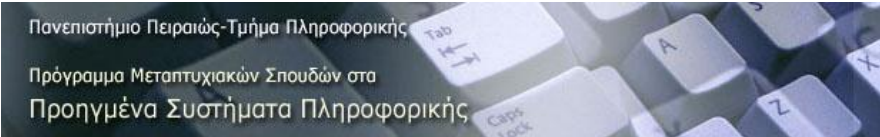




Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον
Όνοματεπώνυμο Φοιτητή	Ζούγρης Αναστάσιος του Παναγιώτη
Αριθμός Μητρώου	ΜΠΣΠ 08009
Κατεύθυνση	Ευφυείς Τεχνολογίες Αλληλεπίδρασης Ανθρώπου-Υπολογιστή
Επιβλέπων	Μαρία Βίβου, Καθηγήτρια



Πανεπιστήμιο Πειραιώς-Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών στα
Προηγμένα Συστήματα Πληροφορικής

Ημερομηνία Παράδοσης **Οκτώβριος 2011**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

Μαρία Βίρβου
Καθηγήτρια

(υπογραφή)

Γεώργιος Τσιχριντζής
Καθηγητής

(υπογραφή)

Ευάγγελος Φούντας
Καθηγητής

Μεταπτυχιακή Διατριβή

- Το κείμενο να είναι γραμμένο σε font size 10pt με μονό διάστιχο (single spacing) και απόσταση παραγράφων 3pt (after).
- Η πρώτη παράγραφος κάθε ενότητας να μην έχει εσοχή πρώτης γραμμής ενώ οι επόμενες να έχουν εσοχή πρώτης γραμμής.
- Τα περιθώρια σελίδας να είναι 3cm και στις τέσσερις πλευρές (πάνω, κάτω, αριστερά, δεξιά).
- Τα Headings να είναι όλα με font **Arial Black** και όχι Bold. Το **Heading 1 να είναι 12pt**, το **Heading 2 να είναι 11pt**, το **Heading 3 να είναι 10pt**. Να μην χρησιμοποιείτε Heading 4 και πέρα.
- Να μην αφήνετε κενές γραμμές πριν ή μετά από τα headings και κάθε επίπεδο heading να απέχει 18pt before και 6pt after.
- Οι λεζάντες (captions) στα σχήματα και τους πίνακες να είναι αριστερά στοιχισμένες και να είναι **Arial bold 9pt**.
- Σε κάθε σελίδα να υπάρχει footer (Arial 8pt) με τον τίτλο της διατριβής στα αριστερά. Στο footer επίσης να υπάρχει αρίθμηση σελίδας στα δεξιά και πάλι με font Arial 8pt.
- Σε κάθε σελίδα να υπάρχει header (Arial 8pt) με το όνομα του φοιτητή στα δεξιά και το λεκτικό «Μεταπτυχιακή Διατριβή» στα αριστερά.
- Τα header και footer να απέχουν από τα άκρα του χαρτιού 2.5cm (στο Page Setup).
- Η διατριβή να περιέχει απαραίτητα:
 - Περίληψη (Abstract) σε χωριστή σελίδα (μισή σελίδα Ελληνικά και μισή στα Αγγλικά).
 - Εισαγωγή – Σύντομη Περιγραφή Προβλήματος/Αντικειμένου (μέχρι 3 σελίδες).
 -
 - Συμπεράσματα – Περίληψη
 - Βιβλιογραφία

ΠΕΡΙΕΧΟΜΕΝΑ

1	Περίληψη.....	4
2	Εισαγωγή.....	5
3	Επισκόπηση Πεδίου.....	6
3.1	Μοντελοποίηση Χρήστη.....	6
3.1.1	Στρατηγικές εξατομίκευσης.....	6
3.1.2	Μαθησιακός προσανατολισμός.....	7
3.2	Προσαρμοστικά Εκπαιδευτικά Περιβάλλοντα.....	8
3.2.1	Παραδείγματα εφαρμογών.....	9
3.2.2	Συμπεράσματα.....	11
4	Περιγραφή ανάγκης για την εφαρμογή Tic-Tac.....	13
4.1	Διδακτική της ανάγνωσης της ώρας.....	13
4.2	Μαθησιακές δυσκολίες.....	14
5	Υπάρχουσες λύσεις.....	16
5.1	Ελληνόφωνο υλικό.....	16
5.2	Αγγλόφωνο υλικό.....	19
6	Τρόπος παρουσίασης θεωρίας και υπόβαθρο.....	21
6.1	Η λειτουργικότητα της εφαρμογής.....	21
6.2	Σενάριο χρήσης.....	28
7	Υλοποίηση.....	36
7.1	Παρουσίαση της εφαρμογής.....	36
7.1.1	Το Περιβάλλον Υλοποίησης.....	36
7.1.2	Αντικειμενοστρεφής Προγραμματισμός.....	36
7.1.3	Φόρμες.....	37
7.1.4	Βοηθητικές Βιβλιοθήκες και Custom Controls.....	43
7.1.5	Βάση Δεδομένων - Πίνακες και συσχετίσεις.....	45
7.2	Μοντελοποίηση των χρηστών της εφαρμογής.....	51
7.3	Προσαρμοστικότητα.....	57
8	Συμπεράσματα – εφαρμογές.....	63
8.1	Συμπεράσματα.....	63
8.2	Επεκτάσεις.....	63
9	Βιβλιογραφία.....	64
10	Παράρτημα.....	65
10.1	Κώδικας - frmLogin.cs.....	65
10.2	Κώδικας - frmMain.cs.....	68
10.3	Κώδικας - frmPracticeRoom.cs.....	70

10.4	Κώδικας - frmSettings.cs.....	72
10.5	Κώδικας - frmStatisticsRoom.cs	74
10.6	Κώδικας - frmStudyRoom.cs	80
10.7	Κώδικας - AdviceGenerator.cs	82
10.8	Κώδικας - Answer.cs.....	90
10.9	Κώδικας - ClockParams.cs.....	91
10.10	Κώδικας - Question.cs	92
10.11	Κώδικας - Randomizer.cs.....	93
10.12	Κώδικας - Settings.cs.....	95
10.13	Κώδικας - SimilarityExtensions.cs	96
10.14	Κώδικας - StringEnum.cs	116
10.15	Κώδικας - StringValueAttribute.cs	121
10.16	Κώδικας - StudySession.cs	122
10.17	Κώδικας - Test.cs.....	122
10.18	Κώδικας - TestEvaluator.cs.....	124
10.19	Κώδικας - TestVisualizer.cs	127
10.20	Κώδικας - User.cs	136
10.21	Κώδικας - VisibilityFunctions.cs.....	138
10.22	Κώδικας - ErrorTypes.cs	138
10.23	Κώδικας - QuestionTypes.cs.....	138
10.24	Κώδικας - Skins.cs.....	139
10.25	Κώδικας - TimeZones.cs.....	140
10.26	Κώδικας - Clock.vb	141

1 Περίληψη

Στα πλαίσια της μεταπτυχιακής αυτής διατριβής αναπτύχθηκε μία εφαρμογή για την εκμάθηση της ώρας που απευθύνεται σε παιδιά ηλικίας 5-9 ετών. Η εφαρμογή αυτή υλοποιεί μοντελοποίηση και προσαρμοστικότητα χρηστών. Το κεφάλαιο 2 αποτελεί μία εισαγωγή στην εξατομικευμένη διδασκαλία και ακολουθεί μία επισκόπηση των πεδίων της μοντελοποίησης και της προσαρμοστικότητας στο κεφάλαιο 3. Στα κεφάλαια 4 και 5 γίνεται μία ανάλυση της ανάγκης για την εφαρμογή TicTac και μία εμβάθυνση στις τρέχουσες υπάρχουσες λύσεις. Η λειτουργικότητα της εφαρμογής καθώς και αναλυτικά σενάρια χρήσης περιγράφονται στο κεφάλαιο 6. Ό,τι αφορά την υλοποίηση και τον σχεδιασμό της εφαρμογής και της βάσης δεδομένων συγκεντρώνεται στο κεφάλαιο 7. Πιθανές επεκτάσεις και συμπεράσματα υπάρχουν στο κεφάλαιο 8 και βιβλιογραφία στο 9. Στο κεφάλαιο 10 περιλαμβάνεται ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση και χωρίζεται ανά κλάση ή φόρμα.

During this master thesis implementation, an application was developed, with the objective to teach time to children of age 5-9 years. This application implements modeling and user adaptability. Chapter 2 is an introduction to individualized teaching, followed by an overview of the fields of modeling and adaptability in Chapter 3. In Chapters 4 and 5 is an analysis of the need to implement TicTac and a deepening current solutions. The functionality of the application and usage scenarios are described in detail in Chapter 6. All implementation and application design issues and the database development are covered in Chapter 7. Possible extensions and conclusions are in Chapter 8 and the literature in Chapter 9. Finally Chapter 10 contains the code used to implement TicTac and is divided by class or form.

2 Εισαγωγή

Βασικός στόχος των σύγχρονων ερευνών αποτελεί η εύρεση λύσεων που θα μετατρέψουν την εκπαίδευση σε μια εξατομικευμένη και μαθητοκεντρική διαδικασία. Προς αυτή την κατεύθυνση, η εξατομικευμένη μάθηση αποτελεί κυρίαρχη τάση, αφού στοχεύει στην παροχή μαθησιακού υλικού που καλύπτει τις προσωπικές ανάγκες και τις επιδιώξεις του κάθε χρήστη ξεχωριστά. Παράλληλα, η ηλεκτρονική μάθηση αποτελεί έναν τομέα που τεχνολογικά επιτρέπει την ανάπτυξη εξατομικευμένων προσαρμοστικών εφαρμογών, οι οποίες θα προσφέρουν εξατομικευμένη μάθηση καλύπτοντας τις επιμέρους ανάγκες των εκπαιδευόμενων. Η ανάπτυξη διαδικτυακών εξατομικευμένων εκπαιδευτικών εφαρμογών παρουσιάζει σημαντική πρόοδο και αναπτύσσει λύσεις που βασίζονται στην αρχιτεκτονική αντικειμένων και στην προσαρμοστική τεχνολογία. Παρόλα αυτά, από τα υπάρχοντα προσαρμοστικά εκπαιδευτικά περιβάλλοντα κατά κανόνα λείπει η κατανόηση του τρόπου με τον οποίο οι εκπαιδευόμενοι μαθαίνουν και ειδικότερα, τα περιβάλλοντα αυτά συνήθως αδυνατούν να παρακολουθούν τους παράγοντες που συμβάλλουν στη διαδικασία απόκτησης γνώσης.

Η ανθρώπινη γνώση και σκέψη πηγάζουν στην ουσία από τα διακριτά χαρακτηριστικά των ατόμων και την ίδια τη φύση της ανθρώπινης επικοινωνίας. Η μάθηση είναι αποτέλεσμα αλληλεπίδρασης ετερόκλητων παραγόντων, ψυχολογικών και κοινωνικών, όπως το γενικότερο κλίμα της τάξης στη διαδικασία της μάθησης, τις επιμέρους στρατηγικές μάθησης, το περιβάλλον, τις χρησιμοποιούμενες μαθησιακές θεωρίες και μοντέλα διδασκαλίας, τα μαθησιακά στυλ, το γνωστικό υπόβαθρο, αλλά και τις δεξιότητες του κάθε μαθητή.

Η εξατομικευμένη διδασκαλία είναι ένας διδακτικός τρόπος, ο οποίος στηρίζεται πάνω στην συνειδητοποίηση του γεγονότος πως τα παιδιά είναι όλα διαφορετικά το ένα από το άλλο, πως παρουσιάζουν το καθένα διαφορετικές ικανότητες, διαφορετικές κλίσεις και πως κατέχουν το καθένα χαρακτηριστικά που είναι ολότελα δικά τους. Το σχολείο είναι αυτό το οποίο οφείλει να ασχοληθεί επισταμένα με την ατομικοποίηση της παιδαγωγικής μεταχείρισης των μαθητών. Η αναζήτηση προς αποτελεσματικότερες διδακτικές στρατηγικές, άρχισε να μπαίνει σε εφαρμογή, όταν άρχισε να γίνεται αντιληπτό πως για να αποκτήσει το άτομο αληθινή μόρφωση είναι ανάγκη να δραστηριοποιηθεί αυτοβούλως. Είναι σημαντικό στην κατεύθυνση αυτή το να παρατηρήσει, να σκεφτεί, να συγκρίνει, να βγάλει συμπεράσματα, να δοκιμάσει λύσεις στα προβλήματά του και να δώσει απαντήσεις στα ερωτήματα και τις απορίες του. Να γίνει δηλαδή ενεργό μέλος της μαθησιακής πράξης και να συμμετέχει και το ίδιο στη μόρφωσή του. Έτσι θα αποφεύγονται προβληματικές στρατηγικές που οδηγούν σε άχρηστες και μη αφομοιωμένες γνώσεις, και θα ενθαρρύνεται η αφύπνιση εσωτερικών δυνάμεων και η ανάπτυξη ικανοτήτων.

Κατά την εφαρμογή της εξατομικευμένης διδασκαλίας πρέπει να λαμβάνονται υπόψη οι διαφορετικές ικανότητες, κλίσεις και χαρακτηριστικά που κατέχουν οι μαθητές, με τελικό στόχο να γίνονται ολοένα και περισσότερο ικανοί για ανεξάρτητη εργασία. Αυτό γιατί η διδασκαλία αυτή θα πρέπει να βρίσκεται σε συμφωνία με τη φύση του παιδιού και τη φυσική διαδικασία της μάθησης. Κατά την εφαρμογή των τεχνικών εξατομικευμένης διδασκαλίας, ο δάσκαλος δεν βλέπει μόνο την ύλη της διδασκαλίας αλλά συνυπολογίζει και τον παράγοντα μαθητή. Δεν βλέπει παράλληλα μόνο το σχολείο σαν ένα απομονωμένο μικρόκοσμο, αλλά και τη ζωή σαν όλον. Ορισμένες από τις βασικές αρχές της εξατομικευμένης διδασκαλίας είναι οι εξής:

- Ο μαθητής θα πρέπει να εργάζεται ατομικά.
- Ο μαθητής θα πρέπει να εργάζεται σύμφωνα με τους δικούς του ρυθμούς.
- Ο μαθητής θα πρέπει να προχωρά με μικρά και σταδιακά βήματα.
- Κάθε επιμέρους βήμα θα πρέπει να εντάσσεται σε μία διαδικασία προσεκτικά επιλεγμένη. Ο μαθητής θα πρέπει να συμμετέχει ενεργά και να λαμβάνει ανατροφοδότηση για κάθε επιμέρους βήμα που κάνει.
- Ο μαθητής θα πρέπει να ενημερώνεται το συντομότερο δυνατόν για την ορθότητα της κάθε απάντησής του.

3 Επισκόπηση Πεδίου

3.1 Μοντελοποίηση Χρήστη

Οι παράγοντες που επηρεάζουν τη μάθηση ενός ατόμου είναι πολλοί και ανάλογοι με τον ιδιαίτερο τύπο της ψυχοπνευματικής κατάστασης του κάθε ατόμου. Παρατηρείται ότι τα άτομα διαφέρουν μεταξύ τους ως προς τις νοητικές ικανότητες (π.χ. προσοχή, παρατηρητικότητα, δημιουργική φαντασία, κρίση, ταχύτητα μάθησης, περιέργεια), τις ειδικές ικανότητες (π.χ. γλωσσική ικανότητα, μαθηματική ικανότητα, κινητικές ικανότητες), αλλά ακόμα και ως προς τα ενδιαφέροντα, τις ανάγκες καθώς και ψυχολογικούς παράγοντες (αυτοέλεγχος, αυτοεκτίμηση, αυτοπροσδιορισμός, αυτοπραγμάτωση). Άμεση συνέπεια αυτού του γεγονότος αποτελεί η διαφοροποίηση των ατόμων ως προς τις επιδόσεις τους, το οποίο με τη σειρά του αποτελεί έναν επιπλέον παράγοντα ύπαρξης ατομικών διαφορών.

Στο πλαίσιο των ατομικών διαφορών έχουν αναπτυχθεί πολλά επιστημονικά μοντέλα με βασικότερα αυτά των μαθησιακών στυλ και της θεωρίας των μαθησιακών διαφορών. Τα μαθησιακά στυλ είναι ο τρόπος με τον οποίο ο κάθε εκπαιδευόμενος εφαρμόζει στρατηγικές και συμπεριφορές στη μαθησιακή διαδικασία. Πρακτικά, είναι σημαντικό το μοντέλο να μην περιγράφει μόνο το πώς οι μαθητές κατηγοριοποιούνται αλλά και πώς οι παρεχόμενες οδηγίες κατά τη μαθησιακή διαδικασία πρέπει να ενσωματωθούν σε κάθε μία κατηγορία μαθησιακού τύπου. Αυτό σημαίνει ότι εκτός των περιγραφικών πληροφοριών (π.χ. οι μαθητές κατηγοριοποιούνται σε 'ενεργητικούς' και 'παθητικούς'), το μοντέλο θα πρέπει να παρέχει κατευθυντήριες οδηγίες, οι οποίες μπορούν να οδηγήσουν σε ειδικούς κανόνες αναφορικά με το σχεδιασμό και την προσαρμογή του συστήματος (π.χ. τι τύπος εκπαιδευτικού υλικού θα πρέπει να συλλεχθεί για τους ενεργητικούς και τους παθητικούς μαθητές).

Το κόστος του μοντέλου κάθε μαθησιακού τύπου και των εργαλείων αξιολόγησης είναι ένας ακόμη παράγοντας ο οποίος θα πρέπει να απασχολήσει τους σχεδιαστές των συστημάτων. Η κατάσταση που αφορά τα διαθέσιμα διδακτικά εργαλεία αξιολόγησης, παρουσιάζει ποικιλομορφία. Μερικά παρέχονται δωρεάν και άλλα πρέπει να αγοραστούν. Στη τελευταία περίπτωση θα πρέπει οι σχεδιαστές να αναλογιστούν το κόστος του μοντέλου και το εργαλείο αξιολόγησης. Κατόπιν, θα πρέπει να σκεφτούν τη διαθεσιμότητα του τεστ σε σχέση με τον αριθμό των χρηστών.

3.1.1 Στρατηγικές εξατομίκευσης

Σε αυτό το σημείο παρατίθενται οι πέντε στρατηγικές εξατομίκευσης που συναντήσαμε στη βιβλιογραφία και τα βασικότερα σημεία της κάθε στρατηγικής. Αυτές οι στρατηγικές εμφανίζονται σε αρκετές διαδικτυακές εκπαιδευτικές εφαρμογές, είναι προτιμότερο όμως να χρησιμοποιούνται σε συνδυασμό για καλύτερα αποτελέσματα.

- **Name-Recognized Personalization:** Αυτή η στρατηγική βασίζεται στην αναγνώριση του κάθε ατόμου από το όνομά του, και απαντάται σε μεγάλο βαθμό σε διαδικτυακές εφαρμογές, όπου το όνομα του εκπαιδευόμενου εμφανίζεται στις οδηγίες ή οι προηγούμενες ενέργειες και οι επιτυχίες του αποθηκεύονται και παρουσιάζονται όποτε αυτό είναι χρήσιμο. Αποτελεί μια εύκολη και συνήθη στρατηγική για εξατομίκευση.
- **Self-Described Personalization:** Η στρατηγική αυτή βασίζεται στην παροχή πληροφοριών από τους ίδιους τους εκπαιδευόμενους, μέσα από κάποιες στρατηγικές συλλογής δεδομένων όπως τα ερωτηματολόγια σε ένα pre-course quiz, για τις προσωπικές τους προτιμήσεις και ιδιότητες. Η παρουσίαση του μαθησιακού υλικού αποτελεί μια εξατομικευμένη παροχή μαθησιακού υλικού για κάθε εκπαιδευόμενο και γίνεται βάση των απαντήσεων που έχουν δοθεί.
- **Segmented Personalization:** Αυτή η στρατηγική βασίζεται στο διαχωρισμό των εκπαιδευόμενων σε τμήματα με βάση κοινά χαρακτηριστικά που αυτοί παρουσιάζουν, όπως για παράδειγμα ίδια βαθμίδα εκπαίδευσης, ίδιο εργασιακό αντικείμενο. Το μαθησιακό υλικό βασίζεται στις ανάγκες του κάθε group που έχει οριστεί και ο κάθε εκπαιδευόμενος λαμβάνει το ανάλογο περιεχόμενο. Ο διαχωρισμός σε επίπεδα δυσκολίας περιλαμβάνεται στην τεχνική αυτή, η οποία είναι και η τεχνική εξατομίκευσης

που επιλέξαμε να εφαρμόσουμε στην εργασία μας, θεωρώντας την περισσότερο λειτουργική και υλοποιήσιμη.

- **Cognitive-Based Personalization:** Η στρατηγική αυτή βασίζεται στην απόδοση του ανάλογου μαθησιακού υλικού που αντιστοιχεί σε κάθε στυλ μάθησης των εκπαιδευόμενων. Για παράδειγμα κάποιος που προτιμά να διαβάζει κείμενο από το να ακούει οδηγίες θα έχει τη δυνατότητα να επιλέξει κείμενο αντί φωνής. Τα στυλ μάθησης των εκπαιδευόμενων ανακαλύπτονται μέσα από διαδικασίες όπως συλλογή δεδομένων, παρακολούθηση των ενεργειών του κάθε εκπαιδευόμενου, σύγκριση ενεργειών διαφορετικών χρηστών και πρόβλεψη των προθέσεων του χρήστη. Η συγκεκριμένη εφαρμογή αποτελεί μια περισσότερο σύνθετη επιλογή και απαντάται κυρίως σε υψηλά επίπεδα διαδικτυακών εφαρμογών.
- **Whole-Person Personalization:** Η στρατηγική αυτή είναι η πιο σύνθετη από όλες τις προηγούμενες και βασίζεται στους μαθησιακούς προσανατολισμούς (learning orientations). Στόχος της συγκεκριμένης στρατηγικής είναι να όχι απλά να βοηθήσει τους εκπαιδευόμενους να επιτύχουν μαθησιακούς στόχους, αλλά και να βελτιώσει τη γενικότερη μαθησιακή τους ικανότητα και τις διαδικτυακές μαθησιακές σχέσεις. Το εξατομικευμένο αυτό σύστημα συλλέγει στοιχεία, παρακολουθεί την πρόοδο και βελτιώνει την απόδοσή του με το πέρασμα του χρόνου.

Βασικός στόχος λοιπόν των τελευταίων επιστημονικών ερευνών είναι να ανακαλύψουν εκείνες τις διεργασίες που είναι υπεύθυνες για τον τρόπο που σκεφτόμαστε και μαθαίνουμε με απώτερο στόχο να γίνουν κατανοητές οι ατομικές μαθησιακές διαφορές και να επιτυγχάνεται καλύτερο επίπεδο εξατομίκευσης. Η κατανόηση αυτών των ατομικών διαφορών όμως, θα πρέπει να βασίζεται όχι μόνο στις γνωστικές διεργασίες που λαμβάνουν χώρα κατά τη μάθηση και τη δόμηση της γνώσης, αλλά και στις ατομικές ανάγκες των εκπαιδευόμενων, γεγονός που μας οδηγεί σε μια whole- person προσέγγιση της μάθησης. Η παραδοσιακή προσέγγιση της εξατομίκευσης δεν καλύπτει τις ανάγκες των εκπαιδευόμενων σε περιβάλλοντα ηλεκτρονικής μάθησης, όπου οι εκπαιδευόμενοι θέλουν ή σκοπεύουν να γίνουν περισσότερο αυτό-υποστηριζόμενοι και αυτό-καθοδηγούμενοι, ανεξάρτητοι από τον εκπαιδευτή.

Σημαντικό είναι ότι οι περισσότεροι εκπαιδευόμενοι που προέρχονται από τα περιβάλλοντα της τάξης αδυνατούν να χειριστούν διαδικτυακά εκπαιδευτικά περιβάλλοντα. Αυτό το γεγονός είναι ιδιαίτερα ανησυχητικό για το μέλλον της ηλεκτρονικής μάθησης, αλλά παράλληλα, η μελέτη των συναισθημάτων και των προθέσεων των εκπαιδευόμενων που επηρεάζουν την εκπαιδευτική διαδικασία αποτελεί μια εξαιρετική ερευνητική πρόκληση με στόχο μια αυτό-καθοδηγούμενη ηλεκτρονική μάθηση.

3.1.2 Μαθησιακός προσανατολισμός

Με τον όρο μαθησιακός προσανατολισμός ή μαθησιακό προφίλ εννοούμε τον μοναδικό, προσωπικό τρόπο με τον οποίο ο κάθε εκπαιδευόμενος κατανοεί, αξιολογεί και διαχειρίζεται τη μάθησή του για να επιτύχει στόχους. Οι μαθησιακοί προσανατολισμοί διαφέρουν από τα στυλ μάθησης. Ενώ τα στυλ μάθησης αναγνωρίζουν το σημαντικό ρόλο των γνωστικών παραγόντων, οι μαθησιακοί προσανατολισμοί αναγνωρίζουν το ρόλο των συναισθημάτων και των προθέσεων. Παράλληλα, αναγνωρίζουν ότι «όλα τα άτομα έχουν διαφορετικές μαθησιακές εμπειρίες και ωριμάζουν σα μαθητές, αφού γίνονται σταδιακά περισσότερο βέβαιοι, σκέφτονται περισσότερο σύνθετα, και εξελίσσονται στην κατανόηση και γίνονται εν γένει καλύτεροι στη διαχείριση αυξανόμενων σύνθετων προσωπικών συναισθηματικών, βουλητικών, κοινωνικών και γνωστικών μαθησιακών παραγόντων.

Το μοντέλο μαθησιακού προσανατολισμού παρέχει τα προφίλ των μαθησιακών διαφορών που απαντώνται στους εκπαιδευόμενους, με στόχο την κατηγοριοποίηση των τελευταίων σε ομάδες βάσει των χαρακτηριστικών τους και την παροχή των κατάλληλων οδηγιών μάθησης για κάθε κατηγορία ξεχωριστά. Αυτά τα προφίλ των εκπαιδευόμενων αναγνωρίζουν τις διαφορές στη μαθησιακή προσέγγιση και διαχωρίζουν τους εκπαιδευόμενους ανάλογα με τις πεποιθήσεις, τις επιθυμίες, τα συναισθήματα, την πρόθεση για προσπάθεια και τα γνωστικά και κοινωνικά στυλ στη μάθηση. Περιγράφουν την προδιάθεση του κάθε ατόμου στο να αναλάβει το έλεγχο, να ορίσει στόχους, να επιτύχει τους στόχους αυτούς, να διαχειριστεί το μαθησιακό υλικό, αλλά και να λύσει προβλήματα στη διαδικασία της μάθησης και είναι οι ακόλουθοι:

- **Transforming Learner:** Πρόκειται για έναν ισχυρά αυτό-παρακινούμενο εκπαιδευόμενο που χρησιμοποιεί ανακαλυπτική μάθηση και αυτό- διαχειρίζεται τους στόχους, τη μάθηση, την πρόοδο και τα αποτελέσματα. Βρίσκεται σε σύγχυση όταν περιορίζεται ή του δίνεται μικρή μαθησιακή αυτονομία.
- **Performing Learner:** Γίνεται ένας αυτό-παρακινούμενος, συγκεντρωμένος εκπαιδευόμενος όταν το αντικείμενο είναι ελκυστικό. Θέτει και πετυχαίνει βραχυπρόθεσμους task-oriented στόχους που αποτελούν μέτρια-προς- υψηλά standards. Προτιμά την καθοδήγηση και την αλληλεπίδραση για να επιτύχει στόχους.
- **Conforming Learner:** Πρόκειται για έναν εκπαιδευόμενο που λαμβάνει μικρά ρίσκα, κινητοποιείται εξωτερικά και ακολουθεί και προσπαθεί να επιτύχει απλούς task-oriented στόχους που έχουν οριστεί από άλλους. Χρειάζεται συνεχή καθοδήγηση και περιμένει ενίσχυση όταν επιτυγχάνει βραχυπρόθεσμους στόχους. Μεγιστοποιεί την προσπάθεια σε υποστηρικτικά περιβάλλοντα
- **Resistant Learner:** Αποφεύγει να χρησιμοποιήσει τη γνώση για να επιτύχει ακαδημαϊκούς στόχους που ορίζονται από άλλους. Αποφεύγει συστηματικά τη μάθηση (είναι απαθής, συγχυσμένος, αποθαρρυσμένος ή ανυπάκουος). Θέτει προσωπικούς στόχους που δεν αποτελούν επίσημες μαθησιακές απαιτήσεις ή προσδοκίες.

Οι Transforming Learners προτιμούν λοιπόν ελαστικά, ανακαλυπτικά περιβάλλοντα με χαλαρή δομή που προωθούν ελκυστικούς στόχους, στρατηγικές, τη λύση προβλημάτων και την αυτό-διαχειριζόμενη μάθηση. Οι Performing Learners προτιμούν ημι-δομημένα, αλληλεπιδραστικά περιβάλλοντα που ενθαρρύνουν την ομαδικότητα και παρέχουν λεπτομέρειες, στόχους και διαδικασίες. Προτιμούν ανταγωνιστικά περιβάλλοντα που προσφέρουν καθοδήγηση, εξάσκηση και ανάδραση ώστε να ενισχύσουν την αυτό-κινητοποίηση, τη λύση προβλημάτων, τον αυτό-έλεγχο, ενώ παράλληλα ελαχιστοποιούν την εξερεύνηση, την περίσσεια προσπάθεια και τα δύσκολα standards. Οι Conforming Learners προτιμούν απλά, ασφαλή, δομημένα περιβάλλοντα που βοηθούν τους χρήστες να πετύχουν εύκολους, χαμηλού ρίσκου στόχους. Οι Resistant Learners τέλος, λόγω των περιορισμένων κινήτρων τους, δραστηριοποιούνται καλύτερα σε περιβάλλοντα που στοχεύουν στην ανακίνηση του ενδιαφέροντος και την εντυπωσιακή παρουσίαση της ύλης. Προτείνεται λοιπόν η παροχή καθοδήγησης και συνεργατικών εκπαιδευτικών εργαλείων, που θα λαμβάνουν υπόψη τη μαθησιακή φύση του εκπαιδευόμενου και θα τον βοηθούν να μάθει εύκολα και αποτελεσματικότερα.

3.2 Προσαρμοστικά Εκπαιδευτικά Περιβάλλοντα

Τα προσαρμοστικά εκπαιδευτικά περιβάλλοντα ενσωματώνουν λειτουργίες που διαφοροποιούνται ανάλογα με την αλληλεπίδραση με το χρήστη. Η λειτουργικότητά τους είναι λοιπόν δυναμική και παρέχουν στοιχεία προσαρμογής στη μαθησιακή συμπεριφορά του. Μπορεί να περιλαμβάνουν εφαρμογές που λειτουργούν τοπικά ή εφαρμογές που βασίζονται στον παγκόσμιο ιστό.

Τα Προσαρμοστικά Εκπαιδευτικά Περιβάλλοντα που βασίζονται στον Παγκόσμιο Ιστό (Web-based Adaptive Educational Systems) αποτελούν εξέλιξη δύο προγενέστερων τύπων Εκπαιδευτικών Περιβαλλόντων, των Νοημόνων Εκπαιδευτικών Συστημάτων (Intelligent Tutoring Systems) και των Προσαρμοστικών Συστημάτων Υπερμέσων (Adaptive Hyper-media Systems). Αποτελούν επομένως συνδυασμό δύο διαφορετικών προσεγγίσεων στην περιοχή της υποστηριζόμενης μάθησης από υπολογιστή (computer assisted learning) της κατευθυνόμενης δασκαλο-κεντρικής προσέγγισης των παραδοσιακών συστημάτων Τεχνητής Νοημοσύνης και της ευέλικτης μαθητο-κεντρικής προσέγγισης της ελεύθερης πλοήγησης ενός συστήματος υπερμέσων.

Τα Προσαρμοστικά Εκπαιδευτικά Περιβάλλοντα βασίζονται πάνω στην τεχνική που χαρακτηρίζει τη λειτουργικότητα και των Προσαρμοστικών Εκπαιδευτικών Υπερμέσων (Adaptive Educational Hypermedia -ΑΕΗ). Τα ΑΕΗ είναι μια σχετικά νέα κατεύθυνση της έρευνας στον τομέα των προσαρμοστικών και βασισμένων στο χρήστη εκπαιδευτικών εφαρμογών. Τα Προσαρμοστικά Εκπαιδευτικά Υπερμέσα συγκεντρώνουν χαρακτηριστικά των Ευφυών Συστημάτων Διδασκαλίας (Intelligent Tutoring Systems) και των Προσαρμοστικών Υπερμέσων (Adaptive Hypermedia). Στα Προσαρμοστικά Εκπαιδευτικά Υπερμέσα ο χρήστης έχει αρκετή ελευθερία επιλογής στην πλοήγηση, σε αντίθεση με τα Ευφυή Συστήματα

Διδασκαλίας στα οποία το σύστημα ελέγχει σε μεγάλο βαθμό τι παρουσιάζεται στο χρήστη. Τα Προσαρμοστικά Εκπαιδευτικά Υπερμέσα προσπαθούν να προσαρμόσουν το περιεχόμενο και τις συνδέσεις μιας σελίδας υπερκειμένου (hypertext) στις απαιτήσεις του χρήστη. Έτσι οι δύο βασικές υπηρεσίες που προσφέρουν στον χρήστη είναι η προσαρμοστική παρουσίαση (adaptive presentation) και η προσαρμοστική πλοήγηση (adaptive navigation).

3.2.1 Παραδείγματα εφαρμογών

Στη συνέχεια παρατίθενται κάποιες σημαντικές εφαρμογές στον τομέα των Προσαρμοστικών Εκπαιδευτικών Προγραμμάτων, με στόχο να ερευνησουμε την υπάρχουσα κατάσταση στις προσαρμοστικές εκπαιδευτικές εφαρμογές, έτσι ώστε στη συνέχεια να αναζητήσουμε τρόπους γεφύρωσης του χάσματος ανάμεσα στην υπάρχουσα και στην επιθυμητή κατάσταση του σχεδιασμού εξατομικευμένων προσαρμοστικών εφαρμογών που θα ενσωματώνουν όλες εκείνες τις εκπαιδευτικές παραμέτρους που συνθέτουν τη μαθησιακή διαδικασία.

Σημαντικό θέμα στο σχεδιασμό ενός Προσαρμοστικού Περιβάλλοντος Μάθησης αποτελεί το είδος της προσαρμοστικότητας που θα εφαρμόσει, και πιο συγκεκριμένα τα χαρακτηριστικά του συστήματος τα οποία διαφοροποιούνται ανάλογα με τον εκπαιδευόμενο. Έχουν αναπτυχθεί διάφορες τεχνολογίες που υποστηρίζουν την εξατομικευμένη μάθηση, διαμορφώνοντας το ίδιο το περιεχόμενο της αλληλεπίδρασης εκπαιδευόμενου και συστήματος ή που απλά υποστηρίζουν τον εκπαιδευόμενο κατά τη διάρκεια της μελέτης του λειτουργώντας συμβουλευτικά.

Στη συνέχεια παρατίθενται κάποιες από τις σημαντικότερες εφαρμογές (εργαλεία και standards) με σκοπό να ερευνησουμε την υπάρχουσα κατάσταση στον τεχνολογικό τομέα των προσαρμοστικών εκπαιδευτικών εφαρμογών. Στόχος είναι να ερευνησουμε την προοπτική χρήσης κάποιων παρόμοιων ή ακόμη και αυτών των εργαλείων για το σχεδιασμό πραγματικά εξατομικευμένων προσαρμοστικών περιβαλλόντων που θα συνεργάζονται με τον τομέα της Διαχείρισης εκπαιδευτικού Υλικού (Educational Management).

Το iclass και το SCORM αποτελούν πολύ σημαντικές και πλήρεις υλοποιήσεις και για το λόγο αυτό αναλύονται περισσότερο από το ACCT και το APeLS - που αποτελούν μοντέλα κατασκευής μαθημάτων - καθώς και το MOT και Knowledge Tree, τα οποία αποτελούν Web-based περιβάλλοντα μάθησης.

Στόχος του iclass είναι η ανάπτυξη ενός έξυπνου και ανοιχτού εκπαιδευτικού συστήματος και περιβάλλοντος βασισμένου στη γνωστική θεώρηση και προσαρμοσμένου στις ξεχωριστές ανάγκες του κάθε εκπαιδευόμενου. Το iclass ξεκίνησε τον Ιανουάριο του 2004 και υποστηρίζεται από 11 χώρες οι οποίες φιλοξενούν ερευνητικά προγράμματα που έχουν τους εξής στόχους:

- Την εκτενή έρευνα για τον προσδιορισμό και σχεδιασμό της επόμενης γενιάς εκπαιδευτικών προϊόντων και υπηρεσιών.
- Την κατασκευή και δοκιμή πρωτοτύπων του iclass.
- Την αποτίμηση των αποτελεσμάτων της δραστηρικής εφαρμογής του εργαλείου μέχρι και τη δευτεροβάθμια εκπαίδευση.

Οι προσδοκίες του iclass εμπεριέχουν την κατασκευή ενός εξελιγμένου εκπαιδευτικού συστήματος, στηριγμένου σε μία αρχιτεκτονική αλληλουχίας της γνώσης και προσαρμοσμένου στο επίπεδο αντίληψης και τρόπου μάθησης του εκάστοτε χρήστη, δημιουργώντας δυναμικά, εξατομικευμένα εκπαιδευτικά αντικείμενα. Παράλληλα, στοχεύει στην κατασκευή ενός διαδεδωμένου, συνεργατικού περιβάλλοντος που θα προσφέρει συνεχή πρόσβαση σε όλους τους ενδιαφερόμενους (γονείς, δασκάλους, μαθητές, πολιτεία, εκδότες) ώστε να εμπλουτίζεται το περιεχόμενο και οι υπηρεσίες που θα παρέχει, προωθώντας την άμεση επικοινωνία.

Ορισμένα από τα βασικά οφέλη που προσφέρει το iclass είναι συγκεντρωτικά τα εξής:

- Διαμόρφωση νέας παιδαγωγικής προσέγγισης ικανής να προσαρμόζει τις εκπαιδευτικές και μαθησιακές διαδικασίες στις ανάγκες του κάθε εκπαιδευόμενου.
- Εξατομίκευση σύμφωνη με τις διαφορετικές κουλτούρες και τρόπους μάθησης.
- Ένα κατευθυντήριο διαγνωστικό σύστημα ικανό όχι μόνο να παρουσιάζει εξατομικευμένο περιεχόμενο στους χρήστες του, αλλά να εντοπίζει τα προβλήματα που παρουσιάστηκαν κατά την εκπαιδευτική διαδικασία και να τα αναφέρει στους εκάστοτε συσχετιζόμενους με το σύστημα, για την περαιτέρω βελτίωση και ανάπτυξη του.

Όλοι οι ενδιαφερόμενοι για την εκπαίδευση θα επωφεληθούν από τα αποτελέσματα αυτής της προσπάθειας. Τέτοιοι είναι οι μαθητές, οι δάσκαλοι, οι διευθυντές σχολείων, οι εκπαιδευτικές αρχές, οι γονείς καθώς και άλλοι παράγοντες όπως οι υπεύθυνοι ανάπτυξης περιεχομένου, οι εκδότες και όσοι εμπορεύονται και προωθούν εκπαιδευτικές υπηρεσίες. Στο πρόγραμμα συμμετέχουν, όπως αναφέρθηκε, 11 χώρες καθώς και κορυφαίοι Ευρωπαίοι ερευνητές της γνωστικής επιστήμης, παιδαγωγοί και επιστήμονες της τεχνητής νοημοσύνης. Το iclass υποστηρίζεται από αρκετές εταιρίες λογισμικού καθώς και από τρία δίκτυα διεθνών σχολείων στα οποία θα δοκιμαστεί το πρόγραμμα και θα αποτιμηθούν οι δυνατότητες του. Το σύστημα του iclass μπορεί συνεπώς να αλληλεπιδράσει με όλους του συμμετέχοντες της εκπαιδευτικής κοινότητας. Οι πληροφορίες που θα συλλέγονται από εξωγενείς δραστηριότητες, όπως επίσκεψη μουσείων, παρακολούθηση τηλεόρασης, διαδίκτυο, καθώς και δραστηριότητες εντός του σχολικού περιβάλλοντος θα επηρεάζουν το σύστημα. Απ' την άλλη πλευρά, το iclass θα προτείνει κάποιες δραστηριότητες όπως ανάγνωση βιβλίων, εκπαιδευτικές εκδρομές, παρακολούθηση κάποιων προγραμμάτων στην τηλεόραση κλπ. με σκοπό την συμπλήρωση της εκπαιδευτικής διαδικασίας.

Το πρότυπο SCORM (Sharable Content Object Reference Model) ουσιαστικά αποτελεί μία συλλογή τεχνικών προτύπων, τα οποία έχουν αναπτυχθεί από διάφορους σχετικούς οργανισμούς, όπως είναι οι ARIADNE, AICC, IMS και IEEE με στόχο να δώσουν τη δυνατότητα στα διάφορα συστήματα ηλεκτρονικής μάθησης να εισάγουν, να διαμοιράζονται, να επαναχρησιμοποιούν και να εξάγουν ηλεκτρονικό εκπαιδευτικό υλικό, βάσει συγκεκριμένων προτύπων. Το SCORM έχει βασισθεί σε προϋπάρχουσες προδιαγραφές και πρότυπα των παραπάνω οργανισμών, τα οποία επεκτείνει και εξειδικεύει, ορίζοντας επιπλέον λεπτομέρειες και οδηγίες υλοποίησης της εφαρμογής τους. Η δημιουργία του βασίστηκε σε έξι βασικές απαιτήσεις οι οποίες κρίθηκαν ως στρατηγικής σημασίας για την περαιτέρω επέκταση, τη βελτίωση της αποτελεσματικότητας και την μείωση του κόστους της ηλεκτρονικής μάθησης.

- Προσβασιμότητα (Accessibility): Δυνατότητα εντοπισμού και πρόσβασης μονάδων εκπαιδευτικού υλικού που βρίσκονται σε μία απομακρυσμένη τοποθεσία, και εκπαιδευτικής χρήσης τους από οπουδήποτε και μέσω οποιουδήποτε συστήματος ηλεκτρονικής μάθησης
- Προσαρμοσιμότητα (Adaptability): Δυνατότητα προσαρμογής του εκπαιδευτικού υλικού και όλης της μαθησιακής εμπειρίας (learning experience) στις ειδικές ανάγκες των οργανισμών και των εκπαιδευόμενων που το χρησιμοποιούν.
- Οικονομικότητα (Affordability): Αύξηση της εκπαιδευτικής απόδοσης και παραγωγικότητας μέσω μείωσης του απαιτούμενου για την εκπαίδευση χρόνου και κόστους.
- Ανθεκτικότητα (Durability): Ανθεκτικότητα στις τεχνολογικές εξελίξεις, ώστε το προϋπάρχον εκπαιδευτικό υλικό να μπορεί να χρησιμοποιείται χωρίς την ανάγκη εκτεταμένων και δαπανηρών μετατροπών.
- Διαλειτουργικότητα (Interoperability): Δυνατότητα να πάρουμε μονάδες εκπαιδευτικού υλικού που αναπτύχθηκαν κάπου με ένα εργαλείο ή πλατφόρμα, και να τις χρησιμοποιήσουμε κάπου αλλού με ένα διαφορετικό εργαλείο ή πλατφόρμα.
- Επαναχρησιμοποιησιμότητα (Reusability): Ευελιξία ενσωμάτωσης και χρήσης μονάδων εκπαιδευτικού υλικού σε πολλά διαφορετικά εκπαιδευτικά πακέτα για διαφορετικούς σκοπούς και διαφορετικές ομάδες – στόχους (target groups)

Οι ιστοσελίδες εκπαιδευτικού υλικού που προτυποποιείται από το SCORM είναι «έξυπνες» και περιέχουν κώδικα με τον οποίο επικοινωνούν με το σύστημα ηλεκτρονικής μάθησης όπου εκτελούνται. Το σύστημα ηλεκτρονικής μάθησης είναι επίσης «έξυπνο» επικοινωνεί με το «έξυπνο» εκπαιδευτικό υλικό και γνωρίζει τι υλικό και πότε κάποιος θα πρέπει να παρουσιαστεί σε κάθε εκπαιδευόμενο.

Το ACCT αποτελεί ένα παιδαγωγικού προσανατολισμού σύστημα κατασκευής μαθημάτων. Ο τρόπος λειτουργίας του προσφέρει δυνατότητες για την εφαρμογή κάθε δυνατού εκπαιδευτικού σεναρίου. Η διαδικασία κατασκευής ενός μαθήματος περιλαμβάνει την κατασκευή XML αρχείων τα οποία ορίζουν την δομή του καθώς και τις λειτουργίες του συστήματος, εξαρτώμενες από τις επιλογές του χρήστη. Το μοντέλο ApeLS (Adaptive Personalized eLearning Service) είναι παρόμοιο με του ACCT, αλλά προσφέρει και αυξημένες δυνατότητες εισαγωγής και επεξεργασίας μεταδεδομένων καθώς και δομημένη χρήση αφηγηματικών μοντέλων για την υλοποίηση ενός εκπαιδευτικού σεναρίου. Σημαντικές είναι οι ομάδες

περιεχομένων οι οποίες παρέχουν πακέτα δεδομένων που αφορούν το ίδιο αντικείμενο αλλά με άλλη προσέγγιση (π.χ. ένα πακέτο για θεωρία ηλεκτρισμού για επίπεδο δημοτικού και ένα για επίπεδο λυκείου). Το APeLS αλληλεπιδρά με το μαθησιακό στυλ του κάθε εκπαιδευομένου και αξιοποιεί την χρήση μεταδεδομένων για την καταγραφή και κατηγοριοποίηση του σαν χρήστη. Αποτέλεσμα είναι η κατασκευή εξατομικευμένων μαθημάτων με την χρήση ενός απλού σχετικό μοντέλου.

Το MOT αποτελεί ένα Web-Based σύστημα κατασκευής μαθημάτων. Ο χρήστης μπορεί να κάνει login στο σύστημα σαν καθηγητής και σαν μαθητής. Σαν καθηγητής μπορεί να επιλέξει εκπαιδευτικό υλικό, πόρους, εργαλεία κλπ και να κατασκευάσει ένα ολοκληρωμένο online μάθημα. Σαν μαθητής μπορεί να δει μία λίστα από τα διαθέσιμα μαθήματα, να επιλέξει όποια επιθυμεί ή όποια του έχουν ζητηθεί και να τα παρακολουθήσει. Το MOT στηρίζεται στις δυνατότητες του Semantic Web και προσφέρει σημαντικές λειτουργίες στους τομείς της προσαρμοστικότητας του περιβάλλοντος στο μαθησιακό στυλ του εκάστοτε εκπαιδευομένου καθώς και στην δυναμική εξέλιξη των μαθημάτων. Η web-based λογική του MOT εξυπηρετεί την ευκολία στην πρόσβαση και την αμεσότητα στην χρήση, τομείς που είναι πολύ σημαντικοί για την ηλεκτρονική μάθηση.

Το Knowledge Tree τέλος, αποτελεί ένα online σύστημα διαχείρισης εγγράφων. Βασικός του στόχος είναι να εξυπηρετεί αυξημένες ανάγκες για την διαχείριση, αποθήκευση και επεξεργασία εγγράφων. Ο προσανατολισμός του συγκεκριμένου Web – Based εργαλείου είναι η οργάνωση των μικρών ή μεσαίων επιχειρήσεων και προς το παρόν δεν υφίσταται κάποια εφαρμογή του στην ηλεκτρονική μάθηση. Παρόλα αυτά πολλές είναι οι δυνατότητες του και στην εκπαίδευση, καθώς μπορεί να αποτελέσει ένα ανεπτυγμένο συνεργατικό εργαλείο απομακρυσμένης διαχείρισης εργασιών. Οι δυνατότητες του Knowledge Tree περιγράφονται συνοπτικά παρακάτω:

- Ασφάλεια και σταθερότητα συστήματος
- Πρόσβαση μέσω Browser, Windows σύστημα, Microsoft Office, και third- party εφαρμογές.
- Απεριόριστες δυνατότητες συνεργατικότητας στην κατασκευή του περιεχομένου
- Ισχυρός μηχανισμός προσθήκης μεταδεδομένων στα έγγραφα
- Σκανάρισμα εγγράφων “Local to Web” (αποθήκευση κατ’ευθείαν στη βάση δεδομένων του Knowledge Tree)
- Πλήρες Προσαρμοστικό Περιβάλλον για κάθε χρήστη

3.2.2 Συμπεράσματα

Η ανάπτυξη διεθνών προτύπων στο χώρο της ηλεκτρονικής μάθησης με στόχο την εξασφάλιση διαλειτουργικότητας και μέγιστης συμβατότητας μεταξύ των διαφορετικών συστημάτων ηλεκτρονικής μάθησης καθώς και την δυνατότητα διαμοιρασμού, επαναχρησιμοποίησης και ολοκλήρωσης μονάδων εκπαιδευτικού υλικού είναι κρίσιμης σημασίας για την ανάπτυξη της ηλεκτρονικής μάθησης και αποτελεί τη βάση για τη δημιουργία ευέλικτων και αποδοτικών λύσεων.

