



**ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**



**Πτυχιακή εργασία**

**Δημιουργία ηλεκτρονικού βιβλίου με προσαρμοστικό τρόπο επικοινωνίας  
ανθρώπου υπολογιστή για άτομα με μαθησιακές δυσκολίες**



**Του φοιτητή**  
**Νεοφίδη Κωνσταντίνου**  
**Αρ. Μητρώου: 052765**

**Επιβλέπων καθηγητής**  
**Χατζάρα Κωνσταντίνα**

**Θεσσαλονίκη 2010**

## Περίληψη

Η ηλεκτρονική μάθηση σε συνεργασία με τον κλάδο των ηλεκτρονικών υπολογιστών προσπαθεί να ανεβάσει κατά ένα επίπεδο την εκπαιδευτική διαδικασία. Η ηλεκτρονική μάθηση είναι ένα εργαλείο το οποίο αν χρησιμοποιηθεί σωστά μπορεί να αμβλύνει δυο από τα κυριότερα προβλήματα της εκπαιδευτικής διαδικασίας που είναι ο χώρος και ο χρόνος. Ένα εργαλείο το οποίο μπορεί να χρησιμοποιηθεί στην ηλεκτρονική μάθηση είναι αναμφισβήτητα το ηλεκτρονικό βιβλίο το οποίο προσφέροντας όλες τις ευκολίες που μπορεί να προσφέρει ένα ηλεκτρονικό μέσο έρχεται να κλέψει λίγη από τη δόξα και τη μαγεία του κλασσικού παραδοσιακού βιβλίου. Μια εποχή σαν και αυτή στην οποία υπάρχει μια μεγάλη ποικιλία ανθρώπων, τεχνικών, μεθόδων κτλ. Αυξάνεται η ανάγκη για εξειδίκευση και εξατομίκευση. Η τάση του να αυξάνεται η προσαρμοστικότητα μιας εφαρμογής έχει ως σκοπό να προσφέρει το μέγιστο από αυτή και να δώσει μια αίσθηση στον χρήστη ότι η εφαρμογή σχεδιάστηκε αποκλειστικά για τον ίδιο. Ένας ευφυής πράκτορας μπορεί να “ζωντανέψει” μια εφαρμογή, η εφαρμογή επικοινωνεί με τον χρήστη μέσω του πράκτορα που με την παρουσία του κάνει ελκυστικότερη μια εφαρμογή.

<b>ΠΕΡΙΛΗΨΗ.....</b>	<b>2</b>
<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>5</b>
<b>1 ΗΛΕΚΤΡΟΝΙΚΗ ΜΑΘΗΣΗ (E-LEARNING) .....</b>	<b>6</b>
1.1 ΕΙΣΑΓΩΓΙΚΑ .....	6
1.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ.....	6
1.3 ΟΡΙΣΜΟΣ .....	7
1.4 ΠΛΕΟΝΕΚΤΗΜΑΤΑ E- LEARNING .....	8
1.5 ΜΕΙΟΝΕΚΤΗΜΑΤΑ E- LEARNING.....	9
1.6 ΕΠΙΛΟΓΟΣ .....	9
<b>2 ΗΛΕΚΤΡΟΝΙΚΟ ΒΙΒΛΙΟ.....</b>	<b>10</b>
2.1 ΕΙΣΑΓΩΓΙΚΑ .....	10
2.2 ΟΡΙΣΜΟΣ .....	10
2.3 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ.....	10
2.4 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΗΛΕΚΤΡΟΝΙΚΟΥ ΒΙΒΛΙΟΥ .....	11
2.5 ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΗΛΕΚΤΡΟΝΙΚΟΥ ΒΙΒΛΙΟΥ .....	12
2.6 ΛΟΓΙΣΜΙΚΑ ΚΑΙ ΣΥΣΚΕΥΕΣ ΑΝΑΓΝΩΣΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΒΙΒΛΙΩΝ.....	12
2.7 ΕΠΙΛΟΓΟΣ .....	13
<b>3 ΠΡΟΣΑΡΜΟΣΤΙΚΑ ΥΠΕΡΜΕΣΑ .....</b>	<b>14</b>
3.1 ΕΙΣΑΓΩΓΙΚΑ .....	14
3.2 ΟΡΙΣΜΟΣ .....	14
3.3 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ.....	15
3.4 ΕΙΔΗ ΠΡΟΣΑΡΜΟΣΤΙΚΟΤΗΤΑΣ .....	15
3.5 ΤΕΧΝΙΚΕΣ ΚΑΙ ΜΕΘΟΔΟΙ ΠΡΟΣΑΡΜΟΣΤΙΚΩΝ ΥΠΕΡΜΕΣΩΝ .....	16
3.5.1 Προσαρμοστική παρουσίαση.....	16
3.5.1.1 Μέθοδοι της προσαρμοστικής παρουσίασης .....	16
3.5.1.2 Τεχνικές της προσαρμοστικής παρουσίασης .....	17
3.5.2 Προσαρμοστική πλοήγηση.....	18
3.5.2.1 Μέθοδοι της προσαρμοστικής πλοήγησης .....	19
3.5.2.2 Τεχνικές της προσαρμοστικής πλοήγησης .....	20
3.6 ΤΑ ΚΥΡΙΑ ΤΜΗΜΑΤΑ ΕΝΟΣ ΠΡΟΣΑΡΜΟΣΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΥΠΕΡΜΕΣΟΥ.....	21
3.7 ΕΠΙΛΟΓΟΣ .....	22
<b>4 ΕΥΦΥΕΙΣ ΠΡΑΚΤΟΡΕΣ .....</b>	<b>23</b>
4.1 ΕΙΣΑΓΩΓΙΚΑ .....	23
4.2 ΟΡΙΣΜΟΣ .....	23
4.3 ΕΠΙΠΛΕΟΝ ΟΡΙΣΜΟΙ.....	24
4.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΠΡΑΚΤΟΡΩΝ .....	25
4.5 ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΕΥΦΥΩΝ ΠΡΑΚΤΟΡΩΝ .....	26
4.5.1 Ορθολογικοί Πράκτορες (Rational Agents).....	27
4.5.2 Αντιδραστικοί πράκτορες (Reactive Agents) .....	27
4.5.3 Η Αρχιτεκτονική BDI .....	28
4.5.4 Διαστρωματωμένη Αρχιτεκτονική ή Αρχιτεκτονική Πολλαπλών Επιπέδων (Layered Agents)..	32
4.6 ΕΥΦΥΕΙΣ ΠΡΑΚΤΟΡΕΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ .....	35
4.7 ΕΥΦΥΗ ΕΚΠΑΙΔΕΥΤΙΚΑ ΣΥΣΤΗΜΑΤΑ .....	36
4.7.1 MATS (Multi-Agent Tutoring System) .....	37
4.7.2 Ένας πράκτορας διαμεσολάβησης για εκπαιδευτικά συστήματα βοηθούμενα από υπολογιστές	38
4.7.3 Χρήση κινούμενων χαρακτήρων (life-like animated characters) σε εκπαιδευτικές εφαρμογές.	38
4.7.4 Intelligent Helpdesk.....	39
4.7.5 Adaptive Statistics Tutor (AST) .....	40
4.7.6 AlgeBrain, ένας βοηθός διδασκαλίας άλγεβρας .....	41
4.8 ΕΠΙΛΟΓΟΣ .....	42
<b>5 ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ .....</b>	<b>43</b>
5.1 ΣΤΟΙΧΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ .....	43
5.1.1 ΡΗΡ .....	43

5.1.2	MySQL.....	43
5.1.3	Adobe Photoshop.....	44
5.1.4	Adobe Flash.....	44
5.1.5	ESOPOS.....	44
	ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	44
5.2.1	Οθόνη εισόδου χρήστη στο σύστημα.....	45
5.2.2	Οθόνη εγγραφής χρήστη στο σύστημα.....	48
5.2.3	Οθόνη Ερωτηματολογίου.....	49
5.2.4	Οθόνη 1 <sup>ης</sup> Άσκησης.....	51
5.2.5	Οθόνη 2 <sup>ης</sup> Άσκησης.....	53
5.2.6	Οθόνη 3 <sup>ης</sup> Άσκησης.....	58
5.2.7	Animation εζόδου.....	62
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>		<b>63</b>
	ΕΛΛΗΝΙΚΗ.....	63
	ΞΕΝΗ.....	63
	ΙΣΤΟΣΕΛΙΔΕΣ.....	65
<b>ΠΑΡΑΡΤΗΜΑ Α.....</b>		<b>66</b>
A.1	ΚΩΔΙΚΑΣ ΕΙΣΟΔΟΥ ΧΡΗΣΤΗ ΣΤΟ ΣΥΣΤΗΜΑ.....	66
A.2	ΚΩΔΙΚΑΣ ΕΓΓΡΑΦΗΣ ΧΡΗΣΤΗ ΣΤΟ ΣΥΣΤΗΜΑ.....	70
A.3	ΚΩΔΙΚΑΣ ΑΝΑΝΕΩΣΗΣ ΤΗΣ ΒΑΣΗΣ.....	73
A.4	ΚΩΔΙΚΑΣ ΓΙΑ ΝΑ ΑΚΟΛΟΥΘΟΥΝ ΤΑ ΜΑΤΙΑ ΤΟΥ ΠΡΑΚΤΟΡΑ ΤΟ ΠΟΝΤΙΚΙ.....	75
A.5	ΚΩΔΙΚΑΣ ΑΝΕΝΕΡΓΙΑΣ ΧΡΗΣΤΗ.....	76
A.6	ΚΩΔΙΚΑΣ ΚΙΝΗΣΗΣ ΤΟΥ ΠΡΑΚΤΟΡΑ ΣΤΟ ΕΠΙΤΡΑΠΕΖΙΟ ΤΗΣ ΆΣΚΗΣΗΣ 2.....	77

## Εισαγωγή

Η παρούσα πτυχιακή εργασία έχει ως θέμα την δημιουργία ενός ηλεκτρονικού βιβλίου με προσαρμοστικό τρόπο επικοινωνίας ανθρώπου υπολογιστή για άτομα με μαθησιακές δυσκολίες. Στα πλαίσια της δημιουργείται με την βοήθεια του Adobe Flash μια εφαρμογή που παραπέμπει σε βιβλίο και έχει ως στόχο να μπορεί να χρησιμοποιηθεί με την ίδια ευκολία, με τον ίδιο διασκεδαστικό τρόπο και από παιδιά με μια μαθησιακή δυσκολία όπως η δυσαριθμησία αλλά και από παιδιά χωρίς αυτήν. Να έχει δηλαδή την δυνατότητα η εφαρμογή να προσαρμόζεται στις ιδιαιτερότητες κάθε χρήστη.

Όσο αφορά την δομή του συγγράμματος, αυτό αποτελείται από 5 κεφάλαια :

Το πρώτο κεφάλαιο ασχολείται με την ηλεκτρονική μάθηση. Ξεκινά με την ιστορική αναδρομή της ηλεκτρονικής μάθησης και συνεχίζει με τον ορισμό της. Τέλος το κεφάλαιο κλείνει με τα πλεονεκτήματα και μειονεκτήματα της ηλεκτρονικής μάθησης.

Το δεύτερο κεφάλαιο αφορά το ηλεκτρονικό βιβλίο. Στην αρχή του κεφαλαίου γίνεται ο ορισμός του ηλεκτρονικού βιβλίου και στην συνέχεια αναφέρονται κάποια ιστορικά γεγονότα γύρω από αυτό. Στην συνέχεια παρουσιάζονται κάποια από τα πλεονεκτήματα και τα μειονεκτήματα της χρήσης ηλεκτρονικών βιβλίων. Τέλος παρουσιάζονται κάποια λογισμικά και συσκευές με την βοήθεια των οποίων γίνεται δυνατή η ανάγνωση τέτοιων βιβλίων.

Το τρίτο κεφάλαιο αναφέρεται στα προσαρμοστικά υπερμέσα. Το κεφάλαιο ξεκινά με τον ορισμό και την ιστορική αναδρομή, έπειτα παρουσιάζονται τα είδη της προσαρμοστικότητας και γίνεται μια αναλυτική περιγραφή των τεχνικών και των μεθόδων των προσαρμοστικών υπερμέσων. Τέλος γίνεται μια αναφορά στα κύρια τμήματα ενός προσαρμοστικού συστήματος υπερμέσου.

Το τέταρτο κεφάλαιο ασχολείται με τους ευφυείς πράκτορες. Στην αρχή του κεφαλαίου γίνεται μια προσπάθεια να οριστεί η έννοια του ευφυή πράκτορα, στην συνέχεια γίνεται μια παράθεση των χαρακτηριστικών ενός ευφυή πράκτορα. Έπειτα γίνεται μια λεπτομερής αναφορά στις αρχιτεκτονικές που χρησιμοποιούνται γύρω από τους ευφυείς πράκτορες. Το κεφάλαιο κλείνει με λίγα λόγια για την συμβολή και την χρήση των ευφυών πρακτόρων στην εκπαίδευση και την παρουσίαση κάποιων ευφυών εκπαιδευτικών συστημάτων.

Τέλος στο πέμπτο κεφάλαιο γίνεται η παρουσίαση της εφαρμογής.

# **1 Ηλεκτρονική μάθηση (e-learning)**

## **1.1 Εισαγωγικά**

Με τον ερχομό των ηλεκτρονικών υπολογιστών αλλά και μετέπειτα του διαδικτύου πολλοί τομείς που αφορούν την ζωή μας επηρεάστηκαν, ένας από αυτούς ήταν και αυτός της εκπαίδευσης. Οι νέες τεχνολογίες προσφέρουν όλο και περισσότερες δυνατότητες αναβάθμισης της εκπαιδευτικής διαδικασίας. Τέλος το γεγονός ότι βιώνουμε την εποχή της δια βίου μάθησης κάνει ακόμα πιο αναγκαία την ηλεκτρονική μάθηση.

## **1.2 Ιστορική αναδρομή**

Η ανάγκη για μάθηση και κατά συνέπεια για γνώση σε όλα τα μήκη και πλάτη της γης οδήγησαν στην ανακάλυψη ενός νέου τρόπου μάθησης αυτού της εκπαίδευσης από απόσταση η οποία στα πρώιμα βήματα της είχε την μορφή αλληλογραφίας μέσω ταχυδρομείου. Αρχικά χρησιμοποιούνταν έντυπα μέσα και ειδικότερα τα εκπαιδευτικά βιβλία της εποχής. Αργότερα, καθώς εξελισσόταν η τεχνολογία χρησιμοποιήθηκαν μέσα όπως το ραδιόφωνο, η τηλεόραση και το τηλέφωνο για αμεσότερη εξ αποστάσεως διδασκαλία ενώ τα τελευταία χρόνια υιοθετήθηκαν όλες σχεδόν οι μορφές της σύγχρονης τεχνολογίας (Keegan, 2001).

Παρόλο που στην αρχή ο διδάσκων με τον διδασκόμενο απλά αντάλλασσαν μηνύματα, μετέπειτα, η επικοινωνία τους αυτή υιοθέτησε κάποιες πολυμεσικές δυνατότητες όπως το υπερκείμενο, το βίντεο, την εικόνα και τον ήχο. Αυτό είχε ως συνέπεια πολλά εκπαιδευτικά ιδρύματα να δημιουργήσουν ολοκληρωμένα εκπαιδευτικά περιβάλλοντα που βασίζονταν αποκλειστικά στις τεχνολογίες αυτές (Rosenberg, 2000).

### 1.3 Ορισμός

Η έννοια e-learning ή ηλεκτρονική μάθηση είναι αρκετά γενική και περιλαμβάνει οποιαδήποτε μορφή εκπαίδευσης χρησιμοποιεί τους πόρους του δικτύου ή γενικότερα τις δυνατότητες των ηλεκτρονικών υπολογιστών. Συνήθως ο εκπαιδευόμενος έχει τον έλεγχο του ρυθμού προόδου ενώ παράλληλα του δίνεται η δυνατότητα να ζητήσει υποστήριξη από τον εκπαιδευτή ή κάποιον ειδικό πάνω στο θέμα.

Όπως συμβαίνει στην κλασσική εκπαίδευση δηλαδή σε μια παραδοσιακή αίθουσα διδασκαλίας έτσι και στην ηλεκτρονική μάθηση ενδέχεται να εμπεριέχεται το στοιχείο της συνεργατικής εκπαίδευσης αλλά και της αλληλεπίδρασης μεταξύ μαθητών-εκπαιδευτών και μεταξύ μαθητών.

Η διδασκαλία χωρίζεται σε δυο κατηγορίες (Μπαλαούρας, 2002):

- **Σύγχρονη διδασκαλία.** Σε αυτή την κατηγορία οι συμμετέχοντες δεν είναι ανάγκη να βρίσκονται όλοι μαζί στην ίδια φυσική αίθουσα αλλά μπορούν να βρίσκονται όπου θέλουν π.χ. σπίτι. Χρησιμοποιώντας τεχνολογίες τηλεδιάσκεψης μπορούμε να δημιουργήσουμε μια εικονική αίθουσα διδασκαλίας η οποία θα δίνει την αίσθηση στους συμμετέχοντες ότι βρίσκονται όλοι μαζί στον ίδιο χώρο προσφέροντας παράλληλα τις ίδιες ή και παραπάνω δυνατότητες με αυτές που προσφέρονται σε μία κανονική αίθουσα.  
Η Συνομιλία (chat) δίνει την δυνατότητα σε άτομα της ίδιας ομάδας εργασίας να γράψουν και να στείλουν μηνύματα στους υπόλοιπους σε πραγματικό χρόνο μέσω του δικτύου.
- **Ασύγχρονη διδασκαλία.** Στην κατηγορία αυτή οι συμμετέχοντες έχουν την δυνατότητα να εργαστούν με το εκπαιδευτικό υλικό όπου και όποτε θέλουν έχοντας παράλληλα δυνατότητα ασύγχρονης επικοινωνίας με τους υπόλοιπους συμμετέχοντες και με τον εκπαιδευτή. Το εκπαιδευτικό

υλικό μπορεί να έχει δοθεί είτε από την αρχή είτε να δίδεται σταδιακά. Ο εκπαιδευτής είναι υπεύθυνος για τον ρυθμό ερχόμενος όμως και σε συνεννόηση με του εκπαιδευόμενους.

Το Ηλεκτρονικό ταχυδρομείο (E-mail) δίνει την δυνατότητα σε άτομα της ίδιας ομάδας εργασίας να επικοινωνήσουν μεταξύ τους ασύγχρονα χωρίς να είναι απαραίτητο ο παραλήπτης να απαντήσει στον αποστολέα την ίδια στιγμή.

Σύμφωνα με τον ιστοχώρο [www.teleteaching.gr](http://www.teleteaching.gr) , στον οποίο επιστημονικός υπεύθυνος είναι ο καθηγητής του ΕΜΠ κ.Μάγκλαρης παραθέτονται μερικά από τα πλεονεκτήματα και τα μειονεκτήματα του e- learning.

#### **1.4 Πλεονεκτήματα e- learning**

Μερικά από τα σημαντικότερα πλεονεκτήματα του e- learning είναι τα εξής:

- Είναι διαθέσιμο ανεξάρτητα από το που βρίσκεται ο καθένας, αυξάνοντας κατά αυτόν τον τρόπο το κίνητρο για μάθηση (Keegan, 2001).
- Είναι πάντα διαθέσιμο, το γεγονός αυτό μας δίνει την δυνατότητα να το επαναλάβουμε όσες φορές θέλουμε.
- Εξοικονομούνται χρήματα από όλες τις πλευρές (εκπαιδευόμενοι, εκπαιδευτές κτλ.) καθώς αυτό που αρκεί είναι να υπάρχει πρόσβαση σε έναν Η/Υ και στο διαδίκτυο.
- Είναι εξαιρετικά αποτελεσματικό, όταν με την βοήθεια πολυμέσων, βίντεο, ήχου, κειμένου, εικόνων και διαλογικής συνεργασίας.
- Ο αριθμός των εκπαιδευόμενων είναι πρακτικά απεριόριστος.
- Ο εκπαιδευόμενος διαλέγει τον τρόπο διδασκαλίας.
- Ο εκπαιδευόμενος καθορίζει τον χρόνο και τον ρυθμό μελέτης του και αν απαιτείται υπάρχει η δυνατότητα πραγματοποίησης κάποιων συναντήσεων ανάλογα με τις ανάγκες του (εξατομίκευση διδασκαλίας).
- Η αίσθηση που δίνεται στους εκπαιδευόμενους ότι ελέγχουν όλη την διαδικασία ενισχύει τον πειραματισμό και την αυτενέργεια κατά τη διεργασία της μάθησης.



- «Αδύναμες» ομάδες ή άτομα έχουν πλέον περισσότερες δυνατότητες πρόσβασης και ίσης συμμετοχής καθώς αμβλύνονται τα εμπόδια όπως το κόστος, ο χρόνος και ο χώρος.

### **1.5 Μειονεκτήματα e-learning**

- Οι εκπαιδευτές θα πρέπει να εξοικειωθούν με τις διάφορες μεθόδους και πρακτικές της ηλεκτρονικής μάθησης.
- Το γεγονός ότι απαιτείται πρόσβαση σε πολυμεσικό ηλεκτρονικό εκπαιδευτικό υλικό απαιτεί την απόκτηση επιπρόσθετου εξοπλισμού ή αναβάθμισης του υπάρχοντος .
- Η έλλειψη άμεσης προσωπικής επαφής των εκπαιδευομένων, τόσο μεταξύ τους όσο και με τον εκπαιδευτή, η οποία αποτελεί και σημαντικό παράγοντα ανταλλαγής απόψεων και γνώσης.
- Θα πρέπει να διασφαλιστεί ότι η αξιολόγηση των εκπαιδευομένων από απόσταση θα είναι αδιάβλητη και αποτελεσματική .
- Οι υποχρεώσεις του εκπαιδευτή αυξάνονται καθώς υποχρεώνεται έκτος από το χρόνο του μαθήματος να αφιερώνει και χρόνο για τη προετοιμασία του μαθήματος, για τη δημιουργία και συντήρηση του ψηφιακού υλικού καθώς και για την ασύγχρονη επικοινωνία με τους μαθητές.

### **1.6 Επίλογος**

Πρέπει να κατανοήσουμε ότι η ηλεκτρονική μάθηση δεν έχει σαν σκοπό να αντικαταστήσει τον τωρινό τρόπο διδασκαλίας. Η ηλεκτρονική μάθηση έρχεται να συμπληρώσει την παρούσα εκπαιδευτική διαδικασία, να βοηθήσει το διδάσκοντα να προσφέρει περισσότερη, πιο πλήρη και σφαιρική γνώση στους μαθητές. Οι νέες τεχνολογίες πρέπει να χρησιμοποιηθούν για να εμπλουτίσουν το μάθημα και να το κάνουν πιο ενδιαφέρον. Η τεχνολογία έχει εισβάλλει παντού και η εξοικείωση με αυτή είναι απαραίτητη για όλους και ειδικά για τους αυριανούς πολίτες και εργαζόμενους. Είναι λοιπόν αναγκαίο για τους μαθητές να έρθουν σε επαφή με νέες τεχνολογίες, να μάθουν να τις χρησιμοποιούν και να εκμεταλλεύονται τις δυνατότητες που τους δίνουν. Μέσα από την ηλεκτρονική μάθηση η επαφή και εξοικείωση αυτή γίνεται με τρόπο φυσικό και ευχάριστο για τους μαθητές.

## **2 Ηλεκτρονικό βιβλίο**

### **2.1 Εισαγωγικά**

Η είσοδος μας στην εποχή της ψηφιοποίησης έχει επιφέρει κάποιες αλλαγές, μια από αυτές είναι και το γεγονός ότι το κλασσικό βιβλίο δίνει την θέση του στο ηλεκτρονικό βιβλίο, το οποίο φιλοδοξεί να πατήσει πάνω στα θετικά του κλασσικού βιβλίου και να προσφέρει στον αναγνώστη ακόμα μεγαλύτερη ευκολία και απόλαυση στο διάβασμα.

### **2.2 Ορισμός**

Ηλεκτρονικό βιβλίο είναι ένα «ψηφιακό αντικείμενο με κείμενα ή/και άλλου είδους περιεχόμενο, το οποίο προκύπτει ενσωματώνοντας στη γνωστή έννοια του βιβλίου χαρακτηριστικά που υπάρχουν σε ηλεκτρονικά περιβάλλοντα. Τα ηλεκτρονικά βιβλία τυπικά έχουν λειτουργίες αναζήτησης και παραπομπών, υπερσυνδέσεις, σελιδοδείκτες, επισημειώσεις, υπογραμμίσεις, πολυμεσικά αντικείμενα και διαδραστικά εργαλεία» (Vassiliou and Rowley, 2008)

### **2.3 Ιστορική αναδρομή**

Όλα ξεκίνησαν όταν ο Alan Kay, φοιτητής στο πανεπιστήμιο της Utah, παρουσίασε ένα ηλεκτρονικό βιβλίο με το όνομα Dynabook.

