



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

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	(ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΥΠΟΣΤΗΡΙΞΗΣ ΜΑΘΗΜΑΤΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ SCRATCH ΓΙΑ ΜΑΘΗΤΕΣ ΜΙΚΡΗΣ ΗΛΙΚΙΑΣ, ΜΕ ΜΙΑ ΠΛΑΤΦΟΡΜΑ ΤΗΛΕ-ΕΚΠΑΙΔΕΥΣΗΣ ΚΑΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ ΧΡΗΣΤΩΝ ΜΕΣΩ MOODLE) (EVALUATING THE SUPPORT OF COMPUTER PROGRAMMING SCRATCH LESSONS FOR YOUTH STUDENTS, VIA AN E-LEARNING USER MODELLING PLATFORM USING MOODLE)
Όνοματεπώνυμο Φοιτητή	ΚΑΤΣΙΝΑ ΠΑΝΑΓΙΩΤΑ
Πατρώνυμο	ΦΩΤΙΟΣ
Αριθμός Μητρώου	ΜΠΣΠ / 13047
Επιβλέπων	ΜΑΡΙΑ ΒΙΡΒΟΥ, ΚΑΘΗΓΗΤΡΙΑ

Τριμελής Εξεταστική Επιτροπή

Μαρία Βίββου
Καθηγήτρια

Τσιχριντζής Γεώργιος
Καθηγητής

Ευθύμιος Αλέπης
Επίκουρος

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή πραγματεύεται την μελέτη και αξιολόγηση μίας πλατφόρμας τηλε-εκπαίδευσης και μοντελοποίησης μαθητών, για την υποστήριξη μαθημάτων προγραμματισμού σε μαθητές μικρής ηλικίας και πιο συγκεκριμένα ηλικίας 10 με 11 χρονών. Τα μαθήματα προγραμματισμού σε μαθητές της πέμπτης τάξης του δημοτικού στα δημοτικά σχολεία Ενιαίου Αναμορφωμένου Εκπαιδευτικού Προγράμματος (Ε.Α.Ε.Π.), συνήθως περιλαμβάνουν την σταδιακή και με αργούς ρυθμούς επαφή των μαθητών με τις πρώτες και βασικές έννοιες του προγραμματισμού και των γλωσσών προγραμματισμού. Αυτό επιτυγχάνεται με την ενασχόληση των μαθητών με ειδικά περιβάλλοντα προγραμματισμού για μικρές ηλικίες (8-16) όπως η εφαρμογή Scratch που έχει δημιουργηθεί από το γνωστό πανεπιστήμιο MIT στις ΗΠΑ. Το περιβάλλον αυτό είναι από τα πιο συνήθη σε προτίμηση και χρήση από τους καθηγητές για τέτοιες δραστηριότητες στο δημοτικό και σε παιδιά κατάλληλης ηλικίας. Η μελέτη αυτής της μεταπτυχιακής διατριβής περιλαμβάνει ένα διαδικτυακό σύστημα τηλε-εκπαίδευσης δημιουργημένο με την πλατφόρμα Moodle, την δημιουργία χρηστών-μαθητών που χρησιμοποιούν το σύστημα ξεχωριστά για τον κάθε μαθητή, και την παρουσίαση-εκτέλεση-καταγραφή τριών μαθημάτων (courses) δημιουργημένα (θεωρία-παραδείγματα-τεστ) σε κατάλληλη μορφή (SCORM πακέτα) ώστε να εκτελούνται μέσω Moodle. Κατά την παρουσίαση-εκτέλεση των μαθημάτων από τους μαθητές-χρήστες του συστήματος, καταχωρούνται τα αποτελέσματα των απαντήσεων του μαθητή στον λογαριασμό-μοντέλο του μαθητή, για περαιτέρω αξιολόγηση από τον καθηγητή του. Τέλος γίνεται μία προσπάθεια σύγκρισης και αξιολόγησης των αποτελεσμάτων των μαθημάτων που έγιναν μόνο με εκπαίδευση στην τάξη, με αυτά που έγιναν με συνδυασμό εκπαίδευσης στην τάξη και τηλε-εκπαίδευσης. Τα συμπεράσματα της έρευνας είναι εξαιρετικά ενδιαφέροντα και αποτελούν σημαντικό βήμα στήριξης για μελλοντική συνέχεια αυτής της έρευνας.

Abstract

The present msc thesis discusses the study and evaluation of an e-learning user modeling platform for students, concerning the support of computer programming lessons for young students of ages 10 to 11 years old. Computer programming lessons for students of the fifth grade of primary schools in Greece that comply with the United Reformed Educational Program (U.R.E.P), usually include the gradual and slow-paced contact of the students with the initial and basic concerns of programming and programming languages. This is achieved with the occupation of the students with special programming environments created for young students (8-16) like the application Scratch which has been created by the famous MIT University of the USA. This environment is one of the most popular environments in use by computer science teachers for such activities in primary schools and for students of the appropriate age. The study of this msc thesis includes a web based e-learning system created with Moodle platform, the creation of individual student-users that use the system, and the presentation-execution-report of three courses created in special SCORM packages form (theory-examples-test), able to work through the Moodle site-platform. During the presentation-execution of the courses by the student-users of the system, the results of the student's answers are being reported in his account-user model, for further consideration and evaluation by his teacher. Finally there exists a comparison and evaluation of the reports of the lessons which took place thoroughly in the classroom, with the lessons that took place combining classroom lessons and e-learning lessons. The conclusions of the study are of high interest and compose basis for future continuation of this study.

1. Εισαγωγή – Σύνομη Περιγραφή Προβλήματος/Αντικειμένου

1.1 Προγραμματισμός για μαθητές μικρής ηλικίας

Τις τελευταίες δεκαετίες έχει αναγνωριστεί στην εκπαιδευτική κοινότητα η αξία της αλγοριθμικής σκέψης. Η αλγοριθμική έχει εξελιχθεί σε μία εξαιρετικά σημαντική περιοχή γνώσης για την εκπαίδευση, καθώς αναδεικνύει την ισχύ και την αποτελεσματικότητα του αναλυτικού τρόπου ανθρώπινης σκέψης για την επίλυση προβλημάτων. Η κατανόηση της δυναμικής που χαρακτηρίζει την αλγοριθμική οδηγεί σήμερα τα εκπαιδευτικά συστήματα των τεχνολογικά προηγμένων χωρών, στο να καλλιεργούν την αλγοριθμική σκέψη των μαθητών από την πρωτοβάθμια εκπαίδευση.

Σύμφωνα με αρκετούς ερευνητές, η υιοθέτηση της κλασικής προσέγγισης διδασκαλίας του προγραμματισμού αποτελεί έναν από τους σημαντικότερους παράγοντες που καθιστά την εκμάθηση του προγραμματισμού δύσκολη. Κάποια από τα σημαντικότερα προβλήματα που αντιμετωπίζουν οι αρχάριοι προγραμματιστές που διδάσκονται τις αρχές του προγραμματισμού με την κλασική προσέγγιση, όπως προκύπτει από εμπειρικές μελέτες ερευνητών, είναι τα εξής:

- Οι γλώσσες προγραμματισμού γενικού σκοπού διαθέτουν, κατά κανόνα, ένα μεγάλο ρεπερτόριο εντολών και είναι πολύπλοκες. Το υλικό που πρέπει να καλύψει ο μαθητής, το οποίο περιλαμβάνει το θεωρητικό πλαίσιο μιας γλώσσας προγραμματισμού σε συνδυασμό με τις βασικές αρχές του προγραμματισμού, είναι πολύ μεγάλο. Το γεγονός αυτό έχει ως αποτέλεσμα ο μαθητής να δυσκολεύεται να αποκτήσει το απαιτούμενο υπόβαθρο. (Brusilovsky et al. 1997, Ruckert & Halpern 1993).
- Η προσοχή των μαθητών επικεντρώνεται στην εκμάθηση των συντακτικών κανόνων των γλωσσών προγραμματισμού γενικού σκοπού που είναι αρκετά πολύπλοκοι, και έτσι αφιερώνουν πολύ χρόνο στην ίδια την γλώσσα προγραμματισμού, και όχι στην ανάπτυξη ικανοτήτων επίλυσης προβλημάτων μέσω της αλγοριθμικής σκέψης (Brusilovsky et al. 1997, Ziegler & Crews 1999).
- Το προγραμματιστικό περιβάλλον δεν παρέχει δυνατότητες οπτικού προγραμματισμού και έτσι ο μαθητής δεν έχει στήριξη στην κατανόηση των βασικών ενεργειών και δομών ελέγχου. Δεν μπορεί να δει την διαδικασία εκτέλεσης του προγράμματος βήμα-βήμα και το αντιμετωπίζει απλά ως μία διαδικασία εισόδου-εξόδου.

Λόγω των παραπάνω δυσκολιών που παρουσιάζουν οι εμπορικοί μεταγλωττιστές που απευθύνονται οι περισσότεροι σε επαγγελματίες και όχι σε μαθητές, αναπτύχθηκαν διάφορα ειδικά προγραμματιστικά εργαλεία για εκπαιδευτικούς σκοπούς εκμάθησης των βασικών αρχών προγραμματισμού, και ειδικότερα για μαθητές μικρότερων ηλικιών.

Στην Ελλάδα, τα Προγράμματα Σπουδών (Υπουργείο Παιδείας & Θρ. 2014) για τα εισαγωγικά μαθήματα προγραμματισμού, τόσο στην πρωτοβάθμια, όσο και στη δευτεροβάθμια εκπαίδευση, συνιστούν, ιδιαίτερα στους μικρότερους μαθητές, την αξιοποίηση προγραμματιστικών μικρόκοσμων, δηλαδή εργαλείων και περιβαλλόντων οπτικού προγραμματισμού κατάλληλων για την εκμάθηση του προγραμματισμού σε μικρές ηλικίες, όπως είναι το Scratch (MIT Media Lab 2002), το Microworlds Pro (LCSI 1999) και άλλα.

Η βασική ιδέα των μικρόκοσμων και των μικρογλωσσών προγραμματισμού είναι η δημιουργία μιας γλώσσας προγραμματισμού που αποτελείται από ένα περιορισμένο ρεπερτόριο εντολών με απλή σύνταξη και σημασία. Οι μικρόκοσμοι είναι μικρά, αλλά σχετικά ολοκληρωμένα υποσύνολα αναπαράστασης του "πραγματικού" περιβάλλοντος, αναπαριστούν ένα μέρος του θεωρητικού αυτού "κόσμου" και μπορούν να κατανοηθούν με ένα τρόπο εμποτιστικό, βιωματικό και εξερευνητικό.

Ο μαθητής μαθαίνει τις βασικές έννοιες του προγραμματισμού ελέγχοντας ένα αντικείμενο πρωταγωνιστή (π.χ. μία γάτα ή μία χελώνα) που ζει μέσα στον μικρόκοσμο. Αναπτύσσει βήμα

προς βήμα τα προγράμματα, τα εκτελεί και παρακολουθεί το αποτέλεσμα εκτέλεσης της κάθε εντολής μέσα από την οπτικοποίηση της εκτέλεσης που προσφέρει ο μικρόκοσμος (Xinogalos et al. 2006). Αυτό έχει σαν αποτέλεσμα να γίνονται εύκολα αντιληπτά τα τυχόν λάθη του προγραμματιστή κυρίως στις νεότερες ηλικίες.

Το βασικότερο πλεονέκτημα των προγραμματιστικών μικρόκοσμων είναι ότι προκαλούν το ενδιαφέρον των μαθητών γιατί το περιβάλλον τους βασίζεται σε κάποιο φυσικό μοντέλο, που είναι ήδη γνωστό στους μαθητές, και η εμφάνιση του είναι σαν παιχνίδι. Λόγω των προηγούμενων, οι μαθητές εξ αρχής αντιμετωπίζουν την ενασχόληση με το μάθημα, που θα τους βοηθήσει στην κατανόηση των βασικών αρχών προγραμματισμού, θετικά και με καλή διάθεση.

Παρόλα τα παραπάνω, η διδασκαλία των βασικών αρχών προγραμματισμού με την βοήθεια μικρογλωσσών προγραμματισμού και προγραμματιστικών μικρόκοσμων αντιμετωπίζει κάποιες δυσκολίες στην εφαρμογή της. Ο διαφορετικός ρυθμός εκμάθησης των βασικών λειτουργιών του προγράμματος, η διαφορετική ευχέρεια χρήσης του ηλεκτρονικού υπολογιστή και του προγραμματιστικού περιβάλλοντος του μικρόκοσμου, ο αριθμός των διδακτικών ωρών που είναι δύο την εβδομάδα, αλλά και οι οδηγίες του αναλυτικού προγράμματος σπουδών (10-12 ώρες συνολικά) για την ενότητα του "προγραμματίζω με τον υπολογιστή", είναι κάποιες από αυτές. Επιπρόσθετα αυτές οι διδακτικές ώρες μπορεί να πραγματοποιηθούν σε διάστημα περίπου 2 μηνών ανάλογα με τις ώρες που θα τύχει να χαθούν, κάτι που μειώνει την επαφή των μαθητών μικρής ηλικίας με τις έννοιες που πραγματεύεται το μάθημα.

Λόγω των δυσκολιών που μόλις περιγράψαμε γίνεται εύκολα κατανοητή η ανάγκη ύπαρξης κάποιων μαθημάτων από αυτά που περιλαμβάνει η διδασκαλία, σε μορφή ψηφιακή ως ολοκληρωμένα ηλεκτρονικά διδακτικά μαθήματα, τα οποία να είναι διαθέσιμα διαδικτυακά, για χρήση με ασύγχρονη εξ αποστάσεως εκπαίδευση ή αλλιώς τηλε-εκπαίδευση. Με αυτό τον τρόπο οι μαθητές θα μπορούν να επαναλαμβάνουν κάποια από τα μαθήματα από το σπίτι τους σε χρόνο και με το ρυθμό που επιθυμούν οι ίδιοι, να παγιώνουν καλύτερα την γνώση τους και να δοκιμάζονται σε ερωτήσεις για αυτά. Ο καθηγητής από την πλευρά του θα μπορεί μέσω της πλατφόρμας τηλε-εκπαίδευσης να ενημερώνεται για το ποιοι μαθητές ασχολήθηκαν περαιτέρω με τα διδασκόμενα μαθήματα, καθώς και την επίδοσή τους στην κατανόηση αυτών.

1.2 Εξ αποστάσεως εκπαίδευση – τηλεεκπαίδευση – ηλεκτρονική μάθηση

Εξ αποστάσεως εκπαίδευση είναι η υποβοηθούμενη από τα μέσα επικοινωνίας εκπαίδευση (παραδοσιακό ταχυδρομείο, email, ραδιόφωνο, τηλεόραση, υπολογιστές, τηλεδιάσκεψη και άλλα) με μικρή ή καθόλου διαπροσωπική ή σε τάξη επαφή μεταξύ εκπαιδευτή και εκπαιδευόμενου.

Τα τελευταία χρόνια, η εξ αποστάσεως εκπαίδευση ή αλλιώς τηλε-εκπαίδευση (distance learning or e-learning) υλοποιείται σχεδόν αποκλειστικά με την υποστήριξη του υπολογιστή, και πιο συγκεκριμένα σε περιβάλλον διαδικτύου. Για το λόγο αυτό, τείνει να είναι σχεδόν ταυτόσημη με έννοιες όπως, ηλεκτρονική μάθηση (e-learning), μάθηση υποβοηθούμενη από υπολογιστή (computer assisted learning), και μάθηση μέσω διαδικτύου (online learning).

Η τηλε-εκπαίδευση μπορεί να διαχωριστεί σε *σύγχρονη* και *ασύγχρονη*. Στη *σύγχρονη* τηλε-εκπαίδευση, η διαδικασία της διδασκαλίας και της μάθησης γίνονται ταυτόχρονα. Ο εκπαιδευτής παραδίδει το μάθημα σε ζωντανή σύνδεση, όχι απαραίτητα αμφίδρομη, και ο εκπαιδευόμενος, αν και βρίσκεται σε διαφορετικό τόπο, παρακολουθεί το μάθημα στον ίδιο χρόνο. Η τηλε-εκπαίδευση με την υποστήριξη της τηλεδιάσκεψης είναι χαρακτηριστικό παράδειγμα.

Στην *ασύγχρονη* τηλε-εκπαίδευση, που είναι και πιο διαδεδομένη, ο εκπαιδευόμενος μαθαίνει όχι μόνο σε διαφορετικό χώρο από τον εκπαιδευτή, αλλά και σε διαφορετικό χρόνο από τη διαδικασία της παράδοσης ή δημιουργίας του μαθήματος. Χαρακτηριστικό παράδειγμα είναι τα μαθήματα τηλε-εκπαίδευσης που χρησιμοποιούν διαδικτυακά συστήματα διαχείρισης του εκπαιδευτικού υλικού ή/και της μάθησης (Content Management Systems CMS ή Learning Management Systems, LMS).

Ένα σύστημα CMS επιτρέπει στον διδάσκοντα να δημιουργήσει ένα δικτυακό μάθημα όπου μπορούν να «ανέβουν» (upload) κείμενα σε έναν από τους συνήθεις τύπους (όπως κειμένου,

παρουσίασης, ήχου) κλπ. χωρίς να χρειάζεται να μετατρέπονται σε web format δηλαδή να μετατραπούν σε υλικό για ιστοσελίδες (ενσωμάτωση στον κώδικα της ιστοσελίδας). Εκεί οι μαθητές μπορούν ελεύθερα να βρουν το υλικό και να το μελετήσουν.

Ένα σύστημα LMS διανέμει και διαχειρίζεται όλες τις μαθησιακές ανάγκες. Τα διαδικτυακά μαθήματα είναι πια μέρος της ιστοσελίδας και έχουν ενσωματωθεί στον κώδικα της. Το σύστημα LMS καθιστά διαθέσιμα τα μαθήματα, τα οποία έχουν υλικό που έχει "ανέβει" εκεί από τον εκπαιδευτή, κάνει εγγραφές μαθητών, δημιουργούνται λογαριασμοί για τους μαθητές, και προχωρά στην επιβεβαίωση αυτών των εγγραφών. Οι μαθητές μετά την είσοδο τους, μπορούν να έχουν πρόσβαση στα μαθήματα ή/και δοκιμασίες σε αυτά, υπάρχουν υπενθυμίσεις για το πρόγραμμα μαθημάτων, ανακοινώσεις για την ολοκλήρωση του μαθήματος στο διδάσκοντα και τον μαθητή. Τέλος, καταγράφονται στοιχεία χρήσης του εκπαιδευτικού υλικού, παράγονται αναφορές για τον αριθμό των μαθητών που έχουν εγγραφεί, υπολογίζεται η βαθμολογία απόδοσης των μαθητών σε συγκεκριμένα μαθήματα, και αποθηκεύονται τα στοιχεία απαντήσεων των μαθητών.

Όποια μορφή δημιουργίας και αν έχει το εκάστοτε σύστημα τηλε-εκπαίδευσης που έχει δημιουργηθεί, δεν είναι δυνατή η αξιοποίηση του περιεχομένου ή/και των υπόλοιπων λειτουργιών που προσφέρει χωρίς την χρήση του διαδικτύου. Τα οφέλη του διαδικτύου γενικότερα, όσον αφορά το λογισμικό τηλε-εκπαίδευσης είναι αδιαμφισβήτητα, αφού καθιστά αυτά τα συστήματα ηλεκτρονικής διδασκαλίας διαθέσιμα από οποιονδήποτε, σε οποιαδήποτε ώρα, από οποιοδήποτε μέρος. Για το λόγο αυτό, τις τελευταίες δεκαετίες, πολλές ερευνητικές προσπάθειες έχουν γίνει για την βασισμένη στο διαδίκτυο εκπαίδευση (Alpert et al. 1999, Lopez et al. 1998, Virvou & Alerpis 2003), καθώς και για την διδασκαλία μέσω διαδικτύου (Virvou & Tsigira 2001, Gamper & Knapp 2002, Virvou & Katsionis 2003).

Στην περίπτωση της υποστήριξης μαθημάτων προγραμματισμού μικρής ηλικίας θα μας απασχολήσουν περισσότερο τα συστήματα ασύγχρονης τηλε-εκπαίδευσης και διαχείρισης της μάθησης (LMS) που χρησιμοποιούν τη βοήθεια της ασύγχρονης τηλε-εκπαίδευσης μέσω διαδικτύου, για να μπορεί ο μαθητής να κάνει είσοδο στον λογαριασμό του στο σύστημα σε οποιοδήποτε χρόνο και τόπο, να μελετάει το διαθέσιμο υλικό/μαθήματα, να εκτελεί δραστηριότητες/δοκιμασίες, και να γίνεται καταγραφή των δράσεων του στο σύστημα. Σε αυτά υπάρχει το πλεονέκτημα της μάθησης στον χρόνο και με το ρυθμό που επιθυμεί ο εκπαιδευόμενος, κάτι που είναι πολύ σημαντικό στην εκμάθηση των βασικών αρχών προγραμματισμού από μαθητές μικρής ηλικίας μέσω ενός προγραμματιστικού μικρόκοσμου, λόγω των δυσκολιών εννοιολογικών-χειρισμού-περιβάλλοντος που παρουσιάζονται και εξηγήσαμε σε προηγούμενη παράγραφο.

Όπως αναφέραμε και προηγουμένως, κατά την παρακολούθηση αυτών των μαθημάτων, και την εκτέλεση δραστηριοτήτων και δοκιμασιών για αυτά, γίνεται καταγραφή των δράσεων αυτών του μαθητή στα συστήματα ασύγχρονης τηλε-εκπαίδευσης και διαχείρισης της μάθησης (LMS). Αυτά τα στοιχεία που καταγράφονται αποτελούν κομμάτια για το μοντέλο του μαθητή-χρήστη του συστήματος, το οποίο σταδιακά δημιουργείται/σχηματίζεται μέσα στο σύστημα αυτό. Όσο περισσότερο χρησιμοποιεί ένας μαθητής το σύστημα τηλε-εκπαίδευσης, και όσα περισσότερα στοιχεία καταγράφονται για αυτόν, τόσο περισσότερο ενημερωμένο είναι το μοντέλο του μαθητή-χρήστη αυτού. Η μοντελοποίηση του μαθητή-χρήστη του συστήματος τηλε-εκπαίδευσης, ενημερώνοντας το προσωπικό του προφίλ με τις δράσεις του και τα αποτελέσματα τους, είναι πολύ σημαντική για τον καθηγητή του μαθήματος, ο οποίος μπορεί να παρακολουθεί με αυτό τον τρόπο την πρόοδο του, και να τον καθοδηγήσει καλύτερα για τις επόμενες ενέργειες του.

1.3 Μοντελοποίηση χρηστών-μαθητών

Η μοντελοποίηση χρηστών είναι διεργασία που εκτελείται από το σύστημα προκειμένου να παρέχει μία αναπαράσταση των απόψεων του χρήστη, των πιθανών του στόχων κατά την αλληλεπίδραση με το σύστημα, του επιπέδου γνώσης που έχει όσον αφορά στη χρήση του λογισμικού, των προτιμήσεων, ενδιαφερόντων του κλπ.

Ένα μοντέλο χρήστη αναπαριστά μία συλλογή από προσωπικά δεδομένα που σχετίζονται με ένα συγκεκριμένο χρήστη. Έτσι, αποτελούν την βάση για οποιαδήποτε προσαρμογή ή αλλαγή του συστήματος προς τον χρήστη. Το ποια δεδομένα θα καταγράφονται στο μοντέλο του χρήστη εξαρτώνται από το είδος και τον σκοπό της εφαρμογής. Μπορεί να περιλαμβάνουν προσωπικές πληροφορίες όπως όνομα και ηλικία, ενδιαφέροντα, ικανότητες και γνώσεις, στόχους και σχέδια, προτιμήσεις, καθώς και διάφορα στοιχεία για την συμπεριφορά του και την αλληλεπίδραση του με το σύστημα.

- Στατικό μοντέλο χρήστη: είναι το πιο βασικό είδος μοντέλου χρήστη. Όταν έχουν συγκεντρωθεί τα κυρίως δεδομένα συνήθως δεν αλλάζουν ξανά, είναι στατικά. Μελλοντικές αλλαγές στις συνήθειες ενέργειες του χρήστη δεν καταγράφονται πια και δεν υπάρχει περαιτέρω προσπάθεια αλλαγής του μοντέλου του χρήστη.
- Δυναμικό μοντέλο χρήστη: επιτρέπουν μία πιο ανανεώσιμη αναπαράσταση του χρήστη. Αλλαγές στις συνήθειες ενέργειες του, η πρόοδος του, οι αλληλεπιδράσεις του με το σύστημα καταγράφονται και επηρεάζουν ξανά και ξανά το μοντέλο του χρήστη. Το μοντέλο ενημερώνεται με αυτόν τον τρόπο και λαμβάνει υπόψη του τις ανάγκες και τους στόχους του χρήστη.

Η απαίτηση για την ενσωμάτωση τεχνικών μοντελοποίησης χρηστών στην ανάπτυξη λογισμικού έχει γίνει επιτακτική στις μέρες μας λόγω της πολυπλοκότητας του χειρισμού συστημάτων λογισμικού, και της εξάπλωσης της χρήσης συστημάτων λογισμικού από πολλές διαφορετικές ομάδες χρηστών λόγω της χρήσης του διαδικτύου. Ως αποτέλεσμα αυτής της ανάγκης έχει δημιουργηθεί μεγάλο παγκόσμιο ερευνητικό ενδιαφέρον στην μοντελοποίηση χρηστών τόσο από την ακαδημαϊκή κοινότητα όσο και από τις εταιρίες ανάπτυξης λογισμικού.

Ποικίλες προσεγγίσεις έχουν γίνει για την μοντελοποίηση μαθητή σε διάφορα συστήματα διδασκαλίας από την ερευνητική κοινότητα (Katsionis & Virvou 2008, Kabassi & Virvou 2006). Ένα τέτοιο είναι και ο Δάσκαλος Παθητικής Φωνής (Virvou et al. 2000, Virvou & Maras, 1999). Σε αυτό το σύστημα διδασκαλίας η πρόοδος του μαθητή και τα συνήθη λάθη του καταγράφονται στο μοντέλο του μαθητή. Πληροφορίες από το μοντέλο του μαθητή, χρησιμοποιούνται για την επίλυση της αμφιβολίας στις περιπτώσεις όπου δεν είναι ξεκάθαρο το τι μπορεί να είναι η βαθύτερη αιτία για ένα λάθος.

Ολοκληρώνοντας, θα πρέπει να σημειώσουμε ότι ειδικά στην περίπτωση του εκπαιδευτικού λογισμικού τηλε-εκπαίδευσης για μαθητές μικρής ηλικίας που είναι διαθέσιμο μέσω διαδικτύου αυτή η ανάγκη για μοντελοποίηση των χρηστών-μαθητών είναι ακόμα μεγαλύτερη για να μπορέσει ο καθηγητής να λάβει την ενημέρωση που χρειάζεται σχετικά με την δράση του μαθητή στο εκπαιδευτικό σύστημα και την πρόοδο του, αφού η δράση του μαθητή γίνεται μακριά από την σχολική τάξη και χωρίς την επίβλεψη από τον καθηγητή, και έτσι να προσαρμόσει την διδασκαλία που προσφέρεται μέσω του συστήματος τηλε-εκπαίδευσης.

2. Προγραμματιστικοί Μικρόκοσμοι

Στην παρούσα ενότητα, θα ασχοληθούμε με μια ιδιαίτερη κατηγορία προγραμματιστικών περιβαλλόντων, τους προγραμματιστικούς μικρόκοσμους. Με τον όρο προγραμματιστικά περιβάλλοντα εννοούμε συστήματα τα οποία επιτρέπουν την έκδοση και εκτέλεση προγραμμάτων. Οι προγραμματιστικοί μικρόκοσμοι αποτελούν ξεχωριστή κατηγορία προγραμματιστικών περιβαλλόντων, αφού παρουσιάζουν μια σειρά κοινών χαρακτηριστικών.

Η διδασκαλία-εκμάθηση του προγραμματισμού, όπως είναι γνωστό, παρουσιάζει αρκετές δυσκολίες. Τις τελευταίες δεκαετίες, έχει γίνει αξιόλογη ερευνητική δουλειά στην περιοχή της Διδακτικής της Πληροφορικής και ειδικότερα στην περιοχή της διδασκαλίας του προγραμματισμού. Όπως προκύπτει από τη σχετική βιβλιογραφία, ένας από τους σημαντικότερους παράγοντες που έχει διαπιστωθεί ότι αποτελεί πηγή δυσκολιών για την εκμάθηση του προγραμματισμού έγκειται στο γεγονός ότι, όπως φαίνεται, η κλασική προσέγγιση διδασκαλίας (Brusilovsky et al. 1997) είναι ασύμβατη με τις πραγματικές διδακτικές ανάγκες των μαθητών.

Με τον όρο κλασική προσέγγιση διδασκαλίας, εννοούμε την διδασκαλία που συνίσταται:

- στη χρήση μιας γλώσσας γενικού σκοπού (όπως οι Pascal, C, κλπ),
- ενός επαγγελματικού περιβάλλοντος προγραμματισμού για την γλώσσα αυτή, και
- στην επίλυση ενός συνόλου προβλημάτων επεξεργασίας αριθμών και συμβόλων.

Όπως έχει ήδη αναφερθεί στην εισαγωγική ενότητα, σύμφωνα με αρκετούς ερευνητές, η υιοθέτηση της κλασικής προσέγγισης διδασκαλίας του προγραμματισμού αποτελεί έναν από τους σημαντικότερους παράγοντες που καθιστά την εκμάθηση του προγραμματισμού δύσκολη. Τα σημαντικότερα προβλήματα που αντιμετωπίζουν οι αρχάριοι προγραμματιστές που διδάσκονται τις αρχές του προγραμματισμού με την κλασική προσέγγιση, όπως προκύπτει από εμπειρικές παρατηρήσεις και μελέτες ερευνητών αλλά και μαρτυρίες των ίδιων των μαθητών, είναι τα εξής:

- *Οι γλώσσες προγραμματισμού γενικού σκοπού διαθέτουν, κατά κανόνα, ένα μεγάλο ρεπερτόριο εντολών και είναι πολύπλοκες.* Το υλικό που πρέπει να καλύψει ο μαθητής, το οποίο περιλαμβάνει το θεωρητικό πλαίσιο μιας γλώσσας προγραμματισμού σε συνδυασμό με τις βασικές αρχές του προγραμματισμού, είναι πολύ μεγάλο. Το γεγονός αυτό έχει ως αποτέλεσμα ο μαθητής να δυσκολεύεται να αποκτήσει το απαιτούμενο υπόβαθρο. (Brusilovsky et al. 1997, Ruckert & Halpern 1993).
- *Η προσοχή των μαθητών επικεντρώνεται στην εκμάθηση των συντακτικών κανόνων των γλωσσών προγραμματισμού γενικού σκοπού που είναι αρκετά πολύπλοκοι, και έτσι αφιερώνουν πολύ χρόνο στην ίδια την γλώσσα προγραμματισμού, και όχι στην ανάπτυξη ικανοτήτων επίλυσης προβλημάτων μέσω της αλγοριθμικής σκέψης* (Brusilovsky et al. 1997, Ziegler & Crews 1999). Ενώ λοιπόν ο σκοπός του μαθήματος της εισαγωγής στον προγραμματισμό είναι η χρήση μιας γλώσσας προγραμματισμού για την διδασκαλία εννοιών του προγραμματισμού και την υλοποίηση αλγορίθμων, τελικά η προσοχή επικεντρώνεται στη γλώσσα προγραμματισμού.
- *Το προγραμματιστικό περιβάλλον δεν παρέχει δυνατότητες οπτικού προγραμματισμού και έτσι ο μαθητής δεν έχει στήριξη στην κατανόηση των βασικών ενεργειών και δομών ελέγχου.* Δεν μπορεί να δει την διαδικασία εκτέλεσης του προγράμματος βήμα-βήμα και το αντιμετωπίζει απλά ως μία διαδικασία εισόδου-εξόδου.
- *Οι εμπορικοί μεταγλωττιστές δεν ικανοποιούν τις ανάγκες των αρχάριων προγραμματιστών.* Οι περισσότεροι εμπορικοί μεταγλωττιστές απευθύνονται σε επαγγελματίες και όχι σε μαθητές, προκαλώντας έτσι αρκετά προβλήματα όταν

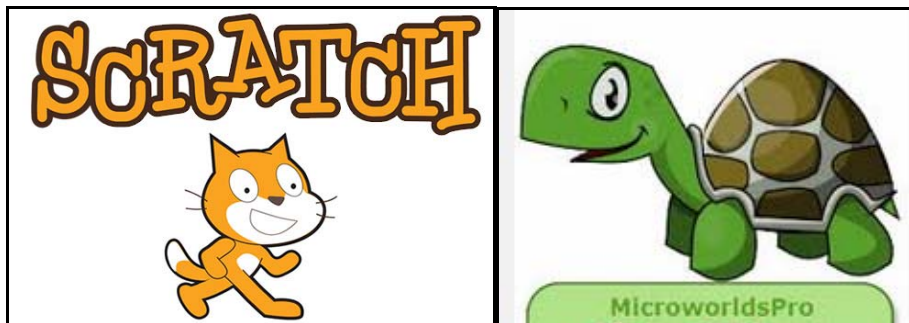
χρησιμοποιούνται στο πλαίσιο ενός μαθήματος εισαγωγής στον προγραμματισμό (Freund & Roberts 1996).

- Η *διανοητική πολυπλοκότητα που απαιτεί η εκφορά ενός αλγορίθμου σε μια γλώσσα προγραμματισμού είναι μεγάλη*. Οι προγραμματιστές πολύ συχνά κατά την ανάπτυξη προγραμμάτων δημιουργούν νοητά μοντέλα του προγράμματός τους, του τρόπου λειτουργίας του και χρήσης των δεδομένων. Η διανοητική «απόσταση» ανάμεσα στις νοητές αναπαραστάσεις των αλγορίθμων ενός αρχάριου προγραμματιστή και στην περιγραφή τους σε μια γλώσσα προγραμματισμού είναι πολύ μεγάλη, εξαιτίας της «φύσης» της γλώσσας (Pane & Myers 1996).
- Η *επίλυση ενδιαφερόντων προβλημάτων απαιτεί την εκμάθηση ενός μεγάλου υποσυνόλου της γλώσσας* και την ανάπτυξη αρκετά μεγάλων προγραμμάτων, απαιτεί δηλαδή την επικέντρωση της προσοχής στην εκμάθηση της γλώσσας (Brusilovsky et als 1997).

Το γεγονός ότι η κλασική προσέγγιση διδασκαλίας του προγραμματισμού δεν ικανοποιεί τις διδακτικές ανάγκες των μαθητών είχε ως αποτέλεσμα την ανάπτυξη ειδικών μεθοδολογιών διδασκαλίας, γλωσσών προγραμματισμού και εκπαιδευτικών εργαλείων που ονομάζονται: **Μικρόκοσμοι ή Μικρογλώσσες Προγραμματισμού**.

Η βασική ιδέα των μικρόκοσμων (microworlds) και των μικρογλωσσών (mini-languages) προγραμματισμού είναι η δημιουργία μιας γλώσσας προγραμματισμού που αποτελείται από ένα περιορισμένο ρεπερτόριο εντολών με απλή σύνταξη και σημασία. Οι βασικές ιδέες που διέπουν τη λειτουργία των μικρόκοσμων είναι, έμμεσα και χωρίς πλήρη γνώση, να δίνονται λύσεις στα συνήθη προβλήματα που παρουσιάζουν τα διαδεδομένα περιβάλλοντα προγραμματισμού. Η λύση στα συνήθη προβλήματα δίνεται μέσα από την δημιουργία οικειότητας με το περιβάλλον, μέσω της ύπαρξης μίας συγκεκριμένης πρωταγωνιστικής οντότητας.

Ο μαθητής μαθαίνει τις βασικές έννοιες του προγραμματισμού ελέγχοντας ένα πρωταγωνιστή (μία γάτα, μία χελώνα, ένα ρομπότ ή κάποια άλλη οντότητα) που ζει στον μικρόκοσμο, όπως η γάτα που περιλαμβάνει το περιβάλλον του Scratch (MIT Media Lab 2002), ή η χελώνα που περιλαμβάνει το περιβάλλον του Microworlds Pro (LCSI 1999).



Εικόνα 1: Οι πρωταγωνιστές των μικρόκοσμων του Scratch και Microworlds Pro.

Τα προγραμματιστικά περιβάλλοντα που βασίζονται στην προσέγγιση των μικρόκοσμων ενσωματώνουν συνήθως δυναμικές προσομοιώσεις εκτέλεσης των προγραμμάτων (program animation). Ο μαθητής δηλαδή, έχει τη δυνατότητα να εκτελεί τα προγράμματα που αναπτύσσει βήμα προς βήμα, παρακολουθώντας το αποτέλεσμα εκτέλεσης της κάθε εντολής, στην κατάσταση του μικρόκοσμου. Βασικό πλεονέκτημα της προσέγγισης των μικρόκοσμων, εκτός βέβαια από το περιορισμένο ρεπερτόριο εντολών και την οπτικοποίηση της εκτέλεσης ενός προγράμματος, είναι και το γεγονός ότι οι μικρόκοσμοι βασίζονται (στην πλειοψηφία τους) σε κάποιο φυσικό μοντέλο, που είναι ήδη γνωστό στους μαθητές και όχι σε κάποιο άγνωστο και δυσνόητο (για τους μαθητές) μοντέλο αναφοράς, όπως η μηχανή του Von Neumann.

Συγκεκριμένα, ο χρήστης βλέπει τον πρωταγωνιστή ή τους πρωταγωνιστές του μικρόκοσμου να ανταποκρίνονται σε κάθε μήνυμα εκτελώντας την αντίστοιχη μέθοδο, μεταβάλλοντας κατ' Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

αυτό τον τρόπο την κατάσταση τους και την κατάσταση του περιβάλλοντος στο οποίο ζουν. Αν και το σύνολο των διαθέσιμων μηνυμάτων/μεθόδων είναι μικρό, ωστόσο υπάρχει η δυνατότητα επίλυσης τόσο απλών όσο και αρκετά πολύπλοκων προβλημάτων. Εκτός από τις εντολές ελέγχου τις οποίες εκτελούν οι πρωταγωνιστές του μικρόκοσμου υπάρχουν και αρκετά ερωτήματα (συναρτήσεις) στα οποία μπορούν να απαντήσουν, έχουν δηλαδή τη δυνατότητα να ελέγχουν την κατάσταση του κόσμου όπου «ζουν». Επίσης, οι περισσότεροι μικρόκοσμοι περιλαμβάνουν όλες τις βασικές δομές ελέγχου και ένα μηχανισμό δημιουργίας και προσθήκης νέων μεθόδων στις υπάρχουσες δομές.

Οι μικρόκοσμοι μπορούν να χρησιμοποιηθούν αρκετά αποτελεσματικά στα πλαίσια εκμάθησης εννοιών του προγραμματισμού και την απόκτηση δεξιοτήτων επίλυσης προβλημάτων και αλγοριθμικού τρόπου σκέψης. Η διδασκαλία του προγραμματισμού με την προσέγγιση των μικρόκοσμων παρουσιάζει αρκετά πλεονεκτήματα:

- Η γλώσσα προγραμματισμού αποτελείται από ένα περιορισμένο ρεπερτόριο εντολών με απλή σύνταξη και σημασιολογία.
- Βασίζονται σε υπαρκτά μοντέλα που είναι ήδη γνωστά στο σπουδαστή, μειώνοντας έτσι δραματικά τη διανοητική «απόσταση» ανάμεσα στα νοητά μοντέλα ή την περιγραφή σε φυσική γλώσσα των αλγορίθμων και στην περιγραφή τους στη γλώσσα προγραμματισμού.
- Τα προβλήματα που καλούνται να λύσουν οι σπουδαστές παρουσιάζουν ιδιαίτερο ενδιαφέρον.
- Η εκτέλεση ενός προγράμματος είναι ορατή, αποκαλύπτοντας έτσι τη σημασία των διδασκόμενων δομών, καθώς και τις έννοιες που σχετίζονται με τη δομή και την εκτέλεση των προγραμμάτων.
- Υπάρχει δυνατότητα προσαρμογής του μικρόκοσμου στις ανάγκες του κοινού στο οποίο απευθύνεται.

Οι προγραμματιστικοί μικρόκοσμοι χρησιμοποιούνται για την εισαγωγή των μαθητών στον προγραμματισμό, εδώ και αρκετά χρόνια. Ένας μεγάλος αριθμός διδασκόντων τόσο σε Πανεπιστήμια, όσο και σε σχολεία, από διαφορετικές χώρες, αναφέρουν θετικά αποτελέσματα από τη χρήση προγραμματιστικών μικρόκοσμων (Brusilovsky et al. 1997). Ωστόσο, πρέπει να επισημανθεί ότι τα αποτελέσματα αυτά βασίζονται περισσότερο στις εντυπώσεις των διδασκόντων, και λιγότερο σε εμπειρική αξιολόγηση της αποτελεσματικότητας των μικρόκοσμων κατά την χρήση τους από τους μαθητές.

Βέβαια, η αξιολόγηση της διδακτικής προσέγγισης που βασίζεται στη χρήση των μικρόκοσμων δεν είναι εύκολη και θα πρέπει σε κάθε περίπτωση να λαμβάνονται υπόψη οι ιδιαίτερες συνθήκες χρήσης τους, το προφίλ των μαθητών και τα ιδιαίτερα χαρακτηριστικά του προγραμματιστικού μικρόκοσμου που χρησιμοποιείται. Αρκετοί ερευνητές επισημαίνουν ότι οι μικρόκοσμοι μπορούν να χρησιμοποιηθούν αποτελεσματικά αλλά με προσοχή. Οι Pattis et al. (1995) και Brusilovsky et al. (1997) επισημαίνουν ότι οι μικρόκοσμοι μπορούν να χρησιμοποιηθούν:

- στο πλαίσιο ενός μαθήματος εισαγωγής στον προγραμματισμό, το οποίο μπορεί να επικεντρωθεί αποκλειστικά στη διδασκαλία εννοιών του προγραμματισμού ή να προετοιμάσει παράλληλα τους μαθητές για την ευκολότερη μετάβαση σε μια γλώσσα προγραμματισμού γενικού σκοπού,
- σε προχωρημένα στάδια σπουδών για τη στήριξη των μαθητών στην κατανόηση δύσκολων εννοιών (π.χ. αναδρομή, δείκτες κλπ),
- για την ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων και αλγοριθμικού τρόπου σκέψης,
- από άτομα που θέλουν να αποκτήσουν κάποιες βασικές γνώσεις γύρω από τον προγραμματισμό, αλλά και την επιστήμη των υπολογιστών γενικότερα.

Επίσης, όπως ήδη αναφέρθηκε, στην αξιολόγηση της διδακτικής προσέγγισης των μικρόκοσμων, θα πρέπει οπωσδήποτε να λαμβάνονται υπ' όψη τα ιδιαίτερα χαρακτηριστικά του προγραμματιστικού περιβάλλοντος που χρησιμοποιείται, αφού ακόμα και για τον ίδιο μικρόκοσμο έχουν, σε αρκετές περιπτώσεις, υλοποιηθεί περιβάλλοντα με διαφορετικά χαρακτηριστικά και λειτουργίες. Αν και οι προγραμματιστικοί μικρόκοσμοι που έχουν αναπτυχθεί είναι πολλοί και με σημαντικές διαφορές, μπορούν να ταξινομηθούν σε γενικές κατηγορίες όπως οι μικρόκοσμοι βασισμένοι στη Logo (Logo like) όπως το περιβάλλον του Microworlds Pro (LCSI 1999), μικρόκοσμοι υποκατηγορίες των logo like βασισμένοι στα ρομπότ (Karel like) όπως το περιβάλλον των Karel, Karel++, Object Karel (Pattis et als 1995), μικρόκοσμοι βασισμένοι μεν σε logo-like προγραμματιστικά περιβάλλοντα, αλλά υλοποιημένοι με Οπτικό Προγραμματισμό με Πλακίδια όπως το περιβάλλον του Scratch (MIT Media Lab 2002), μικρόκοσμοι παιχνίδια, και πολλά άλλα είδη (Υπουργείο Παιδείας & Θρ. 2013).

2.1 Διάφορα περιβάλλοντα προγραμματιστικών μικρόκοσμων

Η «οικογένεια» περιβαλλόντων προγραμματιστικών μικρόκοσμων Logo (Logo-like environments)

Η γλώσσα **Logo** είναι δημιούργημα του Αμερικανού ερευνητή Seymour Papert (Papert & Harel 1991). Ο S. Papert υπήρξε μαθητής του J. Piaget και η γλώσσα Logo υπήρξε καρπός της προσπάθειας του να κατασκευάσει έναν μικρόκοσμο στον Η.Υ., ο οποίος να «υλοποιεί» τη θεωρία του J. Piaget, τον κονστρουκτιβισμό (constructivism), αλλά και την επέκταση της θεωρίας αυτής που επινόησε ο ίδιος ο Papert, το λεγόμενο κονστρουκτιονισμό (constructionism). Το όνομά της προέρχεται από την ελληνική λέξη «ΛΟΓΟΣ».

