ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Προγραμματιστικοί Μικρόκοσμοι και η Εφαρμογή τους στη Διδασκαλία του Προγραμματισμού

ΜΑΡΙΑ ΔΡΟΣΙΝΟΥ
Α.Μ.:36961

Επιβλέπων καθηγητής:
Ιωάννης Ψαρομήλιγκος

ΑΘΗΝΑ, 2013

Περίληψη

Η παρούσα πτυχιακή άσκηση αποτελεί μελέτη των Προγραμματιστικών Μικρόκοσμων σε θεωρητικό αλλά και πρακτικό επίπεδο. Αναλύεται ο ρόλος της Πληροφορικής στην εκπαίδευση και δίνεται έμφαση στις δυσκολίες που αντιμετωπίζουν οι μαθητές στην εκμάθηση του προγραμματισμού. Στη συνέχεια, περιγράφονται προγραμματιστικές γλώσσες που βοηθούν στην διδακτική προσέγγιση των προγραμματιστικών εννοιών. Για την κατανόηση αυτών των εννοιών από μαθητές της δευτεροβάθμιας εκπαίδευσης, αναπτύχθηκαν απλά παραδείγματα βασισμένα στα κεφάλαια 1 και 2 (Εισαγωγή στην έννοια του Αλγόριθμου και στον Προγραμματισμό και Ο Προγραμματισμός στην Πράξη) της πρώτης ενότητας του βιβλίου Πληροφορικής της Γ' Γυμνασίου. Αναφέρονται στις ιδιότητες του αλγορίθμου, περιλαμβάνουν θεωρητικές ερωτήσεις, δείχνουν τον ρόλο του πίνακα και καθιστούν κατανοητή την έννοια της μεταβλητής. Γενικά όλα τα παραδείγματα στηρίζονται κυρίως στην δομή επιλογής και επανάληψης. Για τα παραδείγματα χρησιμοποιήθηκε η εκπαιδευτική γλώσσα Scratch και, το εξελιγμένο παρακλάδι της, BYOB.

Abstract

This thesis is a study of programming environments in both theoretical and practical level. Special focus is given on the role of IT in education and also on the difficulties that students face in learning programming. Furthermore, there is the description of programming languages that assist in teaching approaches in programming concepts. To make these concepts understood by students in secondary education, simple examples were developed based on chapters 1 and 2 (Introduction to the concept of the Algorithm and the Programming and Programming in the Practice) of the first section of the book Information Technology of Third Grade. These referred to the properties of the algorithm, including theoretical questions, show the role of the table and make the meaning of the variable understood. Generally, all examples are based mainly on selection of the structure and repetition. For the practical part, the programming, language learning environment Scratch is being used, with its advanced offshoot, BYOB.

Ευχαριστίες

Θα ήθελα να εκφράσω θερμές ευχαριστίες στον εποπτεύοντα καθηγητή μου Ιωάννη Ψαρομήλιγκο, ο οποίος γνωρίζοντας το ενδιαφέρον μου για τους Προγραμματιστικούς Μικρόκοσμους μου έδωσε την κατάλληλη εργασία. Κάτω από την καθοδήγησή του και δίνοντας μου τα ακατάλληλα εφόδια πέτυχα την ολοκλήρωσή της.

Ευχαριστώ τους φίλους μου που την διάβασαν και μετά από συζητήσεις με βοήθησαν να φτάσω στο τελικό αποτέλεσμα, καθώς και την αδερφή μου που τέσταρε τα παραδείγματα.

Τέλος, ευχαριστώ τους γονείς μου που πρόθυμα με στήριξαν με τα εύστοχα σχολιά τους, την κριτική και την παρότρυνση τους.

ΠΕΡΙΕΧΟΜΕΝΑ

Κεφάλαιο 1

Νέες Τεχνολογίες στην Εκπαίδευση

1.1 Εισαγωγή

Η πλειονότητα των ανθρώπων, μέχρι τα μέσα του 19ου αιώνα, διέθετε πάρα πολύ χρόνο για την καλλιέργεια και συλλογή των καρπών. Οι γεωργικές δραστηριότητες ήταν ένας τρόπος ζωής.

Η βιομηχανική επανάσταση στα μέσα του 19ου αιώνα επέφερε σημαντικές αλλαγές. Για πρώτη φορά στην ιστορία άρχισαν να δουλεύουν στις βιομηχανίες περισσότεροι άνθρωποι από ότι στους αγρούς. Η εποχή της βιομηχανίας είχε αρχίσει.

Σήμερα ζούμε στην εποχή της πληροφορίας. Καθώς τα εργοστάσια γίνονται περισσότερο αυτοματοποιημένα, εργάζονται λιγότεροι άνθρωποι για την παραγωγή αγαθών. Η ίδια χρονική περίοδος χαρακτηρίζεται από την έκρηξη της πληροφορίας. Συνώνυμο της πληροφορίας είναι ο όρος δεδομένα (data). Νέα συστήματα επικοινωνίας μας επιτρέπουν να αξιοποιούμε τα δεδομένα (πληροφορίες) με σχετική ευκολία. Για το λόγο αυτό πολλοί άνθρωποι αναφέρονται στην εποχή μας χαρακτηρίζοντάς την ως εποχή της πληροφορίας.

Ο Η/Υ έχει εισχωρήσει σε όλους τους τομείς της επιστήμης και κάθε άλλης παραγωγικής δραστηριότητας συμβάλλοντας έτσι, με έμμεσο και άμεσο τρόπο, στην ίδια τη ραγδαία εξέλιξή τους. Είναι προφανές πως οι κοινωνικές επιπτώσεις από τη νέα αυτή παραγωγική δύναμη είναι σημαντικές, σύνθετες και, ως ένα σημείο, απρόβλεπτες. Επηρεάζουν άμεσα την ποιότητα της ζωής μας, ακόμη και τη διαμόρφωση του χαρακτήρα μας και τις κοινωνικές μας σχέσεις. Είναι ένα πεδίο όπου χάνονται και κερδίζονται οικονομικοί και πολιτικοί πόλεμοι, όμως και ένα μέσο που ανοίγει νέους ορίζοντες επικοινωνίας.

Ο κίνδυνος είναι μεγάλος και ακουμπά κυρίως τις ευάλωτες κοινωνικές ομάδες, μια από τις οποίες είναι και τα παιδιά. Το καλύτερο όπλο απέναντι στο μύθο του Η/Υ αλλά και το καλύτερο εφόδιο απέναντι στη σημερινή κοινωνία δεν είναι άλλο από την σωστή εκπαίδευση.

Από την πληθώρα των μέσων που προσφέρει σήμερα η εκπαιδευτική τεχνολογία, σημαντική θέση κατέχει ο ηλεκτρονικός υπολογιστής, του οποίου η εισαγωγή και χρήση στην εκπαίδευση, όσον αφορά τη διδακτική υποστήριξη διαφόρων αντικειμένων, είναι πια γεγονός. Η εκπαίδευση δε θα μπορούσε και ούτε

πρέπει να μείνει απαθής και αποστασιοποιημένη από αυτή τη νέα πραγματικότητα για δυο κυρίως λόγους. Πρώτο, επειδή ο χαρακτήρας της εκπαίδευσης πρέπει να αναπροσαρμόζεται στις εκάστοτε απαιτήσεις της κοινωνίας και δεύτερο, επειδή μπορεί κάλλιστα η εκπαίδευση να χρησιμοποιήσει, εντάσσοντας στους μηχανισμούς της, τον ίδιο τον υπολογιστή είτε ως εργαλείο διδασκαλίας, είτε ως επικοινωνιακό μέσο. Επομένως, πιστεύουμε ότι ο χώρος προσφέρεται για δημιουργική ενασχόληση και ειδικότερα για την διερεύνηση των δυνατοτήτων συμβολής της πληροφορικής στην εποπτικοποίηση της διδακτικής πράξης, γεγονός που αποτελεί συστηματική επιδίωξη της σύγχρονης διδακτικής.

Τέλος, η εισαγωγή και ενσωμάτωση της πληροφορικής τεχνολογίας στην εκπαίδευση αποτελεί προτεραιότητα κάθε σύγχρονης κοινωνίας και διασφαλίζει την ισότιμη συμμετοχή των πολιτών στην ανταγωνιστική Κοινωνία της Πληροφορίας.

1.2 Η Πληροφορική στην Εκπαίδευση

Στη σύγχρονη εποχή η πληροφορική έχει εισχωρήσει σε όλους τους τομείς της επιστήμης και κάθε άλλης παραγωγικής δραστηριότητας με αποτέλεσμα να μην μένει ανεπηρέαστη και η εκπαίδευση. Η πλειονότητα των εκπαιδευτικών συστημάτων έχει επηρεαστεί βαθύτατα από τις ραγδαίες τεχνολογικές εξελίξεις, οι οποίες καθιστούν την εισαγωγή των ΤΠΕ απαραίτητη σε όλες τις βαθμίδες και μας οδηγούν στο να κάνουμε λόγο για εκπαίδευση βασισμένη στον υπολογιστή και για την πληροφορική ως αντικείμενο μάθησης.

Η ένταξη των ΤΠΕ στην εκπαίδευση ακολούθησε μια εξελικτική πορεία ώσπου να αποκτήσει τη σημερινή της μορφή. Τα στάδια που ακολούθησε είναι γνωστά ως χρονολογικές φάσεις ένταξης. Αναλυτικότερα, μέχρι το 1970 κυριαρχούσε η εκπαιδευτική τεχνολογία και οι διδακτικές μηχανές. Την περίοδο 1970 - 1980 οι ΤΠΕ αντιμετωπίζονται ως γνωστικό αντικείμενο το οποίο θα μπορούσε να διδαχθεί σε όλες τις βαθμίδες της εκπαίδευσης. Η προσέγγιση αυτή είναι γνωστή ως τεχνοκρατική. Τη δεκαετία του 80' η πληροφορική εκλαμβάνεται ως ένα μέσο διαθεματικής προσέγγισης όλων των μαθημάτων. Μέσω των ΤΠΕ οι μαθητές μυούνται στην έρευνα, στην απόκτηση γνώσης και στη μάθηση (ολοκληρωμένη προσέγγιση). Η τελευταία περίοδος ξεκινά από το 1990 και συνεχίζει ακόμα και στις

μέρες μας. Η πληροφορική και οι ΤΠΕ αποτελούν κοινωνικό φαινόμενο καθώς, επίσης, και σημαντικό στοιχείο της εκπαίδευσης και της γενικότερης κουλτούρας. Το εκπαιδευτικό σύστημα είναι αναγκασμένο να ανταποκριθεί ανάλογα στις σημαντικές αυτές τεχνολογικές αλλαγές, ώστε να αντεπεξέλθει στις σύγχρονες απαιτήσεις μόρφωσης και κατάρτισης και στις ραγδαίες εξελίξεις της αγοράς εργασίας.

Τείνει να γίνει ευρύτερα αποδεκτό ότι «η διδασκαλία και η χρήση των νέων τεχνολογιών στην εκπαίδευση έχουν ως κύριο στόχο να διευρυνθούν οι γνωστικές ικανότητες των παιδιών και να εμπλουτιστούν οι ευκαιρίες και οι τρόποι διδασκαλίας και μάθησης». Πιο συγκεκριμένα , ο χώρος των Νέων Τεχνολογιών είναι μια ευκαιρία για τη δημιουργία ενός περιβάλλοντος που, πέρα από την παιδαγωγική του διάσταση, παρέχει δυνατότητες επανεξέτασης και επαναθεώρησης γενικών και ειδικών προβλημάτων στους χώρους μάθησης και εκπαίδευσης όπως προβλήματα παιδαγωγικών θεωριών, διδακτικού σχεδιασμού, εξατομικευμένης διδασκαλίας και αλληλεπίδρασης ανάμεσα στους συμμετέχοντες στην εκπαιδευτική διαδικασία. Οι νέες τεχνολογίες με ένα σχετικά χαμηλό κόστος επιτρέπουν στα παιδιά απομακρυσμένων σχολείων, να έχουν πρόσβαση σε βιβλιοθήκες όλου του κόσμου, καθώς επίσης και τη δυνατότητα της επικοινωνίας με άλλα παιδιά αλλά και με ειδικούς για κάποιο θέμα που τα ενδιαφέρει. Για τα παιδιά αυτά οι νέες τεχνολογίες αποτελούν ζωτικής σημασίας εργαλεία.

Είναι βέβαια γνωστό πως οι υπολογιστές δεν είναι μαγικά εργαλεία και ότι από μόνοι τους δεν μπορούν να αλλάξουν τη μαθησιακή διαδικασία Ερευνητικά δεδομένα τονίζουν τον ουσιαστικό ρόλο που μπορεί να διαδραματίσει η εκπαιδευτική τεχνολογία στη δημιουργία υποστηρικτικού μαθησιακού περιβάλλοντος, υπό το πρίσμα της Γνωστικής Επιστήμης. Διαφαίνεται λοιπόν καθαρά η αναγκαιότητα μιας διεπιστημονικής συνεργασίας για τη μελέτη του τρόπου ενσωμάτωσης των υπολογιστών στο αναλυτικό πρόγραμμα του σχολείου, έτσι ώστε να μπορέσουν οι τελευταίοι να μην αφομοιωθούν από τις τρέχουσες εκπαιδευτικές πρακτικές, αλλά να αποτελέσουν την ναυαρχίδα ουσιαστικών αλλαγών στη μαθησιακή διαδικασία.

Η πληροφορική στην εκπαίδευση είναι μια αναμφισβήτητη πραγματικότητα, και με αυτό δεν εννοούμε εισαγωγή απλώς ενός νέου εργαλείου αλλά την ανάπτυξη μιας νέας διάστασης της εκπαιδευτικής τεχνολογίας. Με την αλληλεπίδραση γονέων,

εκπαιδευτικών, μαθητών, βιβλίων, κοινωνικών ομάδων και του παιδαγωγικού πλαισίου, το εκπαιδευτικό σύστημα θα αλλάξει στις επόμενες δεκαετίες και ο πυρήνας αυτής της αλλαγής θα είναι ο υπολογιστής.

Στις περισσότερες χώρες η ένταξη των Νέων Τεχνολογιών στην Πρωτοβάθμια Εκπαίδευση αντιμετωπίζεται όχι ως αυτόνομο γνωστικό αντικείμενο αλλά ως εργαλείο διαθεματικών δραστηριοτήτων. Δίνεται λοιπόν έμφαση στην αναζήτηση πληροφοριών, στην επικοινωνία, και στην χρήση εκπαιδευτικών λογισμικών.

Επιπλέον σύμφωνα με έρευνα της Eurydice σχετικά με την χρήση των ΤΠΕ στην εκπαίδευση, οι Τεχνολογίες των Πληροφοριών και της Επικοινωνίας αποτελούν μέρος του προγράμματος σπουδών των μαθητών σχεδόν παντού στην Ευρώπη. Στην Εκπαίδευση, οι ΤΠΕ χρησιμοποιούνται κυρίως ως εργαλείο για τη διδασκαλία άλλων μαθημάτων και οι προβλεπόμενες δραστηριότητες περιλαμβάνουν τη χρήση λογισμικού, την αναζήτηση πληροφοριών και τα δίκτυα επικοινωνιών για την επέκταση της γνώσης σε διάφορα μαθήματα Σε πολλές χώρες, ο χρόνος που αφιερώνεται για τις ΤΠΕ είναι ελαστικός. Οι επίσημες κατευθυντήριες γραμμές σχετικά με τις προσεγγίσεις που υιοθετούνται μοιάζουν αρκετά σε όλες τις χώρες.

Η ελληνική εκπαίδευση θα πρέπει να ανταποκριθεί στην τεχνολογία και αυτή η ανταπόκριση δεν συνίσταται απλώς στο να διδάξει προγραμματισμό στα νεαρά άτομα και να αναπτύξει δεξιότητες στο πληκτρολόγιο και στα ηλεκτρονικά κυκλώματα (κάτι που, όπως φαίνεται, υιοθετεί και αποκλειστικά επιδιώκει ως σήμερα το υπουργείο Παιδείας), αλλά πρέπει να περιέχει προσπάθειες καθοδήγησης της τεχνολογίας με εγγυητές τις δημοκρατικές αξίες και την απελευθέρωση του ανθρώπου. Δεν θα πρέπει να επιτρέψουμε στην τεχνολογία να αλλάξει μηχανικά την εκπαίδευση. Οποιαδήποτε αλλαγή ή μεταμόρφωση της εκπαίδευσης που θα προξενηθεί θα πρέπει να ελέγχεται από τους εκπαιδευτικούς και τους ίδιους τους μαθητές και όχι από τις μηχανές.

Γενικότερα θα λέγαμε ότι επικρατεί μια μεγάλη σύγχυση στα θέματα πληροφορικής στην εκπαίδευση. Αυτό βασικά οφείλεται :

- Στην πολυμορφία και πολυσημία της επιστήμης της πληροφορικής, καθώς και στη συνεχή ραγδαία εξέλιξη της ίδιας της τεχνολογίας. Κάτι που σήμερα είναι νέο σε ένα χρόνο μπορεί να θεωρηθεί ξεπερασμένο. Ακόμη και ο πιο «σοφός» ειδικός δεν

μπορεί να γνωρίσει όλα τα εργαλεία της επιστήμης του. Συχνά οι απόψεις των «εμπειρογνωμόνων» διαφέρουν τόσο πολύ και παρουσιάζουν τέτοια απροβλεψιμότητα, που θα έλεγε κανείς ότι δεν αναφέρονται σε θετικές αλλά σε κοινωνικές επιστήμες.

- Στην έλλειψη γνώσης – πέρα μιας αόριστης υποψίας – σχετικά με το πώς οι τεχνολογικές επιλογές συμμετέχουν σε οργανωτικές, οικονομικές και πολιτικές επιλογές και, γενικότερα στην απουσία μιας ολοκληρωμένης μελέτης του ρόλου της σύγχρονης τεχνολογίας ως κοινωνικού καταλύτη.
- Σε εμπόδια που οφείλονται στις μη δεκτικές στάσεις του τεχνολογικά μη καταρτισμένου ή – όπως το αποκαλούν ορισμένοι – τεχνολογικά αναλφάβητου κοινού, που πιστεύει ότι αυτά τα ζητήματα είναι απρόσιτα και τα αφήνει στην αυθεντία των ειδικών.

Η εισαγωγή της πληροφορικής στην Εκπαίδευση συνοδεύεται από μια σειρά προβλημάτων τα οποία αφορούν τόσο τα επιμέρους προβλήματα της εκπαίδευσης και των Νέων Τεχνολογιών όσο και τα προβλήματα τα οποία προκύπτουν από την αλληλεπίδραση των δύο αυτών χώρων.

Η συνθετότητα και το εύρος των προβλημάτων αυτών διαφαίνεται στους λόγους εισαγωγής της πληροφορικής στην Εκπαίδευση, οι οποίοι θα μπορούσαν να συνοψιστούν στις παρακάτω κατηγορίες λογικών βάσεων:

- Λόγοι κοινωνικής φύσεως, με βασικό επιχείρημα την αναγκαιότητα της προετοιμασίας των παιδιών, ώστε να λειτουργήσουν αυτά κατάλληλα σαν πολίτες μιας κοινωνίας πλαισιωμένης από τις Νέες Τεχνολογίες
- Λόγοι επαγγελματικής φύσεως, με επιχείρημα την αναγκαιότητα της προετοιμασίας των παιδιών, ώστε να λειτουργήσουν αυτά κατάλληλα ως επαγγελματίες εργαζόμενοι σε μια τεχνολογική κοινωνία
- Η πορεία των Η/Υ στην Εκπαίδευση ακολούθησε έναν δρόμο κοινό με αυτόν άλλων καινοτομιών, όπως π.χ. κατά την εισαγωγή των οπτικοακουστικών μέσων διδασκαλίας, η εμφάνιση των οποίων δημιούργησε την πεποίθηση για μια επανάσταση στο χώρο της εκπαίδευσης
- Χαρακτηριστικό σημείο ομοιότητας ήταν ο ενθουσιασμός, ο οποίος μονόδρομα πάντα οδηγούσε στο να αποτελεί η χρήση των «μέσων» αυτοσκοπό και

να παραμερίζεται το γεγονός ότι αυτές αποτελούν συστατικό μόνο μέρος της εκπαιδευτικής διαδικασίας και όχι το σύνολο.

Κοινό σημείο της πορείας αυτής υπήρξε η προσπάθεια ταξινόμησης της χρήσης των Η/Υ, όπως εξάλλου και των οπτικοακουστικών μέσων, ως προς την καταλληλότητά τους για κάθε γνωστικό αντικείμενο ή διδακτική ενότητα, με κριτήρια τόσο στεγανά διαχωρισμένα που οπωσδήποτε δεν έχουν σχέση με αυτό που θα θέλαμε να εννοούμε σαν εκπαιδευτική πράξη, αλλά σαν μια προσπάθεια ανακάλυψης συνταγών για τη λύση των προβλημάτων.

Οι διάφοροι παράγοντες που υπεισέρχονται στην χρήση των Νέων Τεχνολογιών στην εκπαίδευση, φαίνεται ότι σχετίζονται με το μηχανικό μέρος του εξοπλισμού (hardware), το λογισμικό υλικό (software) και την εκπαίδευση των εκπαιδευτικών.

Η ελληνική πραγματικότητα, παρά τα όποια προβλήματα που δημιουργεί η έλλειψη δυνατότητας εξοπλισμού όλων των σχολείων με μηχανήματα – λόγω μη ύπαρξης εθνικής βιομηχανίας παραγωγής εξοπλισμού – απέφυγε ως ένα μεγάλο βαθμό το οξύ πρόβλημα των μηχανημάτων που δε χρησιμοποιούνται.

1.2.1 Η Πληροφορική ως Αντικείμενο και ως Εργαλείο Μάθησης

Η χρήση των Η/Υ ανοίγει νέους ορίζοντες στην εκπαιδευτική διαδικασία με αποτέλεσμα να δημιουργείται η ανάγκη για την μελέτη της ίδιας της επιστήμης αλλά και η χρησιμοποίηση της για την μελέτη άλλων μαθημάτων. Έτσι αναπτύχθηκαν δυο διαφορετικές προσεγγίσεις ως αναφορά την Πληροφορική στην εκπαίδευση.

- *Η Πληροφορική ως εργαλείο μάθησης - Εκπαίδευση βασισμένη στον υπολογιστή*

Η εκπαίδευση ή εξάσκηση βασισμένη στον υπολογιστή (Computer Based Training ή CBT) ορίζεται ως μια διαλογική αυτοεκμάθηση με ασκήσεις που "τρέχουν" σε έναν υπολογιστή. Έτσι, ο μαθητής μπορεί να μάθει μια γλώσσα, να μάθει γεωγραφία, να εξασκεί στα μαθηματικά κλπ. ενώ οι ενήλικες μπορούν να χρησιμοποιήσουν τέτοιου είδους εφαρμογές για να μάθουν πληκτρολόγηση, χρήση

εφαρμογών, να μιλούν ξένες γλώσσες κλπ. Η εκπαίδευση βασισμένη στον υπολογιστή αναφέρεται σε οποιοδήποτε μάθημα κι έχει ως κύρια μέθοδο μετάδοσης της γνώσης τον ηλεκτρονικό υπολογιστή.

- *Η πληροφορική ως γνωστικό αντικείμενο*

Η προσέγγιση αυτή η έχει σαν στόχο την απόκτηση γνώσεων για την λειτουργία των ηλεκτρονικών υπολογιστών, την εισαγωγή στον προγραμματισμό και γενικότερα τις λειτουργίες της λογικής διαδικασίας. Η Πληροφορική αντιμετωπίζεται ως σύνθεση τριών βασικών επιστημονικών όρων. Της θεωρίας, των πειραματικών επιστημών και της τεχνολογίας. Η θεωρία περιλαμβάνει το πως λειτουργεί ο ηλεκτρονικός υπολογιστής, οι πειραματικές επιστήμες, θέματα που αφορούν την ανάπτυξη των υλικών ενώ η τεχνολογία αφορά τις επιλογές εκείνες που σε συνδυασμό με τα παραπάνω οδηγούν στην επίλυση καθημερινών προβλημάτων και αναγκών.

1.3 Ο Ρόλος του Η/Υ στο Σχολείο

Πριν από μερικά χρόνια η χρήση του υπολογιστή ως εκπαιδευτικού εργαλείου ήταν πολύ μικρότερη απ' ότι σήμερα. Στη διάρκεια της βιομηχανικής επανάστασης οι παιδαγωγοί μιλούσαν για μια εκπαίδευση με τα τρία R (Reading, wRiting, aRithmetic). Σήμερα οι παιδαγωγοί μιλούν για μία εκπαίδευση με τα τρία C (Children, Computer, Communication), για μια εκπαίδευση με τα τρία Π (Παιδιά, υΠολογιστές, εΠικοινωνία) . Στη σύγχρονη εποχή θεωρείται αναγκαίο ένα σύνολο γνώσεων και δεξιοτήτων, που το ονομάζουμε τεχνολογική εκπαίδευση ή τεχνολογικό εναλφαβητισμό.

Δεδομένου ότι σήμερα στις κοινωνίες μας θα θεωρείται κανείς αναλφάβητος αν δε γνωρίζει να χρησιμοποιεί κατά κάποιο τρόπο τον υπολογιστή, ο Η/Υ θα πρέπει να διδάσκεται στο μαθητή ως γνωστικό αντικείμενο που θα τον προετοιμάζει για την εκπαιδευτική και την επαγγελματική επιβίωση και ανάπτυξή του. Η χρήση του υπολογιστή, συνδέεται με ένα σύνολο δεξιοτήτων που θα είναι απαραίτητες για τις καθημερινές ανάγκες του αυριανού ανθρώπου. Από την απλή αλληλογραφία και

σύνταξη κειμένων μέχρι τις απαιτήσεις της εργασίας σε όλους τους επαγγελματικούς χώρους και την επιστημονική έρευνα, με την οποία κάθε εκπαιδευτικός θα χρειάζεται ίσως να είναι εξοικειωμένος.

Η εισαγωγή των υπολογιστών στις αίθουσες διδασκαλίας είναι πια γεγονός και για τη χώρας μας. Το ενδιάμεσο στάδιο, για την μετάβαση από την διδακτική θεωρία στην πράξη, είναι η εύρεση του κατάλληλου εργαλείου. Εργαλεία έχουν βρεθεί και ίσως όχι χρησιμοποιηθεί πολύ αρκετά, ο λόγος του δασκάλου, ο πίνακας, η τηλεόραση, το βίντεο, κ.λ.π. Αυτό που διεθνώς έχει πλέον πιστοποιηθεί και προτείνουμε, είναι η χρήση του υπολογιστή σαν εργαλείου, σαν εποπτικό μέσο αν θέλετε στην εκπαιδευτική διαδικασία.

Λέμε ότι ο υπολογιστής είναι σήμερα το καλύτερο εργαλείο που μπορεί να μας πάει από την θεωρία στην πράξη. Όντως έτσι είναι και αυτό μπορούμε να το δούμε πολύ βιαστικά από δύο κύρια χαρακτηριστικά του. Πρώτα απ' όλα μπορεί να επεξεργαστεί μεγάλο όγκο δεδομένων πολύ γρήγορα και αφετέρου συνδυάζει πολλά μέσα. Μπορεί να χρησιμοποιηθεί σαν πίνακας, σαν βίντεο, σαν κασετόφωνο ή σαν συνδυασμός όλων αυτών με τις νέες τεχνολογίες των πολυμέσων και είναι στη διάθεσή μας αυτή τη στιγμή, ένα ιδεατό, θα μπορούσαμε να πούμε, εργαλείο που ανάλογα με το μοντέλο που ίσως χρησιμοποιεί ο εκπαιδευτικός, με τις ανάγκες που θέλει να καλύψει και με τους στόχους που έχει βάλει, να χρησιμοποιήσει αυτό το εργαλείο όπως αυτός θέλει.

Ένα άλλο χαρακτηριστικό του υπολογιστή, είναι ότι μας δίνει τη δυνατότητα να έχουμε πληροφορία οποιουδήποτε τύπου τη στιγμή που τη θέλουμε, μπορεί να εκμεταλλεύεται πολλά συστήματα συμβόλων (κείμενο, ήχος, εικόνα, βίντεο, τρισδιάστατη αναπαράσταση) και πάλι κατά την επιλογή του δασκάλου.

Με τον υπολογιστή σαν εκπαιδευτικό εργαλείο, έχει έρθει μια επανάσταση στα εκπαιδευτικά πράγματα. Ο μαθητής πλέον, δεν είναι ένας παθητικός δέκτης αυτών που διαδραματίζονται στην τάξη, αλλά γίνεται ενεργός συμμέτοχος. Είναι ένα ενεργό στοιχείο της εκπαιδευτικής διαδικασίας. Δρα, κάτι κάνει, παίζει αν θέλετε, δημιουργεί, οικοδομεί, φτιάχνει.

Ο Υπολογιστής σαν διδακτικό εργαλείο καλείται λοιπόν να παίξει έναν σημαντικότατο ρόλο μέσο των πολυμέσων (multimedia) . Μπορεί να χρησιμοποιηθεί

με τους παρακάτω τρόπους: Για εμπέδωση ήδη διδαχθείσας ύλης με προγράμματα εξάσκησης.

Σαν καθοδηγητικό μέσο για μάθηση με προγράμματα διδασκαλίας και επίδειξης, που με διαλογικό τρόπο προσφέρουν σταδιακή γνώση σε κάποιο γνωστικό αντικείμενο.

Σαν μέσο προσομοίωσης και μοντελοποίησης σε καταστάσεις είτε του μικρόκοσμου είτε του μακρόκοσμου, είτε υποκαθιστώντας ένα κλασικό εργαστήριο όπου, αλλάζοντας παραμέτρους, παρατηρούνται τα αποτελέσματα. Τα πολυμέσα και τα προγράμματα που βοηθούν στην ειδικότερη και γενικότερη μάθηση είναι πεδίο τρανό για αυτού του είδους τη χρήση, ιδιαίτερα στις μικρές ηλικίες γιατί μπορεί με άνεση να συνδυάσει το παιχνίδι με τη μάθηση. Παίζοντας ο νεαρός χρήστης, μπορεί να μάθει κυκλοφοριακή αγωγή, να γράφει, να διαβάζει ή να λύνει απλές αριθμητικές ασκήσεις. Αλλά παρ' όλο που τέτοιοι τίτλοι υπήρχαν άφθονοι από παλιά , ήταν ακατάλληλοι για την Ελλάδα, μιας και ήταν ξενόγλωσσοι.

Τα πράγματα όμως έχουν αλλάξει πολύ τον τελευταίο καιρό και τώρα υπάρχει μια μικρή αλλά ικανή βάση ελληνικών ή εξελληνισμένων προγραμμάτων, η οποία συνεχώς διευρύνεται.

Η χρήση υπολογιστών για την εκπαίδευση των παιδιών και την κατανόηση θεμελιωδών κανόνων του κόσμου που μας περιβάλλει, έχει αποδείξει πως αυτή μπορεί να είναι ουσιαστική και το σπουδαιότερο ταχύρυθμη. Σε αντίθεση με την τηλεόραση ή την προβολή κινηματογραφικών θεμάτων στο παιδί, η επαφή με τον υπολογιστή προσφέρει την αμεσότητα της αλληλεπίδρασης.

Ο Η/Υ μπορεί να χρησιμοποιηθεί ως εποπτικό μέσο διδασκαλίας, όπως είπαμε, σε όλα τα μαθήματα από τη γλώσσα, τα μαθηματικά μέχρι και τις τέχνες. Έτσι :

- Το μάθημα γίνεται πιο κατανοητό, πιο ευχάριστο και δίνει ερεθίσματα για περισσότερη εμβάθυνση – με την παρέμβαση πάντοτε του δασκάλου.
- Ο Η/Υ έχει απεριόριστη υπομονή και δεν «τραβάει ποτέ τα αυτιά των παιδιών» ούτε επηρεάζεται από υποσυνείδητες προκαταλήψεις. Το παιδί δεν έχει λόγο να τον ντρέπεται ούτε να φοβάται μην το περάσει για ηλίθιο, κάτι που εμείς οι δάσκαλοι δεν μπορούμε πάντα να το αποφύγουμε.

- Η ενίσχυση που δίνεται στο μαθητή από τη σωστή απάντηση είναι άμεση κι αυτό ενισχύει και την ίδια τη μάθηση.
- Επιτρέπει στο μαθητή να προχωρήσει στην εργασία του με ρυθμό ανάλογο με τις δικές του ικανότητες. Έτσι αν κάποιος μαθητής, για οποιοδήποτε λόγο, έχει διακόψει τη φοίτηση ενός μαθήματος, μπορεί να αρχίσει από εκεί που σταμάτησε την τελευταία φορά, χωρίς να δημιουργεί κενό που μπορεί να είναι καθοριστικό για την πρόοδό του. Αλλά και το αντίθετο : Αν κάποιος μαθητής τελειώσει την εργασία που του ανατέθηκε, μπορεί να συνεχίσει με άλλη εργασία ανώτερου επιπέδου και έτσι δεν υπάρχει αυτή η ισοπέδωση των ικανοτήτων και ενδιαφερόντων.
- Η ασχολία των μαθητών με τους Η/Υ επιτρέπει στο δάσκαλο να επιτηρεί όλη την τάξη και επί πλέον να καταγράφει τις αδυναμίες των μαθητών και να τα βοηθάει ατομικά. Ευνοείται έτσι η εξατομικευμένη και προγραμματισμένη διδασκαλία. Εξαιτίας αυτής της βαθμιαίας πορείας ο μαθητής ελέγχει την πρόοδό του και φθάνει σε σημείο να απαντά σχεδόν πάντα σωστά.
- Τα εκπαιδευτικά προγράμματα έχουν τη δυνατότητα να διαδίδονται πιο εύκολα ακόμα και στα πιο απομακρυσμένα χωριά. Έτσι η διάδοση της γνώσης γίνεται ευκολότερη και δημιουργούνται ίσως ευκαιρίες για κάθε παιδί του χωριού και της πόλης.
- Ο Η/Υ ως διδακτικό μέσο έχει αποδειχθεί εξαιρετικά κατάλληλο με εκπληκτικά αποτελέσματα – ιδιαίτερα σε ορισμένες τάξεις του δημοτικού – για την πρόοδο μαθητών με μαθησιακές δυσκολίες ποικίλης αιτιολογίας (νοητική ή αισθητηριακή υστέρηση, ιδιαιτερότητες συμπεριφοράς, υπερκινητικότητα, απέχθεια προς το βιβλίο και τα μαθήματα κ.α.).
- Οι Υπολογιστές ανταποκρίνονται στις ανάγκες ατόμων διαφορετικής ιδιοσυγκρασίας. Επιτρέπει, δηλαδή, σε μαθητές διαφορετικής προσωπικότητας να αξιοποιούν στο μέγιστο δυνατό βαθμό τις ικανότητές τους. Ακόμη και για τα παιδιά που δεν τα πάνε καλά με τις μηχανές γενικά και έχουν π.χ. περισσότερο θεωρητικά ενδιαφέροντα υπάρχουν απεριόριστες δυνατότητες, ώστε μέσω του υπολογιστή να μπορούν να συνδέονται στο μέλλον με βιβλιοθήκες και να έχουν πρόσβαση σε οποιοδήποτε κείμενο του ενδιαφέροντός τους ή και επικοινωνία με κάποιο επιστήμονα.

- Ο υπολογιστής είναι σημαντικός όχι μόνο γι' αυτό που κάνει, αλλά και για το πώς σε κάνει να αισθάνεσαι. Ο υπολογιστής ικανοποιεί και την ανάγκη για αναζήτηση ταυτότητας. Για παράδειγμα, οι «χάκερς» ή οι πολύ έμπειροι μαθητές γύρω από τον υπολογιστή, που μπορεί και να μην έχουν άλλες ακαδημαϊκές επιδόσεις, χρησιμοποιούν την κυριαρχία τους πάνω στη μηχανή για να δείξουν τις ικανότητές τους στον τομέα αυτό τουλάχιστον. Τα παιδιά αισθάνονται ότι η γνώση που αποκτάται με τη χρήση του υπολογιστή «ανήκει και σ' αυτά και όχι μόνο στους δασκάλους».

Τα παιδιά, χρησιμοποιώντας τον υπολογιστή μαθαίνουν να συνεργάζονται ανταλλάσσοντας προγράμματα και μελέτες, συζητώντας τον τρόπο με τον οποίο λύνουν κάποια προβλήματα, μαθαίνοντας ως παρατηρητές, παίζοντας ένα παιχνίδι κ.τ.λ. Η διδασκαλία, για παράδειγμα, σε μικρά παιδιά του Δημοτικού ήταν αποδοτική, όταν τα παιδιά εργάζονταν τρία τρία και μάλιστα χωρίς τον κλασικό ρόλο του δασκάλου. Σε μια τυπική σχολική τάξη εξάλλου δεν προσφέρονται σε όλους τόσες πολλές ευκαιρίες ανάπτυξης δικτύων αμφίδρομης επικοινωνίας, ώστε να κινδυνεύει η κοινωνικότητα του μαθητή από τη χρήση του υπολογιστή. Αντίθετα ο υπολογιστής συμβάλει στη μείωση ιεράρχησης της σχέσης δασκάλου – μαθητή. Στην περίπτωση αυτή ο δάσκαλος είναι κι αυτός ένας μαθητευόμενος.

1.4 Θεωρίες Μάθησης στον Σχεδιασμό Εκπαιδευτικού Λογισμικού

1.4.1 Συμπεριφοριστική Προσέγγιση του Skinner

Ο Skinner (1904-1990) ανέπτυξε τη θεωρία του, εισάγοντας την έννοια των συνεπειών στη συμπεριφορά της ανθρώπινης μάθησης. Όπως περιέγραφε στη θεωρία του, η πιθανότητα επανάληψης μίας συμπεριφοράς, εξαρτάται σε μεγάλο βαθμό από την ποσότητα της ευχαρίστησης (ή του πόνου) που προξένησε η ίδια η συμπεριφορά στο άτομο, ή που έχει επιφέρει στο παρελθόν σε αυτό. Σύμφωνα με αυτόν λοιπόν, εισάγεται στην ερμηνεία της ανθρώπινης συμπεριφοράς, η επιλογή και η ελεύθερη βούληση.

Εκείνη την εποχή, ο Skinner συμπεριέλαβε στο λεξιλόγιο του συμπεριφορισμού τις βασικές έννοιες της θετικής και αρνητικής ενίσχυσης και της

τιμωρίας. Σύμφωνα με το μοντέλο της συντελεστικής μάθησης του Skinner, οι άνθρωποι μαθαίνουν να συμπεριφέρονται μέσω μίας διαδικασίας δοκιμής και σφάλματος, όπου θυμούνται με ποιές συμπεριφορές αποκόμισαν θετικά ή ευχάριστα αποτελέσματα και με ποιές αρνητικά. Οι απόψεις του προέρχονταν από την παρατήρηση της συμπεριφοράς σε ποντίκια, τα οποία είχε απομονώσει σε κλουβιά - τα λεγόμενα «Skinner Boxes»- ήταν στερημένα από τροφή και μπορούσαν να λαμβάνουν τροφή πατώντας κάθε φορά ένα μοχλό. Στη συνέχεια, μελέτησε τον τρόπο με τον οποίο κινούνται και λειτουργούν αυτά στο περιβάλλον τους, καθώς και την αλλαγή της συμπεριφοράς τους, ανάλογα με τις εκάστοτε διαφορετικές μορφές ενίσχυσης, που τους δίνονταν σε κάθε πειραματική συνθήκη.

