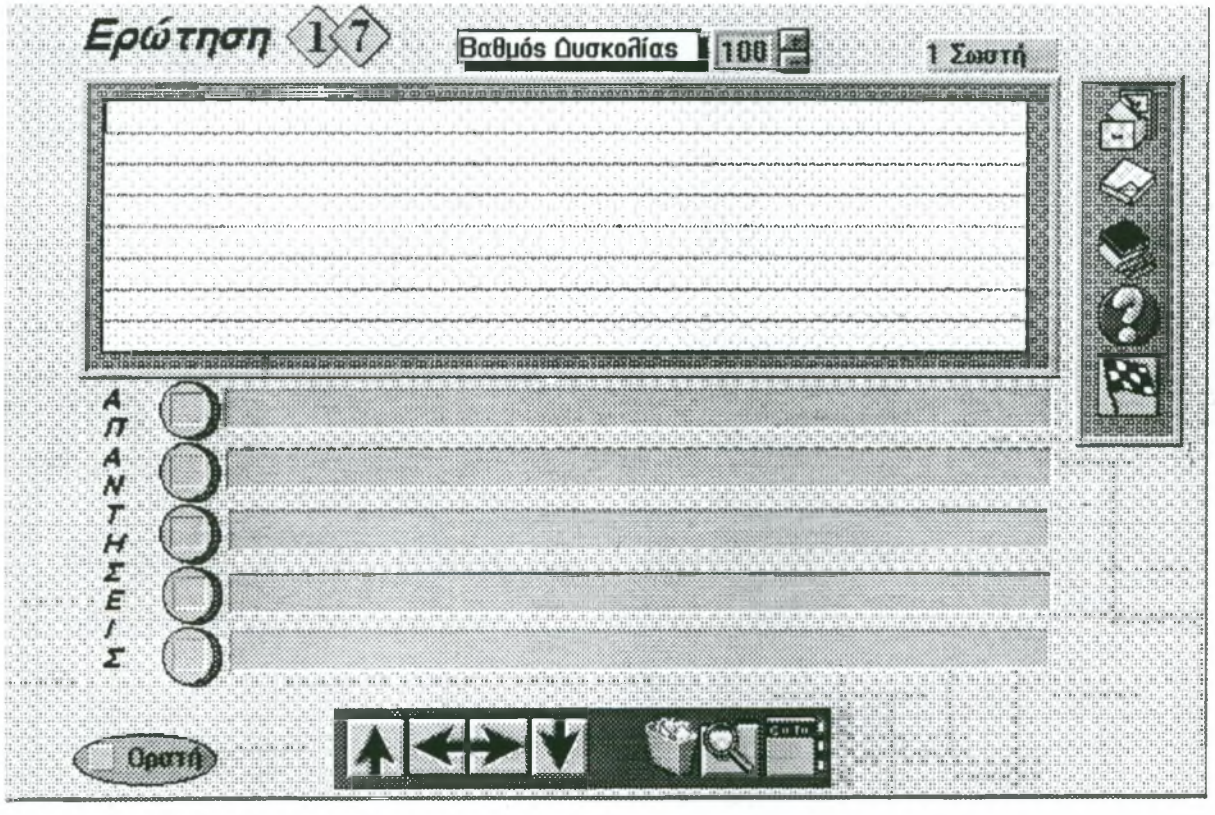
Επιστρέφει στην **Οθόνη Δημιουργίας Κωδικού**

Δημιουργεί μία κενή ερώτηση στη Βάση Δεδομένων και μεταβαίνει στην **Οθόνη Εισαγωγής Ερώτησης**.

Εμφανίζει το πλαίσιο διαλόγου **Μετάβαση** (θα εξηγηθεί παρακάτω)

Εμφανίζει το πλαίσιο διαλόγου **Αναζήτηση** (θα εξηγηθεί παρακάτω).

Οθόνη Εισαγωγής Ερώτησης



Η οθόνη αυτή είναι η βασικότερη του **Περιβάλλοντος Συγγραφής και Διόρθωσης Τεστ**. Ο καθηγητής εδώ συμπληρώνει όλα τα στοιχεία τα οποία θα απαρτίσουν μία ολοκληρωμένη ερώτηση του τεστ.

Πιο συγκεκριμένα, ο καθηγητής αφού γράψει το κείμενο της ερώτησης και των πέντε πιθανών απαντήσεων, συμπληρώνει τις θέσεις αριστερά της ερώτησης με το ανάλογο γράμμα. Δηλαδή T (true) ή F (false) αν η ερώτηση είναι αληθής ή ψευδής

αντίστοιχα. Στη συνέχεια με τη βοήθεια του εργαλείου αυξομείωσης δίνει στην ερώτηση ένα επίπεδο δυσκολίας (από 10 - 100). Οι παραπάνω ενέργειες γίνονται για την εισαγωγή μίας νέας ερώτησης στη Βάση Δεδομένων. Αν ο καθηγητής επιθυμεί η συγκεκριμένη ερώτηση να συμπεριληφθεί στο Τεστ στο οποίο θα εξεταστεί ο μαθητής, αρκεί να τσεκάρει το πλήκτρο «Ορατή» στη βάση της οθόνης.

Για να φύγει από τη συγκεκριμένη κατάσταση εισαγωγής στοιχείων πρέπει να συμπληρώσει όλα τα στοιχεία της ερώτησης. Υπάρχει, βέβαια, και η δυνατότητα να σβύσει την ερώτηση με το ανάλογο εργαλείο.

Η οθόνη περιέχει τα εξής εργαλεία:

Δείχνει τον αύξοντα αριθμό της κάθε ερώτησης.

Θέση για να εισαχθεί **T (true)** ή **F (false)** και δηλώνει αν η απάντηση θα είναι αληθής ή ψευδής

Εργαλείο αυξομείωσης. Με τη βοήθειά του αλλάζει ο βαθμός δυσκολίας της ερώτησης.

Σβύνει την τρέχουσα ερώτηση της Βάσης Δεδομένων των Ερωτήσεων.

Πλήκτρο επιλογής. Όταν είναι τσεκαρισμένο η συγκεκριμένη ερώτηση θα συμπεριληφθεί στο Τεστ που θα δημιουργηθεί.

- *Τέσσερα πλήκτρα πλοήγησης στη Βάση Δεδομένων*

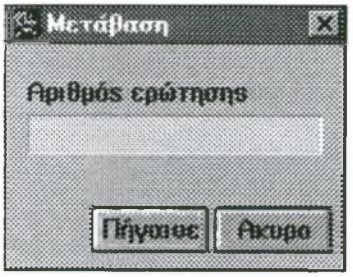
Εμφανίζει την πρώτη ερώτηση της Βάσης Δεδομένων των Ερωτήσεων

Εμφανίζει την προηγούμενη ερώτησης της Βάσης Δεδομένων των Ερωτήσεων.

Εμφανίζει την επόμενη ερώτηση της Βάσης Δεδομένων των Ερωτήσεων (ή δημιουργεί μία καινούργια κενή ερώτηση)

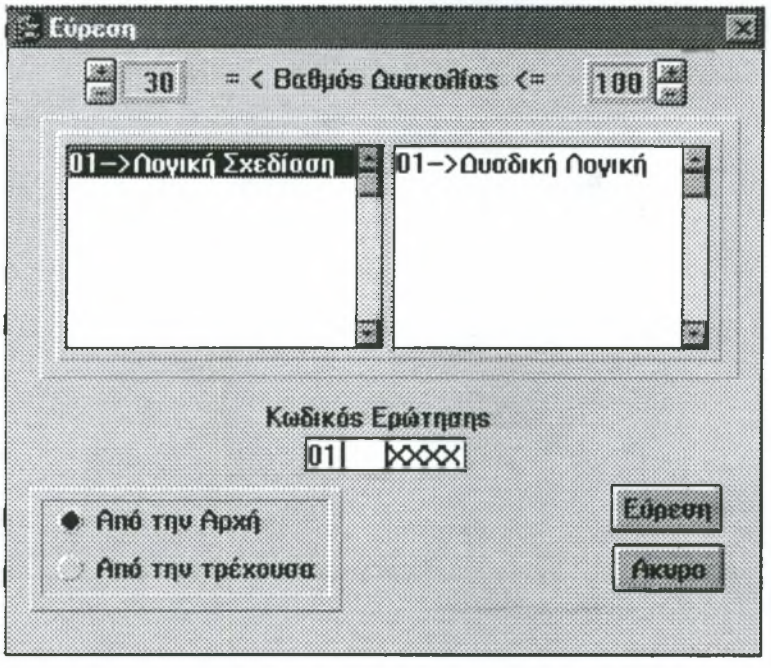
Εμφανίζει την τελευταία ερώτηση της Βάσης Δεδομένων των Ερωτήσεων

<u>Το πλαίσιο διαλόγου Μετάβαση</u>



Πατώντας το πλήκτρο μετάβασης (παρουσιάστηκε παραπάνω) εμφανίζεται το πλαίσιο διαλόγου **Μετάβαση.** Δίνοντας τον αριθμό της ερώτησης, ο καθηγητής, μπορεί να μεταβεί σ' αυτήν και, είτε να την επιλέξει για το Τεστ είτε να την τροποποιήσει.

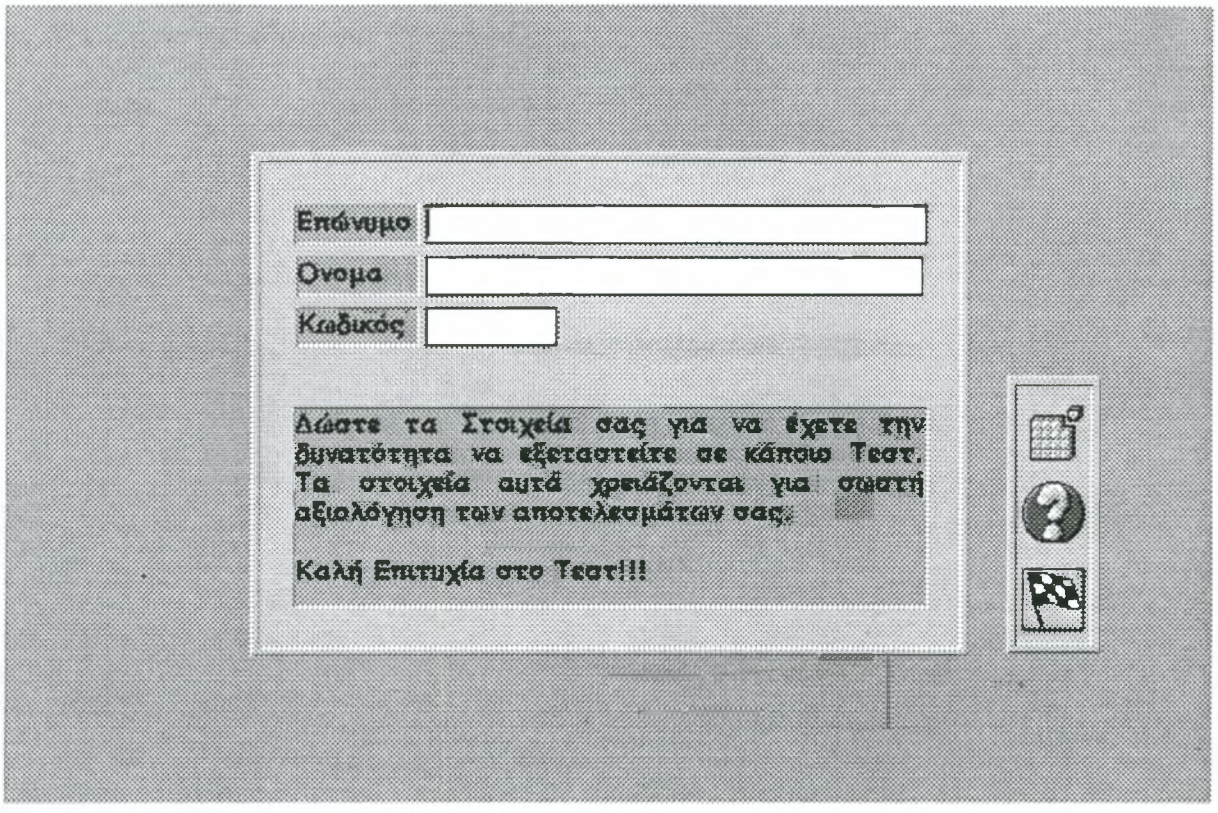
Το πλαίσιο διαλόγου Εύρεση



Εμφανίζεται επιλέγοντας το πλήκτρο εύρεσης (αναφέρθηκε παραπάνω). Με τη βοήθειά του δίνεται η δυνατότητα στον καθηγητή να αναζητήσει κάποια ερώτηση με τα χαρακτηριστικά που επιυμεί. Πιο συγκεκριμένα, μπορεί να επιλέξει από τις δύο λίστες το μάθημα και το κεφάλαιο στο οποίο ανήκει η ερώτηση και το επίπεδο δυσκολίας της. Η αναζήτηση μπορεί να γίνει είτε από την αρχή της Βάσης Δεδομένων είτε από την τρέχουσα ερώτηση. Μόλις βρεθεί η ερώτηση που πληρεί τα χαρακτηριστικά που έδωσε ο καθηγητής αυτόματα γίνεται η μετάβαση σ'αυτήν.

3.4.3 Περιβάλλον Επίλυσης Τεστ

Οθόνη Στοιχείων Εξεταζομένου



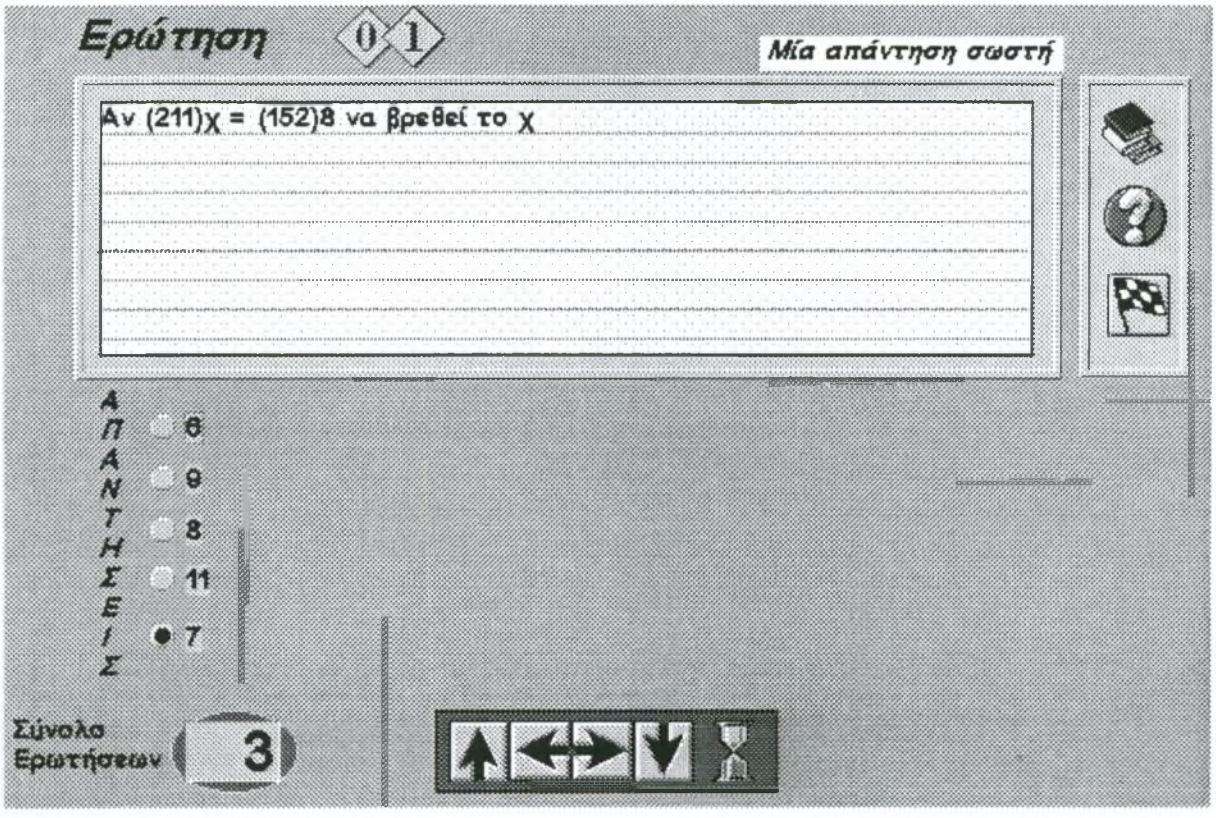
Στην πρώτη οθόνη του **Περιβάλλοντος Επίλυσης του Τεστ** ο μαθητής δίνει τα στοιχεία του (Επώνυμο, Όνομα, Κωδικός) και αφού επιλέξει το Τέστ στο οποίο επθυμεί να εξεταστεί, μεταβαίνει στην **Οθόνη Εξέτασης Ερώτησης**. Τα πλήκτρα που βρίσκονται στην οθόνη αυτή είναι:

Εξετάζει αν ο μαθητής έχει δώσει όλα του τα στοιχεία, επιλέγει κάποιο τέστ (αρχείο με επέκταση *.TFS) στο οποίο θα εξεταστεί, και μεταβαίνει στην **Οθόνη Εξέτασης Ερώτησης**.

Επιστροφή στην **Οθόνη Στοιχείων Εξεταζομένου**.

Τα υπόλοιπα πλήκτρα έχουν εξεταστεί παραπάνω.

Οθόνη Εξέτασης Ερώτησης



Η οθόνη αυτή μοιάζει αρκετά με την οθόνη **Οθόνη Εισαγωγής Ερώτησης** του Περιβάλλοντος Συγγραφής και Διόρθωσης Τεστ. Τα νέα στοιχεία που υπάρχουν εδώ είναι τα εξής:

Δείκτης που δείχνει το συνόλο των ερωτήσεων του Τεστ

Εμφανίζει το Πλαίσιο διαλόγου **Ρολόϊ - Χρονόμετρο** στο οποίο ο μαθητής μπορεί να δει την ώρα του συστήματος και τον χρόνο που του απομένει για να λύσει το Τεστ.

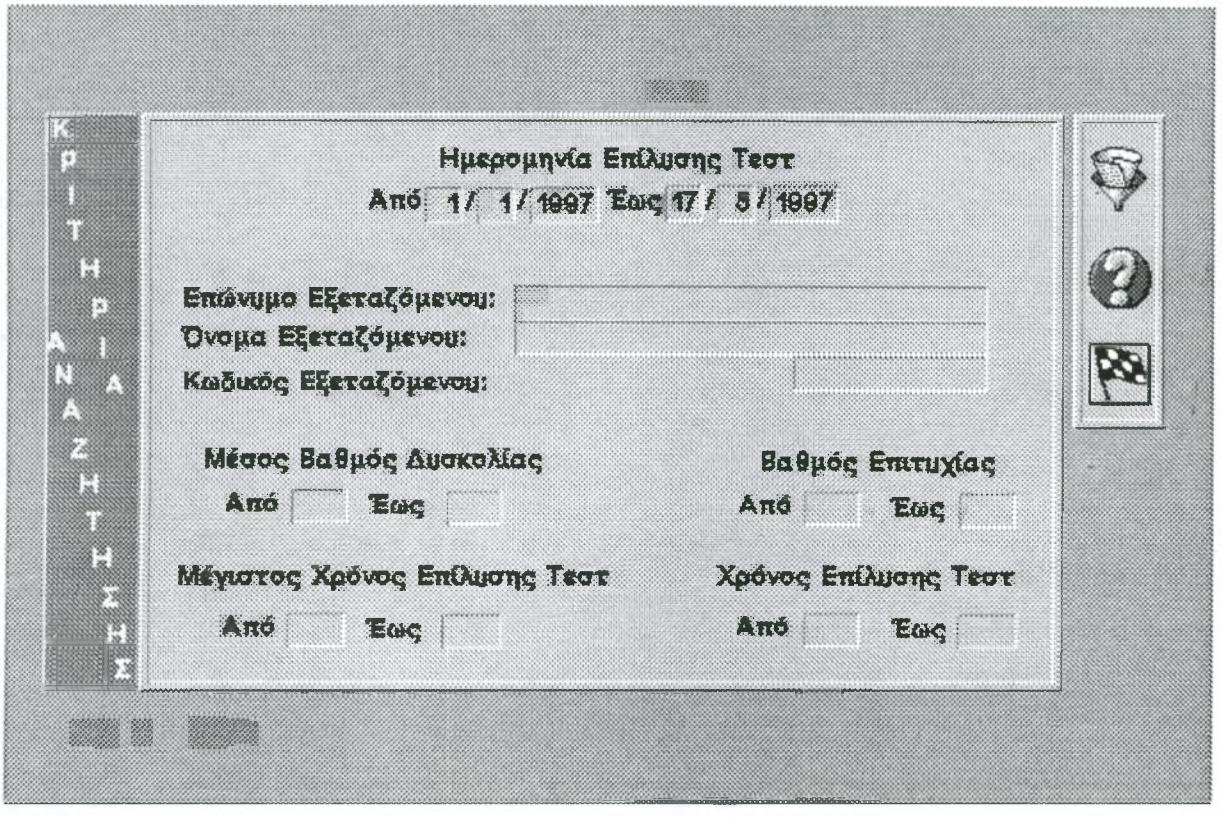
Τα υπόλοιπα πλήκτρα έχουν περιγραφεί παραπάνω και δεν αναφέρονται εδώ.

Το παράθυρο Ρολόϊ - Χρονόμετρο

Πατώντας το πλήκτρο της κλεψύδρας στην μπάρα εργαλειών στη βάση της οθόνης, εμφανίζεται το παράθυρο **Ρολόϊ - Χρονόμετρο**. Σ' αυτό ο μαθητής βλέπει την ώρα του συστήματος και τον χρόνο που του μένει για να τελειώσει το Τεστ.

3.4.4 Περιβάλλον Ελέγχου Αποτελεσμάτων

Οθόνη Κριτηρίων



Εδώ ο καθηγητής εισάγει κάποια στοιχεία (κριτήρια αναζήτησης) ώστε να δεί τα αποτελέσματα των Μαθητών σε κάποιο Τεστ. Πιο συγκεκριμένα μπορεί να χρησιμοποιήσει το εύρος ημερομηνίας, τα στοιχεία του εξεταζόμενου (επώνυμο, όνομα, κωδικό), το εύρος του Μέσου Βαθμού Δυσκολίας, το Εύρος του Βαθμού

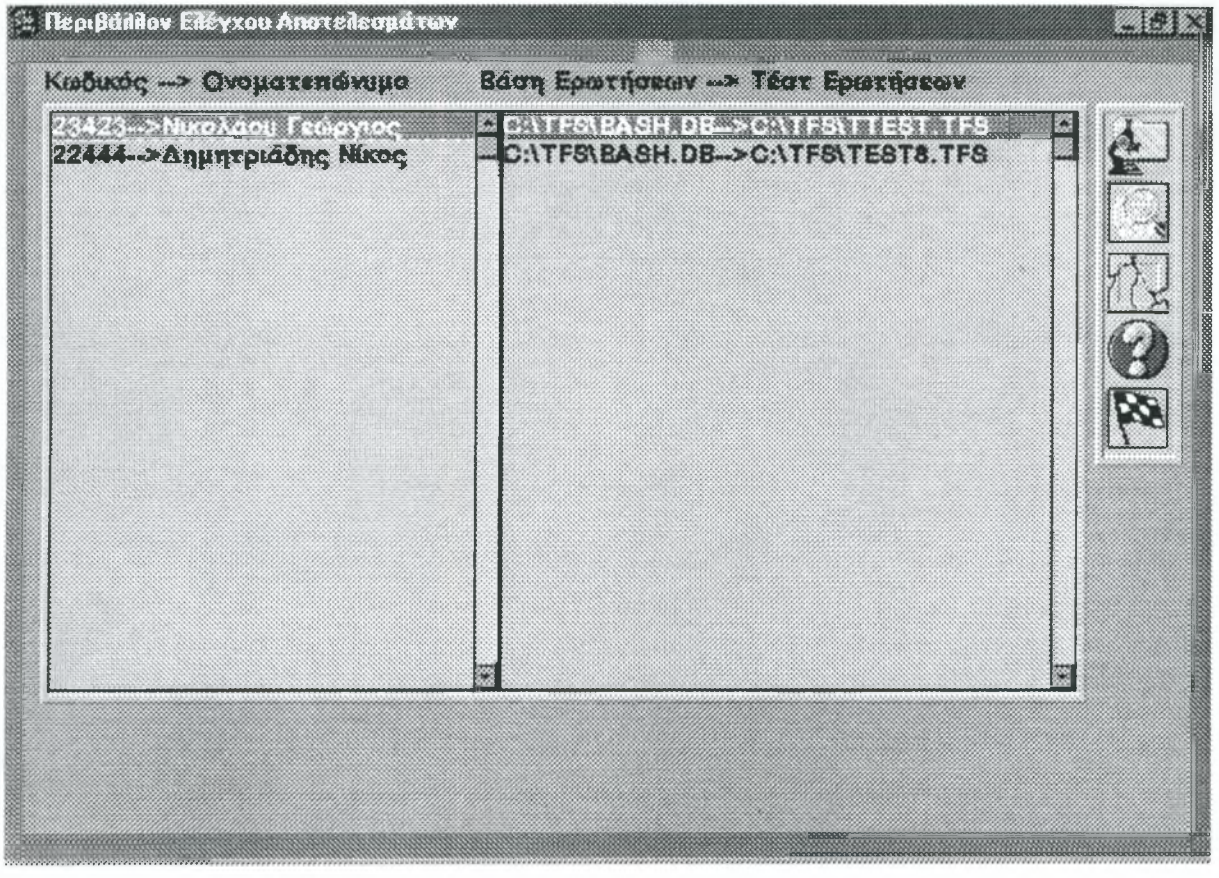
Επιτυχίας, το εύρος του Χρόνου Επίλυσης του Τεστ (ο χρόνος που χρειάστηκε ο Μαθητής για να τελειώσει το τεστ) και το εύρος του Μέγιστου Χρόνου Επίλυσης Τεστ (ο χρόνος που έχει καθορισθεί από τον καθηγητή σαν χρόνος επίλυσης του τεστ).

Τα στοιχεία που δίνει ο καθηγητής είναι προαιρετικά, δηλαδή αν κανένα κριτήριο δεν καθοριστεί, τότε στην **Οθόνη Επιλογής Τεστ** που ακολουθεί θα εμφανίζονται όλα τα Τεστ που έχουν δοθεί. Τα πλήκτρα της οθόνης είναι:

Φιλτράρει όλα τα στοιχεία (τα τεστ που υπάρχουν αποθηκευμένα σε κάποια βάση δεδομένων) σύμφωνα με τα κριτήρια που έχει δώσει ο καθηγητής.

Τα υπόλοιπα πλήκτρα έχουν εξηγηθεί παραπάνω.

Οθόνη Επιλογής Τεστ



Στην οθόνη αυτή διακρίνουμε δύο λίστες με στοιχεία από τα τεστ που έχουν ικανοποιήσει τα κριτήρια της προηγούμενης οθόνης. Στην αριστερή λίστα φαίνεται ο κωδικός και το ονοματεπώνυμο του εξεταζόμενου ενώ στην δεξιά το όνομα του Τεστ και το όνομα της Βάσης απ' όπου προέρχεται το Τεστ. Ο καθηγητής επιλέγει κάποιο από τα στοιχεία της αριστερής λίστας, για να δεί το αποτέλεσμα που έχει φέρει ο εξεταζόμενος ή για να το διαγράψει.

Τα πλήκτρα αυτής της οθόνης είναι

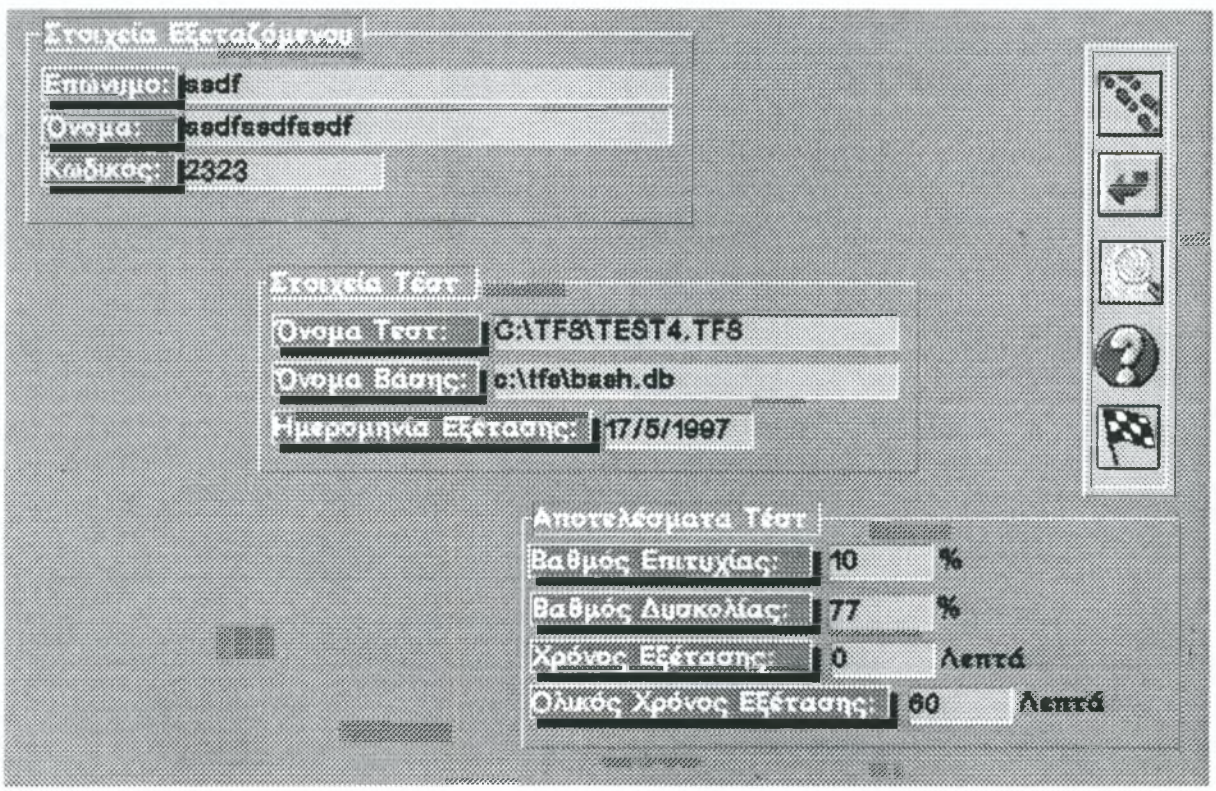
Μεταβαίνει στην **Οθόνη Αποτελεσμάτων** για το συγκεκριμένο τεστ που έχει επιλεχθεί από την αριστερή λίστα.