Η γλώσσα προγραμματισμού Logo σχεδιάστηκε στο MIT στα τέλη της δεκαετίας του 60 για εκπαιδευτικούς κυρίως σκοπούς από ομάδα ερευνητών στην Τεχνητή Νοημοσύνη, με επικεφαλής όπως αναφέραμε, τον Seymour Papert. Η Logo αποτελεί διάλεκτο της Lisp, της γλώσσας της τεχνητής νοημοσύνης, και έχει αξιοποιηθεί στην εκπαίδευση περισσότερο από οποιαδήποτε άλλη γλώσσα προγραμματισμού. Η Logo αρχικά περιοριζόταν στην επεξεργασία καταλόγων και λέξεων, γρήγορα όμως επεκτάθηκε και στη δημιουργία γραφικών με τη βοήθεια ενός μικρού χαρακτήρα που κατευθύνεται από υπολογιστή.

Η βασική ιδέα της Logo, είναι η δημιουργία μιας «Μαθηματικό-χώρας», δηλαδή ενός περιβάλλοντος που θα λειτουργεί με βάση τα Μαθηματικά, με πρωταγωνιστή μία χελώνα. Ωστόσο, καθώς η χελώνα προγραμματίζεται, η γλώσσα Logo μπορεί να χρησιμοποιηθεί και στη διδασκαλία του προγραμματισμού – μάλιστα ο Brian Harvey από το M.I.T. έχει συγγράψει ένα τρίτομο έργο για προγραμματισμό με τη γλώσσα Logo (Harvey 1997). Η γνωστή χελώνα της Logo εδώ και 40 σχεδόν χρόνια, έχει χρησιμοποιηθεί σε δεκάδες παραλλαγές, δημιουργώντας μια «οικογένεια» λογισμικών με ανάλογα χαρακτηριστικά. Τα χαρακτηριστικά της Logo είναι:

- Ένας «πρωταγωνιστής» που υποτίθεται ότι παριστάνει μια χελώνα και κινείται πάνω σε μια επίπεδη, λευκή επιφάνεια, χαράσσοντας μια γραμμή από τα σημεία από τα οποία περνάει.
- Μια απλή γλώσσα προγραμματισμού (συναρτησιακή, functional) για τον καθορισμό της κίνησης της χελώνας, με πολύ απλές δομές δεδομένων (λίστες), με στοιχειώδεις λειτουργίες εισόδου-εξόδου, με δυνατότητες αναδρομής και επεκτάσιμη, μέσω συναρτήσεων που δημιουργεί ο χρήστης.
- Επειδή η κίνηση της χελώνας είναι άμεση, ο χρήστης διαπιστώνει με προφανή τρόπο αν πέτυχε τον προκαθορισμένο στόχο του (για παράδειγμα αν η χελώνα «σχημάτισε» ένα τετράγωνο) και μπορεί να διορθώσει το πρόγραμμά του, σε περίπτωση λάθους.

Η οικογένεια της Logo περιλαμβάνει πολλές παραλλαγές του πρώτου βασικού περιβάλλοντος. Η γλώσσα Logo, ενσωματώνεται ως βασική γλώσσα προγραμματισμού σε

διάφορα περιβάλλοντα (όπως το ABAKEIO (2001), το MicroWorlds Pro (LCSI 1999), το HyperStudio (1995) κ.ά.). Ορισμένες παραλλαγές της Logo έχουν τρισδιάστατη αναπαράσταση, ή ενσωματώνουν διάφορες πρόσθετες ιδιότητες (για παράδειγμα, οι χελώνες έχουν «μάζα» και αδράνεια). Μερικές αξιοσημείωτες περιπτώσεις συστημάτων της οικογένειας Logo είναι η LadyBug («πασχαλίτσα»), ένα πολύ απλό περιβάλλον στο οποίο ο προγραμματισμός πραγματοποιείται χωρίς καθόλου γραπτό κείμενο, η StarLogo, ένα περιβάλλον «μαζικά παράλληλου προγραμματισμού», στο οποίο ο χρήστης μπορεί να διαχειρίζεται ταυτόχρονα εκατοντάδες ή χιλιάδες «χελώνες» και τα Lego Mindstorms, κατασκευές, εμπνευσμένες από τη γλώσσα Logo (Υπουργείο Παιδείας & Θρ. 2013).

Τα logo-like προγραμματιστικά περιβάλλοντα διαθέτουν μια απλή, σαφή και ισχυρή γλώσσα προγραμματισμού, που επιτρέπει σύντομο κύκλο ανάπτυξης με άμεση εκτέλεση εντολών. Τα γραφικά της χελώνας επιτρέπουν στους μαθητές να δουν τα αποτελέσματα της εκτέλεσης ενός προγράμματος με άμεσο τρόπο, γεγονός πάρα πολύ σημαντικό για τον σχεδιασμό εκπαιδευτικών εφαρμογών για μαθητές μικρής ηλικίας (Eisenberg 1995).

Σε ένα κλασικό περιβάλλον logo οι μαθητές μπορούν να έρθουν σε επαφή σχεδόν με το σύνολο της αλγοριθμικής που μπορεί να εφαρμοστεί μέσα από τον δομημένο ιεραρχικά και τμηματικά προγραμματισμό. Οι δυνατότητες των logo-like γλωσσών μπορούν να χτίσουν νοητικές υποδομές και για μετά το δημοτικό και το γυμνάσιο.

Τα Logo-like περιβάλλοντα αξιοποιούνται για τη σχεδίαση και την ανάπτυξη υπολογιστικών εργαλείων, τα οποία προσφέρουν στους μαθητές τη δυνατότητα έκφρασης κι αξιοποίησης των σκέψεων, ιδεών και διαισθήσεών τους και υποστηρίζουν τη διαδικασία οικοδόμησης της γνώσης διαμορφώνοντας πλούσια σε ευκαιρίες προβληματισμού και πειραματισμού περιβάλλοντα μάθησης (Hoyle & Noss 1992, Papert, 1980).

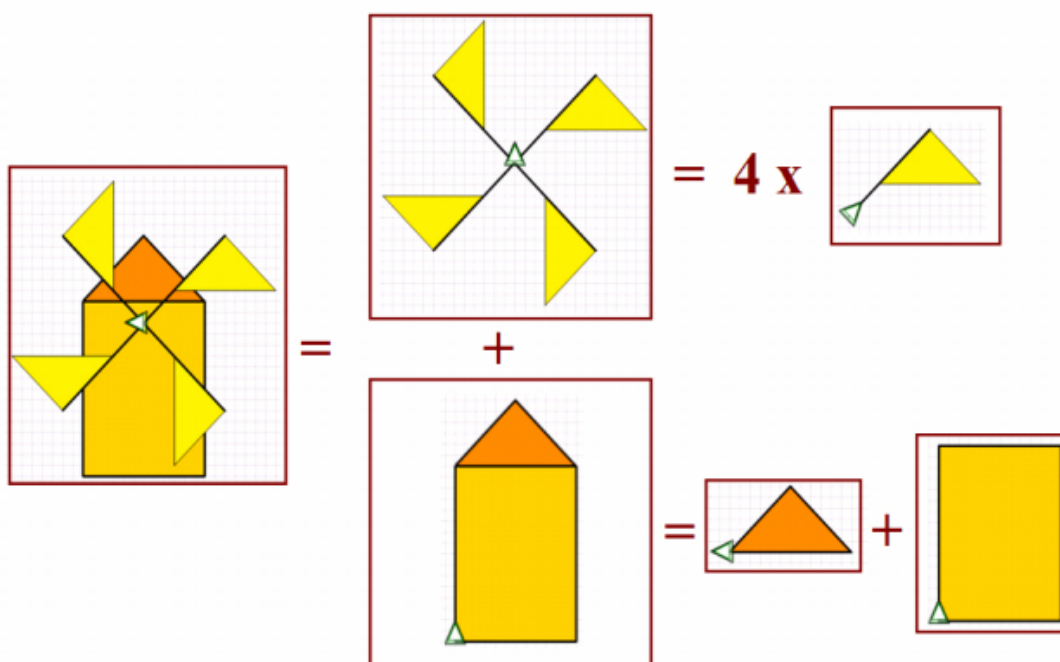
Στο περιβάλλον της Logo, το παιδί μπορεί εύκολα να φτιάξει πολλές δικές του λέξεις-διεργασίες (δυνατότητα επεκτασιμότητας), επιπλέον καλλιεργείται η δυσπιστία στο κυρίαρχο εκπαιδευτικό κριτήριο του «σωστού-λάθους» και του «αληθινού-ψεύτικου». Ειδικά στην αλληλουχία εντολών που δομεί το παιδί για να κατευθύνει τη χελώνα η ερώτηση που τίθεται σχετικά με το πρόγραμμα που δημιουργείται, δεν είναι αν είναι σωστό ή λάθος αλλά αν διορθώνεται ή πως αλλιώς μπορεί να φτιαχτεί.

Οι μαθητές, στο περιβάλλον της **Logo**:

- Πειραματίζονται με εντολές Logo για να τις καταλάβουν και να νιώσουν αυτοπεποίθηση με τη χρήση τους.
- Σχεδιάζουν τη δουλειά τους και την οργανώνουν στις διάφορες συνιστώσες της.
- Γράφουν μια σειρά εντολών για να ολοκληρώσουν κάθε μικρό υποέργο.
- Οικοδομούν ένα πρόγραμμα όπου ολοκληρώνουν όλα τα υποέργα στη σωστή σειρά.
- Δοκιμάζουν το πρόγραμμά τους και αξιολογούν τη δουλειά τους.
- Εκσφαλτώνουν το πρόγραμμα τους εντοπίζοντας και διορθώνοντας τα λάθη ή αναδομούν την προσέγγισή τους.

Τα νέα ΠΣ εκμεταλλεύονται τις δυνατότητες της logo δίνοντας έμφαση σε διαχρονικές έννοιες του προγραμματισμού αλλά και σε έννοιες που είναι χρήσιμες στην καθημερινή ζωή, και όχι σε συγκεκριμένες και εξειδικευμένες δυνατότητες εφήμερων γλωσσών προγραμματισμού.

Η Logo ακολουθεί τη λογική του δομημένου προγραμματισμού. Σύμφωνα με αυτόν η συνολική διεργασία υποδιαιρείται σε μικρότερα φυσικά μέρη έτσι ώστε να μπορούμε να διορθώσουμε τα σφάλματα κάθε μέρους ξεχωριστά. Μπορούμε δηλαδή να δημιουργήσουμε υποδιαδικασίες που δοκιμάζονται και ελέγχονται ως προς τη λειτουργία τους και στη συνέχεια να τις συνθέσουμε σε μια υπερδιαδικασία.

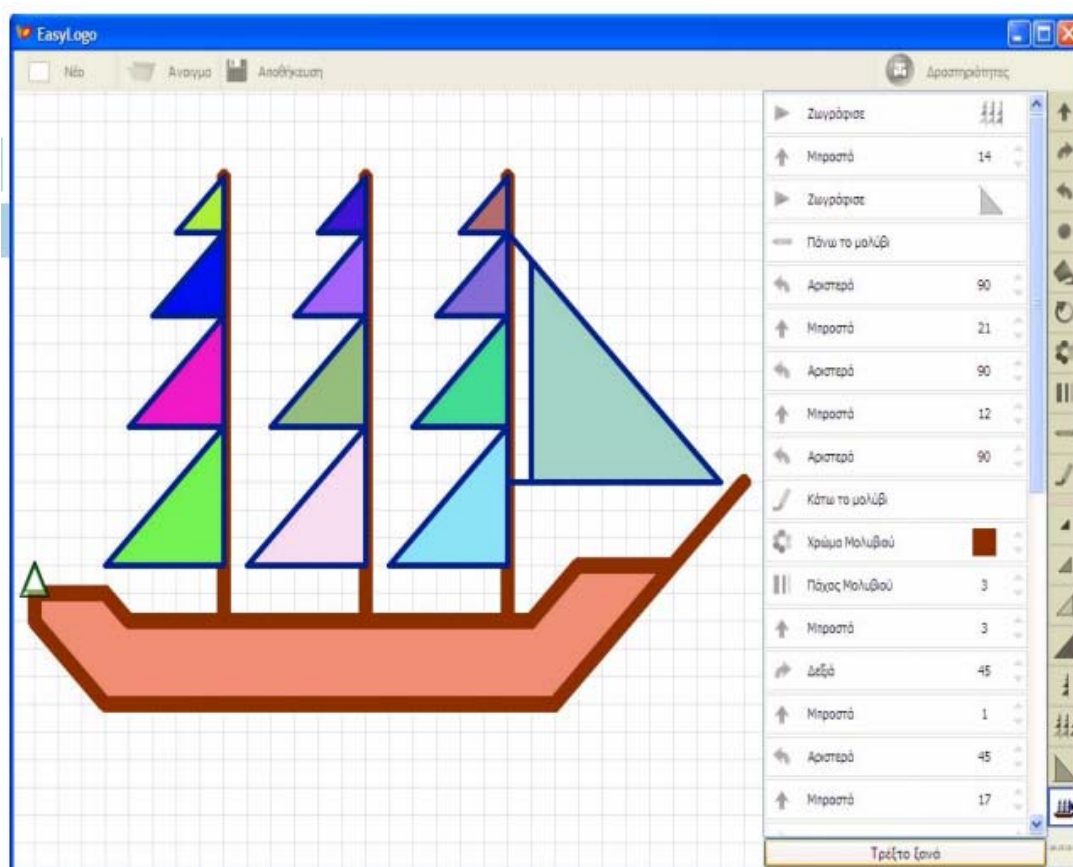


Εικόνα 2: Αναλυτικές δυνατότητες των σύγχρονων περιβαλλόντων Logo.

Σύμφωνα με το Bruner κάποια γνώση αναπαρίσταται ως δράση (λειτουργική γνώση), κάποια ως εικόνα και μια τρίτη ως σύμβολο. Ο Papert στο μικρόκοσμο της χελώνας δίνει την ευκαιρία στα παιδιά να μετέρχονται τους τρεις τρόπους αναπαράστασης της γνώσης, γεγονός που την καθιστά περισσότερο εύπεπτη και σταθερή.

Πολλά Logo-like προγραμματιστικά περιβάλλοντα μικρόκοσμων είναι κατάλληλα και για παιδιά που δεν έχουν κατακτήσει τη γραφή (λόγω ηλικίας ή ειδικών αναγκών). π.χ. η **EasyLogo** (Comenius University, 1994) περιβάλλον οπτικού προγραμματισμού χωρίς πλακίδια, που διαθέτει ένα περιορισμένο σύνολο εντολών και παρέχει τις παρακάτω δυνατότητες:

- Οπτικό Περιβάλλον Προγραμματισμού (χωρίς πλακίδια).
- Οι εντολές ταυτοποιούνται με οπτικά σύμβολα και λέξεις.
- «Αυτονόητος» τρόπος δημιουργίας-ορισμού διαδικασιών.
- Δεν χρησιμοποιεί τις διαδικασίες με αναδρομικό τρόπο.
- Το πρόγραμμα «τρέχει» συνεχώς παρέχοντας έτσι στον προγραμματιστή άμεση ανατροφοδότηση.
- Λειτουργεί μόνο με τιμές και δεν διαθέτει μεταβλητές.
- Δεν έχει εντολή αποφάσεων.



Εικόνα 3: Το περιβάλλον της EasyLogo, κατάλληλο και για παιδιά που δεν έχουν κατακτήσει ακόμα την γραφή.

Ένας από τους προγραμματιστικούς μικρόκοσμούς που προσφέρουν προγραμματιστικά περιβάλλοντα όπως το ABAKEIO (2001), και το MicroWorlds Pro (LCSI 1999), είναι ο **Χελωνόκοσμος**.

Ο χελωνόκοσμος είναι ένα περιβάλλον συμβολικής έκφρασης για τη δημιουργία και επεξεργασία μοντέλων και πειραματισμών με αυτά.

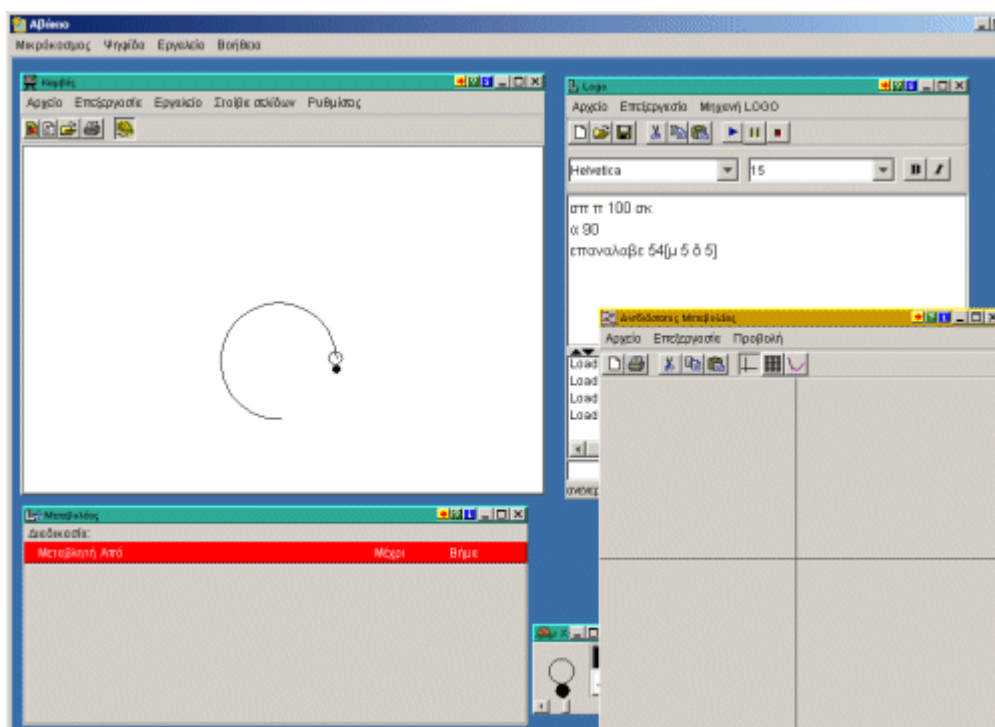
Το λογισμικό, περιβάλλον του χελωνόκοσμου είναι ένα εκπαιδευτικό περιβάλλον διερευνητικής μάθησης. Διαθέτει μια επιφάνεια εργασίας στην οποία μπορεί κάποιος να συνθέσει εκπαιδευτικούς «Μικρόκοσμούς» για πειραματισμό και διερεύνηση φαινομένων, εννοιών, υποθέσεων και συσχετισμών.

Κάθε Μικρόκοσμος δημιουργείται με τη διασύνδεση μικρών, προ-παρασκευασμένων κομματιών λογισμικού που εκτελούν μια συγκεκριμένη λειτουργία και αποσκοπούν στη δημιουργία ενός εκπαιδευτικά αξιοποιήσιμου συνόλου. Αυτά τα κομμάτια λογισμικού ονομάζονται ψηφίδες.

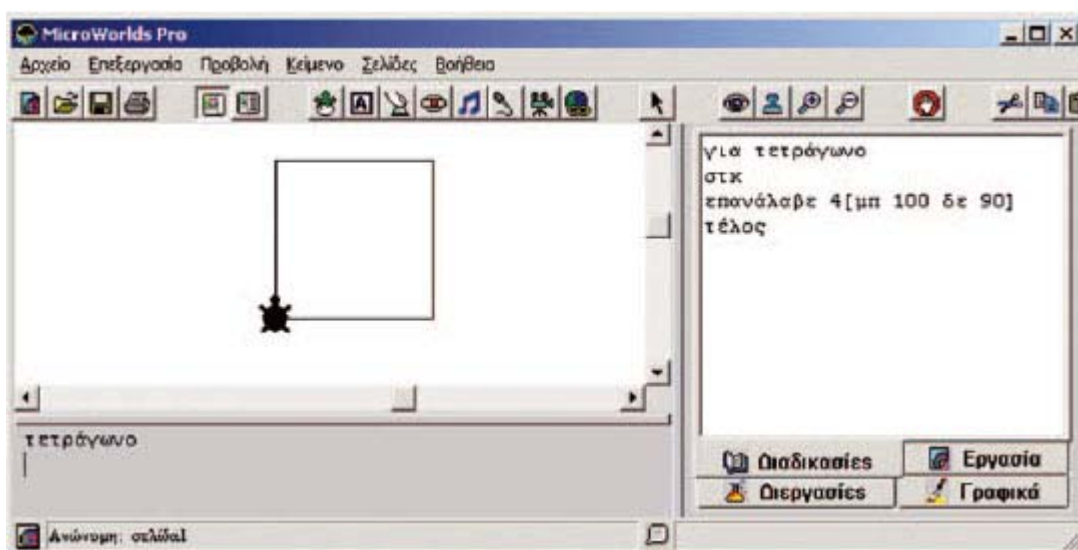
Έτσι λοιπόν, το περιβάλλον του χελωνόκοσμου αποτελείται από διακριτές αλλά συνδεδεμένες περιοχές εργασίας, που όπως αναφέραμε ονομάζονται ψηφίδες. Κάθε ψηφίδα είναι ορισμένη για κάποιες συγκεκριμένες εργασίες ή λειτουργίες.

- Υπάρχει η ψηφίδα Χελώνα.
- Υπάρχει η ψηφίδα Καμβάς στην οποία κινείται και αφήνει το ίχνος της η Χελώνα.
- Υπάρχει η ψηφίδα Logo στην οποία γράφουμε με συμβολικό τρόπο τις οδηγίες εκείνες τις οποίες θέλουμε να εκτελέσει η χελώνα.
- Υπάρχει η ψηφίδα Μεταβολέας με την οποία έχουμε τη δυνατότητα να εμφανίζουμε τις μεταβλητές του σχήματος και να αλλάζουμε τις τιμές τους με δυναμικό τρόπο.

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

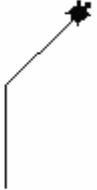


Εικόνα 4: Το περιβάλλον του Χελωνόκοσμου, του περιβάλλοντος προγραμματιστικών μικρόκοσμων που προσφέρει το Αβάκειο.

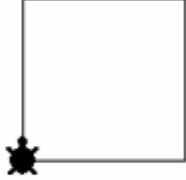


Εικόνα 5: Το περιβάλλον του Χελωνόκοσμου, του περιβάλλοντος προγραμματιστικών μικρόκοσμων που προσφέρει το MicroWorlds Pro.

Η χελώνα εμφανίζεται στην οθόνη ως εικονική αναπαράσταση μιας χελώνας έτσι ώστε να φαίνονται καθαρά η θέση και η κατεύθυνσή της (δηλαδή, προς τα πού είναι στραμμένη). Αυτά είναι τα βασικά στοιχεία της κατάστασης μιας χελώνας. Οι πιο σημαντικές εντολές Logo που ελέγχουν τη χελώνα είναι αυτές που αλλάζουν την κατάστασή της. Για παράδειγμα, οι εντολές **μπροστά** (ή, για συντομία, **μπ**) και **πίσω** (ή **πι**) τη μετακινούν, ενώ οι εντολές **αριστερά** (ή **αρ**) και **δεξιά** (ή **δε**) την περιστρέφουν. Η χελώνα επίσης μεταφέρει ένα στυλό. Αν τοποθετήσετε το στυλό κάτω (**ΣτυλόΚάτω** ή, για συντομία, **ΣΤΚ**), η χελώνα αφήνει ίχνη στο πέρασμά της.

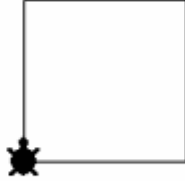
ΕΝΤΟΛΕΣ	ΑΠΟΤΕΛΕΣΜΑ
<p>ΣτυλόΚάτω μπροστά 100 δεξιά 45 μπροστά 100</p>	
<p>Η εντολή μπροστά 100 μετακινεί τη χελώνα 100 βήματα μπροστά. Η εντολή δεξιά 45 περιστρέφει τη χελώνα κατά 45 μοίρες προς τα δεξιά, χωρίς να αλλάζει η θέση της.</p>	

Εικόνα 6: Παράδειγμα εκτέλεσης εντολών στο περιβάλλον του Χελωνόκοσμου.

ΕΝΤΟΛΕΣ	ΑΠΟΤΕΛΕΣΜΑ
<p>μπροστά 100 δεξιά 90 μπροστά 100 δεξιά 90 μπροστά 100 δεξιά 90 μπροστά 100 δεξιά 90</p>	
<p>Η εντολή μπροστά 100 μετακινεί τη χελώνα 100 βήματα μπροστά. Η εντολή δεξιά 90 περιστρέφει τη χελώνα κατά 90 μοίρες προς τα δεξιά, χωρίς να αλλάζει η θέση της.</p>	

Εικόνα 7: Παράδειγμα δημιουργίας τετραγώνου στο περιβάλλον του Χελωνόκοσμου.

Το ίδιο αποτέλεσμα μπορούμε να έχουμε χρησιμοποιώντας την εντολή της επανάληψης. Η εντολή **επανάλαβε** δέχεται ως είσοδο μια λίστα οδηγιών. Η λίστα οδηγιών είναι ένα σύνολο εντολών που θα εκτελεστούν από τη Logo. Η ακόλουθη οδηγία δίνει εντολή στη χελώνα να σχεδιάσει ένα τετράγωνο, με επανάληψη των εντολών μπροστά 100 και δεξιά 90 τέσσερις φορές.

ΕΝΤΟΛΕΣ	ΑΠΟΤΕΛΕΣΜΑ
<p>επανάλαβε 4 [μπροστά 100 δεξιά 90]</p>	
<p>Αν δώσετε διαφορετικές τιμές στην εντολή μπροστά, σχηματίζονται τετράγωνα διαφορετικών μεγεθών.</p>	

Εικόνα 8: Παράδειγμα χρήσης επαναληπτικής δομής στο περιβάλλον του Χελωνόκοσμου.

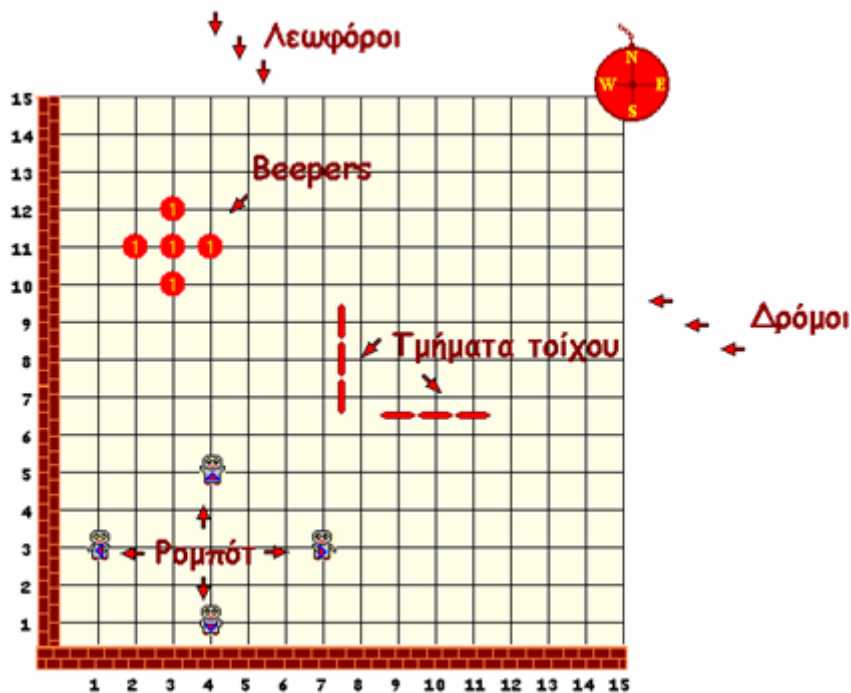
Το προγραμματιστικό περιβάλλον του Χελωνόκοσμου προσφέρει και άλλες πιο **σύνθετες εντολές, μεταβλητές και διαδικασίες** όπου δεν είναι δόκιμο να αναλύσουμε σε αυτό το σημείο.

Στην εξέλιξή της η **logo** έχει αποκτήσει εκδόσεις που κάθε μια ενσωματώνει πολλά ιδιαίτερα χαρακτηριστικά και δυνατότητες όπως: κατασκευή υπερμεσικών εφαρμογών, μοντελοποίηση, έλεγχος ρομπότ και ανάπτυξη αλληλεπιδραστικών παιχνιδιών που προκαλούν το ενδιαφέρον των παιδιών και που τα εμπλέκουν σε αυθεντικές διαδικασίες παραγωγής.

Η «οικογένεια» περιβαλλόντων προγραμματιστικών μικρόκοσμων βασισμένων στα ρομπότ Karel-like

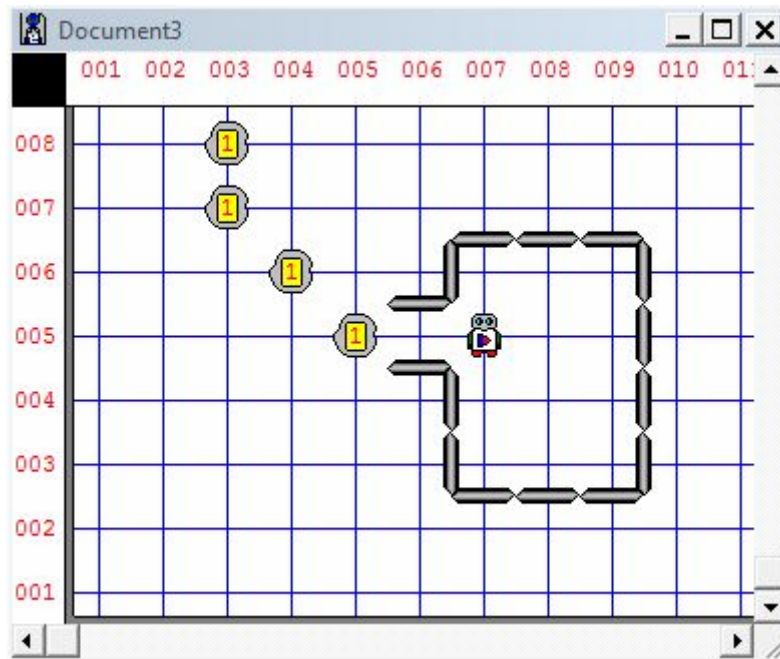
Η οικογένεια Logo περιλαμβάνει και την «υπο-οικογένεια» **Karel**, ενός υποθετικού ρομπότ, προσομοιωμένου στην οθόνη, που κινείται πάνω σε ένα «πλέγμα» ορθογώνιως τεμνόμενων δρόμων, με παραλλαγές όπως το Karel++, το JKarelRobot, το Jeroo, το objectKarel και άλλα, ανάλογα συστήματα (Υπουργείο Παιδείας & Θρ. 2013).

Ένας από τους πιο δημοφιλείς μικρόκοσμους προγραμματισμού βασισμένος στα ρομπότ, όπως επισημαίνουν αρκετοί ερευνητές (Brusilovsky, Calabrese, Hvorenky, Miller, Kouchnirenko, 1997) είναι ο “Karel the Robot” (Pattis et al. 1995) που σχεδιάστηκε προκειμένου να χρησιμοποιηθεί για το μάθημα της εισαγωγής στον δομημένο προγραμματισμό. Ο Karel, ο πρωταγωνιστής του μικρόκοσμου, εκτελεί διάφορες αποστολές (προγράμματα) σε ένα κόσμο που αποτελείται από οριζόντιους δρόμους και κάθετες λεωφόρους. Στον κόσμο του Karel μπορεί να υπάρχουν τμήματα τοίχου που τοποθετούνται μεταξύ των διασταυρώσεων δημιουργώντας εμπόδια (π.χ. λαβύρινθους) που καλείται να ξεπεράσει ο Karel και beepers, μικροί πλαστικοί κώνοι που παράγουν ένα ήχο (μπιπ). Ο Karel έχει τη δυνατότητα να κινείται προς την τρέχουσα κατεύθυνση κατά 1 μπλοκ, να στρίβει κατά 90 μοίρες, να εντοπίζει beepers που βρίσκονται στην ίδια διασταύρωση μ’ αυτόν, να εντοπίζει τοίχους που βρίσκονται μπροστά του σε απόσταση μισού μπλοκ χρησιμοποιώντας μια κάμερα, να σηκώνει και να κατεβάζει beepers με το μηχανικό του χέρι και τέλος μπορεί να καθορίζει προς ποια κατεύθυνση βλέπει χρησιμοποιώντας την πυξίδα του. Η γλώσσα προγραμματισμού εκτός από τις εντολές ελέγχου του Karel (move, turnleft, pickbeeper, putbeeper) περιλαμβάνει όλες τις βασικές δομές ελέγχου (if, if-else, while, loop).



Εικόνα 9: Ο προγραμματιστικός μικρόκοσμος βασισμένος στα ρομπότ, Karel.

Η μεταγενέστερη έκδοση του ρομπότ Karel, γνωστή ως **Karel++** (Bergin et al., 1997), υποστηρίζει τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού. Στον Karel++ υπάρχουν δύο μοντέλα (κλάσεις) ρομπότ που ανταποκρίνονται σε ένα περιορισμένο αριθμό μηνυμάτων. Οι μαθητές μπορούν να δημιουργήσουν ρομπότ των δύο βασικών μοντέλων, αλλά για πιο απαιτητικές αποστολές μπορούν να δημιουργήσουν και τα δικά τους μοντέλα ρομπότ. Τα νέα μοντέλα ρομπότ κληρονομούν όλες τις δυνατότητες των βασικών μοντέλων και τις επεκτείνουν.



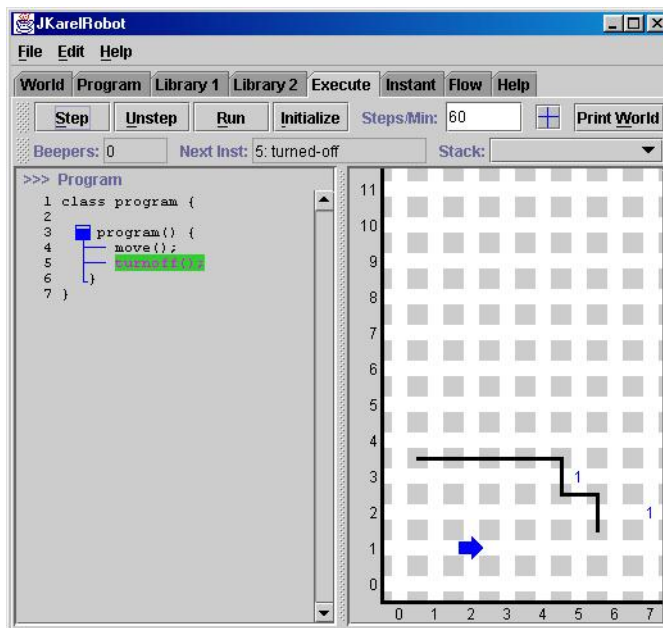
Εικόνα 10: Ο προγραμματιστικός μικρόκοσμος βασισμένος στα ρομπότ, Karel++.

Ο μικρόκοσμος **JKarelRobot** (Buck and Stucki, 2000) βασίζεται στο ρομπότ Karel (Pattis et al. 1995) και παρέχει τη δυνατότητα προγραμματισμού του ομώνυμου ρομπότ χρησιμοποιώντας μια γλώσσα προγραμματισμού που μοιάζει με την Pascal, τη Java ή τη Lisp. Ο μικρόκοσμος JKarelRobot αποτελείται από τις παρακάτω διαφορετικές καρτέλες:

- την καρτέλα World, στην οποία ο σπουδαστής μπορεί να δημιουργήσει και να ανακαλέσει καταστάσεις του κόσμου, καθώς επίσης και να καθορίσει την αρχική κατάσταση του Karel που είναι το μοναδικό ρομπότ που μπορεί να χρησιμοποιηθεί,
- την καρτέλα Program όπου ο σπουδαστής αναπτύσσει, μεταγλωττίζει και διαχειρίζεται τα προγράμματα του,
- την καρτέλα Execute στην οποία ο σπουδαστής εκτελεί τα προγράμματά του βήμα προς βήμα είτε προς τα εμπρός είτε προς τα πίσω,
- την καρτέλα Instant όπου ο σπουδαστής μπορεί να γράψει μία ή περισσότερες εντολές και να τις εκτελέσει όπως σε ένα ερμηνευτή,
- την καρτέλα Flow που χρησιμοποιείται για την ανάπτυξη διαγραμμάτων ροής για υπάρχοντα προγράμματα,
- την καρτέλα Help στην οποία παρέχεται ένα πολύ σύντομο tutorial.

Στην ουσία, στο περιβάλλον του JKarelRobot η δημιουργία της αρχικής κατάστασης του κόσμου, η ανάπτυξη του προγράμματος και η εκτέλεση του πραγματοποιούνται σε 3 διαφορετικές καρτέλες. Επιπλέον, παρόλο που ο σπουδαστής χρειάζεται σε κάθε καρτέλα να εκτελέσει μια συγκεκριμένη ακολουθία ενεργειών πατώντας κάποια κουμπιά, όλες οι καρτέλες και τα κουμπιά είναι συνεχώς ενεργά και δεν παρέχεται καμία καθοδήγηση στον χρήστη.

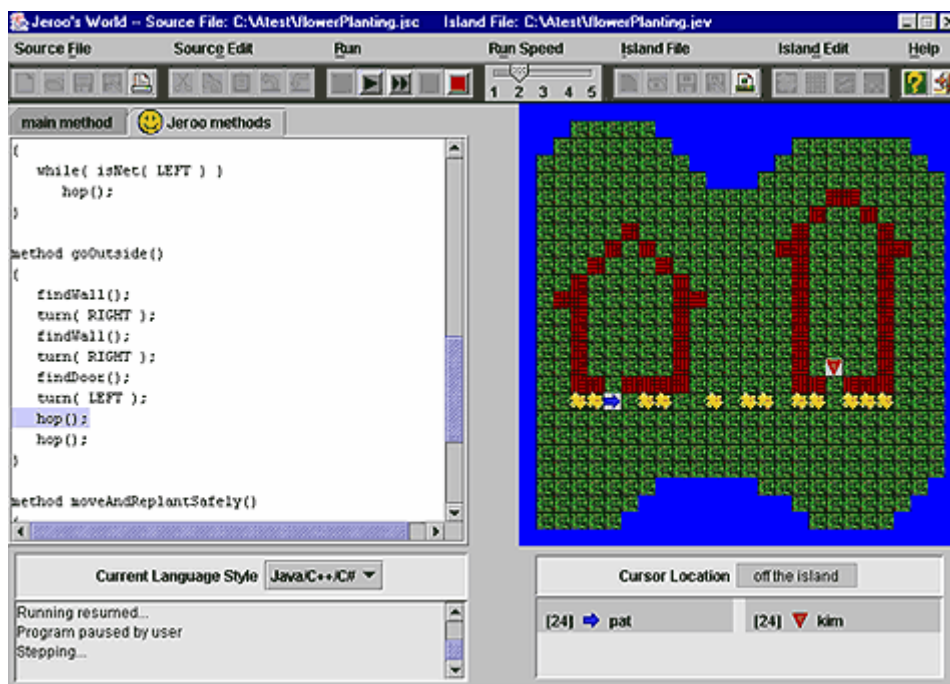
Όπως ήδη αναφέρθηκε στην αρχή, ο σπουδαστής έχει τη δυνατότητα να αναπτύξει τα προγράμματά του σε μια Java-like γλώσσα. Ωστόσο, στα προγράμματα δεν χρησιμοποιούνται εντολές δημιουργίας και αρχικοποίησης αντικειμένων, αφού το περιβάλλον υποστηρίζει τη χρήση ενός μόνο αντικειμένου (ρομπότ) το οποίο αρχικοποιείται στα πλαίσια καθορισμού της αρχικής κατάστασης του κόσμου. Ο περιορισμός αυτός, δηλαδή η ύπαρξη ενός μόνο αντικειμένου στο περιβάλλον του JKarelRobot μπορεί να οδηγήσει τους σπουδαστές στην παρανόηση ότι κλάση και αντικείμενο είναι έννοιες ταυτόσημες. Επίσης, στο tutorial του JKarelRobot δεν γίνεται καμία αναφορά στις βασικές έννοιες του αντικειμενοστρεφούς παραδείγματος προγραμματισμού: αντικείμενο, μήνυμα, κλάση, κληρονομικότητα, πολυμορφισμός.



Εικόνα 11: Ο προγραμματιστικός μικρόκοσμος βασισμένος στα ρομπότ, JKarelRobot.

Ο μικρόκοσμος **Jeroo** (Sanders and Dorn, 2003) βασίζεται στο ρομπότ Karel, αλλά χρησιμοποιεί μια γλώσσα προγραμματισμού που επιτρέπει μια πιο εύκολη μετάβαση στη C++ και τη Java. Το γραφικό ενδιάμεσο αποτελείται από ένα μόνο παράθυρο, ενώ το περιβάλλον υποστηρίζει την ανάπτυξη αναδρομικών μεθόδων και τη βηματική εκτέλεση των προγραμμάτων.

Ωστόσο, ο μικρόκοσμος Jeroo παρουσιάζει τους εξής περιορισμούς: (1) υπάρχει μία μόνο κλάση και δεν υπάρχει δυνατότητα δημιουργίας νέων κλάσεων, (2) οι σπουδαστές μπορούν να δημιουργήσουν μέχρι τέσσερα αντικείμενα της υπάρχουσας κλάσης, (3) δεν υποστηρίζεται η έννοια της κληρονομικότητας, (4) οι σπουδαστές μπορούν να επεκτείνουν την κλάση Jeroo με void μεθόδους, αλλά δεν μπορούν να αναπτύξουν κατηγορήματα (predicates). Σύμφωνα βέβαια με τους δημιουργούς του Jeroo, οι παραπάνω δυνατότητες δεν ενσωματώθηκαν στον μικρόκοσμο γιατί θεωρήθηκαν περιττές, αφού σκοπός του μικρόκοσμου είναι οι σπουδαστές να μάθουν τις βασικές έννοιες που σχετίζονται με τη χρήση αντικειμένων για την επίλυση προβλημάτων και την ανάπτυξη μεθόδων που καθορίζουν τη συμπεριφορά των αντικειμένων.

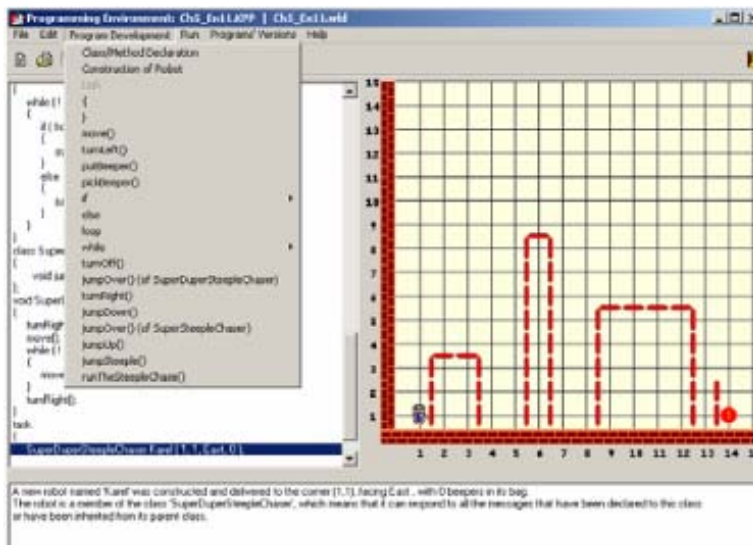


Εικόνα 12: Ο προγραμματιστικός μικρόκοσμος βασισμένος στα ρομπότ, Jeroo.