Με τον τρόπο αυτό, ο Skinner, διατύπωσε τη θεωρία της συντελεστικής μάθησης, κατά την οποία η συμπεριφορά μπορεί να διαμορφωθεί με την ύπαρξη ή την έλλειψη της ενίσχυσης και την εφάρμοσε σε μία ευρεία κλίμακα συμπεριφορών, σε ανθρώπους και ζώα, όπως περιγράφει στο βιβλίο του «The behavior of Organisms», το 1938.

Ο Skinner υπήρξε ο πρόδρομος των μηχανών διδασκαλίας (teaching machines), πριν αναπτυχθούν τα σύγχρονα εκπαιδευτικά περιβάλλοντα. Πίστευε ότι οι μηχανές αυτές θα μπορούσαν να δημιουργήσουν περιβάλλοντα ευνοϊκά για τη μάθηση που θα ανέτρεπαν τα μειονεκτήματα του σχολικού συστήματος, εφόσον θα ασχολούνταν με τις απαντήσεις των μαθητών και θα ενίσχυαν τις σωστές απαντήσεις αμέσως μετά τη διατύπωση τους από τους μαθητές, κάτι που δεν γινόταν στο πλαίσιο της συνηθισμένης διδασκαλίας. Πίστευε επίσης ότι οι διδακτικές μηχανές θα μπορούσαν να εφαρμόσουν ορισμένες γενικές αρχές της διδασκαλίας, η οποία θα στηριζόταν στον προγραμματισμό των διαδοχικών ερωτήσεων προς το μαθητή γι' αυτό και η διδασκαλία αυτή ονομάστηκε προγραμματισμένη διδασκαλία.

1.4.2 Piaget, Θεωρία της Γενετικής Επιστημολογίας

Ο Piaget ενδιαφέρθηκε ιδιαίτερα για την ανάπτυξη της γνώσης στο μέσο παιδί, σε αυτό που ονόμασε "epistemic subject", και όχι στο μεμονωμένο παιδί. Δεδομένου ότι η γνώση εξελίσσεται, για τον Piaget κάθε βήμα ανάπτυξης είναι ζωτικής σημασίας και αξιόπιστο. Αλλά οι ιδέες των παιδιών είναι πολύ διαφορετικές από εκείνες του

ενηλίκου και κατά συνέπεια υπάρχει μια ανάγκη να γίνουν σεβαστές οι απόψεις των παιδιών για τον κόσμο και η οποιαδήποτε διαδικασία μάθησης να προσπαθήσει να στηριχτεί σε αυτές.

Κατά τον Piaget δεν έχει καμία σημασία πόσο παράξενη ή διαφορετική μας φαίνεται ή ιδέα ενός παιδιού. Αυτό που έχει επιτακτική σημασία προκειμένου να την κατανοήσουμε πλήρως είναι ο τρόπος που το παιδί αντιλαμβάνεται το περιβάλλον του τη δεδομένη χρονική στιγμή.

Ο Piaget προσδιόρισε αλλά και περιέγραψε τέσσερα στάδια της νοητικής ανάπτυξης του παιδιού:

α) Αισθητικοκινητικό (0-18 μηνών)

β) Προεννοιολογικό (18 μηνών-6 ετών)

γ) "Συγκεκριμένων λογικών ενεργειών" (6 ετών -15 ετών)

δ) "Τυπικών λογικών ενεργειών" (15-16 ετών)

Τα παραπάνω αυτά στάδια που ανέλυσε ο Piaget αποτέλεσαν σημείο αναφοράς για τον τρόπο διδασκαλίας για όλους τους εκπαιδευτικούς ανά τον κόσμο, αλλά ταυτόχρονα έριξαν φως σε παρεξηγημένες έννοιες του παρελθόντος γύρω από την ψυχολογία του παιδιού. Ο εγωκεντρισμός (egocentrism) για παράδειγμα που παρατηρείται κατά το αισθητικοκοινωνικό στάδιο είχε κατά το παρελθόν οδηγήσει σε εντελώς λανθασμένα συμπεράσματα, όπως για παράδειγμα ότι ένα παιδί είναι "φύσει κακό" ως φορέας του προπατορικού αμαρτήματος.

Ενδιαφέρον έχει να δούμε τον τρόπο που δούλεψε ο Piaget προκειμένου να οδηγηθεί στα παραπάνω συμπεράσματα αλλά και να ανατρέψει ό,τι αναληθές υπήρχε σε παρελθούσες έρευνες γύρω από την ψυχολογία των παιδιών.

Δεν περιορίστηκε σε βιβλία και θεωρίες παλαιότερων ερευνητών που στις περισσότερες περιπτώσεις αντιμετώπιζαν τα παιδιά ως μια μικρογραφία των μεγάλων. Πρωτοπόρος και ανήσυχο πνεύμα θεώρησε ότι η απάντηση σε όλα τα ερωτήματα βρισκόταν στα ίδια τα παιδιά. Ο ίδιος πέρασε το μεγαλύτερο μέρος της επαγγελματικής του σταδιοδρομίας παρακολουθώντας παιδιά να παίζουν και συνομιλώντας μαζί τους, στα πλαίσια που ήταν αυτό δυνατό δεδομένου ότι κάποια από αυτά μπορούσαν στοιχειωδώς να μιλήσουν.

Μετά από χρόνια ερευνών και αναρίθμητων τέτοιων συζητήσεων ο Piaget κατέληξε στο συμπέρασμα πως κάτω από τις φαινομενικά αφελείς και χαριτωμένες απαντήσεις των παιδιών στα ερωτήματα του, υπήρχε μία λογική που είχαν αναπτύξει τα παιδιά με την οποία όμως κανείς μέχρι τότε δεν είχε ασχοληθεί.

Μέχρι το τέλος της ζωής του στην ηλικία των 84 ετών ο Piaget είχε αφιερώσει 75 χρόνια ερευνών στη μελέτη της ψυχολογίας των παιδιών. Τα νούμερα αυτά δεν είναι καθόλου υπερβολικά αν σκεφτεί κανείς ότι πραγματοποίησε την πρώτη του "επιστημονική" έκδοση στην ηλικία των 10 ετών! Η σπουδαιότητα των ανακαλύψεών του απασχόλησε ακόμα και τον Einsten για τις οποίες υποστήριξε: "Ηταν τόσο απλές που μόνο μία ιδιοφυία θα μπορούσε να τις σκεφτεί".

1.4.3 Θεωρίες Δομητισμού-Papert

Ο Papert επικεντρώνει το ενδιαφέρον του στον μαθητή και συγκεκριμένα στον τρόπο που αυτός θα μπορέσει να μάθει τα μαθηματικά εμπειρικά, όπως εμπειρικά μαθαίνει και την μητρική του γλώσσα. Η μάθηση που στηρίζεται σε ένα παράδειγμα, μία οπτική παράσταση, μία χειροπιαστή εμπειρία, βοηθά στην κατανόηση αφηρημένων εννοιών και λογικών πράξεων.

Αυτό βέβαια δεν σημαίνει ότι ο μαθητής θα φτάσει μόνος του σε γενικεύσεις χωρίς τη βοήθεια του δασκάλου. Πέρα από την υποβοήθηση, χρειάζονται και τα κατάλληλα κίνητρα. Τέτοια κίνητρα είναι η ικανοποίηση κάποιου ειδικού ενδιαφέροντος, η επίλυση προβλημάτων σχετικών με τη ζωή, κάτι προσωπικά ή κοινωνικά χρήσιμο, το παιχνίδι κτλ.

Ο ίδιος ανέπτυξε τη μαθηματική του σκέψη όταν ήταν μαθητής, προσφέροντας στον εαυτό του χειροπιαστά αντικείμενα με τα οποία «μπορεί κάποιος να σκεφτεί». Με τον ίδιο τρόπο βλέπει τη χελώνα της γλώσσας προγραμματισμού Logo, ως αντικείμενο που βοηθά στο να γίνει το αφηρημένο συγκεκριμένο.

1.4.4 Η Κοινωνικό-πολιτιστική Προσέγγιση του Vygotsky

Η κοινωνικό-πολιτιστική προσέγγιση του Vygotsky θεωρείται το πλέον κατάλληλο θεωρητικό πλαίσιο για να στηριχθεί η συνεργατική μέθοδος οργάνωσης της σχολικής τάξης. Η προσέγγιση αυτή έδωσε έμφαση στο κοινωνικό και

πολιτιστικό πλαίσιο μέσα στο οποίο τα παιδιά μαθαίνουν και τόνισε τη σημασία που έχει η καθοδήγηση, η συνεργασία και αλληλεπίδραση με πιο ικανούς ενήλικες ή συμμαθητές. Ο Vygotsky υποστηρίζει ότι η κατασκευή της γνώσης από τους μαθητευόμενους πραγματοποιείται σε ένα περιβάλλον που ευνοεί τη συλλογική επικοινωνία και την κοινωνική αλληλεπίδραση. Η κοινωνική αλληλεπίδραση έχει επιπτώσεις στη γνωστική ανάπτυξη του παιδιού καθώς η βιολογική και ψυχολογική ανάπτυξη του ατόμου δεν πραγματοποιείται σε συνθήκες απομόνωσης. Σύμφωνα πάντα με τον ίδιο ψυχολόγο και παιδαγωγό οι διανοητικές ικανότητες είναι άμεσα εξαρτημένες με μια διαλεκτική σχέση από το πολιτιστικό πλαίσιο μέσα στο οποίο τα παιδιά αναπτύσσονται και το παιδί, με την κατάλληλη καθοδήγηση και βοήθεια από ένα πιο έμπειρο άτομο, μπορεί να πετύχει πολύ περισσότερα.

Ο Vygotsky μελέτησε την γνωστική ανάπτυξη ως αποτέλεσμα μιας διαλεκτικής διαδικασίας κατά την οποία το παιδί μαθαίνει να επιλύει προβληματικές καταστάσεις χρησιμοποιώντας την εμπειρία κάποιου άλλου (γονέα, αδερφών, δάσκαλου, συμμαθητών κ.τ.λ). Ο ίδιος πρότεινε μια έννοια που θεωρείται πολύ σημαντική για την εκπαιδευτική διαδικασία, τη «ζώνη της επικείμενης ανάπτυξης» (Zone of Proximal Development – ZPD). Η έννοια αυτή αναφέρεται στο διάστημα μεταξύ του τι είναι ικανό ένα παιδί να πετύχει μόνο του λύνοντας προβλήματα και τις αντίστοιχες ικανότητες που έχει, όταν κάνει το ίδιο κάτω από την καθοδήγηση ενός ενήλικα ή πιο ικανών συμμαθητών. Η ζώνη της επικείμενης ανάπτυξης είναι μεταφορικά η γέφυρα με την οποία το παιδί περνά από αυτά που δεν γνωρίζει σε αυτά που είναι ικανό να μάθει με τη βοήθεια και συνεργασία άλλων. Η μάθηση κατά τον Vygotsky λαμβάνει χώρα σε αυτό ακριβώς το διάστημα. Το πρώτο επίπεδο (όσα μπορεί να κάνει ένα παιδί μόνο του) ορίζεται ως πραγματική ανάπτυξη. Δυο παιδιά μπορούν να έχουν το ίδιο επίπεδο πραγματικής ανάπτυξης, όταν είναι ικανά να δώσουν λύση στον ίδιο αριθμό προβλημάτων ίδιου βαθμού δυσκολίας. Αν, όμως, σε δεύτερο επίπεδο το ένα από αυτά λάβει την απαιτούμενη βοήθεια από έναν ενήλικο ή έναν ικανό συμμαθητή, σίγουρα θα γίνει πιο ικανό στην επίλυση προβλημάτων. Ακριβώς αυτό το δεύτερο επίπεδο (το τι μπορεί να κάνει ένα παιδί με την συνεργασία ή καθοδήγηση) το ονομάζει δυνητική ανάπτυξη (potential development). Επομένως, τα παιδιά δεν μαθαίνουν από μόνα τους και η μάθηση δε λαμβάνει χώρα σε ένα κενό

αλλά είναι κοινωνικά και πολιτιστικά προσδιορισμένη. Στο πλαίσιο αυτό η συνεργατική μέθοδος εργασίας μπορεί να αποτελέσει ένα εργαλείο που έχει τη δυνατότητα να βοηθήσει τους μαθητές να αποκτήσουν ένα ενεργητικό μοντέλο μάθησης. Οι συνεργατικές τεχνικές δημιουργούν ένα ενεργητικό και δυναμικό περιβάλλον μάθησης, «όταν οι μαθητές εμπλέκονται ενεργά στην διατύπωση ερωτημάτων στη δική τους γλώσσα και στη συλλογική διαπραγμάτευση των μαθησιακών δραστηριοτήτων αντί να αρκούνται στην απλή αναπαραγωγή του υλικού ή της πληροφορίας που παρουσιάστηκε από το δάσκαλο ή περιέχεται σε κάποιο βιβλίο»

1.4.5 Ο Εποικοδομητισμός του Bruner

Ο Bruner ανήκει στην κατηγορία των γνωστικών ψυχολόγων της μάθησης, που δίνει έμφαση στη διευκόλυνση της μάθησης μέσα από την κατανόηση των δομών και των επιστημονικών αρχών ενός αντικειμένου και των τρόπων του σκέπτεσθαι του μαθητευόμενου, καθώς και στην υιοθέτηση της ανακαλυπτικής μεθόδου, ή της καθοδηγούμενης ανακάλυψης με την ανάπτυξη εσωτερικών κινήτρων μάθησης από μέρους του μαθητευόμενου.

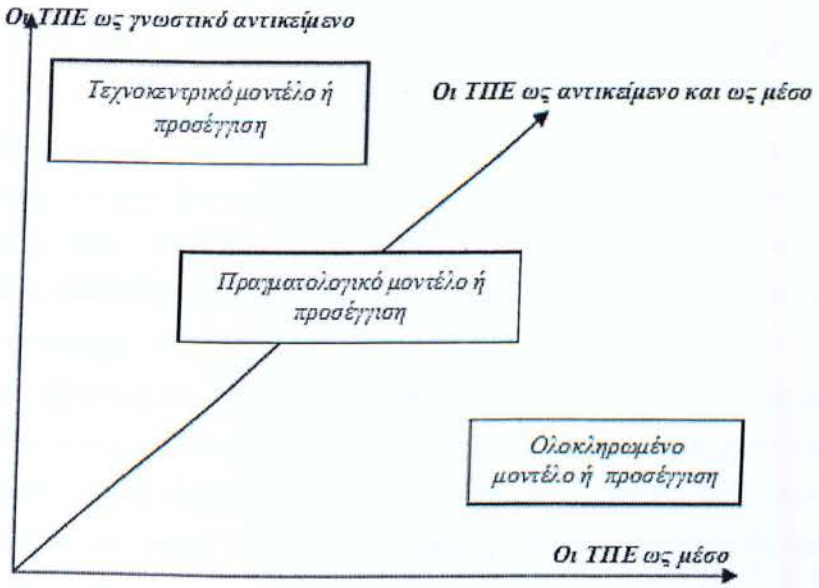
Σχετικά με την απόκτηση της γνώσης ο Bruner υποστηρίζει την ανακαλυπτική-διερευνητική μάθηση, κατά την οποία ο μαθητής με τις δικές του δυνάμεις προσπαθεί να εμβαθύνει στο αντικείμενο και να ανακαλύψει τις θεμελιώδεις αρχές και σχέσεις που διέπουν τα επιμέρους στοιχεία του. Εδώ η λογική σκέψη του ατόμου παίζει ρόλο, όμως ο Bruner θεωρεί ότι το άτομο πρέπει να προχωρήσει παραπέρα και να καλλιεργήσει τη διαισθητική σκέψη, που του επιτρέπει να κάνει πνευματικά άλματα, να πρωτοτυπεί, να εφευρίσκει και να συλλαμβάνει ριζοσπαστικές λύσεις σε προβληματικές καταστάσεις.

Επίσης, ο Bruner υποστηρίζει ότι όλοι οι μαθητές είναι δυνατόν να μάθουν οτιδήποτε και σε οποιαδήποτε ηλικία, εφόσον υπάρχει η κατάλληλη δομή και οργάνωση της ύλης, καθώς και η απαραίτητη μεθόδευση της διδασκαλίας. Η θέση αυτή του Bruner προκάλεσε αρκετές αντιδράσεις, αφού προσέκρουσε στις μέχρι τότε αποδεκτές αντιλήψεις για το θέμα αυτό, αλλά και επέφερε επαναστατικές αλλαγές

τόσο στη φύση των αναλυτικών προγραμμάτων όσο και στην οργάνωση και διεξαγωγή της διδασκαλίας.

1.5 Μοντέλα Ένταξης της Πληροφορικής στην Εκπαίδευση

Οι προσεγγίσεις που παρουσιάστηκαν στις προηγούμενες παραγράφους δεν αλληλοαναιρούνται αλλά αλληλοσυμπληρώνονται και αλληλοεξαρτώνται. Έτσι, μέσα στην καθημερινή εκπαιδευτική πρακτική, φαίνεται να επικρατούν τρεις τάσεις χρήσης των νέων τεχνολογιών της πληροφορίας και της επικοινωνίας στην εκπαιδευτική διαδικασία:



- *Τεχνοκεντρικό Μοντέλο*

Η τεχνοκεντρική προσέγγιση υπήρξε η πρώτη χρονολογικά προσέγγιση εισαγωγής της πληροφορικής στην εκπαίδευση. Κυριάρχησε κατά την δεκαετία του 1970, κυρίως στις υψηλότερες βαθμίδες της εκπαίδευσης. Από τη στιγμή που ένα μάθημα καθιερώνεται στο αναλυτικό πρόγραμμα, είναι εύλογο να τεθούν και τα συνακόλουθα ερωτήματα που αφορούν το περιεχόμενο του, τους στόχους του, τα ερωτήματα που θέτει η διδασκαλία του, και συνεπώς την ύπαρξη μιας διδακτικής του προσέγγισης.

Το μοντέλο αυτό χαρακτηρίζεται από τεχνοκρατικό ντετερμινισμό και έχει ως βασική επιδίωξη την απόκτηση γνώσεων πάνω στη λειτουργία των υπολογιστών και την εισαγωγή στον προγραμματισμό τους. Η πληροφορική στα πλαίσια αυτά θεωρείται ως αυτοτελές γνωστικό αντικείμενο, και στην διεθνή βιβλιογραφία απαντάται με τον όρο απομονωμένη τεχνική προσέγγιση ή κάθετη προσέγγιση.

- *Ολοκληρωμένο Μοντέλο*

Η ολοκληρωμένη προσέγγιση αναφέρεται στην ένταξη των νέων τεχνολογιών μέσα σε όλα τα μαθήματα ως έκφραση μιας ολιστικής, διαθεματικής προσέγγισης της μάθησης. Το μοντέλο αυτό εμφανίστηκε σχετικά πρόσφατα και χαρακτηρίζεται από το ότι η διδασκαλία της χρήσης των νέων τεχνολογιών και η χρήση τους ενσωματώνεται στα επιμέρους γνωστικά αντικείμενα του προγράμματος σπουδών (αποδίδεται με τον όρο οριζόντια ή ολιστική προσέγγιση). Σύμφωνα με την προσέγγιση αυτή, τα θέματα που αφορούν στους υπολογιστές και στις ΤΠΕ γενικότερα, διδάσκονται μέσα από όλα τα γνωστικά αντικείμενα του σχολείου και δεν συνιστούν ιδιαίτερο γνωστικό αντικείμενο.

Οι υποστηρικτές αυτής της προσέγγισης πιστεύουν ότι η διασπορά της διδασκαλίας και της χρήσης της πληροφορικής σε όλο το φάσμα του προγράμματος σπουδών και όχι η ένταξή του σε ένα ιδιαίτερο αντικείμενο, μπορεί να βοηθήσει την ουσιαστική και από κοινού δημιουργική συμμετοχή εκπαιδευτικών και μαθητών στην εκπαιδευτική διαδικασία.

Η προσέγγιση αυτή προϋποθέτει σημαντικά διαφορετικές εκπαιδευτικές αντιλήψεις, τόσο στην επιλογή της γνώσης και της διδακτικής πρακτικής όσο και στην εκπαίδευση και την κατάρτιση των εκπαιδευτικών και στην υλικοτεχνική υποδομή.

- *Πραγματολογικό Μοντέλο*

Η πραγματολογική προσέγγιση συνιστά συνδυασμό των δυο προηγούμενων προσεγγίσεων. Συνιστά μια μεταβατική, "εφικτή" λύση, απαραίτητη για ένα

τουλάχιστον χρονικό διάστημα μέχρι την πλήρη ένταξη των τεχνολογιών σε όλο το αναλυτικό πρόγραμμα.

Το πρότυπο αυτό, χαρακτηρίζεται από τη διδασκαλία ενός αμιγούς μαθήματος γενικών γνώσεων πληροφορικής και την προοδευτική ένταξη της χρήσης των νέων τεχνολογιών ως μέσο στήριξης της μαθησιακής διαδικασίας σε όλα τα γνωστικά αντικείμενα του προγράμματος σπουδών.

Στη βιβλιογραφία αποδίδεται και με τον όρο εφικτή ή μικτή προσέγγιση. Η έμφαση στα πλαίσια αυτής της προσέγγισης, δίνεται στις γνωστικές και τις κοινωνικές διαστάσεις της χρήσης της πληροφορικής στην εκπαιδευτική διαδικασία.

Συνδυάζει τα παιδαγωγικά πλεονεκτήματα της ολοκληρωμένης προσέγγισης με την ανάγκη για τεχνολογικό αλφαβητισμό.

1.6 Ο Προγραμματισμός ως Δραστηριότητα Μάθησης

Σε μια κοινωνία που αναθέτει όλο και περισσότερες καθημερινές εργασίες στους Η/Υ και όπου οι τεχνολογίες της πληροφορίας και της επικοινωνίας αναπτύσσονται ραγδαία, η γενική σχολική εκπαίδευση θα πρέπει να προσφέρει στους μαθητές τη δυνατότητα να αποκτούν μια βαθύτερη κατανόηση της τεχνολογίας των υπολογιστών, των διαφόρων εφαρμογών της, καθώς και των περιορισμών της. Κλειδί για την κατανόηση αυτή αποτελεί η εξοικείωση των μαθητών με βασικές αρχές και έννοιες του προγραμματισμού υπολογιστών, οι οποίες αποτελούν εξάλλου τη βάση για τη λογική και την αφηρημένη συλλογιστική. Για το λόγο αυτό αποτελεί κοινή παραδοχή το ότι ο προγραμματισμός θα πρέπει να κατέχει κεντρικό ρόλο στα σχολικά προγράμματα σπουδών Πληροφορικής. Ωστόσο, παρά τα αναμφισβήτητα οφέλη του προγραμματισμού για τους μαθητές, στην πράξη έχει αποδειχθεί ότι η εκμάθηση του είναι δύσκολη υπόθεση για τους αρχάριους όλων των ηλικιών.

Πιο συγκεκριμένα, ο προγραμματισμός του υπολογιστή δεν αποτελεί απλά μια ικανότητα μεγάλης οικονομικής σημασίας αλλά έχει αυξημένο εκπαιδευτικό ενδιαφέρον. Ο προγραμματισμός θεωρείται δραστηριότητα με την οποία καλλιεργούνται ανώτερες μορφές σκέψης όπως η αναλυτική, η συνθετική, η αναγνώριση προτύπων, κ.α. Με τον προγραμματισμό είναι δυνατό να βελτιωθεί η ικανότητα επίλυσης προβλημάτων. Ο προγραμματισμός θεωρείται επίσης δεξιότητα

κλειδί για την προσέγγιση και κατανόηση άλλων θεμάτων των ΤΠΕ. Ειδικά θέματα της διδακτικής του προγραμματισμού έχουν απασχολήσει Έλληνες και ξένους ερευνητές.

Ο επαγγελματικός προγραμματισμός των Η/Υ έχει εξελιχθεί με το πέρασμα του χρόνου. Τα σύγχρονα περιβάλλοντα αποτελούνται από πληθώρα εργαλείων και κυρίως από προκαθορισμένες ιεραρχίες κλάσεων αντικειμένων οι οποίες κάνουν την μάθησης αρκετά δύσκολη στην αρχή και απομακρύνουν τον προγραμματιστή από τη μηχανή, παρέχοντας του ένα αφηρημένο περιβάλλον μέσα στο οποίο θα εκτελεστούν τα προγράμματα που αναπτύσσει.

Η εκπαίδευση των νέων προγραμματιστών γίνεται συνήθως αξιοποιώντας τα εμπορικά-επαγγελματικά περιβάλλοντα ανάπτυξης εφαρμογών ή άλλα δωρεάν διαθέσιμα που τους μοιάζουν. Τα περιβάλλοντα αυτά παρά την εκτεταμένη τεκμηρίωση, απευθύνονται στον επαγγελματία και δεν είναι κατά ανάγκη κατάλληλα για τον μαθητευόμενο προγραμματιστή. Τα επαγγελματικά περιβάλλοντα, παρά την αίσθηση αυθεντικότητας που δίνουν, είναι ακόμα συχνότερα ακατάλληλα για χρήση στην δευτεροβάθμια και πρωτοβάθμια εκπαίδευση. Αρκεί κανείς να αναλογισθεί τον τυπικό μαθητή της γενικής εκπαίδευσης που δεν έχει υποχρεωτικά σκοπό να αναπτύξει επαγγελματική δεξιότητα προγραμματιστή αλλά θα μπορούσε να εμπλακεί σε δραστηριότητες μάθησης που περιλαμβάνουν και προγραμματισμό. Ένα άλλο ζήτημα, που προκύπτει από την έλλειψη κατάλληλων περιβαλλόντων, είναι η δυσκολία στην ηλικιακή κατανομή του περιεχομένου του προγραμματισμού, επειδή η πολυπλοκότητα των συνηθισμένων περιβαλλόντων προγραμματισμού τα καθιστά εφαρμόσιμα μόνο σε μεγάλες τάξεις.

Οι προσπάθειες εφαρμογής του προγραμματισμού στην εκπαίδευση έχουν αρχίσει να συσσωρεύουν γνώση και εμπειρίες που εκφράζονται και μέσα από τον σχεδιασμό εκπαιδευτικών περιβαλλόντων.

1.6.1 Δυσκολίες των Αρχάριων Προγραμματιστών

Η διδασκαλία και η εκμάθηση του προγραμματισμού χαρακτηρίζονται κατά κανόνα από ορισμένες "δυσκολίες", οι οποίες εκδηλώνονται κυρίως κατά την κατασκευή ενός αλγορίθμου ή ενός προγράμματος. Ορισμένες δυσκολίες

εκδηλώνονται και σε άλλες περιπτώσεις: όταν, για παράδειγμα, ο προγραμματιστής επιχειρεί να αιτιολογήσει ή να προβλέψει τη συμπεριφορά ενός αλγορίθμου ή επιχειρεί να τον διορθώσει. Ορισμένες δυσκολίες μοιάζουν να είναι, κατά κάποιο τρόπο, εγγενείς στον ίδιο τον προγραμματισμό, με την έννοια ότι συναντώνται κατά τρόπο συστηματικό, και είναι σχεδόν ανεξάρτητες από τη μέθοδο διδασκαλίας του αντίστοιχου αντικειμένου. Επίσης, πολλές απ' αυτές παρουσιάζουν εξαιρετική ανθεκτικότητα στο χρόνο και συναντώνται είτε σε μαθητές δημοτικού είτε σε σπουδαστές και φοιτητές (Pea 1986). Σήμερα γνωρίζουμε αρκετά καλά τα προβλήματα που αντιμετωπίζουν οι αρχάριοι προγραμματιστές στους τομείς αυτούς - και σε μερικές περιπτώσεις γνωρίζουμε διδακτικές μεθόδους για την υπέρβαση τους. Υπάρχουν πολλές θεωρίες που προσπαθούν να εξηγήσουν τι είναι αυτό που καθιστά την εκμάθηση του προγραμματισμού τόσο δύσκολη (Brusilovsky 1997, Du Boulay 1989, Brooks 1977). Θέματα που έχουν ερευνηθεί κυρίως αφορούν:

- προβλήματα της διδασκαλίας και κατανόησης των μεταβλητών

Η ανάθεσης τιμής σε μεταβλητή παρουσιάζει δυσκολίες και είναι γενικά αποδεκτό ότι αποτελεί ένα από τα προβλήματα που συναντούν οι μαθητές στη διδασκαλία του προγραμματισμού. Η δυσκολία που παρουσιάζει η έννοια της μεταβλητής πηγάζει ορισμένες φορές από την επιλογή του συμβόλου ανάθεσης τιμής. Ως αποτέλεσμα, εκφράσεις της μορφής "X=X+1" προκαλούν σύγχυση και σύγκρουση μεταξύ Προγραμματισμού και Μαθηματικών. Η συγκεκριμένη έκφραση δεν έχει νόημα στα Μαθητικά, αφού δεν είναι δυνατό ένας αριθμός X να είναι ίσος με τον εαυτό του συν τη μονάδα, ενώ στον Προγραμματισμό παριστάνει μια εκχώρηση. Ορισμένοι ερευνητές διατυπώνουν επιπλέον την άποψη ότι στις εκφράσεις του τύπου "X=Y+1", υπάρχει μια σύγχυση σχετικά με το αν το σύμβολο "=" είναι το σύμβολο ισότητας ή το σύμβολο της εκχώρησης τιμής. Αρκετές γλώσσες προγραμματισμού κάνουν χρήση του "=" ως συμβόλου εκχώρησης τιμής, προφανώς για λόγους απλότητας, γεγονός που δημιουργεί σύγχυση με το αλγεβρικό σύμβολο της ισότητας.

Ο Du Boulay (1989) θεωρεί ότι οι παρερμηνείες στην κατανόηση των μεταβλητών βασίζονται στα παραδείγματα που χρησιμοποιούν οι εκπαιδευτικοί στην

τάξη (λ.χ. η παρομοίωση της μεταβλητής με ένα κουτί ή ένα συρτάρι με μια ετικέτα, μπορεί να παρερμηνευτεί από τους μαθητές και να θεωρήσουν ότι η μεταβλητή μπορεί να έχει περισσότερες από μία τιμές). Έτσι ορισμένοι μαθητές δεν κατανοούν ότι η νέα τιμή καταχωρείται στη θέση της παλιάς η οποία χάνεται. Θεωρούν τη μεταβλητή σαν μια λίστα που περιέχει όλες τις τιμές που έχουν εκχωρηθεί στη μεταβλητή, οι οποίες και μπορούν να ανακτηθούν.

Οι Bayman & Mayer (1983) καταλήγουν στο συμπέρασμα ότι δύο είδη εκχώρησης τιμών σε μεταβλητές μπορεί να οδηγήσουν σε παρερμηνείες. Η πρώτη περίπτωση είναι η αρχικοποίηση (X=0) και η άλλη είναι η εξίσωση (X=Y+1). Οι μαθητές συχνά θεωρούν ότι ο υπολογιστής έχει καταγράψει κάπου την πληροφορία ή την έχει εκτυπώσει στην οθόνη ενώ αντίθετα η πληροφορία έχει αποθηκευθεί σε συγκεκριμένη θέση στη μνήμη. Πολλοί μαθητές θεωρούν ότι αποθηκεύεται η εξίσωση και όχι η τιμή. Οι συγγραφείς καταλήγουν στο συμπέρασμα ότι "οι αρχάριοι προγραμματιστές χρειάζονται ειδική εκπαίδευση στα θέματα που αφορούν θέσεις μνήμης και κάτω από ποιες συνθήκες οι τιμές που αποθηκεύονται στις θέσεις αυτές μπορούν να αντικατασταθούν".

Η Samurcay (1985) καταλήγει σε τέσσερις τρόπους ανάθεσης τιμών σε μεταβλητές:

- Ανάθεση σταθερής τιμής (constant value, A=3).
- Ανάθεση τιμής που προκύπτει από υπολογισμό (calculated value, A=2*B+1).
- Αντιγραφή (duplication, A=B).
- Συσσώρευση (accumulation, A=A+1).

Οι τρεις πρώτες περιπτώσεις δεν εμφανίζουν ιδιαίτερα διδακτικά προβλήματα αφού οι κατάλληλες αναπαραστάσεις που απαιτούνται έχουν ήδη οικοδομηθεί από τον χώρο των μαθηματικών. Όμως στην περίπτωση της συσσώρευσης η έννοια της μεταβλητής προκαλεί σύγχυση και απαιτεί διαφορετική αντιμετώπιση. Στην περίπτωση αυτή πρέπει να γίνει σαφής διάκριση μεταξύ αριστερού και δεξιού τμήματος της εντολής ανάθεσης, και να κατανοήσουν οι μαθητές ότι το αριστερό τμήμα σχετίζεται με τη θέση μνήμης ενώ το δεξί τμήμα με την τιμή που θα πάρει η μεταβλητή.

Η Samurcay επίσης κατατάσσει τις μεταβλητές σε δύο κατηγορίες - εσωτερικές και εξωτερικές - και περιγράφει τη χρήση των τεσσάρων τρόπων ανάθεσης σε κάθε κατηγορία μεταβλητών. Εξωτερικές είναι οι μεταβλητές που αποτελούν είσοδο ή έξοδο (αποτέλεσμα) σε ένα πρόγραμμα. Αυτές είναι υπό τον έλεγχο του χρήστη, όταν εκτελεί το πρόγραμμα. Εσωτερικές είναι οι μεταβλητές που είναι αναγκαίες μόνο στη (προγραμματιστική) λύση ενός προβλήματος και είναι υπό τον έλεγχο του προγραμματιστή. Η Samurcay πιστεύει ότι οι αρχάριοι δυσκολεύονται περισσότερο με τον χειρισμό των εσωτερικών μεταβλητών. Αυτό συμβαίνει γιατί, στην περίπτωση αυτή, απαιτείται μια διαρκής αναπαράσταση της εσωτερικής λειτουργίας του υπολογιστή.

- προβλήματα της διδασκαλίας και κατανόησης των επαναληπτικών δομών

Οι επαναληπτικές δομές αποτελούν δομικό στοιχείο σχεδόν κάθε προγράμματος. Οι ερευνητές συμφωνούν στο συμπέρασμα ότι οι επαναληπτικές δομές αποτελούν μια περιοχή όπου παρατηρούνται δυσκολίες στους αρχάριους προγραμματιστές. Οι δυσκολίες αυτές μπορεί να οφείλονται είτε σε αδυναμία γενίκευσης (Hoc 1989) είτε σε αδυναμία ανάπτυξης ενός κατάλληλου νοητικού μοντέλου της δομής του βρόχου (Kessler 1989). Ο Hoc (1989) παρατηρεί ότι υπάρχει τάση στους νέους προγραμματιστές να γράφουν ξανά το ίδιο κομμάτι κώδικα αντί να χρησιμοποιήσουν βρόχο. Ο Du Boulay (1989) θεωρεί ότι οι βρόχοι δημιουργούν στους αρχαρίους αρκετά προβλήματα. Αρκετοί δεν κατανοούν πως γίνεται η αλλαγή τιμής στον μετρητή ενός βρόχου FOR, μια και δεν αποτυπώνεται η ενέργεια αυτή στον κώδικα, αλλά γίνεται εξ υποθέσεως. Οι Rogalski και Samurcay (1990) καταγράφουν τις ενέργειες που λαμβάνουν χώρα στη δημιουργία ενός βρόχου. Τρεις τύποι διεργασιών μεταβλητών εμπεριέχονται σε ένα βρόχο:

- Αρχικοποίηση (initialization), όπου γίνεται η απόδοση των αρχικών τιμών στις μεταβλητές του βρόχου.
- Ενημέρωση (updating), όπου γίνεται η (απαραίτητη) αναπροσαρμογή των τιμών των μεταβλητών.
- Έλεγχος (test), όπου καθορίζεται η συνθήκη τερματισμού του βρόχου.

Οι αρχάριοι συνήθως δεν μπορούν να καθορίσουν το τμήμα ενημέρωσης και συχνά γράφουν τον κώδικα ενός βρόχου παραλείποντάς το. Επίσης δυσκολεύονται να «συνθέσουν» μια συνθήκη εξόδου για ένα τμήμα κώδικα (το σώμα του βρόχου) με το οποίο δεν έχουν ακόμα καταπιαστεί. Η έρευνα του Soloway (1983) καταλήγει στο συμπέρασμα ότι οι αρχάριοι γράφουν πιο σωστά το τμήμα των βρόχων ενός προγράμματος αν χρησιμοποιήσουν μια δομή που επιτρέπει την έξοδο από το μέσο του βρόχου και όχι από την αρχή ή το τέλος.

Τα πιο συχνά λάθη και παρανοήσεις των μαθητών που σχετίζονται με βρόχους, σύμφωνα με τον Sleeman (1988) είναι:

- Θεωρούν ότι μια εντολή που βρίσκεται αμέσως μετά το τέλος του βρόχου, συμπεριλαμβάνεται σε αυτόν.
- Η τελευταία εντολή ενός βρόχου εκτελείται πολλές φορές, ενώ όλες οι άλλες εντολές εκτελούνται μία φορά.
- Πιστεύουν ότι μια μεταβλητή κρατά περισσότερες από μία τιμές και έτσι χειρίζονται μια εντολή επιλογής σαν βρόχο.
- Πιστεύουν ότι η μεταβλητή που χρησιμοποιείται ως μετρητής στο βρόχο FOR ή δεν έχει τιμή μέσα στο βρόχο, ή ότι είναι σωστό να αλλάζει η τιμή της μέσα στο βρόχο.

- προβλήματα της διδασκαλίας και κατανόησης των εντολών επιλογής

Οι αρχάριοι προγραμματιστές συναντούν τις ακόλουθες δυσκολίες στην κατανόηση των δομών επιλογής:

- Εμφωλευμένες δομές επιλογής: Εδώ ο βαθμός δυσκολίας αυξάνεται ανάλογα με το βάθος εμφώλευσης (Rogalski 1990).
- Σύνθετες εκφράσεις – συνθήκες: Οι πολύπλοκες εκφράσεις που χρησιμοποιούν Boolean συναρτήσεις ή συνδυασμό προτάσεων με τους λογικούς τελεστές AND, OR και NOT, δυσχεραίνουν σαφώς την κατανόηση της λειτουργίας μιας εντολής ελέγχου (Δαγδιλέλης 1996).