Επιστρέφει στην **Οθόνη Κριτηρίων** για να δώσει ο καθηγητής καινούργια κριτήρια αναζήτησης

Διαγράφει τα επιλεγμένα στοιχεία στις δύο λίστες.

Τα υπόλοιπα πλήκτρα έχουν εξηγηθεί παραπάνω.

Οθόνη Αποτελεσμάτων



Η Οθόνη Αποτελεσμάτων χωρίζεται σε τρία τμήματα. Το πρώτο έχει τα στοιχεία του εξεταζόμενου (επώνυμο, όνομα, κωδικός), το δεύτερο τα στοιχεία του τεστ (όνομα τεστ, όνομα βάσης και ημερομηνία εξέτασης) και το τρίτο τα αποτελέσματα του τεστ (βαθμός επιτυχίας, βαθμός δυσκολίας, χρόνος εξέτασης και ολικός χρόνος εξέτασης).

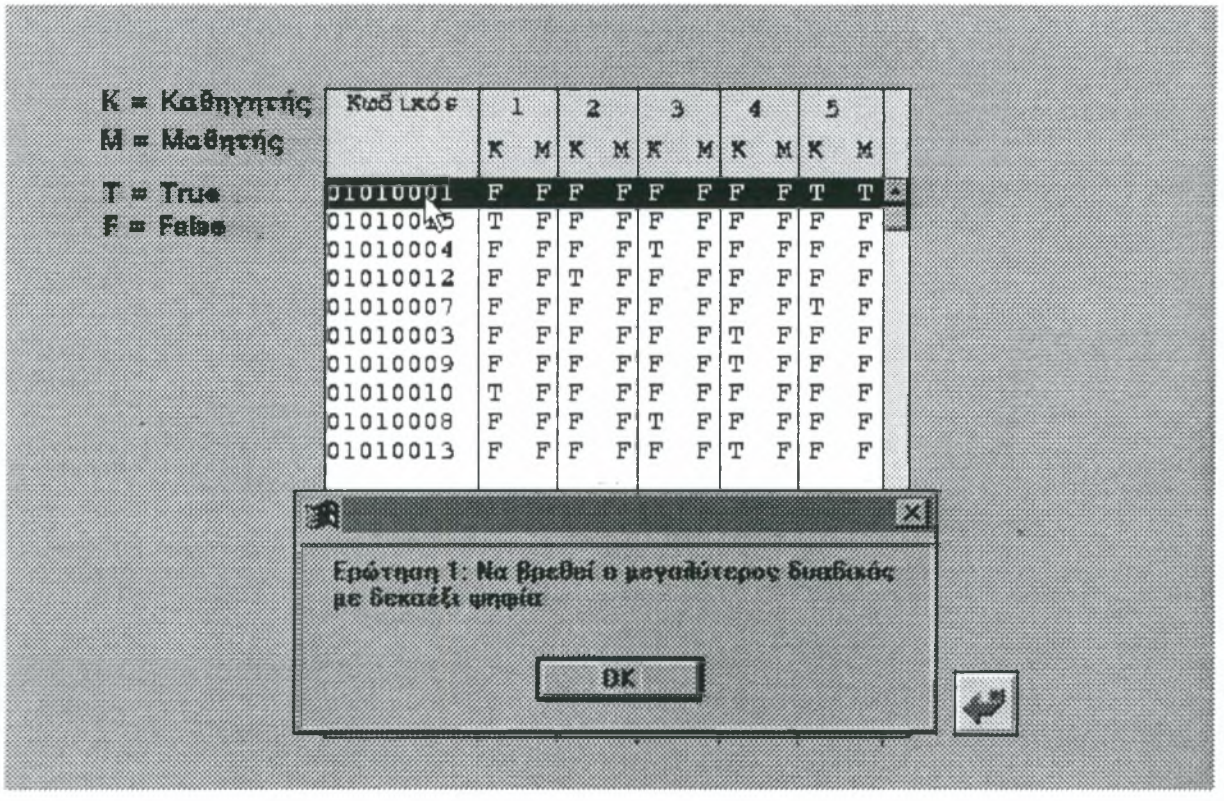
Τα πλήκτρα αυτής της οθόνης είναι:

Μεταβαίνει στην **Οθόνη Απαντήσεων**

Μεταβαίνει στην προηγούμενη Οθόνη

Τα υπόλοιπα πλήκτρα έχουν εξηγηθεί παραπάνω

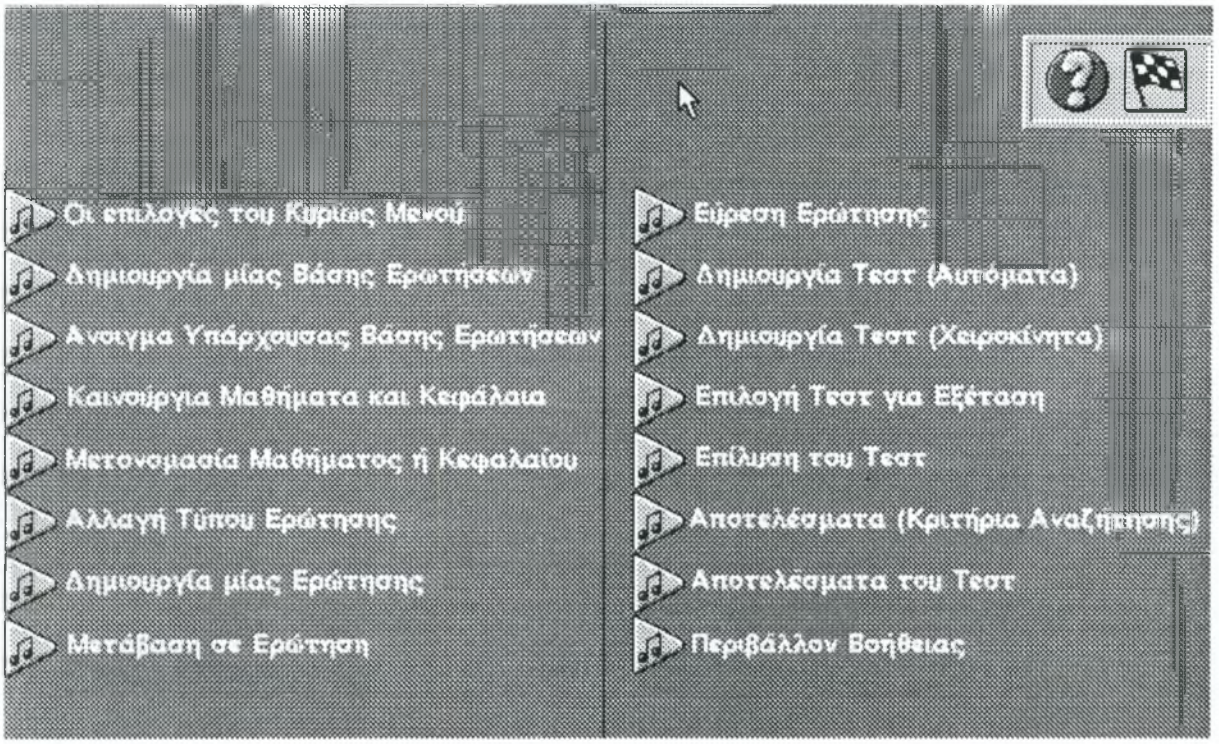
Οθόνη Απαντήσεων



Στην **Οθόνη Απαντήσεων** εμφανίζεται μία λίστα με όλες τις ερωτήσεις και όλες τις απαντήσεις της ερώτησης του τεστ. Στην πρώτη στήλη φαίνεται ο κωδικός της κάθε ερώτησης, και στις υπόλοιπες 5 στήλες οι απαντήσεις της ερώτησης. Η πρώτη τιμή (T ή F) της κάθε απάντησης είναι η απάντηση που όρισε ο καθηγητής ενώ η δεύτερη η απάντηση που έδωσε ο εξεταζόμενος. Αν ο καθηγητής κάνει κλικ με το mouse σε κάποια ερώτηση εμφανίζεται στην οθόνη του το κείμενο αυτής της ερώτησης από την Βάση Δεδομένων των Ερωτήσεων.

Τα πλήκτρα της οθόνης έχουν εξηγηθεί παραπάνω.

3.4.5 Περιβάλλον Βοήθειας



Στο Περιβάλλον Βοήθειας εμφανίζεται μία Οθόνη με τις κυριότερες εργασίες του τμήματος αξιολόγησης. Η τρόπος που εκτελείται κάθε εργασία παρουσιάζεται με τη βοήθεια ενός video clip. Οποιαδήποτε στιγμή θέλει ο χρήστης μπορεί να διακόψει το βίντεο πατώντας το πλήκτρο ESC.

Τα πλήκτρα αυτής της οθόνης είναι:

Εμφανίζει το βίντεο κλιπ με το αντίστοιχο θέμα.

ΚΕΦΑΛΑΙΟ 4: Ο ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΗΣ ΨΗΦΙΑΚΗΣ ΣΧΕΔΙΑΣΗΣ

Στο κεφάλαιο αυτό παρουσιάζονται θέματα που έχουν σχέση με τον προγραμματισμό της εφαρμογής. Λόγω της μεγάλης έκτασης, του αριθμού των αντικειμένων και του όγκου του συνδεδεμένου με αυτά κώδικα, επιχειρείται σχολιασμός του κώδικα των βασικότερων σημείων και των τμημάτων εκείνων που παρουσιάζουν το μεγαλύτερο ενδιαφέρον από άποψη προγραμματισμού.

4.1 ΜΗΝΥΜΑΤΑ ΤΟΥ ΤΒ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΕΙ Η ΕΦΑΡΜΟΓΗ

Τα Μηνύματα (συμβάντα) του ΤΒ που χρησιμοποιούνται σ' όλη την εφαρμογή (με αλφαβητική σειρά) είναι τα εξής:

- **ButtonClick**

Στέλνεται στο αντικείμενο όταν ο χρήστης κάνει κλικ επάνω του με το mouse. Αν ο χρήστης πατήσει το mouse πάνω στο αντικείμενο και το απελευθερώσει έξω από το αντικείμενο, τότε το συμβάν δεν απενεργοποιείται.

- **ButtonDown**

Στέλνεται στο αντικείμενο όταν ο χρήστης πιέσει το αριστερό πλήκτρο του mouse επάνω του. (Το μήνυμα buttonClick στέλνεται αφού ο χρήστης πιέσει και μετά απελευθερώσει το αριστερό πλήκτρο του mouse).

- **ButtonUp**

Στέλνεται στο αντικείμενο όταν ο χρήστης πιέσει και μετά απελευθερώσει το αριστερό πλήκτρο του mouse επάνω του (με την απελευθέρωση). Η σειρά των μηνυμάτων που στέλνονται στο αντικείμενο που δέχεται κλικ από το mouse είναι buttonDown - buttonUp - buttonClick.

- **Clear**

Σβύνει την τρέχουσα επιλογή δίχως να την τοποθετεί στον Clipboard.

- **Cut**

Σβύνει την τρέχουσα επιλογή και την τοποθετεί στον Clipboard, αντικαθιστώντας τα ήδη υπάρχοντα περιεχόμενα του Clipboard.

- **EnterApplication**

Στέλνεται όταν ένα Βιβλίο ανοίγει στο Κυρίως Παράθυρο (Main window) του ToolBook. Αυτό το μήνυμα στέλνεται στην σελίδα που εμφανίζεται στο Κυρίως Παράθυρο, λίγο πριν το μήνυμα EnterBook.

• **EnterPage**
Στέλνεται σε μία σελίδα όταν ο χρήστης μεταβαίνει σ' αυτήν τη σελίδα ή όταν ανοίγει το βιβλίο σ' αυτήν σελίδα, ή όταν οποιαδήποτε ενέργεια προκαλεί το Toolbook να μεταβεί σ' αυτήν τη σελίδα, για παράδειγμα, εμφανίζοντας έναν viewer ή εκτελώντας μία εντολή flip.

• **Idle**
Στέλνεται όταν δεν συμβαίνει τίποτα στο επίπεδο αναγνώστη (Reader Level). Το μήνυμα Idle στέλνεται στην τρέχουσα σελίδα.

• **KeyChar**
Στέλνεται στο αντικείμενο που έχει το επίκεντρο (focus), όταν ένα πλήκτρο πατιέται στο Επίπεδο Αναγνώστη. Αν κανένα αντικείμενο δεν έχει το focus, στέλνεται στην τρέχουσα σελίδα. Ανάλογα με το ποιο πλήκτρο πατήθηκε και ποιο αντικείμενο έχει το focus, το αποτέλεσμα είναι και διαφορετικό. Για παράδειγμα αν το focus είναι σε ένα πεδίο, η προκαθορισμένη απόκριση του ToolBook για αλφαριθμητικά πλήκτρα, είναι να εισάγει τον κατάλληλο χαρακτήρα στο πεδίο.
Ένα μήνυμα KeyChar δεν στέλνεται όταν ο χρήστης πατήσει ένα πλήκτρο συντόμευσης (π.χ. το F3), ή ένα πλήκτρο που δεν εμφανίζει κάποιον χαρακτήρα (π.χ. τα βελάκια). Για να ελεγχτεί η απόκριση όταν ο χρήστης πατά τέτοια πλήκτρα (όπως το F3) πρέπει να γραφεί κώδικας που να χειρίζεται το μήνυμα KeyDown ή το KeyUp, και όχι το KeyChar.

• **KeyUp**
Στέλνεται στο αντικείμενο με το focus, όταν ο χρήστης πατά κάποιο πλήκτρο στο Επίπεδο Αναγνώστη. Αν δεν υπάρχει αντικείμενο με focus, στέλνεται στην τρέχουσα σελίδα. Το μήνυμα KeyUp δεν στέλνεται όταν ο χρήστης πατά συνδυασμό πλήκτρων συντόμευσης (π.χ. το Ctrl+O που ισοδυναμεί με την εντολή Open του μενού File).

• **LeaveApplication**
Στέλνεται όταν ένα Βιβλίο κλείνει στο Κυρίως Παράθυρο του Toolbook. Το μήνυμα αυτό στέλνεται λίγο πριν το μήνυμα LeaveBook και μόνο στην σελίδα που εμφανίζεται στο Κυρίως Παράθυρο.

• **LeaveField**
Στέλνεται στο πεδίο που έχει το focus, ακριβώς πριν το λάβει κάποιο άλλο αντικείμενο.

• **LeavePage**
Στέλνεται ακριβώς πριν μία σελίδα λάβει το επίκεντρο (focus). Αυτό μπορεί να είναι αποτέλεσμα είτε πλοήγησης από σελίδα σε σελίδα είτε κλεισίματος ενός viewer. Στέλνεται στη σελίδα που χάνει το focus.

• **LeaveWindow**
Στέλνεται σε έναν viewer όταν ο χρήστης «μεταβεί» (switches) σε μία διαφορετική εφαρμογή ή κάνει κλικ σε έναν διαφορετικό viewer στην ίδια εφαρμογή.

• **MMNotify**
Στέλνεται όταν μία πολυμεσική εντολή τερματίσει. Το μήνυμα MMNotify στέλνεται στο αντικείμενο που καθορίζεται στην παράμετρο του notify της συγκεκριμένης πολυμεσικής εντολής.

• **MouseEnter**
Στέλνεται στο αντικείμενο όταν ο χρήστης περάσει το mouse πάνω από το αντικείμενο.

• **MouseLeave**
Στέλνεται σε ένα αντικείμενο όταν ο χρήστης ενώ είχε το mouse πάνω από το αντικείμενο το απομάκρυνε έξω από τα όριά του.

• **Paste**
Στέλνεται στην σελίδα όταν επιλέγεται το Paste από το μενού Edit. Είναι δυνατόν επίσης να σταλεί χρησιμοποιώντας την εντολή «send paste» Η προεπιλεγμένη απόκριση του ToolBook είναι να «επικολλεί» τα περιεχόμενα του Clipboard στο τρέχον Βιβλίο. Για να «επικολληθεί» ένα αντικείμενο χρησιμοποιώντας εντολές της γλώσσας OpenScript, πρέπει να τεθεί η ιδιότητα focus στην τιμή null και να εκτελεστεί η «send paste». Η ιδιότητα focus πρέπει να τεθεί στην τιμή null γιατί το ToolBook δεν μπορεί να επικολλήσει ένα αντικείμενο αν το επίκεντρο είναι σε κάποιο πεδίο.

• **Sized**
Στέλνεται στο Επίπεδο Συγγραφέα (Author Level) σε ένα αντικείμενο όταν οι ιδιότητες bounds, size, ή vertices του αντικειμένου αλλάζουν. Αν κάποιο αντικείμενο σε μία πολλαπλή επιλογή αλλάζει μέγεθος, μόνο το συγκεκριμένο αντικείμενο δεχεται το μήνυμα Sized, και όχι τα υπόλοιπα επιλεγμένα αντικείμενα. Αν ένα group αλλάζει μέγεθος , μόνο το αντικείμενο group δέχεται το μήνυμα Sized και όχι το κάθε ένα μέλος του group χωριστά.

• **TextScrolled**
Στέλνεται σε ένα πεδίο (field) όταν ο χρήστης κάνει κλικ στην γραμμή κύλισης (scroll bar) του πεδίου. Για να σταλεί το μήνυμα TextScrolled στο πεδίο πρέπει η ιδιότητά του BorderStyle να έχει τεθεί στην τιμή «scrolling».

4.2 ΑΝΑΦΟΡΕΣ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΙΣΑΓΩΓΗΣ ΣΤΗΝ ΕΦΑΡΜΟΓΗ

4.2.1 Υλοποίηση του κινούμενου βιβλίου που αλλάζει σελίδες

Κατά την εισαγωγή στην εφαρμογή της Ψηφιακής Σχεδίασης , όπως είδαμε στο Κεφάλαιο 3, εμφανίζεται ένα κινούμενο βιβλίο το οποίο ανοίγει και αποκαλύπτει τις εσωτερικές σελίδες του. Με την είσοδο, λοιπόν, στην πρώτη σελίδα του CONTENTS.TBK εκτελείται ο handler enterpage.

```
to handle enterPage
  system clicktimes
  clicktimes = 1
  send animate1
  show group "l1"
  show group "r1"
  show button "nextbtn"
  show field "nextlbl"
  forward
end
```

Καλείται η διαδικασία «animate1» η οποία εκτελεί την είσοδο του βιβλίου στην οθόνη, και εμφανίζονται τα groups «l1» και «r1» που περιέχουν τα fields με τα κείμενα της πρώτης αριστερής και δεξιάς σελίδας του βιβλίου, και το button "nextbtn" το οποίο αν πατηθεί εκτελεί το επόμενο βήμα (για τη συνέχεια). Η system μεταβλητή clicktimes περιέχει τον αριθμό των φορών που γίνεται κλικ στο button "nextbtn".

Η διαδικασία «animate1» μετακινεί το group «book» (κλειστό βιβλίο) κατά μήκος της οθόνης, και κατόπιν το ανοίγει εμφανίζοντας και κρύβοντας διαδοχικά τα σχετικά αντικείμενα (εικόνες βιβλίου που προβαλλόμενες διαδοχικά η μία μετά την άλλη δημιουργούν την ψευδαίσθηση της κίνησης). Τέλος, αυξάνει το μέγεθός του.

```
-- Είσοδος του βιβλίου στην οθόνη
to handle Animate1
-- Μετακίνησε το βιβλίο κατά μήκος της οθόνης
  send moveObj 5580,1995, "group book"
  step i from 2 to 3
        hide group ("cover" & i-1)
        show group ("cover" & i)
  end
-- Άνοιξε το βιβλίο
  hide group "cover3"
  hide group "book"
  show group "book1"
  step i from 6 to 8
        hide group ("cover" & i-1)
```

```
            if i = 7
                    hide irregularPolygon "binding1"
            end
            show group ("cover" & i)
    end
    hide group "cover8"
    step i from 2 to 4
            hide group ("book" & i-1)
            show group ("book" & i)
    end
    -- Αύξησε το μέγεθος του βιβλίου
    send moveObj 1600,1696,"group book4",6400,4109
  end
```

Η μετακίνηση ενός αντικειμένου και η αλλαγή του μεγέθους του γίνεται με την κλήση της διαδικασίας moveObj.

```
  -- Μετακινεί ένα αντικείμενο (dragObj) από την τρέχουσα θέση του, στη θέση
  -- x2,y2 και από το τρέχον μέγεθός του στο sX,sY
  to handle moveObj  x2, y2, dragObj, sX, sY
    sizeX = item 1 of size of dragObj
    sizeY = item 2 of size of dragObj
     ---Έλεγχος αν πρέπει να αλλάξει το μέγεθος του αντικειμένου (sX και sY
     ---έχουν τιμή κατά την κλήση της moveObj)
    if sX <> NULL and sY <> NULL
            sX = sX - sizeX
            sY = sY - sizeY
    else
            sX = 0
            sY = 0
    end
    position of dragObj = x2,y2
    size of dragObj = sX + sizeX, sY + sizeY
  end
```

Κατόπιν, ο χρήστης κάνει κλικ στο button "nextbtn" για να γυρίσει η σελίδα του βιβλίου. Αν η μεταβλητή clicktimes δεν είναι 3 ή 4 τότε καλείται η διαδικασία turnpage με παράμετρο την clicktimes+1 για το γύρισμα της σελίδας. Αν η clicktimes έχει τιμή 3 (έχουν δηλ. γυρίσει 2 φύλλα του βιβλίου) τότε καλείται η διαδικασία «animate2» για τη μετακίνηση και την «απενεργοποίηση» του βιβλίου (τοποθέτηση στο «παρασκήνιο»). Αν η clicktimes έχει την τιμή 4 τότε γίνεται μετάβαση στην επόμενη σελίδα επιλογής τμήματος της εφαρμογής.

```
  to handle buttonclick
    system clicktimes
    if clicktimes <> 3 and clicktimes <> 4 then
      send turnpage clicktimes + 1
    else
```

```
      if clicktimes = 3 then
        hide group "l3"
        hide group "r3"
        send animate2
      end if
      if clicktimes = 4 then
        hide group "grey book" of this background
        hide self
        hide field "nextlbl"
        transition "blinds" to next page
      end if
    end if
    clicktimes = clicktimes + 1
  end
```

Η διαδικασία turnpage εκτελεί το γύρισμα της σελίδας εμφανίζοντας και κρύβοντας διαδοχικά εικόνες σελίδων που δημιουργούν την ψευδαίσθηση της κίνησης. Επίσης, ανάλογα με την τιμή της system μεταβλητής clicktimes εμφανίζονται και κρύβονται αντίστοιχα τα groups που περιέχουν τα αντίστοιχα fields με τα κείμενα κάθε σελίδας («l2,«r2», «l3», «r3»).

```
  -- Γύρισμα της σελίδας του βιβλίου
  to handle turnPage p
    ml = "l" & (p-1)
    mr = "r" & (p-1)
    ll = "l" & p
    rr = "r" & p
   show group "page1"
   step i from 2 to 4
          hide group (page&i-1)
          show group (page&i)
          if i = 2 then
            hide group mr
          end if
          if i = 3 then
            show group rr
          end if
   end
   hide group "page4"
   show irregularPolygon "page5"
   hide group ml
   step i from 6 to 7
          hide irregularPolygon ("page" & i-1)
          show irregularPolygon ("page" & i)
   end
   show group ll
   hide irregularPolygon "page7"
  end
```

4.2.2 Υλοποίηση του πίνακα περιεχομένων

Ο πίνακας περιεχομένων βρίσκεται στην τρίτη σελίδα του βιβλίου CONTENTS.TBK. Στο αριστερό τμήμα της οθόνης εμφανίζονται τα 7 πεδία με τους τίτλους των κεφαλαίων, τα οποία είναι ομαδοποιημένα σε group. Για την επιλογή κεφαλαίου ο χρήστης κάνει κλικ στο αντίστοιχο πεδίο (συνεπώς, κάνει κλικ και στο group). Εκτελείται η εξής ρουτίνα:

```
to handle buttondown
  system kef
-- Επανάφερε το στυλ των τίτλων όλων των κεφαλαίων, στο αρχικό
  send raisekef
-- Άλλαξε το στυλ του τίτλου του κεφαλαίου που επιλέχθηκε και εμφάνισε τις
-- αντίστοιχες ενότητες
  send openunit target
  kef = last char of name of target
  su = "sub" & kef
-- Επανάφερε το στυλ των τίτλων των ενοτήτων στο αρχικό και απόκρυψε τις
-- υποενότητες
  send raiseenot group su
end
```

Οι διαδικασίες που καλούνται βρίσκονται στο script της σελίδας. Με την openunit εμφανίζεται το group su που περιέχει τα fields με τους τίτλους των ενοτήτων του κεφαλαίου που έχει επιλεγεί, αλλάζει το στυλ του field του επιλεγμένου κεφαλαίου σε inset από raised και εμφανίζεται το group ss που βρίσκεται ανάμεσα στο field του κεφαλαίου που επιλέχθηκε και στο φάτνωμα των ενοτήτων για να δίνει την εντύπωση της συνέχειας του στυλ inset για το επιλεγμένο field και το φάτνωμα.

```
-- Εμφανίζονται οι ενότητες και αλλάζει το στυλ εμφάνισης
-- του τίτλου του κεφαλαίου που επιλέχθηκε
to handle openunit unit
  borderstyle of unit = inset
  rgbfill of unit = 192,192,192
  unitnum = char 5 of name of unit
  ss = "s" & unitnum
  su = "sub" & unitnum
  visible of group ss = true
  visible of group su = true
end
-------------------------------------------------------------------------------------------
--
-- Οι τίτλοι όλων των ενοτήτων και το υπόβαθρό τους επανέρχονται στην
-- αρχική τους κατάσταση ενώ εξαφανίζονται οι υποενότητες
to handle raiseenot subg
  step i from 1 to 7
    fss = "ss" & i
```

```
    ff = "ypo" & i
    visible of field ff = false
    visible of group fss = false
  end
  visible of group "ss8" = false
  step i from 1 to itemcount(subg)
    borderstyle of item i of subg = raised
    rgbfill of item i of subg = 192,192,192
  end
end
```

Η επιλογή ενότητας γίνεται κάνοντας κλικ στο αντίστοιχο field του group της ενότητας. Για την επιλογή ενότητας του 2ου Κεφαλαίου, για παράδειγμα, αρχικά καλείται η raiseenot με παράμετρο το group των ενοτήτων ώστε τα fields των ενοτήτων να επανέλθουν στην αρχική τους κατάσταση, ενώ στη συνέχεια καλείται η διαδικασία visypo με παραμέτρους τη διαφορά του αριθμού των υποενοτήτων κάθε ενότητας από το 7, και το όνομα του field της ενότητας που επιλέχθηκε (η μεταβλητή target αναφέρεται πάντα στο αντικείμενο που δέχεται το κλικ του ποντικιού). Τέλος, αναγράφεται στα fields των υποενοτήτων ο τίτλος καθεμιάς.

```
to handle buttonclick
  system chap
  chap = last char of name of target
  send raiseenot self
  if last char of name of target = "1" then
    send visypo 6,target
    text of field "ypo1" = "Γενικά"
  end
  if last char of name of target = "2" then
    send visypo 5,target
    text of field "ypo1" = "Γενικά"
    text of field "ypo2" = "Η δίτιμη άλγεβρα Boole"
  end
  if last char of name of target = "3" then
    send visypo 4,target
    text of field "ypo1" = "Δυϊσμός"
    text of field "ypo2" = "Βασικά θεωρήματα"
    text of field "ypo3" = "Προτεραιότητα τελεστών"
  end
  if last char of name of target = "4" then
    send visypo 4,target
    text of field "ypo1" = "Γενικά"
    text of field "ypo2" = "Αλγεβρικοί μετασχηματισμοί"
    text of field "ypo3" = "Συμπλήρωμα μιας συνάρτησης"
  end
  if last char of name of target = "5" then
    send visypo 2,target
    text of field "ypo1" = "Ελαχιστόροι και μεγιστόροι"
```

```
    text of field "ypo2" = "Άθροισμα ελαχιστόρων"
    text of field "ypo3" = "Γινόμενο μεγιστόρων"
    text of field "ypo4" = "Μετατροπές κανον. μορφών"
    text of field "ypo5" = "Πρότυπες μορφές"
  end
  if last char of name of target = "6" then
    send visypo 6,target
    text of field "ypo1" = "Γενικά"
  end
  if last char of name of target = "7" then
    send visypo 5,target
    text of field "ypo1" = "Γενικά"
    text of field "ypo2" = "Επέκταση πολλαπλών εισόδων"
  end
end
```

Η διαδικασία visypo αρχικά αλλάζει το στυλ εμφάνισης της επιλεγείσας ενότητας, στη συνέχεια εμφανίζει το group fss που βρίσκεται ανάμεσα στο field της ενότητας που επιλέχθηκε και στο φάτνωμα των υποενοτήτων για να δίνει την εντύπωση της συνέχειας του στυλ inset για το επιλεγμένο field και το φάτνωμα και τέλος εμφανίζει τόσα fields υποενοτήτων όσες οι υποενότητες της επιλεγείσας ενότητας.

```
-- Εμφανίζονται οι υποενότητες της συγκεκριμένης ενότητας
-- και αλλάζει το στυλ εμφάνισης του τίτλου της ενότητας που επιλέχθηκε
to handle visypo vis, enot
  borderstyle of enot = inset
  rgbfill of enot = 224,224,224
  fss = "ss" & last char of name of enot
  visible of group fss = true
  step i from 1 to 7-vis
    ff = "ypo" & i
    visible of field ff = true
  end
end
```

Η μετάβαση στη επιλεγείσα υποενότητα γίνεται κάνοντας κλικ στο αντίστοιχο field, οπότε ανάλογα με τις τιμές των system μεταβλητών kef, chap, subchap που περιέχουν τους αριθμούς του επιλεγμένου κεφαλαίου, ενότητας και υποενότητας, γίνεται η μετάβαση στη σελίδα chappage που είναι η πρώτη σελίδα της επιλεγείσας υποενότητας. Ο χρήστης κάνει κλικ στο group που περιέχει τα fields των υποενοτήτων και το κλικ το λαμβάνει το αντίστοιχο field (τιμή της μεταβλητής target).

```
to handle buttonclick
  system kef, chap, subchap
  subchap = last char of name of target
-- Kef - Ο αριθμός κεφαλαίου
-- Chap - Ο αριθμός ενότητας
```

```
-- Subchap - Ο αριθμός υποενότητας
-- Chappage - Ο αριθμός της πρώτης σελίδας κάθε υποενότητας
 rgbfill of target = 224,224,224
 if kef = 1 then
   runbook = "unit1.tbk"
   if chap = 1 then
     chappage = 1
   end if
   if chap = 2 then
     chappage = 5
   end if
   if chap = 3 then
     chappage = 8
   end if
   if chap = 4 then
     if subchap = 1 then
       chappage = 11
     end if
```

.