Ο μικρόκοσμος **objectKarel** βασίζεται στον Karel++ και υποστηρίζει τη διδασκαλία και εκμάθηση όλων των βασικών αντικειμενοστρεφών εννοιών: αντικείμενο, μήνυμα, κλάση, κληρονομικότητα, πολυμορφισμός και υποσκέλιση (Ξυνόγαλος, 2006). Επιπλέον, υποστηρίζει τις καθιερωμένες δομές επιλογής και επανάληψης. Ο μικρόκοσμος objectKarel ενσωματώνει:

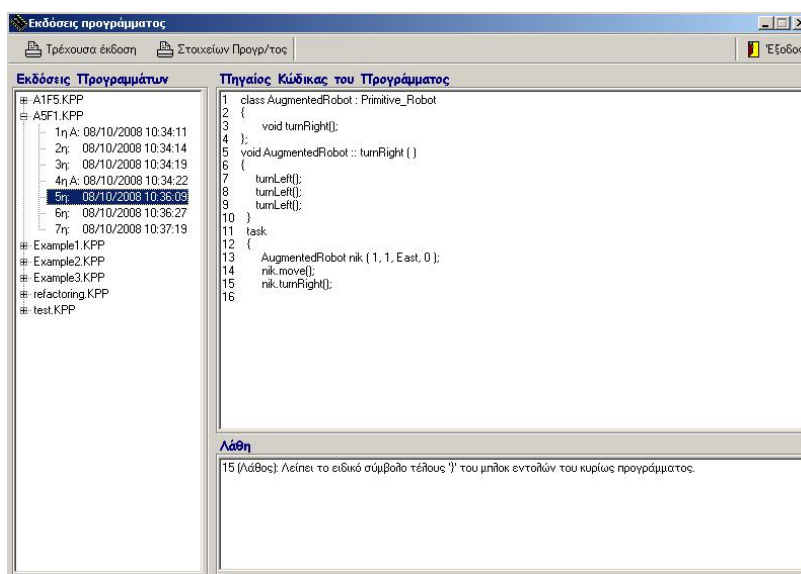
- το απαραίτητο διδακτικό υλικό (σύντομη και περιεκτική θεωρία, εγχειρίδιο χρήσης)
- δραστηριότητες για την εξοικείωση των μαθητών με τις έννοιες πριν την εφαρμογή τους στα πλαίσια ανάπτυξης προγραμμάτων
- ένα συντάκτη δομής για την εύκολη ανάπτυξη προγραμμάτων και την αποφυγή επικέντρωσης στις λεπτομέρειες της γλώσσας προγραμματισμού
- τη δυνατότητα της επεξηγηματικής οπτικοποίησης, εμφάνισης δηλαδή μηνυμάτων σε φυσική γλώσσα για την τρέχουσα κάθε φορά εντολή κατά τη βηματική εκτέλεση των προγραμμάτων.

Στο σημείο αυτό αξίζει ωστόσο να αναφερθούν κάποιες επιπρόσθετες δυνατότητες που παρέχονται στον εκπαιδευτικό. Για τα περισσότερα προβλήματα που καλούνται να λύσουν οι μαθητές πρέπει να υπάρχει μια συγκεκριμένη αρχική κατάσταση του κόσμου, η οποία δημιουργείται χρησιμοποιώντας τεχνικές άμεσης διαχείρισης (κάνοντας κλικ μεταξύ διασταυρώσεων εμφανίζονται τμήματα τοίχου και κάνοντας κλικ πάνω σε διασταυρώσεις beepers). Επίσης, υπάρχει δυνατότητα επέμβασης στην τρέχουσα κατάσταση του κόσμου ανά πάσα στιγμή. Ο εκπαιδευτικός μπορεί, στα πλαίσια παρουσίασης παραδειγμάτων, να αλλάξει την κατάσταση του κόσμου κατά την εκτέλεση ενός προγράμματος και να ζητήσει από τους μαθητές να προσδιορίσουν ποιο θα είναι το αποτέλεσμα της εκτέλεσής του από εκείνο το σημείο και μετά, ελέγχοντας έτσι το βαθμό κατανόησης των διδασκόμενων εννοιών. Επίσης, ο μαθητής έχει τη δυνατότητα να πειραματιστεί και να διερευνήσει ποιο θα είναι το αποτέλεσμα της εκτέλεσης ενός τμήματος κώδικα ή ενός προγράμματος για μια διαφορετική κατάσταση του κόσμου και να λύσει τυχόν απορίες, χωρίς να απαιτείται να διακόψει την εκτέλεση του προγράμματος, να αλλάξει την κατάσταση του κόσμου και να εκτελέσει ξανά το πρόγραμμα από την αρχή.



Εικόνα 13: Ο προγραμματιστικός μικρόκοσμος βασισμένος στα ρομπότ, objectKarel.

Επίσης, στον μικρόκοσμο objectKarel έχει ενσωματωθεί η δυνατότητα καταγραφής των ενεργειών των σπουδαστών κατά την ανάπτυξη και αποσφαλμάτωση των προγραμμάτων. Συγκεκριμένα, κάθε φορά που ο μαθητής μεταγλωττίζει ένα πρόγραμμα το σύστημα αποθηκεύει αυτόματα τόσο τον πηγαίο κώδικα όσο και τα αποτελέσματα της μεταγλώττισης. Η ιστορία των μεταγλωττίσεων παρουσιάζεται σε ένα διαφορετικό παράθυρο με τη μορφή ενός δένδρου δύο επιπέδων. Στο πρώτο επίπεδο παρουσιάζονται τα ονόματα όλων των αρχείων των προγραμμάτων και στο δεύτερο επίπεδο ο αύξων αριθμός, η ημερομηνία και η ώρα της κάθε έκδοσης. Οι γραμμές του πηγαίου κώδικα είναι αριθμημένες για την ευκολότερη μελέτη των εκδόσεων ενός προγράμματος. Η δυνατότητα της καταγραφής των ενεργειών των μαθητών επιτρέπει: (1) την καταγραφή των δυσκολιών και των παρανοήσεων των μαθητών κατά την εισαγωγή τους στον αντικειμενοστρεφή προγραμματισμό, (2) τη διερεύνηση των στρατηγικών επίλυσης προβλημάτων των μαθητών, αφού κάθε έκδοση του προγράμματος αντιπροσωπεύει ένα βήμα της διαδικασίας επίλυσής τους, και (3) στο διδάσκοντα να προσαρμόσει το μάθημα στις ανάγκες των μαθητών.



Εικόνα 14: Παρουσίαση των εκδόσεων και των αντίστοιχων αποτελεσμάτων μεταγλώττισης ενός προγράμματος, στον μικρόκοσμο objectKarel.

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

Άλλα περιβάλλοντα προγραμματιστικών μικρόκοσμων

Το **Stagecast Creator** (Stagecast Software, 1997) είναι ένα περιβάλλον οπτικού προγραμματισμού που εμφανίστηκε αρχικά το 1997 από την εταιρεία Stagecast Software. Το Stagecast Creator ξεκίνησε σαν πρόγραμμα με το όνομα KidSim από τους Allen Cypher και David Canfield Smith στο τμήμα της Apple Advanced Technology Group (ATG). Το 1994 το πρόγραμμα μετονομάστηκε Cocoa το οποίο μετά από διάφορες εταιρικές αλλαγές στην Apple κατέληξε στην σημερινή ονομασία του περιβάλλοντος.

Το περιβάλλον αυτό δημιουργήθηκε με σκοπό την χρήση του για την διδασκαλία προγραμματισμού σε παιδιά. Μπορεί να χρησιμοποιηθεί για την δημιουργία δισδιάστατων προσομοιώσεων, κινούμενων σχεδίων και παιχνιδιών σε οποιαδήποτε πλατφόρμα υποστηρίζει Java.

Η δημιουργία του κόσμου μέσα στο Stagecast Creator ξεκινάει με την επιλογή / δημιουργία του φόντου της σκηνής που θα υπάρχει ο υπό σχεδίαση κόσμος. Υπάρχει η δυνατότητα ύπαρξης πολλών «πιστών» που θα εναλλάσσονται κατά την εκτέλεση του κόσμου. Εκτός από το φόντο σε κάθε πίστα μπορούν να προστεθούν με εισαγωγή / δημιουργία αντικείμενα και προγραμματιζόμενοι χαρακτήρες. Ο προγραμματισμός αυτών των χαρακτήρων βασίζεται στο μοντέλο προγραμματισμός με επίδειξη (programming by demonstration). Σε κάθε χαρακτήρα ανατίθεται μια λίστα οδηγιών που καθορίζουν την συμπεριφορά και τις ιδιότητες / μεταβλητές του. Οι οδηγίες δημιουργούνται με την επίδειξη του τι κάνει ο χαρακτήρας σε μια συγκεκριμένη περίπτωση. Κάθε οδηγία έχει την μορφή πριν/μετά κατάσταση, που δηλώνει ότι όταν οι «πριν» ορισμένες συνθήκες της οδηγίας ισχύουν τότε οι «μετά» ορισμένες ενέργειες αυτής θα εκτελεστούν.

Σημαντικό χαρακτηριστικό του περιβάλλοντος είναι μια σειρά διαδραστικών οδηγιών εκμάθησης του Stagecast Creator (και στα ελληνικά) που έχει πρόσβαση ο χρήστης με την εκκίνηση του περιβάλλοντος.

Βασικά συστατικά του εκπαιδευτικού περιβάλλοντος οπτικού προγραμματισμού Stagecast Creator είναι:

- Το κεντρικό παράθυρο του περιβάλλοντος όπου περιέχεται η σκηνή που λαμβάνει χώρα η εκτέλεση του κόσμου που έχει δημιουργηθεί και είναι δυνατές οι τυχόν αλληλεπιδράσεις του χρήστη με το περιβάλλον.
- Μενού με βασικές ενέργειες διαχείρισης αρχείων, πληροφοριών κ.ά..
- Μενού με εικονίδια ενεργειών στο κάτω μέρος της σκηνής όπως εισαγωγής χαρακτήρα, σχεδίασης χαρακτήρα, δημιουργία κανόνα χαρακτήρα, πλήκτρα ελέγχου εκτέλεσης του κόσμου, αντιγραφή και διαγραφή χαρακτήρα/αντικειμένου.
- Μενού με εικονίδια για διαχείριση των πόρων του κόσμου όπου μπορεί επιλογή / εισαγωγή χαρακτήρων-αντικειμένων, ειδικών χαρακτήρων -αντικειμένων, επιλογή / δημιουργία / τροποποίηση πιστών-σκηνών κόσμου, επιλογή / εισαγωγή ήχων και επιλογές του υπό σχεδίαση κόσμου.
- Παράθυρο ιδιοτήτων χαρακτήρα - αντικειμένου όπου φαίνονται οι οδηγίες που ορίζουν την συμπεριφορά του καθώς και οι μεταβλητές που καθορίζουν τα χαρακτηριστικά του και οι μεταβλητές που έχει δημιουργήσει ο χρήστης.
- Παράθυρο ορισμού / τροποποίησης λεπτομερειών οδηγίας χαρακτήρα-αντικειμένου.



Εικόνα 15: Ο προγραμματιστικός μικρόκοσμος δημιουργίας παιχνιδιών και προσομοιώσεων, Stagecast Creator.

Το **Kodu** (Microsoft Research, 2009) είναι ένα περιβάλλον οπτικού προγραμματισμού, που πρωτοεμφανίστηκε το 2009, αναπτύχθηκε από τα FUSE Labs της Microsoft με σκοπό την σχεδίαση παιχνιδιών και γενικότερα αλληλεπιδραστικών τρισδιάστατων κόσμων. Σχεδιάστηκε για να λειτουργεί μόνο σε Xbox 360 και λειτουργικά Microsoft. Η σχεδίαση του Kodu βασίστηκε σε αρχές σχεδιασμού άλλων εκπαιδευτικών περιβαλλόντων προγραμματισμού όπως της Logo, του AgentSheets, και Alice. Σχεδιάστηκε έτσι ώστε να είναι ένα δημιουργικό και ευχάριστο περιβάλλον για τον κάθε χρήστη αλλά ιδιαίτερα για να είναι προσιτό και εύχρηστο σε παιδιά μικρής ηλικίας. Κάποιες από τις ιδιαιτερότητες του περιβάλλοντος Kodu είναι το παραμετροποιήσιμο τρισδιάστατο περιβάλλον σχεδίασης και ο ιδιαίτερος τρόπος προγραμματισμού που εμπεριέχει.

Ο προγραμματισμός στο Kodu έχει απαλλαχθεί από την εφαρμογή πολλών βασικών εννοιών προγραμματισμού που βρίσκουν εφαρμογή σε άλλα εκπαιδευτικά περιβάλλοντα προγραμματισμού. Ο προγραμματισμός βασίζεται σε μια σειρά από οδηγίες της μορφής, όταν συμβεί κάτι τότε κάνει αυτό (WHEN... DO...). Ένα παράδειγμα εντολής γραμμένης με περιγραφικό τρόπο είναι: When – see – red – fruit > Do – move – towards – quickly. Βέβαια ο προγραμματισμός στο Kodu γίνεται με οπτικό τρόπο, επιλέγοντας με το ποντίκι συνθήκες και ενέργειες που θα συμβούν όταν ισχύει η συνθήκη, και προσθέτοντας τα στην οδηγία που θα δημιουργηθεί. Για να λάβει χώρα ο προγραμματισμός προηγουμένως πρέπει να έχει δημιουργηθεί ένας κόσμος τριών διαστάσεων (με την πληθώρα των εργαλείων που παρέχονται) και να έχουν προστεθεί τα αντικείμενα και χαρακτήρες (από ένα μεγάλο αριθμό που επίσης παρέχεται από το περιβάλλον) που θα δρουν μέσα σε αυτόν. Η συμπεριφορά κάθε αντικειμένου – χαρακτήρα και ο τρόπος αλληλεπίδρασης αυτού με τον κόσμο είναι αυτά που καθορίζονται με τον προγραμματισμό.

Βασικά συστατικά του περιβάλλοντος Kodu είναι:

- Το κύριο μενού που εμφανίζεται με την είσοδο στο Kodu, όπου μπορεί επιλεγθεί η δημιουργία ενός νέου κόσμου «Νέος κενός κόσμος», η επιλογή εμφάνισης κόσμων που υπάρχουν τοπικά στον υπολογιστή «Ο κόσμος του Kodu» (κόσμοι δημιουργημένοι από τον χρήστη, κόσμοι από την εγκατάσταση του Kodu και κόσμοι οι οποίοι έχουν μεταφορτωθεί τοπικά από την κοινότητα), η επιλογή εμφάνισης κόσμων που υπάρχουν στην κοινότητα του Kodu «Κοινότητα» (κόσμοι του χρήστη εάν έχει μοιραστεί κάποιους αλλά και άλλων χρηστών), η επιλογή ρυθμίσεων

γενικών παραμέτρων του περιβάλλοντος «Επιλογές», η επιλογή «Βοήθεια» και «Έξοδος από το Kodu».

- Το βασικό περιβάλλον σχεδίασης του Kodu με τις οδηγίες χρήσης στο πάνω αριστερά μέρος, το μενού εικονιδίων στο κάτω μέρος με διάφορα εργαλεία και επιλογές όπως Αρχικό μενού, Πάίζει παιχνίδι, Μετακίνηση κάμερας, εργαλείο προσθήκης ή επεξεργασίας χαρακτήρων και αντικειμένων, επιλογή Αλλαγή ρυθμίσεων κόσμου, και διάφορα άλλα εργαλεία.
- Τον κεντρικό χώρο όπου γίνεται η σχεδίαση του κόσμου από τον χρήστη με την χρήση των εργαλείων του μενού εικονιδίων.
- Την περιοχή ορισμού των οδηγιών που καθορίζουν την συμπεριφορά και την αλληλεπίδραση των χαρακτήρων και αντικειμένων του κόσμου.
- Το παράθυρο εκτέλεσης του κόσμου που δημιουργείται, όπου παρατηρείται η συμπεριφορά και η αλληλεπίδραση των αντικειμένων και χαρακτήρων του κόσμου, που έχει ρυθμιστεί σύμφωνα με τις οδηγίες που ορίστηκαν από τον χρήστη.
- Το αρχικό μενού όπου μπορεί ο χρήστης να αποθηκεύσει τον κόσμο που δημιουργεί, να εκτυπώσει τον κώδικα που έχει δημιουργήσει στον κόσμο του αλλά κ.ά.



Εικόνα 16: Ο προγραμματιστικός μικρόκοσμος δημιουργίας παιχνιδιών Kodu, με προγραμματιστικό μοντέλο αντικειμενοστραφές και οδηγούμενο από γεγονότα.

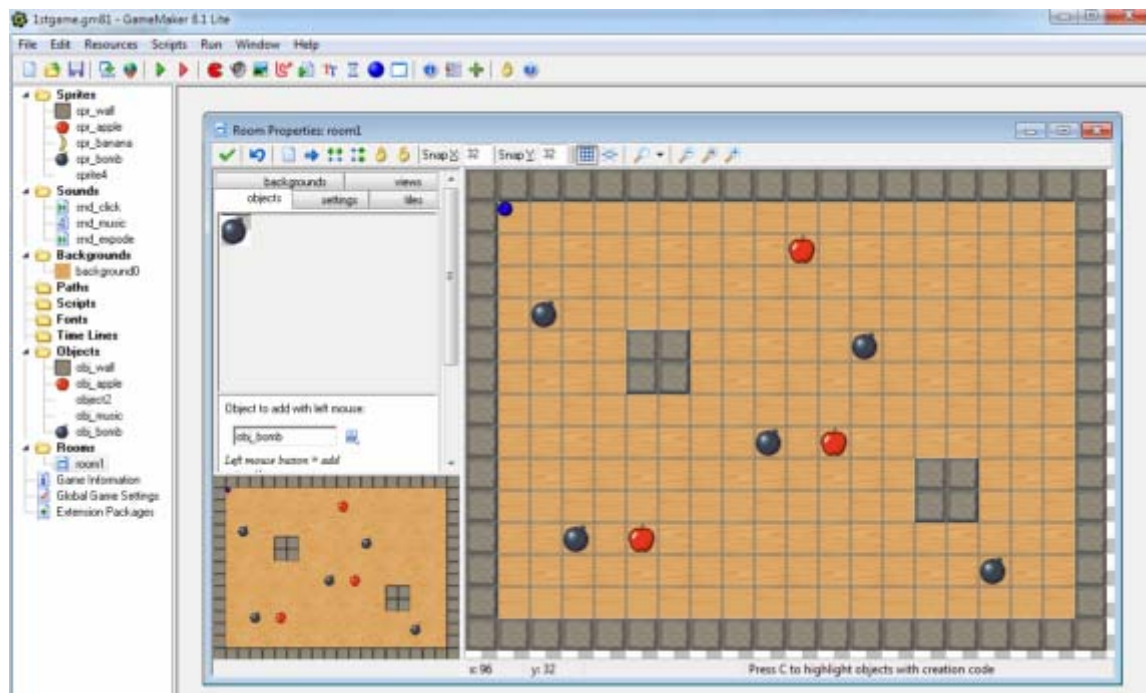
Το **GameMaker** (Overmars, 1999) είναι ένα περιβάλλον ανάπτυξης παιχνιδιών που αρχικά δημιουργήθηκε το 1999 από τον Mark Overmars σε γλώσσα προγραμματισμού Delphi. Τώρα το περιβάλλον πλέον αναπτύσσεται και διατίθεται από την εταιρεία YoYo Games. Το GameMaker σχεδιάστηκε έτσι ώστε να επιτρέπει την γρήγορη δημιουργία παιχνιδιών ακόμα και από χρήστες που δεν έχουν εμπειρία στον προγραμματισμό. Το 2012 κυκλοφόρησε η πρώτη έκδοση του

GameMaker Studio, η οποία γράφτηκε σε C#, και επιτρέπει την δημιουργία παιχνιδιών και την έκδοσή τους σε πολλές διαφορετικές πλατφόρμες. Η τρέχουσα έκδοση του GameMaker Studio υποστηρίζει την δημιουργία παιχνιδιών για Windows, Mac-OS, iOS, Android, σε HTML5, Windows 8, Windows 8 phone και Linux.

Στο περιβάλλον αλληλεπίδρασης του **GameMaker** γίνεται χρήση ενός συστήματος συρσίματος και εναπόθεσης, επιτρέποντας σε χρήστες που δεν είναι εξοικειωμένοι με τον προγραμματισμό να δημιουργούν διαισθητικώς παιχνίδια οργανώνοντας με οπτικό τρόπο εικονίδια στην οθόνη του υπολογιστή. Αυτά τα εικονίδια αντιπροσωπεύουν ενέργειες που συμβαίνουν σε ένα παιχνίδι, όπως κινήσεις, σχεδίαση και απλές δομές ελέγχου. Για πιο προχωρημένους χρήστες το GameMaker περιέχει μια ενσωματωμένη γλώσσα προγραμματισμού την Game Maker Language (GML), επιτρέποντας έτσι την δημιουργία πιο πολύπλοκων παιχνιδιών.

Κάποια βασικά συστατικά του περιβάλλοντος δημιουργίας παιχνιδιών GameMaker είναι:

- Το μενού επιλογών και η μπάρα εργαλείων με εικονίδια - συντομεύσεις λειτουργιών του μενού επιλογών.
- Η περιοχή με τους πόρους του παιχνιδιού όπου παρατίθενται, σε μορφή λίστας, όλα τα είδη των πόρων του παιχνιδιού (γραφικά στοιχεία, ήχοι, φόντα, προγραμματιζόμενα αντικείμενα, δωμάτια, πληροφορίες για το παιχνίδι κ.ά.).
- Παράθυρο ιδιοτήτων γραφικού στοιχείου όπου επιλέγεται/σχεδιάζεται/τροποποιείται η εικόνα που θα αντιστοιχεί σε αυτό.
- Παράθυρο ιδιοτήτων προγραμματιζόμενου αντικείμενου όπου λαμβάνει χώρα ο προγραμματισμός αυτού (ορισμός γεγονότος και των ενεργειών που θα εκτελεστούν όταν αυτό συμβεί). Εδώ επίσης επιλέγεται αν το αντικείμενο αντιστοιχεί σε κάποιο ήδη ορισμένο γραφικό στοιχείο και αν κληρονομεί τις ιδιότητες από κάποιο άλλο (γονικό) αντικείμενο.
- Παράθυρο ιδιοτήτων των δωματίου (πάνω στο οποίο θα εξελίσσεται το παιχνίδι) όπου συμβαίνει η σχεδίαση αυτού.
- Παράθυρο εκτέλεσης του παιχνιδιού.



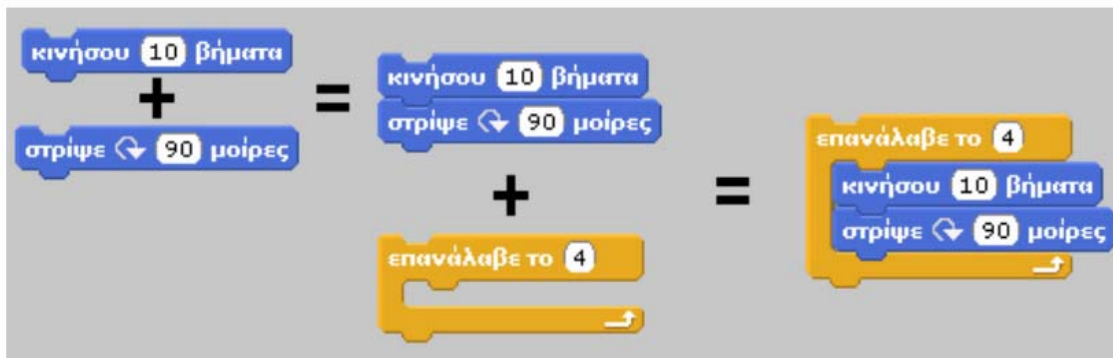
Εικόνα 17: Ο προγραμματιστικός μικρόκοσμος δημιουργίας παιχνιδιών Game Maker.

2.2 Περιβάλλοντα προγραμματιστικών μικρόκοσμων με Οπτικό Προγραμματισμό με Πλακίδια – Η περίπτωση του Scratch του MIT

Η «οικογένεια» περιβαλλόντων προγραμματιστικών μικρόκοσμων βασισμένων σε Οπτικό Προγραμματισμό με Πλακίδια

Μια άλλη δυνατότητα που προσφέρουν ορισμένα σύγχρονα logo-like προγραμματιστικά περιβάλλοντα είναι η διεπαφή «Οπτικού Προγραμματισμού με Πλακίδια» τα οποία συνδυάζονται μεταξύ τους ως κομμάτια παζλ.

Πλεονέκτημα του Οπτικού Προγραμματισμού είναι ότι ο μαθητής αποφεύγει τα συντακτικά λάθη και εστιάζει στον αλγόριθμο.



Εικόνα 18: Περιβάλλον οπτικού προγραμματισμού με πλακίδια (π.χ. Scratch).

Αξίζει να σημειωθεί ότι η μετάβαση από το ένα περιβάλλον στο άλλο βοηθάει τους μαθητές να αναπτύξουν δεξιότητες για μελλοντικές μεταβάσεις σε άλλα προγραμματιστικά περιβάλλοντα που θα είναι απαραίτητες όταν θα αναπτυχθούν νέα προγραμματιστικά περιβάλλοντα που σήμερα ίσως δεν μπορούμε ούτε να φανταστούμε τις δυνατότητές τους.



Εικόνα 19: Μετάβαση από κλασικό Logo like περιβάλλον προγραμματισμού σε περιβάλλον οπτικού προγραμματισμού με πλακίδια.

Βασικά κριτήρια επιλογής προγραμματιστικού περιβάλλοντος από τον εκπαιδευτικό είναι τα εξής:

- Να ικανοποιεί τους στόχους του Προγράμματος Σπουδών (Το Πρόγραμμα Σπουδών όσον αφορά τον προγραμματισμό βασίζεται στη χρήση διαδικασιών).
- Να ανήκει σε ομάδα λογισμικών με κοινή φιλοσοφία (logo).
- Να διαθέτει διεπαφή Οπτικού Προγραμματισμού (με ή χωρίς πλακίδια).
- Να είναι ελεύθερο λογισμικό.
- Να είναι εξελληνισμένο (πολύ σημαντικό για μαθητές δημοτικού).
- Να υπάρχει κοινότητα μάθησης εστιασμένης στους σκοπούς του μαθήματος.
- Να είναι εξοικειωμένος ο εκπαιδευτικός με το λογισμικό.

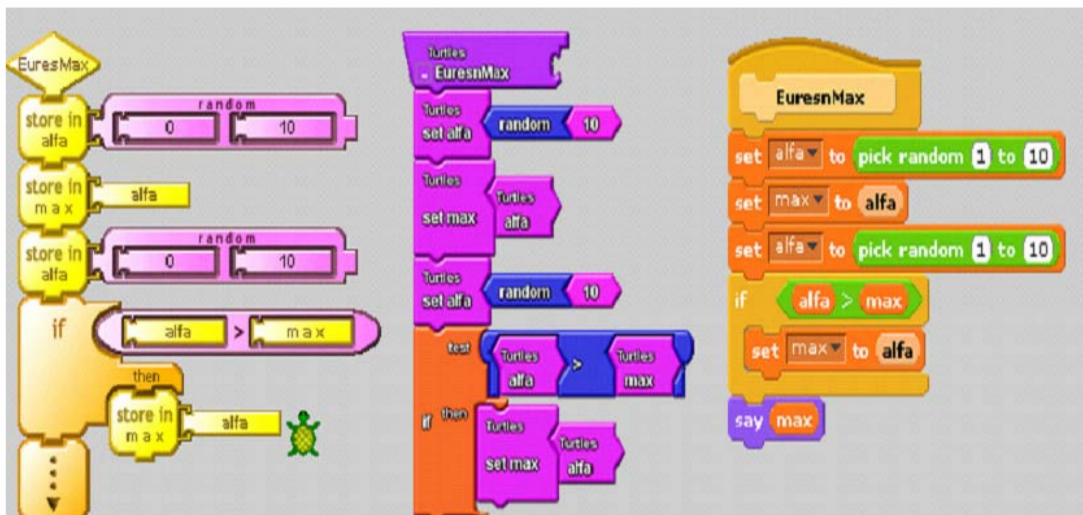


Εικόνα 20: Τεράστια η σημασία του εξελληνισμένου περιβάλλοντος για τους μαθητές μικρής ηλικίας.

Από τα γνωστά logo-like προγραμματιστικά περιβάλλοντα με Οπτικό Προγραμματισμό με Πλακίδια που διαθέτουν διαδικασίες, αυτά που πληρούν τα παραπάνω βασικά κριτήρια επιλογής προγραμματιστικού περιβάλλοντος από τον εκπαιδευτικό, σε μικρότερο ή μεγαλύτερο βαθμό είναι τα παρακάτω:

TurtleArt

StarLogo TNG και BYOB (Scratch)



Εικόνα 21: Περιβάλλοντα οπτικού προγραμματισμού με πλακίδια, με φιλοσοφία logo-like μικρόκοσμων, που διαθέτουν διαδικασίες.

Τα παραπάνω προτεινόμενα περιβάλλοντα προγραμματιστικών μικρόκοσμων, δεν παρέχουν όλα τις ίδιες προγραμματιστικές δυνατότητες.

Το **TurtleArt** (Silverman B., 2004) είναι ένα περιβάλλον οπτικού προγραμματισμού που σχεδιάστηκε το 2004 από τον Brian Silverman και διατηρείται από τον Walter Bender. Αποτελεί ουσιαστικά μια υλοποίηση της γλώσσας Logo σε ένα περιβάλλον οπτικού προγραμματισμού. Το περιβάλλον TurtleArt γράφτηκε αρχικά σε Java, και μετέπειτα ξαναγράφηκε σε Python για τα OLPC (One Laptop Per Child). Το περιβάλλον TurtleArt επικεντρώνεται στην κατασκευή εικόνων, ενώ επιτρέπει την εξερεύνηση του πεδίου της γεωμετρίας όπως και του προγραμματισμού.

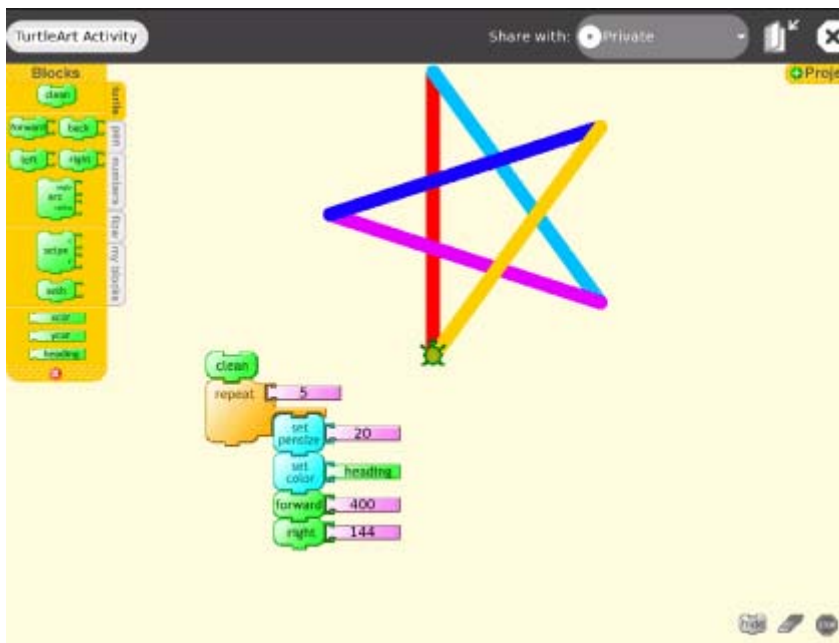
Στο **TurtleArt** ο προγραμματισμός λαμβάνει χώρα με την δημιουργία σειρών πλακιδίων εντολών, με την επιλογή και την εναπόθεση τους στο χώρο προγραμματισμού - εκτέλεσης. Τα πλακίδια έχουν την μορφή κομματιών παζλ ώστε να διευκολύνεται ο χρήστης κατά την δημιουργία του προγράμματος. Η αλληλουχία των πλακιδίων εντολών ουσιαστικά αντιπροσωπεύει ένα πρόγραμμα που περιγράφει μια εικόνα. Τα αποτελέσματα της εκτέλεσης προγραμμάτων, δηλαδή η εμφάνιση του σχεδίου που δημιουργεί το πρόγραμμα, εμφανίζονται στην ίδια περιοχή που γίνεται και ο προγραμματισμός της χελώνας του περιβάλλοντος. Βέβαια υπάρχει και η δυνατότητα απομόνωσης είτε του προγράμματος είτε του αποτελέσματος της εκτέλεσης – του σχεδίου.

Βασικά συστατικά του περιβάλλοντος είναι:

- Το μενού επιλογών στο πάνω μέρος του προγράμματος όπου υπάρχουν μια σειρά από εργασίες αρχείων και άλλων απλών εργασιών.
- Η περιοχή παλετών πλακιδίων εντολών όπου υπάρχουν η παλέτα Turtle (που αφορά πλακίδια κίνησης, ορισμού θέσης και ανάγνωσης χαρακτηριστικών - ιδιοτήτων χελώνας), η παλέτα Pen (με πλακίδια χρήσης πέννας σχεδίασης και ανάγνωσης χαρακτηριστικών - ιδιοτήτων πέννας), η παλέτα Numbers (με πλακίδια δημιουργίας αριθμητικών παραστάσεων κ.ά.), η παλέτα Flow (με πλακίδια ελέγχου ροής και επαναλήψεων) και η παλέτα My Blocks (με πλακίδια δημιουργίας διαδικασιών, μεταβλητών και ανάγνωσης των τιμών αυτών).
- Η περιοχή προγραμματισμού και εμφάνισης των αποτελεσμάτων της σχεδίασης του προγράμματος.

Προγραμματιστικές δυνατότητες του περιβάλλοντος οπτικού προγραμματισμού με πλακίδια, **TurtleArt**:

- Η (πρωτογενής) εντολή σχεδίασης τόξων.
- Διαθέτει μόνο δύο μεταβλητές.
- Οι διαδικασίες της δεν έχουν μηχανισμούς περάσματος παραμέτρων.
- Περιορισμένη στη γεωμετρία της χελώνας, χωρίς πολυμεσικές δυνατότητες.



Εικόνα 22: Περιβάλλον οπτικού προγραμματισμού μικρόκοσμων με πλακίδια, TurtleArt.

Το περιβάλλον προγραμματισμού **StarLogo** είναι ένα εκπαιδευτικό περιβάλλον προγραμματισμού που αναπτύχθηκε από τον Mitchel Resnick, Eric Klopfer, και άλλους στο MIT Media Lab και το MIT Teacher Education Program στη Μασαχουσέτη. Αρχικά η StarLogo δημιουργήθηκε με σκοπό να αποτελέσει μια γλώσσα προσομοίωσης βασισμένη σε πράκτορες. Σχεδιασμένη για την εκπαίδευση, η StarLogo μπορεί να χρησιμοποιηθεί από τους μαθητές για την δημιουργία προσομοιώσεων.

Η πρώτη έκδοση **StarLogo TNG** (MIT Media Lab, 2008) εμφανίστηκε το 2008 αποτελώντας μια σημαντική μετάβαση από τον δισδιάστατο κόσμο της αρχικής StarLogo στον τρισδιάστατο κόσμο της StarLogo TNG (The Next Generation). Το περιβάλλον προγραμματισμού της νέας αυτής έκδοσης επιστρατεύοντας τον τρισδιάστατο αυτό κόσμο, που δημιουργήθηκε με χρήση OpenGL γραφικών, και μιας οπτικής γλώσσας προγραμματισμού, που βασίζεται σε πλακίδια εντολών, ενισχύει την ευκολία εκμάθησης και χρήσης αυτού του περιβάλλοντος. Επιπλέον δίνει και την δυνατότητα εισόδου από το πληκτρολόγιο καθιστώντας το έτσι ένα σπουδαίο εργαλείο για τον προγραμματισμό εκπαιδευτικών παιχνιδιών. Η StarLogo TNG όντας γραμμένη σε Java λειτουργεί στους περισσότερους υπολογιστές.

Ο προγραμματισμός γίνεται με σύρσιμο και εναπόθεση πλακιδίων εντολών στην περιοχή προγραμματισμού δημιουργώντας τελικά ακολουθίες πλακιδίων που αντιπροσωπεύουν το πρόγραμμα που θα εκτελεστεί.

Βασικά συστατικά του περιβάλλοντος προγραμματισμού της StarLogo TNG είναι:

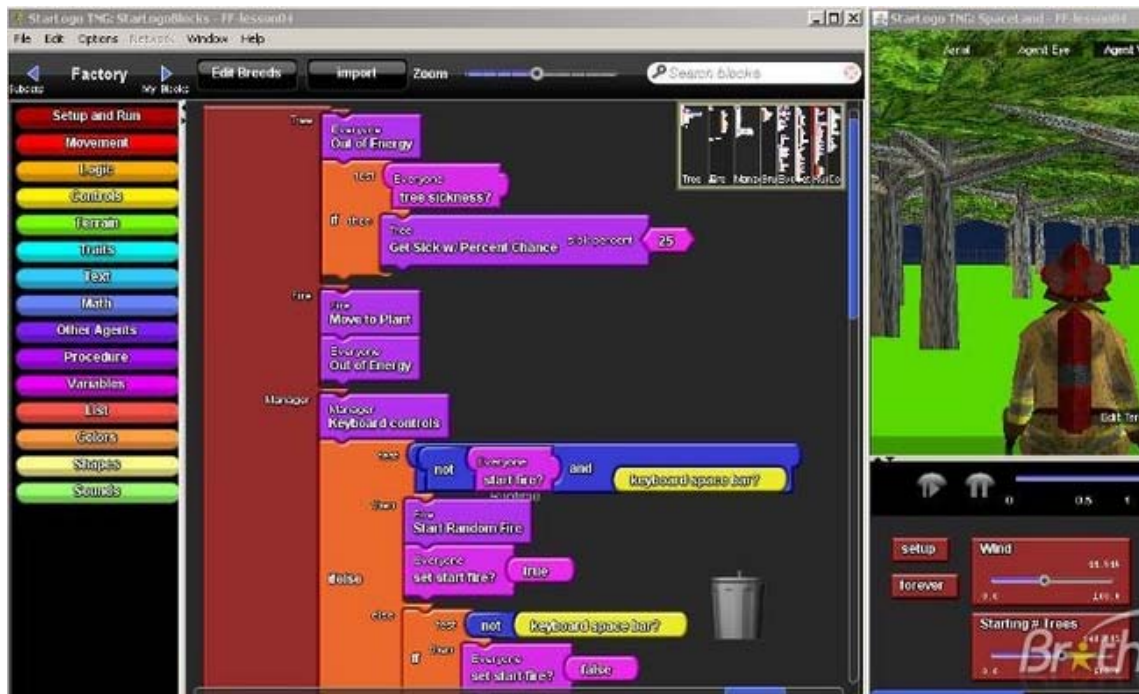
- Το μενού επιλογών, όπου υπάρχουν ενέργειες διαχείρισης αρχείων, επεξεργασίας προγράμματος, διαμόρφωσης περιβάλλοντος εργασίας κ.ά.
- Η περιοχή με τις παλέτες των πλακιδίων εντολών, όπου εκεί υπάρχουν τρεις υποομάδες παλετών: η Factory (με παλέτες εγκατάσταση και εκτέλεση, κίνηση, λογική, έλεγχοι, τερν, ίχνη, κείμενο, μαθηματικά, και άλλα), η My Blocks και η Subsets. Η δύο τελευταίες εμπλουτίζονται με πλακίδια εντολών κατά την διάρκεια δημιουργίας του προγράμματος από τον χρήστη (με την προσθήκη πρακτόρων, την δημιουργία διαδικασιών κ.ά.).
- Η περιοχή προγραμματισμού, όπου εκεί γίνεται η εναπόθεση και η «συναρμολόγηση» των διαφόρων πλακιδίων εντολών ώστε τελικά να διαμορφωθεί τελικά το πρόγραμμα που θα εκτελείται. Η περιοχή αυτή χωρίζεται σε διακριτές

υποπεριοχές, την περιοχή του εκάστοτε πράκτορα που έχει προστεθεί στον κόσμο (τοποθετούνται πλακίδια εντολών που αφορούν τον προγραμματισμό της συμπεριφοράς του πράκτορα), την περιοχή Όλοι (εδώ συμβαίνει προγραμματισμός που αφορά όλους τους πράκτορες του περιβάλλοντος), την περιοχή Εγκατάσταση (όπου συμβαίνει προγραμματισμός που αφορά την αρχικοποίηση του υπό δημιουργία κόσμου), η περιοχή Συγκρούσεις (όπου ορίζεται η συμπεριφορά των πρακτόρων που έχουν επιλεχτεί σε περίπτωση σύγκρουσης), και την περιοχή Εκτέλεση (όπου ουσιαστικά συμβαίνει ο συντονισμός και ορισμός του τρόπου εκτέλεσης όλων των παραπάνω).

- Την σκηνή που είναι ένας νοητός κόσμος τριών διαστάσεων, όπου εκεί λαμβάνει χώρα η εκτέλεση και παρατήρηση του κόσμου που έχει δημιουργηθεί.
- Την περιοχή εμφάνισης χειριστηρίων αρχικοποίησης και εκτέλεσης του κόσμου που έχει δημιουργηθεί όπως και απεικόνισης διαφόρων πληροφοριών που έχουν οριστεί να εμφανίζονται κατά την διάρκεια της εκτέλεσης του κόσμου. Στην ίδια περιοχή συμβαίνει και η δημιουργία, τροποποίηση του φόντου του κόσμου μας (υπάρχει η δυνατότητα ύπαρξης πολλών επιπέδων φόντων ή levels και εναλλαγής μεταξύ αυτών κατά την διάρκεια της εκτέλεσης).

Προγραμματιστικές δυνατότητες του περιβάλλοντος οπτικού προγραμματισμού με πλακίδια, **StarLogo**:

- Διαθέτει μόνο εντολή επανάληψης για προκαθορισμένο πλήθος επαναλήψεων.
- Οι διαδικασίες της έχουν μηχανισμούς περάσματος παραμέτρων.
- Τρισδιάστατο terrain, προσομοιώσεις φυσικών συστημάτων & μοντελοποίηση, «αυθεντικά» παιχνίδια.
- Διαχείριση (ιδιόρρυθμη) πολυμεσικών στοιχείων.



Εικόνα 23: Περιβάλλον οπτικού προγραμματισμού μικρόκοσμων με πλακίδια, StarLogo TNG.

Το περιβάλλον προγραμματιστικών μικρόκοσμων, βασισμένο σε Οπτικό Προγραμματισμό με Πλακίδια, Scratch του MIT.

Το περιβάλλον προγραμματιστικών μικρόκοσμων, βασισμένο σε οπτικό προγραμματισμό με πλακίδια, με κοινή φιλοσοφία Logo-like, το οποίο χρησιμοποιεί διαδικασίες και **ικανοποιεί τους στόχους του Προγράμματος Σπουδών** (Υπουργείο Παιδείας & Θρ. 2014) **με τον δυνατό ικανοποιητικότερο τρόπο** είναι το **Scratch** του MIT. Το Scratch είναι ελεύθερο λογισμικό, είναι εξελληνισμένο (με πολλές ανανεώσεις, βιβλία και μελέτες σε ελληνικά τριτοβάθμια ιδρύματα), και υπάρχει εκτεταμένη κοινότητα μάθησης εστιασμένη στους σκοπούς του μαθήματος με την χρήση του περιβάλλοντος Scratch (Πανεπιστήμιο Θεσσαλίας 2010, Φεσάκης Γ. et al. 2008, Χαραλαμπίδης Σ. 2010). Στον ιστοχώρο του Scratch υπάρχουν και πολλές άλλες πηγές μάθησης και υποστήριξης όπως: εκπαιδευτικά βίντεο, κάρτες Scratch και συχνές ερωτήσεις (<http://info.scratch.mit.edu/Support/>).

Σημαντικό χαρακτηριστικό του περιβάλλοντος και της φιλοσοφίας σχεδίασής του είναι **δυνατότητα διαμοιρασμού έργων** των χρηστών στην διαδικτυακή κοινότητα του Scratch. Ο κάθε χρήστης – δημιουργός έχει όχι μόνο την δυνατότητα δημοσίευσης του έργου του αλλά και της μεταφόρτωσης, χρήσης, τροποποίησης και επέκτασης έργων άλλων χρηστών (<http://scratchplay.wikispaces.com>).

Το **Scratch** (MIT Media Lab 2002) είναι ένα εκπαιδευτικό περιβάλλον οπτικού προγραμματισμού που αναπτύχθηκε το 2006 από το Lifelong Kindergarten group στο MIT Media Lab. Στο περιβάλλον αυτό ο προγραμματισμός λαμβάνει χώρα με ανάθεση ενεργειών, που συμβαίνει με σύρσιμο και εναπόθεση πλακιδίων εντολών (συνθηκών ελέγχου, ενεργειών κ.ά.), στους πράκτορες (agents), οι οποίοι υπάρχουν και ενεργούν σε μια σκηνή δύο διαστάσεων (stage). Με αυτόν τον τρόπο οι μαθητές μπορούν να δημιουργήσουν απλά παιχνίδια, κινούμενα σχέδια, διαδραστικές παρουσιάσεις, και να διηγηθούν ιστορίες. Οι μαθητές μπορούν, χωρίς να υπάρχει ο κίνδυνος συντακτικών λαθών και η ανάγκη απομνημόνευσης εντολών, να επικεντρωθούν στην δημιουργία του έργου τους εμπλεκόμενοι σε μια διαδικασία που περιλαμβάνει σχηματισμό ακολουθιών από πλακίδια εντολών ανάλογα το επιθυμητό αποτέλεσμα, τροποποίηση της σειράς ή και του είδους των πλακιδίων εντολών, εκτέλεση του προγράμματος και παρατήρησης του αποτελέσματος. Η εκτέλεση και τα αποτελέσματα του προγραμματισμού συμβαίνουν πάνω σε μια σκηνή δύο διαστάσεων.