- Εντοπισμός των ορίων εμβέλειας (begin .. end): Η απουσία εντολών που καθορίζουν τα όρια εμβέλειας των τμημάτων μετά τα THEN και ELSE κάνει πιο δύσκολη την αναγνώριση τους από τους μαθητές(Sime 1977).

Ο προγραμματισμός βασίζεται κατά ένα μέρος σε εμφωλευμένες εντολές. Κλασική περίπτωση αποτελούν τα εμφωλευμένα if..then..else. Για πολλούς ερευνητές οι εμφωλευμένες εντολές ελέγχου προκαλούν σύγχυση. Αντίθετα άλλοι ερευνητές θεωρούν ότι η παράθεση ελέγχων είναι σε ελάχιστο βαθμό και σε ορισμένες μόνο περιπτώσεις πιο κατανοητή από τους εμφωλευμένους ελέγχους. Θεωρούν ότι οι εμφωλευμένες εντολές ελέγχου δεν πρέπει να καταργηθούν. Αυτό όμως που φαίνεται ουσιαστικό είναι να υπάρχει μια ορατή αντιστοιχία ανάμεσα στη δομή του κώδικα και τη "φυσική" θέση των εντολών μέσα στο πρόγραμμα. Έχει προταθεί ακόμη κι η χρήση ειδικών συμβόλων που χρησιμοποιούνται από πολλούς συγγραφείς για αναπαράσταση αλγοριθμικών γλωσσών. Σύμφωνα με τους Putnam (1986) και Sleeman (1988), τα πιο συχνά λάθη και παρανοήσεις των μαθητών που σχετίζονται με τις δομές επιλογής είναι:

- Στην περίπτωση που η δομή επιλογής δεν έχει τμήμα ELSE, αναμένουν τη διακοπή εκτέλεσης του προγράμματος και την εμφάνιση μηνύματος λάθους αν η συνθήκη της εντολής IF είναι ψευδής (false).
- Στην περίπτωση που η δομή επιλογής έχει και τμήμα ELSE, αναμένουν την εκτέλεση τόσο του τμήματος THEN όσο και του ELSE.
- Αναμένουν την εκτέλεση του τμήματος THEN ανεξάρτητα από το αν η συνθήκη είναι αληθής ή όχι.
- Στην περίπτωση που η δομή επιλογής δεν έχει τμήμα ELSE, διαχειρίζονται την αμέσως επόμενη εντολή (που δεν ανήκει στην IF..THEN) όπως η ELSE, πιστεύουν δηλαδή ότι η εντολή αυτή εκτελείται μόνο όταν η συνθήκη είναι ψευδής.

- προβλήματα της διδασκαλίας της αναδρομικότητας

Αναδρομή ονομάζεται η δυνατότητα ενός υποπρογράμματος να καλεί τον εαυτό του και είναι μια βασική έννοια του προγραμματισμού. Οι αναδρομικές διαδικασίες προϋποθέτουν μεγάλη εξοικείωση με βασικές έννοιες του

προγραμματισμού και θέτουν σημαντικά προγραμματιστικά αλλά και διδακτικά προβλήματα.

Η αναδρομή είναι ένα χρήσιμο, δυνατό και αποτελεσματικό εργαλείο στα χέρια ενός προγραμματιστή. Παρ' όλα αυτά, η εγγενής πολυπλοκότητα της αναδρομικής διαδικασίας δημιουργεί σημαντικά διδακτικά προβλήματα, τα οποία απαντώνται κυρίως από αρχάριους προγραμματιστές και προέρχονται από:

- Απειρία στην κατανόηση, χρήση και υλοποίηση διαδικασιοστρεφούς προγραμματισμού
- Χαμηλή εκτίμηση της λειτουργικότητας και χρηστικότητας της αναδρομής
- Ελλιπής κατάρτιση για την καταγραφή μιας λύσης με αναδρομή

1.6.2 Εναλλακτικές Διδακτικές Προσεγγίσεις του Προγραμματισμού

Η πιο διαδεδομένη μέθοδος εισαγωγής στον προγραμματισμό είναι η σταδιακή παρουσίαση των δομών μιας γλώσσας προγραμματισμού γενικού σκοπού και η επίλυση προβλημάτων αυξανόμενης δυσκολίας με τη χρήση αυτών των δομών. Όμως οι δυσκολίες που συναντούν οι αρχάριοι προγραμματιστές αποτέλεσαν το κίνητρο για την αναζήτηση νέων μεθόδων διδασκαλίας για τα εισαγωγικά μαθήματα προγραμματισμού. Τα τελευταία χρόνια η διδασκαλία του προγραμματισμού έχει επηρεαστεί από την εμφάνιση φυσικών μηχανικών μοντέλων που συνδέονται με υπολογιστή και μπορούν να κινούνται, να εκτελούν έργα και γενικά να αλληλεπιδρούν και από μαθησιακά μίνι-περιβάλλοντα που βασίζονται σε μίνι-γλώσσες και μικρόκοσμους, δηλαδή εκπαιδευτικά περιβάλλοντα κατάλληλα σχεδιασμένα ώστε να καθοδηγούν πολύ προσεκτικά το χρήστη και να του προσφέρουν μια εκπαιδευτική εμπειρία.

- *Η κλασσική μέθοδος εισαγωγής στον προγραμματισμό*

Η πιο διαδεδομένη μέθοδος εισαγωγής στον προγραμματισμό είναι η σταδιακή παρουσίαση των δομών μιας γλώσσας προγραμματισμού γενικού σκοπού και η επίλυση προβλημάτων αυξανόμενης δυσκολίας με τη χρήση αυτών των δομών. Ωστόσο, η προσέγγιση αυτή κρίνεται ως αναποτελεσματική, ιδιαίτερα για μαθητές

μικρής ηλικίας, καθώς θέτει μια σειρά από εμπόδια στους αρχάριους προγραμματιστές: α) απαιτείται από το μαθητή να εξοικειωθεί ταυτόχρονα τόσο με την αυστηρή σύνταξη και τη σημασιολογία της ίδιας της γλώσσας όσο και με τις βασικές αρχές του προγραμματισμού, β) η διδασκαλία και χρήση μιας πλήρους γλώσσας προγραμματισμού στο σχολείο μπορεί να αποβεί πολύ χρονοβόρα, γ) παρέχεται συνήθως περιορισμένη υποστήριξη όσον αφορά στην κατανόηση των βασικών εντολών και δομών ελέγχου της γλώσσας αφού η διαδικασία της εκτέλεσης του προγράμματος παραμένει κρυμμένη από το μαθητή, ενώ η έλλειψη οπτικής ανάδρασης εμποδίζει την κατανόηση της σημασιολογίας της γλώσσας, δ) οι μαθητές δύσκολα μπορούν να εντοπίσουν και να διορθώσουν λάθη στα προγράμματά τους, ε) τα πρώτα προβλήματα που τίθενται στους μαθητές αφορούν κατά κανόνα στην επεξεργασία αριθμών ή συμβόλων και αποτυγχάνουν να κινήσουν το ενδιαφέρον των μαθητών, ενώ η ανάπτυξη πιο ελκυστικών εφαρμογών απαιτεί την εκμάθηση ενός μεγάλου υποσυνόλου της γλώσσας. Ειδικότερα, όσον αφορά στην παρακίνηση των μαθητών, αν η συγγραφή ενός προγράμματος που εμφάνιζε στην οθόνη τη φράση "Hello world" κινούσε το ενδιαφέρον των μαθητών παλαιότερα, δεν συμβαίνει το ίδιο με τη σημερινή 'γενιά του Nintendo' που έλκεται από πολυμεσικά μαθησιακά περιβάλλοντα που θυμίζουν ηλεκτρονικά παιχνίδια.

- *Η μέθοδος των προγραμματιστικών μινι-περιβαλλόντων*

Ως εναλλακτική, πιο αποτελεσματική προσέγγιση για την εισαγωγή μαθητών αλλά και φοιτητών στον προγραμματισμό έχουν προταθεί μαθησιακά μινι-περιβάλλοντα (mini-environments) που βασίζονται σε μινι-γλώσσες (mini-languages) και μικρόκοσμους (microworlds). Οι μινι-γλώσσες είναι μικρές γλώσσες προγραμματισμού ειδικά σχεδιασμένες για τη διδασκαλία του προγραμματισμού. Οι μαθητές μαθαίνουν να προγραμματίζουν καθοδηγώντας τις ενέργειες μιας ψηφιακής οντότητας (π.χ. χελώνας, ρομπότ) που ζει σε έναν εικονικό κόσμο (μικρόκοσμο). Η οντότητα μπορεί να εκτελέσει ένα μικρό σύνολο εντολών και να απαντήσει σε ορισμένες ερωτήσεις που επιστρέφουν τιμές. Συνήθως, ο μαθητής ελέγχει την οντότητα αρχικά δίνοντας μεμονωμένες εντολές και κατόπιν γράφοντας μικρά προγράμματα στη μινι-γλώσσα, που συνήθως περιλαμβάνει όλες τις βασικές δομές

ελέγχου (π.χ. εκτέλεση υπό συνθήκη, βρόγχους), καθώς και μηχανισμούς για τη δημιουργία νέων εντολών και υποπρογραμμάτων (Brusilovsky et al., 1997).

Τα πλεονεκτήματα της προσέγγισης αυτής είναι τα ακόλουθα:

- Μια μινι-γλώσσα έχει μικρό συντακτικό και απλή σημασιολογία. Επομένως, οι μαθητές μπορούν γρήγορα να τη μάθουν και να τη χρησιμοποιήσουν με ενδιαφέροντα αποτελέσματα, επενδύοντας το χρόνο τους σε σημαντικότερα ζητήματα, όπως η κατανόηση προγραμματιστικών δομών και αρχών, η ανάπτυξη αλγορίθμων και η σχεδίαση προγραμμάτων.
- Το όλο προγραμματιστικό περιβάλλον είναι κτισμένο πάνω σε κάποια οπτικά ελκυστική και παρακινητική για τους μαθητές μεταφορά (metaphor) και επιτρέπει στον εκπαιδευτικό να δημιουργήσει ενδιαφέροντα προβλήματα που σχετίζονται με τις καθημερινές εμπειρίες των μαθητών.
- Οι διάφορες ενέργειες που εκτελεί η οντότητα προκαλούν ορατές αλλαγές στο μικρόκοσμο που αναπαρίσταται στην οθόνη, πράγμα που βοηθά τον αρχάριο προγραμματιστή να αντιληφθεί τι κάνει το πρόγραμμά του και να κατανοήσει τη σημασιολογία των διαφόρων δομών της γλώσσας.
- Τα προβλήματα που συνοδεύουν το περιβάλλον μοιάζουν περισσότερο με σπαζοκεφαλιές παρά με «σοβαρά» προβλήματα και η δραστηριότητα της επίλυσής τους γίνεται ένα είδος παιχνιδιού για τους μαθητές.
- Προάγεται η δημιουργικότητα των μαθητών καθώς και η εποικοδομητική (constructivist) μάθηση μέσα από τον ενεργό πειραματισμό.
- Παρέχεται στους μαθητές η δυνατότητα να οικοδομούν νοητικά μοντέλα και να αναπτύσσουν στρατηγικές επίλυσης προβλημάτων που είναι πιθανό να μεταφερθούν αργότερα σε άλλα πλαίσια.

Από την παραπάνω παρουσίαση καθίσταται προφανές ότι τα προγραμματιστικά μινι-περιβάλλοντα συνιστούν ένα απλό αλλά ισχυρό μέσο για την εισαγωγή των μαθητών στις βασικές αρχές του προγραμματισμού, στην αλγοριθμική σκέψη και στη συστηματική επίλυση προβλημάτων, ενώ παράλληλα παρέχουν τα θεμέλια για τη μετέπειτα εκμάθηση μιας γλώσσας προγραμματισμού γενικού σκοπού.

Κεφάλαιο 2

Προγραμματιστικά Περιβάλλοντα για Αρχάριους

2.1 Εισαγωγή

Η διδασκαλία του προγραμματισμού και της αλγοριθμικής επίλυσης προβλημάτων σε αρχάριους, μαθητές της δευτεροβάθμιας εκπαίδευσης ή φοιτητές, όπως είδαμε στο προηγούμενο κεφάλαιο, συνιστά ένα δύσκολο αλλά εξαιρετικά ενδιαφέρον έργο, τόσο από διδακτική όσο και από γνωστική άποψη. Κατά την επίλυση αλγοριθμικών προβλημάτων χρησιμοποιούνται έννοιες και δομές, οι οποίες είναι δύσκολο να οικοδομηθούν με τα παραδοσιακά διδακτικά μέσα (αλγόριθμος, μεταβλητή, αρχικοποίηση, δομή επιλογής, δομή επανάληψης κ.λπ.). Τα συνήθη προγραμματιστικά περιβάλλοντα και οι γλώσσες προγραμματισμού που χρησιμοποιούνται για εκπαιδευτικούς σκοπούς έχουν σχεδιαστεί για την ανάπτυξη εφαρμογών και όχι για τη διδασκαλία του προγραμματισμού. Είναι, συνεπώς, προσαρμοσμένα στο πλαίσιο γνώσεων και δεξιοτήτων των έμπειρων προγραμματιστών, γεγονός που ενισχύει τις δυσκολίες και τα εμπόδια που συναντούν οι μαθητές και οι αρχάριοι στον προγραμματισμό.

Τα τελευταία χρόνια, αναπτύσσεται μεγάλο ερευνητικό και εκπαιδευτικό ενδιαφέρον για τη χρήση ειδικών περιβαλλόντων προγραμματισμού, όπως περιβάλλοντα Logo (γλώσσες Logo, MicroWolds Pro), εκπαιδευτική ρομποτική (π.χ. LEGO/LOGO, JKarelRobot), μικρογλώσσες, προγραμματιστικοί μικρόκοσμοι (π.χ. Karel, Karel++, KarelJ, JEROO), περιβάλλοντα προσομοίωσης και οπτικοποίησης αλγορίθμων, ολοκληρωμένα εκπαιδευτικά προγραμματιστικά περιβάλλοντα κ.λπ. Τα περιβάλλοντα αυτά παρέχουν νέες δυνατότητες για την οικοδόμηση γνώσεων και την ανάπτυξη δεξιοτήτων στον προγραμματισμό.

2.2 Γλώσσα Προγραμματισμού Logo

"Η LOGO είναι με μεγάλη διαφορά η πιο δυνατή γλώσσα προγραμματισμού, που είναι διαθέσιμη για home computers". Η γλώσσα Logo, η οποία αποτελεί διάλεκτο της Lisp, εφευρέθηκε και αναπτύχθηκε από το Νοτιοαφρικανό μαθηματικό Seymour Papert κατά τη δεκαετία του 1960 στο Εργαστήριο Τεχνητής Νοημοσύνης του MIT (Τεχνολογικό Ινστιτούτο Μασαχουσέτης). Σκοπός του Papert ήταν να προσφέρει στα παιδιά ένα περιβάλλον μάθησης μέσα στο οποίο θα μπορούσαν να

αποκτήσουν τη γνώση των μαθηματικών και της γεωμετρίας αρχικά και κατόπιν τη γνώση του πώς μαθαίνουν με τρόπο φυσικό και αβίαστο όπως ακριβώς μαθαίνουν να μιλούν τη μητρική τους γλώσσα χωρίς να τους τη διδάξει κανείς.

Ο Papert ήθελε να δημιουργήσει ένα μέσο, με το οποίο θα μπορούν να σχεδιαστούν αλληλεπιδραστικά περιβάλλοντα, με τα οποία ο μαθητής θα έρχεται σε επαφή για διδακτικούς σκοπούς. Στα περιβάλλοντα αυτά, ο μαθητής καλείται να λύσει προβλήματα που μπορεί να έχουν διάφορες μορφές ελκυστικές για τα παιδιά, όπως παιχνίδια. Οι μαθητές θα μπορούν να εξωτερικεύουν τη σκέψη τους κατά την επίλυση προβλημάτων, ώστε να μπορούν να σκεφτούν για αυτή. Τα λάθη σε μια τέτοια θεώρηση είναι μιας πρώτης τάξης ευκαιρία για ανακάλυψη ατελειών στη διαισθητική γνώση, των γνωστών παρανοήσεων. Οι μαθητές διορθώνουν τις σκέψεις τους και συντονίζουν τα διαισθητικά γνωστικά μοντέλα με την πραγματικότητα.

Η γλώσσα Logo αποτελεί ένα από τα χαρακτηριστικότερα παραδείγματα του πώς χρησιμοποιείται ο υπολογιστής ως γνωστικό εργαλείο, δηλαδή ως εργαλείο που μπορεί να αλλάξει τον τρόπο με τον οποίο σκεφτόμαστε. Θεωρείται το ιδανικότερο μέσο για να μαθαίνεις κάνοντας (learning by doing) και αναμφίβολα αποτελεί σημαντικό εργαλείο στα χέρια του εκπαιδευτικού για την ανάπτυξη δεξιοτήτων εξερεύνησης, δημιουργικότητας, επίλυσης προβλημάτων, λογικής-αλγοριθμικής σκέψης.

Παρέχει ένα περιβάλλον όπου οι μαθητές αναλαμβάνουν το ρόλο του δασκάλου. Η ανάπτυξη ενός προγράμματος γίνεται, δημιουργώντας ένα κύριο πρόγραμμα το οποίο καλεί άλλες συναρτήσεις και διαδικασίες. Η δύναμή της γίνεται αντιληπτή από τον τρόπο που βοηθά τον προγραμματιστή να συγκεντρωθεί στο πρόβλημά του, χωρίς να τον αποπροσανατολίζουν οι ιδιαιτερότητες και οι περιορισμοί του εργαλείου που χρησιμοποιεί.

2.2.1 Γενική Περιγραφή της Logo

Η Logo είναι μια γλώσσα προγραμματισμού γενικού σκοπού. Οι περισσότερες εκδόσεις της, βασίζονται σε διερμηνευτή. Γεγονός που διευκολύνει την εκμάθηση, αφού είναι δυνατό να εκτελούνται και να δοκιμάζονται άμεσα εντολές διαφόρων μεγεθών. Αυτό που την κάνει ξεχωριστή, είναι η απόκρυψη της έννοιας του

επεξεργαστή από τον προγραμματιστή. Ο χρήστης δε σκέφτεται ότι πρέπει να βάλει σε μια σειρά εντολές του επεξεργαστή, αλλά επικοινωνεί με μια «χελώνα». Η χελώνα της Logo βρίσκεται πάνω σε ένα επίπεδο και φέρει μια σφραγίδα. Η χελώνα γνωρίζει κάποιες απλές εντολές που αφορούν τον προσανατολισμό της (left, right), την κίνηση της (front, back), τα χαρακτηριστικά της γραφίδας (χρώμα, πάχος, υφή κλπ.). Όταν η χελώνα μετακινείται και έχει τη γραφίδα σε επαφή με το επίπεδο, τότε εμφανίζεται ένα ίχνος που μαρτυρά την πορεία της. Ο νεαρός χρήστης καλείται να σχεδιάσει γεωμετρικά και άλλα σχήματα καθοδηγώντας την χελώνα και έτσι εισάγεται στον προγραμματισμό.

Οι νεότερες εκδόσεις της Logo όπως η StarLogo, προσφέρουν δυνατότητες για εύκολη διαχείριση πολυμέσων (εικόνα, ήχος, βίντεο κλπ). Επίσης, μια ενδιαφέρουσα προσθήκη στην Logo, είναι η συνύπαρξη πολλαπλών «χελωνών» σε ένα πρόγραμμα. Η ικανότητα του προγραμματισμού πολλαπλών χελωνών, δίνει την δυνατότητα στον χρήστη να εισαχθεί στον παράλληλο προγραμματισμό.

Παρόλο που η LOGO χρησιμοποιείται τις περισσότερες φορές από παιδιά τα οποία εξερευνούν μόνο ένα μικρό μέρος της, αποτελεί μια πλήρως εξοπλισμένη γλώσσα προγραμματισμού. Σχεδόν οποιοδήποτε πρόγραμμα μπορεί να γραφτεί εξίσου καλά στη LOGO, όπως σε οποιοδήποτε άλλη γλώσσα και συχνά πιο γρήγορα και άνετα. Υπάρχουν κάποια ιδιαίτερα χαρακτηριστικά που την κάνουν να ξεχωρίζει από τις υπόλοιπες γλώσσες ως εκπαιδευτικά εργαλεία. Είναι άμεσα αλληλεπιδραστική, αφού οποιαδήποτε εντολή μπορεί να έχει άμεσα αποτελέσματα στην οθόνη, δίνοντας με αυτό τον τρόπο τη δυνατότητα στο χρήστη να αναλογίζεται κάθε φορά τις συνέπειες κάθε συμβολικής του ενέργειας, εφόσον τις βλέπει να συγκεκριμενοποιούνται στην οθόνη. Επίσης, είναι εύκολη στην εκμάθηση και αυτό συμβαίνει επειδή οι λέξεις απομνημονεύονται εύκολα και είναι εύχρηστες, οι εντολές της LOGO έχουν πολύ στενή σχέση με την αντίστοιχη λέξη που χρησιμοποιούμε για το ίδιο νόημα ή λειτουργία στην καθημερινή γλώσσα. Προωθεί καλές προγραμματιστικές τεχνικές, ενθαρρύνοντας την ανάπτυξη μικρών προγραμμάτων, οι διαδικασίες είναι επαναχρησιμοποιήσιμες για την ανάπτυξη άλλων διαδικασιών, δεν υπάρχουν γραμμές εντολών που ενθαρρύνουν μεταβάσεις μέσα στο πρόγραμμα, κάνει αυτόματο, δυναμικό καταμερισμό της μνήμης και δεν απαιτεί προσδιορισμό

του μεγέθους λέξεων, ή λιστών. Ακόμη, χαρακτηρίζεται σαν μια επεκτάσιμη γλώσσα και δίνει την δυνατότητα να προστεθούν νέες εντολές, να εκτελεσθούν και να δοκιμαστούν διαδικασίες, ανεξάρτητα η μία από την άλλη, ενθαρρύνει τη δημιουργία βιβλιοθήκης από συχνά χρησιμοποιούμενες διαδικασίες και επιτρέπει τη μετονομασία των εντολών της για την εύκολη κατανόησή τους, ή για προσομοίωση άλλων γλωσσών προγραμματισμού.

2.2.2Τα Οφέλη της Logo

Η LOGO αποτελεί την ιδανική λύση για την εισαγωγή στη διδασκαλία των γλωσσών προγραμματισμού, ως μέσο επίλυσης προβλημάτων με υπολογιστή, για την ανάπτυξη εκπαιδευτικών πληροφορικών περιβαλλόντων από τους μαθητές. Είναι πολλά τα θετικά της στοιχεία που έχουν αντίκρισμα στην πορεία της εκπαίδευσης των μαθητών. Το παιδί, προγραμματίζοντας τον υπολογιστή, του δίνει εντολές για να του «μάθει» τι πρέπει να κάνει για να φτιάξει ένα γεωμετρικό σχήμα, π.χ. ένα τετράγωνο. Μέσα από αυτή τη διαδικασία και με τη βοήθεια του δασκάλου του, ανακαλύπτει και το δικό του τρόπο σκέψης.

Ο προγραμματισμός απαιτεί ένα είδος αυστηρής σκέψης με απόλυτα συγκεκριμένη μορφή. Δεν υπάρχουν υπονοούμενα ή αοριστίες όταν δίνουμε εντολές στη χελώνα για το πώς να κινηθεί. Αν υπάρξουν, εμφανίζονται μηνύματα λάθους ή οι κινήσεις τις χελώνας δεν είναι οι προσδοκώμενες. Επομένως, μέσα από τη διαδικασία αυτή το παιδί μαθαίνει να μιλάει με ακρίβεια και σαφήνεια, αναπτύσσει δεξιότητες επικοινωνίας.

Τους βοηθάει να κατανοήσουν τη γνώση που πρέπει να διδάσκεται, να σχεδιάσουν μια προσέγγιση για να μεταδώσουν αυτή τη γνώση και να τη χωρίσουν σε μικρά, κατανοητά κομμάτια. Με αυτόν τον τρόπο, τα παιδιά μαθαίνουν γενικότερα ότι ένα πρόβλημα του οποίου η λύση αρχικά φαίνεται δύσκολη, μπορεί να γίνει εύκολη αν το σπάσουν σε κομμάτια.

Το γεγονός ότι για το φτιάξιμο ενός σχήματος με τη χελώνα δεν υπάρχει μόνο ένας τρόπος, μαθαίνει στα παιδιά ότι και για τα υπόλοιπα προβλήματα, είτε στα μαθητικά είτε στην καθημερινή ζωή, δεν υπάρχει μόνο μία λύση κι ότι δεν είναι η μία λύση σωστή ή πιο σωστή και οι υπόλοιπες λάθος. Τα παιδιά αποκτούν έτσι ευελιξία

στη σκέψη. Μαθαίνουν επίσης να αποδέχονται τη διαφορετικότητα του τρόπου με τον οποίο σκέπτονται οι άλλοι και να αποδέχονται τη διαφορετικότητα εν γένει.

Το γεγονός ότι τα παιδιά κάνουν λάθη, τα οποία στη συνέχεια εκμεταλλεύονται (διατρέχουν τις εντολές που έδωσαν για να βρουν πού είναι το λάθος), τους δίνει την ευκαιρία αφενός να εκτιμήσουν την αξία του λάθους και αφετέρου να είναι επιεικείς με τους άλλους.

Η ορολογία του προγραμματισμού και η ανοιχτή συζήτηση που χρειάζεται να γίνεται κατά τη διάρκεια του προγραμματισμού, δίνουν την ευκαιρία ανάπτυξης αναστοχαστικών δεξιοτήτων, της λεγόμενης μεταγνώσης. Τα παιδιά δεν επικεντρώνονται μόνο στο πρόβλημα αλλά και στη διαδικασία της επίλυσης του προβλήματος. Αναπτύσσεται η αυτογνωσία τους σχετικά με τις προσωπικές γνωστικές τους διεργασίες. Μαθαίνουν πώς μαθαίνουν.

Ο προγραμματισμός παρέχει στο παιδί ένα περιβάλλον μέσα στο οποίο μπορεί να χρησιμοποιήσει γενικές έννοιες, όπως εκείνες του μετασχηματισμού, της συνάρτησης και της μεταβλητής και να παρατηρήσει τις συνέπειες των εφαρμογών του. Είναι αυτό που λέγεται για τη Logo ότι η γλώσσα αυτή κάνει το αφηρημένο ορατό.

2.3 Συστήματα Οπτικοποίησης Αλγορίθμων: Γενικά

Οι εφαρμογές οπτικοποίησης αλγορίθμων έχουν μακρά ιστορία στην εκπαίδευση. Το βίντεο Sorting Out Sorting (Baecker, 1981), το οποίο παρουσίαζε εικόνες δεδομένων που ταξινομούνταν από διαφορετικούς αλγορίθμους θεωρείται η πρώτη σημαντική αναφορά στην οπτικοποίηση αλγορίθμων. Από τότε έχουν αναπτυχθεί πολλά συστήματα προσομοίωσης και οπτικοποίησης αλγορίθμων που βασίζονταν στις διαθέσιμες κάθε φορά τεχνολογίες. Με τον όρο οπτικοποίηση (visualization) αλγορίθμου περιγράφεται μια διαδραστική οπτικοποιημένη παρουσίαση της λογικής που κρύβεται πίσω από τον αλγόριθμο, βασισμένη σε μια σειρά εικόνων και αναπαραστάσεων που εστιάζουν στα βασικά χαρακτηριστικά της συμπεριφοράς του. Τα συνήθη λογισμικά προσομοίωσης-οπτικοποίησης δεν αντικαθιστούν το προγραμματιστικό περιβάλλον, αφού δεν εκτελούν κάποιον

αλγόριθμο, αλλά παρουσιάζουν, οπτικοποιούν και προσομοιώνουν την εκτέλεσή του για προκαθορισμένα δεδομένα εισόδου.

Οι λόγοι που συνηγορούν στη χρήση των προσομοιώσεων- οπτικοποιήσεων στα εισαγωγικά μαθήματα του προγραμματισμού είναι οι εξής:

- επιτρέπουν την οπτικοποίηση της λειτουργίας ενός αλγορίθμου, υποστηρίζοντας τη διάλεξη του εκπαιδευτικού και την εργαστηριακή εξάσκηση των μαθητών
- επιτρέπουν στους μαθητές να απομονώσουν τις μεταβλητές και το ρόλο τους στο πρόγραμμα, με στόχο την κατανόηση πιο σύνθετων υπολογιστικών δομών και διαδικασιών
- βοηθούν τους μαθητές να εμβαθύνουν στη λογική του προγραμματισμού και να κατανοήσουν δύσκολες υπολογιστικές έννοιες και διαδικασίες
- ενεργοποιούν το ενδιαφέρον των μαθητών, παρέχοντας δυνατότητες να εκφράσουν τις δικές τους αναπαραστάσεις για αλγορίθμους και διαδικασίες
- διευκολύνουν την ενεργητική μάθηση μέσα από διαδικασίες παράθεσης υποθέσεων, μεταβολής των τιμών εισόδου και άμεσου ελέγχου των αποτελεσμάτων στην οθόνη
- βοηθούν τους μαθητές να μάθουν μέσα από διερευνητικές δραστηριότητες.

Η τυπική χρήση λογισμικών προσομοίωσης-οπτικοποίησης αλγορίθμων δεν είναι περισσότερο αποτελεσματική σε σχέση με την παραδοσιακή διδασκαλία. Πρόσφατες έρευνες έχουν δείξει ότι τα λογισμικά προσομοίωσης αλγορίθμων θα πρέπει να προωθούν τη διερευνητική μάθηση και να ευνοούν την ενεργητική συμμετοχή και όχι την παθητική συμμόρφωση των μαθητών. Ο μαθητής δεν αρκεί απλά να παρακολουθεί την οπτικοποίηση ως παθητικός θεατής αλλά θα πρέπει να πειραματιστεί με την εκτέλεση του αλγορίθμου, έτσι ώστε να διερευνήσει τις διάφορες λογικές πτυχές του.

Με βάση τον τύπο τους, διακρίνουμε τα εκπαιδευτικά περιβάλλοντα οπτικοποίησης σε δύο μεγάλες κατηγορίες: συστήματα οπτικοποίησης αλγορίθμων και συστήματα οπτικοποίησης προγραμμάτων.

2.3.1 Συστήματα Οπτικοποίησης Αλγορίθμων

Τα συστήματα οπτικοποίησης αλγορίθμων είναι περιβάλλοντα τα οποία παράγουν μια γραφική αναπαράσταση του αλγορίθμου με σκοπό να αναδειχθούν τα ιδιαίτερα χαρακτηριστικά του, τα οποία είναι δύσκολο να οικοδομήσουν ή να προσεγγίσουν οι μαθητευόμενοι. Διακρίνονται σε:

- Συστήματα με γλώσσα συγγραφής σεναρίων

Είναι η πιο δημοφιλής κατηγορία συστημάτων οπτικοποίησης αλγορίθμων. Ο χρήστης έχει τη δυνατότητα να δημιουργήσει τη δική του οπτικοποίηση του αλγορίθμου, αφού εισάγει στον κώδικά του εντολές οπτικοποίησης μέσω μιας γλώσσας σεναρίων (script language). Αυτό δεν είναι κάτι που μπορεί να γίνει εύκολα από έναν αρχάριο προγραμματιστή ή έναν μαθητή που παρακολουθεί τα εισαγωγικά μαθήματα αλγοριθμικής, καθώς θα πρέπει να μάθει μια ακόμη γλώσσα προγραμματισμού, τη γλώσσα σεναρίων του συστήματος. Τα συστήματα οπτικοποίησης-προσομοίωσης αλγορίθμων μπορούν να συμβάλλουν στην υποστήριξη των διαλέξεων του διδάσκοντα, ειδικά σε φοιτητές που έχουν παρακολουθήσει αρκετά μαθήματα προγραμματισμού. Τα πρώτα συστήματα αυτής της κατηγορίας ανήκουν στην οικογένεια των Tango, Polka, Samba και JSamba που αναπτύχθηκαν από την ομάδα του Stasko (1997). Ένα άλλο σύστημα που παρουσιάζει ομοιότητες με το JSamba είναι το JAWAA (Pierson & Rodger, 1998) που οπτικοποιεί δομές δεδομένων όπως είναι η στοίβα, η ουρά κ.α. Νεότερα συστήματα που ακολουθούν την ίδια τεχνική είναι το JHave (Naps, 2005), το Animal (Röbling 2000) και το Alvis (Hundhausen & Brown, 2007), το οποίο είναι ένα από τα καλύτερα συστήματα της κατηγορίας αυτής και χαρακτηρίζεται από τον υψηλό βαθμό αλληλεπίδρασης με τον χρήστη.

- Συστήματα με βιβλιοθήκες γραφικών

Ανάλογα μειονεκτήματα έχουν επίσης και τα συστήματα που χρησιμοποιούν βιβλιοθήκες γραφικών για την οπτικοποίηση του αλγορίθμου. Ο μαθητής/φοιτητής

είναι υποχρεωμένος να εισάγει τις κατάλληλες εντολές στον κώδικά του, ώστε να παρατηρήσει την οπτικοποίηση του αλγορίθμου που σχεδίασε. Χαρακτηριστικά συστήματα του τύπου αυτού είναι το Zeus (Brown, 1991) και το Algorithm Explorer (Carson et al., 2007).

2.3.2 Συστήματα Οπτικοποίησης Προγραμμάτων

Τα συστήματα οπτικοποίησης προγραμμάτων χρησιμοποιούν αναπαραστάσεις και διαδικασίες που είναι χρήσιμες στους προγραμματιστές κατά την εκσφαλμάτωση (debugging) του προγράμματος. Για παράδειγμα, γίνεται παρακολούθηση των τιμών των μεταβλητών σε κάθε βήμα του αλγορίθμου με στόχο την ανίχνευση και τη διόρθωση των λαθών κατά την εκτέλεση του προγράμματος.

Χαρακτηριστικός εκπρόσωπος της κατηγορίας αυτής, παρότι ενσωματώνει αρκετά στοιχεία οπτικοποίησης δομών δεδομένων, είναι το σύστημα Leonardo (Demetrescu & Finocchi, 1999) που αναπτύχθηκε στο Πανεπιστήμιο της Ρώμης. Η οπτικοποίηση του προγράμματος γίνεται χρησιμοποιώντας την γλώσσα συγγραφής σεναρίων Alpha. Ένα άλλο περιβάλλον προσομοίωσης είναι το PlanAni. Το λογισμικό αυτό χρησιμοποιεί διαφορετικές αναπαραστάσεις για τις διάφορες μεταβλητές, οι οποίες σχηματίζονται κατά τη φάση της δήλωσής τους και ανάλογα με το ρόλο τους στο πρόγραμμα. Το PlanAni έχει και ελληνική έκδοση, η οποία έχει εφαρμοστεί πιλοτικά για την κατανόηση της έννοιας της προγραμματιστικής μεταβλητής και της εντολής εκχώρησης. Τέλος, η πιο δημοφιλής εφαρμογή της κατηγορίας αυτής είναι το Jeliot. Το σημαντικότερο χαρακτηριστικό της είναι η αυτόματη οπτικοποίηση των προγραμμάτων σε Java, η οποία όμως επικεντρώνεται στις αντικειμενοστραφείς δομές του προγράμματος και όχι στη συμπεριφορά-λογική του αλγορίθμου.

2.3.3 Δυναμική Οπτικοποίηση Αλγορίθμων

Οι ερευνητικές μελέτες σχετικά με την αποτελεσματικότητα των συστημάτων οπτικοποίησης-προσομοίωσης αλγορίθμων έδειξαν ότι τα μαθησιακά οφέλη είναι περιορισμένα όταν ο μαθητής παρακολουθεί απλά μια οπτικοποίηση χωρίς να εμπλέκεται σε γνωστικές διαδικασίες χειρισμού και αλληλεπίδρασης με τον

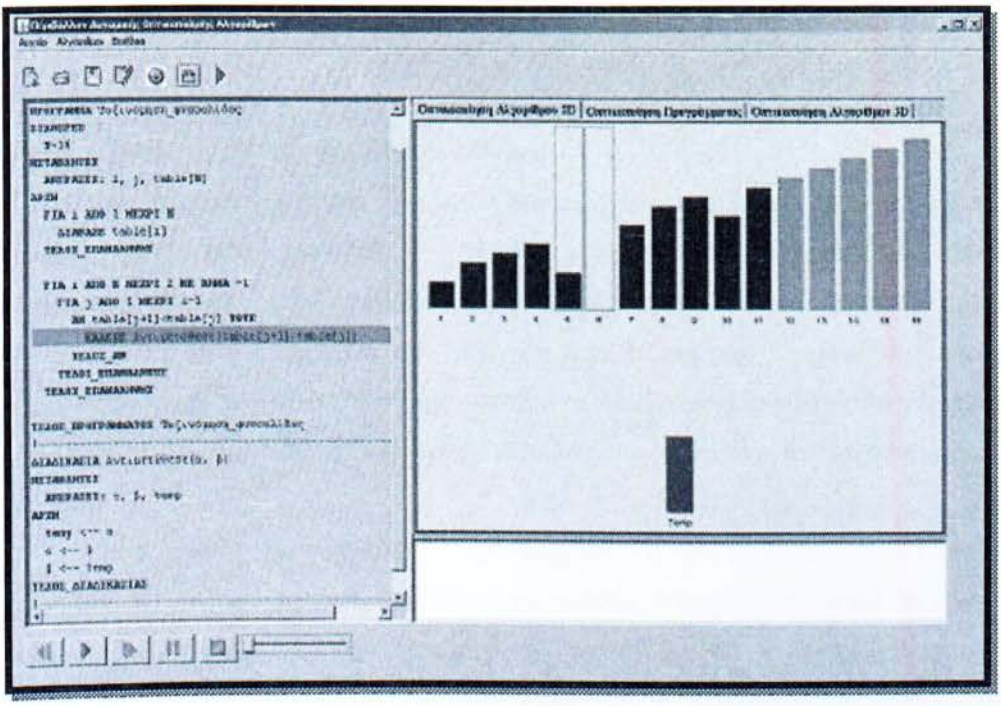
αλγόριθμο. Ο Hundhausen (2002), μετά από την ανάλυση των αποτελεσμάτων 24 πειραματικών διδασκαλιών, κατέληξε στο συμπέρασμα ότι 'όσο πιο ενεργά συμμετέχουν οι μαθητές σε δραστηριότητες με το εκπαιδευτικό λογισμικό, τόσο καλύτερα αποδίδουν αργότερα, όταν καλούνται να υλοποιήσουν τον αλγόριθμο που διδάχθηκαν'.

Στα περισσότερα συστήματα ο μαθητής έχει περιορισμένες δυνατότητες παρέμβασης στην οπτικοποίηση του αλγορίθμου, ορίζοντας απλά τα δεδομένα εισόδου, σταματώντας την εκτέλεση ή μεταβαίνοντας σε όποιο σημείο της εκτέλεσης επιθυμεί. Κανένα από τα εργαλεία αυτά δεν έχει τη δυνατότητα να οπτικοποιεί αυτόματα έναν αλγόριθμο που γράφεται από τον ίδιο το μαθητή σε μια δεδομένη γλώσσα προγραμματισμού.