.

.

4.3 ΑΝΑΦΟΡΕΣ ΣΤΟΝ ΚΩΔΙΚΑ ΤΟΥ ΤΜΗΜΑΤΟΣ ΔΙΔΑΣΚΑΛΙΑΣ

4.3.1 Υλοποίηση του slider πλοήγησης εντός των υποενοτήτων

Το slider πλοήγησης στις σελίδες μιας υποενότητας είναι ένας πρωτότυπος τρόπος μετάβασης από σελίδα σε σελίδα. Όλα τα αντικείμενα και ο κώδικας με τον οποίο υλοποιείται το χαρακτηριστικό αυτό της εφαρμογής βρίσκονται στο background. Συγκεκριμένα στο background βρίσκεται το group «slider» που αποτελείται από ένα τρίγωνο και δύο γραμμές (αντικείμενα σχεδίασης), το group «ff» που αποτελείται από τρεις οριζόντιες γραμμές και 20 γραμμές που αντιστοιχούν στα tickmarks που αντιπροσωπεύουν τις σελίδες της υποενότητας.

Κατά την είσοδο σε κάθε σελίδα στέλνεται το συμβάν enterpage και στον handler του enterpage για κάθε σελίδα καλείται η ρουτίνα entrance η οποία βρίσκεται στο script του background. Στη ρουτίνα αυτή τοποθετούνται στις system μεταβλητές chap, subchap και sel οι αριθμοί της ενότητας, υποενότητας και ο αριθμός της συγκεκριμένης σελίδας στην υποενότητα, αντίστοιχα. Οι αριθμοί αυτοί περιέχονται στο όνομα κάθε σελίδας.

```
to handle entrance
 system chap, subchap, num, chappage
 local array y[20]
 chap = second char of name of this page
```

```
subchap = third char of name of this page
sel = fourth char of name of this page
```

Κατόπιν, ανάλογα με τους παραπάνω αριθμούς, τοποθετείται στη system μεταβλητή num ο αριθμός των σελίδων της υποενότητας και στην chappage ο αριθμός της πρώτης σελίδας της υποενότητας. (η ρουτίνα uncheckmenus εξαφανίζει τα checkmarks από τις επιλογές των drop-down menus).

```
if chap = 1 then
    send uncheckmenus
        check menuItem sub1 in menu "&Ενότητες 1ου Κεφαλαίου" at
reader
    num = 4
    chappage = 1
  end if
  if chap = 2 then
    send uncheckmenus
        check menuItem sub2 in menu "&Ενότητες 1ου Κεφαλαίου" at
reader
    num = 3
    chappage = 5
  end if
  if chap = 3 then
    send uncheckmenus
        check menuItem sub3 in menu "&Ενότητες 1ου Κεφαλαίου" at
reader
    num = 3
    chappage = 8
  end if
```

.

.

.

Στη συνέχεια, αν το num>1 εμφανίζονται τα αντικείμενα που υλοποιούν το slider (αν η υποενότητα περιέχει μόνο μία σελίδα, τότε το slider δεν εμφανίζεται) και ανάλογα με την τιμή του num υπολογίζεται το μέγεθος και η θέση των groups «ff», «slider» και του field «cover». (Το field «cover» είναι ένα διαφανές field που καλύπτει τα παραπάνω groups και τα tickmarks και δέχεται τα κλικς του ποντικιού). Κατόπιν γίνονται ορατά μόνο τόσα tickmarks όσα και ο αριθμός των σελίδων της υποενότητας και τοποθετούνται στη θέση τους. Τέλος, το group «slider» τοποθετείται στο tickmark αντιστοιχεί στη θέση της σελίδας στην υποενότητα.

```
if num >1 then
  visible of group "ff" of self = true
  visible of group "slider" of self = true
```

```
visible of field "cover" of self = true
item 1 of size of group "ff" of self = (num-1)*355
item 1 of position of group "ff" of self = (item 1 of size of self)/2-(item 1 of size of group "ff" of self)/2
item 1 of size of field "cover" of self = (item 1 of size of group "ff" of self) + 330
item 1 of position of field "cover" of self = (item 1 of size of self)/2-(item 1 of size of field "cover" of self )/2
-- Εμφάνιση των num tickmarks
step i from 1 to num
  visible of line i of self = true
end
-- Απόκρυψη των υπόλοιπων tickmarks
step i from num + 1 to 20
  visible of line i of self = false
end
-- Τοποθέτηση των ορατών tickmarks
step i from 1 to num
  item 1 of position of line i of self = (item 1 of position of group "ff" of self) + (i-1)*355
end
-- Αποθήκευση των αντίστοιχων θέσεων του slider ανάλογα με τον αριθμό των
-- σελίδων της υποενότητας
step i from 1 to num
        y[i] = (item 1 of position of line i of self)-160
end step
-- Τοποθέτηση του slider στο αντίστοιχο tickmark της σελίδας της υποενότητας
item 1 of position of group "slider" of self = y[sel]
else
-- Αν υπάρχει μόνο μία σελίδα στην υποενότητα τότε απόκρυψε το slider και τα
-- σχετικά tickmarks
visible of group "ff" of self = false
visible of group "slider" of self = false
visible of field "cover" of self = false
step i from 1 to 20
  visible of line i of self = false
end
end if
```

Για την υλοποίηση της μετάβασης από σελίδα σε σελίδα της υποενότητας, ο χρήστης κάνει κλικ ή σύρει το slider από tickmark σε tickmark. Τα συμβάντα αυτά του ποντικιού τα δέχεται το field «cover» που είναι διάφανο και καλύπτει τα παραπάνω αντικείμενα, όπως αναφέρθηκε προηγουμένως. Αρχικά, αποθηκεύονται στο array pin τα σημεία που βρίσκονται ακριβώς στο μέσο της απόστασης από tickmark σε tickmark, και στο array x οι θέσεις των tickmarks. Κατόπιν, και καθόλη τη διάρκεια που είναι πατημένο το αριστερό πλήκτρο του ποντικιού, στη μεταβλητή pos αποθηκεύεται η θέση του δείκτη του ποντικιού (sysmouseposition) και αυτή συγκρίνεται με τις τιμές των pin. Μόλις βρεθεί η pos μικρότερη από κάποιο pin, τότε το slider μετακινείται στο αντίστοιχο x και γίνεται μετάβαση στην αντίστοιχη σελίδα της υποενότητας

μόνο αν η σελίδα αυτή είναι διαφορετική από την παρούσα σελίδα. Αν η θέση του δείκτη του ποντικιού είναι μεγαλύτερη από το τελευταίο pin, τότε το slider μετακινείται στο τελευταίο x και γίνεται μετάβαση στην τελευταία σελίδα της υποενότητας (με την προϋπόθεση ότι δεν είμαστε ήδη εκεί).

```
-- Μετακίνηση του slider σε κάθε tick mark καθώς μετακινείται το mouse
-- και εμφάνιση της αντίστοιχης σελίδας της υποενότητας
to handle buttonDown
   system num, chappage
  local array x[num], pin[num-1]
  -- Εύρεση των ενδιάμεσων σημείων ανάμεσα στα tick marks
  step i from 1 to num -1
          pin[i] = (((item 1 of position of line (i+1) of this background)-\
            (item 1 of position of line i of this background))/2 )+\
            (item 1 of position of line i of this background)
  end step
  -- Εύρεση των θέσεων των tick marks
  step i from 1 to num
          x[i] = (item 1 of position of line i of this background)-160
  end step
  -- Μετακίνησε το slider στη θέση καθενός tick mark
  while keystate(keyleftbutton) is down
          pos = item 1 of sysmouseposition
          step i from 1 to num-1
            if pos < pin[i] then
              item 1 of position of group  "slider" of this background = x[i]
              p = i + chappage - 1
              if name of this page <> name of page p then
            go to page p
          end if
              break step
            end
        end
        if pos > pin[num-1] then
          item 1 of position of group "slider" of this background = x[num]
          p = num + chappage - 1
          if name of this page <> name of page p then
            go to page p
          end if
        end
  end while
end buttonDown
```

4.3.2 Απόδοση ηχητικών κομματιών με χρήση MCI συναρτήσεων

Ορισμένες σελίδες της θεωρίας περιέχουν ηχητικά κομμάτια με επιπλέον πληροφορίες για το θέμα που παρουσιάζουν. Για την απόδοση αυτών των ηχητικών κομματιών που έχουν τη μορφή αρχείων WAV χρησιμοποιείται το

στάνταρντ σετ εντολών MCI (Media Control Interface) των Windows. Η αποστολή τέτοιων εντολών γίνεται με τη χρήση της συνάρτησης callMCI().

Για να ακούσει ο χρήστης κάποιο ηχητικό κομμάτι πρέπει να κάνει κλικ στο πλήκτρο που βρίσκεται στη μπάρα πλοήγησης και ελέγχου. Το πλήκτρο αυτό είναι ορατό μόνο στις σελίδες εκείνες οι οποίες περιέχουν τις σχετικές ηχητικές πληροφορίες. Το αν είναι ορατό ή όχι το πλήκτρο αυτό καθορίζεται στον handler enterpage κάθε σελίδας μέσω της εντολής

visible of button "listen" of this background = true (ή false)

Επίσης, στον handler enterpage των σελίδων που περιέχουν σχετικές ηχητικές πληροφορίες, παίρνουν αρχικές τιμές οι system μεταβλητές mmfile στην οποία αποθηκεύεται το όνομα του αρχείου WAV το οποίο προορίζεται να ακουστεί, mmalias στην οποία αποθηκεύεται το «ψευδώνυμο» του αρχείου και startstop της οποίας η αρχική τιμή είναι 1 και υποδηλώνει αν το αρχείο WAV «παίζει» (τιμή 0) ή είναι κλειστό (τιμή 1). Όταν ο χρήστης κάνει κλικ στο button «listen» (ο cursor μετατρέπεται σε μεγαφωνάκι στον handler mouseenter του button), τότε ανάλογα με την τιμή της system μεταβλητής startstop εκτελούνται είτε οι εντολές open και play είτε η close με ορίσματα τις mmfile και mmalias. Επίσης, στην play τοποθετείται και το όρισμα self με το οποίο όταν ολοκληρωθεί η υλοποίηση της εντολής στέλνεται το μήνυμα MCInotify στο button. Τέλος, ο cursor αλλάζει ανάλογα με την τιμή της startstop (αν το κομμάτι παίζεται μετατρέπεται σε διαγραμμένο μεγαφωνάκι που υποδεικνύει ότι αν πατηθεί το πλήκτρο, θα σταματήσει ο ήχος, ενώ αν δεν παίζεται επανέρχεται σε μεγαφωνάκι). Ταυτόχρονα, η τιμή της startstop αλλάζει από 1 σε 0 ή από 0 σε 1.

```
-- Κάθε φορά που γίνεται κλικ στο κουμπί "listen" καλούνται είτε οι MCI
-- συναρτήσεις των windows open και play είτε η close με τα ορίσματα που
-- καθορίζονται από τις σχετικές system μεταβλητές mmfile και mmalias στον
-- handler enterpage της παρούσας σελίδας. Αυτό καθορίζεται από τη system
-- μεταβλητή startstop
to handle buttonclick
  system mmfile, mmalias, startstop
  if startstop = 1 then
    get callMCI("open" && mmfile && "alias" && mmalias)
    get callMCI("play" && mmalias, self)
    syscursor = cursor "stop"
  end if
  if startstop = 0 then
    get callMCI("close" && mmalias, self)
    syscursor = cursor "start"
  end if
  if startstop = 1 then
    startstop = 0
  else
    startstop = 1
  end if
end
```

Κάθε φορά που το ποντίκι εισέρχεται στο πλήκτρο «listen» εκτελείται ο handler mouseenter με τον οποίο αλλάζει ο cursor από «start» (μεγαφωνάκι) σε «stop» (διαγραμμένο μεγαφωνάκι) ανάλογα με την τιμή της startstop, ενώ γίνεται true η system μεταβλητή inbutton που δείχνει αν το ποντίκι είναι εντός ή εκτός του πλήκτρου «listen». Ομοίως, όταν εξέλθει του «listen» το ποντίκι, επαναφέρει το δείκτη του ποντικιού στη default κατάστασή του και κάνει false την τιμή της inbutton.

```
to handle mouseenter
  system startstop, inbutton
  inbutton = true
  visible of button "mmlabel" of this background = true
  if startstop = 1 then
    syscursor = cursor "start"
  else
    syscursor = cursor "stop"
  end if
  caption of statusbar = "Ακούγονται επιπλέον πληροφορίες για το θέμα
που παρουσιάζεται στην παρούσα σελίδα"
end
-----------------------------------------------------------------------------------------------
--
to handle mouseleave
  system inbutton
  inbutton = false
  syscursor = default
  visible of button "mmlabel" of this background = false
  caption of statusbar = "Κεφάλαιο:" && text of recordfield id 60 \
                    && "Ενότητα:" && text of recordfield id 61
end
```

Αφού ολοκληρωθεί το ηχητικό κομμάτι, στέλνεται το μήνυμα MCInotify στο button «listen». Αν το κομμάτι έχει ολοκληρωθεί επιτυχώς (status = successful) τότε κλείνει το αρχείο, αλλάζει η τιμή της startstop σε 1 (αρχείο κλειστό), ενώ ανάλογα με την τιμή της inbutton αλλάζει ο cursor (από διαγραμμένο μεγαφωνάκι σε μεγαφωνάκι). Η αλλαγή αυτή γίνεται μόνο αν ο δείκτης του ποντικιού βρίσκεται εντός του button «listen» (inbutton = true).

```
-- Όταν ολοκληρωθεί το ηχητικό κομμάτι, αλλάζει η τιμή της system μεταβλητής
-- startstop
to handle MCINotify status, operation, device
  system mmalias, startstop, inbutton
  if status && operation = "successful play" and device = mmalias then
     get callMCI("close" && mmalias)
     startstop = 1
     if inbutton then
       syscursor = cursor "start"
     end if
  end if
```

```
end MCINotify
```

Τέλος, σε περίπτωση που ο χρήστης εκτελέσει κάποια λειτουργία και προκαλέσει έξοδο από τη σελίδα (μετάβαση σε άλλη σελίδα), κλείνει το αρχείο WAV σε περίπτωση που αυτό είναι ανοιχτό. Οι ενέργειες αυτές γίνονται από τον handler leavepage που είναι τοποθετημένος στο script της σελίδας.

```
to handle leavepage
  system mmalias, startstop
  visible of button "listen" of background "mainbckgrnd" of book
        "unit1.tbk" = false
  if startstop = 0 then
    get callMCI("close" && mmalias)
  end if
end
```

4.3.3 Υλοποίηση του animation με τον αριθμό που περνάει διαδοχικά από 4 αριθμητικά συστήματα

Στο 1° Κεφάλαιο «Δυαδικές έννοιες» και στην ενότητα «Μετατροπή βάσης αριθμού», στο δεξιό κύριο παράθυρο εμφανίζεται ένας αριθμός που κινείται κυκλικά, περνάει μέσα από τετράγωνα πλαίσια που επιγράφονται ως τα τέσσερα αριθμητικά συστήματα (δυαδικό, δεκαδικό, οκταδικό και δεκαεξαδικό), και αλλάζει ανάλογα με το αριθμητικό σύστημα από το οποίο έχει περάσει.

Αρχικά, και με την εκτέλεση του handler enterpage στη σελίδα, τοποθετείται στην αρχική του θέση (εντός του πλαισίου που επιγράφεται δυαδικό σύστημα) το field «number» και αλλάζει η ιδιότητα του text που γίνεται «100110» (ο αριθμός στο δυαδικό σύστημα).

```
position of field "number" = 5430,1470
text of field "number" = "100110"
```

Η κίνηση του field «number» αποτελείται από 20 βήματα για να ολοκληρωθεί ο κύκλος και να επανέλθει στην αρχική του θέση. Από τα 20 αυτά βήματα τα πέντε γίνονται με κατεύθυνση από τα αριστερά προς τα δεξιά, τα επόμενα από πάνω προς τα κάτω, τα επόμενα πέντε από δεξιά προς τα αριστερά και τα τελευταία πέντε από κάτω προς τα πάνω. Οι κινήσεις αυτές υλοποιούνται από τη ρουτίνα kinisi που βρίσκεται στο script της σελίδας.

```
-- O handler αυτός καλείται από τον notifyafter idle handler του πεδίου
-- number και εκτελεί την κίνησή του
to handle kinisi metr
  if metr <= 5 then
    move field "number" by 470,0
  end
  if metr > 5 and metr <= 10 then
    move field "number" by 0,400
```

```
  end
  if metr > 10 and metr <= 15 then
    move field "number" by -470,0
  end
  if metr > 15 then
   move field "number" by 0,-400
  end
end
```

Η ρουτίνα αυτή καλείται από τον notifyafter idle handler του field «number». Το μήνυμα notifyafter idle στέλνεται όταν δεν εκτελείται καμία δραστηριότητα, σε όλα τα αντικείμενα. Κάθε φορά που λαμβάνει αυτό το μήνυμα το πεδίο «number» αυξάνει την τιμή της system μεταβλητής tik η οποία λαμβάνει διαδοχικές ακέραιες τιμές από το 0 ως το 100 και ξανά 0. Η μεταβλητή tik χρησιμοποιείται για να εισάγει κάποια καθυστέρηση ώστε να είναι ορατή η κίνηση του «number». Κάθε φορά που η tik γίνεται 100 αυξάνει η τιμή της system μεταβλητής count η οποία αλλάζει από 1 σε 20 και ξανά 1 και τοποθετείται ως όρισμα στην κλήση της ρουτίνας kinisi για να εκτελεστεί η σωστή κίνηση (αριστερά, δεξιά, πάνω ή κάτω). Κάθε φορά που η count λαμβάνει συγκεκριμένες τιμές (19, 4, 9, 14) που αντιστοιχούν στη θέση του «number» ακριβώς στο όριο καθενός από τα 4 τετράγωνα πλαίσια (buttons), τότε αλλάζει το χρώμα του πλαισίου στιγμιαία και αλλάζει η ιδιότητα text του field παίρνοντας την τιμή που αντιστοιχεί στο αριθμητικό σύστημα στο οποίο εισέρχεται.

```
-- Κίνηση του number όταν δεν γίνεται τίποτα (idle)
notifyafter idle
  system count, tik
  if tik = 100 then
     tik = 0
    if count = null then
      count = 0
    end
    if count = 20 then
      count = 1
    else
      count = count + 1
    end if
-- Ο handler kinisi που βρίσκεται στο script της παρούσας σελίδας, εκτελεί
-- την κίνηση του number
    send kinisi count
    if count = 19 then
      invert of button "b" = true
      pause 10
      text of field "number" = "100110"
      invert of button "b" = false
    end if
    if count = 4 then
      invert of button "d" = true
```

```
      pause 10
      text of field "number" = "77"
      invert of button "d" = false
    end if
    if count = 9 then
      invert of button "o" = true
      pause 10
      text of field "number" = "115"
      invert of button "o" = false
    end if
    if count = 14 then
      invert of button "h" = true
      pause 10
      text of field "number" = "8D"
      invert of button "h" = false
    end if
  end
  tik = tik + 1
end
```

4.4 ΑΝΑΦΟΡΕΣ ΣΤΟΝ ΚΩΔΙΚΑ ΤΟΥ ΤΜΗΜΑΤΟΣ ΕΞΑΣΚΗΣΗΣ

4.4.1 Υλοποίηση του μαυροπίνακα με την κιμωλία

Κατά την εισαγωγή στο τμήμα των ασκήσεων παρουσιάζεται ένας μαυροπίνακας και ένα χέρι το οποίο κρατά μία κιμωλία και γράφει τη λέξη «ΑΣΚΗΣΕΙΣ», ενώ ταυτόχρονα ακούγεται και ο ήχος της κιμωλίας στον πίνακα. Η υλοποίηση και αυτού του εφφέ βασίζεται στο συμβάν idle το οποίο συμβαίνει συνεχώς για όσο διάστημα δεν εκτελείται από την πλευρά του χρήστη καμία δραστηριότητα.

Πάνω στον πίνακα είναι ήδη σχεδιασμένες 26 γραμμές που έχουν id από 95 έως 120 και οι οποίες σχηματίζουν τη λέξη «ΑΣΚΗΣΕΙΣ» και αρχικά δεν είναι ορατές. Με την είσοδο στη σελίδα ανοίγει το αρχείο BLKBRD.WAV που περιέχει τον ήχο της κιμωλίας, με την κλήση της συνάρτησης callMCI(). Επίσης, η system μεταβλητή t παίρνει την τιμή 95.

```
to handle enterpage
  system t
  t = 95
  get callMCI("open blkbrd.wav alias write")
end
```

Μόλις αρχίσει να εκτελείται το συμβάν idle (η system μεταβλητή flag1 έχει αρχικά τιμή null) καλείται η ρουτίνα moveObjectAlongPath με ορίσματα το button «pencil» που είναι το χέρι με την κιμωλία, τη γραμμή με id t και τον αριθμό 5. Η ρουτίνα αυτή την οποία θα εξηγήσουμε παρακάτω, μετακινεί το

button «pencil» κατά μήκος της γραμμής με id t και παίζει τον ήχο της κιμωλίας στον πίνακα. Αφού εκτελεστεί η κίνηση, με την κλήση της callMCI(«seek write to start») μετακινούμαστε ξανά στην αρχή του BLKBRD.WAV και αυξάνεται η τιμή της t. Η όλη διαδικασία σταματά όταν το t γίνει 121 (εμφανιστεί και η γραμμή με id 120 οπότε το flag1 γίνεται 1).

```
to handle idle
  system t, flag1
  if flag1 = null then
    send moveObjectAlongPath button "pencil", line id t, 5
    t = t + 1
    get callMCI("seek write to start")
    if t = 121 then
      flag1 = 1
    end if
  end if
end
```

Με τη ρουτίνα moveObjectAlongPath και ορίσματα moveObj, pathObj, steps μετακινείται το αντικείμενο moveObj κατά μήκος του pathObj. Steps είναι ο αριθμός των τμημάτων (και άρα των κινήσεων) στις οποίες υποδιαιρείται η γραμμή pathObj. Αρχικά τοποθετούνται στη μεταβλητή vert οι κορυφές της γραμμής pathObj και με την pop οι συντεταγμένες της 1^{ης} κορυφής γίνονται τιμές των startX και startY. Τοποθετείται στην αρχική του θέση το moveObj και στη συνέχεια οι συντεταγμένες της δεύτερης κορυφής της γραμμής pathObj γίνονται τιμές των endX και endY. Η οριζόντια και κατακόρυφη απόσταση των δύο κορυφών διαιρείται σε steps μέρη και εκτελείται μετακίνηση του moveObj κατά xmove και ymove για κάθε άξονα steps φορές μέχρι να φτάσει το moveObj τη δεύτερη κορυφή. Αφού εκτελεστεί η κίνηση, η γραμμή γίνεται ορατή δίνοντας την εντύπωση ότι «σχεδιάστηκε» από το moveObj (το button «pencil»). Τέλος, παίζεται ο ήχος της κιμωλίας που γράφει στον πίνακα με την callMCI(«play»). (Η moveObjectAlongPath μπορεί να πάρει σαν pathObj και αντικείμενο angledLine που έχει περισσότερες από δύο vertices, γι' αυτό και στο τέλος του step loop οι τιμές των endX και endY γίνονται τιμές των startX και startY ώστε να εκτελεστεί ξανά το loop με τις συντεταγμένες της επόμενης κορυφής.)

```
to handle moveObjectAlongPath moveObj,pathObj,steps
  vert = vertices of pathObj
  pop vert into startX
  pop vert into startY
  obSize = size of moveObj
  set position of moveObj to startX-90, startY-125
  step i from 1 to itemcount(vert) div 2
        pop vert into endX
        pop vert into endY
        xmove = (endX - startX) / steps
        ymove = (endY - startY) / steps
```

```
            step j from 1 to steps
                    move moveObj by xmove,ymove
            end
            visible of pathobj = true
        get callMCI("play write wait")
            startX = endX
            startY = endY
    end
end
```

4.4.2 Υλοποίηση του χειρισμού του περιβάλλοντος εξάσκησης

Ο χειρισμός του περιβάλλοντος εξάσκησης γίνεται με τη βοήθεια των πλήκτρων που περιγράφηκαν στο Κεφάλαιο 3 «Η χρήση της Ψηφιακής Σχεδίασης» και τα οποία βρίσκονται στο βασικό background (εκτός από το πλήκτρο «ΛΥΣΗ» που βρίσκεται στη σελίδα).

Είσοδος σε σελίδα που περιέχει άσκηση

Κάθε φορά που ο χρήστης εισέρχεται σε κάποια σελίδα στέλνεται το μήνυμα eisodos του οποίου ο handler βρίσκεται στο background. Γίνεται ορατή και τοποθετείται στη σωστή θέση η ταμπέλα που γράφει «Κάντε κλικ εδώ για να δείτε τη λύση» (field «sclick») και δείχνει προς το πλήκτρο «ΛΥΣΗ». (Ως προς τον κατακόρυφο άξονα ευθυγραμμίζεται με το πλήκτρο «ΛΥΣΗ» - group «slabel» που είναι διαφορετικό για κάθε σελίδα). Τέλος, παίρνουν αρχικές τιμές οι system μεταβλητές orato και tik που χρησιμοποιούνται από το handler idle.

```
to handle eisodos
  system orato, tik
  visible of field "sclick" of self = true
  item 1 of position of field "sclick" of self = 4305
  item 2 of position of field "sclick" of self = item 2 of position of group
"slabel" of this page
  orato = true
  tik = 0
end
```

Ο handler idle του background (που εκτελείται όταν δεν υπάρχει καμία δραστηριότητα από την πλευρά του χρήστη), αν η orato είναι αληθής (με την είσοδο στη σελίδα γίνεται, όπως είδαμε παραπάνω) μετακινεί συνεχώς προς τα αριστερά το field «sclick» μέχρι κάποιο σημείο και ξανά το επαναφέρει στην αρχική του θέση. Επίσης, κάθε 50 tik (tik είναι system μεταβλητή που αυξάνει κατά ένα κάθε φορά που εκτελείται ο handler idle μέχρι να φτάσει κάποια τιμή - 50 στην προκειμένη περίπτωση - και ξαναμηδενίζεται) εμφανίζονται και εξαφανίζονται 5 ερωτηματικά που βρίσκονται στο background. Πρέπει να σημειωθεί ότι η παραπάνω δραστηριότητα εκτελείται μέχρι ο χρήστης να πατήσει το πλήκτρο «ΛΥΣΗ». Μόλις το πατήσει, τότε θέτει false την orato και δεν εκτελείται πλέον καμία εντολή του handler idle του background.

```
to handle idle
  system orato, tik
  if orato = true then
    move field "sclick" of self by -15,0
    if item 1 of position of field "sclick" of self < 1100 then
      item 1 of position of field "sclick" of self = 4305
    end
    if tik = 50 then
      if visible of group "qmarks" of self = true then
        visible of group "qmarks" of self = false
      else
        visible of group "qmarks" of self = true
      end if
      tik = 0
    end if
    tik = tik + 1
  end if
end
```

Το πλήκτρο «ΑΣΚΗΣΗ»

Όπως είδαμε στο Κεφάλαιο 3, αν ο χρήστης κάνει κλικ στο πλήκτρο αυτό μεταβαίνει στην πρώτη άσκηση κάποιας κατηγορίας με τυχαίο τρόπο. Στο button αυτό έχει οριστεί μία UserProperty με το όνομα Problems που είναι ένα μονοδιάστατο array 11 θέσεων στις οποίες έχουν αποθηκευτεί οι αριθμοί των σελίδων των πρώτων ασκήσεων από κάθε κατηγορία. (Στο 1° Κεφάλαιο των ασκήσεων οι κατηγορίες είναι 11 ενώ στο 2° είναι 6). Αφού το problems φορτωθεί σε ένα local array (το f), επιλέγεται τυχαία με τη συνάρτηση random μία σελίδα από το f και γίνεται μετάβαση σ' αυτή σελίδα.

```
to handle buttonclick
  local f array[11]
  f = my problems
  sel = random(11)
  selida = f[sel]
  transition "spiral in fast" to page selida
end
```

Το πλήκτρο >>

Με το πλήκτρο αυτό, όπως είδαμε στο Κεφάλαιο 3 γίνεται μετάβαση στην επόμενη άσκηση της ίδιας κατηγορίας. Όταν εισέλθουμε σε σελίδα που περιέχει την 1η άσκηση κάποιας κατηγορίας, εκτελείται ο handler enterpage της σελίδας. Στη σελίδα αυτή έχει οριστεί μία UserProperty με το όνομα exers που είναι ένα μονοδιάστατο δυναμικό array (με μη ορισμένο αριθμό θέσεων) και περιέχει τους αριθμούς των σελίδων που έχουν ασκήσεις της ίδιας κατηγορίας. Το array MyUserProperty του πλήκτρου >> (button «more») γίνεται ίσο με το exers ενώ αν οι θέσεις του exers είναι λιγότερες από 2 τότε το button «more» απενεργοποιείται. (Αν η άσκηση είναι η μοναδική της κατηγορίας, τότε το exers έχει δύο θέσεις με την πρώτη να περιέχει το 0).