Παρά το γεγονός ότι σήμερα υπάρχουν πολλά πρότυπα, προδιαγραφές και οδηγίες στο χώρο της ηλεκτρονικής μάθησης, οι ενδιαφερόμενοι οργανισμοί που προσανατολίζονται στην αγορά ή στην ανάπτυξη συστημάτων ηλεκτρονικής μάθησης και πακέτων ηλεκτρονικού εκπαιδευτικού υλικού, θα πρέπει να εστιάσουν την προσοχή τους στα πρότυπα επικοινωνίας, μεταδεδομένων και πακετοποίησης. Οι υπεύθυνοι έργων ανάπτυξης τέτοιων συστημάτων και όλοι οι εμπλεκόμενοι στα έργα αυτά θα ωφεληθούν σημαντικά, εάν κατανοήσουν το πώς όλες αυτές οι προσφερόμενες δυνατότητες από τις τεχνολογίες ηλεκτρονικής μάθησης και τα σχετικά πρότυπα μπορούν να ικανοποιήσουν τις ιδιαίτερες ανάγκες τους. Τούτο σημαίνει ότι οι προδιαγραφές και τα πρότυπα πρέπει να αποτελέσουν τα θεμέλια πάνω στα οποία το έργο της ηλεκτρονικής μάθησης πρέπει να στηριχθεί, παρέχοντας μία συνεκτική, αποτελεσματική και δομημένη μέθοδο, στην οποία οι φάσεις σχεδιασμού, ανάπτυξης, παράδοσης και συντήρησης του έργου μπορούν να βασισθούν.

Επιπλέον, η διευρυμένη πλέον βάση υποστήριξης των προτύπων αυτών από πολλούς κατασκευαστές και προμηθευτές συστημάτων ηλεκτρονικής μάθησης και ηλεκτρονικού εκπαιδευτικού υλικού δίνει μία προστιθέμενη αξία στο όλο τελικό προϊόν. Το γεγονός αυτό είναι ιδιαίτερα επωφελές τόσο για την πλευρά της ζήτησης όσο και για την πλευρά της προσφοράς

της ηλεκτρονικής μάθησης. Το ζητούμενο είναι η ολοκληρωμένη εξατομίκευση της μάθησης με εύκολο, άμεσο τρόπο και υψηλές δυνατότητες επαναχρησιμότητας. Για την υλοποίηση αυτή υπάρχει ανάγκη για «σύνδεση» της τεχνολογίας με το κατάλληλο θεωρητικό υπόβαθρο των παραγόντων που επηρεάζουν την εκπαίδευση. Η ενοποίηση αυτή θα αποτελέσει το μίγμα για την κατασκευή χώρων εκπαίδευσης με ισχυρές δυνατότητες αναγνώρισης των αναγκών του χρήστη και άμεση προσαρμοστικότητα στο προσωπικό του μαθησιακό στυλ.

4 Περιγραφή ανάγκης για την εφαρμογή Tic-Tac

Στα πλαίσια της εργασίας αυτής αναπτύχθηκε μία εφαρμογή εκπαιδευτικού λογισμικού, που απευθύνεται σε μικρούς μαθητές και έχει σκοπό να τους διδάξει την ανάγνωση της ώρας από ένα αναλογικό ρολόι. Παρουσιάζεται μία σειρά δυναμικά δημιουργούμενα τεστ, τα οποία ο μαθητής συμπληρώνει και παίρνει ανατροφοδότηση για τις απαντήσεις του, βασισμένη στην πρόδοό του. Το λογισμικό σχεδιάστηκε με επίκεντρο τις μαθησιακές δυσκολίες που αντιμετωπίζουν οι μαθητές κατά τη διαδικασία της ανάγνωσης ενός αναλογικού ρολογιού και στοχεύει στη δημιουργία προφίλ για αυτούς και παροχή συμβουλών προσαρμοσμένων στις αδυναμίες τους, επιθυμώντας να διασαφηνίσει ορισμένα συχνά λάθη που πραγματοποιούν στην ανάγνωση της ώρας.

4.1 Διδακτική της ανάγνωσης της ώρας

Σύμφωνα με τη βιβλιογραφία στον τομέα αυτό, η διαδικασία μάθησης της ώρας πολλές φορές προκαλεί δυσκολίες και απογοήτευση στα παιδιά και δημιουργεί στους καθηγητές ερωτήματα σχετικά με τους λόγους από τους οποίους αυτές οι δυσκολίες πηγάζουν. Ένα από τα προβλήματα που συνδέονται με την απόκτηση της ικανότητας μέτρησης για το χρόνο, είναι το ότι, λόγω της αφηρημένης φύσης του, δεν είναι εύκολα αισθητός σε μονάδα μέτρησης, όπως το μήκος ή το βάρος. Έτσι αποτελεί ένα από τα λιγότερο μελετηθέντα πεδία στο χώρο της παιδαγωγικής.

Μιας και η έννοια του χρόνου περιλαμβάνει πολλές επιμέρους έννοιες που αλληλοσυνδέονται, όπως το χρονικό διάστημα και τα χρονικά σημεία, ενώ αντίθετα με άλλα μεγέθη είναι δυναμικά μεταβαλλόμενο, εισάγεται η απαίτηση μιας διαφορετικής προσέγγισης στη διαδικασία της διδασκαλίας του. Καταρχάς, οι υποδιαιρέσεις του χρόνου διαφέρουν από τις δεκαδικές υποδιαιρέσεις του μήκους και του βάρους: Μία ώρα είναι εξήντα λεπτά, ένα λεπτό εξήντα δευτερόλεπτα και ούτω καθεξής. Επιπλέον, δεν υπάρχουν μηδενικά μεγέθη χρόνου πάνω σε ένα ρολόι, δεν υπάρχει δηλαδή μηδενική ώρα, αλλά μηδενικό χρονικό διάστημα, το οποίο και πάλι δεν αναπαρίσταται, παρά με στασιμότητα των δεικτών. Συνεπώς, το μετρικό σύστημα για το χρόνο διαφοροποιείται από τα άλλα, εισάγει διαφορετικές μαθησιακές δυσκολίες και συνήθως αντιμετωπίζεται με άλλες διδακτικές τεχνικές.

Μια κοινώς αποδεκτή προσέγγιση, είναι ο διαχωρισμός της έννοιας του χρόνου από την τεχνική του να «διαβάζουμε την ώρα». Έτσι μπορεί να αποφεύγεται ανεπιθύμητη σύγχυση στη διαδικασία και να διδάσκονται έννοιες απλούστερα κατανοήσιμες. Η διδασκαλία της ανάγνωσης της ώρας συνήθως διαχωρίζεται σε τρεις επιμέρους έννοιες:

- Το χρονικό σημείο
- Τα χρονικά διαστήματα
- Τη χρονική διάρκεια

Το χρονικό σημείο αφορά μια δεδομένη στιγμή στο χρόνο. Η αντίληψη της έννοιας αυτής είναι εκείνη που περισσότερο αφορά την ανάγνωση ενός ρολογιού και ενός ημερολόγιου. Όμως, ξεκινώντας από την ικανότητα ανάγνωσης του χρονικού σημείου, επέρχεται ευκολότερα και η αντίληψη μιας χρονικής διάρκειας. Λόγω της αναγωγής του προβλήματος ανάγνωσης ενός ψηφιακού ρολογιού στην ανάγνωση αριθμών από μια συσκευή, συνήθως η ανάγνωση της ώρας από ένα τέτοιο ρολόι επιτυγχάνεται με μεγαλύτερη ευκολία. Οι απόψεις όμως δίστανται στο αν είναι σωστή πρακτική το να διδάσκεται πρώτα η ανάγνωση της ώρας από ένα ψηφιακό μέσο, ως απλούστερη. Εάν ο σκοπός είναι απλά η δυνατότητα του παιδιού να γνωρίζει «τι ώρα είναι» για πρακτικούς λόγους, τότε η μέθοδος αυτή συστήνεται.

Εντούτοις, αν και με μια πρώτη προσέγγιση η ψηφιακή ώρα διαβάζεται πιο εύκολα, αρκετοί υποστηρίζουν ότι είναι περισσότερο σωστό να ξεκινούμε με τη διδασκαλία της αναλογικής ώρας. Ο λόγος είναι, ότι αν και δυσκολότερο να αποκωδικοποιηθεί, το αναλογικό ρολόι παρέχει μια περισσότερο άμεση εικόνα του χρόνου. Για παράδειγμα, όταν διαβάζουμε ότι είναι εννέα η ώρα, έχουμε το μικρό δείκτη στο εννέα και το μεγάλο στο δώδεκα. Όταν ο μαθητής κατανοήσει ότι δώδεκα για το λεπτοδείκτη είναι το σημείο μηδέν, έχει αποκτήσει πλήρη εικόνα του πώς δείχνουν οι ακέραιες ώρες στο ρολόι.

Κάτι τέτοιο δεν ισχύει και για το ψηφιακό ρολόι. Αν ξεπεράσουμε το γεγονός ότι οι αριθμοί συνήθως δεν αναπαρίστανται όπως συνηθίζεται σε τυπωμένα ή ηλεκτρονικά κείμενα, λόγω της τεχνολογίας υγρών κρυστάλλων, η ερμηνεία ενός ψηφιακού ρολογιού απαιτεί αρκετή προσπάθεια. Καταρχάς, ένας μικρός μαθητής δε γνωρίζει την έννοια του μηδέν πριν από έναν

αριθμό. Έτσι, η ώρα 09:00 ενδεχομένως να μην κατανοείται αμέσως ως «εννιά». Επιπλέον, το παιδί πρέπει να γνωρίζει τι αναπαριστά η άνω και κάτω τελεία στην απεικόνιση αυτή, ότι περνούμε δηλαδή από τις ώρες στα λεπτά. Τέλος, εισάγεται η έννοια του εσπερινού δωδεκάωρου, κάτι το οποίο ίσως είναι το δυσκολότερο για το μαθητή. Δηλαδή, όσον αφορά την ώρα 21:00, δεν είναι καθόλου προφανές αν δε διδαχθεί ότι αντιστοιχεί στην 09:00 μ.μ.

Για το λόγο λοιπόν πως η μάθηση της αναλογικής ώρας, αν και λίγο πιο επίπονη, παρέχει καλύτερη κατανόηση της έννοιας του χρόνου στο παιδί και καλύτερες βάσεις για την αντιμετώπιση θεμάτων όπως η χρονική διάρκεια και το χρονικό διάστημα, διαλέξαμε την υλοποίηση ενός αναλογικού ρολογιού στη διδασκαλία μας, αφήνοντας την επέκταση για ψηφιακό ρολόι σε μελλοντική εργασία.

4.2 Μαθησιακές δυσκολίες

Θα αναφέρουμε τώρα τα συνηθέστερα από τα προβλήματα που μπορεί να συναντήσουν οι μαθητές κατά τη διαδικασία μάθησης της ανάγνωσης ενός αναλογικού ρολογιού. Σχετικές έρευνες δείχνουν ότι δυσκολίες που μπορεί να αντιμετωπιστούν, σχετίζονται με τα εξής:

Μη παρατήρηση του ενός από τους δύο δείκτες.

Δυσκολία κατανόησης της σημασίας του ότι οι δείκτες περιστρέφονται.

Δυσκολία κατανόησης της σημασίας του να βρίσκεται ο δείκτης των ωρών λίγο πριν ή λίγο μετά μια συγκεκριμένη ώρα.

Δυσκολία αντίληψης των υποδιαίρεσεων του ρολογιού που υποδηλώνουν τα λεπτά.

Δυσκολία αντίληψης του γεγονότος ότι τα λεπτά δεν αναγράφονται σαν αριθμοί σε ένα αναλογικό ρολόι, ότι δηλαδή το «ένα» πάνω στο ρολόι σημαίνει «πέντε» για τα λεπτά και ούτω καθεξής.

Οι μαθητές λοιπόν χρειάζονται μια αρκετά εκτενή τράπεζα γνώσης προκειμένου να κατορθώσουν την ανάγνωση ενός αναλογικού ρολογιού. Αυτή η γνώση περιλαμβάνει και την αντίληψη του κυκλικού τρόπου μέτρησης για το χρόνο, που επεκτείνεται στη μέτρηση των ημερών της εβδομάδας, των μηνών και των ετών και ολοκληρώνεται με την ανάγνωση ημερολογίων και την κατανόηση των ζωνών ώρας.

Η μέτρηση χρονικών διαστημάτων επιπλέον, προϋποθέτει τη γνώση και χρησιμοποίηση των χρονικών μονάδων, έχοντας κατανοήσει την αφηρημένη και υποκειμενική έννοια του χρόνου. Το γεγονός ότι ο χρόνος δεν είναι κάτι απτό, εισάγει δυσκολίες στη μέτρησή του και τον υπολογισμό του. Η επαφή των μαθητών με επαναλαμβανόμενες έννοιες και δραστηριότητες στην καθημερινή τους ζωή, μπορεί να παράσχει μια αρχική εξοικείωση με την περιοδικότητα του χρόνου και να δώσει σε συγκεκριμένες δραστηριότητες μια χρονική ετικέτα, πχ. «ξυπνάμε στις οχτώ το πρωί και τρώμε στις δώδεκα το μεσημέρι». Όμως για να γίνουν πλήρως κατανοητές αυτές οι φαινομενικά αυθαίρετες μετρήσεις του χρόνου, πρέπει να γίνει σύνδεση μεταξύ μαθηματικών εννοιών και καθημερινών καταστάσεων, κάτι το οποίο καταφέρεται διαφορετικά με την εμπειρία. Η χρήση σχολικής λογοτεχνίας που να περιλαμβάνει έννοιες χρονικές, όπως το «πριν», το «μετά», το «βράδυ» κλπ. μπορεί να βοηθήσει στην κατεύθυνση αυτή.

Αν και παλαιότερα, η διδασκαλία της ώρας δεν επικεντρωνόταν τόσο σε έννοιες όπως τα δευτερόλεπτα και η έκφραση των ωρών σε κλίμακα 24-ώρου, η εισαγωγή της τεχνολογίας στην καθημερινή ζωή, με τη χρήση προσωπικών υπολογιστών και κινητών τηλεφώνων, που κατά κόρον μεταχειρίζονται μια τέτοια αποτύπωση του χρόνου, έφερε αυτές τις έννοιες πιο κοντά στην καθημερινότητά μας, κάνοντας τη διδασκαλία της ώρας μια μεγαλύτερη πρόκληση.

Κατά συνέπεια, το αναλογικό ρολόι αποτελεί μια απεικόνιση του χρόνου περισσότερο προσανατολισμένη στη λειτουργία της όρασης, μπορεί να παρουσιαστεί σε μέσο διδασκαλίας με κλιμακωτή δυσκολία και δίνει μια περισσότερο οπτικοποιημένη αίσθηση του χρόνου που περνά στο μαθητή, απαλλάσσοντάς τον από έννοιες πολύπλοκότερες από όσο απαιτείται. Είναι μεν περισσότερο ξένο σε δομή από ένα ψηφιακό, θέτει όμως γερές βάσεις για μια βαθύτερη κατανόηση του χρόνου. Η διδασκαλία της ώρας από ένα αναλογικό ρολόι θα πρέπει να επικεντρώνεται στην επίλυση των μαθησιακών δυσκολιών που προαναφέραμε και να μη προβαίνει σε περισσότερο εντατικές μεθόδους από όσο χρειάζεται. Υπάρχουν πολλές περιπτώσεις, όπου μαθητές αντιμετωπίζουν δυσκολίες της κατανόησης του χρόνου ακόμα και μετά το Δημοτικό σχολείο. Εφόσον ακολουθηθούν συστηματοποιημένες και κλιμακωτές δυσκολίας μέθοδοι διδασκαλίας, είναι δυνατόν να συνειδητοποιηθεί σωστά η έννοια του χρόνου

Μεταπτυχιακή Διατριβή

Ζούγρης Π. Αναστάσιος

και της μέτρησής του, έστω και σε κόστος της συνολικής διάρκειας που θα απαιτηθεί για τη διδασκαλία αυτή.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

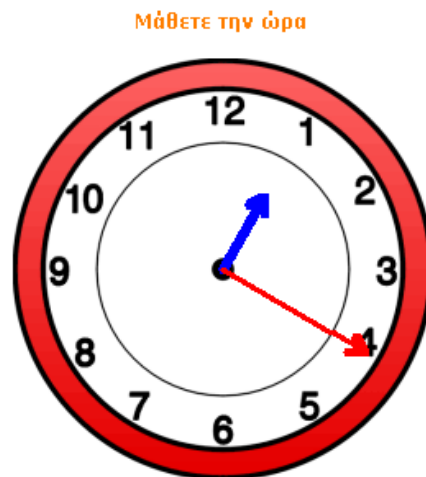
5 Υπάρχουσες λύσεις

Στον τομέα της διδασκαλίας της ώρας, υπάρχει ένας αριθμός εφαρμογών που μνημονεύονται συχνά σε διαδικτυακές αναφορές και περιλαμβάνουν εκπαιδευτικά εργαλεία που απευθύνονται σε μαθητές, αλλά και καθηγητές. Θα αναφέρουμε τα περισσότερο ενδιαφέροντα εργαλεία που εντοπίσαμε, παρουσιάζοντας συνοπτικά τη λειτουργικότητά τους, για να καταλήξουμε στο πώς συνεισφέρει η εφαρμογή μας στην κατεύθυνση αυτή.

5.1 Ελληνόφωνο υλικό

Όσον αφορά το ελληνόφωνο υλικό, το περισσότερο πλούσιο βρέθηκε στον Ιστότοπο «Πράσινη Πρίζα», που περιέχει μια σειρά από συνδέσμους σε εκπαιδευτικά λογισμικά κάθε είδους, απευθυνόμενα σε μικρούς μαθητές. Στον ιστότοπο αυτό αφιερώνεται ειδικός χώρος για τη μάθηση της ώρας, όπου δίνονται σύνδεσμοι για λογισμικά και παρέχεται εκπαιδευτικό υλικό που μπορεί να χρησιμοποιηθεί από δασκάλους. Παρέχονται δικτυακές εφαρμογές, καρτέλες εξάσκησης για χρήση από τους καθηγητές και σύνδεσμοι στα ανάλογα εδάφια σχολικών βιβλίων.

Θα παρουσιαστούν τώρα ορισμένα από τα περισσότερο ενδιαφέροντα λογισμικά και φύλλα δραστηριοτήτων που εντοπίστηκαν μέσω του ιστότοπου <http://www.prasinipriza.com/>:

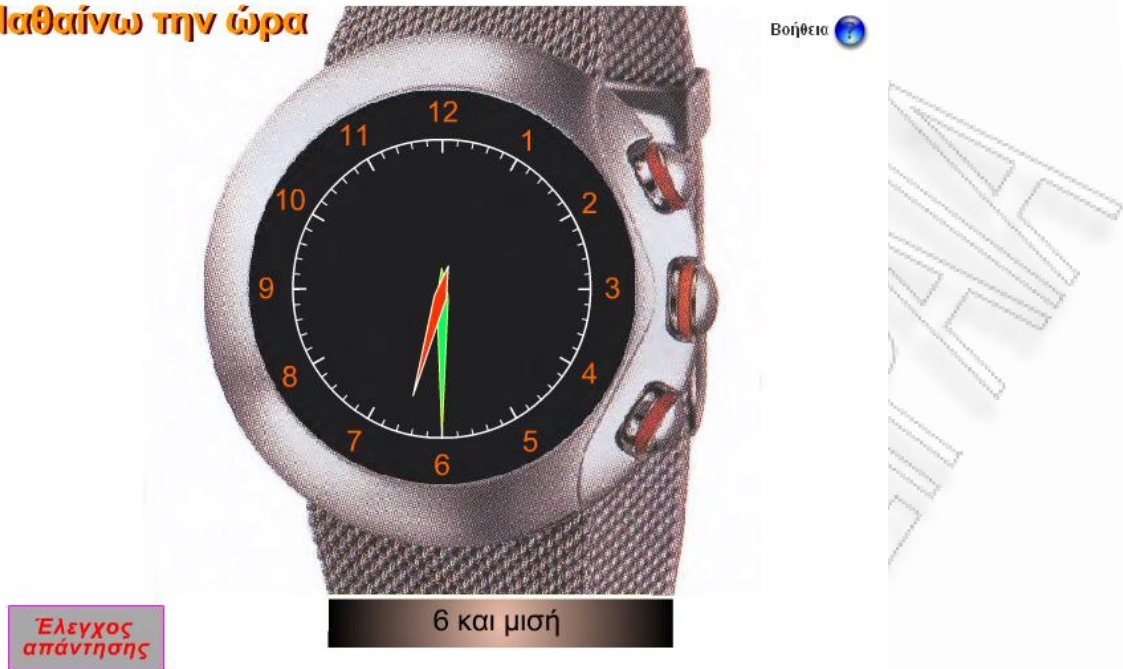


Τα καταφέρατε ! Επιτυχίες: 2 από 2.

OK

Εικόνα 1: Πρόγραμμα που αφήνει το μαθητή να τοποθετήσει τους δείκτες του ρολογιού στη ζητούμενη ώρα, παρέχει ανατροφοδότηση και σκορ. Πηγή <http://junior.imo.gr/games/time/index.html>

Μαθαίνω την ώρα



δασκική: Κοσμάς Αθανασιάδης

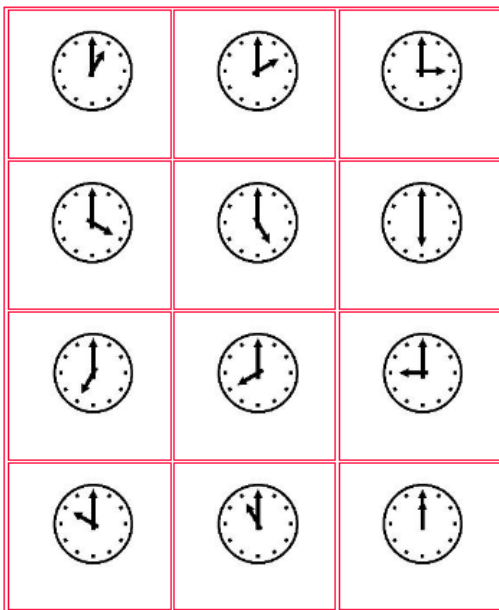
Εικόνα 2: Πρόγραμμα διασκευασμένο από αγγλόφωνη πηγή, το οποίο παρέχει ελληνικές ηχογραφημένες οδηγίες και ζητά από το μαθητή να τοποθετήσει τους δείκτες στη σωστή ώρα. Πηγή: <http://e-math.eduportal.gr/>



Εικόνα 3: Πρόγραμμα παρόμοιο με το προηγούμενο, περισσότερο προσαρμοσμένο σε μικρότερους μαθητές και με δυνατότητα σκορ και αριθμού ευκαιριών. Πηγή: <http://e-math.eduportal.gr/>

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

ΚΑΡΤΕΛΕΣ ΜΕ ΤΙΣ ΩΡΕΣ ΤΟΥ ΡΟΛΟΓΙΟΥ
γράφουμε την ώρα κάτω από κάθε καρτέλα
κόβουμε τις καρτέλες και τις βάζουμε στη σειρά

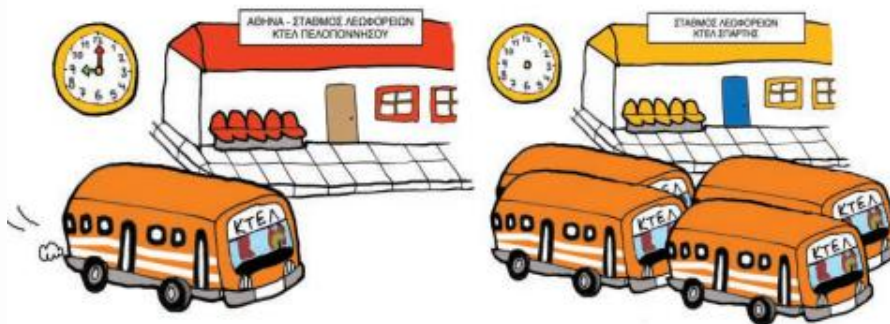


πίσω στη σελίδα www.prasinipriza.com
[ΠΑΙΧΝΙΔΙΑ](#) [ΜΑΘΗΜΑΤΙΚΑ](#) [ΩΡΑ-ΡΟΛΟΙ](#)
© Αθηνά Κυριακίδου Χριστοφόρου Σπ. Με επιμέλεια όλων των διακρίσεων.

Εικόνα 4: Καρτέλες προς εκτύπωση για δραστηριότητα από μαθητές.
Πηγή <http://www.prasinipriza.com/>

Εργασία

Τι ώρα θα δείχνει το ρολόι στο τέλος του ταξιδιού;



Ξεκίνησε στις το πρωί. Έφτασε στη Σπάρτη μετά από 3 ώρες και μισή.

Δηλαδή έφτασε στις και μισή το μεσημέρι.

Συμπέρασμα

Όταν η ώρα είναι «και μισή», ο μεγάλος δείκτης (λεπτοδείκτης) δείχνει το 6.

Παραδείγματα:



3 και μισή



6 και μισή



7 και μισή


Εικόνα 5: Εδάφιο του βιβλίου των Μαθηματικών Δευτέρας Δημοτικού, όπου διδάσκονται οι μισές και οι ολόκληρες ώρες. Πηγή <http://pi-schools.sch.gr/>

ΤιςΤας: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

5.2 Αγγλόφωνο υλικό

Όσον αφορά τις αγγλόφωνες πηγές, μπορούν να εντοπιστούν περισσότερο εύχρηστες εφαρμογές, με γραφικά περιβάλλοντα πιο ευχάριστα για τους μικρούς μαθητές. Θα παρουσιάσουμε ορισμένα λογισμικά που εντοπίσαμε και παρέχονται σε ελεύθερη διαδικτυακή πρόσβαση από διάφορους ιστότοπους:

Learn to tell time!



Enter the correct time below, shown on the clock above.
(Hours between 1 and 12, Minutes between 0 and 55 in increments of 5.)

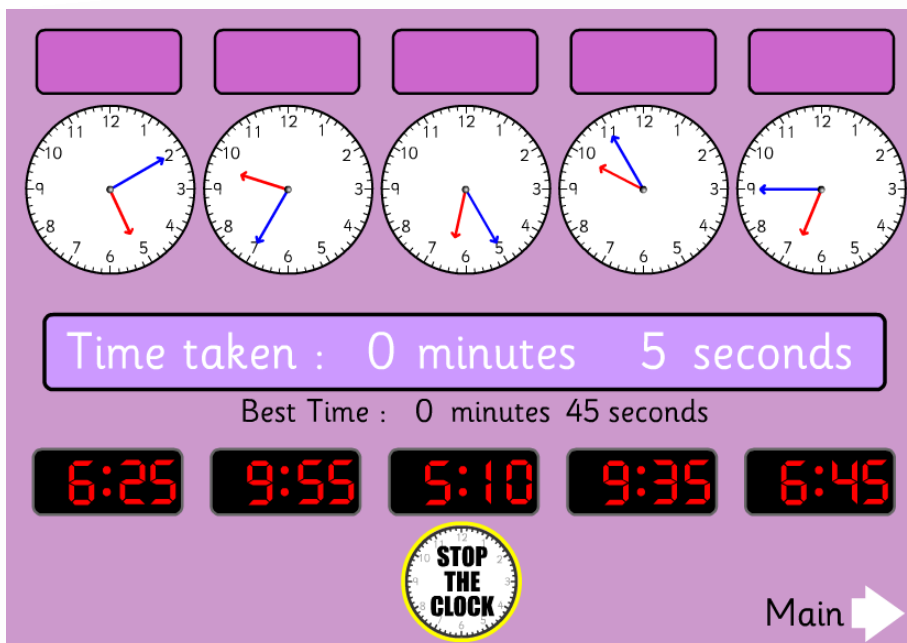
Hours: Mins :

Εικόνα 6: Το απλούστερο από τα εργαλεία που συναντήσαμε, παρέχει δυνατότητα πληκτρολόγησης της ζητούμενης ώρας και ανατροφοδότηση. Πηγή <http://playtolearn.com/>



Εικόνα 7: Ευχάριστα χρησιμοποιούμενη εφαρμογή, όπου ο χρήστης διαλέγει το ρολόι που δείχνει τη ζητούμενη ώρα και ένα κινούμενο σχέδιο επιβεβαιώνει ή διορθώνει την απάντησή του. Πηγή: <http://www.ictgames.com/>

ΤicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον



Εικόνα 8: Σειρά παιχνιδιών με όνομα "Stop the clock!", όπου ο μαθητής αντιστοιχίζει τα αναλογικά με τα ψηφιακά ρολόγια και η απόδοσή του χρονομετρείται. Αποτελείται από πέντε παιχνίδια, με την ίδια διεπαφή αλλά αυξανόμενη δυσκολία στις ασκήσεις. Πηγή: <http://www.oswego.org/ocsd-web/games/StopTheClock/>

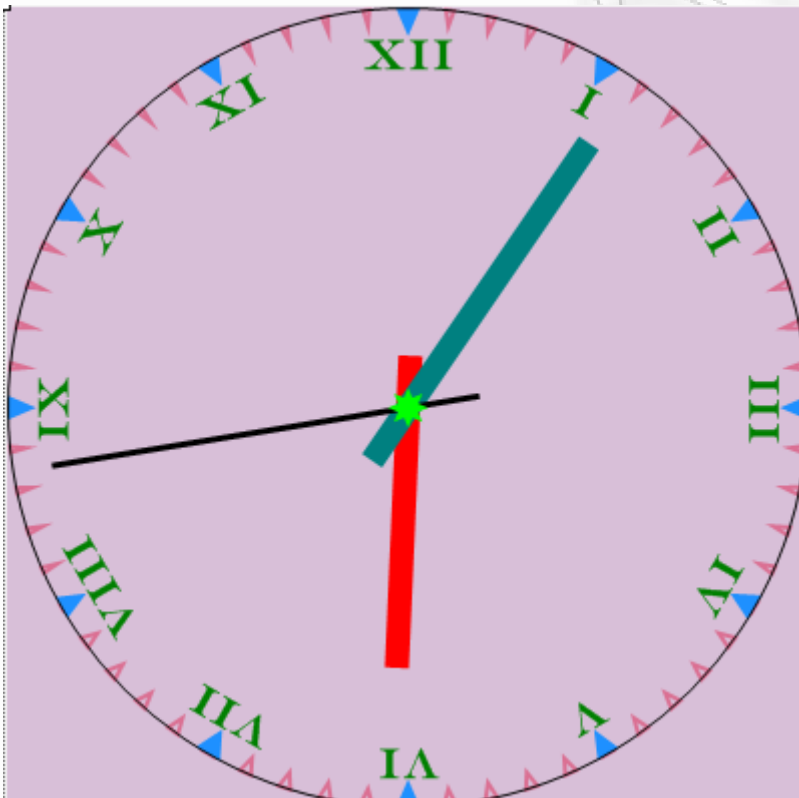
Οι εφαρμογές που παρέχονται για την εκπαίδευση στην ανάγνωση της ώρας ποικίλουν όπως είδαμε, όμως δεν παρέχουν συνήθως δυνατότητες μοντελοποίησης χρηστών ή προσαρμοστικότητας. Είναι αμετάβλητες σε διεπαφές και προσφέρουν την ίδια λειτουργικότητα σε όλους τους χρήστες. Η προσέγγιση που εμείς ακολουθήσαμε στόχευε στη δημιουργία μιας περισσότερο ευέλικτης και προσαρμοστικής εφαρμογής, με λειτουργίες λήψης στατιστικών στοιχείων και παροχής αντίστοιχων συμβουλών και με στοιχεία που διαφοροποιούνται ανάλογα με τις ομάδες χρηστών που τη χειρίζονται και ανάλογα με την απόδοση αυτών στις ασκήσεις μάθησης της ώρας.

6 Τρόπος παρουσίασης θεωρίας και υπόβαθρο

Η εφαρμογή που αναπτύχθηκε λοιπόν, επικεντρώνεται στην κάλυψη ασκήσεων σχετικών με τη μάθηση της ώρας σε ένα αναλογικό ρολόι. Στοχεύει στην εισαγωγή του χρήστη σε ασκήσεις βαθμωτής δυσκολίας, στην αξιολόγηση της απόδοσής του και στην παροχή ανατροφοδότησης για αυτή. Ενσωματώνει στοιχεία μοντελοποίησης του χρήστη, ενώ παρέχει προσαρμοστικότητα σε αρκετές από τις λειτουργίες της. Η γλώσσα που χρησιμοποιήθηκε στα τεστ, στη διδασκαλία και στη διεπαφή είναι τα Ελληνικά.

6.1 Η λειτουργικότητα της εφαρμογής

Ο πυρήνας της λειτουργικότητας της εφαρμογής είναι το αναλογικό ρολόι. Πρωταρχικός παράγοντας στη σχεδίαση του ρολογιού αποτέλεσε η αναγνωσιμότητα, η προσαρμοσιμότητα και η δυνατότητα επέκτασης και παραμετροποίησης. Το αναλογικό ρολόι που κατασκευάστηκε μπορεί να προσαρμοστεί σε μέγεθος, φόντο, τρόπο εμφάνισης δεικτών, σύμβολα για λεπτά και ώρες, και να λειτουργήσει και σαν ένα κανονικό ρολόι δείχνοντας ζωντανά την ώρα. Μπορεί επίσης να οριστεί ποιοι δεικτες θα φαίνονται. Για παράδειγμα η τρέχουσα υλοποίηση δεν συμπεριλαμβάνει δευτερόλεπτα οπότε και ο δεικτης των δευτερολέπτων δεν είναι ορατός. Έχει όμως υλποποιηθεί στο control και μπορεί να χρησιμοποιηθεί σε πιθανή μελλοντική επέκταση.



Εικόνα 9: Αναλογικό ρολόι που χρησιμοποιείται από την εφαρμογή και μπορεί να παραμετροποιηθεί.


Η εφαρμογή ξεκινάει με την εισαγωγή του χρήστη σε μία κεντρική οθόνη. Την πρώτη φορά που ένας χρήστης εισάγεται στο περιβάλλον, καλείται να πραγματοποιήσει εγγραφή σε μία ειδική φόρμα, όπου συμπληρώνει τα στοιχεία για το όνομα και το επώνυμό του, την ηλικία του και το φύλο του. Καταχωρεί επίσης ένα όνομα χρήστη και ένα κωδικό.



Εικόνα 10: Οθόνη εισόδου

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

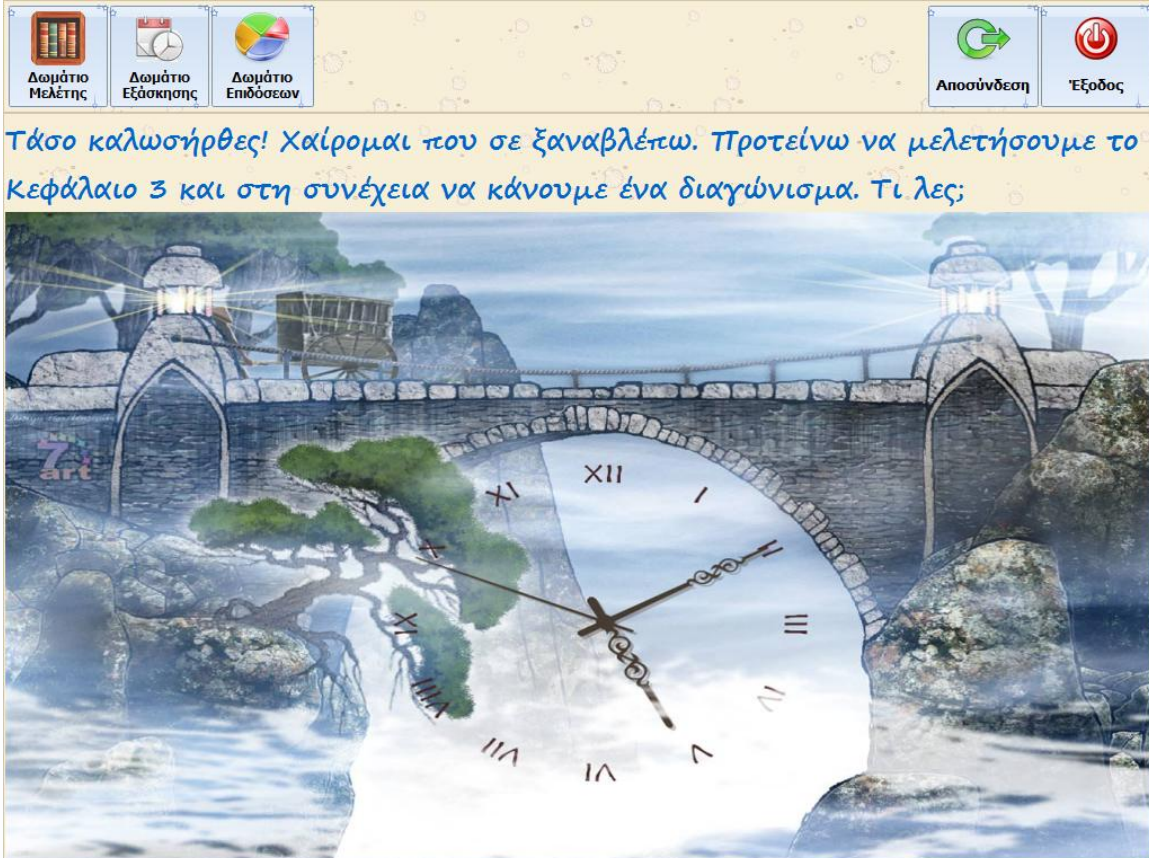
Βασικά Στοιχεία			
Όνομα:	<input type="text"/>	Όνομα Χρήστη:	<input type="text"/>
Επώνυμο:	<input type="text"/>	Κωδικός Χρήστη:	<input type="text"/>
Ηλικία:	<input type="text"/>	Επαλήθευση Κωδικού:	<input type="text"/>
Φύλο:	<input checked="" type="checkbox"/> Αγόρι <input type="checkbox"/> Κορίτσι		

Βοηθητικά Στοιχεία	
Από τις εικόνες προτιμάς...	
	
<input type="checkbox"/> Έχεις ξανακούσει για την ώρα!	

Έξοδος	Εγγραφή	Είσοδος
---------------	----------------	----------------

Εικόνα 11:Θθόνη εγγραφής χρήστη

Στη συνέχεια, ο χρήστης οδηγείται σε μία οθόνη καλωσορίσματος που του παρέχει μια σύντομη συμβουλή για το πως να ξεκινήσει την περιήγησή του στην εφαρμογή, όπου μπορεί να επιλέξει την ανάγνωση της θεωρίας για τη μάθηση της ώρας, την εκπόνηση εργασιών - διαγωνισμάτων που απευθύνονται στο επίπεδό του ή σε χαμηλότερα επίπεδα, καθώς και να παρακολουθεί τις επιδόσεις του μέχρι του παρόντος.

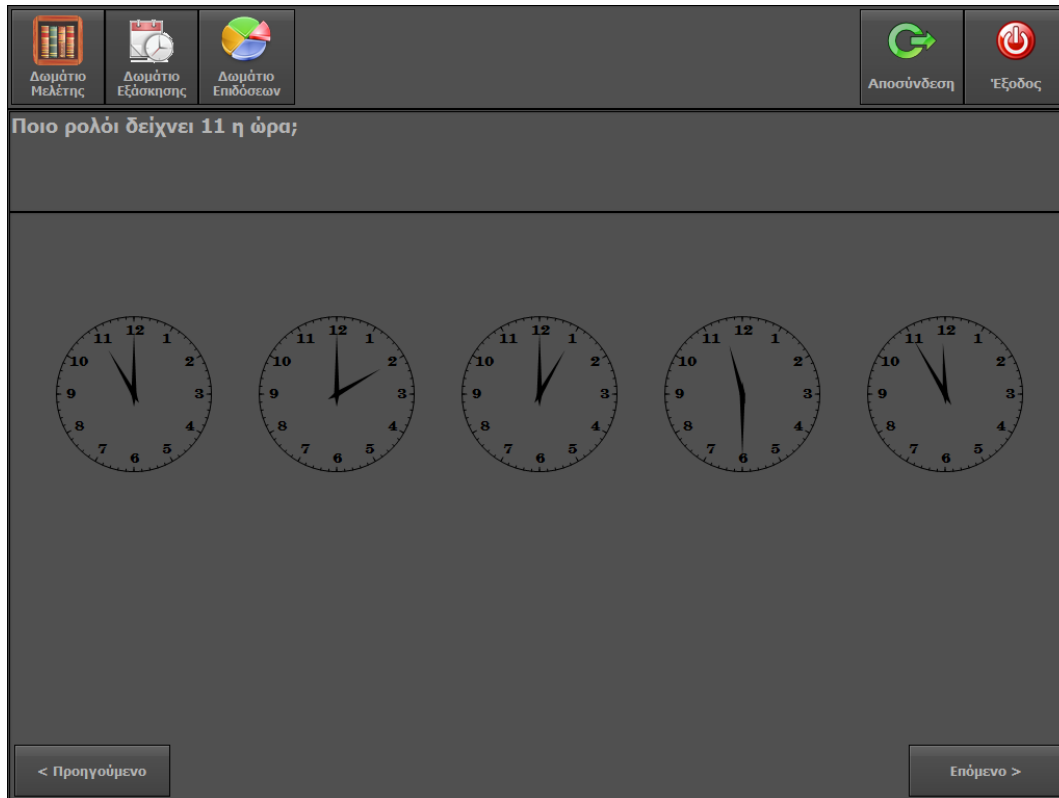


Εικόνα 12: Κεντρική οθόνη καλωσορίσματος. Το πρόγραμμα προτείνει στο χρήστη να μελετήσει το Κεφάλαιο 3 και μετά να κάνει ένα διαγώνισμα.

Οι χρήστες διαχωρίζονται σε 4 επίπεδα, που ανταποκρίνονται στη δυσκολία των ασκήσεων που θα κληθούν να αντιμετωπίσουν. Για κάθε επίπεδο χρήστη παρέχονται ασκήσεις του επιπέδου που του αναλογεί, αλλά και των χαμηλότερων επιπέδων. Τα επίπεδα δυσκολίας των ασκήσεων θα εξηγηθούν αναλυτικότερα αμέσως παρακάτω. Σημειώνουμε, πως το επίπεδο του χρήστη είναι γνωστό μόνο στην εφαρμογή και όχι στον ίδιο, λόγω του ότι θεωρήσαμε πως κάτι τέτοιο δε θα προσέφερε κάτι στη χρηστικότητα του λογισμικού. Αντ' αυτού, ο χρήστης δέχεται συστάσεις για ανάγνωση των κεφαλαίων που αντιστοιχούν στο επίπεδό του και εκπόνηση ασκήσεων του ανάλογου επιπέδου.

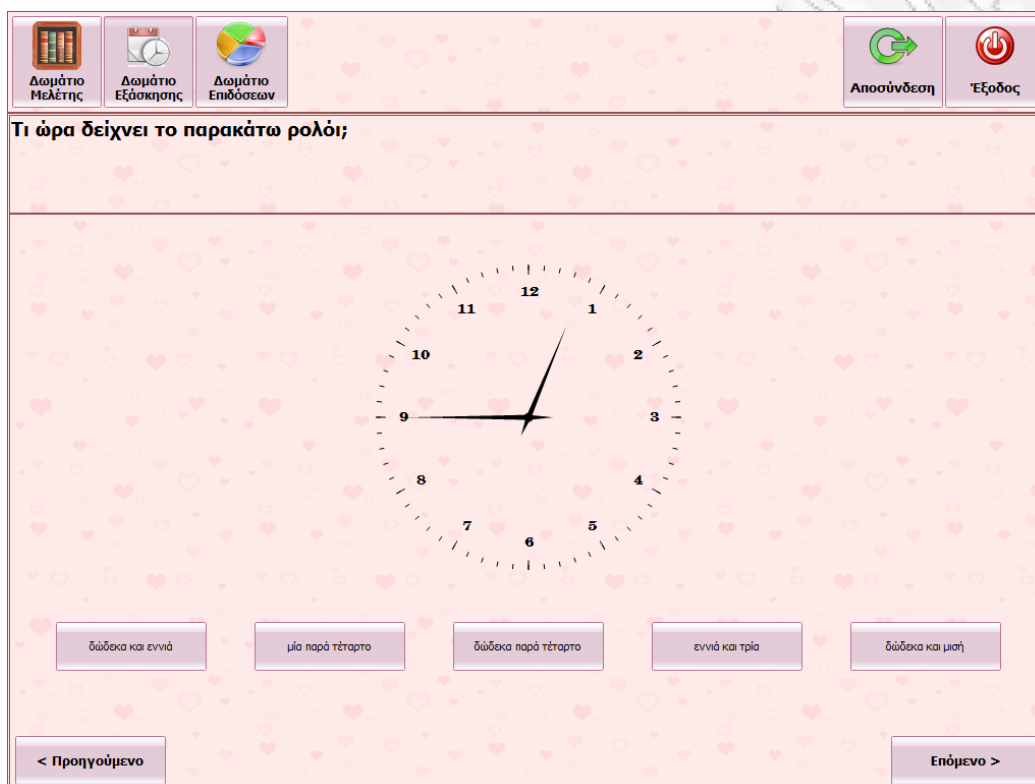
Οι ασκήσεις χωρίζονται λοιπόν σε 4 επίπεδα δυσκολίας. Ο διαχωρισμός των επιπέδων επιλέχθηκε σύμφωνα με προσεγγίσεις που προτείνουν τη σταδιακή δυσκολίας διδασκαλία της ώρας και αφορά τη λεπτομέρεια των ζητούμενων ασκήσεων. Αναλυτικότερα:

- **Επίπεδο 1:** Οι ασκήσεις αφορούν έλεγχο γνώσης ακέραιων ωρών. Ο χρήστης χρειάζεται να γνωρίζει μόνο την απεικόνιση στο ρολόι των ωρών «δώδεκα ακριβώς», «μία ακριβώς» και ούτω καθεξής. Η θεωρία που αφορά τα πρώτο επίπεδο βρίσκεται στο πρώτο κεφάλαιο στην ανάλογη φόρμα της διεπαφής.



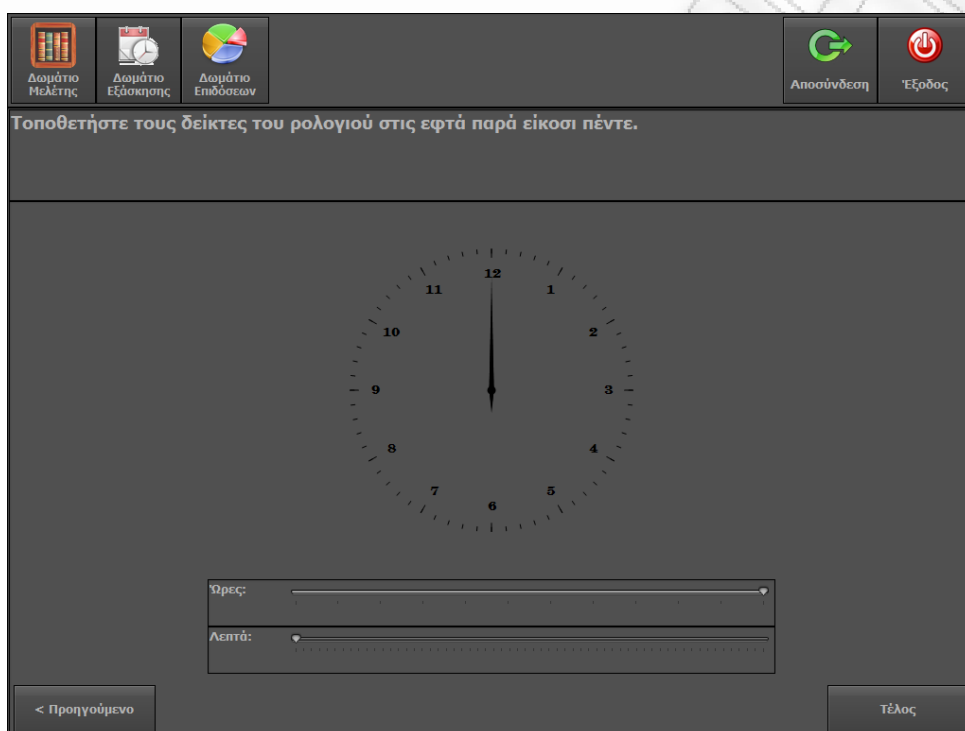
Εικόνα 13: Ερώτημα από διαγώνισμα πρώτου επιπέδου.

- **Επίπεδο 2:** Οι ασκήσεις αφορούν έλεγχο γνώσης της ώρας και υποδιαιρέσεων σε επίπεδο τετάρτου. Ο χρήστης χρειάζεται να γνωρίζει την απεικόνιση στο ρολόι των ωρών «δώδεκα και τέταρτο», «δώδεκα και μισή» και ούτω καθεξής. Η θεωρία που αφορά το δεύτερο επίπεδο βρίσκεται στο πρώτο και δεύτερο κεφάλαιο στην ανάλογη φόρμα της διεπαφής.



