

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ-ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ



ΕΚΠΑΙΔΕΥΤΙΚΗ ΡΟΜΠΟΤΙΚΗ-ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ ΜΕ
SCRIBBLER

Δημήτρης Ι. Κοντογιάννης

Επιβλέπων: Γεώργιος Παλαιγεωργίου

Επιβλέπων: Ηλίας Χούστης

Περιεχόμενα

1. Εισαγωγή.....	4
1.1 Ρομποτική και εκπαιδευτική ρομποτική	4
1.2 Το Scribbler	4
1.3 Περίληψη	5
2. Μαθησιακές δυσκολίες	6
2.1 Ο προγραμματισμός ως δραστηριότητα επίλυσης προβλήματος.....	6
2.2 Έμπειροι vs αρχάριοι προγραμματιστές.....	7
2.3 Διάκριση γνώσεων προγραμματισμού.....	7
2.4 Η εκμάθηση του προγραμματισμού	8
2.5 Πλεονεκτήματα από την εκμάθηση του προγραμματισμού	9
2.6 Διδακτικές προσεγγίσεις στον προγραμματισμό.....	10
2.7 Μαθησιακές δυσκολίες και παρανοήσεις σε βασικές προγραμματιστικές έννοιες και δομές 15	
2.7.1 Μαθησιακές δυσκολίες – Εισαγωγή.....	15
2.7.2 Μεταβλητές.....	16
2.7.3 Οι δομές ελέγχου και η έννοια της επιλογής	17
2.7.4 Δομές Επανάληψης.....	19
2.7.5 Αναδρομή.....	20
2.7.6 Πίνακες.....	21
2.7.7 Εντολές εισόδου-εξόδου	21
3. Προγραμματιστικά περιβάλλοντα για αρχάριους προγραμματιστές.....	23
3.1 Ταξινόμηση	23
3.2 Διδακτικά συστήματα προγραμματισμού	23
3.3 Μηχανισμοί του προγραμματισμού.....	23
3.3.1 Εκφράζοντας ένα πρόγραμμα	24
3.3.2 Δομώντας προγράμματα.....	34
3.3.3 Κατανοώντας την εκτέλεση ενός προγράμματος	36
3.4 Υποστήριξη μάθησης.....	39
3.4.1 Συνεργατική μάθηση.....	39
3.4.2 Παροχή περιεχομένου με επιπλέον κίνητρο	41
3.5 Συστήματα ενδυνάμωσης.....	42
3.5.1 Ο κώδικας είναι πολύ δύσκολος.....	42
3.5.2 Βελτιώνοντας τις γλώσσες προγραμματισμού	47
3.5.3 Δραστηριότητες που ενισχύει ο προγραμματισμός	53
3.6 Σύνοψη κεφαλαίου	55

4. Διδακτικό πλάνο.....	58
4.1 Περιγραφή διδακτικού πλάνου.....	58
4.2 Πλαίσιο εκμάθησης: «Ερευνητική δραστηριότητα στον πλανήτη Άρη».....	58
4.2.1 Στόχοι διδασκαλίας.....	58
4.2.2 1η Διδακτική ώρα	61
4.2.3 2η Διδακτική ώρα	73
4.2.4 3η Διδακτική ώρα	82
5.Εφαρμογή διδασκαλίας και αξιολόγηση.....	92
5.1 Εφαρμογή διδασκαλίας.....	92
5.1.1 Προφίλ ομάδων	93
5.2 Μέθοδοι αξιολόγησης.....	95
5.3 Αποτελέσματα αξιολόγησης.....	96
5.3.1 Ποιοτική ανάλυση στην εφαρμογή της διδασκαλίας	96
5.3.2 Αποτελέσματα φύλλου αξιολόγησης έμπειρων χρηστών.....	106
5.3.3 Σύγκριση αποτελεσμάτων φύλλων αξιολόγησης	112
5.3.4 Αποτελέσματα αξιολόγησης με συνεδρίες	119
6.Συζήτηση.....	124
ΠΑΡΑΡΤΗΜΑ	127
Φύλλα αξιολόγησης	127
Φύλλο αξιολόγησης εμπειρίας	127
Φύλλο γενικής αξιολόγησης	130
Φύλλο αξιολόγησης για έμπειρους στον προγραμματισμό	132
Βιβλιογραφία.....	138