Τέλος, η system μεταβλητή d που λειτουργεί ως δείκτης στο array MyUserProperty παίρνει την τιμή 2.

```
to handle enterpage
  system d
  local s array[]
-- Το array exers περιέχει τους αριθμούς των σελίδων που περιέχουν προβλήματα
-- της ίδιας κατηγορίας με αυτό της τρέχουσας σελίδας
  s = exers of this page
-- Το array exers γίνεται ίσο με το array myUserProperty του button "more"
  set myUserProperty of button "more" of this background to s
  if dimensions(s) < 2 then
    enabled of button "more" of this background = false
  else
    enabled of button "more" of this background = true
  end if
-- Η μεταβλητή d λειτουργεί ως δείκτης στο array που περιέχει τις σελίδες με
-- προβλήματα της ίδιας κατηγορίας
  d = 2
```

Αν ο χρήστης κάνει κλικ στο πλήκτρο >>, τότε γίνεται μετάβαση στη σελίδα μέσα στο array MyUserProperty την οποία δείχνει η system μεταβλητή d (η οποία με την είσοδο στη σελίδα έχει πάρει την τιμή 2 και δείχνει στη δεύτερη άσκηση). Κατόπιν, η d αυξάνει κατά 1 και αν έχει υπερβεί τον αριθμό των θέσεων του array (έχουμε φτάσει στην τελευταία άσκηση της κατηγορίας), απενεργοποιείται το button «more».

```
to handle buttonclick
  system d
  local s array[]
  s = my myUserProperty
  pg = s[d]
  transition "rain right fast" to page pg
  d = d + 1
  if d > dimensions(s) then
    my enabled = false
  end if
end
```

Το πλήκτρο «Θεωρία»

Με το πλήκτρο αυτό ο χρήστης μεταφέρεται στο κομμάτι της θεωρίας που είναι σχετικό με την άσκηση της παρούσας σελίδας. Ο αριθμός της σελίδας του βιβλίου της θεωρίας είναι αποθηκευμένος στη UserProperty theory κάθε σελίδας που περιέχει άσκηση. Έτσι, κάνοντας κλικ ο χρήστης στο πλήκτρο αυτό μεταβαίνει στην αντίστοιχη με την άσκηση θεωρία. Η system μεταβλητή flag4 παίρνει την τιμή 1 έτσι ώστε μετά την ολοκλήρωση του handler buttonclick να εκτελεστεί ο handler mouseleave χωρίς να σβηστεί το caption του statusbar (το οποίο πλέον παρουσιάζει πληροφορίες για το νέο βιβλίο). Αυτό γίνεται μόνο όταν ο δείκτης του ποντικιού εισέλθει στο button και

εξέλθει χωρίς να γίνει κλικ. Η flag4 παίρνει την τιμή 0 στο handler enterbackground που εκτελείται με την είσοδο στο βασικό background.

```
to handle buttonclick
  system flag4
  flag4 = 1
  pageno = theory of this page
  syslockscreen = true
  transition "zoom out" to page pageno of book "unit1.tbk"
end
-----------------------------------------------------------------------------------------------------
--
to handle mouseenter
  syscursor = 44
   caption of statusbar = "Παρουσιάζει το τμήμα της θεωρίας που είναι
σχετικό με το παρόν πρόβλημα"
end
-----------------------------------------------------------------------------------------------------
--
to handle mouseleave
  system flag4

  syscursor = default
  if flag4 = 0 then
    caption of statusbar = " "
  end if
end
```

4.4.3 Υλοποίηση της άσκησης με τα συμπληρώματα ως προς 10

Στις ασκήσεις του Κεφαλαίου 1 «Δυαδικές Έννοιες» υπάρχει μία άσκηση για την εύρεση των συμπληρωμάτων ως προς 10 των αριθμών 123900, 090657, 100000 και 000000. Αφού κάνει κλικ ο χρήστης στο πλήκτρο «ΛΥΣΗ» για να δει τη λύση, παρουσιάζονται οι κανόνες βάσει των οποίων υπολογίζονται τα συμπληρώματα ως προς 10, και παρακάτω, στο αριστερό μέρος οι αριθμοί των οποίων τα συμπληρώματα είναι προς υπολογισμό, και στο δεξιό τα συμπληρώματά τους. Ο χρήστης προκειμένου να δει τον τρόπο υπολογισμού των συμπληρωμάτων, προτρέπεται να περάσει το ποντίκι πάνω από τα ψηφία των 4 αριθμών. Με το πέρασμα του ποντικιού πάνω από κάποιο ψηφίο, το αντίστοιχο ψηφίο του συμπληρώματός του γίνεται κόκκινο, ενώ επίσης, γίνεται κόκκινος και ο κανόνας βάσει του οποίου υπολογίστηκε το ψηφίο του συμπληρώματος.

Η υλοποίηση του παραπάνω εφφέ γίνεται ως εξής:

Καθένα από τα ψηφία των 3 πρώτων αριθμών και ο 4^{ος} (000000) είναι hotwords των οποίων το hotwordstyle είναι frame (το οποίο είναι ορισμένο και ως bookdefault). Όταν ο δείκτης του ποντικιού εισέλθει σε μία από αυτές τις hotwords, τότε γίνεται ορατό ένα field που το κείμενό του είναι ακριβώς το ίδιο με τον κανόνα υπολογισμού του συμπληρώματος, όπως και η θέση του, ενώ

το χρώμα του είναι κόκκινο. Το field αυτό ήταν πριν αόρατο. Ομοίως, το ίδιο συμβαίνει και με το αντίστοιχο ψηφίο του συμπληρώματος, το οποίο και αυτό είναι ένα προηγουμένως αόρατο field με την ίδια θέση και κείμενο με το ψηφίο του συμπληρώματος και χρώμα κόκκινο. Έτσι δημιουργείται η εντύπωση ότι ο κανόνας και το ψηφίο του συμπληρώματος «φωτίζονται». Επίσης αλλάζει το hotwordstyle της hotword εντός της οποίας έχει εισέλθει ο δείκτης του ποντικιού και γίνεται color όπου το default color του βιβλίου είναι το κόκκινο. Έτσι, και το αρχικό ψηφίο «φωτίζεται» και μάλιστα αφαιρείται και το πλαίσιο με το οποίο περιβαλλόταν (αρχικά το hotwordstyle ήταν frame). Όταν ο δείκτης του ποντικιού αποχωρήσει, τα «κόκκινα» πεδία γίνονται και πάλι αόρατα και το hotwordstyle της hotword επανέρχεται στο αρχικό του.

```
to handle mouseenter
  show field id 59
  show field id 35
  hotwordstyle of self = color
end
```

```
to handle mouseleave
  hide field id 59
  hide field id 35
  hotwordstyle of self = bookdefault
end
```

4.5 ΑΝΑΦΟΡΕΣ ΣΤΟΝ ΚΩΔΙΚΑ ΤΟΥ ΤΜΗΜΑΤΟΣ ΑΞΙΟΛΟΓΗΣΗΣ

4.5.1 Συνεχής μουσική σε όλα τα Βιβλία:

```
to handle idle
    if idmid of this book = null                -- αν δεν έχει κομμάτι να παίξει
        idmid of this book = 1                  -- παίζει το πρώτο κομμάτι
        fileMid of this book = "c:\tfs\mid\tt01.mid"
    end
    MMFile = FileMid of this book
    -- με τη συνάρτηση CallMCI γίνεται διαχείρηση διάφορων Πολυμέσων
    get CallMCI("status " & MMFile & " mode")
    if It = "playing"                           -- αν παίζει ήδη μουσική
    else                                        -- αλλιώς..
        MMFile = FileMid of this book
        get callMci("close " & MMFile)          -- κλείνει το προηγούμενο
                                                -- αρχείο
        x = IdMid of this book
        -- η IdMid δείχνει κάθε φορά το επόμενο μουσικό κομμάτι
        IdMid of this book = IdMid of this book +1
        if IdMid of this book >= 35             -- αν τελειώσουν τα μουσικά
                                                -- αρχεία
```

```
                IdMid of this book = 1          -- αρχίζει ξανά από το πρώτο
            end
            if x div 10 = 0          -- εξετάζει αν ο αριθμός είναι διψήφιος
                x = "0" & x          -- αν όχι του προσθέτει μπροστά ένα μηδέν
            end
            fileMid of this book = "c:\tfs\mid\tt" & x & ".mid"
            MMFile = FileMid of this book
            get CallMCI("open " & MMFile)    -- ανοίγει το νέο αρχείο
                                             -- μουσικής
            get CallMCI("play " & MMFile)    -- και το παίζει
        end
end
```

Στον παραπάνω κώδικα οι Μεταβλητές IdMid και FileMid είναι User Properties. Η λειτουργία τους είναι να διατηρούν την μουσική συνεχώς ανοιχτή. Πιο συγκεκριμένα, κάθε στιγμή η IdMid κρατά τον αριθμό του μουσικού κομματιού που θα ακολουθήσει ενώ η FileMid το όνομα του αρχείου που παίζει εκείνη την στιγμή. Αυτές οι δύο User Properties βρίσκονται σε κάθε Βιβλίο της Εφαρμογής. Κάθε φορά που γίνεται μετάβαση από κάποιο περιβάλλον σε άλλο (π.χ. από το Κυρίως Μενού στο Περιβάλλον Επίλυσης Τεστ) αυτές οι δύο μεταβλητές ενημερώνονται άμεσα κατά την μετάβαση. Δηλαδή αν ο χρήστης βρίσκεται στο Περιβάλλον Βοήθειας και οι τιμές της IdMid = 3 και της FileMid = «c:\tfs\mid\tt02.mid» τότε και στο Κυρίως Μενού οι ομώνυμες User Properties έχουν τις ίδιες τιμές. Ο παραπάνω κώδικας υπάρχει απαράλλαχτος στα 4 περιβάλλοντα του τμήματος αξιολόγησης αλλά και στο Κυρίως Μενού.

4.5.2 Μέγιστος αριθμός χαρακτήρων σε ένα πεδίο

```
-- Με το που πατηθεί κάποιο πλήκτρο
to handle keyChar keyPressed
        local INT maxCharAllowed                -- ο μέγιστος αριθμός
                                                -- χαρακτήρων
        local INT currentCount                  -- ο τρέχων αριθμός
                                                -- χαρακτήρων
        maxCharAllowed = 1
        currentCount = charCount(my text)       -- οι χαρακτήρες του πεδίου
        -- αν οι χαρακτήρες ξεπεράσουν σε αριθμό τον μέγιστο επιτρεπόμενο
        if (currentCount >= maxCharAllowed) or (keyPressed = keyEnter and
currentCount >= maxCharAllowed - 1)             -- ή αν πατηθεί το enter
            beep 1                                      -- ακούγεται ένα beep
        else
            -- απαγορεύει επιπλέον τα πλήκτρα διάφορα του 2 και του 3
            if keyPressed>51 or keyPressed< 50
            else
                forward                         -- προωθεί το μήνυμα προς τα
                                                -- πάνω
```

```
end
```

Ο παραπάνω κώδικας απαγορεύει την εισαγωγή σε ένα πεδίο περισσότερων χαρακτήρων από κάποιο δεδομένο αριθμό (maxCharAllowed). Επιπλέον περιορίζει το είδος των χαρακτήρων που μπορούν να εισαχθούν στο πεδίο. Προωθώντας (forward) το μήνυμα προς τα πάνω στην ιεραρχία επιτρέπει την εκτύπωση του τρέχοντος χαρακτήρα στο πεδίο.

4.5.3 Αυτόματη Γραμμή Κύλισης σε Πεδίο

```
-- πριν μπει ο χρήστης στην συγκεκριμένη σελίδα στέλνεται μήνυμα
-- ενημέρωσης στο αντικείμενο που έχει τον κώδικα (στο πεδίο)
notifyBefore enterPage
        send showOrHideScrollbar                -- καλεί την υπορουτίνα που
                                                -- εμφανίζει ή κρύβει
                                                -- τις scroll bars του πεδίου
end
-----------------------------------------------------------------------------------------------
-----
-- αφού φύγει από το πεδίο στέλνεται μήνυμα ενημέρωσης στον εαυτό του
notifyAfter leaveField
        send showOrHideScrollbar
end
-----------------------------------------------------------------------------------------------
-----
-- μετά το πάτημα κάποιου πλήκτρου στέλνεται μήνυμα ενημέρωσης
notifyAfter keyUp
        send showOrHideScrollbar
end
-----------------------------------------------------------------------------------------------
-----
-- αφού γίνει «επικόληση» κάτι από τον ClipBoard στέλνεται μήνυμα
--  ενημέρωσης
notifyAfter paste
        send showOrHideScrollbar
end
-----------------------------------------------------------------------------------------------
-----
-- αφού κοπεί κάτι και σταλεί στον Clipboard στέλνεται μήνυμα ενημέρωσης
notifyAfter cut
        send showOrHideScrollbar
end
```

```
end
-----------------------------------------------------------------------------------------------
-----
-- αφού σβυστεί κάτι στο πεδίο στέλνεται μήνυμα ενημέρωσης
notifyAfter clear
        send showOrHideScrollbar
end
-----------------------------------------------------------------------------------------------
-----
-- η παρακάτω υπορουτίνα εμφανίζει ή εξαφανίζει τις scroll bars στο πεδίο
to handle showOrHideScrollbar
        if my textOverflow > 0 or my scroll > 0
                set my borderStyle to scrolling     -- εμφανίζει τις scroll bars
        else
                set my borderStyle to rectangle     -- εξαφανίζει τις scroll bars
        end
end
```

Τα μηνύματα μπορούν να σταλούν άμεσα σε κάποια αντικείμενα και στην συνέχεια να προωθηθούν ή όχι προς τα πάνω στην ιεραρχία. Υπάρχουν όμως κάποια μηνύματα που δεν στέλνονται ποτέ σε κάποια αντικείμενα (π.χ. το μήνυμα Idle στέλνεται μόνο στην σελίδα). Γι' αυτό τον λόγο χρησιμοποιούνται τα Μηνύματα ενημέρωσης. Τα μηνύματα αυτά δεν είναι τίποτα άλλο παρά αντίγραφα των κανονικών μηνυμάτων τα οποία στέλνονται σε όσα αντικείμενα δεν έχουν άμεση προσπέλαση σ' αυτά. Με άλλα λόγια όταν κάποιο αντικείμενο θέλει να εκμεταλευτεί το μήνυμα Idle δεν έχει παρά να χρησιμοποιήσει κάποιο αντίγραφο του μηνύματος (Μήνυμα Ενημέρωσης) αντί το ίδιο το μήνυμα. Αυτό επιτυγχάνεται με την βοήθεια των Χειριστών Ενημέρωσης (Notify Handlers) οι οποίοι είναι οι Notifybefore και NotifyAfter. Με τη χρήση τον Χειριστών Ενημέρωσης τα αντικείμενα αποκτούν ξεχωριστή οντότητα και παύουν να εξαρτώνται από το Βιβλίο στο οποίο ανήκουν.
Χρησιμοποιώντας τα μηνύματα ενημέρωσης (Notify messages) ο προηγούμενος κώδικας εμφανίζει και εξαφανίζει τις γραμμές κύλισης σε ένα πεδίο όταν χρειάζεται. Ελέγχει δηλαδή τις ιδιότητες TextOverflow και Scroll και ανάλογα (αν είναι θετικά νούμερα) βάζει ή (αν είναι αρνητικά νούμερα ή μηδέν) βγάζει τις γραμμές κύλισης.

4.5.4 Συνάρτηση εύρεσης των ημερών ενός μήνα

```
-- Συνάρτηση: daysInMonth(<μήνας>,<χρόνος 4 ψηφίων>)
-- Επιστρέφει τον αριθμό των ημερών σε έναν μήνα.
-- Σημείωση: αυτή η συνάρτηση απαιτεί ο χρόνος ναέχει 4 ψηφία.
-- Αυτή η συνάρτηση καλεί την συνάρτηση leapYear για έλεγχο δίσεκτου
έτους.
-- Παράδειγμα: για να βρείτε πόσες μέρες έχει ο Φεβρουάριος του 1992,
-- θέστε x = daysInMonth(2,1992)
to get daysInMonth m,y
        if m <> 2       -- αν ο μήνας δεν είναι ο Φεβρουάριος
```

```
            -- επιστρέφει κάποιο στοιχείο της λίστας
            return item m of "31,28,31,30,31,30,31,31,30,31,30,31"
        else        -- αλλιώς
            -- καλεί την συνάρτηση leapYear(y) για να ελέγξει αν το έτος
            -- y είναι δίσεκτο
            if leapYear(y)
                return 29       -- αν είναι δίσεκτο επιστρέφει 29 μέρες
            else
                return 28       -- αλλιώς επιστρέφει 28 μέρες
            end
        end
end
```

4.5.5 Έλεγχος Δίσεκτου Έτους

```
-- Επιστρέφει TRUE αν το όρισμα y είναι δίσεκτο έτος, αλλιώς επιστρέφει
FALSE .
-- Σημείωση: Αυτή η συνάρτηση απαιτεί τον χρόνο με 4 ψηφία.
-- Παράδειγμα: Για να βρείς αν το 1992 είναι δίσεκτο γράψε
--  If leapYear(1992)
to get isLeapYear y
        -- αν ο χρόνος είναι πολλαπλάσιο του 4 και του 100
        -- ή αν ο χρόνος είναι πολλαπλάσιο του 400
        if ((y mod 4 = 0 and y mod 100 <> 0) or (y mod 400 = 0))
            return TRUE         -- είναι δίσεκτος
        else
            return FALSE        -- δεν είναι δίσεκτος
        end
end
```

4.5.6 Σύγκριση δύο ημερομηνιών

Η παρακάτω συνάρτηση συγκρίνει δύο ημερομηνίες. Αν τις βρεί ίσες επιστρέφει 0 αλλιώς αν η πρώτη είναι μεγαλύτερη επιστρέφει 1, και αν η δεύτερη είναι μεγαλύτερη επιστρέφει -1.

Η μορφή των ημερομηνιών είναι «d,m,y» δηλαδή πρόκειται για τρείς αριθμούς χωρισμένους με κόμμα «,».

```
to get comparedate d1,d2
        Conditions
            -- όταν η δεύτερη χρονιά είναι μεγαλύτερη...
            When item 3 of d2 > item 3 of d1
            return -1                                   -- επιστρέφει -1
            -- αν οι χρονιές είναι ίσες...
            When item 3 of d2 = item 3 of d1
            Conditions
                -- αν ο δεύτερος μήνας είναι μεγαλύτερος...
                When item 2 of d2 > item 2 of d1
```

```
                return -1                                   -- επιστρέφει -1
                -- αλλιώς αν οι μήνες είναι ίσοι...
                When item 2 of d2 = item 2 of d1
                Conditions
                        -- αν η δεύτερη μέρα είναι μεγαλύτερη...
                        When item 1 of d2 > item 1 of d1
                        return -1                           -- επιστρέφει -1
                        -- αν τέλος και οι μέρες είναι ίσες...
                        When item 1 of d2 = item 1 of d1
                        return 0                            -- επιστρέφει 0
                        -- αν η πρώτη μέρα είναι μεγαλύτερη...
                        When item 1 of d2 < item 1 of d1
                        return 1                            -- επιστρέφει 1
                end
                -- όταν ο δεύτερος μήνας είναι μικρότερος...
                When item 2 of d2 < item 2 of d1
                return 1                                    -- επιστρέφει 1
        end
        -- αν ο πρώτος χρόνος είναι μεγαλύτερος...
        When item 3 of d2 < item 3 of d1
        return 1                                            -- επιστρέφει 1
    end
    -- αν δεν ισχύει τίποτα από τα προηγούμενα...
    return 0                                                -- επιστρέφει 0
end
```

4.5.7 Υλοποίηση του Παραθύρου Ρολόϊ - Χρονόμετρο

```
notifybefore idle               -- όσο περνά ο χρόνος
    system oldtime1,ftime,ltime         -- καθολικές μεταβλητές
    get systime             -- βρίσκει την ώρα του συστήματος
    format time it as "h,min,sec"       -- μετατρέπει την ώρα σε μορφή HMS
    set clocktime of self to it         -- καλεί την υπορουτίνα clocktime με
όρισμα
                                        -- την τρέχουσα ώρα
    if oldtime1 <> systime              -- αν έχει αλλάξει η ώρα
        oldtime1 = systime
        systimeformat = "min"
        text of field "left" =  ltime - systime & " Λεπτ"
        -- ενημέρωσε το πεδίο που δείχνει πόσος χρόνος έμεινε
        if ltime-systime =1 then
            text of field "left" = text of field "left" & "ό"
        else
            text of field "left" = text of field "left" & "ά"
        end
        if ltime-systime <=0 then   -- αν έχει λήξει ο χρόνος του Τεστ
```

```
                    send checktest        -- κάλεσε την ρουτίνα checktest
                                          -- για να υπολογιστεί ο βαθμός
                    in viewer id 0
                          ftime=null
                          ltime=null
                          go to page "contents"        -- γύρνα στα
                                                       -- περιεχόμενα
                    end
                    close viewer "clockv"                -- κλείσε το παράθυρο
                                                         -- Ρολόϊ
              end
        end
        sysTimeFormat = "h,min,sec"
end
-----------------------------------------------------------------------------------------------
-----
-- ενημερώνει τους δείκτες του ρολογιού
to set clocktime to hms
        -- ελέγχει για να δεί αν η ώρα έχει αλλάξει
        set oldTime to curTime of self      -- αλλάζει την User property oldTime
        if hms = oldTime
              break
        else
              set curTime of self to hms  -- αν έχει αλλάξει ο χρόνος αλλάζει
την
                                          -- user property curTime
        end
        set coords to bounds of self        -- οι κορυφές του αντικειμένου ρολόι
        -- το μήκος και το ύψος του ρολογιού
        set ctr to (item 3 of coords + item 1 of coords)/2,(item 4 of coords +
item 2 of coords)/2
        -- υπολογίζει την ακτίνα του κύκλου
        set radius to ((item 3 of coords) - (item 1 of coords))/2
        set hour to item 1 of hms mod 12          -- βρίσκει την ώρα
        set minutes to item 2 of hms mod 60       -- βρίσκει τα λεπτά
        set secs to item 3 of hms mod 60          -- βρίσκει τα δευτερόλεπτα
        set sls to syslockscreen
        set syslockscreen to true                 -- παγώνει την οθόνη
        -- μετακινεί τον ωροδείκτη βρίσκοντας την γωνία μετακίνησης
        set hourAngle to (hour/12 + minutes/60/12 + secs/60/60/12)
        set vertices of line "hour" of self to
handVertices(ctr,hourAngle,.6,radius)
        -- μετακινεί τον λεπτοδείκτη βρίσκοντας την γωνία μετακίνησης
        set minutesAngle to (minutes/60 + secs/60/60)
        set vertices of line "minute" of self to
handVertices(ctr,minutesAngle,.9,radius)
        -- μετακινεί τον δευτερολεπτοδείκτη βρίσκοντας την γωνία μετακίνησης
        set secsAngle to (secs/60)
```

```
        set vertices of line "second" of self to
handVertices(ctr,secsAngle,.95,radius)
        set syslockscreen to sls                    -- ξεπαγώνει την οθόνη
end
-----------------------------------------------------------------------------------------------------
-----
-- επιστρέφει τις κορυφές του δείκτη βασισμένο στην γωνία και το μήκος
to get handVertices ctr,angle,len,radius
   return \
     item 1 of ctr + (.1 * radius * cos((angle-.5) * (2 * PI) - (PI / 2))),\
     item 2 of ctr + (.1 * radius * sin((angle-.5) * (2 * PI) - (PI / 2))),\
     item 1 of ctr + (len * radius * cos(angle * (2 * PI) - (PI / 2))), \
     item 2 of ctr + (len * radius * sin(angle * (2 * PI) - (PI / 2)))
end
-----------------------------------------------------------------------------------------------------
---------------
to get clocktime
        return curTime of self
end
-----------------------------------------------------------------------------------------------------
-----
-- κρατά το ρολόϊ κυκλικό αν αλλάξει το μέγεθός του
to handle sized
        set t to target
        if t is self
                set b to bounds of t
                set maxside to max(item 3 of b - item 1 of b,item 4 of b - item 2
of b)
                set item 3 of b to item 1 of b + maxside
                set item 4 of b to item 2 of b + maxside
                set syssuspendmessages to true
                set bounds of t to b
                set syssuspendmessages to false
                send setTicks          -- ξαναζωγραφίζει τα λεπτά
        end
end
-----------------------------------------------------------------------------------------------------
-----
-- ξανασχεδιάζει τα σημεία των λεπτών (καλείται όταν το ρολόϊ αλλάζει
μέγεθος)
-- τα σημεία αυτά είναι γραμμές που έχουν αύξων αριθμό από 0 έως
-- 59 για όνομα

to handle setTicks
        set syslockscreen to true            --παγώνει την οθόνη
        set coords to bounds of self
        set ctr to (item 3 of coords + item 1 of coords)/2,(item 4 of coords +
item 2 of coords)/2
        set radius to ((item 3 of coords) - (item 1 of coords))/2
```

```
        step i from 0 to 59
                set t to line i of self
                conditions
                        when i mod 15 is 0
                                set sz to .15
                        when i mod 5 is 0
                                set sz to .1
                        else
                                set sz to .05
                                if radius < 1440/2
                                        -- believe it or not, this test is needed for
performance
                                        if visible of t
                                                hide t
                                        end
                                else
                                        if not visible of t
                                                show t
                                        end
                                end
                end
                set angle to (i/60) * (2 * PI) - (PI / 2)
                set vertices of t to\
                        item 1 of ctr + ((1-sz) * radius * cos(angle)),\
                        item 2 of ctr + ((1-sz) * radius * sin(angle)),\
                        item 1 of ctr + (radius * cos(angle)), \
                        item 2 of ctr + (radius * sin(angle))
        end
end
```

4.5.8 Ρουτίνες Πλοήγησης στη Βάση Δεδομένων

- **Ανάγνωση της τιμής κάποιου Πεδίου της Βάσης Δεδομένων**

Η παρακάτω συνάρτηση διαβάζει από την Βάση Δεδομένων, με όνομα Βάσης αυτό της παραμέτρου DbTable, την τιμη του πεδίου fieldName και την τιμή την επιστρέφει στην ειδική μεταβλητή It

```
to get fieldValue1 DbTable , fieldName
        -- Βρίσκει τον τύπο του Πεδίου
        DBfieldType = getPXFieldType(DBTable,fieldName)
        clear SysError          -- καθαρίζει την μεταβλητή sysError
        retVal = -1     -- Θέτει την επιστρέφόμενη τιμή στην τιμή λάθους
                        -- αυτή θα επιστρέψει αν δεν βρεθεί το πεδίο
        conditions
        -- όταν το πεδίο είναι τύπου memo "BLOB" (Binary Large OBject) -
        when first Char of DBfieldType = "M"
                -- ανοίγει το BLOB για διάβασμα
```

```
            hPrivateBlob = openPXBlobRead(DBTable, fieldName)
            if hPrivateBlob < 0
                send PXError hPrivateBlob
            else
                retVal = getPXMemoBlob(hPrivateBlob)
                get closePXBlob(hPrivateBlob, 0)
            end
        -- όταν το πεδίο είναι τύπου graphics BLOB
        when first Char of DBfieldType = "G"
            hPrivateBlob = openPXBlobRead(DBTable, fieldName)
            if hPrivateBlob < 0
                send PXError hPrivateBlob
            else
                retVal = getPXGraphicBlob(hPrivateBlob)
                set item 2 of retVal to
getPXGraphicBlobPalette(hPrivateBlob)
                get closePXBlob(hPrivateBlob, 0)
            end
        -- όταν το πεδίο είναι τύπου binary BLOB (the "B" stands for binary)
        when first Char of DBfieldType = "B"
            hPrivateBlob = openPXBlobRead(DBTable, fieldName)
            if hPrivateBlob < 0
                send PXError hPrivateBlob
            else
                retVal = getPXBlob(hPrivateBlob,
getBlobSize(hPrivateBlob), 0)
                get closePXBlob(hPrivateBlob, 0)
            end
        -- όταν το πεδίο είναι τύπου formatted text (μορφοποιημένο κείμενο)
        when first Char of DBfieldType = "F"
            sysError = "This field type not supported: Formatted text"
        -- όταν το πεδίο είναι τύπου OLE BLOB
        when first Char of DBfieldType = "O"
            sysError = "This field type not supported: OLE Blob"
        -- σε όλες τις άλλες περιπτώσεις
        else
            retVal = getPXFieldValue(DBTable,fieldName)
            if sysError is not null
                send PXError retVal
            end
        end
        return retVal
end
```

• Αποθήκευση κάποιας τιμής στο Πεδίο μίας Βάσης Δεδομένων

Η παρακάτω συνάρτηση αποθηκεύει μία τιμή (value) σε κάποιο πεδίο (fieldName). Δεν υπάρχει το όρισμα της Βάσης Δεδομένων γιατί δηλώνεται μέσα στην υπορουτίνα σαν "database"

```
to set fieldValue fieldName to value
	DbTable = "database"                    -- το όνομα της Βάσης
Δεδομένων
	-- Βρίσκει τον τύπο του πεδίου
	DBfieldType = getPXFieldType(DBTable,fieldName)
	clear SysError                          -- καθαρίζει την SysError
	conditions
	-- όταν το πεδίο είναι τύπου memo BLOB
	when first char of DBfieldType = "M"
		-- προστίθεται για την /0 λέξη τερματισμού
		BLOBSize = charCount(value) + 1
		hPrivateBlob = openPXBlobWrite(DBTable, fieldName,
BLOBSize, 0)
		if hPrivateBlob < 0
			send PXError hPrivateBlob
		else
			get setPXMemoBlob(hPrivateBlob, value)
			sysError = getPXErrorString(it)
			get closePXBlob(hPrivateBlob, 1)
		end if
	-- όταν το πεδίο είναι τύπου graphics BLOB από κάποιο αρχείο

	when first char of DBfieldType = "G"
		BLOBSize = getPXFileSize(value) + 8    -- για επικεφαλίδα
		hPrivateBlob = openPXBlobWrite(DBTable, fieldName,
BLOBSize, 0)
		if hPrivateBlob < 0
			send PXError hPrivateBlob
		else
			get setPXGraphicBlobFromFile(hPrivateBlob, value)
			sysError = getPXErrorString(it)
			get closePXBlob(hPrivateBlob, 1)
		end if
	-- όταν το πεδίο είναι τύπου binary BLOBs
	when first char of DBfieldType = "B"
		sysError = "This field type not supported: Binary"
	-- όταν το πεδίο είναι τύπου formatted memo BLOB
	when first char of DBfieldType = "F"
		sysError = "This field type not supported: Formatted text"
	-- όταν το πεδίο είναι τύπου OLE BLOB
	when first char of DBfieldType = "O"
		sysError = "This field type not supported: OLE Blob"
	-- σε όλες τις άλλες περιπτώσεις
```

```
        else
                get setPXfieldvalue(DBTable,fieldName, value)
        sysError = getPXErrorString(it)          -- το τρέχον λάθος
        end conditions
end fieldValue
```

- **Αναζήτηση τιμής σε κάποιο Πεδίο της Βάσης Δεδομένων**

Η παρακάτω συνάρτηση αναζητά σε μία Βάση Δεδομένων να βρεί κάποια τιμή ενός πεδίου. Το όρισμα "mode" καθορίζει πώς θα ψαχτεί ο πίνακας. Αν mode = 0 τότε αυτό σημαίνει ότι η αναζήτηση θα γίνει από την πρώτη εγγραφή (record). Αν είναι 1 αυτό σημαίνει ότι, η αναζήτηση θα γίνει από την επόμενη εγγραφή. Αν 2, ξεκινά από την αρχή και ψάχνει για συγκεκριμένη εγγραφή. Η τιμή του ορίσματας mode = 2 δουλεύει χωρίς προβλήματα μόνο σε πίνακες που έχουν index. Οι τιμές Mode=0 και Mode=1 αν αποτύχουν δεν επηρεάζουν την τρέχουσα εγγραφή. Χρησιμοποιώντας ένα νούμερο για τιμή του fieldName αναγκάζουμε την αναζήτηση βάσει κλειδιού.

```
to get searchTable fieldName, fieldValue, mode, indexID
        DbTable = "database"            -- το όνομα της Βάσης Δεδομένων
        -- αρχικοποιεί την επιστρεφόμενη τιμή
        retVal = "SUCCESS"
        -- πρέπει να υπάρχει κάποια τιμή για mode
        if mode = NULL
                mode = 0
        end if
        -- αν το πεδίο είναι τύπου Real (πραγματικός αριθμός)
        if isType(REAL, fieldName)
                get searchPXKey(DBTable, fieldName, fieldValue, mode)
        else
                -- πρέπει να υπάρχει ένα IndexID
                if indexID = NULL
                        indexID = 0
                end if
                get emptyPXRecord(DBTable)      -- καθαρίζει τον buffer
                get setPXFieldValue(DBTable, fieldName, fieldValue)
                get searchPXField(DBTable, fieldName, indexID, mode)
        end if

        if it < 0       -- Η εγγραφή δεν βρέθηκε
                sysError = getPXErrorString(it)
                if mode = 2
                        if it = -89
                                retVal = "Closest"
                        else            -- αν δηλαδή το it = -101, δηλ τέλος πίνακα
                                retVal = "End Of Table"
                        end if
                else                    -- αλλιώς επιστρέφει ότι δεν το βρήκε
                        retVal = "Not Found"
```

```
        end if
    end if
    return retVal
end searchTable
```

ΚΕΦΑΛΑΙΟ 5: ΕΠΙΛΟΓΟΣ - ΣΥΜΠΕΡΑΣΜΑΤΑ

Η αποφοίτηση ενός νέου επιστήμονα από το Πανεπιστήμιο και η είσοδός του είτε στον επαγγελματικό χώρο είτε στο στίβο των μεταπτυχιακών σπουδών, έρχεται ως αποτέλεσμα μιας εξελικτικής εκπαιδευτικής διαδικασίας. Φαινομενικά, η μετάβαση αυτή είναι συνεχής και προκύπτει ως το λογικό πέρας της ακατάπαυστης εισροής γνώσεων κατά τη διάρκεια της εκπαιδευτικής διαδικασίας. Η πείρα, όμως, έχει αποδείξει ότι μεταξύ της αποφοίτησης και της πλήρους ένταξης στο νέο ανταγωνιστικό πεδίο, υπάρχει ένα μεγάλο κενό. Οι νεοεισερχόμενοι βρίσκονται μπροστά στο κενό αυτό, στη μία πλευρά του οποίου υπάρχει το πεδίο της θεωρίας το οποίο έχουν διανύσει με τόσο κόπο, επιτυχώς μέχρι τώρα, και στην άλλη το πεδίο της πράξης και της εφαρμογής.