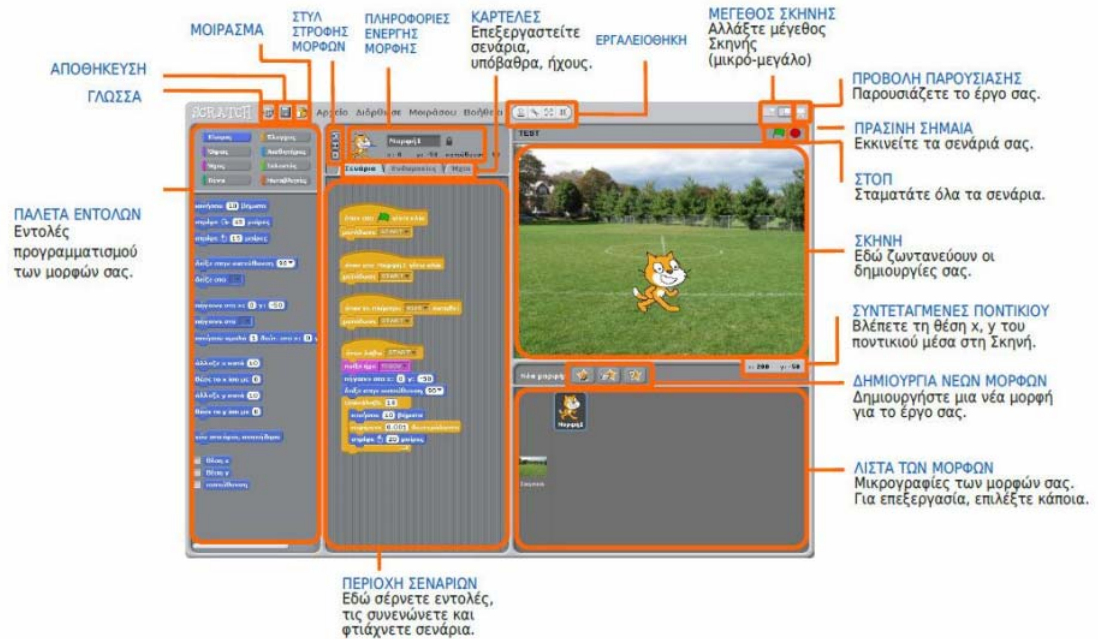
Τα έργα (project) στο **Scratch** οικοδομούνται από αντικείμενα που λέγονται μορφές (sprite). Μπορούμε να αλλάξουμε την εμφάνιση μιας μορφής δίνοντας της μια διαφορετική ενδυμασία (costume). Μπορούμε να την κάνουμε να μοιάζει με άνθρωπο, με τρένο, με πεταλούδα ή με οτιδήποτε άλλο. Για ενδυμασία μπορούμε να χρησιμοποιήσουμε οποιαδήποτε εικόνα: μπορούμε να δημιουργήσουμε μία εικόνα στον Επεξεργαστή Ζωγραφικής, να εισάγουμε μια από τον σκληρό μας δίσκο ή μία από το διαδίκτυο.

Μπορούμε να δώσουμε οδηγίες σε μια μορφή ώστε να κινηθεί, να παίξει μουσική ή να αλληλεπιδράσει με άλλες μορφές. Για να πούμε στη μορφή τι να κάνει, συνενώνουμε εικονικές εντολές (που μοιάζουν με τουβλάκια – block) μεταξύ τους σε στήλες που ονομάζονται σενάρια ενεργειών (κώδικες, script). Όταν κάνουμε κλικ σε ένα σενάριο, το Scratch «τρέχει» τις εντολές από την κορυφή του σεναρίου έως τον πάτο.

Βασικά συστατικά του περιβάλλοντος προγραμματισμού είναι:

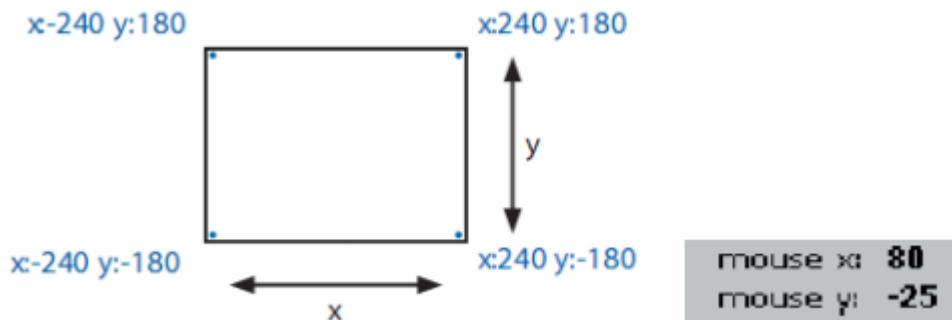
- Το **μενού επιλογών** στην πάνω πλευρά του περιβάλλοντος.
- Η περιοχή με τις **παλέτες των πλακιδίων εντολών** (παλέτες κίνηση, όψεις, ήχος, πένα, έλεγχος, αισθητήρες, τελεστές και μεταβλητές), στην αριστερή πλευρά του περιβάλλοντος.
- Η **περιοχή σεναρίων** όπου γίνεται η εναπόθεση των πλακιδίων εντολών (όπου επίσης ορίζονται οι ενδυμασίες και οι ήχοι που αντιστοιχούν στον επιλεγμένο πράκτορα ή τα υπόβαθρα και οι ήχοι που αντιστοιχούν στην σκηνή του έργου), στην κεντρική περιοχή του περιβάλλοντος.

- Η **σκηνή** όπου λαμβάνει χώρα η εκτέλεση και η παρατήρηση των αποτελεσμάτων του προγραμματισμού που έχει συμβεί, στην δεξιά πάνω περιοχή του περιβάλλοντος.
- Η **περιοχή** όπου γίνεται ο ορισμός (επιλογή έτοιμων εικόνων ή και σχεδίαση νέων) των **πρακτόρων-μορφών**, στην δεξιά κάτω περιοχή του περιβάλλοντος.



Εικόνα 24: Περιοχές του περιβάλλοντος εργασίας μικρόκοσμων με πλακίδια, Scratch.

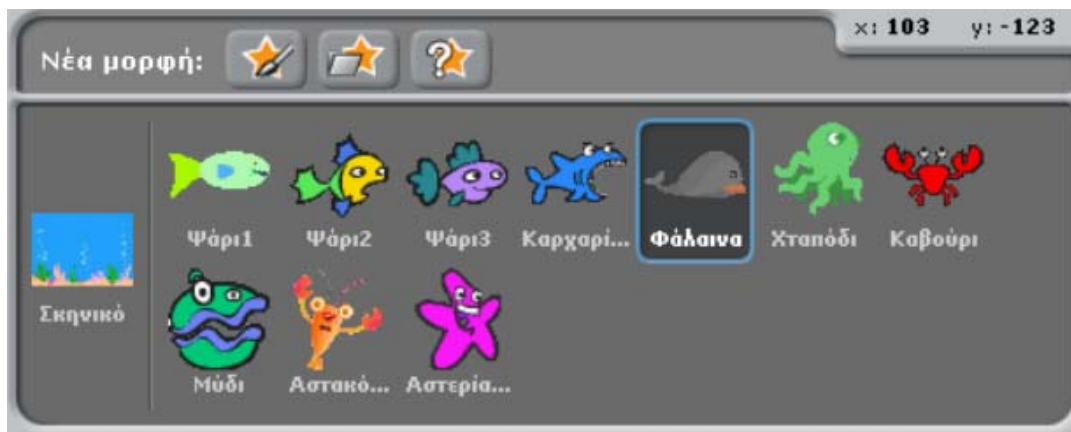
Η **Σκηνή** είναι ο χώρος όπου ζωντανεύουν οι ιστορίες μας, τα παιχνίδια και τα κινούμενα σχέδια. Οι μορφές κινούνται και αλληλεπιδρούν μεταξύ τους επάνω στη Σκηνή. Η Σκηνή έχει μήκος 480 μονάδες και ύψος 360 μονάδες. Είναι χωρισμένη σε άξονες x και y. Το κέντρο της Σκηνής έχει συντεταγμένες x:0 και y:0. Στη μέση της σκηνής το x και το y είναι ρυθμισμένο στο 0. Για να βρεις τη θέση του x και του y στη σκηνή κούνησε το ποντίκι και κοίτα το “mouse x-y display” που βρίσκεται ακριβώς κάτω απ’ τη σκηνή στα δεξιά.



Εικόνα 25: Η περιοχή της «Σκηνής» του Scratch.

Η **Λίστα των μορφών** παρουσιάζει μικρογραφίες όλων των μορφών του έργου. Το όνομα της κάθε μορφής εμφανίζεται κάτω από τη μικρογραφία της, μαζί με το πόσα κοστούμια έχει και το πόσες ενέργειες έχει. Για να δούμε ή να επεξεργαστούμε τα σενάρια, τις ενδυμασίες και τους ήχους μιας μορφής, κάνουμε κλικ πάνω στη μικρογραφία της στη Λίστα των μορφών ή κάνουμε διπλό κλικ πάνω στην ίδια τη μορφή μέσα στη Σκηνή. Η επιλεγμένη μορφή είναι μαρκαρισμένη με μπλε περίγραμμα μέσα στη Λίστα των μορφών.

Για να δούμε, να εξάγουμε, να αντιγράψουμε ή να διαγράψουμε μία μορφή-χαρακτήρα, κάνουμε δεξί κλικ στη μικρογραφία της μορφής από τη λίστα με τις μορφές. Για να εμφανιστεί μία μορφή η οποία είναι εκτός σκηνής ή κρυμμένη, πατάμε το shift και ταυτόχρονα κάνουμε κλικ στη μικρογραφία της μορφής στη λίστα με τις μορφές αυτό θα φέρει την μορφή στη μέση της σκηνής και θα την φανερώσει. Μπορούμε να αλλάξουμε τη σειρά των μορφών στη λίστα με τις μορφές κάνοντας κλικ στη μικρογραφία της μορφής και σύροντας το ποντίκι εκεί που θέλουμε να πάει η μορφή. Όπως η μορφή αλλάζει την εμφάνισή της αλλάζοντας κοστούμι, έτσι και η σκηνή μπορεί να αλλάξει την εμφάνισή της αλλάζοντας φόντο (background). Για να δούμε και να διαμορφώσουμε τις ενέργειες, το φόντο και τους ήχους που σχετίζονται με τη σκηνή κάνουμε κλικ στο εικονίδιο «σκηνή» που βρίσκεται στην αριστερή μεριά της λίστας με τις μορφές.

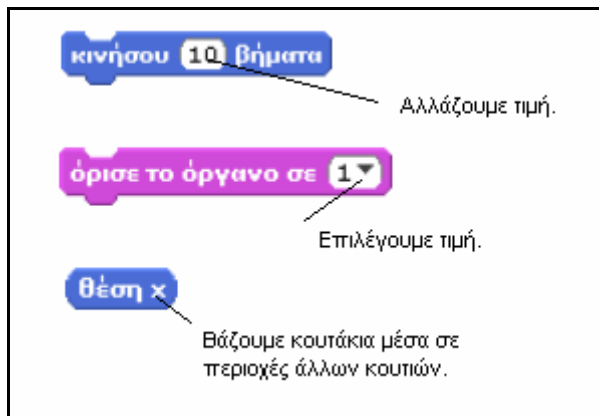


Εικόνα 26: Η περιοχή της «Λίστας των μορφών», του Scratch.

Για να προγραμματίσουμε μια μορφή, σέρνουμε εντολές (τουβλάκια) από την **Παλέτα εντολών** προς την **Περιοχή σεναρίων**. Για να εκτελεστεί μια εντολή, κάνουμε κλικ επάνω της. Δημιουργούμε σενάρια ενεργειών (προγράμματα) συνενώνοντας εντολές μεταξύ τους σε στήλες και κάνουμε κλικ οπουδήποτε στη στήλη για να εκτελεστεί ολόκληρο το σενάριο, από την αρχή ως το τέλος. Για να καταλάβουμε τι ακριβώς κάνει μια εντολή, κάνουμε δεξί κλικ πάνω της και επιλεγούμε τη βοήθεια από το αναδυόμενο μενού. Όταν σέρνουμε μία εντολή μέσα στη Περιοχή σεναρίων, μία λευκή υπογράμμιση υποδεικνύει πού μπορούμε να την αφήσουμε ώστε να δημιουργήσει μια σωστή ένωση με άλλη εντολή. Μπορούμε να εισάγουμε τουβλάκια στο μέσον της στήλης ή στο τέλος.

Για να μετακινήσουμε μια στήλη την επιλέγουμε από το πρώτο τουβλάκι. Αν επιλέξουμε να βγάλουμε ένα τουβλάκι από τη μέση της στήλης, όλα τα τουβλάκια κάτω από αυτό θα το ακολουθήσουν. Για να αντιγράψουμε μια στήλη με τουβλάκια από μία μορφή σε μία άλλη, σύρουμε τη στήλη στη μικρογραφία της άλλης μορφής στη λίστα με τις μορφές.

Κάποια τουβλάκια έχουν άσπρα κουτάκια στα οποία μπορούν να διαμορφωθούν (παράμετροι-ορίσματα) όπως είναι ένας αριθμός, η επιλογή από μία λίστα αριθμών κ.α. Για να αλλάξουμε την τιμή, κάνουμε κλικ μέσα στην άσπρη περιοχή που έχει το τουβλάκι, έτσι ώστε να διαμορφώσουμε το νούμερο, ή κάνουμε κλικ στο βελάκι της λίστας τιμών για να δούμε τις επιλογές της λίστας και να επιλέξουμε αυτή που θέλουμε. Μπορούμε επίσης να τοποθετήσουμε στρογγυλά κουτάκια (μεταβλητές) μέσα στην περιοχή που έχει το τουβλάκι.



Εικόνα 27: Ενέργειες σε τουβλάκια της «περιοχής ενεργειών», του Scratch.

Για να καθαρίσουμε το χώρο ενεργειών, κάνουμε δεξί κλικ και επιλέγουμε clean up από το μενού. Για να εξάγουμε ένα στιγμιότυπο από το χώρο ενεργειών, κάνουμε κλικ και επιλέγουμε save picture of scripts.

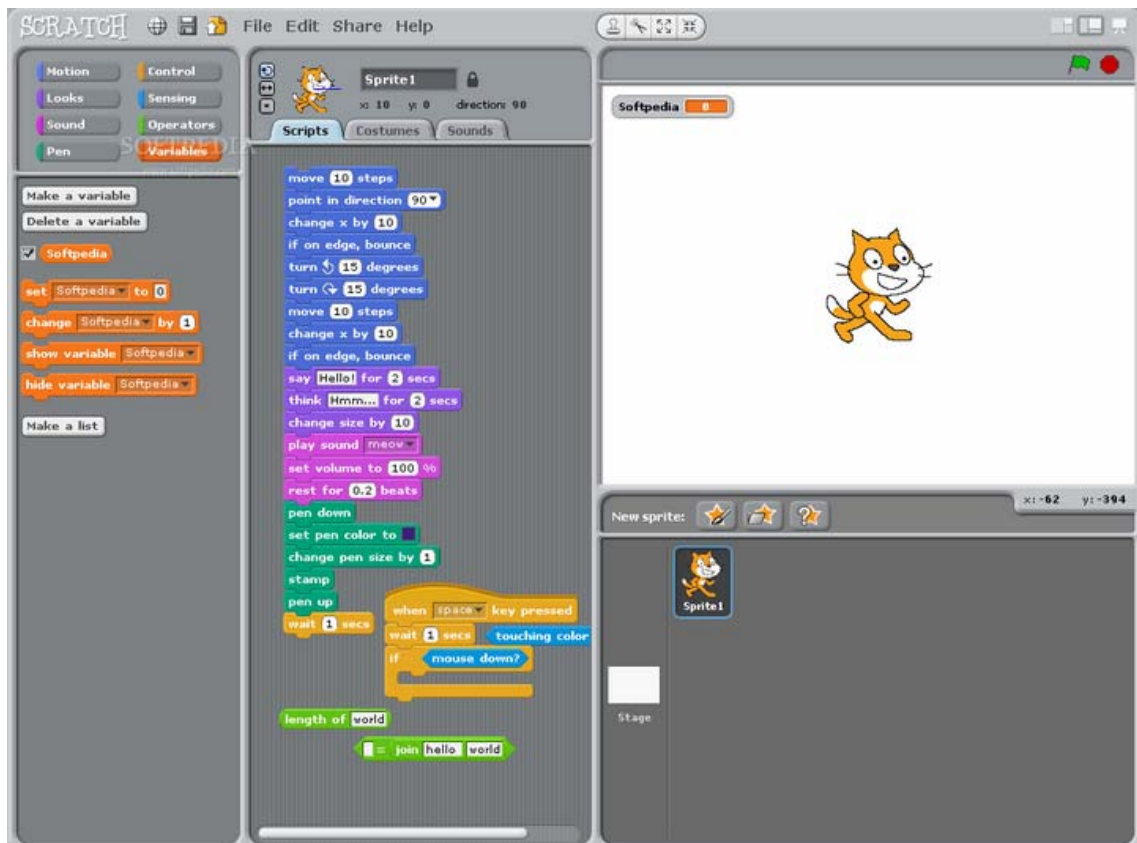


Εικόνα 28: Η «παλέτα εντολών» και η «περιοχή ενεργειών», του Scratch.

Υπάρχουν και παραλλαγές προγραμματιστικών περιβαλλόντων που έχουν βασιστεί στο Scratch όπως το BYOB (Build Your Own Block) όπου υπάρχει δυνατότητα επίδειξης εννοιών του αντικειμενοστραφούς προγραμματισμού, δημιουργίας νέων πλακιδίων εντολών και πιο εξελιγμένων μορφών δομών δεδομένων. Μία ακόμα πιο εξελιγμένη παραλλαγή του Scratch είναι το Panther το οποίο όμως δεν σχεδιάστηκε με σκοπό την χρήση του από αρχάριους προγραμματιστές.

Προγραμματιστικές δυνατότητες του περιβάλλοντος οπτικού προγραμματισμού με πλακίδια, **Scratch**:

- Διαχείριση (κλασική) πολυμεσικών στοιχείων.
- Οι διαδικασίες της έχουν ισχυρούς μηχανισμούς περάσματος παραμέτρων.
- Ποικιλία εντολών επανάληψης.
- Χρησιμοποιώντας τις διαδικασίες μπορεί ο προγραμματιστής να κατασκευάσει τις δικές του εντολές.



Εικόνα 29: Περιβάλλον οπτικού προγραμματισμού μικρόκοσμων με πλακίδια, Scratch.

Άλλα περιβάλλοντα προγραμματιστικών μικρόκοσμων με οπτικό προγραμματισμό με πλακίδια

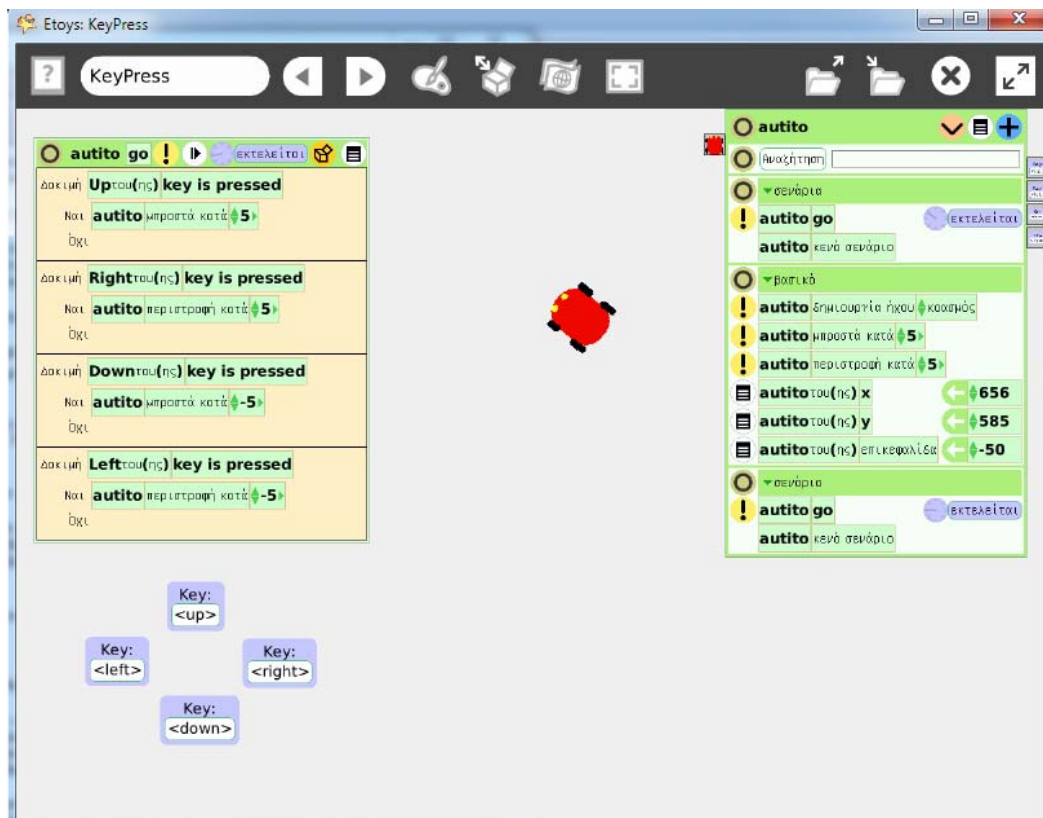
Το περιβάλλον προγραμματισμού με το όνομα **Etoys**, πρωτύτερα γνωστό ως Squeak, εμφανίστηκε το 1996 όταν την ανάπτυξη του ανέλαβε μια ομάδα ανάπτυξης καθοδηγούμενη από τον Alan Kay στην Disney Imagineering Research. Η σχεδίαση και ανάπτυξή του έγινε με σκοπό της υποστήριξης της κονστρουκτιβιστικής μάθησης, εμπνευσμένες από τον Seymour Papert και την γλώσσα Logo που δημιούργησε. Το περιβάλλον Etoys επηρέασε και αυτό με την σειρά του, σε μεγάλο βαθμό, ένα άλλο βασιζόμενο σε Squeak περιβάλλον προγραμματισμού γνωστό σαν Scratch.

Το περιβάλλον **Etoys** έχει βασιστεί στην έννοια εκτέλεσης ενεργειών στην οθόνη του υπολογιστή από προγραμματιζόμενες εικονικές οντότητες. Το περιβάλλον παρέχει μια μεγάλη γκάμα έτοιμων αντικειμένων – οντοτήτων (ήδη προγραμματισμένων - λειτουργικών) αλλά και δίνει πρωτίστως την δυνατότητα δημιουργίας αντικειμένων από τον χρήστη. Με την δημιουργία – σχεδίαση ενός νέου αντικειμένου στο περιβάλλον εργασίας είναι στην διάθεση του Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

προγραμματιστή μια σειρά από ιδιότητες του αντικείμενου (θέση στον x και y άξονα, προσανατολισμός κ.ά.) όπως και μια σειρά ενεργειών που μπορεί να κάνει το αντικείμενο (μετακίνηση, στροφή, κ.ά.). Με την χρήση όλων αυτών και την δυνατότητα δημιουργίας σεναρίων (scripts) που μας δίνει το περιβάλλον μπορεί ο χρήστης του περιβάλλοντος να δημιουργήσει πολλών ειδών έργα όπως απλά παιχνίδια, παρουσιάσεις, προσομοιώσεις. Ο προγραμματισμός γίνεται με οπτικό τρόπο, με το σύρσιμο και την εναπόθεση πλακιδίων όπως και σε νεότερα εκπαιδευτικά περιβάλλοντα οπτικού προγραμματισμού, και μπορεί να ειπωθεί ότι το περιβάλλον Etoys αποτέλεσε ένα μεταβατικό στάδιο από τα Logo περιβάλλοντα προγραμματισμού βασιζόμενα σε κείμενο στα μετέπειτα εκπαιδευτικά περιβάλλοντα οπτικού προγραμματισμού.

Τα βασικά συστατικά του περιβάλλοντος εργασίας Etoys είναι:

- Το μενού εικονιδίων του περιβάλλοντος στο πάνω μέρος (ζωγραφική, προμήθειες, επιλογή γλώσσας, άνοιγμα και αποθήκευση κ.ά.).
- Την περιοχή σχεδίασης, εμφάνισης και διαχείρισης παλετών πλακιδίων ενεργειών και ιδιοτήτων επιλεγμένων αντικειμένων, προγραμματισμού και εκτέλεσης - παρατήρησης του δημιουργημένου έργου.



Εικόνα 30: Περιβάλλον οπτικού προγραμματισμού μικρόκοσμων με πλακίδια, Etoys.

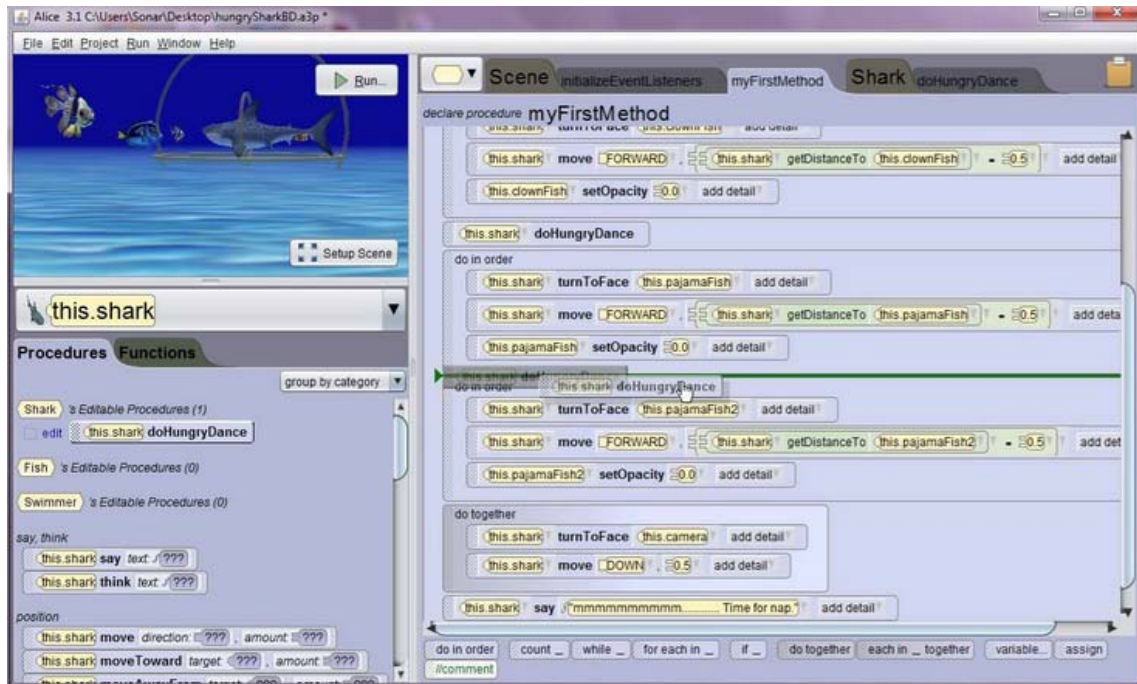
Το **Alice** (Carnegie Mellon University, 1999) είναι ένα εκπαιδευτικό περιβάλλον οπτικού προγραμματισμού που αναπτύχθηκε στο Carnegie Mellon University και πρωτοεμφανίστηκε το 1999. Χωρίς να είναι απαραίτητες πρότερες γνώσεις προγραμματισμού, ο χρήστης μπορεί να δημιουργήσει κινούμενα γραφικά για την αφήγηση μιας ιστορίας, διαδραστικά παιχνίδια ή και βίντεο. Το Alice αποτελεί ειδικότερα μια εκπαιδευτική αντικειμενοστραφής γλώσσα προγραμματισμού που δημιουργήθηκε για να αντιμετωπίσει τα προβλήματα που εμφανίζονταν στην διδασκαλία του προγραμματισμού λόγω της συντακτικής και σημασιολογικής

πολυπλοκότητας των εμπορικών γλωσσών προγραμματισμού που αποτελούσαν ανασταλτικό παράγοντα για τους αρχάριους προγραμματιστές.

Στο περιβάλλον **Alice** ο χρήστης μπορεί να δημιουργήσει ένα τρισδιάστατο εικονικό κόσμο της επιλογής του με την τοποθέτηση αντικείμενων μέσα σε αυτόν και στην συνέχεια να τον προγραμματίσει, απλά με σύρσιμο και εναπόθεση πλακιδίων εντολών. Υπάρχει μια μεγάλη ποικιλία αντικειμένων που παρέχονται από το περιβάλλον και μπορούν να προστεθούν στον εικονικό κόσμο που δημιουργείται. Το κάθε αντικείμενο έχει μια σειρά μεθόδων, ιδιοτήτων και συναρτήσεων, σε μορφή πλακιδίων, που ο χρήστης μπορεί να επιλέξει και να χρησιμοποιήσει μέσα στο πρόγραμμα του. Κάθε πλακίδιο αντιστοιχεί σε τυποποιημένες δηλώσεις μιας γλώσσας προγραμματισμού, όπως η Java, C + +, C #. Στο Alice ο χρήστης δεν χρειάζεται να απομνημονεύσει εντολές ή κάποιους συντακτικούς κανόνες και παρόλα αυτά είναι ικανός να προγραμματίζει σε μια γλώσσα που είναι πλήρως αντικειμενοστραφής και οδηγούμενη από γεγονότα. Επιπλέον το Alice επιτρέπει στους χρήστες - μαθητές να δουν αμέσως τα προγράμματα τους να τρέχουν, δίνοντας τους τη δυνατότητα έτσι να κατανοήσουν εύκολα τη σχέση μεταξύ των εντολών και της συμπεριφοράς των αντικειμένων στον εικονικό κόσμο. Με το χειρισμό των αντικειμένων στο εικονικό κόσμο τους, οι μαθητές αποκτούν εμπειρία με όλες τις δομές προγραμματισμού που συνήθως διδάσκονται σε ένα εισαγωγικό μάθημα προγραμματισμού.

Βασικά συστατικά του περιβάλλοντος Alice είναι:

- Το μενού επιλογών στο πάνω μέρος του περιβάλλοντος μαζί με τα κουμπιά εκτέλεσης, ανάρτησης/επανάληψης και τον κάδο ανακύκλωσης.
- Η περιοχή αντικειμένων του κόσμου, όπου φαίνεται μια λίστα των αντικειμένων που έχουν προστεθεί στον κόσμο και από όπου μπορεί να επιλεγεί το αντικείμενο το οποίο θα προγραμματιστεί.
- Η περιοχή με τις λεπτομέρειες του αντικειμένου που έχει επιλεγεί, όπου υπάρχουν τα πλακίδια που αντιστοιχούν σε ιδιότητες, μεθόδους και συναρτήσεις του αντικειμένου.
- Η περιοχή προγραμματισμού του επιλεγμένου αντικειμένου, όπου αποτελεί την περιοχή όπου συμβαίνει η σύνθεση των πλακιδίων εντολών ή ιδιοτήτων των αντικειμένων του κόσμου έτσι ώστε να λάβει χώρα ουσιαστικά ο προγραμματισμός του κόσμου.
- Η περιοχή προγραμματισμού των γεγονότων στα οποία θα αντιδράει ο κόσμος ή τα αντικείμενα του κόσμου που δημιουργείται.
- Η περιοχή προεπισκόπησης του κόσμου που έχει δημιουργηθεί, όπου μπορεί ο χρήστης να έχει μια εποπτική εικόνα του εικονικού κόσμου που δημιουργείται (της μορφής του κόσμου και αντικειμένων όπως και της θέσης αυτών) όπως και να τροποποιεί τον κόσμο αυτό (προσθήκη / μετακίνηση αντικειμένων κ.ά.). Δεν είναι δυνατή σε αυτή την περιοχή η προεπισκόπηση της εκτέλεσης του προγράμματος που δημιουργείται.
- Η σκηνή που εμφανίζεται όταν επιλεγεί να εκτελεστεί το πρόγραμμα (επιλογή κουμπιού Play), όπου πλέον φαίνεται η κίνηση των αντικειμένων του κόσμου, η αλληλεπίδρασή τους και οτιδήποτε έχει οριστεί στο πρόγραμμα.



Εικόνα 31: Περιβάλλον οπτικού προγραμματισμού μικρόκοσμων με πλακίδια, Alice.

Επιπλέον οι σχεδιαστές του περιβάλλοντος Alice προσπάθησαν να δημιουργήσουν ένα περιβάλλον προγραμματισμού που να απευθύνεται και σε κατηγορίες χρηστών που συνήθως έδειχναν μικρό ενδιαφέρον για τον προγραμματισμό. Αυτός είναι ο λόγος που δημιουργήθηκε και το Storytelling Alice το οποίο είναι προσανατολισμένο περισσότερο στην δημιουργία και στην αφήγηση ιστοριών (με χαρακτήρες με αυξημένα χαρακτηριστικά αλληλεπίδρασης και σκηνικά κόσμων φτιαγμένα για αυτό τον σκοπό).

3. Συστήματα τηλε-εκπαίδευσης

Ο χώρος της εκπαίδευσης χαρακτηρίζεται διαχρονικά από σημαντικές αλλαγές που έχουν άμεση σχέση με τις αλλαγές τόσο στο κοινωνικό περιβάλλον όσο και **με την είσοδο της τεχνολογίας στην εκπαίδευση**. Μια σημαντική αλλαγή στη μορφή της εκπαίδευσης αποτέλεσε η **εκπαίδευση από απόσταση**. Το κύριο χαρακτηριστικό αυτής της εκπαίδευσης είναι ο τοπικός και χρονικός διαχωρισμός του εκπαιδευόμενου από τον εκπαιδευτή (Ματραλής, 1999). Ο εκπαιδευτής δεν έρχεται πλέον σε άμεση επαφή με τον εκπαιδευόμενο κατά τη διάρκεια του μαθήματος αλλά επικοινωνούν μέσα από ένα σύστημα επικοινωνίας. Αυτό το σύστημα επικοινωνίας ξεκίνησε αρχικά με την αλληλογραφία μέσω ταχυδρομείου, χρησιμοποίησε το ραδιόφωνο, την τηλεόραση, το τηλέφωνο και τα τελευταία χρόνια υιοθέτησε όλες σχεδόν τις μορφές της σύγχρονης τεχνολογίας από κινητά τηλέφωνα και ηλεκτρονικούς υπολογιστές, μέχρι δορυφόρους (Keegan, 2000).

Αιτίες για την ανάπτυξη της εκπαίδευσης από απόσταση είναι η αλλαγή των δημογραφικών χαρακτηριστικών των εκπαιδευομένων, η αλματώδης αύξηση της γνώσης και της παράλληλης ανάγκης συνεχούς ενημέρωσης και κατάρτισης. Επιπλέον στην ανάπτυξη αυτή βοήθησαν η χρήση των υπολογιστών, η **τεράστια ανάπτυξη του Διαδικτύου** αλλά και η δημιουργία ειδικών προγραμμάτων (Naidu, 2003). Τα προγράμματα αυτά συνδυάζουν την ύπαρξη ισχυρών εργαλείων παρουσίασης του εκπαιδευτικού υλικού, με την υποστήριξη της αλληλεπίδρασης του εκπαιδευόμενου τόσο με το εκπαιδευτικό υλικό όσο και με τους απομακρυσμένους εκπαιδευτές.

Υπάρχουν πολλοί τύποι εκπαίδευσης από απόσταση όπως η τηλε-εκπαίδευση e-Learning (Virvou & Katsionis 2003), η εκμάθηση σε απευθείας σύνδεση Online learning, η εικονική εκμάθηση Virtual learning (Virvou & Katsionis 2008), η διανεμημένη (κατανεμημένη) εκμάθηση Distributed learning, και η εκμάθηση μέσω δικτύου Web-based learning (Virvou & Alepis 2003, Nam, Smith-Jackson, 2007), οι οποίοι έχουν πάρα πολλά κοινά χαρακτηριστικά.

Σήμερα, η εκπαίδευση από απόσταση ή αλλιώς τηλε-εκπαίδευση με τη χρήση των ΤΠΕ (Τεχνολογίες της Πληροφορίας και Επικοινωνίας) διεξάγεται πλέον με συστήματα που συνδυάζουν τις τεχνολογίες του διαδικτύου, με σκοπό τη δημιουργία ενός μαθησιακού περιβάλλοντος μέσα στο οποίο πραγματοποιείται η εκπαιδευτική διαδικασία. Κάθε μία προσέγγιση έχει συγκεκριμένες τεχνικές απαιτήσεις τόσο σε υλικό (Server, τερματικά κλπ) όσο και σε λογισμικό εξοπλισμό. **Λογισμικό** που αναφέρεται κυρίως σε ηλεκτρονικές πλατφόρμες εμπορικές (Blackboard κτλ) ή ανοικτού κώδικα (Moodle, e-class κτλ).

Τηλεκπαίδευση χαρακτηρίζεται κάθε είδους εκπαιδευτική δραστηριότητα η οποία με χρήση νέων τεχνολογιών (internet, χρήση Η/Υ) επιτρέπει στον εκπαιδευτή, να βρίσκεται σε διαφορετικό φυσικό χώρο από τους εκπαιδευόμενους, να οργανώνει και να ολοκληρώνει εκπαιδευτικές και μαθησιακές λειτουργίες ανεξάρτητες από τον συγκεκριμένο τόπο, χρόνο και ρυθμό μάθησης των εκπαιδευομένων. Αυτά που απασχολούν πια τον εκπαιδευτή είναι η μορφή που πρέπει να έχει το υλικό των μαθημάτων του, και η ταχύτητα μετάδοσης της πληροφορίας, ώστε να εξυπηρετηθεί η εικονική τάξη του.

Η πληροφορία πρέπει να είναι στη μορφή που θέλουμε, τη χρονική στιγμή που τη θέλουμε. Η σημερινή τεχνολογία παρέχει:

- Την πολυμορφία της πληροφορίας: Κείμενο, ήχος, εικόνα, γράφημα, κινούμενη εικόνα, ταινία (Πολυμέσα).
- Το Διαδίκτυο: εξασφαλίζει τη διακίνηση και τη διάχυση της πληροφορίας.
- Οι επικοινωνίες: προσφέρουν την απαραίτητη ταχύτητα για τη μεταφορά της πληροφορίας.
- Πλατφόρμες **σύγχρονης ή ασύγχρονης** επικοινωνίας – τηλεεκπαίδευσης.
- **Συστήματα Διαχείρισης Μαθημάτων** (CMS - LMS)

Στην **ασύγχρονη** τηλεεκπαίδευση εκπαιδευόμενοι και εκπαιδευτής διαχωρίζονται χωρικά και χρονικά. Το εκπαιδευτικό υλικό τοποθετείται στο διαδίκτυο και οι εκπαιδευόμενοι έχουν τη δυνατότητα να εργαστούν με αυτό από οπουδήποτε και οποτεδήποτε. Η **σύγχρονη** τηλεεκπαίδευση υλοποιείται μέσω του διαδικτύου σε πραγματικό χρόνο σε περιβάλλον εικονικής τάξης. Εκπαιδευόμενοι και εκπαιδευτής διαχωρίζονται χωρικά αλλά βρίσκονται στον ίδιο χρόνο στο διαδίκτυο.

Κατηγορίες Συστημάτων Διαχείρισης Μαθημάτων (ΣΔΜ)

Στα **Συστήματα Διαχείρισης Μαθημάτων (ΣΔΜ)** (όπως είναι τα Blackboard, Joomla, Moodle, e-class, PostNuke, Drupal κ.ά.), υπάρχει μια σύγχυση σε σχέση με τις πραγματικές λειτουργίες τους. Τα Συστήματα Διαχείρισης Μαθημάτων (ΣΔΜ) διακρίνονται κυρίως σε δυο κατηγορίες, τα *Συστήματα Διαχείρισης Περιεχομένου* (Content Management System - CMS) και τα *Συστήματα Διαχείρισης Μάθησης* (Learning Management Systems - LMS). Πηγή αυτής της σύγχυσης είναι οι ομοιότητες των δυο συστημάτων. Και τα δυο επιτελούν λειτουργίες εγγραφής σπουδαστών, επικοινωνίας με αυτούς, αποτίμησης της απόδοσης και ενεργοποίησης μαθησιακού υλικού, αλλά έχουν και ορισμένες διαφορές μεταξύ τους.

Τα **Συστήματα Διαχείρισης Περιεχομένου** (Content Management System-CMS) είναι εφαρμογές με τις οποίες μπορεί κανείς να διαχειριστεί συγκεντρωμένες πληροφορίες ηλεκτρονικής μορφής στο διαδίκτυο. Ένα CMS μπορεί να περιέχει μορφές μαθησιακού υλικού όπως: Μαθήματα, Πολυμέσα, Ιστοσελίδες, Δοκιμασίες ελέγχου της μάθησης κ.λπ. Όταν ένα CMS χρησιμοποιείται για τις ανάγκες της εκπαίδευσης, μπορεί να χαρακτηριστεί ως Σύστημα Διαχείρισης Περιεχομένου Μάθησης (Content Course Management System). Τέτοια συστήματα επιτρέπουν στο διδάσκοντα να δημιουργήσει ένα διαδικτυακό μάθημα, όπου θα μπορεί να εισαχθεί (upload) εκπαιδευτικό υλικό σε έναν από τους συνήθεις τύπους (όπως κειμένου, παρουσίασης, ήχου, βίντεο κλπ)) χωρίς να χρειάζεται κάποια απαραίτητη μετατροπή σε web format, δηλαδή να μετατραπεί σε υλικό κατάλληλο για ιστοσελίδες (ενσωμάτωση στον κώδικα της ιστοσελίδας). Τέτοια συστήματα διαχείρισης περιεχομένου είναι ενδεικτικά το **joomla και το e-class**.

Γενικά ένα CMS, απαιτεί σχετικά περιορισμένες δεξιότητες και αυτό το καθιστά δημοφιλή επιλογή. Καλύπτει συνήθως τα εξής χαρακτηριστικά:

- Online ανάρτηση υλικού μαθημάτων.
- Αξιολόγηση σπουδαστή. Η αξιολόγηση αυτή μπορεί να υλοποιηθεί με online ερωτήσεις, τεστ κλπ.
- Φόρουμ συζητήσεων (e-forum). Οι συζητήσεις μπορούν να διεξάγονται με την επίβλεψη μιας ομάδας προκειμένου να ανταλλάσσονται σημειώσεις και να συζητώνται συγκεκριμένα θέματα στο ενδιαμέσο των μαθημάτων.

Ως μειονεκτήματα θα μπορούσαν να θεωρηθούν:

- Μειωμένη ευελιξία. Τα ονόματα των συγκεκριμένων τμημάτων που αποτελούν ένα CMS σπάνια μπορούν να αλλάξουν ή να μεταβληθούν.
- Ανεπαρκής παροχή εργαλείων συγγραφής. Το διαδραστικό e-learning μέσα από εργαλεία συγγραφής όπως το Dreamweaver, ή το Flash, δεν μπορεί να διανεμηθεί μέσω των CMS. Για το σκοπό αυτό οι διδάσκοντες απαραίτητα θέτουν συνδέσμους προς το απομονωμένο υλικό που έχουν δημιουργήσει και που βρίσκεται αποθηκευμένο αλλού.
- Αδυναμία στον έλεγχο και την καταγραφή. Δεν μπορεί να επαληθεύσει την ταυτότητα των σπουδαστών που λαμβάνουν μέρος στην εξέταση ούτε μπορεί να σώσει το τεστ πριν το υποβάλλει ο μαθητής στον διδάσκοντα.
- Έλλειψη εμπορικής υποστήριξης.

Μια ευρύτερη κατηγορία συστημάτων διαχείρισης μαθημάτων είναι τα **Συστήματα Διαχείρισης Μάθησης LMS** (Learning Management Systems-LMS). Τα LMS είναι συστήματα τα οποία, διανέμουν και διαχειρίζονται όλες τις μαθησιακές ανάγκες. Είναι αναγκαίο στο σημείο αυτό να διευκρινιστεί ότι η διαχείριση αναφέρεται περισσότερο στην πληροφορία που συντελεί στη μάθηση και όχι σε αυτήν καθαυτή τη μάθηση. Μέσω ενός ολοκληρωμένου, βασισμένου στο Web περιβάλλοντος διεπαφής, το LMS επιτρέπει στους διαχειριστές να εκτελούν διαχειριστικές εργασίες, όπως να καθιστούν διαθέσιμα τα μαθήματα, να εγγράφουν νέους εκπαιδευόμενους και να προχωρούν στην επιβεβαίωση αυτών των εγγραφών, να ελέγχουν την καταλληλότητα των σπουδαστών, να δημιουργούν υπενθυμίσεις για το πρόγραμμα μαθημάτων, να δημιουργούν τεστ, να καταγράφουν την ολοκλήρωση των μαθημάτων και ενοτήτων και τις επιδόσεις των εκπαιδευομένων, να δημιουργούν σχετικές αναφορές για την ολοκλήρωση του μαθήματος στο διδάσκοντα, και ακολούθως να ενημερώνεται και ο σπουδαστής. Τέλος το LMS συγκεντρώνει παράγει αναφορές για τον αριθμό των σπουδαστών που έχουν εγγραφεί σε ένα συγκεκριμένο μάθημα, και συγκεντρώνει τη βαθμολογία απόδοσης των σπουδαστών σε συγκεκριμένα μαθήματα.