Ένας ακόμα πρωτοπόρος πάνω σε αυτό το θέμα είναι και ο Andries Van Dam, ο οποίος αναγνωρίζεται από πολλούς ως αυτός που επινόησε τον όρο «ηλεκτρονικό βιβλίο (electronic book)», πριν από 40 χρόνια. Το '67 και το '68, ο Van Dam ηγήθηκε μιας ομάδας η οποία και ανέπτυξε το πρώτο hypertext σύστημα. «Το Hyper Text Editing System έτρεχε σε ένα τερματικό IBM/360 με 128 K μνήμης και χρηματοδοτήθηκε από την IBM.

Κατά την διάρκεια του 1970, χρησιμοποιήθηκε ένα σύστημα ανάκτησης και επεξεργασίας αρχείων (File Retrieval and Editing System - FRESS) από φοιτητές και μέλη πανεπιστημίου του Brown. Το FRESS είχε δυναμική ιεραρχία (όπως κεφάλαια/ενότητες) , διπλής κατεύθυνσης συνδέσμους παραπομπών και συνδέσμους με λέξεις κλειδιά.

Το 1971, ο Michael Hart ξεκίνησε το σχέδιο Guttenberg , και στα πλαίσια του πληκτρολόγησε την «Διακήρυξη της Ανεξαρτησίας» και διέθεσε το κείμενο μέσω πολλών δικτύων. Από το '71, η ομάδα του προγράμματος Guttenberg έχει διαθέσει κοντά στους 2,100 τίτλους.

Σύμφωνα με το άρθρο «*The Seybold Report on Internet Publishing*», οι εκδοτικοί οίκοι ψηφιοποιούν βιβλία για περισσότερο από μία δεκαετία. Η Sony προσπάθησε να υλοποιήσει διάφορες εκδόσεις ενός φορητού ηλεκτρονικού βιβλίου ανεπιτυχώς στις αρχές του 90. Ιδιαίτερη άνθηση γνώρισαν εγκυκλοπαίδειες οι οποίες συνδύαζαν στοιχεία αλληλεπίδρασης και multimedia και είχαν τη μορφή CD-ROM. Το Adobe Acrobat (το οποίο και χρησιμοποιούσε αρχεία που έγιναν γνωστά ως PDF – Portable Document Format), παρουσιάστηκε ως εμπορικό λογισμικό για την έκδοση και παρουσίαση ηλεκτρονικών βιβλίων. (Ανδριώτης, 2007)

## **2.4 Πλεονεκτήματα ηλεκτρονικού βιβλίου**

Μερικά από τα πιο σημαντικά πλεονεκτήματα των ηλεκτρονικών βιβλίων είναι τα εξής:

- Παραμένουν αναλλοίωτα στον χρόνο καθώς δεν φθείρονται, έτσι μπορούν να παραμείνουν για αρκετό καιρό στην κατοχή κάποιου χρήστη.
- Ο χώρος αποθήκευσης που απαιτούνται για τα ηλεκτρονικά βιβλία είναι σαφώς πιο λίγος από τον χώρο που απαιτείται για την αποθήκευση συμβατικών βιβλίων.
- Η αναζήτηση μιας λέξης ή μιας φράσης μπορεί να γίνει εύκολα και γρήγορα.
- Καθώς το κόστος αντιγραφής ενός ηλεκτρονικού βιβλίου είναι μικρό ή μηδενικό, μπορούμε να δημιουργήσουμε εύκολα όσα αντίτυπα θέλουμε.
- Άτομα με προβλήματα όρασης ή ακοής μπορούν να βοηθηθούν καθώς το μέγεθος της γραμματοσειράς του κειμένου μπορεί να αυξομειωθεί. Επίσης με τη βοήθεια ειδικού λογισμικού το κείμενο μπορεί να μετατραπεί σε ήχο.
- Ένα ηλεκτρονικό βιβλίο μπορεί να διαβαστεί ταυτόχρονα από πολλούς αναγνώστες.
- Το διαδίκτυο κάνει διαθέσιμα τα ηλεκτρονικά βιβλία σε κάθε γωνιά της γης.
- Από οικολογικής άποψης δεν χρειάζεται χαρτί για την εκτύπωση των ηλεκτρονικών βιβλίων.

## **2.5 Μειονεκτήματα ηλεκτρονικού βιβλίου**

Από την άλλη πλευρά υπάρχουν και μερικά μειονεκτήματα τα σημαντικότερα από τα οποία είναι τα εξής:

- Τα σημαντικότερα προβλήματα που συναντιούνται γύρω από το ηλεκτρονικό βιβλίο έχουν να κάνουν με θέματα πνευματικής ιδιοκτησίας. Υπάρχουν φαινόμενα αντιγραφής και διακίνησης συγγραμμάτων χωρίς την συγκατάθεση του συγγραφέα.
- Η αλλαγή του περιεχομένου ενός ηλεκτρονικού βιβλίου είναι εύκολη υπόθεση.
- Για την ανάγνωση ενός ηλεκτρονικού βιβλίου απαιτείται συχνά κάποια συσκευή ή κάποιο λογισμικό το οποίο καμιά φορά μπορεί να είναι δαπανηρό.
- Η ανάγνωση ενός τέτοιου βιβλίου μπορεί να γίνει κουραστική αν διαρκεί παρατεταμένη ώρα.

## **2.6 Λογισμικά και συσκευές ανάγνωσης ηλεκτρονικών βιβλίων**

- **Microsoft Reader.** Λογισμικό ανάγνωσης και διαχείρισης ηλεκτρονικών βιβλίων, αναπτυγμένο από την Microsoft.
- **Adobe Reader.** Πρόκειται για ίσως το δημοφιλέστερο αυτή τη στιγμή λογισμικό ανάγνωσης ηλεκτρονικών βιβλίων. Δέχεται αρχεία τύπου PDF (Portable Document Format) , format αναπτυγμένο από την ίδια την Adobe.
- **BeBook.** Το BeBook είναι ένας φορητός αναγνώστης ηλεκτρονικών βιβλίων. Υποστηρίζει περισσότερες από 20 μορφές δημοφιλών αρχείων κειμένου και εικόνας. Επίσης υποστηρίζει αρχεία mp3 για αναπαραγωγή ακουστικών βιβλίων (audiobooks).

## 2.7 Επίλογος

Παρά την πρόοδο της τεχνολογίας το βιβλίο εξακολουθεί να υπερέχει στις προτιμήσεις των αναγνωστών. Η μετάβαση από το συμβατικό βιβλίο στο ηλεκτρονικό δεν θα είναι εύκολη, θα είναι όμως ένα βήμα μπροστά. Στην εκπαίδευση, όπου ο αναγνώστης έχει συνηθίσει διαβάζει από ένα κλασσικό βιβλίο μπορούμε να κρατήσουμε το interface του κλασσικού βιβλίου για να παρουσιάσουμε το εκπαιδευτικό μας υλικό, αυτό που θα αλλάξει θα είναι ο τρόπος με τον οποίον θα γίνει αυτή η παρουσίαση. Μια σκέψη θα ήταν από στατική να γίνει δυναμική. Ένα πολύ σημαντικό στοιχείο το οποίο εισάγει το ηλεκτρονικό βιβλίο είναι η διάδραστικότητα. Πλέον υπάρχει ανάδραση, το γεγονός αυτό μπορεί να συντελέσει στην αύξηση του ενδιαφέροντος του αναγνώστη και στην επιθυμία του να συνεχίσει το διάβασμα. Επίσης το γεγονός ότι το ηλεκτρονικό βιβλίο έχει την δυνατότητα να προσαρμόζεται στις ανάγκες και ιδιαιτερότητες του κάθε αναγνώστη συνιστά ένα από τα βασικά του πλεονεκτήματα.

## 3 Προσαρμοστικά υπερμέσα

### 3.1 Εισαγωγικά

Κάθε άνθρωπος είναι διαφορετικός, έχει διαφορετικά χαρακτηριστικά και διαφορετικές ανάγκες. Τα προσαρμοστικά υπερμέσα κάνουν μια εφαρμογή να είναι ικανή να αντεπεξέλθει στις συγκεκριμένες γνώσεις και ανάγκες του χρήστη. Χρησιμοποιούν την γνώση η οποία αποθηκεύεται στο μοντέλο του χρήστη έτσι ώστε να προσαρμόσουν τόσο τη πληροφορία όσο και τους συνδέσμους πάνω στον συγκεκριμένο χρήστη.

### 3.2 Ορισμός

*«Με τον όρο **προσαρμοστικά υπερμέσα** εννοούμε όλα τα υπερκείμενα και τα υπερμέσα, τα οποία αναγνωρίζουν κάποια χαρακτηριστικά του χρήστη και με την βοήθεια τους σχηματίζουν το λεγόμενο μοντέλο του χρήστη (*user model*) το οποίο στη συνέχεια χρησιμοποιούν για να προσαρμόσουν το σύστημα πάνω στον χρήστη» (Brusilovsky, 1996)*

Ένα Σύστημα Προσαρμοστικών Υπερμέσων πρέπει:

- να είναι βασισμένο σε ένα υπερκείμενο ή και υπερμέσο
- να περιλαμβάνει ένα μοντέλο γνωστικής περιοχής (*domain model*)
- να περιλαμβάνει ένα μοντέλο χρήστη (*user model*), το οποίο αποθηκεύει τα χαρακτηριστικά των χρηστών
- να προσαρμόζει κάποια οπτικά ή και λειτουργικά μέρη του συστήματος χρησιμοποιώντας το μοντέλο του χρήστη.

### 3.3 Ιστορική αναδρομή

Η πρώτη εμφάνιση των προσαρμοστικών υπερμέσων έγινε στα τέλη της δεκαετίας του 80' και στις αρχές του 90'. Ορόσημο στην ιστορία αυτών των συστημάτων ήταν το έτος 1996 τόσο από πλευράς μελέτης και ανάπτυξης, όσο και από πλευράς ολοκλήρωσης και υλοποίησης.

Πριν το 1996, όλα τα συστήματα υπερμέσων παρουσίαζαν παρόμοια χαρακτηριστικά. Επιπλέον ακολουθούσαν την ίδια λογική, η συμπεριφορά πλοήγησης του χρήστη ήταν εκείνη που καθόριζε το μοντέλο του χρήστη (*user model*), στο οποίο αποθηκευόντουσαν τα χαρακτηριστικά του χρήστη. Εν τέλει το μοντέλο του χρήστη χρησιμοποιούνταν ως βάση για την εφαρμογή της προσαρμογής.

Μετά το 1996 με την είσοδο του ιστού η μελέτη και η υλοποίηση τέτοιων συστημάτων μεταβαίνει σε μια άλλη διάσταση καθώς αυξάνεται ο αριθμός και ο τύπος τους. Μερικά από τα συστήματα αυτά εξελίσσονται ενώ άλλα δημιουργούνται, επίσης την περίοδο αυτή γνωρίζουν ραγδαία ανάπτυξη τα εκπαιδευτικά υπερμέσα (*educational hypermedia systems*), τα συστήματα on-line πληροφοριών (*on-line information systems*) καθώς και τα συστήματα ανάκτησης πληροφοριών (*information retrieval hypermedia*). (<http://esp.it.teithe.gr/documents/PE09/prosarmostikaupermesa.doc>)

### 3.4 Είδη προσαρμοστικότητας

- **Στατική προσαρμοστικότητα (*adaptability*)**. Στα υπερμέσα με στατική προσαρμοστικότητα το περιβάλλον αλλάζει μια φορά αυτόματα κατά την έναρξη κάθε συνόδου.
- **Δυναμική προσαρμοστικότητα (*adaptivity*)**. Στα υπερμέσα με δυναμική προσαρμοστικότητα το περιβάλλον αλλάζει καθ' όλη την διάρκεια της συνόδου. Τα χαρακτηριστικά των χρηστών τα οποία ενδέχεται να αλλάζουν κατά την διάρκεια της συνόδου και δεν είναι γνωστά από την αρχή αλλά ανιχνεύονται από το σύστημα, είναι εκείνα που τελικά κάνουν το περιβάλλον να αλλάζει.

- **Προσαρμοστικότητα από τους χρήστες (*tailorability*).** Στα υπερμέσα αυτά δίνεται η δυνατότητα στον ίδιο τον χρήστη να κάνει κάποιες αλλαγές χωρίς την παρέμβαση του συστήματος.

### **3.5 Τεχνικές και μέθοδοι προσαρμοστικών υπερμέσων**

#### **3.5.1 Προσαρμοστική παρουσίαση**

Ενδέχεται μερικές φορές να θέλουμε η παρουσίαση των πληροφοριών ενός συγκεκριμένου θέματος να γίνεται με διαφορετικούς τρόπους. Αυτό που θα καθορίζει τον τρόπο με τον οποίο θα παρουσιάζεται μια πληροφορία είναι τα διάφορα χαρακτηριστικά ενός χρήστη. π.χ. γνώσεις ,στόχοι, κτλ

##### **3.5.1.1 Μέθοδοι της προσαρμοστικής παρουσίασης**

Υπάρχουν πέντε μέθοδοι προσαρμοστικής παρουσίασης (Wieringa and Dubois, 1994) :

1. οι επιπρόσθετες επεξηγήσεις
2. οι προαπαιτούμενες επεξηγήσεις
3. οι συγκριτικές επεξηγήσεις
4. οι παραλλαγές επεξήγησης
5. η ταξινόμηση

Αυτό που κάνει η μέθοδος των επιπρόσθετων επεξηγήσεων είναι να παρέχει κάποια επιπλέον βοήθεια με την μορφή επεξηγήσεων, απεικονίσεων κτλ σε όσους χρήστες κρίνεται ότι είναι αναγκαίο. Από την άλλη το σύστημα αποκρύπτει τέτοιες βοήθειες από χρήστες που δεν τις έχουν ανάγκη.

Με την προσθήκη μιας προαπαιτούμενης επεξήγησης το σύστημα έχει ως σκοπό να καταστήσει το υπόλοιπο της σελίδας πιο κατανοητό κάτι που



ενδεχομένως να μην ήταν δυνατό σε αντίθετη περίπτωση. αποτελούν κατά κάποιο τρόπο μια ειδική περίπτωση.

Η μέθοδος της συγκριτικής επεξήγησης χρησιμοποιείται όταν υπάρχουν πληροφορίες για περιεχόμενα που είναι παρόμοια με αυτό που παρουσιάζεται στην τρέχουσα σελίδα.

Υπάρχει περίπτωση δυο ή περισσότεροι χρήστες να χρειάζονται την ίδια περίπου επεξήγηση, σε αυτήν την περίπτωση χρησιμοποιείται η μέθοδος των παραλλαγών επεξήγησης η οποία αλλάζει τον τρόπο παρουσίασης της κάθε επεξηγήσης ανάλογα με τις ανάγκες.

Χρησιμοποιώντας ως βάση την γνώση και κάποια άλλα χαρακτηριστικά του χρήστη, η μέθοδος της ταξινόμησης ταξινομεί τα κομμάτια από τα οποία αποτελείται η πληροφορία κάνοντας έτσι πιο εύκολη την ζωή του χρήστη.

### **3.5.1.2 Τεχνικές της προσαρμοστικής παρουσίασης**

Υπάρχουν πέντε είδη τεχνικών για την εφαρμογή των μεθόδων που αναφέρθηκαν παραπάνω (Brusilovsky et.al, 1996,2001). Αυτές είναι :

1. η εισαγωγή / αφαίρεση των κομματιών κειμένου
2. η αλλαγή κομματιών κειμένου
3. το «αναδιπλούμενο» κείμενο
4. η ταξινόμηση των κομματιών κειμένου
5. το «θόλωμα» των κομματιών κειμένου

Η τεχνική της εισαγωγής / αφαίρεσης κομματιών κειμένου ή γνωστή αλλιώς και ως τεχνική του «υπό όρου κειμένου», χρησιμοποιείται όταν θέλουμε να εμφανίσουμε κάποια επιλεγμένα κομμάτια μιας πληροφορίας. Όπως γίνεται εύκολα αντιληπτό η τεχνική αυτή μπορεί να χρησιμοποιηθεί για την εφαρμογή των μεθόδων των επιπρόσθετων, προαπαιτούμενων και συγκριτικών επεξηγήσεων. Σε γενικές γραμμές αυτό που κάνει η τεχνική αυτή είναι να συνδέει κάποιο κομμάτι πληροφορίας με την βοήθεια μιας (*Boolean*) συνθήκης με τα στοιχεία του μοντέλου χρηστών (*user model*). Έτσι κατά την εμφάνιση μιας σελίδας, το σύστημα παρουσιάζει μόνο τα κομμάτια για τα οποία η

(*Boolean*) συνθήκη είναι αληθής (*true*). Το θετικό στοιχείο της τεχνικής αυτής είναι ότι κάποιος μπορεί εύκολα να την καταλάβει και να την μεταβάλλει. Είναι μια τεχνική χαμηλού επιπέδου κάτι το οποίο κάνει το γράψιμο των συνθηκών να φαίνεται περισσότερο σαν διαδικασία προγραμματισμού παρά σαν δημιουργία ενός προσαρμοστικού υπερμέσου. Το γεγονός ότι είναι δύσκολη η χρησιμοποίηση της μεθόδου ταξινόμησης συγκαταλέγεται στα μειονεκτήματα της τεχνικής αυτής.

Με την τεχνική της αλλαγής κομματιών μπορεί να γίνει εύκολα η εφαρμογή της μεθόδου των παραλλαγών επεξήγησης. Με την τεχνική αυτή τα συστήματα των προσαρμοστικών υπερμέσων αποθηκεύουν πολλές παραλλαγές του ίδιου κομματιού πληροφορίας και με βάση το μοντέλου του χρήστη προβάλλουν αυτή που πρέπει.

Η τεχνική του πτυσσόμενου κειμένου χρησιμοποιείται περισσότερο για την εφαρμογή των μεθόδων των επιπρόσθετων, προαπαιτούμενων και συγκριτικών επεξηγήσεων, λιγότερο για την μέθοδο των παραλλαγών επεξήγησης και καθόλου για την μέθοδο της ταξινόμησης.

Η τεχνική της ταξινόμησης κομματιών κειμένου όπως είναι ευλόγως κατανοητό, ταξινομεί τις πληροφορίες από τις πιο σχετικές στις λιγότερο σχετικές χρησιμοποιώντας ως κριτήριο κάποιο ή κάποια από τα χαρακτηριστικά του χρήστη(γνώσεις, απαιτήσεις, υπόβαθρο, κτλ) .

Σκοπός της τεχνικής του «θολώματος» των κομματιών κειμένου είναι να αλλάξει την σημασία ενός συγκεκριμένου κομματιού κάνοντας το να μην είναι σχετικό για τον χρήστη (Hothi and Hall, 1998).

### **3.5.2 Προσαρμοστική πλοήγηση**

Η περιήγηση μας στο διαδίκτυο αλλά και σε άλλα ψηφιακά μέσα κατακλύζεται από ένα μεγάλο πλήθος συνδέσμων, εδώ μπαίνει η ιδέα της προσαρμοστικής πλοήγησης έχει ως σκοπό να προσαρμόσει το σύνολο των υπαρχόντων συνδέσμων με τέτοιο τρόπο ώστε ο χρήστης κάθε φορά να ακολουθεί συνδέσμους που σχετίζονται με το θέμα που επιθυμεί.

### 3.5.2.1 Μέθοδοι της προσαρμοστικής πλοήγησης

Ο Brusilovsky (Brusilovsky, 1996) αναφέρει πέντε μεθόδους προσαρμοστικής πλοήγησης, οι οποίες είναι :

1. ολική πλοήγηση
2. τοπική πλοήγηση
3. ολική υποστήριξη προσανατολισμού
4. τοπική υποστήριξη προσανατολισμού
5. διαχείριση εξατομικευμένων όψεων

Με τον όρο ολική πλοήγηση αναφερόμαστε στο γεγονός ότι το σύστημα μας προτείνει ένα σύνολο συνδέσμων π.χ. σελίδων λέγοντας μας πια είναι η πιο σημαντική.

Με τον όρο τοπική πλοήγηση αναφερόμαστε στο γεγονός ότι το σύστημα μας προτείνει το επόμενο «βήμα» ή *ενέργεια* που πρέπει να κάνουμε. Ένα παράδειγμα τοπικής πλοήγησης είναι η χρήση ενός κουμπιού με την ένδειξη «next».

Όταν το σύστημα μας παρουσιάζει μια περίληψη όλης της δομής ενός υπερχώρου αυτό ονομάζεται ολική υποστήριξη προσανατολισμού η οποία συνήθως παρέχεται με την μορφή ενός πίνακα περιεχομένων, ενός πλαισίου ή μίας σελίδας.

Από την άλλη όταν το σύστημα μας παρουσιάζει ένα τμήμα του υπερχώρου τότε αυτό ονομάζεται τοπική υποστήριξη προσανατολισμού η οποία έχει την μορφή ενός τμήματος της αντίστοιχης υλοποίησης της ολικής υποστήριξης.

Η δημιουργία όψεων μπορεί να βοηθήσει στην περιήγηση ενός χρήστη σε έναν υπερχώρο απλουστεύοντας την πολυπλοκότητα του και ελαχιστοποιώντας το μέγεθος του.

### 3.5.2.2 Τεχνικές της προσαρμοστικής πλοήγησης

Υπάρχουν έξι τεχνικές προσαρμοστικής πλοήγησης (Πρεντζας και Χατζηλυγερούδης, 2001 ):

1. η άμεση πλοήγηση
2. η ταξινόμηση συνδέσμων
3. η απόκρυψη συνδέσμων
  - 3.1.1. απόκρυψη συνδέσμων
  - 3.1.2. διαγραφή συνδέσμων
  - 3.1.3. απενεργοποίηση συνδέσμων
4. ο σχολιασμός συνδέσμων
5. η δημιουργία συνδέσμων
6. η προσαρμογή χαρτών

Με την βοήθεια της άμεσης πλοήγησης το σύστημα χρησιμοποιώντας το μοντέλο χρήστη επιλέγει τον επόμενο «καλύτερο» κόμβο, κάτι το οποίο μπορούμε να χρησιμοποιήσουμε την εφαρμογή της μεθόδου της τοπικής πλοήγησης.

Με την βοήθεια της ταξινόμησης συνδέσμων το σύστημα χρησιμοποιώντας το μοντέλο χρήστη ταξινομεί όλους τους συνδέσμους μιας σελίδας. Η συγκεκριμένη τεχνική μπορεί να χρησιμοποιηθεί για την εφαρμογή της μεθόδου της ολικής πλοήγησης.

Με την βοήθεια της απόκρυψης συνδέσμων ο χώρος πλοήγησης απλοποιείται αφού σύνδεσμοι που οδηγούν σε σελίδες οι οποίες δεν μας ενδιαφέρουν, αποκρύπτονται.

Η βασική ιδέα της τεχνικής της διαγραφής συνδέσμων είναι η διαγραφή των συνδέσμων που οδηγούν σε ένα μη επιθυμητό ή μη επιτρεπτό κείμενο.

Η τεχνική της απενεργοποίησης των συνδέσμων έχει ως στόχο να καταστήσει κάποιους συνδέσμους ανενεργούς. Η τεχνική αυτή λειτουργεί σε συνδυασμό με την απόκρυψη ή τον σχολιασμό των μη ενεργών συνδέσμων.

Η προσθήκη σχολίων σε συγκεκριμένους συνδέσμους βοηθά τον χρήστη στην περιήγηση του. Υπάρχουν τρεις μέθοδοι για την εφαρμογή της τεχνικής σχολιασμού των συνδέσμων :

- Στη μέθοδο αυτή ο σχολιασμός μας δείχνει πόσο σχετικός είναι ο σύνδεσμος με το θέμα, ας πούμε ότι οι σύνδεσμοι με πράσινο χρώμα είναι πιο σχετικοί από τους συνδέσμους με πορτοκαλί χρώμα.
- Σε αυτή την μέθοδο ο σχολιασμός του συνδέσμου μας δείχνει πόσο καλά ξέρει ο χρήστης το περιεχόμενο της σελίδας.
- Στην περίπτωση αυτή ο σχολιασμός μας δείχνει το κατά πόσο είναι έτοιμος ο χρήστης να δει το περιεχόμενο της σελίδας.

Ένα ενδιαφέρον γεγονός είναι ότι αυτές οι τρεις μέθοδοι μπορούν να συνδυαστούν και να χρησιμοποιηθούν ταυτόχρονα.

Η τεχνική της δημιουργίας συνδέσμων μπορεί να υλοποιηθεί με τρεις τρόπους:

- εύρεση νέων χρήσιμων συνδέσμων μεταξύ των κειμένων και προσθήκη τους στην λίστα των ήδη υπαρχόντων συνδέσμων
- δημιουργία συνδέσμων για την ευκολότερη και γρηγορότερη πλοήγηση μεταξύ των περιεχομένων
- δυναμική πρόταση σχετικών συνδέσμων

### **3.6 Τα κύρια τμήματα ενός προσαρμοστικού συστήματος υπερμέσου**

Θα μπορούσαμε να πούμε ότι η δομή των προσαρμοστικών υπερμέσων αποτελείται από τρία τμήματα τα οποία είναι:

- Το πρώτο τμήμα περιλαμβάνει τα συστατικά των υπερμέσων δηλαδή περιέχει την οργάνωση, την αναπαράσταση και την πλοήγηση του χώρου του υπερμέσου.