Απαιτείται, συνεπώς, ένα σύστημα με υψηλό βαθμό αλληλεπίδρασης, το οποίο να ενσωματώνει δυνατότητες αυτοματοποιημένης παραγωγής της οπτικοποίησης του κώδικα του μαθητή. Η δυνατότητα αυτή ενισχύει σημαντικά τη μαθησιακή διαδικασία, καθώς προσθέτει μια άλλη διάσταση στην αλληλεπίδραση του μαθητή με το λογισμικό, όπως για παράδειγμα συμβαίνει με τη γλώσσα προγραμματισμού Logo. Ο μαθητής βλέπει τη γραφική αναπαράσταση των βασικών χαρακτηριστικών της συμπεριφοράς του αλγορίθμου που έχει υλοποιήσει, ενώ παράλληλα έχει τη δυνατότητα να ανιχνεύσει άμεσα και να διορθώσει τα λογικά λάθη του. Επιπλέον, έχει τη δυνατότητα να λειτουργήσει διερευνητικά και να λάβει ανάδραση από το ίδιο το σύστημα ελέγχοντας την οπτικοποίηση του υπό μελέτη αλγορίθμου, ανακαλύπτοντας σημαντικά χαρακτηριστικά της δομής και της συμπεριφοράς του, μελετώντας τον τρόπο που ανταποκρίνεται ο αλγόριθμος στις αλλαγές που δοκιμάζει κ.ο.κ.. Αυτό ενισχύεται ακόμη περισσότερο όταν βλέπει στην ίδια οθόνη τον κώδικα του αλγορίθμου και την οπτικοποίησή του, έτσι ώστε να είναι εύκολη η αντιστοίχηση της εκτελούμενης εντολής με το αποτέλεσμα της οπτικοποίησης.

Σε αυτή την κατηγορία ανήκει το λογισμικό δυναμικής οπτικοποίησης αλγορίθμων DAVE, το οποίο σχεδιάστηκε για να καλύψει τις παραπάνω απαιτήσεις. Βασικός στόχος είναι όχι απλά η εμπλοκή και ο πειραματισμός του μαθητή με προκαθορισμένους αλγορίθμους αλλά, κυρίως, η αυτόματη οπτικοποίηση

αλγορίθμων που έχει σχεδιάσει ο ίδιος. Το λογισμικό DAVE παρέχει τη δυνατότητα στο μαθητή να συσχετίσει κάθε εντολή-δομή του κώδικα, που έχει αναπτύξει ο ίδιος, με κατάλληλες γραφικές αναπαραστάσεις δεδομένων έτσι ώστε να αναδεικνύονται τα ειδικά χαρακτηριστικά κάθε αλγορίθμου. Με βάση την προσέγγιση αυτή, ο μαθητής βρίσκεται σε άμεση και συνεχή επαφή με τον κώδικα του αλγορίθμου, καθώς κάθε αλλαγή στον κώδικα έχει άμεσο αντίκτυπο στην οπτικοποίηση. Για να δημιουργήσει μια νέα οπτικοποίηση αρκεί να γράψει τον κώδικα του υπό μελέτη αλγορίθμου στη γλώσσα προγραμματισμού και δεν χρειάζεται να μάθει μια νέα γλώσσα σεναρίων ή να χειριστεί βιβλιοθήκες γραφικών.



Σχήμα 1: Οθόνη του λογισμικού DAVE με τον αλγόριθμο ταξινόμησης φυσαλίδας

2.4 Περιβάλλοντα Εισαγωγής στον Προγραμματισμό

Τα περιβάλλοντα προγραμματισμού που έχουν σχεδιαστεί με στόχο την διδασκαλία του προγραμματισμού είναι αναμφίβολα λιγότερα και λιγότερο ώριμα από τα αντίστοιχα για το διαδικασιακό παράδειγμα προγραμματισμού (procedural

programming paradigm) (Kölling and Rosenberg, 2001), το οποίο έχει διδαχθεί και ερευνηθεί η διδασκαλία και η εκμάθηση του για πολύ περισσότερο χρόνο.

Στην ενότητα αυτή θα παραθέσουμε τις πιο διαδεδομένες κατηγορίες εκπαιδευτικών εργαλείων και προγραμματιστικών περιβαλλόντων που έχουν σχεδιαστεί για την διδασκαλία του προγραμματισμού. Αυτές είναι:

- προγραμματιστικοί μικρόκοσμοι
- περιβάλλοντα εκπαιδευτικής ρομποτικής
- περιβάλλοντα που δίνουν έμφαση στη δυναμική προσομοίωση εκτέλεσης των προγραμμάτων
- περιβάλλοντα που στοχεύουν στην αναπαράσταση της λύσης ενός προβλήματος με πολλαπλές μορφές

2.4.1 Προγραμματιστικοί Μικρόκοσμοι

Με την εισαγωγή των γραφικών δυνατοτήτων στη Logo, εμφανίστηκε η έννοια του μικρόκοσμου από την ομάδα μάθησης και κοινής λογικής του εργαστηρίου μέσων του MIT (MIT Media Lab Learning and Common Sense Group). Μικρόκοσμος στην Logo, είναι ένα σύστημα από έτοιμα προγράμματα και μέσα (εικόνες κλπ) με το οποίο ο μαθητής μπορεί να διερευνήσει επιλογές, να ελέγξει υποθέσεις και να ανακαλύψει δεδομένα που ισχύουν μόνο για τον συγκεκριμένο κόσμο.

Ένας μικρόκοσμος είναι ένα μικρό κομμάτι της πραγματικότητας. Ο Papert αναφέρει ότι είναι αυστηρά περιορισμένος και πλήρως ορισμένος, αλλά και ότι είναι πλούσιος. Οι μικρόκοσμοι έχουν δημιουργηθεί και σχεδιαστεί σαν ασφαλείς χώροι για εξερεύνηση. Μπορείς να δοκιμάσεις τα πάντα.

Οι βασικές προδιαγραφές που διέπουν έναν υπολογιστικό μικρόκοσμο είναι οι ακόλουθες:

- Ένας μικρόκοσμος πρέπει να διαθέτει ένα σύνολο από υπολογιστικά αντικείμενα τα οποία μοντελοποιούν τις μαθηματικές, φυσικές ή επιστημονικές ιδιότητες του χώρου στον οποίο αντιστοιχεί ο μικρόκοσμος καθώς και συνδέσεις σε πολλαπλού τύπου αναπαραστάσεις των υποκείμενων ιδιοτήτων των αντικειμένων ή των μοντέλων του.

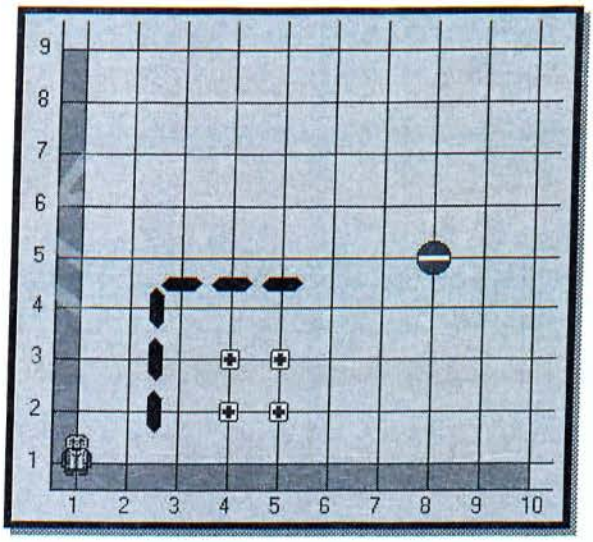
- Ένας μικρόκοσμος πρέπει να επιτρέπει να συνδυάζονται αντικείμενα ή τελεστές ώστε να δημιουργούνται πιο σύνθετα αντικείμενα, όπως κατασκευάζονται οι φράσεις από τις λέξεις μιας γλώσσας.
- Ένας μικρόκοσμος πρέπει να διαθέτει ένα σύνολο από δραστηριότητες που ενθαρρύνουν το μαθητή να χρησιμοποιήσει τα αντικείμενα και τους τελεστές του για να λύσει ένα πρόβλημα, να διερευνήσει μία κατάσταση ή να πετύχει ένα στόχο.

Οι προγραμματιστικοί μικρόκοσμοι (programming microworlds), έχουν αναπτυχθεί για καθαρά εκπαιδευτικούς σκοπούς, βασίζονται σε μια φυσική μεταφορά (metaphor) (χελώνα, ρομπότ, καγκουρό, τρισδιάστατα αντικείμενα σε εικονικούς κόσμους) – δηλαδή πρωταγωνιστές που "ζουν" μέσα στο περιβάλλον του μικρόκοσμου–, είναι εύχρηστοι, ενσωματώνουν δυνατότητες οπτικοποίησης και κίνησης (animation), συνήθως χρησιμοποιούν μια εκπαιδευτική γλώσσα προγραμματισμού και παρέχουν τη δυνατότητα της δυναμικής προσομοίωσης της εκτέλεσης των προγραμμάτων, δηλαδή της βήμα προς βήμα εκτέλεσης τους και συγχρόνως απεικόνιση του αποτελέσματος της εκτέλεσης στην κατάσταση του μικρόκοσμου. Στην κατηγορία αυτή, οι μικρόκοσμοι που έχουν αναπτυχθεί για την εισαγωγή στον προγραμματισμό είναι οι Karel the Robot in Java, KarelJ.Robot, JKarelRobot, Jeroo, Alice, objectKarel, Kara, Microworlds Logo, Squeak και πολλοί ακόμα, τους οποίους θα δούμε στη συνέχεια.

- Οικογένεια εργαλείων βασισμένα στο Karel the Robot

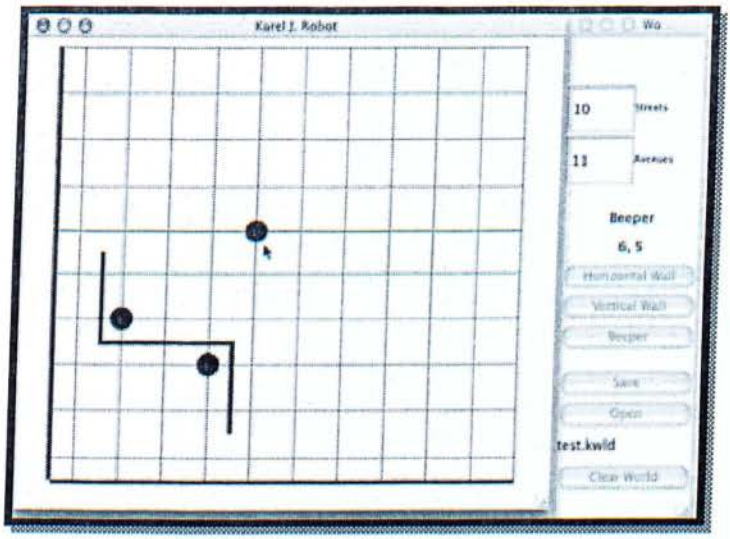
Ο Karel, αναπτύχθηκε στο Πανεπιστήμιο του Waterloo στο Ontario του Καναδά, στο τμήμα της Επιστήμης των Υπολογιστών, είναι ένα προγραμματιζόμενο ρομπότ, δυνατότητες ελαφρώς περισσότερες από αυτές της χελώνας, το οποίο ζει σε έναν κόσμο δυο διαστάσεων που αποτελείται από δρόμους (streets), λεωφόρους (avenues), beepers και εμπόδια (οριζόντια και κάθετα). Αποτελεί έναν ευρύτατα χρησιμοποιούμενο μικρόκοσμο που σχεδιάστηκε για την εισαγωγή των νέων χρηστών στον προγραμματισμό. Το πρώτο Karel (1985) έχει ως στόχο την εκμάθηση της γλώσσας Pascal. Ο προγραμματιστής που χρησιμοποιεί το δεδομένο περιβάλλον, όπως κα στη Logo, χρησιμοποιεί τον προγραμματισμό για να μάθει σε μια τεχνητή-νοητή οντότητα να κάνει κάτι ή να λύνει ένα πρόβλημα. Ο προγραμματιστής δεν

ελέγχει άμεσα τον ίδιο τον υπολογιστή ή έστω την αφαιρετική του αναπαράσταση που περιέχει το λειτουργικό σύστημα αλλά ένα φανταστικό αντικείμενο και τον κόσμο του.

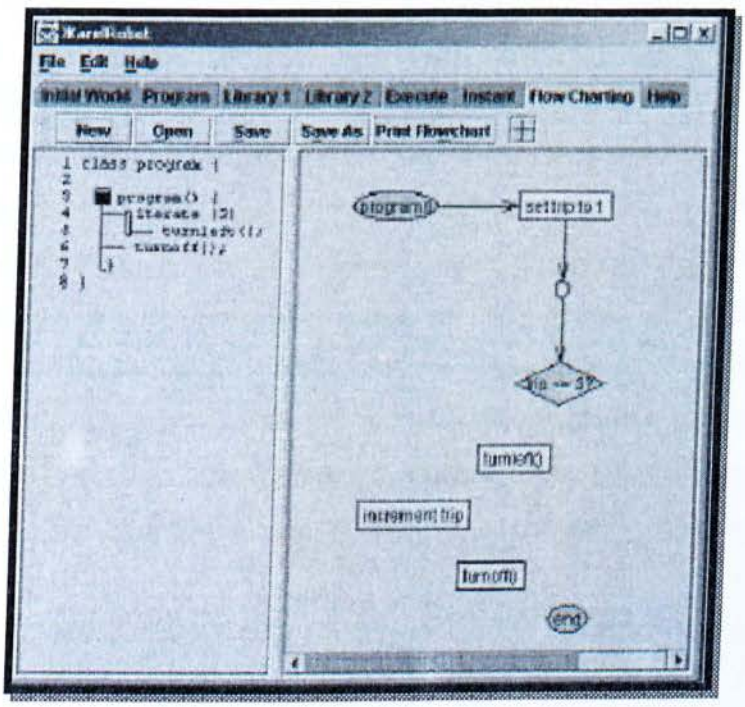


Σχήμα 2: Ο κόσμος του Karel

Ως φυσική εξέλιξη του Karel εμφανίστηκαν παρόμοια περιβάλλοντα για να καλύψουν τις ανάγκες νέων υποδειγμάτων προγραμματισμού που διαδέχθηκαν τον δομημένο διαδικαστικό προγραμματισμό που πρεσβεύει η Pascal. Οπότε έχουμε τα περιβάλλοντα Karel++, το οποίο εισάγει τον αρχάριο προγραμματιστή στην ιδέα του αντικειμενοστραφούς προγραμματισμού και βασίστηκε σε μία σύνταξη της γλώσσας προγραμματισμού που προσομοιάζει την C++ και την Java. Ο KarelJRobot αποτελεί την μετάφραση που ο Bergin έκανε στο μικρόκοσμο Karel++ σε αμιγώς γλώσσα Java. Ο JKarelRobot είναι ένας μικρόκοσμος γραμμένος σε Java, ο οποίος υποστηρίζει διαγράμματα ροής και τη δυνατότητα διαγραμματικής παρουσίασης του προγράμματος με τη χρήση διαγραμμάτων δομής. Αναπτύχθηκε από τους Buck και Stucki και χρησιμοποιήθηκε στο Otterbein College.



Σχήμα 3: Περιβάλλον KarelJRobot



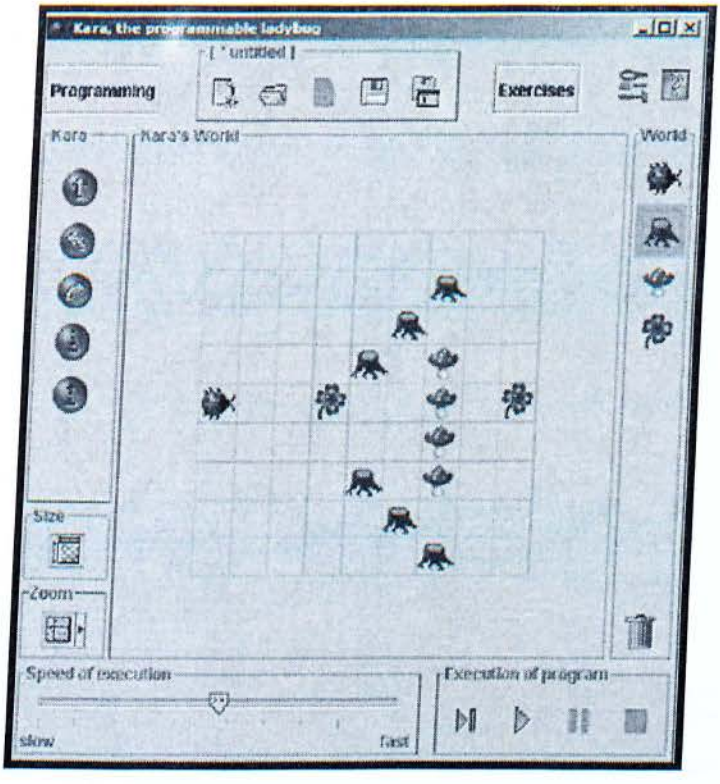
Σχήμα 4: JKarelRobot Flow Chart

- Τα περιβάλλοντα Kara

Πρόκειται για περιβάλλοντα, στα οποία χρησιμοποιείται η θεωρία υπολογισμού ως όχημα για την διδασκαλία βασικών εννοιών του προγραμματισμού και της Πληροφορικής . Τα περιβάλλοντα Kara προσφέρουν:

- μια εισαγωγή στον προγραμματισμό βασισμένη στις μηχανές πεπερασμένων καταστάσεων για άτομα χωρίς προηγούμενη προγραμματιστική εμπειρία (Kara-PFSM)
- μια εισαγωγή στο προγραμματισμό σε Java για άτομα χωρίς προηγούμενη προγραμματιστική εμπειρία (περιβάλλον JavaKara)
- μια εισαγωγή σε βασικές έννοιες ταυτόχρονου προγραμματισμού (περιβάλλον MultiKara)
- μια εισαγωγή στη θεωρία υπολογισμού βασισμένη στις δισδιάστατες μηχανές Turing (περιβάλλον TuringKara)

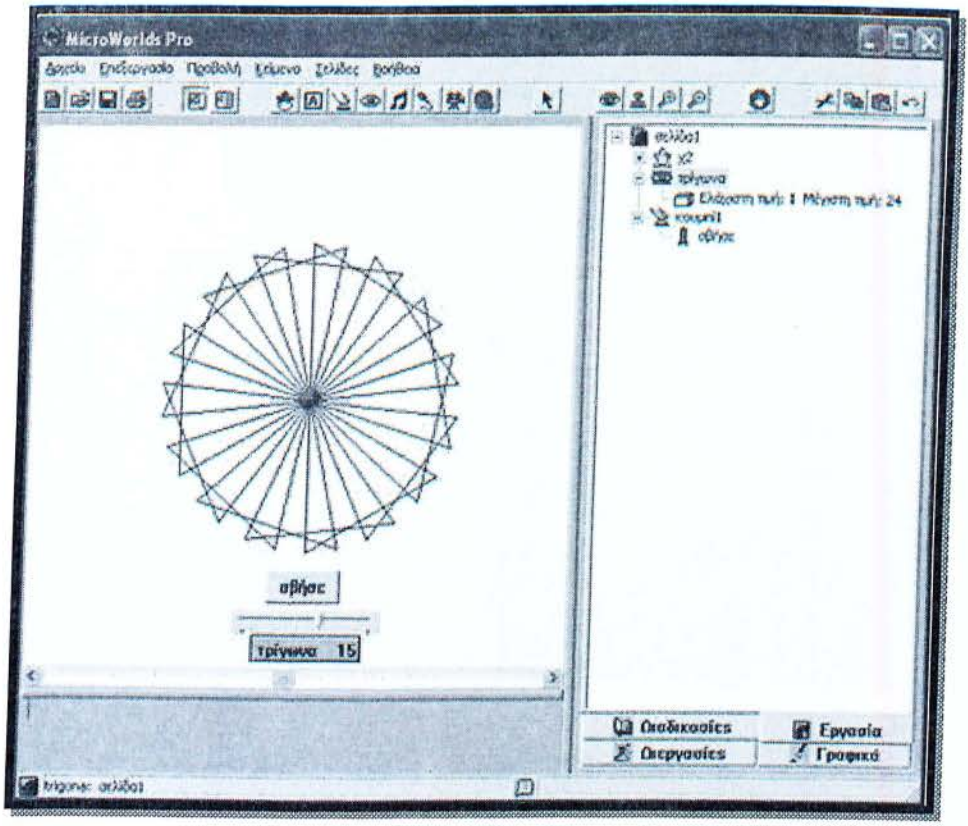
Τα περιβάλλοντα Kara-PFSM και JavaKara παρουσιάζουν ιδιαίτερο ενδιαφέρον λόγω της χρησιμότητας τους στην σχολική εκπαίδευση. Ο βασικός πρωταγωνιστής είναι ένα έντομο που κινείται επί ενός δισδιάστατου πλέγματος που έχει τη μορφή σκακιέρας. Το έντομο μπορεί να προχωρήσει ένα τετράγωνο προς τα εμπρός κατά μήκος της τρέχουσας διεύθυνσης, να στρίψει δεξιά ή αριστερά κατά 90° επί του τετραγώνου που βρίσκεται, να σπρώξει προς τα εμπρός κατά ένα τετράγωνο ένα μανιτάρι εφόσον αυτό βρεθεί στην πορεία του, να ρίξει ένα φύλλο στο έδαφος ή να μαζέψει ένα φύλλο από το έδαφος επί του τετραγώνου στο οποίο βρίσκεται. Χρησιμοποιώντας τα κατάλληλα πλαίσια ελέγχου, ο μαθητής έχει τη δυνατότητα με τεχνικές drag and drop να τοποθετήσει στο πλέγμα όσους κορμούς, μανιτάρια και φύλλα επιθυμεί, με εξαίρεση το ίδιο το έντομο για το οποίο υπάρχει μόνον ένα στιγμιότυπο.



Σχήμα 5: Η κεντρική οθόνη του περιβάλλοντος Kara-PFSM

- Microworlds Pro

Το Microworlds Pro είναι ένα περιβάλλον γενικής χρήσης που καλλιεργεί σύνθετες δεξιότητες και μαθησιακές τάσεις και επιτρέπει τη διερεύνηση-επανάληψη-αξιολόγηση δύσκολων εννοιών. Βασίζεται στη γλώσσα Logo και εξαρχής σχεδιάστηκε για την εκπαίδευση. Είναι ένα πλούσιο πολυμεσικό περιβάλλον με ειδικά μελετημένο σχεδιασμό, που ευνοεί με πολλούς τρόπους την ανάπτυξη συνθετικών εργασιών στο πλαίσιο πολλών μαθημάτων. Το περιβάλλον μπορεί να αξιοποιηθεί από διάφορες βαθμίδες της εκπαίδευσης, μια που η διαχείριση των προγραμματιζόμενων αντικειμένων είναι δυνατό να γίνει και με απτικό τρόπο, με τα προβλεπόμενα εργαλεία τα οποία μπορεί κανείς να χειριστεί με το ποντίκι.



Σχήμα 6: Το περιβάλλον Microworlds Pro

- Stagecast Creator

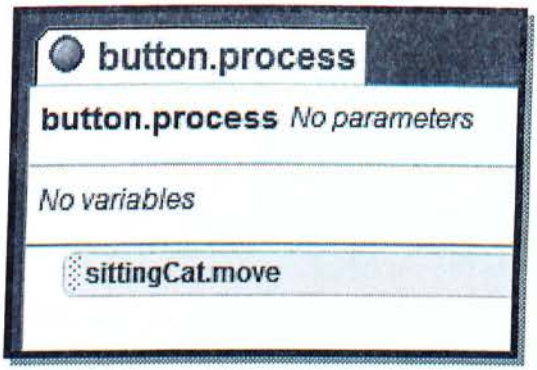
Το Stagecast Creator (Smith & Cypher, 1999) είναι ένα προγραμματιστικό περιβάλλον σχεδιασμένο για νεαρούς προγραμματιστές. Χρησιμοποιεί ως βάση έναν μικρόκοσμο μέσα στον οποίο οι χαρακτήρες αλληλεπιδρούν σύμφωνα με τους κανόνες που έχει γράψει ο χρήστης. Για την εισαγωγή των κανόνων δεν χρειάζεται η χρησιμοποίηση του πληκτρολογίου. Όταν εκτελείται το πρόγραμμα το Stagecast Creator προσπαθεί να ταιριάξει τον κόσμο γύρω από ένα αντικείμενο με τον όρο ενός κανόνα. Όταν βρεθεί μια τέτοια αντιστοιχία, ο κόσμος γύρω από τον χαρακτήρα αλλάζει όπως καθορίζεται από τον κανόνα. Σκοπός του εργαλείου είναι να εισάγει τον αρχάριο προγραμματιστή σε βασικές έννοιες όπως οι μεταβλητές, η επανάληψη, η υπό όρους εκτέλεση μιας εντολής και οι διαδικασίες. Ταυτόχρονα αποτελεί έναν ιδανικό χώρο για την κατασκευή μοντέλων στη φυσική και σε άλλες θετικές επιστήμες.

Σχήμα 7: Το περιβάλλον του Stagecast Creator

- Alice

Στο περιβάλλον Alice προγραμματίζεται ένας ιδεατός κόσμος όπου "κατοικούν" αντικείμενα με τη μορφή τρισδιάστατων μοντέλων. Τα αντικείμενα αυτά έχουν προγραμματιζόμενη συμπεριφορά. Η Alice μοιράζεται πολλά χαρακτηριστικά με τις συνήθεις γλώσσες αντικειμενοστραφούς προγραμματισμού και έτσι το κάθε αντικείμενο έχει ιδιότητες και μεθόδους (Cooper et al., 2000).

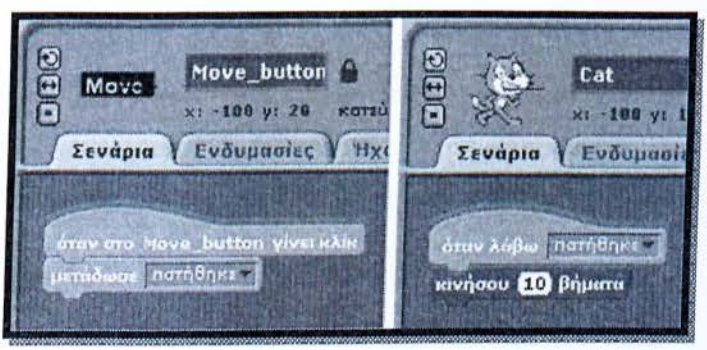
Στο Alice μια μέθοδος μπορεί να καλέσει μια μέθοδο ενός άλλου αντικειμένου. Αν και η υλοποίηση γίνεται με γραφικά, σέρνοντας δηλαδή το αντίστοιχο μπλοκ μέσα στο σώμα της μεθόδου, το αντικείμενο στο οποίο αναφέρεται η κάθε μέθοδος διαχωρίζεται από τη μέθοδο με τον τελεστή της τελείας που είναι συνηθισμένος στις αντικειμενοστραφείς γλώσσες προγραμματισμού. Στο Σχήμα 8 φαίνεται με ποιο τρόπο μπορεί το αντικείμενο button να καλέσει μια μέθοδο του αντικειμένου sittingCat.



Σχήμα 8: Η μέθοδος process του αντικειμένου button καλεί τη μέθοδο move του αντικειμένου sittingCat μέσω του τελεστή της τελείας "."

- Scratch

Στη Scratch ο μαθητής προγραμματίζει τη συμπεριφορά του σκηνικού και των μορφών που αποτελούν την εφαρμογή του. Ο χειρισμός των πολυμέσων είναι διαισθητικός και υπάρχουν εντολές που αλλάζουν την εμφάνιση των μορφών η αναπαράγουν κάποιον ήχο. Εξωτερικές εικόνες και ήχοι μπορούν να εισαχθούν στις εφαρμογές και ο προγραμματιστής ελέγχει τον τρόπο με τον οποίο εμφανίζονται στην κατασκευή του (Resnick et al., 2009). Ο άμεσος τρόπος συγχρονισμού μεταξύ των αντικειμένων είναι τα μηνύματα (Νικολός & Κόμης, 2010). Στο Σχήμα 9 όταν το «κουμπί» πατηθεί στέλνει ένα μήνυμα που η «γάτα» το λαμβάνει και κινείται. Το μήνυμα δεν απευθύνεται στη «γάτα» αλλά γίνεται εκπομπή του μηνύματος και όλα τα αντικείμενα της Scratch μπορούν να το λάβουν.

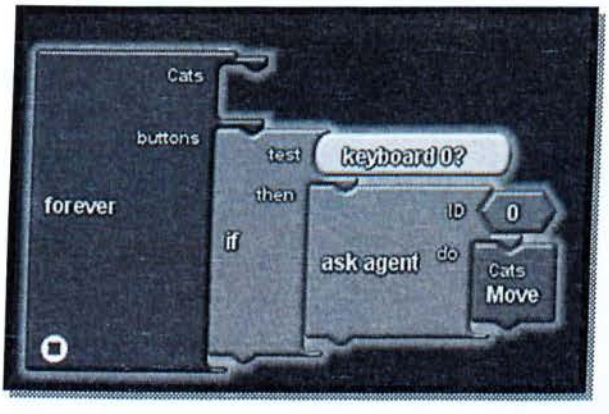


Σχήμα 9: Επικοινωνία αντικειμένων μέσω μηνυμάτων στη Scratch

- Starlogo

Το περιβάλλον Starlogo είναι ένα οπτικό περιβάλλον προγραμματισμού στο οποίο τα συντακτικά λάθη αποφεύγονται μέσω του γραφικού τρόπου σύνταξης των προγραμμάτων. Οι χελώνες που αποτελούν το βασικό αντικείμενο στη StarLogo μπορούν εύκολα να χρησιμοποιηθούν σαν προγραμματιζόμενοι χαρακτήρες ενός παιχνιδιού, που ελέγχονται από τον χρήστη (Begel & Klopfer, 2007).

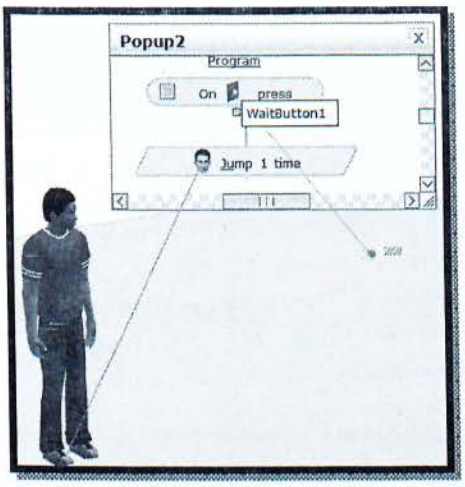
Στη Starlogo υπάρχει η εντολή "ask agent" η οποία μπορεί να υλοποιήσει απλά παραδείγματα όπως αυτό που αναφέραμε (Σχήμα 10). Αν και η εμβέλεια είναι όπως στη Scratch και δεν μπορούν εντολές που αναφέρονται σε μία κλάση αντικειμένων να κληθούν από άλλες, η εντολή «ask agent» δίνει αυτή τη δυνατότητα επικοινωνίας και συγχρονισμού των στιγμιότυπων μεταξύ τους.



Σχήμα 10: Η εντολή ask agent

- Yenka

Στο περιβάλλον προγραμματισμού Yenka ο χρήστης προγραμματίζει με διαγράμματα ροής. Εντολές που αναφέρονται σε διαφορετικά αντικείμενα μπορούν να συνδυαστούν σε ένα πρόγραμμα όπως φαίνεται στο Σχήμα 11.

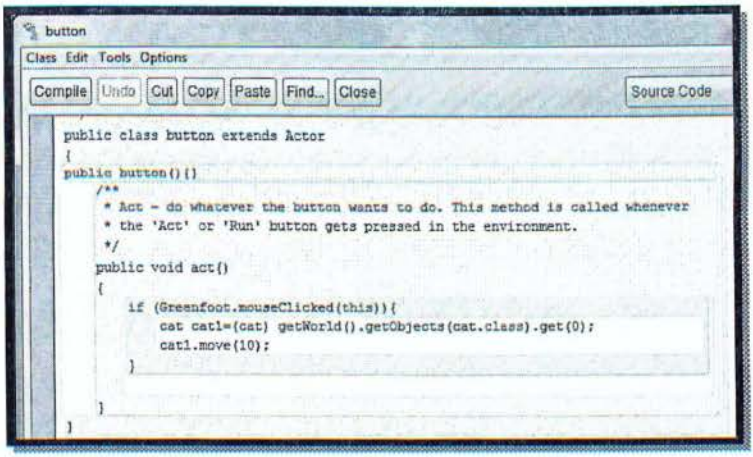


Σχήμα 11: Το απλό πρόβλημα συγχρονισμού στο περιβάλλον Yenka

- Greenfoot

Το περιβάλλον Greenfoot ο μαθητής προγραμματίζει αντικείμενα, που ονομάζονται Ηθοποιοί (Actors), τα οποία βρίσκονται σε έναν Κόσμο (World). Το περιβάλλον έχει δημιουργηθεί με βάση το BlueJ και χρησιμοποιείται η γλώσσα προγραμματισμού Java (Al-Bow et al., 2008). Στο περιβάλλον αυτό ορίζονται κλάσεις και στιγμιότυπα, είναι ένα περιβάλλον που υλοποιεί πλήρως το αντικειμενοστραφές προγραμματιστικό παράδειγμα.

Στο περιβάλλον Greenfoot ένα αντικείμενο μπορεί να δώσει εντολή σε κάποιο άλλο αντικείμενο να εκτελέσει μια εντολή. Ο προγραμματιστής θα πρέπει να βρει πρώτα το όνομα του στιγμιότυπου στο οποίο θέλει να αναφερθεί και στη συνέχεια να χρησιμοποιήσει τον τελεστή της τελείας. Η Greenfoot υλοποιεί τη συνάρτηση getObjects για τον σκοπό αυτό (Σχήμα 12). Σε αυτό το περιβάλλον δίνεται η δυνατότητα να οριστούν ιδιωτικές (private) μέθοδοι και σε αυτή την περίπτωση δεν προσπελαύνονται από άλλα αντικείμενα.

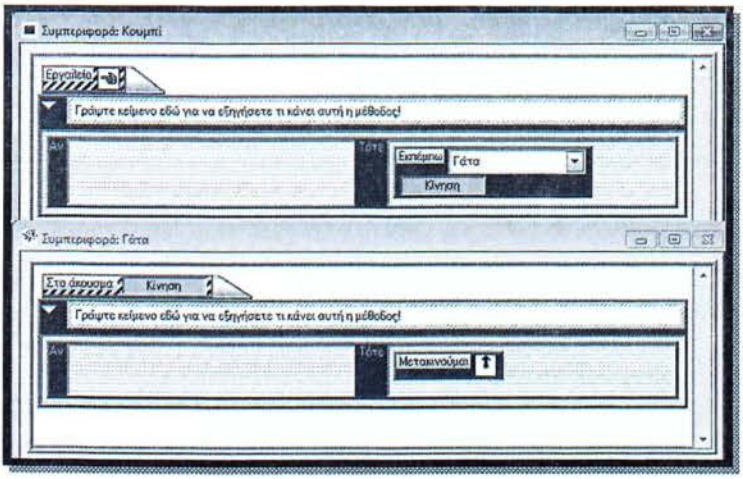


Σχήμα 12: Η μέθοδος Act() για το αντικείμενο Button καλεί την μέθοδο move του αντικειμένου cat1.

2.4.2 Περιβάλλοντα Ανάπτυξης Παιχνιδιών

- Agentsheets

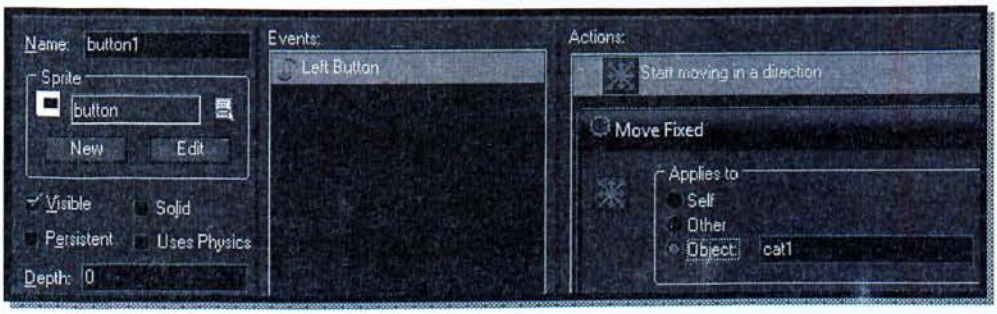
Το περιβάλλον Agentsheets είναι ένα εκπαιδευτικό διαδικτυακό εργαλείο χρησιμοποιείται για τη δημιουργία προσομοιώσεων παιχνιδιών από ένα ευρύ φάσμα προγραμματιστών, από αρχάριους έως επαγγελματίες, όπως επίσης και από εκπαιδευτικούς πολλών χωρών προκειμένου να διδάξουν προγραμματισμό στους μαθητές τους. Περιέχει μια drag-and-drop γλώσσα που δίνει τη δυνατότητα στους μαθητές να δημιουργήσουν παιχνίδια και να τα δημοσιεύσουν στο Διαδίκτυο. Βασικό στοιχείο του προγραμματισμού με το Agentsheets είναι οι πράκτορες. Οι πράκτορες έχουν μια γραφική απεικόνιση στο περιβάλλον και μπορούν να προγραμματιστούν με κανόνες που διέπουν τη συμπεριφορά των πρακτόρων. Οι συνθήκες και οι εντολές έχουν τοπικό χαρακτήρα, δηλαδή αφορούν τον πράκτορα στον οποίο αναφέρονται. Ωστόσο, έχει υλοποιηθεί η εντολή "εκπέμπω" (μετάδοση μηνύματος) με την οποία ένας πράκτορας μπορεί να δώσει εντολή σε κάποιον άλλο πράκτορα.



Σχήμα 13: Επικοινωνία μεταξύ πρακτόρων στο περιβάλλον Agentsheets

- GameMaker

Το GameMaker είναι ένα περιβάλλον οπτικού προγραμματισμού στο οποίο μπορούν να προγραμματιστούν παιχνίδια με γραφικά δύο διαστάσεων. Ο προγραμματισμός γίνεται μέσω αντικειμένων των οποίων η συμπεριφορά προγραμματίζεται (Hernandez et al., 2010). Στο GameMaker δημιουργούνται γεγονότα (events) για τα αντικείμενα που αποτελούν το πρόγραμμα. Ένα trigger που αφορά ένα αντικείμενο μπορεί να δώσει εντολή σε ένα άλλο αντικείμενο να εκτελέσει μια ενέργεια. Για τον σκοπό αυτό έχει υλοποιηθεί η επιλογή «Applies to» που σε αυτή την περίπτωση σημαίνει «Εφαρμόζεται σε». Στο Σχήμα 14, όταν πατηθεί με το ποντίκι το κουμπί (αντικείμενο button1) τότε μπορεί να δοθεί η εντολή move fixed η οποία θα εφαρμοστεί στη γάτα (αντικείμενο cat1).