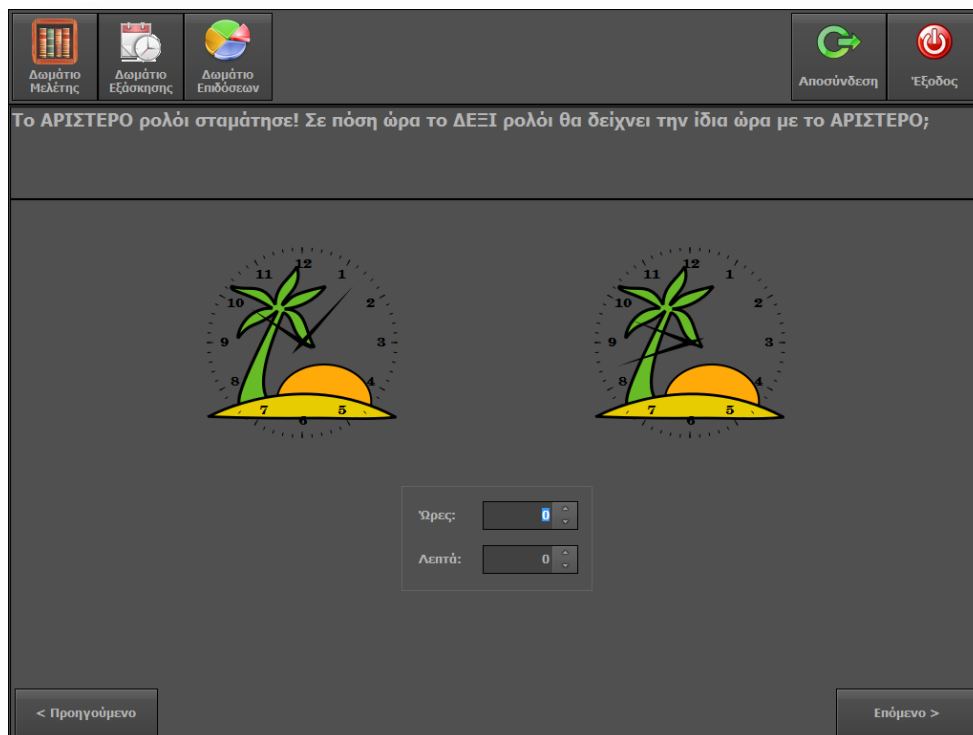
Εικόνα 14: Ερώτημα από διαγώνισμα δεύτερου επιπέδου.

- **Επίπεδο 3:** Οι ασκήσεις αφορούν έλεγχο γνώσης και απεικόνισης της ώρας σε επίπεδο πενταλέπτου στο αναλογικό ρολόι. Η θεωρία που αφορά το τρίτο επίπεδο βρίσκεται στο πρώτο, δεύτερο και τρίτο κεφάλαιο στην ανάλογη φόρμα της διεπαφής.



Εικόνα 15: Ερώτημα από διαγώνισμα τρίτου επιπέδου.

- **Επίπεδο 4:** Οι ασκήσεις προϋποθέτουν την πλήρη γνώση απεικόνισης της ώρας στο αναλογικό ρολόι, εισάγουν όμως περισσότερο πολύπλοκες ερωτήσεις, που αφορούν χρονικά διαστήματα. Η θεωρία που αφορά το τέταρτο επίπεδο βρίσκεται σε όλα τα κεφάλαια στη φόρμα της διεπαφής.



Εικόνα 16: Ερώτημα από διαγώνισμα τέταρτου επιπέδου.

Η θεωρία για τη διδασκαλία της ώρας έχει επιλεγεί από μία διαδικτυακή πηγή προσανατολισμένη στη διδακτική της ανάγνωσης του αναλογικού ρολογιού και είναι αποτυπωμένη σε απλή γλώσσα, κατανοητή από όλα τα επίπεδα μαθητών.

Οι κατηγορίες ασκήσεων που παρέχονται από την εφαρμογή είναι έξι και περιγράφονται αναλυτικά στη παράγραφο «Μοντελοποίηση».

Η εφαρμογή λοιπόν παρέχει ένα σύνολο από ερωτήσεις στο χρήστη, ομαδοποιημένες σε τεστ των 5, 10 ή 15 ερωτήσεων του επιπέδου του και αξιολογεί τις απαντήσεις του στο τέλος των τεστ. Μετά το πέρας των τεστ δίνει μία γενική εικόνα της απόδοσής του, παρέχει συμβουλές βασισμένες σε αυτή και ανάλογα με το πόσο καλά τα κατάφερε τον προάγει, του στερεί ή τον διατηρεί στο ίδιο επίπεδο. Παράλληλα παρέχει διαφορετική διεπαφή, ανάλογα με το φύλλο του χρήστη, ενώ αξιολογεί τις μαθησιακές του δυσκολίες, μέσω του είδους των λαθών που πραγματοποιεί.

6.2 Σενάριο χρήσης

Ένα αναλυτικό σενάριο χρήσης της εφαρμογής λοιπόν θα ήταν το εξής:




Ο χρήστης εισέρχεται για πρώτη φορά στην εφαρμογή. Δεν υπάρχει καταχωρημένος στη βάση, οπότε συμπληρώνει τη φόρμα εγγραφής.

Βασικά Στοιχεία

Όνομα:	<input type="text" value="Χριστίνα"/>	Όνομα Χρήστη:	<input type="text" value="christine"/>
Επώνυμο:	<input type="text" value="Γκιόκα"/>	Κωδικός Χρήστη:	<input type="password" value="****"/>
Ηλικία:	<input type="text" value="5"/>	Επαλήθευση Κωδικού:	<input type="password" value="****"/>
Φύλο:	<input type="checkbox"/> Αγόρι <input checked="" type="checkbox"/> Κορίτσι		

Βοηθητικά Στοιχεία

Από τις εικόνες προτιμάς...



Έχεις ξανακούσει για την ώρα!

Εικόνα 17: Οθόνη καταχώρησης στοιχείων του χρήστη. Όλα τα βασικά πεδία πρέπει να συμπληρωθούν. Τα βοηθητικά στοιχεία δεν συμπληρώνονται υποχρεωτικά ωστόσο η συμπλήρωσή της βοηθά στην καλύτερη προσαρμογή του προγράμματος στις ανάγκες – επιθυμίες του χρήστη.

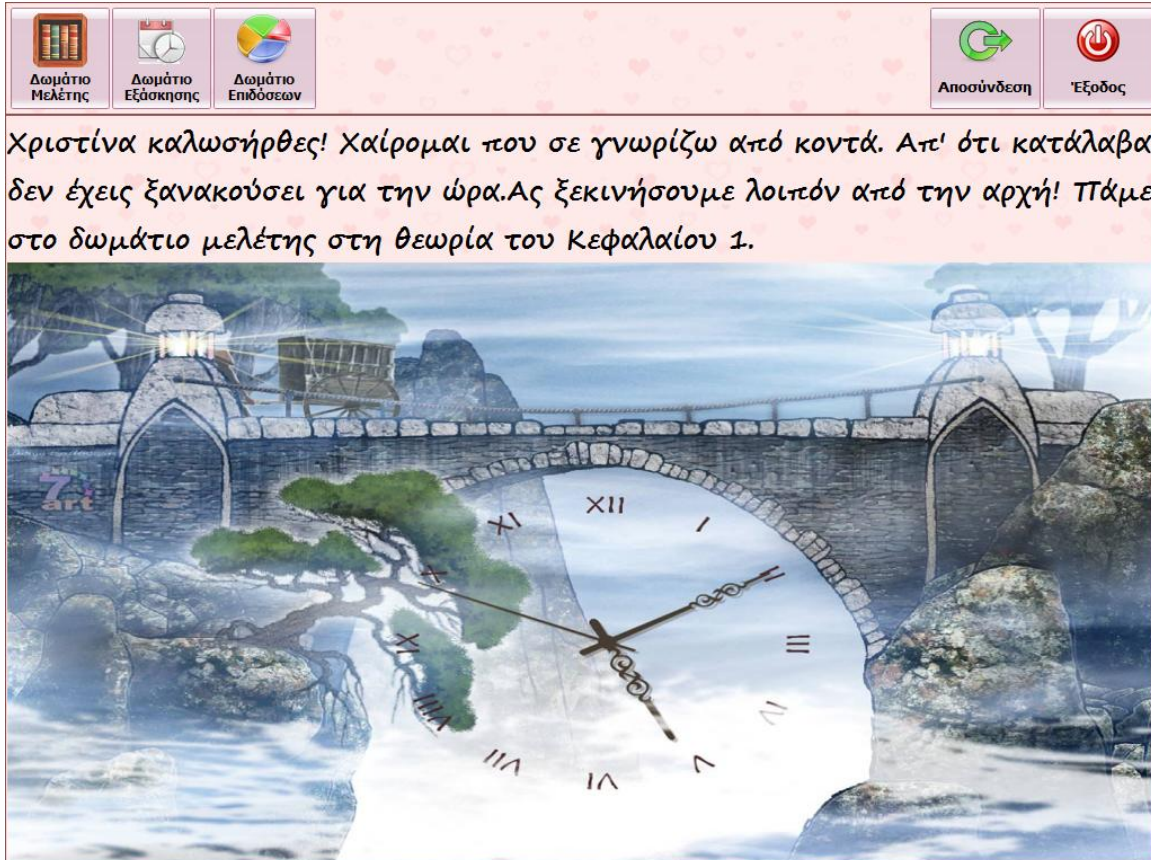
Η Χριστίνα είναι πλέον εγγεγραμμένη χρήστης και μπορεί να εισέλθει στην εφαρμογή μέσω της οθόνης εισόδου.

ΤicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

29

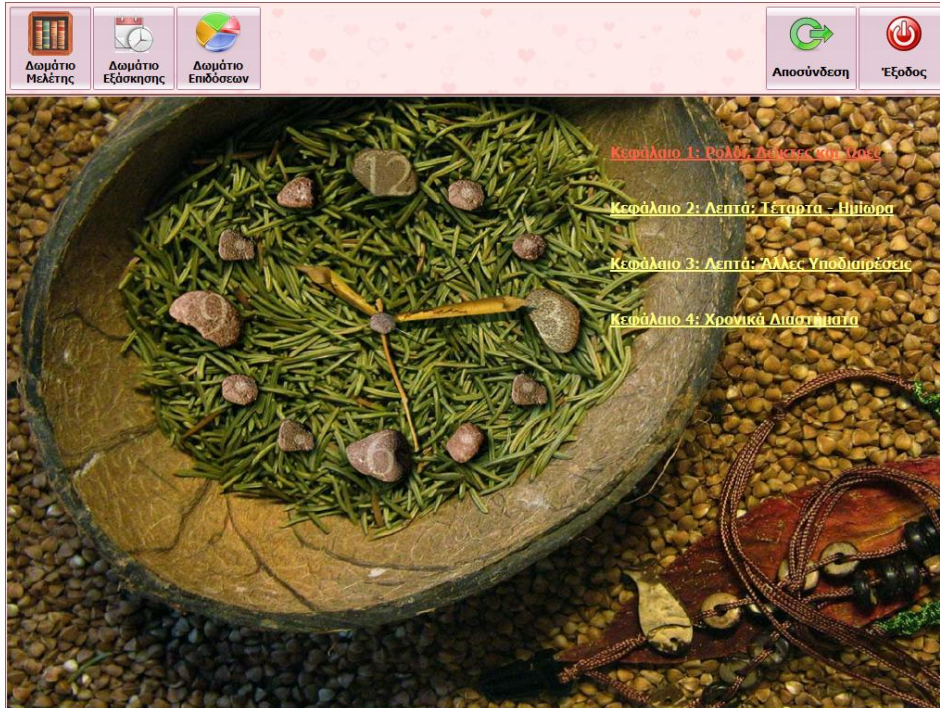


Εικόνα 18: Οθόνη εισόδου στο πρόγραμμα έπειτα από επιτυχημένη καταχώρηση.



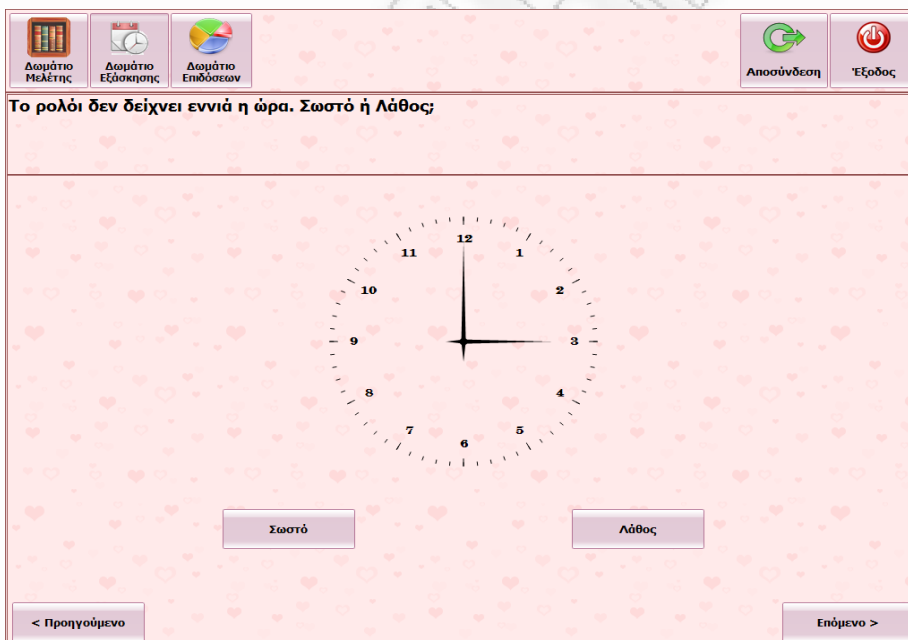
Εικόνα 19: Οθόνη υποδοχής

Έχει ηλικία 5 ετών και δεν έχει ξανακούσει για την ώρα, οπότε κατατάσσεται αυτόματα στο 1ο επίπεδο και γι' αυτό το πρόγραμμα της προτείνει να διαβάσει το κεφάλαιο 1. Επίσης «γνωρίζοντας» πως απευθύνεται σε κορίτσι, προσαρμόζει τη διεπαφή χρωματικά ώστε να είναι πιο φιλική σε ένα κορίτσι. Μετά λοιπόν την υποδοχή, η Χριστίνα οδηγείται στο δωμάτιο μελέτης όπου της προτείνεται να διαβάσει το Κεφάλαιο 1.



Εικόνα 20: Κεντρική οθόνη της εφαρμογής. Προτείνεται στο χρήστη να μελετήσει το Κεφάλαιο 1 που αντιστοιχεί στο επίπεδό του.

Αφού διαβάσει τη θεωρία, επιθυμεί να εκπονήσει ένα τεστ, το οποίο θα ανταποκρίνεται στο επίπεδο που της ανατέθηκε κατά στερεότυπο. Το τεστ θα περιλαμβάνει 10 ερωτήσεις, στις οποίες ο χρήστης θα επιτύχει ποσοστό 70%.



Εικόνα 21: Εκπόνηση διαγωνίσματος στο Κεφάλαιο 1.

Τελευταίο Διαγώνισμα

Χριστίνα, η επίδοσή σου στο τελευταίο διαγώνισμα ήταν ικανοποιητική. Αν και ο χρόνος που αφιέρωσες για μελέτη ήταν ελάχιστος, ο τρόπος που αντιμετώπισες το διαγώνισμα, έκρυψε καλά τις αδυναμίες σου. Συγκριτικά με το προηγούμενο διαγώνισμα, τα πήγες περίπου το ίδιο καλά, πάντως να ξέρεις ότι μπορείς και καλύτερα. Έκανες μερικά λάθη, κυρίως κατανόησης και σε δυσκόλεψαν οι ερωτήσεις πολλαπλής επιλογής με επιλογές κεκλιμένου.

60 %

Πληροφορίες	Αριθμός Διαγωνίσματος	Επίπεδο Δυσκολίας	Αριθμός Ερωτήσεων	Βαθμολογία	Χρόνοι		
					Ώρα Έναρξης	Ώρα Λήξης	Διάρκεια
	49	Πολύ Εύκολο	10	60%	20:55	20:58	00:03:05
					2/3/2011		

Η Συμβουλή Μου

Πιθανότατα δεν έχεις κατανοήσει στον απαιτούμενο βαθμό κάποια πράγματα στη θεωρία του Κεφαλαίου 1. Μία καλή επανάληψη πιστεύω θα σε βοηθούσε πολύ.

Εικόνα 22: Αποτελέσματα πρώτου τεστ.

Εμφανίζονται τα αποτελέσματα του τεστ που πραγματοποίησε η Χριστίνα, τα συνολικά στατιστικά της για το Κεφάλαιο 1 και συμβουλές που αφορούν τις επιδόσεις της και τα λάθη στα οποία έχει υποπέσει. Λόγω των επιδόσεων της ανεβαίνει στο επίπεδο 2. Ωστόσο το πρόγραμμα τη συμβουλεύει να κάνει μία επανάληψη στη θεωρία του κεφαλαίου 1 γιατί αντιλαμβάνεται ότι ο χρόνος που αφιέρωσε η Χριστίνα ήταν λίγος. Ακολουθώντας τη συμβουλή αυτή η Χριστίνα θα μελετήσει ξανά τη θεωρία του πρώτου κεφαλαίου πριν συνεχίσει και μελετήσει τη θεωρία του Κεφαλαίου 2.

Τα ρολόγια έχουν ένα μεγάλο δείκτη για να λείει τα λεπτά, και ονομάζεται λεπτοδείκτης και ένα μικρό δείκτη για να λείει την ώρα, και λέγεται ωροδείκτης. Ο ωροδείκτης είναι κοντός και πλατύς, ενώ ο λεπτοδείκτης μακρύς και λεπτός.

Ο Ωροδείκτης

Ο Ωροδείκτης λείει την ώρα της ημέρας. Όταν μετακινείται από έναν αριθμό στον επόμενο έχει περάσει μία ώρα. Στην επόμενη εικόνα ο ωροδείκτης δείχνει στο νούμερο 1, λέγοντάς μας ότι η ώρα είναι μία.

Προηγούμενη Σελίδα Επόμενη Σελίδα Επιστροφή

Εικόνα 23: Ο χρήστης μελετά τη θεωρία του Κεφαλαίου 1

Στη συνέχεια θα εκπονήσει τεστ του δευτέρου κεφαλαίου αποσπώντας χαμηλή επίδοση. Συνεπώς, η Χριστίνα θα υποβιβαστεί στο πρώτο επίπεδο και θα χρειαστεί να επαναλάβει τις ασκήσεις και τη θεωρία του Κεφαλαίου 1.

Τελευταίο Διαγώνισμα

Χριστίνα, η επίδοσή σου στο τελευταίο διαγώνισμα ήταν κακή. Επιπλέον, ο χρόνος που αφιέρωσες για μελέτη ήταν ελάχιστος, και το αποτέλεσμα μοιάζει να είναι συνέπεια της μικρής προσπάθειας που κατέβαλες. Χρειάζεται πολλή δουλειά... Προσπάθησε! Συγκριτικά με το προηγούμενο διαγώνισμα, η επίδοσή σου κυμάνθηκε στα ίδια χαμηλά επίπεδα, σίγουρα προσπάθησες; Έκανες πάρα πολλά λάθη, κυρίως σύγχυσης μεταξύ "παρά" και "και", και σε δυσκόλεψαν οι ερωτήσεις που απαιτείται υπολογισμός κάποιου χρονικού διαστήματος.

10 %

Διαγώνισμα	Πληροφορίες				Χρόνοι		
	Αριθμός Διαγωνίσματος	Επίπεδο Δυσκολίας	Αριθμός Ερωτήσεων	Βαθμολογία	Ώρα Έναρξης	Ώρα Λήξης	Διάρκεια
Ερωτήσεις	50	Εύκολο	10	10%	10:49	10:52	00:02:55
	49	Πολύ Εύκολο	10	60%	20:55	20:58	00:03:05
						2/3/2011	

Η Συμβουλή Μου

Η επίδοσής σου τελευταία είναι ιδιαίτερα χαμηλή. Για αυτό πιστεύω ότι το καλύτερο θα ήταν να επιστρέψεις στο Κεφάλαιο 1. Διάβασε καλά τη θεωρία, χωρίς να βιάζεσαι και όταν πιστεύεις ότι έχει προετοιμαστεί κατάλληλα κάνε πάλι ένα διαγώνισμα. Προσπάθησε να αποφύγεις τα εύκολα λάθη και πιστεύω ότι θα τα πας πολύ καλύτερα αυτή τη φορά. Καλή επιτυχία!

Εικόνα 24: Τα αποτελέσματα του χρήστη είναι αρκετά χαμηλά. Έτσι το πρόγραμμα τον συμβουλεύει για τα λάθη του και τον επαναφέρει στο επίπεδο 1.

Έτσι η Χριστίνα επαναλαμβάνει ένα τεστ του πρώτου κεφαλαίου και συγκεντρώνει σκορ 100%. Το πρόγραμμα την συγχαίρει και την επαφέρει στο δεύτερο επίπεδο.

Τελευταίο Διαγώνισμα

Χριστίνα, η επίδοσή σου στο τελευταίο διαγώνισμα ήταν άριστη. Επιπλέον, ο χρόνος που αφιέρωσες για μελέτη ήταν αρκετός, γεγονός που δείχνει ότι η προσπάθειά σου ανταμείβεται. Συνέχισε την καλή προσπάθειά! Συγκριτικά με το προηγούμενο διαγώνισμα, η βελτίωση είναι κάτι παραπάνω από αισθητή, και αυτό αντικατοπτρίζεται στα αποτελέσματά. Συνέχισε έτσι!

100 %

Διαγωνίσματα	Πληροφορίες				Χρόνοι		
	Αριθμός Διαγωνίσματος	Επίπεδο Δυσκολίας	Αριθμός Ερωτήσεων	Βαθμολογία	Ώρα Έναρξης	Ώρα Λήξης	Διάρκεια
Ερωτήσεις	51	Πολύ Εύκολο	10	100%	10:55	10:56	00:01:06
Επίπεδα	50	Εύκολο	10	10%	10:49	10:52	00:02:55
Χρόνοι	49	Πολύ Εύκολο	10	60%	20:55	20:58	00:03:05
Ρεκόρ						2/3/2011	

Η Συμβουλή Μου

Στα τελευταία διαγωνίσματα, τα έχεις πάει αρκετά καλά και πιστεύω ότι είσαι πλέον έτοιμη να περάσεις στο Κεφάλαιο 2. Τα αποτελέσματά σου θα ήταν ακόμα καλύτερα αν ήσουν πιο προσεκτική και συγκεντρωμένη. Προσπάθησε στο μέλλον να αποφεύγεις τα εύκολα λάθη. Καλή Επιτυχία!

Εικόνα 25: Άριστη βαθμολογία στο τεστ επιπέδου 1. Προαγωγή στο επίπεδο 2.

Τελικά η Χριστίνα εξέρχεται από τη εφαρμογή, διατηρώντας το τρίτο επίπεδο. Εάν συγκεντρώσει το απαιτούμενο ποσοστό ορθών απαντήσεων, θα ανέβει επίπεδο στο επόμενο τεστ που θα εκπονήσει. Όταν ο χρήστης εξέλθει από την εφαρμογή και επιστρέψει αργότερα, τα στοιχεία του έχουν αποθηκευτεί στη βάση και υπάρχει μνήμη για την απόδοση και τις μαθησιακές δυσκολίες του. Εάν μάλιστα συμπληρώσει πάλι ένα τεστ, παρέχεται ανατροφοδότηση σχετικά με την απόδοσή του σε σχέση με την προηγούμενη φορά που συμπλήρωσε τεστ.

7 Υλοποίηση

7.1 Παρουσίαση της εφαρμογής

Για την υλοποίηση της εφαρμογής επιλέχθηκε περιβάλλον .NET και συγκεκριμένα η γλώσσα προγραμματισμού C# στο εργαλείο ανάπτυξης Microsoft Visual Studio 2010 και το σχεσιακό σύστημα διαχείρισης Βάσεων Δεδομένων Microsoft SQL Server 2008 R2. Επίσης για την σχεδίαση της διεπαφής με το χρήστη χρησιμοποιήθηκαν τα components της DevExpress.

Η εφαρμογή λειτουργεί τοπικά στον υπολογιστή του χρήστη, παρέχοντας μία προσαρμοστική διεπαφή, με λειτουργίες μοντελοποίησης για τον ίδιο. Η δομή της εφαρμογής λοιπόν, περιλαμβάνει μία βάση σε SQL, με την οποία επικοινωνεί η διεπαφή και πραγματοποιεί το ζητούμενο αριθμό ερωτήσεων για κάθε χρήστη, αξιολογώντας την πρόοδο του και παρέχοντας ανάλογες συμβουλές αλλά και ανάλογης δυσκολίας ερωτήσεις.

7.1.1 Το Περιβάλλον Υλοποίησης

Το Microsoft .NET Framework είναι ένα πλαίσιο λογισμικού που είναι διαθέσιμο από διάφορα λειτουργικά συστήματα και προϊόντα της Microsoft. Περιλαμβάνει μια ιδιαίτερα μεγάλη βιβλιοθήκη κλάσεων σε συνηθισμένα προγραμματιστικά προβλήματα και μια εικονική μηχανή που είναι σε θέση να διαχειρίζεται την εκτέλεση προγραμμάτων γραμμένα ειδικά για το πλαίσιο αυτό. Οι έτοιμες κλάσεις που αποτελούν τη βασική βιβλιοθήκη κλάσεων του πλαισίου καλύπτουν ένα μεγάλο εύρος προγραμματιστικών αναγκών σε μια πληθώρα τομέων που συμπεριλαμβάνουν τα εξής: interface, data access, database connectivity, cryptography, web application development, numeric algorithms και network communications. Η βιβλιοθήκη κλάσεων χρησιμοποιείται από προγραμματιστές που τη συνδυάζουν με το δικό τους κώδικα προκειμένου να αναπτύξουν εφαρμογές.

Μια από τις καινοτομίες του .NET Framework είναι το Common Language Infrastructure (CLI). Σκοπός του είναι να παρέχει μια πλατφόρμα ουδέτερη σε γλώσσες προγραμματισμού για την ανάπτυξη και εκτέλεση εφαρμογών συμπεριλαμβάνοντας συναρτήσεις για τη διαχείριση εξαιρέσεων, τη συλλογή «σκουπιδιών» και την ασφάλεια.

Η C# είναι μία πολυδιάστατη γλώσσα προγραμματισμού και περιλαμβάνει προγραμματιστικές τεχνικές όπως ο δομημένος, δηλωτικός, συναρτησιακός, generic, αντικειμενοστρεφής προγραμματισμός. Η C# θεωρείται πως είναι μία ισχυρή, απλή, μοντέρνα, γενικής χρήσης, αντικειμενοστρεφής γλώσσα προγραμματισμού.

7.1.2 Αντικειμενοστρεφής Προγραμματισμός

Αντικειμενοστρεφής προγραμματισμός ονομάζεται ένα προγραμματιστικό υπόδειγμα το οποίο εμφανίστηκε στα τέλη της δεκαετίας του 1960 και καθιερώθηκε τη δεκαετία του 1990. Αντικατέστησε σε μεγάλο βαθμό το παραδοσιακό υπόδειγμα του δομημένου προγραμματισμού ενώ πρόκειται για μια μεθοδολογία ανάπτυξης προγραμμάτων όπου ο χειρισμός σχετιζόμενων δεδομένων και των διαδικασιών που επενεργούν σε αυτά γίνεται από κοινού, μέσω μιας δομής δεδομένων που τα περιβάλλει ως αυτόνομη οντότητα με ταυτότητα και δικά της χαρακτηριστικά. Αυτή η δομή δεδομένων καλείται αντικείμενο και αποτελεί πραγματικό στιγμιότυπο στη μνήμη ενός σύνθετου, και πιθανώς οριζόμενου από τον χρήστη τύπου δεδομένων που ονομάζεται κλάση. Η κλάση προδιαγράφει τόσο δεδομένα όσο και τις διαδικασίες οι οποίες επιδρούν επάνω τους. Αυτή υπήρξε και η πρωταρχική καινοτομία του αντικειμενοστρεφούς προγραμματισμού.

Υπό αυτό το πρίσμα μπορεί να οριστεί μια προδιαγραφή δομής αποθήκευσης η οποία να περιέχει τόσο ιδιότητες όσο και πράξεις ή χειρισμούς επ'αυτών των ιδιοτήτων. Αξιοσημείωτο είναι ότι τα αντικείμενα καταλαμβάνουν χώρο στη μνήμη του υπολογιστή ενώ οι κλάσεις αποτελούν απλώς «καλούπια». Οι αιτίες που ώθησαν στην ανάπτυξη του αντικειμενοστρεφούς προγραμματισμού ήταν οι ίδιες με αυτές που οδήγησαν στην ανάπτυξη του δομημένου προγραμματισμού π.χ. ευκολία συντήρησης, οργάνωσης, χειρισμού και επαναχρησιμοποίησης κώδικα μεγάλων και πολύπλοκων εφαρμογών. Τελικά όμως η αντικειμενοστρέφεια επικράτησε

και συνεχίζει να επικρατεί καθώς ήταν σε θέση να ανταπεξέλθει σε προγράμματα πολύ μεγαλύτερου όγκου και πολυπλοκότητας.

Σημαντικό χαρακτηριστικό του αντικειμενοστρεφούς προγραμματισμού είναι επίσης η έννοια της κληρονομικότητας. Κληρονομικότητα ονομάζεται η ιδιότητα των κλάσεων να επεκτείνονται σε νέες κλάσεις, ρητά δηλωμένες ως κληρονόμους (υποκλάσεις ή θυγατρικές κλάσεις), οι οποίες μπορούν να επαναχρησιμοποιήσουν τις μεταβιβάσιμες μεθόδους και ιδιότητες της γονικής τους κλάσης αλλά και να προσθέσουν δικές τους. Στιγμιότυπα των θυγατρικών κλάσεων μπορούν να χρησιμοποιηθούν όπου απαιτούνται στιγμιότυπα των γονικών (εφόσον η θυγατρική είναι κατά κάποιον τρόπο μία πιο εξειδικευμένη έκδοχή της γονικής), αλλά το αντίστροφο δεν ισχύει. Παράδειγμα κληρονομικότητας είναι μία γονική κλάση Vehicle (=Όχημα) και οι δύο πιο εξειδικευμένες υποκλάσεις της Car (=Αυτοκίνητο) και Bicycle (=Ποδήλατο), οι οποίες λέμε ότι "κληρονομούν" από αυτήν. Πολλαπλή κληρονομικότητα είναι η δυνατότητα που προσφέρουν ορισμένες γλώσσες προγραμματισμού μία κλάση να κληρονομεί ταυτόχρονα από περισσότερες από μία γονικές. Από μία υποκλάση μπορούν να προκύψουν νέες υποκλάσεις που κληρονομούν από αυτήν, με αποτέλεσμα μία ιεραρχία κλάσεων που συνδέονται μεταξύ τους "ανά γενιά" με σχέσεις κληρονομικότητας.

7.1.3 Φόρμες

Το σχεδιαστικό τμήμα της εφαρμογής αποτελείται από 6 βασικές φόρμες, που συγκροτούν όλο το τμήμα της διεπαφής. Αυτές οι φόρμες είναι οι εξής:

- Η φόρμα εισόδου (*frmLogin.cs*) αφορά στην οθόνη εισόδου του χρήστη και στην οθόνη εγγραφής.



Εικόνα 26: Σχεδίαση της φόρμας εισόδου, Login Screen

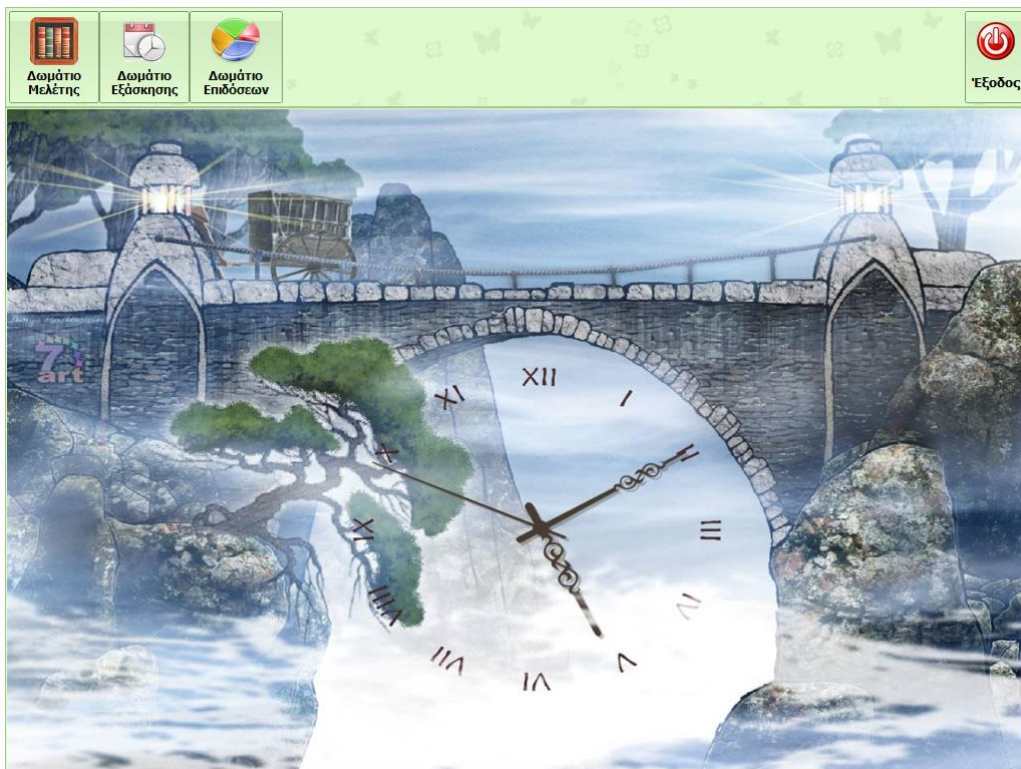
Βασικά Στοιχεία

Όνομα:	<input type="text"/>	Όνομα Χρήστη:	<input type="text"/>
Επώνυμο:	<input type="text"/>	Κωδικός Χρήστη:	<input type="text"/>
Ηλικία:	<input type="text"/>	Επαλήθευση Κωδικού:	<input type="text"/>
Φύλο:	<input checked="" type="checkbox"/> Αγόρι <input type="checkbox"/> Κορίτσι		

Έξοδος Εγγραφή Είσοδος

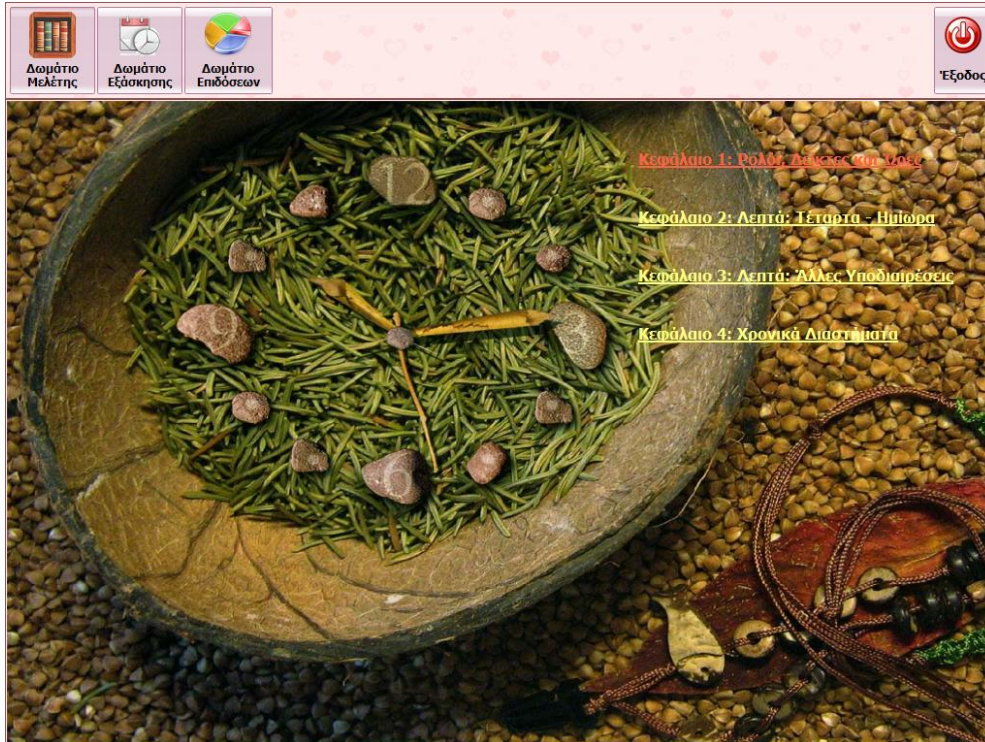
Εικόνα 27: Σχεδίαση της φόρμας εισόδου, Registration Form

- Η κύρια φόρμα (*frmMain.cs*) διαθέτει τις βασικές επιλογές περιήγησης στην εφαρμογή και φιλοξενεί όλες τις άλλες φόρμες της εφαρμογής. Έχει υλοποιηθεί *skin controller* έτσι ώστε να μπορεί να προσαρμοστεί η εμφάνισή της, αλλά και των φορμών που περιέχει στις απαιτήσεις του χρήστη.



Εικόνα 28: Σχεδίαση της κύριας φόρμας. Αποτελεί το δοχείο για όλες τις υπόλοιπες φόρμες.

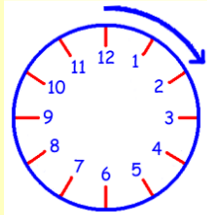
- Η φόρμα μελέτης (frmStudyRoom.cs) είναι το περιβάλλον μελέτης του χειριστή. Φιλοξενεί τα κεφάλαια προς ανάγνωση σημειώνοντας αυτό που προτείνει η μοντελοποίηση. Επιλέγοντας κάποιο από αυτά εμφανίζεται η θεωρία που αντιστοιχεί σε αυτό το κεφάλαιο και ξεκινάει η μέτρηση του χρόνου μελέτης για το συγκεκριμένο κεφάλαιο και τον συγκεκριμένο χρήστη.



Εικόνα 29: Φόρμα μελέτης. Εμφανίζονται τα διαθέσιμα προς μελέτη κεφάλαια. Με κόκκινο σημειώνεται αυτό που προτείνεται από τη μοντελοποίηση.

Το Πρόσωπο του Ρολογιού

Ας ονομάσουμε τη μπροστινή επιφάνεια από ένα ρολόι πρόσωπο. Θα μιλήσουμε για τα διαφορετικά μέρη του προσώπου ενός ρολογιού. Ρίξε μια ματιά σε αυτή την εικόνα.



Τα κόκκινα σημάδια χωρίζουν το ρολόι σε 12 τμήματα. Κάθε μέρος είναι μία ώρα.

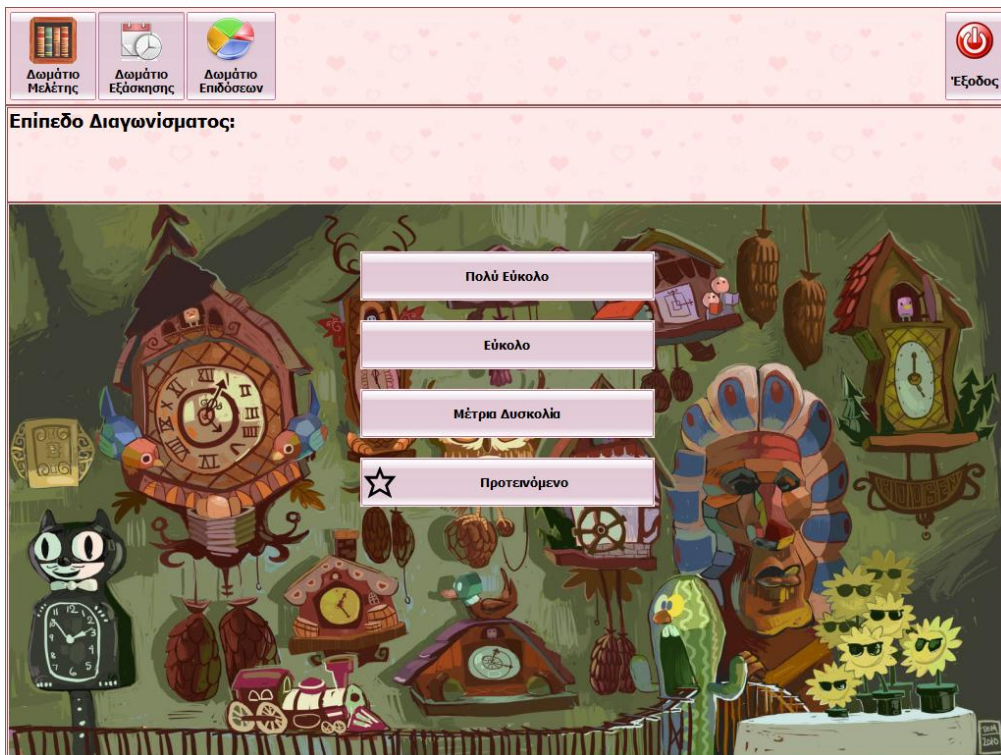
Πολλά ρολόγια έχουν έναν αριθμό ή ένα σύμβολο δίπλα σε κάθε σήμα, για να είναι πιο εύκολο να πεις την ώρα. Σημειώστε ότι ο αριθμός 12 είναι πάντα στην κορυφή και οι αριθμοί 1 έως 11 βρίσκονται γύρω

Προηγούμενη Σελίδα Επόμενη Σελίδα Επιστροφή

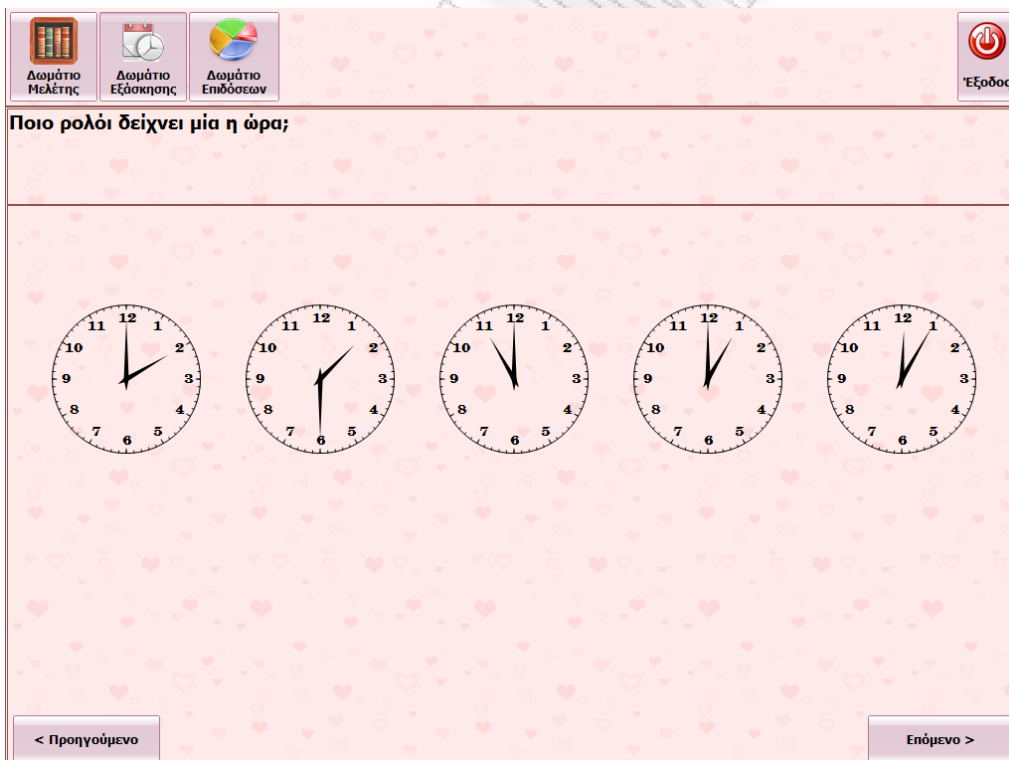
Εικόνα 30: Φόρμα μελέτης, η διεπαφή ανάγνωσης.

- Η φόρμα εξάσκησης (frmPracticeRoom.cs) υλοποιεί το περιβάλλον ενός διαγωνίσματος. Διατυπώνει τις ερωτήσεις και εφόσον αυτές υπάρχουν τις διαθέσιμες απαντήσεις.

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον



Εικόνα 31: Φόρμα εξάσκησης. Επιλογή επιπέδου δυσκολίας του διαγωνίσματος.



Εικόνα 32: Φόρμα εξάσκησης. Διεπαφή διαγωνίσματος

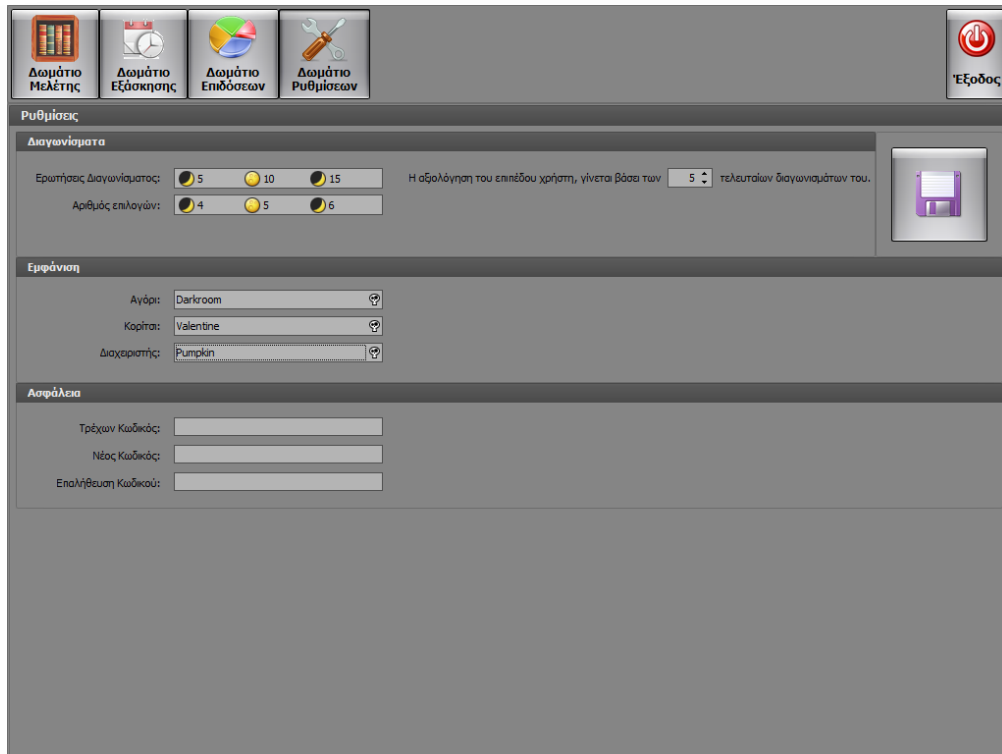
TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

- Φόρμα στατιστικών (*frmStatisticsRoom.cs*). Οπτικοποιεί τα αποτελέσματα τόσο της μελέτης της θεωρίας όσο και αυτά των διαγωνισμάτων με τρόπο φιλικό προς το χρήστη και εύκολα αντιληπτό.

The screenshot displays the 'frmStatisticsRoom.cs' application interface. At the top, there are navigation icons for 'Δωμάτιο Μελέτης', 'Δωμάτιο Εξάσκησης', and 'Δωμάτιο Επιδόσεων', along with a 'Έξοδος' button. The main content area is titled 'Τελευταίο Διαγώνισμα' and includes a feedback message: 'Τόσο, η επίδοσή σου στο τελευταίο διαγώνισμα ήταν κακή. Επιπλέον, ο χρόνος που αφιέρωσες για μελέτη ήταν ελάχιστος, και το αποτέλεσμα μοιάζει να είναι συνέπεια της μικρής προσπάθειας που κατέβαλλες. Χρειάζεται πολλή δουλειά... Προσπάθησε! Συγκριτικά με το προηγούμενο διαγώνισμα, η επίδοσή σου κυμάνθηκε στα ίδια χαμηλά επίπεδα, σίγουρα προσπάθησες Έκανες πάρα πολλά λάθη, κυρίως κατανόησης και σε δυσκόλεψαν οι ερωτήσεις που απαιτείται υπολογισμός κάποιου χρονικού διαστήματος.' A progress indicator shows '0 %' with a sad face icon. Below this, there are two columns: 'Τύποι Ερωτήσεων' and 'Τύποι Λαθών'. The 'Τύποι Ερωτήσεων' column lists: 'Πολλαπλής Επιλογής - Επιλογές Εικόνας' (0 στις 2 (0%)), 'Πολλαπλής Επιλογής - Επιλογές Κεμένου' (0 στις 1 (0%)), 'Κατασκευαστική - Ρύθμιση Ύψους' (0 στις 1 (0%)), 'Κατασκευαστική - Υπολογισμός Διάρκειας' (0 στις 3 (0%)), 'Αξιολόγηση - Σωστό ή Λάθος' (0 στις 3 (0%)), and 'Ανοιχτές Ερωτήσεις'. The 'Τύποι Λαθών' column lists: 'Γενικά Λάθη - Κατανόησης' (7 από 10 (70%)), 'Σύγχυση μικρού - μεγάλου δείκτη' (0 από 10 (0%)), 'Σύγχυση μεταξύ εννοιών "παρά" - "και"' (0 από 10 (0%)), 'Λάθη που αφορούν διάρκεια' (3 από 10 (30%)), 'Ανάγνωση του λεπτοδείκτη' (0 από 10 (0%)), 'Ανάγνωση του ωροδείκτη' (0 από 10 (0%)), and 'Ορθογραφικά Λάθη' (0 από 10 (0%)). At the bottom, there is a 'Η Συμβουλή Μου' section with a message: 'Πιθανότατα δεν έχεις κατανοήσει στον απαιτούμενο βαθμό κάποια πράγματα στη θεωρία του Κεφαλαίου 3. Μια καλή επανάληψη πιστεύω θα σε βοηθήσει πολύ. Τα διαγωνίσματα σε βοηθούν σημαντικά να καταλάβεις το αντικείμενο. Με λίγη εξάσκηση ακόμα νομίζω πως θα τα πηγαίνεις περφημα! Τα αποτελέσματά σου θα ήταν ακόμα καλύτερα αν ήσουν πιο προσεκτικός και συγκεντρωμένος. Προσπάθησε στο μέλλον να αποφεύγεις τα εύκολα λάθη.' A smiley face icon is also present.