Τυπικά ένα LMS είναι ένα λογισμικό που βασίζεται στο Internet το οποίο διαχειρίζεται, παρακολουθεί και παρέχει πληροφορίες σχετικά με την αλληλεπίδραση μεταξύ του εκπαιδευόμενου και του περιεχομένου, καθώς και μεταξύ του εκπαιδευόμενου και του εκπαιδευτή. Τέτοια συστήματα διαχείρισης μάθησης (LMS) είναι το **Blackboard, το ILIAS και το Moodle**.

Οι βασικές λειτουργίες ενός τέτοιου συστήματος λοιπόν περιλαμβάνουν:

- Δημιουργία μαθημάτων.
- Εγγραφή και έλεγχος σπουδαστών.
- Προγραμματισμό μαθημάτων.
- Παρακολούθηση συμμετοχής στο μάθημα.
- Εξέταση - Online Τεστ.
- Διεξαγωγή συζητήσεων.
- Μεταφορά πληροφορίας σε άλλα συστήματα.

Ως μειονεκτήματα θα μπορούσαν να θεωρηθούν:

- Ραγδαία εξέλιξη της τεχνολογίας - αυξημένο κόστος. Σύντομα ένα τέτοιο σύστημα καθίσταται πλεονάζον ή περιττό εξαιτίας της επερχόμενης τεχνολογίας που περιλαμβάνεται στις νεότερες εκδόσεις του.
- Προβλήματα προσαρμογής. Οι διαφορετικές ανάγκες των διάφορων οργανισμών που θα υιοθετήσουν ένα τέτοιο σύστημα δημιουργεί προβλήματα προσαρμογής στις ανάγκες αυτές. Το σύστημα επιδέχεται περιορισμένες αλλαγές. Εκτεταμένες αλλαγές μπορεί να κάνουν περισσότερη ζημιά παρά να ωφελήσουν.
- Απαραίτητη η σχετική εκπαίδευση των χρηστών στη χρήση του συστήματος.

Στον παρακάτω πίνακα παρουσιάζονται συνοπτικά σε αντιπαράθεση μερικά από τα πλεονεκτήματα και τα μειονεκτήματα των παραπάνω συστημάτων διαχείρισης μαθημάτων.

	CMS	LMS
Πλεονεκτήματα	<ul style="list-style-type: none"> • Εύκολα στην χρήση • Απαιτούν περιορισμένες ικανότητες • Ελάχιστο κόστος • Online ανάρτηση υλικού • Αξιολόγηση σπουδαστών με χρήση ερωτήσεων, τεστ • Ομάδες συζητήσεων • Ανταλλαγή υλικού μεταξύ χρηστών 	<ul style="list-style-type: none"> • Αυτοματοποιημένες δυνατότητες διαχείρισης • Δυνατότητα εγγραφών ή επεξεργασίας μαθητών σε ομάδες • Υπενθυμίσεις για το πρόγραμμα μαθημάτων • Έλεγχος καταλληλότητας σπουδαστών και διαχείριση πόρων
Μειονεκτήματα	<ul style="list-style-type: none"> • Έλλειψη εμπορικής υποστήριξης • Ορισμένα δεν διαθέτουν εργαλεία συγγραφής 	<ul style="list-style-type: none"> • Πολύπλοκα • Απαιτούν αυξημένες ικανότητες • Αυξημένο κόστος

Πίνακας 1: Σύγκριση συστημάτων διαχείρισης μαθημάτων (CMS συστήματα διαχείρισης περιεχομένου – LMS συστήματα διαχείρισης μάθησης).

Παρόλο λοιπόν που και τα δυο συστήματα ενσωματώνουν δυνατότητες για ηλεκτρονική μάθηση, εν τούτοις το ένα σύστημα δεν μπορεί να αντικαταστήσει το άλλο, λόγω των ανόμοιων μαθησιακών δραστηριοτήτων που υποστηρίζουν. Είναι απαραίτητο να γίνει κατανοητό ότι η εκπαίδευση κερδίζει σε μακροπρόθεσμη γνώση ενώ η επιμόρφωση κερδίζει γνώση για άμεση εφαρμογή. Συνεπώς, το CMS υποστηρίζει καλύτερα μακροπρόθεσμες συνεδρίες μαθημάτων ενώ το LMS υποστηρίζει έναν αριθμό από σύντομα επιμορφωτικά γεγονότα - φυσικά, θα πρέπει να επισημανθεί ότι αυτές οι διαφορές δεν έχουν απόλυτο χαρακτήρα, αλλά είναι πολύ σχετικές.

Πρόσφατα στα συστήματα διαχείρισης μαθησιακού υλικού έχει προστεθεί και ο όρος LCMS (LMS+CMS) δηλαδή τα **συστήματα διαχείρισης περιεχομένου μάθησης** (Learning Content Management Systems - LCMS). Πρόκειται για ένα περιβάλλον όπου οι διαχειριστές μπορούν να δημιουργούν, να αποθηκεύουν, να επαναχρησιμοποιούν, να διαχειρίζονται και να διανείμουν μαθησιακό περιεχόμενο από μια κεντρική δεξαμενή, συνήθως μια βάση δεδομένων. Σε ένα τέτοιο σύστημα, ο διαχειριστής εισάγει και αποθηκεύει πόρους που θα χρησιμοποιηθούν για τη δημιουργία ενός αντικειμένου μάθησης και ταυτόχρονα δημιουργεί και αποθηκεύει το περιεχόμενο της μάθησης. Έτσι το LMS έχει πρόσβαση στο CMS και ανακτά το περιεχόμενο της μάθησης. Το LCMS συνδυάζει την ισχύ του CMS με την αρτιότητα του LMS και πια αντιμετωπίζεται από τους περισσότερους ως ένα πλήρες σύστημα eLearning με δημιουργία και διαχείριση περιεχομένου.

Υπάρχουν και άλλα συστήματα διαχείρισης μαθησιακού υλικού όπως τα εικονικά συστήματα μάθησης (Virtual Learning Environment - VLE), τα ολοκληρωμένα μαθησιακά περιβάλλοντα (Enterprise Learning Management Systems - ELMS) κ.α.

Πρότυπα περιγραφής μαθησιακού υλικού - το πρότυπο SCORM

Τα *Συστήματα Διαχείρισης Μάθησης (LMS)* στηρίζονται σε μια ποικιλία από πλατφόρμες ανάπτυξης, από αρχιτεκτονικές βασισμένες σε Java EE μέχρι Microsoft.NET, και συνήθως χρησιμοποιούν στο βάθος μια στιβαρή βάση δεδομένων. Πολύ γρήγορα φάνηκε η ανάγκη ύπαρξης **προτύπων** για την περιγραφή του μαθησιακού υλικού, ώστε τα συστήματα αυτά να προσφέρουν μεταφερσιμότητα (portability) των μαθησιακών πόρων, διαλειτουργισμότητα (interoperability) μεταξύ τους και εύκολη αναζήτηση. Έτσι, τα **κυριότερα πρότυπα** που έχουν μέχρι στιγμής αναπτυχθεί είναι τα παρακάτω:

- Το πρότυπο AICC (Aviation Industry CBT Committee).
- Το πρότυπο IMS Global Learning Consortium.
- Το πρότυπο Learning Object Metadata Standard της IEEE (IEEE LOM).
- Το πρότυπο **SCORM** (Sharable Content Object Reference Model), που αναπτύχθηκε από το Υπουργείο Εθνικής Άμυνας των ΗΠΑ, με σκοπό να συνενώσει τα υπόλοιπα πρότυπα και σήμερα είναι ίσως το πιο δημοφιλές. Τα πακέτα SCORM μπορούν να φορτωθούν σε οποιοδήποτε συμβατό με αυτό Σύστημα Διαχείρισης Μάθησης.

Το Sharable Content Object Reference Model (**SCORM**) (Rustici Software, 2000), είναι ένα σύνολο προδιαγραφών για την ανάπτυξη, τη συσκευασία (packaging) και τη διανομή εκπαιδευτικού υλικού υψηλής ποιότητας, όποτε και οπουδήποτε αυτό απαιτείται. Το SCORM εξασφαλίζει:

- την επαναχρησιμοποίηση
- την προσβασιμότητα και
- την ανθεκτικότητα του εκπαιδευτικού υλικού στις αλλαγές της τεχνολογίας, καθώς και τη διαλειτουργικότητα μεταξύ συστημάτων ηλεκτρονικής μάθησης.

Το **SCORM v1.2** (Rustici Software, 2001) αποτελείται από δύο μέρη:

1. Το SCORM Content Aggregation Model.
2. Το SCORM Run-time Environment.

Το **SCORM Content Aggregation Model**:

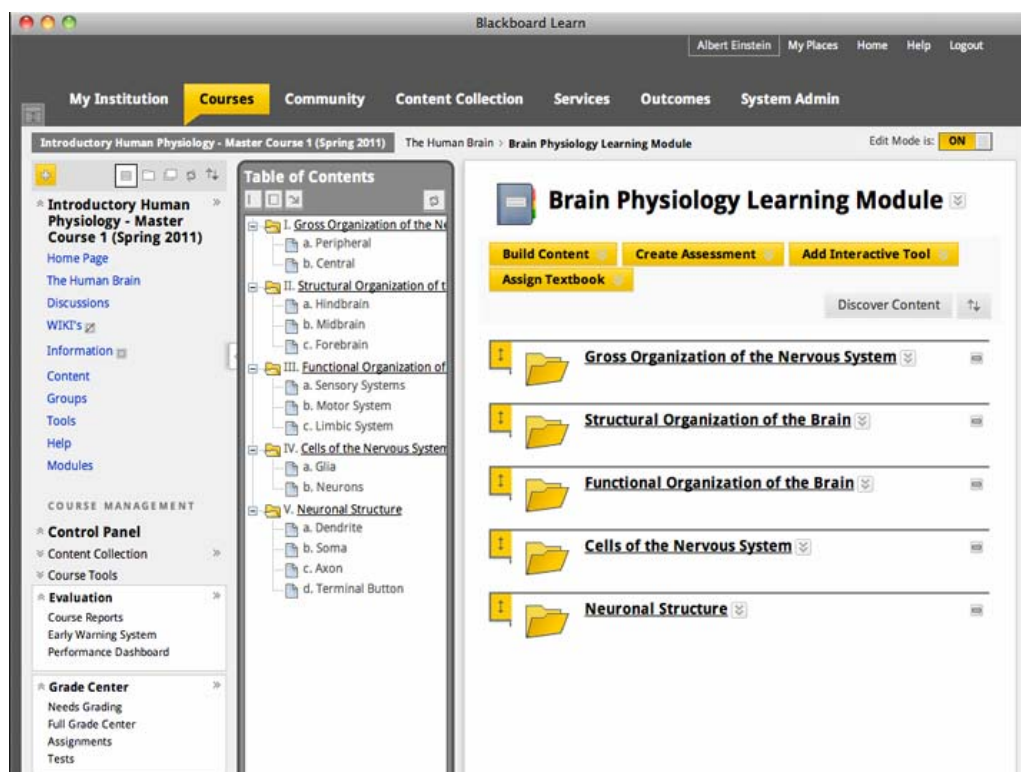
- Παρέχει έναν κοινό τρόπο για τη σύνθεση εκπαιδευτικού υλικού από αναζητήσιμες, επαναχρησιμοποιήσιμες, διαμοιραζόμενες και διαλειτουργικές πηγές.
- Ορίζει πώς το εκπαιδευτικό υλικό μπορεί να ταυτοποιηθεί (identified) και να περιγραφεί, να ομαδοποιηθεί ως ένα μάθημα ή ως ένα μέρος μαθήματος και να μετακινηθεί μεταξύ συστημάτων που μπορεί να περιέχουν Learning Management Systems (LMS) και αποθήκες εκπαιδευτικού υλικού.
- Ορίζει τις τεχνικές μεθόδους για την πραγματοποίηση των παραπάνω διαδικασιών.
- Περιέχει προδιαγραφές για την ομαδοποίηση του εκπαιδευτικού υλικού και την περιγραφή του με μεταδεδομένα.
- Συσκευάζει το υλικό σε ένα zip αρχείο (SCORM πακέτο ή Package Interchange File (PIF)). Το αρχείο αυτό περιέχει, όχι μόνο τα αρχεία του μαθήματος, αλλά και ένα XML αρχείο (manifest file σύμφωνα με το SCORM), το οποίο περιγράφει τα περιεχόμενα του μαθήματος, τη δομή και την αλληλουχία τους.

Το **SCORM Run-time Environment** επιτυγχάνει τη διαλειτουργικότητα μεταξύ SCORM εκπαιδευτικού υλικού και LMS ανεξάρτητα από τα εργαλεία που έχουν χρησιμοποιηθεί για τη δημιουργία του.

3.1 Διάφορα συστήματα διαχείρισης μαθημάτων (CMS και LMS) μέσω τηλε-εκπαίδευσης

Blackboard (LMS) (Blackboard Inc, 1997): Πιθανά το πιο διαδεδομένο και το πιο πετυχημένο εμπορικά ΣΔΜ. Η εταιρεία που το αναπτύσσει ιδρύθηκε το 1997 και απασχολεί περισσότερους από χίλιους εργαζόμενους, υποστηρίζοντας με το λογισμικό της σε περισσότερα από 5.000 εκπαιδευτικά ιδρύματα και οργανισμούς.

Η πλατφόρμα Blackboard είναι ένα ολοκληρωμένο Σύστημα Διαχείρισης Μάθησης (Learning Management System), που προορίζεται για εκπαιδευτικά ιδρύματα και προσφέρει σε γενικές γραμμές τριπλή λειτουργία μέσω διαδικτύου: τη διδασκαλία του διδακτικού υλικού, την επικοινωνία μεταξύ εκπαιδευτή και εκπαιδευομένων και την αξιολόγηση των εκπαιδευομένων. Αποτελεί μια ευέλικτη πλατφόρμα την οποία οι εκπαιδευτές μπορούν να προσαρμόσουν ανάλογα με τη θεωρία μάθησης ή το μοντέλο διδασκαλίας που θέλουν να χρησιμοποιήσουν, ενώ παρέχει δυνατότητα επέκτασης ανάλογα με τις ανάγκες του ιδρύματος/οργανισμού που προσφέρει τα εκπαιδευτικά προγράμματα, πολυγλωσσική υποστήριξη και αρχιτεκτονική που επιτρέπει την ενσωμάτωση άλλων εφαρμογών.



Εικόνα 32: Περιβάλλον συστήματος διαχείρισης μάθησης, Blackboard.

Δυνατότητες:

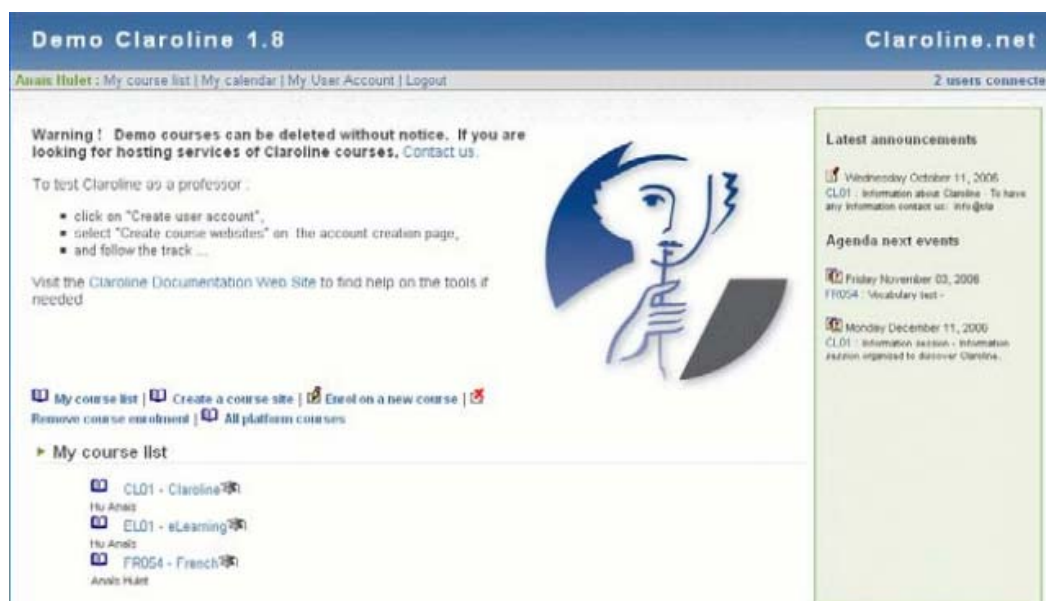
- Εργαλεία συγγραφής περιεχομένου. Να μπορεί ο διαχειριστής να δημιουργεί ή να εισάγει διδακτικό υλικό κατά μονάδες ή πακέτα, όπως πληροφορίες, ανακοινώσεις, σημειώσεις, διαλέξεις, ασκήσεις, βιβλιογραφία κ.ά.
- Προκατασκευασμένο υλικό μαθημάτων.
- Προσαρμοστικότητα. Ο διδάσκων καθορίζει την πορεία μάθησης ορίζοντας ή απορρίπτοντας την πρόσβαση κάποιου σπουδαστή σε συγκεκριμένο υλικό.
- Δημιουργία πολλαπλών φόρουμ συζήτησης.

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

- Online αξιολόγηση και επισκόπηση της προόδου ενός μαθητή.
- Σύστημα έγκαιρης προειδοποίησης (κατατάσσει τους μαθητές σταδιακά σε profiles με βάση τη συμμετοχή και τις επιδόσεις τους και ειδοποιεί τον διδάσκοντα προκειμένου αυτός να επιλέξει την πορεία μάθησης).
- Δημιουργία μαθησιακές κοινότητες με online συζητήσεις.
- Εύκολη ανάρτηση πληροφορίας χωρίς γνώση HTML.
- Ηλεκτρονικό Εμπόριο. Τα ιδρύματα μπορούν να χρεώνουν online.
- Επιτρέπει στους χρήστες να δημιουργούν blogs.
- Δημιουργία portfolio και δυνατότητα προσωπικών ρυθμίσεων στην εμφάνιση του Portfolio.
- Templates για κάθε είδους υλικό.
- Δημιουργία αναφορών.
- Κατάλογος μαθησιακών αντικειμένων (για αναζήτηση).

Claroline (LMS) (UCL 2000): Η ανάπτυξη της πλατφόρμας ξεκίνησε από το UCL (Καθολικό Πανεπιστήμιο της Louvain) στο Βέλγιο το 2000 και σύμφωνα με τους κατασκευαστές της δημιουργήθηκε για να υποστηρίξει παραδοσιακή διδασκαλία με διαλέξεις, αυτόνομη μάθηση, μεικτή διδασκαλία ή εξ αποστάσεως εκπαίδευση. Βασίζεται στη γλώσσα προγραμματισμού PHP και χρησιμοποιεί ως σύστημα διαχείρισης βάσεων δεδομένων τη MySQL. Προσφέρεται με άδεια χρήσης GPL. Μεταφρασμένη σε 35 γλώσσες και με χρήση σε 95 χώρες, έχει δημιουργήσει μια μεγάλη κοινότητα χρηστών και προγραμματιστών που ασχολούνται με την εξέλιξη ή προσαρμογή της πλατφόρμας.

Σύμφωνα με την ομάδα τηλεκπαίδευσης TELEDU του GUnet, κρίνεται ως φιλικό και εύχρηστο, ενώ διαθέτει εγγενή υποδομή και υποστήριξη πολυγλωσσικών ιστοσελίδων. Οι δημιουργοί του δηλώνουν ότι υπολογίστηκε οι δυνατότητες του συστήματος να ξεκινούν από απλή διανομή εγγράφων και αρχείων μέχρι υποστήριξη της μάθησης με τη μέθοδο project ή επίλυση προβλημάτων και συνεργατική ηλεκτρονική μάθηση, ενώ με τις δυνατότητες εξέλιξής του να λειτουργεί ως καταλύτης για παιδαγωγικές καινοτομίες και εξέλιξη των εκπαιδευτικών οργανισμών.



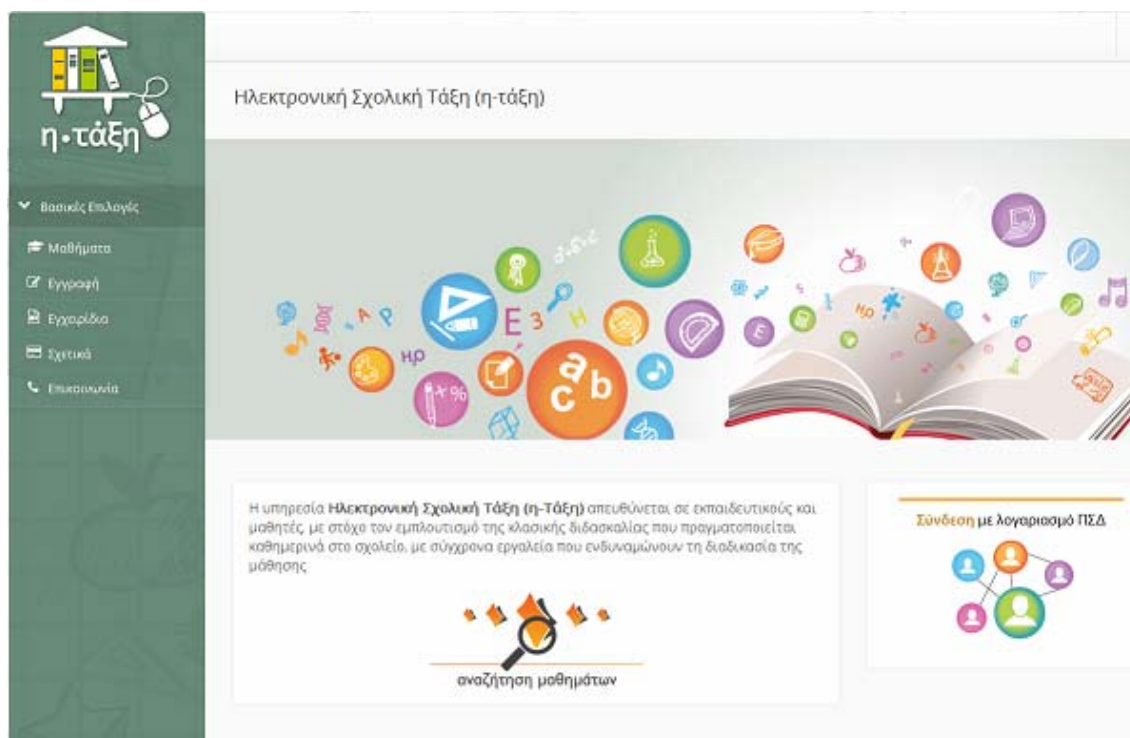
Εικόνα 33: Περιβάλλον συστήματος διαχείρισης μάθησης, Claroline.

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

E-class (CMS ή/και LMS) (Greek Academic network, 2000): Σχεδιάστηκε, αναπτύχθηκε και υποστηρίζεται από την ομάδα Ασύγχρονης Τηλεκπαίδευσης του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών (ΕΚΠΑ) για λογαριασμό του Ακαδημαϊκού Διαδικτύου GUnet. Στην αρχική του έκδοση πριν από μερικά χρόνια είχε βασιστεί στην πλατφόρμα Claroline 1.3, ωστόσο με τις επόμενες εκδόσεις του έχει διαφοροποιηθεί από αυτήν. Παράλληλα, η ίδια ομάδα έχει τροποποιήσει κατάλληλα την πλατφόρμα τόσο για τις ανάγκες του Πανεπιστημίου Αθηνών (<http://eclass.uoa.gr>) όσο και για τις ανάγκες της Δευτεροβάθμιας Εκπαίδευσης μέσω του Πανελληνίου Σχολικού Δικτύου (<http://eclass.sch.gr>). Στην έκδοση 2.1, από 15 Οκτωβρίου 2008, η πλατφόρμα μετονομάστηκε σε Open e Class.

Δυνατότητες:

- Έχει απλή διεπαφή.
- Αποδοχή πολλών μορφών περιεχομένου (word, pdf, powerpoint, flash, windows media, κλπ).
- Εύκολη οργάνωση του περιεχομένου (μετακινήσεις, αντιγραφές).
- Εμφάνιση ανάλογη με τους γνωστούς browsers για διευκόλυνση των χρηστών.
- FTP Publishing.
- Visual HTML Editing.
- Ενσωματωμένη μηχανή αναζήτησης.
- Γρήγορη και εύκολη παραγωγή αυτοβαθμολογούμενων διαγωνισμάτων.
- Πολύγλωσση υποστήριξη.



Εικόνα 34: Περιβάλλον συστήματος διαχείρισης περιεχομένου ή/και μάθησης, e-class.

ILIAS (LMS) (University of Cologne 1997): Η ανάπτυξη του ξεκίνησε το 1997/1998 από το Πανεπιστήμιο της Κολωνίας στη Γερμανία με γλώσσα προγραμματισμού PHP και χρησιμοποιεί ως σύστημα διαχείρισης βάσεων δεδομένων τη MySQL. Διατίθεται με άδεια χρήσης GPL. Παρόλο που προσφέρει τη δυνατότητα παραμετροποίησης της επιφάνειας εργασίας ανάλογα με τις επιθυμίες του χρήστη, η πλατφόρμα κρίνεται αρκετά δύσχρηστη για τον διδάσκοντα, σύμφωνα με την ομάδα τηλεκπαίδευσης TELEDU του GUnet, αφού απαιτεί την εγκατάσταση επιπλέον λογισμικού και διαμόρφωσης του περιβάλλοντος της PHP, ενώ και ο τρόπος εισαγωγής εκπαιδευτικού υλικού στην πλατφόρμα είναι αρκετά περιορισμένος.



Εικόνα 35: Περιβάλλον συστήματος διαχείρισης μάθησης, ILIAS.

Joomla (CMS) (Open Source Matters, Inc 2005): Το Joomla είναι ένα δωρεάν σύστημα διαχείρισης περιεχομένου το οποίο χρησιμοποιείται για την ανάπτυξη δυναμικών ιστοσελίδων και την διαχείριση του περιεχομένου τους μέσα από ένα γραφικό περιβάλλον. Χρησιμοποιεί διακομιστή Web τεχνολογίας Apache, είναι γραμμένο στην γλώσσα scripting PHP και αποθηκεύει τα δεδομένα του σε βάση δεδομένων MySQL.

Ο διαχειριστής προτύπων (template manager) που βρίσκεται στο διαχειριστικό περιβάλλον του Joomla, παρέχει το υπόβαθρο για την εγκατάσταση και επεξεργασία των προτύπων. Τα πρότυπα (templates) και τα αρχεία που συνδέονται με αυτά, καθορίζουν την εμφάνιση και το στυλ της ιστοσελίδας και διατηρούνται ξεχωριστά από το υπόλοιπο περιεχόμενό της. Τα πρότυπα μπορούν να χρησιμοποιηθούν σε διαφορετικά τμήματα της ιστοσελίδας. Συγκεκριμένα υπάρχει η δυνατότητα χρήσης διαφορετικών προτύπων σε διαφορετικούς συνδέσμους της ιστοσελίδας.

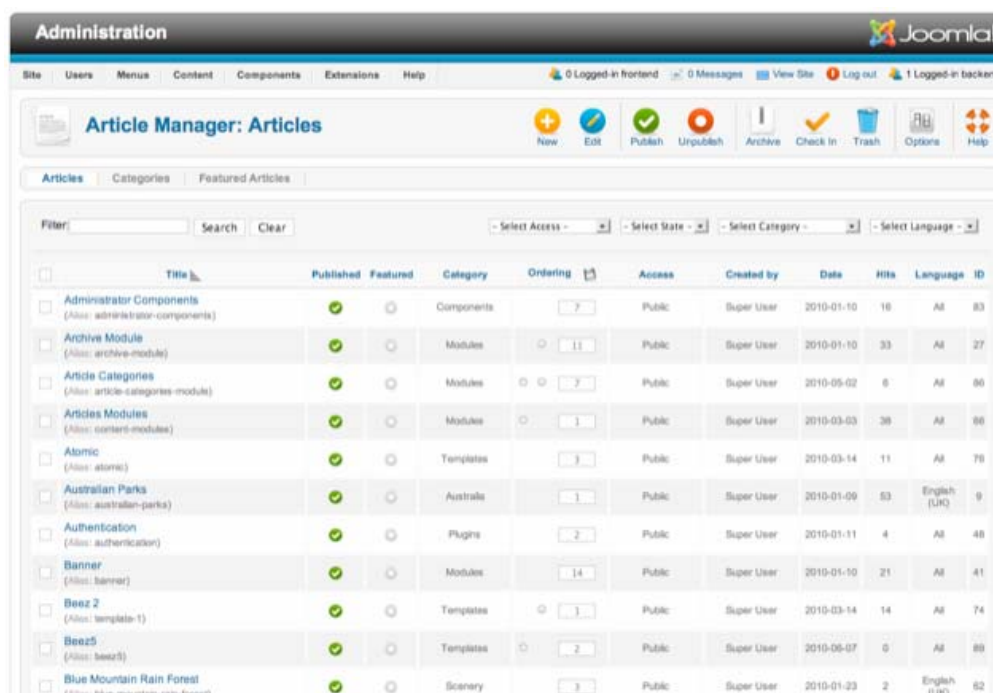
Στο Joomla, το περιεχόμενο (κείμενα, φωτογραφίες κτλ.) είναι εντελώς ανεξάρτητο από τον τρόπο με τον οποίο θέλουμε να εμφανίζεται στην ιστοσελίδα. Ένας ή περισσότεροι σύνδεσμοι, ενσωματώνονται συνήθως στην ιστοσελίδα με τη βοήθεια ενός μενού. Με τη χρήση των μενού, επιτυγχάνεται η πλοήγηση και η πρόσβαση στις διάφορες περιοχές της ιστοσελίδας.

Δυνατότητες:

- Εύκολο στην εγκατάσταση.
- Επεξεργασία του περιεχομένου μόνο με απλή γνώση word.
- Διαθέσιμη συλλογή πολυμεσικών αρχείων και αρχείων εικόνας.

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

- Δυνατότητα ηλεκτρονικού εμπορίου.
- Χώρος για φόρουμ και συζητήσεις.
- Δυνατότητα δημιουργίας blogs.
- Email Newsletters.
- Συλλογή δεδομένων και δημιουργία αναφορών.
- Υπηρεσίας συνδρομητικής εγγραφής.
- Δημιουργία καταλόγων περιεχομένου.



Εικόνα 36: Περιβάλλον συστήματος διαχείρισης περιεχομένου, Joomla.

3.2 Το σύστημα διαχείρισης μάθησης (LMS) μέσω τηλε-εκπαίδευσης Moodle

Το όνομα **Moodle** (Dougiamas, 2001) προέρχεται από το ακρώνυμο των λέξεων Modular Object Oriented Dynamic Learning Environment και είναι ένα πακέτο λογισμικού για την δημιουργία διαδικτυακών μαθημάτων. Δημιουργήθηκε το 1999 από τον Αυστραλό Martin Dougiamas ως τμήμα της διδακτορικής του διατριβής. Οι δυνατότητες του δεν περιορίζονται στην εκπαίδευση από απόσταση αλλά μπορεί να λειτουργήσει συμπληρωματικά και στην κλασική διδασκαλία. Μέσα από την πλατφόρμα Moodle ο εκπαιδευτικός μπορεί να παρουσιάσει το μάθημα του με τέτοιο τρόπο ώστε να προκαλεί ενδιαφέρον στον εκπαιδευόμενο χρησιμοποιώντας διάφορα μέσα. Για παράδειγμα, διάθεση υλικού του μαθήματος σε διαφορετικές μορφές, προσθήκη δραστηριοτήτων, σύγχρονη και ασύγχρονη επικοινωνία μεταξύ μελών, ανάθεση εργασιών, αξιολόγηση της επίδοσης (Βερναδάκης κ.α., 2007).

Το **Moodle** παρέχεται δωρεάν σαν λογισμικό Open Source (κάτω από την άδεια GNU-Public License). Αυτό πρακτικά σημαίνει ότι το Moodle έχει πνευματικά δικαιώματα αλλά μπορεί να χρησιμοποιηθεί, να γίνουν αντιγραφές του, και τροποποιήσεις του υπό τον όρο να παρέχεται ο πηγαίος κώδικας και σε άλλους, να μην αλλάξει ή αφαιρεθεί η επίσημη άδεια και να εφοδιάζετε με αυτήν οποιαδήποτε παράγωγη δουλειά. Το Moodle μπορεί να τρέξει σε οποιοδήποτε σύστημα υπολογιστή υποστηρίζει τη γλώσσα PHP, και διαφορετικούς τύπους βάσεων δεδομένων αλλά Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

κυρίως χρησιμοποιεί την MySQL. Είναι μια εφαρμογή Web/Client, δηλαδή βρίσκεται εγκατεστημένο σε κάποιο Server και οι χρήστες έχουν πρόσβαση σ' αυτό μέσω ενός φυλλομετρητή, όπως είναι ο Internet Explorer, ο Mozilla Firefox κ.α. Επομένως από πλευράς χρήστη (μαθητή, καθηγητή και διαχειριστή) απαιτούνται μόνο η ύπαρξη μιας σύνδεσης στο Internet και το κατάλληλο λογισμικό περιήγησης.

Το Moodle ενσωματώνει λειτουργίες οι οποίες το καθιστούν ένα πολυσύνθετο, ευέλικτο και αρκετά ασφαλές εργαλείο στην ασύγχρονη εκπαίδευση από απόσταση. Είναι ένα εξαιρετικό **σύστημα διαχείρισης μάθησης (LMS)**. Είναι διαδεδομένο σε όλο τον κόσμο. Μεταξύ των οργανισμών που το χρησιμοποιούν είναι το MIT, το Yale και άλλα πανεπιστήμια στην Αμερική και στην Ευρώπη. Στην Ελλάδα η πλατφόρμα έχει εγκατασταθεί σε περισσότερους από 45 φορείς εκπαίδευσης και κατάρτισης, μεταξύ των οποίων το Εθνικό Μετσόβιο Πολυτεχνείο και τα Πανεπιστήμια Μακεδονίας και Θεσσαλίας.

Βασικά Χαρακτηριστικά:

- Οργάνωση του εκπαιδευτικού υλικού ανάλογα με τις απαιτήσεις που υφίστανται σε κάθε περίπτωση (π.χ. ανά εβδομάδα ή ανά θεματική ενότητα).
- Υποστήριξη μιας μεγάλης ποικιλίας δραστηριοτήτων διαφορετικού τύπου (Forums, Journals, Quizzes, Resources, Choices, Surveys και Assignments).
- Αυτόματη εγγραφή των φοιτητών μέσα από το Διαδίκτυο οι οποίοι στη συνέχεια εφ' όσον έχουν τα κατάλληλα δικαιώματα μπορούν να εγγραφούν στα μαθήματα της αρεσκείας τους χωρίς την παρέμβαση του διαχειριστή του μαθήματος.
- Παροχή υψηλού επιπέδου ασφαλείας.
- Αυτόματη βαθμολόγηση των διαγωνισμάτων με απευθείας ενημέρωση του φοιτητή.
- Δυνατότητα δημιουργίας του προσωπικού προφίλ για τους εγγεγραμμένους φοιτητές.
- Δυνατότητα ηλεκτρονικής υποβολής των εργασιών των φοιτητών στο σύστημα. Για αυτές τις εργασίες υπάρχει η δυνατότητα καθορισμού προθεσμίας υποβολής (Deadline).
- Δυνατότητα καταγραφής και ελέγχου των διάφορων τύπων δραστηριοτήτων των φοιτητών από το διαχειριστή του συστήματος.
- Υποστήριξη 75 και πλέον διαφορετικών φυσικών γλωσσών, μεταξύ των οποίων περιλαμβάνεται και η Ελληνική γλώσσα.

Ο σχεδιασμός και η δομή του Moodle επιτρέπει: (α) την επικοινωνία των μαθητών μεταξύ τους καθώς και με τον εκπαιδευτικό, με την αποστολή και λήψη μηνυμάτων με σύγχρονες (chat) και ασύγχρονες συζητήσεις (forum), (β) στην ενασχόληση των μαθητών με δραστηριότητες όπως η δημιουργία κοιζ ή λεξικών ή η συνεισφορά σε πηγές και ψηφιακό πολυμεσικό υλικό, (γ) η συμμετοχή σε συλλογική εργασία και σε διαδικασίες αυτοαξιολόγησης καθώς και εποικοδομητικού σχολιασμού. Υπό αυτή τη σκοπιά και όπως έχει επισημανθεί στην εισαγωγή του παρόντος εκπαιδευτικού υλικού, οι μαθητές έχουν δυνατότητες αυτενέργειας, καλλιεργούν την κρίση τους, την ικανότητα επίλυσης προβλημάτων και λήψης αποφάσεων σε ένα συνεργατικό πλαίσιο και ενισχύουν τις δεξιότητες επικοινωνίας τους.

Αναλυτικότερα, ο **συνολικός σχεδιασμός** του λογισμικού Moodle:

- Προωθεί τη παιδαγωγική θεωρία του κοινωνικού δομισμού (συνεργασία, δραστηριότητες).
- Παρέχει μονάδες με πολλά λειτουργικά στοιχεία.
- Εύκολη ενεργοποίηση/απενεργοποίηση των λειτουργιών του Moodle, ανάλογα με την εκάστοτε υλοποίηση.
- Εύκολη τροποποίηση του γραφικού περιβάλλοντος.
- Παρέχει κατάλογο μαθημάτων, με σύντομη περιγραφή για κάθε μάθημα και δυνατότητα πρόσβασης από επισκέπτες.

- Επιτρέπει την κατηγοριοποίηση και αναζήτηση των μαθημάτων. Μια μόνο πλατφόρμα Moodle μπορεί να υποστηρίξει ένα σημαντικό αριθμό μαθημάτων.
- Δημιουργία μπλοκ με ειδήσεις, συνδέσμους κλπ.
- Έμφαση στην ασφάλεια. Όλες οι φόρμες υποβάλλονται σε έλεγχο, τα cookies κρυπτογραφούνται.
- Οι περισσότερες περιοχές εισαγωγής κειμένου χρησιμοποιούν ενσωματωμένο επεξεργαστή κειμένου.

Στο Moodle πραγματοποιούνται τέσσερα **βασικά είδη διαδικασιών**, τα οποία αναλύουμε παρακάτω.

1. Αποθήκευση στο Moodle:

- Αρχεία (κείμενο, ήχο, βίντεο κ.α.)
- Ιστοσελίδες (HTML, ενσωμάτωση αντικειμένων, υπερσυνδέσεις, εισαγωγή εικόνων)
- Φακέλους (δημιουργία, μετονομασία, οργάνωση)
- Συνδέσμους (απευθείας σε ιστοσελίδες)
- Βάσεις δεδομένων (κατασκευή αποθετηρίου, αρχεία, σύνδεσμοι, εικόνες κ.α.)
- Πακέτα LMS (αντικείμενα μάθησης flash, SCORM)
- Χαρτοφυλάκιο (διατήρηση χαρτοφυλακίων, εγγράφων)
- Ετικέτες (οργάνωση περιεχομένου)

2. Επικοινωνία στο Moodle:

- Ομάδες Συζήτησης (forum): Οι μαθητές και οι εκπαιδευτικοί μπορούν να ανταλλάξουν ιδέες με την ανάρτηση σχολίων. Υπάρχουν τέσσερις βασικοί τύποι forum. Οι δημοσιεύσεις στα forum μπορεί να βαθμολογούνται από το εκπαιδευτικό ή τους άλλους μαθητές.
- Συζήτηση (chat): Η μονάδα συνομιλίας επιτρέπει στους συμμετέχοντες να έχουν σε πραγματικό χρόνο συζήτηση σε ένα μάθημα του Moodle.
- Μηνύματα: Τα "Μηνύματα" αναφέρονται τόσο σε αυτόματες ειδοποιήσεις από το Moodle για νέες δημοσιεύσεις, ειδοποιήσεις υποβολής, ανάθεσης κλπ, καθώς επίσης και σε συζητήσεις χρησιμοποιώντας τη λειτουργία instant messaging.
- Rss: Συνδρομή σε blogs, forums, wikis, εξωτερικές πηγές.
- Διάλογος: Ιδιωτικές Συζητήσεις μεταξύ χρηστών. Ανατροφοδότηση μαθητών με σχόλια ιδιωτικά.
- Ημερολόγιο: Το ημερολόγιο μπορεί να εμφανίσει γεγονότα του ιστοτόπου, του μαθήματος, και ομάδας χρηστών ή/και χρήστη επιπρόσθετα των προθεσμιών για εργασίες, κουίζ κ.λπ.

3. Συνεργασία στο Moodle:

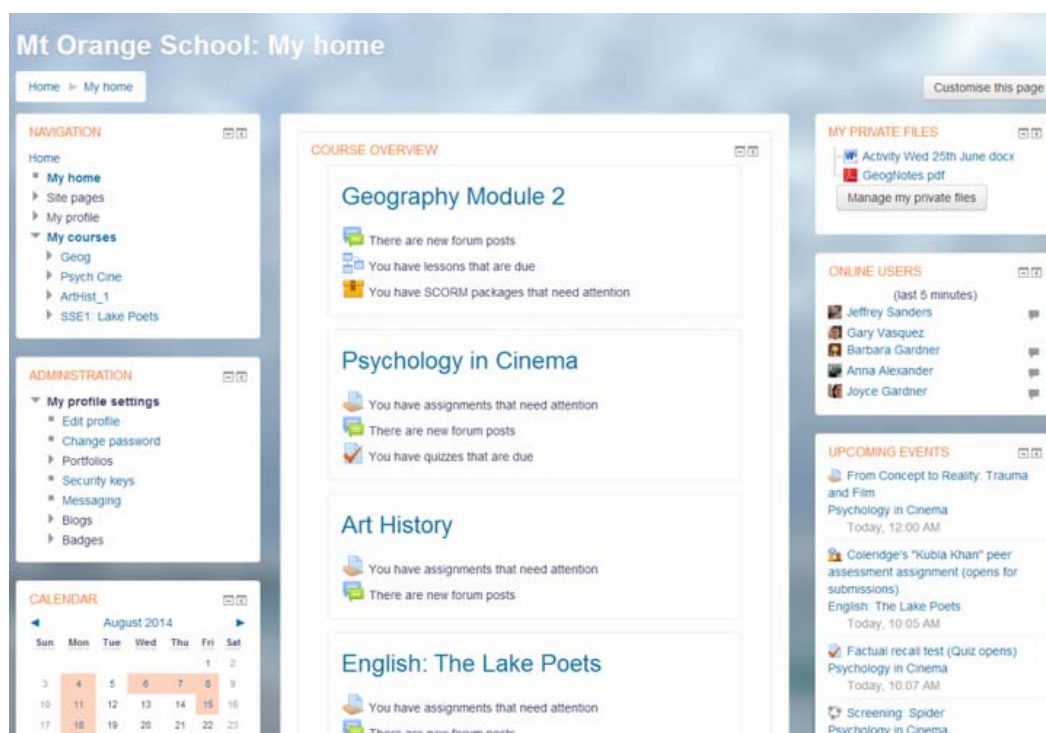
- Blogs: Τα Blogs στο Moodle είναι βάση των χρηστών - ο κάθε χρήστης έχει το δικό του blog. Οι χρήστες μπορούν επίσης να εγγράψουν τα εξωτερικά blogs τους, όπως στο Blogger ή Wordpress, έτσι ώστε οι αναρτήσεις τους να περιλαμβάνονται αυτόματα στο blog τους στο Moodle.
- Wiki: Στο Moodle τα wiki μπορεί να είναι ένα ισχυρό εργαλείο για τη συλλογική εργασία. Ολόκληρη η τάξη μπορεί να επεξεργαστεί ένα έγγραφο μαζί, ή ο κάθε μαθητής μπορεί να έχει το δικό του wiki και να εργάζονται σε αυτό με τον εκπαιδευτικό και τους συμμαθητές τους.
- Γλωσσάρι: Η μονάδα γλωσσάριο επιτρέπει στους συμμετέχοντες να δημιουργήσουν και να διατηρήσουν μια λίστα από ορισμούς, όπως σε ένα λεξικό.