Σχήμα 14: Υλοποίηση του απλού παραδείγματος με GameMaker

- Kodu

Με τη γλώσσα προγραμματισμού Kodu δημιουργούνται εικονικοί κόσμοι. Στους κόσμους αυτούς εισάγονται αντικείμενα τα οποία μπορούν να προγραμματιστούν. Ο προγραμματισμός γίνεται με συγκεκριμένους κανόνες-συνθήκες, τις οποίες καθορίζει ο χρήστης και πρέπει να ικανοποιούνται μόνες ή συνδυαστικά (Κολόσακα κ.α., 2011).

Στη Kodu, κανένα αντικείμενο δεν μπορεί να δεχθεί εντολές από κάποιο άλλο αντικείμενο. Ωστόσο, τα αντικείμενα μπορούν να ανιχνεύσουν γεγονότα που δεν τα αφορούν. Στο Σχήμα 15 η μορφή Kodu ανιχνεύει το γεγονός «έγινε κλικ με το ποντίκι σε έναν κόκκινο πιεστήρα» και εκτοξεύει μια βολή.

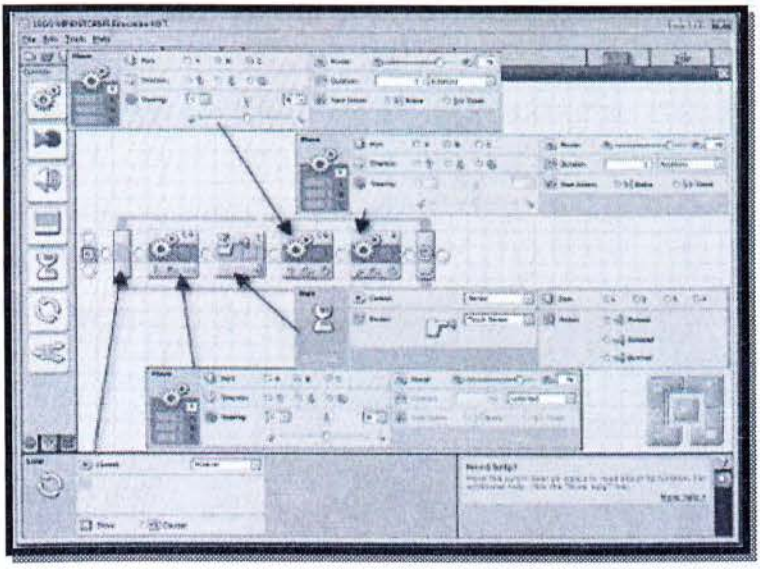
Σχήμα 15: Το Kodu ανιχνεύει ένα γεγονός που αφορά ένα άλλο αντικείμενο

2.4.3 Περιβάλλοντα εκπαιδευτικής ρομποτικής

- LEGO Mindstorms

Τα Lego Mindstorms αποτελούν ένα ευέλικτο μέσο για σχεδιασμό και κατασκευές σε περιορισμένο χρόνο και με μικρό κόστος. Είναι προγραμματιζόμενα ρομποτικά παιχνίδια που έχουν κατασκευαστεί από την εταιρεία Lego. Το περιβάλλον που χρησιμοποιείται για τον προγραμματισμό των ρομποτικών κατασκευών είναι το εκπαιδευτικό λογισμικό LEGO ® Mindstorms ® NXT που βασίζεται στη χρήση εικονιδίων και είναι μια εκπαιδευτική έκδοση του επαγγελματικού λογισμικού LabView του National Instruments. Το λογισμικό έχει μια διαισθητική διεπαφή drag and drop και ένα γραφικό προγραμματιστικό περιβάλλον, το οποίο καθιστά την εφαρμογή προσιτή για έναν αρχάριο, αλλά και

εξίσου δυναμική για έναν εξειδικευμένο χρήστη. Προσφέρει «οικοδομικά» υλικά (τουβλάκια, γρανάζια, τροχούς κλπ.), αισθητήρες για τη μετάδοση δεδομένων στον Η/Υ και συσκευές εξόδου (κινητήρες, λαμπτήρες κλπ.). Τα Lego Mindstorms χρησιμοποιούν «τουβλάκια» κώδικα για τη δημιουργία προγραμμάτων που θα χειρίζονται παιχνίδια φτιαγμένα από Lego. Οι μαθητές πρέπει να ενώσουν τα «τουβλάκια» κώδικα για να δημιουργήσουν κάποιο πρόγραμμα, ακολουθώντας τις βασικές αρχές του προγραμματισμού χωρίς να χρειαστεί να μάθουν κάποια συγκεκριμένη γλώσσα προγραμματισμού ούτε να προβληματιστούν για τη σύνταξη των εντολών. Ένα, επίσης, σημαντικό πλεονέκτημα είναι πως το αποτέλεσμα του προγράμματος δεν εμφανίζεται απλά στην οθόνη αλλά μεταφέρεται στον πραγματικό κόσμο με τη βοήθεια των παιχνιδιών της Lego.



Σχήμα 16: Το περιβάλλον των Lego Mindstorms

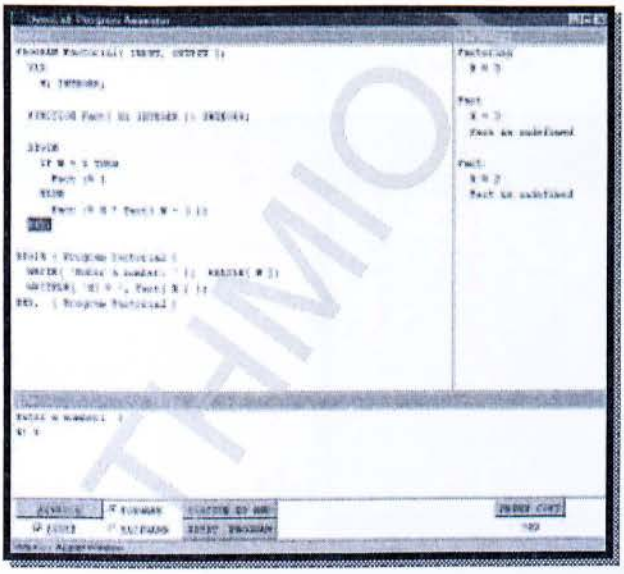
2.4.4 Περιβάλλοντα που δίνουν έμφαση στη Δυναμική Προσομοίωση Εκτέλεσης των Προγραμμάτων

- Dynalab

Το Dynalab (πλήρες όνομα: dynamic computer science laboratory infrastructure featuring program animation) είναι ένα σύστημα δυναμικής

προσομοίωσης εκτέλεσης προγραμμάτων. Στο περιβάλλον του Dynalab ο μαθητής μπορεί να αναπτύξει, να αποθηκεύσει, να ανακαλέσει, να επεξεργαστεί, να αποθηκεύσει και να εκτελέσει ένα πρόγραμμα σε Pascal. Το σημαντικό χαρακτηριστικό του περιβάλλοντος είναι η δυνατότητα της βήμα προς βήμα εκτέλεσης ενός προγράμματος - τόσο προς τα εμπρός όσο και προς τα πίσω - με ταυτόχρονη οπτικοποίηση των δεδομένων, εμφάνιση δηλαδή των μεταβολών των τιμών των μεταβλητών κατά την εκτέλεση ενός προγράμματος. Παρέχεται δηλαδή στο μαθητή ένα «παράθυρο» στο εσωτερικό της μηχανής που μαθαίνει να χειρίζεται βοηθώντας τον έτσι:

- να ελέγξει και να αποσφαλματώσει τα προγράμματά του
- να κατανοήσει τη ροή εκτέλεσης ενός προγράμματος
- να κατανοήσει το δυναμικό χαρακτήρα των μεταβλητών και να εξαλείψει τις σχετικές παρανοήσεις που δημιουργούνται - κατά κύριο λόγο - από τον παραλληλισμό της έννοιας της μεταβλητής στον προγραμματισμό με την έννοια της στα μαθηματικά
- να εξαλείψει παρανοήσεις που σχετίζονται με τη σημασιολογία των δομών επιλογής και επανάληψης



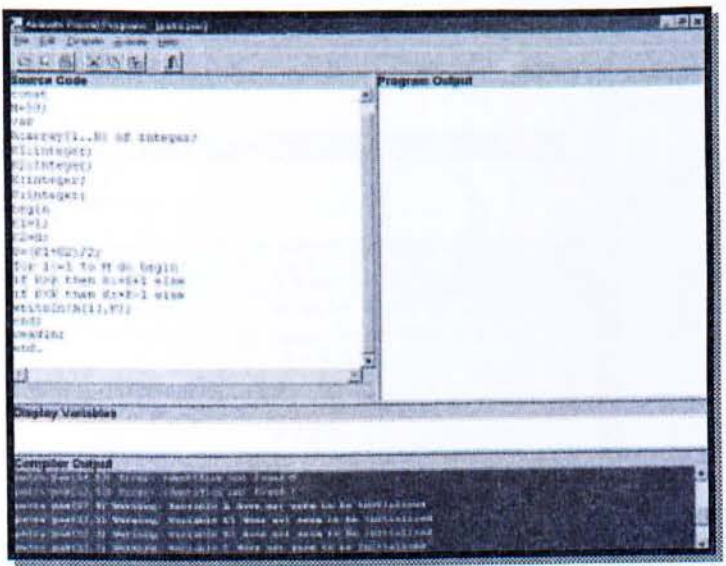
Σχήμα 17: Το περιβάλλον του Dynalab

- AnimPascal

Το AnimPascal αποτελεί και αυτό ένα σύστημα δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων. Σε αντίθεση με το Dynalab, το AnimPascal δεν υποστηρίζει την προς τα πίσω εκτέλεση ενός προγράμματος, η οποία δίνει στο μαθητή τη δυνατότητα να διερευνήσει το τμήμα του προγράμματος που τον δυσκολεύει, αποφεύγοντας έτσι την ανάγκη επανεκκίνησης της εκτέλεσής του. Επίσης, δεν υποστηρίζει την οπτικοποίηση της εκτέλεσης των υποπρογραμμάτων. Ωστόσο, ο μεταγλωττιστής Free Pascal που χρησιμοποιείται δεν ενημερώνει απλά το μαθητή για τα σφάλματα (errors), αλλά τον προειδοποιεί και για πιθανές παραλείψεις (warnings, hints, notes) του προγράμματός του.

Επιπλέον, το AnimPascal ενσωματώνει τη δυνατότητα καταγραφής των ενεργειών των μαθητών, αποθηκεύει δηλαδή τον πηγαίο κώδικα και τα αποτελέσματα της μεταγλώττισης κάθε φορά που ο μαθητής μεταγλωττίζει το πρόγραμμά του. Οι εκδόσεις των προγραμμάτων παρουσιάζονται σε ένα διαφορετικό παράθυρο και παρέχουν τη δυνατότητα στο διδάσκοντα:

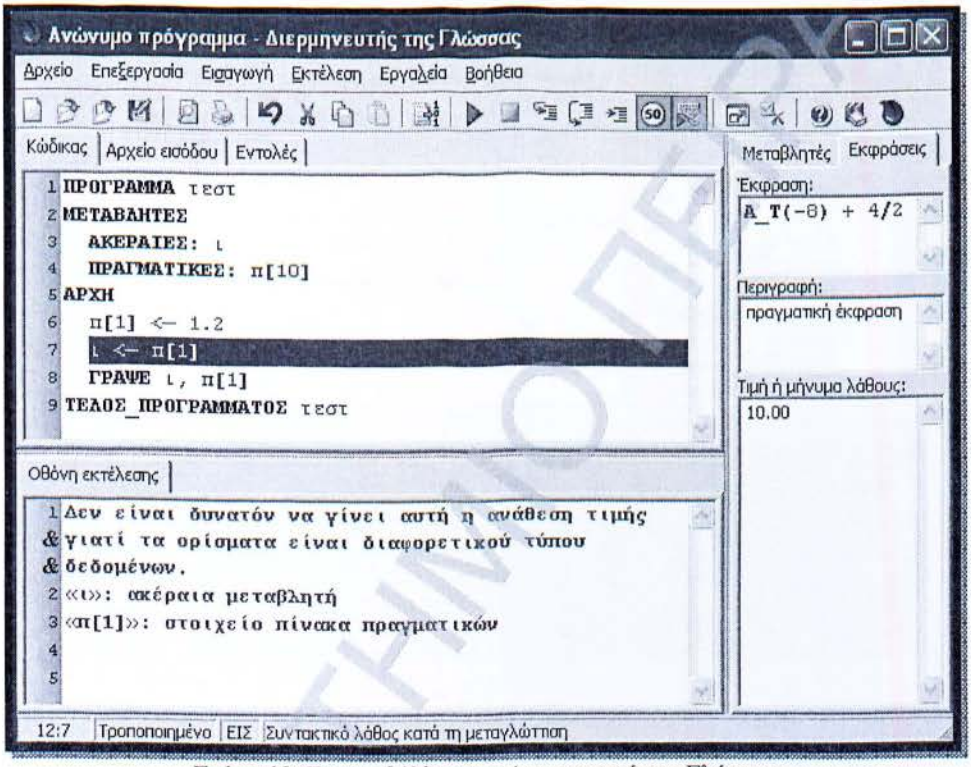
- να διερευνήσει τις δυσκολίες και τις παρανοήσεις των μαθητών, καθώς επίσης και τις στρατηγικές επίλυσης προβλημάτων που εφαρμόζουν
- να προσαρμόσει το μάθημα στις ανάγκες των μαθητών.



Σχήμα 18: Το περιβάλλον του AnimPascal

- Διερμηνευτής της Γλώσσας

Ο Διερμηνευτής της ΓΛΩΣΣΑΣ είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης αλγορίθμων σε μορφή ψευδοκώδικα, ειδικά σχεδιασμένο για τη ΓΛΩΣΣΑ προγραμματισμού που διδάσκεται στα πλαίσια του μαθήματος «Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον» (ΑΕΠΠ) της Γ΄ Γενικού Λυκείου. Μερικά από τα χαρακτηριστικά του είναι η παρακολούθηση της ροής του προγράμματος με βήμα-προς-βήμα εκτέλεση, των μεταβλητών, συνθηκών και εκφράσεων, σημεία διακοπής. Βασικό πλεονέκτημα θεωρείται η υλοποίηση του στην ελληνική γλώσσα και η σύμπνοια με το ισχύον Ενιαίο Πρόγραμμα Προγραμμάτων Σπουδών Πληροφορικής. Ο διερμηνευτής στοχεύει στους αρχάριους προγραμματιστές, όχι στους επαγγελματίες, και προσπαθεί να υποβοηθήσει την ανάπτυξη της αλγοριθμικής σκέψης. Έτσι, για παράδειγμα, αντίθετα με τους επαγγελματικούς compilers, σε περίπτωση συντακτικών λαθών ή λαθών χρόνου εκτέλεσης εμφανίζει αναλυτικά μηνύματα εξηγώντας πού ακριβώς υπήρξε πρόβλημα.

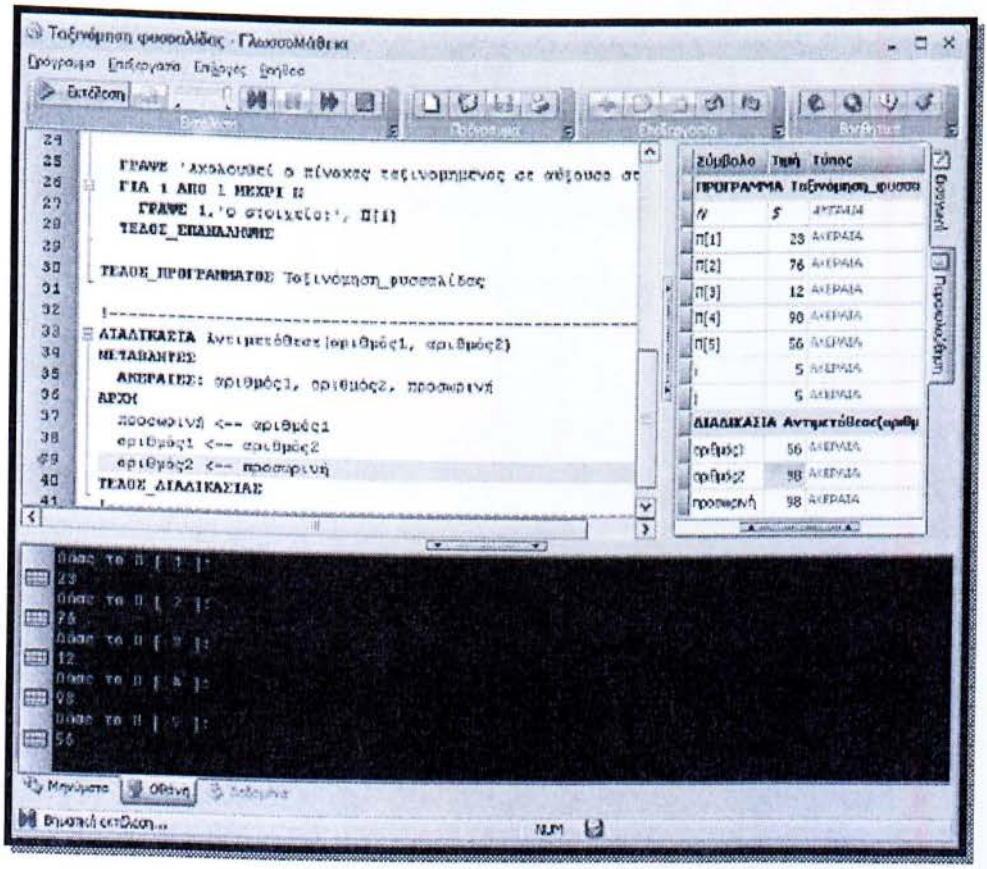


Σχήμα 19: Το περιβάλλον του Διερμηνευτή της Γλώσσας

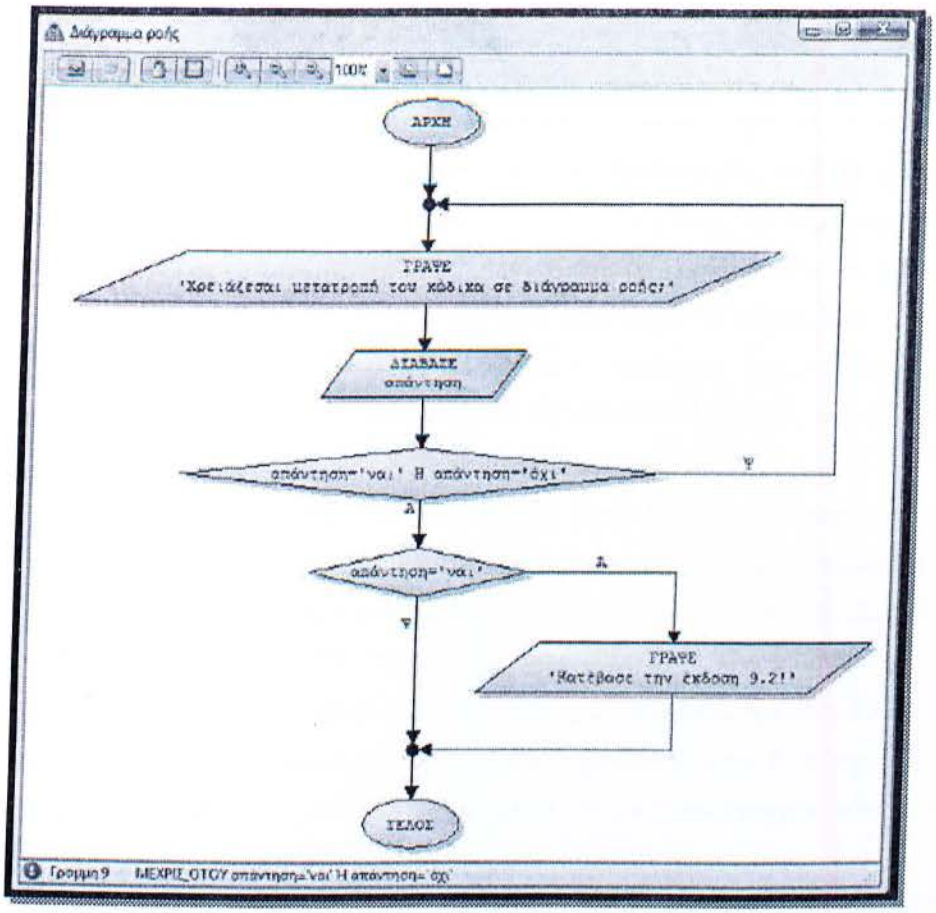
2.4.5 Περιβάλλοντα που στοχεύουν στην Αναπαράσταση της Λύσης ενός Προβλήματος με Πολλαπλές Μορφές

- Γλωσσομάθεια

Η Γλωσσομάθεια είναι ένα ολοκληρωμένο εκπαιδευτικό περιβάλλον προγραμματισμού που υλοποιεί τη γλώσσα που διδάσκεται στα πλαίσια του μαθήματος «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον» (ΑΕΠΠ), του Γενικού Λυκείου. Περιλαμβάνει τη συγγραφή προγραμμάτων μέσω της σχεδίασης διαγραμμάτων ροής με γραφικό τρόπο και κατά την εκτέλεση δίνει έμφαση στην εμφάνιση αναλυτικών μηνυμάτων σφάλματος, κατανοητών ακόμα και από αρχάριους χρήστες. Το πρόγραμμα είναι δυνατόν να εκτελεστεί με επιλεγόμενη ταχύτητα με ταυτόχρονη παρακολούθηση των τιμών μεταβλητών και εκφράσεων.



Σχήμα 20: Το περιβάλλον της Γλωσσομάθειας



Σχήμα 21: Το περιβάλλον της Γλωσσομάθειας(Προβολή διαγράμματος ροής)

2.5 Συμπεράσματα

Για να αντιμετωπιστούν οι δυσκολίες, που παρουσιάζει η εκμάθηση του προγραμματισμού, αναπτύχθηκαν τα εκπαιδευτικά προγραμματιστικά περιβάλλοντα. Αποδεικνύονται εύχρηστα εργαλεία, τα οποία ενσωματώνουν ή όχι δυνατότητες οπτικοποίησης και κίνησης, συνήθως χρησιμοποιούν μια εκπαιδευτική γλώσσα προγραμματισμού και παρέχουν ή όχι τη δυνατότητα της δυναμικής προσομοίωσης της εκτέλεσης των προγραμμάτων. Το λογισμικό οπτικοποίησης αλγορίθμων DAVE παρέχει στο μαθητή δυνατότητες αλληλεπίδρασης σε υψηλό βαθμό, με στόχο την ανάπτυξη αλγορίθμων και την οικοδόμηση επαρκών αναπαραστάσεων για τις αλγοριθμικές δομές. Ο μαθητής αναπτύσσει ο ίδιος τον αλγόριθμο που πρόκειται να οπτικοποιηθεί και δεν περιορίζεται απλά στον καθορισμό των δεδομένων εισόδου του

αλγορίθμου. Έτσι έχει τη δυνατότητα να πειραματιστεί με τον δικό του αλγόριθμο και να ανιχνεύσει τα λάθη του μέσω οικείων αναπαραστάσεων που προκύπτουν από την οπτικοποίηση. Επιπλέον, έχει τη δυνατότητα να επικεντρωθεί στα σημαντικά χαρακτηριστικά κάθε αλγορίθμου και όχι στις τεχνικές λεπτομέρειες της χρησιμοποιούμενης γλώσσας προγραμματισμού. Η προτεινόμενη προσέγγιση από ένα ανοικτό εκπαιδευτικό λογισμικό οπτικοποίησης-προσομοίωσης αλγορίθμων, όπως είναι το DAVE, δίνει έμφαση στον παιδαγωγικό σχεδιασμό της μάθησης του προγραμματισμού και στη μετατόπιση από το συντακτικό-τεχνικό μέρος στην καλλιέργεια δεξιοτήτων επίλυσης προβλημάτων.

Το ολοκληρωτικά γραφικό περιβάλλον Kara-PFSM θα μπορούσε να χρησιμοποιηθεί για μια ευχάριστη και παιγνιώδη εισαγωγή στον προγραμματισμό στις τελευταίες τάξεις του Δημοτικού σχολείου και στο Γυμνάσιο. Το JavaKara θα μπορούσε να αξιοποιηθεί στο Λύκειο για την εισαγωγή των μαθητών στον αλγοριθμικό τρόπο σκέψης μέσα από δραστηριότητες επίλυσης προβλημάτων με όχημα μια ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού, την Java, την οποία μαθητές της βαθμίδας αυτής μπορούν να οικειοποιηθούν αν προσεγγίσουν μέσα από κατάλληλες εκπαιδευτικές δραστηριότητες.

Τα περιβάλλοντα Kara έχουν χρησιμοποιηθεί με επιτυχία σε μαθήματα Πληροφορικής σε διάφορα επίπεδα της εκπαίδευσης κυρίως στη Γερμανία, Αυστρία και Ελβετία. Για παράδειγμα, σε επίπεδο ανώτατης εκπαίδευσης, τα MultiKara και TuringKara έχουν αξιοποιηθεί σε μαθήματα θεωρητικής Πληροφορικής σε Τμήματα Πληροφορικής, ενώ τα Kara-PFSM και JavaKara έχουν χρησιμοποιηθεί σε παιδαγωγικά τμήματα με στόχο την εισαγωγή μελλοντικών εκπαιδευτικών σε βασικές προγραμματιστικές αρχές και στη διδασκαλία τους. Όσον αφορά στη δευτεροβάθμια εκπαίδευση, έρευνες (Reichert, 2003) που έγιναν σε μαθητές Γυμνασίου που είχαν χρησιμοποίησαν το περιβάλλον Kara-PFSM στα σχολεία τους, και οι οποίοι είχαν μηδενική ή περιορισμένη προηγούμενη εμπειρία στον προγραμματισμό, έδειξαν ότι: α) οι αντιδράσεις των μαθητών ήταν πολύ θετικές, β) το περιβάλλον κέντρισε το ενδιαφέρον των μαθητών και έδρασε παρακινητικά, ιδιαίτερα για τους μαθητές χωρίς προηγούμενη προγραμματιστική εμπειρία, γ) οι μαθητές βρήκαν τη διεπαφή της εφαρμογής πολύ απλή και εύχρηστη, δ) οι μαθητές εκτίμησαν ιδιαίτερα το ότι μετά

από μια πολύ σύντομη φάση αρχικής εξοικείωσης με το περιβάλλον άρχισαν πολύ γρήγορα να επιλύουν ενδιαφέροντα προβλήματα, ε) το περιβάλλον επέτρεψε στους μαθητές να εστιάσουν στην επίλυση προβλημάτων και στον έλεγχο της ορθότητας των προγραμμάτων τους, χωρίς να αναλώνονται στη συγγραφή κώδικα σε κάποια γλώσσα προγραμματισμού. Επίσης, έρευνα σε εκπαιδευτικούς που είχαν χρησιμοποίησαν το JavaKara στα σχολεία τους (Reichert, 2003) έδειξε ότι το περιβάλλον ενδείκνυται για τη διδασκαλία Java σε αρχάριους. Τέλος, η απλή και διαισθητική διεπαφή χρήστη των περιβαλλόντων Kara ελαχιστοποιεί το χρόνο εκμάθησης της χρήσης των περιβαλλόντων αυτών, πράγμα που έχει ιδιαίτερη σημασία για τη σχολική εκπαίδευση όπου οι ώρες διδασκαλίας της Πληροφορικής είναι συνήθως περιορισμένες.

Αποτελέσματα πιλοτικών ερευνών έχουν δείξει ότι η αξιοποίηση του εκπαιδευτικού πακέτου Lego Mindstorms μπορεί να αποτελέσει πολύ καλό εκπαιδευτικό εργαλείο, το οποίο κάτω από συγκεκριμένες προϋποθέσεις μπορεί να αποδειχτεί πολύτιμος βοηθός του εκπαιδευτικού και να δώσει δυνατότητα να γίνουν πράξη οι θεωρίες για την κατασκευή της γνώσης μέσα από έρευνα, δοκιμή και απόρριψη.

Με τη χρήση όλων των γραφικών περιβαλλόντων, οι μαθητές μπορούν να κατανοήσουν τις αρχές της προγραμματιστικής δομής. Κατά τη διάρκεια της εφαρμογής του μπορούν να αναπτύξουν δεξιότητες κριτικής σκέψης μέσα από τη διαδικασία οικοδόμησης του προγράμματος και να εξοικειωθούν στον προγραμματισμό στο συγκεκριμένο περιβάλλον. Επιπλέον το ίδιο το πρόγραμμα τους βοηθάει να εξοικειωθούν εύκολα με τα εικονίδια και τις διαθέσιμες εντολές.

Μέσα από κατάλληλες εργασίες αναπτύσσονται επιπλέον δραστηριοτήτων οι οποίες θα ανοίγουν νέους ορίζοντες στους μαθητές σε νέα γνωστικά πεδία όπως Αυτοματισμό, Ρομποτική και Τεχνητή Νοημοσύνη που πρόσφατα έχουν ενταχθεί στις πρώτες βαθμίδες της εκπαίδευσης.

ΚΕΦΑΛΑΙΟ 3

Περιβάλλοντα Γραφικού Προγραμματισμού

ALICE - SCRATCH – BYOB

3.1 ALICE

Το Alice είναι ένα καινοτόμο τρισδιάστατο (3D) προγραμματιστικό περιβάλλον, το οποίο δίνει την δυνατότητα να σχεδιάσουμε εύκολα κινούμενα σχέδια για την αφήγηση μιας ιστορίας, να παίξουμε ένα διαδραστικό παιχνίδι ή να μοιραστούμε βίντεο στον παγκόσμιο ιστό. Αναπτύχθηκε πρώτα στο Πανεπιστήμιο της Βιρτζίνια,Carnegie Mellon (από το 1997), από μια ερευνητική ομάδα με επικεφαλή τον Randy Pausch.

Είναι ένα ελεύθερο διαθέσιμο εργαλείο διδασκαλίας και έχει σχεδιαστεί για να είναι η πρώτη έκθεση ενός μαθητή στον αντικειμενοστραφή προγραμματισμό. Πιο συγκεκριμένα, επιτρέπει στους μαθητές να μάθουν θεμελιώδης έννοιες προγραμματισμού στο πλαίσιο της δημιουργίας ταινιών κινούμενων σχεδίων και απλών βιντεοπαιχνιδιών. Στο Alice, τα τρισδιάστατα αντικείμενα (π.χ. άνθρωποι, ζώα, και οχήματα) κατοικούν σε ένα εικονικό κόσμο και οι μαθητές δημιουργούν ένα πρόγραμμα για να εμψυχώσει αυτά τα αντικείμενα.

Στο διαδραστικό περιβάλλον του Alice, οι μαθητές μπορούν να χρησιμοποιήσουν τα υπάρχοντα γραφικά με τη μέθοδο της μεταφοράς και απόθεσης, ώστε να κατασκευάσουν ένα πρόγραμμα, όπου οι οδηγίες ανταποκρίνονται σε τυποποιημένες δηλώσεις μιας αντικειμενοστραφής γλώσσας προγραμματισμού, όπως Java, C++, και C#. Το Alice, επιτρέπει στους μαθητές να δουν αμέσως πως τρέχει το δικό τους πρόγραμμα κινουμένων σχεδίων, δίνοντας τους την δυνατότητα να κατανοήσουν εύκολα τη σχέση μεταξύ των δηλώσεων του προγραμματισμού και της συμπεριφοράς των αντικειμένων, όσον αφορά την κίνηση τους. Καθώς χειρίζονται τα αντικείμενα στον εικονικό κόσμο, οι μαθητές αποκτούν εμπειρία με όλες τις προγραμματιστικές δομές που συνήθως διδάσκονται σε ένα εισαγωγικό μάθημα προγραμματισμού.

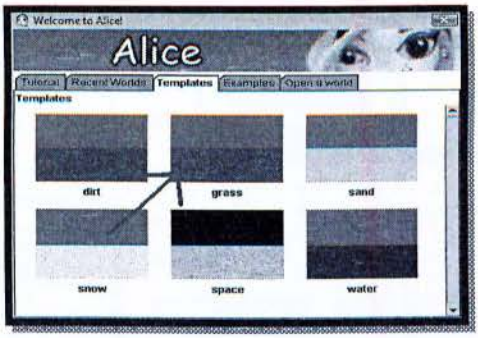
Τα παιδιά έχουν βρει μια καινούρια χώρα των θαυμάτων, όχι σε ταινίες της Disney, αλλά στον κόσμο του προγραμματισμού των ηλεκτρονικών υπολογιστών.

Στο πανεπιστήμιο Duke, του Durham της Βόρειας Καρολίνας αναπτύχθηκε ένα πρόγραμμα Περιπέτειες με το Alice(Adventures in Alice Programming) για να ενσωματώσει τη γλώσσα προγραμματισμού Alice σε γυμνάσια και λύκεια στην πολιτεία. Είναι μια προσπάθεια να εξοικειωθούν οι μαθητές με την επιστήμη των υπολογιστών εισάγοντας μια φιλική προς το χρήστη γλώσσα προγραμματισμού. Το Alice μπορεί να χρησιμοποιηθεί με δυο

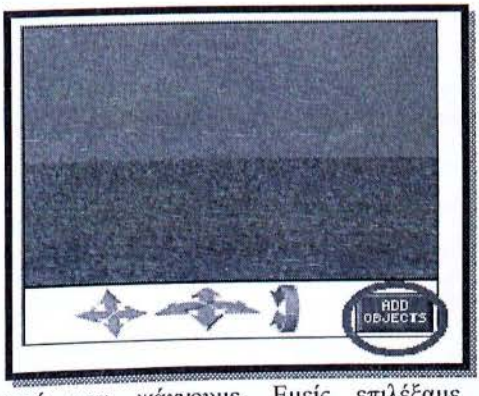
τρόπους, σε αρχικό επίπεδο οι μαθητές μπορούν να μάθουν αρκετά για το πώς κατασκευάζονται στο Alice διαδραστικοί κόσμοι ώστε να ενταχθούν σε ενότητες μαθημάτων και σε υψηλότερο επίπεδο μπορεί να αποτελέσει μια πλήρη σειρά μαθημάτων για την εκμάθηση του προγραμματισμού. Σε κάθε τρέχουσα ενότητα, στην οποία ο μαθητής δημιουργεί μια αφίσα ή μια παρουσίαση, ο φοιτητής μπορεί τώρα να κατασκευάσει μια διαδραστική ιστορία ή ένα παιχνίδι ως παρουσίαση.

Το έργο χρηματοδοτήθηκε αρχικά από το Εθνικό Ίδρυμα Επιστημών και έλαβε χώρα σε έξι περιφέρειες-Ντούρχαμ, Βόρεια Καρολίνα, στη Βιρτζίνια Μπιτς, στο Σαν Χοσέ, Καλιφόρνια, στο Ντένβερ, Κολοράντο, Τσάρλεστον, Νότια Καρολίνα και της Οξφόρδης, Μισισίπι-από το 2006 έως 2009.

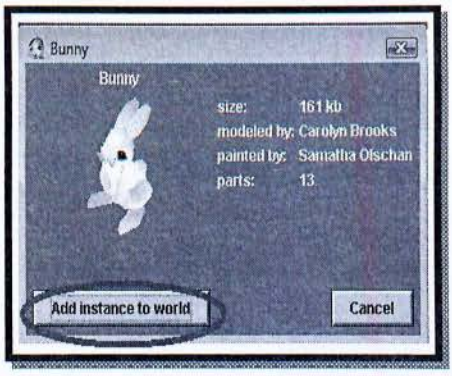
Παρακάτω αναπτύσσουμε ένα απλό παράδειγμα, στο οποίο έχουμε ένα λαγουδάκι που θα λέει "I love Alice!", ώστε να μπορέσουμε να κατανοήσουμε τις βασικές λειτουργίες του Alice. Εφόσον έχουμε ανοίξει το Alice, επιλέγουμε ένα φόντο για τον εικονικό κόσμο μας. Στον κόσμος μας θα τοποθετήσουμε αντικείμενα, τα οποία στη συνέχεια θα έχουν ένα ρόλο. Υπάρχουν έξι διαφορετικές επιλογές φόντου. Για το παράδειγμα μας, επιλέγουμε το γρασίδι.



Το Alice περιέχει διάφορα είδη αντικειμένων, τα οποία προστίθενται στον κόσμο μας ώστε να αποκτήσει ενδιαφέρων. Για να προσθέσουμε ένα αντικείμενο κάνουμε κλικ στο κουμπί Add Objects (Προσθήκη Αντικειμένων):

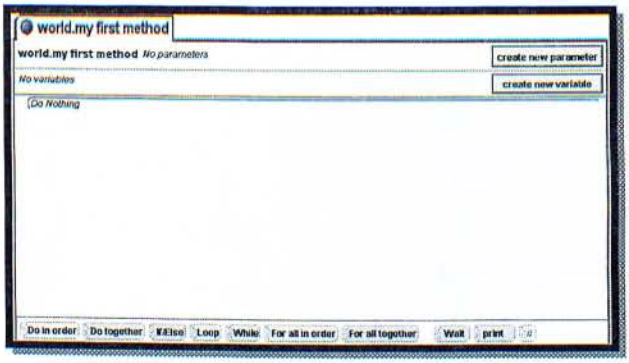


Τα αντικείμενα είναι χωρισμένα σε ομάδες. Οπότε μπορούμε εύκολα να βρούμε αυτό που ψάχνουμε. Εμείς επιλέξαμε το λαγουδάκι από τον φάκελο με τα ζωάκια.

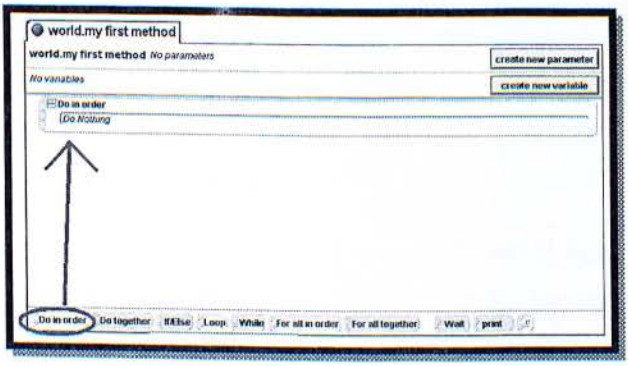


Πατάμε το κουμπί Add Instance to World (Προσθήκη) για να προστεθεί το αντικείμενο μας στον κόσμο του Alice.