Εικόνα 33: Φόρμα στατιστικών. Παρέχει feedback για το τελευταίο διαγώνισμα και συμβουλές για το πως πρέπει να κινηθεί ο χρήστης από τώρα και στο εξής.

- Η φόρμα ρυθμίσεων (*frmSettings.cs*) παρέχει στο χρήστη τη δυνατότητα να μετβάλλει τη λειτουργία του προγράμματος, να τροποποιήσει την εμφάνισή του, και να αλλάξει κωδικούς πρόσβασης.



Εικόνα 34: Φόρμα ρυθμίσεων.

Η λειτουργικότητα της C# παρέχει τη δυνατότητα ελέγχου συμβάντων όταν πραγματοποιείται οποιαδήποτε αλλαγή πάνω σε μία φόρμα. Έτσι, έχουν προβλεφθεί σενάρια χρήσης για κάθε πιθανό γεγονός αλληλουχίας συμβάντων, παρέχοντας και τον αντίστοιχο έλεγχο σφαλμάτων ενσωματωμένο στις δυνατότητες κάθε φόρμας. Έτσι εξασφαλίζεται και η ακεραιότητα της βάσης δεδομένων καθώς τα προς εισαγωγή δεδομένα παράγονται μέσα από τυποποιημένες διαδικασίες κανονικοποίησης.

7.1.4 Βοηθητικές Βιβλιοθήκες και Custom Controls

➤ Η βιβλιοθήκη Utils

Η βιβλιοθήκη “Utils” είναι μία μεγάλη συλλογή από κλάσεις, μεθόδους, objects, interfaces, enums και resources που χρησιμοποιούνται με μεγάλη συχνότητα από την εφαρμογή υλοποιώντας μεγάλο τμήμα της λειτουργικότητας και των διαδικασιών που εκτελούνται στο παρασκήνιο. Παρακάτω επεξηγούνται συνοπτικά οι πιο σημαντικές από αυτές:

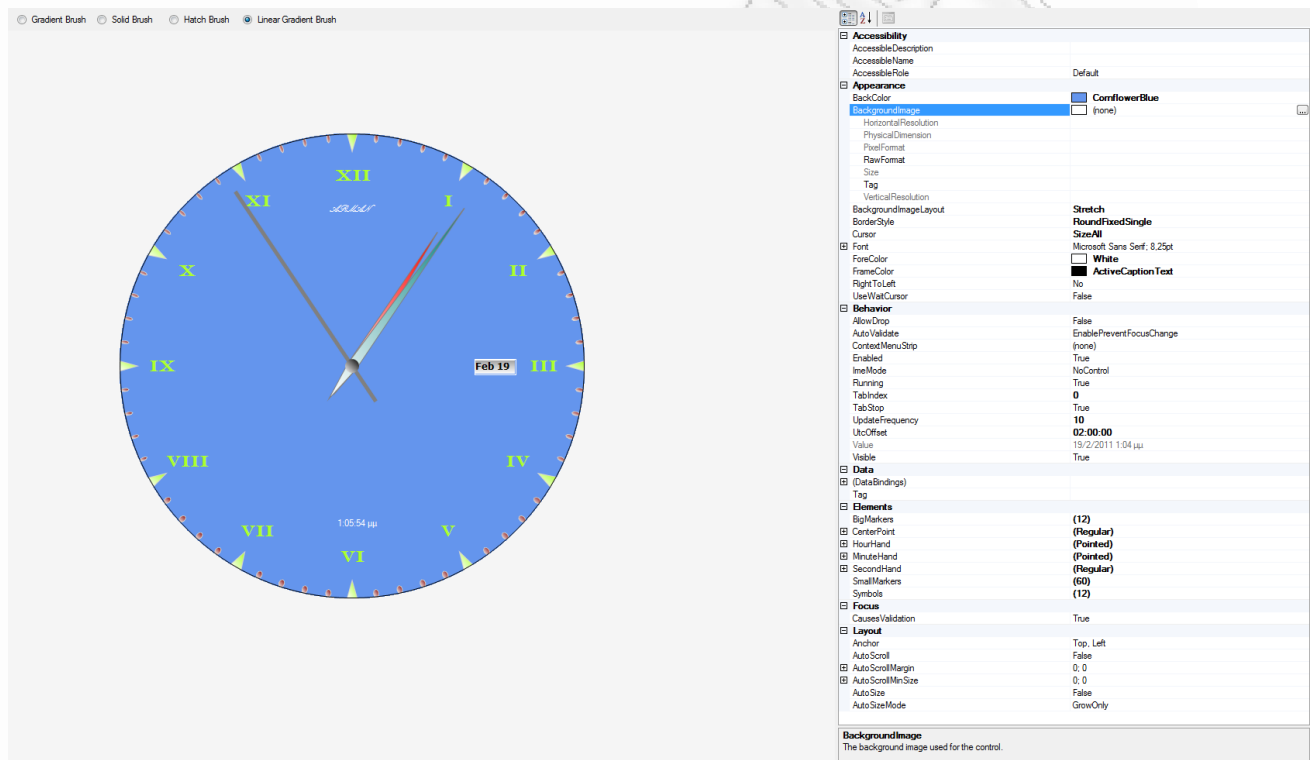
- *AdviceGenerator*. Είναι η γεννήτρια συμβουλών και γενικότερα όλου του feedback που παρέχεται σε μορφή κειμένου. Διαθέτει μία συλλογή από μερικές προκατασκευασμένες προτάσεις και όποια στιγμή της ζητηθεί, δημιουργεί μία μικρή παράγραφο με νόημα που ανταποκρίνεται στην εικόνα του χρήστη. Αν και υπάρχουν έτοιμα templates, οι τελικοί συνδυασμοί που μπορούν να δημιουργηθούν είναι αρκετοί ώστε η λειτουργία αυτής της κλάσης να μπορεί να θεωρηθεί απολύτως δυναμική. Επίσης το feedback που παρέχει είναι σε ζωντανό χρόνο. Για παράδειγμα αν ο χρήστης κάνει ένα διαγώνισμα στο επίπεδο 1 και προβιβαστεί στο επίπεδο 2, με το πέρας του διαγωνίσματος ο χειριστής θα βλέπει συμβουλές που ανταποκρίνονται στο έως τώρα επίπεδό του (1) και κατα συνέπεια στο διαγώνισμα που μόλις ολοκλήρωσε. Αν όμως επανέλθει και ζητήσει συμβουλή για 2^η φορά προφανώς είναι σωστότερο να

του δοθούν συμβουλές στο πως να κινηθεί απο εδώ και πέρα και κατα συνέπεια να αφορούν στο νέο του επίπεδο (2).

- *Answer*: Είναι το αντικείμενο που αντιπροσωπεύει μια οποιαδήποτε απάντηση σε μία ερώτηση. Σχετίζεται λοιπόν με την ερώτηση στην οποία ανήκει, αναφέρεται στον πιθανό τύπο λαθών που μοντελοποιεί ή αναφέρει ότι πρόκειται για τη σωστή απάντηση, και φυσικά έχει και το κείμενο της απάντησης που αφορά στον τρόπο με τον οποίο αυτή θα οπτικοποιηθεί στο χρήστη (αν η ερώτηση είναι πολλαπλής επιλογής) ή τον τρόπο με τον οποίο θα διασταυρωθεί με την απάντηση που θα δώσει ο χρήστης (σε μία ερώτηση άλλου τύπου).
- *ClockParams*: Είναι μια κλάση που ευθύνεται για τις παραμέτρους που πρέπει είτε να ληφθούν από το control του ρολογιού, είτε να σταλούν προς αυτό. Πέρα από τις τιμές για το μικρό και το μεγάλο δείκτη μπορεί να λάβει και μια τιμή για χρονικό διάστημα της ημέρας στο οποίο ανήκει η συγκεκριμένη ώρα (πρωί, μεσημέρι, απόγευμα, βράδυ) και να το οπτικοποιήσει στο ρολόι.
- *Question*: Είναι το αντικείμενο που αντιπροσωπεύει μία ερώτηση ενός διαγωνίσματος. Πέρα από το διαγώνισμα στο οποίο ανήκει, έχει ιδιότητες για τον τύπο της, το κείμενο (εκφώνηση) και τις πιθανές παραμέτρους που μπορεί να λάβει, και διαθέτει μεθόδους για να δημιουργήσει δυναμικά τις πιθανές απαντήσεις της εφόσον είναι ερώτηση πολλαπλής επιλογής. Τέλος διατηρεί τα στατιστικά για το χρόνο που απασχόλησε το χρήστη βοηθώντας στην εξαγωγή πολυτιμων συμπερασμάτων.
- *Randomizer*: Πρόκειται για ένα σύνολο μεθόδων που έχουν σαν στόχο την όσο το δυνατόν καλύτερη τυχαιοποίηση στις επιλογές του προγράμματος. Για παράδειγμα δε θα ήταν σωστό από τις πιθανές απαντήσεις μιας ερώτησης η σωστή απάντηση να είναι πάντα η δεύτερη. Έτσι έχουν υλοποιηθεί τέσσερις μέθοδοι που εξασφαλίζουν ότι οι επιλογές, όπου αυτές είναι απαραίτητες θα είναι όσο πιο τυχαίες γίνεται. Τρεις από αυτές είναι της μορφής Shuffle όπου δεδομένης μίας συλλογής αντικειμένων πραγματοποιεί μετάθεση αυτών σε τυχαία σειρά. Και μία είναι της μορφής SelectRandomN όπου δεδομένης μιας συλλογής αντικειμένων καλείται να επιλέξει σε τυχαία σειρά κάποια από αυτά τα αντικείμενα. Συχνά χρησιμοποιείται συνδυασμός αυτών των συναρτήσεων ώστε το αποτέλεσμα να είναι πραγματικά τυχαίο.
- *Settings*: Η κλάση που υλοποιεί τις ρυθμίσεις του προγράμματος.
- *StudySession*: Είναι το αντικείμενο ενός session μελέτης. Διατηρεί πληροφορία για το ποιός χρήστης μελέτησε πιο κεφάλαιο, πόσο χρόνο αφιέρωσε. Και ευθύνεται για την καταχώρηση των αντίστοιχων στατιστικών στη βάση δεδομένων.
- *Test*: Είναι το αντικείμενο που αντιπροσωπεύει ένα διαγώνισμα. Πέρα από το επίπεδο στο οποίο ανήκει, και τις πιθανές παραμέτρους που μπορεί να λάβει, διαθέτει μεθόδους για να δημιουργήσει δυναμικά τις ερωτήσεις που του ανήκουν. Ένα διαγώνισμα δημιουργείται κατ' απαίτηση του χρήστη και παρόλο που η δημιουργία του είναι δυναμική, πρέπει να αποθηκευτεί σαν πληροφορία μετά τη λήξη του, ώστε να εξαχθούν στατιστικά από αυτό. Έτσι όλη η χρήσιμη πληροφορία αποθηκεύεται στη βάση δεδομένων σε μορφή xml.
- *TestEvaluator*: Βαθμολογεί και αξιολογεί ένα διαγώνισμα. Πραγματοποιώντας και τις απαραίτητες συγκρίσεις με προηγούμενα διαγωνίσματα.
- *TestVisualizer*: Είναι η γέφυρα μεταξύ του αντικειμένου Test και του user interface. Ευθύνεται για τη δυναμική σχεδίαση των controls που διαφέρουν από ερώτηση σε ερώτηση καθώς και για την πλοήγηση μέσα στο test.
- *User*: ο χρήστης του προγράμματος γύρω από τον οποίο περιστρέφονται και προσαρμόζονται οι υπόλοιπες κλάσεις ώστε να του παρασχεθεί όσο το δυνατόν καλύτερη μαθησιακή εμπειρία.
- *VisibilityFunctions*: Μέθοδοι που φροντίζουν για την εμφάνιση/απόκρυψη των φορμών – παιδιών μέσα στην κύρια φόρμα.

➤ Ρολόι

Το αναλογικό ρολόι (project AnalogClockControl) δημιουργήθηκε σαν custom control σε visual basic. Λαμβάνει παραμετρικά εισόδους για τις τιμές του μικρού και του μεγάλου δείκτη εμφανίζοντας την ανάλογη ώρα. Το output αυτού του project είναι ένα αρχείο δυναμικής βιβλιοθήκης (.dll), το οποίο μπορεί να χρησιμοποιηθεί από οποιαδήποτε γλώσσα προγραμματισμού του .NET Framework. Χρησιμοποιείται δηλαδή σαν ανεξάρτητο component, που ενσωματώνεται σε οποιοδήποτε project. Με την προσθήκη ενός timer μπορεί να γίνει δυναμικό και να λειτουργεί σαν κανονικό ρολόι, ενώ είναι πλήρως προσαρμόσιμο σε μέγεθος και χρώματα. Πρωταρχικός στόχος αποτελεί η αναγνωσιμότητα και η ευχρηστία του ρολογιού και επιλέχθηκε η κατάλληλη σχεδίαση, ώστε να είναι ευδιάκριτες οι λεπτομέρειες σε επίπεδο λεπτού, ακόμα και στα μικρά μεγέθη ρολογιών.

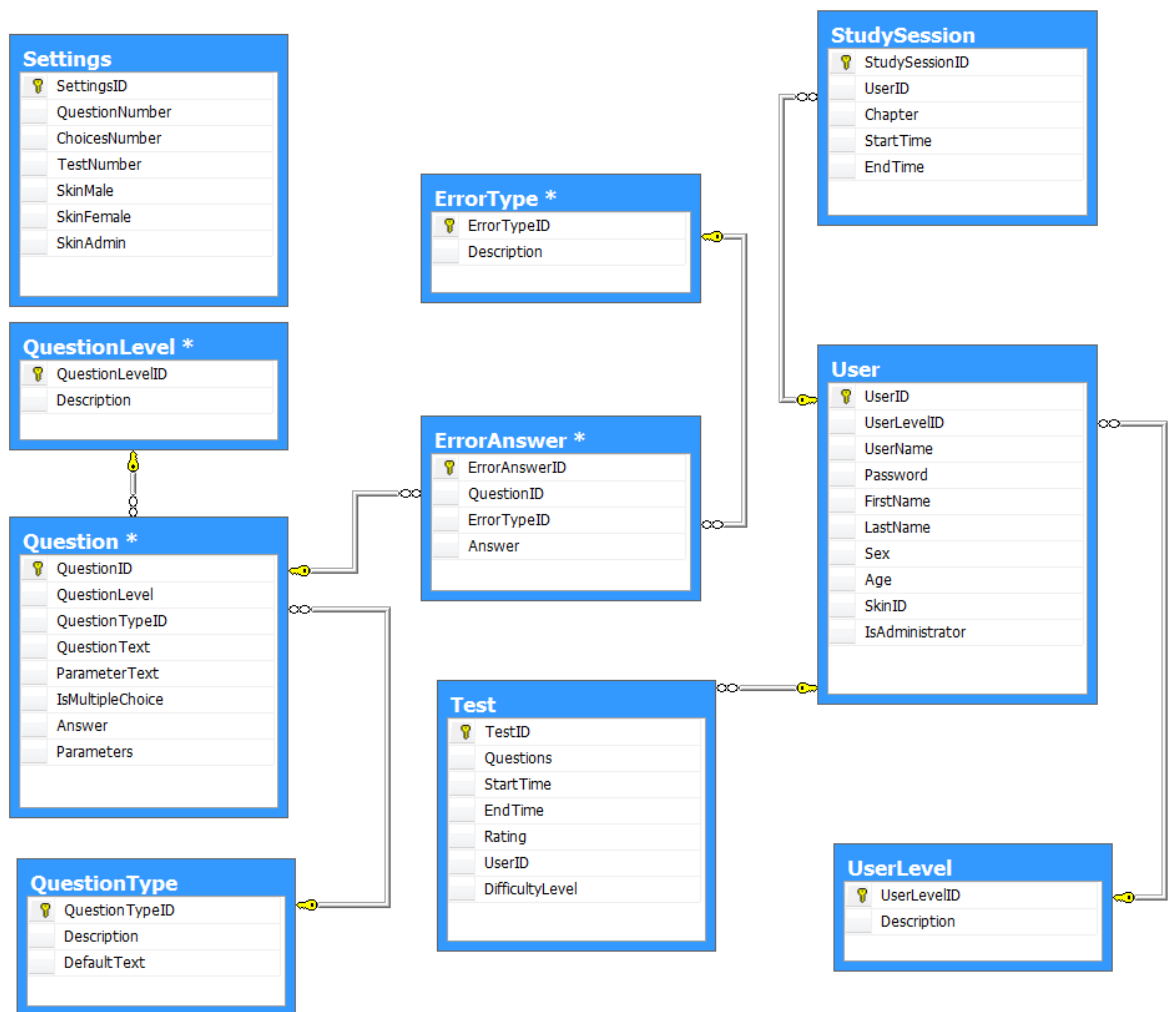


Εικόνα 35: Custom Control και παράμετροι – ιδιότητες για το αναλογικό ρολόι.

Οι δείκτες του ρολογιού σχεδιάζονται σε πραγματικό χρόνο, πάνω σε ένα σύστημα πολικών συντεταγμένων, μέσω κυκλικών επιμέρους μετατοπίσεων και σχεδίασης γραμμών και αριθμών στις ανάλογες θέσεις. Με απλούστερα λόγια, διαχωρίζουμε τον κύκλο σε 60 ίσα μέρη, και μετρώντας τις μοίρες, σχεδιάζουμε μικρές και μεγαλύτερες γραμμές, ανάλογα με το αν βρισκόμαστε σε θέση λεπτού ή πενταλέπτου, ενώ όποτε χρειάζεται η εκτύπωση αριθμού, δίνουμε εντολή σχεδίασης συμβολοσειράς πάνω στο ρολόι και τη μετατοπίζουμε ελαφρώς, ώστε να βρεθεί στην ανάλογη θέση.

7.1.5 Βάση Δεδομένων - Πίνακες και συσχετίσεις

Η βάση δεδομένων της εφαρμογής αποτελεί πολύ βασικό στοιχείο της λειτουργικότητάς της, επιτρέποντας την ανάκτηση ερωτήσεων, την ανάλυση στατιστικών στοιχείων και τη μοντελοποίηση των χρηστών. Στο διάγραμμα φαίνεται το λογικό διάγραμμα της βάσης, που παρέχει τις συσχετίσεις μεταξύ των πινάκων αυτής.



Εικόνα 36: Λογικό σχήμα της βάσης δεδομένων που κατασκευάστηκε.

Οι πίνακες που δημιουργήθηκαν είναι οι εξής:

- **User**: Διατηρεί στοιχεία πρόσβασης που παρέχονται από το registration ή από την καταχώρηση του χρήστη. Από αυτά, η ηλικία χρησιμοποιείται για να τον κατατάξει βάσει στερεοτύπου στο κατάλληλο επίπεδο, το φύλο και το όνομα του χρήστη χρησιμοποιούνται στη μοντελοποίηση (βλ. ανάλογα εδάφια).

	UserID	UserLevelID	UserName	Password	FirstName	LastName	Sex	Age	SkinID	IsAdministrator
▶	1	2	1	1	Τάσος	Ζούργης	M	6	1	False
	2	0	admin	admin	Administrator	Admin	A	99	3	True
	3	3	2	2	Christine	G	F	7	2	False
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Εικόνα 37: Στιγμιότυπο του πίνακα User.

- **UserLevel**: Περιέχει τα επίπεδα στα οποία κατατάσσεται ο χρήστης μετά την εγγραφή του. Υπάρχει αντιστοιχία στα επίπεδα τόσο με τις ερωτήσεις όσο και με τη θεωρία. Μόνη διαφοροποίηση είναι η εγγραφή «Χωρίς Επίπεδο» που αναθέτει στο πρόγραμμα την κατάταξη του χρήστη στο πιο κατάλληλο επίπεδο βάσει της μοντελοποίησης. Είναι επίσης το επίπεδο κάθε χρήστη-διαχειριστή.

	UserLevelID	Description
▶	0	Χωρίς Επίπεδο
	1	Πολύ Εύκολο
	2	Εύκολο
	3	Μέτρια Δυσκολία
	4	Δύσκολο
*	NULL	NULL

Εικόνα 38: Στιγμιότυπο του πίνακα UserLevel

- **Question:** Διατηρεί στοιχεία για την ερώτηση, που αφορούν το επίπεδό της, τον τύπο της, το κείμενο που θα εμφανίζεται σαν εκφώνηση (εφόσον αυτό διαφέρει από το προεπιλεγμένο που έχει οριστεί για τον κάθε τύπο ερώτησης), το παραμετροποιημένο κείμενο που ενδέχεται να εμφανίζεται στην εκφώνηση και να διαφέρει από ερώτηση σε ερώτηση, αν είναι multiple choice ερώτηση, ποια είναι η σωστή απάντηση, και άλλες παραμέτρους που αφορούν στην οπτικοποίηση της ερώτησης στην οθόνη.

	QuestionID	QuestionLevel	QuestionTypeID	QuestionText	ParameterText	IsMultipleChoice	Answer	Parameters
▶	1	1	1		μία η ώρα	True	01:00	
	2	1	1		σις έξι	True	06:00	
	3	1	1		11 η ώρα	True	11:00	
	4	1	1		σις τρεις η ώρα	True	03:00	
	5	1	2		08:00	True	οκτώ η ώρα	
	6	1	2		10:00	True	δέκα η ώρα	
	7	1	2		02:00	True	2:00	
	8	1	2		05:00	True	πέντε η ώρα	
	9	1	3		εφτά η ώρα	False	07:00	
	10	1	3		τέσσερις ακριβώς	False	04:00	
	11	1	3		εννιά η ώρα	False	01:00	
	12	1	3		έντακα	False	11:00	
	13	1	4	Το ΔΕΞΙ ρολόι σ...		False	02:00	2-12:00-02:00
	14	1	4	Το ΔΕΞΙ ρολόι σ...		False	03:00	2-11:00-02:00
	15	1	4	Το ΔΕΞΙ ρολόι σ...		False	04:00	2-04:00-08:00
	16	1	4	Το ΔΕΞΙ ρολόι σ...		False	01:00	2-07:00-08:00
	17	1	5		Τα δύο ρολόγια ...	False	true	2-07:00:1-07:00:1
	18	1	5		Τα δύο ρολόγια ...	False	false	2-01:00:3-12:05:3
	19	1	5		Το ρολόι δείχνει ...	False	false	1-02:00
	20	1	5		Το ρολόι δεν δεί...	False	true	1-03:00

Εικόνα 39: Στιγμιότυπο του πίνακα Question.

- **QuestionType:** Περιέχει πληροφορίες για τους διαφορετικούς τύπους ερωτήσεων, όπως η επεξήγηση για τον κάθε τύπο ερώτησης και το προεπιλεγμένο κείμενο που εμφανίζεται στην εκφώνηση εφόσον δεν έχει καθοριστεί κάτι διαφορετικό.

	QuestionTypeID	Description	DefaultText
▶	1	Multiple Choice - Ρολόγια ως επιλογές (Επιλογές Εικόνας).	Ποιο ρολόι δείχνει {0};
	2	Multiple Choice - Κουμπιά ως επιλογές (Επιλογές Κειμένου).	Τι ώρα δείχνει το παρακάτω ρολόι;
	3	Κατασκευαστική - Ρύθμιση ρολογιού σε καθορισμένη ώρα.	Τοποθετήστε τους δείκτες του ρολογιού στις {0}.
	4	Κατασκευαστική - Ρύθμιση χρονικού εύρους.	Πόσο χρονικό διάστημα {0};
	5	Σωστό ή Λάθος - Αξιολογείται η δεδομένη πρόταση.	{0}. Σωστό ή Λάθος;
	6	Ανοιχτού Τύπου - Απαιτείται πληκτρολόγηση της απάντησης.	
*	NULL	NULL	NULL

Εικόνα 40: Στιγμιότυπο του πίνακα QuestionType.

- **QuestionLevel:** Περιέχει τα επίπεδα στα οποία κατηγοριοποιούνται οι ερωτήσεις βάση της μοντελοποίησης. Κάθε επίπεδο ανταποκρίνεται σε διαφορετικό σε επίπεδο δυσκολίας γνωστικό αντικείμενο. Η στήλη της περιγραφής δεν χρησιμοποιείται από το πρόγραμμα, αλλά για καλύτερη κατανόηση της αρχιτεκτονικής δεδομένων του προγράμματος.

	QuestionLevelID	Description
	1	Το χαμηλότερο επίπεδο δυσκολίας. Ανταποκρίνεται κυρίως σε αρχάριους και παιδιά πολύ μικρής ηλικίας.
	2	Το δεύτερο επίπεδο δυσκολίας. Εξοικειώνει το χρήστη με τις βασικές υποδιαιρέσεις της ώρας.
	3	Το τρίτο επίπεδο δυσκολίας. Με περισσότερες υποδιαιρέσεις της ώρας και εισαγωγή στις χρονικές ζώνες.
▶	4	Το μεγαλύτερο επίπεδο δυσκολίας. Σύνθετες ερωτήσεις με μεγάλο εύρος.
*	NULL	NULL

Εικόνα 41: Στιγμιότυπο του πίνακα QuestionLevel.

- **ErrorAnswer:** Διατηρεί μοντελοποιημένα σφάλματα για τους διαφορετικούς τύπους σφαλμάτων και για κάθε ερώτηση. Επιλέγονται τυχαία από το πρόγραμμα όταν επιλέγεται δυναμικά μία ερώτηση προς εξέταση.

	ErrorAnswerID	QuestionID	ErrorTypeID	Answer
▶	1	1	2	12:05
	2	1	3	11:00
	3	1	5	01:05
	4	1	6	02:00
	5	1	4	01:30
	6	2	2	12:30
	7	2	5	06:30
	8	2	6	01:00
	9	2	4	06:05
	10	2	1	06:15
	11	3	2	11:55
	12	3	3	01:00
	13	3	4	11:30
	14	3	5	10:55
	15	3	6	02:00
	16	4	2	12:15
	17	4	3	09:00
	18	4	4	03:30
	19	4	5	03:15
	20	4	6	12:30
	21	5	2	δώδεκα παρά είκ...
	22	5	3	τέσσερις η ώρα
	23	5	5	οκτώ και δώδεκ...
	24	5	6	εννιά η ώρα
	25	5	4	επτά η ώρα

Εικόνα 42: Στιγμιότυπο του πίνακα ErrorAnswer.

- **ErrorType:** Περιέχει τους μοντελοποιημένους τύπους λαθών στα οποία μπορεί να υποπέσει ο χρήστης κατά τη διάρκεια ενός διαγωνίσματος.

	ErrorTypeID	Description
	1	Γενικά Λάθη - Κατανόησης
	2	Σύγχυση μικρού - μεγάλου δείκτη
	3	Σύγχυση μεταξύ εννοιών "παρά" - "και"
	4	Λάθη που αφορούν διάρκεια
	5	Ανάγνωση του λεπτοδείκτη
	6	Ανάγνωση του ωροδείκτη
	7	Ορθογραφικά Λάθη
▶*	NULL	NULL

Εικόνα 43: Στιγμιότυπο του πίνακα ErrorType.

- **Test:** Στον πίνακα Test αποθηκεύεται η πληροφορία για κάθε διαγώνισμα που ολοκλήρωσε ένας χρήστης. Πέρα από την ώρα έναρξης και λήξης, τη βαθμολογία που απέσπασε στο διαγώνισμα, τον κωδικό του υπο εξέταση χρήστη και το επίπεδο δυσκολίας του διαγωνίσματος, υπάρχει η στήλη Questions από την οποία εξάγεται σε συνδυασμό με τους άλλους πίνακες το μεγαλύτερο μέρος των στατιστικών του προγράμματος. Είναι μία στήλη xml που για κάθε ερώτηση του διαγωνίσματος έχει πληροφορία για το αν είναι σωστή. Αν δεν είναι σε τι τύπο σφάλματος υπέπεσε ο χρήστης και πόση ώρα τον απασχόλησε.

TestID	Questions	StartTime	EndTime	Rating	UserID	DifficultyLevel
24	<questions><question id="3" correct="True" errortype="0" start="2010-11-17 19:40:09.4009" end="2010-11-17 19:40:11.4011" dur...	2010-11-17 19:...	2010-11-17 19:...	50	1	1
25	<questions><question id="10" correct="False" errortype="1" start="2010-11-18 22:26:21.2621" end="2010-11-18 22:26:23.2623" d...	2010-11-18 22:...	2010-11-18 22:...	60	1	1
26	<questions><question id="6" correct="False" errortype="1" start="2010-11-30 20:03:59.0359" end="2010-11-30 20:04:03.0403" du...	2010-11-30 20:...	2010-11-30 20:...	10	1	1
27	<questions><question id="1" correct="False" errortype="1" start="2010-11-30 20:04:57.0457" end="2010-11-30 20:04:59.0459" du...	2010-11-30 20:...	2010-11-30 20:...	20	1	1
28	<questions><question id="2" correct="False" errortype="1" start="2010-11-30 20:05:46.0546" end="2010-11-30 20:05:48.0548" du...	2010-11-30 20:...	2010-11-30 20:...	10	1	1
29	<questions><question id="4" correct="True" errortype="0" start="2010-12-02 20:25:17.2517" end="2010-12-02 20:25:38.2538" dur...	2010-12-02 20:...	2010-12-02 20:...	100	1	1
30	<questions><question id="2" correct="False" errortype="2" start="2010-12-02 20:26:37.2637" end="2010-12-02 20:26:41.2641" du...	2010-12-02 20:...	2010-12-02 20:...	0	1	1
31	<questions><question id="8" correct="False" errortype="3" start="2010-12-03 20:35:51.3551" end="2010-12-03 20:35:55.3555" du...	2010-12-03 20:...	2010-12-03 20:...	40	3	1
32	<questions><question id="10" correct="True" errortype="0" start="2010-12-03 20:37:38.3738" end="2010-12-03 20:37:43.3743" d...	2010-12-03 20:...	2010-12-03 20:...	100	3	1
33	<questions><question id="11" correct="False" errortype="1" start="2010-12-05 13:00:42.0042" end="2010-12-05 13:00:48.0048" d...	2010-12-05 13:...	2010-12-05 13:...	60	3	1
34	<questions><question id="12" correct="True" errortype="0" start="2010-12-05 13:03:49.0349" end="2010-12-05 13:03:53.0353" d...	2010-12-05 13:...	2010-12-05 13:...	100	3	1
35	<questions><question id="3" correct="False" errortype="1" start="2010-12-06 20:50:30.5030" end="2010-12-06 20:50:31.5031" du...	2010-12-06 20:...	2010-12-06 20:...	0	3	1

Εικόνα 44: Στιγμιότυπο του πίνακα Test.

- **StudySession:** Κάθε φορά που ο χρήστης μελετάει κάποιο κεφάλαιο της θεωρίας, καταγράφεται αυτή την πληροφορία στον πίνακα study session. Έτσι εξάγονται συμπεράσματα για το κατά πόσο αποτελεσματικά μελετάει, και σε τι βαθμό οι επιδόσεις στα διαγωνίσματα εξαρτώνται από τη θεωρία.

	StudySessionID	UserID	Chapter	StartTime	EndTime
▶	1	1	1	2010-11-15 20:...	2010-11-15 20:...
	2	1	1	2010-11-20 17:...	2010-11-20 17:...
	3	1	1	2010-11-20 17:...	2010-11-20 17:...
	4	1	1	2010-11-20 20:...	2010-11-20 20:...
	5	1	1	2010-11-20 20:...	2010-11-20 20:...
	6	2	1	2010-11-20 20:...	2010-11-20 20:...
	7	2	2	2010-11-20 20:...	2010-11-20 20:...
	8	1	1	2010-11-21 12:...	2010-11-21 12:...
	9	1	1	2010-11-30 18:...	2010-11-30 18:...
	10	1	2	2010-12-09 18:...	2010-12-09 18:...
	11	1	3	2010-12-09 18:...	2010-12-09 18:...
	12	1	4	2010-12-09 18:...	2010-12-09 18:...
	13	3	1	2010-12-18 19:...	2010-12-18 19:...
	14	1	1	2011-01-29 10:...	2011-01-29 10:...
	15	1	1	2011-01-29 10:...	2011-01-29 10:...
	16	1	4	2011-01-29 12:...	2011-01-29 12:...

Εικόνα 45: Στιγμιότυπο του πίνακα StudySession.

- **Settings:** Στον πίνακα Settings αποθηκεύονται από τον διαχειριστή κεντρικές ρυθμίσεις του προγράμματος όπως οι ερωτήσεις του κάθε διαγωνίσματος, ο αριθμός επιλογών στις απαντήσεις πολλαπλής επιλογής, ο αριθμός των διαγωνισμάτων βάσει των οποίων αξιολογείται ο χρήστης και συνεπώς κατατάσσεται στα διάφορα επίπεδα, και τέλος

κάποιες επιλογές που αφορούν στην εμφάνιση της εφαρμογής για κάθε ομάδα χρηστών.

	SettingsID	QuestionNumber	ChoicesNumber	TestNumber	SkinMale	SkinFemale	SkinAdmin
▶	1	10	5	5	5	33	24
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Εικόνα 46: Στιγμιότυπο του πίνακα Settings.

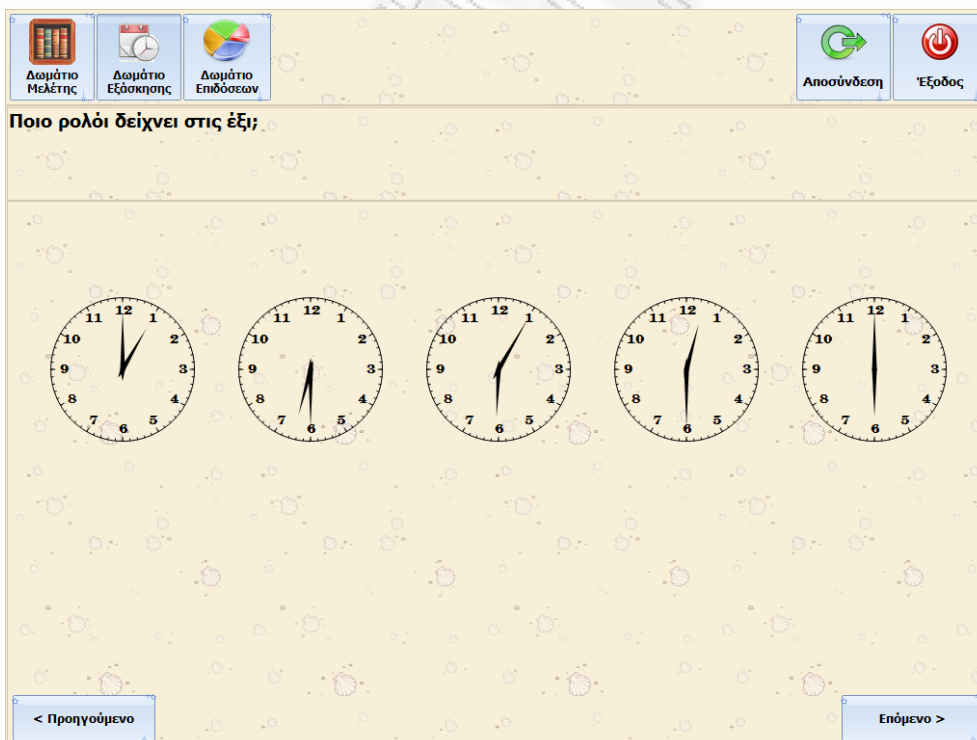
Είναι σημαντικό να ξεκαθαριστεί ότι οι ερωτήσεις, οι υποψήφιος απαντήσεις, οι κατηγορίες λαθών και οι τύποι ερωτήσεων, αν και κατασκευασμένα εκ των προτέρων δε δεσμεύονται από τη λειτουργία της εφαρμογής, καθώς με νέες προσθήκες στη βάση, μπορούν να φτάσουν οποιοδήποτε επιθυμητό αριθμό. Έτσι, η εφαρμογή αποτελεί μια διεπαφή πάνω σε μία βάση, η οποία μπορεί να είναι οσοδήποτε μεγάλη και πολύπλοκη επιθυμεί κανείς.

7.2 Μοντελοποίηση των χρηστών της εφαρμογής

Η μοντελοποίηση για το χρήστη ξεκινάει από την εγγραφή του στη βάση δεδομένων της εφαρμογής. Όταν ένας νέος μαθητής εγγράφεται στο αρχείο, γίνεται μία αξιολόγηση βάσει της ηλικίας του και της ενδεχόμενης προηγούμενης εμπειρίας του και κατατάσσεται κατά στερεότυπο αυτόματα σε ένα αρχικό επίπεδο. Για παράδειγμα, όταν έχει ηλικία μέχρι έξι χρονών, κατατάσσεται στο επίπεδο 1, όταν είναι πάνω από έξι χρονών στο επίπεδο 2 και εφόσον έχει προηγούμενη εμπειρία με την ώρα κερδίζει ακόμα ένα επίπεδο.

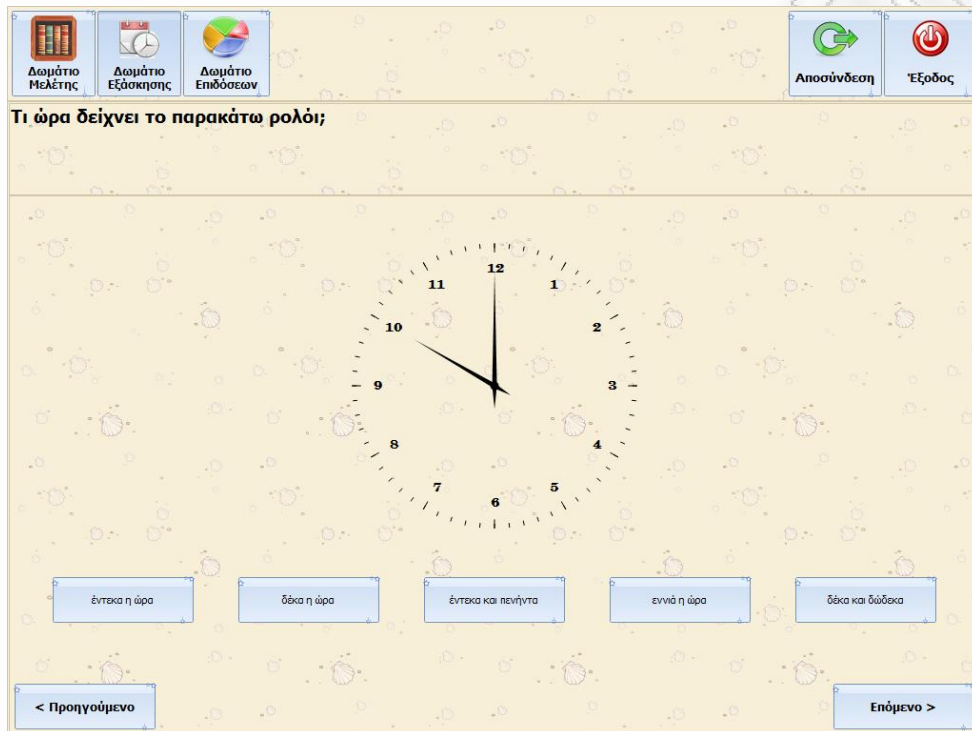
Οι ερωτήσεις που απευθύνονται στο χρήστη έχουν μοντελοποιηθεί σε έξι κατηγορίες, με κάθε μία από αυτές να προσπαθεί να αναπτύξει και να εξετάσει διαφορετικές δεξιότητες του χρήστη. Αναλυτικά οι κατηγορίες είναι:

- Ερωτήσεις πολλαπλής επιλογής με επιλογές εικόνες (ρολόγια): ο χρήστης πρέπει να είναι σε θέση να αναγνωρίσει όλες τις εικόνες και να διαλέξει αυτή που ανταποκρίνεται καλύτερα στο ζητούμενο της ερώτησης. Στόχο έχει να αναπτύξει την παρατηρητικότητα του χρήστη και τη δυνατότητα να οπτικοποιήσει αυτό που έχει μελετήσει.



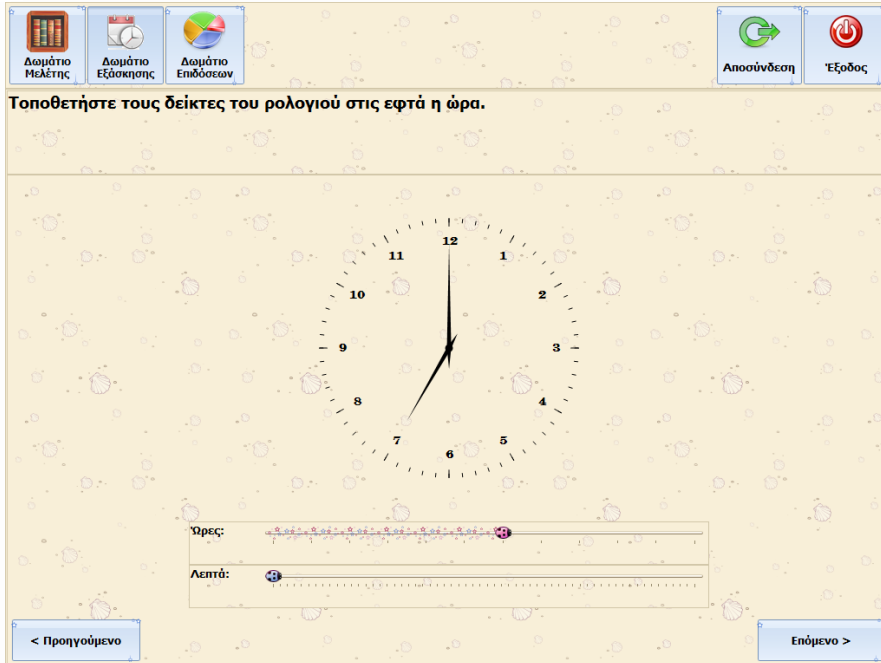
Εικόνα 47: Ερώτηση πολλαπλής επιλογής με επιλογές εικόνες.

- Ερωτήσεις πολλαπλής επιλογής με επιλογές κειμένου. Ο χρήστης βλέπει μια εικόνα και κάποιες επιλογές κειμένου και πρέπει να επιλέξει την επιλογή που περιγράφει αυτό που βλέπει. Σε αντίθεση ο χρήστης πρέπει να είναι σε θέση να μετατρέψει την εικόνα που βλέπει σε κείμενο και να το συγκρίνει με της απαντήσεις που βλέπει. Στόχος είναι να αναπτύξει την αναλυτική σκέψη.



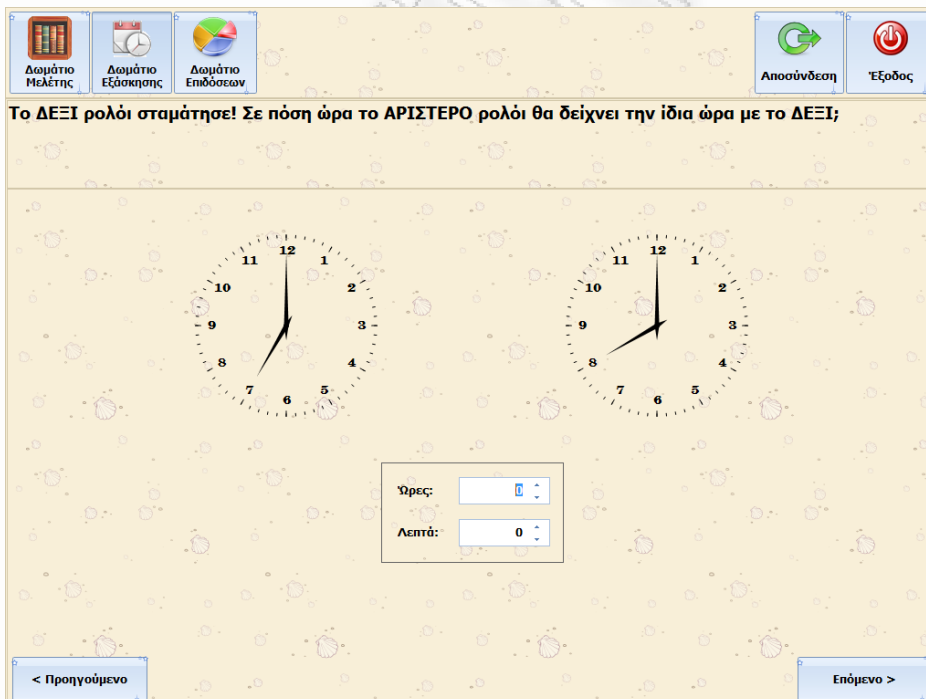
Εικόνα 48: Ερώτηση πολλαπλής επιλογής με επιλογές κειμένου.

- Ρύθμιση ρολογιού σε ζητούμενη ώρα. Είναι μία κατασκευαστική ερώτηση αφού ζητά από το χρήστη να ρυθμίσει το ρολόι σε συγκεκριμένη ώρα. Δίνει τη δυνατότητα στο χρήστη να πειραματιστεί με πιθανές απαντήσεις προτού δώσει την τελική του απάντηση και ταυτόχρονα να βλέπει τις ενδιαμέσες απαντήσεις.



Εικόνα 49: Κατασκευαστική ερώτηση - Ρύθμιση ρολογιού σε ζητούμενη ώρα.

- Εύρεση ζητούμενης διάρκειας. Και αυτή η κατηγορία θεωρείται κατασκευαστική καθώς απαιτεί συχνά από το χρήστη να κάνει κάποια ανάλυση προκειμένου να καταλήξει στη λύση. Βέβαια σε αντίθεση με την προηγούμενη κατηγορία, ο χρήστης δεν έχει κάποια ενδιαμέση εικόνα οπότε και συχνά χρειάζεται να αφιερώσει περισσότερο χρόνο. Στόχος είναι η κατανόηση των εννοιών που αφορούν χρονική διάρκεια και η ανάπτυξη της κριτικής σκέψης.



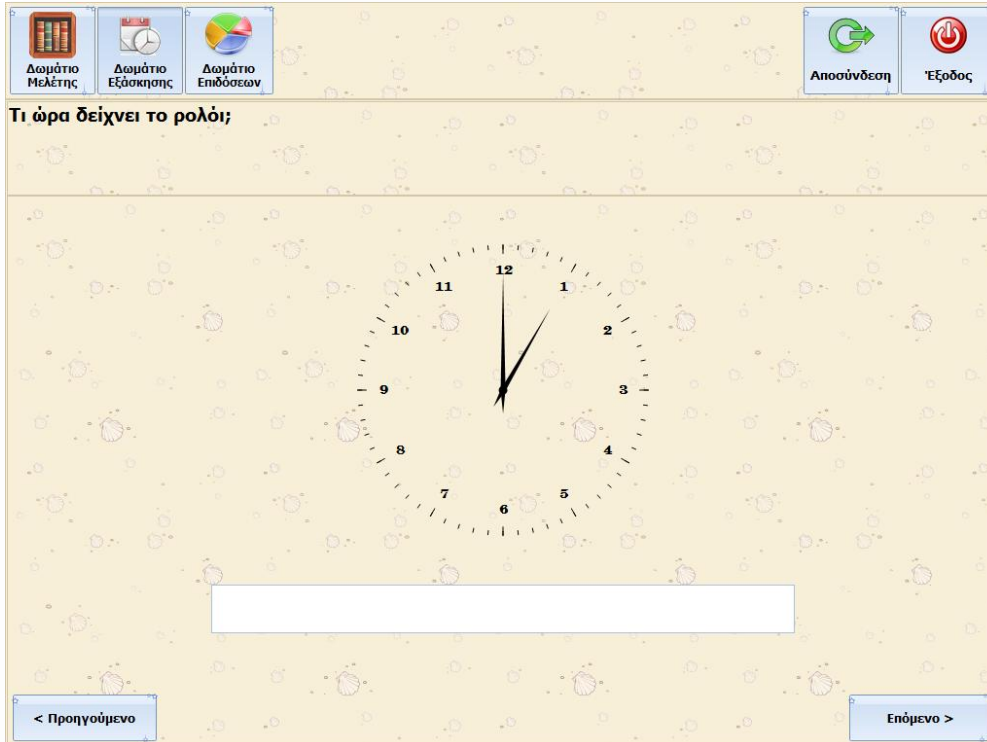
Εικόνα 50: Κατασκευαστική ερώτηση - Ρύθμιση χρονικής διάρκειας.