1. Εισαγωγή

1.1 Ρομποτική και εκπαιδευτική ρομποτική

Οι αυξημένες ανάγκες της κοινωνίας για τεχνολογικά ενήμερους πολίτες επιβάλλουν τη διερεύνηση νέων υλικών αλλά και μεθόδων που θα εξοικειώσουν τους μαθητές με τη λειτουργική χρήση της τεχνολογίας και θα τους επιτρέψουν να διερευνούν βαθύτερα τον πραγματικό κόσμο. Η εκπαιδευτική ρομποτική αποτελεί ένα σύγχρονο εκπαιδευτικό περιβάλλον, όπου ο χρήστης είναι σε θέση, με τη βοήθεια μιας απλής γλώσσας προγραμματισμού, να συνθέσει και να κατευθύνει μια τεχνολογική οντότητα. Πέρα από την προφανή συμβολή της στον τεχνολογικό αλφαριθμητισμό, η εκπαιδευτική ρομποτική μπορεί να βρει άμεση εφαρμογή στις φυσικές επιστήμες, στα μαθηματικά και στη λύση προβλήματος, ενώ οι έρευνες δείχνουν ότι ταυτόχρονα προάγει τη συνεργατική μάθηση, την αυτοπεποίθηση και τη δημιουργικότητα των παιδιών[1].

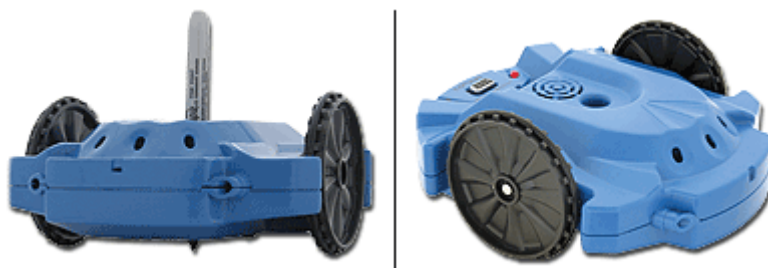
Αρχικά, η ρομποτική χρησιμοποιήθηκε στο χώρο της βιομηχανίας για τον αυτοματισμό και την άρση βαρέων φορτίων [2]. Αργότερα, ο Seymour Papert απλοποίησε τη ρομποτική για εκπαιδευτικούς σκοπούς χρησιμοποιώντας τη γλώσσα προγραμματισμού Logo σε συνδυασμό με το υλικό της εταιρείας Lego[3]. Σε ένα περιβάλλον εκπαιδευτικής ρομποτικής (E.P.), μια τεχνολογική οντότητα όπως π.χ. ένα μοντέλο αυτοκινήτου-ρομπότ, κατευθύνεται από το χρήστη, με τη βοήθεια μια γλώσσας προγραμματισμού. Όπως σημειώνει ο Δαγδιλέλης και οι συνεργάτες του [4], η ρομποτική οντότητα μπορεί να είναι εικονική, δηλαδή να απεικονίζεται μόνο στην οθόνη του υπολογιστή ή πραγματική και να δρα στο έδαφος. Η E.P., μπορεί να βρει εφαρμογή στις φυσικές επιστήμες, στα μαθηματικά, στη λύση προβλήματος, αλλά παράλληλα μπορεί να βελτιώσει τις δεξιότητες συνεργασίας, την αυτοπεποίθηση, τη δημιουργικότητα, τα κίνητρα των μαθητών και τις δεξιότητες χειρισμού του υπολογιστή[5]. Επιπλέον, η E.P. παρέχει στους μαθητές το κατάλληλο περιβάλλον για να διερευνήσουν τα προβλήματα του πραγματικού κόσμου, συμβάλλοντας έτσι στη δημιουργία τεχνολογικά ενημερωμένων πολιτών σύμφωνα με τις ανάγκες της σύγχρονης κοινωνίας.

Στις μέρες μας, μεγάλα εκπαιδευτικά ιδρύματα παρέχουν εναλλακτικά courses διδασκαλίας προγραμματισμού με τη χρήση ρομπότ, εκτιμώντας την ως τον πλέον ενδεδειγμένο τρόπο για μια εισαγωγή στην προγραμματιστική λογική.

1.2 To Scribbler

Μια τέτοια ρομποτική εφαρμογή που χρησιμοποιείται κατά κόρον για την διδασκαλία προγραμματισμού σε αρχάριους, είναι το Scribbler robot.