- Το δεύτερο τμήμα περιλαμβάνει τα συστατικά των ευφυών εκπαιδευτικών συστημάτων. Περιέχει όλες τις «έξυπνες» λειτουργίες και ενέργειες του συστήματος
- Το τμήμα που περιλαμβάνει τα «εποπτικά» συστατικά. Περιέχει τους κανόνες που προσαρμόζουν τις λειτουργίες των δύο προηγούμενων τμημάτων.

### **3.7 Επίλογος**

Πριν από μερικά χρόνια, ο σχεδιασμός των προσαρμοστικών υπερμέσων ήταν μια μεγάλη και χρονοβόρα διαδικασία, η οποία θα μπορούσε να πραγματοποιηθεί μόνο από ειδικευμένες ομάδες με μεγάλη εμπειρία προγραμματισμού. Όμως, η ολοένα αυξανόμενη ανάπτυξη και η διαθεσιμότητα πλαισίων και εργαλείων δημιουργίας έχει ήδη μεταβάλλει την κατάσταση των προηγούμενων χρόνων. Έτσι δίνεται η δυνατότητα ακόμα και σε αρχάριους σχεδιαστές να πειραματιστούν στον σχεδιασμό συστημάτων με την πολυπλοκότητα και την δυσκολία των προσαρμοστικών υπερμέσων. Αυτό έχει σαν αποτέλεσμα την δυνατότητα δημιουργίας νέων συστημάτων που φέρουν νέα χαρακτηριστικά και μεθόδους.

## **4 Ευφυείς πράκτορες**

### **4.1 Εισαγωγικά**

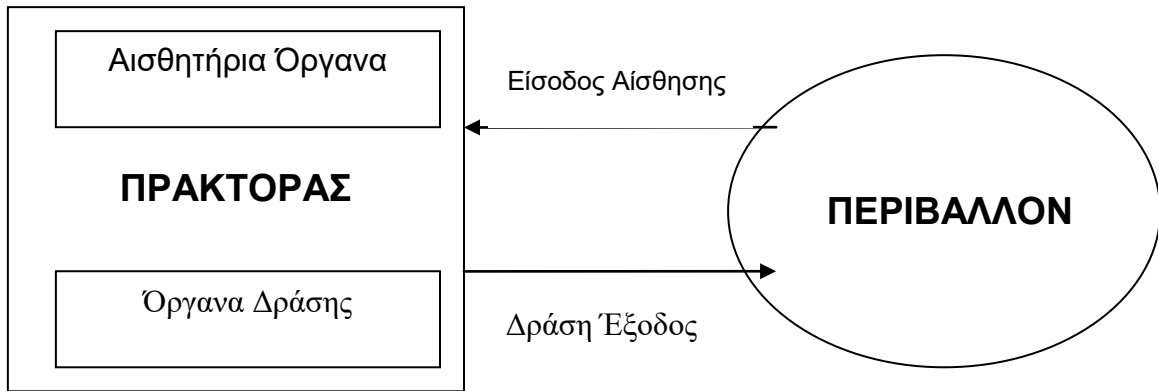
Μια σχετικά νέα έννοια, αυτή του ευφυή πράκτορα και γενικά του πράκτορα αποδεικνύεται όλο και πιο σημαντική για διάφορα ερευνητικά πεδία όπως η τεχνητή νοημοσύνη και η ηλεκτρονική μάθηση. Παρακάτω, αφού γίνει μια προσπάθεια να οριστεί η έννοια του πράκτορα και του ευφυή πράκτορα θα αναφερθούν χαρακτηριστικά και αρχιτεκτονικές τους.

### **4.2 Ορισμός**

Καθώς η έννοια του πράκτορα είναι σχετικά νέα δεν υπάρχει ένας μοναδικός διεθνώς αποδεκτός ορισμός. Αυτό που κάνει τον ορισμό δύσκολο είναι ότι διάφορα χαρακτηριστικά συσχετιζόμενα με ένα πράκτορα και την λειτουργία του είναι εντελώς διαφορετικής σημασίας για διαφορετικούς επιστημονικούς τομείς. Η ικανότητα ενός πράκτορα μπορεί να είναι υψίστης σημασίας για κάποιες εφαρμογές, ενώ για κάποιες άλλες να είναι ασήμαντο έως ανεπιθύμητο.

Παρ' όλο αυτά μπορεί να γίνει μια προσπάθεια ορισμού της έννοιας του πράκτορα. Σύμφωνα, λοιπόν, με τους M.Wooldribge και N.Jennings (Wooldribge and Jennings, 1995): «Ένας πράκτορας είναι ένα υπολογιστικό σύστημα που είναι εγκατεστημένο σε κάποιο περιβάλλον και έχει την ικανότητα αυτόνομης δράσης στο περιβάλλον προκειμένου να ανταποκριθεί στους σκοπούς σχεδίασης του.»

Μία έννοια η οποία είναι άμεσα συνδεδεμένη με αυτή του πράκτορα, είναι η έννοια της αυτονομίας. Με λίγα λόγια είναι η ικανότητα του πράκτορα να λειτουργεί χωρίς καμία εξωτερική παρέμβαση. (Αθανασόγλου ,2009)



**Σχήμα 1.1.** Ένας πράκτορας σε αλληλεπίδραση με το περιβάλλον του. Ο πράκτορας παίρνει αισθητήρια είσοδο από το περιβάλλον του και παράγει σαν έξοδο δράσεις που το επηρεάζουν (Πηγή: Multiagent Systems, MIT Press, 1999).

Όσο αναφορά τους ευφυείς πράκτορες ο M. Wooldridge και N. R. Jennings (1995) υποστηρίζουν ότι *«ευφυής πράκτορας είναι αυτός που έχει την ικανότητα ευέλικτης αυτονομίας δράσης προκειμένου να διεκπεραιώσει τους σκοπούς σχεδίασης του»*.

### 4.3 Επιπλέον ορισμοί

Πράκτορας κατά **Russel** και **Norvig**:

“Πράκτορας είναι οτιδήποτε μπορεί να αντιληφθεί το περιβάλλον του μέσω αισθητήρων και να αντιδράσει πάνω στο περιβάλλον μέσω μηχανισμών δράσης”

Πράκτορας κατά Hayes-Roth

Οι ευφυείς πράκτορες κάνουν συνεχώς τις εξής τρεις λειτουργίες:

1. αντιλαμβάνονται τις δυναμικές συνθήκες του περιβάλλοντος
2. δρουν πάνω στο περιβάλλον ώστε να το αλλάξουν
3. συλλογίζονται ώστε να ερμηνεύσουν αυτά που αντιλαμβάνονται, να λύσουν προβλήματα, να συμπεράνουν και να καθορίσουν τη δράση τους



#### 4.4 Χαρακτηριστικά Πρακτόρων

Όπως είπαμε δεν μπορούμε να δώσουμε έναν αυστηρό ορισμό του ευφυή πράκτορα, αυτό που μπορούμε να κάνουμε όμως είναι να καθορίσουμε κάποια ουσιαστικά χαρακτηριστικά τους. Έτσι διάφοροι ερευνητές, δίνοντας βάρος σε διαφορετικές πτυχές της έννοιας ανάλογα με το πρόβλημα που προσπαθούν να επιλύσουν, κάνουν λόγο για "ευφυείς πράκτορες", για "αυτόνομους πράκτορες", γενικότερα για "υπολογιστικούς πράκτορες" ή για συνδυασμούς των όρων αυτών. (Hayes-Roth B., 1990; Maes P., 1994; Hedberg S. R., 1995; Jennings N.R. & Wooldridge M., 1996; Sycara K. & Zeng D., 1996).

Τα χαρακτηριστικά αυτά είναι τα εξής:

- **Αυτονομία (*autonomy*)**, δηλαδή το να λειτουργεί η οντότητα χωρίς εξωτερική παρέμβαση.
- **Προνοητικότητα ή προενεργητικότητα (*pro-activeness*)**, δηλαδή το να σκέφτεται η οντότητα μακροπρόθεσμα και να λαμβάνει πρωτοβουλίες για ενέργειες οι οποίες θα την βοηθήσουν στην περάτωση του στόχου της.
- **Αντιδραστικότητα (*reactivity*) ή Αποκρισιμότητα (*responsiveness*)**, δηλαδή η ικανότητα γρήγορης και «σωστής» απόκρισης στις αλλαγές του περιβάλλοντός της.
- **Κοινωνική ικανότητα (*social ability*)**, δηλαδή η ικανότητα αλληλεπίδρασης με άλλους πράκτορες και ανθρώπους.

Εκτός από τα παραπάνω χαρακτηριστικά τα οποία χαρακτηρίζονται ως κύρια υπάρχουν και μερικά δευτερεύοντα και συχνά συνθέτουν τον «χαρακτήρα» ενός πράκτορα, μερικά από αυτά είναι τα εξής:

- η ειλικρίνεια δηλαδή ο πράκτορας προσπαθεί να λέει και κάνει αυτό που πιστεύει
- η καλή προαίρεση, ο πράκτορας πάντα ξεκινά με σκοπό να βοηθήσει
- η λογική, ο πράκτορας δεν θα κάνει άσκοπες πράξεις θα κάνει μόνο αυτές που θα τον οδηγήσουν στον στόχο του.

## 4.5 Αρχιτεκτονικές Ευφυών Πρακτόρων

Η αρχιτεκτονική ενός πράκτορα, καθορίζει από ποια μέρη θα αποτελείται και πως αυτά τα μέρη θα συνεργάζονται μεταξύ τους. (Αθανασόγλου ,2009) Το γεγονός αυτό με την σειρά του θα καθορίσει τις ενέργειες του πράκτορα και τη μελλοντική του κατάσταση, σε συνάρτηση με το ποια είναι τα δεδομένα που προσλαμβάνονται στα αισθητήρια όργανα του πράκτορα και το ποια είναι η τωρινή του εσωτερική κατάσταση.

Η συμπεριφοράς ενός πράκτορα μπορεί να χαρακτηριστεί ως μια συνεχή ακολουθία λειτουργιών αντίληψης, νοητικής διεργασίας και δράσης, κάτι που αποτελεί βασικό χαρακτηριστικό σχεδιασμού όσον αφορά την αρχιτεκτονική του. Ένα ευρέως χρησιμοποιούμενο αρχιτεκτονικό μοντέλο προτρέπει οι λειτουργίες αυτές να πραγματοποιούνται από αντίστοιχα τμήματα λογισμικού, τα οποία μπορούν να χαρακτηριστούν ως νοητικό τμήμα, και όργανα δράσης (perception – cognition – action) (Hayes-Roth, 1990,1995). Το νοητικό τμήμα είναι υπεύθυνο για την υλοποίηση τις συμπερασματικής διαδικασίας, δηλαδή ποιες ενέργειες θα εκτελεστούν, ποια σχέδια δράσης θα διαμορφωθούν κτλ. Όπως επίσης είναι το τμήμα το οποίο είναι υπεύθυνο για την δυνατότητα που έχει ο πράκτορας να «μαθαίνει».

Διάφορες αρχιτεκτονικές προσεγγίσεις για την σχεδίαση και υλοποίηση ευφυών πρακτόρων είναι οι εξής:

- **Ορθολογικοί Πράκτορες (*Rational Agents*):** η λήψη αποφάσεων δράσης γίνεται μέσω λογικής επαγωγής.
- **Αντιδραστικοί πράκτορες (*Reactive Agents*):** οι αποφάσεις λαμβάνονται μέσω απευθείας αντιστοιχίας από κατάσταση σε δράση.
- **BDI Πράκτορες ή Πράκτορες βασισμένοι σε Πιστεύω – Επιθυμίες - Προθέσεις (*Belief-Desire-Intention, BDI Agents*):** η λήψη αποφάσεων γίνεται με την διαχείριση ειδικών δομών που αναπαριστούν τα πιστεύω, τις επιθυμίες και τις προθέσεις του πράκτορα.
- **Διαστρωματομένη Αρχιτεκτονική (*Layered Agents*):** η λήψη αποφάσεων γίνεται μέσω διαφορετικών επιπέδων λογισμικού, το κάθε ένα από τα οποία αποτελεί κρίση του περιβάλλοντος σε διαφορετικό επίπεδο αφαίρεσης.

### 4.5.1 Ορθολογικοί Πράκτορες (Rational Agents)

Για να αποκτήσει το σύστημα μας ευφυή συμπεριφορά θα πρέπει να του παρέχουμε τα εξής:

- μια συμβολική αναπαράσταση τόσο του περιβάλλοντος του
- μια συμβολική αναπαράσταση της επιθυμητής συμπεριφοράς του
- ένα σύστημα συντακτικής διαχείρισης των παραπάνω αναπαραστάσεων

Σε γενικές γραμμές Ορθολογικός Πράκτορας είναι ένας πράκτορας που ενεργεί έτσι ώστε να επιτυγχάνει το καλύτερο αποτέλεσμα ή όταν υπάρχει αβεβαιότητα, το καλύτερο αναμενόμενο αποτέλεσμα.

Με την βοήθεια παραδοσιακών μαθηματικών τεχνικών, έγινε εφικτή η μοντελοποίηση τέτοιων συστημάτων. Έτσι για τις συμβολικές αναπαραστάσεις χρησιμοποιούνται *λογικές φόρμουλες* ενώ για την συντακτική διαχείριση τους *λογικά συμπεράσματα* ή *απόδειξη θεωρήματος*.

Η ιδέα της δημιουργίας πρακτόρων οι οποίοι θα δρούσαν βάση αποδείξεων θεωρημάτων υπήρξε ελκυστική. Έτσι το επόμενο βήμα ήταν να δημιουργηθούν φορμαλιστικές θεωρίες που περιγράφουν πως θα πρέπει να συμπεριφέρεται ένας ευφυής πράκτορας.

### 4.5.2 Αντιδραστικοί πράκτορες (Reactive Agents)

Η αναζήτηση εναλλακτικών προσεγγίσεων στην συμβολική τεχνητή νοημοσύνη οδήγησε στην ανάπτυξη της αρχιτεκτονικής των αντιδραστικών πρακτόρων. Ως αποτέλεσμα αυτής της αναζήτησης ήταν να προκύψουν διάφορες προσεγγίσεις, οι οποίες αν και δύσκολο να κατηγοριοποιηθούν είχαν κάποια κοινά χαρακτηριστικά που αποτέλεσαν την βάση για την δημιουργία της αρχιτεκτονικής των αντιδραστικών πρακτόρων. Αυτά ήταν:

- Η απόρριψη των συμβολικών αναπαραστάσεων και της εξαγωγής αποφάσεων από την συντακτική διαχείριση τους.
- Η ιδέα ότι η ευφυής και λογική συμπεριφορά για ένα πράκτορα είναι ενδογενώς συνδεδεμένη με το *περιβάλλον* του και κατά συνέπεια αποτέλεσμα της *αλληλεπίδρασης* του πράκτορα με το περιβάλλον του.
- Η ιδέα ότι η ευφυής συμπεριφορά έρχεται σαν αποτέλεσμα αλληλεπίδρασης επιμέρους απλών συμπεριφορών.

Σε αυτή την αρχιτεκτονική προσέγγιση θα ασχοληθούμε με δυο κύρια χαρακτηριστικά.

Κατά το πρώτο χαρακτηριστικό η συνεχής αλληλεπίδραση του πράκτορα με το περιβάλλον του έχει ως αποτέλεσμα την λήψη κάποιων αποφάσεων σχετικά με την δράση του. Η δράση του πράκτορα είναι αποτέλεσμα μιας συνάρτησης που αντιστοιχίζει την τρέχουσα είσοδο σε κίνηση (αντίδραση).

Κατάσταση  $\longrightarrow$  Κίνηση

Το γεγονός ότι σε μια κατάσταση είναι δυνατόν να ενεργοποιηθούν παραπάνω από μία συναρτήσεις που θα δώσουν την επόμενη κίνηση (αντίδραση) για τον πράκτορα κάνει επιτακτική την ανάγκη ενός μηχανισμού ο οποίος ανάλογα με τον σκοπό σχεδίασης του κάθε πράκτορα θα επιλέγει την επόμενη κίνηση βάση κάποιων κριτηρίων όπως κόστος, ταχύτητα, κτλ. Αυτό αποτελεί το δεύτερο κύριο χαρακτηριστικό.

### 4.5.3 Η Αρχιτεκτονική BDI

Η αρχιτεκτονική των Πιστεύω-Επιθυμιών-Προθέσεων (BDI: Belief-Desire-Intention) είναι στενά συνδεδεμένη με την έννοια της πρακτικής κρίσης όπου: πρακτική κρίση είναι η διαδικασία να αποφασίζεις κάθε στιγμή πια κίνηση πρέπει να κάνεις για να πετύχεις τους στόχους σου. Αυτό γίνεται με τον να αποφασίσεις ποιους σκοπούς θες να πετύχεις και πως θα τους πετύχεις

Το πρώτο είναι γνωστό σαν *σκέψη (deliberation)*, ενώ το δεύτερο σαν *κρίση μέσων-ολοκλήρωσης (means-ends reasoning)*.

Για να γίνει ευκολότερη η κατανόηση του μοντέλου BDI θα παρατεθεί ένα απλό παράδειγμα πρακτικής κρίσης. Έστω ότι κάποιος τελειώνει το λύκειο και βρίσκεται έτοιμος να αποφασίσει τι θα σπουδάσει. Η διαδικασία απόφασης συνήθως ξεκινά με την προσπάθεια εντοπισμού των πιθανών διαθέσιμων εκδοχών. Για παράδειγμα αν ο μαθητής έχει καλούς βαθμούς μια εκδοχή είναι να σπουδάσει ιατρική. Μετά την δημιουργία του συνόλου εναλλακτικών θα πρέπει να γίνει επιλογή κάποιων για να επιχειρηθούν. Οι επιλεγμένες εκδοχές γίνονται προθέσεις, οι οποίες καθορίζουν τις κινήσεις του πράκτορα. Οι προθέσεις αυτές θα επανατροφοδοτήσουν την μελλοντική πρακτική κρίση του πράκτορα. Αν ο πράκτορας αποφασίσει να σπουδάσει ιατρική θα πρέπει να αφιερώσει χρόνο και προσπάθεια γι' αυτό τον σκοπό στο μέλλον.

Οι προθέσεις παίζουν καθοριστικό ρόλο στην διαδικασία κρίσης. Συνήθως οι προθέσεις οδηγούν σε δράση. Αν πραγματικά ο πράκτορας έχει πρόθεση να σπουδάσει ιατρική, αναμένεται απ' αυτόν να δρα με βάση αυτή την πρόθεση για να την πετύχει. Για παράδειγμα, αναμένεται να διαβάζει πολλές ώρες. Επίσης, αν μια ακολουθία κινήσεων αποτύχει αναμένεται από τον πράκτορα να ξαναπροσπαθήσει (δεν εγκαταλείπει στην πρώτη αποτυχία). Έτσι, αν ο πράκτορας δεν συγκεντρώσει τα απαιτούμενα μόρια εισαγωγής για την ιατρική, θα ξαναπροσπαθήσει.

Επιπλέον, από την στιγμή που ο πράκτορας έχει κάποια πρόθεση, αυτό θα περιορίζει την μελλοντική του πρακτική κρίση. Δηλαδή, εφόσον υπάρχει πρόθεση για επίτευξη κάποιου στόχου, ενδεχόμενες κινήσεις, οι οποίες έρχονται σε αντιπαράθεση με την υπάρχουσα πρόθεση, δεν θα γίνονται αποδεκτές. Για παράδειγμα, όταν ο πράκτορας σκοπεύει (έχει πρόθεση) να σπουδάσει ιατρική (απαιτούνται αρκετές ώρες διαβάσματος), η εκδοχή των καθημερινών πάρτι αποκλείεται, γιατί τα δύο αυτά γεγονότα (ενδεχόμενα) είναι αμοιβαία αποκλειόμενα. Ένα άλλο σημαντικό χαρακτηριστικό των προθέσεων είναι ότι επιμένουν στην επίτευξή τους. Έτσι, ο πράκτορας που έχει αποφασίσει να σπουδάσει ιατρική θα επιμένει σ' αυτήν την προσπάθεια αφιερώνοντας χρόνο και διαθέσιμους πόρους. Παρόλα αυτά η επιμονή σε μια πρόθεση δεν πρέπει να ξεπερνά κάποια όρια-αν γίνει ξεκάθαρο ότι ο πράκτορας δεν μπορεί να σπουδάσει ιατρική θα εγκαταλείψει την πρόθεση αυτή. Επίσης, όταν οι λόγοι υιοθέτησης μιας πρόθεσης πάψουν να

υπάρχουν και σ' αυτή την περίπτωση ο πράκτορας την εγκαταλείπει. Για παράδειγμα, αν ο λόγος επιλογής της ιατρικής ήταν ότι θεωρήθηκε εύκολη περίπτωση ζωής, αλλά στην συνέχεια ανακαλύπτεται ότι συμπεριλαμβάνει επιβαρημένο ωράριο (ανεπιθύμητη διαδικασία για τον πράκτορα), τότε ο λόγος υιοθέτησης της πρόθεσης παύει να ισχύει και η συγκεκριμένη πρόθεση είναι λογικό να εγκαταλειφθεί.

Τέλος, οι προθέσεις είναι στενά συνδεδεμένες με τα πιστεύω για το μέλλον. Αν ο πράκτορας έχει πρόθεση να σπουδάσει ιατρική τότε θα πιστεύει ότι σπουδάσει ιατρική. Θα ήταν παράλογο για κάποιο πράκτορα να υιοθετήσει μια πρόθεση που δεν πιστεύει πραγματικά ότι μπορεί να υιοθετήσει μια πρόθεση που δεν πιστεύει πραγματικά ότι μπορεί να τον οδηγήσει σε επίτευξη στόχου. Έχοντας σαν πρόθεση (ο πράκτορας) να σπουδάσει ιατρική, θα πρέπει τουλάχιστον να πιστεύει ότι υπάρχει μια καλή πιθανότητα γι' αυτό.

Σύμφωνα με τον Wooldridge (Wooldridge, 2000) οι προθέσεις παίζουν σημαντικό ρόλο στην πρακτική κρίση:

- Οι **προθέσεις** οδηγούν τα *μέσα-ολοκλήρωσης* κρίσης. Αν ο πράκτορας έχει ξεκαθαρίσει την πρόθεση που να σπουδάσει ιατρική, τότε θα προσπαθήσει να το πετύχει, πράγμα που σημαίνει (μεταξύ άλλων) ότι θα αποφασίσει το *πώς* θα το πετύχει (π.χ. διαβάζοντας αρκετές ώρες).
- Οι **προθέσεις** περιορίζουν την μελλοντική *σκέψη*. Ο πράκτορας που έχει πρόθεση να γίνει ακαδημαϊκός δεν θα εξετάσει εκδοχές που δεν συμφωνούν με την υπό διεκπεραίωση πρόθεση. Για παράδειγμα ένας λογικός πράκτορας δεν θα εξετάσει το ενδεχόμενο να γίνει τραγουδιστής ενώ παράλληλα έχει ξεκινήσει την διεκπεραίωση της πρόθεσης του να σπουδάσει ιατρική.
- Οι **προθέσεις** επιμένουν να πραγματοποιηθούν. Κατά κανόνα μια πρόθεση δεν εγκαταλείπεται χωρίς κάποιο ισχυρό λόγο. Μια πρόθεση εγκαταλείπεται είτε επειδή ο πράκτορας πιστεύει ότι την έχει φέρει σε πέρας, είτε γιατί έχει πειστεί ότι δεν πρόκειται να την επιτύχει, είτε γιατί οι λόγοι επιδίωξης της δεν ισχύουν πια.
- Οι **προθέσεις** επηρεάζουν τα *πιστεύω*, στα οποία βασίζεται η μελλοντική πρακτική κρίση. Πιστεύοντας ένας πράκτορας ότι θα σπουδάσει ιατρική μπορεί να σχεδιάζει μελλοντικές κινήσεις θεωρώντας το δεδομένο.

Το γεγονός ότι ο πράκτορας καμιά φορά θα πρέπει να εγκαταλείπει μια πρόθεση του είναι ένα σύννηθες πρόβλημα στην σχεδίαση ευφυών πρακτόρων που υλοποιούν πρακτική κρίση. Έτσι κατά διαστήματα ο πράκτορας είναι καλό να σταματάει και να επανεξετάζει τις προθέσεις του, αυτό όμως έχει σίγουρα κάποιο κόστος, τόσο σε χρόνο όσο και σε υπολογιστικούς πόρους. Έτσι, γενικά έχουν παρατηρηθεί οι εξής συμπεριφορές:

- Από την μια έχουμε πράκτορες που δεν σταματούν να επανεξετάσουν της προθέσεις τους άρα συνεχίζουν να προσπαθούν για την πραγματοποίησή τους χωρίς να υπάρχει λόγος.
- Από την άλλη έχουμε πράκτορες που σταματάνε την δράση τους πολύ συχνά για να επανεξετάσουν τις προθέσεις και με αυτό τον τρόπο ξοδεύουν χρόνο και πόρους, ριψοκινδυνεύοντας μάλιστα να μην πραγματοποιήσουν ποτέ τις προθέσεις τους λόγω καθυστέρησης.