- Ερωτήσεις τύπου Σωστό/Λάθος. Αξιολογείται δεδομένη πρόταση. Ο χρήστης αναπτύσσει ικανότητες αξιολόγησης και μαθαίνει να εντοπίζει σημεία – παγίδες.



Εικόνα 51: Ερώτηση τύπου Σωστό/Λάθος.

- Ερωτήσεις ανοικτού τύπου. Ζητείται από το χρήστη να πληκτρολογήσει ολόκληρη την απάντηση. Καλλιεργείται η ανεξαρτησία του χρήστη και αποκτά εμπιστοσύνη στις γνώσεις που έχει αποκτήσει.



Εικόνα 52: Ερώτηση ανοικτού τύπου.

Παράλληλα, τα λάθη που μπορεί να πραγματοποιήσει ο χρήστης έχουν μοντελοποιηθεί σε επτά κατηγορίες, καθεμία από τις οποίες σχετίζεται με μία διαφορετική μαθησιακή δυσκολία. Οι κατηγορίες αυτές είναι οι εξής:

- Λάθη σύγχυσης μεταξύ των δεικτών του ρολογιού. Στα λάθη αυτά ο χρήστης αναγιγνώσκει το λεπτοδείκτη σαν ωροδείκτη και το αντίστροφο.



Εικόνα 53: Σε ένα τέτοιο λάθος, ο χρήστης θα διάβαζε στην παραπάνω εικόνα την ώρα 04:10 αντί 02:20.

- Λάθη που σχετίζονται με την κίνηση του ωροδείκτη. Στα λάθη αυτά ο χρήστης αγνοεί την παράλληλη κίνηση του ωροδείκτη, με αποτέλεσμα να δυσκολεύεται να κατανοήσει σε ποια ώρα αναφέρεται ο ωροδείκτης όταν βρίσκεται σε ένα σημείο μεταξύ δύο ωρών.



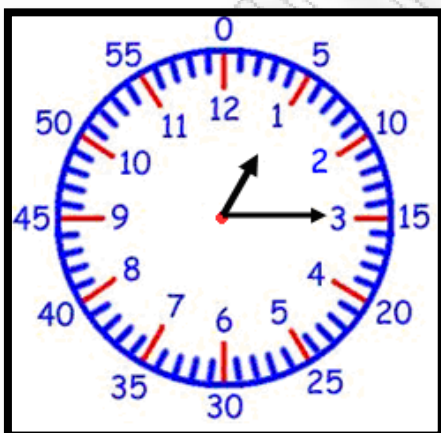
Εικόνα 54: Σε ένα τέτοιο λάθος, ο χρήστης θα διάβαζε στην παραπάνω εικόνα την ώρα 01:45 αντί 12:45.

- Λάθη σύγχυσης σχετικής με τη φορά του ρολογιού. Στα λάθη αυτά ο χρήστης συγχύζει τις ώρες «... και ...» με τις «... παρά ...».



Εικόνα 55: Σε ένα τέτοιο λάθος, ο χρήστης θα διάβαζε στην παραπάνω εικόνα την ώρα 07:37 αντί 07:23.

- Λάθη ανάγνωσης των διαβαθμίσεων που αφορούν τα λεπτά. Σε αυτό τον τύπο λαθών ο χρήστης διαβάζει τον αναγραφόμενο αριθμό του ρολογιού, που αφορά μέτρηση ωρών και τον χρησιμοποιεί για να μετρήσει λεπτά.



Εικόνα 56: Σε ένα τέτοιο λάθος, ο χρήστης θα διάβαζε στην παραπάνω εικόνα την ώρα 01:03 αντί 01:15.

- Λάθη που αφορούν χρονική διάρκεια. Ο χρήστης μπορεί να υποπέσει σε ένα τέτοιο λάθος είτε απαντώντας λάθος σε μία ερώτηση που αφορά διάρκεια και εφόσον το

λάθος δεν ανήκει σε κάποια από τις παραπάνω κατηγορίες, είτε απαντώντας σε μία οποιαδήποτε ερώτηση άλλου τύπου να γίνει εμφανές ότι συγχέει κάποια έννοια που αφορά διάρκεια. Για παράδειγμα αν σε μία κατασκευαστική ερώτηση που ζητάει από το χρήστη να θέσει την ώρα στις οκτώ και τέταρτο, αυτός τοποθετήσει τους δείκτες στις οκτώ και μισή αυτό είναι πιθανότατα σημάδι ότι ο χρήστης δεν μπορεί να διαχωρίσει το τέταρτο από το ημίωρο, έννοιες που αφορούν διάρκεια.

- Λάθη ορθογραφικά. Είναι λάθη που μπορούν να προκύψουν είτε σε ερωτήσεις ανοικτού τύπου όπου ο χρήστης δίνει μία απάντηση σωστά αλλά με λάθος ορθογραφία, είτε σε αλλου τύπου ερωτήσεις όπου ο χρήστης πιθανότατα γνώριζε την απάντηση αλλά από κεκτημένη ταχύτητα πιθανότατα έδωσε λάθος απάντηση. Για παράδειγμα σε μία κατασκευαστική ερώτηση που ζητά από το χρήστη να τοποθετήσει τους δείκτες του ρολογιού στις 8.10 εκείνος τους τοποθετεί στις 8.11, πράγμα που πιθανότατα σημαίνει ότι αυτό το λάθος είναι πιθανότερο να συνέβη όχι επειδή ο χρήστης δεν γνώριζε την απάντηση αλλά μάλλον επειδή βιάστηκε να απαντήσει.
- Λάθη γενικά ή λάθη κατανόησης. Τα λάθη αυτά, είναι εκείνα που δεν εμπίπτουν σε καμία από τις προηγούμενες κατηγορίες. Μπορεί να οφείλονται σε πολλούς και διαφορετικούς παράγοντες, όμως δεν έχουν προβλεφθεί από την τρέχουσα υλοποίηση. Γενικότερα είναι λάθη για τα οποία δεν είναι καθόλου προφανής ο λόγος που υπέπεσε σε αυτά ο χρήστης. Μοντελοποιούνται σε μία κατηγορία χάριν απλότητας, όμως θα μπορούσαν να είναι πολλές άλλες μικρές και πολύ σπανιότερες σε συχνότητα εμφάνισης κατηγορίες.

Κάθε παρεχόμενη ερώτηση λοιπόν, ενσωματώνει στις λανθασμένες απαντήσεις της κάποια από τα προαναφερθέντα είδη λαθών. Έτσι μπορούμε να γνωρίζουμε σε ποιους τομείς αντιμετωπίζει ο χρήστης δυσκολίες και με ποιους τομείς ο χρήστης είναι εξοικειωμένος. Στο τέλος κάθε διαγωνίσματος αλλά και όταν αποκτά πρόσβαση στην εφαρμογή σε μια επόμενη φορά, ο μαθητής μπορεί να δει συμβουλές για την απόδοσή του και για τμήματα της ύλης που τον δυσκολεύουν.

7.3 Προσαρμοστικότητα

Η εφαρμογή παρέχει προσαρμοστικότητα στη λειτουργία της, η οποία επηρεάζεται από την επιτυχία του χρήστη στις τεθείσες ασκήσεις και την αλλαγή επιπέδων, που εισάγει νέα δυσκολία στις ζητούμενες εργασίες. Έτσι, ανάλογα με την απόδοση του χρήστη στα τεστ κάθε επιπέδου, γίνεται αξιολόγηση των πετυχημένων απαντήσεών του σε ένα σύνολο από όλες τις απαντήσεις. Εάν ο αριθμός των πετυχημένων απαντήσεων του χρήστη φτάσει ένα επιθυμητό ποσοστό επί του συνόλου, τότε εκείνος προάγεται σε επίπεδο και καλείται να αντιμετωπίσει αυξημένης δυσκολίας ασκήσεις. Αντίθετα, εάν παρουσιάσει εξαιρετικά χαμηλά σκορ, το πρόγραμμα υποβιβάζει το επίπεδό του, αναγκάζοντάς τον να επαναλάβει τη θεωρία και τα τεστ του προηγούμενου. Έτσι, εξερχόμαστε από τη συμπεριφορά που επιβάλλει το στερεότυπο της ηλικίας και προσαρμοζόμαστε στις μαθησιακές ανάγκες του συγκεκριμένου χρήστη. Για παράδειγμα, εάν ο χρήστης δηλώσει εσφαλμένη τιμή για την ηλικία του, η λειτουργικότητα αυτή επιτρέπει τη σύγκλιση σε κοντινότερης στις πραγματικές ικανότητές του δυσκολίας ασκήσεων.

Παράλληλα, ο μαθητής βλέπει στην οθόνη της απόδοσής του τα είδη των λαθών που έχει πραγματοποιήσει και συμβουλές πάνω στα λάθη αυτά, καθώς και σχόλια για την ως τώρα πρόοδό του. Τα λάθη που μπορεί να πραγματοποιήσει ένας μαθητής έχουν μοντελοποιηθεί ανάλογα με τις μαθησιακές δυσκολίες που ανιχνεύσαμε στη βιβλιογραφία και αναφέρονται αναλυτικά στο εδάφιο που αφορά τη μοντελοποίηση του χρήστη.

Δωμάτιο Μελέτης

Δωμάτιο Εξάσκησης

Δωμάτιο Επιδόσεων

Αποσύνδεση

Έξοδος

Τελευταίο Διαγώνισμα

Τόσο, η επίδοσή σου στο τελευταίο διαγώνισμα ήταν πολύ καλή. Αν και ο χρόνος που αφιέρωσες για μελέτη ήταν ελάχιστος, ο τρόπος που αντιμετώπισες το διαγώνισμα, έκρυψε καλά τις αδυναμίες σου. Συγκριτικά με το προηγούμενο διαγώνισμα, η βελτίωση είναι κάτι παραπάνω από αισθητή, και αυτό αντικατοπτρίζεται στα αποτελέσματα. Συνέχισε έτσι! Έκανες σχετικά λίγα λάθη, κυρίως σε ότι αφορά στον υπολογισμό διάρκειας, και σε δυσκόλεψαν οι ερωτήσεις που απαιτείται να ρυθμίσεις το ρολόι σε καθορισμένη ώρα.

70 %

Διαγωνίσματα	Πληροφορίες				Χρόνοι		
	Αριθμός Διαγωνίσματος	Επίπεδο Δυσκολίας	Αριθμός Ερωτήσεων	Βαθμολογία	Ώρα Έναρξης	Ώρα Λήξης	Διάρκεια
Ερωτήσεις	46	Μέτρια Δυσκολία	10	70%	18:36	18:38	00:01:58
Επίπεδα	43	Μέτρια Δυσκολία	10	0%	12:26	12:26	00:00:09
Χρόνοι	42	Εύκολο	10	60%	12:20	12:22	00:01:18
Χρόνοι	39	Πολύ Εύκολο	5	100%	17:13	17:14	00:00:37
Χρόνοι	38	Πολύ Εύκολο	15	20%	15:55	15:56	00:00:54
Χρόνοι	37	Πολύ Εύκολο	15	80%	15:44	15:46	00:01:55
Χρόνοι	30	Πολύ Εύκολο	5	0%	20:26	20:27	00:00:22
Χρόνοι	29	Πολύ Εύκολο	5	100%	20:25	20:26	00:00:56

Η Συμβουλή Μου

Πιθανότατα δεν έχεις κατανοήσει στον απαιτούμενο βαθμό κάποια πράγματα στη θεωρία του Κεφαλαίου 3. Μία καλή επανάληψη πιστεύω θα σε βοηθήσει πολύ. Τα αποτελέσματά σου θα ήταν ακόμα καλύτερα αν ήσουν πιο προσεκτικός και συγκεντρωμένος. Προσπάθησε στο μέλλον να αποφύγεις τα εύκολα λάθη.

Εικόνα 57: Οθόνη επιδόσεων και συμβουλών.

Ένα ακόμη στοιχείο προσαρμοστικότητας, είναι η προσαρμογή της διεπαφής ανάλογα με το φύλο και τις προτιμήσεις του μαθητή. Ενώ τόσο στη θεωρία όσο και στα διαγωνίσματα η εφαρμογή προτείνει στο χρήστη επιλογές που αντιστοιχούν στο επίπεδό του, ώστε να τον διευκολύνει να επιλέξει το ανάλογο υλικό μελέτης και εξέτασης.

Όσον αφορά στα παρεχόμενα διαγωνίσματα, η διεπαφή παρέχει προεπιλογή διαγωνίσματος στο επίπεδο του χρήστη, αφήνοντας ανοιχτή την επιλογή για όλα τα προηγούμενα επίπεδα, ενώ δε επιτρέπει την είσοδο του χρήστη σε διαγωνίσματα υψηλότερου επιπέδου από το δικό του. Ο λόγος που υλοποιήθηκε κάτι τέτοιο είναι η αποφυγή καταστάσεων κατά τις οποίες ένας χρήστης εισέρχεται σε διαγωνίσματα ακατάλληλης δυσκολίας, επιτυγχάνει πολύ χαμηλά σκορ και έπειτα δυσκολεύεται κατά την άνοδο επιπέδου.

Τέλος, και με σκοπό η χρήση της εφαρμογής να είναι για το χρήστη μία ευχάριστη εμπειρία, οι επιδόσεις και τα στατιστικά που παρέχονται στο χρήστη είναι απλά και ευχάριστα στην όψη ενώ κάποια από αυτά μπορεί να τα δει σαν παιχνίδι και να προσπαθήσει να βελτιώσει τις επιδόσεις τους. Για παράδειγμα ο χρήστης βλέπει στα ρεκόρ ότι έχει 4 συνεχόμενες σωστές απαντήσεις και κάνει ακόμα ένα τεστ για να ξεπεράσει την προηγούμενη επίδοσή του. Κάτι αντίστοιχο μπορεί να συμβεί και με τους χρόνους. Στις εικόνες που ακολουθούν φαίνονται όλες οι δυνατές οθόνες που αφορούν στις επιδόσεις και στα στατιστικά που εξάγονται και πώς αυτές προσαρμόζονται ώστε να αντανακλούν τις επιδόσεις του.

Δωμάτιο Μελέτης

Δωμάτιο Εξάσκησης

Δωμάτιο Επιδόσεων

Αποσύνδεση

Έξοδος

Τελευταίο Διαγώνισμα

Τόσο, η επίδοσή σου στο τελευταίο διαγώνισμα ήταν πολύ καλή. Αν και ο χρόνος που αφιέρωσες για μελέτη ήταν ελάχιστος, ο τρόπος που αντιμετώπισες το διαγώνισμα, έκρυψε καλά τις αδυναμίες σου. Συγκριτικά με το προηγούμενο διαγώνισμα, η βελτίωση είναι κάτι παραπάνω από αισθητή, και αυτό αντικατοπτρίζεται στα αποτελέσματα. Συνέχισε έτσι! Έκανες σχετικά λίγα λάθη, κυρίως σε ότι αφορά στον υπολογισμό διάρκειας, και σε δυσκόλεψαν οι ερωτήσεις που απαιτείται να ρυθμίσεις το ρολόι σε καθορισμένη ώρα.

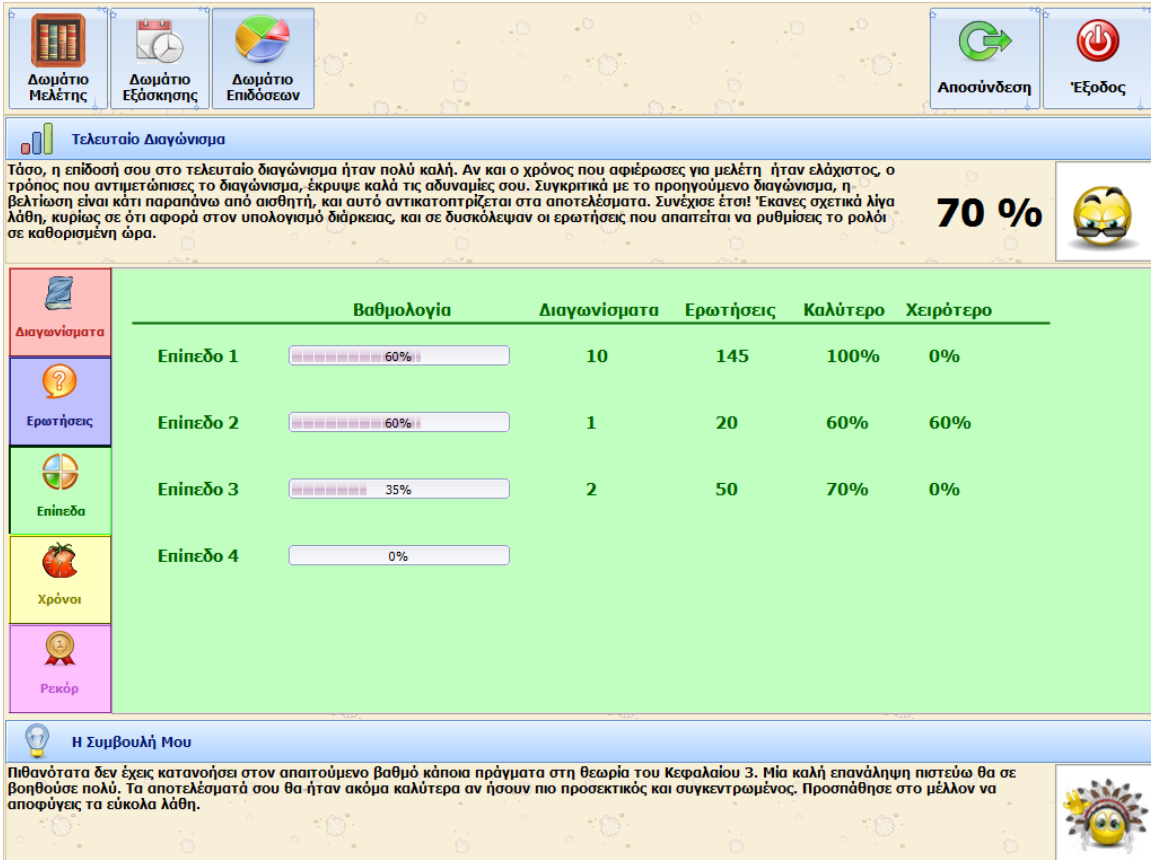
70 %

	Τύποι Ερωτήσεων	Τύποι Λαθών
	Πολλαπλής Επιλογής - Επιλογές Εικόνας 2 στις 4 (50%)	Γενικά Λάθη - Κατανόησης 7 από 13 (53%)
	Πολλαπλής Επιλογής - Επιλογές Κειμένου 1 στις 2 (50%)	Σύγχυση μικρού - μεγάλου δείκτη 0 από 13 (0%)
	Κατασκευαστική - Ρύθμιση Ώρας 0 στις 2 (0%)	Σύγχυση μεταξύ εννοιών "πρά" - "και" 0 από 13 (0%)
	Κατασκευαστική - Υπολογισμός Διάρκειας 2 στις 6 (33%)	Λάθη που αφορούν διάρκεια 5 από 13 (38%)
	Αξιολόγηση - Σωστό ή Λάθος 2 στις 6 (33%)	Ανάγνωση του λεπτοδείκτη 0 από 13 (0%)
	Ανοιχτές Ερωτήσεις	Ανάγνωση του ωροδείκτη 0 από 13 (0%)
		Ορθογραφικά Λάθη 1 από 13 (7%)

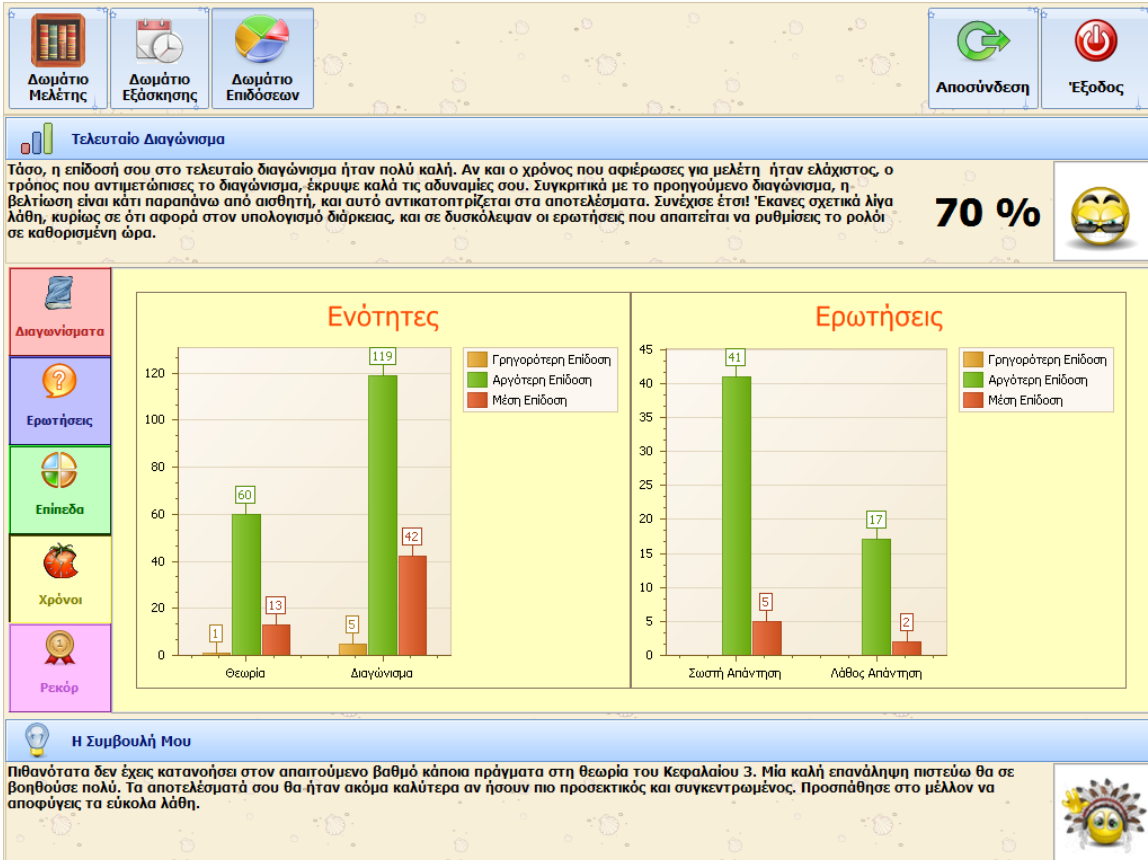
Η Συμβουλή Μου

Πιθανότατα δεν έχεις κατανόησει στον απαιτούμενο βαθμό κάποια πράγματα στη θεωρία του Κεφαλαίου 3. Μία καλή επανάληψη πιστεύω θα σε βοηθήσει πολύ. Τα αποτελέσματά σου θα ήταν ακόμα καλύτερα αν ήσουν πιο προσεκτικός και συγκεντρωμένος. Προσπάθησε στο μέλλον να αποφύγεις τα εύκολα λάθη.

Εικόνα 58: Οι επιδόσεις του μαθητή, ανά κατηγορία ερώτησης και ανα τύπο λάθους. Το πρόγραμμα επισημαίνει στο χρήστη τα δυνατά και τα αδύναμα σημεία του .



Εικόνα 59: Οι συγκεντρωτικές επιδόσεις του χρήστη ανά επίπεδο. Πόσα διαγωνίσματα έκανε, πόσες ερωτήσεις απάντησε, καλύτερη και χειρότερη επίδοση.



Εικόνα 60: Καλύτεροι, χειρότεροι και μέσοι χρόνοι για θεωρίες - διαγωνίσματα και σωστές - λάθος απαντήσεις.

The screenshot displays the 'TicTac' application interface. At the top, there are navigation buttons for 'Διαμάτιο Μελέτης', 'Διαμάτιο Εξάσκησης', 'Διαμάτιο Επιδόσεων', 'Αποσύνδεση', and 'Έξοδος'. Below this is a 'Τελευταίο Διαγώνισμα' (Last Exam) section with a score of 70% and a smiley face icon. The main area features a sidebar with categories: 'Διαγωνίσματα', 'Ερωτήσεις', 'Επίπεδα', 'Χρόνοι', and 'Ρεκόρ'. The 'Ρεκόρ' (Record) section is active, showing a 'Συνεχόμενες Σωστές Απαντήσεις' (Consecutive Correct Answers) window with a star icon and '7 ερωτήσεις' (7 questions). A 'Record 1 of 4' indicator is visible at the bottom of this window. Below the main area is a 'Η Συμβουλή Μου' (My Advice) section with a lightbulb icon and a paragraph of text, accompanied by a sad smiley face icon.

Εικόνα 61: Ρεκόρ του μαθητή (θετικά και αρνητικά) σε κάποιες συγκεκριμένες κατηγορίες

8 Συμπεράσματα – εφαρμογές

8.1 Συμπεράσματα

Συμπερασματικά η εφαρμογή TicTac συνδυάζει πλεονεκτήματα από τις υπάρχουσες εκπαιδευτικές εφαρμογές, όπως η απλότητα, η φιλική διεπαφή και η προσαρμοστικότητα της διεπαφής στο χρήστη. Ωστόσο εισάγει και πολλά καινοτόμα χαρακτηριστικά και λειτουργίες που την διαφοροποιούν ουσιαστικά από αυτές τις εφαρμογές και την καθιστούν μία ολοκληρωμένη λύση λογισμικού στην εκμάθηση της ώρας. Η δυναμικότητα της εφαρμογής κατά δημιουργία ερωτήσεων και διαγωνισμάτων συνδυάζεται με ένα πλήρως μοντελοποιημένο περιβάλλον στο οποίο ο χρήστης μπορεί να περιηγηθεί χωρίς δυσκολίες. Η σχηματική και πολυδιάστατη απεικόνιση των στατιστικών πληροφορεί το χρήστη με ακρίβεια για τις γνώσεις και τις αδυναμίες του. Η συγκρότηση των διαγωνισμάτων έχει υλοποιηθεί με τρόπο τέτοιο ώστε να ευνοείται η κατανόηση των ερωτήσεων και όχι η απομνημόνευσή τους. Ταυτόχρονα η ανάπτυξη έχει γίνει προσεκτικά και με γνώμονα την επεκτασιμότητα του λογισμικού.

8.2 Επεκτάσεις

Το πρόγραμμα έχει σχεδιαστεί ώστε να επιτρέπει ένα σύνολο επεκτάσεων στη λειτουργικότητά του. Οι περισσότερες χαρακτηριστικές είναι οι εξής:

- Επέκταση της λειτουργίας της εφαρμογής και σε ψηφιακό ρολόι.
- Υλοποίηση ηλεκτρονικού βοηθού που να παρέχει οπτικοακουστικές συμβουλές κατά την περιήγηση του χρήστη.
- Κατάταξη των ερωτήσεων σε επίπεδα, όχι μόνο ανάλογα με το βαθμό δυσκολίας τους αλλά και με βάση τις απαντήσεις των χρηστών.
- Υλοποίηση διαχειριστικής διεπαφής για δημιουργία και μεταβολή ερωτήσεων και διαγωνισμάτων
- Καταγραφή στατιστικών σε επίπεδο συνεδρίας (session) και παροχή αντίστοιχου feedback προς τους χρήστες.
- Υλοποίηση διαδραστικού περιβάλλοντος μελέτης της θεωρίας, με επιπλέον ακουστικό υλικό και που θα επιδιώκει αλληλεπίδραση του χρήστη.

9 Βιβλιογραφία

1. Personalisation Systems: Multi-purpose User Modelling. Victor Manuel Garcia Barrios. VDM Verlag. 2008.
2. Basics of Adaptive Personalisation and User Modelling: A Complete Approach to Automatically Adapting Information Systems to their Users. Paul de Vrieze. 2009. VDM Verlag.
3. Προσαρμοστικά Εκπαιδευτικά Υπερμέσα: Αρχές και Υπηρεσίες, Δ. Πρέντζας, Ι. Χατζηλυγερούδης, Πανεπιστήμιο Πατρών, Πολυτεχνική Σχολή, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Ινστιτούτο Τεχνολογίας Υπολογιστών, δικτυακή πηγή, 2009
4. Papanikolaou, K. A., Magoulas, G. D., Grigoriadou, M. (2000). Computational Intelligence in Adaptive Educational Hypermedia. International Joint Conference on Neural Networks, 2000.
5. Leli V. Misirlaki S., Papaderos A., ,Adaptive Learning Environments & Educational Management, Academic Assignment, Web Source, 2009.
6. Harris S., It's about time: difficulties in developing time concepts, Web Source, 2009.
7. Brusilovsky, P. (1999) "Adaptive and Intelligent Technologies for Web-based Education", Special Issue on Intelligent Systems and Teleteaching, Künstliche Intelligenz, 4, pp19-25.
8. Brusilovsky, P. (2003). "Developing Adaptive Education Hypermedia Systems: From Design Models to Authoring Tools", in Murray, T., Blessing S., & Ainsworth, S. (eds.), Authoring Tools for Advanced Learning Technologies, Kluwer Academic Publishers, NL.

10 Παράρτημα

10.1 Κώδικας - frmLogin.cs

```

using System;
using System.Data;
using System.Drawing;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using Utils;

namespace TicTacApplication.Forms
{
    public partial class frmLogin : XtraForm
    {
        public string Username { get; set; }
        private int _skinId = 1;
        public frmLogin()
        {
            InitializeComponent();
            LookAndFeel.SkinName =
Utils.EnumExtensions.StringEnum.GetStringValue(Skins.Darkroom, 0);
        }

        #region Form Control Events
        private void txEdits_Enter(object sender, EventArgs e)
        {
            TextEdit activeEdt = sender as TextEdit;
            if (activeEdt.EditValue == null)
            {
                activeEdt.Tag = activeEdt.Properties.NullText;
                activeEdt.Text = string.Empty;
                activeEdt.Properties.NullText = string.Empty;
                activeEdt.Properties.Appearance.TextOptions.HAlignment =
DevExpress.Utils.HorzAlignment.Near;
                activeEdt.Properties.Appearance.ForeColor = Color.WhiteSmoke;
                Font f = activeEdt.Properties.Appearance.Font;
                activeEdt.Properties.Appearance.Font = new Font(f.FontFamily,
f.Size, FontStyle.Regular);
                if (activeEdt == txPassword)
                    activeEdt.Properties.PasswordChar = '*';
            }
        }

        private void txEdits_Leave(object sender, EventArgs e)
        {
            TextEdit activeEdt = sender as TextEdit;
            if (null == activeEdt.EditValue ||
DBNull.Value.Equals(activeEdt.EditValue) ||
string.IsNullOrEmpty(activeEdt.EditValue.ToString()))
            {
                activeEdt.EditValue = null;
                activeEdt.Properties.NullText = activeEdt.Tag.ToString();
                activeEdt.Properties.Appearance.TextOptions.HAlignment =
DevExpress.Utils.HorzAlignment.Center;
                activeEdt.Properties.Appearance.ForeColor = Color.DimGray;
                Font f = activeEdt.Properties.Appearance.Font;
            }
        }
    }
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        activeEdt.Properties.Appearance.Font = new Font(f.FontFamily,
f.Size, FontStyle.Italic);
        if (activeEdt == txPassword)
            activeEdt.Properties.PasswordChar = '\0';
    }
}

private void btnRegister_Click(object sender, EventArgs e)
{
    if (tabControl.SelectedTabPage != tpRegister)
        tabControl.SelectedTabPage = tpRegister;
    else
        DoRegister();
}

private void btnLogin_Click(object sender, EventArgs e)
{
    if (tabControl.SelectedTabPage != tpLogin)
        tabControl.SelectedTabPage = tpLogin;
    else
        DoLogin();
}

#endregion

#region Register - Login Functions
private void DoRegister()
{
    string sFirstName = txFirstName.Text;
    string sLastName = txLastName.Text;
    string sUserName = txRgUserName.Text;
    string sPassword = (txRgPassword.Text == txRepeatPassword.Text) ?
txRgPassword.Text : string.Empty;
    int iAge = (txAge.EditValue != null &&
!txAge.EditValue.Equals(DBNull.Value)) ? Convert.ToInt32(txAge.EditValue) : -1;
    char cSex = chkMale.Checked ? 'M' : 'F';

    if (!string.IsNullOrEmpty(sFirstName) &&
!string.IsNullOrEmpty(sLastName) && !string.IsNullOrEmpty(sUserName) &&
!string.IsNullOrEmpty(sPassword) && iAge != -1)
    {
        userBindingSource.AddNew();
        DataRowView drv = userBindingSource.Current as DataRowView;
        drv.BeginEdit();
        drv["FirstName"] = sFirstName;
        drv["LastName"] = sLastName;
        drv["UserName"] = sUserName;
        drv["Password"] = sPassword;
        drv["Age"] = iAge;
        drv["Sex"] = (iAge < 18) ? cSex : 'A';
        drv["SkinID"] = _skinId;
        drv["UserLevelID"] = GenerateUserLevel();
        drv["IsAdministrator"] = (iAge >= 18);
        drv.EndEdit();
        userTableAdapter.Update(ticTacDataSet.User);
        tabControl.SelectedTabPage = tpLogin;
    }
    else
    {

```

```

        //ErrorHandling
        XtraMessageBox.Show("Παρουσιάστηκε σφάλμα κατά την εγγραφή.");
    }
}

private void DoLogin()
{
    string sUsername = (txUsername.EditValue == null ||
txUsername.EditValue.Equals(DBNull.Value) ||
string.IsNullOrEmpty(txUsername.EditValue.ToString())) ? string.Empty :
txUsername.EditValue.ToString();
    string sPassword = (txPassword.EditValue == null ||
txPassword.EditValue.Equals(DBNull.Value) ||
string.IsNullOrEmpty(txPassword.EditValue.ToString())) ? string.Empty :
txPassword.EditValue.ToString();
    if (string.IsNullOrEmpty(sUsername) ||
string.IsNullOrEmpty(sPassword))
    {
        //ErrorHandling
        XtraMessageBox.Show("Εισάγετε στοιχεία εισόδου");
    }
    else
    {
        if (userTableAdapter.VerifyUser(sUsername, sPassword) > 0)
        {
            Username = sUsername;
            DialogResult = DialogResult.OK;
        }
        else
        {
            //ErrorHandling
            XtraMessageBox.Show("Λάθος στοιχεία εισόδου");
        }
    }
}
#endregion

private void pictureEdit1_Click(object sender, EventArgs e)
{
    foreach (Control c in grpFeedback.Controls)
    {
        if (c is PictureEdit)
            (c as PictureEdit).BorderStyle =
DevExpress.XtraEditors.Controls.BorderStyles.Default;
    }
    (sender as PictureEdit).BorderStyle =
DevExpress.XtraEditors.Controls.BorderStyles.Style3D;
    _skinId = Convert.ToInt32((sender as PictureEdit).Tag);
}

private int GenerateUserLevel()
{
    int iAge = (txAge.EditValue != null &&
!txAge.EditValue.Equals(DBNull.Value)) ? Convert.ToInt32(txAge.EditValue) : -1;
    int resval = 1;
    if (chkPreviousExperience.Checked)
        resval++;
    if (iAge >= 6)

```



```

        resval++;
        return resval;
    }
}
}

```

10.2 Κώδικας - frmMain.cs

```

using System;
using Utils;
using Utils.EnumExtensions;
using System.Windows.Forms;

namespace TicTacApplication.Forms
{
    public partial class frmMain : DevExpress.XtraEditors.XtraForm
    {
        public User User { get; set; }

        private bool _testComplete;
        private Settings _settings;
        public frmMain(User user)
        {
            InitializeComponent();
            _settings = new Settings();
            User = user;
            SetSkin();
            chkSettings.Visible = User.IsAdmin;

            //Show Welcome Form
            frmWelcome fWelcome = new frmWelcome(User);
            VisibilityFunctions.ClearPanel(pnlMain);
            VisibilityFunctions.ShowFormInPanel(fWelcome, pnlMain);
        }

        private void SetSkin()
        {
            Skins skin;
            if (User.Sex == 'M')
                skin = (Skins)_settings.MaleSkinID;
            else if (User.Sex == 'F')
                skin = (Skins)_settings.FemaleSkinID;
            else
                skin = (Skins)_settings.AdminSkinID;

            defaultLookAndFeel1.LookAndFeel.SetSkinStyle(StringEnum.GetStringValue(skin, 0));
        }

        public void SetSkin(int skinId)
        {
            defaultLookAndFeel1.LookAndFeel.SetSkinStyle(StringEnum.GetStringValue((Skins)skinId, 0));
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

    }

    private void checkButton_CheckedChanged(object sender, EventArgs e)
    {
        if ((pnlMain.Controls.Count > 0) && (pnlMain.Controls[0] is
frmStatisticsRoom))
        {
            if ((pnlMain.Controls[0] as frmStatisticsRoom).PerformLevelCheck)
                User.CheckLevelPromotion();
        }
        else if ((pnlMain.Controls.Count > 0) && (pnlMain.Controls[0] is
frmSettings))
        {
            _settings = new Settings();
            SetSkin();
        }
        if (chkStudy.Checked)
        {
            frmStudyRoom fStRoom = new frmStudyRoom
            {
                User = User
            };
            VisibilityFunctions.ClearPanel(pnlMain);
            VisibilityFunctions.ShowFormInPanel(fStRoom, pnlMain);
        }
        else if (chkPractice.Checked)
        {
            frmPracticeRoom fPracticeRoom = new frmPracticeRoom
            {
                User = User
            };
            fPracticeRoom.OnTestCompleted += fPracticeRoom_OnTestCompleted;
            VisibilityFunctions.ClearPanel(pnlMain);
            VisibilityFunctions.ShowFormInPanel(fPracticeRoom, pnlMain);
        }
        else if (chkStats.Checked)
        {
            frmStatisticsRoom fStatsRoom = new frmStatisticsRoom
            {
                User = User,
                PerformLevelCheck = _testComplete
            };
            _testComplete = false;
            VisibilityFunctions.ClearPanel(pnlMain);
            VisibilityFunctions.ShowFormInPanel(fStatsRoom, pnlMain);
        }
        else if (chkSettings.Checked)
        {
            frmSettings fSettings = new frmSettings
            {
                User = User
            };
            VisibilityFunctions.ClearPanel(pnlMain);
            VisibilityFunctions.ShowFormInPanel(fSettings, pnlMain);
        }
    }

    private void fPracticeRoom_OnTestCompleted(object sender, EventArgs e)

```

```

    {
        _testComplete = true;
        chkStats.Checked = true;
    }

    private void frmMain_FormClosing(object sender,
System.Windows.Forms.FormClosingEventArgs e)
    {
        User.CheckLevelPromotion();
    }

    private void btnLogout_Click(object sender, EventArgs e)
    {
        Application.Restart();
    }
}
}

```

10.3 Κώδικας - frmPracticeRoom.cs

```

using System;
using System.Drawing;
using DevExpress.XtraEditors;
using TicTacApplication.TicTacDataSetTableAdapters;
using Utils;

namespace TicTacApplication.Forms
{
    public partial class frmPracticeRoom : TicTacApplication.Forms.frmRoom
    {
        public event EventHandler OnTestCompleted;
        public User User { get; set; }
        public Test Test { get; set; }
        public TestVisualizer Visualizer { get; set; }

        private Settings Settings { get; set; }

        public frmPracticeRoom()
        {
            InitializeComponent();
            Settings = new Settings();
        }

        private void frmPracticeRoom_Shown(object sender, EventArgs e)
        {
            InitForm();
        }

        private void InitForm()
        {
            btnStartTest.Visible = true;
            lblQuestion.Text = "Για να ξεκινήσεις ένα διαγώνισμα πάτησε το κουμπί
δεξιά...";
        }

        private void StartTest(int level)
        {

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        Test = new Test(Settings.ChoicesNumber, Settings.QuestionNumber,
level, User.ID);
        Test.StartTime = DateTime.Now;
        Test.OnTestCompleted += Test_OnTestCompleted;
        Visualizer = new TestVisualizer(pnlAnswer, Test, lblQuestion);
        Test.GenerateQuestions();
        Visualizer.LoadQuestion();
    }

    void Test_OnTestCompleted(object sender, EventArgs e)
    {
        if (null != OnTestCompleted)
        {
            OnTestCompleted(this, e);
        }
    }

    private void btnStartTest_Click(object sender, EventArgs e)
    {
        btnStartTest.Visible = false;
        PromptLevelSelect();
    }

    private void PromptLevelSelect()
    {
        lblQuestion.Text = "Επίπεδο Διαγωνίσματος:";
        for (int i = 1; i <= User.Level; i++)
        {
            using (UserLevelTableAdapter userLevelTableAdapter = new
UserLevelTableAdapter())
            {
                string desc =
userLevelTableAdapter.GetLevelDescriptionByID(i);
                AddLevelButton(i, desc);
            }
        }
        AddLevelButton(-1, "Προτεινόμενο");
    }

    private void AddLevelButton(int level, string description)
    {
        int iButtonWidth = 300;
        int iButtonHeight = 50;

        int iFreeWidth = pnlAnswer.Width - iButtonWidth;
        if (iFreeWidth < 0) return;

        int iXMargin = (int)((float)iFreeWidth / (float)2);
        int iYMargin = (level == 1) ? 50 : (level == -1) ? 50 + (User.Level *
(iButtonHeight + 20)) : 50 + ((level - 1) * (iButtonHeight + 20));

        CheckButton button = new CheckButton { Size = new Size(iButtonWidth,
iButtonHeight), Location = new Point(iXMargin, iYMargin) };
        button.LookAndFeel.UseDefaultLookAndFeel = true;
        if (level == -1)
        {
            button.Image = Properties.Resources.star;
            button.ImageLocation = ImageLocation.MiddleLeft;
        }
    }

```

```

        Font f = button.Appearance.Font;
        button.Appearance.Font = new Font(f.FontFamily, 10F, FontStyle.Bold);
        button.AllowFocus = false;
        button.Text = description;
        button.Tag = (level != -1) ? level : User.Level;
        button.Click += button_Click;
        pnlAnswer.Controls.Add(button);
        button.BringToFront();
    }

    void button_Click(object sender, EventArgs e)
    {
        int lvl = Convert.ToInt32((sender as SimpleButton).Tag);
        Utils.VisibilityFunctions.ClearPanel(pnlAnswer);
        StartTest(lvl);
    }
}
}
}

```

10.4 Κώδικας - frmSettings.cs

```

using System;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using Utils;
using System.Data;
using Utils.EnumExtensions;

namespace TicTacApplication.Forms
{
    public partial class frmSettings : TicTacApplication.Forms.frmRoom
    {
        public User User { get; set; }

        public frmSettings()
        {
            InitializeComponent();
        }

        private void frmSettings_Load(object sender, EventArgs e)
        {
            DataTable dtSkins = SkinDataSource();
            lookUpEditSkinMale.Properties.DataSource = dtSkins;
            lookUpEditSkinFemale.Properties.DataSource = dtSkins;
            lookUpEditSkinAdmin.Properties.DataSource = dtSkins;
            settingsTableAdapter.Fill(ticTacDataSet.Settings);
            lookUpEditSkinAdmin.EditValueChanged +=
aLookUpEditSkin_EditValueChanged;
            lookUpEditSkinMale.EditValueChanged +=
aLookUpEditSkin_EditValueChanged;
            lookUpEditSkinFemale.EditValueChanged +=
aLookUpEditSkin_EditValueChanged;
        }
    }
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον


```

private void btnSave_Click(object sender, EventArgs e)
{
    settingsBindingSource.EndEdit();
    settingsTableAdapter.Update(ticTacDataSet.Settings);
    SavePassword();
}

private void SavePassword()
{
    txCurrentPassword.DoValidate();
    txNewPassword.DoValidate();
    txConfirmPassword.DoValidate();
    if (txCurrentPassword.Text.Trim() == string.Empty ||
        txNewPassword.Text.Trim() == string.Empty ||
        txConfirmPassword.Text.Trim() == string.Empty)
        return;
    if (txCurrentPassword.EditValue.Equals(User.Password))
    {
        if (txNewPassword.EditValue.Equals(txConfirmPassword.EditValue)
&& txNewPassword.EditValue != null)
        {
            User.Password = txNewPassword.EditValue.ToString();
            User.SaveUserData();
        }
        else
        {
            XtraMessageBox.Show("Η επαλήθευση απέτυχε, ο νέος κωδικός δεν
αποθηκεύτηκε.", "Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else if (txCurrentPassword.EditValue != null)
    {
        XtraMessageBox.Show("Ο τρέχων κωδικός ήταν λάθος, ο νέος κωδικός
δεν αποθηκεύτηκε.", "Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void rgChoicesNo_SelectedIndexChanged(object sender, EventArgs e)
{
    (sender as RadioGroup).DoValidate();
}

private void spnTestNo_EditValueChanged(object sender, EventArgs e)
{
    (sender as SpinEdit).DoValidate();
}

private static DataTable SkinDataSource()
{
    DataTable resval = new DataTable();
    resval.Columns.Add("ID", typeof(int));
    resval.Columns.Add("Description", typeof(string));

    for (int i = 0; i < 35; i++)
    {
        DataRow row = resval.NewRow();
        row["ID"] = i;
        row["Description"] = StringEnum.GetStringValue((Skins)i, 0);
    }
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        resval.Rows.Add(row);
    }
    return resval;
}

private void aLookUpEditSkin_EditValueChanged(object sender, EventArgs e)
{
    if((sender as LookUpEdit).EditValue != DBNull.Value)
    {
        int skinId = (int)((sender as LookUpEdit).EditValue);
        (Parent.Parent as frmMain).SetSkin(skinId);
        (sender as LookUpEdit).DoValidate();
    }
}
}
}
}

```

10.5 Κώδικας - frmStatisticsRoom.cs

```

using System;
using System.Data;
using System.Drawing;
using System.Windows.Forms;
using System.Resources;
using DevExpress.XtraEditors;
using TicTacApplication.TicTacDataSetTableAdapters;
using Utils;
using Utils.EnumExtensions;

namespace TicTacApplication.Forms
{
    public partial class frmStatisticsRoom : TicTacApplication.Forms.frmRoom
    {
        public User User { get; set; }

        private string _lastTestScore;
        public string LastTestScore
        {
            get { return _lastTestScore; }
            set
            {
                _lastTestScore = value;
                lblScore.Text = _lastTestScore + " % ";
            }
        }

        public bool PerformLevelCheck { get; set; }

        private StatisticsTableAdapter _StatisticsTableAdapter = new
        StatisticsTableAdapter();

        public frmStatisticsRoom()
        {
            InitializeComponent();
        }
    }
}

```

```
RecordsGridControl.LookAndFeel.SetSkinStyle(StringEnum.GetStringValue(Skins.Valentine, 0));
```

```
TestGridControl.LookAndFeel.SetSkinStyle(StringEnum.GetStringValue(Skins.DevExpressStyle, 0));
}
```

```
private void frmStatisticsRoom_Load(object sender, EventArgs e)
{
    userLevelTableAdapter.Fill(ticTacDataSet.UserLevel);
    TestGridControl.DataSource =
    _StatisticsTableAdapter.GetTestsByUserID(User.ID);
}

private void InitForm()
{
    ResizeControls();
    FillLastTestStats();
    AdviceGenerator ag = new AdviceGenerator(User);
    lblLastTestEval.Text = ag.EvaluateLastTest();
    lblAdvice.Text = ag.GenerateAdvice();
}

private void ResizeControls()
{
    foreach (Control c in pnlStatButtons.Controls)
    {
        if (c is SimpleButton)
        {
            (c as SimpleButton).Height = (pnlStatButtons.Height - 6) / 5;
        }
    }
    foreach (Control c in tpQuestion.Controls)
    {
        if (c is GroupControl)
        {
            (c as GroupControl).Width = (pnlStats.Width -
pnlStatButtons.Width - 10) / 2;
        }
    }
}

private void FillLastTestStats()
{
    DataTable dtLastTest =
    _StatisticsTableAdapter.GetLastTestStatsByUserID(User.ID);
    if (dtLastTest.Rows.Count > 0)
    {
        LastTestScore = dtLastTest.Rows.Count > 0 ?
dtLastTest.Rows[0]["Rating"].ToString() : "-";
        int iRating = (Convert.ToInt32(LastTestScore));
        SkinForm(iRating);
    }
}

private void FillLevelStats()
{
```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        DataTable dtLevelStats =
        _StatisticsTableAdapter.GetLevelStatsByUserID(User.ID,5);
        foreach (DataRow dr in dtLevelStats.Rows)
        {
            int l = Convert.ToInt32(dr["DifficultyLevel"]);
            (tpLevel.Controls["pb" + l.ToString()] as
ProgressBarControl).Position = Convert.ToInt32(dr["AverageRating"]);
            (tpLevel.Controls["lbt" + l.ToString()] as LabelControl).Text =
dr["TotalTests"].ToString();
            (tpLevel.Controls["lbq" + l.ToString()] as LabelControl).Text =
dr["TotalQuestions"].ToString();
            (tpLevel.Controls["lbb" + l.ToString()] as LabelControl).Text =
dr["BestRating"].ToString()+"%";
            (tpLevel.Controls["lbw" + l.ToString()] as LabelControl).Text =
dr["WorstRating"].ToString()+"%";
        }
    }

    private void FillQuestionStats()
    {
        int idx = 0;
        DataTable dtQuestionTypeStats =
        _StatisticsTableAdapter.GetQuestionTypeStatsByUserIDLevelID(User.ID, User.Level);
        foreach (DataRow dr in dtQuestionTypeStats.Rows)
        {
            int t = Convert.ToInt32(dr["QuestionTypeID"]);
            double tq = Convert.ToDouble(dr["TotalQuestions"]);
            double cq = Convert.ToDouble(dr["CorrectQuestions"]);
            double prc = cq / tq * 100;
            if (idx == 0 || idx == dtQuestionTypeStats.Rows.Count - 1)
            {
                LabelControl l = (grpQuestionType.Controls["lbqt" +
t.ToString()] as LabelControl);
                l.BorderStyle =
DevExpress.XtraEditors.Controls.BorderStyles.Simple;
                l.Appearance.BorderColor = (idx == 0) ? Color.ForestGreen :
Color.Tomato;
            }
            (grpQuestionType.Controls["lbqt" + t.ToString()] as
LabelControl).Text = String.Format("{0} στις {1}{2}", cq, tq, string.Format("
({0}%)", (int)prc));
            idx++;
        }
        idx = 0;
        int errorCount =
(int)_StatisticsTableAdapter.GetTotalErrorCount(User.ID, User.Level);
        DataTable dtErrorTypeStats =
        _StatisticsTableAdapter.GetErrorTypeStatsByUserLevelID(User.ID, User.Level);
        foreach (DataRow dr in dtErrorTypeStats.Rows)
        {
            int t = Convert.ToInt32(dr["ErrorTypeID"]);
            int e = Convert.ToInt32(dr["Errors"]);
            if (idx == 0 || idx == dtErrorTypeStats.Rows.Count-1)
            {
                LabelControl l = (grpErrorType.Controls["lbet" +
t.ToString()] as LabelControl);
                l.BorderStyle =
DevExpress.XtraEditors.Controls.BorderStyles.Simple;