Απόψεις του Scribbler robot



Το Scribbler robot κατασκευάστηκε από την Parallax με σκοπό το κόστος του να διατηρηθεί σε χαμηλό επίπεδο έτσι ώστε να μπορεί ένα εκπαιδευτικό ίδρυμα να διαθέσει το απαιτούμενο ποσό για τη προμήθεια πολλών τέτοιων μοντέλων (Οκτώβριος 2009: \$79.95) [6]. Το ρομπότ συνοδεύει -εκτός από το απαραίτητο εγχειρίδιο χρήσης, τους οδηγούς εγκατάστασης και το καλώδιο σύνδεσης με Η/Υ- μια σειρά από μαρκαδόρους που προσαρτώνται στο κέντρο του και του δίνουν τη δυνατότητα να ζωγραφίσει.

Αναλυτικότερα, το Scribbler διαθέτει θύρες και ελεγκτή υπερύθρων, αισθητήρες φωτός, υποδοχή για μαρκαδόρο στο μέσο του μηχανήματος, ένα μικρό ηχείο καθώς και 3 LEDs. Για τη λειτουργία του καταναλώνονται αλκαλικές μπαταρίες ενώ η σύνδεσή του με Η/Υ είναι σειριακή. Στο εσωτερικό του Scribbler λειτουργεί ένας επαναπρογραμματιζόμενος "εγκέφαλος", ο Basic Stamp 2, ο οποίος μπορεί να λάβει προγράμματα μέσω της σειριακής θύρας είτε από ένα γραφικό περιβάλλον, ειδικά διαμορφωμένο για αρχάριους σε θέματα προγραμματισμού, είτε από τον Basic Stamp Editor, ένα πρόγραμμα βασισμένο στην Pbasic που απευθύνεται σε έμπειρους προγραμματιστές. Η Pbasic είναι μια γλώσσα που αναπτύχθηκε ειδικά για τη διαχείριση του Basic Stamp και η δομή της μοιάζει αρκετά με τις περισσότερες γλώσσες προγραμματισμού BASIC.

Με τη βοήθεια αυτών των δύο εργαλείων, το Scribbler Robot μπορεί να κινείται στο χώρο, να ζωγραφίζει σχέδια, να παράγει ήχους, να ανιχνεύει φωτεινά σημεία σε ένα δωμάτιο, να αποφεύγει εμπόδια κ.ά..

1.3 Περίληψη

Η παρούσα έρευνα ξεκινά με την παρουσίαση, εξέταση και ανάλυση των τυπικών μαθησιακών δυσκολιών που αντιμετωπίζουν οι μαθητές κατά την διδασκαλία του προγραμματισμού (2^ο Κεφάλαιο). Ακολουθεί το 3^ο Κεφάλαιο όπου και ταξινομούνται τα προγραμματιστικά εργαλεία που δημιουργήθηκαν ώστε να εξαλείψουν αυτές τις δυσκολίες καθώς και οι τεχνικές που χρησιμοποίησαν οι κατασκευαστές τους. Στο 4^ο κεφάλαιο καταστρώνεται ένα πλήρες διδακτικό πλάνο για διδασκαλία του προγραμματισμού με την βοήθεια του Scribbler robot. Το πλάνο αυτό εφαρμόστηκε σε τρεις ομάδες μαθητών με διαφορετικό επίπεδο προγραμματιστικών γνώσεων. Η παρουσίαση της εφαρμογής της διδασκαλίας, των ιδιαιτεροτήτων που παρουσιάστηκαν σε αυτήν καθώς και η αξιολόγηση της περιλαμβάνονται στο 5^ο κεφάλαιο. Η έρευνα ολοκληρώνεται με μια γενικότερη θεώρηση των αποτελεσμάτων της εφαρμοσθείσας διδασκαλίας και της εμπειρίας που αποκομίστηκε από αυτήν.