Αυτό που πρέπει να γίνει είναι να βρεθεί μια ιδανική συμπεριφορά η οποία θα εξισορροπεί τις παραπάνω δυο συμπεριφορές.

Πάνω σε αυτή την ιδέα εργάστηκαν ο David Kinny και ο Michael Georgeff, κάνοντας μια σειρά πειραμάτων με μια πλατφόρμα BDI πρακτόρων που ονόμασαν dMARS (Kinny D. and Georgeff M., 1991). Το συμπέρασμα αυτής της μελέτης ήταν ότι οι πράκτορες οι οποίοι δεν σταματούν για να επανεξετάσουν τις προθέσεις τους δηλαδή οι «τολμηροί» πράκτορες αλλά και οι πράκτορες που επανεξέταζαν πολύ συχνά τις προθέσεις τους δηλαδή οι «προσεκτικοί» πράκτορες ήταν ικανοί να δρουν και να αποδίδουν σε διάφορα περιβάλλοντα.

Πολύ σημαντική παράμετρος των πειραμάτων τους ήταν ο ρυθμός αλλαγής του κόσμου  $\gamma$  (με  $\gamma$  συμβολίζεται ο ρυθμός αλλαγής του περιβάλλοντος στο οποίο βρίσκεται ο πράκτορας). Τα κύρια αποτελέσματα των Kinny και Georgeff έχουν ως εξής:

- Όταν ο ρυθμός  $\gamma$  είναι χαμηλός που σημαίνει ότι το περιβάλλον δεν αλλάζει γρήγορα, οι τολμηροί πράκτορες αποδίδουν καλύτερα συγκριτικά με τους

προσεκτικούς και αυτό γιατί δεν χρειάζεται να σταματάνε συχνά έτσι ώστε να επανεξετάσουν τις προθέσεις τους.

- Όταν ο ρυθμός  $\gamma$  είναι υψηλός δηλαδή το περιβάλλον αλλάζει συχνά, τότε αποδίδουν ικανοποιητικότερα οι προσεκτικοί πράκτορες από ότι οι τολμηροί, και αυτό γιατί οι τολμηροί συνεχίζουν να προσπαθούν να επιτύχουν ένα στόχο που καμιά φορά έχει πάψει να έχει νόημα. Τέλος το γεγονός ότι οι προσεκτικοί πράκτορες σταματάνε συχνά για να επανεξετάσουν τις προθέσεις τους, τους κάνει να έχουν μια αντικειμενικότερη άποψη για το περιβάλλον τους καθώς αντιλαμβάνονται καλύτερα τυχόν νέα δεδομένα.

Το συμπέρασμα είναι ότι κάθε τύπος περιβάλλοντος απαιτεί και τον δικό του τύπο πράκτορα, έτσι στατικά περιβάλλοντα απαιτούν προσεκτικούς πράκτορες ενώ δυναμικά περιβάλλοντα απαιτούν τολμηρούς πράκτορες.

#### **4.5.4 Διαστρωματωμένη Αρχιτεκτονική ή Αρχιτεκτονική Πολλαπλών Επιπέδων (Layered Agents)**

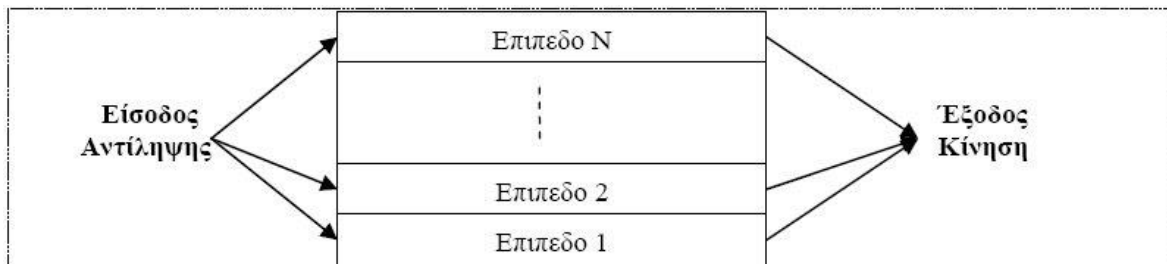
Πρόκειται ουσιαστικά για μια ομάδα αρχιτεκτονικών όπου διάφορα υποσυστήματα συντίθεται σαν μια ιεραρχία επιπέδων. Στην διαστρωματωμένη αρχιτεκτονική κάθε επίπεδο είτε αποτελεί μια ξεχωριστή συνάρτηση αντιστοίχισης της αντίληψης (εισόδου) σε κίνηση, είτε αποτελεί ένα φίλτρο επεξεργασίας της αντίληψης μέχρι την εξαγωγή της τελικής απόφασης κίνησης.

Δύο είναι οι κύριες προσεγγίσεις διαστρωματωμένων αρχιτεκτονικών:

*Οριζόντιας Διαστρωμάτωσης* (Σχήμα 1.2), όπου κάθε επίπεδο λογισμικού είναι απευθείας συνδεδεμένο τόσο με το λογισμικό που είναι υπεύθυνο για την αντίληψη, όσο και με το λογισμικό που είναι υπεύθυνο για την επόμενη κίνηση. Στην προσέγγιση αυτή κάθε επίπεδο λογισμικού αποτελεί και μια διαφορετική εκδοχή συνάρτησης που αντιστοιχίζει την δεδομένη είσοδο αντίληψης σε κάποια κίνηση (έξοδο). Αυτός που αποφασίζει ποια θα είναι η επόμενη κίνηση που θα



επιλεγεί ανάμεσα σε όλες τις διαφορετικές εκδοχές, είναι το λογισμικό εξόδου το οποίο με βάση κάποια κριτήρια κάνει το τελικό φιλτράρισμα και επιλέγει την επόμενη κίνηση.



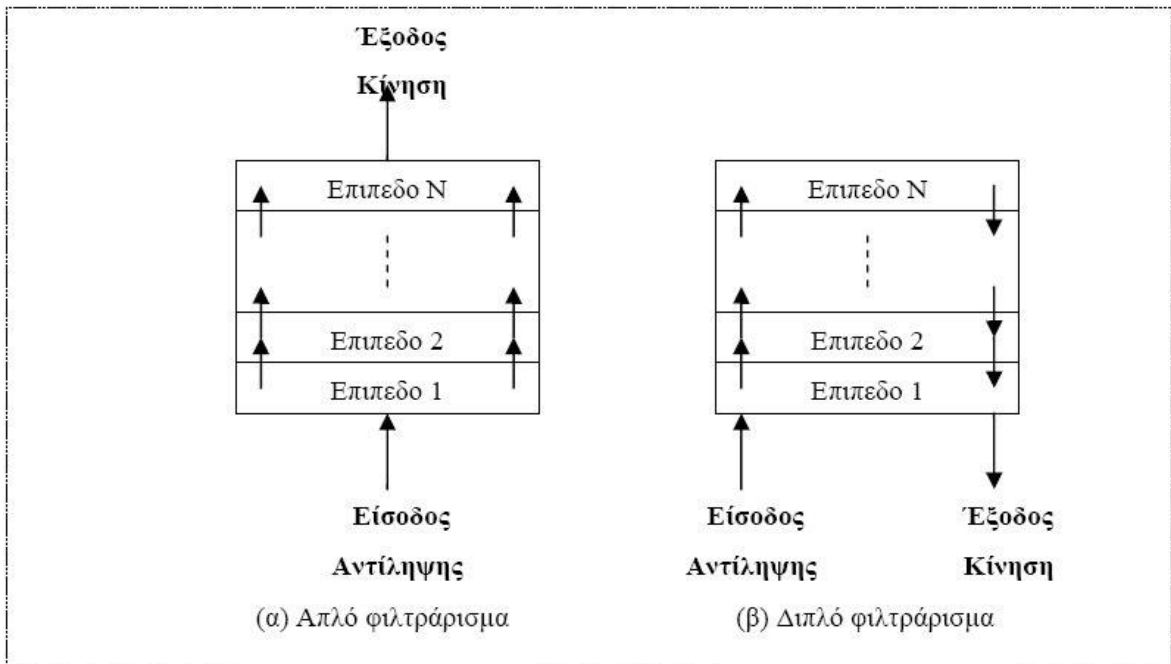
**Σχήμα 1.2.** Αρχιτεκτονική Πολλαπλών Επιπέδων Οριζόντιας διαστρωμάτωσης. (Πηγή: Wooldridge and Jennings, eds, Intelligent Agents: Theories, Architectures and Languages)

*Κάθετης Διαστρωμάτωσης* (Σχήμα 1.3), όπου κάθε επίπεδο λογισμικού επικοινωνεί άμεσα με το προηγούμενο και το επόμενο (εφόσον υπάρχει). Στην περίπτωση αυτή κάθε επίπεδο είναι και ένα φίλτρο που δέχεται είσοδο από το προηγούμενο του και δίνει την έξοδο του στο επόμενο.

Διακρίνονται δυο περιπτώσεις:

Αυτή του απλού φιλτραρίσματος, όπου το φιλτράρισμα ξεκινά από το πρώτο επίπεδο και ολοκληρώνεται με την έκδοση της επόμενης κίνησης από το τελευταίο επίπεδο.

Αυτή του διπλού φιλτραρίσματος, όπου το τελευταίο επίπεδο ανατροφοδοτεί το προηγούμενό του κ.ο.κ. και το πρώτο επίπεδο είναι εκείνο έχει ως έξοδο την επόμενη κίνηση.



**Σχήμα 1.3.** Αρχιτεκτονική Πολλαπλών Επιπέδων Κάθετης διαστρωμάτωσης. (Πηγή: Wooldridge and Jennings, eds, Intelligent Agents: Theories, Architectures and Languages)

## **4.6 Ευφυείς πράκτορες στην εκπαίδευση**

Η χρήση των πρακτόρων σε περιβάλλοντα εκπαίδευσης αποδεικνύεται εξαιρετικά χρήσιμη καθώς μπορούν σε ικανοποιητικό βαθμό να εξαλείψουν τα συνήθη προβλήματα στα περιβάλλοντα αυτά.(Aroyo and Kommers, 1999) Κατά την σχεδίαση εκπαιδευτικών συστημάτων που χρησιμοποιούν πράκτορες δίνεται μεγάλο βάρος στην προσαρμοστικότητα. Σκοπός είναι να προσαρμόζεται το σύστημα στις ανάγκες και προτιμήσεις του χρήστη, να εξατομικεύεται δηλαδή. Συνέπεια των παραπάνω είναι ένα τέτοιο σύστημα να υπερτερεί σε αποτελεσματικότητα και ευελιξία από ένα σύστημα που παρέχει το ίδιο περιεχόμενο σε όλους τους χρήστες. Τα εκπαιδευτικά αυτά συστήματα χρησιμοποιούν κάποια γνώσιμα του χρήστη για να προσαρμόσουν την συμπεριφορά τους, τέτοια γνώσιμα μπορεί να είναι : η γνώση, στόχοι, υπόβαθρο και οι προτιμήσεις του χρήστη. Μόλις εντοπιστούν αυτά το επόμενο βήμα είναι να βρεθεί ένας τρόπος έκφρασης της προσαρμοστικότητας στο σύστημα. Υπάρχουν κάποια στοιχεία του συστήματος τα οποία μπορούν να αλλάζουν ανάλογα με τον χρήστη, τέτοια στοιχεία είναι :

- η σειρά παρουσίασης της εκπαιδευτικής ύλης (curriculum sequencing)
- η ανάλυση των λύσεων των μαθητών (intelligent analysis of student solutions)
- η βοήθεια με αλληλεπίδραση κατά την επίλυση προβλημάτων (interactive problem solving support)
- τρόπος παρουσίασης του περιεχομένου (adaptive content presentation)
- ο τρόπος πλοήγησης (adaptive navigation support)
- η υποστήριξη συνεργασίας ανάμεσα στους μαθητές (adaptive collaboration support) (Brusilovsky, 1998)

Έτσι με την αλλαγή στην σειρά παρουσίασης του υλικού επιδιώκουμε να βρούμε για τον κάθε μαθητή την πλέον κατάλληλη σειρά εκπαιδευτικού υλικού (παραδείγματα, ερωτήσεις, προβλήματα) με την παρακολούθηση της οποίας θα βελτιστοποιηθεί η μαθησιακή διαδικασία.

Η ευφυής ανάλυση των λύσεων των μαθητών ασχολείται με τις τελικές απαντήσεις των μαθητών σε εκπαιδευτικά προβλήματα και έχει ως στόχο να

εντοπίσει ποιο ακριβώς είναι το λάθος , καθώς και την ελλιπή γνώση που οδήγησε σε εσφαλμένη απάντηση.

Ο στόχος της βοήθειας με αλληλεπίδραση είναι, μέσω των ευφυών πρακτόρων να παρακολουθεί στενά τις ενέργειες του μαθητή και να του παρέχει βοήθεια σε κάθε βήμα της επίλυσης ενός προβλήματος.

Ο στόχος της προσαρμοστικότητας στην παρουσίαση είναι να προσαρμόζεται το περιεχόμενο της σελίδας στον χρήστη ανάλογα με κάποια χαρακτηριστικά του όπως οι στόχοι, η γνώση και άλλες πληροφορίες που είναι αποθηκευμένες στο προφίλ του.

Ο στόχος της προσαρμοστικότητας στην πλοήγηση είναι να προσαρμοστούν οι σύνδεσμοι των σελίδων που θα επισκεφτεί ο μαθητής με τέτοιο τρόπο έτσι ώστε να δημιουργήσουν ένα 'μονοπάτι' το οποίο θα είναι το πιο κατάλληλο για τις ανάγκες του.

Η υποστήριξη συνεργασίας ανάμεσα στους μαθητές έχει ως στόχο να δημιουργήσει ομάδες συνεργασίας χρησιμοποιώντας την γνώση του συστήματος για τους χρήστες η οποία βρίσκεται αποθηκευμένη στα μοντέλα μαθητών.

Τέλος κάτι το οποίο παρουσιάζει ενδιαφέρον και δεν θα μπορούσε να παραληφθεί είναι η μοντελοποίηση των μαθητών με σκοπό να δημιουργηθούν τα αντίστοιχα προφίλ μαθητών. Συνήθως κάθε μοντέλο μαθητή χωρίζεται σε δυο τμήματα, το πρώτο περιλαμβάνει κάποιες γενικές πληροφορίες για τον μαθητή όπως το όνομα του και τα ενδιαφέροντα του, ενώ το δεύτερο περιλαμβάνει πιο σημαντικές πληροφορίες όπως η γνώση του έτσι όπως αποθηκεύεται από το σύστημα κατά την αλληλεπίδραση του χρήστη με αυτό. (Vozalis et al, 2001)

#### **4.7 Ευφυή εκπαιδευτικά συστήματα**

Παρακάτω θα παρατεθούν μερικά παραδείγματα εκπαιδευτικών συστημάτων τα οποία είτε χρησιμοποιούν ευφυείς πράκτορες είτε διαθέτουν κάποιο είδος ευφυΐας.

#### 4.7.1 MATS (Multi-Agent Tutoring System)

Το σύστημα MATS (K. Solomos and N. Avouris, 1999) το οποίο είναι ένα ανοικτό κατανεμημένο εκπαιδευτικό σύστημα πολλαπλών πρακτόρων και βασίζεται σε μια αρχιτεκτονική η οποία χρησιμοποιεί πολλαπλούς πράκτορες και υποστηρίζει εκπαιδευτικές μονάδες οι οποίες λειτουργούν στο χώρο του διαδικτύου και έχουν την δυνατότητα να συνεργάζονται για να πετύχουν ένα συγκεκριμένο εκπαιδευτικό στόχο. Οι πράκτορες κινούνται ελεύθερα στο χώρο του διαδικτύου και περιλαμβάνουν διαφορετικές γνώσεις. Το ζητούμενο είναι η κατασκευή μηχανισμών ανταλλαγής γνώσης (knowledge interchange mechanisms) για αυτούς τους πράκτορες, οι οποίοι διακρίνονται σε δυο βασικές κατηγορίες:

- στους πράκτορες διαμεσολάβησης (Broker Agents)
- στους εκπαιδευτικούς πράκτορες (Tutoring Agents)

Σε ένα τυπικό σενάριο λειτουργίας του συστήματος, οι πράκτορες διαμεσολάβησης περιμένουν να δεχτούν αιτήσεις από τους εκπαιδευτικούς πράκτορες. Κάθε εκπαιδευτικός πράκτορας στέλνει τα χαρακτηριστικά του γνωρίσματα στον πλησιέστερο πράκτορα διαμεσολάβησης. Έτσι όταν ο μαθητής έρχεται σε επαφή με κάποιον εκπαιδευτικό πράκτορα, αυτός θα προχωρήσει στην εκπαίδευση του μαθητή με βάση τις γνώσεις που διαθέτει. Αν σε κάποιο σημείο ο εκπαιδευτικός πράκτορας διαπιστώσει ότι δεν μπορεί να ανταπεξέλθει στις απαιτήσεις του μαθητή, θα στείλει ένα μήνυμα για αναζήτηση συνεργασίας στον πράκτορα διαμεσολάβησης. Το μήνυμα αυτό θα περιλαμβάνει πληροφορίες σχετικά με το συγκεκριμένο μαθητή καθώς και τις γνώσεις που αυτός αναζητά. Ο πράκτορας διαμεσολάβησης μόλις δεχτεί την αίτηση θα διατρέξει όλα τα μοντέλα των εκπαιδευτικών πρακτόρων που έχει αποθηκευμένα και θα επιλέξει το καταλληλότερο με βάση τις γνώσεις του αλλά και τη διαθεσιμότητα του. Υπάρχει περίπτωση ένας εκπαιδευτικός πράκτορας να δεχτεί ταυτόχρονα περισσότερες προτάσεις για συνεργασία, έτσι χρησιμοποιείται ένας αλγόριθμος επιλογής, για να προκύψει τελικά η αίτηση που θα ικανοποιηθεί.

#### **4.7.2 Ένας πράκτορας διαμεσολάβησης για εκπαιδευτικά συστήματα βοηθούμενα από υπολογιστές**

Το σύστημα των Dang, Ghenniwa και Kamel (Dang et al, 1999) έχει ως σκοπό να δημιουργήσει ένα περιβάλλον μάθησης και συνεργασίας ανάμεσα στο μαθητή και το εκπαιδευτικό λογισμικό. Προτείνει την ύπαρξη ενός πράκτορα διαμεσολάβησης που πρέπει να είναι αυτόνομος, δυναμικός, οδηγούμενος από τους στόχους του και συνεργατικός. Ο ρόλος αυτού του πράκτορα είναι να δρα σαν ένας διαμεσολαβητής ανάμεσα στο μαθητή και το εκπαιδευτικό σύστημα και όταν αυτό κρίνεται αναγκαίο να έρχεται σε επαφή και με τους δυο, ενώ παράλληλα θα παρακολουθεί τις κινήσεις (αντιδράσεις, αδυναμίες) του μαθητή ώστε να προσαρμόζει ανάλογα σε αυτές, τις εκπαιδευτικές του μεθόδους. Σημαντικό χαρακτηριστικό του συστήματος αυτού είναι ότι επιτρέπει στο μαθητή να τροποποιήσει το προφίλ που έχει δημιουργήσει το σύστημα για αυτόν. Έτσι ο μαθητής μπορεί να καθορίσει το επίπεδο γνώσεων που πιστεύει ότι έχει για κάποιο θέμα, επηρεάζοντας έτσι την επιλογή της διδακτέας ύλης. Αυτό έχει ως αποτέλεσμα να αποφεύγεται η κακή επιλογή θεμάτων τουλάχιστον στα αρχικά στάδια, όπου το προφίλ του μαθητή είναι ακόμα ελλιπές. Αν όμως ο πράκτορας διαπιστώσει ότι ο μαθητής αντιμετωπίζει προβλήματα με το επίπεδο της ύλης που του διδάσκεται, θα προσαρμόσει άμεσα το προφίλ του μαθητή και θα προχωρήσει στις αναγκαίες αλλαγές σχετικά με τα επιλεγόμενα εκπαιδευτικά θέματα. Ένα ακόμα ενδιαφέρον χαρακτηριστικό του συστήματος είναι ότι ο σχεδιασμός του πράκτορα διαμεσολάβησης είναι ανεξάρτητος από το εκπαιδευτικό σύστημα με το οποίο συνεργάζεται κάθε φορά. Έτσι ο ίδιος πράκτορας μπορεί να χρησιμοποιηθεί σε συνεργασία με διαφορετικά συστήματα, χωρίς να υπάρχει ανάγκη για αλλαγές στην αρχιτεκτονική του.

#### **4.7.3 Χρήση κινούμενων χαρακτήρων (life-like animated characters) σε εκπαιδευτικές εφαρμογές**

Μερικές εκπαιδευτικές εφαρμογές χρησιμοποιούν κινούμενους χαρακτήρες για να βελτιώσουν την αποτελεσματικότητά τους. Η χρήση τέτοιων χαρακτήρων έχει τα εξής πλεονεκτήματα:

- αυξάνει το ενδιαφέρον και την προσοχή του μαθητή για την διδασκόμενη ύλη
- μέσα από τις εκφράσεις και τις κινήσεις του χαρακτήρα δίνονται επιπλέον βοήθειες στο μαθητή σχετικά με τις επιλογές του
- δημιουργείται σε αυτόν η εντύπωση ότι το εκπαιδευτικό υλικό είναι λιγότερο δύσκολο. (Johnson et al, 1998) (Andre et al, 1997)

Ένα παράδειγμα τέτοιας εφαρμογής είναι το WebPersona (Andre et al, 1997). Το WebPersona κατασκευάζει αυτόματα αλληλεπιδραστικές εκπαιδευτικές παρουσιάσεις (interactive education presentations) στο διαδίκτυο οι οποίες στη συνέχεια παρουσιάζονται με τη βοήθεια ενός κινούμενου χαρακτήρα που έχει τη μορφή ανθρώπου. Αυτό που κάνει το σύστημα ευφυές είναι το γεγονός ότι οι παρουσιάσεις δημιουργούνται δυναμικά ανάλογα με τις αντιδράσεις του χρήστη αλλά και τις ενέργειες του πράκτορα. Οι κινήσεις του πράκτορα βασίζονται σε ένα μοντέλο συμπεριφοράς το οποίο περιλαμβάνει ένα σύνολο στοιχειωδών κινήσεων που εντάσσονται στις ακόλουθες γενικές δράσεις:

- κινήσεις για την παρουσίαση ενός θέματος (Presentation acts)
- αντιδράσεις του χαρακτήρα που εξαρτώνται από τις επιλογές του μαθητή (Reactive behaviors on sensed events)
- κινήσεις για χρονικά διαστήματα όπου ο μαθητής παραμένει αδρανής (Idle-time acts)
- πιθανότατα αναφέρονται σε κινήσεις του χαρακτήρα που οδηγούν το μαθητή στο επόμενο θέμα (Low-level navigation acts).

Η επόμενη κίνηση του πράκτορα επιλέγεται με βάση τις ενέργειες του χρήστη.

#### **4.7.4 Intelligent Helpdesk**

Το σύστημα Intelligent Helpdesk (Greer, 1998) κατασκευάστηκε για να βοηθήσει μαθητές που παρακολουθούν κάποιο πανεπιστημιακό μάθημα να βρουν βοήθεια για θέματα που τους δυσκολεύουν. Η αρχιτεκτονική του συστήματος, με βάση την οποία πραγματοποιείται και η αναπαράσταση της γνώσης, αποτελείται από μια δομή δυο επιπέδων.

Στο πρώτο επίπεδο, (επίπεδο των θεμάτων ή topics layer) αποθηκεύονται στοιχεία που αφορούν την οργάνωση του μαθήματος και τις δραστηριότητες που πρέπει να

πραγματοποιηθούν κατά τη διάρκεια του - εκπαιδευτική ύλη, ασκήσεις, διαγωνίσματα, εργαστήρια κλπ.

Στο δεύτερο , (επίπεδο των εννοιών ή concepts layer), γίνεται αναφορά στις βαθύτερες έννοιες που διδάσκονται στο συγκεκριμένο μάθημα.

Έτσι προκύπτει μια δομή ζευγών θεμάτων-εννοιών (concept-topic structure). Σε αυτή τη δομή μπορεί να υπάρχουν σύνδεσμοι (links) ανάμεσα σε διαφορετικές έννοιες. Οι σύνδεσμοι καταγράφουν γεγονότα όπως η παρακολούθηση κάποιων θεμάτων. Στο επίπεδο των οποίων αναπαρίσταται η "θέση" του μαθητή στο μάθημα - ποιες ενότητες έχει παρακολουθήσει, τι διαγωνίσματα έχει γράψει κλπ. Το δεύτερο επίπεδο, των εννοιών, είναι αναγκαίο για να γίνουν αντιληπτές οι αληθινές γνώσεις και ο βαθμός κατανόησης της ύλης από τον κάθε μαθητή. Τα δεδομένα αυτά είναι ιδιαίτερα χρήσιμα στη διαδικασία αναζήτησης των κατάλληλων μαθητών για την παροχή βοήθειας όταν αυτή ζητηθεί.