```

```

        l.Appearance.BorderColor = (idx==0) ? Color.Tomato :
Color.ForestGreen;
    }
    if (errorCount != 0)
        (grpErrorType.Controls["lbet" + t.ToString()] as
LabelControl).Text = String.Format("{0} από {1}{2}", e, errorCount,
string.Format(" ({0}%)", (int)(e * 100 / errorCount)));
    else
    {
        (grpErrorType.Controls["lbet" + t.ToString()] as
LabelControl).Text = String.Empty;
        (grpErrorType.Controls["lbet" + t.ToString()] as
LabelControl).BorderStyle =
DevExpress.XtraEditors.Controls.BorderStyles.NoBorder;
    }
    idx++;
}

}

private void SkinForm(int aRating)
{
    string prefix = string.Empty;
    if (User.SkinID == 1)
        prefix = "def";
    else if (User.SkinID == 2)
        prefix = "ani";
    else if (User.SkinID == 3)
        prefix = "emo";

    ResourceManager resourceManager = new
ResourceManager("TicTacApplication.Properties.Resources", GetType().Assembly);

    if (0 <= aRating && aRating <= 20)
    {
        picEdtLastTest.Image = (Bitmap)resourceManager.GetObject(prefix +
"F");
    }
    else if (20 < aRating && aRating <= 35)
    {
        picEdtLastTest.Image = (Bitmap)resourceManager.GetObject(prefix +
"E");
    }
    else if (35 < aRating && aRating <= 50)
    {
        picEdtLastTest.Image = (Bitmap)resourceManager.GetObject(prefix +
"D");
    }
    else if (50 < aRating && aRating <= 65)
    {
        picEdtLastTest.Image = (Bitmap)resourceManager.GetObject(prefix +
"C");
    }
    else if (65 < aRating && aRating <= 80)
    {
        picEdtLastTest.Image = (Bitmap)resourceManager.GetObject(prefix +
"B");
    }
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον


```

        else if (80 < aRating && aRating <= 100)
        {
            picEdtLastTest.Image = (Bitmap)resourceManager.GetObject(prefix +
"A");
        }
        picEdtAdvice.Image = (Bitmap)resourceManager.GetObject(prefix +
"_advisor");
    }

    private void frmStatisticsRoom_Shown(object sender, EventArgs e)
    {
        InitForm();
    }

    private void chkBtnTest_CheckedChanged(object sender, EventArgs e)
    {
        if ((sender as CheckButton).Checked)
        {
            xtraTabControl1.SelectedTabPageIndex = Convert.ToInt32((sender as
CheckButton).Tag);
        }
    }

    private void xtraTabControl1_SelectedPageChanged(object sender,
DevExpress.XtraTab.TabPageChangedEventArgs e)
    {
        if (e.Page == tpLevel)
        {
            FillLevelStats();
        }
        else if (e.Page == tpQuestion)
        {
            FillQuestionStats();
        }
        else if (e.Page == tpTime)
        {
            chartControl1.Width = (tpTime.Width - 40) / 2;
            timeEfficiencyTableAdapter.Fill(ticTacDataSet.TimeEfficiency,
User.ID);
        }
        else if (e.Page == tpRecords)
        {
            RecordsGridControl.DataSource = RecordsDataSource();
        }
    }

    private void gvTests_CustomUnboundColumnData(object sender,
DevExpress.XtraGrid.Views.Base.CustomColumnDataEventArgs e)
    {
        if (e.Column == colDate)
        {
            DataRow dr = gvTests.GetDataRow(e.RowHandle);
            e.Value = dr["StartTime"];
        }
    }

    private DataTable RecordsDataSource()
    {
        DataTable resval = new DataTable();
    }

```

```

resval.Columns.Add("Description", typeof(string));
resval.Columns.Add("Image", typeof(Image));
resval.Columns.Add("Score", typeof(string));

DataRow dr = resval.NewRow();
dr["Description"] = "Συνεχόμενες Σωστές Απαντήσεις";
dr["Image"] = Properties.Resources.tr_p3;
dr["Score"] = string.Format("{0} ερωτήσεις", GetStreak(true));
resval.Rows.Add(dr);

dr = resval.NewRow();
dr["Description"] = "Συνεχόμενες Λάθος Απαντήσεις";
dr["Image"] = Properties.Resources.tr_n1;
dr["Score"] = string.Format("{0} ερωτήσεις", GetStreak(false));
resval.Rows.Add(dr);

dr = resval.NewRow();
dr["Description"] = "Καλύτερο Διαγώνισμα";
dr["Image"] = Properties.Resources.tr_p1;
dr["Score"] = string.Format("{0} στα 100",
_StatisticsTableAdapter.MaxTestRating(User.ID));
resval.Rows.Add(dr);

dr = resval.NewRow();
dr["Description"] = "Χειρότερο Διαγώνισμα";
dr["Image"] = Properties.Resources.tr_n2;
dr["Score"] = string.Format("{0} στα 100",
_StatisticsTableAdapter.MinTestRating(User.ID));
resval.Rows.Add(dr);

return resval;
}

private int GetStreak(bool correct)
{
    int resval = 0;
    int counter = 0;
    DataTable dt =
_StatisticsTableAdapter.GetAllQuestionsByUserID(User.ID);
    foreach (DataRow dr in dt.Rows)
    {
        if (dr["Correct"].Equals(correct))
        {
            counter++;
        }
        else
        {
            if (counter > resval)
                resval = counter;
            counter = 0;
        }
    }
    if (counter > resval)
        resval = counter;
    return resval;
}
}
}
}

```

10.6 Κώδικας - frmStudyRoom.cs

```

using System;
using System.Drawing;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using Utils;

namespace TicTacApplication.Forms
{
    public partial class frmStudyRoom : TicTacApplication.Forms.frmRoom
    {
        public User User { get; set; }

        private int _recommendedChapter;
        public int RecommendedChapter
        {
            get
            {
                return _recommendedChapter;
            }
            set
            {
                _recommendedChapter = value;
                switch (value)
                {
                    case 1:
                        lblChapter1.LinkColor = Color.Tomato;
                        break;
                    case 2:
                        lblChapter2.LinkColor = Color.Tomato;
                        break;
                    case 3:
                        lblChapter3.LinkColor = Color.Tomato;
                        break;
                    case 4:
                        lblChapter4.LinkColor = Color.Tomato;
                        break;
                    default: break;
                }
            }
        }

        private StudySession _studySession;

        public frmStudyRoom()
        {
            InitializeComponent();
        }

        private void aChapter_LinkClicked(object sender,
        LinkLabelLinkClickedEventArgs e)
        {
            string filepath = String.Format("{0}\\rtf\\{1}",
            Application.StartupPath, (sender as LinkLabel).Tag);

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        RichTextBox reader = new RichTextBox { Size = new Size(width - 50,
Height - 105), BackColor = Color.FromArgb(255, 255, 192), Location = new
Point(25, 25), ReadOnly = true, ScrollBars = RichTextBoxScrollBars.None, Font =
new Font("Comic Sans MS", 16.0F) };
        reader.LoadFile(filepath);
        Controls.Add(reader);
        CreatePreviousNextReturnButtons(reader);
        reader.BringToFront();
        string sChapterNo = (sender as
LinkLabel).Name.Replace("lblChapter", "");
        int iChapterNo = Convert.ToInt32(sChapterNo);
        _studySession = new StudySession(User.ID, iChapterNo);
    }
    SimpleButton btnReturn, btnUp, btnDown;
    private void CreatePreviousNextReturnButtons(RichTextBox reader)
    {
        btnUp = new SimpleButton();
        btnUp.Size = new Size(150, 50);
        btnUp.Location = new Point(reader.Left, reader.Bottom + 5);
        btnUp.AllowFocus = false;
        btnUp.LookAndFeel.UseDefaultLookAndFeel = true;
        btnUp.Appearance.TextOptions.WordWrap =
DevExpress.Utils.WordWrap.Wrap;
        Font f = btnUp.Appearance.Font;
        btnUp.Appearance.Font = new Font(f.FontFamily, 10F, FontStyle.Bold);
        btnUp.Text = "Προηγούμενη Σελίδα";
        btnUp.Click += btnUp_Click;
        btnUp.Tag = reader;

        btnDown = new SimpleButton();
        btnDown.Size = new Size(150, 50);
        btnDown.Location = new Point(btnUp.Right+5, reader.Bottom + 5);
        btnDown.AllowFocus = false;
        btnDown.LookAndFeel.UseDefaultLookAndFeel = true;
        btnDown.Appearance.TextOptions.WordWrap =
DevExpress.Utils.WordWrap.Wrap;
        btnDown.Appearance.Font = btnUp.Appearance.Font;
        btnDown.Text = "Επόμενη Σελίδα";
        btnDown.Click += btnDown_Click;
        btnDown.Tag = reader;

        btnReturn = new SimpleButton();
        btnReturn.Size = new Size(150, 50);
        btnReturn.Location = new Point(reader.Right - 150, reader.Bottom +
5);
        btnReturn.AllowFocus = false;
        btnReturn.Appearance.Font = btnUp.Appearance.Font;
        btnReturn.LookAndFeel.UseDefaultLookAndFeel = true;
        btnReturn.Text = "Επιστροφή";
        btnReturn.Click += btnReturn_Click;
        btnReturn.Tag = reader;

        Controls.AddRange(new Control[] { btnReturn, btnUp, btnDown });
    }

    void btnUp_Click(object sender, EventArgs e)
    {
        ((sender as SimpleButton).Tag as RichTextBox).Focus();
    }

```

```

        SendKeys.Send("{PGUP}");
    }

    void btnDown_Click(object sender, EventArgs e)
    {
        ((sender as SimpleButton).Tag as RichTextBox).Focus();
        SendKeys.Send("{PGDN}");
    }

    void btnReturn_Click(object sender, EventArgs e)
    {
        _studySession.EndTime = DateTime.Now;
        _studySession.SaveSession();

        (btnReturn.Tag as RichTextBox).Dispose();
        btnReturn.Dispose();
        btnDown.Dispose();
        btnUp.Dispose();
    }

    private void frmStudyRoom_Load(object sender, EventArgs e)
    {
        RecommendedChapter = User.Level;
    }
}
}

```

10.7 Κώδικας - AdviceGenerator.cs

```

using System;
using System.Text;
using System.Data;
using Utils.EnumExtensions;
using Utils.UtilsDataSetTableAdapters;

namespace Utils
{
    public class AdviceGenerator
    {
        enum TestScores
        {
            [StringValue("άριστη")]
            Excellent = 1,
            [StringValue("πολύ καλή")]
            VeryGood = 2,
            [StringValue("ικανοποιητική")]
            Satisfying = 3,
            [StringValue("μέτρια")]
            Mediocre = 4,
            [StringValue("κάτω του μετρίου")]
            BelowAverage = 5,
            [StringValue("κακή")]
            Poor = 6
        };

        enum ErrorTypesDesc

```



```

    {
        [StringValue("κατανόηση")]
        General = 1,
        [StringValue("σύγχυσης μικρού και μεγάλου δείκτη,")]
        SmallLargeHand = 2,
        [StringValue("σύγχυσης μεταξύ \"παρά\" και \"και\",")]
        PastTo = 3,
        [StringValue("σε ότι αφορά στον υπολογισμό διάρκειας,")]
        Duration = 4,
        [StringValue("σε ότι αφορά στην τιμή του λεπτοδείκτη,")]
        MinuteHandValue = 5,
        [StringValue("σε ότι αφορά στην τιμή του ωροδείκτη,")]
        HourHandValue = 6,
        [StringValue("σε ότι αφορά στην ορθογραφία,")]
        Spelling = 7
    }

    enum QuestionTypesDesc
    {
        [StringValue("πολλαπλής επιλογής με ρολόγια ως επιλογές.")]
        MultipleChoiceClock = 1,
        [StringValue("πολλαπλής επιλογής με επιλογές κειμένου.")]
        MultipleChoiceText = 2,
        [StringValue("που απαιτείται να ρυθμίσεις το ρολόι σε καθορισμένη
ώρα.")]
        SetClock = 3,
        [StringValue("που απαιτείται υπολογισμός κάποιου χρονικού
διαστήματος.")]
        SetRange = 4,
        [StringValue("τύπου Σωστό/Λάθος.")]
        RightWrong = 5,
        [StringValue("ανοιχτού τύπου που πρέπει να πληκτρολογήσεις την
απάντηση.")]
        OpenAnswer = 6
    };

    const string Not = @"δεν";

    string sTestScore = @"{0}, η επίδοσή σου στο τελευταίο διαγώνισμα ήταν
{1}. ";

    public User User { get; set; }
    private int _LastTestLevel = -1;
    private int _LastTestRating = 0;

    private TestTableAdapter _TestTableAdapter = new TestTableAdapter();
    private StudySessionTableAdapter _StudySessionTableAdapter = new
StudySessionTableAdapter();
    private StatisticsTableAdapter _StatisticsTableAdapter = new
StatisticsTableAdapter();

    private bool bStudy = false;
    private bool bPractice = false;
    private bool bAttention = false;
    private bool bLevelUp = false;
    private bool bLevelDown = false;

    public AdviceGenerator(User user)
    {

```

```

        User = user;
    }

    public string EvaluateLastTest()
    {
        StringBuilder sBuilder = new StringBuilder();
        int iScore = GetLastTestScore();
        if(iScore>0)
        {
            EvaluateStudyTime(_LastTestRating);
            bool bPrevTest = EvaluatePreviousTest(_LastTestRating);
            bool bMistakes = EvaluateMistakes(_LastTestRating);
            string sFirstName = User.FirstName.TrimEnd('ς');
            sBuilder.Append(string.Format(sTestScore, sFirstName,
StringEnum.GetStringValue((TestScores)iScore, 0)));
            sBuilder.Append(string.Format(sStudy, arg0_Study, arg1_Study,
arg2_Study, arg3_Study));
            if (bPrevTest)
                sBuilder.Append(string.Format(sPreviousTest, arg0_PrevTest,
arg1_PrevTest));
            if (bMistakes)
                sBuilder.Append(string.Format(sMistakes, arg0_Mistakes,
arg1_Mistakes, arg2_Mistakes));
        }
        return sBuilder.ToString();
    }

    public string GenerateAdvice()
    {
        string sAdvice = string.Empty;
        DataTable dtLevelStats =
_StatisticsTableAdapter.GetCurrentLevelStatsByUserIDLevel(User.ID, new
Settings().QuestionNumber, User.Level);
        if (dtLevelStats.Rows.Count > 0)
        {
            bLevelUp =
(Convert.ToInt32(dtLevelStats.Rows[0]["AverageRating"]) > 65) && (User.Level <
4);
            bLevelDown =
(Convert.ToInt32(dtLevelStats.Rows[0]["AverageRating"]) < 20) && (User.Level >
1);
        }
        if (bLevelDown)
        {
            sAdvice = string.Format(@"Η επιδόσεις σου τελευταία είναι
ιδιαιτέρα χαμηλές. Για αυτό πιστεύω ότι το καλύτερο θα ήταν να επιστρέψεις στο
Κεφάλαιο {0}. Διάβασε καλά τη θεωρία, χωρίς να βιάζεσαι και όταν πιστεύεις ότι
έχει προετοιμαστεί κατάλληλα κάνε πάλι ένα διαγώνισμα. Προσπάθησε να αποφύγεις τα
εύκολα λάθη και πιστεύω ότι θα τα πας πολύ καλύτερα αυτή τη φορά. Καλή
επιτυχία!", User.Level - 1);
        }
        else
        {
            if (bLevelUp)
            {
                string sReady = User.Sex == 'F' ? "έτοιμη" : "έτοιμος";
                string sCareful = User.Sex == 'F' ? "προσεκτική" :
"προσεκτικός";

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

string sConcentrate = User.Sex == 'F' ? "συγκεντρωμένη" :
"συγκεντρωμένος";

sAdvice = string.Format(@"Στα τελευταία διαγωνίσματα, τα
έχεις πάει αρκετά καλά και πιστεύω ότι είσαι πλέον {0} να περάσεις στο Κεφάλαιο
{1}. ", sReady, User.Level + 1);
if (bStudy)
    sAdvice += string.Format("Πάντως για παν ενδεχόμενο, θα
σε συμβούλευα να ρίξεις πρώτα μια ματιά στη θεωρία του Κεφαλαίου {0} για να
είμαστε σίγουροι ότι δεν θα αφήσουμε πίσω κενά.", User.Level);
if (bAttention)
    sAdvice += string.Format("Τα αποτελέσματά σου θα ήταν
ακόμα καλύτερα αν ήσουν πιο {0} και {1}. Προσπάθησε στο μέλλον να αποφύγεις τα
εύκολα λάθη. ", sCareful, sConcentrate);
    sAdvice += "Καλή Επιτυχία!";
}
else
{
string sOwn = User.Sex == 'F' ? "μόνη" : "μόνος";
string sCareful = User.Sex == 'F' ? "προσεκτική" :
"προσεκτικός";
string sConcentrate = User.Sex == 'F' ? "συγκεντρωμένη" :
"συγκεντρωμένος";
if (bStudy)
    sAdvice += string.Format(@"Πιθανότατα δεν έχεις
κατανοήσει στον απαιτούμενο βαθμό κάποια πράγματα στη θεωρία του Κεφαλαίου {0}.
Μία καλή επανάληψη πιστεύω θα σε βοηθούσε πολύ. ", User.Level);
if(bPractice)
    sAdvice += "Τα διαγωνίσματα σε βοηθουν σημαντικά να
καταλάβεις το αντικείμενο. Με λίγη εξάσκηση ακόμα νομίζω πως θα τα πηγαίνεις
περίφημα! ";
if (bAttention)
    sAdvice += string.Format("Τα αποτελέσματά σου θα ήταν
ακόμα καλύτερα αν ήσουν πιο {0} και {1}. Προσπάθησε στο μέλλον να αποφύγεις τα
εύκολα λάθη.", sCareful, sConcentrate);
if (string.IsNullOrEmpty(sAdvice))
    sAdvice = string.Format("Βρίσκεσαι στο σωστό δρόμο! Τα
πηγαίνεις μια χαρά και {0} σου. Συνέχισε την καλή προσπάθεια...", sOwn);
}
}
return sAdvice;
}

private int GetLastTestScore()
{
int resval = -1;
DataTable dtLastTest = _TestTableAdapter.GetDataByUserID(User.ID);
if (dtLastTest.Rows.Count > 0)
{
_LastTestLevel =
Convert.ToInt32(dtLastTest.Rows[0]["DifficultyLevel"]);
_LastTestRating = Convert.ToInt32(dtLastTest.Rows[0]["Rating"]);
if (_LastTestRating > 80 && _LastTestRating <= 100)
    resval = 1;
else if (_LastTestRating > 65 && _LastTestRating <= 80)
    resval = 2;
else if (_LastTestRating > 50 && _LastTestRating <= 65)
    resval = 3;
}
}
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        else if (_LastTestRating > 35 && _LastTestRating <= 50)
            resval = 4;
        else if (_LastTestRating > 20 && _LastTestRating <= 35)
            resval = 5;
        else if (_LastTestRating >= 0 && _LastTestRating <= 20)
            resval = 6;
    }
    return resval;
}

private string sStudy = @"{0} ο χρόνος που αφιέρωσες για μελέτη {1} ήταν
{2}, {3} ";
private string arg0_Study = string.Empty;
private string arg1_Study = string.Empty;
private string arg2_Study = string.Empty;
private string arg3_Study = string.Empty;

private void EvaluateStudyTime(int score)
{
    int totalTime =
(int)_StudySessionTableAdapter.GetTotalStudyTime(User.ID, _LastTestLevel);
    if (totalTime >= 0 && totalTime <= 120)
    {
        bStudy = true;
        arg1_Study = string.Empty;
        arg2_Study = "ελάχιστος";
    }
    else if (totalTime > 120 && totalTime <= 240)
    {
        if (score > 50)
        {
            arg1_Study = Not;
            arg2_Study = "αρκετός";
        }
        else
        {
            bStudy = true;
            arg1_Study = string.Empty;
            arg2_Study = "ανεπαρκής";
        }
    }
    else if (totalTime > 240 && totalTime <= 360)
    {
        bStudy = true;
        arg1_Study = string.Empty;
        arg2_Study = "σχετικά λίγος";
    }
    else if (totalTime > 360 && totalTime <= 480)
    {
        arg1_Study = string.Empty;
        arg2_Study = "οριακά επαρκής";
    }
    else if (totalTime > 480 && totalTime <= 600)
    {
        arg1_Study = string.Empty;
        arg2_Study = "επαρκής";
    }
    else if (totalTime > 600)
    {

```

```

        arg1_Study = string.Empty;
        arg2_Study = "αρκετός";
    }

    if (score > 50 && totalTime < 360)
    {
        arg0_Study = "Αν και";
        if (score <= 70)
        {
            arg3_Study = "ο τρόπος που αντιμετώπισες το διαγώνισμα,
έκρυψε καλά τις αδυναμίες σου.";
            bStudy = true;
        }
        else
            arg3_Study = "στο διαγώνισμα έδειξες την απαιτούμενη
σοβαρότητα, και η προσπάθειά σου ανταμείφθηκε. Μπράβο!";
    }
    else if (score <= 50 && totalTime >= 360)
    {
        bPractice = true;
        arg0_Study = "Ωστόσο,";
        arg3_Study = "πράγμα που πιθανότατα σημαίνει ότι δεν ήσουν
συγκεντρωμένος είτε κατά τη διάρκεια του διαγωνίσματος, είτε κατά τη διάρκεια της
μελέτης.";
        bAttention = true;
    }
    else
    {
        arg0_Study = "Επιπλέον,";
        if (score > 65)
            arg3_Study = "γεγονός που δείχνει ότι η προσπάθεια σου
ανταμείβεται. Συνέχισε την καλή προσπάθεια!";
        else if (score > 50)
            arg3_Study = "και παρ'όλο που δεν τα πήγες και ασχημα,
υπάρχουν μεγάλα περιθώρια βελτίωσης. Προσπάθησε σκληρότερα την επόμενη φορά!";
        else if (score > 35)
        {
            arg3_Study = "οπότε το αποτέλεσμα δεν προκαλεί έκπληξη. Θα
πρέπει να προσπαθήσεις περισσότερο απο εδώ και πέρα!";
            bAttention = true;
        }
        else
        {
            arg3_Study = "και το αποτέλεσμα μοιάζει να είναι συνέπεια της
μικρής προσπάθειας που κατέβαλλες. Χρειάζεται πολλή δουλειά... Προσπάθησε!";
            bAttention = true;
        }
        bStudy = (totalTime < 360);
        bPractice = (score < 55);
    }
}

private string sPreviousTest = @"Συγκριτικά με το προηγούμενο διαγώνισμα,
{0}, {1} ";
private string arg0_PrevTest = string.Empty;
private string arg1_PrevTest = string.Empty;

private bool EvaluatePreviousTest(int score)

```



```

    {
        bool resval = false;
        DataTable dtPrevTest =
        _TestTableAdapter.GetPreviousTestByUserIDLevel(User.ID, _LastTestLevel);
        if (dtPrevTest.Rows.Count > 0)
        {
            resval = true;
            int prevScore = Convert.ToInt32(dtPrevTest.Rows[0]["Rating"]);
            int iDiff = score - prevScore;
            if (iDiff >= -5 && iDiff <= 5)
            {
                if (score > 50)
                {
                    arg0_PrevTest = "τα πήγες περίπου το ίδιο καλά";
                    arg1_PrevTest = (score > 70) ? "αξίζεις συγχαρητήρια για
τη σταθερά καλή σου επίδοση!" : "πάντως να ξέρεις ότι μπορείς και καλύτερα.";
                }
                else
                {
                    arg0_PrevTest = "η επίδοσή σου κυμάνθηκε στα ίδια χαμηλά
επίπεδα";
                    arg1_PrevTest = "σίγουρα προσπάθησες.";
                }
            }
            else if (iDiff > 5 && iDiff <= 30)
            {
                if (score < 50)
                {
                    arg0_PrevTest = "αυτή τη φορά τα πήγες καλύτερα";
                    arg1_PrevTest = "χρειάζεται ωστόσο αρκετή δουλειά
ακόμα.";
                }
                else
                {
                    arg0_PrevTest = "προσπάθησες και αυτό φάνηκε";
                    arg1_PrevTest = "αυτή τη φορά τα πήγες ισοθιτά καλύτερα,
συνέχισε έτσι!";
                }
            }
            else if (iDiff > 30)
            {
                bAttention = true;
                arg0_PrevTest = "η βελτίωση είναι κάτι παραπάνω από ισοθιτή";
                if (score < 50)
                    arg1_PrevTest = "μπορεί να χρειάζεται ακόμη δουλειά, όμως
η προσπάθειά σου είναι αξιόλογη!";
                else
                    arg1_PrevTest = "και αυτό αντικατοπτρίζεται στα
αποτελέσματα. Συνέχισε έτσι!";
            }
            else if (iDiff < -5 && iDiff >= -30)
            {
                if (score > 50)
                {
                    arg0_PrevTest = "αν και η επίδοσή σου είναι ακόμα
αποδεκτή";
                    arg1_PrevTest = "την προηγούμενη φορά τα είχες παεί λίγο
καλύτερα.";
                }
            }
        }
    }

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

else
{
    arg0_PrevTest = "μάλλον αυτή τη φορά έχασες τη
συγκέντρωσή σου";
    arg1_PrevTest = "το αποτέλεσμα απέχει από το αναμενόμενο.
Έχεις ήδη αποδείξει ότι μπορείς, αλλά θέλεις;";
}
}
else if (iDiff < -30)
{
    bAttention = true;
    if (score > 50)
    {
        arg0_PrevTest = "αν και η επίδοσή σου είναι ακόμα
αποδεκτή";
        arg1_PrevTest = "την προηγούμενη φορά τα είχες παεί πολύ
καλύτερα.";
    }
    else
    {
        arg0_PrevTest = "μάλλον αυτή τη φορά βιάστηκες και έχασες
τη συγκέντρωσή σου";
        arg1_PrevTest = "το αποτέλεσμα απέχει πολύ από αυτό που
μπορείς.";
    }
}
}
return resval;
}

string sMistakes = @"Έκανες {0} λάθη, κυρίως {1} και σε δυσκόλεψαν οι
ερωτήσεις {2} ";
private string arg0_Mistakes = string.Empty;
private string arg1_Mistakes = string.Empty;
private string arg2_Mistakes = string.Empty;

private bool EvaluateMistakes(int score)
{
    DataTable dtLastTestStats =
_StatisticsTableAdapter.GetLastTestStatsByUserID(User.ID);
    if (dtLastTestStats.Rows.Count > 0)
    {
        if (score == 100)
            return false;
        else if (score > 80 && score < 100)
            arg0_Mistakes = "λίγα";
        else if (score > 65 && score <= 80)
            arg0_Mistakes = "σχετικά λίγα";
        else if (score > 50 && score <= 65)
            arg0_Mistakes = "μερικά";
        else if (score > 35 && score <= 50)
            arg0_Mistakes = "αρκετά";
        else if (score > 20 && score <= 35)
            arg0_Mistakes = "πολλά";
        else if (score >= 0 && score <= 20)
            arg0_Mistakes = "πάρα πολλά";

        int[] errorTypeArray = new int[7];
        int[] questTypeArray = new int[6];
    }
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        for (int i = 0; i < 7; i++)
        {
            errorTypeArray[i] =
dtLastTestStats.Select(string.Format("ErrorType = {0}", i + 1)).Length;
        }
        for (int i = 0; i < 6; i++)
        {
            questTypeArray[i] =
dtLastTestStats.Select(string.Format("QuestionTypeID = {0} AND Correct=0", i +
1)).Length;
        }
        arg1_Mistakes =
StringEnum.GetStringValue((ErrorTypesDesc)(GetArrayMaxElementIdx(errorTypeArray)
+ 1), 0);
        arg2_Mistakes =
StringEnum.GetStringValue((QuestionTypesDesc)(GetArrayMaxElementIdx(questTypeArra
y) + 1), 0);
        return true;
    }
    return false;
}

private static int GetArrayMaxElementIdx(int[] array)
{
    int idx = -1;
    int max = 0, tmp = 0;

    foreach (int i in array)
    {
        if (i > max)
        {
            max = i;
            idx = tmp;
        }
        tmp++;
    }
    return idx;
}
}
}
}

```

10.8 Κώδικας - Answer.cs

```

using System;

namespace Utils
{
    public class Answer : IDisposable
    {
        public Question Question { get; set; }
        public int ErrorTypeID { get; set; }
        public string AnswerText { get; set; }
        public bool IsCorrect { get; set; }

        public void Dispose()
    }
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```
    {  
        GC.SuppressFinalize(this);  
    }  
}
```

10.9 Κώδικας - ClockParams.cs

```
using System;  
using System.Drawing;  
  
namespace Utils  
{  
    public class ClockParams  
    {  
        public DateTime Value { get; set; }  
        public TimeZones TimeZone { get; set; }  
        public Image Image { get; set; }  
  
        public ClockParams(string sParams)  
        {  
            TimeZone = TimeZones.tzNone;  
            ReadParams(sParams);  
        }  
  
        private void ReadParams(string sParams)  
        {  
            string[] sArrayParams = sParams.Split(':');  
            if (sArrayParams.Length == 2 || sArrayParams.Length == 3)  
            {  
                int hours = Convert.ToInt32(sArrayParams[0]);  
                int mins = Convert.ToInt32(sArrayParams[1]);  
                Value = DateTime.Now.Date.AddHours(hours).AddMinutes(mins);  
                if (sArrayParams.Length == 3)  
                {  
                    int tzone = Convert.ToInt32(sArrayParams[2]);  
                    TimeZone = (TimeZones)tzone;  
                    SetTimeZoneImage();  
                }  
            }  
            else  
                throw new ArgumentOutOfRangeException();  
        }  
  
        private void SetTimeZoneImage()  
        {  
            switch (TimeZone)  
            {  
                case TimeZones.tzMorning:  
                    Image = AdditionalResources.Resources.morning;  
                    break;  
                case TimeZones.tzMidday:  
                    Image = AdditionalResources.Resources.midday;  
                    break;  
                case TimeZones.tzAfternoon:  
                    Image = AdditionalResources.Resources.afternoon;  
            }  
        }  
    }  
}
```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        break;
    case TimeZones.tzNight:
        Image = AdditionalResources.Resources.night;
        break;
    default:
        Image = null;
        break;
    }
}

public void Save(AnalogClock.Clock clock)
{
    clock.Value = Value;
    if (Image != null)
    {
        clock.BackgroundImage = Image;
        clock.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Zoom;
    }
}
}
}

```

10.10 Κώδικας - Question.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Data;
using Utils.UtilsDataSetTableAdapters;

namespace Utils
{
    public class Question : IDisposable
    {
        ErrorAnswerTableAdapter errorAnswerTableAdapter = new
ErrorAnswerTableAdapter();

        public int QuestionID { get; set; }
        public QuestionTypes Type { get; set; }
        public string DefaultText { get; set; }
        public string QuestionText { get; set; }
        public string ParameterText { get; set; }
        public string Parameters { get; set; }
        private string _givenAnswer;
        public string GivenAnswer
        {
            get { return _givenAnswer; }
            set
            {
                if (_givenAnswer != value)
                {
                    _givenAnswer = value;
                    End = DateTime.Now;
                }
            }
        }
    }
}

```

TitTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον


```

    }
    public bool IsMultipleChoice { get; set; }
    public bool IsAnswerCorrect { get; set; }
    public DateTime Start { get; set; }
    public DateTime End { get; set; }
    public ErrorTypes ErrorType { get; set; }
    public Test Test { get; set; }
    public List<Answer> Answers = new List<Answer>();
    public UtilsDataSet.ErrorAnswerDataTable AllQuestionWrongAnswers;

    public void GenerateAnswers(string correctAnswer)
    {
        using (Answer a = new Answer { AnswerText = correctAnswer,
ErrorTypeID = -1, IsCorrect = true, Question = this })
        {
            Answers.Add(a);
        }

        if (IsMultipleChoice)
        {
            AllQuestionWrongAnswers =
errorAnswerTableAdapter.GetDataByQuestionID(QuestionID);
            IEnumerable<int> randomWrongAnswerIds =
Randomizer.PickNItemsInRandomOrder(AllQuestionWrongAnswers.Select(e =>
e.ErrorAnswerID), Test.MultipleChoices - 1);
            foreach (int id in randomWrongAnswerIds)
            {
                var ans = AllQuestionWrongAnswers.Where(e => e.ErrorAnswerID
== id).First();
                Answer a = new Answer { AnswerText = ans.Answer, ErrorTypeID
= ans.ErrorTypeID, IsCorrect = false, Question = this };
                Answers.Add(a);
            }
            //Triple Shuffle
            Answers = Randomizer.ShuffleA(Answers).ToList();
            Answers = Randomizer.ShuffleB(Answers).ToList();
            Answers = Randomizer.ShuffleC(Answers).ToList();
        }
    }

    public void Dispose()
    {
        GC.SuppressFinalize(this);
    }
}
}
}

```

10.11 Κώδικας - Randomizer.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;

```

```

namespace Utils
{
    public static class Randomizer
    {
        public static IEnumerable<int> PickNItemsInRandomOrder(IEnumerable<int>
itemCollection, int count)
        {
            Random random = new Random(DateTime.Now.Millisecond);

            Dictionary<double, int> randomSortTable = new Dictionary<double,
int>();

            foreach (int i in itemCollection)
                randomSortTable[random.NextDouble()] = i;

            return randomSortTable.OrderBy(KVP => KVP.Key).Take(count).Select(KVP
=> KVP.Value);
        }

        public static IList<T> ShuffleA<T>(IList<T> list)
        {
            int count = list.Count;
            Random random = new Random(DateTime.Now.Millisecond);

            Dictionary<double, T> randomSortTable = new Dictionary<double, T>();

            foreach (var v in list)
                randomSortTable[random.NextDouble()] = v;

            return randomSortTable.OrderBy(KVP => KVP.Key).Take(count).Select(KVP
=> KVP.Value).ToList();
        }

        public static readonly Random random = new Random();
        public static IList<T> ShuffleB<T>(IList<T> toShuffle)
        {
            List<T> deck = new List<T>(toShuffle);
            int N = deck.Count;

            for (int i = 0; i < N; ++i)
            {
                int r = i + (int)(random.Next(N - i));
                T t = deck[r];
                deck[r] = deck[i];
                deck[i] = t;
            }

            return deck;
        }

        public static IList<T> ShuffleC<T>(this IList<T> list)
        {
            using (RNGCryptoServiceProvider provider = new
RNGCryptoServiceProvider())
            {
                int n = list.Count;
                while (n > 1)
                {
                    byte[] box = new byte[1];

```

```

do
    provider.GetBytes(box);
while (!(box[0] < n * (Byte.MaxValue / n)));
int k = (box[0] % n);
n--;
T value = list[k];
list[k] = list[n];
list[n] = value;
}
}
return list;
}
}
}

```

10.12 Κώδικας - Settings.cs

```

using System;

namespace Utils
{
    public class Settings
    {
        public int QuestionNumber { get; set; }
        public int ChoicesNumber { get; set; }
        public int EvalTestNumber { get; set; }
        public int MaleSkinID { get; set; }
        public int FemaleSkinID { get; set; }
        public int AdminSkinID { get; set; }

        public Settings()
        {
            LoadCurrentSettings();
        }

        public void LoadCurrentSettings()
        {
            using (UtilsDataSetTableAdapters.SettingsTableAdapter
settingsTableAdapter = new UtilsDataSetTableAdapters.SettingsTableAdapter())
            {
                UtilsDataSet.SettingsRow drSettings =
settingsTableAdapter.GetData().Rows[0] as UtilsDataSet.SettingsRow;
                QuestionNumber = drSettings.QuestionNumber;
                ChoicesNumber = drSettings.ChoicesNumber;
                EvalTestNumber = drSettings.TestNumber;
                MaleSkinID = drSettings.SkinMale;
                FemaleSkinID = drSettings.SkinFemale;
                AdminSkinID = drSettings.SkinAdmin;
            }
        }
    }
}

```

10.13 Κώδικας - SimilarityExtensions.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

// A set of extension methods for determining string similarity
// Collected, refactored, or ported by Steve Hawley, steve.hawley@atalasoft.com
// Credit for original code is attached to the public method
// Use freely, but at your own risk - not guaranteed to be correct

namespace Utils
{
    public static class SimilarityExtensions
    {
        static SimilarityExtensions()
        {
            _soundExTable = new List<KeyValuePair<string, string>>();
            _soundExTable.Add(new KeyValuePair<string, string>("bfpv", "1"));
            _soundExTable.Add(new KeyValuePair<string, string>("cgjksxz", "2"));
            _soundExTable.Add(new KeyValuePair<string, string>("dt", "3"));
            _soundExTable.Add(new KeyValuePair<string, string>("l", "4"));
            _soundExTable.Add(new KeyValuePair<string, string>("mn", "5"));
            _soundExTable.Add(new KeyValuePair<string, string>("r", "6"));
        }

        // Extension method for getting Levenshtein distance, based on:
        // http://www.merriampark.com/ldcsharp.htm
        // by Lasse Johansen
        public static int LevenshteinDistance(this string s, string t)
        {
            if (s == null)
                throw new ArgumentNullException("s");
            if (t == null)
                throw new ArgumentNullException("t");
            int n = s.Length;
            int m = t.Length;
            int[,] d = new int[n+1,m+1];

            if (n == 0 || m == 0)
                return Math.Max(m, n);

            for (int i = 0; i <= n; i++)
            {
                d[i, 0] = i;
            }
            for (int i = 0; i < m; i++)
            {
                d[0, i] = i;
            }

            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < m; j++)
                {

```

```

        int cost = (t[j] == s[i]) ? 0 : 1;

        d[i + 1, j + 1] = Math.Min(Math.Min(d[i, j + 1] + 1, d[i + 1,
j] + 1), d[i, j] + cost);
    }
}

return d[n, m];
}

// Extension method for getting SoundEx, based on:
// http://blogs.techrepublic.com.com/programming-and-development/?p=656
// by Zach Smith

static List<KeyValuePair<string, string>> _soundExTable;

public static string SoundEx(this string s)
{
    if (s == null)
        throw new ArgumentNullException("s");
    if (s.Length == 0)
        return "0000";

    StringBuilder sb = new StringBuilder();
    string previousCode = "";
    sb.Append(Char.ToLower(s[0]));

    for (int i=1; i < s.Length; i++)
    {
        char currentLetter = Char.ToLower(s[i]);

        KeyValuePair<string, string> found =
_soundExTable.FirstOrDefault(kvp => kvp.Key.Contains(currentLetter));
        string currentCode = found.Value;
        if (currentCode != null) {
            if (currentCode != previousCode) {
                sb.Append(currentCode);
                if (sb.Length == 4)
                    break;
                previousCode = currentCode;
            }
        }
    }
    if (sb.Length < 4)
        sb.Append(new String('0', 4 - sb.Length));
    return sb.ToString().ToUpper();
}

private static bool ContainsSubstring(this string s, string t, int start,
int count)
{
    if (s == null)
        throw new ArgumentNullException("s");
    if (t == null)
        throw new ArgumentNullException("t");
    return s.Contains(t.Substring(start, count));
}

public static int SoundExDifference(string soundex1, string soundex2)

```



```

    {
        if (soundex1 == null)
            throw new ArgumentNullException("soundex1");
        if (soundex2 == null)
            throw new ArgumentNullException("soundex2");
        if (soundex1.Length != 4)
            throw new ArgumentOutOfRangeException("soundex1", "soundex
strings need to be 4 characters long");
        if (soundex2.Length != 4)
            throw new ArgumentOutOfRangeException("soundex2", "soundex
strings need to be 4 characters long");

        int result = 0;

        if (soundex2.ContainsSubstring(soundex1, 1, 3))
        {
            result = 3;
        }
        else if (soundex2.ContainsSubstring(soundex1, 2, 2))
        {
            result = 2;
        }
        else if (soundex2.ContainsSubstring(soundex1, 1, 2))
        {
            result = 2;
        }
        else
        {
            if (soundex2.ContainsSubstring(soundex1, 1, 1))
                result++;
            if (soundex2.ContainsSubstring(soundex1, 2, 1))
                result++;
            if (soundex2.ContainsSubstring(soundex1, 3, 1))
                result++;
        }
        if (soundex1[0] == soundex2[0])
            result++;
        return result == 0 ? 1 : result;
    }

    public static int SoundEx(this string s, string t)
    {
        return SoundExDifference(s.SoundEx(), t.SoundEx());
    }

    public static bool StartsWith(this string s, params string[] candidate)
    {
        string match = candidate.FirstOrDefault(t => s.StartsWith(t));
        return match != default(string);
    }

    public static bool SubstringIs(this string s, int start, int length,
params string[] candidate)
    {
        if (start < 0)
            return false;
        string sub = s.Substring(start, length);
        string match = candidate.FirstOrDefault(t => t == sub);
    }

```

```

        return match != default(string);
    }

    public static bool IsVowelOrY(this char c)
    {
        return "AEIOUY".Contains(Char.ToUpper(c));
    }

    static string[] _slavoGermanicRoots = new string[] { "W", "K", "CZ",
"WITZ" };
    private static bool IsSlavoGermanic(string s)
    {
        return _slavoGermanicRoots.FirstOrDefault(t => s.Contains(t)) !=
default(string);
    }

    private static void AddMetaphoneCharacter(String primaryCharacter, String
alternateCharacter, StringBuilder primaryKey, StringBuilder alternateKey)
    {
        //Is the primary character valid?
        if (primaryCharacter.Length > 0)
        {
            primaryKey.Append(primaryCharacter);
        }

        //Is the alternate character valid?
        if (alternateCharacter != null)
        {
            //Alternate character was provided. If it is not zero-length,
append it, else
            //append the primary string as long as it wasn't zero length and
isn't a space character
            if (alternateCharacter.Length > 0)
            {
                if (alternateCharacter[0] != ' ')
                {
                    alternateKey.Append(alternateCharacter);
                }
            }
            else
            {
                //No, but if the primary character is valid, add that instead
                if (primaryCharacter.Length > 0 && (primaryCharacter[0] != '
'))
                {
                    alternateKey.Append(primaryCharacter);
                }
            }
        }
        else if (primaryCharacter.Length > 0)
        {
            //Else, no alternate character was passed, but a primary was, so
append the primary character to the alternate key
            alternateKey.Append(primaryCharacter);
        }
    }

    private const int METAPHONE_KEY_LENGTH = 4;

```

```

// double metaphone from Adam Neslon here
// http://www.codeproject.com/KB/recipes/dmetaphone5.aspx

public static KeyValuePair<string, string> DoubleMetaphone(this string s)
{
    if (s == null)
        throw new ArgumentNullException("s");
    string word = s.ToUpper();
    int length = word.Length;
    int last = length - 1;
    int current = 0;

    StringBuilder primaryKey = new StringBuilder();
    StringBuilder alternateKey = new StringBuilder();
    bool isSlavOrGermanic = IsSlavoGermanic(word);

    if (length < 1)
        return new KeyValuePair<string, string>("", "");

    //skip these when at start of word
    if (s.StartsWith("GN", "KN", "PN", "WR", "PS"))
        current += 1;

    //Initial 'X' is pronounced 'Z' e.g. 'Xavier'
    if (word[0] == 'X')
    {
        AddMetaphoneCharacter("S", null, primaryKey, alternateKey); // 'Z'
        current += 1;
    }

    ////////////////main loop//////////////////////////////////////
    while ((primaryKey.Length < METAPHONE_KEY_LENGTH) ||
    (alternateKey.Length < METAPHONE_KEY_LENGTH))
    {
        if (current >= length)
            break;

        switch (word[current])
        {
            case 'A':
            case 'E':
            case 'I':
            case 'O':
            case 'U':
            case 'Y':
                if (current == 0)
                    //all init vowels now map to 'A'
                    AddMetaphoneCharacter("A", null, primaryKey,
alternateKey);
                current++;
                break;

            case 'B':

                //"-mb", e.g., "dumb", already skipped over...
                AddMetaphoneCharacter("P", null, primaryKey,
alternateKey);

```

```

        if (word[current + 1] == 'B')
            current += 2;
        else
            current++;
        break;

    case 'Ç':
        AddMetaphoneCharacter("S", null, primaryKey,
alternateKey);

        current += 1;
        break;

    case 'C':
        //various germanic
        if ((current > 1)
            && !word[current - 2].IsVowelOrY()
            && word.SubstringIs((current - 1), 3, "ACH")
            && ((word[current + 2] != 'I') && ((word[current + 2]
!= 'E')
word.SubstringIs((current - 2), 6, "BACHER", "MACHER"))))
        {
            AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);

            current += 2;
            break;
        }

        //special case 'caesar'
        if ((current == 0) && word.SubstringIs(current, 6,
"CAESAR"))
        {
            AddMetaphoneCharacter("S", null, primaryKey,
alternateKey);

            current += 2;
            break;
        }

        //italian 'chianti'
        if (word.SubstringIs(current, 4, "CHIA"))
        {
            AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);

            current += 2;
            break;
        }

        if (word.SubstringIs(current, 2, "CH"))
        {
            //find 'michael'
            if ((current > 0) && word.SubstringIs(current, 4,
"CHAE"))
            {
                AddMetaphoneCharacter("K", "X", primaryKey,
alternateKey);

                current += 2;
                break;
            }
        }
    }
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

//greek roots e.g. 'chemistry', 'chorus'
if ((current == 0)
    && (word.SubstringIs((current + 1), 5, "HARAC",
"HARIS")
        || word.SubstringIs((current + 1), 3, "HOR",
"HYM", "HIA", "HEM")))
    {
        AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);
        current += 2;
        break;
    }

//germanic, greek, or otherwise 'ch' for 'kh' sound
if ((word.SubstringIs(0, 4, "VAN ", "VON ") ||
word.SubstringIs(0, 3, "SCH"))
    // 'architect but not 'arch', 'orchestra',
'orchid'
    || word.SubstringIs((current - 2), 6, "ORCHES",
"ARCHIT", "ORCHID")
    || word.SubstringIs((current + 2), 1, "T", "S")
    || ((word.SubstringIs((current - 1), 1, "A", "O",
"U", "E") || (current == 0))
        //e.g., 'wachtler', 'wechsler', but not 'tichner'
        && word.SubstringIs((current + 2), 1, "L",
"R", "N", "M", "B", "H", "F", "V", "W", " ")))
    {
        AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);
    }
    else
    {
        if (current > 0)
        {
            if (word.SubstringIs(0, 2, "MC"))
                //e.g., "McHugh"
                AddMetaphoneCharacter("K", null,
primaryKey, alternateKey);
            else
                AddMetaphoneCharacter("X", "K",
primaryKey, alternateKey);
        }
        else
            AddMetaphoneCharacter("X", null, primaryKey,
alternateKey);
    }
    current += 2;
    break;
}
//e.g., 'czerny'
if (word.SubstringIs(current, 2, "CZ") &&
!word.SubstringIs((current - 2), 4, "WICZ"))
{
    AddMetaphoneCharacter("S", "X", primaryKey,
alternateKey);
    current += 2;
    break;
}

```



```

    }

    //e.g., 'focaccia'
    if (word.SubstringIs((current + 1), 3, "CIA"))
    {
        AddMetaphoneCharacter("X", null, primaryKey,
alternateKey);

        current += 3;
        break;
    }

    //double 'C', but not if e.g. 'McClellan'
    if (word.SubstringIs(current, 2, "CC") && !((current ==
1) && (word[0] == 'M'))))
        // 'bellocchio' but not 'bacchus'
        if (word.SubstringIs((current + 2), 1, "I", "E", "H")
&& !word.SubstringIs((current + 2), 2, "HU"))
            {
                // 'accident', 'accede' 'succeed'
                if (((current == 1) && (word[current - 1] ==
'A'))
                || word.SubstringIs((current - 1), 5,
"UCCEE", "UCCES"))
                    AddMetaphoneCharacter("KS", null, primaryKey,
alternateKey);