- **Βάση Δεδομένων:** Η μονάδα δραστηριότητας Βάση Δεδομένων επιτρέπει στον καθηγητή και / ή στους μαθητές να δημιουργήσουν και να διαχειριστούν και να αναζητήσουν ένα σύνολο καταχωρήσεων για οποιοδήποτε θέμα. Η μορφή και η δομή αυτών των καταχωρήσεων μπορεί να περιλαμβάνει εικόνες, αρχεία, URL, αριθμούς και κείμενο μεταξύ άλλων.
- **Μάθημα-Ενότητα:** Ένα μάθημα εμφανίζει περιεχόμενο με έναν ενδιαφέρον και ευέλικτο τρόπο. Αποτελείται από ένα πλήθος σελίδων. Κάθε σελίδα κανονικά τελειώνει με μια ερώτηση και έναν αριθμό από πιθανές απαντήσεις. Ανάλογα με την απάντηση του φοιτητή είτε προχωράνε στην επόμενη σελίδα είτε πηγαίνουν στην προηγούμενη. Η πλοήγηση μέσα στο μάθημα μπορεί να είναι απλή ή πολύπλοκη και εξαρτάται από την δομή του υλικού που παρουσιάζεται.

4. Αξιολόγηση στο Moodle:

- **Κουίζ:** Η μονάδα δραστηριότητας κουίζ επιτρέπει στον καθηγητή να σχεδιάσει και να κατασκευάσει κουίζ που αποτελούνται από μια μεγάλη ποικιλία τύπων ερωτήσεων, συμπεριλαμβανομένων των πολλαπλής επιλογής, σωστό-λάθος και ερωτήσεις με σύντομες απαντήσεις. Αυτές οι ερωτήσεις φυλάσσονται στην Τράπεζα Ερωτήσεων και μπορούν να επαναχρησιμοποιηθούν σε διαφορετικά κουίζ.
- **Έρευνα:** Η μονάδα Έρευνα είναι μια δραστηριότητα που παρέχει έναν αριθμό ελεγμένων στοιχείων ερευνών (Survey), οι οποίες έχουν βρεθεί χρήσιμες στην αξιολόγηση της μάθησης σε online. Οι καθηγητές μπορούν να τα χρησιμοποιήσουν για να συλλέξουν δεδομένα από τους μαθητές τους, που θα τους βοηθήσουν να μάθουν την τάξη τους και να προβληματιστούν σχετικά με τη διδασκαλία τους.
- **Ερωτηματολόγιο:** Η μονάδα ερωτηματολόγιο είναι μια δραστηριότητα όπως η έρευνα. Επιτρέπει στους εκπαιδευτικούς να δημιουργήσουν ένα ευρύ φάσμα ερωτήσεων για να πάρουν ανατροφοδότηση από τους μαθητές π.χ. σε ένα μάθημα ή δραστηριότητες. Με το ερωτηματολόγιο δεν εξετάζεται ή αξιολογείται ο μαθητής, απλά συγκεντρώνονται δεδομένα.
- **Κλίμακες:** Οι Κλίμακες είναι ένας τρόπος για την αξιολόγηση ή βαθμολόγηση των επιδόσεων των μαθητών. Οι διαχειριστές μπορούν να δημιουργήσουν τυποποιημένες κλίμακες που είναι διαθέσιμες σε όλο τον ιστότοπο, και οι εκπαιδευτικοί μπορούν να δημιουργήσουν προσαρμοσμένες κλίμακες μόνο για το δικό τους μάθημα.
- **Ανάθεση Εργασιών:** Η μονάδα αυτή επιτρέπει στους καθηγητές να συλλέγουν εργασίες από τους μαθητές, να αναθεωρούν και να παρέχουν ανατροφοδότηση συμπεριλαμβανομένων των βαθμών. Η εργασία που υποβάλλουν οι μαθητές είναι ορατές μόνο στο εκπαιδευτικό και όχι στους άλλους μαθητές, εκτός εάν έχει επιλεγεί ανάθεση εργασίας σε ομάδα.
- **Βαθμολόγιο:** Όλοι οι βαθμοί για κάθε μαθητή σε ένα μάθημα βρίσκονται στο Βαθμολόγιο. Η αναφορά βαθμών συλλέγει τα στοιχεία που έχουν βαθμολογηθεί από τα διάφορα μέρη του Moodle που αξιολογούνται, και σας επιτρέπει να δείτε και να αλλάξετε τους βαθμούς, καθώς και να τους χωρίσετε σε κατηγορίες και να υπολογίσετε σύνολα με διάφορους τρόπους.

Σε κάθε μάθημα μπορούμε να βάλουμε όσα θέλουμε από τα βασικά δομικά συστατικά που αναφέραμε. Αλλά μπορούμε φυσικά και να προσθέσουμε όσα πρόσθετα (<https://moodle.org/plugins/>) θέλουμε για να επεκτείνουμε τις δυνατότητες του Moodle αλλά και των μαθημάτων που διαθέτουμε μέσω του Moodle.



Εικόνα 37: Περιβάλλον συστήματος διαχείρισης μάθησης, Moodle.

4. Σχεδίαση μαθημάτων - Υλοποίηση συστήματος

Η έρευνα στο χώρο της **πληροφορικής στην εκπαίδευση** έχει δείξει πως η αξιοποίηση των υπολογιστών και ειδικότερα δραστηριοτήτων με **πολυμεσικές δραστηριότητες**, μπορούν να έχουν ως αποτέλεσμα σημαντικά μαθησιακά αποτελέσματα (Virvou, Katsionis, & Manos 2005) καθώς και βελτίωση των κινήτρων για μάθηση (Virvou & Katsionis 2008).

Οι κατεξοχήν εφαρμογές για εκμάθηση βασικών εννοιών και αρχών του προγραμματισμού που προσφέρουν **πολυμεσικές δραστηριότητες** είναι οι **Προγραμματιστικοί Μικρόκοσμοι**. Οι προγραμματιστικοί μικρόκοσμοι (programming microworlds) αναπτύσσονται για καθαρά εκπαιδευτικούς σκοπούς, και βασίζονται σε μια φυσική μεταφορά (χελώνα, ρομπότ, γάτα, τρισδιάστατα αντικείμενα σε εικονικούς κόσμους), δηλαδή πρωταγωνιστές που “ζουν” μέσα στο περιβάλλον του μικρόκοσμου, είναι εύχρηστοι, ενσωματώνουν δυνατότητες οπτικοποίησης, ήχου, εφέ, και κίνησης (animation). Αυτοί οι μικρόκοσμοι, συνήθως χρησιμοποιούν μια εκπαιδευτική γλώσσα προγραμματισμού και παρέχουν τη δυνατότητα της δυναμικής προσομοίωσης της εκτέλεσης των προγραμμάτων, δηλαδή της βήμα προς βήμα εκτέλεσης του και συγχρόνως απεικόνιση του αποτελέσματος της εκτέλεσης στην κατάσταση του μικρόκοσμου.

Ένας από τους πλέον διαδεδομένους **προγραμματιστικούς μικρόκοσμούς** που προσφέρουν **πολυμεσικές δραστηριότητες** είναι το **Scratch**. Το Scratch είναι μια καινούρια logo-like γλώσσα προγραμματισμού, βασισμένη στον **Οπτικο Προγραμματισμό με Πλακίδια** που καθιστά εύκολο το να δημιουργήσεις τις διαδραστικές ιστορίες σου, κινούμενα σχέδια, παιχνίδια, μουσική και τέχνη. Η σημαντικότερη δυνατότητα που προσφέρουν τα σύγχρονα logo-like προγραμματιστικά περιβάλλοντα, είναι η διεπαφή «Οπτικού Προγραμματισμού με Πλακίδια» τα οποία συνδυάζονται μεταξύ τους ως κομμάτια παζλ. Πλεονέκτημα του Οπτικού Προγραμματισμού με Πλακίδια είναι ότι ο μαθητής αποφεύγει τα συντακτικά λάθη και εστιάζει στον αλγόριθμο.

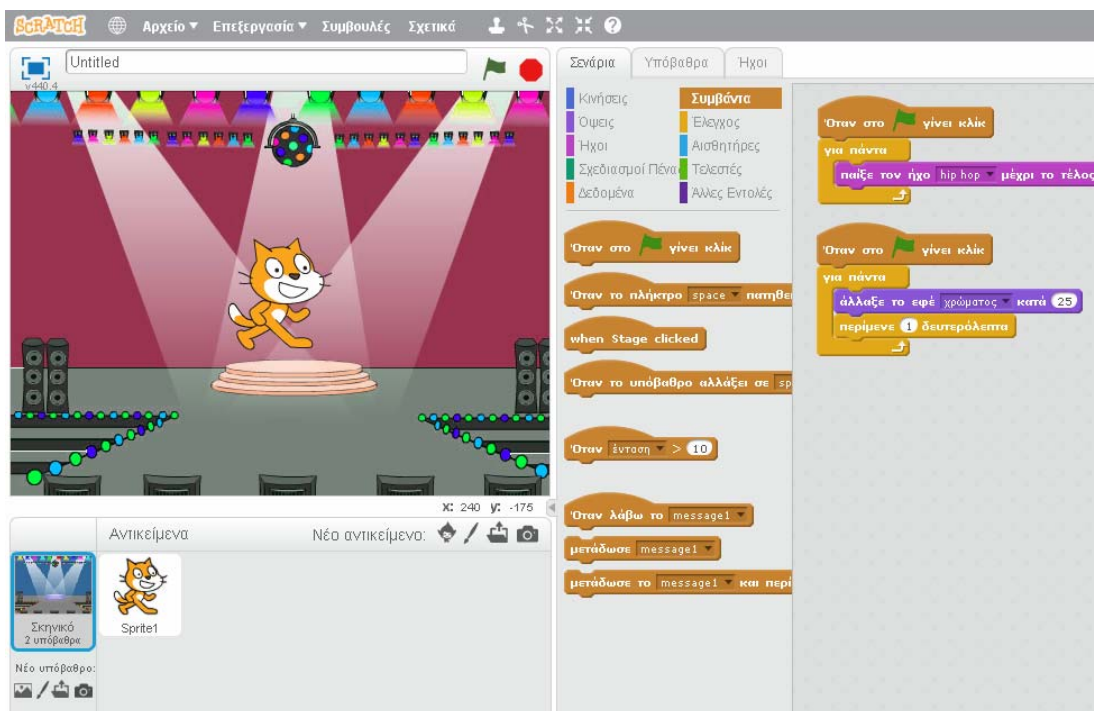
Έτσι, η καινοτομία του **Scratch** έγκειται στο γεγονός ότι δε χρειάζεται να πληκτρολογήσει κανείς ούτε μια γραμμή κώδικα για να γράψει ένα πρόγραμμα. Ξεχάστε λοιπόν το συντακτικό, γιατί αυτή η γλώσσα δομείται όπως ένα παζλ, είναι τόσο απλό. Το μόνο που χρειάζεται να κάνει κανείς είναι να σύρει μπλοκ (κομμάτια κώδικα) από το αριστερό μέρος της οθόνης και να τα κολλήσει με όποιον τρόπο θέλει στο μέσον της (σενάριο), ώστε το παραχθέν σενάριο να επιφέρει την επιθυμητή ενέργεια κατά την εκτέλεση του.

Για παράδειγμα, ένα σενάριο μπορεί να μοιάζει κάπως έτσι:



Εικόνα 38: Ένα σενάριο στο Scratch.

Το παραπάνω σενάριο θα κάνει το γατί σας (ο αρχικός χαρακτήρας της Scratch, ο οποίος βέβαια μπορεί να αντικατασταθεί από πληθώρα άλλων) να κινηθεί 30 βήματα, να νιαουρίσει, να κάνει άλλα τριάντα βήματα και να ξανά-νιαουρίσει! Όλα τα μπλοκ που φαίνονται στην εικόνα είναι έτοιμα, το μόνο που χρειάστηκε ήταν να κάνουμε ένα κολλάζ και να αλλάξουμε τις τιμές μετά τα “επανάλαβε” και “κινήσου”. Είναι τόσο απλό!



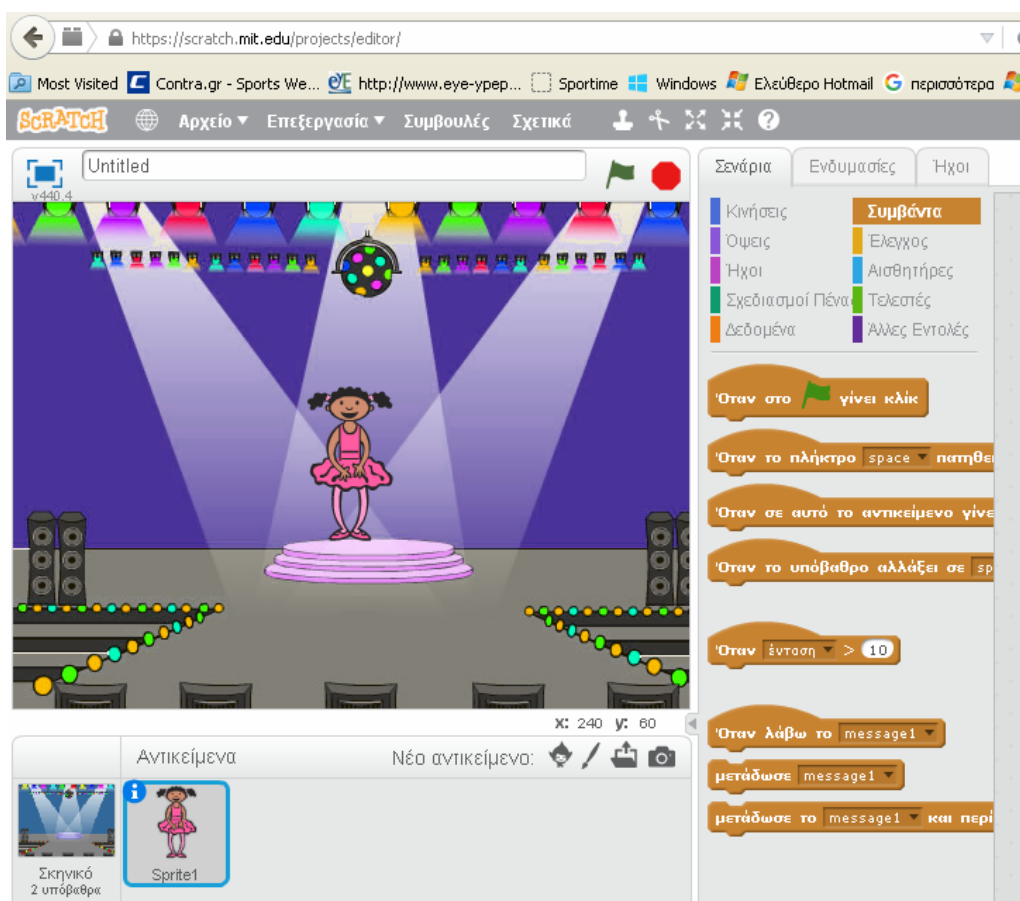
Εικόνα 39: Σενάριο στο περιβάλλον προγραμματισμού με πλακίδια, Scratch.

Εισαγωγή του Scratch στη μαθησιακή διαδικασία αρχών προγραμματισμού

Όσον αφορά του **μαθητές**, αν θέλαμε να αναφέρουμε κάποιους σημαντικούς λόγους για την εισαγωγή του περιβάλλοντος προγραμματιστικών μικρόκοσμων Scratch στην εκπαιδευτική διαδικασία, θα έπρεπε να σταθούμε στην υψηλή εμπλοκή των μαθητών, στην εσωτερίκευση του περιεχομένου καθώς και τις ευκαιρίες συνεργατικής μάθησης που προσφέρουν τέτοιου είδους δραστηριότητες.

Οι κατασκευαστές της Scratch (MIT Media Lab 2002) έχουν υποστηρίξει ότι τα παιδιά με τη βοήθεια του Scratch αναπτύσσουν μαθητικές δεξιότητες του 21ου αιώνα. Το θεωρητικό παιδαγωγικό υπόβαθρο στο οποίο στηρίζεται το Scratch είναι η πρακτική της μάθησης μέσω σχεδιασμού (learning by design) η οποία έχει τις ρίζες της στην κονστρουκτιβιστική θεωρία μάθησης του Piaget, καθώς και στην εκπαιδευτική προσέγγιση του κονστρουκτιονισμού που αναπτύχθηκε από τον Seymour Papert (Papert 1991).

Το περιβάλλον του Scratch προσφέρει και άλλα πολύ σημαντικά στοιχεία για τους μαθητές. Ικανοποιεί του στόχους του προγράμματος σπουδών, είναι logo-like περιβάλλον οπτικού προγραμματισμού με πλακίδια που χρησιμοποιεί διαδικασίες. Επιπρόσθετα, είναι εντελώς εξελληνισμένο, γεγονός που αποτελεί πάρα πολύ σημαντικό πλεονέκτημα για την χρήση του για διδασκαλία σε μαθητές μικρής ηλικίας. Τέλος, υπάρχει online έκδοση (<https://scratch.mit.edu/projects/editor/>) του προγράμματος για να μπορεί να την χρησιμοποιεί ο μαθητής (και ο εκπαιδευτικός φυσικά) από οπουδήποτε και με οποιαδήποτε συσκευή.

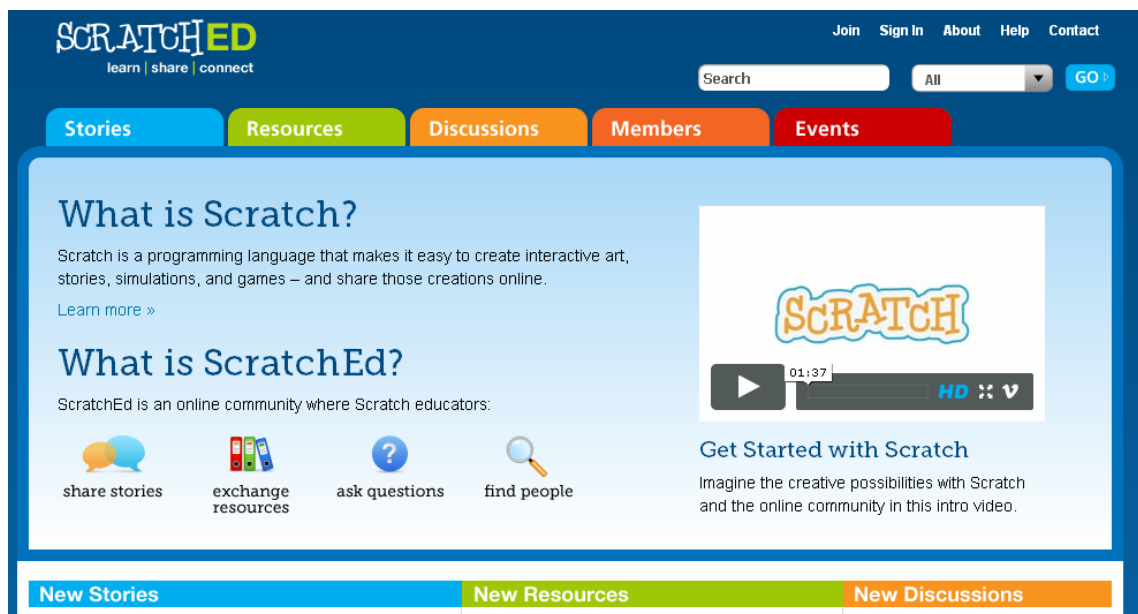


Εικόνα 40: Σενάριο στο ONLINE περιβάλλον προγραμματισμού με πλακίδια, Scratch.

Από την πλευρά των **εκπαιδευτικών** η δύναμη του Scratch κρύβεται στην ευκολία με την οποία μπορεί κανείς να μοιραστεί τις δημιουργίες του/της με τον υπόλοιπο κόσμο. Οι απίστευτες δυνατότητες του ίντερνετ ξεδιπλώνονται με το πάτημα ενός κουμπιού. Ο κάθε δημιουργός μπορεί να ανεβάσει στο δίκτυο της Scratch τα έργα του/της, για να τα μοιραστεί με φίλους από όλο τον κόσμο, σε μια δημιουργική και εκπαιδευτική συναλλαγή.

Το Scratch είναι ελεύθερο λογισμικό, είναι εξελληνισμένο (με πολλές ανανεώσεις, βιβλία και μελέτες σε ελληνικά τριτοβάθμια ιδρύματα), και υπάρχει εκτεταμένη κοινότητα μάθησης εστιασμένη στους σκοπούς του μαθήματος με την χρήση του περιβάλλοντος Scratch. Στον ιστοχώρο του Scratch υπάρχουν και πολλές άλλες πηγές μάθησης και υποστήριξης όπως: εκπαιδευτικά βίντεο, κάρτες Scratch και συχνές ερωτήσεις (<http://info.scratch.mit.edu/Support/>).

Σημαντικό χαρακτηριστικό του περιβάλλοντος και της φιλοσοφίας σχεδίασής του είναι **δυνατότητα διαμοιρασμού έργων** των χρηστών στην διαδικτυακή κοινότητα του Scratch. Ο κάθε χρήστης – δημιουργός έχει όχι μόνο την δυνατότητα δημοσίευσης του έργου του αλλά και της μεταφόρτωσης, χρήσης, τροποποίησης και επέκτασης έργων άλλων χρηστών (<http://scratchplay.wikispaces.com>).



Εικόνα 41: Η online διαδικτυακή κοινότητα ScratchEd, όπου οι εκπαιδευτές που χρησιμοποιούν Scratch μπορούν: να μοιράζονται σενάρια, να στέλνουν υλικό, να κάνουν ερωτήσεις, κ.α.

Εισαγωγή συστήματος τηλε-εκπαίδευσης Moodle στην μαθησιακή διαδικασία αρχών προγραμματισμού με την βοήθεια του περιβάλλοντος Scratch

Αν μάλιστα, αυτές οι **πολυμεσικές δραστηριότητες Οπτικού Προγραμματισμού με Πλακίδια** των μαθημάτων είναι διαθέσιμες και από απόσταση ώστε ο μαθητής μέσω ασύγχρονης **τηλε-εκπαίδευσης** να μπορεί να τις χρησιμοποιήσει όποτε μπορεί ή θέλει, τότε τα επιθυμητά μαθησιακά αποτελέσματα που προαναφέραμε μπορούν να γίνουν ακόμα καλύτερα.

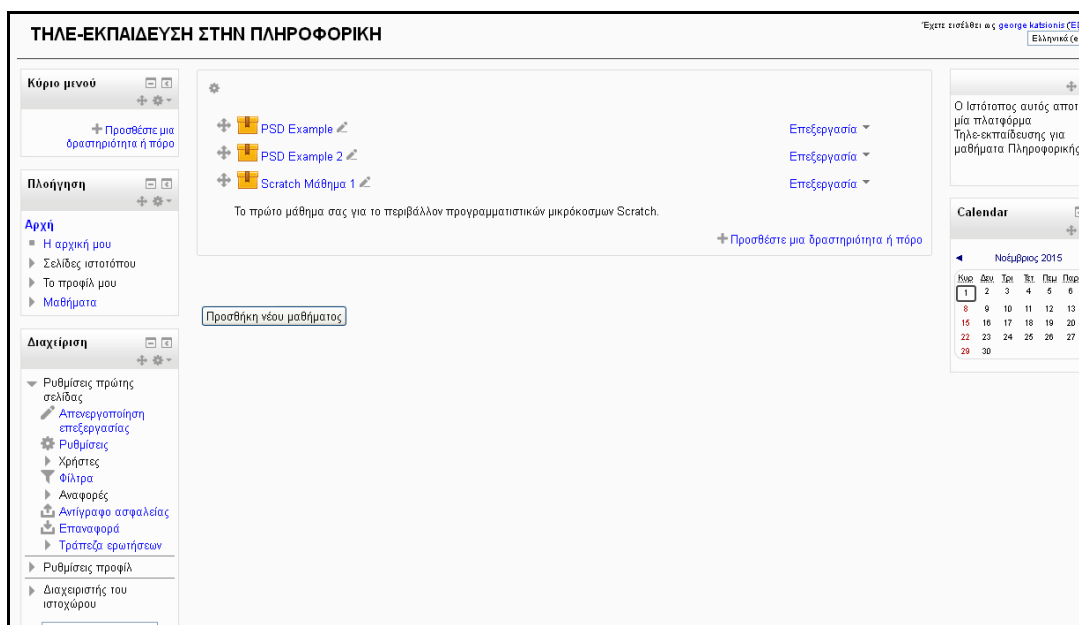
Το μοντέλο της **τηλεκπαίδευσης** δεν έχει σκοπό να αντικαταστήσει τη μέθοδο της κλασικής εκπαίδευσης, αλλά να την διευκολύνει και να την αναπτύξει, δίνοντας κίνητρο σε εκπαιδευτικούς και μαθητές να ασχοληθούν περισσότερο με τη διδασκαλία του μαθήματος, δημιουργώντας ένα υβριδικό μοντέλο διδασκαλίας (Κυνηγός 2006). Όπως ακριβώς η εξέλιξη της διδασκαλίας έχει εμπλουτίσει τα σχολεία με εργαστήρια πληροφορικής, έτσι και η τηλεεκπαίδευση θα συμβάλει

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

στο δύσκολο έργο του εκπαιδευτικού απλουστεύοντας και εμπλουτίζοντας κάποιες εργασίες μέσα από πληθώρα υπολειτουργιών με αλληλεπιδραστικό χαρακτήρα. Αυτονόητη είναι η διευκόλυνση που παρέχει ένα τέτοιο σύστημα σε δυσπρόσιτα μέρη ή σε περιπτώσεις κακοκαιρίας και ας μην ξεχνάμε ότι το μοντέλο αυτό ήδη λειτουργεί στα πανεπιστήμια σε όλο τον κόσμο. Ο εκπαιδευτής και ο εκπαιδευόμενος δεν είναι απαραίτητο να συνυπάρχουν στον ίδιο χώρο και χρόνο. Ο εκπαιδευτής μπορεί να παρουσιάσει το μάθημά του και ο εκπαιδευόμενος να μπορεί να ανατρέξει σε αυτό οποιαδήποτε στιγμή μπορεί ή θέλει.

Ένα από τα πιο διαδεδομένα **συστήματα διαχείρισης μάθησης** το οποίο παρέχει υπηρεσίες **τηλε-εκπαίδευσης** είναι το **Moodle**. Το **Moodle** (Dougiamas, 2001) είναι ένα πακέτο λογισμικού για την δημιουργία διαδικτυακών μαθημάτων. Οι δυνατότητες του δεν περιορίζονται στην εκπαίδευση από απόσταση αλλά μπορεί να λειτουργήσει συμπληρωματικά και στην κλασική διδασκαλία. Μέσα από την πλατφόρμα Moodle ο εκπαιδευτικός μπορεί να παρουσιάσει το μάθημα του με τέτοιο τρόπο ώστε να προκαλεί ενδιαφέρον στον εκπαιδευόμενο χρησιμοποιώντας διάφορα μέσα. Για παράδειγμα, διάθεση υλικού του μαθήματος σε διαφορετικές μορφές, προσθήκη δραστηριοτήτων, σύγχρονη και ασύγχρονη επικοινωνία μεταξύ μελών, ανάθεση εργασιών, αξιολόγηση της επίδοσης, κ.α.

Το **Moodle** επιλέγηκε ως πλατφόρμα διαχείρισης μάθησης επειδή είναι **Ανοικτό/Ελεύθερο Λογισμικό** το οποίο δεν προϋποθέτει κόστος (άδειες χρήσης, κτλ), μπορεί να τροποποιηθεί ελεύθερα, είναι εξελληνισμένο, υπάρχουν οδηγίες χρήσης και διαχείρισης που έχουν δημιουργηθεί στην Ελλάδα από το Πανεπιστήμιο Μακεδονίας, και ήδη χρησιμοποιείται τόσο από το Πανελλήνιο Σχολικό Δίκτυο, το Πανεπιστήμιο Μακεδονίας, το Ανοικτό Πανεπιστήμιο Κύπρου, καθώς και από το Υπουργείο Παιδείας και Πολιτισμού Κύπρου. Το Moodle, σύμφωνα με τους δημιουργούς του, είναι ένα **Κοινωνικό Σύστημα Διαχείρισης Μάθησης**, βασιζόμενο στις θεωρίες του Κοινωνικού Εποικοδομητισμού. Μέσα από τα εργαλεία που προσφέρει, ο εκπαιδευτικός μπορεί να δημιουργήσει σχετικά εύκολα, περιβάλλοντα στα οποία οι μαθητές μπορούν να συνομιλήσουν μεταξύ τους σε πραγματικό χρόνο (chat, instant messaging) ή με ασύγχρονες μεθόδους(email, forum), καθώς και να δημιουργήσει περιβάλλοντα για συστήματα τηλε-εκπαίδευσης.



Εικόνα 42: Περιβάλλον συστήματος διαχείρισης μάθησης και υπηρεσιών τηλε-εκπαίδευσης, Moodle.

Όπως προαναφέρθηκε, η πλατφόρμα Moodle διανέμεται ως λογισμικό ανοιχτού κώδικα (open source), μέσω Γενικής Άδειας Δημόσιας Χρήσης, (GNU). Το γεγονός αυτό συνεπάγεται

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

πως είναι δυνατή η λήψη του κώδικα από το Διαδίκτυο, η **ελεύθερη και χωρίς περιορισμούς χρήση** του, καθώς και επεμβάσεις, διορθώσεις και επαυξήσεις στον κώδικα αυτό. Το **κόστος συντήρησης** της προτεινόμενης υπηρεσίας είναι μηδενικό, όπως επίσης και ότι ο εκπαιδευτικός φορέας-μπορεί να δημιουργήσει υλικό και ηλεκτρονικές τάξεις για γνωστικά αντικείμενα που τον ενδιαφέρουν χωρίς περιορισμό.

Το Moodle βασίζεται στον Apache Web Server, στη βάση δεδομένων MySQL και στην τεχνολογία ανάπτυξης PHP, ενώ όλη η λειτουργία και διαχείρισή της γίνεται με τη χρήση ενός browser. Η λειτουργία του είναι κατάλληλα σχεδιασμένη έτσι ώστε να παρέχει εύκολη πρόσβαση ακόμα και σε χρήστες που συνδέονται με **συνδέσεις χαμηλής ταχύτητας**. Ταυτόχρονα είναι **συμβατή με τα διεθνή πρότυπα του SCORM** player + packages 1.2, του W3C και SENDA που εξασφαλίζουν συμβατότητα με τον οποιοδήποτε browser, με το οποιοδήποτε λειτουργικό σύστημα καθώς και ειδικό τρόπο εμφάνισης για άτομα με ειδικές ανάγκες. Υποστηρίζει πλήρως την ελληνική γραμματοσειρά και στο περιβάλλον διεπαφής όσο και στο υλικό του συστήματος.

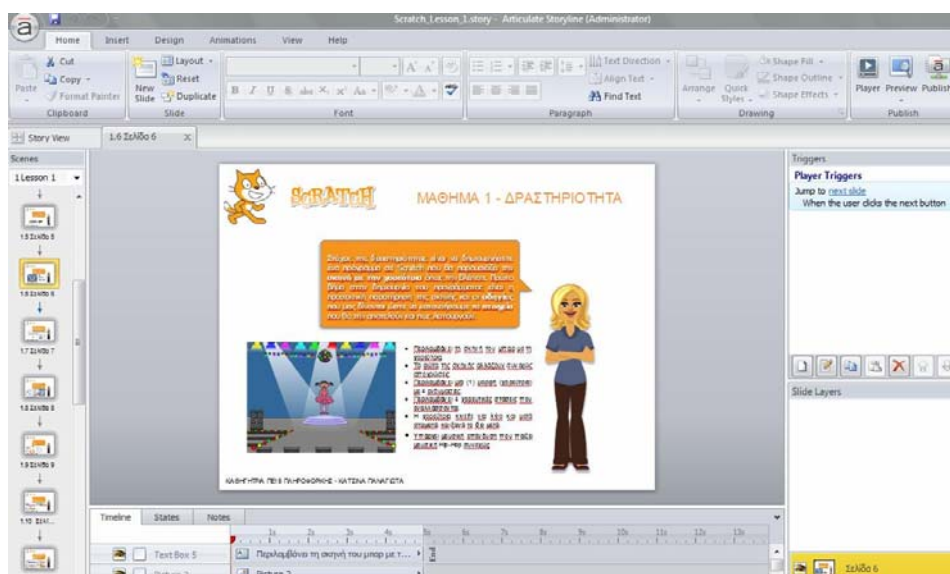
4.1 Σχεδίαση των μαθημάτων διδασκαλίας Scratch, θεωρία-δραστηριότητες-ερωτήσεις, σε μορφή συμβατή με το διεθνές πρότυπο SCORM 1.2, για χρήση σε τηλε-εκπαίδευση μέσω Moodle.

Η εξάπλωση του διαδικτύου, όπως επίσης και η εξέλιξη της ηλεκτρονικής μάθησης ενθάρρυναν την εμπλοκή των εκπαιδευτικών στη δημιουργία ηλεκτρονικών μαθημάτων, χωρίς να χρειάζεται, στην πλειονότητα τους, να έχουν προηγμένες τεχνολογικές δυνατότητες. Τα εργαλεία που υποστηρίζουν την ηλεκτρονική μάθηση (to learn anywhere anytime) ονομάζονται **εργαλεία συγγραφής** (authoring tools) και καθιστούν δυνατή την παραγωγή πολυμεσικών και διαδραστικών εφαρμογών. Στην «ανοιχτή και ευέλικτη» μάθηση (Σοφός & Κρον, 2010), η προσέγγιση της μάθησης λαμβάνεται από την πλευρά των μαθητών ως ενεργή διαδικασία. Ακόμη, ο εκπαιδευτικός με την σχεδίαση και δημιουργία μαθημάτων ψηφιακού περιεχομένου, προεκτείνει τα στενά όρια μιας συμβατικής διδασκαλίας η οποία στηρίζεται σε αναλυτικά προγράμματα. Μετασχηματίζει την διδακτέα ύλη στην κατάλληλη ψηφιακή δομή και μορφή υποστηρίζοντας τους πολλαπλούς τρόπους αναπαραγωγής της γνώσης. Η **τηλε-εκπαίδευση ή ηλεκτρονική μάθηση** η οποία στηρίζεται σε **εργαλεία συγγραφής ηλεκτρονικών μαθημάτων**, έχει ως στόχο τη μείωση του κόστους ανάπτυξης του μαθησιακού υλικού και τη διευκόλυνση των εκπαιδευτικών στη δημιουργία ηλεκτρονικών μαθημάτων, που θα είναι διαθέσιμα μέσω πλατφόρμας τηλε-εκπαίδευσης.

Τα **εργαλεία συγγραφής** είναι λογισμικά που επιτρέπουν τη σχεδίαση και ανάπτυξη **διαδραστικού εκπαιδευτικού ψηφιακού υλικού**. Είναι γενικά αποδεκτό πως για την παραγωγή του ίδιου περιεχομένου με ένα εργαλείο συγγραφής χρειάζεται αρκετά λιγότερος χρόνος σε σχέση με το κλασικό προγραμματισμό. Η σύνθεση πολυμεσικών εφαρμογών, με την χρήση αυτών των εφαρμογών, απαιτεί ελάχιστες γνώσεις προγραμματισμού και απευθύνεται σε αυτούς που ασχολούνται με τον εκπαιδευτικό σχεδιασμό γνωστικών αντικειμένων. Με τα εργαλεία συγγραφής μπορούμε να σχεδιάσουμε και να συνθέσουμε πολυμεσικά και υπερκειμενικά μαθησιακά αντικείμενα, συνθέτοντας κείμενο, εικόνα, video, ήχο, γραφικά, quiz, κ.α.

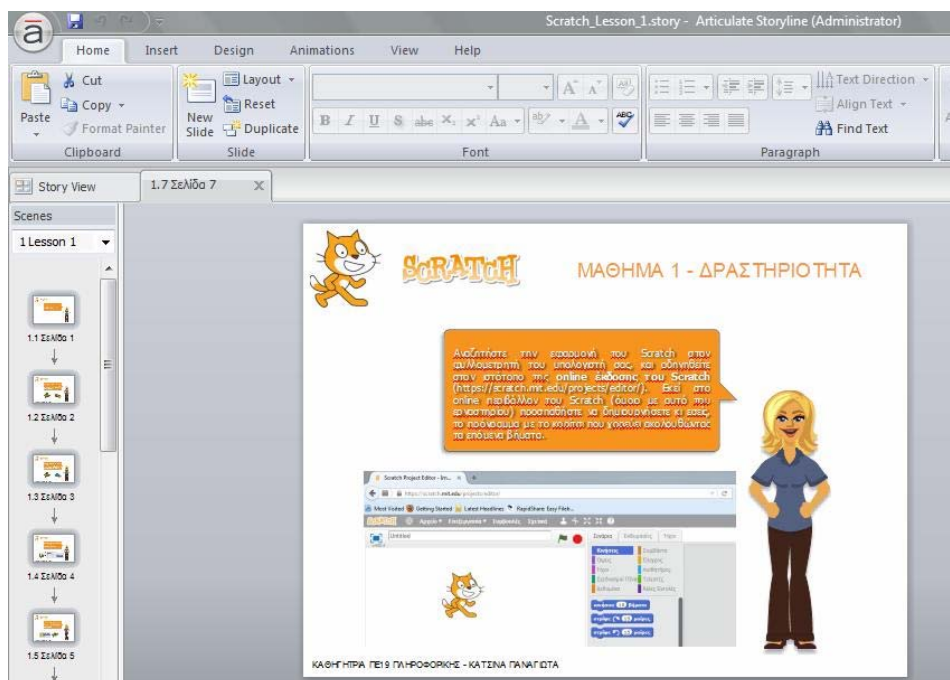
Τα παραγόμενα μαθήματα συσκευάζονται με διάφορους τρόπους, σύμφωνα με τα **διεθνή πρότυπα SCORM/AICC**, και διανέμονται είτε σε απλές ιστοσελίδες, είτε σε **συστήματα LMS** (Σύστημα Διαχείρισης Μάθησης). Η συγγραφή μαθησιακού υλικού (Learning Objects) ακολουθεί τα ακόλουθα στάδια: δημιουργία, μετατροπή, τεμαχισμός, συναρμολόγηση και επαναχρησιμοποίηση.

Η **σχεδίαση των μαθημάτων** του διδακτικού υλικού, που αφορούν το περιβάλλον προγραμματιστικών μικρόκοσμων Scratch, και πρέπει να είναι σε μορφή διεθνών προτύπων (SCORM) συμβατών με διαδικτυακά συστήματα διαχείρισης μάθησης και τηλε-εκπαίδευσης, όπως είναι το Moodle, έγινε με την δοκιμαστική έκδοση του προγράμματος **Articulate StoryLine** (Articulate Global Inc 2012).



Εικόνα 43: Περιβάλλον εργαλείου συγγραφής, Articulate StoryLine.

Το **Articulate StoryLine** ανήκει στα εργαλεία ανάπτυξης και δημοσίευσης σειράς μαθημάτων τα οποία είναι ειδικά δημιουργημένα εργαλεία για σχεδίαση, ανάπτυξη και δημοσίευση **ηλεκτρονικών μαθημάτων**, και προσφέρουν έτοιμα **πρότυπα** (templates) για διάφορους τύπους μαθημάτων. Επιπρόσθετα, μπορούν να αναδειχθούν σε πολύτιμα εργαλεία για εκείνους που διαθέτουν τις κατάλληλες δεξιότητες να συνθέσουν οποιασδήποτε μορφής ηλεκτρονικό μάθημα, που να περιλαμβάνει πολλές μορφές διδακτικού υλικού, χωρίς την υποστήριξη παραδοσιακών προγραμματιστικών δομών. Τέλος, το διδακτικό υλικό που δημιουργείται μπορεί να δημοσιευθεί σε **μορφή-πρότυπο** που να υποστηρίζεται από οποιοδήποτε website, από οποιαδήποτε συσκευή αναπαραγωγής (desktop, laptop, tablet, etc) και από τις περισσότερες eLearning πλατφόρμες συστημάτων διαχείρισης μάθησης (LMS).



Εικόνα 44: Διδακτικό Υλικό στο περιβάλλον του εργαλείου συγγραφής, Articulate StoryLine.

Ηλεκτρονικά μαθήματα δημιουργημένα σύμφωνα με τις προδιαγραφές που έχουμε θέσει (Scratch, SCORM, τηλε-εκπαίδευση)

Ακολουθούν παραδείγματα των ηλεκτρονικών μαθημάτων για τον προγραμματιστικό μικρόκοσμο του περιβάλλοντος Οπτικού προγραμματισμού με πλακίδια Scratch, τα οποία δημιουργήθηκαν στο Articulate StoryLine, έγιναν εξαγωγή από αυτό σύμφωνα με διεθνές πρότυπο ηλεκτρονικών μαθημάτων SCORM 1.2, και έγιναν εισαγωγή στο σύστημα τηλε-εκπαίδευσης Moodle.

Παραδείγματα θεωρίας:

Scratch_Lesson_1

Resources

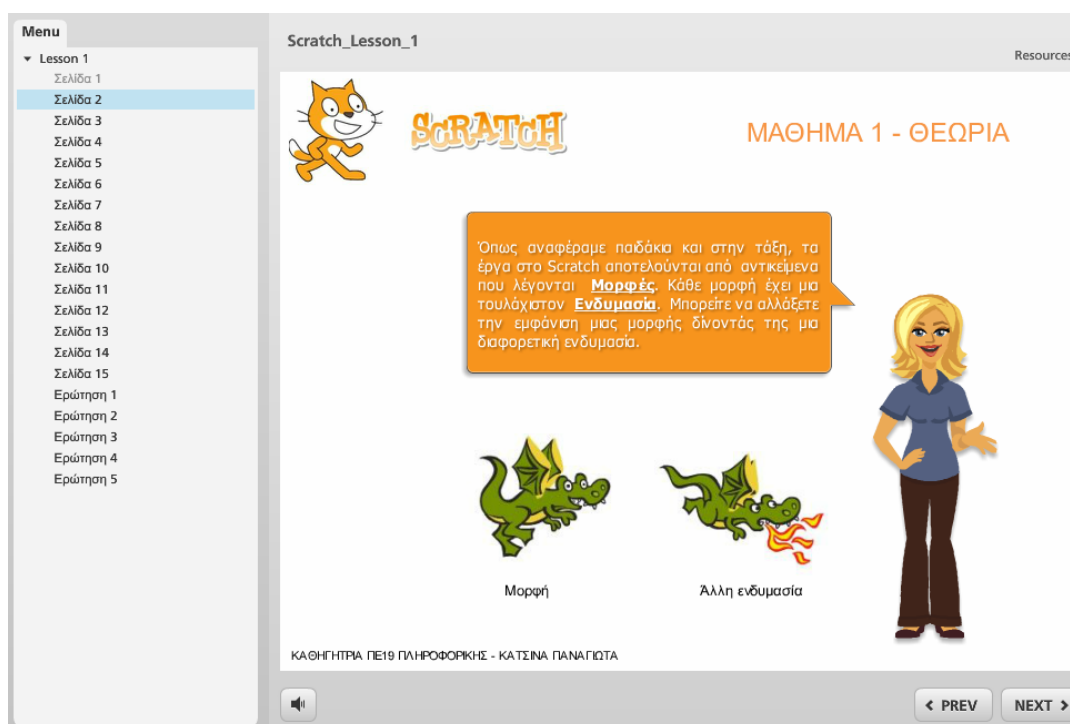
ΜΑΘΗΜΑ 1 - ΘΕΩΡΙΑ

Γεια σας παιδιάκια. Είμαι η κυρία σας, η Παναγιώτα. Καλώς ήλθατε στο 1ο μάθημα για το περιβάλλον προγραμματιστικών μικρόκοσμων SCRATCH.