Το σύστημα δίνει ιδιαίτερη έμφαση στην αλληλοβοήθεια μεταξύ μαθητών που παρακολουθούν το ίδιο μάθημα. Όταν ένας μαθητής ζητήσει βοήθεια για κάποιο θέμα, αρχικά πραγματοποιείται μια αναζήτηση στη δομή θεμάτων-εννοιών για να βρεθούν ανάλογα θέματα και σχετιζόμενες έννοιες. Με βάση τα αποτελέσματα της αναζήτησης θα προταθούν στο μαθητή τα κατάλληλα άρθρα, news bulletins ή FAQs. Αν δεν βρεθεί κάτι χρήσιμο για να του προταθεί, τότε το σύστημα προχωράει στην αναζήτηση άλλων μαθητών που γνωρίζουν το συγκεκριμένο θέμα. Αφού ειδικός αλγόριθμος εντοπίσει το πλέον κατάλληλο άτομο για να προσφέρει βοήθεια, θα ανοίξει ένα παράθυρο διαλόγου, όπου οι δυο μαθητές μπορούν να συζητήσουν και ο ένας να λύσει τις απορίες του άλλου.

#### **4.7.5 Adaptive Statistics Tutor (AST)**

Το σύστημα AST (Specht et al, 1997) αποτελεί ένα βοηθό διδασκαλίας που κατασκευάζει σειρές μαθημάτων στατιστικής στο Διαδίκτυο προσαρμοσμένες σε διαφορετικούς μαθητές. Η ευφυΐα του συστήματος βρίσκεται στον τρόπο με τον οποίο χρησιμοποιεί τις πληροφορίες που συλλέγει από το μαθητή για να προσαρμόσει την συνολική του συμπεριφορά απέναντι σε αυτόν.

Η αρχιτεκτονική του συστήματος βασίζεται στις τρεις βασικές πηγές γνώσης που χρησιμοποιούν οι αληθινοί καθηγητές. Το πρώτο κομμάτι του συστήματος



σχετίζεται με την εκπαιδευτική γνώση (domain expert module). Το δεύτερο κομμάτι του συστήματος είναι το παιδαγωγικό (pedagogical expert module) και περιλαμβάνει παιδαγωγικές στρατηγικές (pedagogical strategies) και διαγνωστικές γνώσεις (diagnostic knowledge). Το τελευταίο κομμάτι του συστήματος σχετίζεται με την μοντελοποίηση των μαθητών. Κρατάει πληροφορίες για τις προτιμήσεις του κάθε μαθητή, για τους εκπαιδευτικούς του στόχους, για τα μαθήματα που έχει παρακολουθήσει και τις επιδόσεις του σε αυτά. Κάθε ενέργεια του μαθητή καταγράφεται και επηρεάζει τον τρόπο με τον οποίο ανανεώνεται το αντίστοιχο μοντέλο.

Η συμπεριφορά του AST αλλάζει για κάθε μαθητή, ανάλογα με το μοντέλο που του αντιστοιχεί. Αυτή η προσαρμοστικότητα του συστήματος εκφράζεται:

- στην πλοήγηση (adaptive navigation support)
- στη σειρά παράδοσης των μαθημάτων (adaptive sequencing)
- στον τρόπο εξέτασης (adaptive testing)
- στην επιλογή της πιο κατάλληλης διδακτικής μεθόδου (adaptation of the default teaching strategy).

#### **4.7.6 AlgeBrain, ένας βοηθός διδασκαλίας άλγεβρας**

Το σύστημα AlgeBrain (Alpert et al, 1999) είναι ένας βοηθός διδασκαλίας στον τομέα της άλγεβρας γυμνασίου. Στόχος του είναι να αποτελέσει έναν παιδαγωγικό βοηθό για τον καθηγητή του μαθήματος, που θα συνοδεύει τους μαθητές κατά την επίλυση των προβλημάτων και θα τους παρέχει υποστήριξη, αν αυτό κρίνεται απαραίτητο. Ένα ενδιαφέρον γνώρισμα του συστήματος είναι ότι η εκπαιδευτική του συμπεριφορά δεν είναι συγκεντρωμένη σε έναν κεντρικό server, όπως συμβαίνει στις περισσότερες υπάρχουσες εφαρμογές. Αντίθετα, έχει επιλεγεί μια λύση όπου το κυρίως μέρος του συστήματος βρίσκεται σε έναν κεντρικό server - στον οποίο είναι αποθηκευμένα τα μοντέλα όλων των μαθητών για να είναι προσπελάσιμα από οποιοδήποτε τερματικό, ενώ σε κάθε client που τρέχει τοπικά την εφαρμογή ενεργοποιείται ένα Java Applet με το user interface - για να υπάρχει ταχύτερη απόκριση στις ενέργειες του χρήστη. Το Java Applet έρχεται σε επικοινωνία με το server μόνο όταν αυτό είναι αναγκαίο.

Κεντρικό κομμάτι στην αρχιτεκτονική του AlgeBrain είναι ένα έμπειρο σύστημα βασισμένο σε κανόνες (rule-based expert system), το οποίο παρακολουθεί τις ενέργειες του μαθητή και τις συγκρίνει με αυτές ενός άριστου λύτη (expert solver) στο ίδιο πρόβλημα.

#### **4.8 Επίλογος**

Οι πράκτορες και πιο συγκεκριμένα οι ευφυείς πράκτορες μπορούν αν χρησιμοποιηθούν να δώσουν άλλη διάσταση σε ένα σύστημα. Το πλήθος των αρχιτεκτονικών υλοποίησης τους μας παρέχει μια μεγάλη ποικιλία, έτσι ώστε να διαλέξουμε τον πράκτορα που ταιριάζει πιο πολύ στις ανάγκες μας. Τέλος τα ανθρώπινα χαρακτηριστικά που τυχόν μπορεί να έχει ένας πράκτορας είναι αυτά που τον κάνουν αρεστό και οικείο στους χρήστες.

## 5 Παρουσίαση εφαρμογής

### 5.1 Στοιχεία που χρησιμοποιήθηκαν

- PHP
- MySQL
- Adobe Photoshop CS3
- Adobe Flash CS4 Professional
- ESOPOS

#### 5.1.1 PHP

Η PHP είναι μια Server-Side γλώσσα προγραμματισμού σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Ένα από τα πλεονεκτήματα της PHP είναι η ικανότητα της να χρησιμοποιηθεί σε συνδυασμό με το server βάσεων δεδομένων MySQL.

Η PHP λειτουργεί ως ενδιάμεσος μεταξύ της MySQL και του Flash, είναι ο συνδετικός κρίκος που ενώνει αυτά τα δυο στοιχεία. Το Flash δεν είναι σε θέση να επικοινωνήσει απ'ευθείας με την MySQL και έτσι χρειάζεται την PHP.

(<http://el.wikipedia.org/wiki/PHP>)

#### 5.1.2 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων. Μια βάση δεδομένων μερικά από τα σημαντικότερα πράγματα που μας επιτρέπει να κάνουμε είναι να αποθηκεύουμε, να αναζητούμε, να ταξινομούμε και να ανακαλούμε δεδομένα. Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα μας, με αυτό τον τρόπο δίνεται η δυνατότητα να εργαστούν πολλοί χρήστες ταυτόχρονα. Η MySQL χρησιμοποιεί την SQL(Structured Query

Language), η οποία είναι η τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων. Η MySQL είναι διαθέσιμη με άδεια ανοικτού κώδικα.

(<http://en.wikipedia.org/wiki/MySQL>)

### 5.1.3 Adobe Photoshop

Το **Adobe Photoshop**, ή απλά **Photoshop**, είναι ένα πρόγραμμα επεξεργασίας γραφικών το οποίο μας παρέχει άπειρες δυνατότητες. Αναπτύχθηκε και κυκλοφόρησε από την Adobe Systems. Είναι ένα απαραίτητο εργαλείο για τους επαγγελματίες γραφίστες. (<http://el.wikipedia.org/wiki/Photoshop>)

### 5.1.4 Adobe Flash

Το Adobe Flash είναι μια πολυμεσική πλατφόρμα την οποία χρησιμοποιούμε για να προσθέσουμε animation, video και το στοιχείο της αλληλεπίδρασης σε μια εφαρμογή ή σε μια σελίδα web. ([http://en.wikipedia.org/wiki/Adobe\\_Flash](http://en.wikipedia.org/wiki/Adobe_Flash))

### 5.1.5 ESOPOS

Το ESOPOS είναι ένα σύστημα το οποίο μετατρέπει ελληνικό κείμενο σε ομιλία. Έχει υλοποιηθεί από το εργαστήριο επεξεργασίας σήματος και βιοατρικής μηχανικής του τμήματος ηλεκτρολόγων μηχανικών και μηχανικών υπολογιστών του ΑΠΘ. (<http://esopos.ee.auth.gr/index.html>)

## 5.2 Παρουσίαση της εφαρμογής

Στην MySQL έχουμε δημιουργήσει μια βάση με όνομα users και ένα πίνακα με όνομα users. Ο πίνακας έχει τα εξής πεδία:

- Username : Όνομα χρήστη
- Password : Κωδικός χρήστη
- userLevel : Επίπεδο χρήστη (Τιμές 0 ή 1)
- dyscalculia : Αν ο χρήστης έχει δυσαριθμησία (Τιμές TRUE ή FALSE)
- page : Σε ποια σελίδα ο χρήστης έκανε αποσύνδεση

Με την βοήθεια του Flash δημιουργούμε ένα νέο αρχείο Flash με διαστάσεις 1024X768. Στη συνέχεια εισάγουμε στη βιβλιοθήκη του Flash όλα τα αντικείμενα που θα χρησιμοποιήσουμε στην εφαρμογή μας (εικόνες, ήχοι, κτλ). Έπειτα τοποθετούμε στο stage του Flash τα αντικείμενα που θα χρησιμοποιήσουμε για να δημιουργήσουμε την πρώτη οθόνη (Εικόνα 1), που θα δει ο χρήστης αλλά και το animation το οποίο θα τον μεταφέρει στη σελίδα όπου γίνεται η είσοδος του χρήστη στο σύστημα. Σε κάθε οθόνη υπάρχει η background εικόνα της βιβλιοθήκης και πάνω σε αυτή ένα αριστερό και ένα δεξιό φύλλο του βιβλίου μας.



Εικόνα 1. Πρώτη οθόνη.

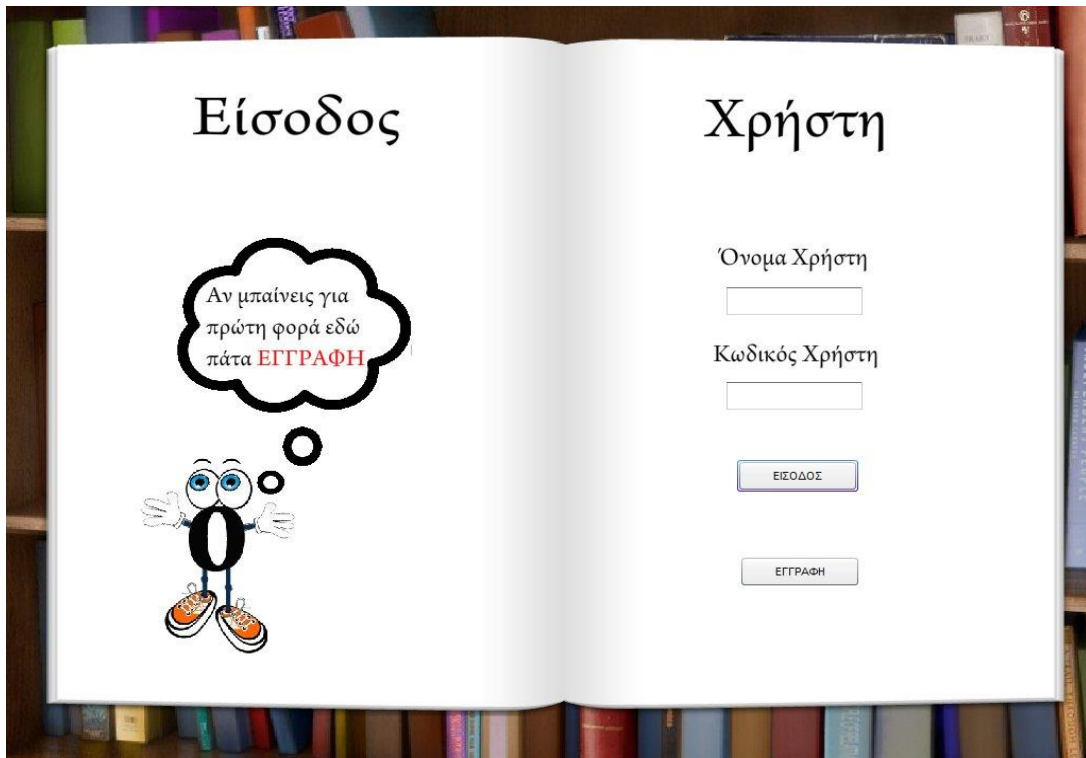
### 5.2.1 Οθόνη εισόδου χρήστη στο σύστημα

Με την βοήθεια του παρακάτω κώδικα μόλις ο χρήστης φτάσει σε αυτή την οθόνη, αναπαράγεται ένα clip ήχου το οποίο του λέει «Γεια».

```
var mySound:Sound = new Sound();  
mySound.load(new URLRequest("sounds/geia.mp3"));  
mySound.play();
```

Στην οθόνη στην οποία γίνεται η είσοδος χρήστη (Εικόνα 2) έχουμε τοποθετήσει την αριστερή και δεξιά σελίδα του βιβλίου μας και πάνω σε αυτές έχουμε τοποθετήσει κάποια αντικείμενα. Τα αντικείμενα είναι:

- 4 αντικείμενα τύπου static text τα οποία μας δίνουν πληροφορίες για την σελίδα. Τα κείμενα τους είναι: «Είσοδος», «Χρήστη», «Όνομα Χρήστη», «Κωδικός Χρήστη».
- 2 αντικείμενα τύπου text input στα οποία ο χρήστης εισάγει το όνομα και κωδικό χρήστη.
- 2 κουμπιά. Το ένα με ετικέτα «ΕΙΣΟΔΟΣ» και το άλλο με ετικέτα «ΕΓΓΡΑΦΗ». Μόλις πατηθεί το «ΕΙΣΟΔΟΣ» το σύστημα με τη βοήθεια κώδικα (Παράρτημα Α) ελέγχει αν υπάρχει στη βάση ο συγκεκριμένος χρήστης. Αν υπάρχει τότε μας πάει στην επόμενη οθόνη αλλιώς εμφανίζει μήνυμα για το τι φταίει. Μόλις πατηθεί το «ΕΓΓΡΑΦΗ» το σύστημα μας πάει στην οθόνη εγγραφής νέου χρήστη στο σύστημα.
- 1 αντικείμενο τύπου dynamic text το οποίο μας πληροφορεί για την διαδικασία εισόδου του χρήστη στο σύστημα. Αν ο χρήστης δεν έχει συμπληρώσει κάποιο πεδίο τότε το αντικείμενο αυτό εμφανίζει το κείμενο «Συμπλήρωσε όλα τα πεδία». Αν έχει εισάγει λάθος όνομα χρήστη τότε το αντικείμενο αυτό εμφανίζει το κείμενο «Αυτό το όνομα χρήστη δεν υπάρχει». Αν έχει εισάγει λάθος κωδικό χρήστη τότε το αντικείμενο αυτό εμφανίζει το κείμενο «Λάθος κωδικός».
- 1 αντικείμενο τύπου movie clip (ο πράκτορας). Με την βοήθεια κώδικα (Παράρτημα Α) τα μάτια του πράκτορα παρακολουθούν το ποντίκι, αυτό συμβαίνει σε κάθε οθόνη στην οποία εμφανίζεται ο πράκτορας. Επίσης το δεξί χέρι του με τη βοήθεια animation χαιρετάει τον χρήστη. Ο πράκτορας δημιουργήθηκε με την βοήθεια του Photoshop.
- 1 αντικείμενο τύπου movie clip (η φούσκα που μας δείχνει τι σκέφτεται ο πράκτορας)
- 1 αντικείμενο τύπου movie clip με όνομα cloudTTS το οποίο αρχικά είναι αόρατο (Εικόνα 2.1). Γίνεται ορατό με την βοήθεια κώδικα (Παράρτημα Α) εξαφανίζοντας παράλληλα την φούσκα που μας δείχνει την σκέψη του πράκτορα αν ο χρήστης δεν κάνει καμία ενέργεια για 30 δευτερόλεπτα, αυτό συμβαίνει σε κάθε οθόνη στην οποία εμφανίζεται ο πράκτορας.



Εικόνα 2. Οθόνη εισόδου χρήστη στο σύστημα.



Εικόνα 2.1 Οθόνη στην οποία φαίνεται το αντικείμενο το οποίο γίνεται ορατό αν ο χρήστης παραμείνει ανενεργός για κάποιο διάστημα.

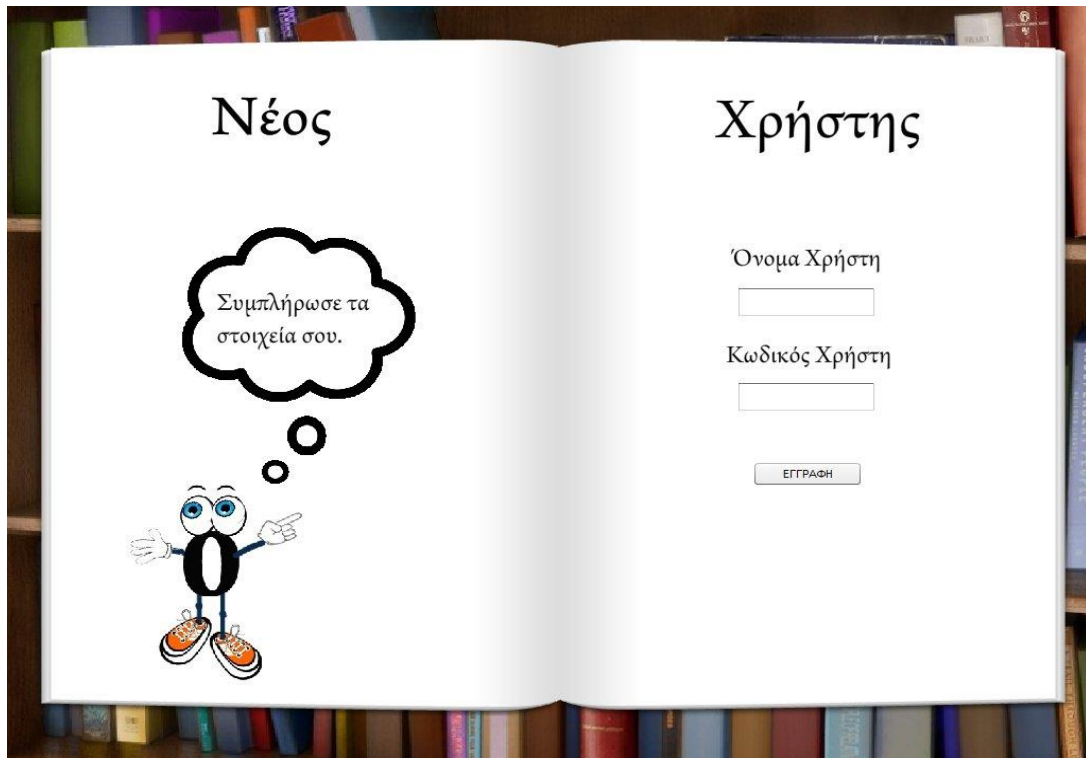
### 5.2.2 Οθόνη εγγραφής χρήστη στο σύστημα

Στην οθόνη όπου γίνεται η εγγραφή του χρήστη στο σύστημα (Εικόνα 3) έχουμε την αριστερή και δεξιά σελίδα του βιβλίου μας και τα εξής αντικείμενα πάνω τους:

- 4 αντικείμενα τύπου static text τα οποία μας δίνουν πληροφορίες για την σελίδα. Τα κείμενα τους είναι: «Νέος», «Χρήστης», «Όνομα Χρήστη», «Κωδικός Χρήστη».
- 1 αντικείμενο τύπου dynamic text το οποίο μας πληροφορεί για την διαδικασία εγγραφής του χρήστη στο σύστημα. Αν ο χρήστης εισάγει κάποιο όνομα χρήστη το οποίο έχει άλλος χρήστης, τότε του εμφανίζεται μήνυμα που του λέει «Αυτό το όνομα χρήστη υπάρχει ήδη». Επίσης αν ο χρήστης δεν συμπληρώσει όλα τα πεδία τότε εμφανίζεται κατάλληλο μήνυμα «Συμπλήρωσε όλα τα πεδία».
- 2 αντικείμενα τύπου text input στα οποία ο χρήστης εισάγει το όνομα και κωδικό χρήστη που επιθυμεί.
- 1 αντικείμενο τύπου movie clip (ο πράκτορας).
- 1 αντικείμενο τύπου movie clip (η φούσκα που μας δείχνει τι σκέφτεται ο πράκτορας)
- 1 αντικείμενο τύπου movie clip το cloudTTS το οποίο εξηγήσαμε στην προηγούμενη οθόνη.
- 1 κουμπί με ετικέτα «ΕΓΓΡΑΦΗ», το οποίο μόλις πατηθεί με την βοήθεια κώδικα (Παράρτημα Α) δημιουργεί στον πίνακα της βάσης μας μια νέα εγγραφή με τα δεδομένα του χρήστη. Τα πεδία του πίνακα με ετικέτες username και password παίρνουν αντίστοιχα τις τιμές που εισήγαγε ο χρήστης στα 2 αντικείμενα τύπου text input. Τα υπόλοιπα πεδία παίρνουν κάποιες προεπιλεγμένες τιμές. Η προεπιλεγμένη τιμή για το κάθε ένα από αυτά είναι :  
dyscalculia = False (Αρχικά θεωρούμε ότι ο χρήστης δεν έχει δυσαριθμησία)  
userLevel = 0 (Αρχικά θεωρούμε ότι ο χρήστης βρίσκεται στο εύκολο επίπεδο δυσκολίας)  
page = 142 (Είναι ο αριθμός του frame της οθόνης εισόδου χρήστη στο σύστημα)

Στη συνέχεια το σύστημα μας οδηγεί στην επόμενη οθόνη.





Εικόνα 3. Οθόνη εγγραφής χρήστη στο σύστημα.

### 5.2.3 Οθόνη Ερωτηματολογίου

Στην οθόνη του ερωτηματολογίου (Εικόνα 4) έχουμε τα εξής αντικείμενα:

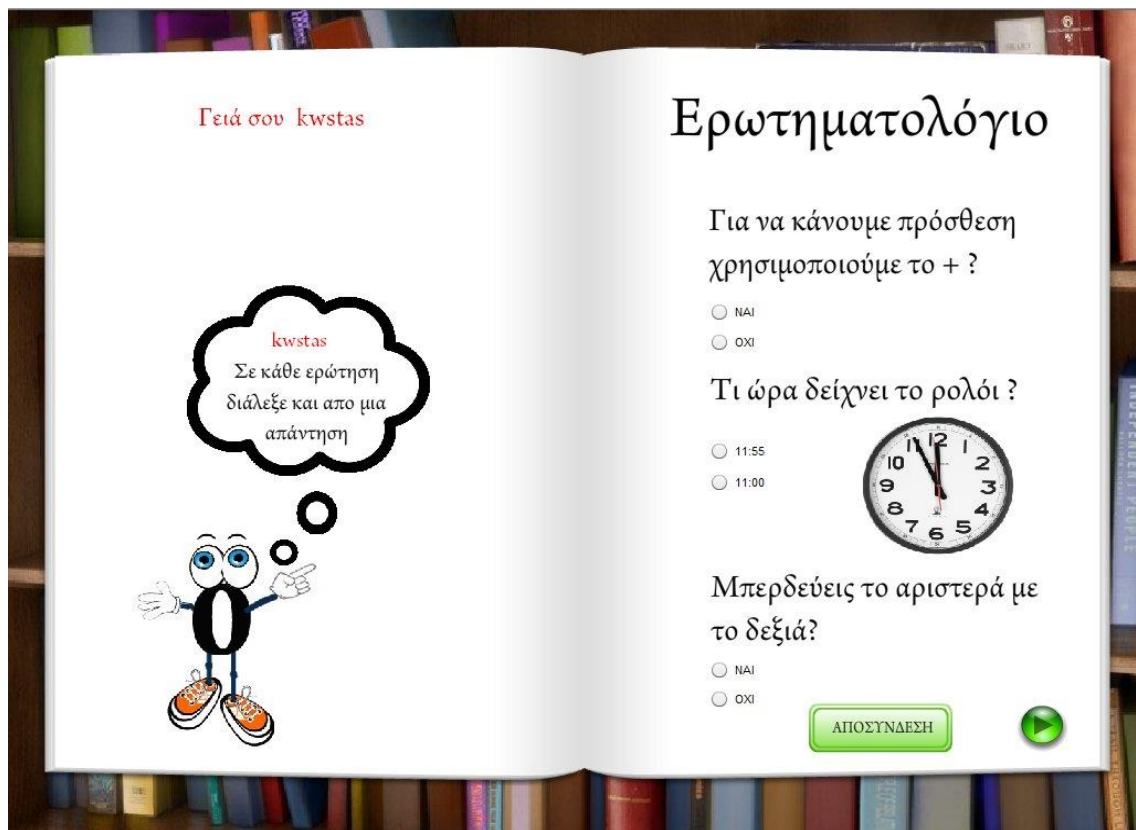
- 1 αντικείμενο τύπου dynamic text με όνομα statusText και το οποίο εμφανίζει το κείμενο: Γειά σου και το όνομα του χρήστη.  
`statusText.text="Γειά σου "+pageVars.user;`
- 1 αντικείμενο τύπου movie clip (ο πράκτορας).
- 1 αντικείμενο τύπου movie clip (η φούσκα που μας δείχνει τι σκέφτεται ο πράκτορας)
- 1 αντικείμενο τύπου movie clip το cloudTTS το οποίο εξηγήσαμε στην προηγούμενη οθόνη.
- 4 αντικείμενα τύπου static text τα 3 από τα οποία είναι ερωτήσεις για να καταλάβουμε βάση των απαντήσεων τους αν ο χρήστης έχει δυσαριθμήςια. Τα κείμενα τους είναι: «Ερωτηματολόγιο», «Για να κάνουμε πρόσθεση χρησιμοποιούμε το + ?», «Τι ώρα δείχνει το ρολόι ?», «Μπερδεύεις το αριστερά με το δεξιά?».
- 1 εικόνα ρολογιού

- 6 αντικείμενα τύπου radio button τα οποία αποτελούν ζεύγη απαντήσεων στις ερωτήσεις των παραπάνω αντικειμένων.