                // 'bacci', 'bertucci', other italian
                else
                    AddMetaphoneCharacter("X", null, primaryKey,
alternateKey);

                current += 3;
                break;
            }
            else
            { //Pierce's rule
                AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);

                current += 2;
                break;
            }
        }

        if (word.SubstringIs(current, 2, "CK", "CG", "CQ"))
        {
            AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);

            current += 2;
            break;
        }

        if (word.SubstringIs(current, 2, "CI", "CE", "CY"))
        {
            //italian vs. english
            if (word.SubstringIs(current, 3, "CIO", "CIE",
"CIA"))
                AddMetaphoneCharacter("S", "X", primaryKey,
alternateKey);

            else
                AddMetaphoneCharacter("S", null, primaryKey,
alternateKey);

            current += 2;

```

```

        break;
    }

    //else
    AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);

    //name sent in 'mac caffrey', 'mac gregor
    if (word.SubstringIs((current + 1), 2, "C", "Q", "G"))
        current += 3;
    else
        if (word.SubstringIs((current + 1), 1, "C", "K", "Q")
            && !word.SubstringIs((current + 1), 2, "CE",
"CI"))
            current += 2;
        else
            current += 1;
        break;

    case 'D':
        if (word.SubstringIs(current, 2, "DG"))
            if (word.SubstringIs((current + 2), 1, "I", "E",
"Y"))
                {
                    //e.g. 'edge'
                    AddMetaphoneCharacter("J", null, primaryKey,
alternateKey);

                    current += 3;
                    break;
                }
            else
                {
                    //e.g. 'edgar'
                    AddMetaphoneCharacter("TK", null, primaryKey,
alternateKey);

                    current += 2;
                    break;
                }

            if (word.SubstringIs(current, 2, "DT", "DD"))
                {
                    AddMetaphoneCharacter("T", null, primaryKey,
alternateKey);

                    current += 2;
                    break;
                }

            //else
            AddMetaphoneCharacter("T", null, primaryKey,
alternateKey);

            current += 1;
            break;

    case 'F':
        if (word[current + 1] == 'F')
            current += 2;
        else
            current += 1;

```

```

AddMetaphoneCharacter("F", null, primaryKey,
alternateKey);
break;
case 'G':
if (word[current + 1] == 'H')
{
if ((current > 0) && !word[current - 1].IsVowelOrY())
{
AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);
current += 2;
break;
}
if (current < 3)
{
//'ghislane', ghiradelli
if (current == 0)
{
if (word[current + 2] == 'I')
AddMetaphoneCharacter("J", null,
primaryKey, alternateKey);
else
AddMetaphoneCharacter("K", null,
primaryKey, alternateKey);
current += 2;
break;
}
}
//Parker's rule (with some further refinements) -
if (((current > 1) && word.SubstringIs((current - 2),
1, "B", "H", "D"))
|| ((current > 2) && word.SubstringIs((current -
3), 1, "B", "H", "D"))
|| ((current > 3) && word.SubstringIs((current -
4), 1, "B", "H")))
{
current += 2;
break;
}
else
{
//e.g., 'laugh', 'McLaughlin', 'cough', 'gough',
'rough', 'tough'
if ((current > 2)
&& (word[current - 1] == 'U')
&& word.SubstringIs((current - 3), 1, "C",
"G", "L", "R", "T"))
{
AddMetaphoneCharacter("F", null, primaryKey,
alternateKey);
}
else
if ((current > 0) && word[current - 1] !=
'I')

```

```

AddMetaphoneCharacter("K", null,
primaryKey, alternateKey);

        current += 2;
        break;
    }
}
if (word[current + 1] == 'N')
{
    if ((current == 1) && word[0].IsVowelOrY() &&
!isSlavOrGermanic)
    {
        AddMetaphoneCharacter("KN", "N", primaryKey,
alternateKey);
    }
    else
        //not e.g. 'cagney'
        if (!word.SubstringIs((current + 2), 2, "EY")
&& (word[current + 1] != 'Y') &&
!isSlavOrGermanic)
        {
            AddMetaphoneCharacter("N", "KN", primaryKey,
alternateKey);
        }
        else
            AddMetaphoneCharacter("KN", null, primaryKey,
alternateKey);
        current += 2;
        break;
    }
}
//'tagliaro'
if (word.SubstringIs((current + 1), 2, "LI") &&
!isSlavOrGermanic)
{
    AddMetaphoneCharacter("KL", "L", primaryKey,
alternateKey);
    current += 2;
    break;
}
//-ges-, -gep-, -gel-, -gie- at beginning
if ((current == 0)
&& ((word[current + 1] == 'Y')
|| word.SubstringIs((current + 1), 2, "ES",
"EP", "EB", "EL", "EY", "IB", "IL", "IN", "IE", "EI", "ER")))
{
    AddMetaphoneCharacter("K", "J", primaryKey,
alternateKey);
    current += 2;
    break;
}
}
// -ger-, -gy-
if ((word.SubstringIs((current + 1), 2, "ER") ||
(word[current + 1] == 'Y'))
&& !word.StartsWith("DANGER", "RANGER", "MANGER")
&& !word.SubstringIs((current - 1), 1, "E", "I"))

```

```

    && !word.SubstringIs((current - 1), 3, "RGY", "OGY"))
    {
        AddMetaphoneCharacter("K", "J", primaryKey,
alternateKey);
        current += 2;
        break;
    }

    // italian e.g. 'biaggi'
    if (word.SubstringIs((current + 1), 1, "E", "I", "Y") ||
word.SubstringIs((current - 1), 4, "AGGI", "OGGI"))
    {
        //obvious germanic
        if ((word.StartsWith("VAN ", "VON ", "SCH"))
            || word.SubstringIs((current + 1), 2, "ET"))
            AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);
        else
            //always soft if french ending
            if (word.SubstringIs((current + 1), 4, "IER "))
                AddMetaphoneCharacter("J", null, primaryKey,
alternateKey);
            else
                AddMetaphoneCharacter("J", "K", primaryKey,
alternateKey);
        current += 2;
        break;
    }

    if (word[current + 1] == 'G')
        current += 2;
    else
        current += 1;
    AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);
    break;

    case 'H':
        //only keep if first & before vowel or btw. 2 vowels
        if (((current == 0) || word[current - 1].IsVowelOrY())
            && word[current + 1].IsVowelOrY())
        {
            AddMetaphoneCharacter("H", null, primaryKey,
alternateKey);
            current += 2;
        }
        else//also takes care of 'HH'
            current += 1;
        break;

    case 'J':
        //obvious spanish, 'jose', 'san jacinto'
        if (word.SubstringIs(current, 4, "JOSE") ||
word.StartsWith("SAN "))
        {
            if (((current == 0) && (word[current + 4] == ' ')) ||
word.StartsWith("SAN "))
                AddMetaphoneCharacter("H", null, primaryKey,
alternateKey);

```



```

else
{
    AddMetaphoneCharacter("J", "H", primaryKey,
alternateKey);
}
current += 1;
break;
}

if ((current == 0) && !word.SubstringIs(current, 4,
"JOSE"))
    AddMetaphoneCharacter("J", "A", primaryKey,
alternateKey); //Yankelovich/Jankelowicz
else
    //spanish pron. of e.g. 'bajador'
    if (word[current - 1].IsVowelOrY()
        && !isSlavOrGermanic
        && ((word[current + 1] == 'A') || (word[current +
1] == 'O'))))
        AddMetaphoneCharacter("J", "H", primaryKey,
alternateKey);
else
    if (current == last)
        AddMetaphoneCharacter("J", " ", primaryKey,
alternateKey);
else
    if (!word.SubstringIs((current + 1), 1, "L",
"T", "K", "S", "N", "M", "B", "Z")
        && !word.SubstringIs((current - 1), 1,
"S", "K", "L"))
        AddMetaphoneCharacter("J", null,
primaryKey, alternateKey);

    if (word[current + 1] == 'J') //it could happen!
        current += 2;
    else
        current += 1;
    break;

case 'K':
    if (word[current + 1] == 'K')
        current += 2;
    else
        current += 1;
    AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);
    break;

case 'L':
    if (word[current + 1] == 'L')
    {
        //spanish e.g. 'cabrillo', 'gallegos'
        if (((current == (length - 3))
            && word.SubstringIs((current - 1), 4, "ILLO",
"ILLA", "ALLE"))
            || ((word.SubstringIs((last - 1), 2, "AS", "OS")
|| word.SubstringIs(last, 1, "A", "O"))
            && word.SubstringIs((current - 1), 4,
"ALLE")))

```

```

        {
            AddMetaphoneCharacter("L", " ", primaryKey,
alternateKey);
            current += 2;
            break;
        }
        current += 2;
    }
    else
        current += 1;
        AddMetaphoneCharacter("L", null, primaryKey,
alternateKey);
        break;
    case 'M':
        if ((word.SubstringIs((current - 1), 3, "UMB")
            && (((current + 1) == last) ||
word.SubstringIs((current + 2), 2, "ER")))
            //'dumb', 'thumb'
            || (word[current + 1] == 'M'))
            current += 2;
        else
            current += 1;
            AddMetaphoneCharacter("M", null, primaryKey,
alternateKey);
            break;
    case 'N':
        if (word[current + 1] == 'N')
            current += 2;
        else
            current += 1;
            AddMetaphoneCharacter("N", null, primaryKey,
alternateKey);
            break;
    case 'Ñ':
        current += 1;
        AddMetaphoneCharacter("N", null, primaryKey,
alternateKey);
        break;
    case 'P':
        if (word[current + 1] == 'H')
        {
            AddMetaphoneCharacter("F", null, primaryKey,
alternateKey);
            current += 2;
            break;
        }
        //also account for "campbell", "raspberry"
        if (word.SubstringIs((current + 1), 1, "P", "B"))
            current += 2;
        else
            current += 1;
            AddMetaphoneCharacter("P", null, primaryKey,
alternateKey);
            break;

```

```

case 'Q':
    if (word[current + 1] == 'Q')
        current += 2;
    else
        current += 1;
    AddMetaphoneCharacter("K", null, primaryKey,
alternateKey);
    break;

case 'R':
    //french e.g. 'rogier', but exclude 'hochmeier'
    if ((current == last)
        && !isSlavOrGermanic
        && word.SubstringIs((current - 2), 2, "IE")
        && !word.SubstringIs((current - 4), 2, "ME", "MA"))
        AddMetaphoneCharacter("", "R", primaryKey,
alternateKey);
    else
        AddMetaphoneCharacter("R", null, primaryKey,
alternateKey);

    if (word[current + 1] == 'R')
        current += 2;
    else
        current += 1;
    break;

case 'S':
    //special cases 'island', 'isle', 'carlisle', 'carlyslle'
    if (word.SubstringIs((current - 1), 3, "ISL", "YSL"))
    {
        current += 1;
        break;
    }

    //special case 'sugar-'
    if ((current == 0) && word.StartsWith("SUGAR"))
    {
        AddMetaphoneCharacter("X", "S", primaryKey,
alternateKey);
        current += 1;
        break;
    }

    if (word.SubstringIs(current, 2, "SH"))
    {
        //germanic
        if (word.SubstringIs((current + 1), 4, "HEIM",
"HOEK", "HOLM", "HOLZ"))
            AddMetaphoneCharacter("S", null, primaryKey,
alternateKey);
        else
            AddMetaphoneCharacter("X", null, primaryKey,
alternateKey);
        current += 2;
        break;
    }
}

```

```

//italian & armenian
if (word.SubstringIs(current, 3, "SIO", "SIA") ||
word.SubstringIs(current, 4, "SIAN"))
{
    if (!isSlavOrGermanic)
        AddMetaphoneCharacter("S", "X", primaryKey,
alternateKey);
    else
        AddMetaphoneCharacter("S", null, primaryKey,
alternateKey);
    current += 3;
    break;
}

//german & anglicisations, e.g. 'smith' match 'schmidt',
'snider' match 'schneider'
//also, -sz- in slavic language altho in hungarian it is
pronounced 's'
if (((current == 0)
    && word.SubstringIs((current + 1), 1, "M", "N", "L",
"W"))
    || word.SubstringIs((current + 1), 1, "Z"))
{
    AddMetaphoneCharacter("S", "X", primaryKey,
alternateKey);
    if (word.SubstringIs((current + 1), 1, "Z"))
        current += 2;
    else
        current += 1;
    break;
}

if (word.SubstringIs(current, 2, "SC"))
{
    //Schlesinger's rule
    if (word[current + 2] == 'H')
        //dutch origin, e.g. 'school', 'schooner'
        if (word.SubstringIs((current + 3), 2, "OO",
"ER", "EN", "UY", "ED", "EM"))
        {
            //schermerhorn, 'schenker'
            if (word.SubstringIs((current + 3), 2, "ER",
"EN"))
            {
                AddMetaphoneCharacter("X", "SK",
primaryKey, alternateKey);
            }
            else
                AddMetaphoneCharacter("SK", null,
primaryKey, alternateKey);
            current += 3;
            break;
        }
        else
        {
            if ((current == 0) && !word[3].IsVowelOrY()
&& (word[3] != 'W'))
                AddMetaphoneCharacter("X", "S",
primaryKey, alternateKey);
        }
    }
}

```

```

else
    AddMetaphoneCharacter("X", null,
primaryKey, alternateKey);
    current += 3;
    break;
}

"Υ"))
    if (word.SubstringIs((current + 2), 1, "I", "E",
alternateKey);
    {
        AddMetaphoneCharacter("S", null, primaryKey,
        current += 3;
        break;
    }
    //else
    AddMetaphoneCharacter("SK", null, primaryKey,
alternateKey);
    current += 3;
    break;
}

//french e.g. 'resnais', 'artois'
2, "AI", "OI"))
    if ((current == last) && word.SubstringIs((current - 2),
alternateKey);
        AddMetaphoneCharacter("", "S", primaryKey,
else
        AddMetaphoneCharacter("S", null, primaryKey,
alternateKey);

    if (word.SubstringIs((current + 1), 1, "S", "Z"))
        current += 2;
    else
        current += 1;
    break;

case 'T':
    if (word.SubstringIs(current, 4, "TION"))
    {
        AddMetaphoneCharacter("X", null, primaryKey,
alternateKey);
        current += 3;
        break;
    }

    if (word.SubstringIs(current, 3, "TIA", "TCH"))
    {
        AddMetaphoneCharacter("X", null, primaryKey,
alternateKey);
        current += 3;
        break;
    }

    if (word.SubstringIs(current, 2, "TH")
        || word.SubstringIs(current, 3, "TTH"))
    {
        //special case 'thomas', 'thames' or germanic
        if (word.SubstringIs((current + 2), 2, "OM", "AM"))

```



```

|| word.SubstringIs(0, 4, "VAN ", "VON ")
|| word.SubstringIs(0, 3, "SCH"))
{
    AddMetaphoneCharacter("T", null, primaryKey,
alternateKey);
}
else
{
    AddMetaphoneCharacter("θ", "T", primaryKey,
alternateKey);
}
current += 2;
break;
}

if (word.SubstringIs((current + 1), 1, "T", "D"))
    current += 2;
else
    current += 1;
AddMetaphoneCharacter("T", null, primaryKey,
alternateKey);
break;

case 'V':
    if (word[current + 1] == 'V')
        current += 2;
    else
        current += 1;
    AddMetaphoneCharacter("F", null, primaryKey,
alternateKey);
    break;

case 'W':
    //can also be in middle of word
    if (word.SubstringIs(current, 2, "WR"))
    {
        AddMetaphoneCharacter("R", null, primaryKey,
alternateKey);
        current += 2;
        break;
    }
    if ((current == 0)
        && (word[current + 1].IsVowelOrY() ||
word.SubstringIs(current, 2, "WH")))
    {
        //Wasserman should match Vasserman
        if (word[current + 1].IsVowelOrY())
            AddMetaphoneCharacter("A", "F", primaryKey,
alternateKey);
        else
            //need Uomo to match Womo
            AddMetaphoneCharacter("A", null, primaryKey,
alternateKey);
    }

    //Arnow should match Arnoff
    if (((current == last) && word[current - 1].IsVowelOrY())

```

```

|| word.SubstringIs((current - 1), 5, "EWSKI",
"EWSKY", "OWSKI", "OWSKY")
|| word.SubstringIs(0, 3, "SCH"))
{
AddMetaphoneCharacter("", "F", primaryKey,
alternateKey);
current += 1;
break;
}

//polish e.g. 'filipowicz'
if (word.SubstringIs(current, 4, "WICZ", "WITZ"))
{
AddMetaphoneCharacter("TS", "FX", primaryKey,
alternateKey);
current += 4;
break;
}

//else skip it
current += 1;
break;

case 'X':
//french e.g. breaux
if (!(current == last)
&& (word.SubstringIs((current - 3), 3, "IAU",
"EAU")
|| word.SubstringIs((current - 2), 2, "AU",
"OU"))))
AddMetaphoneCharacter("KS", null, primaryKey,
alternateKey);

if (word.SubstringIs((current + 1), 1, "C", "X"))
current += 2;
else
current += 1;
break;

case 'Z':
//chinese pinyin e.g. 'zhao'
if (word[current + 1] == 'H')
{
AddMetaphoneCharacter("J", null, primaryKey,
alternateKey);
current += 2;
break;
}
else
if (word.SubstringIs((current + 1), 2, "ZO", "ZI",
"ZA")
|| (isSlavOrGermanic && ((current > 0) &&
word[current - 1] != 'T'))))
{
AddMetaphoneCharacter("S", "TS", primaryKey,
alternateKey);
}
else

```

```

        AddMetaphoneCharacter("S", null, primaryKey,
alternateKey);

        if (word[current + 1] == 'Z')
            current += 2;
        else
            current += 1;
        break;

    default:
        current += 1;
        break;
    }
}

//Finally, chop off the keys at the proscribed length
string primaryKeyString = primaryKey.ToString(0,
Math.Min(primaryKey.Length, METAPHONE_KEY_LENGTH));
string alternateKeyString = alternateKey.ToString(0,
Math.Min(alternateKey.Length, METAPHONE_KEY_LENGTH));

return new KeyValuePair<string, string>(primaryKeyString,
alternateKeyString);
}

private struct SimilarityUnit
{
    public int Position1, Position2;
    public int End1, End2;
    public int Maximum;
}

private static SimilarityUnit SimilarStringHelper(string txt1, int len1,
string txt2, int len2)
{
    SimilarityUnit unit = new SimilarityUnit
    {
        Maximum = 0
    };

    int j1 = 0, j2 = 0, l = 0;
    for (int i1 = 0; i1 < len1; i1++)
    {
        for (int i2 = 0; i2 < len2; i2++)
        {
            l = 0; j1 = 0; j2 = 0;
            while ((i1 + j1 < len1) && (i2 + j2 < len2))
            {
                char ch1 = txt1[i1 + j1];
                j1++;
                char ch2 = txt2[i2 + j2];
                j2++;
                if (ch1 != ch2)
                {
                    j1--;
                    j2--;
                    break;
                }
            }
            l++;
        }
    }
}

```

```

    }
    if (l > unit.Maximum)
    {
        unit.Maximum = l;
        unit.Position1 = i1;
        unit.End1 = j1;
        unit.Position2 = i2;
        unit.End2 = j2;
    }
}
return unit;
}

private static int SimilarString(string s, int sLen, string t, int tLen)
{
    SimilarityUnit similar = SimilarStringHelper(s, sLen, t, tLen);
    int sum = similar.Maximum;
    if (sum != 0)
    {
        if (similar.Position1 != 0 && similar.Position2 != 0)
        {
            sum += SimilarString(s, similar.Position1, t,
similar.Position2);
        }
        if ((similar.Position1 + similar.End1 < sLen) &&
(similar.Position2 + similar.End2 < sLen))
        {
            sum += SimilarString(s.Substring(similar.Position1 +
similar.End1), s.Length - similar.Position1 - similar.End1,
t.Substring(similar.Position2 + similar.End2), t.Length -
similar.Position2 - similar.End2);
        }
    }
    return sum;
}

// ported from PHP source here:
// http://lxr.php.net/source/php-src/ext/standard/string.c#4536

public static float SimilarText(this string s, string t)
{
    if (s == null)
        throw new ArgumentNullException("s");
    if (t == null)
        throw new ArgumentNullException("t");

    int sim = SimilarString(s, s.Length, t, t.Length);
    return sim * 200.0f / (s.Length + t.Length);
}
}
}

```

10.14 Κώδικας - StringEnum.cs

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

using System;
using System.Collections;
using System.Reflection;

namespace Utils.EnumExtensions
{
    #region Class StringEnum

    /// <summary>
    /// Helper class for working with 'extended' enums using <see
    cref="StringValueAttribute"/> attributes.
    /// </summary>
    public class StringEnum
    {
        #region Instance implementation

        private Type _enumType;
        private static Hashtable _stringValues = new Hashtable();

        /// <summary>
        /// Creates a new <see cref="StringEnum"/> instance.
        /// </summary>
        /// <param name="enumType">Enum type.</param>
        public StringEnum(Type enumType)
        {
            if (!enumType.IsEnum)
                throw new ArgumentException(String.Format("Supplied type must be
an Enum. Type was {0}", enumType.ToString()));

            _enumType = enumType;
        }

        /// <summary>
        /// Gets the string value associated with the given enum value.
        /// </summary>
        /// <param name="valueName">Name of the enum value.</param>
        /// <returns>String Value</returns>
        public string GetStringValue(string valueName, int langIndex)
        {
            Enum enumType;
            string stringValue = null;
            try
            {
                enumType = (Enum)Enum.Parse(_enumType, valueName);
                stringValue = GetStringValue(enumType, langIndex);
            }
            catch (Exception) { } //Swallow!

            return stringValue;
        }

        /// <summary>
        /// Gets the string values associated with the enum.
        /// </summary>
        /// <returns>String value array</returns>
        public Array GetStringValues()
        {
            ArrayList values = new ArrayList();
            //Look for our string value associated with fields in this enum

```



```

        foreach (FieldInfo fi in _enumType.GetFields())
        {
            //Check for our custom attribute
            StringValueAttribute[] attrs =
            fi.GetCustomAttributes(typeof(StringValueAttribute), false) as
            StringValueAttribute[];
            if (attrs.Length > 0)
                values.Add(attrs[0].Value);
        }

        return values.ToArray();
    }

    /// <summary>
    /// Gets the values as a 'bindable' list datasource.
    /// </summary>
    /// <returns>IList for data binding</returns>
    public IList GetListValues()
    {
        Type underlyingType = Enum.GetUnderlyingType(_enumType);
        ArrayList values = new ArrayList();
        //Look for our string value associated with fields in this enum
        foreach (FieldInfo fi in _enumType.GetFields())
        {
            //Check for our custom attribute
            StringValueAttribute[] attrs =
            fi.GetCustomAttributes(typeof(StringValueAttribute), false) as
            StringValueAttribute[];
            if (attrs.Length > 0)
                values.Add(new
            DictionaryEntry(Convert.ChangeType(Enum.Parse(_enumType, fi.Name),
            underlyingType), attrs[0].Value));
        }

        return values;
    }

    /// <summary>
    /// Return the existence of the given string value within the enum.
    /// </summary>
    /// <param name="stringValue">String value.</param>
    /// <returns>Existence of the string value</returns>
    public bool IsStringDefined(string stringValue)
    {
        return Parse(_enumType, stringValue) != null;
    }

    /// <summary>
    /// Return the existence of the given string value within the enum.
    /// </summary>
    /// <param name="stringValue">String value.</param>
    /// <param name="ignoreCase">Denotes whether to conduct a case-
    insensitive match on the supplied string value</param>
    /// <returns>Existence of the string value</returns>
    public bool IsStringDefined(string stringValue, bool ignoreCase)
    {

```

```

        return Parse(_enumType, stringValue, ignoreCase) != null;
    }

    /// <summary>
    /// Gets the underlying enum type for this instance.
    /// </summary>
    /// <value></value>
    public Type EnumType
    {
        get { return _enumType; }
    }

#endregion

#region Static implementation

    /// <summary>
    /// Gets a string value for a particular enum value.
    /// </summary>
    /// <param name="value">Value.</param>
    /// <returns>String Value associated via a <see
    cref="StringValueAttribute"/> attribute, or null if not found.</returns>
    public static string GetStringValue(Enum value, int langIndex)
    {
        string output = null;
        Type type = value.GetType();

        if (_stringValues.ContainsKey(value))
        {
            output = (langIndex == 0) ?
                (_stringValues[value] as StringValueAttribute).Value :
                (_stringValues[value] as StringValueAttribute).EnValue;
        }
        else
        {
            //Look for our 'StringValueAttribute' in the field's custom
attributes
            FieldInfo fi = type.GetField(value.ToString());
            StringValueAttribute[] attrs =
fi.GetCustomAttributes(typeof(StringValueAttribute), false) as
StringValueAttribute[];
            if (attrs.Length > 0)
            {
                _stringValues.Add(value, attrs[0]);
                output = (langIndex == 0) ? attrs[0].Value :
attributes[0].EnValue;
            }
        }
        return output;
    }

    /// <summary>
    /// Parses the supplied enum and string value to find an associated enum
value (case sensitive).
    /// </summary>
    /// <param name="type">Type.</param>
    /// <param name="stringValue">String value.</param>

```

```

    /// <returns>Enum value associated with the string value, or null if not
found.</returns>
    public static object Parse(Type type, string stringValue)
    {
        return Parse(type, stringValue, false);
    }

    /// <summary>
    /// Parses the supplied enum and string value to find an associated enum
value.
    /// </summary>
    /// <param name="type">Type.</param>
    /// <param name="stringValue">String value.</param>
    /// <param name="ignoreCase">Denotes whether to conduct a case-
insensitive match on the supplied string value</param>
    /// <returns>Enum value associated with the string value, or null if not
found.</returns>
    public static object Parse(Type type, string stringValue, bool
ignoreCase)
    {
        object output = null;
        string enumStringValue = null;

        if (!type.IsEnum)
            throw new ArgumentException(String.Format("Supplied type must be
an Enum. Type was {0}", type.ToString()));

        //Look for our string value associated with fields in this enum
        foreach (FieldInfo fi in type.GetFields())
        {
            //Check for our custom attribute
            StringValueAttribute[] attrs =
fi.GetCustomAttributes(typeof(StringValueAttribute), false) as
StringValueAttribute[];
            if (attrs.Length > 0)
                enumStringValue = attrs[0].Value;

            //Check for equality then select actual enum value.
            if (string.Compare(enumStringValue, stringValue, ignoreCase) ==
0)
            {
                output = Enum.Parse(type, fi.Name);
                break;
            }
        }
        return output;
    }

    /// <summary>
    /// Return the existence of the given string value within the enum.
    /// </summary>
    /// <param name="stringValue">String value.</param>
    /// <param name="enumType">Type of enum</param>
    /// <returns>Existence of the string value</returns>
    public static bool IsStringDefined(Type enumType, string stringValue)
    {
        return Parse(enumType, stringValue) != null;
    }

```

```

    /// <summary>
    /// Return the existence of the given string value within the enum.
    /// </summary>
    /// <param name="stringValue">String value.</param>
    /// <param name="enumType">Type of enum</param>
    /// <param name="ignoreCase">Denotes whether to conduct a case-
insensitive match on the supplied string value</param>
    /// <returns>Existence of the string value</returns>
    public static bool IsStringDefined(Type enumType, string stringValue,
bool ignoreCase)
    {
        return Parse(enumType, stringValue, ignoreCase) != null;
    }

    #endregion
}

#endregion
}

```

10.15 Κώδικας - StringValueAttribute.cs

```

using System;

namespace Utils.EnumExtensions
{
    #region Class StringValueAttribute

    /// <summary>
    /// Simple attribute class for storing String Values
    /// </summary>
    public class StringValueAttribute : Attribute
    {
        private string _value;
        private string _enValue;

        /// <summary>
        /// Creates a new <see cref="StringValueAttribute"/> instance.
        /// </summary>
        /// <param name="value">Value.</param>
        public StringValueAttribute(string value)
        {
            _value = value;
        }
        public StringValueAttribute(string value, string envalue)
            : this(value)
        {
            _enValue = envalue;
        }
        protected StringValueAttribute()
        {
        }
    }
}

```

```
    /// <summary>
    /// Gets the value.
    /// </summary>
    /// <value></value>
    public string Value
    {
        get { return _value; }
    }
    public string EnValue
    {
        get { return _enValue; }
    }
}
#endregion
}
```

10.16 Κώδικας - StudySession.cs

```
using System;
using Utils.UtilsDataSetTableAdapters;

namespace Utils
{
    public class StudySession
    {
        public int UserID { get; set; }
        public int Chapter { get; set; }
        public DateTime StartTime { get; set; }
        public DateTime EndTime { get; set; }

        private static StudySessionTableAdapter _studySessionTableAdapter = new
StudySessionTableAdapter();

        public StudySession(int userId, int chapter)
        {
            UserID = userId;
            Chapter = chapter;
            StartTime = DateTime.Now;
        }

        public void SaveSession()
        {
            _studySessionTableAdapter.Insert(UserID, Chapter, StartTime,
EndTime);
        }
    }
}
```

10.17 Κώδικας - Test.cs

```
using System;
using System.Collections.Generic;
```

ΤιςΤαc: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον


```

using System.Linq;
using System.Data;
using System.Xml;
using Utils.UtilsDataSetTableAdapters;

namespace Utils
{
    public class Test : IDisposable
    {
        public event EventHandler OnTestCompleted;

        QuestionTableAdapter questionTableAdapter = new QuestionTableAdapter();

        public int MultipleChoices { get; set; }
        public int QuestionsPerTest { get; set; }
        public int TestLevel { get; set; }

        public XmlDocument QuestionsXml { get; set; }
        public DateTime StartTime { get; set; }
        public DateTime EndTime { get; set; }
        public float Rating { get; set; }
        public int UserID { get; set; }

        public List<Question> Questions = new List<Question>();
        public UtilsDataSet.QuestionDataTable AllTestLevelQuestions;
        public Test(int choices, int questions, int level, int userId)
        {
            MultipleChoices = choices;
            QuestionsPerTest = questions;
            TestLevel = level;
            UserID = userId;
        }

        public void TestComplete()
        {
            if (null != OnTestCompleted)
            {
                OnTestCompleted(this, new EventArgs());
            }
        }

        public void GenerateQuestions()
        {
            AllTestLevelQuestions =
            questionTableAdapter.GetDataByQuestionLevel(TestLevel);
            IEnumerable<int> randomQuestionIds =
            Randomizer.PickNItemsInRandomOrder(AllTestLevelQuestions.Select(q =>
            q.QuestionID), QuestionsPerTest);
            foreach (int id in randomQuestionIds)
            {
                var qr = AllTestLevelQuestions.Where(qu => qu.QuestionID ==
            id).First();
                Question q = new Question { Test = this, QuestionID =
            qr.QuestionID, Type = (QuestionTypes)qr.QuestionTypeID, DefaultText =
            qr.DefaultText, QuestionText = qr.QuestionText, Parameters=qr.Parameters,
            ParameterText = qr.ParameterText, IsMultipleChoice = qr.IsMultipleChoice,
            IsAnswerCorrect = false, ErrorType = ErrorTypes.Empty, GivenAnswer = string.Empty
            };
                q.GenerateAnswers(qr.Answer);
            }
        }
    }
}

```

```

        q.Start = q.End = DateTime.MinValue;
        Questions.Add(q);
    }
}

public void Dispose()
{
    GC.SuppressFinalize(this);
}
}
}

```

10.18 Κώδικας - TestEvaluator.cs

```

using System;
using System.Linq;
using System.Xml;
using System.Data;
using Utils;
using Utils.UtilsDataSetTableAdapters;

namespace Utils
{
    public class TestEvaluator : IDisposable
    {
        public Test SubmittedTest { get; set; }
        public TestEvaluator(Test t)
        {
            SubmittedTest = t;
        }

        public void EvaluateTest()
        {
            foreach (Question q in SubmittedTest.Questions)
            {
                Answer correctAns = q.Answers.Where(a => a.IsCorrect ==
true).First();
                if (q.Type == QuestionTypes.qtOpenAnswer)
                {
                    string[] sArrayAnswers = correctAns.AnswerText.Split(':');
                    if (sArrayAnswers.Contains(q.GivenAnswer))
                        q.IsAnswerCorrect = true;
                    else
                    {
                        bool bFoundError = false;
                        DataTable dtErrors = (new
ErrorAnswerTableAdapter()).GetDataByQuestionID(q.QuestionID);
                        foreach (DataRow row in dtErrors.Rows)
                        {
                            if (bFoundError) break;
                            sArrayAnswers = row["Answer"].ToString().Split(':');
                            if (sArrayAnswers.Contains(q.GivenAnswer))
                            {
                                q.ErrorType = (ErrorTypes)row["ErrorTypeID"];
                                bFoundError = true;
                            }
                        }
                    }
                }
            }
        }
    }
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

    }
    if (!bFoundError)
    {
        foreach (DataRow row in dtErrors.Rows)
        {
            if (bFoundError) break;
            sArrayAnswers =
row["Answer"].ToString().Split(':');
            foreach (string s in sArrayAnswers)
            {
                if
(SimilarityExtensions.SimilarText(q.GivenAnswer, s) >= 75)
                {
                    q.ErrorType = ErrorTypes.esSpelling;
                    bFoundError = true;
                }
            }
        }
        if (!bFoundError)
            q.ErrorType = ErrorTypes.etGeneral;
    }
}
else if (q.GivenAnswer.Equals(correctAns.AnswerText)) //Correct
Answer
{
    q.IsAnswerCorrect = true;
}
else if (!q.IsMultipleChoice) //Wrong Answer not multiple choice
{
    q.IsAnswerCorrect = false;
    object errTypeObj = (new
ErrorAnswerTableAdapter()).GetErrorType(q.QuestionID, q.GivenAnswer);
    if (null == errTypeObj)
    {
        if (q.Type == QuestionTypes.qtSetRange)
            errTypeObj = ErrorTypes.etDuration;
        else
            errTypeObj = ErrorTypes.etGeneral;
    }
    q.ErrorType = (ErrorTypes)errTypeObj;
}
else //Wrong Answer multiple choice
{
    if (string.IsNullOrEmpty(q.GivenAnswer)) //Empty Answer -
General Error
    {
        q.IsAnswerCorrect = false;
        q.ErrorType = ErrorTypes.etGeneral;
    }
    else //Selected Answer - Specified Error Type
    {
        Answer selectedAns = q.Answers.Where(a =>
a.AnswerText.Equals(q.GivenAnswer)).First();
        q.IsAnswerCorrect = false;
        q.ErrorType = (ErrorTypes)selectedAns.ErrorTypeID;
    }
}
}
}

```

```

        SubmittedTest.Rating = 100 * ((float)(SubmittedTest.Questions.Where(q
=> q.IsAnswerCorrect).Count()) / (float)(SubmittedTest.Questions.Count));
    }

    public void SerializeTestQuestions()
    {
        XmlDocument xdoc = new XmlDocument();

        XmlNode xnode_questions = xdoc.CreateElement("questions");
        xdoc.AppendChild(xnode_questions);

        foreach (Question q in SubmittedTest.Questions)
        {
            XmlNode xnode_question = xdoc.CreateElement("question");

            XmlAttribute idAttribute = xdoc.CreateAttribute("id");
            idAttribute.Value = q.QuestionID.ToString();
            xnode_question.Attributes.Append(idAttribute);

            XmlAttribute correctAttribute = xdoc.CreateAttribute("correct");
            correctAttribute.Value = q.IsAnswerCorrect.ToString();
            xnode_question.Attributes.Append(correctAttribute);

            XmlAttribute errorTypeAttribute =
            xdoc.CreateAttribute("errortype");
            errorTypeAttribute.Value = ((int)q.ErrorType).ToString();
            xnode_question.Attributes.Append(errorTypeAttribute);

            XmlAttribute startDateTimeAttribute =
            xdoc.CreateAttribute("start");
            startDateTimeAttribute.Value = q.Start.ToString("yyyy'-MM'-'dd
HH':'mm':'ss'. 'mmss");
            xnode_question.Attributes.Append(startDateTimeAttribute);

            XmlAttribute endDateTimeAttribute = xdoc.CreateAttribute("end");
            endDateTimeAttribute.Value = q.End.ToString("yyyy'-MM'-'dd
HH':'mm':'ss'. 'mmss");
            xnode_question.Attributes.Append(endDateAttribute);

            XmlAttribute durationAttribute =
            xdoc.CreateAttribute("duration");
            TimeSpan ts = (q.End - q.Start);
            durationAttribute.Value = ts.ToString();
            xnode_question.Attributes.Append(durationAttribute);

            xnode_questions.AppendChild(xnode_question);
        }
        SubmittedTest.QuestionsXml = xdoc;
    }

    public void SaveTest()
    {
        TestTableAdapter test = new TestTableAdapter();
        test.Add(SubmittedTest.QuestionsXml.InnerXml,
SubmittedTest.StartTime, SubmittedTest.EndTime, SubmittedTest.Rating,
SubmittedTest.UserID, SubmittedTest.TestLevel);
    }
}

```

```

    }

    public void Dispose()
    {
        GC.SuppressFinalize(this);
    }
}

```

10.19 Κώδικας - TestVisualizer.cs

```

using System;
using System.Drawing;
using System.Linq;
using System.Windows.Forms;
using AnalogClock;
using CustomControls;
using DevExpress.XtraEditors;

namespace Utils
{
    public class TestVisualizer
    {
        public PanelControl TargetPanel { get; set; }
        public LabelControl Label { get; set; }
        public Test Test { get; set; }
        public Question CurrentQuestion { get; set; }

        public int _qidx { get; set; }

        public TestVisualizer(PanelControl p, Test t, LabelControl l)
        {
            TargetPanel = p;
            Test = t;
            Label = l;
            _qidx = 0;
        }

        public void LoadNextQuestion()
        {
            _qidx++;
            SetQuestionMinEndTime();
            LoadQuestion();
        }

        public void LoadPrevQuestion()
        {
            _qidx--;
            SetQuestionMinEndTime();
            LoadQuestion();
        }

        private void SetQuestionMinEndTime()
        {
            if (CurrentQuestion.End == DateTime.MinValue)
                CurrentQuestion.End = DateTime.Now;
        }
    }
}

```



```

    }

    public void LoadQuestion()
    {
        Question q = Test.Questions[_qidx];
        CurrentQuestion = q;
        if (CurrentQuestion.Start == DateTime.MinValue)
            CurrentQuestion.Start = DateTime.Now;
        Label.Text = string.IsNullOrEmpty(q.QuestionText) ?
string.Format(q.DefaultText, q.ParameterText) : string.Format(q.QuestionText,
q.ParameterText);
        CreateControls();
        if (!string.IsNullOrEmpty(q.GivenAnswer))
            LoadAnswer();
        CreateNextPreviousButton();
    }
    private void LoadAnswer()
    {
        switch (CurrentQuestion.Type)
        {
            case QuestionTypes.qtMultipleChoiceClock:
                {
                    var query = from ctl in
TargetPanel.Controls.OfType<Clock>()
                                where
ctl.Tag.Equals(CurrentQuestion.GivenAnswer)
                                select ctl;
                    Clock clk = query.First();
                    HighlightClock(clk, Color.Red, Color.YellowGreen);
                } break;
            case QuestionTypes.qtMultipleChoiceText:
                {
                    var query = from ctl in
TargetPanel.Controls.OfType<CheckButton>()
                                where
ctl.Text.Equals(CurrentQuestion.GivenAnswer)
                                select ctl;
                    CheckButton cb = query.First();
                    cb.Checked = true;
                } break;
            case QuestionTypes.qtSetClock:
                {
                    string[] stringValue =
CurrentQuestion.GivenAnswer.Split(':');
                    int hrs = Convert.ToInt32(stringValue[0]);
                    int mins = Convert.ToInt32(stringValue[1]);

                    var query = from ctl in
TargetPanel.Controls.OfType<TimeSetter>()
                                select ctl;
                    TimeSetter ts = query.First();
                    ts.Hours = hrs;
                    ts.Minutes = mins;
                } break;
            case QuestionTypes.qtSetRange:
                {
                    string[] stringValue =
CurrentQuestion.GivenAnswer.Split(':');
                    int hrs = Convert.ToInt32(stringValue[0]);

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        int mins = Convert.ToInt32(stringValues[1]);

        var query = from ctl in
TargetPanel.Controls.OfType<DurationSetter>()
                    select ctl;
        DurationSetter ds = query.First();
        ds.Hours = hrs;
        ds.Minutes = mins;
    } break;
    case QuestionTypes.qtRightWrong:
    {
        var query = from ctl in
TargetPanel.Controls.OfType<CheckButton>()
                    where
ctl.Tag.ToString().Equals(CurrentQuestion.GivenAnswer)
                    select ctl;
        CheckButton checkButton = query.First();
        checkButton.Checked = true;
    } break;
    case QuestionTypes.qtOpenAnswer:
    {
        var query = from ctl in
TargetPanel.Controls.OfType<TextEdit>()
                    select ctl;
        TextEdit TxEdit = query.First();
        TxEdit.EditValue = CurrentQuestion.GivenAnswer;
    } break;
    default: break;
    }
}

private void CreateControls()
{
    switch (CurrentQuestion.Type)
    {
        case QuestionTypes.qtMultipleChoiceClock:
        {
            AddClockControls(Test.MultipleChoices);
        } break;
        case QuestionTypes.qtMultipleChoiceText:
        {
            AddClockControls(1);
            AddButtons(Test.MultipleChoices);
        } break;
        case QuestionTypes.qtSetClock:
        {
            AddClockControls(1);
            AddTimeSetterControl();
        } break;
        case QuestionTypes.qtSetRange:
        {
            string[] sParams = CurrentQuestion.Parameters.Split('-');
            int iNoOfClocks = Convert.ToInt32(sParams[0]);
            AddClockControls(iNoOfClocks);
            AddDurationSetterControl();
        } break;
        case QuestionTypes.qtRightWrong:
        {
            string[] sParams = CurrentQuestion.Parameters.Split('-');

```

```

        if (sParams.Length > 0)
        {
            if (!string.IsNullOrEmpty(sParams[0]))
            {
                int iNoOfClocks = Convert.ToInt32(sParams[0]);
                AddClockControls(iNoOfClocks);
            }
        }
        AddButtons(2);
    } break;
case QuestionTypes.qtOpenAnswer:
    {
        string[] sParams = CurrentQuestion.Parameters.Split('-');
        int iNoOfClocks = Convert.ToInt32(sParams[0]);
        AddClockControls(iNoOfClocks);
        AddTextEdit();
    } break;
default: break;
}
}

private void AddClockControls(int iNoOfClocks)
{
    if (iNoOfClocks < 1 || iNoOfClocks >= 7)
        return;
    bool isButton = (iNoOfClocks > 2);
    int iClockEdge = isButton ? 150 : (iNoOfClocks) == 1 ? 300 : 200;
    int iOffsetY = isButton ? 100 : 50;
    int iTotalClockWidth = iClockEdge * iNoOfClocks;
    int iFreeWidth = TargetPanel.Width - iTotalClockWidth;

    if (iFreeWidth < 0)
        return;

    int iMargin = (int)((float)iFreeWidth / (float)(iNoOfClocks + 1));
    int x = 0, y = iOffsetY;
    for (int i = 0; i < iNoOfClocks; i++)
    {
        x += iMargin;
        TargetPanel.Controls.Add(CreateClock(x, y, iClockEdge,
iClockEdge, isButton, i + 1));
        x += iClockEdge;
    }
}

private void AddButtons(int iNoOfButtons)
{
    if (iNoOfButtons < 2 || iNoOfButtons > 6)
        return;

    int iButtonWidth = 150;
    int iButtonHeight = 50;
    int iTotalButtonWidth = iButtonWidth * iNoOfButtons;
    int iFreeWidth = TargetPanel.Width - iTotalButtonWidth;

    if (iFreeWidth < 0)
        return;

    int iMargin = (int)((float)iFreeWidth / (float)(iNoOfButtons + 1));

```

```

        int x = 0, y = GetFreeLocY();
        if (iNoOfButtons >= 4)
        {
            for (int i = 0; i < iNoOfButtons; i++)
            {
                x += iMargin;
                TargetPanel.Controls.Add(CreateCheckButton(x, y,
iButtonWidth, iButtonHeight, CurrentQuestion.Answers[i]));
                x += iButtonWidth;
            }
        }
        else
        {
            CreateTrueFalseButtons(iMargin, y, iButtonWidth, iButtonHeight);
        }
    }

    private void AddTimeSetterControl()
    {
        int iTimeSetterWidth = 600;
        int iFreeWidth = TargetPanel.Width - iTimeSetterWidth;
        if (iFreeWidth < 0)
            return;

        int iMargin = (int)((float)iFreeWidth / (float)2);
        int x = iMargin, y = GetFreeLocY();
        TargetPanel.Controls.Add(CreateTimeSetter(x, y, iTimeSetterWidth,
100));
    }

    private void AddDurationSetterControl()
    {
        int iDurationSetterWidth = 200;
        int iFreeWidth = TargetPanel.Width - iDurationSetterWidth;
        if (iFreeWidth < 0)
            return;

        int iMargin = (int)((float)iFreeWidth / (float)2);
        int x = iMargin, y = GetFreeLocY();
        TargetPanel.Controls.Add(CreateDurationSetter(x, y,
iDurationSetterWidth, 110));
    }

    private void AddTextEdit()
    {
        int iTxEdtWidth = 600;
        int iTxEdtHeight = 50;
        int iFreeWidth = TargetPanel.Width - iTxEdtWidth;
        if (iFreeWidth < 0) return;
        int iMargin = (int)((float)iFreeWidth / (float)2);
        TextEdit txEdit = new TextEdit();
        txEdit.Properties.AutoHeight = false;
        txEdit.Size = new Size(iTxEdtWidth, iTxEdtHeight);
        txEdit.Location = new Point(iMargin, GetFreeLocY());
        Font f = txEdit.Properties.Appearance.Font;
        txEdit.Properties.Appearance.Font = new Font(f.FontFamily, 14F,
FontStyle.Regular);
        txEdit.Properties.CharacterCasing = CharacterCasing.Upper;
        txEdit.EditValueChanged += txEdit_EditValueChanged;
    }

```

```

        TargetPanel.Controls.Add(txEdit);
        txEdit.Focus();
    }

    private DurationSetter CreateDurationSetter(int x, int y, int width, int
height)
    {
        DurationSetter resval = new DurationSetter { Size = new Size(width,
height), Location = new Point(x, y) };
        resval.OnDurationChanged += durationSetter_OnDurationChanged;
        return resval;
    }

    private TimeSetter CreateTimeSetter(int x, int y, int width, int height)
    {
        TimeSetter resval = new TimeSetter { Size = new Size(width, height),
Location = new Point(x, y) };
        resval.OnTimeChanged += timeSetter_OnTimeChanged;
        return resval;
    }

    private void CreateTrueFalseButtons(int margin, int y, int width, int
height)
    {
        CheckButton btnTrue = new CheckButton();
        CheckButton btnFalse = new CheckButton();

        btnTrue.Text = "Σωστό";
        btnFalse.Text = "Λάθος";
        btnTrue.Size = btnFalse.Size = new Size(width, height);
        btnTrue.Location = new Point(margin, y);
        btnTrue.Tag = "true";
        btnFalse.Tag = "false";
        btnTrue.GroupIndex = btnFalse.GroupIndex = 1;
        btnFalse.Location = new Point(2 * margin + width, y);
        btnFalse.LookAndFeel.UseDefaultLookAndFeel =
btnTrue.LookAndFeel.UseDefaultLookAndFeel = true;
        btnTrue.CheckedChanged += btnTrueFalse_CheckedChanged;
        btnFalse.CheckedChanged += btnTrueFalse_CheckedChanged;
        btnFalse.AllowFocus = btnTrue.AllowFocus = false;
        Font f = btnTrue.Appearance.Font;
        btnTrue.Appearance.Font = btnFalse.Appearance.Font = new
Font(f.FontFamily, 10F, FontStyle.Bold);
        TargetPanel.Controls.AddRange(new Control[] { btnTrue, btnFalse });
    }

    private void CreateNextPreviousButton()
    {
        SimpleButton buttonNext = new SimpleButton();
        SimpleButton buttonPrev = new SimpleButton();

        buttonNext.Name = "btnNext";
        buttonPrev.Name = "btnPrev";
        buttonNext.Size = buttonPrev.Size = new Size(150, 50);
        buttonNext.Location = new Point(TargetPanel.Width - 155,
TargetPanel.Height - 55);
        buttonPrev.Location = new Point(5, TargetPanel.Height - 55);
        buttonNext.LookAndFeel.UseDefaultLookAndFeel =
buttonPrev.LookAndFeel.UseDefaultLookAndFeel = true;
    }

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον


```

        Font f = buttonNext.Appearance.Font;
        buttonNext.Appearance.Font = buttonPrev.Appearance.Font = new
Font(f.FontFamily, 10F, FontStyle.Bold);
        buttonNext.Text = "Επόμενο >";
        buttonPrev.Text = "< Προηγούμενο";
        buttonNext.Enabled = (_qidx < Test.QuestionsPerTest - 1);
        buttonPrev.Enabled = (_qidx > 0);
        buttonNext.Click += next_Click;
        buttonPrev.Click += prev_Click;
        if (_qidx == Test.QuestionsPerTest - 1)
        {
            buttonNext.Text = "Τέλος";
            buttonNext.Enabled = true;
            buttonNext.Click -= next_Click;
            buttonNext.Click += finish_Click;
        }
        TargetPanel.Controls.AddRange(new Control[] { buttonNext, buttonPrev
});
    }

    private Clock CreateClock(int locX, int locY, int width, int height, bool
buttonStyle, int clockNo)
    {
        Clock resval = new Clock { Name = "clock" + clockNo.ToString(), Value
= DateTime.Now, Running = false, Size = new Size(width, height), Location = new
Point(locX, locY) };
        resval.HourHand.Style = HandStyle.Pointed;
        resval.MinuteHand.Style = HandStyle.Pointed;
        resval.SecondHand.Visible = false;

        if (CurrentQuestion.Type == QuestionTypes.qtSetClock)
            resval.Value = DateTime.Now.Date;
        else if (CurrentQuestion.Type == QuestionTypes.qtMultipleChoiceText)
        {
            ClockParams p = new ClockParams(CurrentQuestion.ParameterText);
            p.Save(resval);
        }
        else if (CurrentQuestion.Type == QuestionTypes.qtSetRange
|| CurrentQuestion.Type == QuestionTypes.qtRightWrong
|| CurrentQuestion.Type == QuestionTypes.qtOpenAnswer)
        {
            string[] sParams = CurrentQuestion.Parameters.Split('-');
            ClockParams p = new ClockParams(sParams[clockNo]);
            p.Save(resval);
        }

        if (buttonStyle)
        {
            Answer answer = CurrentQuestion.Answers[clockNo - 1];
            string[] sValues = answer.AnswerText.Split(':');
            int hrs = Convert.ToInt32(sValues[0]);
            int mins = Convert.ToInt32(sValues[1]);
            resval.Value = DateTime.Now.Date.AddHours(hrs).AddMinutes(mins);
            resval.Tag = answer.AnswerText;
            resval.BorderStyle = BorderStyles.RoundFixedSingle;
            resval.Click += aClock_Click;
        }
        return resval;
    }
}

```

```

private SimpleButton CreateCheckButton(int locX, int locY, int width, int
height, Answer answer)
{
    CheckButton resval = new CheckButton { GroupIndex = 1, Tag =
answer.IsCorrect, Text = answer.AnswerText, Size = new Size(width, height),
Location = new Point(locX, locY) };
    resval.LookAndFeel.UseDefaultLookAndFeel = true;
    resval.AllowFocus = false;
    resval.CheckedChanged += checkButton_CheckedChanged;
    return resval;
}

private void HighlightClock(Clock selectedClock, Color frameColor, Color
backgroundColor)
{
    foreach (Control c in TargetPanel.Controls)
    {
        if (c is Clock)
        {
            if (selectedClock == c as Clock)
            {
                selectedClock.FrameColor = frameColor;
                selectedClock.BackColor = backgroundColor;
            }
            else
            {
                (c as Clock).FrameColor = Color.Black;
                (c as Clock).BackColor = Color.Transparent;
            }
        }
    }
}

private int GetFreeLocY()
{
    int maxY = 0;
    foreach (Control c in TargetPanel.Controls)
    {
        if (c.Bottom > maxY)
            maxY = c.Bottom;
    }
    return maxY + 50;
}

#region Control Events
private void next_Click(object sender, EventArgs e)
{
    Utils.VisibilityFunctions.ClearPanel(TargetPanel);
    LoadNextQuestion();
}

private void prev_Click(object sender, EventArgs e)
{
    Utils.VisibilityFunctions.ClearPanel(TargetPanel);
    LoadPrevQuestion();
}

private void finish_Click(object sender, EventArgs e)

```

```

    {
        SetQuestionMinEndTime();
        Test.EndTime = DateTime.Now;
        using (TestEvaluator testEval = new TestEvaluator(Test))
        {
            testEval.EvaluateTest();
            testEval.SerializeTestQuestions();
            testEval.SaveTest();
        }
        Test.TestComplete();
    }

    void aClock_Click(object sender, EventArgs e)
    {
        HighlightClock((sender as Clock), Color.Red, Color.YellowGreen);
        CurrentQuestion.GivenAnswer = (sender as Clock).Tag.ToString();
    }

    private void timeSetter_OnTimeChanged(object sender, EventArgs e)
    {
        TimeSetter ts = sender as TimeSetter;
        foreach (Control c in TargetPanel.Controls)
        {
            if (c is Clock)
            {
                (c as Clock).Value =
                DateTime.Now.Date.AddHours(ts.Hours).AddMinutes(ts.Minutes);
                string sMinutes = string.Format("{0:00}", ts.Minutes);
                string sHours = string.Format("{0:00}", ts.Hours);
                CurrentQuestion.GivenAnswer = string.Format("{0}:{1}",
                sHours, sMinutes);
                break;
            }
        }
    }

    private void durationSetter_OnDurationChanged(object sender, EventArgs e)
    {
        DurationSetter ds = sender as DurationSetter;
        string sMinutes = string.Format("{0:00}", ds.Minutes);
        string sHours = string.Format("{0:00}", ds.Hours);
        CurrentQuestion.GivenAnswer = string.Format("{0}:{1}", sHours,
        sMinutes);
    }

    private void btnTrueFalse_CheckedChanged(object sender, EventArgs e)
    {
        CurrentQuestion.GivenAnswer = (sender as CheckButton).Tag.ToString();
    }

    private void txEdit_EditValueChanged(object sender, EventArgs e)
    {
        CurrentQuestion.GivenAnswer = (sender as
        TextEdit).EditValue.ToString();
    }

    private void checkButton_CheckedChanged(object sender, EventArgs e)
    {
        CurrentQuestion.GivenAnswer = (sender as CheckButton).Text;
    }

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        CurrentQuestion.IsAnswerCorrect = Convert.ToBoolean((sender as
        CheckButton).Tag);
    }
    #endregion
}
}

```

10.20 Κώδικας - User.cs

```

using System;
using System.Linq;
using System.Data;

namespace Utils
{
    public class User : IDisposable
    {
        public string Username { get; set; }
        public string Password { get; set; }
        public string LastName { get; set; }
        public string FirstName { get; set; }
        public int Level { get; set; }
        public char Sex { get; set; }
        public int Age { get; set; }
        public int ID { get; set; }
        public int SkinID { get; set; }
        public bool IsAdmin { get; set; }

        public User(string username)
        {
            Username = username;
            LoadUserData();
        }

        private void LoadUserData()
        {
            using (UtilsDataSetTableAdapters.UserTableAdapter userTableAdapter =
            new UtilsDataSetTableAdapters.UserTableAdapter())
            {
                UtilsDataSet.UserRow user =
                userTableAdapter.LoadUserDataByUsername(Username).First();
                Level = user.UserLevelID;
                Sex = Convert.ToChar(user.Sex);
                Age = user.Age;
                ID = user.UserID;
                Password = user.Password;
                SkinID = user.SkinID;
                IsAdmin = user.IsAdministrator;
                FirstName = user.FirstName;
                LastName = user.LastName;
            }
        }

        public void SaveUserData()
        {
            using (UtilsDataSetTableAdapters.UserTableAdapter userTableAdapter =
            new UtilsDataSetTableAdapters.UserTableAdapter())

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        {
            UtilsDataSet.UserRow user =
userTableAdapter.LoadUserDataByUsername(Username).First();
            user.UserLevelID = Level;
            user.Sex = Sex.ToString();
            user.Age = Age;
            ID = user.UserID;
            user.Password = Password;
            user.SkinID = SkinID;
            user.IsAdministrator = IsAdmin;
            user.FirstName = FirstName;
            user.LastName = LastName;
            userTableAdapter.Update(user);
        }
    }

    public void CheckLevelPromotion()
    {
        using (UtilsDataSetTableAdapters.StatisticsTableAdapter
statisticsTableAdapter = new UtilsDataSetTableAdapters.StatisticsTableAdapter())
        {
            DataTable dt =
statisticsTableAdapter.GetCurrentLevelStatsByUserIDLevel(ID, new
Settings().EvalTestNumber, Level);
            if (dt.Rows.Count == 0) return;
            if (Convert.ToInt32(dt.Rows[0]["AverageRating"]) > 50)
            {
                if (Level > 0 && Level < 4)
                {
                    Level++;
                    using (UtilsDataSetTableAdapters.UserTableAdapter
userTableAdapter = new UtilsDataSetTableAdapters.UserTableAdapter())
                    {
                        userTableAdapter.Promote(Level, ID);
                    }
                }
            }
            else if (Convert.ToInt32(dt.Rows[0]["AverageRating"]) < 20)
            {
                if (Level > 1 && Level <= 4)
                {
                    Level--;
                    using (UtilsDataSetTableAdapters.UserTableAdapter
userTableAdapter = new UtilsDataSetTableAdapters.UserTableAdapter())
                    {
                        userTableAdapter.Promote(Level, ID);
                    }
                }
            }
        }
    }

    public void Dispose()
    {
        GC.SuppressFinalize(this);
    }
}
}

```


10.21 Κώδικας - VisibilityFunctions.cs

```
using System;
using System.Windows.Forms;
using DevExpress.XtraEditors;

namespace Utils
{
    public class VisibilityFunctions
    {
        public static void ShowFormInPanel(Form f, PanelControl p)
        {
            f.TopLevel = false;
            f.FormBorderStyle = FormBorderStyle.None;
            f.Dock = DockStyle.Fill;
            f.Parent = p;
            f.Show();
        }

        public static void ClearPanel(PanelControl p)
        {
            foreach (Control c in p.Controls)
            {
                c.Dispose();
            }
            p.Controls.Clear();
        }
    }
}
```

10.22 Κώδικας - ErrorTypes.cs

```
using System;

namespace Utils
{
    public enum ErrorTypes
    {
        Empty = 0,
        etGeneral = 1,
        etSmallLargeHand = 2,
        etPastTo = 3,
        etDuration = 4,
        etMinuteHandValue = 5,
        etHourHandValue = 6,
        esSpelling = 7
    };
}
```

10.23 Κώδικας - QuestionTypes.cs

ΤιςΤαc: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```
using System;

namespace Utils
{
    public enum QuestionTypes
    {
        qtNone = 0,
        qtMultipleChoiceClock = 1,
        qtMultipleChoiceText = 2,
        qtSetClock = 3,
        qtSetRange = 4,
        qtRightWrong = 5,
        qtOpenAnswer = 6
    };
}
```

10.24 Κώδικας - Skins.cs

```
using System;
using Utils.EnumExtensions;

namespace Utils
{
    public enum Skins
    {
        [StringValue("Black")]
        Black = 0,
        [StringValue("Blue")]
        Blue = 1,
        [StringValue("Caramel")]
        Caramel = 2,
        [StringValue("Coffee")]
        Coffee = 3,
        [StringValue("Dark Side")]
        DarkSide = 4,
        [StringValue("Darkroom")]
        Darkroom = 5,
        [StringValue("DevExpress Style")]
        DevExpressStyle = 6,
        [StringValue("Foggy")]
        Foggy = 7,
        [StringValue("Glass Oceans")]
        GlassOceans = 8,
        [StringValue("High Contrast")]
        HighContrast = 9,
        [StringValue("iMaginary")]
        iMaginary = 10,
        [StringValue("Lilian")]
        Lilian = 11,
        [StringValue("Liquid Sky")]
        LiquidSky = 12,
        [StringValue("London Liquid Sky")]
        LondonLiquidSky = 13,
        [StringValue("McSkin")]
        McSkin = 14,
        [StringValue("Money Twins")]
    }
}
```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

    MoneyTwins = 15,
    [StringValue("Office 2007 Black")]
    Office2007Black = 16,
    [StringValue("Office 2007 Blue")]
    Office2007Blue = 17,
    [StringValue("Office 2007 Green")]
    Office2007Green = 18,
    [StringValue("Office 2007 Pink")]
    Office2007Pink = 19,
    [StringValue("Office 2007 Silver")]
    Office2007Silver = 20,
    [StringValue("Office 2010 Black")]
    Office2010Black = 21,
    [StringValue("Office 2010 Blue")]
    Office2010Blue = 22,
    [StringValue("Office 2010 Silver")]
    Office2010Silver = 23,
    [StringValue("Pumpkin")]
    Pumpkin = 24,
    [StringValue("Seven")]
    Seven = 25,
    [StringValue("Seven Classic")]
    SevenClassic = 26,
    [StringValue("Sharp")]
    Sharp = 27,
    [StringValue("Sharp Plus")]
    SharpPlus = 28,
    [StringValue("Springtime")]
    Springtime = 29,
    [StringValue("Stardust")]
    Stardust = 30,
    [StringValue("Summer 2008")]
    Summer2008 = 31,
    [StringValue("The Asphalt World")]
    TheAsphaltWorld = 32,
    [StringValue("Valentine")]
    Valentine = 33,
    [StringValue("Xmas 2008 Blue")]
    Xmas2008 = 34
};
}

```

10.25 Κώδικας - TimeZones.cs

```

using System;

namespace Utils
{
    public enum TimeZones
    {
        tzNone = 0,
        tzNight = 1,
        tzMorning = 2,
        tzMidday = 3,
        tzAfternoon = 4
    };
}

```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

10.26 Κώδικας - Clock.vb

```
'Author: Arman Ghazanchyan
'Created date: 06/18/2007
'Last updated: 03/02/2009

<Assembly: CLSCompliant(True)>

#Region " Enumeration "

''' <summary>
''' Represents border-style enumeration for AnalogClock.Clock control.
''' </summary>
<Flags()> _
Public Enum BorderStyles As Integer
    None = 0
    Round = 1
    FixedSingle = 2
    RoundFrame = 4
    RoundFixedSingle = (Round Or RoundFrame)
    FixedSingleRoundFrame = (FixedSingle Or RoundFrame)
End Enum

''' <summary>
''' Represents smoothing mode enumeration for AnalogClock.Clock control.
''' </summary>
Public Enum SmoothMode As Integer
    [Default] = 0
    HighSpeed = 1
    HighQuality = 2
    None = 3
    AntiAlias = 4
End Enum

#End Region

''' <summary>
''' Represents a AnalogClock.Clock control for windows.
''' </summary>
<DebuggerNonUserCode()> _
<System.ComponentModel.DefaultProperty("Name"), _
System.ComponentModel.DefaultEvent("TimeChanged"), _
DebuggerDisplay("DateTime = {DateTime}"), _
System.ComponentModel.Description("Displays analog clock."), _
ToolboxBitmap(GetType(Clock), "clock.bmp"), _
System.ComponentModel.DefaultBindingProperty("UtcOffset")> _
Public Class Clock

#Region " Event Handlers "

    <System.ComponentModel.Category("Property Changed"),
System.ComponentModel.Description("Occurs when the clock's time changed.")> _
    Public Event TimeChanged As System.EventHandler

    <System.ComponentModel.Category("Property Changed"),
System.ComponentModel.Description("Occurs when the clock's UTC changed.")> _
```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

    Public Event UtcChanged As System.EventHandler
    <System.ComponentModel.Category("Property Changed"),
    System.ComponentModel.Description("Occurs when the clock's update changed.")> _
    Public Event ClockRunningChanged As System.EventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when an element property changed.")> _
    Public Event ElementPropertyChanged As
    System.ComponentModel.PropertyChangedEventHandler
    <System.ComponentModel.Category("Property Changed"),
    System.ComponentModel.Description("Occurs when the clock's frame color
    changed.")> _
    Public Event FrameColorChanged As EventHandler
    <System.ComponentModel.Category("Property Changed"),
    System.ComponentModel.Description("Occurs when the clock's border style
    changed.")> _
    Public Event BorderStyleChanged As EventHandler
    <System.ComponentModel.Category("Property Changed"),
    System.ComponentModel.Description("Occurs when the clock's update frequency
    changed.")> _
    Public Event UpdateFrequencyChanged As EventHandler
    <System.ComponentModel.Category("Appearance"),
    System.ComponentModel.Description("Occurs when the clock's background needs
    repainting.")> _
    Public Event BackgroundPaint As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when a small-marker object needs
    repainting.")> _
    Public Event SmallMarkerPainting As EventHandler(Of
    AnalogClock.PaintEventArgs)
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when a small-marker objects had been
    painted.")> _
    Public Event SmallMarkerPainted As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when all small-marker objects had been
    painted.")> _
    Public Event SmallMarkersPainted As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when a big-marker object needs
    repainting.")> _
    Public Event BigMarkerPainting As EventHandler(Of AnalogClock.PaintEventArgs)
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when a big-marker objects had been
    painted.")> _
    Public Event BigMarkerPainted As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when all big-marker objects had been
    painted.")> _
    Public Event BigMarkersPainted As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Appearance"),
    System.ComponentModel.Description("Occurs when the clock's frame needs
    repainting.")> _
    Public Event FramePainting As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Appearance"),
    System.ComponentModel.Description("Occurs when the clock's frame had been
    painted.")> _
    Public Event FramePainted As System.Windows.Forms.PaintEventHandler

```



```

    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when a symbol object needs
    repainting.")> _
    Public Event SymbolPainting As EventHandler(Of AnalogClock.PaintEventArgs)
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when a symbol objects had been
    painted.")> _
    Public Event SymbolPainted As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when all symbol objects had been
    painted.")> _
    Public Event SymbolsPainted As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when a custom symbol painting is
    required.")> _
    Public Event CustomSymbolPainting As PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when the hour-hand object needs
    repainting.")> _
    Public Event HourHandPainting As EventHandler(Of AnalogClock.PaintEventArgs)
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when the hour-hand object had been
    painted.")> _
    Public Event HourHandPainted As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when the minute-hand object needs
    repainting.")> _
    Public Event MinuteHandPainting As EventHandler(Of
    AnalogClock.PaintEventArgs)
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when the minute-hand object had been
    painted.")> _
    Public Event MinuteHandPainted As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when the second-hand object needs
    repainting.")> _
    Public Event SecondHandPainting As EventHandler(Of
    AnalogClock.PaintEventArgs)
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when the second-hand object had been
    painted.")> _
    Public Event SecondHandPainted As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when the center-point object needs
    repainting.")> _
    Public Event CenterPointPainting As EventHandler(Of
    AnalogClock.PaintEventArgs)
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when the center-point object had been
    painted.")> _
    Public Event CenterPointPainted As System.Windows.Forms.PaintEventHandler
    <System.ComponentModel.Category("Elements"),
    System.ComponentModel.Description("Occurs when a custom element's GraphicsPath
    object is required.")> _
    Public Event CustomElementRequest As EventHandler(Of
    AnalogClock.CustomElementEventArgs)

#End Region

```

ΤιςΤαc: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```
#Region " Members "
```

```
Private _borderStyle As AnalogClock.BorderStyles
Private _rect As RectangleF
Private _dateTime As DateTime
Private _frameColor As Color = Color.Black
Private _radius As Single
Private _utcOffset As TimeSpan
Private WithEvents _hourHand As Hand
Private WithEvents _minuteHand As Hand
Private WithEvents _secondHand As Hand
Private WithEvents _centerPoint As Center
Private _bigMarkers(11) As Marker
Private _smallMarkers(59) As Marker
Private _symbols(11) As Symbol
Private _symbolsInitalized As Boolean
Private _bigMarkersInitalized As Boolean
Private _smallMarkersInitalized As Boolean
```

```
#End Region
```

```
#Region " Properties "
```

```
''' <summary>
''' Gets clock's hour-hand object.
''' </summary>
<System.ComponentModel.Category("Elements"),
System.ComponentModel.Description("Represents clock's hour-hand object."), _
System.ComponentModel.DesignerSerializationVisibilityAttribute(System.ComponentMo
del.DesignerSerializationVisibility.Content)> _
Public ReadOnly Property HourHand() As Hand
Get
Return Me._hourHand
End Get
End Property

''' <summary>
''' Gets clock's minute-hand object.
''' </summary>
<System.ComponentModel.Category("Elements"),
System.ComponentModel.Description("Represents clock's minute-hand object."), _
System.ComponentModel.DesignerSerializationVisibilityAttribute(System.ComponentMo
del.DesignerSerializationVisibility.Content)> _
Public ReadOnly Property MinuteHand() As Hand
Get
Return Me._minuteHand
End Get
End Property

''' <summary>
''' Gets clock's second-hand object.
''' </summary>
<System.ComponentModel.Category("Elements"),
System.ComponentModel.Description("Represents clock's second-hand object."), _
System.ComponentModel.DesignerSerializationVisibilityAttribute(System.ComponentMo
del.DesignerSerializationVisibility.Content)> _
```

TicTac: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

Public ReadOnly Property SecondHand() As Hand
    Get
        Return Me._secondHand
    End Get
End Property

''' <summary>
''' Gets clock's center-point object.
''' </summary>
<System.ComponentModel.Category("Elements"),
System.ComponentModel.Description("Represents clock's center-point object."), _
System.ComponentModel.DesignerSerializationVisibilityAttribute(System.ComponentMo
del.DesignerSerializationVisibility.Content)> _
Public ReadOnly Property CenterPoint() As Center
    Get
        Return Me._centerPoint
    End Get
End Property

''' <summary>
''' Gets or sets the clock's big-marker array.
''' </summary>
<System.ComponentModel.Category("Elements"),
System.ComponentModel.Description("Represents an array of clock's big-markers."),
_
System.ComponentModel.Editor(GetType(ElementArrayEditor),
GetType(Drawing.Design.UITypeEditor)), _
System.ComponentModel.DesignerSerializationVisibilityAttribute(System.ComponentMo
del.DesignerSerializationVisibility.Content), _
System.ComponentModel.TypeConverterAttribute(GetType(ElementArrayConverter))>
-
Public Property BigMarkers() As Marker()
    Get
        Return Me._bigMarkers
    End Get
    Set(ByVal bigMarkers As Marker())
        If bigMarkers Is Nothing Then
            Throw New ArgumentNullException("bigMarkers")
        Else
            If bigMarkers.Length <> 12 Then
                Throw New ArgumentOutOfRangeException( _
                    "bigMarkers", bigMarkers, "The clock must have 12 big-
markers!")
            Else
                If Not Me._bigMarkersInitalized OrElse (Me._bigMarkers(0)
 IsNot bigMarkers(0)) Then
                    Me.InitalizeBigMarkers(bigMarkers)
                    Me._bigMarkersInitalized = True
                End If
                Me._bigMarkers = bigMarkers
                Me.Invalidate(False)
            End If
        End If
    End Set
End Property

''' <summary>

```

```

''' Gets or sets the clock's small-marker array.
''' </summary>
<System.ComponentModel.Category("Elements"),
System.ComponentModel.Description("Represents an array of clock's small-
markers."), _
System.ComponentModel.Editor(GetType(ElementArrayEditor),
GetType(Drawing.Design.UITypeEditor)), _

System.ComponentModel.DesignerSerializationVisibilityAttribute(System.ComponentMo
del.DesignerSerializationVisibility.Content), _
System.ComponentModel.TypeConverterAttribute(GetType(ElementArrayConverter))>
-
Public Property SmallMarkers() As Marker()
Get
Return Me._smallMarkers
End Get
Set(ByVal smallMarkers As Marker())
If smallMarkers Is Nothing Then
Throw New ArgumentNullException("smallMarkers")
Else
If smallMarkers.Length <> 60 Then
Throw New ArgumentOutOfRangeException( _
"smallMarkers", smallMarkers, "The clock must have 60 small-
markers!")
Else
If Not Me._smallMarkersInitalized OrElse Me._smallMarkers(0)
IsNot smallMarkers(0) Then
Me.InitalizeSmallMarkers(smallMarkers)
Me._smallMarkersInitalized = True
End If
Me._smallMarkers = smallMarkers
Me.Invalidate(False)
End If
End If
End Set
End Property

''' <summary>
''' Gets or sets the clock's symbol array.
''' </summary>
<System.ComponentModel.Category("Elements"),
System.ComponentModel.Description("Represents an array of clock's symbols."), _
System.ComponentModel.Editor(GetType(ElementArrayEditor),
GetType(Drawing.Design.UITypeEditor)), _

System.ComponentModel.DesignerSerializationVisibilityAttribute(System.ComponentMo
del.DesignerSerializationVisibility.Content), _
System.ComponentModel.TypeConverterAttribute(GetType(ElementArrayConverter))>
-
Public Property Symbols() As Symbol()
Get
Return Me._symbols
End Get
Set(ByVal symbols As Symbol())
If symbols Is Nothing Then
Throw New ArgumentNullException("symbols")
Else
If symbols.Length <> 12 Then
Throw New ArgumentOutOfRangeException( _

```

```

        "symbols", symbols, "The clock must have 12 symbols!")
    Else
        If Not Me._symbolsInitalized OrElse Me._symbols(0) IsNot
symbols(0) Then
            Me.InitalizeSymbols(symbols)
            Me._symbolsInitalized = True
            End If
            Me._symbols = symbols
            Me.Invalidate(False)
        End If
    End If
End Set
End Property

''' <summary>
''' Gets or sets the clock's Coordinated Universal Time (UTC) offset.
''' </summary>
<System.ComponentModel.DefaultValue(GetType(TimeSpan), "00:00:00"), _
System.ComponentModel.Bindable(True), _
System.ComponentModel.Category("Behavior"), _
System.ComponentModel.RefreshPropertiesAttribute(System.ComponentModel.RefreshPro
perties.Repaint), _
System.ComponentModel.Description( _
"Indicates the clock's Coordinated Universal Time (UTC) offset.")> _
Public Property UtcOffset() As TimeSpan
    Get
        Return Me._utcOffset
    End Get
    Set(ByVal utcOffset As TimeSpan)
        If utcOffset < TimeSpan.Parse("-23:59:00") OrElse utcOffset >
TimeSpan.Parse("23:59:00") Then
            Throw New ArgumentOutOfRangeException( _
"utcOffset", utcOffset, "The value must be in the range -23:59:00
to 23:59:00.")
        Else
            If Me._utcOffset <> utcOffset Then
                Me._utcOffset = utcOffset
                Me.UpdateRunningTime()
                Me.OnUtcChanged(New System.EventArgs)
            End If
        End If
    End Set
End Property

''' <summary>
''' Gets or sets the clock's frame color.
''' </summary>
<System.ComponentModel.DefaultValue(GetType(Color), "Black"), _
System.ComponentModel.Bindable(True), _
System.ComponentModel.Category("Appearance"), _
System.ComponentModel.Description("Indicates the clock's frame color.")> _
Public Property FrameColor() As Color
    Get
        Return Me._frameColor
    End Get
    Set(ByVal value As Drawing.Color)
        If Me._frameColor <> value Then
            Me._frameColor = value
        End Set
    End Set
End Property

```



```

        Me.Invalidate(False)
        Me.OnFrameColorChanged(New System.EventArgs)
    End If
End Set
End Property

''' <summary>
''' Gets or sets a value indicating whether the clock is running.
''' </summary>
<System.ComponentModel.DefaultValue(True), _
System.ComponentModel.Bindable(True),
System.ComponentModel.Category("Behavior"), _
System.ComponentModel.Description("Indicates whether the clock is running.")>
-
Public Property Running() As Boolean
    Get
        Return Me.UpdateTimer.Enabled
    End Get
    Set(ByVal value As Boolean)
        If Me.UpdateTimer.Enabled <> value Then
            Me.UpdateTimer.Enabled = value
            Me.UpdateRunningTime()
            Me.OnClockRunningChanged(New System.EventArgs)
        End If
    End Set
End Property

''' <summary>
''' Gets or sets the clock's border style.
''' </summary>
<System.ComponentModel.DefaultValue(GetType(BorderStyles), "None"), _
System.ComponentModel.Bindable(True),
System.ComponentModel.Category("Appearance"), _
System.ComponentModel.Description("Indicates the clock's border style")> _
Public Overloads Property BorderStyle() As AnalogClock.BorderStyles
    Get
        Return Me._borderStyle
    End Get
    Set(ByVal borderStyle As AnalogClock.BorderStyles)
        If Not [Enum].IsDefined(GetType(BorderStyles), borderStyle) Then
            Throw New
System.ComponentModel.InvalidEnumArgumentException("borderStyle", borderStyle,
GetType(BorderStyles))
        Else
            If Me._borderStyle <> borderStyle Then
                Me._borderStyle = borderStyle
                Me.SetClockRegion()
                Me.Refresh()
                Me.OnBorderStyleChanged(New System.EventArgs)
            End If
        End If
    End Set
End Property

''' <summary>
''' Gets or sets the clock's update frequency (in milliseconds).
''' </summary>
<System.ComponentModel.DefaultValue(100), _

```

```

System.ComponentModel.Bindable(True),
System.ComponentModel.Category("Behavior"), _
System.ComponentModel.Description("Indicates the clock's update frequency (in
milliseconds).")> _
    Public Property UpdateFrequency() As Integer
        Get
            Return Me.UpdateTimer.Interval
        End Get
        Set(ByVal updateFrequency As Integer)
            If (updateFrequency > 100 OrElse updateFrequency < 1) AndAlso
Math.IEEEERemainder(100, updateFrequency) <> 0 Then
                Dim sb As New System.Text.StringBuilder
                sb.AppendLine("The value should be in the range 1 to 100 and must
divide 100 without a remainder.")
                sb.AppendLine("Example:")
                sb.AppendLine("1")
                sb.AppendLine("2")
                sb.AppendLine("4")
                sb.AppendLine("5")
                sb.AppendLine("10")
                sb.AppendLine("20")
                sb.AppendLine("25")
                sb.AppendLine("50")
                sb.AppendLine("100")
                Throw New ArgumentOutOfRangeException("updateFrequency",
updateFrequency, sb.ToString)
            Else
                Me.UpdateTimer.Interval = updateFrequency
                Me.OnUpdateFrequencyChanged(New EventArgs)
            End If
        End Set
    End Property

''' <summary>
''' Gets the clock's date and time object.
''' </summary>
<System.ComponentModel.Category("Behavior"),
System.ComponentModel.Description("The clock's date and time.")> _
    Public Property Value() As System.DateTime
        Get
            Return Me._dateTime
        End Get
        Set(ByVal value As System.DateTime)
            Me._dateTime = value
            Me.UpdateRunningTime()
        End Set
    End Property

''' <summary>
''' Gets clock's drawing rectangle.
''' </summary>
<System.ComponentModel.Browsable(False)> _
    Public ReadOnly Property Rectangle() As RectangleF
        Get
            Return Me._rect
        End Get
    End Property

''' <summary>

```

```

''' Gets clock's radius.
''' </summary>
<System.ComponentModel.Browsable(False)> _
Public ReadOnly Property Radius() As Single
    Get
        Return Me._radius
    End Get
End Property

''' <summary>
''' Gets a value that indicates whether the clock has been initialized.
''' </summary>
<System.ComponentModel.Browsable(False)> _
Public ReadOnly Property Initialized() As Boolean
    Get
        Return (Me._bigMarkersInitalized And Me._smallMarkersInitalized And
Me._symbolsInitalized)
    End Get
End Property

#End Region

#Region " Methods "

Sub New()

    ' This call is required by the Windows Form Designer.
    InitializeComponent()
    ' Add any initialization after the InitializeComponent() call.

    Me.SetClockRectangle()
    ' Create hour-hand object.
    Me._hourHand = New Hand("HourHand", Me._radius, 0.65F, 5.0F, 0.2F, 30)
    ' Create minute-hand object.
    Me._minuteHand = New Hand("MinuteHand", Me._radius, 0.8F, 5.0F, 0.2F, 6)
    ' Create second-hand object.
    Me._secondHand = New Hand("SecondHand", Me._radius, 0.9F, 1.0F, 0.2F, 6)
    ' Create center-point object.
    Me._centerPoint = New Center(Me._radius, 0.03F)
    ' Update the graphics paths of the elements.
    Me.HourHand.UpdateBasePath()
    Me.MinuteHand.UpdateBasePath()
    Me.SecondHand.UpdateBasePath()
    Me.CenterPoint.UpdateBasePath()
End Sub

''' <summary>
''' Calculates and returns a point in the clock's cordinates.
''' </summary>
''' <param name="startPoint">A starting point in the clock's
''' area from which the point should be calculated.</param>
''' <param name="angle">An angle from x-axis (in radins).</param>
''' <param name="radius">The distans from the starting point to the
point.</param>
Public Shared Function GetPoint(ByVal startPoint As PointF, ByVal angle As
Double, ByVal radius As Single) As PointF
    Dim y, x As Single
    y = CSng(startPoint.Y - (Math.Sin(angle) * radius))
    x = CSng(startPoint.X + (Math.Cos(angle) * radius))

```

```

    Return New PointF(x, y)
End Function

''' <summary>
''' Updates the clock's hands.
''' </summary>
Private Sub UpdateRunningTime()
    If Me.Running
        Me._dateTime = Date.UtcNow.AddTicks(Me._utcOffset.Ticks)
    End If
    Static Dim sec As Integer = -1
    If Me._secondHand.Motion = HandMotion.Step Then
        Me._secondHand.Value = Me._dateTime.Second
    Else
        Me._secondHand.Value = Me._dateTime.Second + Me._dateTime.Millisecond
/ 1000
    End If
    If Me.MinuteHand.Motion = HandMotion.Step Then
        Me._minuteHand.Value = Me._dateTime.Minute
    Else
        Me._minuteHand.Value = Me._dateTime.Minute + Me._dateTime.Second / 60
    End If
    Me._hourHand.Value = Me._dateTime.Hour + Me._dateTime.Minute / 60
    If sec <> Me._dateTime.Second Then
        sec = Me._dateTime.Second
        Me.OnTimeChanged(New System.EventArgs)
    End If
End Sub

''' <summary>
''' Sets the clock's rectangle and radius.
''' </summary>
Private Sub SetClockRectangle()
    If Me.Width < Me.Height Then
        Me._rect = New RectangleF(0.0F, 0.0F, Me.Width, Me.Width)
    ElseIf Me.Width > Me.Height Then
        Me._rect = New RectangleF(0.0F, 0.0F, Me.Height, Me.Height)
    Else
        Me._rect = New RectangleF(0.0F, 0.0F, Me.Width, Me.Width)
    End If
    Me._radius = Me._rect.Width / 2.0F
End Sub

''' <summary>
''' Sets the clock's region.
''' </summary>
Private Sub SetClockRegion()
    If (Me._borderStyle And BorderStyles.Round) <> 0 Then
        Dim pa As New Drawing.Drawing2D.GraphicsPath
        pa.AddEllipse(Me._rect.X - 1.0F, Me._rect.Y - 1.0F, Me._rect.Width +
1.0F, Me._rect.Height + 1.0F)
        pa.CloseAllFigures()
        Me.Region = New Region(pa)
        pa.Dispose()
    Else
        Dim pa As New Drawing.Drawing2D.GraphicsPath
        pa.AddRectangle(Me._rect)
        pa.CloseAllFigures()
        Me.Region = New Region(pa)
    End If
End Sub

```

```

        pa.Dispose()
    End If
End Sub

''' <summary>
''' Initializes the clock's big-markers by adding event handlers and updating
the base-path.
''' </summary>
''' <param name="markerArray">An array of AnalogClock.Marker object.</param>
Private Sub InitializeBigMarkers(ByVal markerArray As Marker())
    For Each item As Marker In markerArray
        AddHandler item.PropertyChanged, AddressOf Me.Element_PropertyChanged
        AddHandler item.CustomElementRequest, AddressOf
Me.Elements_CustomElementRequest
        AddHandler item.ElementPainting, AddressOf
Me.BigMarkers_ElementPainting
        AddHandler item.ElementPainted, AddressOf
Me.BigMarkers_ElementPainted
        item.UpdateBasePath()
    Next
End Sub

''' <summary>
''' Initializes the clock's small-markers by adding event handlers and
updating the base-path.
''' </summary>
''' <param name="markerArray">An array of AnalogClock.Marker object.</param>
Private Sub InitializeSmallMarkers(ByVal markerArray As Marker())
    For Each item As Marker In markerArray
        AddHandler item.PropertyChanged, AddressOf Me.Element_PropertyChanged
        AddHandler item.CustomElementRequest, AddressOf
Me.Elements_CustomElementRequest
        AddHandler item.ElementPainting, AddressOf
Me.SmallMarkers_ElementPainting
        AddHandler item.ElementPainted, AddressOf
Me.SmallMarkers_ElementPainted
        item.UpdateBasePath()
    Next
End Sub

''' <summary>
''' Initializes the clock's symbols by adding event handlers.
''' </summary>
''' <param name="symbolArray">An array of AnalogClock.Symbol object.</param>
Private Sub InitializeSymbols(ByVal symbolArray As Symbol())
    For Each item As Symbol In symbolArray
        AddHandler item.PropertyChanged, AddressOf Me.Element_PropertyChanged
        AddHandler item.CustomSymbolPainting, AddressOf
Me.Symbols_ElementCustomPainting
        AddHandler item.ElementPainting, AddressOf Me.Symbols_ElementPainting
        AddHandler item.ElementPainted, AddressOf Me.Symbols_ElementPainted
    Next
End Sub

#End Region

#Region " Events "

```



```

Private Sub Clock_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    If Not Me._smallMarkersInitalized Then
        ' Create Small-Markers.
        For i As Integer = 0 To 59
            Dim angle As Integer = 90 - ((i) * 6)
            If angle < 0 Then
                angle += 360
            End If
            Dim item As New Marker("SmallMarker" & angle, Me._radius, 1.0F, _
                angle, Color.Black, 1.0F, 0.03F, True)
            Me._smallMarkers(i) = item
            AddHandler item.PropertyChanged, AddressOf
Me.Element_PropertyChanged
            AddHandler item.CustomElementRequest, AddressOf
Me.Elements_CustomElementRequest
            AddHandler item.ElementPainting, AddressOf
Me.SmallMarkers_ElementPainting
            AddHandler item.ElementPainted, AddressOf
Me.SmallMarkers_ElementPainted
            item.UpdateBasePath()
        Next
        Me._smallMarkersInitalized = True
    End If
    If Not Me._bigMarkersInitalized Then
        ' Create Big-Markers.
        For i As Integer = 0 To 11
            Dim angle As Integer = 90 - ((i) * 30)
            If angle < 0 Then
                angle += 360
            End If
            Dim item As New Marker("BigMarker" & angle, Me._radius, 1.0F, _
                angle, Color.Black, 1.0F, 0.06F)
            Me._bigMarkers(i) = item
            AddHandler item.PropertyChanged, AddressOf
Me.Element_PropertyChanged
            AddHandler item.CustomElementRequest, AddressOf
Me.Elements_CustomElementRequest
            AddHandler item.ElementPainting, AddressOf
Me.BigMarkers_ElementPainting
            AddHandler item.ElementPainted, AddressOf
Me.BigMarkers_ElementPainted
            item.UpdateBasePath()
        Next
        Me._bigMarkersInitalized = True
    End If
    If Not Me._symbolsInitalized Then
        ' Create Symbols.
        For i As Integer = 0 To 11
            Dim angle As Integer = 90 - ((i) * 30)
            If angle < 0 Then
                angle += 360
            End If
            Dim item As New Symbol("Symbol" & angle, Me._radius, 0.82F,
angle, i, New Point(1, 1))
            Me._symbols(i) = item
            AddHandler item.PropertyChanged, AddressOf
Me.Element_PropertyChanged
        Next
    End If
End Sub

```

```

        AddHandler item.CustomSymbolPainting, AddressOf
Me.Symbols_ElementCustomPainting
        AddHandler item.ElementPainting, AddressOf
Me.Symbols_ElementPainting
        AddHandler item.ElementPainted, AddressOf
Me.Symbols_ElementPainted
        Next
        Me._symbolsInitalized = True
    End If

    ' Update the hands' values.
    Me.UpdateRunningTime()
End Sub

Private Sub UpdateTimer_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles UpdateTimer.Tick
    Me.UpdateRunningTime()
End Sub

Private Sub Clock1_EnabledChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.EnabledChanged
    Me.Running = Me.Enabled
End Sub

Private Sub Elements_CustomElementRequest(ByVal sender As Object, ByVal e As
CustomElementEventArgs) _
Handles _centerPoint.CustomElementRequest, _hourHand.CustomElementRequest, _
_minuteHand.CustomElementRequest, _secondHand.CustomElementRequest

    RaiseEvent CustomElementRequest(sender, e)
End Sub

Private Sub _hourHand_ElementPainted(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles _hourHand.ElementPainted
    RaiseEvent HourHandPainted(sender, e)
End Sub

Private Sub _hourHand_ElementPainting(ByVal sender As Object, ByVal e As
PaintEventArgs) Handles _hourHand.ElementPainting
    RaiseEvent HourHandPainting(sender, e)
End Sub

Private Sub _minuteHand_ElementPainted(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles _minuteHand.ElementPainted
    RaiseEvent MinuteHandPainted(sender, e)
End Sub

Private Sub _minuteHand_ElementPainting(ByVal sender As Object, ByVal e As
PaintEventArgs) Handles _minuteHand.ElementPainting
    RaiseEvent MinuteHandPainting(sender, e)
End Sub

Private Sub _secondHand_ElementPainted(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles _secondHand.ElementPainted
    RaiseEvent SecondHandPainted(sender, e)
End Sub

Private Sub _secondHand_ElementPainting(ByVal sender As Object, ByVal e As
PaintEventArgs) Handles _secondHand.ElementPainting

```

ΤιςΤαc: Εκπαιδευτική εφαρμογή εκμάθησης της ώρας με μοντελοποίηση χρηστών σε προσαρμοστικό περιβάλλον

```

        RaiseEvent SecondHandPainting(sender, e)
    End Sub

    Private Sub _centerPoint_ElementPainted(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles _centerPoint.ElementPainted
        RaiseEvent CenterPointPainted(sender, e)
    End Sub

    Private Sub _centerPoint_ElementPainting(ByVal sender As Object, ByVal e As
PaintEventArgs) Handles _centerPoint.ElementPainting
        RaiseEvent CenterPointPainting(sender, e)
    End Sub

    Private Sub BigMarkers_ElementPainting(ByVal sender As Object, ByVal e As
PaintEventArgs)
        RaiseEvent BigMarkerPainting(sender, e)
    End Sub

    Private Sub BigMarkers_ElementPainted(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs)
        RaiseEvent BigMarkerPainted(sender, e)
    End Sub

    Private Sub SmallMarkers_ElementPainting(ByVal sender As Object, ByVal e As
PaintEventArgs)
        RaiseEvent SmallMarkerPainting(sender, e)
    End Sub

    Private Sub SmallMarkers_ElementPainted(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs)
        RaiseEvent SmallMarkerPainted(sender, e)
    End Sub

    Private Sub Symbols_ElementPainting(ByVal sender As Object, ByVal e As
PaintEventArgs)
        RaiseEvent SymbolPainting(sender, e)
    End Sub

    Private Sub Symbols_ElementPainted(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs)
        RaiseEvent SymbolPainted(sender, e)
    End Sub

    Private Sub Symbols_ElementCustomPainting(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs)
        RaiseEvent CustomSymbolPainting(sender, e)
    End Sub

    Private Sub Element_PropertyChanged(ByVal sender As Object, ByVal e As
System.ComponentModel.PropertyChangedEventArgs) Handles _
_hourHand.PropertyChanged, _minuteHand.PropertyChanged,
_secondHand.PropertyChanged, _centerPoint.PropertyChanged

        Me.Invalidate(False)
        RaiseEvent ElementPropertyChanged(sender, e)
    End Sub

#End Region

```

```

#Region " On Event "

''' <summary>
''' Raises the System.Windows.Forms.Control.Paint event.
''' </summary>
Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)

    ' Paint clock's background.
    Me.OnBackgroundPaint(e)

    ' Paint all SmallMarkers
    For Each m As Marker In Me._smallMarkers
        m.Paint(e)
    Next
    Me.OnSmallMarkersPainted(e)

    ' Paint all BigMarkers
    For Each m As Marker In Me._bigMarkers
        m.Paint(e)
    Next
    Me.OnBigMarkersPainted(e)

    ' Paint all Symbols
    For Each s As Symbol In Me._symbols
        s.Paint(e)
    Next
    Me.OnSymbolsPainted(e)

    ' Paint the Hour-Hand
    Me._hourHand.Paint(e)

    ' Paint the Minute-Hand
    Me._minuteHand.Paint(e)

    ' Paint the Second-Hand
    Me._secondHand.Paint(e)

    'Paint the Center-Point
    Me._centerPoint.Paint(e)

    ' Paint the clock's frame.
    Me.OnFramePainting(e)

    ' Call the base paint event.
    MyBase.OnPaint(e)
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.BackgroundPaint event.
''' </summary>
''' <param name="e">A System.Windows.Forms.PaintEventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnBackgroundPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
    RaiseEvent BackgroundPaint(Me, e)
End Sub

''' <summary>

```

```

''' Raises the AnalogClock.Clock.SmallMarkersPainted event.
''' </summary>
''' <param name="e">A System.Windows.Forms.PaintEventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnSmallMarkersPainted(ByVal e As
System.Windows.Forms.PaintEventArgs)
    RaiseEvent SmallMarkersPainted(Me, e)
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.BigMarkersPainted event.
''' </summary>
''' <param name="e">A System.Windows.Forms.PaintEventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnBigMarkersPainted(ByVal e As
System.Windows.Forms.PaintEventArgs)
    RaiseEvent BigMarkersPainted(Me, e)
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.SymbolssPainted event.
''' </summary>
''' <param name="e">A System.Windows.Forms.PaintEventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnSymbolsPainted(ByVal e As
System.Windows.Forms.PaintEventArgs)
    RaiseEvent SymbolsPainted(Me, e)
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.FramePainting event.
''' </summary>
''' <param name="e">A System.Windows.Forms.PaintEventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnFramePainting(ByVal e As
System.Windows.Forms.PaintEventArgs)
    RaiseEvent FramePainting(Me, e)
    Dim p As New Drawing.Pen(Me._frameColor)
    'Draw circle
    If (Me._borderStyle And BorderStyles.RoundFrame) <> 0 Then
        e.Graphics.DrawEllipse( _
            p, Me._rect.X, Me._rect.Y, Me._rect.Width - 1.0F, Me._rect.Height -
1.0F)
    End If
    'Draw Square
    If (Me._borderStyle And BorderStyles.FixedSingle) <> 0 Then
        e.Graphics.DrawRectangle( _
            p, Me._rect.X, Me._rect.Y, Me._rect.Width - 1.0F, Me._rect.Height -
1.0F)
    End If
    p.Dispose()
    Me.OnFramePainted(e)
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.FramePainted event.
''' </summary>
''' <param name="e">A System.Windows.Forms.PaintEventArgs
''' that contains the event data.</param>

```



```

Protected Overridable Sub OnFramePainted(ByVal e As
System.Windows.Forms.PaintEventArgs)
    RaiseEvent FramePainted(Me, e)
End Sub

''' <summary>
''' Raises the System.Windows.Forms.Control.Resize event.
''' </summary>
''' <param name="e">A System.EventArgs
''' that contains the event data.</param>
Protected Overrides Sub OnResize(ByVal e As System.EventArgs)
    Me.SetClockRectangle()
    Me.SetClockRegion()
    Me._centerPoint.ClockRadius = Me._radius
    Me._hourHand.ClockRadius = Me._radius
    Me._minuteHand.ClockRadius = Me._radius
    Me._secondHand.ClockRadius = Me._radius
    If Me._bigMarkersInitalized Then
        For Each item As Marker In Me._bigMarkers
            item.ClockRadius = Me._radius
        Next
    End If
    If Me._symbolsInitalized Then
        For Each item As Marker In Me._smallMarkers
            item.ClockRadius = Me._radius
        Next
    End If
    If Me._symbolsInitalized Then
        For Each item As Symbol In Me._symbols
            item.ClockRadius = Me._radius
        Next
    End If
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.TimeChanged event.
''' </summary>
''' <param name="e">A System.EventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnTimeChanged(ByVal e As System.EventArgs)
    RaiseEvent TimeChanged(Me, e)
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.UtcChanged event.
''' </summary>
''' <param name="e">A System.EventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnUtcChanged(ByVal e As System.EventArgs)
    RaiseEvent UtcChanged(Me, e)
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.FrameColorChanged event.
''' </summary>
''' <param name="e">A System.EventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnFrameColorChanged(ByVal e As System.EventArgs)
    RaiseEvent FrameColorChanged(Me, e)

```

```
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.BorderStyleChanged event.
''' </summary>
''' <param name="e">A System.EventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnBorderStyleChanged(ByVal e As System.EventArgs)
    RaiseEvent BorderStyleChanged(Me, e)
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.UpdateFrequencyChanged event.
''' </summary>
''' <param name="e">A System.EventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnUpdateFrequencyChanged(ByVal e As
System.EventArgs)
    RaiseEvent UpdateFrequencyChanged(Me, e)
End Sub

''' <summary>
''' Raises the AnalogClock.Clock.ClockRunningChanged event.
''' </summary>
''' <param name="e">A System.EventArgs
''' that contains the event data.</param>
Protected Overridable Sub OnClockRunningChanged(ByVal e As System.EventArgs)
    RaiseEvent ClockRunningChanged(Me, e)
End Sub

#End Region

End Class
```