Για την πλήρωση αυτού του χάσματος απαιτείται μια δοκιμασία η οποία θα αποτελέσει το βάπτισμα του πυρός, θα βεβαιώσει τους άλλους για τις ικνότητες των νεοεισερχομένων και θα πείσει τους ίδιους για την επάρκεια των ικανοτήτων αυτών που απαιτούνται για τη συνέχιση των προσπαθειών τους στο νέο χώρο.

Η παρούσα πτυχιακή εργασία αποτέλεσε την υλοποίηση του στόχου που περιγράφηκε πιο πάνω.

Στα πλαίσια, λοιπόν, της πτυχιακής μας εργασίας στο 4° έτος του τμήματος Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας, αναπτύξαμε μια εφαρμογή Hypermedia, ένα εκπαιδευτικό πρόγραμμα που αφορά το μάθημα των Ψηφιακών Ηλεκτρονικών. Η εφαρμογή αυτή, που αναπτύχθηκε με τη χρήση του πακέτου Multimedia Toolbook της Asymetrix v. 3.0, περιλαμβάνει το τμήμα της διδασκαλίας της Ψηφιακής θεωρίας, το τμήμα της εξάσκησης πάνω σε μια σειρά σχετικών ασκήσεων και το τμήμα της αξιολόγησης πάνω στη διδαχθείσα ύλη, το οποίο αναπτύχθηκε ξεχωριστά και μπορεί να χρησιμεύσει ως τμήμα αξιολόγησης για οποιαδήποτε μαθήματα.

Ο χαρακτηρισμός της εφαρμογής ως Hypermedia, δικαιολογείται πλήρως με βάση τους ορισμούς που δόθηκαν στο 1° κεφάλαιο για τη σχετική ορολογία. Η εφαρμογή χρησιμοποιεί όλων των μορφών τις πηγές πληροφορίας (κείμενο, υπερκείμενο, ήχο, προσομοίωση κίνησης - animation, video), ενώ ταυτόχρονα είναι πλήρως αλληλεπιδραστική με το χρήστη, αφού του παρέχει ένα πλήθος δυνατοτήτων πλοήγησης, ελέγχου και γενικά χειρισμού της εφαρμογής, ανάλογα με τις επιθυμίες του. Επιπλέον, η διασύνδεση του χρήστη με το πρόγραμμα (user interface) έχει σχεδιαστεί με γνώμονα την ευκολία στη χρήση, την απλότητα, την εύκολη κατανόηση των χρησιμοποιημένων συμβόλων, και την ομοιογένεια, χρησιμοποιώντας εικονίδια, επεξηγηματικά μενού, buttons κ.λπ. με τη βοήθεια του mouse ή του πληκτρολογίου.

Κατά τη διάρκεια της ενασχόλησής μας με την εργασία αυτή, βρεθήκαμε αντιμέτωποι με τις ιδιαιτερότητες και τις απαιτήσεις πραγματικών καταστάσεων και επιβεβαιώθηκε επανειλλημένα ότι ένα πράγμα δεν είναι πάντα τόσο απλό όσο φαίνεται αρχικά. Δεν ήταν λίγες οι φορές που αναθεωρήσαμε τις σκέψεις για την υλοποίηση κάποιου τμήματος της εργασίας ή και για το δρόμο που θα ακολουθούσε αυτή η ίδια η εργασία. Ωστόσο, η

αποκόμιση της όποιας πείρας αποζημίωσε με το παραπάνω κάθε προσπάθεια και κόπο.

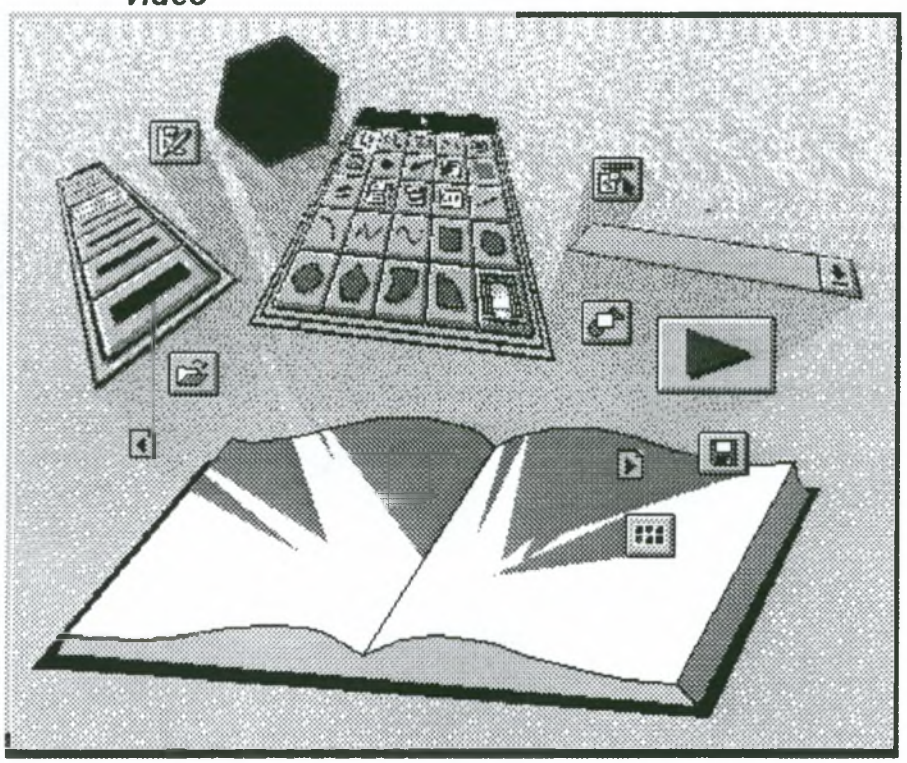
Αναμφισβήτητα, τα περιθώρια βελτίωσης είναι μεγάλα. Άλλωστε ο περιορισμένος διατιθέμενος χρόνος για την υλοποίηση μιας Hypermedia εφαρμογής που ικανοποιεί κάποιες ελάχιστες απαιτήσεις, όπως προηγουμένως ανφέρθηκαν, οπωσδήποτε αποτελεί ανασταλτικό παράγοντα. Είτε βελτίωση στον κώδικα ώστε αυτός να γίνει απλούστερος και συνεκτικότερος, είτε ακόμη και στο ίδιο το user interface για να γίνει ακόμη πιο ελκυστικό και «μεταδοτικότερο», αποτελούν σημεία παρέμβασης.

Η παρούσα μορφή της εφαρμογής, σίγουρα, είναι μια βάση για παραπέρα μελέτη και εμβάθυνση. Το αν θα αποτελέσει και το εφαλτήριο μελλοντικών προοπτικών στο ευρύτατο και ραγδαία αναπτυσσόμενο πεδίο των Hypermedia - Multimedia εξαρτάται μόνο από τη φαντασία, την όρεξη και τις ικανότητες των εμπλεκομένων.

ΠΑΡΑΡΤΗΜΑ Α: ΕΙΣΑΓΩΓΗ ΣΤΟ MULTIMEDIA - TOOLBOOK ΕΚΔ. 3.0 ΤΗΣ ASYMETRIX

Η ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ αναπτύχθηκε με τη χρήση του «πακέτου» **Multimedia Toolbook Version 3.0** της εταιρείας **Asymetrix**. Το Toolbook της Asymetrix είναι ένα *εργαλείο συγγραφής* (*authoring tool*) για την ανάπτυξη πολυμεσικών εφαρμογών. Αυτό σημαίνει πως με τη χρήση του Toolbook μπορούμε να αναπτύξουμε εφαρμογές που να συνδυάζουν ποικίλους τρόπους παρουσίασης της πληροφορίας, δηλαδή,

κείμενο
ήχο
εικόνα
γραφικά και κινούμενο σχέδιο
video



Το Toolbook χαίρει ιδιαίτερης εκτίμησης ανάμεσα στους χρήστες και δικαιολογημένα, αφού παρουσιάζει κάποια πλεονεκτήματα στη χρήση του σε σχέση με τα ανταγωνιστικά πακέτα [5]:

♦ Ευκολία στην κατανόηση, στη χρήση και στην εκμάθηση αφού προσφέρει ένα πολύ φιλικό περιβάλλον στο χρήστη.

♦ Προσομοιώνει ένα «ηλεκτρονικό βιβλίο» το οποίο ο χρήστης μπορεί να ξεφυλλίσει. Άρα πρόκειται για κάτι το οικείο σε όλους.

♦ Ισχυρά χαρακτηριστικά με την ενσωματωμένη γλώσσα προγραμματισμού που διαθέτει.

Στη συνέχεια θα επιχειρήσουμε μια σύντομη εισαγωγή στο Toolbook, αναφερόμενοι στα κυριότερα χαρακτηριστικά του.

Α.1 ΓΕΝΙΚΑ ΓΙΑ ΤΟ TOOLBOOK

Α.1.1 Εισαγωγή

Ένα βιβλίο περιλαμβάνει:

- Graphic Objects (Αντικείμενα Γραφικών) (είναι σχήματα που δημιουργήθηκαν στο ΤΒ ή εισήχθησαν (imported) ή επικολλήθηκαν στο ΤΒ από κάποια άλλη εφαρμογή).
- Fields (Πεδία) (είναι αντικείμενα στα οποία μπορεί να γραφεί κείμενο).
- Viewers (είναι παράθυρα που δημιουργούνται στο ΤΒ και μπορούν να απεικονίζουν τις σελίδες ενός βιβλίου).
- Pages (Σελίδες. Είναι η βασική μονάδα ενός βιβλίου).
- Buttons (Πλήκτρα)
- Background (Παρασκήνιο) (Ειδικό αντικείμενο που το περιεχόμενό του φαίνεται από κάθε σελίδα του βιβλίου).

Ο χρήστης μπορεί να «χτίσει» ένα ηλεκτρονικό βιβλίο δημιουργώντας σελίδες, τοποθετώντας επάνω τους αντικείμενα και μετά, γράφοντας κάποιο κώδικα (Scripts = σενάρια) με την βοήθεια της γλώσσας OpenScript του ΤΒ, να αναγκάσει τα αντικείμενα να εκτελέσουν κάποιες επιθυμητές ενέργειες.

Α.1.2 Επίπεδο Συγγραφέα και Αναγνώστη

Στο ΤΒ μπορεί να δημιουργηθεί (σχεδιαστεί) αλλά και να εκτελεστεί μία εφαρμογή. Η σχεδίαση γίνεται στο **επίπεδο συγγραφέα (Author Level)**, στο επίπεδο δηλαδή που υπάρχουν τα κατάλληλα εργαλεία και είναι προσιτές οι ανάλογες εντολές για την δημιουργία και τροποποίηση κάποιου αντικειμένου, ενώ η εκτέλεση γίνεται στο **επίπεδο αναγνώστη (Reader Level)**, στο επίπεδο εκείνο που οι εφαρμογές ελέγχονται και εκτελούνται.

Με άλλα λόγια, στο επίπεδο συγγραφέα ο χρήστης δημιουργεί και τροποποιεί τα βιβλία του, τα οποία «εκτελεί» και ελέγχει στο επίπεδο αναγνώστη [6].

Επίσης ο χρήστης μπορεί να γράψει κώδικα (σενάρια) για το κάθε αντικείμενο και έτσι να ελέγξει τη συμπεριφορά του. Η εμφάνιση του ΤΒ στο επίπεδο του συγγραφέα αντανακλά τις ισχυρές δυνατότητες ανάπτυξης που

έχει. Τα μενού και οι γραμμές εργαλείων (tool bars) περιέχουν εντολές και εργαλεία σχεδίασης τα οποία ο χρήστης μπορεί να χρησιμοποιήσει ώστε να τροποποιήσει τα αντικείμενα και τις ιδιότητές τους.

Συνοπτικά διακρίνουμε τις δυνατότητες του χρήστη σε κάθε επίπεδο στις ακόλουθες:

Author Level

- Δημιουργία μίας εφαρμογής
- Δημιουργία αντικειμένου
- Αλλαγή ιδιοτήτων αντικειμένου
- Συγγραφή σεναρίων

Reader Level

- Διόρθωση (editing) σε πεδία κειμένου
- Εκτύπωση
- Εκτέλεση εφαρμογής [6]

Το ΤΒ εκτελεί τις εφαρμογές του χρήστη στο Reader Level. Εξ ορισμού (by default), το Reader Level προσφέρει αρκετές λειτουργίες που προσφέρονται και στο Author Level, όπως εισαγωγή κειμένου σε διορθώσιμα πεδία και εκτύπωση της εφαρμογής. Είναι δυνατόν, επιπλέον, να ρυθμιστεί ώστε να απαγορεύεται στον χρήστη η μετάβαση στο Author Level.

Για να αλλάξουμε από Author Level σε Reader Level και αντίστροφα:

1. Πατάμε **F3** ή διαλέγουμε από το **EDIT** μενού την επιλογή **Reader**
2. Πατάμε **F3** ή διαλέγουμε από το **EDIT** μενού την επιλογή **Author**

Α.1.3 Αντικειμενοστραφής προγραμματισμός

Το ΤΒ είναι ένα αντικειμενοστραφές οδηγούμενο από συμβάντα περιβάλλον προγραμματισμού.

Στο περιβάλλον του ΤΒ, αρχικά, αναπτύσσεται μία εφαρμογή με την δημιουργία αντικειμένων, και μετά ορίζεται η συμπεριφορά αυτών στο τμήμα Επεξεργασίας Σεναρίων (Script Editor) των αντικειμένων.

Script (σενάριο) είναι μία σειρά καταστάσεων που ορίζουν τη συμπεριφορά των αντικειμένων στο ΤΒ, δηλαδή, είναι το σύνολο εκείνων των εντολών που γράφονται με την βοήθεια της γλώσσας OpenScript, και οι οποίες ορίζουν πως θα απαντήσει το ΤΒ σε συγκεκριμένα συμβάντα.

Το ΤΒ δεν είναι ένα παραδοσιακό γραμμικό περιβάλλον προγραμματισμού, επειδή δεν χρειάζεται να γράφονται οι εφαρμογές ορίζοντας δεδομένα και ρουτίνες [6].

Α.1.4 Οδηγούμενο από Συμβάντα (Events-driven)

Το ΤΒ είναι ένα πρόγραμμα οδηγούμενο από συμβάντα, επειδή τα σενάρια στις εφαρμογές του ΤΒ δεν εκτελούνται έως ότου ενεργοποιηθούν από κάποια συμβάντα.

Οποιαδήποτε ενέργεια στις εφαρμογές του ΤΒ είναι ένα συμβάν. Οι χρήστες της εφαρμογής, αλλά και άλλες εφαρμογές, μπορούν να ενεργοποιήσουν συμβάντα.

Για παράδειγμα, το κλικ στο πλήκτρο του mouse, η μετακίνηση του δείκτη του και η επιλογή μίας εντολής του μενού, ερμηνεύονται από το ΤΒ σαν συμβάντα. Ακόμη και όταν τίποτα δεν συμβαίνει, αυτό είναι ένα συμβάν (το συμβάν Idle).

Όταν λαμβάνει χώρα ένα συμβάν, το ΤΒ λαμβάνει ένα μήνυμα (message). Τα μηνύματα συνδέουν τα συμβάντα με τις ενέργειες που τα προκάλεσαν στις διάφορες εφαρμογές του ΤΒ.

Κάθε συμβάν δημιουργεί ένα ή περισσότερα μοναδικά messages τα οποία στέλνονται στο αντικείμενο που επηρεάζεται από το συμβάν. Για παράδειγμα αν γίνει κλικ στο πλήκτρο του mouse όταν ο δείκτης του είναι πάνω από ένα αντικείμενο, τότε στέλνεται το μήνυμα **ButtonClick** στο αντικείμενο.

Με τη χρήση της γλώσσας προγραμματισμού OpenScript, του ΤΒ, γράφεται ένα σενάριο για ένα αντικείμενο, το οποίο καθορίζει πώς θα αποκριθεί η εφαρμογή σε ένα συγκεκριμένο message που στέλνεται στο αντικείμενο.

Ένας **χειριστής μηνυμάτων (handler)** είναι ένα σύνολο εντολών της γλώσσας OpenScript μέσα σε ένα script, και καθορίζει ποια θα είναι η απάντηση του ΤΒ σε ένα συγκεκριμένο συμβάν, (όπως π.χ. όταν ο χρήστης κάνει click σε ένα αντικείμενο). Ένα σενάριο μπορεί να περιέχει περισσότερους από έναν handlers, που ο καθένας να απαντά σε διαφορετικό συμβάν. Σε ένα script αντικειμένου, ο handler, αποφασίζει ποιο message θα εκτελεστεί μέσα από το σενάριο, και τι εντολή του ΤΒ θα εκτελεστεί.

Π.χ.

```
To Handle ButtonUp
        Beep
End
```

- Ο παραπάνω handler χειρίζεται το μήνυμα ButtonUp (απελευθέρωση του αριστερού πλήκτρου του Mouse), και η απάντησή του είναι να στέλνει στο μεγαφωνάκι του υπολογιστή ένα Beep.

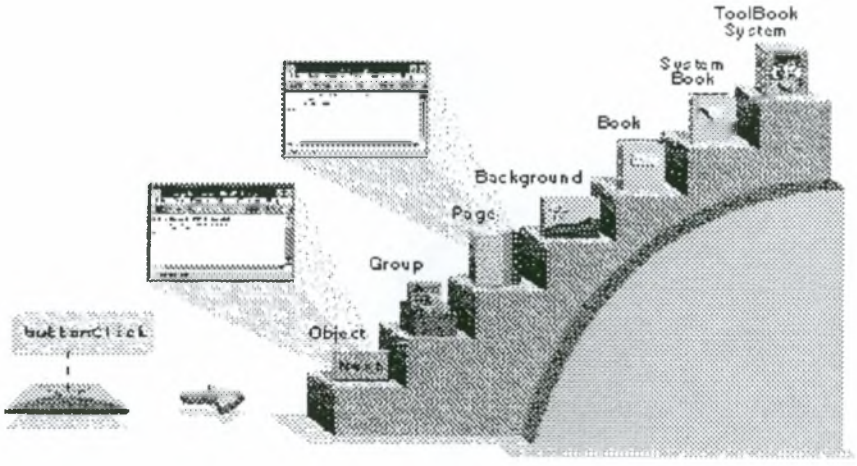
Α.1.5 Η Ιεραρχία των αντικειμένων

Η εσωτερική οργάνωση του ΤΒ καλείται ιεραρχία των αντικειμένων. Η ιεραρχία κατά αύξουσα σειρά είναι:

- Fields, Buttons, Graphic Objects
- Groups
- Pages
- Backgrounds
- Books
- System Books (Είναι ένα βιβλίο που το σενάριό του το διαμοιράζονται άλλα βιβλία. Τα System Books μπορούν να

χρησιμοποιηθούν για να ορίσουν γενικά σενάρια για χρήση από άλλα βιβλία).

- Σύστημα του ToolBook



Κάθε group, page, background ή book που περιέχει κάποιο αντικείμενο, θεωρείται πατέρας του αντικειμένου αυτού.

Τα messages στο ΤΒ ταξιδεύουν από αντικείμενο σε αντικείμενο προς τα πάνω στην ιεραρχία.

Αν ένα script αντικειμένου δεν περιέχει handler ενός μηνύματος που στέλνεται στο αντικείμενο τότε το μήνυμα ταξιδεύει προς τα πάνω στην ιεραρχία, στον πατέρα του αντικειμένου, μέχρι να βρει κάποια απάντηση, ή μέχρι να φτάσει στο σύστημα του ToolBook.

Η ροή του ταξιδιού του μηνύματος μπορεί να αλλάξει στέλνοντας ένα μήνυμα κατευθείαν στο αντικείμενο ή προσπερνώντας (forward) την ιεραρχία του αντικειμένου [6].

Α.2 ΑΝΤΙΚΕΙΜΕΝΑ ΚΑΙ ΟΙ ΙΔΙΟΤΗΤΕΣ ΤΟΥΣ

Α.2.1 Αντικείμενα που σχεδιάζει το Toolbook

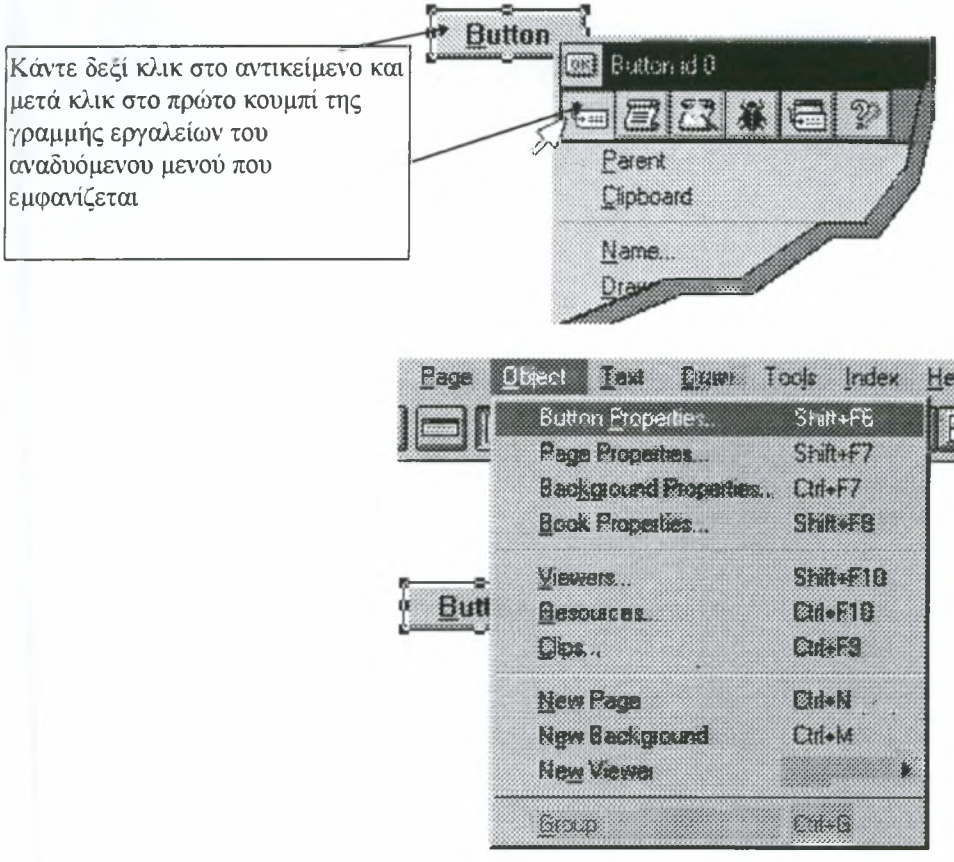
Οι εφαρμογές του ΤΒ περιέχουν αντικείμενα. Όλα τα αντικείμενα έχουν ιδιότητες οι οποίες ορίζουν την συμπεριφορά και την εμφάνιση τους. Από αυτές τις ιδιότητες αρκετές μπορούν να τροποποιηθούν. Όταν ανοίγει ένα νέο βιβλίο, το ΤΒ σχεδιάζει από μόνο του κάποια αντικείμενα όπως είναι η σελίδα, το Background (Παρασκήνιο) και ο Main Viewer (Κύριο Παράθυρο) - ένα παράθυρο που δημιουργείται στο ΤΒ και μπορεί να εμφανίζει οποιαδήποτε σελίδα ενός οποιουδήποτε βιβλίου.

Όταν ο χρήστης ανοίγει ένα Book μπορεί να προσθέσει κι άλλες Pages, backgrounds ή και Viewers ή να σχεδιάσει άλλα αντικείμενα όπως Lines (γραμμές), Buttons (πλήκτρα), και Fields (πεδία κειμένου).

Κάθε αντικείμενο στο ΤΒ έχει Properties για τις οποίες μπορούν να οριστούν τιμές, όπως και Properties που τις τιμές τους τις ορίζει το ίδιο το ΤΒ. Για παράδειγμα μπορεί ο χρήστης να αλλάξει το όνομα ενός αντικειμένου

αλλά το TB είναι εκείνο που δίνει στο αντικείμενο το μοναδικό του ID (το οποίο δεν μπορεί να αλλάξει από τον χρήστη) [6].

Όταν σχεδιάζεται ένα αντικείμενο μπορεί να αλλάξει η εμφάνισή του και η συμπεριφορά του αν αλλαχτούν οι ιδιότητές του. Επιλέγοντας κάποιο αντικείμενο ο χρήστης μπορεί να ανοίξει το πλαίσιο διαλόγου Object's Properties (Ιδιότητες Αντικειμένου) είτε χρησιμοποιώντας το ανάλογο εικονίδιο στην ToolBar (Γραμμή Εργαλείων), είτε την αντίστοιχη επιλογή από το μενού **OBJECT**, είτε τέλος κάνοντας δεξί κλικ στο αντικείμενο και κλικ στο αναδυόμενο μενού που εμφανίζεται το σχετικό εικονίδιο (βλέπε σχήμα).



Βιβλία (Books)

Το TB με το που ανοίγει ένα νέο αρχείο δημιουργεί ένα Book, μία Page, ένα Background και έναν Viewer. Ο χρήστης όμως έχει τη δυνατότητα να προσθέσει περισσότερες Pages, Backgrounds και Viewers ανάλογα με το σκοπό της εφαρμογής που δημιουργεί. Ένα Book είναι συγχρόνως ένα αρχείο του TB (με επέκταση .TBK) και ένα αντικείμενο με ιδιότητες.

Οι ιδιότητες του αντικειμένου Book περιλαμβάνουν εκτός των άλλων και το File Name (όνομα του αρχείου), το προεπιλεγμένο (default) μήκος σελίδας (Page Size), ορισμένες λεπτομέρειες για το αν το Book σώζεται όταν κλείνει, και οποιαδήποτε τυχόν Passwords (κωδικοί) για το βιβλίο.

Pages

Μία Page είναι η βασική μονάδα ενός Book. Οι Pages μπορούν να περιέχουν και άλλα αντικείμενα. Τα αντικείμενα που δημιουργήθηκαν σε μία Page εμφανίζονται μόνο στη συγκεκριμένη Page, αλλά αν βρίσκονται στην

επιφάνεια του Background φαίνονται και στις άλλες Pages που διαμοιράζονται το συγκεκριμένο Background.

Είναι σπουδαίο να τεθεί η ιδιότητα Name της Page έτσι ώστε να γίνεται εύκολα αναφορά σ' αυτήν αλλά και στα αντικείμενα που είναι τοποθετημένα σ' αυτήν.

Background

Ένα Background σχεδιάζεται για να εμφανίζει το περιεχόμενό του σε μία ή περισσότερες Pages στο Book. Τα αντικείμενα στο Background εμφανίζονται σε κάθε Page η οποία ανήκει στο συγκεκριμένο Background. Ένα Background επίσης ορίζει ποιο θα είναι το χρώμα των Pages που κείνται επί αυτού του background.

Η μετάβαση από την Page στο Background γίνεται είτε με την επιλογή **Background** του μενού **VIEW** είτε πατώντας το πλήκτρο **F4**.

Viewers

Ένας Viewer είναι ένα παράθυρο που δημιουργείται στο ΤΒ και μπορεί να εμφανίζει οποιαδήποτε Page από κάποιο Book. Ένας Viewer μπορεί να έχει στοιχεία όπως System Menu (Το εικονίδιο στην πάνω αριστερή γωνία), Caption Bar (Γραμμή Τίτλου), Minimize/Maximize Buttons (Πλήκτρα Ελαχιστοποίησης και Μεγιστοποίησης), Border (Περίγραμμα) και Status Bar (Γραμμή Κατάστασης). Η Status Bar του κάθε Viewer μπορεί να εξαφανιστεί (ή να εμφανιστεί) πιέζοντας το **F12.**

Όταν ο χρήστης ανοίγει ένα νέο αρχείο το ΤΒ αυτόματα δημιουργεί έναν Main Viewer με ID 0. Οι ιδιότητες αυτού του παραθύρου μπορούν να αλλάξουν από τον χρήστη (όπως η Caption), όμως το παράθυρο το ίδιο δεν μπορεί να σβηστεί όπως μπορεί να γίνει με τους άλλους Viewers [6].