Αν ο χρήστης απαντήσει σε παραπάνω από 1 στις 3 ερωτήσεις λάθος τότε το σύστημα συμπεραίνει ότι ο χρήστης έχει δυσαριθμησία αλλιώς συμπεραίνει ότι ο χρήστης δεν έχει δυσαριθμησία, κρατάει δηλαδή την προεπιλεγμένη τιμή.

```
if (count >1){  
    globals.data.dyscalculia=true;  
}
```

- 1 κουμπί το οποίο θα υπάρχει από εδώ και πέρα σε όλες τις σελίδες και το οποίο μας οδηγεί στην επόμενη σελίδα.
- 1 κουμπί το οποίο θα υπάρχει από εδώ και πέρα σε όλες τις σελίδες και το οποίο μόλις πατηθεί αποθηκεύει την κατάσταση του χρήστη στη βάση (Παράρτημα Α) και ξεκινάει πάλι από το 1<sup>ο</sup> frame.



Εικόνα 4. Οθόνη ερωτηματολογίου.

## 5.2.4 Οθόνη 1<sup>ης</sup> Άσκησης

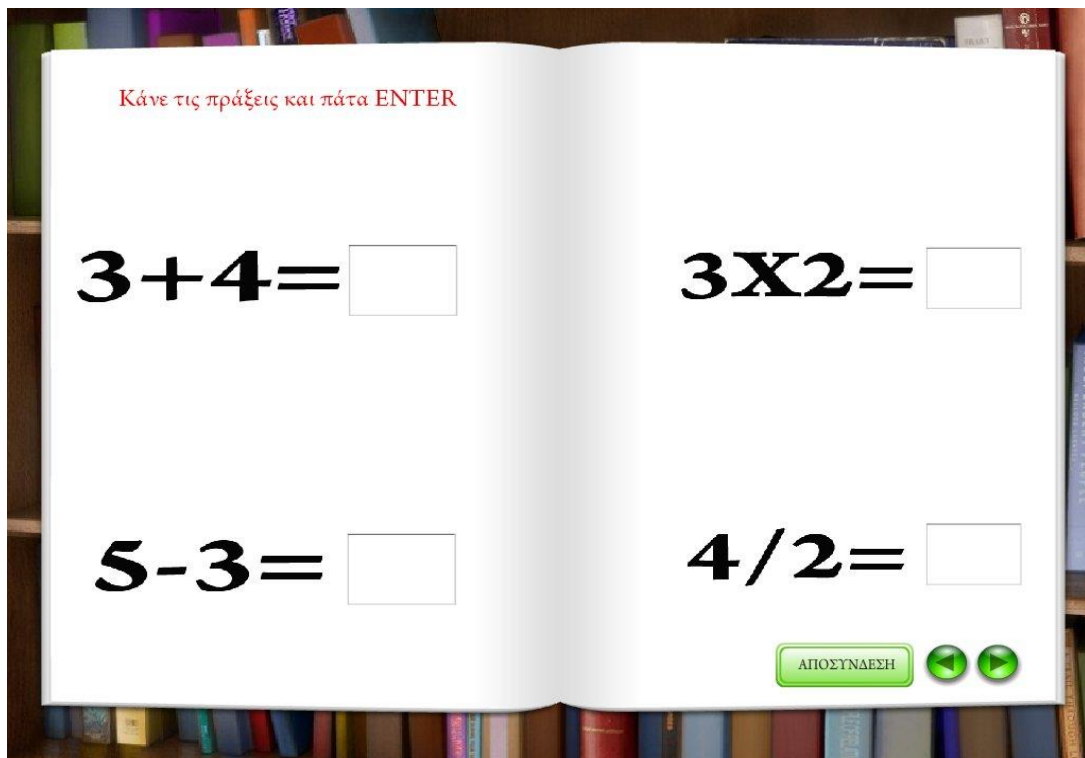
Στην οθόνη της 1<sup>ης</sup> άσκησης (Εικόνα 5) έχουμε δυο αντικείμενα τύπου `movie clip` που είναι η σελίδες μας και μέσα σε αυτές έχουμε τοποθετήσει τα αντικείμενα μας.

Όσο αναφορά την αριστερή σελίδα μέσα σε αυτή έχουμε:

- 2 αντικείμενα τύπου `movie clip` που αναπαριστούν τις πράξεις μας ( $3+4=$  ,  $5-3=$ )
- 3 αντικείμενα τύπου `dynamic text`. Το 1<sup>ο</sup> πληροφορεί τον χρήστη ότι μόλις πληκτρολογήσει την πράξη του μέσα στο αντίστοιχο `text input` πρέπει να πατήσει «ENTER». Τα άλλα δύο πληροφορούν τον χρήστη αν έχει κάνει σωστά την πράξη. (Εικόνα 5.1 , 5.2)
- 2 αντικείμενα τύπου `text input` μέσα στα οποία ο χρήστης γράφει την απάντηση του. Μόλις ο χρήστης γράψει την απάντηση του και πατήσει ENTER `if (e.keyCode == Keyboard.ENTER)` τότε τα αντίστοιχα `dynamic text` εμφανίζουν το μήνυμα «ΜΠΡΑΒΟ» σε περίπτωση σωστής απάντησης και «Προσπάθησε ξανά» σε περίπτωση λανθασμένης. Αν η απάντηση του χρήστη είναι σωστή, τότε το `text input` γίνεται μη τροποποιήσιμο καθώς δεν έχει νόημα να αλλάξει ο χρήστης την απάντηση του. `ex1ProsthesisInput.editable=false;` Σε αντίθετη περίπτωση ο χρήστης μπορεί να αλλάζει την απάντηση του όσες φορές θέλει.
- Τέλος έχουμε 2 αντικείμενα τύπου `movie clip` τα οποία είναι το βοήθημα μας (Εικόνα 5.3) για χρήστες με δυσαριθμησία. Πιο συγκεκριμένα αν στην προηγούμενη οθόνη (οθόνη ερωτηματολογίου) είχαμε διαγνώσει ότι ο χρήστης έχει δυσαριθμησία τότε από την τρέχουσα οθόνη και έπειτα θα εμφανίζονται κάποια βοηθήματα με σκοπό να βοηθήσουν τον χρήστη να απαντήσει σωστά στις εκάστοτε ερωτήσεις (πράξεις).

Όσο αναφορά την δεξιά σελίδα αυτό που αλλάζει είναι οι πράξεις, οι οποίες είναι : ( $3 \times 2=$  ,  $4/2=$ ) και το γεγονός ότι κάτω δεξιά έχει τρία κουμπιά.

1. Το κουμπί Αποσύνδεση
2. Το κουμπί εμπρός
3. το κουμπί πίσω



Εικόνα 5. Οθόνη 1<sup>η</sup> άσκησης.

Προσπάθησε ξανά

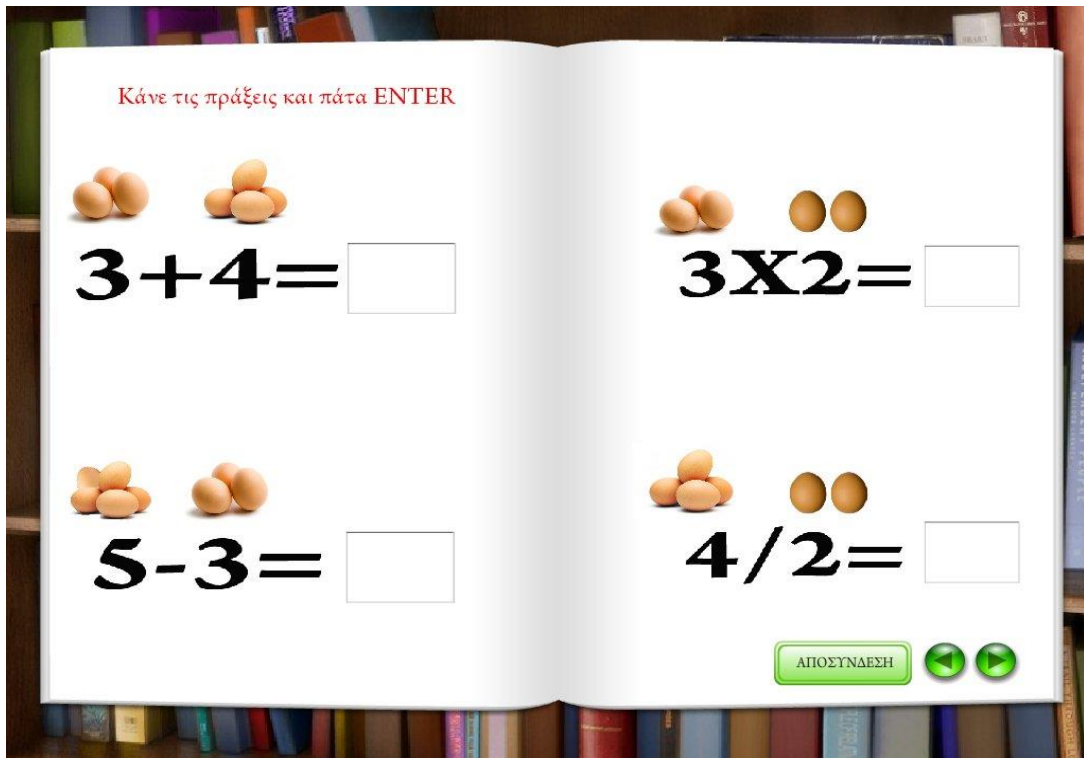
2

Εικόνα 5.1. Ο χρήστης έχει κάνει την πράξη λάθος.

ΜΠΡΑΒΟ

2

Εικόνα 5.2. Ο χρήστης έχει κάνει την πράξη σωστά.



Εικόνα 5.3. Το βοήθημα μας εμφανίζεται με την μορφή αυγών.

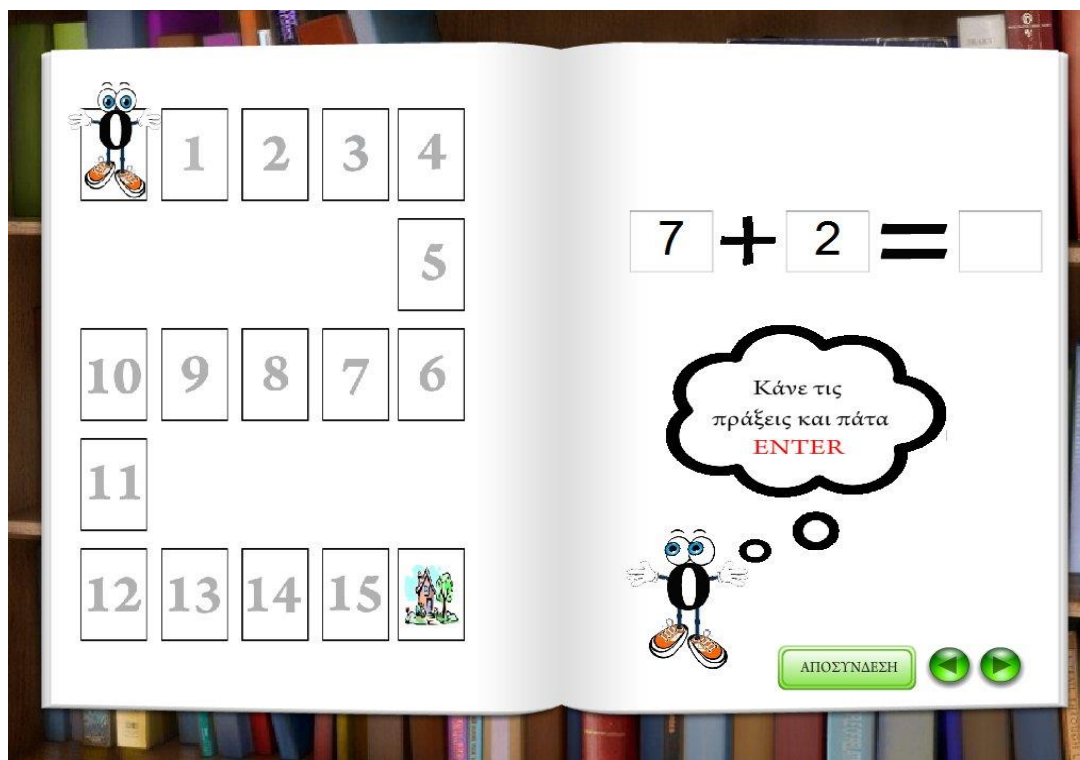
### 5.2.5 Οθόνη 2<sup>ης</sup> Άσκησης

Στην οθόνη της 2<sup>ης</sup> άσκησης (Εικόνα 6) έχουμε δυο αντικείμενα τύπου monie clip που είναι η σελίδες μας και μέσα σε αυτές έχουμε τοποθετήσει τα αντικείμενα μας.

Όσο αναφορά την αριστερή σελίδα μέσα σε αυτή έχουμε:

- 2 αντικείμενα τύπου monie clip. Το ένα είναι ο πράκτορας και το άλλο είναι το επιτραπέζιο μας πάνω στο οποίο κινείται ο πράκτορας.

Σκοπός του χρήστη είναι, κάνοντας σωστά τις πράξεις που του ζητούνται να κάνει τον πράκτορα να φτάσει σπίτι του.



Εικόνα 6. Οθόνη 2<sup>ης</sup> άσκησης.

Όσο αναφορά την δεξιά σελίδα έχουμε τα εξής αντικείμενα:

- 3 κουμπιά. Το «Αποσύνδεση», το κουμπί εμπρός και το κουμπί πίσω.
- 3 αντικείμενα τύπου text input. Στα δυο πρώτα εμφανίζονται με την βοήθεια της συνάρτησης  $(\text{Math.ceil}(\text{Math.random()} * 8) + 2;)$  τυχαίοι αριθμοί από το 1 έως το 10. Στο τρίτο ο χρήστης γράφει την απάντησή του και πατάει «ENTER». Μόλις γίνει αυτό, το σύστημα ελέγχει αν η απάντηση είναι σωστή και με τη βοήθεια κώδικα (Παράρτημα Α) κινεί τον πράκτορα (της αριστερής σελίδας) κατά μια θέση εμπρός πάνω στο επιτραπέζιο. Παράλληλα ξαναχρησιμοποιείται η συνάρτηση  $(\text{Math.ceil}(\text{Math.random()} * 8) + 2;)$  για να εμφανίσει δυο άλλους τυχαίους αριθμούς.
- 4 αντικείμενα τύπου movie clip τα οποία αναπαριστούν τους τελεστές των πράξεων ανάμεσα στους δυο αριθμούς και το ίσον. Οι τελεστές μας είναι (+, -, x) Κάθε φορά που ο χρήστης γράφει την απάντησή του στο text input και πατάει το ENTER πέραν από τους παραπάνω αριθμούς αναπαράγουμε και άλλους, αυτή τη φορά έτσι ώστε να προσδιορίσουμε πια πράξη θα κάνουμε και να εμφανίσουμε και το αντίστοιχο movie clip. Διακρίνονται δυο περιπτώσεις:

- 1<sup>η</sup> Αν ο χρήστης στην προηγούμενη οθόνη (Άσκηση 1) είχε κάνει λιγότερες από τρεις σωστές τότε παραμένει στο εύκολο επίπεδο (`globals.data.userLevel=0;`) και οι τυχαίοι αριθμοί που αφορούν τις πράξεις θα είναι 0 ή 1 (`praksh = Math.round(Math.random()*1);`). Δηλαδή είτε πρόσθεση είτε αφαίρεση. Άρα θα εμφανίσουμε είτε το `movie clip` της αφαίρεσης είτε το `movie clip` της πρόσθεσης.

```
if (praksh == 0){
    ex2Plus.alpha=100;
    ex2Minus.alpha=0;
    ex2Mult.alpha=0;
}
else if (praksh == 1){
    ex2Plus.alpha=0;
    ex2Minus.alpha=100;
    ex2Mult.alpha=0;
}
```

- 2<sup>η</sup> Αν ο χρήστης στην προηγούμενη οθόνη είχε κάνει τουλάχιστον τρεις σωστές πράξεις τότε ανέβαινε στο δύσκολο επίπεδο (`globals.data.userLevel=1;`) και οι τυχαίοι αριθμοί που αφορούν τις πράξεις θα ήταν ή 0 ή 1 ή 2 (`praksh = Math.round(Math.random()*2);`). Άρα θα εμφανίσουμε το `movie clip` που πρέπει.

```
else if (praksh == 2){
    ex2Plus.alpha=0;
    ex2Minus.alpha=0;
    ex2Mult.alpha=100;
}
```

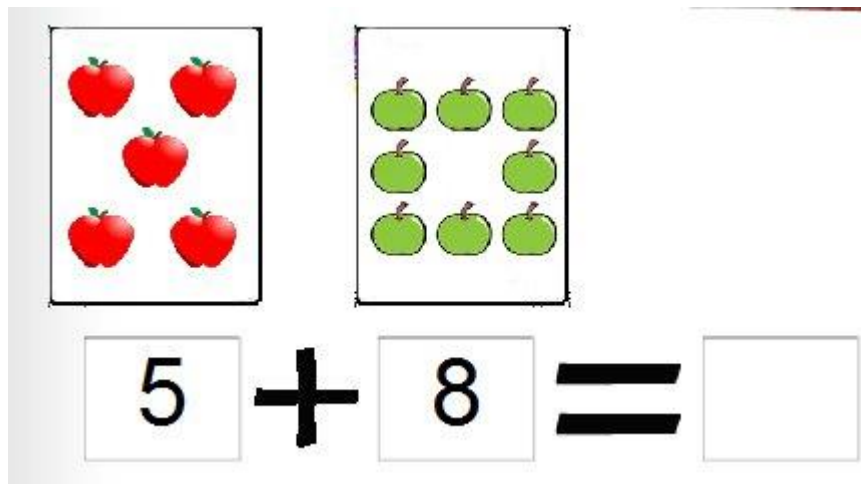
Το 0 αντιστοιχεί στην πράξη τις πρόσθεσης.

Το 1 αντιστοιχεί στην πράξη της αφαίρεσης.

Το 2 αντιστοιχεί στην πράξη του πολλαπλασιασμού.

- 2 αντικείμενα τύπου `movie clip` (Εικόνα 6.1) τα οποία είναι το βοήθημα μας για τους χρήστες με δυσαριθμησία. `if(globals.data.dyscalculia==true)`. Σε

κάθε αριθμό από το 1 ως το 10 αντιστοιχεί και μια κάρτα η οποία αναπαριστά τόσα φρούτα όσα και ο αριθμός.



**Εικόνα 6.1** Κάρτες που εμφανίζονται ως βοήθημα για χρήστες με δυσαριθμησία.

- 2 αντικείμενα τύπου monie clip (ο πράκτορας και η φούσκα). Κάθε φορά που ο χρήστης κάνει μια πράξη σωστά ο πράκτορας χαμογελά (Εικόνα 6.2) ενώ κάθε φορά που ο χρήστης κάνει μια πράξη λάθος τότε ο πράκτορας λυπάται (Εικόνα 6.3). Επίσης κάθε φορά που ο χρήστης κάνει τρεις πράξεις στη σειρά σωστές τότε εμφανίζεται το μήνυμα «Μπράβο συνέχισε έτσι» (Εικόνα 6.4) στη φούσκα και αναπαράγεται ένα clip ήχου το οποίο λέει: «Μπράβο συνέχισε έτσι». Κάθε φορά που ο χρήστης κάνει 3 λάθος πράξεις στην σειρά τότε εμφανίζεται το μήνυμα «Μη Βιάζεσαι Σκέψου καλύτερα» στη φούσκα (Εικόνα 6.5) και αναπαράγεται ένα clip ήχου το οποίο λέει: «Πρόσεχε». Με αυτόν τον τρόπο δημιουργείται η αίσθηση ότι ο πράκτορας μιλάει στον χρήστη.



**Εικόνα 6.2** Ο πράκτορας χαμογελά.



**Εικόνα 6.3** Ο πράκτορας λυπάται.





Εικόνα 6.4 Μήνυμα που εμφανίζεται μετά από τρεις σωστές πράξεις.



Εικόνα 6.5 Μήνυμα που εμφανίζεται μετά από τρεις λανθασμένες πράξεις.

Τέλος αν ο χρήστης κάνει τις πράξεις και καταφέρει να φτάσει ο πράκτορας (της αριστερής σελίδας) σπίτι του τότε συμβαίνουν τα εξής:

- Ο πράκτορας (της δεξιάς σελίδας) χειροκροτεί (Εικόνα6.6).
- Αναπαράγεται ένα clip ήχου το οποίο λέει «ΜΠΡΑΒΟ».
- Στην φούσκα εμφανίζεται το μήνυμα «ΜΠΡΑΒΟ» (Εικόνα6.6).



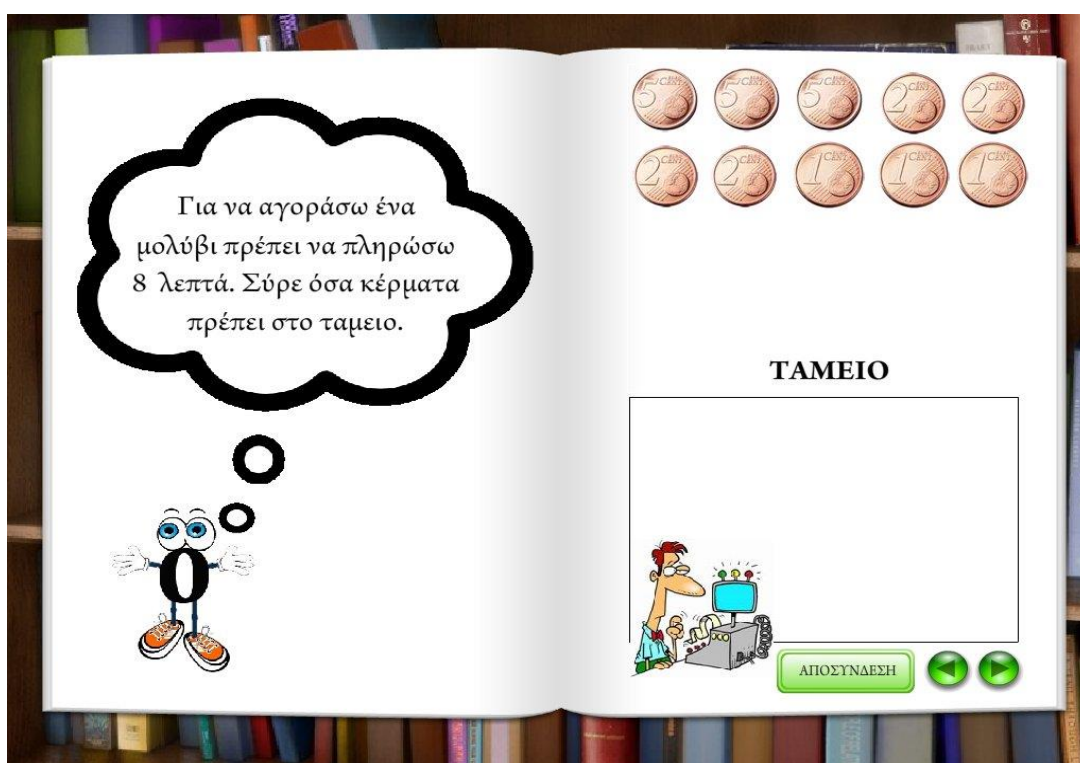
Εικόνα 6.6 Ο πράκτορας χειροκροτεί και η φούσκα εμφανίζει το μήνυμα «ΜΠΡΑΒΟ».