Στην οθόνη του Alice βλέπουμε την μέθοδο επεξεργασίας, δηλαδή μια περιοχή όπου θα πούμε στο λαγουδάκι μας τι πρέπει να κάνει. Μοιάζει με αυτό:

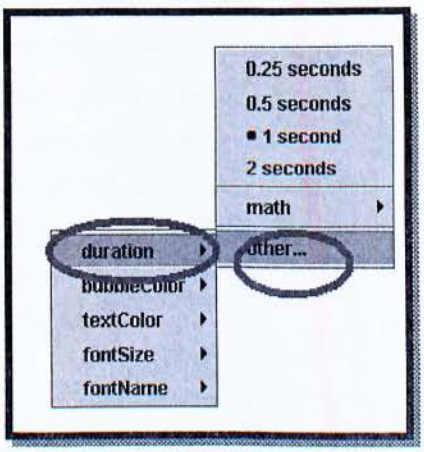


Για να μιλήσουμε στο λαγουδάκι μας και να του δώσουμε οδηγίες, χρησιμοποιούμε εντολές, οι οποίες ονομάζονται μέθοδοι. Το λαγουδάκι ξέρει ήδη ορισμένες εντολές. Επιλέγουμε μια μέθοδο από το μενού που βλέπουμε παρακάτω. Στην περίπτωση μας θα επιλέξουμε την μέθοδο της ομιλίας, bunny say.

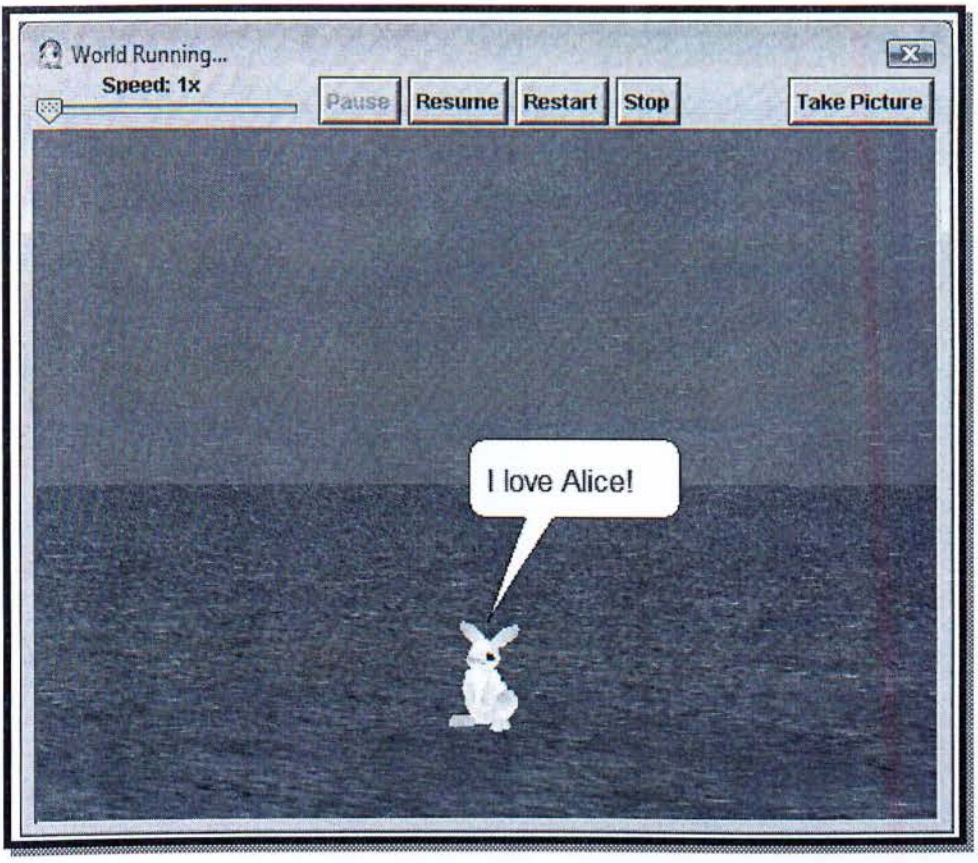


Πριν προσθέσουμε την μέθοδο είναι απαραίτητο να τοποθετήσουμε στην μέθοδο επεξεργασίας το κουμπί Do in order, οι δηλώσεις που βρίσκονται μέσα σε αυτό το μπλοκ θα εκτελεσθούν σε σειρά.

Εφόσον έχουμε επιλέξει την μέθοδο Say, από το μενού που θα εμφανιστεί κάνουμε κλικ στο other (άλλο) και γράφουμε το κείμενο μας, "I love Alice!". Τέλος, ρυθμίζουμε και την διάρκεια που θα εμφανίζεται το κείμενο.



Πατάμε Play για να δούμε πως μοιάζει ο εικονικός κόσμος μας.



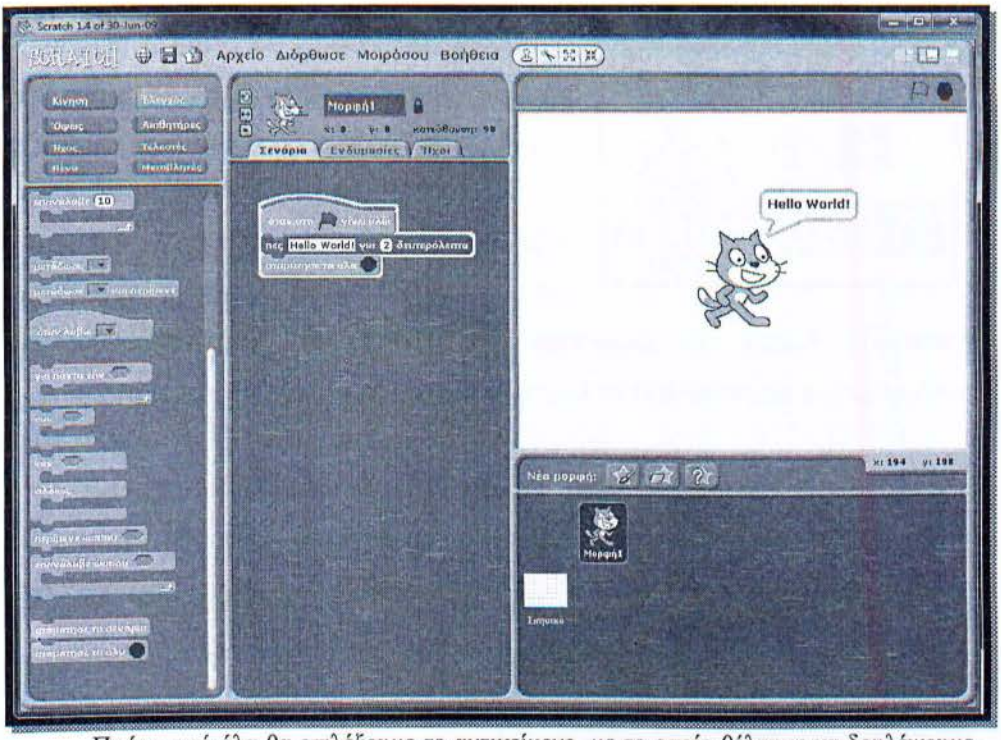
3.2 SCRATCH

Το Scratch είναι μια δυναμική, οπτική γλώσσα προγραμματισμού, της οποίας η υλοποίηση έχει βασιστεί στην ανοικτού λογισμικού αντικειμενοστραφή γλώσσα προγραμματισμού Squeak. Έχει αναπτυχθεί από μια ομάδα ερευνητών στο Lifelong Kindergarten Group στο MIT Media Lab. Σχεδιάστηκε με σκοπό να κάνει τον προγραμματισμό πιο ελκυστικό και προσβάσιμο σε παιδιά διαφορετικών ηλικιών, μπορεί να χρησιμοποιηθεί από ηλικίες 8-16 ετών αλλά μέχρι και από φοιτητές κολεγίου. Δίνει την δυνατότητα στους νέους να δημιουργούν δικές τους διαδραστικές ιστορίες, κινούμενα σχέδια, παιχνίδια και μουσική. Δουλεύοντας με το Scratch τα παιδιά έχουν την ευκαιρία να μάθουν υπολογιστικές έννοιες όπως την επανάληψη, την υπόθεση, μεταβλητές, τύπους δεδομένων.

Είναι αρκετά δημοφιλή στον χώρο της εκπαίδευσης, γεγονός που οφείλεται στο πόσο εύκολα μπορούμε να δημιουργήσουμε ένα πρόγραμμα. Οι εντολές και οι δομές δεδομένων

είναι απλές και η δομή του προγράμματος μπορεί να σχεδιαστεί όπως ένα παζλ, με αποσπώμενα κομμάτια κώδικα που μπορούν να μετακινηθούν και να προσαρμοστούν μαζί. Επίσης, εμείς μπορούμε να κάνουμε αλλαγές στον κώδικα των προγραμμάτων μας καθώς εκτελούνται, με αυτό τον τρόπο μπορούμε να πειραματιστούμε εύκολα σε καινούριες ιδέες.

Παρακάτω θα αναπτύξουμε ένα κλασσικό πρόγραμμα στο Scratch για να κατανοήσουμε τα βασικά χαρακτηριστικά του.

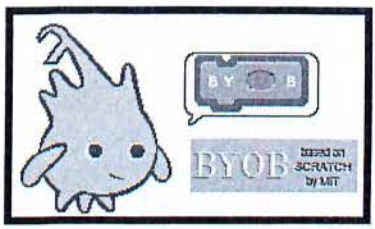


Πρώτα από όλα θα επιλέξουμε το αντικείμενο, με το οποίο θέλουμε να δουλέψουμε. Εδώ είναι επιλεγμένο το Μορφή1 και θα το προγραμματίσουμε ώστε να λέει τη φράση "Hello World". Για να εκτελέσει το Μορφή1 συγκεκριμένες λειτουργίες πρέπει να προγραμματιστεί κατάλληλα. Από τις παλέτες αριστερά θα επιλέξουμε ένα μπλοκ εντολών και θα το τοποθετήσουμε το ένα κάτω από το άλλο ή παράλληλα. Αρχικά, από το Έλεγχος επιλέγουμε πότε θα ξεκινήσει τα πρόγραμμα (συνήθως με το when clicked), στη συνέχεια από το μπλοκ Όψεις επιλέγουμε το say Hello! for 2 secs, δηλαδή ρυθμίζουμε την διάρκεια που το Μορφή1 θα λέει "Hello World!", εδώ δίνουμε την τιμή των 2 δευτερόλεπτων. Τέλος, ορίζουμε πότε θα τερματίζεται το πρόγραμμά μας, στην περίπτωσή μας μετά από 2 δευτερόλεπτα.

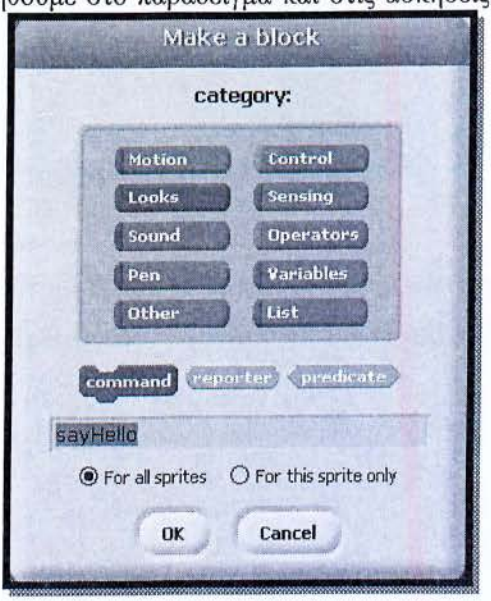
Κατά τον ίδιο τρόπο μπορούμε να προγραμματίσουμε το Μορφή1 να κινείται, να τραγουδάει, γενικά να κάνει οτιδήποτε θέλουμε εμείς. Τοποθετώντας και προγραμματίζοντας περισσότερα αντικείμενα μπορούμε να φτιάξουμε ακόμα και πολύπλοκα παιχνίδια.

3.3 BYOB

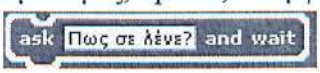
Το BYOB (Build Your Own Blocks) είναι μια επέκταση του Scratch και επιτρέπει στο χρήστη να κατασκευάσει δικά του μπλοκ εντολών. Είναι ένα ελεύθερο λογισμικό που δημιουργήθηκε από τους Jens Mönig και Brian Harvey. Καθίσταται, επίσης, ένα πολύ χρήσιμο πρόγραμμα στην εκπαίδευση μιας και περιέχει δυνατότητες, τις οποίες δεν προσφέρει το Scratch. Πρόσφατα μετονομάστηκε σε SNAP! και παρουσιάστηκε από το Πανεπιστήμιο της Καλιφόρνια στο Μπέρκλεϋ.



Στην παλέτα Variables παρατηρούμε ότι υπάρχουν πολλά επιπλέον κουμπιά σε σχέση με το Scratch. Μας ενδιαφέρει περισσότερο το τελευταίο, που ονομάζεται Make a Block, καθώς με αυτό θα ασχοληθούμε στο παράδειγμα και στις ασκήσεις. Κάνοντας κλικ βλέπουμε ότι εμφανίζεται ένα παράθυρο διαλόγου, όπου μπορούμε να κατασκευάσουμε ένα δικό μας μπλοκ εντολών. Επιλέγουμε ένα όνομα για το μπλοκ, σε ποια κατηγορία θα ανήκει και αν θέλουμε να είναι διαθέσιμο για όλα τα sprites. Πατάμε OK και το καινούριο μπλοκ είναι πλέον διαθέσιμο στην παλέτα που ορίσαμε.



Υποθέτουμε ότι θέλουμε να δημιουργήσουμε ένα μπλοκ με το όνομα sayHello. Πηγαίνουμε στο Variables, πατάμε το κουμπί Make a block, επιλέγουμε

την κατηγορία Looks και command, δίνουμε το όνομα sayHello και πατάμε OK. Το νέο μπλοκ, που μόλις φτιάξαμε, βρίσκεται στην παλέτα Looks. Στη συνέχεια, θα ανοίξει ο Block Editor. Εδώ τοποθετούμε τις εντολές που θα περιέχει το μπλοκ μας. Εμείς θέλουμε το αντικείμενο μας, Sprite1, να εμφανίζει ένα μήνυμα, με το οποίο θα ρωτάει το όνομα του 

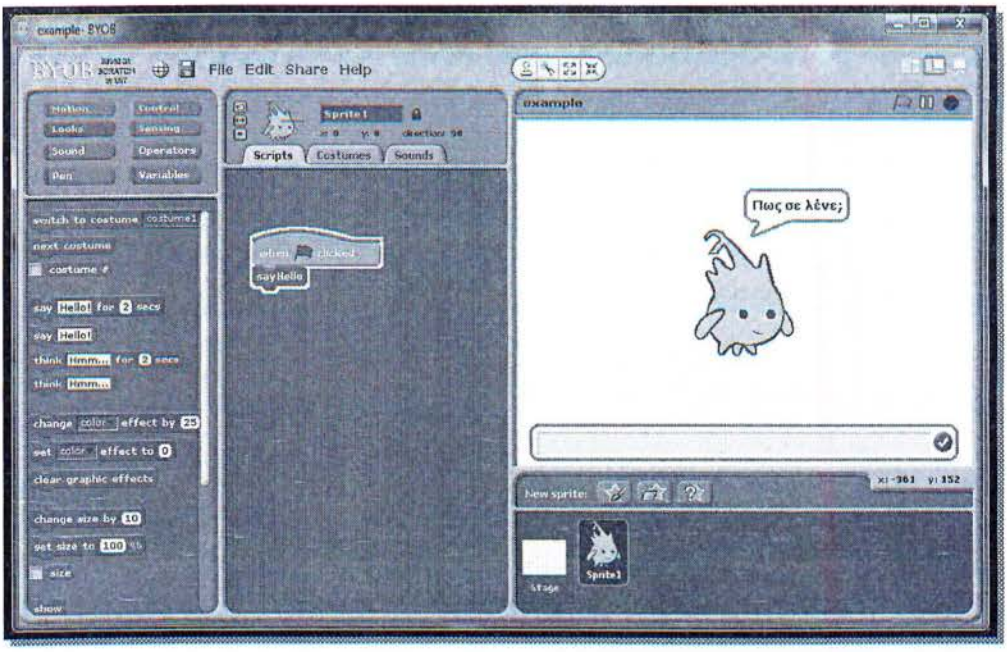
χρήστη και θα χαιρετάει. Οπότε από την παλέτα Sensing επιλέγουμε το ask, ,για να ρωτήσουμε το όνομα του χρήστη. Στη συνέχεια, τοποθετούμε το μπλοκ 

από την παλέτα Looks. Έπειτα, προσθέτουμε από το Operators, το μπλοκ join μέσα στο μπλοκ του say και τέλος το μπλοκ της απάντησης 

.

Πατάμε OK και επιστρέφουμε στο αρχικό παράθυρο. Από την παλέτα control επιλέγουμε το 

και από κάτω προσθέτουμε, από την παλέτα Looks, το μπλοκ sayHello, που μόλις προγραμματίσαμε. Ο ολοκληρωμένος κώδικας το νέου μπλοκ μας φαίνεται παρακάτω:


example- BYOB
File Edit Share Help
Sprite1
Scripts
Costumes
Sounds
example
Πως σε λένε;
New sprite:
Stage
Sprite1

ΚΕΦΑΛΑΙΟ 4

Εφαρμογές για Εκπαιδευτικούς Σκοπούς με τη χρήση Προγραμματιστικών Μικρόκοσμων

4.1 Παραδείγματα με την χρήση του Scratch και του Byob

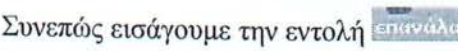
Σκοπός της πρώτης ενότητας του βιβλίου της Πληροφορικής στην Γ' Γυμνασίου, «Γνωρίζω τον υπολογιστή ως ενιαίο σύστημα-Προγραμματισμός», είναι οι μαθητές να αναπτύξουν κριτικές δεξιότητες για την αντιμετώπιση προβλημάτων με τη χρήση υπολογιστή και να επιλύσουν απλά προβλήματα σε προγραμματιστικό περιβάλλον. Ο εκπαιδευτικός πρέπει να αναπτύξει στους μαθητές την πειραματική και ερευνητική διάθεση, δίνοντας τους απλά και διασκεδαστικά προβλήματα προς επίλυση. Οι μαθητές ενθαρρύνονται να κατανοήσουν το προς επίλυση πρόβλημα, να το αναλύσουν και με την βοήθεια ενός κατάλληλου προγραμματιστικοί περιβάλλοντος να συνθέσουν τη λύση του.

Στην ενότητα αυτή, θα υλοποιήσουμε παραδείγματα που αφορούν τα κεφάλαια, «Εισαγωγή στην έννοια του αλγόριθμου και στον προγραμματισμό» και «Ο προγραμματισμός στην πράξη», της πρώτης ενότητας του σχολικού βιβλίου. Μέσα από τα παραδείγματα, ο μαθητής θα μάθει να προγραμματίζει. Θα κατανοήσει την έννοια του προβλήματος, του αλγορίθμου και του προγραμματισμού. Κυρίως, θα μπορέσει να κατανοήσει τις προγραμματιστικές δομές όπως είναι η μεταβλητή, η δομή της επιλογής και της επανάληψης.

Τέλος, οι ασκήσεις που ακολουθούν έχουν δημιουργηθεί με χρήση των λογισμικών Scratch και Byob.

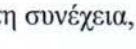
4.1.1 Δημιουργία σχημάτων

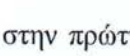
Στο ακόλουθο παράδειγμα δημιουργούμε ένα σενάριο, με το οποίο θα κατασκευάσουμε έναν αλγόριθμο με απλές οδηγίες, ώστε να σχηματιστούν στην οθόνη πολλαπλά τετράγωνα και ένα λουλούδι. Παρουσιάζονται μέσα από το παράδειγμα οι ιδιότητες ενός αλγόριθμου. Αλλάζοντας διάφορα βήματα του αλγόριθμου οι μαθητές προβληματίζονται αν, μετά την εκτέλεση του, παράγονται τα ίδια αποτελέσματα. Τονίζεται η σημασία της σαφήνειας και περατότητας του αλγόριθμου. Επίσης, θα δει την δομή της επανάληψης αλλά και την εμφωλευμένη επανάληψη. Δηλαδή έχουμε την δυνατότητα να εισάγουμε μια επανάληψης μέσα σε μια άλλη επανάληψη κάτι το οποίο μας δίνει νέες δυνατότητες.

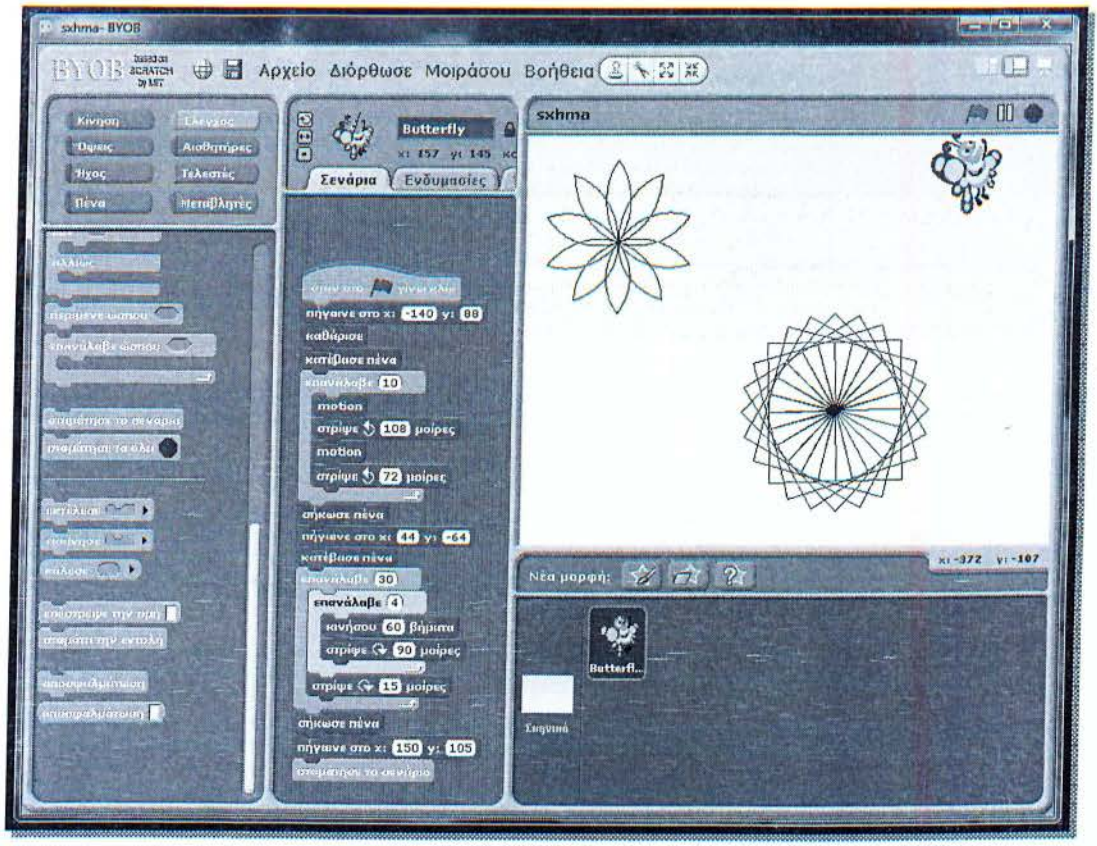
Πρώτα απ' όλα θα κατασκευάσουμε ένα λουλούδι με 10 πέταλα. Από το αρχείο επιλέγουμε το εικονοστοιχείο που θέλουμε να προγραμματίσουμε και έπειτα, από τις παλέτες αριστερά επιλέγουμε το σύνολο των εντολών τοποθετώντας τη μια κάτω από την άλλη, ώστε το αντικείμενο μας να εκτελέσει συγκεκριμένες λειτουργίες. Πρώτα απ' όλα, από το Έλεγχος επιλέγουμε πότε θα ξεκινάει το πρόγραμμα μας, συνήθως με το  . Στη συνέχεια θέλουμε να ορίσουμε μία επανάληψη, αφού η διαδικασία που ακολουθεί θα επαναλαμβάνεται 10 φορές. Συνεπώς εισάγουμε την εντολή  . Μέσα στον βρόγχο της επανάληψης θα τοποθετήσουμε ένα μπλοκ εντολών, το οποίο θα περιλαμβάνει τις εντολές κίνησης

του αντικειμένου μας για την δημιουργία του κάθε πέταλου. Οπότε από την παλέτα Μεταβλητές επιλέγουμε Δημιουργήστε μια εντολή, την κατηγορία Κίνηση και δίνουμε το όνομα motion. Στο παράθυρο που εμφανίζεται θα προγραμματίσουμε το νέο μπλοκ. Οι εντολές που περιλαμβάνει είναι οι εξής

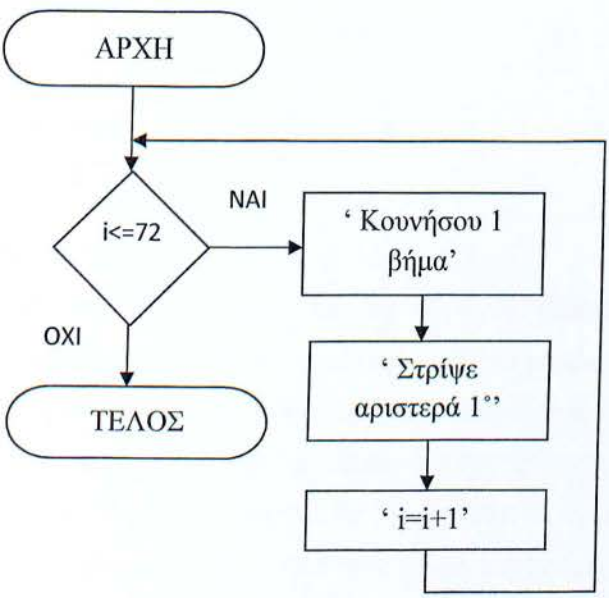
Το αντικείμενό μας θα κινηθεί 72 φορές και κάθε φορά θα κάνει ένα βήμα και θα στρίβει μία μοίρα.

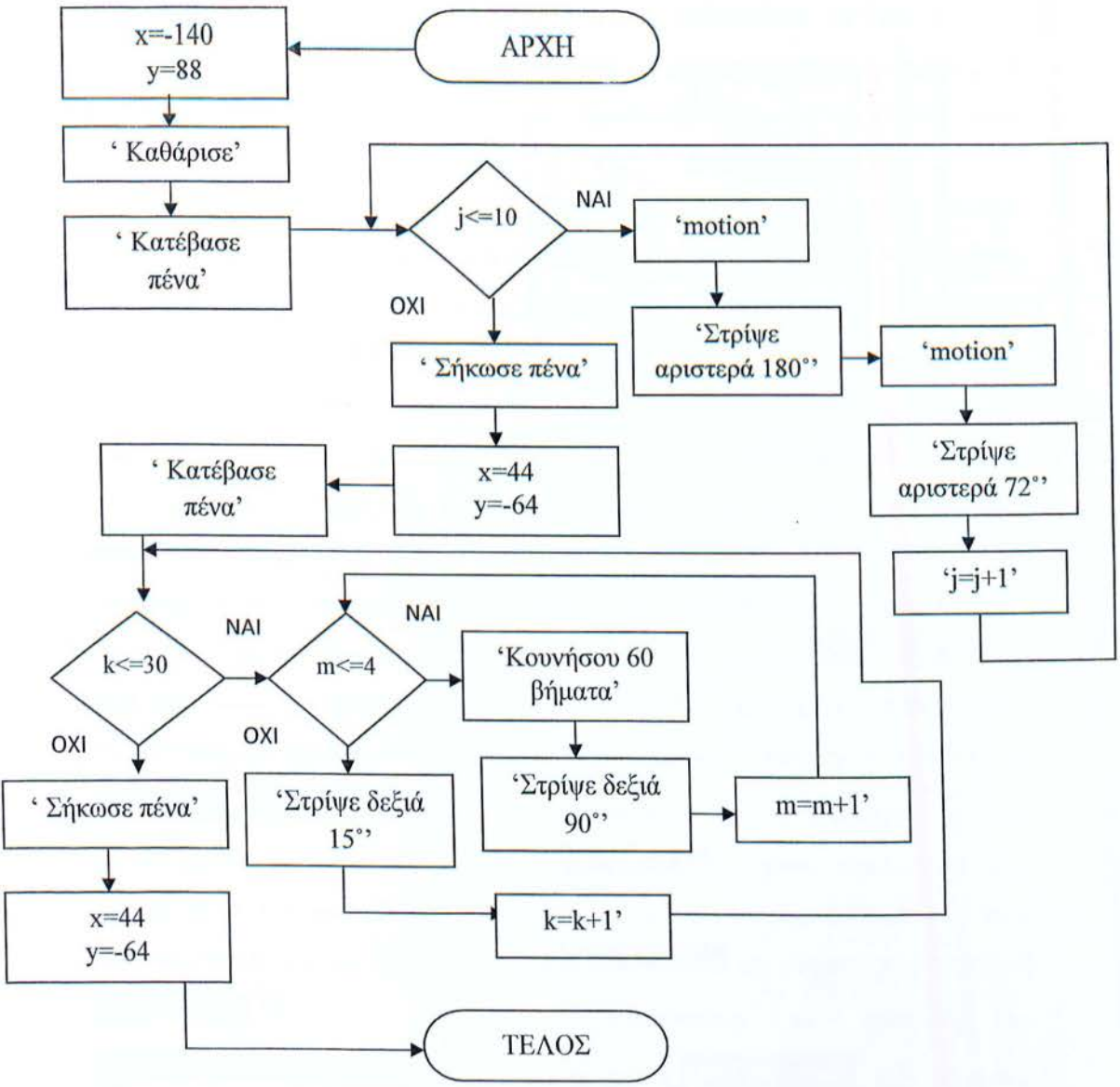
Από την παλέτα Κίνηση βρίσκουμε την εντολή motion,  και την τοποθετούμε στην αρχική επανάληψη. Στη συνέχεια, τοποθετούμε την εντολή  , ξανά την εντολή  και τέλος την εντολή . Με αυτόν τον τρόπο δημιουργούμε κάθε πέταλο.

Για να δημιουργήσουμε τα τετράγωνα, πρέπει να τοποθετήσουμε την εντολή  και μέσα στην πρώτη επανάληψη την εντολή  . Η εξωτερική επανάληψη δείχνει πόσα τετράγωνα θέλουμε να σχηματίσουμε και η εσωτερική ότι θα δημιουργήσουμε ένα τετράγωνο. Μέσα στην δεύτερη επανάληψη τοποθετούμε τις εντολές  και για το μήκος των πλευρών και την αλλαγή της κατεύθυνσης του κάθε τετραγώνου. Για να σχεδιαστεί το επόμενο τετράγωνο μέσα στην εξωτερική επανάληψη θα τοποθετήσουμε την εντολή  . Ολοκληρωμένο το παράδειγμα μας είναι:



Ακολουθούν τα διαγράμματα ροής του μπλοκ motion και του προγράμματος.

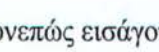


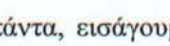
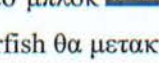
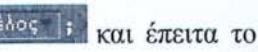
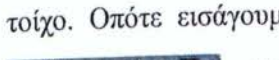
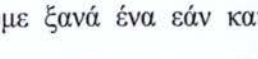


4.1.2 Λαβύρινθος

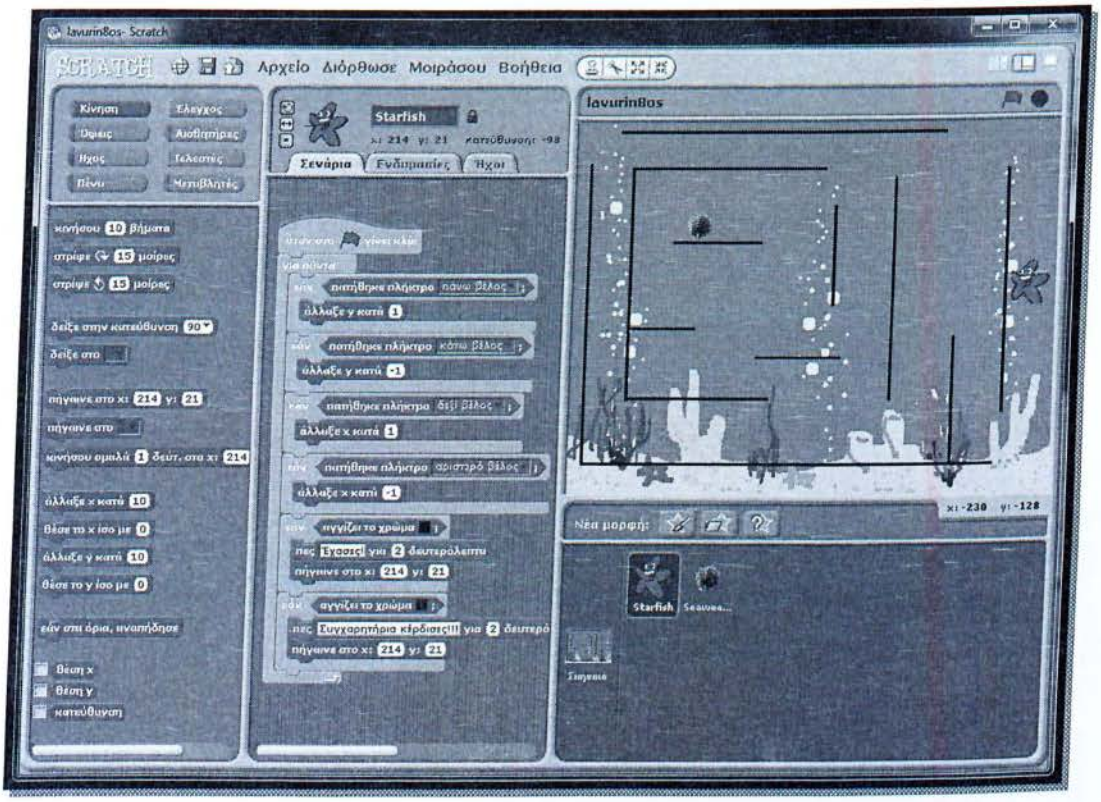
Ο λαβύρινθος είναι ένα παιχνίδι. Σκοπός είναι να καθοδηγήσουμε τον αστερία μέσα σε ένα λαβύρινθο να βρει την τροφή του χωρίς να πέσει πάνω στους τοίχους. Η καθοδήγηση του αστερία θα γίνεται από το πληκτρολόγιο με την βοήθεια των τεσσάρων βελών. Το παιχνίδι θα ξεκινά με το πάτημα της πράσινης σημαίας. Εάν ο αστερίας ακουμπήσει σε ένα τοίχο το παιχνίδι θα τερματίζεται με την εμφάνιση του μηνύματος «Έχασες!!!». Εάν ο αστερίας φτάσει στην τροφή του το παιχνίδι θα

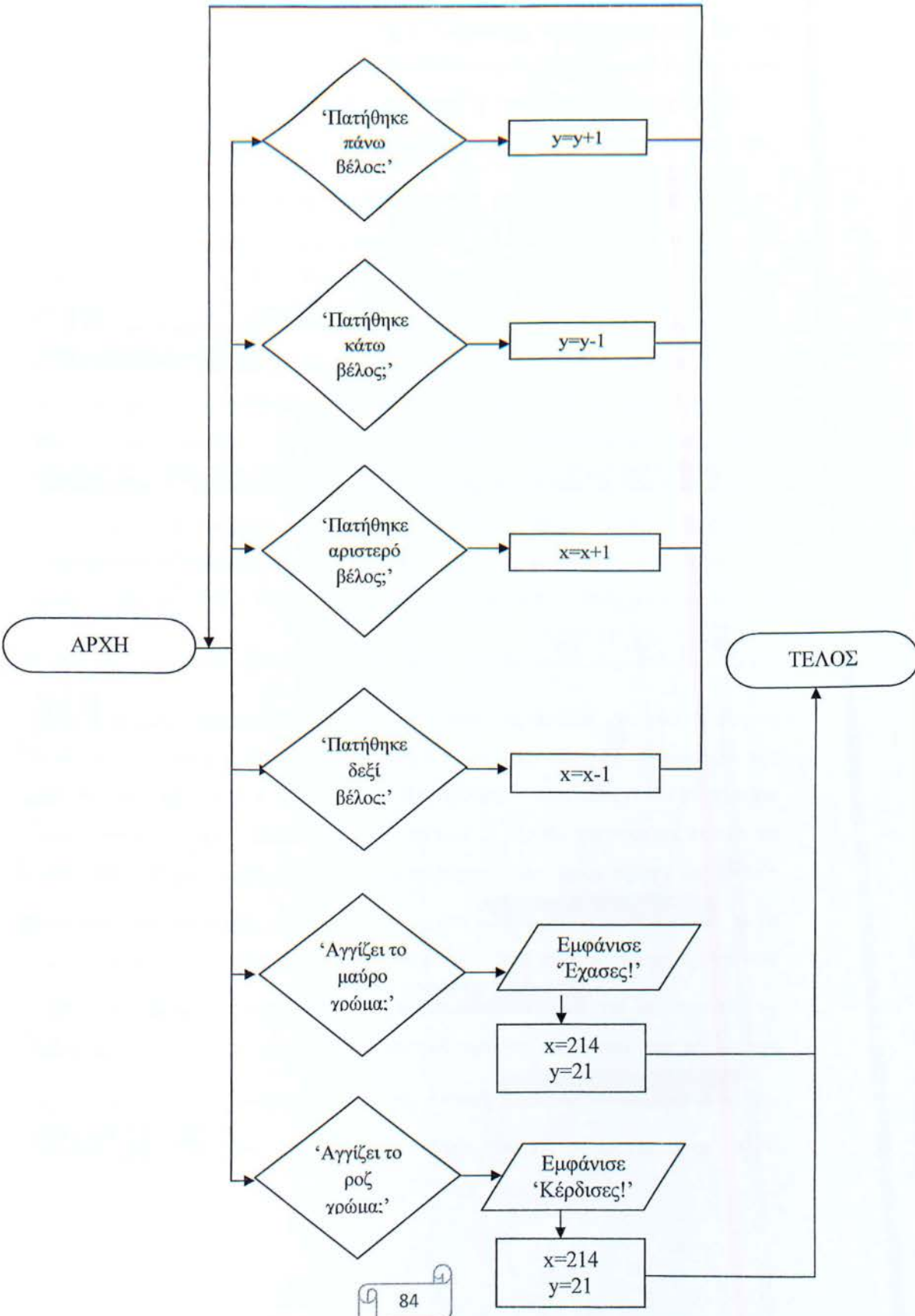
διακόπτεται με την εμφάνιση του μηνύματος «Συγχαρητήρια κέρδισες!!!». Το παράδειγμα αυτό, θα μπορούσε να βοηθήσει στην διδασκαλία της δομής επιλογής. Ο μαθητής αποκτά την δυνατότητα να εξοικειωθεί με τις συνθήκες, που χρησιμοποιούμε σε έναν έλεγχο.