2. Μαθησιακές δυσκολίες

2.1 Ο προγραμματισμός ως δραστηριότητα επίλυσης προβλήματος

Για πολλά χρόνια, λανθασμένα, η διδασκαλία της πληροφορικής και ειδικότερα του προγραμματισμού ήταν ταυτισμένη με τη διδασκαλία μιας γλώσσας προγραμματισμού, ενώ τώρα πλέον το ενδιαφέρον εστιάζεται στις μορφές συλλογισμού που χρησιμοποιούν οι έμπειροι και αρχάριοι προγραμματιστές καθώς και στις μεθόδους για την επίλυση προγραμματιστικών προβλημάτων. Γι αυτό το λόγο ο προγραμματισμός πλέον μελετάται στο πλαίσιο ανθρώπινων δραστηριοτήτων που αφορούν την οικοδόμηση μεθόδων και τεχνικών και την ανακάλυψη ή τη βελτιστοποίηση αλγορίθμων.

Υπάρχουν διάφορες αντιλήψεις για το τι ακριβώς είναι ένα πρόγραμμα και κατά συνέπεια διαφορετικές προσεγγίσεις για τη διδασκαλία του προγραμματισμού. Η προσέγγιση που χρησιμοποιείται ακόμα στη δευτεροβάθμια εκπαίδευση είναι ότι το πρόγραμμα περιγράφει ένα σύνολο από δυνατούς υπολογισμούς. Μια άλλη αντίληψη προσδιορίζει τα προγράμματα ως συναρτήσεις που δέχονται εισόδους (inputs) και έναν κανόνα με τον οποίο συνδυάζονται οι εισοδοί για να παράγουν μια τιμή ή έξοδο (output). Η Τρίτη και πιο πρόσφατη αντίληψη του προγράμματος, συνίσταται στον ορισμό από τον προγραμματιστή αντικειμένων και σχέσεων μεταξύ τους. Ωστόσο σε όλες τις προσεγγίσεις, ο προγραμματιστής βρίσκει μπροστά σε έναν προς επίτευξη στόχο. Συνεπώς, ο προγραμματισμός συνίσταται στην οργάνωση των διαδικασιών που θα επιτρέψουν την επίτευξη του στόχου με τη χρήση μιας γλώσσας προγραμματισμού πάνω σε μια συγκεκριμένη μηχανή.

Η δραστηριότητα του προγραμματισμού στην ψυχολογία θεωρείται ως ένα έργο επίλυσης προβλήματος. Η επίλυση του προβλήματος συνίσταται στην ανάπτυξη στρατηγικών και στην εγκαθίδρυση διαδικασιών που επιτρέπουν τη σύνδεση ανάμεσα στην αρχική και την τελική κατάσταση. Η ομάδα McCracken καθόρισε την στρατηγική επίλυσης προβλήματος ως διαδικασία πέντε βημάτων:

- αφαιρετική προσέγγιση του προβλήματος από την περιγραφή του,
- παραγωγή υποπροβλημάτων,
- μετατροπή των υποπροβλημάτων σε υπο-λύσεις,
- επανασύνθεση και
- αξιολόγηση και επανάληψη.

Δεν υπάρχουν αποδείξεις ότι η εκμάθηση προγραμματισμού προωθεί τη βελτίωση γενικών δεξιοτήτων επίλυσης προβλημάτων, αν και μπορεί να βελτιώσει ικανότητες όπως η μετάφραση του προβλήματος σε εξισώσεις.

Στο πλαίσιο αυτό, βασικός στόχος της διδασκαλίας του προγραμματισμού πρέπει να είναι η ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων, δηλαδή η ανάπτυξη της ικανότητας των μαθητών να εφαρμόζουν τις γνώσεις τους για την επίλυση προβλημάτων που δεν έχουν διδαχθεί πιο πριν. Η επίλυση προβλήματος είναι μια σύνθετη νοητική διαδικασία που εμπερικλείει το συντονισμό ενός συνόλου από απαιτητικές και αλληλοσυνδεόμενες δεξιότητες. Οι δεξιότητες αυτές περιλαμβάνουν:

- την κατανόηση και αναπαράσταση της αρχικής κατάστασης του προβλήματος, συμπεριλαμβανομένου και του προσδιορισμού των ειδών της πληροφορίας που απαιτούνται για τη λύση του,
- τη συλλογή και οργάνωση της κατάλληλης πληροφορίας,

- την κατασκευή και διαχείριση ενός σχεδίου δράσης ή μιας στρατηγικής και η αναζήτηση ευρετικών τεχνικών,
- το διαχωρισμό ενός σύνθετου προβλήματος σε απλούστερα, η λύση των οποίων είναι ήδη γνωστή,
- τον έλεγχο υποθέσεων και τη λήψη απόφασης.