### 5.2.6 Οθόνη 3<sup>ης</sup> Άσκησης

Στην οθόνη της 3<sup>ης</sup> και τελευταίας άσκησης (Εικόνα 7) έχουμε δυο αντικείμενα τύπου monie clip που είναι η σελίδες μας και μέσα σε αυτές έχουμε τοποθετήσει τα αντικείμενα μας.

Όσο αναφορά την αριστερή σελίδα μέσα σε αυτή έχουμε:

- 3 αντικείμενα τύπου monie clip. Ο πράκτορας, η φούσκα και ένα μολύβι το οποίο εμφανίζεται εάν ο χρήστης ολοκληρώσει με επιτυχία την άσκηση.



Εικόνα 7 Οθόνη 3<sup>ης</sup> άσκησης.

Όσο αναφορά την δεξιά σελίδα, έχουμε τα εξής:

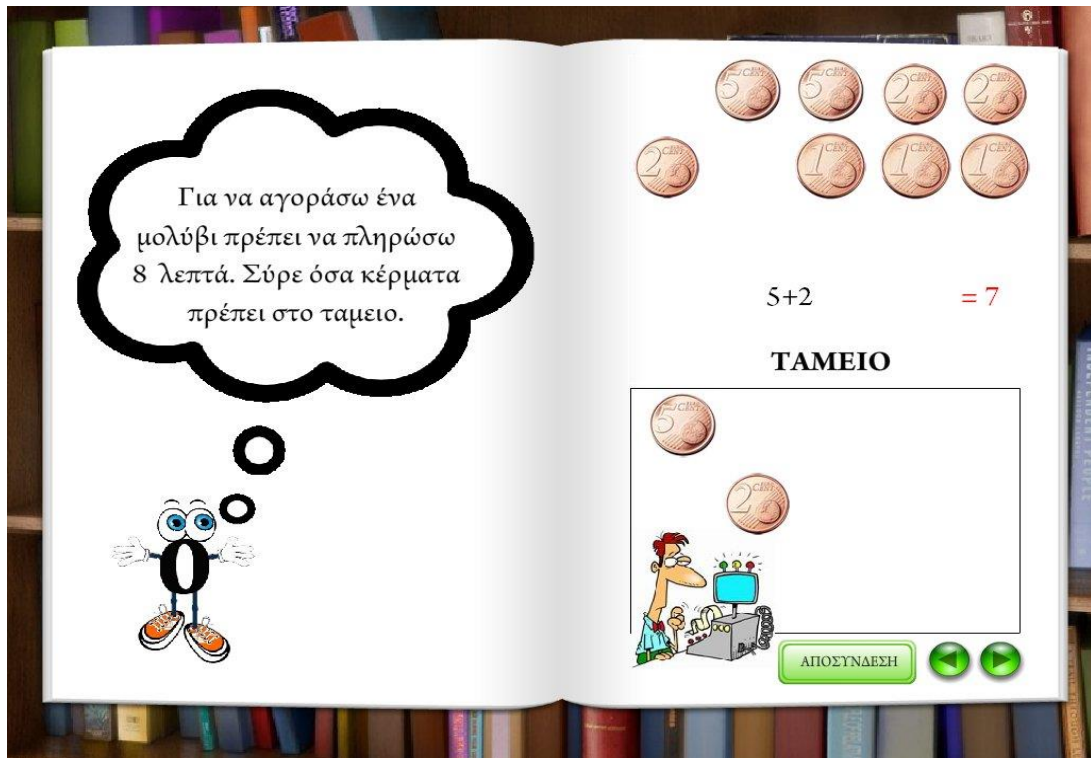
- 3 κουμπιά. Το κουμπί «Αποσύνδεση», το κουμπί εμπρός και το κουμπί πίσω.
- 10 αντικείμενα τύπου monie clip τα οποία αντιπροσωπεύουν κέρματα euro. Έχουμε τρία κέρματα των πέντε λεπτών, τέσσερα κέρματα των δυο λεπτών και τρία κέρματα του ενός λεπτού. Σκοπός του χρήστη είναι να σύρει όσα κέρματα του ζητείται στο ταμείο. Διακρίνουμε δυο περιπτώσεις:

- 1<sup>η</sup> Αν ο χρήστης στην προηγούμενη οθόνη (άσκηση 2) έκανε πάνω από πέντε λάθη και ολοκλήρωσε την άσκηση με επιτυχία τότε μεταβαίνει στο εύκολο επίπεδο και άρα του ζητείται να σύρει στο ταμείο 8 λεπτά
- 2<sup>η</sup> Αν ο χρήστης στην προηγούμενη οθόνη έκανε κάτω από 6 λάθη και ολοκλήρωσε την άσκηση με επιτυχία τότε μεταβαίνει στο δύσκολο επίπεδο και άρα του ζητείται να σύρει στο ταμείο 17 λεπτά.

Αν ο χρήστης δεν ολοκλήρωσε την άσκηση με επιτυχία τότε αυτομάτως μεταβαίνει στο εύκολο επίπεδο δυσκολίας.

```
if(globals.data.countEx2>5 && globals.data.ex3==true ){
    globals.data.userLevel=0;
}
else if(globals.data.countEx2<6 && globals.data.ex3==true){
    globals.data.userLevel=1;
}
else{
    globals.data.userLevel=0;
}
```

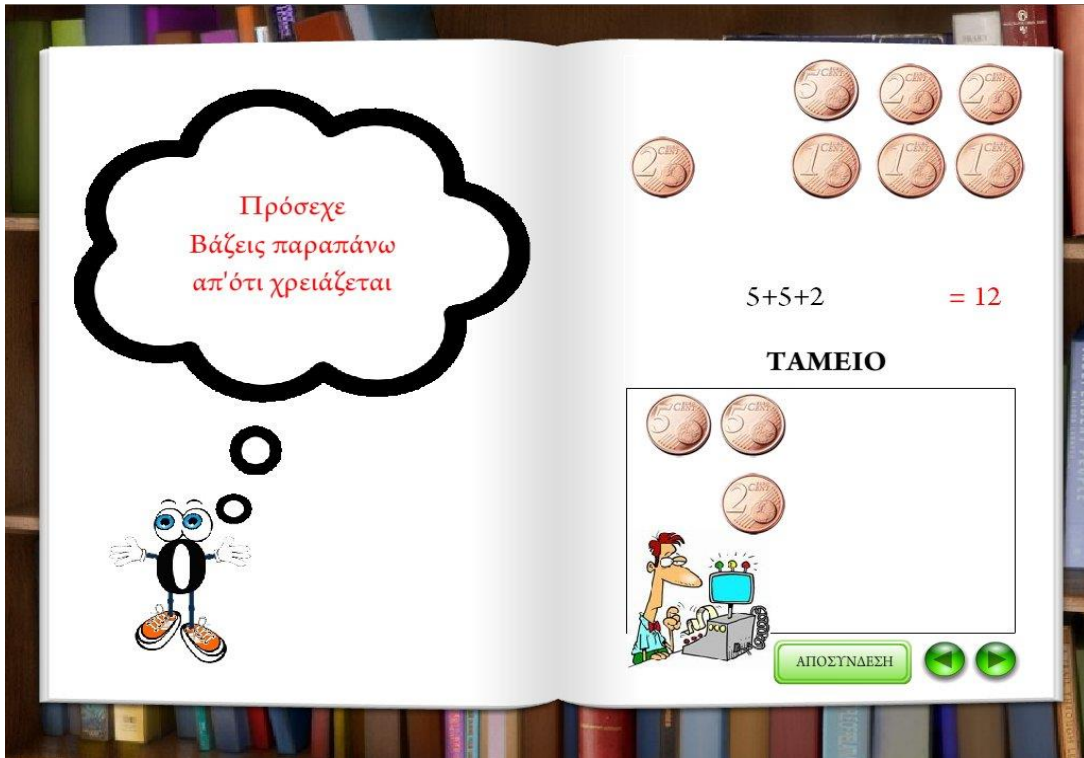
- 2 αντικείμενα τύπου movie clip από τα οποία το ένα αντιπροσωπεύει τον χώρο που βρίσκονται κέρματα στην αρχή και το δεύτερο το ταμείο.
- 1 αντικείμενο τύπου static text με κείμενο «Ταμείο».
- 1 εικόνα που αναπαριστά ένα ταμιά.
- 2 αντικείμενα τύπου dynamic text. Τα αντικείμενα μας αποτελούν το βοήθημα της εφαρμογής για τους χρήστες που έχουν δυσαριθμησία (Εικόνα 7.1). Το 1<sup>ο</sup> αντικείμενο δείχνει με την μορφή πράξεων πόσα κέρματα έχουμε σύρει. π.χ. έστω ότι έχουμε σύρει ένα κέρμα των πέντε λεπτών , ένα κέρμα των 2 λεπτών και ένα κέρμα του ενός λεπτού τότε το βοήθημα θα είχε την εξής μορφή: 5+2+1. Το 2<sup>ο</sup> αντικείμενο εμφανίζει το άθροισμα που προκύπτει από τα κέρματα που έχουμε σύρει στο ταμείο. π.χ έστω ότι έχουμε σύρει ένα κέρμα των πέντε λεπτών , ένα κέρμα των 2 λεπτών και ένα κέρμα του ενός λεπτού τότε το βοήθημα θα εμφανίσει : =8.



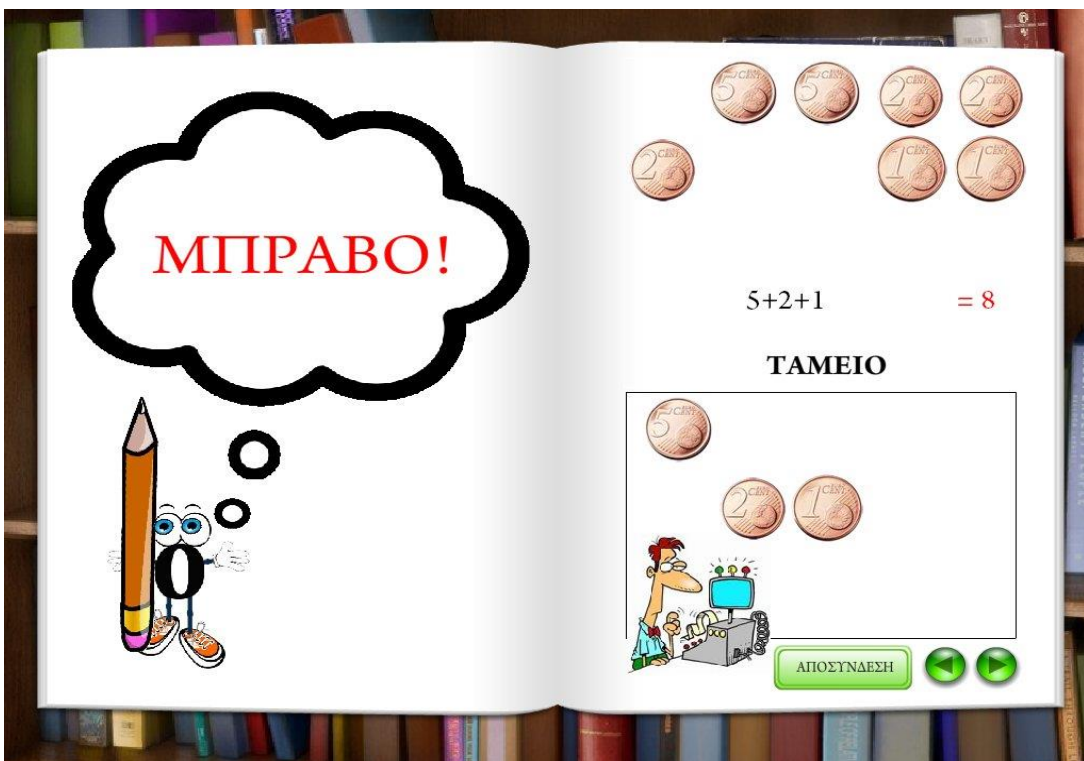
Εικόνα 7.1 Οθόνη 3<sup>ης</sup> άσκησης με βοήθημα.

Αν ο χρήστης σύρει παραπάνω κέρματα από ότι του ζητείται τότε στη φούσκα εμφανίζεται το μήνυμα «Πρόσεχε Βάζεις παραπάνω απ' ότι χρειάζεται» (Εικόνα 7.2) και ταυτόχρονα αναπαράγεται ένα clip ήχου που λέει «Πρόσεχε».

Αν όμως ο χρήστης σύρει όσα κέρματα χρειάζεται στο ταμείο τότε στη φούσκα εμφανίζεται το μήνυμα «ΜΠΡΑΒΟ», αναπαράγεται ένα clip ήχου που λέει «ΜΠΡΑΒΟ» και εμφανίζει τον πράκτορα να κρατά ένα μολύβι (Εικόνα 7.3).



Εικόνα 7.2 Μήνυμα στην φούσκα που προειδοποιεί τον χρήστη.



Εικόνα 7.3 Οθόνη στην οποία ο χρήστης έχει εκτελέσει με επιτυχία την άσκηση.

### **5.2.7 Animation εξόδου**

Μόλις ο χρήστης πατήσει το κουμπί εμπρός στην οθόνη με την 3<sup>η</sup> άσκηση τότε εκτελείται ένα animation το οποίο δείχνει ότι κλείνει το βιβλίο και τοποθετείται πίσω στην βιβλιοθήκη και εκεί κλείνει αυτόματα το παράθυρο της εφαρμογής.

## Βιβλιογραφία

### Ελληνική

Παντελής Μπαλαούρας (2002), Σύγχρονη Τηλεκπαίδευση: Αξιολόγηση της πορείας εισαγωγής της στην Τριτοβάθμια Εκπαίδευση, Εθνικό Καποδιστριακό Πανεπιστήμιο Αθηνών.

Πρέντζας Δ. και Χατζηλυγερούδης Ι. (2001), «Προσαρμοστικά Εκπαιδευτικά Υπερμέσα: Αρχές και Υπηρεσίες». Στο 1ο Πανελλήνιο Συνέδριο για την Ανοικτή και Εξ Αποστάσεως Εκπαίδευση, Ελληνικό Ανοικτό Πανεπιστήμιο, Πάτρα, διαθέσιμο στο:

[http://www.eap.gr/news/EXAGGELIA\\_SYNEDRIOU/synedrio/html/sect5/40.htm](http://www.eap.gr/news/EXAGGELIA_SYNEDRIOU/synedrio/html/sect5/40.htm)

Αθανάσογλου Ι. (2009), Πολυπρακτορικά συστήματα Υφιστάμενη κατάσταση & Συγκριτική μελέτη εφαρμογών, Θεσσαλονίκη.

Keegan, D. (2001), Οι βασικές Αρχές της Ανοικτής και Εξ Αποστάσεως εκπαίδευσης, εκδόσεις: Μεταίχμιο, Αθήνα.

### Ξένη

Brusilovsky, P., Schwarz, E., and Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. Third International Conference on Intelligent Tutoring Systems, 1996, Lecture Notes in Computer Science, Vol. 1086, Springer Verlag, Berlin, 261-269.

Brusilovsky, P. and G. Weber: 1996, "Collaborative Example Selection in an Intelligent Example Based Programming Environment". *International Conference on Learning Sciences, ICLS-96*, Evanston, USA.

Kinny D. and Georgeff M., (1991), The dMARS Architecture: A Specification of the Distributed Multi-Agent Reasoning System, *Autonomous Agents and Multi-Agent Systems*, vol 9 (1-2) pp. 5-53

Hayes-Roth, B. Architectural Foundations for Real-Time Performance in Intelligent Agents. 1990.

Hayes-Roth, B. An Architecture for Adaptive Intelligent Systems. 1995.

Hothi, J. and Hall, W. (1998) An Evaluation of Adapted Hypermedia Techniques Using Static User Modelling. In: *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, USA*. pp. 45-50.

Jennings, N. R. (1995). Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 74(2)

Maes, P. (1994b). Social interface agents: Acquiring competence by learning from users and other agents. In Etzioni, O., editor, *Software Agents - Papers from the 1994 Spring Symposium (Technical Report SS-94-03)*, pages 71-78. AAAI Press.

Michael Wooldridge (2000). Reasoning about Rational Agents. The MIT press. ISBN 0-262-23213-8

Rosenberg, M. (2000), *E-learning: Strategies for delivering knowledge in the digital age*, McGraw-Hill, New York.

Seybold Report on Internet Publishing, The. ISSN: 1090-4808.

Sycara K. & Zeng D., (1996), Distributed Intelligent Agents, *IEEE Expert: Intelligent Systems and Their Applications*, vol 11 (6).

Vassiliou, M., Rowley, J. 2008. 'Progressing the Definition of 'e-book'', *Library Hi Tech*, 26.3,355-368,

<http://www.emeraldinsight.com/Insight/viewPDF.jsp?contentType=Article&FileName=html/Output/Published/EmeraldFullTextArticle/Pdf/2380260303.pdf>.



Wieringa, R. and Dubois, E. (1994). "Integrating Semi-formal and Formal Software Specification Techniques", *Information System Journal*, 19(4), pp. 33-54.

WOOLDRIDGE, M. and JENNINGS, N.R., 1995. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, vol. 10 (2).

## **Ιστοσελίδες**

<http://esp.it.teithe.gr/documents/PE09/prosarmostikaupermesa.doc>

<http://el.wikipedia.org/wiki>

[www.teleteaching.gr](http://www.teleteaching.gr)

<http://portal.gunet.gr/index.pl?iid=3648>

## ΠΑΡΑΡΤΗΜΑ Α

### Α.1 Κώδικας εισόδου χρήστη στο σύστημα

```
1 var loginURL:URLRequest=new URLRequest("http://localhost/login.php");
2 var pageLoader:URLLoader = new URLLoader();
3 var pageVars:URLVariables = new URLVariables();
4 pageLoader.addEventListener(Event.COMPLETE, loadPageSuccessful);
5 pageLoader.addEventListener(IOErrorEvent.IO_ERROR, loadPageError);
6 statusText.text="";
```

Με την βοήθεια του παραπάνω κώδικα ορίζουμε κάποιες μεταβλητές τις οποίες θα χρησιμοποιήσουμε για να φορτώσουμε το αρχείο login.php και να ανταλλάξουμε με αυτό πληροφορίες. Με την χρήση των σειρών 4 και 5 θα είμαστε σε θέση να ξέρουμε αν η φόρτωση ήταν επιτυχής ή αν είχαμε κάποιο σφάλμα. Τέλος 'καθαρίζουμε' το statusText έτσι ώστε να είναι έτοιμο να παρουσιάσει πληροφορίες.

```
1 logBtn.addEventListener(MouseEvent.CLICK, logBtnPressed);
2
3 function logBtnPressed(evt:MouseEvent):void {
4     checkLogin(userInputText.text,passInputText.text);
5 }
```

Το logBtn είναι το κουμπί με ετικέτα «ΕΙΣΟΔΟΣ», μόλις λοιπόν το πατήσει ο χρήστης τότε εκτελείται η συνάρτηση logBtnPressed η οποία με την σειρά της εκτελεί την συνάρτηση checkLogin δίνοντας της δυο ορίσματα. Τα δυο αυτά ορίσματα προέρχονται από τα δυο αντικείμενα τύπου TextInput στα οποία ο χρήστης εισάγει το όνομα χρήστη και τον κωδικό χρήστη.

```
1 function checkLogin(user,pass) {
2     if (user!=""&&pass!="") {
3         logBtn.label="loading...";
4         userInputText.enabled = false;
5         passInputText.enabled = false;
6         pageVars.user=user;
7         pageVars.pass=pass;
8         loginURL.method=URLRequestMethod.POST;
9         loginURL.data=pageVars;
10        pageLoader.load(loginURL);
11    } else {
12        statusText.text="Συμπλήρωσε όλα τα πεδία";
13        logBtn.label="ΕΙΣΟΔΟΣ";
14    }
15 }
```

Η συνάρτηση αυτή καταρχήν ελέγχει αν ο χρήστης συμπλήρωσε και τα δυο πεδία δηλαδή αν κάποιο από τα string δεν είναι κενό αλλιώς του εμφανίζει σχετικό μήνυμα. Στη συνέχεια αλλάζει την ετικέτα από το κουμπί logBtn και κάνει ανενεργά τα δυο textInput έτσι ώστε να μην μπορεί ο χρήστης να αλλάξει τα δεδομένα του την ώρα που το σύστημα τα επεξεργάζεται. Έπειτα βάζει το όνομα χρήστη και των κωδικό χρήστη σε κατάλληλες μεταβλητές, δείχνει τον τρόπο με των οποίων θα αποσταλούν και φορτώνει το αρχείο login.php.

Login.php :

```
1 <?php
2 //enter mySQL details
3 $connection = mysql_connect('localhost', 'root', '123')
4 or die ("Could not connect to mysql because ".mysql_error());
5 //database
6 mysql_select_db('users')
7 or die ("Could not select database because ".mysql_error());
8 $table = 'users';
9 //collecting the sent vars
10 $user = $_POST['user'];
11 $pass = $_POST['pass'];
12 $userQuery = mysql_query("SELECT * FROM $table WHERE username = '$user'");
13 //checking if username exists
14 if(mysql_num_rows($userQuery) == 0){
15 print("loginStatus=username");
16 exit;
17 }
18 //checking if the password is right
19 $passQuery = mysql_query("SELECT * FROM $table WHERE username = '$user' AND password = '$pass'");
20 if(mysql_num_rows($passQuery) == 0){
21 print("loginStatus=password");
22 exit;
23 }
24 //retrieving data
25 $resultQuery = mysql_query("SELECT * FROM $table WHERE username = '$user' AND password = '$pass'");
26 if (!$resultQuery) {
27     print ("Could not successfully run query" );
28     exit;
29 }
30 $row = mysql_fetch_assoc($resultQuery);
31 echo "userLVL=$row[userLevel]&dyscal=$row[dyscalculia]&pageT=$row[page]&";
32
33 //success message
34 print ("loginStatus=success");
35 ?>
```

Ο παραπάνω κώδικας ιδρύει μια σύνδεση με την MySql, επιλέγει την βάση και τον πίνακα. Στην συνέχεια παίρνει τις δυο μεταβλητές που περιέχουν το όνομα χρήστη και τον κωδικό χρήστη που πληκτρολόγησε ο χρήστης και οι οποίες στάλθηκαν από το flash. Εκτελεί ένα ερώτημα SQL για το συγκεκριμένο όνομα χρήστη με σκοπό να δει πόσες εγγραφές θα επιστρέψει αυτό. Αν επιστρέψει 0 εγγραφές αυτό σημαίνει ότι δεν υπάρχει στη βάση τέτοιος χρήστης και εκτυπώνει

print("loginStatus=username"); που θα χρησιμοποιηθεί από τη συνάρτηση loadPageSuccessful του flash , αλλιώς συνεχίζει παρακάτω.

Παρακάτω επίσης εκτελεί ένα ερώτημα SQL το οποίο έχει ως σκοπό να επιστρέψει εγγραφές για τον συγκεκριμένο χρήστη και τον συγκεκριμένο κωδικό.

Αν επιστρέψει 0 εγγραφές τότε ο κωδικός για τον συγκεκριμένο χρήστη δεν είναι σωστός και εκτυπώνει print("loginStatus=password"); που θα χρησιμοποιηθεί από τη συνάρτηση loadPageSuccessful του flash , αλλιώς συνεχίζει παρακάτω.

Τέλος εκτελεί ένα ερώτημα SQL το οποίο έχει ως σκοπό να επιστρέψει εγγραφές για τον συγκεκριμένο χρήστη και τον συγκεκριμένο κωδικό. Αν εκτελεστεί με επιτυχία τότε ο κώδικας εκτυπώνει με την μορφή string ζευγάρια μεταβλητών με τις τιμές τους τις οποίες θα χρησιμοποιήσει το flash, επίσης εκτυπώνει print("loginStatus=success"); Που σημαίνει ότι όλα πήγαν καλά και που θα χρησιμοποιηθεί από τη συνάρτηση loadPageSuccessful του flash.

```
1 function loadPageSuccessful(event:Event):void {
2     var newVars:URLVariables=new URLVariables(pageLoader.data);
3     userInputText.enabled = true;
4     passInputText.enabled = true;
5
6     switch (newVars.loginStatus) {
7         case 'success' :
8             statusText.text='Επιτυχής εισαγωγή';
9             globals.data.userLevel=newVars.userLVL;
10            if(newVars.dyscal=="false"){
11                globals.data.dyscalculia=false;
12            }
13            else if(newVars.dyscal=="true"){
14                globals.data.dyscalculia=true;
15            }
16            globals.data.userName=userInputText.text;
17            globals.data.userPass=passInputText.text;
18
19            gotoAndPlay(newVars.pageT);
20
21            break;
22            case 'username' :
23                statusText.text='Αυτό το όνομα χρήστη δεν υπάρχει';
24                logBtn.label="ΕΙΣΟΔΟΣ";
25                break;
26            case 'password' :
27                statusText.text='Λάθος κωδικός';
28                logBtn.label="ΕΙΣΟΔΟΣ";
29                break;
30        }
31    }
```

Η συνάρτηση `loadPageSuccessful` εκτελείται αφότου εκτελεστεί το αρχείο `login.php` και αυτό γιατί είχαμε βάλει έναν `eventListener` ο οποίος θα ενεργοποιούσε την συνάρτηση `loadPageSuccessful` μόλις ολοκληρωνώταν η εκτέλεση του `login.php`. Η συνάρτηση αυτή φορτώνει σε μια μεταβλητή αυτά που εκτυπώθηκαν στο `login.php` και με την βοήθεια μιας `switch` διακρίνει τρεις περιπτώσεις:

1. Αν εκτυπώθηκε `loginStatus=username` αυτό σημαίνει ότι το συγκεκριμένο όνομα χρήστη που πληκτρολογήθηκε δεν είναι έγκυρο και άρα το `statusText` εμφανίζει το μήνυμα «Αυτό το όνομα χρήστη δεν υπάρχει».
2. Αν εκτυπώθηκε `loginStatus=password` αυτό σημαίνει ότι ο κωδικός που πληκτρολογήθηκε για το συγκεκριμένο χρήστη δεν είναι έγκυρος και άρα το `statusText` εμφανίζει το μήνυμα «Λάθος κωδικός».
3. Αν εκτυπώθηκε `loginStatus=success` αυτό σημαίνει ότι όλα πήγαν καλά και άρα το `statusText` εμφανίζει το μήνυμα «Επιτυχής εισαγωγή».