Αρχικά σχεδιάζουμε τον λαβύρινθο και εισάγουμε τα δυο αντικείμενα μας, τα οποία θα προγραμματίσουμε από τις παλέτες αριστερά ώστε να εκτελούν συγκεκριμένες λειτουργίες. Καταρχάς, ορίζουμε το πότε θα ξεκινάει το πρόγραμμα εισάγοντας την εντολή  από την παλέτα Έλεγχος. Στη συνέχεια θέλουμε να ορίσουμε μία επανάληψη, αφού η διαδικασία που ακολουθεί θα επαναλαμβάνεται για πάντα. Συνεπώς εισάγουμε την εντολή  .

Σε αυτό το σημείο, θα προγραμματίσουμε τον βρόγχο της επανάληψης, ο οποίος θα περιλαμβάνει τις συνθήκες, οποίες μας δίνουν την δυνατότητα να ελέγξουμε αν το αντικείμενο μας θα κινηθεί ή θα εμφανίσει κάποιο μήνυμα. Μέσα στον κλάδο της εντολής για πάντα, εισάγουμε την εντολή εάν  ώστε να δημιουργήσουμε μια εντολή ελέγχου. Στο εξάγωνο που υπάρχει, θα τοποθετήσουμε τις συνθήκες που θέλουμε να ελεγχθούν. Σαν πρώτη επιλογή, βάζουμε ως συνθήκη το  από την παλέτα Αισθητήρες. Στη συνέχεια, από την παλέτα Κίνηση προσθέτουμε το μπλοκ  , το οποίο σημαίνει ότι όταν πατηθεί το πάνω βελάκι το Starfish θα μετακινηθεί. Κατά τον ίδιο τρόπο, στο δεύτερο εάν, βάζουμε ως επιλογή  και στη συνέχεια το μπλοκ  .Στο τρίτο εάν, τοποθετούμε ως συνθήκη το  και έπειτα το μπλοκ  . Στο τελευταίο εάν, που αφορά την κίνηση του αντικειμένου βάζουμε σαν προδιαγραφή το  και το μπλοκ  . Στη συνέχεια, πρέπει να προγραμματίσουμε την περίπτωση που το αντικείμενο μας βρει σε ένα τοίχο. Οπότε εισάγουμε ξανά ένα εάν και σαν συνθήκη τοποθετούμε το μπλοκ  και έπειτα τα μπλοκ  και  . Το πρώτο μπλοκ σημαίνει ότι αν το Starfish αγγίξει το μαύρο χρώμα τότε θα εμφανιστεί το μήνυμα του δεύτερου μπλοκ και το αντικείμενο στη συνέχεια θα μετακινηθεί στην θέση που ορίζει το τρίτο μπλοκ. Τέλος, έχουμε την

περίπτωση που το Starfish φτάσει στην τροφή του. Η συνθήκη που εισάγουμε στο εάν είναι αγγίζει το χρώμα ; , δηλαδή αν ακουμπήσει το πράσινο χρώμα τότε θα έχουμε κερδίσει. Άρα, οι εντολές που εισάγονται είναι πες Συγχαρητήρια κέρδισες!!! για 2 δευτερόλεπτα , με την οποία θα εμφανιστεί το μήνυμα «Συγχαρητήρια κέρδισες!!!» και πήγαινε στο x: 214 y: 21 κατά την οποία το αντικείμενο επιστρέφει στην αρχική του θέση. Ολοκληρωμένος ο κώδικας και το διάγραμμα ροής φαίνονται παρακάτω:




ΑΡΧΗ
'Πατήθηκε πάνω βέλος:'
y=y+1
'Πατήθηκε κάτω βέλος;'
y=y-1
'Πατήθηκε αριστερό βέλος;'
x=x+1
'Πατήθηκε δεξί βέλος:'
x=x-1
'Αγγίζει το μαύρο γρώμα:'
Εμφάνισε 'Έχασες!'
x=214
y=21
'Αγγίζει το ροζ γρώμα:'
Εμφάνισε 'Κέρδισες!'
x=214
y=21
ΤΕΛΟΣ

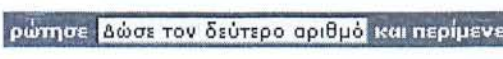
4.1.3 Αριθμομηχανή

Στο βιβλίο της Πληροφορικής της Γ' Γυμνασίου, υπάρχει μία δραστηριότητα κατά την οποία τα παιδιά πρέπει να κατασκευάσουν μία αριθμομηχανή που να κάνει τις 4 βασικές πράξεις. Εμείς δημιουργήσαμε αυτό το σενάριο στη γλώσσα Scratch.

Πρώτον, δημιουργούμε δυο μεταβλητές a και b στην παλέτα Μεταβλητές και τις εμφανίζουμε στη σκηνή. Τοποθετούμε την εντολή 

για να ορίσουμε το πότε θα ξεκινάει το σενάριο. Στη συνέχεια ζητάμε από το χρήστη να μας δώσει δυο αριθμούς. Από την παλέτα Αισθητήρες παίρνουμε το μπλοκ 

και από την παλέτα Μεταβλητές το μπλοκ 

, με το οποίο ορίζουμε την μεταβλητή a να πάρει την τιμή που περιέχει η εντολή απάντηση, η οποία έχει την τιμή που έδωσε ο χρήστης. Με τον ίδιο τρόπο ζητάμε τον δεύτερο αριθμό. Τοποθετούμε το μπλοκ 

και έπειτα 

.

Για της 4 πράξεις θα δημιουργήσουμε 4 κουμπιά, με τα ονόματα «Πρόσθεση», «Αφαίρεση», «Πολλαπλασιασμός» και «Διαίρεση». Τα κουμπιά εισάγονται ως νέες μορφές από το αρχείο και από την καρτέλα Ενδυμασίες επιλέγουμε το κουμπί Διόρθωσε για να προσθέσουμε τα κατάλληλα σύμβολα, 

, 

, 

και 

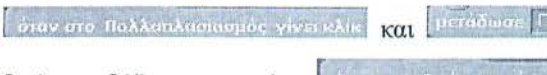
. Σκοπός κάθε κουμπιού είναι η εκτέλεση μίας ομάδας εντολών. Κάνοντας κλικ σε κάθε κουμπί θέλουμε η μορφή Cat να εμφανίζει το αποτέλεσμα της αντίστοιχης πράξης. Για το λόγο αυτό είναι απαραίτητο να μεταδοθεί ένα μήνυμα για κάθε κουμπί στο οποίο κάνουμε κλικ. Το μήνυμα το λαμβάνει η γάτα και εκτελεί το αντίστοιχο σενάριο υπολογισμού του αποτελέσματος. Ας δούμε πρώτα τον κώδικα των κουμπιών. Καταρχάς, τοποθετούμε το μπλοκ 

, με το οποίο δηλώνουμε ότι η επόμενη εντολή θα εκτελεστεί εφόσον κάνουμε κλικ στο Πρόσθεση. Έπειτα, βάζουμε την εντολή 

, για να μεταδοθεί το μήνυμα «Προσθέτω». Ακολούθως, κατασκευάζουμε και τον κώδικα των υπόλοιπων κουμπιών. Για την αφαίρεση έχουμε τις εντολές 

και 

. Στον κώδικα του πολλαπλασιασμού τοποθετούμε τα μπλοκ



και 

. Τέλος, για την διαίρεση βάζουμε το μπλοκ 

και 

. Στο σενάριο του αντικειμένου Cat, για την εκτέλεση της πρόσθεσης τοποθετούμε την εντολή 

, με την οποία όταν το Cat λάβει το μήνυμα Προσθέτω θα εμφανίσει στην οθόνη το αποτέλεσμα της πράξης 

. Για την εκτέλεση της αφαίρεσης οι εντολές είναι 

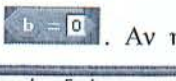
και 

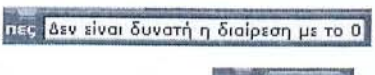
. Για τον πολλαπλασιασμό 

και 

.Για την διαίρεση τοποθετούμε την εντολή 

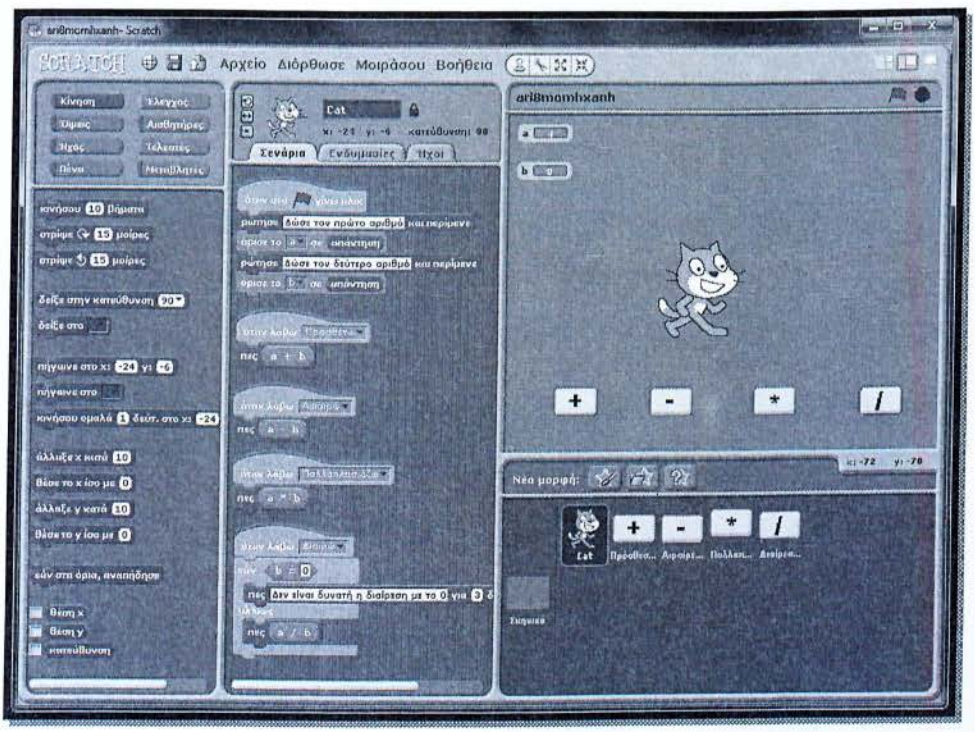
και έπειτα ελέγχουμε αν ο δεύτερος αριθμός ισούται με μηδέν. Οπότε τοποθετούμε την εντολή 

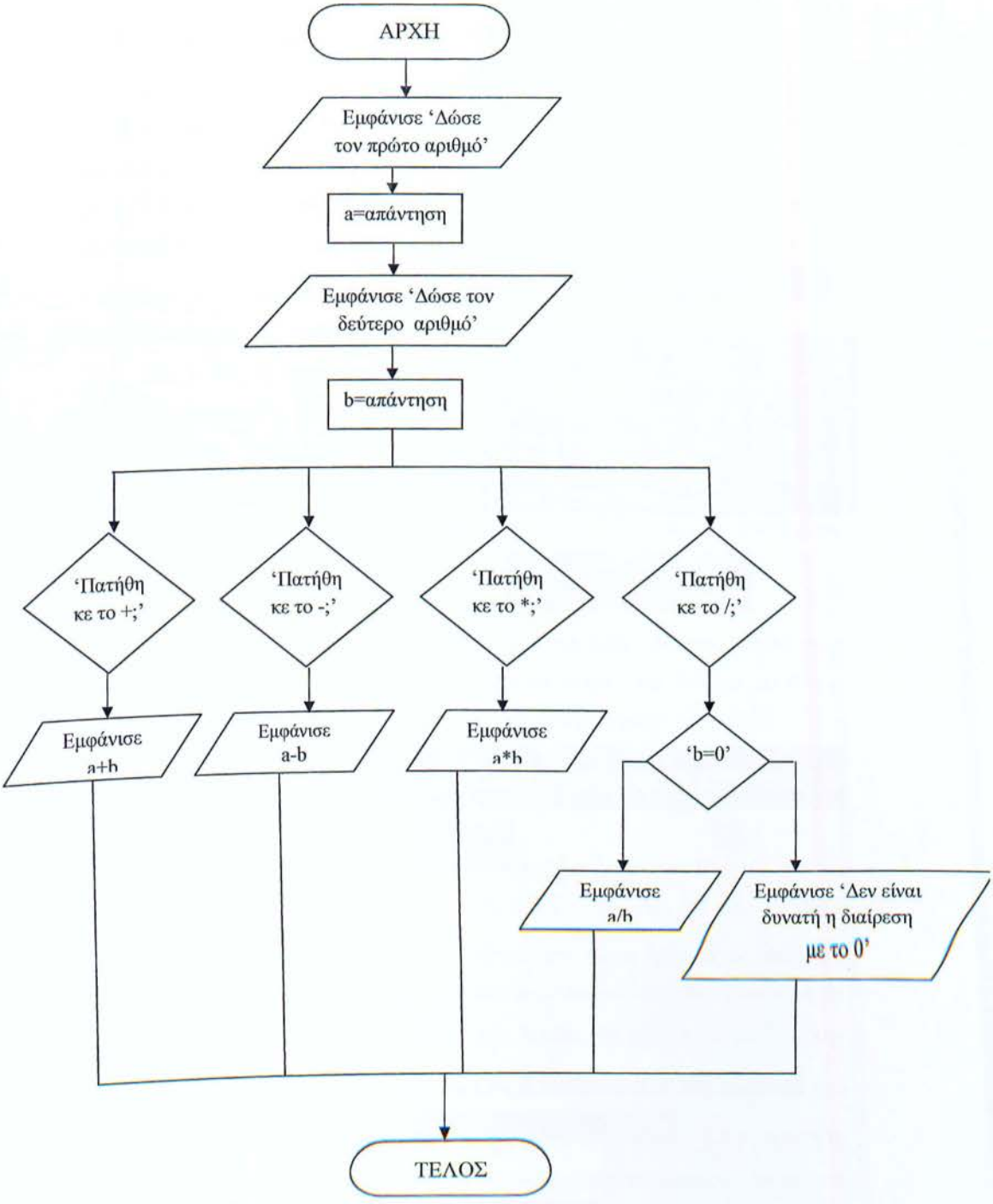
και ως συνθήκη βάζουμε το 

. Αν η συνθήκη αληθεύει τότε θα εμφανιστεί το μήνυμα 

αλλιώς θα μας δώσει το αποτέλεσμα της διαίρεσης με την εντολή 

. Ολοκληρωμένο το παράδειγμα και το διάγραμμα ροής φαίνονται παρακάτω:

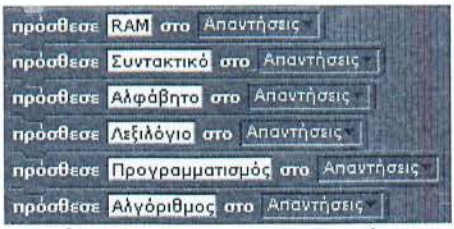



ΑΡΧΗ
Εμφάνισε 'Δώσε τον πρώτο αριθμό'
a=απάντηση
Εμφάνισε 'Δώσε τον δεύτερο αριθμό'
b=απάντηση
'Πατήθηκε το +;'
'Πατήθηκε το -;'
'Πατήθηκε το *;'
'Πατήθηκε το /;'
Εμφάνισε a+b
Εμφάνισε a-b
Εμφάνισε a*b
'b=0'
Εμφάνισε a/b
Εμφάνισε 'Δεν είναι δυνατή η διαίρεση με το 0'
ΤΕΛΟΣ

4.1.4 Παιχνίδι γνώσεων

Σε αυτό το παράδειγμα θα υλοποιήσουμε ακόμα μια δραστηριότητα από το βιβλίο της Πληροφορικής. Θα κατασκευάσουμε ένα παιχνίδι γνώσεων, στο οποίο οι ερωτήσεις αφορούν το πρώτο κεφάλαιο του βιβλίου. Πρώτα απ' όλα, δημιουργούμε ένα πίνακα που θα περιέχει τις απαντήσεις. Από την παλέτα Μεταβλητές επιλέγουμε Δημιουργήστε μια λίστα και δίνουμε το όνομα Απαντήσεις. Για να προσθέσουμε αντικείμενα στη λίστα χρησιμοποιούμε την εντολή 

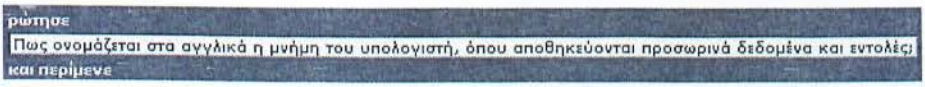
. Οπότε ολοκληρωμένος ο πίνακας θα είναι:



Στη συνέχεια δημιουργούμε δυο μεταβλητές, οι οποίες θα αποθηκεύουν πόσες σωστές και λάθος απαντήσεις έδωσε ο χρήστης. Από την παλέτα Μεταβλητές επιλέγοντας Δημιουργήστε μια μεταβλητή και δίνουμε τα ονόματα Σωστές και Λανθασμένες. Τις αρχικοποιούμε με τιμή μηδέν 

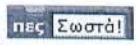
.

Έπειτα δημιουργούμε ένα μπλοκ εντολών για κάθε ερώτηση. Επιλέγουμε Δημιουργήστε μια εντολή, την κατηγορία που θα ανήκει και δίνουμε το όνομα Ερώτηση1. Από την παλέτα Αισθητήρες τοποθετούμε την εντολή

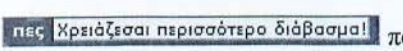


και έπειτα εισάγουμε το μπλοκ 

από την παλέτα Έλεγχος για να ελέγχουμε τις απαντήσεις. Ως συνθήκη βάζουμε την εντολή 

, η οποία ελέγχει αν στη λίστα Απαντήσεις υπάρχει η απάντηση που έδωσε ο χρήστης. Αν η απάντηση υπάρχει τότε θα εμφανιστεί το μήνυμα 

και η τιμή της μεταβλητής Σωστές θα αυξηθεί κατά 1 με την εντολή 

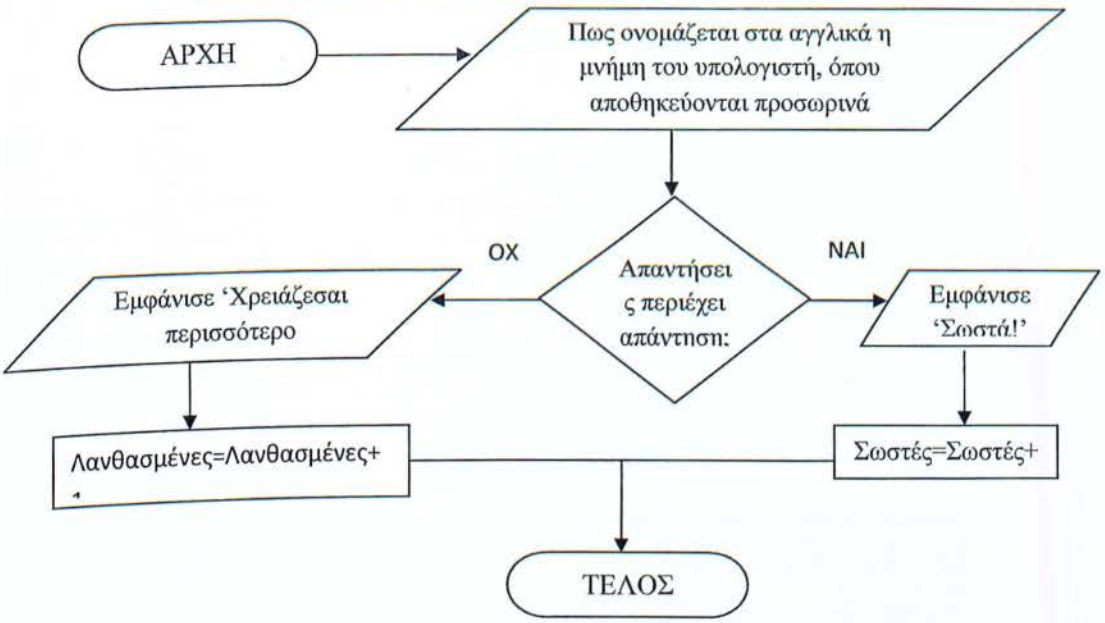
διαφορετικά 

που σημαίνει ότι απάντησε λάθος και 

. Κάθε ερώτηση εμφανίζεται εφόσον ο χρήστης κάνει κλικ σε κάποιο αριθμητικό πλήκτρο, για αυτό το λόγο τοποθετούμε την εντολή 

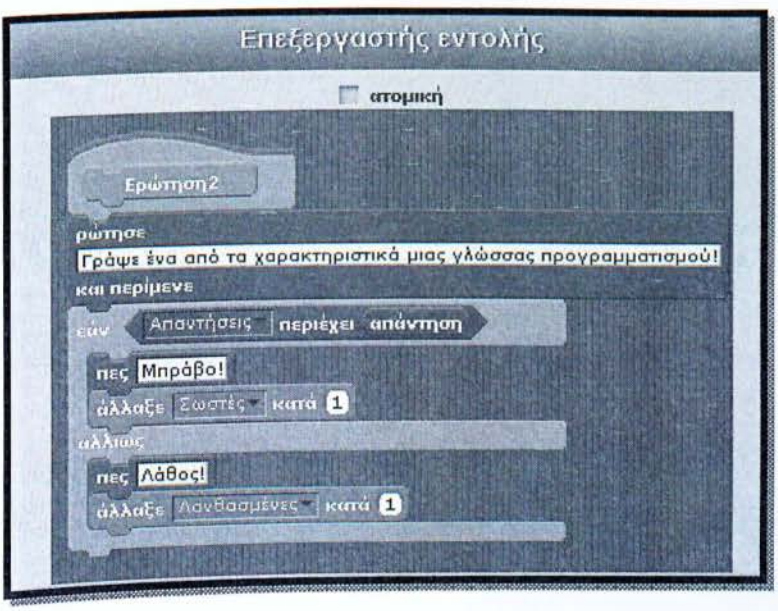
.

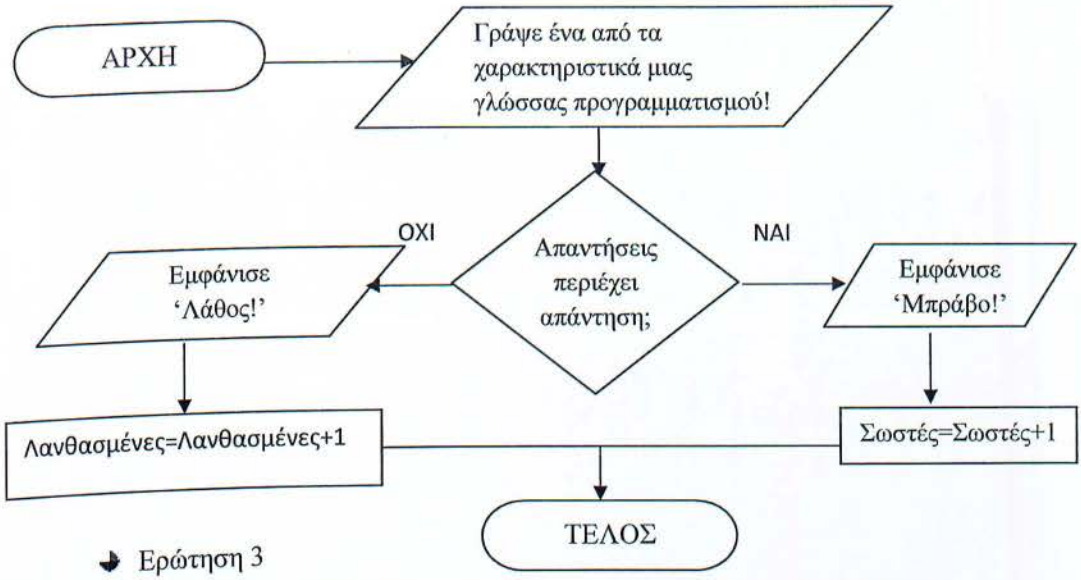
Παρακάτω βλέπουμε το διάγραμμα ροής του πρώτου μπλοκ



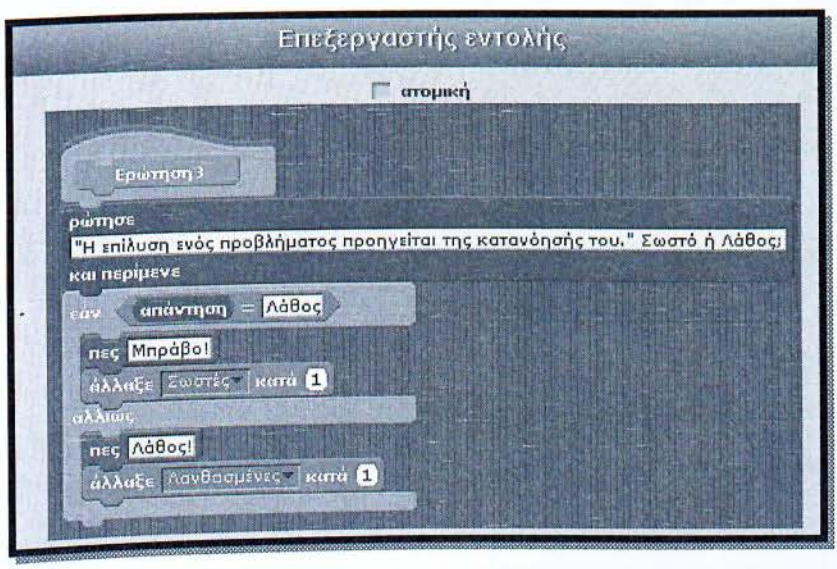
Ακολουθούν οι κώδικες των υπόλοιπων ερωτήσεων μαζί με τα διαγράμματα ροής τους:

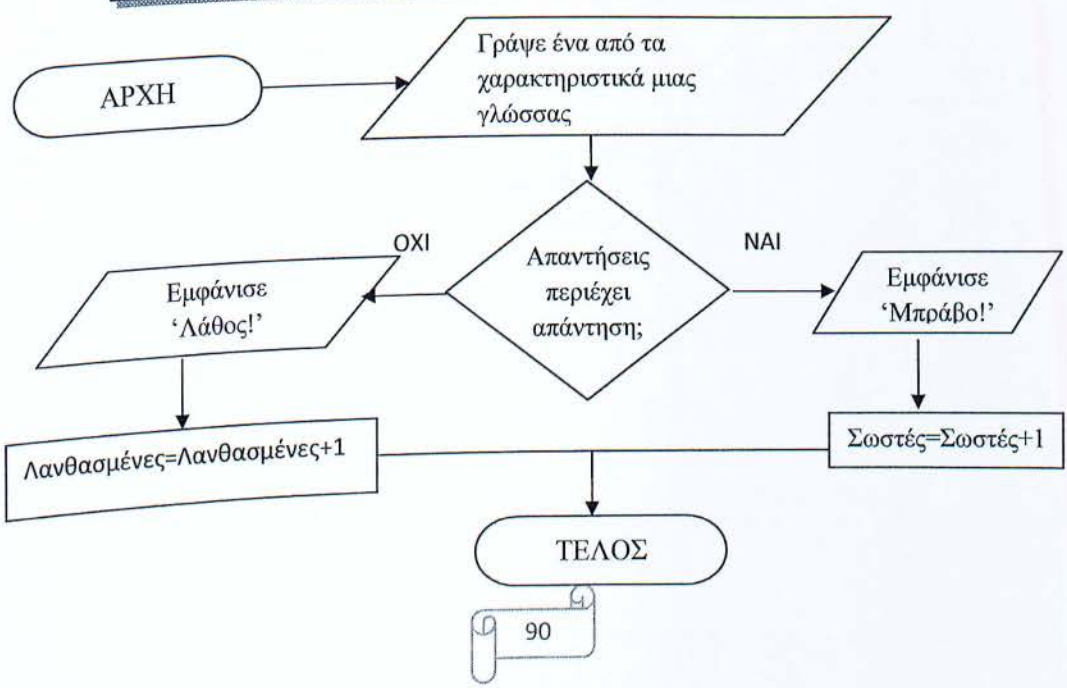
- Ερώτηση 2



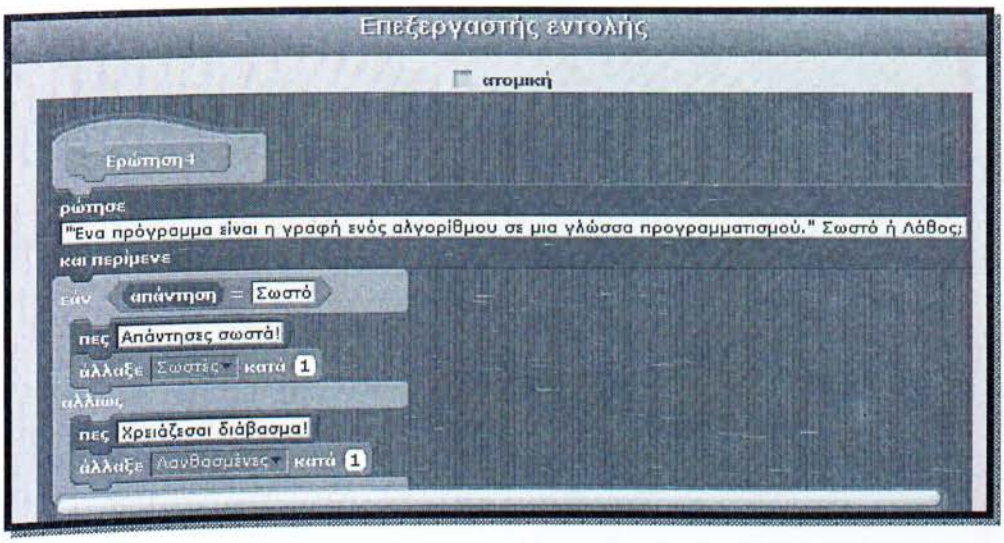

ΑΡΧΗ
Γράψε ένα από τα χαρακτηριστικά μιας γλώσσας προγραμματισμού!
Απαντήσεις περιέχει απάντηση;
ΟΧΙ
ΝΑΙ
Εμφάνισε 'Λάθος!'
Εμφάνισε 'Μπράβο!'
Λανθασμένες=Λανθασμένες+1
Σωστές=Σωστές+1
ΤΕΛΟΣ

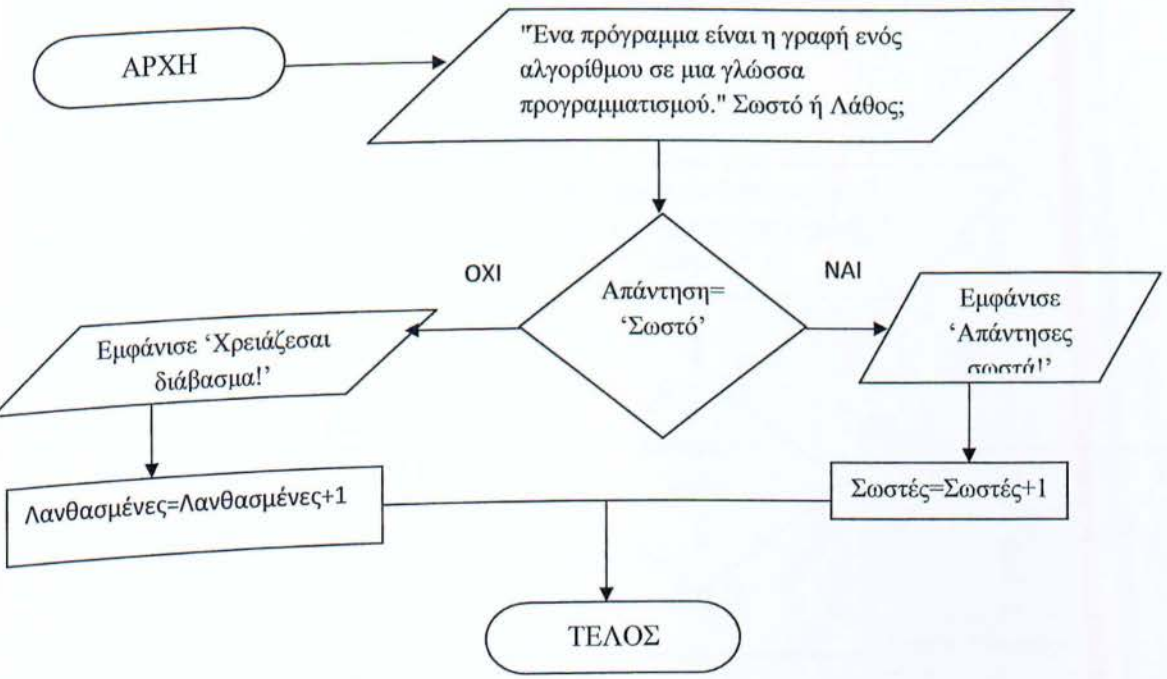
- Ερώτηση 3


Επεξεργαστής εντολής
ατομική
Ερώτηση 3
ρώτησε
"Η επίλυση ενός προβλήματος προηγείται της κατανόησής του." Σωστό ή Λάθος;
και περίμενε
εάν απάντηση = Λάθος
πες Μπράβο!
άλλαξε Σωστές κατά 1
αλλιώς
πες Λάθος!
άλλαξε Λανθασμένες κατά 1

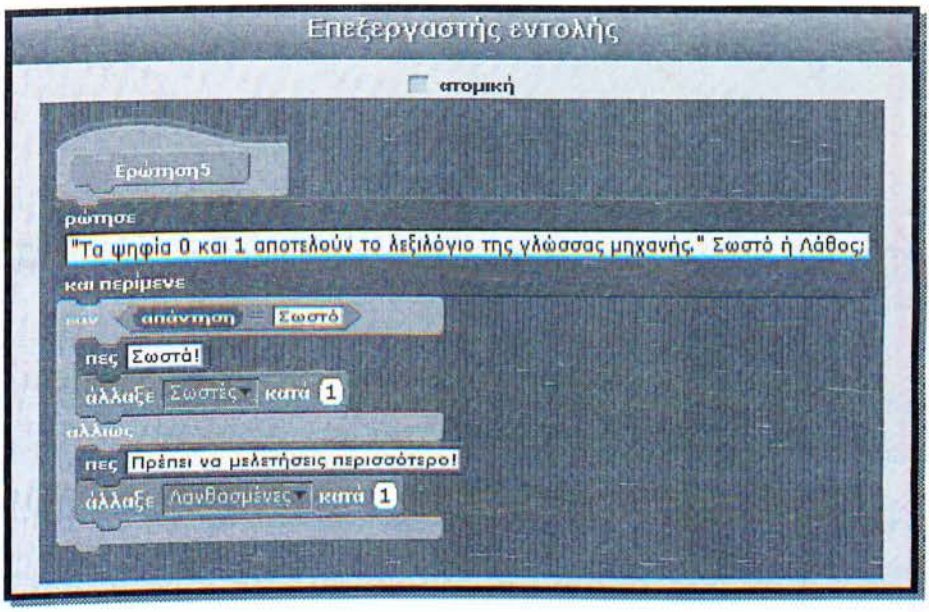

ΑΡΧΗ
Γράψε ένα από τα χαρακτηριστικά μιας γλώσσας
Απαντήσεις περιέχει απάντηση;
ΟΧΙ
ΝΑΙ
Εμφάνισε 'Λάθος!'
Εμφάνισε 'Μπράβο!'
Λανθασμένες=Λανθασμένες+1
Σωστές=Σωστές+1
ΤΕΛΟΣ

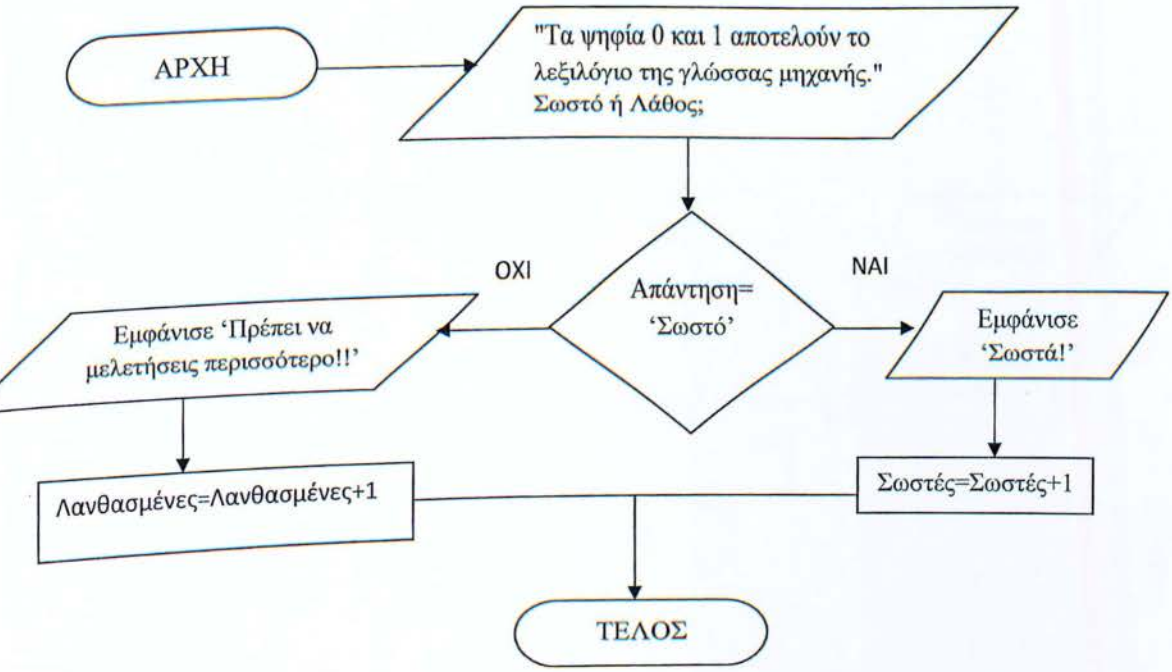
- Ερώτηση 4


Επεξεργαστής εντολής
ατομική
Ερώτηση 4
ρώτησε
"Ενα πρόγραμμα είναι η γραφή ενός αλγορίθμου σε μια γλώσσα προγραμματισμού." Σωστό ή Λάθος;
και περίμενε
εάν απάντηση = Σωστό
πες Απάντησες σωστά!
άλλαξε Σωστές κατά 1
αλλιώς
πες Χρειάζεσαι διάβασμα!
άλλαξε Λανθασμένες κατά 1