2.2 Έμπειροι vs αρχάριοι προγραμματιστές

Το να μάθει κάποιος να προγραμματίζει δεν είναι κάτι απλό. Οι ειδικοί συμφωνούν πως χρειάζονται περίπου 10 χρόνια για να μετατραπεί ένας αρχάριος σε έμπειρο προγραμματιστή[7]. Πριν αρχίσουμε την αναφορά στους αρχάριους προγραμματιστές πρέπει να αναφερθούμε στα στοιχεία που διαθέτουν οι έμπειροι.

Οι έμπειροι προγραμματιστές έχουν βαθιά γνώση του θέματός τους και εφαρμόζουν όλα όσα γνωρίζουν. Αναγνωρίζουν εύκολα τα προβλήματα που απαιτούν παρόμοια λύση(πιθανώς με την εφαρμογή προτύπων ή σχημάτων), τείνουν να πλησιάζουν ένα πρόγραμμα μέσω των αντικειμένων και των δομών δεδομένων, είναι γρηγορότεροι και ακριβέστεροι, έχουν καλύτερη συντακτική και σημασιολογική γνώση και καλύτερες τακτικές και στρατηγικές δεξιότητες. Οι έμπειροι προγραμματιστές οργανώνουν τη γνώση τους σύμφωνα με τα λειτουργικά χαρακτηριστικά, όπως είναι η φύση του εκάστοτε αλγορίθμου(παρά στις επιφανειακές λεπτομέρειες όπως η γλωσσική σύνταξη). Εκτός από τις γενικές στρατηγικές επίλυσης προβλημάτων(όπως το διαίρει-και-βασίλευε), χρησιμοποιούν και εξειδικευμένες στρατηγικές και προσεγγίζουν το πρόβλημα από πάνω προς τα κάτω για το αποσυνθέσουν και να το κατανοήσουν. Επίσης, είναι ευέλικτοι στη ν κατανόηση του προβλήματος και είναι πρόθυμοι να εγκαταλείψουν αμφισβητούμενες υποθέσεις. Τέλος, στις στρατηγικές τους περιλαμβάνονται δοκιμές του κώδικα και τεχνικές αποσφαλμάτωσης του.

Σε αντίθεση με τους έμπειρους προγραμματιστές, οι αρχάριοι δεν μπορούν να εφαρμόσουν τη σχετική γνώση που κατέχουν, πάνω στο πρόγραμμα. Δηλαδή ενώ κατέχουν τη θεωρητική γνώση πάνω στο αντικείμενο, όταν καλούνται να μετουσιώσουν αυτή τη γνώση για να φέρουν εις πέρας ένα πρόβλημα, δεν μπορούν να δώσουν μια καθαρή λύση. Τείνουν να προσεγγίζουν το πρόγραμμα γραμμή προς γραμμή και όχι σαν μια ολότητα που αποτελείται από διάφορες δομές. Έτσι επιχειρούν τοπικές μικρές διορθώσεις και δεν εξετάζουν ολόκληρη την επαναδιατύπωση του προβλήματος. Τέλος στερούνται τεχνικών επίλυσης προβλημάτων και ξοδεύουν πολύ λίγο χρόνο στο σχεδιασμό όπως και στη δοκιμή του κώδικα. Εντούτοις, υπάρχουν και άλλες πιθανές εξηγήσεις που εξηγούν γιατί δυσκολεύονται στον προγραμματισμό, όπως για παράδειγμα δεν μπορούν να καταλάβουν τις δομές που χρησιμοποιούνται σε ένα πρόγραμμα(π.χ. πίνακες, αναδρομή)[9]. Στις μαθησιακές δυσκολίες θα αναφερθούμε εκτενέστερα σε επόμενο κεφάλαιο.