Στη συνέχεια το `flash` αφού εξάγει πληροφορίες από το `string` που εκτυπώθηκε στην γραμμή 31 του αρχείου `login.php`, τις χρησιμοποιεί. Πιο συγκεκριμένα στη μεταβλητή `globals.data.userLevel` αποθηκεύει το επίπεδο του χρήστη που ανέκτησε από τη βάση, στη μεταβλητή `globals.data.dyscalculia` αποθηκεύει την τιμή που ανέκτησε από τη βάση και τέλος χρησιμοποιεί την τιμή της μεταβλητής `pageT` που ανέκτησε από την βάση. Η τιμή αυτή είναι ο αριθμός του `frame` στο οποίο ο χρήστης είχε κάνει πιο παλιά αποσύνδεση `gotoAndPlay(newVars.pageT);`

```
1 function loadPageError($evt:IOErrorEvent):void {
2     statusText.text="Σφάλμα με τον server";
3     userInputText.enabled = true;
4     passInputText.enabled = true;
5     logBtn.label="ΕΙΣΟΔΟΣ";
6 }
```

Ο παραπάνω κώδικας εκτελείται σε περίπτωση που υπάρξει πρόβλημα με την εκτέλεση/φόρτωση του αρχείου `login.php`. Σε αυτή την περίπτωση το `statusText` εμφανίζει το μήνυμα «Σφάλμα με τον server» και προειδοποιεί τον χρήστη για το γεγονός αυτό.

## A.2 Κώδικας εγγραφής χρήστη στο σύστημα

```
1 var cUserURL:URLRequest=new URLRequest("http://localhost/newAccount.php");
2 var pageLoader1:URLLoader = new URLLoader();
3 pageLoader1.addEventListener(Event.COMPLETE, loadPageSuccessful);
4 pageLoader1.addEventListener(IOErrorEvent.IO_ERROR, loadPageError);
5 statusText.text="";
```

Ο κώδικας αυτός είναι σχεδόν ίδιος με αυτόν που χρησιμοποιήσαμε πιο πάνω για την εισαγωγή του χρήστη στο σύστημα. Αυτή τη φορά θα χρειαστεί να φορτώσουμε το αρχείο newAccount.php.

```
1 regBtn.addEventListener(MouseEvent.CLICK, regBtnPressed);
2
3 function regBtnPressed(evt:MouseEvent):void {
4     newAccount(userTextInput.text,passTextInput.text);
5 }
```

Το regBtn είναι το κουμπί με ετικέτα «ΕΓΓΡΑΦΗ», μόλις λοιπόν το πατήσει ο χρήστης τότε εκτελείται η συνάρτηση regBtnPressed η οποία με την σειρά της εκτελεί την συνάρτηση newAccount δίνοντας της δυο ορίσματα. Τα δυο αυτά ορίσματα προέρχονται από τα δυο αντικείμενα τύπου textInput στα οποία ο χρήστης εισάγει το όνομα χρήστη και τον κωδικό χρήστη που επιθυμεί.

```
1 function newAccount(user,pass):void {
2     if (user!=""&&pass!="") {
3         regBtn.label="loading...";
4         userTextInput.enabled = false;
5         passTextInput.enabled = false;
6         pageVars.user=user;
7         pageVars.pass=pass;
8         cUserURL.method=URLRequestMethod.POST;
9         cUserURL.data=pageVars;
10        pageLoader1.load(cUserURL);
11    } else {
12        statusText.text="Συμπλήρωσε όλα τα πεδία";
13        regBtn.label="ΕΓΓΡΑΦΗ";
14    }
15 }
```

Καταρχήν αυτή η συνάρτηση ελέγχει αν ο χρήστης συμπλήρωσε και τα δυο πεδία αλλιώς του εμφανίζει σχετικό μήνυμα. Στη συνέχεια αλλάζει την ετικέτα από το κουμπί regBtn και κάνει ανενεργά τα δυο textInput έτσι ώστε να μην μπορεί ο

χρήστη να αλλάξει τα δεδομένα του την ώρα που το σύστημα τα επεξεργάζεται. Έπειτα βάζει το όνομα χρήστη και τον κωδικό χρήστη σε κατάλληλες μεταβλητές, δείχνει τον τρόπο με τον οποίο θα αποσταλούν και φορτώνει το αρχείο newAccount.php.

newAccount.php:

```
1 <?php
2 //enter mySQL details
3 $connection = mysql_connect('localhost', 'root', '123')
4 or die ("Could not connect to mysql because ".mysql_error());
5 //database
6 mysql_select_db('users')
7 or die ("Could not select database because ".mysql_error());
8 $table = 'users';
9 //checking if username exists
10 $userQuery = mysql_query("SELECT * FROM $table WHERE username = '".$_POST['user']."'");
11 //if username is used the var accountstatus gets sent back with username
12 if(mysql_num_rows($userQuery) != 0){
13 print("accountStatus=username");
14 exit;
15 }
16 //inserting values in to the database and adding md5 encryption to password
17 mysql_query("INSERT into $table (username,password,userLevel,dyscalculia,page)
18 VALUES ('".$_POST['user']."','".$_POST['pass']."',0,'false',142)");
19 print("accountStatus=success");
20 print("&end=end");
21 ?>
```

Στις γραμμές 3-8 αυτός ο κώδικας ιδρύει μια σύνδεση με την MySQL, επιλέγει την βάση και τον πίνακα, αν συμβεί κάποιο σφάλμα σε αυτή τη διαδικασία τότε εκτυπώνει το μήνυμα «Could not connect to mysql because» συν το αντίστοιχο σφάλμα mysql. Έπειτα στις γραμμές 10-14 εκτελεί ένα ερώτημα SQL για να μας επιστρέψει εγγραφές από τον πίνακα για το όνομα χρήστη που επιθυμεί και έχει εισάγει ο χρήστης πιο πριν στο αντίστοιχο textInput. Αν επιστραφούν εγγραφές αυτό σημαίνει ότι αυτό το όνομα χρήστη υπάρχει και χρησιμοποιείται από άλλο χρήστη έτσι εκτυπώνει print("accountStatus=username"); (Που θα χρησιμοποιηθεί παρακάτω) και κλείνει. Παρόλα αυτά αν το όνομα χρήστη δεν χρησιμοποιείται δηλαδή επιστραφούν 0 εγγραφές, τότε εκτελείται ένα ερώτημα SQL το οποίο εισάγει στον πίνακα μας μια νέα εγγραφή με τα εξής στοιχεία:

- Το πεδίο username του πίνακα παίρνει το όνομα χρήστη που πληκτρολόγησε ο χρήστης.
- Το πεδίο password του πίνακα παίρνει τον κωδικό χρήστη που πληκτρολόγησε ο χρήστης.
- Το πεδίο userLevel παίρνει την προεπιλεγμένη τιμή 0 καθώς αρχικά θεωρούμε ότι ο χρήστης βρίσκεται στο εύκολο επίπεδο.

- Το πεδίο `dyscalculia` παίρνει την προεπιλεγμένη τιμή `false` καθώς αρχικά υποθέτουμε ότι ο χρήστης δεν έχει δυσαριθμησία.
- Τέλος το πεδίο `page` παίρνει την προεπιλεγμένη τιμή `142` η οποία είναι ένας αριθμός `frame` ο οποίος αντιστοιχεί στην οθόνη εισόδου του χρήστη στο σύστημα.

Αφου γίνουν όλα αυτά εκτυπώνει `print("accountStatus=success");` Και `print("&end=end");` (Που θα χρησιμοποιηθεί παρακάτω) και κλείνει.

```

1 function loadPageSuccessfull(event:Event):void {
2     var newVars:URLVariables=new URLVariables(pageLoader1.data);
3     userTextInput.enabled = true;
4     passTextInput.enabled = true;
5     switch (newVars.accountStatus) {
6         case 'success' :
7             statusText.text='Ο χρήστης δημιουργήθηκε \n Πάτήστε ΕΙΣΟΔΟΣ';
8             regBtn.label="ΕΙΣΟΔΟΣ";
9             globals.data.userName=userTextInput.text;
10            globals.data.userPass=passTextInput.text;
11            stage.removeEventListener("mouseMove", eyesFollow);
12            gotoAndPlay(117);
13            break;
14            case 'username' :
15                statusText.text='Αυτό το όνομα χρήστη υπάρχει ήδη';
16                regBtn.label="ΕΓΓΡΑΦΗ";
17                break;
18        }
19    }

```

Ο παραπάνω κώδικας εκτελείται αφού εκτελεστεί το αρχείο `newAccount.php`. και αυτό γιατί είχαμε βάλει έναν `eventListener` ο οποίος θα ενεργοποιούσε την συνάρτηση `loadPageSuccessfull` μόλις ολοκληρωνόταν η εκτέλεση του `newAccount.php`. Η συνάρτηση αυτή φορτώνει σε μια μεταβλητή αυτά που εκτυπώθηκαν στο `newAccount.php` και με την βοήθεια μιας `switch` διακρίνει δυο περιπτώσεις:

1. Αν το `accountStatus` έχει πάρει την τιμή `username` αυτό σημαίνει ότι το όνομα χρήστη που διάλεξε ο χρήστης μας χρησιμοποιείται και άρα εκτυπώνει το μήνυμα «Αυτό το όνομα χρήστη υπάρχει ήδη» προτρέποντας τον να γράψει κάποιο άλλο.
2. Αν το `accountStatus` έχει πάρει την τιμή `success` αυτό σημαίνει ότι ο χρήστης μας έχει εγγραφεί με επιτυχία στο σύστημα και κάνει τα εξής: το `statusText` εμφανίζει το κείμενο «Ο χρήστης δημιουργήθηκε» και από κάτω «Πάτήστε ΕΙΣΟΔΟΣ» στην συνέχεια αποθηκεύει σε δυο `global` μεταβλητές το όνομα



χρήστη και τον κωδικό και πάει στο frame 117 το οποίο με τη βοήθεια ενός animation μας πάει στην οθόνη του ερωτηματολογίου.

```
1 function loadPageError($evt:IOErrorEvent):void {
2     statusText.text="Σφάλμα με τον server";
3     userInputText.enabled = true;
4     passInputText.enabled = true;
5     logBtn.label="ΕΙΣΟΔΟΣ";
6 }
```

Ο παραπάνω κώδικας εκτελείται σε περίπτωση που υπάρξει πρόβλημα με την εκτέλεση/φόρτωση του αρχείου newAccount.php. Σε αυτή την περίπτωση το statusText εμφανίζει το μήνυμα «Σφάλμα με τον server» και προειδοποιεί τον χρήστη για το γεγονός αυτό.

### A.3 Κώδικας ανανέωσης της βάσης

```
1 logOutBtn.addEventListener(MouseEvent.CLICK, logOutBtnPressed);
2
3 function logOutBtnPressed(evt:MouseEvent):void {
4     stage.removeEventListener("mouseMove", eyesFollow);
5     updateTable();
6     gotoAndPlay(1);
7 }
```

Το logOutBtn είναι το κουμπί με ετικέτα «ΑΠΟΣΥΝΔΕΣΗ» και το οποίο πατάει ο χρήστης μόλις θέλει να αποσυνδεθεί από το σύστημα. Με το πάτημα αυτού του κουμπιού αφαιρείται από το stage ένας eventListener ο οποίος είναι υπεύθυνος να εκτελεί την συνάρτηση eyesFollow κάθε φορά που κουνιέται το ποντίκι (κώδικας που θα εξετάσουμε παρακάτω). Στη συνέχεια εκτελεί την συνάρτηση updateTable() η οποία ενημερώνει τον πίνακα και αφού γίνει αυτό παίζει το frame 1.

```

1 function updateTable():void {
2     var updateURL:URLRequest=new URLRequest("http://localhost/update.php");
3     var updatePageLoader:URLLoader = new URLLoader();
4     var updatePageVars:URLVariables = new URLVariables();
5     if (pageVars.user!=""&&pageVars.pass!="") {
6         updatePageVars.user=globals.data.userName;
7         updatePageVars.pass=globals.data.userPass;
8         updatePageVars.dyscalculia=String(globals.data.dyscalculia);
9         updatePageVars.page=currentFrame;
10        updatePageVars.userLevel=globals.data.userLevel;
11        updateURL.method=URLRequestMethod.POST;
12        updateURL.data=updatePageVars;
13        updatePageLoader.load(updateURL);
14    }
15 }

```

Η συνάρτηση αυτή με λίγα λόγια δημιουργεί μια μεταβλητή στην οποία στην συνέχεια αποθηκεύει τις τιμές των ιδιοτήτων του χρήστη, επιλέγει την μέθοδο με την οποία θα αποστείλει τα δεδομένα και τέλος φορτώνει το αρχείο update.php.

update.php:

```

1 <?php
2 //enter mySQL details
3 $connection = mysql_connect('localhost', 'root', '123')
4 or die ("Could not connect to mysql because ".mysql_error());
5 //database
6 mysql_select_db('users')
7 or die ("Could not select database because ".mysql_error());
8 $table = 'users';
9 //collecting the sent vars
10 $user = $_POST['user'];
11 $pass = $_POST['pass'];
12 $userLevel = $_POST['userLevel'];
13 $dyscalculia = $_POST['dyscalculia'];
14 $page = $_POST['page'];
15 $userQuery = mysql_query("SELECT * FROM $table WHERE username = '$user'");
16 //checking if username exists
17 if(mysql_num_rows($userQuery) == 0){
18     exit;
19 }
20 //checking if the password is right
21 $passQuery = mysql_query("SELECT * FROM $table WHERE username = '$user' AND password = '$pass'");
22 if(mysql_num_rows($passQuery) == 0){
23     exit;
24 }
25 $updateTableQuery = mysql_query("UPDATE $table SET userLevel = '$userLevel',
26 dyscalculia = '$dyscalculia' , page = '$page' WHERE username = '$user' AND password = '$pass'");
27 if(mysql_num_rows($updateTableQuery) == 0){
28     exit;
29 }
30 ?>

```

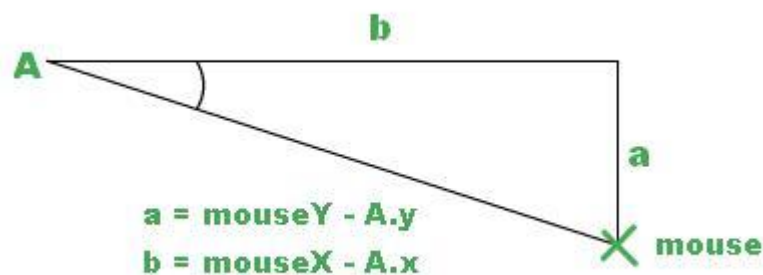
Στις γραμμές 3-8 αυτός ο κώδικας ιδρύει μια σύνδεση με την MySQL, επιλέγει την βάση και τον πίνακα, αν συμβεί κάποιο σφάλμα σε αυτή τη διαδικασία τότε εκτυπώνει το μήνυμα «Could not connect to mysql because» συν το αντίστοιχο σφάλμα mysql. Έπειτα στις γραμμές 10-14 ορίζει μεταβλητές και αποθηκεύει σε αυτές τις τιμές των αντίστοιχων μεταβλητών που έλαβε από την συνάρτηση

updateTable()). Παρακάτω στις γραμμές 15-24 ελέγχει αν το όνομα χρήστη και ο κωδικός χρήστη είναι έγκυροι. Τέλος εκτελεί ένα ερώτημα SQL με το ποίο ενημερώνει τα πεδία του πίνακα με τις νέες τιμές.

#### A.4 Κώδικας για να ακολουθούν τα μάτια του πράκτορα το ποντίκι

```
1 stage.addEventListener("MouseMove", eyesFollow);
2
3 function eyesFollow(e:MouseEvent):void {
4     var a1 = mouseY - numCartoonEye1.y;
5     var b1 = mouseX - numCartoonEye1.x;
6     var radians1 = Math.atan2(a1,b1);
7     var degrees1 = radians1 / (Math.PI / 180);
8     numCartoonEye1.rotation = degrees1;
9
10    var a2 = mouseY - numCartoonEye2.y;
11    var b2 = mouseX - numCartoonEye2.x;
12    var radians2 = Math.atan2(a2,b2);
13    var degrees2 = radians2 / (Math.PI / 180);
14    numCartoonEye2.rotation = degrees2;
15 }
```

Προσθέτουμε στο stage έναν eventListener ο οποίος κάθε φορά που κουνιέται το ποντίκι πρέπει να εκτελεί την συνάρτηση eyesFollow. Η βασική ιδέα της συνάρτησης είναι ότι θα πρέπει να γυρίζουμε τα μάτια σε σχέση με το ποντίκι. Για να το κάνουμε αυτό θα υπολογίσουμε την γωνία που σχηματίζεται μεταξύ του ματιού και του ποντικιού.

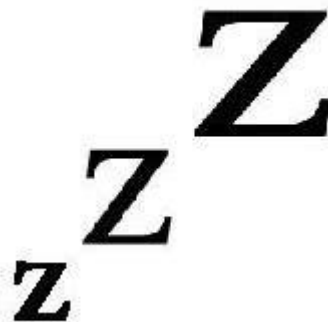


Για να βρούμε την γωνία θα χρησιμοποιήσουμε την συνάρτηση **atan2** του Flash η οποία θα έχει ως ορίσματα την θέση Y και X του ποντικιού. Η γωνία όμως που επιστρέφει η atan2 είναι σε ακτίνια όποτε πρέπει να τα μετατρέψουμε σε μοίρες, αυτό γίνεται με την βοήθεια της γραμμής 7 και 13. Τέλος γυρνάμε το κάθε μάτι ανάλογα με την γωνία μας.

## A.5 Κώδικας ανενεργίας χρήστη

```
1 stage.addEventListener("mouseMove", eyesFollow);
2 var timeToSleep:Number=30000;
3 var myInterval:uint = setInterval (timeToSleepFunction, timeToSleep);
4
5 function eyesFollow(e:MouseEvent):void {
6
7     if(currentFrame==89 || currentFrame==142 || currentFrame==116 ){
8         TTS();
9     }
10    else{
11        clearInterval(myInterval);
12    }
13 }
14
15 function timeToSleepFunction():void{
16     if(currentFrame==89 || currentFrame==142 || currentFrame==116 ){
17         cloud.alpha=0;
18         cloudTTS.play();
19         cloudTTS.alpha=100;
20     }
21 }
22
23 function TTS():void{
24     clearInterval(myInterval);
25     myInterval = setInterval (timeToSleepFunction, timeToSleep);
26     timeToSleep=30000;
27     cloud.alpha=100;
28     cloudTTS.stop();
29     cloudTTS.alpha=0;
30 }
```

Με την βοήθεια της `setInterval` θέλουμε να εκτελείται κάθε 30000 milliseconds (30 δευτερόλεπτα) η συνάρτηση `timeToSleepFunction` η οποία έχει ως σκοπό να ελέγξει αν βρισκόμαστε στο κατάλληλο frame και στη συνέχεια να αποκρύψει την φούσκα του πράκτορα και να εμφανίσει το `cloudTTS` το οποίο είναι το εξής:



Τέλος κάθε φορά που κουνάμε το ποντίκι παράλληλα με την εκτέλεση του κώδικα που είναι υπεύθυνος για την κίνηση των ματιών (του πράκτορα) εκτελείται και η συνάρτηση TTS() η οποία έχει ως σκοπό να κάνει την timeToSleepFunction να ξεκινήσει σε 30 δευτερόλεπτα και τέλος να αποκρύψει το cloudTTS και να εμφανίσει την φούσκα του πράκτορα.

## A.6 Κώδικας κίνησης του πράκτορα στο επιτραπέζιο της Άσκησης 2

```
1 function forwardWalk():void{
2     cartoonSmile.alpha=100;
3     cartoonSadSmile.alpha=0;
4     if(CinRow3<3){
5         cloudBoard.CinRow.alpha=0;
6         cloudBoard.IncInRow.alpha=0;
7         cloudBoard.StringBoardGame.alpha=100;
8         CinRow3++;
9         IncInRow3=1;
10    }
11    else if(CinRow3==3 && cartX < 15){
12        cloudBoard.CinRow.alpha=100;
13        cloudBoard.IncInRow.alpha=0;
14        cloudBoard.StringBoardGame.alpha=0;
15        CinRow3=1;
16        IncInRow3=1;
17        bravoSSC.play();
18    }
19    if(cartX==15){
20        cartX=16;
21        cartY=16;
22        MovieClip(this.parent).ex2PageL.numberCartoonBGame.x=cartoonCoord[0][16];
23        MovieClip(this.parent).ex2PageL.numberCartoonBGame.y=cartoonCoord[1][16];
24        eftaseF();
25    }
26    else{
27        MovieClip(this.parent).ex2PageL.numberCartoonBGame.x=cartoonCoord[0][++cartX];
28        MovieClip(this.parent).ex2PageL.numberCartoonBGame.y=cartoonCoord[1][++cartY];
29    }
```

Ο κώδικας αυτός εκτελείται κάθε φορά που ο χρήστης κάνει σωστά μια πράξη. Έτσι στις γραμμές 2-3 εμφανίζουμε το χαμόγελο και εξαφανίζουμε το λυπημένο χαμόγελο. Στις γραμμές 4-18 ελέγχουμε αν ο χρήστης έχει δώσει τρεις σωστές απαντήσεις στη σειρά, αν ναι τότε εξαφανίζουμε την κανονική φούσκα και εμφανίζουμε μια άλλη που μέσα της λέει «Μπράβο Συνέχισε έτσι», παράλληλα αναπαράγεται ένα clip ήχου που και αυτό λέει «Μπράβο Συνέχισε έτσι». Αν ο χρήστης δεν έχει δώσει τρεις σωστές απαντήσεις στη σειρά τότε αυξάνεται ο

μετρητής των σωστών απαντήσεων στη σειρά και αρχικοποιείται αυτός των λάθος απαντήσεων στη σειρά.

Στις γραμμές 19-29 ελέγχουμε αν ο πράκτορας έχει φτάσει στο 15<sup>ο</sup> κουτάκι.

Αν όχι τότε απλά τον μετακινούμε στο επόμενο κουτάκι. Αυτό γίνεται με την βοήθεια συντεταγμένων X και Y που αντιστοιχούν σε κάθε κουτάκι και τις οποίες έχουμε αποθηκεύσει σε ένα δισδιάστατο πίνακα.

Αν ναι τότε τον πάμε στο κουτάκι 16 και εκτελούμε την συνάρτηση eftaseF().

```
1 function eftaseF():void{
2     bravoSC.play();
3     cartoonSmile.alpha=0;
4     cartoonSadSmile.alpha=0;
5     eftase=true;
6     globals.data.ex3=true;
7     ex2Apotelesma.editable=false;
8     cloudBoard.StringBoardGame.alpha=0;
9     cloudBoard.bravoBoardGame.alpha=100;
10    numCartoonBoardGame.alpha=0;
11    numCartoonApp.alpha=100;
12 }
```

Με το που εκτελείται αυτή η συνάρτηση γίνονται τα εξής:

- αναπαράγεται ένα clip ήχου που και αυτό λέει «Μπράβο»
- ειδοποιούμε την επόμενη άσκηση ότι αυτή ολοκληρώθηκε με επιτυχία
- απενεργοποιούμε το textInput στο οποίο ο χρήστης έγραφε τις απαντήσεις του
- εμφανίζουμε μέσα στη φούσκα το μήνυμα «Μπράβο» και αποκρύπτουμε το προεπιλεγμένο
- εμφανίζουμε έναν άλλο πράκτορα ο οποίος χειροκροτεί

Αντίστοιχος του κώδικα με τον οποίο μετακινείται ο πράκτορας ένα κουτάκι μπροστά υπάρχει και αυτός με τον οποίο μετακινείται ένα κουτάκι πίσω.