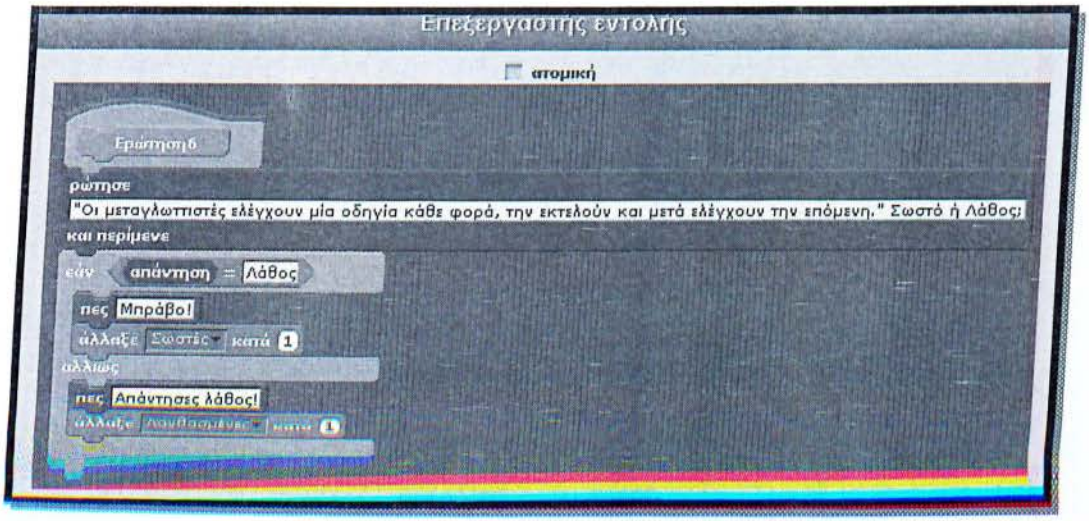

ΑΡΧΗ
"Ενα πρόγραμμα είναι η γραφή ενός αλγορίθμου σε μια γλώσσα προγραμματισμού." Σωστό ή Λάθος;
Απάντηση= 'Σωστό'
ΟΧΙ
ΝΑΙ
Εμφάνισε 'Χρειάζεσαι διάβασμα!'
Εμφάνισε 'Απάντησες
Λανθασμένες=Λανθασμένες+1
Σωστές=Σωστές+1
ΤΕΛΟΣ

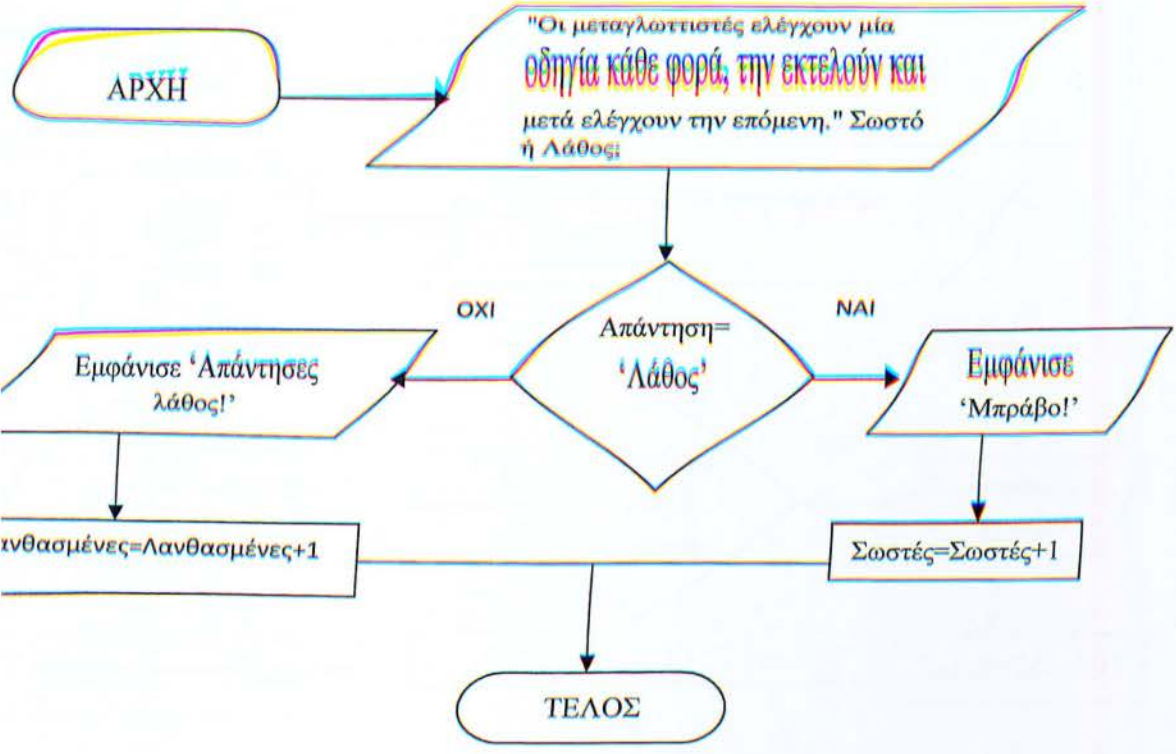
- Ερώτηση 5



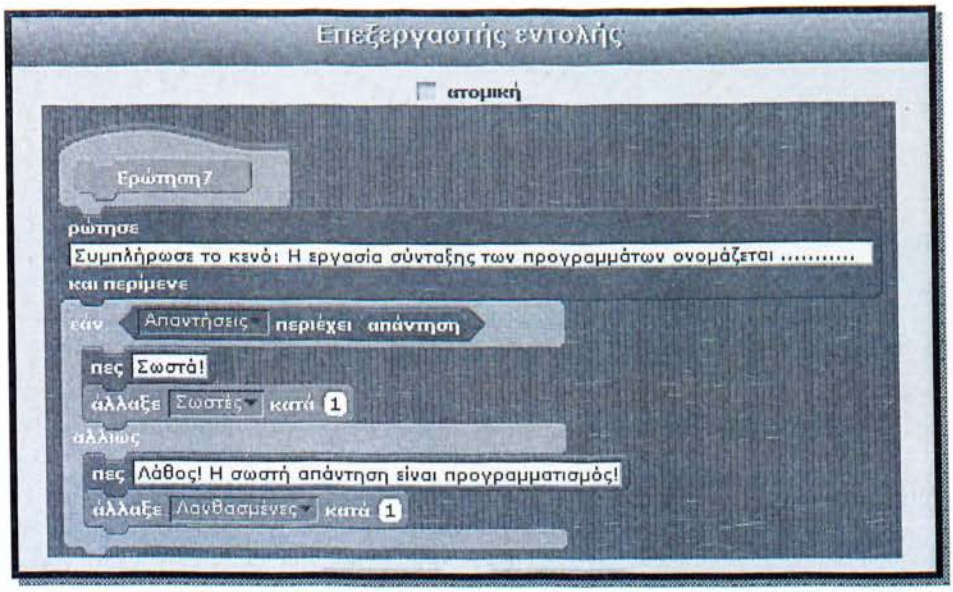


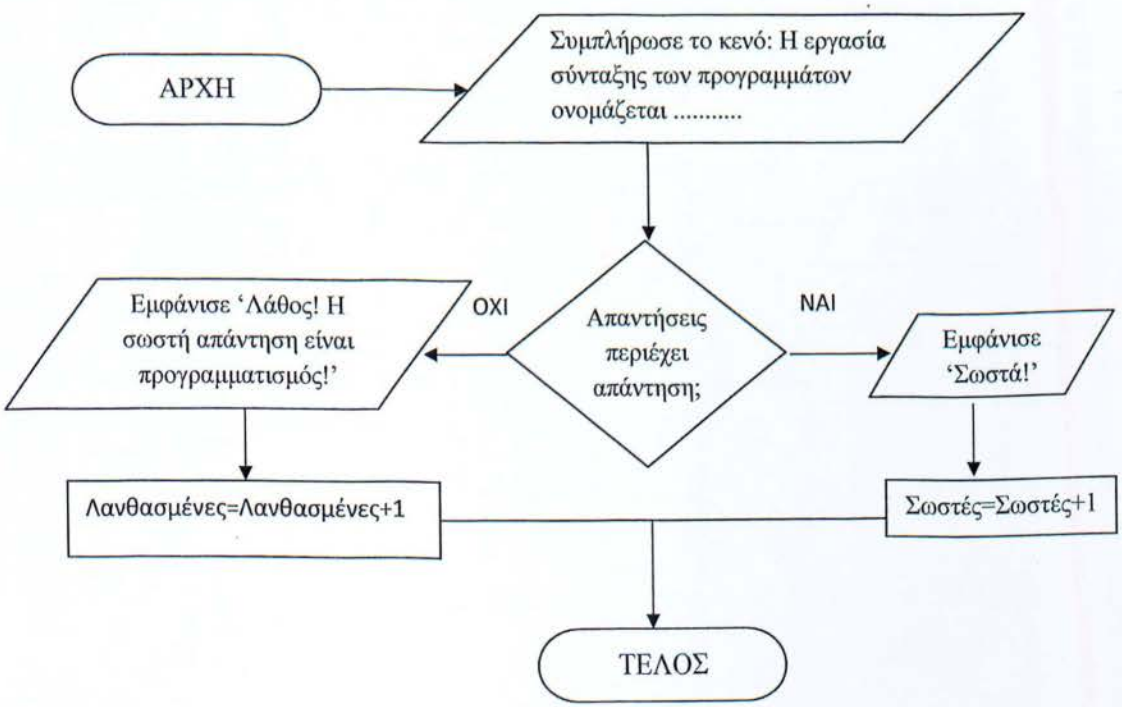
➧ Ερώτηση 6


Επεξεργαστής εντολής
ατομική
Ερώτηση 6
ρώτησε
"Οι μεταγλωττιστές ελέγχουν μία οδηγία κάθε φορά, την εκτελούν και μετά ελέγχουν την επόμενη." Σωστό ή Λάθος;
και περίμενε
εάν απάντηση = Λάθος
πες Μπράβο!
άλλαξε Σωστές κατά 1
αλλιώς
πες Απάντησες λάθος!

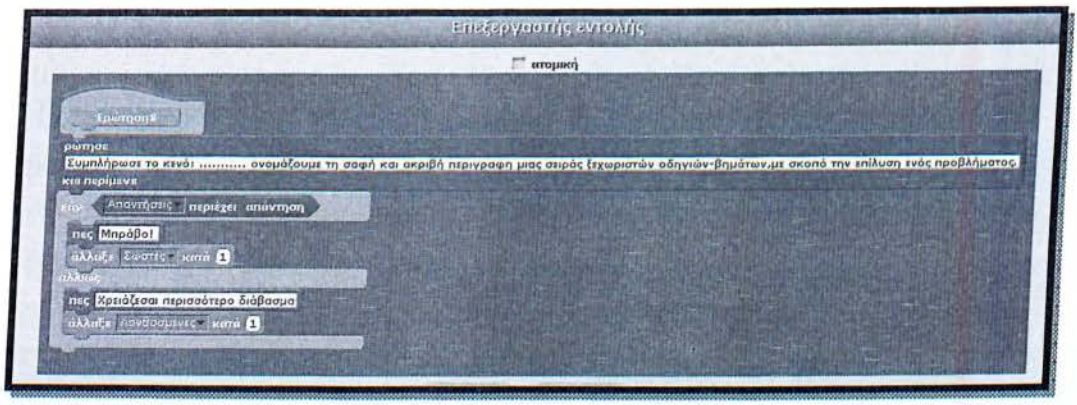

ΑΡΧΗ
"Οι μεταγλωττιστές ελέγχουν μία οδηγία κάθε φορά, την εκτελούν και μετά ελέγχουν την επόμενη." Σωστό ή Λάθος;
Απάντηση= 'Λάθος'
ΟΧΙ
ΝΑΙ
Εμφάνισε 'Απάντησες λάθος!'
Εμφάνισε 'Μπράβο!'
ανθασμένες=Λανθασμένες+1
Σωστές=Σωστές+1
ΤΕΛΟΣ

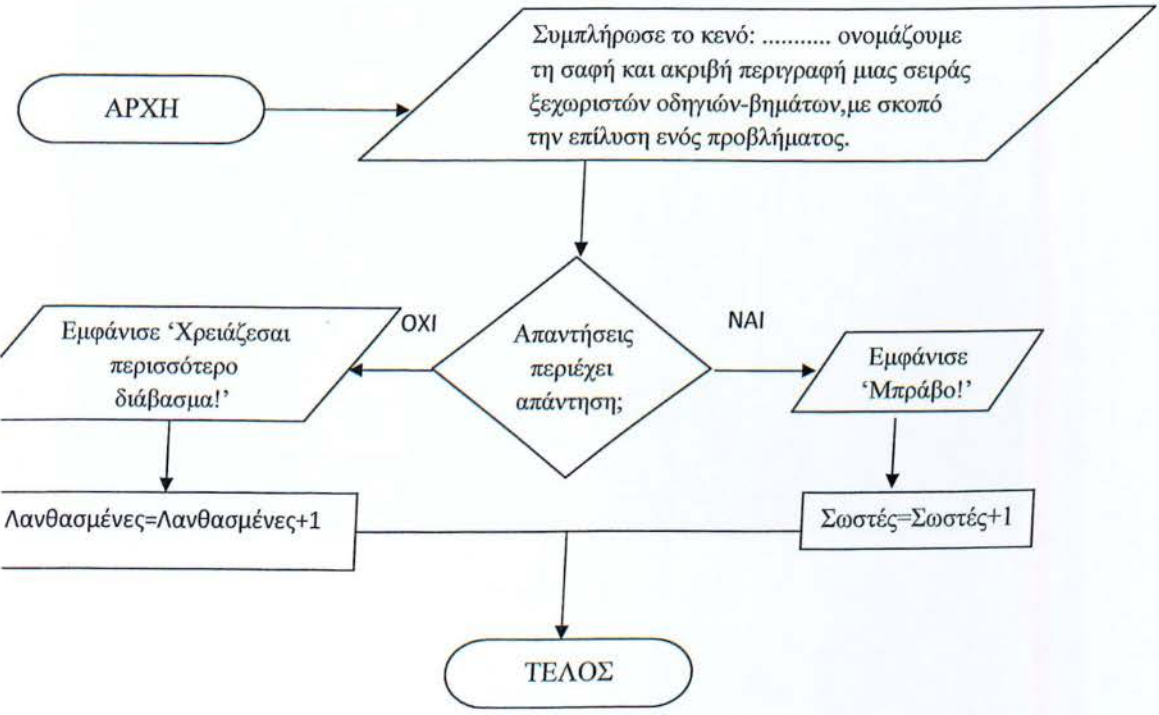
- Ερώτηση 7


Επεξεργαστής εντολής
ατομική
Ερώτηση 7
ρώτησε
Συμπλήρωσε το κενό: Η εργασία σύνταξης των προγραμμάτων ονομάζεται .....
και περίμενε
εάν Απαντήσεις περιέχει απάντηση
πες Σωστά!
άλλαξε Σωστές κατά 1
αλλιώς
πες Λάθος! Η σωστή απάντηση είναι προγραμματισμός!
άλλαξε Λανθασμένες κατά 1


ΑΡΧΗ
Συμπλήρωσε το κενό: Η εργασία σύνταξης των προγραμμάτων ονομάζεται .....
Απαντήσεις περιέχει απάντηση;
ΟΧΙ
ΝΑΙ
Εμφάνισε 'Λάθος! Η σωστή απάντηση είναι προγραμματισμός!'
Εμφάνισε 'Σωστά!'
Λανθασμένες=Λανθασμένες+1
Σωστές=Σωστές+1
ΤΕΛΟΣ

- Ερώτηση 8


Επεξεργαστής εντολής
ατομική
Ερωτηση8
ρωτησε
Συμπλήρωσε το κενό: ..... ονομάζουμε τη σαφή και ακριβή περιγραφη μιας σειράς ξεχωριστών οδηγιών-βημάτων,με σκοπό την επίλυση ενός προβλήματος.
και περίμενε
εαν Απαντήσεις περιέχει απάντηση
πες Μπράβο!
άλλαξε Σωστές κατά 1
αλλιώς
πες Χρειάζεσαι περισσότερο διάβασμα
άλλαξε Λανθασμένες κατά 1


ΑΡΧΗ
Συμπλήρωσε το κενό: ..... ονομάζουμε τη σαφή και ακριβή περιγραφή μιας σειράς ξεχωριστών οδηγιών-βημάτων,με σκοπό την επίλυση ενός προβλήματος.
Απαντήσεις περιέχει απάντηση;
ΟΧΙ
ΝΑΙ
Εμφάνισε 'Χρειάζεσαι περισσότερο διάβασμα!'
Εμφάνισε 'Μπράβο!'
Λανθασμένες=Λανθασμένες+1
Σωστές=Σωστές+1
ΤΕΛΟΣ

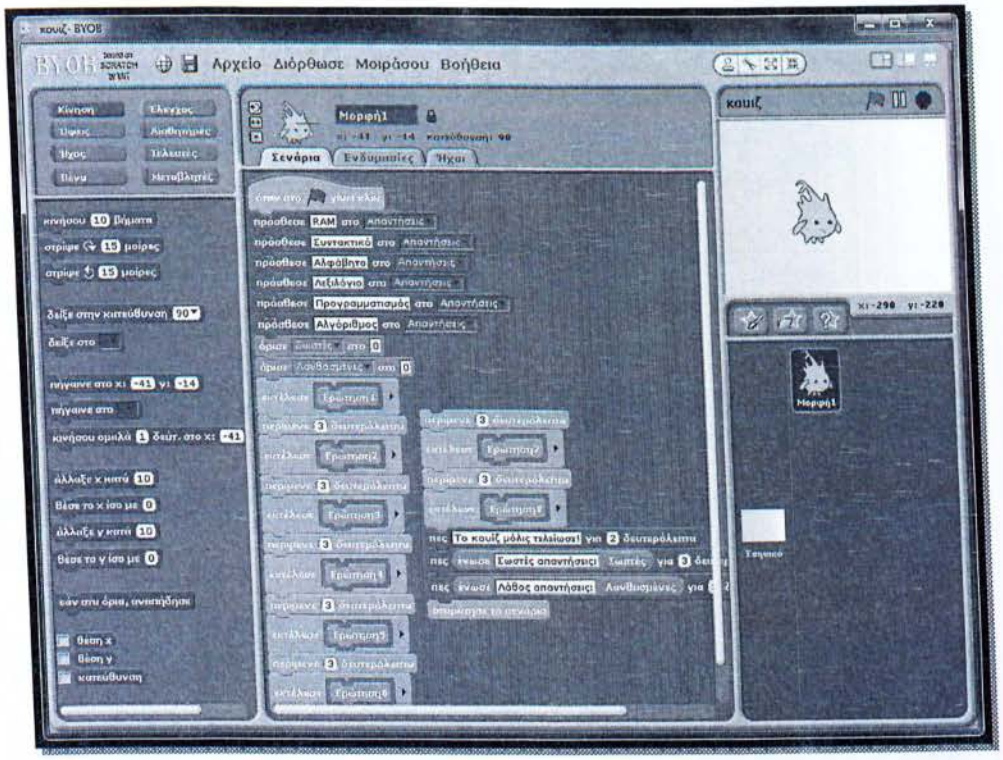
Για την εκτέλεση των μπλοκ τοποθετούμε την εντολή 

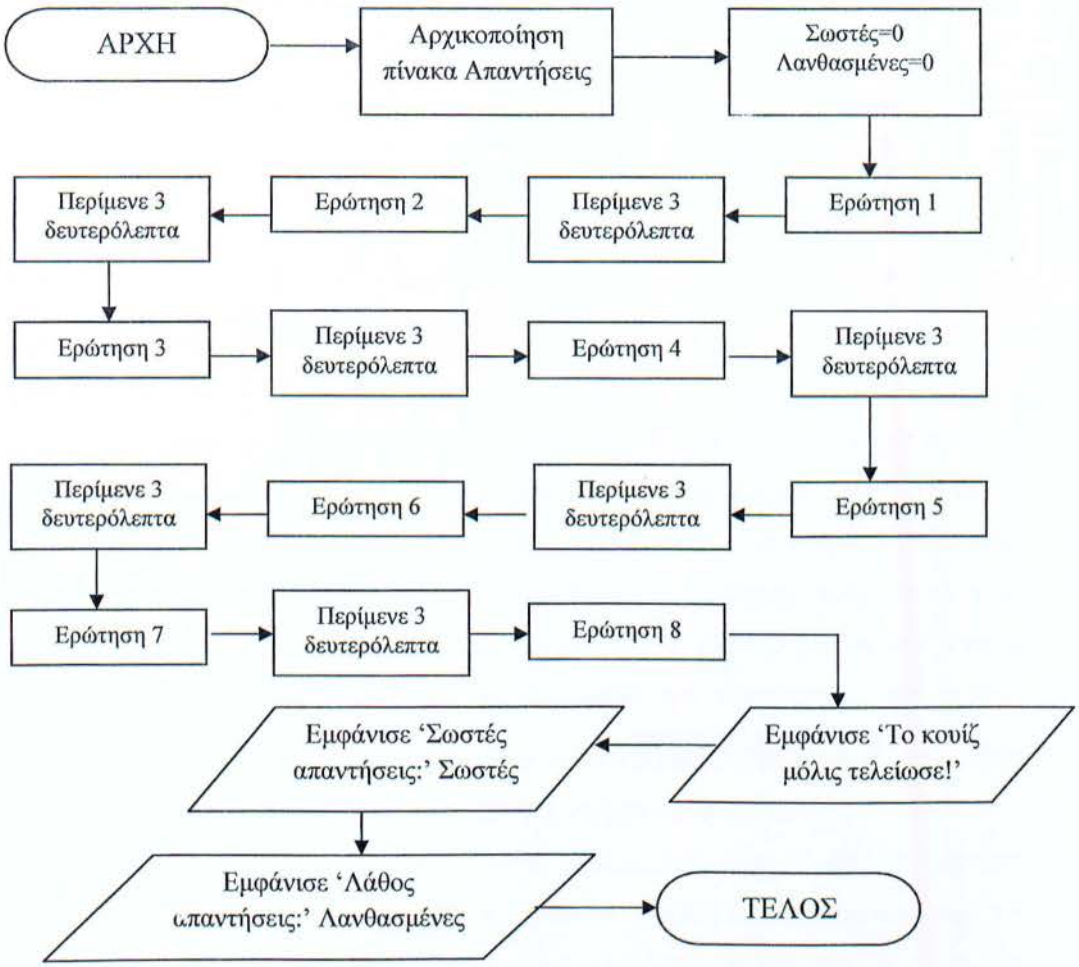
τόσες φορές όσες είναι οι ερωτήσεις μας και στο εσωτερικό της εντολής βάζουμε κάθε φορά το μπλοκ της ερώτησης. Στο τέλος τοποθετούμε την εντολή 

, με την οποία εμφανίζονται πόσες σωστές απαντήσεις έχουμε και την εντολή 

για το πλήθος των λανθασμένων απαντήσεων.

Ολοκληρωμένος ο κώδικας με το διάγραμμα ροής είναι παρακάτω:



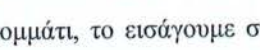


4.1.5 Παίζοντας με τις μπάλες

Το ακόλουθο παράδειγμα είναι ένα παιχνίδι το οποίο συνδυάζει ήχο και κίνηση. Υπάρχουν τρεις μπάλες με διαφορετικό μέγεθος η κάθε μία. Οι μπάλες χοροπηδούν ταυτόχρονα προς όλες τις κατευθύνσεις. Κάθε φορά που ο χρήστης πετυχαίνει μία μπάλα με το ποντίκι ακούγεται ένας ήχος και το σκορ αυξάνεται.

Αρχικά, εισάγουμε το σκηνικό του έργου, επιλέγουμε το «brick-wall2» που περιέχεται στο φάκελο «outdoors» στην βιβλιοθήκη του Scratch. Έπειτα, εισάγουμε τα αντικείμενα μας, τρεις μπάλες οι οποίες θα κινούνται συνεχώς. Θα διαλέξουμε μπάλες με διαφορετικό μέγεθος. Οπότε μπορούμε να χρησιμοποιήσουμε μία μπάλα

θαλάσσης ως την μεγαλύτερη, μια μπάλα του μπάσκετ ως μεσαίου μεγέθους και μία μπάλα του τένις ως την μικρότερη μπάλα.

Στη συνέχεια, θα προσθέσουμε ήχο στο παράδειγμα μας. Πηγαίνουμε στην καρτέλα Σενάρια του σκηνικού, τοποθετούμε την εντολή  , για να ορίσουμε πότε θα ξεκινάει η μουσική και έπειτα την εντολή  , ώστε η μουσική να επαναλαμβάνεται συνεχώς. Στο εσωτερικό της επανάληψης τοποθετούμε την εντολή  . Το μουσικό κομμάτι, το εισάγουμε στην καρτέλα Ήχοι από την βιβλιοθήκη ήχων του Scratch.

Ας δούμε τον κώδικα των αντικειμένων μας. Πρώτα απ' όλα δημιουργούμε μια μεταβλητή από την παλέτα Μεταβλητές με το όνομα score, στην οποία θα αποθηκεύονται οι πόντοι που συλλέγει κάθε παίκτης. Κάθε φορά που γίνεται κλικ πάνω σε μία μπάλα, οι πόντοι πρέπει να αυξάνονται. Αν ο παίκτης πετύχει την μπάλα θαλάσσης κερδίζει 1 πόντο, την μπάλα του μπάσκετ 2 και την μπάλα του τένις 4 πόντους. Επομένως, επιλέγουμε την πρώτη μπάλα και τοποθετούμε την εντολή  και στη συνέχεια την εντολή  , ώστε κάθε φορά που ξεκινάει το πρόγραμμα η μεταβλητή να έχει μηδενική τιμή.

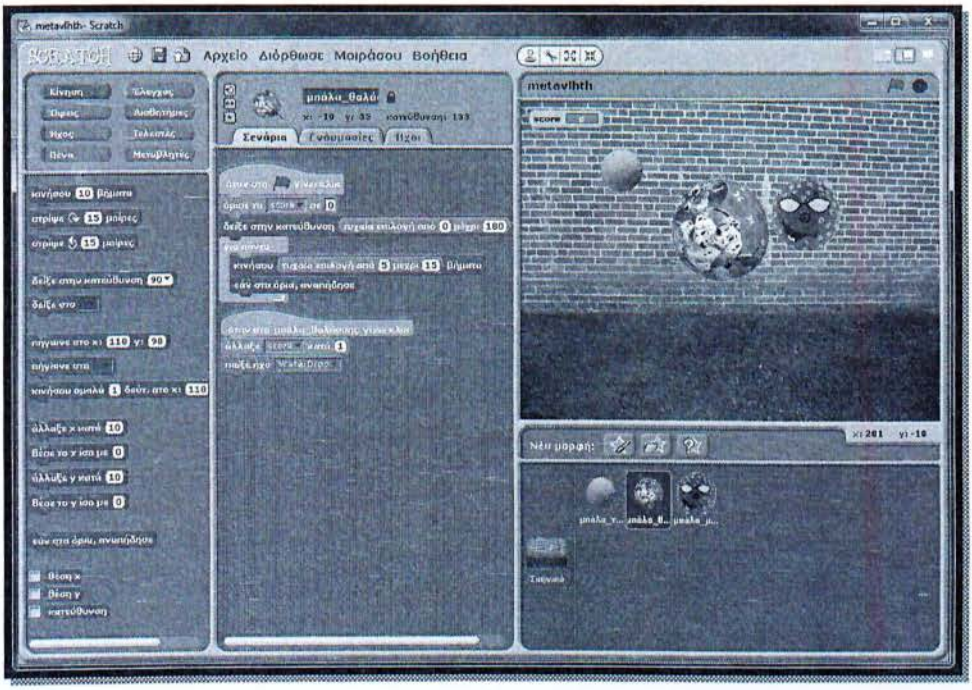
Οι εντολές κίνησης και για τις τρεις μπάλες είναι ίδιες. Αυτό που θέλουμε είναι οι μπάλες να κινούνται από την έναρξη του παιχνιδιού μέχρις ότου ο παίκτης να τερματίσει ο ίδιος το παιχνίδι. Όταν φτάσουν στα όρια της σκηνής, θα πρέπει να συνεχίσουν να κινούνται αλλάζοντας κατεύθυνση. Οπότε, τοποθετούμε την εντολή  για να ορίσουμε τα όρια που θα κινείται η κάθε μπάλα. Εφόσον η κίνηση θα είναι συνεχής, χρησιμοποιούμε την εντολή  και στο εσωτερικό της προσθέτουμε τις εντολές  , για να κινηθεί η μπάλα με τυχαία αρχική κατεύθυνση και με μη σταθερά βήματα σε κάθε επανάληψη, και  , για να αναπηδήσει και να αλλάξει κατεύθυνση στα όρια.

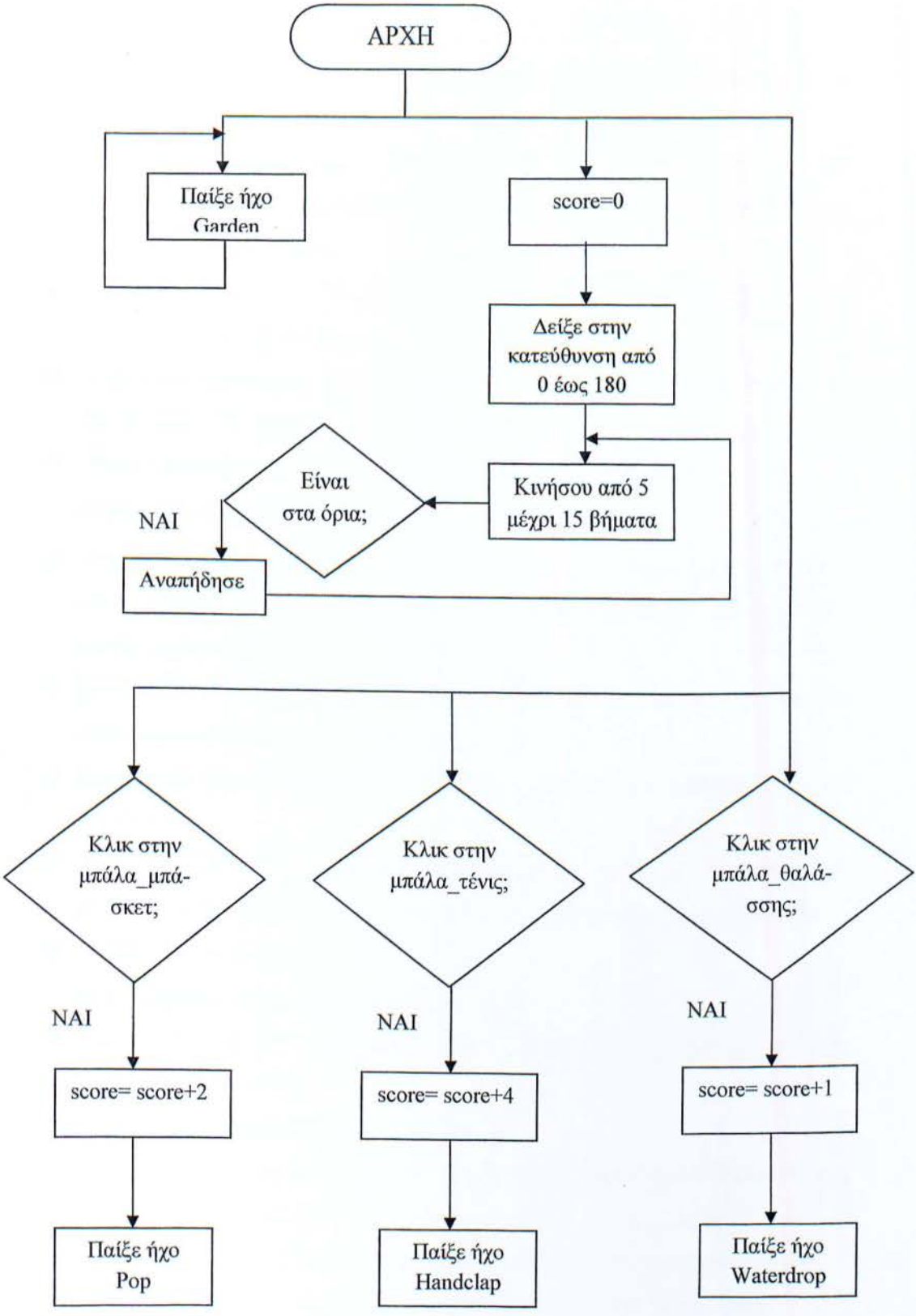
Τέλος, κάθε φορά που θα γίνεται κλικ σε μία μπάλα θέλουμε να αυξάνονται οι πόντοι και να ακούγεται ένα ήχος. Οπότε θα χρησιμοποιήσουμε την εντολή  , έπειτα την εντολή  για την

αύξηση της τιμής της μεταβλητής score και τελευταία την εντολή παίξε ήχο HandClap, για να ακούγεται ο ήχος.

Για τις μπάλες του μπάσκετ και της θαλάσσης προσθέτουμε τις εντολές:

Ολοκληρωμένο το παράδειγμα και το διάγραμμα ροής ακολουθούν παρακάτω:




ΑΡΧΗ
Παίξε ήχο Garden
score=0
Δείξε στην κατεύθυνση από 0 έως 180
Κινήσου από 5 μέχρι 15 βήματα
Είναι στα όρια;
ΝΑΙ
Αναπήδησε
Κλικ στην μπάλα_μπάσκετ;
Κλικ στην μπάλα_τένις;
Κλικ στην μπάλα_θαλάσσης;
ΝΑΙ
ΝΑΙ
ΝΑΙ
score= score+2
score= score+4
score= score+1
Παίξε ήχο Pop
Παίξε ήχο Handclap
Παίξε ήχο Waterdrop

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Μπαβελής Ανδρέας. Οι νέες τεχνολογίες στην εκπαίδευση: Προβλήματα και Προοπτικές
- Γιώργος Παπαευθυμίου. Διπλωματική Εργασία: Διδακτική του Προγραμματισμού για μαθητές Γυμνασίου με χρήση του εργαλείου γραφικού προγραμματισμού Scratch
- Δήμητρα Πούλου. Πτυχιακή Εργασία: Ο προγραμματισμός στην Πρωτοβάθμια Εκπαίδευση και στην Προσχολική Ηλικία
- Δουλούδης Θεοχάρης. Πτυχιακή Εργασία: Εκπαιδευτική Ρομποτική – Οι Παγκόσμιοι Ολυμπιακοί Αγώνες Πληροφορικής
- Πληροφορική και εφαρμογές στην καθημερινή ζωή, Εκπαίδευση https://dsepwiki.wikispaces.com
- Γιώργος Φεσάκης & Αγγελική Δημητρακοπούλου. Επισκόπηση του χώρου των εκπαιδευτικών περιβαλλόντων προγραμματισμού υπολογιστών: τεχνολογικές και παιδαγωγικές προβολές
- Βασίλης Κόμης. Εισαγωγή στις εκπαιδευτικές εφαρμογές των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών
- Real Centre of English. Back to School. Τα παιδιά και η ψυχολογία τους κατά τον Piaget
- Βασίλειος Εφόπουλος, Γεώργιος Ευαγγελίδης, Βασίλειος Δαγδιλέλης & Αλέξανδρος Κλεφτοδήμος. Οι Δυσκολίες των Αρχάριων Προγραμματιστών
- Θεωρίες για τη μάθηση http://www.deutsch.gr/img/theoriesmathisis.pdf
- Γιώτα Πλιακού. Οι Συμπεριφοριστικές Τεχνικές της Θετικής και της Αρνητικής Ενίσχυσης http://www.ibrt.gr/ekpaideysi/2_enisxisi.pdf
- Μάργαρης & Παπαστεργίου. 4ο Πανελλήνιο Συνέδριο Διδακτική της Πληροφορικής. Εισάγοντας αρχάριους στον προγραμματισμό με τα περιβάλλοντα Kara: Μια προσέγγιση βασισμένη στη θεωρία υπολογισμού
- Θωμάς Κακλαμάνης. Συνεργατική μάθηση και ΤΠΕ στην Εκπαίδευση

http://www.pi-schools.gr/download/publications/epitheorisi/teyxos10/130-144.pdf

- Χρήση εκπαιδευτικού λογισμικού
  http://users.sch.gr/nikbalki/epim_kse/files/Basic/Enotita_3_eid.pdf
- Π.Τ.Δ.Ε. Σημειώσεις Σεμιναρίου «Τα μήλα των Εσπερίδων», 2004-05
  http://www.theodoros.gr/seimiwseis/70_Logo_mila.pdf
- Ξυνόγαλος Στέλιος. 2ο Συνέδριο στη Σύρο – ΤΠΕ στην Εκπαίδευση. Σενάρια διδασκαλίας του προγραμματισμού στη δευτεροβάθμια εκπαίδευση
- Γεώργιος Φεσάκης. Διδακτορική Διατριβή: Εκπαιδευτική Αξιοποίηση Υπολογιστικών Περιβαλλόντων Μοντελοποίησης και Ειδικότερα των Σχεσιακών Συστημάτων Διαχείρισης Βάσεων Δεδομένων
  http://ltee.org/uploads/ltee_pubs/gf_2003a_PHD_V_4_FINAL.pdf
- Σταυρούλα Γεωργαντάκη. Διδακτορική Διατριβή: Μία Τεχνολογικά Υποστηριζόμενη Διδακτική Προσέγγιση για τον Αντικειμενοστραφή Προγραμματισμό
- Ελένη Ελευθεριώτη, Ανθή Καρατράντου & Χρήστος Παναγιωτακόπουλος. Χρησιμοποιώντας τα Lego Mindstorms NXT για τη διδασκαλία του προγραμματισμού σε ένα διαθεματικό πλαίσιο: μια πιλοτική μελέτη
- Ε. Βράχνος & Α. Τζιμογιάννης. Σχεδιασμός ενός Περιβάλλοντος Δυναμικής Προσομοίωσης Οπτικοποίησης Αλγορίθμων: Το σύστημα DAVE
- Γλωσσομάθεια http://spinet.gr/glossomatheia/
- Ο Διερμηνευτής της Γλώσσας http://alkisg.mysch.gr/
- Καψιμαλή Βασιλική. Μεταπτυχιακή Διπλωματική Εργασία: Τεχνολογικά Υποστηριζόμενη Διδακτική της Πληροφορικής με χρήση του εργαλείου Scratch
- 7ο Πανελλήνιο Συνέδριο Καθηγητών Πληροφορικής. Επικοινωνία μεταξύ αντικειμένων σε σύγχρονα περιβάλλοντα εισαγωγής στον προγραμματισμό
- Alice:
  http://www.alice.org και http://www.cs.duke.edu/csed/alice/aliceInSchools/
- Scratch:
  http://scratch.mit.edu/

- BYOB:
  http://byob.berkeley.edu/
- Ομάδα Φοιτητών, (2010), Δημιουργώ παιχνίδια στο Scratch. Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων, Πανεπιστήμιο Θεσσαλίας. http://www.scratchplay.gr/index.html
- Στυλιανός-Μαρίνος Χαραλαμπίδης. Προσαρμογή του σχολικού βιβλίου Πληροφορικής της Γ' Γυμνασίου στο περιβάλλον προγραμματισμού Scratch
- Εκπαιδευτικό υλικό για το μάθημα της Πληροφορικής στο Γυμνάσιο
  http://blogs.sch.gr/daskalakis