2.3 Διάκριση γνώσεων προγραμματισμού

Όπως τονίζεται από τον Davies[10], υπάρχει διάκριση ανάμεσα στη θεωρητική γνώση του προγραμματισμού (για παράδειγμα πως λειτουργεί ένας βρόγχος for) και στις προγραμματιστικές στρατηγικές (για παράδειγμα η σωστή χρησιμοποίηση ενός βρόγχου

for μέσα στο πρόγραμμα). Τα περισσότερα εισαγωγικά εγχειρίδια προγραμματισμού αφιερώνουν το μεγαλύτερο μέρος του περιεχομένου τους στην παρουσίαση της θεωρητικής γνώσης για μια ιδιαίτερη γλώσσα προγραμματισμού. Ένας σημαντικός περιορισμός είναι ότι αποτυγχάνουν συχνά να εξετάσουν τον τρόπο με τον οποίο η γνώση χρησιμοποιείται ή εφαρμόζεται πρακτικά, ενώ θα έπρεπε να υπάρχει αναφορά και ανάλυση των στρατηγικών που υιοθετούνται από τους προγραμματιστές στην παραγωγή και κατανόηση των προβλημάτων.

Μια άλλη σημαντική διάκριση είναι αυτή μεταξύ των μελετών που ερευνούν την κατανόηση προγράμματος (όπου δίνεται ο κώδικας ενός προγράμματος και οι σπουδαστές πρέπει να κατανοήσουν το πως αυτό λειτουργεί), και εκείνων που επικεντρώνονται στην παραγωγή προγραμμάτων (όπου οι σπουδαστές πρέπει να δημιουργήσουν ένα μέρος ή ένα ολόκληρο πρόγραμμα για να εκτελεστεί κάποιος στόχος ή να λυθεί ένα πρόβλημα). Τα θέματα συσχετίζονται, ειδικά επειδή κατά τη διάρκεια της παραγωγής, η ανάπτυξη και η διόρθωση του κώδικα περιλαμβάνει απαραίτητα την αναθεώρηση και κατανόηση του. Όμως υπάρχει πολύ μικρή αντιστοιχία ανάμεσα στην ικανότητα του να διαβάξεις ένα πρόγραμμα και στην ικανότητα να γράψεις. Και τα δύο πρέπει να διδαχθούν μαζί με στρατηγικές αποσφαλμάτωσης και δοκιμής του κώδικα.

Διάφορες πρόσφατες μελέτες ερευνούν τα ζητήματα που σχετίζονται σχετικά με τον αντικειμενοστραφή και τον διαδικαστικό προγραμματισμό. Γενικά δεν υπάρχει κάποιος κανόνας για το ποιος από τους δύο διευκολύνει περισσότερο την κατανόηση. Οι γλώσσες προγραμματισμού που ακολουθούν το διαδικαστικό παράδειγμα, επιτρέπουν μετασχηματισμούς που οδηγούν από μια κατάσταση σε μια άλλη. Το πρόγραμμα είναι, συνεπώς, η έκφραση μιας διαδικασίας που οδηγεί από μια αρχική κατάσταση (τα δεδομένα) σε μια τελική κατάσταση (τα αποτελέσματα) χρησιμοποιώντας τις επιτρεπτές από τη γλώσσα πράξεις. Αντίθετα όμως, όταν χρησιμοποιείται ένα προγραμματιστικό περιβάλλον που ακολουθεί το αντικειμενοστραφές παράδειγμα, η δραστηριότητα του προγραμματιστή συνίσταται σε μια δραστηριότητα σχεδιασμού αντικειμένων ξεκινώντας από ορισμένες ιδιότητές τους. Μερικοί αξιώνουν ότι ο αντικειμενοστραφής προγραμματισμός είναι πιο εύκολος στη χρήση και πιο φυσικός. Τέτοιες αξιώσεις είναι βασισμένες στο επιχείρημα ότι τα αντικείμενα αποτελούν φυσικά στοιχεία του προβλήματος, με αποτέλεσμα να διευκολύνεται ο σχεδιασμός. Από την άλλη μελέτες δείχνουν ότι ο προσδιορισμός των αντικειμένων δεν είναι εύκολη διαδικασία, τα αντικείμενα που αναγνωρίζονται στο πρόβλημα δεν είναι πάντα απαραίτητα στο πρόγραμμα και οι αρχάριοι πρέπει να κατασκευάσουν ένα πρότυπο των διαδικαστικών πτυχών μιας λύσης προκειμένου να σχεδιάσουν κατάλληλα τα αντικείμενα και τις κλάσεις. Σε μια έρευνα[11] που έγινε σε μαθητές που μάθαιναν τη γλώσσα C (διαδικαστική) και σε άλλους που μάθαιναν C++ (αντικειμενοστραφής), βγήκε το συμπέρασμα όταν επρόκειτο για μικρό πρόγραμμα, δεν υπήρξε διαφορά στην κατανόηση μεταξύ των δύο γλωσσών. Τα αποτελέσματα όμως ήταν τελείως διαφορετικά σε μεγάλα προγράμματα, καθώς η δυσκολία που αντιμετώπισαν οι αρχάριοι ήταν στην κατανόηση του ελέγχου ροής του προγράμματος της αντικειμενοστραφούς γλώσσας.