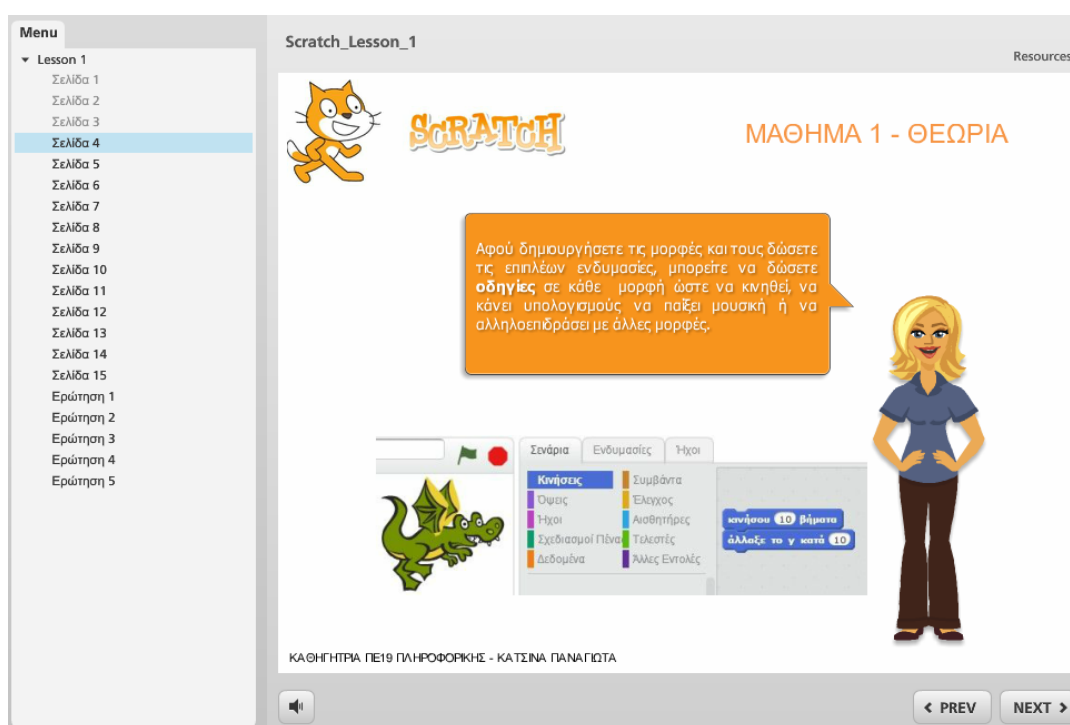
ΚΑΘΗΓΗΤΡΙΑ ΠΕ19 ΠΛΗΡΟΦΟΡΙΚΗΣ - ΚΑΤΣΙΝΑ ΠΑΝΑΓΙΩΤΑ

< PREV NEXT >

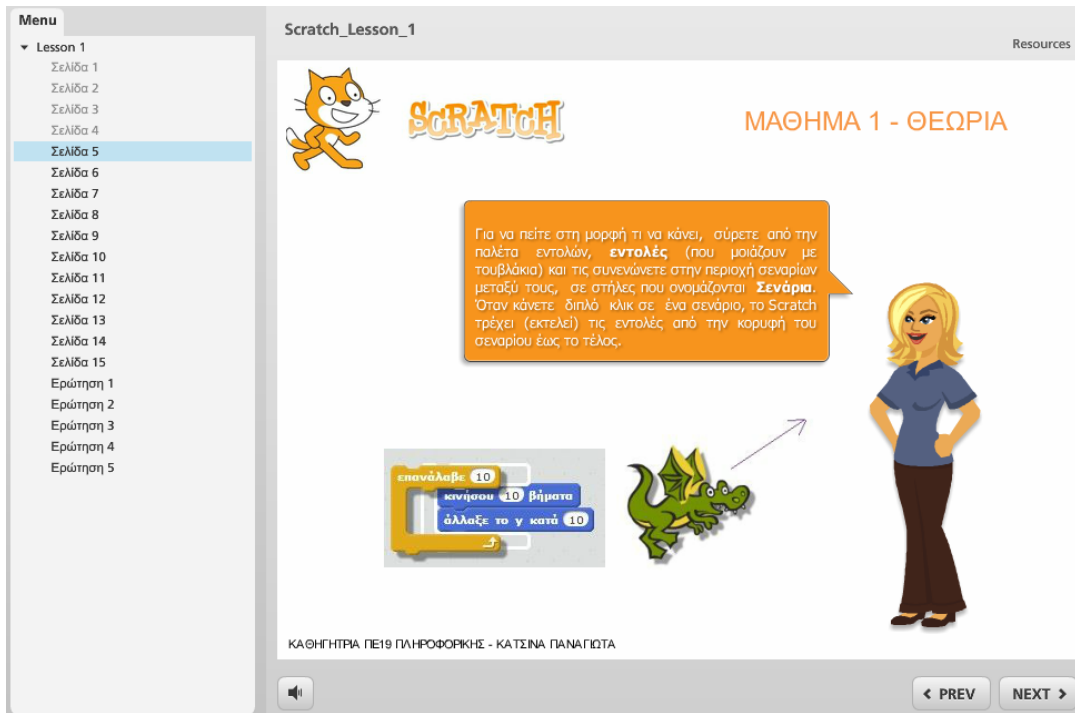
Εικόνα 45: Διαδικτυακό Υλικό Θεωρίας από το εργαλείο συγγραφής, Articulate StoryLine.



Εικόνα 46: Διαδικτυακό Υλικό Θεωρίας από το εργαλείο συγγραφής, Articulate StoryLine.

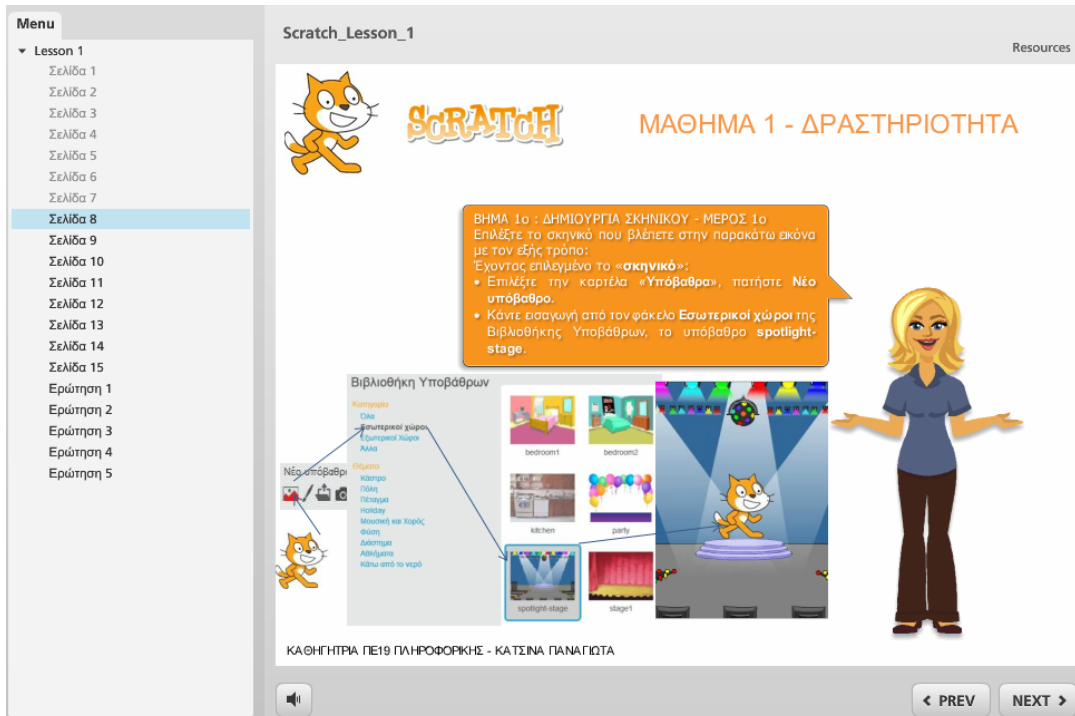


Εικόνα 47: Διαδικτυακό Υλικό Θεωρίας από το εργαλείο συγγραφής, Articulate StoryLine.

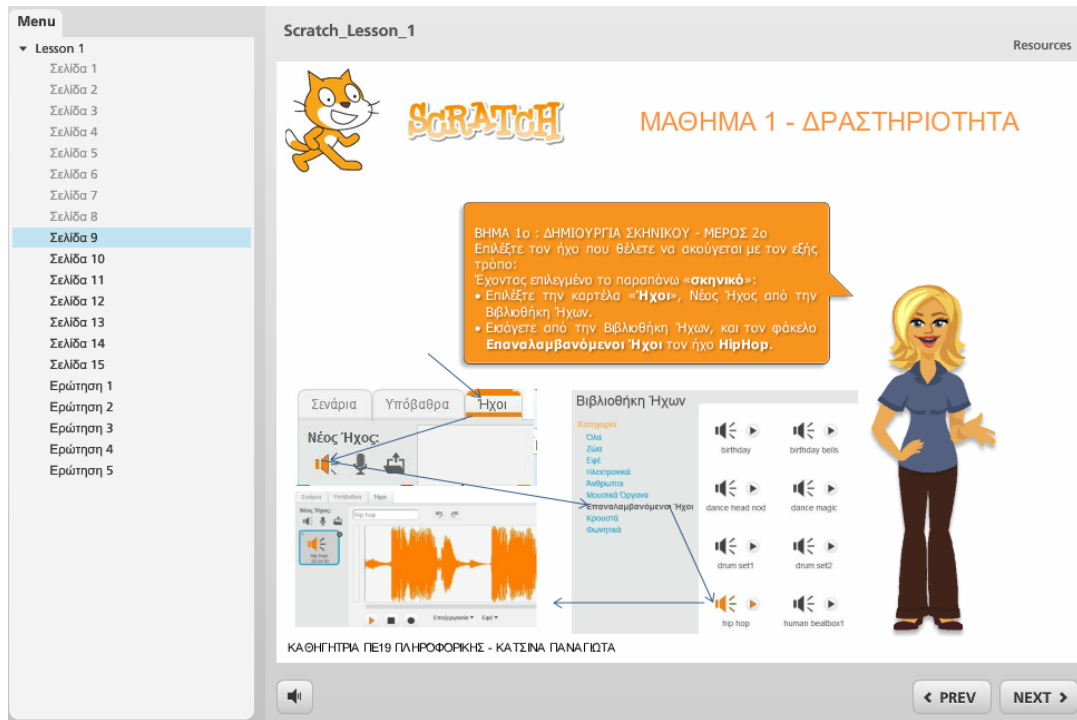


Εικόνα 48: Διαδικτυακό Υλικό Θεωρίας από το εργαλείο συγγραφής, Articulate StoryLine.

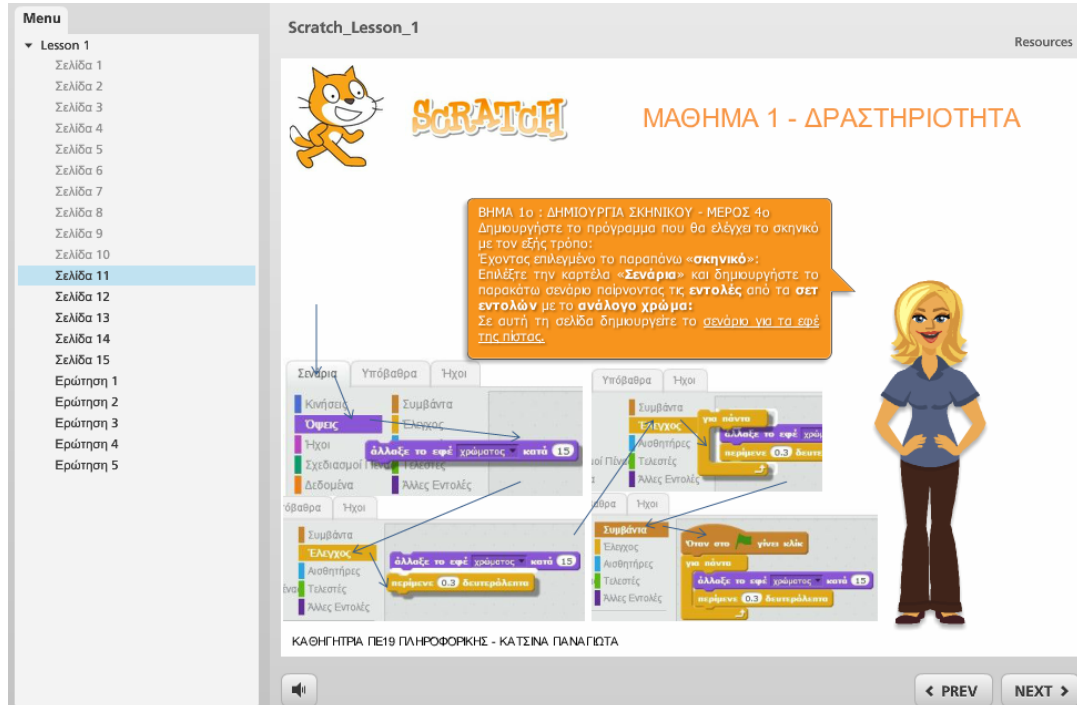
Παραδείγματα δραστηριοτήτων:



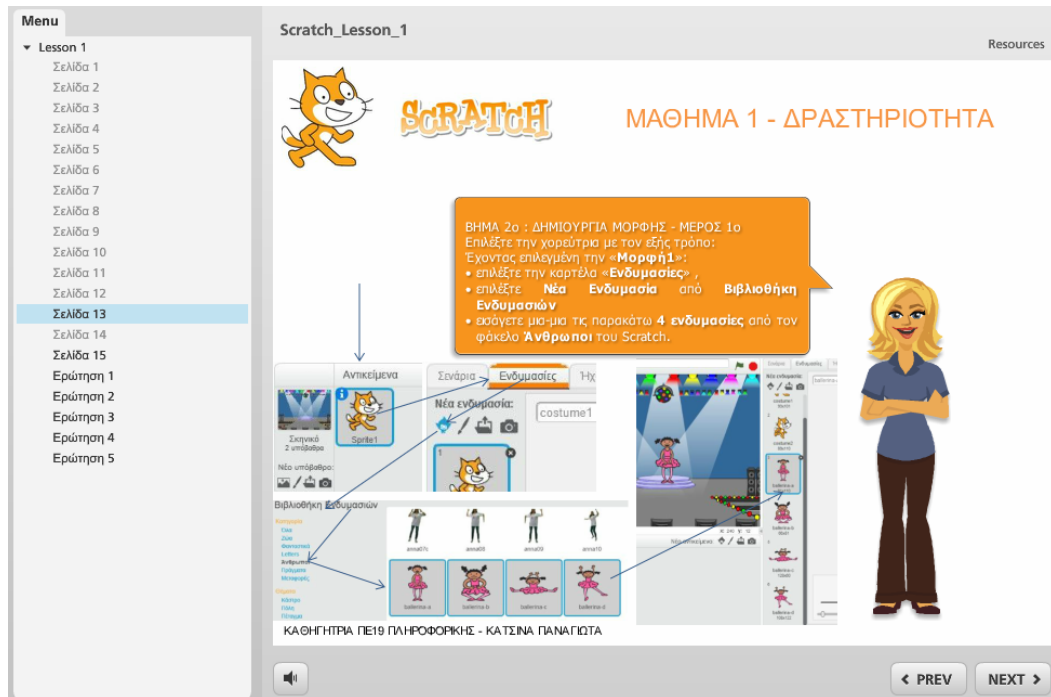
Εικόνα 49: Διαδικτυακό Υλικό Δραστηριοτήτων από το εργαλείο συγγραφής, Articulate StoryLine.



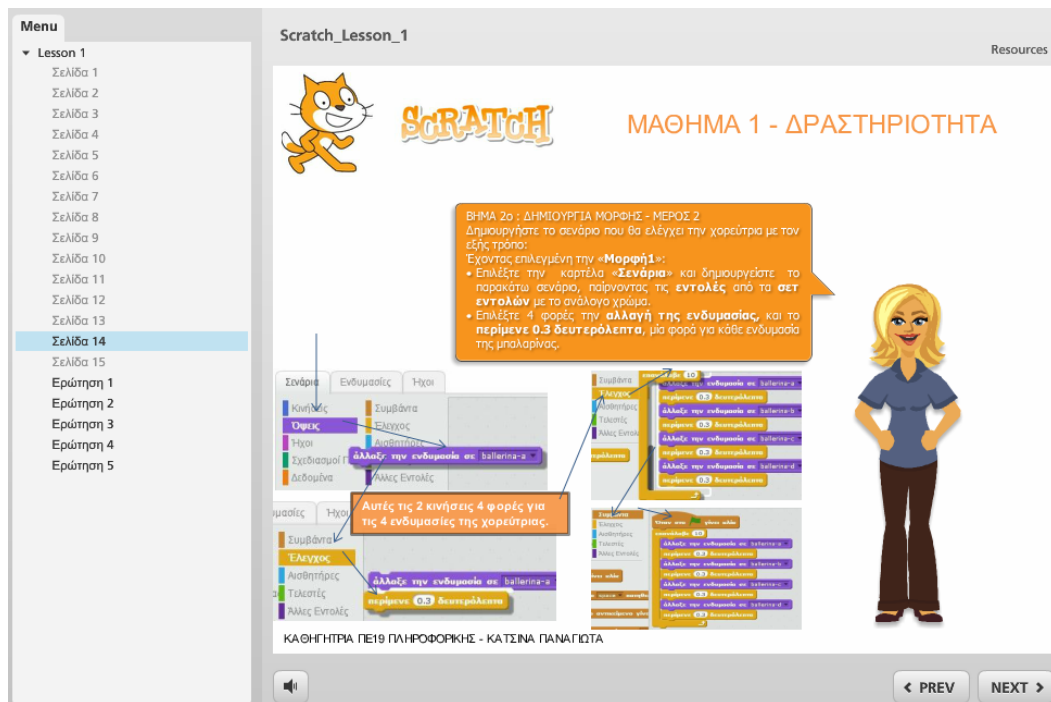
Εικόνα 50: Διαδικτυακό Υλικό Δραστηριοτήτων από το εργαλείο συγγραφής, Articulate StoryLine.



Εικόνα 51: Διαδικτυακό Υλικό Δραστηριοτήτων από το εργαλείο συγγραφής, Articulate StoryLine.



Εικόνα 52: Διαδικτυακό Υλικό Δραστηριοτήτων από το εργαλείο συγγραφής, Articulate StoryLine.



Εικόνα 53: Διαδικτυακό Υλικό Δραστηριοτήτων από το εργαλείο συγγραφής, Articulate StoryLine.

Παραδείγματα ερωτήσεων:

Scratch_Lesson_1 07:31 / 10:00 Resources

MAΘΗΜΑ 1 - ΕΡΩΤΗΣΕΙΣ

Ποιος είναι ο **ρόλος** του αριθμού **10** στην **εντολή επανάλαβε** του σεναρίου της μορφής της χορεύτριας, που σας παρουσιάζουμε στην διπλανή εικόνα;

- Να κάνει η μορφή της χορεύτριας 10 βήματα προς τα αριστερά.
- Να χορέψουν 10 χορεύτριες ταυτόχρονα.
- Να κάνει η μορφή της χορεύτριας 10 βήματα προς τα δεξιά.
- Να επαναληφθεί το χορευτικό με τις 4 φιγούρες, για 10 φορές.

ΚΑΘΗΓΗΤΡΙΑ ΠΕ19 ΠΛΗΡΟΦΟΡΙΚΗΣ - ΚΑΤΣΙΝΑ ΠΑΝΑΓΙΩΤΑ

SUBMIT

Εικόνα 54: Διαδικτυακό Υλικό Ερωτήσεων από το εργαλείο συγγραφής, Articulate StoryLine.

Scratch_Lesson_1 09:19 / 10:00 Resources

MAΘΗΜΑ 1 - ΕΡΩΤΗΣΕΙΣ

Τι θα συμβεί αν στην **εντολή επανάλαβε** (δείτε πάλι την διπλανή εικόνα), αλλάξουμε τον αριθμό **10** με τον αριθμό **2**; Να επιλέξετε **όσες απαντήσεις** είναι σωστές!

- Να επαναληφθεί το χορευτικό με τις 4 φιγούρες, για 2 φορές λιγότερο από πριν.
- Να επαναληφθεί το χορευτικό με τις 4 φιγούρες, για 2 φορές.
- Να επαναληφθεί το χορευτικό με τις 4 φιγούρες, για 12 φορές.
- Να επαναληφθεί το χορευτικό με τις 4 φιγούρες, 8 φορές λιγότερο από πριν.

ΚΑΘΗΓΗΤΡΙΑ ΠΕ19 ΠΛΗΡΟΦΟΡΙΚΗΣ - ΚΑΤΣΙΝΑ ΠΑΝΑΓΙΩΤΑ

SUBMIT

Εικόνα 55: Διαδικτυακό Υλικό Ερωτήσεων από το εργαλείο συγγραφής, Articulate StoryLine.

The screenshot shows a Scratch lesson titled "Scratch_Lesson_1" with a menu on the left listing pages 1 through 15 and questions 1 through 5. The main content area is titled "ΜΑΘΗΜΑ 1 - ΕΡΩΤΗΣΕΙΣ" and contains the following text: "Αν στο σενάριο της μορφής του κοριτσιού-χορεύτριας, κάνουμε κάποια από τις παρακάτω **αλλαγές** τι θα συμβεί; **Ενώστε** μεταξύ τους τις κατάλληλες επιλογές!!". Below the text are three pairs of boxes representing code changes and their effects:

- Box 1: "Αν αφαιρέσουμε όλες τις εντολές "περίμενε 0.3 δευτερόλεπτα"..." / "Αλλάζουν πολύ γρήγορα οι 4 φιγούρες της χορεύτριας για 10 φορές και σταματά!".
- Box 2: "Αν αλλάξουμε την τιμή του "περίμενε 0.3" σε "περίμενε 1.3"..." / "Η κάθε φιγούρα της μένει πιο πολύ ώρα ορατή και το χορευτικό έχει πιο αργό ρυθμό!".
- Box 3: "Αν αφαιρέσουμε τις εντολές "άλλαξε ενδυμασία σε ballerina-b", "άλλαξε ενδυμασία σε ballerina-c"..." / "Οι φιγούρες της χορεύτριας που επαναλαμβάνονται γίνονται μόνο 2!".

At the bottom, it says "ΚΑΘΗΓΗΤΡΙΑ ΠΕ19 ΠΛΗΡΟΦΟΡΙΚΗΣ - ΚΑΤΣΙΝΑ ΠΑΝΑΓΙΩΤΑ" and has a "SUBMIT" button.

Εικόνα 56: Διαδικτυακό Υλικό Ερωτήσεων από το εργαλείο συγγραφής, Articulate StoryLine.

The screenshot shows the same Scratch lesson interface, but with question 5 selected in the menu. The main content area contains the text: "Να **επιλέξετε** στα σενάρια σας, **το σημείο** που βρίσκεται το τουβλάκι που εξασφαλίζει την **αλλαγή του χρώματος** της σκηνής-πίστας!". Below the text are two code snippets:

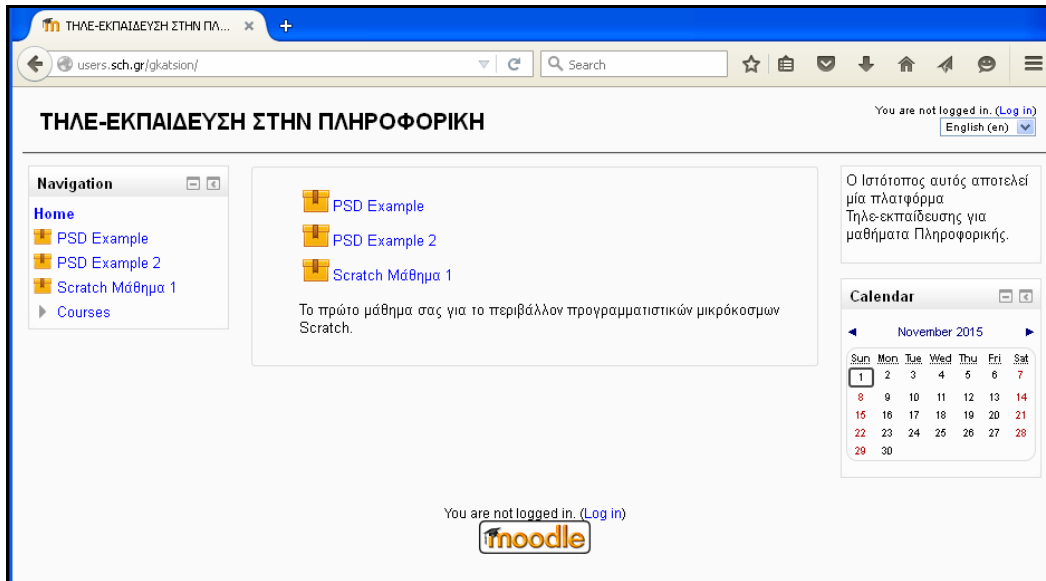
- Snippet 1: "Όταν στο **κλικ** γίνει κλικ" / "επανάληψη 10" / "άλλαξε την ενδυμασία σε ballerina-a" / "περίμενε 1.3 δευτερόλεπτα" / "άλλαξε την ενδυμασία σε ballerina-b" / "περίμενε 1.3 δευτερόλεπτα" / "άλλαξε την ενδυμασία σε ballerina-c" / "περίμενε 1.3 δευτερόλεπτα" / "άλλαξε την ενδυμασία σε ballerina-d" / "περίμενε 1.3 δευτερόλεπτα".
- Snippet 2: "Όταν στο **κλικ** γίνει κλικ" / "για πάντα" / "άλλαξε το **χρώματος** κατά 15" / "περίμενε 0.3 δευτερόλεπτα" / "Όταν στο **κλικ** γίνει κλικ" / "για πάντα" / "παίξε τον ήχο hip hop μέχρι το τέλος".

At the bottom, it says "ΚΑΘΗΓΗΤΡΙΑ ΠΕ19 ΠΛΗΡΟΦΟΡΙΚΗΣ - ΚΑΤΣΙΝΑ ΠΑΝΑΓΙΩΤΑ" and has a "SUBMIT" button.

Εικόνα 57: Διαδικτυακό Υλικό Ερωτήσεων από το εργαλείο συγγραφής, Articulate StoryLine.

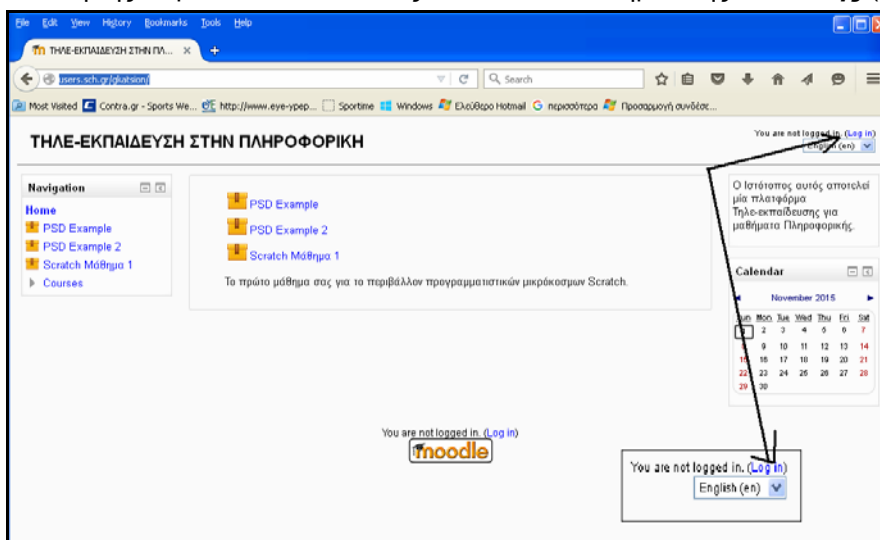
4.2 Υλοποίηση συστήματος τηλε-εκπαίδευσης πλατφόρμας Moodle, που υποστηρίζει SCORM πακέτα διδακτικού υλικού, που δημιουργήθηκαν για την υποστήριξη διδασκαλίας του περιβάλλοντος προγραμματιστικών μικρόκοσμων με πλακίδια Scratch.

Δημιουργήθηκε ιστοσελίδα στον εξυπηρετητή διαδικτύου του πανελληνίου σχολικού δικτύου και στον ειδικό χώρο που παρέχει για τους χρήστες του. Η ιστοσελίδα δημιουργήθηκε με χρήση του συστήματος διαχείρισης μάθησης Moodle, το οποίο παρέχει υπηρεσίες τηλε-εκπαίδευσης. Η **διεύθυνση διαδικτύου** του συστήματος διδασκαλίας που υλοποιήσαμε είναι: <http://users.sch.gr/gkatsion/>.



Εικόνα 58: Το διαδικτυακό σύστημα τηλε-εκπαίδευσης για την υποστήριξη της διδασκαλίας, σε πλατφόρμα Moodle.

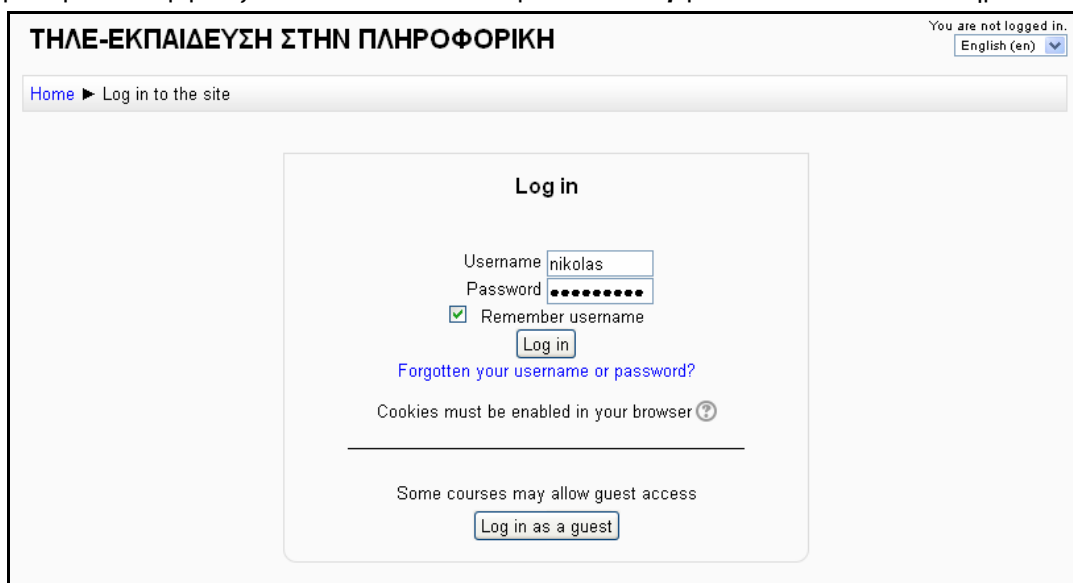
Για τον κάθε μαθητή δημιουργήθηκε **λογαριασμός** στο Moodle, μοναδικός, με όνομα χρήστη και κωδικό πρόσβασης για την διαδικασία σύνδεσης. Έτσι, ο κάθε μαθητής πληκτρολογεί την διεύθυνση της παραπάνω ιστοσελίδας και πατάει στο σημείο της **σύνδεσης** (Login).



Εικόνα 59: Σύνδεση στο διαδικτυακό σύστημα τηλε-εκπαίδευσης, σε πλατφόρμα Moodle.

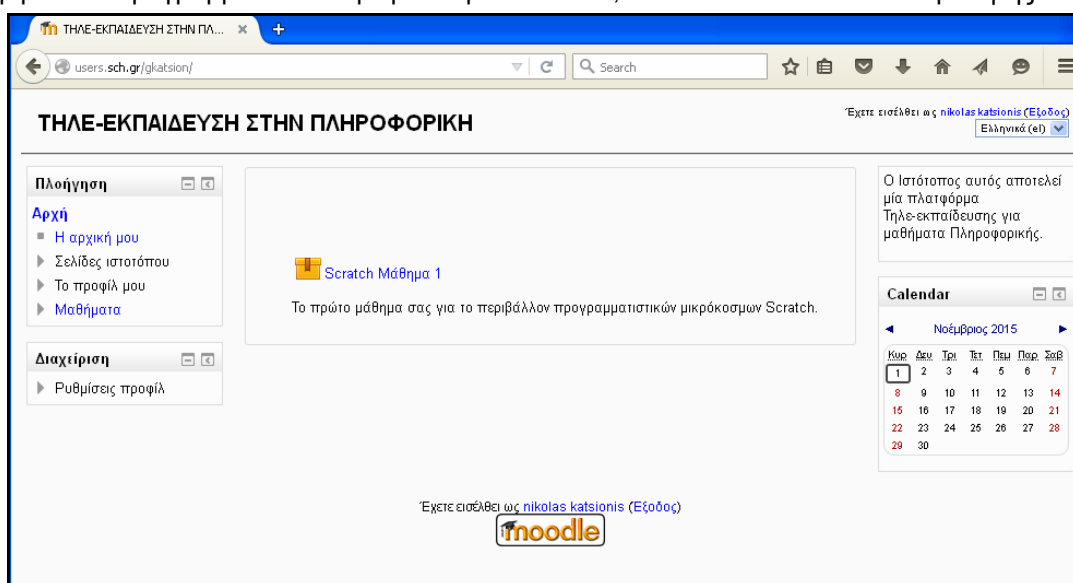
Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

Στην συνέχεια πληκτρολογεί το όνομα χρήστη και τον κωδικό πρόσβασης, στο ειδικό παράθυρο που εμφανίζεται και πατάει στο κουμπί **σύνδεση** για να εισέλθει στο σύστημα.



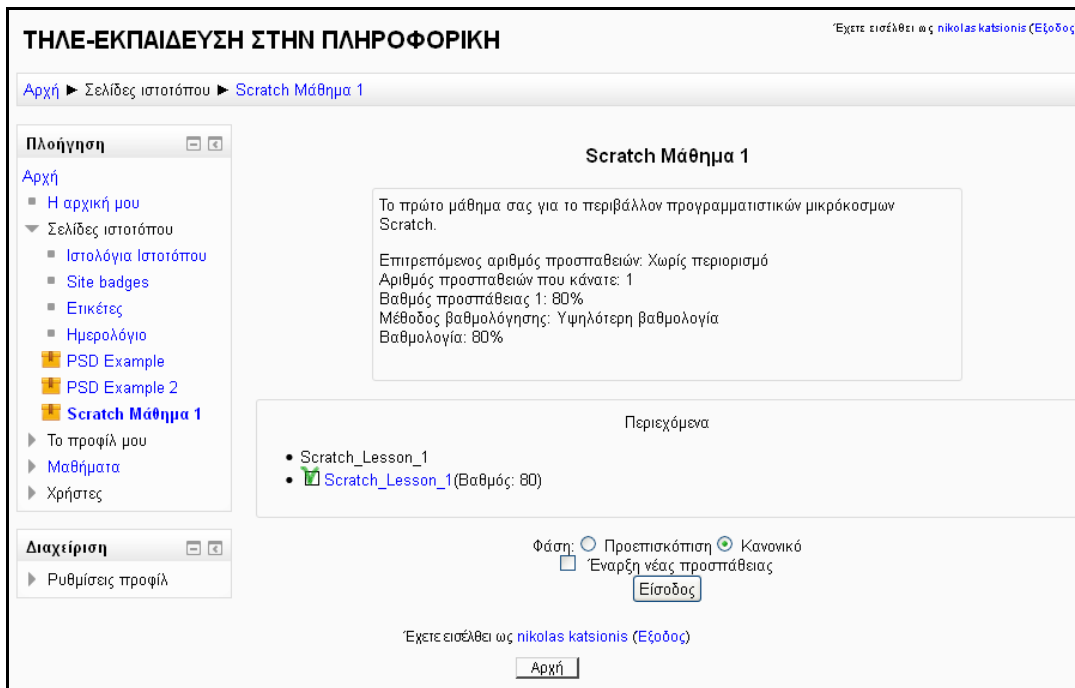
Εικόνα 60: Σύνδεση στο διαδικτυακό σύστημα τηλε-εκπαίδευσης, σε πλατφόρμα Moodle.

Κατά την είσοδο στο σύστημα τηλε-εκπαίδευσης ο μαθητής βλέπει τα διαθέσιμα μαθήματα που υπάρχουν. Στην παρακάτω εικόνα είναι ορατό το **πακέτο** του πρώτου μαθήματος για το περιβάλλον προγραμματιστικών μικρόκοσμων Scratch, πάνω στο οποίο πατάει ο μαθητής.



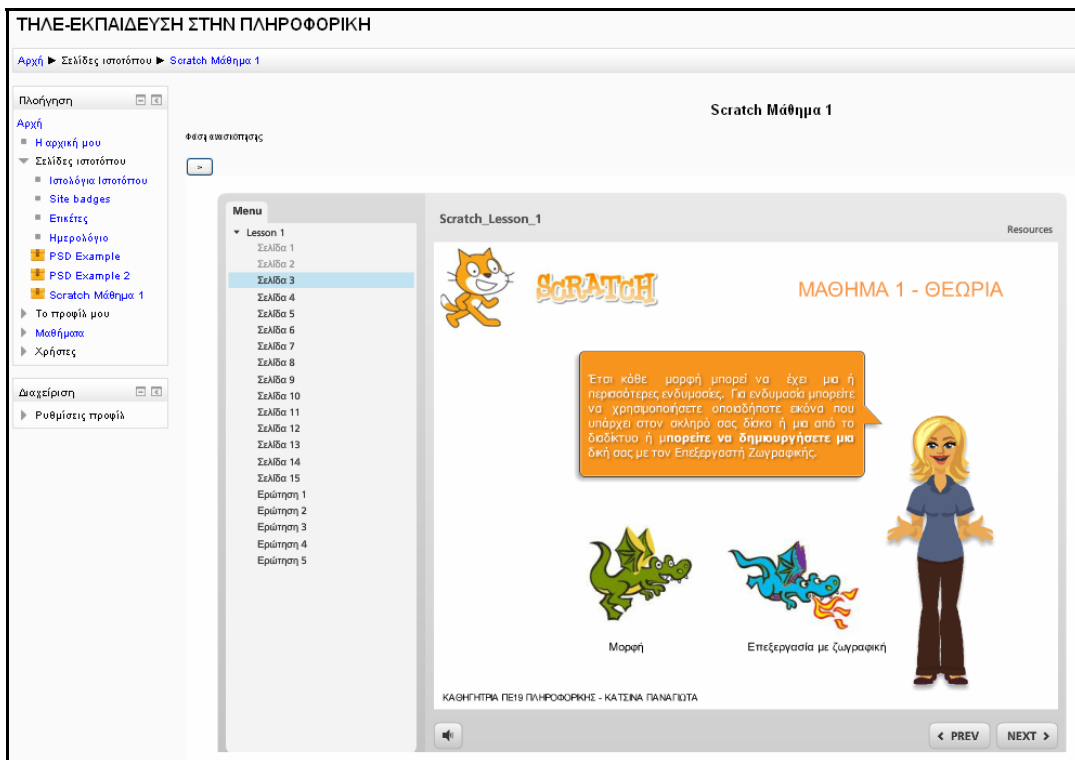
Εικόνα 61: Οθόνη χρήστη, και διαθέσιμα μαθήματα στο σύστημα τηλε-εκπαίδευσης, σε πλατφόρμα Moodle.

Στη συνέχεια γίνεται μία ξεχωριστή «είσοδος» στα πλαίσια του διδακτικού υλικού αυτού. Εκεί ο μαθητής μπορεί να δει **πληροφορίες** για το μάθημα αυτό, πόσες προσπάθειες έχει δικαίωμα να κάνει, πόσες έχει κάνει, ποιος είναι ο βαθμός, ποιο το σύστημα βαθμολόγησης, ποιος είναι ο μέγιστος βαθμός σε αυτή την ενότητα μαθήματος. Ο μαθητής πατάει **είσοδο**.

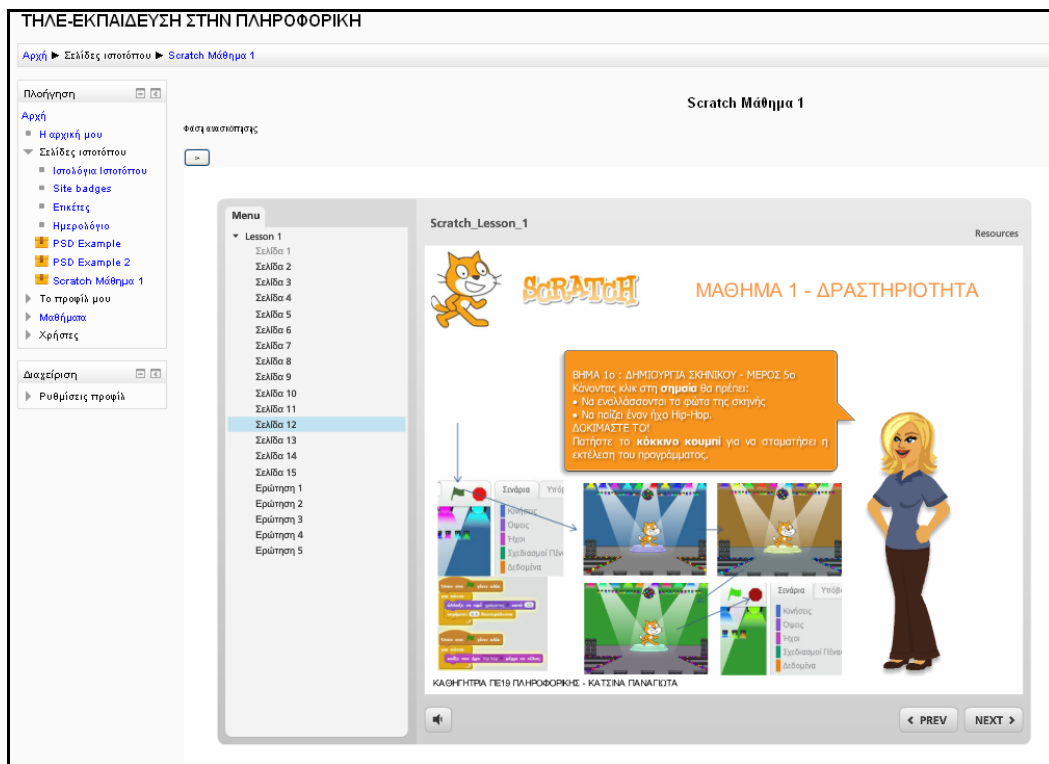


Εικόνα 62: Οθόνη μαθήματος με πληροφορίες για αυτό.

Αν επιλέξει *είσοδο*, εισέρχεται στην διδασκαλία του μαθήματος. Ο μαθητής έχει διαθέσιμες της οθόνες της θεωρίας, των δραστηριοτήτων, και των ερωτήσεων. Ξεκινά με την *θεωρία* και τις *δραστηριότητες*, όπου μελετά το διαθέσιμο διδακτικό υλικό των μαθημάτων.

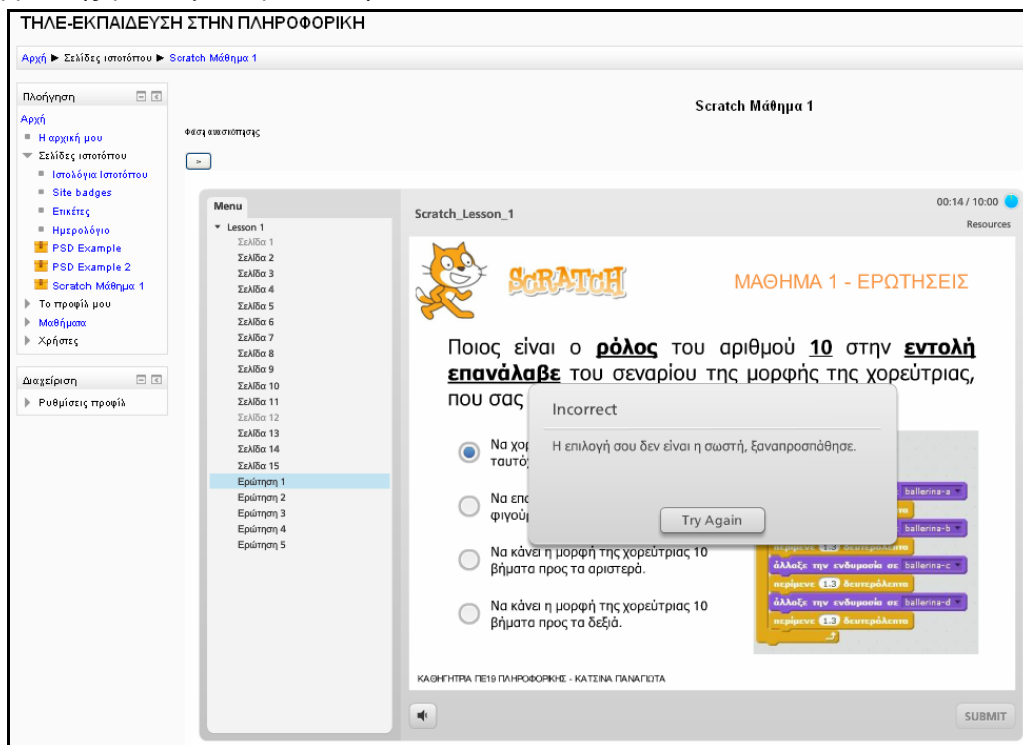


Εικόνα 63: Οθόνη θεωρίας μαθήματος.



Εικόνα 64: Οθόνη δραστηριότητας μαθήματος.