Α.2.2 Αντικείμενα που σχεδιάζει ο χρήστης

Tool Palette (Παλέτα Εργαλείων)

Με τη βοήθεια της Tool Palette, μπορούν να σχεδιαστούν τα ακόλουθα αντικείμενα: Buttons (Πλήκτρα), Fields (Πεδία Κειμένου), Record Fields (Πολλαπλά Πεδία Κειμένου), List Box (Πλαίσια Λίστας), Combo Box (Σύνθετα Πλαίσια Λίστας), OLE (Αντικείμενο Σύνδεσης και Ενσωμάτωσης = Object Linking & Embedded), και Graphic Objects (Αντικείμενα Σχεδίασης). Ακόμη με την επιλογή της ομαδοποίησης είναι δυνατός ο συνδυασμός δύο ή παραπάνω αντικειμένων σε κάποιο αντικείμενο Group.

Αφού επιλεγεί κάποιο αντικείμενο μπορεί να τροποποιηθεί η εμφάνισή του αλλάζοντας τις ιδιότητές του.

ΕΡΓΑΛΕΙΑ ΤΗΣ TOOL PALETTE

Selection Tool (Εργαλείο Επιλογής): Εργαλείο για την επιλογή αντικειμένων.

Magnify Tool (Εργαλείο Μεγέθυνσης/Σμίκρυνσης): Μεγεθύνει την άποψη της περιοχής της σελίδας, (ή με Ctrl σμικρύνει).

Push Button (Πλήκτρο): Σχεδιάζει ένα Button (ένα τρισδιάστατο ορθογώνιο το οποίο φαίνεται να πιέζεται όταν ο χρήστης κάνει κλικ με το mouse επάνω του κατά την εκτέλεση της εφαρμογής (Reader Level)).

Label Button (Πλήκτρο Ετικέτα): Σχεδιάζει ένα Button στην επιφάνεια του οποίου υπάρχει κείμενο με κάποιο γράμμα σαν Shortcut Key (Κλειδί Συντόμευσης).

Radio Button (Πλήκτρο Επιλογής): Σχεδιάζει ένα Button στην επιφάνεια του οποίου υπάρχει ένας κύκλος. Ο κύκλος γίνεται μαύρος ή άσπρος διαδοχικά όταν δέχεται κλικς στο Reader Level.

Check Box (Πλήκτρο Μαρκαρίσματος): Σχεδιάζει ένα Button στην επιφάνεια του οποίου υπάρχει ένα τετράγωνο. Ένα **X** εμφανίζεται μέσα στο τετράγωνο όταν επιλέγεται στο Reader Level.

3D Radio Button (Τρισδιάστατο Πλήκτρο Επιλογής): Σχεδιάζει ένα τρισδιάστατο πλήκτρο επιλογής.

3D Check Box (Τρισδιάστατο Πλήκτρο Μαρκαρίσματος): Σχεδιάζει ένα τρισδιάστατο πλήκτρο μαρκαρίσματος.

Field (Πεδίο κειμένου): Σχεδιάζει ένα αντικείμενο το οποίο μπορεί να περιέχει κείμενο (ή και εικόνα). Το αντικείμενο είναι κατά την δημιουργία του μη διάφανο με απλό περίγραμμα και μπορεί να δεχτεί κείμενο και στο Reader Level (ανάλογα με την τιμή που έχει η ιδιότητα Activate).

Record Field (Πολλαπλό Πεδίο Κειμένου): Σχεδιάζει ένα Field το οποίο μπορεί να βρίσκεται μόνο στο Background. Το Field αυτό στην ουσία είναι ένα σύνολο από απλά Fields που περιέχουν διαφορετικό κείμενο σε κάθε Page που διαμοιράζεται το συγκεκριμένο Background. Από την δημιουργία του το Record Field μπορεί να δεχτεί κείμενο και στο Reader Level.

Borderless Field (Πεδίο χωρίς περίγραμμα): Σχεδιάζει ένα χωρίς περιθώριο διαφανές Field. Το πεδίο αυτό δεν μπορεί να δεχτεί κείμενο στο Reader Level.

List Box (Πλαίσιο Λίστας): Σχεδιάζει ένα μη διάφανο κυλιόμενο πεδίο που μπορεί να εμφανίζει μία λίστα από αντικείμενα που μπορούν να επιλεχθούν. Από τη δημιουργία του το List Box είναι λίστα μίας επιλογής και μη διορθώσιμο στο Reader Level.

Combo Box (Σύνθετο Πλαίσιο Λίστας): Σχεδιάζει ένα List Box που αποτελείται από ένα πλαίσιο κειμένου, ένα Push Button, και μία drop-down (αναδιπλούμενη) λίστα.

OLE Container: Σχεδιάζει ένα Object Linking και Embedding (Αντικείμενο Σύνδεσης και Ενσωμάτωσης) (OLE), το οποίο μπορεί να περιέχει μία αναφορά σε ένα συνδεδεμένο αντικείμενο ή ένα αντίγραφο ενός ενσωματωμένου αντικειμένου.

Line (Γραμμή): Σχεδιάζει μία ευθεία γραμμή.

Arc (Τόξο): Σχεδιάζει ένα τόξο.

Angled Line (Τεθλασμένη Γραμμή): Σχεδιάζει μία τεθλασμένη γραμμή.

Curve (Καμπύλη): Σχεδιάζει μία καμπύλη.

Rectangle (Ορθογώνιο): Σχεδιάζει ένα ορθογώνιο ή ένα τετράγωνο.

Rounded Rectangle (Στρογγυλεμένο Ορθογώνιο): Σχεδιάζει ορθογώνιο με στρογγυλεμένες γωνίες.

Ellipse (Ελλειψοειδές): Σχεδιάζει έλλειψη ή κύκλο.

Polygon (Πολύγωνο): Σχεδιάζει ένα κανονικό πολύγωνο.

Irregular Polygon (Ακανόνιστο Πολύγωνο): Σχεδιάζει ένα κλειστό αντικείμενο με άνισες γωνίες και πλευρές με διαφορετικό μήκος.

Pie (Κυκλικός Τομέας): Σχεδιάζει ένα κλειστό κυκλικό τομέα (δεν μπορεί να ομαδοποιηθεί).

Stage (Σκηνή): Σχεδιάζει μία σκηνή μέσα στην οποία μπορούν να παιχτούν clip (video, image, animation).

Hotwords και groups

Πέρα από τα αντικείμενα που μπορεί να σχεδιάσει ο χρήστης μέσω της Tool Palette, μπορεί ακόμη να ορίσει και hotwords αλλά και groups. Μία **hotword** είναι κείμενο εντός ενός field ή record field που έχει δικό του ID και properties όπως όλα τα αντικείμενα. Όπως τα buttons έτσι και οι hotwords μπορούν να ενεργοποιήσουν συμβάντα και να στείλουν μηνύματα. **Group** είναι ένα αντικείμενο που αποτελείται από δύο ή περισσότερα αντικείμενα. Ο χρήστης μπορεί να συνδυάσει διαφορετικού τύπου αντικείμενα (που μπορεί να περιλαμβάνουν και αυτά με τη σειρά τους κι άλλα groups) σε ένα group, έπειτα να αλλάξει τις ιδιότητές του ή να γράψει ένα σενάριο (script) γι' αυτό. Για να δημιουργηθεί έναν group πρέπει να επιλεγούν τα αντικείμενα που θα αποτελούν το group και κατόπιν να επιλεγεί από το μενού OBJECT η επιλογή group. (Το ίδιο μπορεί να γίνει με το συνδυασμό συντόμευσης Ctrl + G).

Α.2.3 Ιδιότητες που ορίζει ο χρήστης (User Properties)

Μία user property είναι μία ιδιότητα που ο χρήστης μπορεί να δημιουργήσει και να την συσχετίσει με κάποιο αντικείμενο. Το ΤΒ διατηρεί τις τιμές που αποθηκεύονται στις user properties ακόμη και ανάμεσα στις εκδόσεις του. (Το ΤΒ υποστηρίζει την δυνατότητα να «εκτελέσει» ο χρήστης ένα δεύτερο book κατά την ώρα «τρεξίματος» μίας εφαρμογής, δηλ. υπάρχει περίπτωση να βρίσκονται «ανοιχτά» σε κατάσταση Reader Level δύο books - ή ακόμη και το ίδιο book - σε άλλη έκδοση).

Η τιμή που έχει δοθεί σε κάποια user property μπορεί να χρησιμοποιηθεί και σαν καθολική (global) μεταβλητή και να είναι προσπελάσιμη από όλα τα αντικείμενα του book (παρόμοια δηλαδή με τις μεταβλητές συστήματος (system variables)).

Για να καταργηθεί μία user property αρκεί να τεθεί στην τιμή null [6].

Α.3 ΣΥΓΓΡΑΦΗ ΚΩΔΙΚΑ

Α.3.1 Επισκόπηση σεναρίων (scripting overview)

Scripts

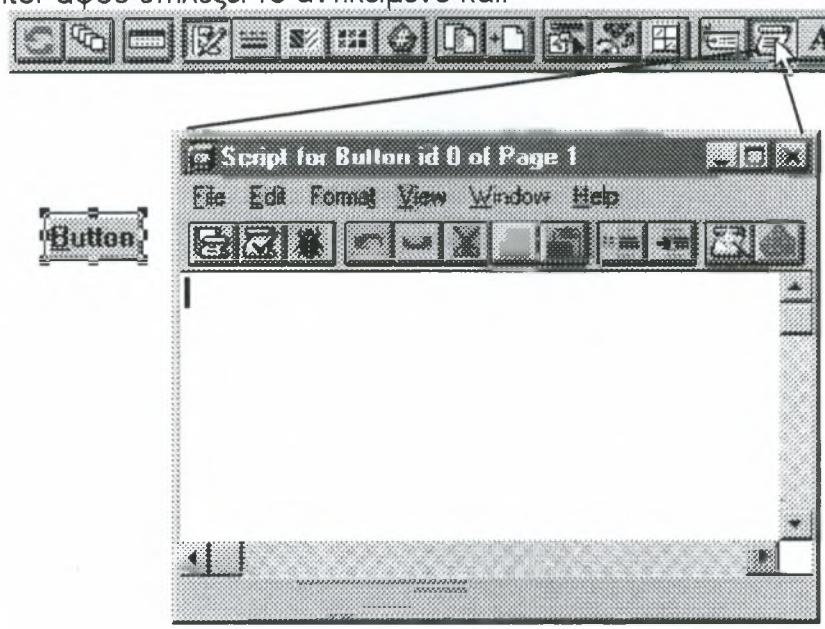
Ενα script αποτελεί μία μοναδική property των αντικειμένων η οποία μπορεί να καθορίσει την συμπεριφορά τους στο ΤΒ. Κάθε αντικείμενο του ΤΒ μπορεί να έχει ένα script. Τα scripts γράφονται με μία ειδική γλώσσα (OpenScript Language) στον OpenScript Editor.

Τα σενάρια δημιουργούνται και διορθώνονται στο Author Level ενώ ελέγχονται στο Author και Reader Level με την βοήθεια του Παραθύρου Εντολών (Command Window).

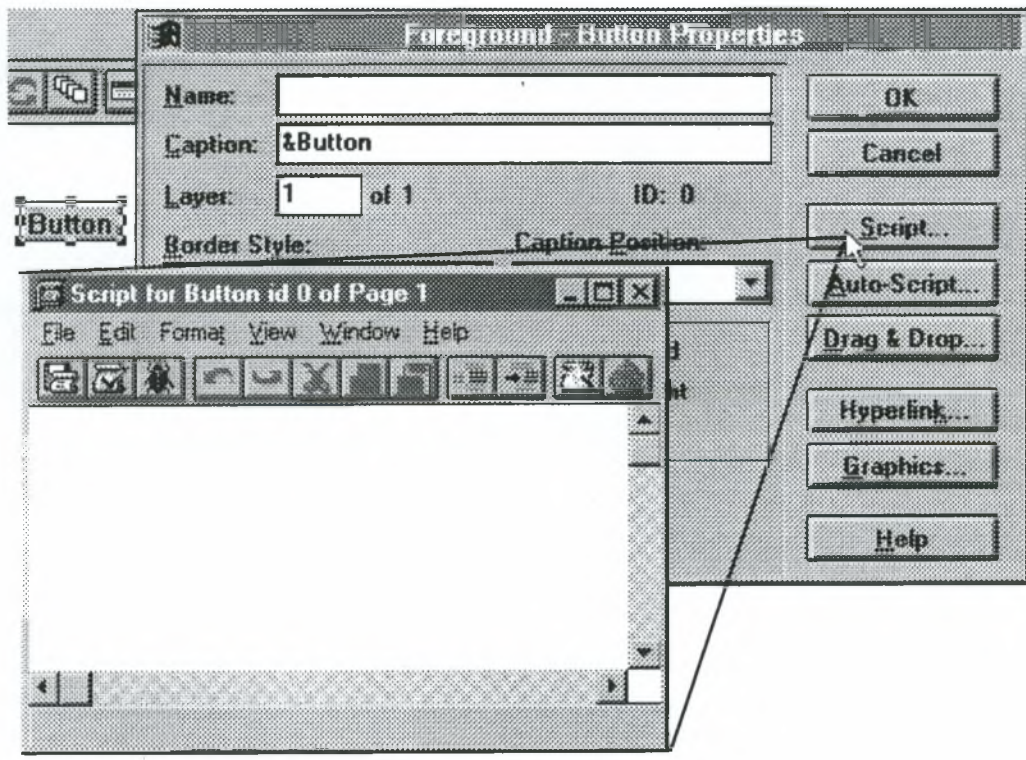
Ο Command Window είναι ένα παράθυρο στο οποίο μπορούν να εκτελεστούν εντολές της γλώσσας OpenScript. Σ'αυτό, ο χρήστης, μπορεί να «τρέξει» όλα τα σενάρια της εφαρμογής του στο Reader Level.

Script Editor

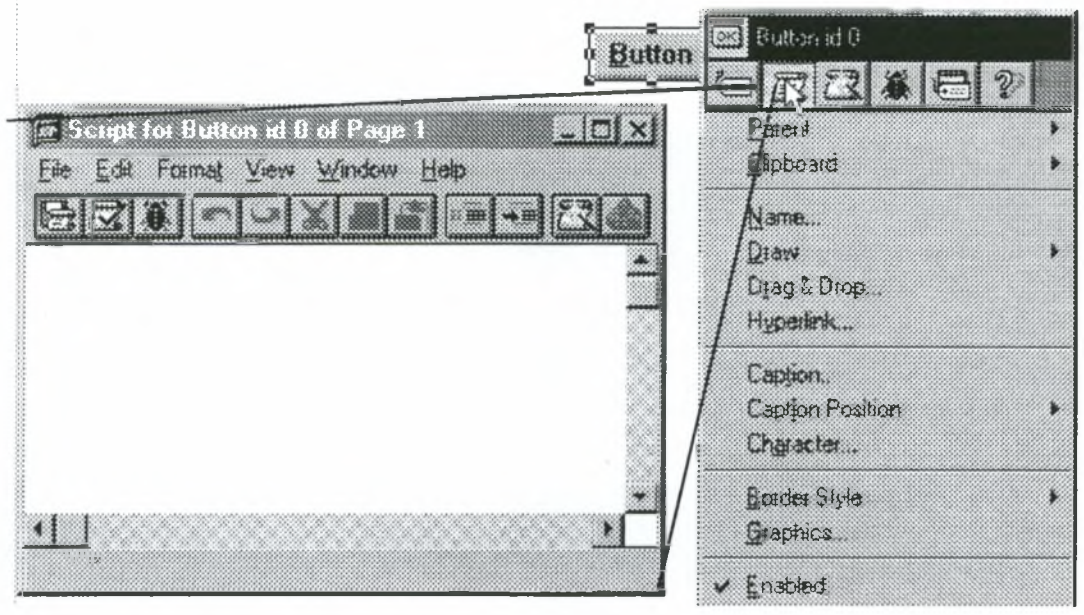
Ο χρήστης μπορεί να επεξεργαστεί το script ενός αντικειμένου στον Script Editor αφού επιλέξει το αντικείμενο και:



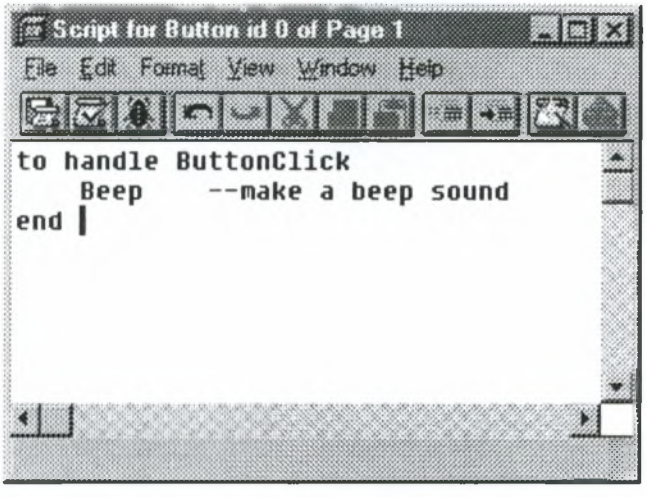
Κάνει κλικ στο πλήκτρο Script του Tool Bar



Ανοίξει το πλαίσιο διαλόγου Object Properties, και μετά κάνει κλικ στο πλήκτρο Script.



- Κάνει κλικ στο πλήκτρο Script στο αναδυόμενο μενού που εμφανίζεται μετά από το δεξί κλικ πάνω στο αντικείμενο που επιλέχτηκε.



Ο Script Editor είναι ένα παράθυρο στο οποίο μπορεί κανείς να δημιουργήσει, διορθώσει, μεταφράσει και ελέγξει τη σύνταξη σεναρίων αντικειμένου. Μπορεί να εκτελεστεί κάθε άλλη δουλειά ενώ ο Script Editor είναι ανοιχτός, περιλαμβανομένης και της εκτέλεσης του σεναρίου που ήδη φαίνεται.

Handlers (Χειριστές Μηνυμάτων)

Τα Scripts αποτελούνται από σειρές δηλώσεων (statements) που ονομάζονται handlers (κάθε handler στο σενάριο ορίζει μία ενέργεια για το ΤΒ, που εκτελείται όταν ένα συμβάν λαμβάνει χώρα). Το συγκεκριμένο συμβάν συμβαίνει στο Reader Level και έτσι ενεργοποιείται ο handler για την εκτέλεση των συγκεκριμένων εντολών [7].

Ενα σενάριο αποτελείται από έναν ή περισσότερους handlers. Οι handlers περιέχουν τα εξής στοιχεία:

- Μία δομή handler η οποία λέει στο TB ποιο μήνυμα να απαντηθεί, πότε το μήνυμα θα απαντηθεί, ή ποια συνάρτηση θα πάρει ή θα δώσει μία τιμή.
- Statements οι οποίες λένε στο TB τι να κάνει όταν ο handler ενεργοποιηθεί από το μήνυμα.
- Comments (σχόλια) που χρησιμοποιούνται για σημειώσεις που πληροφορούν τον χρήστη για ορισμένα σημεία που χρειάζονται διευκρινίσεις στο σενάριο (το TB αγνοεί τα σχόλια).

π.χ.:

```
To Handle ButtonClick
        -- Make a beep Sound    <--Comment
        Beep 1                  <--Statement
End
```

Δομή Handler

Υπάρχουν 4 διαφορετικές δομές handlers:

- *Message Handlers*: καθορίζουν ποιο θα είναι το μήνυμα που το TB θα απαντήσει:

  π.χ.:
  ```
  To Handle ButtonClick
          -- Make a Beep Sound
          Beep 1
  end
  ```
- *To Get Handlers*: οι οποίοι ορίζουν νέες συναρτήσεις ή το πώς θα ληφθεί η τιμή μίας ιδιότητας χρήστη:

  π.χ.:
  ```
  -- Calculates gross margin
          To Get GrossMargin Sales, COGS
          Return Sales - COGS
  end
  ```
- *To Set Handlers*: οι οποίοι ορίζουν procedures (= υπορρουτίνες) για να θέσουν την τιμή κάποιας ιδιότητας χρήστη (user property):

  π.χ.:
  ```
  -- Sets HelpText Property
  To Set HelpText to Value
          Caption of StatusBar = Value
  end
  ```
- *Notify handlers:* ενημερώνουν αντικείμενα με μηνύματα που δεν στέλνονται σ' αυτά αλλά στη σελίδα ή το background.

  π.χ.:
  ```
  -- Disables self on last page
  Notify After EnterPage
          If This Page is Last Page
                  Enabled of self = false
          end
  end
  ```

Ένα Statement είναι μία οδηγία στο TB που του καθορίζει τι ενέργεια να εκτελέσει. Τα Statements του OpenScript μπορούν να είναι εντολές ή δομές ελέγχου (Control Structures) [7].

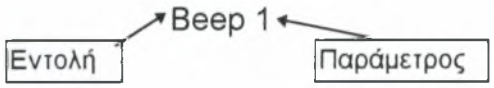
Command Statements:

συντίθενται από μία εντολή και από τυχόν απαιτούμενες παραμέτρους.

Control Structures:

είναι block statements (ομάδες εντολών) οι οποίες τρέχουν κάτω από συγκεκριμένες συνθήκες.

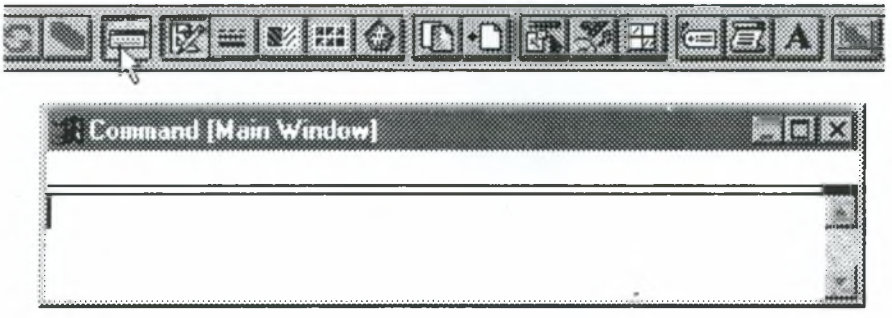
π.χ.:



Το Comment (σχόλιο) είναι μία γραμμή στο σενάριο που το TB αγνοεί: Μπορεί ο χρήστης να συμπεριλάβει σχόλια σαν σημειώσεις για τον εαυτό του. Για να δημιουργηθεί κάποιο σχόλιο απλά γράφονται 2 παύλες (hyphen) μπροστά από το κείμενο που θα γίνει σχόλιο.

Το να περιέχουν τα σενάρια σχόλια είναι μία καλή πρακτική στον προγραμματισμό. Τα σχόλια βοηθούν στην αναγνώριση κάθε τμήματος του σεναρίου και το κάνουν αναγνώσιμο από τον καθένα.

To Command Window



Στο Command Window μπορούν να εκτελεστούν εντολές OpenScript. Εντολές που γράφονται στο Command Window εκτελούνται όταν πατηθεί το πλήκτρο Enter. Είναι δυνατή η «περιήγηση» σε όλες τις εντολές που έχουν εκτελεστεί μέσα στο Command Window με τη βοήθεια των πλήκτρων PgUp και PgDn. Για να εμφανιστεί το Command Window πρέπει να γίνει κλικ στο Command button στο Tool Bar ή να επιλεγεί *Command* από το μενού VIEW, ή τέλος να πατηθεί Shift + F3 [6].

Το Command Window μπορεί να χρησιμοποιηθεί για:

- να θέσει τιμές σε properties
- να στείλει μηνύματα στα αντικείμενα στο επίπεδο συγγραφέα, ώστε να γίνει έλεγχος των σεναρίων
- να υπολογίσει την τιμή κάποιων εκφράσεων
- να εκτελέσει statements

π.χ.:

put visible of Group «NameOfGroup»

Εμφανίζει στο Command Window true ή false (ανάλογα με το αν είναι ορατό το group «NameOfGroup»).

visible of group «NameOfGroup» = false

Κάνει αόρατο το group «NameOfGroup».

send buttonClick to button "NameOfButton"

Στέλνει ένα μήνυμα buttonClick στο πλήκτρο «NameOfButton».

move group "NameOfGroup" by 200,0

Μετακινεί το group «NameOfGroup» κατά 200 οριζόντια και 0 κάθετα από την ήδη υπάρχουσα θέση του.

*put PI*3*

Εμφανίζει στο Command Windows το αποτέλεσμα της πράξης 3*3.1415962.

Α.3.2 Διόρθωση σεναρίων

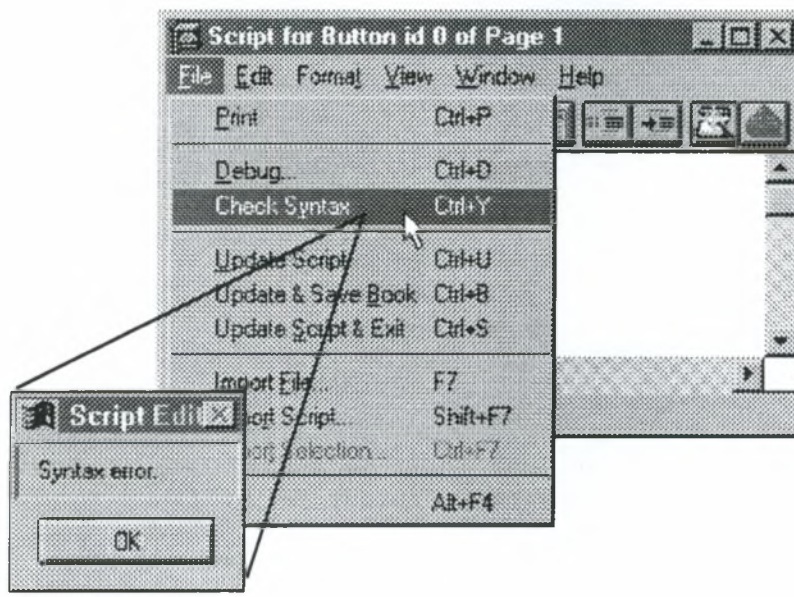
Τύποι λαθών

Μετά τη συγγραφή ενός σεναρίου υπάρχουν 2 τύποι λαθών που μπορεί να συμβούν:

- Syntax Errors (λάθη σύνταξης) στα statements του σεναρίου τα οποία απαγορεύουν στο ΤΒ να μεταφράσει κάποιο σενάριο.
- Run Time Errors (λάθη κατά την ώρα τρεξίματος) τα οποία διακόπτουν τη ροή του προγράμματος ενώ τρέχει.

Syntax Errors

Ενα syntax error είναι ένα λάθος στη σύνταξη ενός statement το οποίο εμποδίζει το ΤΒ να μεταφράσει το σενάριο. Η σύνταξη ενός σεναρίου ελέγχεται διαλέγοντας το Check Syntax από το FILE μενού του Script Editor.



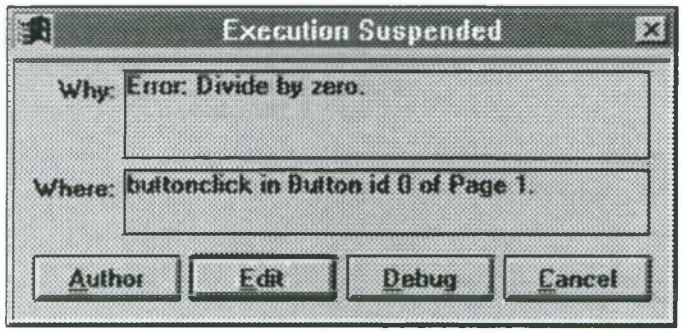
Οταν το ΤΒ ανακαλύπτει ένα syntax error, εμφανίζει με έντονα γράμματα την περιοχή στην οποία βρέθηκε το λάθος, και παρουσιάζει ένα μήνυμα του Script Editor στο οποίο περιγράφεται το συγκεκριμένο λάθος.

Run Time Errors

Τα run time errors εμποδίζουν την εκτέλεση σεναρίου ενώ η εφαρμογή τρέχει. Γι' αυτό και το ΤΒ διακόπτει την ροή του προγράμματος στο Reader

Level και εμφανίζει το πλαίσιο διαλόγου Execution Suspended. Στο πλαίσιο αυτό ο χρήστης έχει τις εξής επιλογές:

- να μεταβεί στο Author Level (επιλογή Author)
- να κάνει διόρθωση (επιλογή Edit) στο σενάριο
- να χρησιμοποιήσει τον απασφαλματωτή (Debugger) (επιλογή Debug)
- να απορρίψει το μήνυμα του Executing Suspended και να συνεχίσει να εκτελεί το σενάριο (επιλογή Cancel)



Στο τμήμα **Why** αναφέρεται γιατί έγινε το λάθος, ενώ στο **Where** πού ακριβώς στο script έγινε το λάθος.

Τα run time errors συχνά συμβαίνουν από λάθος τιμές ή ενέργειες. Τυπικά λάθη είναι:

- Αναφορά σε μη υπάρχοντα αντικείμενα
- Αλλαγή σε property που απαγορεύεται να αλλαχθεί.
- Ανάθεση σε μία property μιας μη νόμιμης τιμής.
- Προσπάθεια να προσπελαστούν δεδομένα με λάθος τρόπο.
- Καθορισμός λάθος παραμέτρου για μία συνάρτηση.

<u>Ο Απασφαλματωτής (Debugger)</u>

Είναι μία εφαρμογή του ΤΒ η οποία εμφανίζεται σε μορφή παραθύρου και παρουσιάζει την αλληλουχία των εντολών σε ένα σενάριο αντικειμένου, ενώ το σενάριο εκτελείται.

Ο χρήστης μπορεί να χρησιμοποιήσει τον Debugger για να:

- δει τις τιμές που παίρνουν οι μεταβλητές του.
- παρακολουθήσει το ιστορικό και την σειρά των κλήσεων (calls) των handlers.
- παρακολουθήσει την εκτέλεση του σεναρίου χρησιμοποιώντας σημεία ελέγχου (breakpoints) σαν σημαίες (flags) στο σενάριο.

Α.4 ΣΤΟΙΧΕΙΑ ΤΗΣ ΓΛΩΣΣΑΣ OPENSCRIPT

Α.4.1 Έτοιμα Μηνύματα

Τα πιο συχνά έτοιμα μηνύματα για το οποία ο χρήστης συνήθως γράφει κώδικα είναι:

- ButtonClick
- enterObject και leaveObject
- menuItemSelected

- Idle

Μπορούν να σταλούν μηνύματα σε κάποια αντικείμενα χρησιμοποιώντας την εντολή Send (το μήνυμα στέλνεται χωρίς να ενεργοποιήσει ο χρήστης κάποιο συμβάν, π.χ.: send buttonclick to button id 0), ή την εντολή forward (η οποία προωθεί το μήνυμα προς τα πάνω στην ιεραρχία).

Παράδειγμα

1.

```
to handle buttonclick
        beep 1
        -- ακούγεται ένα μπιπ μία φορά από το μεγαφωνάκι του υπολογιστή
end
```

2.

```
to handle enterBook
-- με το που εισέρχεται στο βιβλίο...
        send Reader
        -- μπαίνει στο επίπεδο Reader Level
        forward
        -- προωθεί το μήνυμα προς τα πάνω στην ιεραρχία
end
```

3.

```
to handle menuItemSelected item
-- αν επιλεχθεί κάποιο στοιχείο του μενού...
        if item is «Options»
        -- αν αυτό που επιλέχτηκε είναι το «Options»
                send options
                -- καλεί την υπορουτίνα options
        end
        forward
end
```

4.

```
to handle Idle
-- όσο δεν γίνεται τίποτα
        request «Wait input»
        -- εμφανίζει μήνυμα «Wait input»
        forward
end
```

Α.4.2 Μηνύματα Φτιαγμένα από τον χρήστη (procedures)

Ενα μήνυμα χρήστη μπορεί να οριστεί σαν το μήνυμα που δεν είναι προκαθορισμένο από το ΤΒ (π.χ. ΒuttonClick) και το οποίο όταν «στέλνεται» εκτελεί κάποια συγκεκριμένη εργασία που έχει καθορίσει ο χρήστης.

Για παράδειγμα μπορεί να δημιουργηθούν μήνυματα χρήστη με σκοπό τον κατακερματισμό μεγάλων σεναρίων ώστε να τα κάνουν πιο κατανοητά. Ο χρήστης που επιθυμεί να στείλει κάποιο δικό του μήνυμα (να καλέσει δηλαδή κάποια δική του υπορρουτίνα) πρέπει να γράψει κάποιον handler για αυτό το μήνυμα, αλλιώς το ΤΒ θα εμφανίσει μήνυμα λάθους (Execution Suspended Message). π.χ.:

to handle ButtonClick

end *send flash*

to handle flash

-- έτσι ορίζεται η υπορρουτίνα flash

hide self

-- κάνει αόρατο τον εαυτό του (το Button που δέχτηκε το συμβάν)

pause 50 ticks

-- κάνει παύση 50 ticks (1 tick = 1/100 του δευτερολέπτου)

show self

-- κάνει ορατό τον εαυτό του

end

Α.4.3 Αναφορά σε ιδιότητες αντικειμένων

Όπως έχουμε δει, μία ιδιότητα αντικειμένου περιγράφει και ορίζει την εμφάνιση και την συμπεριφορά του αντικειμένου μέσα στο book. Ο χρήστης έχει τη δυνατότητα να διαβάσει την τιμή της κάθε ιδιότητας και να αλλάξει την τιμή αρκετών ιδιοτήτων με τη βοήθεια της γλώσσας OpenScript [7].

Εντολές που μπορούν να χρησιμοποιηθούν από τον χρήστη για να διαβάσει ή να θέσει κάποιες ιδιότητες είναι:

- Get: Διαβάζει την τιμή της ιδιότητας και την τιμή που βρίσκει την «τοποθετεί» στην ειδική μεταβλητή It.
- Ο τελεστής ανάθεσης (=), set, put: Θέτει μία τιμή σε μία ιδιότητα
- Put: γράφει την τιμή μίας ιδιότητας στον Command Window

Π.χ.:

to handle buttonClick

-- με το που γίνει κλικ στο πλήκτρο του mouse

get position of group «Satelite»

-- διαβάζει τις συντεταγμένες του αντικειμένου και τις αποθηκεύει στο It

item 1 of It = 500

-- Αλλάζει το πρώτο στοιχείο του It (την Χ συντεταγμένη)

put It into position of group «Satelite»

-- Ενημερώνει το νέο ζευγάρι τιμών που περιέχει το It στην ιδιότητα Position

end

Ιδιότητες Συστήματος

Μία ιδιότητα συστήματος ορίζει την εξ ορισμού (default) συμπεριφορά του ίδιου του ΤΒ. Π.χ. αν τεθεί η ιδιότητα sysFontFace στην τιμή Arial, το κείμενο σε κάθε καινούργιο field και button που δημιουργείται θα είναι γραμμένο με γραμματοσειρά Arial.

Οι τέσσερις τύποι που προκύπτουν αν ομαδοποιηθούν οι ιδιότητες συστήματος είναι [7]:

- Γενικές (General)
- Εκκίνησης (Startup)
- Εκτυπωτή (Printer)
- Διεθνείς (International)

Οι ιδιότητες των αντικειμένων υπερκαλύπτουν τις ιδιότητες συστήματος (π.χ. αν η ιδιότητα sysFontFace έχει την τιμή Arial και η ιδιότητα fontFace ενός field την τιμή Hellas Arial τότε το κείμενο θα γραφτεί στο field με γραμματοσειρά Hellas Arial).

Π.χ.

to handle EnterApplication

-- με την είσοδο στην εφαρμογή

SysCursor = 4

-- Αλλάζει το δείκτη του mouse σε κλεψύδρα (General)

StartUpDrawDirect = false

-- Απενεργοποιεί την ιδιότητα εκκίνησης DrawDirect (StartUp)

PrinterTopMargin = 1440

-- Ορίζει πάνω περιθώρια για την σελίδα στην εκτύπωση (Printer)

SysDecimal = «,»

-- Ορίζει το σύμβολο των δεκαδικών στου αριθμούς (International)

forward

end

Τιμές και Ιδιότητες

Ο χρήστης μπορεί να ορίσει τους παρακάτω τύπους τιμών σε ιδιότητες:

- Σταθερές (Constants)
- Literal
- Μεταβλητές (Variables)
- Αλφαριθμητικές (String)
- Λογικές (Logical)

Στο ΤΒ πολλές ιδιότητες παίρνουν σαν τιμή οποιαδήποτε τύπο από τους παραπάνω, όμως υπάρχουν και άλλες οι οποίες παίρνουν μόνο καθορισμένης μορφής τιμές:

Π.χ.:

Σωστό:

- *fillColor of ellipse «x» = blue* Σταθερή

- *fillColor of ellipse «x» = 240,50,100* Literal
- *myColor = blue*
  fillColor of ellipse «x» = myColor Μεταβλητή

Λάθος

- *fillColor of ellipse «x» = «blue»* Αλφαριθμητικό
- *fillColor of ellipse «x» = false* Λογική

Οι ιδιότητες, ακόμη, μπορεί να περιέχουν εκφράσεις. Μία έκφραση είναι μία formula (τύπος - δηλ. ένα σύνολο υπολογισμών που δίνει μία τιμή). Τα στοιχεία μίας έκφρασης συνδέονται συνήθως με τελεστές. Η γλώσσα OpenScript περιέχει τους ακόλουθους τύπους τελεστών [7]:

- αριθμητικούς
- λογικούς
- αλφαριθμητικούς

Π.χ.

Αριθμητικούς
to handle buttonClick
-- με το που γίνει κλικ στο mouse
get position of self
-- διαβάζει την θέση του αντικειμένου του οποίου τρέχει ο handler
It = item 1 of It + 100, item 2 of It + 100
-- η μεταβλητή It (που περιέχει την θέση που πήρε από την get) αυξάνει
-- κατά 100,100 (και η πρώτη και η δεύτερη τιμή του)
end
Λογικούς
to handle hide_show obj
-- υπορουτίνα hide_show με όρισμα obj
visible of obj = not (visible of obj)
-- αλλάζει την ιδιότητα visible του αντικειμένου obj (από true σε false και
-- από false σε true)
end
Αλφαριθμητικούς
to handle Idle
-- όσο δεν γίνεται τίποτα...
text of field «x» = «Time:» && sysTime
-- το πεδίο κειμένου «x» εμφανίζει την ώρα συστήματος
end

A.4.4 Statements και συναρτήσεις

Εντολές

Μία εντολή στη γλώσσα OpenScript είναι ένας όρος που εκτελείται από ένα script ή από το Command Window και αναγκάζει το ΤΒ να εκτελέσει μία ενέργεια. Μία statement περιέχει την εντολή που θέλει ο χρήστης να χρησιμοποιήσει και τις παραμέτρους που απαιτεί η εντολή. Μία παράμετρος

μπορεί να είναι είτε ένα αντικείμενο ή κάποια δεδομένα πάνω στα οποία δρα η εντολή.

Built-In Functions (Ετοιμες Συναρτήσεις)

Μία συνάρτηση είναι μία λειτουργία η οποία υπολογίζει μία τιμή χρησιμοποιώντας κάποιες παραμέτρους. Το ΤΒ περιέχει από δημιουργία του πολλές έτοιμες συναρτήσεις τις οποίες ο χρήστης μπορεί να χρησιμοποιήσει όποτε θέλει ώστε να [7]:

- εκτελέσει αριθμητικούς ή οικονομικούς υπολογισμούς
- πάρει πληροφορίες απ'το σύστημα
- χειριστεί αλφαριθμητικά

Π.χ.:

x = abs(-10)	(εκτελεί υπολογισμούς)
x = resourceList(icon, this book)	(Πληροφορίες από το σύστημα)
x = upperCase(«hello»)	(Χειρισμός αλφαριθμητικού)

Συναρτήσεις Χρήστη

Πέρα από τις έτοιμες συναρτήσεις που παρέχει το ΤΒ ο χρήστης έχει τη δυνατότητα να γράψει τις δικές του συναρτήσεις. Αυτό μπορεί να επιτευχθεί με τη βοήθεια του χειριστή «to get» ο οποίος επιστρέφει κάποια τιμή.

Οι συναρτήσεις χρήστη μπορούν να χρησιμοποιηθούν κυρίως για να:

- εκτελεστούν υπολογισμοί (πράξεις) με περισσότερες από μία εισαχθείσες τιμές.
- δημιουργηθούν συναρτήσεις οι οποίες καλούνται επαναληπτικά από άλλους handlers.

Π.χ.:

```
to handle ButtonClick
-- με το που γίνει κλικ στο mouse
        request CallCost(10)
-- καλείται η συνάρτηση CallCost με όρισμα 10 και το  αποτέλεσμά
-- της εμφανίζεται στον χρήστη σε ένα παράθυρο μηνύματος
end

to get CallCost length
-- ορίζεται η συνάρτηση CallCost με όρισμα το length
        if length < 1
        -- αν η παράμετρος length έχει τιμή μικρότερη της μονάδας
                return «Free Call»
        -- επιστρέφει εκεί που καλέστηκε το αλφαριθμητικό «Free Call»
        else
        -- αλλιώς
                return «The Call Cost $» & length * 0.05
                -- πολλαπλασιάζει τη length επί 0.05 και το επιστρέφει σε ένα
                -- αλφαριθμητικό μαζί με την φράση «The Call Cost $»
        end
```

-- κλείνει το μπλόκ του if
end

Δομή διακλάδωσης και Βρόγχου (Loops)

Μία δομή διακλάδωσης περιέχει συνθήκες και statements οι οποίες εκτελούνται όταν οι συνθήκες γίνουν αληθείς. Οι δομές διακλάδωσης στην γλώσσα OpenScript είναι [7]:

- If/else: εκτελεί ένα σύνολο από statements όταν η ορισμένη συνθήκη είναι true (αλήθεια), και κάποιο άλλο (τμήμα else) όταν είναι false. Το τμήμα else είναι προαιρετικό.
- Conditions/When/else: εκτελεί μία από αρκετές πιθανές statements ανάλογα με την τιμή κάποιας έκφρασης. Το τμήμα else είναι προαιρετικό.

Π.χ.:

1.
if fillColor of ellipse «x» = blue
-- αν το χρώμα γεμίσματος της έλλειψης «x» είναι μπλε τότε...
fillColor of ellipse «x» = red
-- κάνει το χρώμα γεμίσματος της έλλειψης «x» κόκκινο
else
-- αλλιώς
fillColor of ellipse «x» = blue
-- κάνει το χρώμα γεμίσματος της έλλειψης «x» μπλέ
end

2.
Conditions
-- Συνθήκες:
When fillColor of ellipse «x» = blue
-- όταν το χρώμα γεμίσματος της έλλειψης «x» είναι μπλέ...
fillColor of ellipse «x» = red
-- κάνει το χρώμα γεμίσματος κόκκινο
When fillColor of ellipse «x» = red
-- όταν το χρώμα γεμίσματος της έλλειψης «x» είναι κόκκινο...
fillColor of ellipse «x» = white
-- κάνει το χρώμα γεμίσματος άσπρο
When fillColor of ellipse «x» = white
-- όταν το χρώμα γεμίσματος της έλλειψης «x» είναι άσπρο
fillColor of ellipse «x» = blue
-- κάνει το χρώμα γεμίσματος μπλέ
end

Μία δομή βρόγχου περιέχει statements που εκτελούνται μέχρι κάποια ορισμένη συνθήκη να πάρει μία καθορισμένη τιμή. Τα είδη της δομής βρόγχου είναι:

- step: εκτελεί ένα σύνολο statements συνεχώς για ένα καθορισμένο αριθμό φορών
- do/until: εκτελεί ένα σύνολο statements μέχρις ότου η καθορισμένη συνθήκη γίνει true. Το σύνολο των statements εκτελείται τουλάχιστον μία φορά.
- while: εκτελεί ένα σύνολο από statements καθόσον η ορισμένη συνθήκη διατηρείται στην τιμή true. Οι statements υπάρχει περίπτωση να μην εκτελεστούν καθόλου (π.χ. όταν η συνθήκη είναι από την αρχή false).

Π.χ.:

1.
step i from 1 to 100
-- για i=1 έως το 100
move ellipse «x» by 100,100
-- μετακινεί την έλλειψη «x» κατά 100 οριζόντια και κατά 100 κάθετα
end

2.
do
move ellipse «x» by 100,100
-- μετακινεί την έλλειψη «x» κατά 100 οριζόντια και κατά 100 κάθετα
until item 1 of position of ellipse «x» > 1000
-- έως ότου η x συντεταγμένη της θέσης της έλλειψης «x» γίνει μεγαλύτερη του 1000

3.
while item 1 of position of ellipse «x» < 1000
-- όσο η x συντεταγμένη της θέσης της έλλειψης «x» είναι μικρότερη του 1000
move ellipse «x» by 100,100
-- μετακίνησε την έλλειψη «x» κατά 100 οριζόντια και κατά 100 κάθετα
end

Α.4.5 Μεταβλητές

Τοπικές μεταβλητές

Η γλώσσα OpenScript υποστηρίζει τοπικές και καθολικές μεταβλητές. Τοπικές είναι οι μεταβλητές εκείνες που η τιμή τους είναι γνωστή μόνο στον handler του αντικειμένου που χρησιμοποιούνται, ενώ καθολικές εκείνες που είναι γνωστές στον κάθε handler του κάθε αντικειμένου στο book.

Μία τοπική μεταβλητή κρατά την τιμή της μόνο όσο ο handler που την περιέχει εκτελείται. Για να χρησιμοποιηθεί μία τοπική μεταβλητή δεν είναι αναγκαία η δήλωση του τύπου της. Απλά, την στιγμή που παίρνει μία τιμή μέσα στον κώδικα δηλώνεται αυτόματα και ο τύπος της [7]:

Π.χ.:

1.
to handle buttonClick

```
step var from 1 to 10
-- εδώ τοπική μεταβλητή είναι η var που λειτουργεί σαν μετρητής
        move ellipse «x» by 100,100
end
end

2.
to handle enterApplication
ask «What is your name ?»
-- ζητά από τον χρήστη να γράψει το όνομά του
var = It
-- η μεταβλητή var παίρνει την τιμή που έχει αποθηκευτεί στην It
-- εδώ τοπική μεταβλητή είναι η var με μορφή αλφαριθμητικό
Forward
end
```

System (Καθολικές) Μεταβλητές

Μία μεταβλητή system κρατάει την τιμή της για όλη την διάρκεια του προγράμματος, και μπορεί να χρησιμοποιηθεί σε περισσότερους από έναν handler.

Για να χρησιμοποιηθεί μία μεταβλητή system πρέπει να δηλωθεί από πριν στον handler που πρόκειται να χρησιμοποιηθεί.

Η system μεταβλητή δεν διατηρεί την τιμή της ανάμεσα σε διαφορετικές εκδόσεις του TB (κάτι που συμβαίνει για τις ιδιότητες χρήστη).

Π.χ.:

```
to handle enterApplication
system sRate
-- η μεταβλητή sRate είναι καθολική και μπορεί να χρησιμοποιηθεί από
κάθε
-- αντικείμενο και από κάθε handler
sRate = 0.05
forward
end

to handle Calculate length
-- ορίζεται η υπορουτίνα Calculate με όρισμα length
system sRate
request «The Call Cost $» & sRate * length
end
```

Η μεταβλητή It

Η μεταβλητή It όπως είδαμε και παραπάνω αποτελεί μία μεταβλητή προσωρινής αποθήκευσης. Οι εντολές της OpenScript που επιστρέφουν ένα αποτέλεσμα στην It είναι οι get, ask και request, οι οποίες όταν εκτελεστούν «βάζουν» το αποτέλεσμά τους στην It.

Κάθε φορά που το TB εκτελεί μία statement που περιέχει μία από αυτές τις εντολές ενημερώνει την It με μία τιμή. Γι' αυτό το λόγο η It μπορεί να χρησιμοποιηθεί με παρόμοιο τρόπο όπως κάθε άλλη τοπική μεταβλητή. Ωστόσο, για να χρησιμοποιηθεί η τιμή που περιέχει η It και μετέπειτα στην εφαρμογή, καλό είναι να αποθηκευτεί σε μία άλλη μεταβλητή [7].

Π.χ.:

1.

to handle buttonClick
get fillColor of ellipse «x»
-- Αποθηκεύει στην It την ιδιότητα fillColor της έλλειψης «x»
fillC = It
-- τώρα η μεταβλητή fillC περιέχει την τιμή της ιδιότητας fillColor
-- της έλλειψης «x»
get strokeColor of ellipse «x»
-- Αποθηκεύει την τιμή της ιδιότητας strokeColor της έλλειψης «x»
-- μεταβλητή It
strokeC= It
-- τώρα η μεταβλητή strokeC περιέχει την τιμή της ιδιότητας strokeColor
-- της έλλειψης «x»
end

2.

to handle exit
-- με το που στείλει ο χρήστης το μήνυμα exit (μήνυμα που ισοδυναμεί με την
-- επιλογή Exit του μενού File
request «Are you sure ?» with «Yes» or «No»
-- Εμφανίζει μήνυμα «Are you sure?» και με δύο πλήκτρα «Yes» και «No»
if It = «Yes»
-- Αν ο χρήστης επιλέξει το πλήκτρο Yes τότε η It παίρνει την τιμή «Yes»
    forward
    -- τότε μόνο προωθεί το μήνυμα προς το TB για να φύγει ο χρήστης
    -- από την εφαρμογή
end
end

<u>Arrays (Πίνακες)</u>

Ενας πίνακας είναι μία τοπική ή καθολική μεταβλητή που ορίζεται σαν λίστα (π.χ. διαστάσεων nx1 ή 1xn) ή σαν πίνακας (nxm). Ο χρήστης μπορεί να δημιουργήσει πίνακες σταθερής διάστασης (από 1 - 16) ή και πίνακες που η διάστασή τους αλλάζει δυναμικά, ενώ υπάρχει η δυνατότητα να προστεθούν στοιχεία (χωρίς να χάνονται οι προηγούμενες τιμές τους).

Είτε οι πίνακες είναι τοπικές είτε καθολικές μεταβλητές πρέπει να δηλωθούν πριν να χρησιμοποιηθούν (με τους όρους local ή system).

Π.χ.:

Local array grades[1][2]
-- ορίζει τον τοπικό πίνακα grades με διαστάσεις 1x2
grades [1][2] = 30
grades [2][3] = 40

Σταθερής διάστασης πίνακες μπορούν να δημιουργηθούν αν στο τμήμα δηλώσεων καθοριστούν και οι διαστάσεις τους, ενώ για να δημιουργηθούν δυναμικοί πίνακες πρέπει να οριστούν οι πίνακες με κενή διάσταση (κενές αγκύλες) [7].

Μπορεί να χρησιμοποιηθεί η συνάρτηση dimensions() για να βρεθεί το μέγεθος ενός δυναμικού πίνακα:

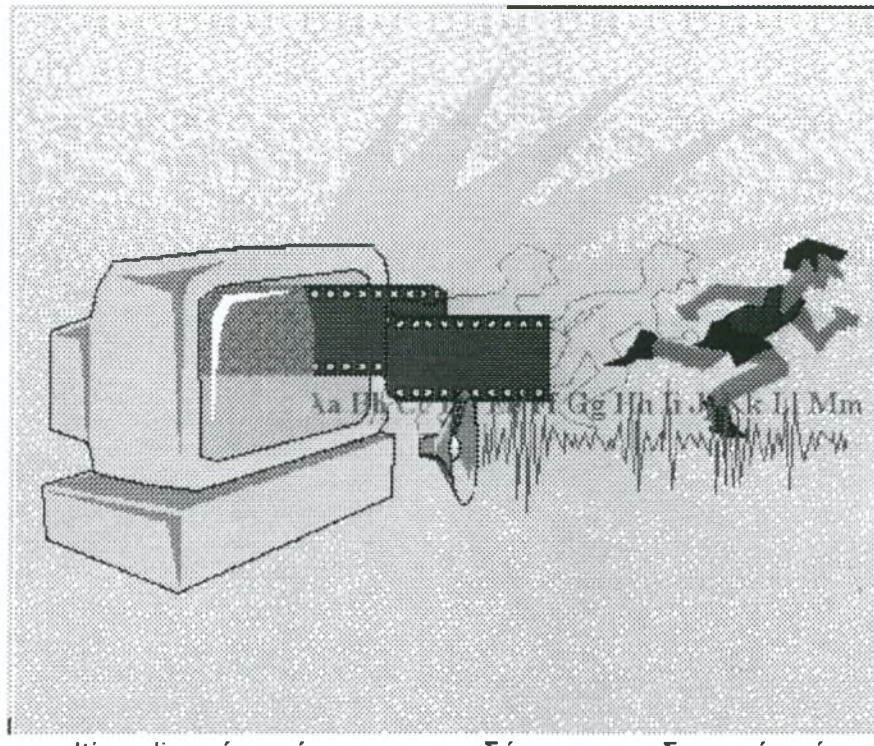
Π.χ.:

```
to handle fixedArray
-- ορίζει την υπορουτίνα fixedArray
        local array questions[10]
        -- δηλώνει τον τοπικό πίνακα questions με διάσταση 10
        step i from 1 to 10
        -- για i = 1 έως το 10
                questions[i] = i
                -- γεμίζει τον πίνακα questions
        end
end

to handle dynamicArray
-- ορίζει την υπορουτίνα dynamicArray
        system array answers[]
        -- δηλώνει τον καθολικό πίνακα answers[] σαν δυναμικό
        dim = dimensions(answers)
        -- βρίσκει την τρέχουσα διάσταση του πίνακα answers
        answers[dim + 1] = true
        -- αυξάνει την διάσταση του πίνακα answers και αποθηκεύει
ακόμη μία τιμή
end
```

Α.5 Ο ΧΕΙΡΙΣΜΟΣ ΤΩΝ MULTIMEDIA ΑΠΟ ΤΟ TB

Α.5.1 Γενικά



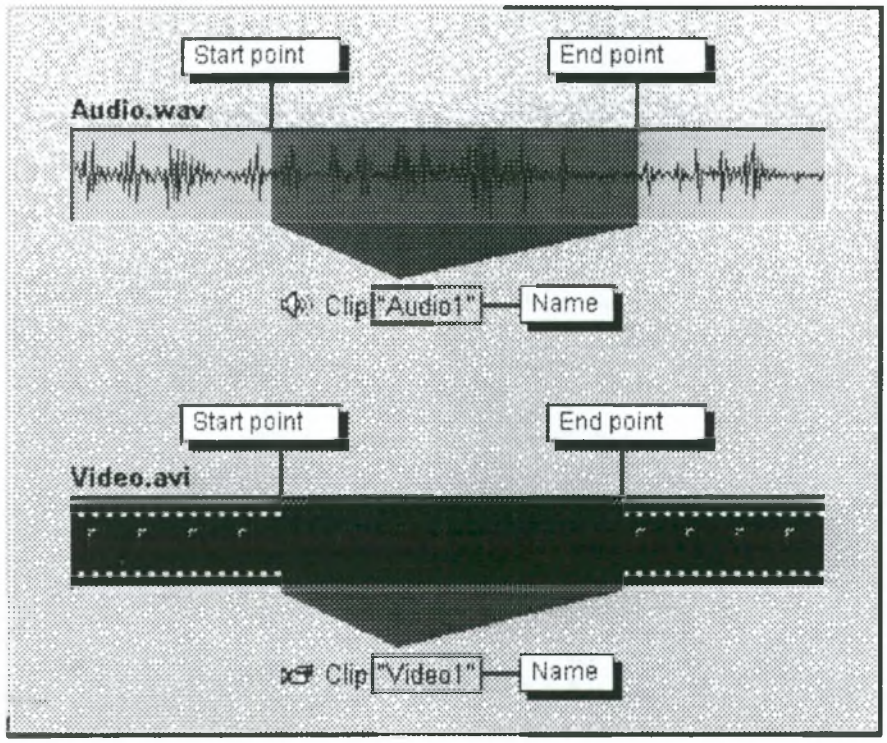
Τα multimedia είναι ένας οποιοσδήποτε συνδυασμός ήχου, video, animation, κειμένου και γραφικών που μπορούν να «παιχθούν» από έναν υπολογιστή [5].

Μπορούμε να προσθέσουμε ήχο σε μια εφαρμογή που δημιουργούμε με το Toolbook , αφού δημιουργήσουμε πρώτα ένα ηχητικό clip (μια αναφορά σε ένα τμήμα ήχου) και μετά γράψουμε ένα σενάριο για να το παίξουμε.

Επίσης, μπορούμε να προσθέσουμε video σε μία εφαρμογή του Toolbook, αφού δημιουργήσουμε ένα video clip, δημιουργήσουμε μία σκηνή (stage) η οποία ορίζει την περιοχή της σελίδας στην οποία θα παιχθεί το clip και γράψουμε ένα σενάριο για να παίξουμε το clip αυτό.

Α.5.2 Clips

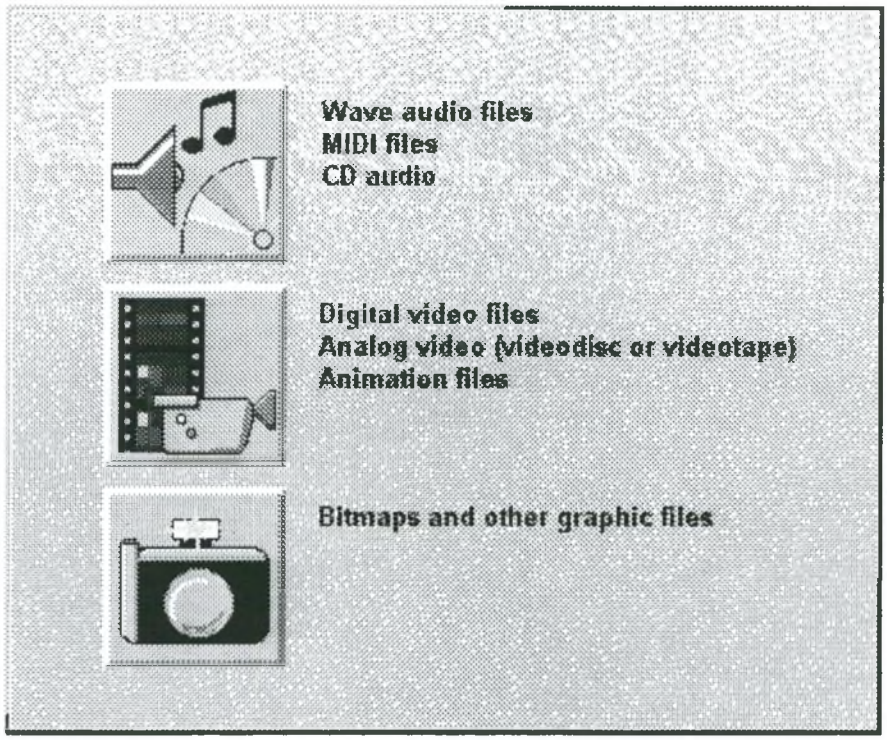
Clip είναι μία αναφορά σε ένα τμήμα οποιασδήποτε multimedia πηγής ή αρχείου και περιλαμβάνει ένα σημείο έναρξης, ένα σημείο τέλους και ένα όνομα. Το Toolbook αποθηκεύει μόνο την αναφορά και όχι ολόκληρο το αρχείο. Μπορούμε να αποθηκεύσουμε ομάδες από clips στις βιβλιοθήκες clips (clip libraries - .cpf files). Η πρόσβαση στα clips γίνεται μέσω του Clip Manager και η δημιουργία νέων clips μέσω του Clip Editor [6].



A.5.3 Multimedia πηγές

Μπορούμε να δημιουργήσουμε clips από τις παρακάτω multimedia πηγές (multimedia αρχεία όπως .WAV files ή συσκευές όπως videodisk player) [6].