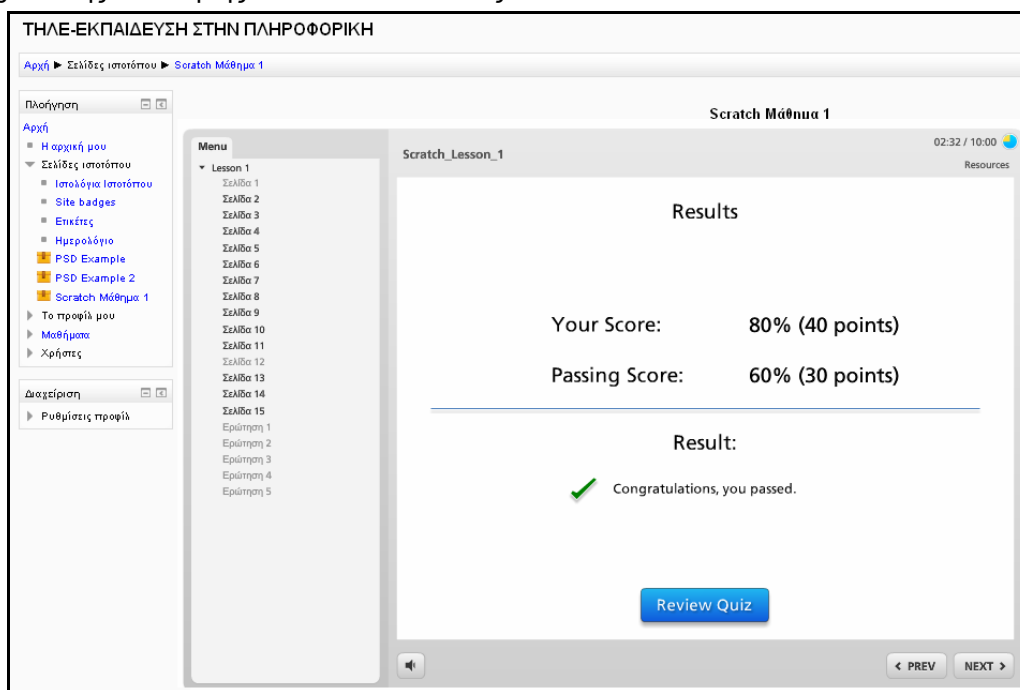
Στην συνέχεια περνάει στις **ερωτήσεις** της διδακτικής ενότητας-μαθήματος, όπου ανάλογα με την απάντηση του δέχεται **μηνύματα** επιτυχίας, αποτυχίας, ή λανθασμένης επιλογής και συμβουλής για να ξαναπροσπαθήσει.



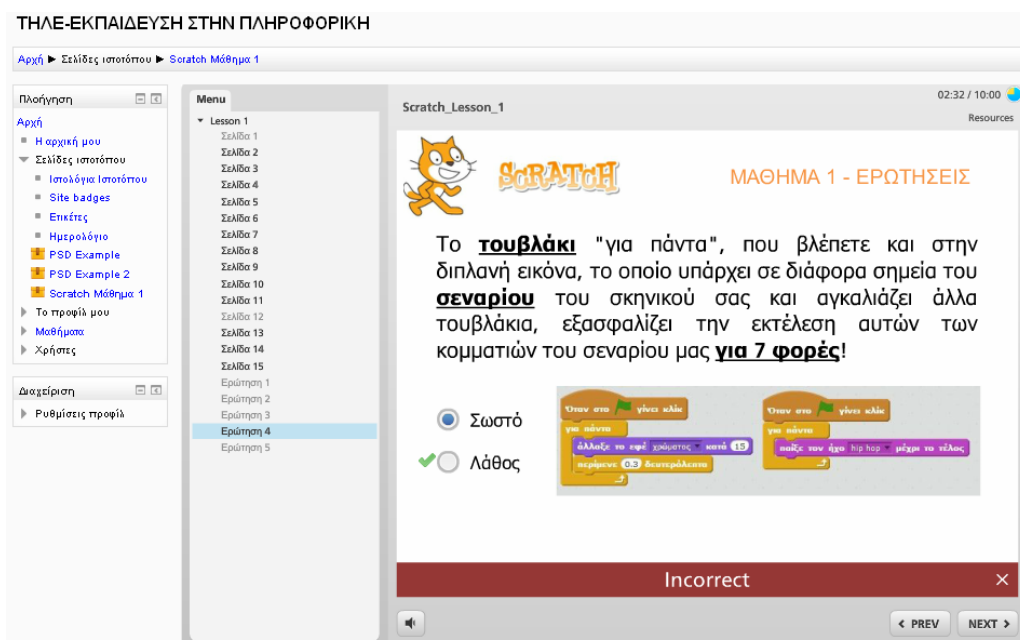
Εικόνα 65: Οθόνη ερώτησης μαθήματος, με ειδικό μήνυμα μετά την απάντηση του μαθητή.

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

Τελειώνοντας τις ερωτήσεις του μαθήματος, ο μαθητής έχει την ευκαιρία να δει τα **αποτελέσματα** των απαντήσεων του, καθώς και μπορεί να κάνει **ανασκόπηση** των απαντήσεων του βλέποντας τις σωστές και τις λάθος απαντήσεις του, συμπεριλαμβανομένου της σωστής απάντησης εκεί που έκανε λάθος.



Εικόνα 66: Οθόνη αποτελεσμάτων των απαντήσεων του μαθητή.



Εικόνα 67: Οθόνη ανασκόπηση των απαντήσεων του μαθητή, με τα λάθη και τις σωστές απαντήσεις.

Αυτές οι πληροφορίες που παρουσιάστηκαν στην υλοποίηση του συστήματος, καταγράφονται στο προφίλ του κάθε μαθητή ξεχωριστά, και είναι διαθέσιμες για αξιοποίηση και αλλαγές από τον καθηγητή-υπεύθυνο του συστήματος τηλε-εκπαίδευσης.

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

5. Αξιολόγηση της Μαθησιακής Διαδικασίας και του Συστήματος

Με τις αυξημένες απαιτήσεις που έθεσε το νέο πλαίσιο λειτουργίας των σχολείων είναι **επιτακτική πλέον η χρήση των εργαλείων της πληροφορικής** για την αξιολόγηση των εκπαιδευόμενων, πέρα βέβαια, από τα προφανή πλεονεκτήματα ακρίβειας και ταχύτητας που προσφέρει ο υπολογιστής.

Ο ρόλος τον δασκάλου μετατρέπεται από έναν απλό μεταδότη γνώσεων προς ένα **καταλύτη της αυτοδιδασκαλίας** των μαθητών. Με τη βοήθεια της τεχνολογίας ο εκπαιδευτικός θα μπορεί να εντοπίζει τις αδυναμίες και τις ιδιαιτερότητες για τον κάθε μαθητή έτσι ώστε να του παρέχει εξ' ατομικευμένη εκπαιδευτική υποστήριξη. Προς αυτή την κατεύθυνση θα βοηθήσει η αξιολόγηση των εκπαιδευόμενων μέσω ηλεκτρονικού υπολογιστή.

Κατά την διάρκεια της εκπαιδευτικής διαδικασίας η αξιολόγηση πρέπει να είναι συνεχής έτσι ώστε να **προσαρμόζεται η μάθηση** στις εκάστοτε ανάγκες των εκπαιδευόμενων. Οι εκπαιδευόμενοι έχουν ανάγκη την αξιολόγηση για να εντοπίσουν το σημείο στο οποίο βρίσκονται και να καθορίσουν τις ενέργειες που χρειάζεται να επιτελέσουν ώστε να επιτύχουν τους εκπαιδευτικούς στόχους τους ενώ οι εκπαιδευτικοί μέσω της αξιολόγησης θα μπορέσουν να εντοπίσουν πιθανές αδυναμίες ή/και παρανοήσεις των εκπαιδευόμενων και να προσαρμόσουν ανάλογα το μάθημά τους.

Η αξιολόγηση συνήθως επιτελείται σε **τρία καθορισμένα στάδια**:

- α) συλλογή δεδομένων: συλλέγονται τα γραπτά και οι απαντήσεις των εκπαιδευόμενων με σκοπό την περαιτέρω ανάλυσή τους
- β) ανάλυση των δεδομένων: γίνεται ανάλυση των δεδομένων που προκύπτουν από την αξιολόγηση. Η ανάλυση αυτή θα επιτρέψει την εξαγωγή χρήσιμων συμπερασμάτων.
- γ) αξιοποίηση των αποτελεσμάτων. Ο εκπαιδευτικός οφείλει να αξιοποιήσει τα συμπεράσματα που εξήχθησαν από την αξιολόγηση και να αναμορφώσει κατάλληλα τη διδασκαλία τον ώστε να επιτευχθεί ποιοτικότερη μάθηση. Ομοίως και οι εκπαιδευόμενοι οφείλουν, βασιζόμενοι στα αποτελέσματα της αξιολόγησης, να καθορίσουν με προσοχή τα επόμενα εκπαιδευτικά τους βήματα με σκοπό την καλύτερη μάθηση.

Οι παραπάνω διαδικασίες (στάδια) απαιτούν **πρόσθετο χρόνο και εργασία** από τους εκπαιδευτικούς. Ο περιορισμένος χρόνος τόσο των μαθητών όσο και των εκπαιδευτικών καθιστά την αξιολόγηση μέσω ηλεκτρονικού υπολογιστή ως μία πολύ καλή λύση στις ολοένα και αυξανόμενες ανάγκες αξιολόγησης. Αυτό το είδος αξιολόγησης μπορεί να βοηθήσει ιδιαίτερα στην διεκπεραίωση των δύο πρώτων σταδίων που προαναφέραμε (δηλαδή στην συλλογή των δεδομένων και στην ανάλυση των δεδομένων) και να δώσει άμεση ανατροφοδότηση τόσο στους εκπαιδευόμενους όσο και στους εκπαιδευτικούς ενώ μειώνει κατά μεγάλο βαθμό το φόρτο εργασίας που επιφορτίζονται οι εκπαιδευτικοί για την βαθμολόγηση των γραπτών των εκπαιδευόμενων.

Τα **πλεονεκτήματα** της αξιολόγησης μέσω ηλεκτρονικού υπολογιστή μπορούν να συνοψιστούν στα ακόλουθα:

- Χαμηλότερο κόστος σε βάθος χρόνου. Αν και η κτήση ηλεκτρονικών υπολογιστών είναι ένα αρχικό κόστος στη συνέχεια το λειτουργικό κόστος για ένα σύστημα ηλεκτρονικής αξιολόγησης είναι πολύ περιορισμένο. Επιπρόσθετα τα περισσότερα σχολεία - εκπαιδευτικοί οργανισμοί διαθέτουν τον απαιτούμενο υλικοτεχνικό εξοπλισμό και δεν επιβαρύνονται το αρχικό κόστος κτήσης τον εξοπλισμού.
- Άμεση ανατροφοδότηση εκπαιδευόμενων. Αυτό το χαρακτηριστικό ηλεκτρονικών συστημάτων αξιολόγησης επιτρέπει τους εκπαιδευόμενους να εντοπίσουν άμεσα και χωρίς τη μεσολάβηση εκπαιδευτικού τις αδυναμίες τους.
- Μεγαλύτερη ευελιξία σε σχέση με χρονικούς και χωρικούς περιορισμούς. Επιτρέπει ακόμα και την αξιολόγηση εκπαιδευόμενων από το σπίτι τους σε ώρες που αυτοί επιλέγουν ή που προκαθορίζονται από τον εκπαιδευτικό.

- Αυξημένη αξιοπιστία. Τόσο η διαδικασία αυτόματης δημιουργίας ερωτηματολογίων όσο και η αξιολόγηση των εκπαιδευόμενων στηρίζεται σε συγκεκριμένους αλγόριθμους επιτρέποντας αυξημένη αξιοπιστία.
- Αμερόληψια. Ο υπολογιστής δεν «γνωρίζει τους εκπαιδευόμενους έτσι δεν θα βοηθήσει κάποιους μαθητές και η βαθμολογία θα είναι αμερόληπτη
- Μεγαλύτερη ικανότητα αποθήκευσης για μεγάλο χρονικό διάστημα. Ο υπολογιστής μπορεί να αποθηκεύσει τις απαντήσεις χιλιάδων εκπαιδευόμενων σε έναν απλό σκληρό δίσκο σε αντίθεση με τον χώρο που θα έπρεπε να δεσμευτεί στην περίπτωση χρήσης χαρτιού.
- Εμπλουτισμένοι τύποι ερωτήσεων οι οποίοι μπορούν να ενσωματώνουν αλληλεπίδραση και πολυμέσα.

Τέλος, η αξιολόγηση της ανάπτυξης των μαθητών σε μαθήματα τα οποία χρησιμοποιούν Τεχνολογίες Πληροφοριών και Επικοινωνίας, όπως τα **περιβάλλοντα προγραμματιστικών μικρόκοσμων**, δεν μπορεί εκ των πραγμάτων να υλοποιηθεί μόνο με τη χρήση συμβατικών τεστ με χαρτί και μολύβι.

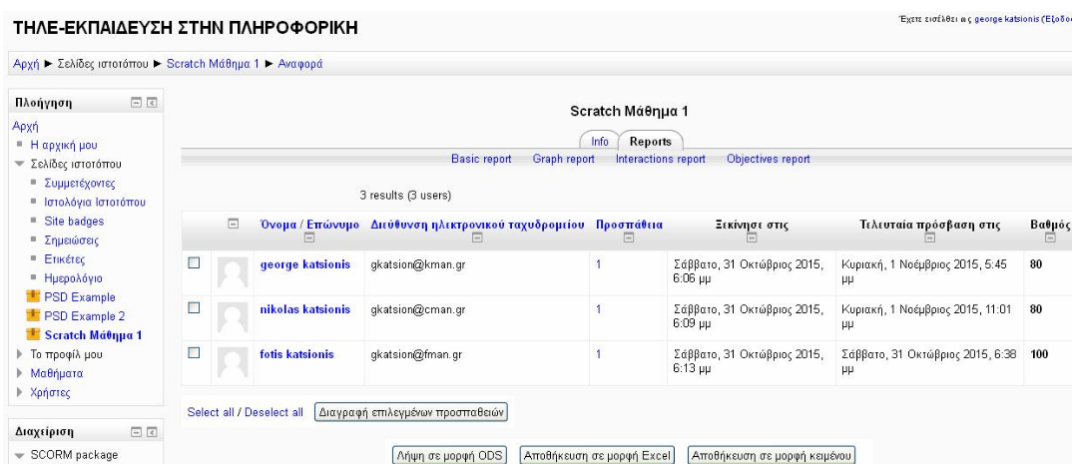
Για την ουσιαστική συνολική αξιολόγησή των μαθητών, στο μάθημα αυτό θα αξιολογείται:

- αφενός η καθημερινή εργασία στην τάξη,
- αφετέρου τα ολοκληρωμένα ψηφιακά έργα που δημιουργούν στα πλαίσια του μαθήματος,
- και εργασίες και δοκιμασίες που θα εκτελούνται-αποθηκεύονται στο ηλεκτρονικό προφίλ του μαθητή,
- το οποίο θα τηρείται στην ψηφιακή πλατφόρμα τηλε-εκπαίδευσης της τάξης.

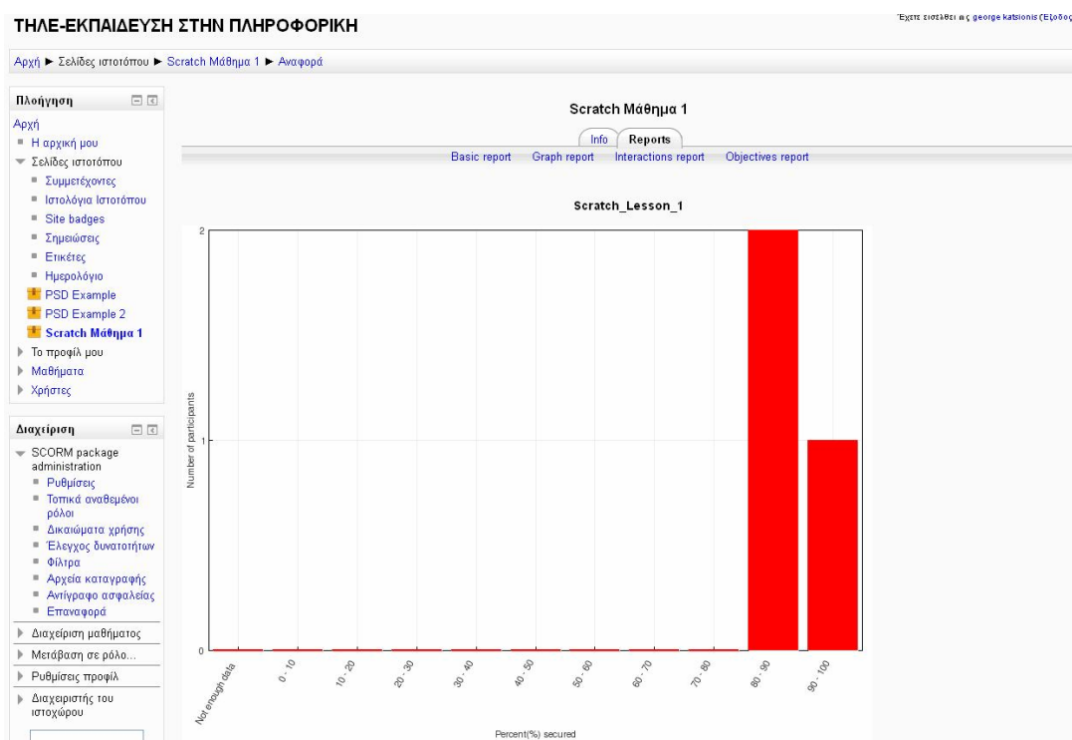
Στην δική μας περίπτωση λοιπόν, του **συστήματος τηλε-εκπαίδευσης για υποστήριξη της διδασκαλίας μαθημάτων προγραμματισμού** στο περιβάλλον προγραμματιστικών μικρόκοσμων με πλακίδια του Scratch, η υποστήριξη της αξιολόγησης του κάθε μαθητή γίνεται από το σύστημα διαχείρισης μάθησης και τηλεκπαίδευσης μέσω Moodle και του αντίστοιχου ιστοτόπου που έχει δημιουργηθεί για αυτό.

Για τον κάθε μαθητή δημιουργήθηκε **λογαριασμός** στον ιστότοπο του συστήματος τηλεκπαίδευσης. Το σύστημα τηλεκπαίδευσης καταγράφει σε βάση δεδομένων τις ενέργειες του μαθητή. Τέτοιες ενέργειες μπορεί να είναι η είσοδος σε μάθημα, η μελέτη ενός μαθήματος, οι προσπάθειες που έχει κάνει να ολοκληρώσει μία διδακτική ενότητα, οι βαθμολογίες-αποτελέσματα που πήρε, και πολλά άλλα.

Έτσι ο καθηγητής - υπεύθυνος του συστήματος τηλε-εκπαίδευσης μπορεί πολύ γρήγορα και με μεγάλη ευκολία να αποκτήσει υλικό που αφορά την συλλογή των δεδομένων του μαθητή από την χρήση του συστήματος, αλλά και υλικό για την ανάλυση των δεδομένων αυτών. Για παράδειγμα βαθμολογίες στα τεστ, % ποσοστό προσπαθειών, γενικά στατιστικά ανάμεσα σε ομάδες μαθητών, διαγράμματα ανάλυσης αποτελεσμάτων και άλλα. Μπορείτε να δείτε τέτοια παραδείγματα αναφορών που προσφέρει το σύστημα τηλεκπαίδευσης που δημιουργήθηκε στις παρακάτω εικόνες.



Εικόνα 68: Οθόνη βασικής αναφοράς στον καθηγητή για τα αποτελέσματα των απαντήσεων των μαθητών, με τον βαθμό που έλαβαν σε κάθε προσπάθεια.



Εικόνα 69: Οθόνη γραφικής αναφοράς στον καθηγητή για τα αποτελέσματα των απαντήσεων των μαθητών, με στατιστική ανάλυση των βαθμών ανά βαθμολογική κατηγορία.

ΤΗΛΕ-ΕΚΠΑΙΔΕΥΣΗ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ Έχετε εισέλθει ως george katsionis (Έξοδος)

Αρχή ▶ Σελίδες ιστοτόπου ▶ Scratch Μάθημα 1 ▶ Αναφορά

Πλοήγηση







- Αρχή
- Η αρχική μου
- Σελίδες ιστοτόπου
 - Συμμετέχοντες
 - Ιστολόγια ιστοτόπου
 - Site badges
 - Σημειώσεις
 - Ετικέτες
 - Ημερολόγιο
 - PSD Example
 - PSD Example 2
 - Scratch Μάθημα 1
- Το προφίλ μου
- Μαθήματα
- Χρήστες

Scratch Μάθημα 1

Info Reports

Basic report Graph report Interactions report Objectives report

3 results (3 users)

Όνομα / Επώνυμο	Διεύθυνση ηλεκτρονικού ταχυδρομίου	Προσπάθεια	Εικίνασε στις	Τελευταία πρόσβαση στις	Βαθμός	Scratch_Lesson_1	Res
 george katsionis	gkatsion@kman.gr	1	Σάββατο, 31 Οκτώβριος 2015, 6:06 μμ	Κυριακή, 1 Νοέμβριος 2015, 5:45 μμ	80	 80	und
 nikolas katsionis	gkatsion@cman.gr	1	Σάββατο, 31 Οκτώβριος 2015, 6:09 μμ	Κυριακή, 1 Νοέμβριος 2015, 11:01 μμ	80	 80	und
 fotis katsionis	gkatsion@fman.gr	1	Σάββατο, 31 Οκτώβριος 2015, 6:13 μμ	Σάββατο, 31 Οκτώβριος 2015, 6:38 μμ	100	 100	und

Select all / Deselect all

Σύμπτυξη όλων

Preferences just for this page

Show

- Summary of question
- Summary of responses
- Summary of right answer

Εικόνα 70: Οθόνη λεπτομερούς αναφοράς στον καθηγητή για τα αποτελέσματα των απαντήσεων των μαθητών, με τον βαθμό που έλαβαν σε κάθε προσπάθεια, αν πέρασαν το μάθημα, αλλά και ανάλυση της κάθε απάντησης τους.

Με την βοήθεια όλων αυτών των δεδομένων και των αναφορών που συλλέγονται για κάθε μαθητή ή/και ομάδα μαθητών μπορεί ο καθηγητής να βοηθηθεί σε διάφορες από τις εργασίες του και ταυτόχρονα να βοηθήσει και αυτός ποιο άμεσα τους μαθητές, προσαρμόζοντας τις συμβουλές του, τις παρατηρήσεις του, την διδασκαλία του μέσα στην τάξη, αλλά και το διαδικτυακό υλικό του στις ανάγκες του μαθητή ή/και της ομάδας μαθητών, βάσει όλων αυτών που παρατηρεί στο κανονικό μάθημα αλλά και αυτών που καταγράφονται στα διαδικτυακά μαθήματα.

Το συγκεκριμένο σύστημα τηλεκπαίδευσης για υποστήριξη της διδασκαλίας προγραμματιστικών μικρόκοσμων Scratch, **δοκιμάστηκε στην πράξη**, από την ίδια την γράφον αυτή την μεταπτυχιακή διατριβή στο σχολείο πρωτοβάθμιας εκπαίδευσης που διδάσκει ως καθηγήτρια Πληροφορικής.

Η δοκιμή του συστήματος έγινε στην Πέμπτη τάξη του δημοτικού, στο ένα τμήμα από τα δύο τμήματα του σχολείου, και για κάποια από τα πρώτα μαθήματα που παραδόθηκαν στην τάξη για τους προγραμματιστικούς μικρόκοσμους Scratch. Συμμετείχαν σε αυτό σχεδόν όλοι οι μαθητές, αφού προϋποθέτει την είσοδο των μαθητών στον ιστοτόπο του συστήματος τηλεκπαίδευσης, κάτι που πρέπει να γίνει σε τέτοιες ηλικίες υπό την επίβλεψη των γονιών τους, και φυσικά από το σπίτι τους.

Αυτή η **αρχική μελέτη και αξιολόγηση** του συστήματος εμφάνισε πολύ ενθαρρυντικά αποτελέσματα, για τους μαθητές και για το τμήμα ολόκληρο που χρησιμοποίησε υποστηρικτικά για το μάθημα την πλατφόρμα τηλεκπαίδευσης, σε σχέση με το άλλο τμήμα της Πέμπτης τάξης το οποίο έκανε μόνο την εκπαίδευση στην τάξη. Γενικά οι περισσότεροι από τους μαθητές που χρησιμοποίησαν το σύστημα μας δήλωσαν:

- Ήταν πολύ θετικό το ότι μπορούσαν να επαναλάβουν το μάθημα από το σπίτι τους, θεωρία, δραστηριότητες, ερωτήσεις.
- Το γεγονός ότι το περιβάλλον προγραμματιστικών μικρόκοσμων Scratch προσφέρει online υπηρεσίες, όπου μπορούσαν να υλοποιούν τις δραστηριότητες του συστήματος τηλεκπαίδευσης άμεσα, τους βοήθησε στην εμπέδωση της παρεχόμενης γνώσης.

- Πολύ καλά σχόλια έγιναν και για το γεγονός ότι το διδακτικό υλικό που δημιουργήθηκε προσφέρει υπηρεσία ανασκόπησης των απαντήσεων του μαθητή, όπου βλέπει τις σωστές απαντήσεις του, τις λάθος, αλλά και ποιο ήταν το σωστό εκεί που έκανε λάθος.

Από την πλευρά του διδάσκοντα τα πλεονεκτήματα χρήσης του περιβάλλοντος τηλεεκπαίδευσης στην υποστήριξη της διδασκαλίας του μαθήματος συνοψίζονται στα παρακάτω:

- Οι μαθητές που χρησιμοποίησαν το σύστημα ήταν πολύ πιο έτοιμοι για το επόμενο μάθημα / διδακτική ενότητα στο περιβάλλον προγραμματιστικών μικρόκοσμων.
- Οι απορίες που διατυπώθηκαν από τους μαθητές ήταν πιο συγκεκριμένες, και αφορούσαν πραγματικές δυσκολίες που συνάντησαν κατά την εφαρμογή των δραστηριοτήτων τους.
- Η συλλογή και ανάλυση των δεδομένων, ατομικά ανά μαθητή, έδωσε την ευκαιρία στον διδάσκοντα να βοηθήσει πιο στοχευμένα, συγκεκριμένους μαθητές.
- Οι απαντήσεις των μαθητών και ακόμη περισσότερο τα λάθη τους, όσα ήταν πολύ-εμφανιζόμενα, καθοδήγησαν τον διδάσκοντα σε αναδιατύπωση συγκεκριμένων κομματιών της θεωρίας ή/και των δραστηριοτήτων με στόχο την διευκόλυνση διοχέτευσης της παρεχόμενης γνώσης.

6. Περίληψη - Συμπεράσματα

Κατά την μελέτη της παραπάνω μεταπτυχιακής διατριβής έγινε μία αρχική αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

Αρχικά αναλύσαμε ότι λόγω των δυσκολιών που παρουσιάζουν οι εμπορικοί μεταγλωττιστές που απευθύνονται οι περισσότεροι σε επαγγελματίες και όχι σε μαθητές, αναπτύχθηκαν διάφορα ειδικά προγραμματιστικά εργαλεία για εκπαιδευτικούς σκοπούς εκμάθησης των βασικών αρχών προγραμματισμού, και ειδικότερα για μαθητές μικρότερων ηλικιών, τα οποία ονομάζονται **προγραμματιστικοί μικρόκοσμοι**.

Βασικό **πλεονέκτημα** των προγραμματιστικών μικρόκοσμων είναι ότι προκαλούν το ενδιαφέρον των μαθητών γιατί το περιβάλλον τους βασίζεται σε κάποιο φυσικό μοντέλο, που είναι ήδη γνωστό στους μαθητές, και η εμφάνιση τους είναι σαν παιχνίδι.

Όμως, η διδασκαλία των βασικών αρχών προγραμματισμού με την βοήθεια προγραμματιστικών μικρόκοσμων αντιμετωπίζει **δυσκολίες στην εφαρμογή** της, λόγω του διαφορετικού ρυθμού εκμάθησης των βασικών λειτουργιών του προγράμματος, της διαφορετικής ευχέρειας χρήσης του ηλεκτρονικού υπολογιστή και του προγραμματιστικού περιβάλλοντος του μικρόκοσμου, και των ωρών διδασκαλίας που είναι λίγες και σε αραιά διάστημα.

Λόγω αυτών των δυσκολιών έγινε κατανοητή η ανάγκη ύπαρξης κάποιων μαθημάτων από αυτά που περιλαμβάνει η διδασκαλία, σε μορφή ψηφιακή ως **ολοκληρωμένα ηλεκτρονικά διδακτικά μαθήματα**, τα οποία να είναι διαθέσιμα διαδικτυακά, για χρήση με ασύγχρονη εξ αποστάσεως εκπαίδευση ή αλλιώς **τηλε-εκπαίδευση**. Αυτά τα μαθήματα λειτουργούν υποστηρικτικά στην διδασκαλία του μαθήματος.

Με αυτό τον τρόπο οι μαθητές θα μπορούν να επαναλαμβάνουν κάποια από τα μαθήματα από το σπίτι τους σε χρόνο και με το ρυθμό που επιθυμούν οι ίδιοι, να παγιώνουν καλύτερα την γνώση τους και να δοκιμάζονται σε ερωτήσεις για αυτά. Ο καθηγητής από την πλευρά του θα μπορεί μέσω της πλατφόρμας τηλε-εκπαίδευσης να ενημερώνεται για το ποιοι μαθητές ασχολήθηκαν περαιτέρω με τα διδασκόμενα μαθήματα, καθώς και την επίδοσή τους στην κατανόηση αυτών, μέσω των **στοιχείων** που καταγράφονται από το σύστημα τηλεεκπαίδευσης.

Αυτά τα στοιχεία που καταγράφονται αποτελούν κομμάτια για το **μοντέλο του μαθητή-χρήστη**. Όσο περισσότερο χρησιμοποιεί ένας μαθητής το σύστημα τηλε-εκπαίδευσης, και όσα περισσότερα στοιχεία καταγράφονται για αυτόν. Η μοντελοποίηση του μαθητή-χρήστη του

συστήματος τηλε-εκπαίδευσης, είναι πολύ σημαντική για τον καθηγητή του μαθήματος, ο οποίος μπορεί να παρακολουθεί με αυτό τον τρόπο την πρόοδο του, και να τον καθοδηγήσει καλύτερα για τις επόμενες ενέργειες του.

Στη συνέχεια, έγινε μία ενδελεχής **ανασκόπηση** των περιβαλλόντων προγραμματιστικών μικρόκοσμων που υπάρχουν καθώς και των συστημάτων τηλεκπαίδευσης που υπάρχουν. Αυτή η ανασκόπηση ήταν ιδιαίτερα σημαντική για την **αιτιολόγηση της επιλογής** του περιβάλλοντος προγραμματιστικών μικρόκοσμων Scratch για την διδασκαλία των βασικών αρχών προγραμματισμού, αλλά και της πλατφόρμας τηλεκπαίδευσης Moodle για την υλοποίηση του συστήματος τηλεκπαίδευσης που παρέχει το διαδικτυακό διδακτικό υλικό που δημιουργήσαμε.

Ακολούθησε η παρουσίαση της **σχεδίασης** του διδακτικού υλικού με το εργαλείο Articulate StoryLine, και η παροχή-**υλοποίηση** του στο σύστημα τηλεκπαίδευσης σε πακέτα του διεθνούς προτύπου προδιαγραφών SCORM για την ανάπτυξη, τη συσκευασία και τη διανομή εκπαιδευτικού υλικού. Επίσης, πραγματοποιήθηκε μία αρχική **αξιολόγηση** της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, μέσω του συστήματος τηλε-εκπαίδευσης που δημιουργήθηκε, και παρουσιάστηκαν κάποια πρώτα συμπεράσματα τόσο από την πλευρά των μαθητών όσο και του καθηγητή.

Τελειώνοντας, πρέπει να αναφέρουμε ότι το σύστημα τηλεκπαίδευσης που δημιουργήθηκε χρειάζεται περαιτέρω αξιολόγηση και δοκιμή στην χρήση του, ώστε να μπορεί να προσφέρει όσο το δυνατό καλύτερα αποτελέσματα στην μαθησιακή διαδικασία. Η συνεχής αξιολόγηση του συστήματος μπορεί να επιφέρει αλλαγές στην σχεδίαση του διδακτικού υλικού, του τρόπου λειτουργίας του, αλλά και των δοκιμασιών που περιλαμβάνει.

Βιβλιογραφία

Alpert S. R., Singley M.K. & Fairweather P.G. (1999) "Deploying Intelligent Tutors on the Web: An Architecture and an Example", *International Journal of Artificial Intelligence in Education*, 10, pp. 183-197.

Articulate Global, Inc (2012).

Bergin, J., Stehlik, M., Roberts, J., Pattis, R.: *Karel++ - A Gentle Introduction to the Art of Object-Oriented Programming*, 2nd edition. Wiley, New York, (1997).

Blackboard Inc. All Rights Reserved. U.S.A, 1997-2015.

Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A. and Miller P. (1997), *Minilanguages: a way to learn programming principles*, *Journal of Education and Information Technologies*, 2, 65-83.

Buck, D. and Stucki, D.J. (2000), *JKarelRobot: A Case Study in Supporting Levels of Cognitive Development in the Computer Science Curriculum*, *ACM SIGCSE Bulletin*, 33(1), 16-20.

Carnegie Mellon University, Alice 2.x 1999-2015.

Comenius University (1994), *EasyLogo*, Dept of Informatics Education, Bratislava, 1994.

Dougiamas M. (2001), *Moodle*, Moodle Pty Ltd, Australia.

Eisenberg, M. (1995). *Creating Software Applications for Children: Some thoughts about Design*. In A. diSessa, C. Hoyles and R. Noss (Eds.), *Computers and Exploratory Learning (NATO ASI Series)*. Heidelberg: Springer-Verlag, 175-196.

Freund, S.N., & Roberts, E.S. (1996). *THETIS: An ANSI C programming environment designed for introductory use*. Philadelphia, PA, USA: ACM. pp. 300–304 (SIGCSE '96 2/96)

Gamper, J. & Knapp, J. (2002): "A Web-Based Language Learning System". In J. Fong et al. (Eds.): *Proceedings of the 1st International Conference on Web-based Learning ICWL 2002*, *Lecture Notes in Computer Science*, Springer-Verlag Berlin Heidelberg 2002, pp. 106-118.

Greek Academic network (GUnet), Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Open eClass platform, 2000, Athens, Greece.

Harvey Brian, *Computer Science Logo Style*, Mit Press, 1997.

Hoyles, C. and Noss, R. (1992) *Deconstructing Microworlds. "Advanced Technologies in the teaching of mathematics and science"* Ferguson D.L. (ed), Springer-Verlag.

HyperStudio 3.0.6 [Computer software] (1995). El Cajon: Roger Wagner

Kabassi K. & Virvou M., 2006, "Multi-Attribute Utility Theory and Adaptive Techniques for Intelligent Web-Based Educational Software", In INSTRUCTIONAL SCIENCE, Vol. 34, pp. 131-158, Springer Netherlands, 2006.

Katsionis G. and Virvou M., 2008, "Personalised e-learning through an Educational Virtual Reality Game using Web Services", In international journal MULTIMEDIA TOOLS AND APPLICATIONS, Volume 39, Number 1 / August, 2008, pp. 47-71, Springer.

Keegan, D. (2000). Οι Βασικές Αρχές της Ανοιχτής και εξ Αποστάσεως Εκπαίδευσης. (Α. Μελίστα, Μετάφρ.). Αθήνα: Εκδόσεις Μεταίχμιο.

LCSI - Logo Computer Systems Inc., 1999, Microworlds Pro.

Lopez J.M., E. Millan, J.L. Perez-de-la-Cruz, F. Triguero, "ILESA: a Web-based Intelligent Learning Environment for the Simplex Algorithm", in Alvegard C. (ed.): Proceedings of CALISCE'98, 4th International Conference on Computer Aided Learning and Instruction in Science and Engineering, 1998, pp. 399-406.

Microsoft Research, Kodu Game Lab, Initial release June 30, 2009.

MIT Media Lab, StarLogo TNG, Mitchel Resnick, Eric Klopfer, and others at MIT Scheller Teacher Education Program in Massachusetts, 2008.

MIT Media Lab Lifelong Kindergarten Group, Scratch visual programming language, First appeared 2002 (latest version Scratch 2, 2013).

Naidu, S. (2003). E-Learning: A guidebook of principles, procedures and practices. CEMCA. Available on line (2010) στο: http://www.cemca.org/e-learning_guidebook.pdf.

Nam, C. S., and Smith-Jackson, T.L. (2007). "Web-Based Learning Environment: A Theory-Based Design Process for Development and Evaluation," Journal of Information Technology Education, Vol. 6, pp. 23-43

Open Source Matters Inc, Joomla Project Team, Joomla Content Management System, 2005.

Overmars Mark, GameMaker: Studio, YoYo Games, 1999.

Pane, J.F.& Myers, B.A. (1996). Usability issues in the design of novice programming systems. School of Computer Science Technical Report CMU-CS-96-132. Pittsburgh, PA: Carnegie Mellon University

Papert S. (1980), Mindstorms: Children, Computers and Powerful Ideas, Basic Books, Inc., Publishers/ New York.

Papert S., Harel I. (1991), Situating constructionism, In Constructionism: Research reports and essays.

Pattis, R. E., Roberts, J. & Stehlik, M. (1995), Karel - The Robot, A Gentle Introduction to the Art of Programming, 2nd edition, Wiley, New York.

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.

Ruckert, M. & Halpern, R. (1993), Educational C, ACM, SIGSCE Bulletin, σελ. 6-9.

Rustici Software, SCORM Version 1.0, January 2000.

Rustici Software, SCORM Version 1.2, October 2001.

Sanders, D. and Dorn, B. (2003), Jerroo: A Tool for Introducing Object-Oriented Programming, ACM SIGCSE Bulletin, 35 (1), 201-204.

Silverman B., Turtle Art, 2004.

Stagecast Software (1997): Allen Cypher, David Canfield Smith.

UCL, University Catholique de Louvain, Claroline - Learning Management System Belgium, 2000.

University of Cologne, ILIAS learning management system, Cologne, 1997.

Virvou M. & Alepis E. (2003), Creating tutoring characters through a Web-based authoring tool for educational software. In Proceedings of the 2003 IEEE International Conference on Systems, Man and Cybernetics.

Virvou M. & Katsionis G, (2003) 'VIRGE: Tutoring English over the Web through a Game', Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2003), Greece, Athens, pp. 469.

Virvou, M., & Katsionis G, (2008) 'On the usability and likeability of virtual reality games for education: The case of VR-ENGAGE'. In the international journal of Computers & Education, An International Journal, 2008, Volume 50, Issue 1, Pages 154-178, January 2008.

Virvou, M., Katsionis, G. & Manos, C. (2005): "Combining Software Games with Education: Evaluation of its Educational Effectiveness", Educational Technology & Society, Journal of International Forum of Educational Technology & Society and **IEEE** Learning Technology Task Force, April 2005, Vol. 8, No 2, pp 54-65.

Virvou M., Maras D. & Tsiriga V., 2000, "Evaluation of an ITS for the Passive Voice of the English Language Using the CIAO!, Framework" Proceedings of ED-MEDIA 2000, World Conferences on Educational Multimedia, Hypermedia and Educational Telecommunications, Montreal (Canada)

Virvou M., Maras D. 1999, "An Intelligent Multimedia Tutor for English as a Second Language", Proceedings of ED-MEDIA/ED-TELECOM 99, World Conference on Educational Multimedia and Educational Telecommunications, Seattle (U.S.A.), Vol. 2, 928 - 932.

Virvou, M. & Tsiriga, V. (2001): "Web Passive Voice Tutor: an Intelligent Computer Assisted Language Learning System over the WWW" In IEEE International Conference on Advanced Learning Technologies 2001, Wisconsin, USA, pp.131-134.

Xinogalos S, Satratzemi M, Dagdilelis V: An introduction to object-oriented programming with a didactic microworld: objectKarel, Computers & Education 47 (2), 148-171, 2006.

Ziegler U. & Crews T. (1999) An Integrated Program Development Tool for Teaching and Learning How to Program, ACM SIGSCE Bulletin, Vol. 31, No. 1, pp. 276-280.

Αβάκιο / E-Slate, Ομάδα Μαθησιακής Τεχνολογίας του Ερευνητικού Ακαδημαϊκού Ινστιτούτου Τεχνολογίας Υπολογιστών (ITY), 2001.

Βερναδάκης Ν, Γιαννούση Μ., Αντωνίου Π., Κέλλης Η. και Κιουμουρτζόγλου Ε., (2007), Η επίδραση ενός συστήματος διαχείρισης μαθημάτων ασύγχρονης εκπαίδευσης για την υποστήριξη της παραδοσιακής διδασκαλίας στην Τριτοβάθμια Εκπαίδευση, Πρακτικά του 4ου Γιεθνές Συνεδρίου για την Ανοικτή και ΕξΑποστάσεως Εκπαίδευση, Τόμος Α, σελ. 309-318.

Κυνηγός, Χ. (2006). Σενάρια: Σχέδια μαθησιακών δραστηριοτήτων με ψηφιακές τεχνολογίες. Το Μάθημα της Διερεύνησης. Παιδαγωγική αξιοποίηση των ψηφιακών τεχνολογιών για τη διδακτική των μαθηματικών. Από την έρευνα στη σχολική τάξη. Αθήνα, Ελληνικά Γράμματα, σελ181-194.

Ματραλής, Χ.(1999). στο: Κόκκος Α., Λιοναράκης Α., Ματραλής Χ. (επιμ.), Ανοικτή και εξ αποστάσεως εκπαίδευση. Θεσμοί και Λειτουργίες, τόμος Α', Πάτρα: Ε.Α.Π.

Ξυνόγαλος, Σ. (2006), Η διδασκαλία του αντικειμενοστρεφούς προγραμματισμού με τον μικρόκοσμο objectKarel: εμπειρίες, προβληματισμοί και προτάσεις, Περιοδικό «ΘΕΜΑΤΑ στην Εκπαίδευση» ειδικό αφιέρωμα: «Σύγχρονη έρευνα στη Διδακτική της Πληροφορικής: ερευνητικοί άξονες, μέθοδοι, τεχνικές, εργαλεία», 7:3, 305-327, Ελληνικά γράμματα.

Πανεπιστήμιο Θεσσαλίας (2010), Δημιουργώ παιχνίδια στο Scratch, Διδακτική της Πληροφορικής ΙΙ, εαρινό εξάμηνο του ακαδημαϊκού έτους 2009-2010, τμήμα Μηχανικών Η/Υ - Δικτύων και Τηλεπικοινωνιών, 2010.

Σοφός, Α. Κρον, F. (2010). Αποδοτική Διδασκαλία με τη Χρήση Μέσων. Από τα προσωπικά και πρωτογενή στα τεταρτογενή και ψηφιακά Μέσα.

Υπουργείο Παιδείας και Θρησκευμάτων (2013), Επιμορφωτικό Υλικό για τον κλάδο ΠΕ 19-20, Κέντρα Στήριξης Επιμόρφωσης, ΤΕΥΧΟΣ 6Α, ΘΕΩΡΙΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ, Ειδικό Μέρος – Θεωρία Σεναρίων, Μάρτιος 2013.

Υπουργείο Παιδείας και Θρησκευμάτων, «ΝΕΟ ΣΧΟΛΕΙΟ (Σχολείο 21^{ου} αιώνα) - Νέο Πρόγραμμα Σπουδών, Οριζόντια Πράξη» MIS: 295450, Πρόγραμμα Σπουδών για τις ΤΠΕ στην Πρωτοβάθμια Εκπαίδευση, ΕΣΠΑ 2007-13/Ε.Π. Ε&ΔΒΜ/Άξονες Προτεραιότητας 1-2-3, Αθήνα, 2014.

Φεσάκης Γ., Δημητρακοπούλου Α., Σεραφείμ Κ., Ζαφειροπούλου Α, Ντούνη Μ., Τούκα Β., Οδηγός του SCRATCH, Γνωριμία με το εκπαιδευτικό περιβάλλον του Scratch (Μετάφραση από το πρωτότυπο: Reference Guide), 4ο Πανελλήνιο Συνέδριο Διδακτική της Πληροφορικής, Πάτρα 28-30 Μαρτίου 2008, Πανεπιστήμιο Πατρών & ΕΤΠΕ.

Χαραλαμπίδης Σ., Προσαρμογή του σχολικού βιβλίου "Πληροφορική Γ Γυμνασίου" στο περιβάλλον προγραμματισμού Scratch, Ζάκυνθος, Μάιος 2010.

Αξιολόγηση της υποστήριξης μαθημάτων προγραμματισμού Scratch για μαθητές μικρής ηλικίας, με μία πλατφόρμα τηλε-εκπαίδευσης και μοντελοποίησης χρηστών μέσω Moodle.