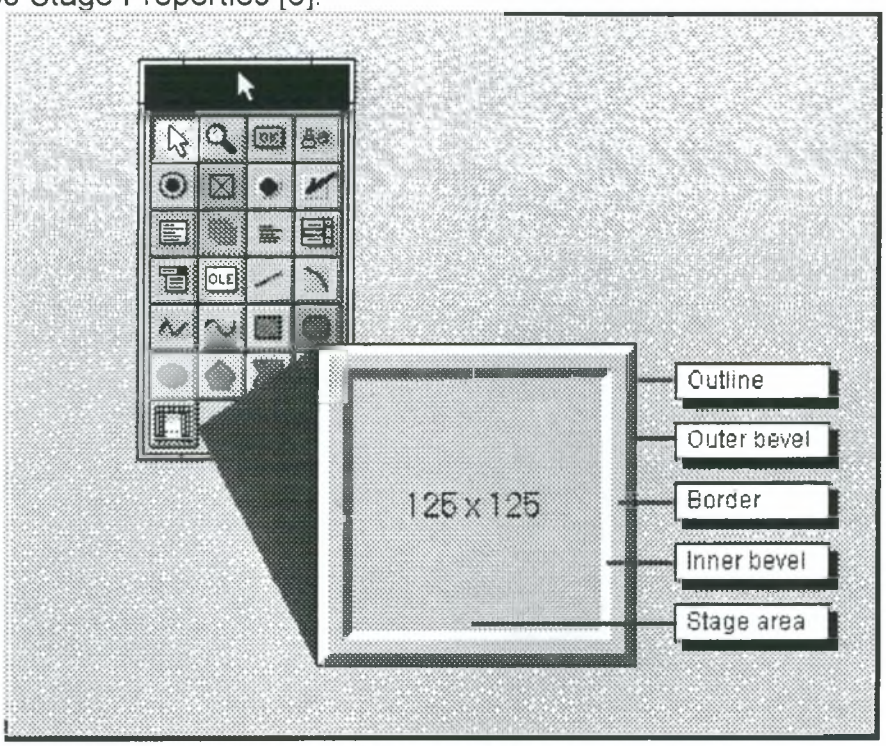
- Wave Audio Files: Ήχος που έχει ηχογραφηθεί σε ψηφιακή μορφή και αποθηκευτεί σε αρχείο με επέκταση .WAV.
- MIDI Files: Musical Instrument Digital Interface Files. Ένα αρχείο MIDI (επέκταση .MID) είναι μια ακολουθία εντολών που απευθύνονται σε ένα synthesizer ή μια κάρτα ήχου (δεν είναι καταγραμμένος ήχος).
- CD Audio: Ήχος που είναι αποθηκευμένος με τη μορφή καναλιού εγγραφής σε Compact Disk και αναπαράγεται από το CD.
- Digital Video Files: Video που έχει μετατραπεί σε ψηφιακή μορφή και έχει αποθηκευτεί σε αρχείο (επέκταση .AVI).
- Αναλογικό video (videoδίσκος ή videoταινία): Video που αναπαράγεται κατευθείαν από μία αναλογική πηγή video όπως συσκευή αναπαραγωγής videoταινίας ή videoδίσκου ή video camera.
- Αρχεία animation: Animations που έχουν δημιουργηθεί με τη χρήση προϊόντων όπως τα Autodesk Animator και Macromedia Director και είναι αποθηκευμένα σαν αρχεία με επεκτάσεις .FLI, .FLC και .MMM. Από τα αρχεία αυτά μπορούν να δημιουργηθούν clips.
- Bitmaps και άλλα αρχεία γραφικών: Εικόνα που συντίθεται από pixels. Τα bitmaps τυπικά χρησιμοποιούνται για να αναπαράγουν εικόνες (όπως φωτογραφίες) που συμπεριλαμβάνουν λεπτομέρειες, σκίαση και χρώμα (επεκτάσεις .BMP, .DIB, .RLE).



A.5.4 Stages (Σκηνές)

Οι σκηνές είναι ένα αντικείμενο του Toolbook μέσα στο οποίο μπορεί να παιχθεί ένα clip animation, ή ψηφιακού ή αναλογικού video ή ακόμη και μια ακίνητη εικόνα. Η σκηνή ορίζει την περιοχή της σελίδας μέσα στην οποία παίζεται το clip.

Μπορούμε να σχεδιάσουμε μία σκηνή με το εργαλείο stage της παλλέτας εργαλείων και να θέσουμε τις ιδιότητές της χρησιμοποιώντας το πλαίσιο διαλόγου Stage Properties [6].



A.5.5 Multimedia εντολές

Μπορούμε να χρησιμοποιήσουμε τις ακόλουθες εντολές για να παίξουμε και να χειριστούμε τα clips [7]:

MMclose: Κλείνει ένα clip. Τη χρησιμοποιούμε για να κλείσουμε ένα clip που έχει ανοίξει με την mmopen.

MMcue: Προτρέπει ένα clip που έχει ανοίξει με την mmopen να παίξει από την αρχή του.

MMopen: Ανοίγει ένα clip, επιβεβαιώνει ότι η αναφορά του είναι έγκυρη και το προετοιμάζει για παίξιμο. Πρέπει να ανοίξουμε ένα clip προτού χρησιμοποιήσουμε οποιαδήποτε multimedia εντολή (εκτός από την mmplay).

MMpause: Κάνει παύση σε ένα ανοιχτό clip στην τρέχουσα θέση του ενώ διατηρεί τον έλεγχο του καναλιού της multimedia συσκευής. Αν εκτελεστεί η mmplay μετά την mmpause τότε το clip παίζει από την τρέχουσα θέση του.

MMplay: Παίζει ένα clip. Εάν το clip είναι μια ακίνητη εικόνα π.χ. bitmap τότε εμφανίζει την εικόνα. Αν το clip δεν είναι ανοιχτό τότε το ανοίγει προτού το παίξει, το κλείνει μετά το παίξιμο και αν το clip παίξει μέχρι το τέλος του χωρίς να σταματήσει, το «ξανατυλίγει» στην αρχή του.

MMrewind: Σταματάει το clip αν αυτό παίζει και θέτει την τρέχουσα θέση του στην αρχή του.

MMseek: Ψάχνει για μια συγκεκριμένη θέση μέσα σε ένα ανοιχτό clip. Εξ ορισμού οι λειτουργίες αναζήτησης έχουν ως αρχικό σημείο το σημείο έναρξης του clip. Αν χρησιμοποιηθεί η παράμετρος from end, τότε οι λειτουργίες αναζήτησης λαμβάνουν χώρα προς τα πίσω από το σημείο τέλους του clip.

MMshow: Απεικονίζει ένα απλό frame ενός ανοιχτού οπτικού clip.

MMstep: Ξεκινάει την αναζήτηση προς τα εμπρός από την τρέχουσα θέση ενός clip για ένα καθορισμένο ποσό (χρόνου).

MMstop: Σταματάει το παίξιμο ενός clip διατηρώντας την τρέχουσα θέση του.

```
to handle enterPage
   mmOpen clip "Satellite"
   mmShow clip "Satellite"
   forward
end

to handle buttonClick
   mmPlay clip "Satellite" in stage "XSAT" wait
   mmClose clip "Satellite"
   mmSeek clip "Space" to 500
   mmPlay clip "Space"
   mmRewind clip "Space"
end
```

Α.5.6 Έλεγχος multimedia συσκευών με τη χρήση MCI εντολών

Στις περισσότερες περιπτώσεις οι multimedia εντολές όπως mmplay, mmstop και mmstep είναι οι μόνες που χρειαζόμαστε για να ελέγξουμε τα clips. Παρόλα αυτά, με τη χρήση της συνάρτησης της συνάρτησης callMCI(), μπορούμε να στείλουμε εντολές MCI (Media Control Interface) οι οποίες δεν έχουν αντίστοιχες εντολές Openscript, για να ελέγξουμε κατευθείαν συσκευές multimedia (π.χ. CD-ROM) και να εκμεταλλευτούμε τις ιδιαίτερες δυνατότητες κάθε συσκευής. Π.χ. μπορούμε να στείλουμε εντολές MCI για να ηχογραφήσουμε αρχεία .WAV ή για να φορτώσουμε και να ξεφορτώσουμε CD's από το CD-ROM drive.

Οι εντολές MCI αποτελούν ένα στάνταρντ σετ εντολών των Windows που χρησιμοποιούνται για τον άμεσο έλεγχο συσκευών multimedia.

Για να στείλουμε μια εντολή MCI, χρησιμοποιούμε τη συνάρτηση callMCI() σε ένα script, περνώντας την εντολή MCI που θέλουμε να εκτελέσουμε, σαν string.

Π.χ.

```
to handle buttonClick
        get callMCI("open c:\windows\chimes.wav alias beep wait")
        get callMCI("play beep")
        get callMCI("close beep")
end
```

Ο έλεγχος των multimedia αρχείων και συσκευών με τη χρήση της συνάρτησης callMCI() είναι ίδιος, στις περισσότερες περιπτώσεις, με τη χρήση των mmopen, mmplay και άλλων Openscript multimedia εντολών. Μπορούμε να ανοίξουμε και να παίξουμε μια συσκευή χρησιμοποιώντας τις MCI εντολές cue και seek και να εξετάσουμε την κατάσταση ενός αρχείου ή μιας συσκευής. Μπορούμε, επίσης, με την callMCI() να ειδοποιήσουμε ένα αντικείμενο όταν

τερματιστεί η εκτέλεση της εντολής. Εντούτοις, το σύνολο των εντολών MCI είναι ελαφρώς διαφορετικό από τις Openscript multimedia εντολές στα παρακάτω [7]:

- Ο τύπος της συσκευής πρέπει να αναφέρεται ρητώς κάθε φορά που στέλνεται μια εντολή MCI χρησιμοποιώντας τις δεσμευμένες λέξεις όπως cdAudio, waveAudio ή videodisk. (Για την απλοποίηση αυτής της διαδικασίας μπορύμε να ορίσουμε ένα «ψευδώνυμο» (alias) για τον τύπο του μέσου ή το όνομα του αρχείου).

- Κάθε συσκευή υποστηρίζει μοναδικές, συγκεκριμένες για τον τύπο αυτό εντολές MCI. Για παράδειγμα, μπορούμε να στείλουμε μια εντολή spin σε ένα videodisk player για να ξεκινήσει η περιστροφή του videoδίσκου. Η εντολή αυτή, όμως, δεν έχει εφαρμογή σε άλλη συσκευή. Π.χ. οι MCI εντολές για τον έλεγχο συσκευών waveform audio είναι οι: capability, close, cue, info, pause, play, record, resume, save, seek, set, status, stop ενώ οι εντολές MCI για τον έλεγχο συσκευών ψηφιακού video είναι οι: capability, close, cue, info, pause, play, put, realize, resume, seek, set, setAudio, setVideo, signal, status, step, stop, update, where, window.

- Η ίδια MCI εντολή για διαφορετικές συσκευές παίρνει διαφορετικές παραμέτρους. Πιο συγκεκριμένα, η εντολή set (για να θέτουμε τις ιδιότητες σε μια συσκευή) απαιτεί διαφορετικές τιμές για κάθε συσκευή.

- Τα οπτικά μέσα που παίζονται από τις εντολές MCI, δεν παίζουν σε σκηνή (stage), οπότε πρέπει να εκτελεστούν MCI εντολές για να ελέγξουμε το μέγεθος και τη θέση του παραθύρου μέσα στο οποίο απεικονίζονται video και animation.

- Όταν ένα οπτικό μέσο που παίζεται με εντολή MCI σταματήσει, το παράθυρο στο οποίο απεικονίζεται, δείχνει το τρέχον frame.

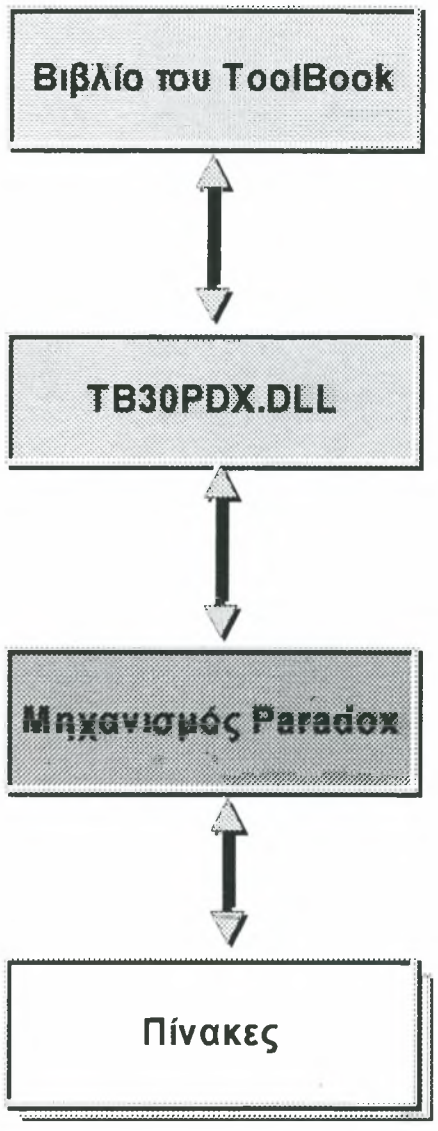
Α.6 Ο ΧΕΙΡΙΣΜΟΣ ΤΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟ ΤΒ

Α.6.1 Χρήση του Μηχανισμού της Paradox

Ο Μηχανισμός της Paradox είναι ένα σύνολο από συναρτήσεις που εκτελούν λειτουργίες χειρισμού δεδομένων π.χ. τη δημιουργία, το άνοιγμα και το κλείσιμο πινάκων, την ανάγνωση και το γράψιμο δεδομένων σ' αυτούς.

Η βιβλιοθήκη συναρτήσεων TB30PDX.DLL του ToolBook περιέχει συναρτήσεις που κάνουν δυνατό τον χειρισμό του Μηχανισμού της Paradox. Ο χρήστης έχει τη δυνατότητα να χειριστεί πίνακες που δημιουργήθηκαν από κάποιον database manager της Paradox ή να δημιουργήσει δικούς του.

Παρακάτω φαίνεται η σχέση και η σύνδεση ενός Βιβλίου του ToolBook με τον μηχανισμό της Paradox (βλέπε σχήμα)



Αποθήκευση Δεδομένων

Τα δεδομένα αποθηκεύονται σε πίνακες. Ένας πίνακας αντιστοιχεί σε ένα αρχείο. Κάθε πίνακας περιέχει μία ή περισσότερες εγγραφές (records) και κάθε εγγραφή περιέχει ένα η περισσότερα πεδία (fields). Ένα πεδίο μπορεί να περιέχει αριθμητικά ή αλφαριθμητικά δεδομένα.

Δεδομένα τύπου BLOB (Binary Large OBjects)

Το BLOB είναι είναι ένα block από δεδομένα δυαδικής μορφής (όπως ένα γραφικό, ένα βίντεο κλιπ, ή ένας ήχος) ή ένα block κειμένου που ξεπερνά τα 255 bytes. Τα στοιχεία τύπου BLOB αποθηκεύονται σε ξεχωριστό αρχείο. Στον πίνακα της βάσης αποθηκεύεται μόνο ένας δείκτης που δείχνει στο αρχείο BLOB.

Indexes (ευρετήρια)

Οι πίνακες μπορούν να περιέχουν indexes που αποτελούνται από οποιαδήποτε πεδίο ή συνδυασμό πεδίων. Εξαιρούνται τα πεδία τύπου BLOB. Μπορούν να υπάρχουν πολλά indexes για ένα πίνακα αλλά μπορεί να χρησιμοποιηθεί μόνο ένα κάθε φορά.

Μεταφορά Δεδομένων

Στο Λογικό Επίπεδο τα δεδομένα μετακινούνται προς και από τον πίνακα με την βοήθεια ενός βοηθητικού buffer. Οι συναρτήσεις της βιβλιοθήκης TB30PDX.DLL οι οποίες διαβάζουν και γράφουν δεδομένα, μετακινούν δεδομένα μέσα και έξω από αυτόν τον buffer, και ποτέ άμεσα στον πίνακα. Για να γραφεί μία τιμή σε ένα πεδίο, πρέπει να γίνει η μετάβαση σε κάποια εγγραφή, να τεθεί η τιμή για το πεδίο στην συγκεκριμένη εγγραφή, και τέλος να γίνει η ενημέρωση των αλλαγών που έγιναν στον πίνακα της Βάσης Δεδομένων.

Στο Φυσικό Επίπεδο τα δεδομένα μετακινούνται με την μορφή blocks με την βοήθεια του swap buffer (cache). Ο μηχανισμός φροντίζει ώστε όλες οι φυσικές μετακινήσεις δεδομένων να γίνονται αυτόματα, αν και παρέχεται η δυνατότητα στον χρήστη να χειριστεί αυτές τις λειτουργίες.

Α.6.2 Κανόνες ονομασίας

Πίνακες

Ένας πίνακας της Βάσης Δεδομένων είναι ένα αρχείο του DOS με επέκταση .DB. Αν κάποια αρχεία indexes και BLOB είναι συσχετισμένα με τον πίνακα, τότε τα αρχεία αυτά θα έχουν το ίδιο όνομα με το αρχείο του πίνακα αλλά με διαφορετική επέκταση. Πιο συγκεκριμένα τα BLOB αρχεία έχουν την επέκταση .MB ενώ τα αρχεία Indexes πρωτεύοντος κλειδιού έχουν την επέκταση .PX. Δευτερεύοντα Indexes έχουν την επέκταση .Xnn ή Ynn όπου ο αριθμός «nn» καθορίζεται από τον Μηχανισμό.

Πεδία

Το όνομα των πεδίων έχει μήκος έως 25 χαρακτήρες. Μπορεί να περιέχει κενό αλλά δεν μπορεί να αρχίζει με κενό. Μπορεί να περιέχει οποιοδήποτε εκτυπώσιμο χαρακτήρα εκτός από το διπλό εισαγωγικό ("), τις τετράγωνες αγκύλες ([ ]), τις κανονικές αγκύλες ({ }), το σύμβολο (#), αριστερές ή δεξιές παρενθέσεις ( ), ή συνδυασμός των παραπάνω. Τέλος δεν επιτρέπεται να υπάρχουν στον ίδιο πίνακα πεδία με το ίδιο όνομα.

Α.6.3 Τύποι Δεδομένων

Υπάρχουν 5 τύποι δεδομένων:

- **Αλφαριθμητικός (A):**
  Όλοι οι χαρακτήρες ASCII εκτός του χαρακτήρα null (ASCII 0). Μέγιστο μήκος 255 χαρακτήρες. Κάθε χαρακτήρας απαιτεί χώρο ενός byte.

- **Αριθμητικός (N):**
  Έχει 15 σημαντικά ψηφία συμπεριλαμβανομένης και της υποδιαστολής. Α ριθμοί με περισσότερα από 15 σημαντικά ψηφία στρογγυλοποιούνται και αποθηκεύονται σε επιστημονική μορφή. Οι πραγματικοί αριθμοί κυμαίνονται από το $\pm 10^{-307}$ έως το $\pm 10^{308}$ και απαιτούν 8 bytes για κάθε πεδίο.

- **Νομισματικός ($):**
  Έχει 15 σημαντικά ψηφία συμπεριλαμβανομένης και της υποδιαστολής. Οι πραγματικοί αριθμοί κυμμαίνονται από το $\pm 10^{-307}$ έως το $\pm 10^{308}$ και απαιτούν 8 bytes για κάθε πεδίο. Η μορφή τους συμφωνεί με την νομισματική μορφή που έχει επιλεγεί από το Control Panel των Windows.

- **Μικρός ακέραιος (S):**
  Προσημασμένος ακέραιος με τιμές ανάμεσα στο -32,767 και το 32,767. Απαιτεί για την αποθήκευσή του 2 bytes.

- **Ημερομηνία (D):**
  Μπορεί να δεχτεί σαν τιμή οποιαδήποτε ημερομηνία μεταξύ της 1^{ης} Ιανουαρίου του 1 μ.Χ. έως της 31^{ης} Δεκεμβρίου 9999. Η Paradox αποθηκεύει τα δεδομένα ως έναν ακέραιο 4 bytes (αριθμό που είναι ίσος με τον αριθμό των ημερών που πέρασαν από την 1/1/1 μ.Χ.).

Και 3 τύποι BLOB:

- **Σχολίων (M):**
  Κείμενο χωρίς μορφοποίηση με μέγεθος μεγαλύτερο το 255 χαρακτήρων.

- **Δυαδικής Μορφής (B):**
  Μη δομημένα δυαδικής μορφής δεδομένα (όπως patterns και ήχος).

- **Γραφικό (G):**
  Εικόνες των Windows με επέκταση .DIB.

Α.6.4 Αρχικοποίηση του Μηχανισμού της Paradox

Πριν χρησιμοποιηθεί ο Μηχανισμός της Paradox, πρέπει να αρχικοποιηθεί (initialize) με την κλήση της συνάρτησης *initializePX()*. Αυτή η συνάρτηση θέτει τις ακόλουθες παραμέτρους σε προκαθορισμένες (πρωταρχικές) τιμές.

Παράμετρος	Περιγραφή	Εύρος	Τιμή
Μέγεθος του Swap buffer	Διαθέσιμο ποσό της μνήμης για προσωρινή αποθήκευση των δεδομένων μέσα και έξω από την μνήμη. Κάθε πίνακας απαιτεί 4 Kb.	(8K - 256K)	32
Μέγιστος αριθμός αρχείων	Ο μέγιστος αριθμός των αρχείων που μπορεί να χειριστεί ο Μηχανισμός.	(3-255)	10
Μέγιστος αριθμός πινάκων	Μέγιστος αριθμός πινάκων που μπορεί να είναι συγχρόνως ανοιχτοί.	(1-64)	5
Τάξη Ταξινόμησης	Ο τύπος της ταξινόμησης που χρησιμοποιείται.	ASCII	

Α.6.5 Επεξεργασία Πινάκων

Η δημουργία ενός πίνακα γίνεται με τη χρήση της συνάρτησης createPXTable() η οποία δέχεται σαν ορίσματα το όνομα του πίνακα, μία λίστα με τα ονόματα των πεδίων (τιμή από 1-255) και μία λίστα με τους αντίστοιχους τύπους και μεγέθη των πεδίων.

Αν ένας πίνακας με το ίδιο όνομα υπάρχει στον ίδιο κατάλογο και δεν είναι ανοιχτός, τότε ο πίνακας (και όλα τα σχετιζόμενα μ'αυτόν αρχεία BLOB και Index) διαγράφεται δίχως προειδοποίηση. Πριν δημιουργηθεί ένας πίνακας, πρέπει να χρησιμοποιηθεί η συνάρτηση doesPXTableExist() για να ελεγχθεί η ύπαρξη πίνακα με το ίδιο όνομα.

Εξ' ορισμού, ο Μηχανισμός δημιουργεί έναν πίνακα μορφής Paradox 4.0. Για τη δημιουργία ενός πίνακα της Paradox έκδοσης 3.5, πρέπει να κληθεί πρώτα η συνάρτηση setPXTableCreateMode() με κατάλληλα ορίσματα. Όλοι οι επόμενοι πίνακες που δημιουργούνται διατηρούν την συγκεκριμένη μορφή.

Α.6.6 Μετάβαση σε Εγγραφές

Οι συναρτήσεις για μετάβαση σε συγκεκριμένες εγγραφές είναι:

gotoPXRecord()	Μετάβαση σε εγγραφή που καθορίζει κάποιος αριθμός. Αυτή η μέθοδος δεν συνίσταται όταν οι πίνακες δουλεύονται σε δίκτυο, επειδή μπορεί να προστεθούν ή να διαγραφούν εγγραφές εν αγνοία του χρήστη και άρα να αλλάξουν οι αριθμοί των εγγραφών).
FirstPXRecord()	Μετάβαση στην πρώτη εγγραφή πίνακα ή Index.
LastPXRecord()	Μετάβαση στην τελευταία εγγραφή πίνακα ή

	Index.
NextPXRecord()	Μετάβαση στην επόμενη εγγραφή σε πίνακα ή Index.
PreviousPXRecord()	Μετάβαση στην προηγούμενη εγγραφή πίνακα ή Index

Το αποτέλεσμα από την χρήση αυτών των συναρτήσεων εξαρτάται από το αν ο πίνακας έχει ένα Index ή όχι και αν αυτό το Index χρησιμοποιείται κατά την μετάβαση.

Αν ο πίνακας δεν έχει index, οι εγγραφές προσπελάζονται με τη σειρά που έχουν εισαχθεί στον πίνακα. Η πρώτη φυσική εγγραφή έχει τον αριθμό 1, η επόμενη είναι η εγγραφή 2, κ.ο.κ.

Αν έχει οριστεί ένα Index κατά το άνοιγμα του πίνακα, τότε οι εγγραφές αναφέρονται στην σειρά που αυτές έχουν στο Index, και η οποία ενδέχεται να μην είναι η ίδια με την φυσικά σειρά στον πίνακα.

Α.6.7. Παροχή πληροφοριών για ένα πεδίο.

Υπάρχουν δύο συναρτήσεις οι οποίες επιστρέφουν πληροφορίες για πεδία:

getPXFieldNames():	επιστρέφει μία λίστα από τα ονόματα των πεδίων που περιέχει ο συγκεκριμένος ανοιχτός πίνακας.
GetPXFieldType()	επιστρέφει τον τύπο του δεδομένου πεδίου στην τρέχουσα εγγραφή.

Α.6.8 Index και αναζήτηση

Ένα Index σε έναν πίνακα λειτουργεί όπως το αλφαβητικό ευρετήριο σε ένα λεξικό. Βοηθάει στην εύρεση πληροφορίας γρήγορα. Όταν χρειάζεται να γίνει αναζήτηση σε μία λέξη σε ένα λεξικό το αλφαβητικό ευρετήριο μειώνει το εύρος αναζήτησης σε λέξεις που αρχίζουν με το ίδιο γράμμα. Με τον ίδιο τρόπο όταν ο Μηχανισμός χρειάζεται να βρει δεδομένα σε ένα πίνακα με index, μεταβαίνει άμεσα στο τμήμα εκείνο του πίνακα που περιέχει την πληροφορία. Αν ο πίνακας δεν έχει index, ο Μηχανισμός αρχίζει την έρευνα από την αρχή και ψάχνει σειριακά μεχρι να βρεί το ζητούμενο.

Όταν ο μηχανισμός ταξινομεί εγγραφές σύμφωνα με κάποιο κριτήριο αυτό γίνεται στην μνήμη του υπολογιστή και όχι στην φυσική εγγραφή στον δίσκο.

Συναρτήσεις Πινάκων

addPXTable()	Αντιγράφει εγγραφές από έναν πίνακα σε άλλο.
ClosePXTable()	Κλείνει έναν πίνακα.
CopyPXTable()	Αντιγράφει μία οικογένεια πινάκων σε μία άλλη.
CreatePXTable()	Δημιουργεί έναν πίνακα
decryptPXTable()	Αποκρυπτογραφεί έναν πίνακα.
DeletePXTable()	Σβύνει μία οικογένεια πινάκων.
DoesPXTableExist()	Ελέγχει αν ένας πίνακας υπάρχει.
EmptyPXTable()	Σβύνει όλες τις εγγραφές από έναν πίνακα.
EncryptPXTable()	Κρυπτογραφεί έναν πίνακα.
GetPXErrorString()	Επιστρέφει το μήνυμα λάθους σε περίπτωση που συμβεί κάποιο λάθος.
GetPXFieldCount()	Επιστρέφει τον αριθμό των πεδίων ενός πίνακα.
GetPXKeyFieldCount()	Επιστρέφει τον αριθμό των πεδίων που είναι κλειδιά.
GetPXMaxTables()	Επιστρέφει τον μέγιστο αριθμό πινάκων που μπορούν να είναι ανοιχτοί ταυτόχρονα.
GetPXRecordCount()	Επιστρέφει τον αριθμό των εγγραφών σεένα πίνακα.
OpenPXTable()	Ανοίγει έναν πίνακα.
PackPXTable()	Συμπιέζει την Βάση Δεδομένων (σβύνοντας εγγραφές που είναι μαρκαρισμένες για διαγραφή).
RenamePXTable()	Αλλάζει το όνομα μίας οικογένειας πινάκων.
SetPXTableCreateMode()	Θέτει την κατάσταση (την έκδοση της Paradox) για την δημιουργία πινάκων.
SetPXTableMaxSize()	Θέτει το μέγιστο μέγεθος για έναν πίνακα.
UpgradePXTable()	Αναβαθμίζει έναν παλιότερης έκδοσης πίνακα της Paradox (Paradox 3.5 ή μεγαλύτερη) σε πίνακα μεταγενέστερης μορφής.

Συναρτήσεις Εγγραφών

appendPXRecord()	Προσθέτει μία άδεια εγγραφή σε έναν πίνακα.
DeletePXRecord()	Σβύνει την τρέχουσα εγγραφή σε έναν πίνακα
emptyPXRecord()	Καθαρίζει από τα περιεχόμενά του την τρέχουσα εγγραφή του πίνακα.
FirstPXRecord()	Μετάβαση στην πρώτη εγγραφή του πίνακα.
GetPXRecordNumber()	Επιστρέφει τον αύξοντα αριθμό της τρέχουσας εγγραφής ενός πίνακα.

GotoPXRecord()	Μετάβαση σε συγκεκριμένη εγγραφή στον πίνακα.
InsertPXRecord()	Εισάγει μία άδεια εγγραφή σε έναν πίνακα.
LastPXRecord()	Μετάβαση στην τελευταία εγγραφή του πίνακα.
NextPXRecord()	Μετάβαση στην επόμενη εγγραφή ενός πίνακα.
PreviousPXRecord()	Μετάβαση στην προηγούμενη εγγραφή ενός πίνακα.
UpdatePXRecord()	Ενημερώνει κάθε αλλαγή της τρέχουσας εγγραφής σε έναν πίνακα.

Συναρτήσεις Πεδίων

emptyPXField()	Αδειάζει την τιμή ενός πεδίου στην τρέχουσα εγγραφή ενός πίνακα.
GetPXFieldNames()	Επιστρέφει μία λίστα με τα ονόματα των πεδίων σε έναν πίνακα.
GetPXFieldType()	Επιστρέφει τον τύπο για ένα συγκεκριμένο πεδίο.
GetPXFieldValue()	Επιστρέφει τα περιεχόμενα ενός πεδίο στην τρέχουσα εγγραφή ενός πίνακα.
SetPXFieldValue()	Αλλάζει τα περιεχόμενα ενός πεδίου της τρέχουσας εγγραφής ενός πίνακα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Π. Πολίτης. «ΥΠΕΡΚΕΙΜΕΝΑ ΥΠΕΡΜΕΣΑ ΠΟΛΥΜΕΣΑ.», Εκδόσεις Νέων Τεχνολογιών, 1994.

[2] Α. Μανιτσάρης, Α. Λιούτας. «Σχεδίαση και Υλοποίηση ενός Συστήματος Υπερμέσων για τη Διδασκαλία της Περιγραφικής Στατιστικής.», 9° Πανελλήνιο Συνέδριο Στατιστικής, Ξάνθη 1996.

[3] Α. Μανιτσάρης, Α. Λιούτας. «Το υποσύστημα Εκπαιδευτικής Αξιολόγησης του Υπερμεσικού Συστήματος Διδασκαλίας Στατιστικής (Multistat).», 10ο Πανελλήνιο Συνέδριο Στατιστικής, Πειραιάς 1997.

[4] M.Morris Mano. «Ψηφιακή Σχεδίαση.», Prentice Hall International, 1991.

[5] Α. Πομπόρτσης et al. «Multimedia στη θεωρία και στην πράξη.», Εκδόσεις Τζιόλα, 1996.

[6] Asymetrix Corp., «Toolbook User's Manual.», Washington DC - USA, 1992.

[7] Asymetrix Corp., «Toolbook Openscript Reference Manual.», Washington DC - USA, 1992.

[8] Σ. Μουρούτσος. «Σχεδιασμός και συγγραφή τίτλων πολυμέσων.», Σεμινάριο «Η χρήση των πολυμέσων στην εκπαίδευση και την επαγγελματική κατάρτιση - Πρόγραμμα Comett», 1995.

[9] ToolBook 3.0 Σημειώσεις, Μανιτσάρης Αθανάσιος, Μάστορας Κωνσταντινος, 1997