2.4 Η εκμάθηση του προγραμματισμού

Κατά τα πρώτα στάδια της μάθησης του προγραμματισμού, οι μαθητές συμπεριφέρονται χρησιμοποιώντας γνωστικές διεργασίες που τους είναι ήδη οικείες. Προσπαθούν να αντιμετωπίσουν νέες καταστάσεις χρησιμοποιώντας καταρχήν γνωστά νοητικά σχήματα

που προέρχονται κυρίως από τις βιωματικές εμπειρίες τους, αλλά και από τις μαθηματικές γνώσεις τους. Η πιο συνήθης παρατήρηση που αφορά τους αρχάριους προγραμματιστές είναι ότι προγραμματίζουν όπως σκέφτονται, δηλαδή συλλογίζονται με τη λογική της επεξεργασίας με το χέρι, χωρίς να λαμβάνουν υπόψη τους τις λειτουργίες της μηχανής.

Στη συνέχεια της μαθησιακής διαδικασίας, οι νοητικές αυτές δομές εξ ανάγκης τροποποιούνται. Η προσαρμογή αυτή οδηγεί το μαθητή να οικοδομήσει σταδιακά νέες δομές αναπαραστάσεων, να προχωρήσει δηλαδή στην εννοιολογική αλλαγή. Η μελέτη της σύνταξης και της σημασιολογίας μιας γλώσσας δεν είναι επαρκής για να φτάσει ο μαθητής σε αυτό το επίπεδο. Η εκμάθηση των εννοιών και των τεχνικών της νέας γλώσσας γίνεται μόνο με γράψιμο προγραμμάτων. Η εμπειρία ξεκινά με γενικές τεχνικές επίλυσης προβλημάτων και τη βασική γνώση. Η πείρα αυξάνεται με την πρακτική εφαρμογή αυτής της γνώσης στο πρόβλημα. Τα δύο αυτά βήματα επαναλαμβάνονται συνεχώς, καθώς προστίθενται συνεχώς νέα προς λύση προβλήματα.

Το επόμενο βήμα στον προγραμματισμό είναι η δοκιμή και η διόρθωση του κώδικα. Η διαδικασία της αποσφαλμάτωσης εξαρτάται πρώτα από όλα από το αν ο μαθητής είναι σε θέση να διαβάσει και να κατανοήσει τα προγράμματα που ο ίδιος έχει γράψει. Η καλύτερη στρατηγική είναι να γίνεται συχνά δοκιμή σε μικρά κομμάτια κώδικα κατά τη διάρκεια της συγγραφής του προγράμματος, μέσα στα οποία να υπάρχουν και εντολές εκτύπωσης.

Συνοψίζοντας υπάρχει μια ιδανική αλυσίδα για την εκμάθηση του προγραμματισμού:

- εκμάθηση ενός νέου χαρακτηριστικού γνωρίσματος ή στοιχείου κάθε φορά,
- συνδυασμός των παλαιών στρατηγικών με νέες,
- εκμάθηση των γενικών δεξιοτήτων επίλυσης προβλήματος συν αυτών που είναι μοναδικές για κάθε πρόβλημα

Τα πρότυπα είναι κρίσιμα για την οικοδόμηση της κατανόησης. Αν ο εκπαιδευτικός τα παραλείψει, τότε οι μαθητές θα δημιουργήσουν μόνοι τους δικά τους, αμφιβόλου ποιότητας. Ο εκπαιδευτικός πρέπει να διδάξει βασικά παραδείγματα σχεδιασμού, γιατί αλλιώς οι μαθητές όταν συναντούν ένα πρόβλημα θα το αντιμετωπίζουν ως τελείως άγνωστο.

2.5 Πλεονεκτήματα από την εκμάθηση του προγραμματισμού

Όσοι υποστηρίζουν την ανάγκη της διδασκαλίας του προγραμματισμού στην εκπαίδευση, αποδέχονται ότι η μάθηση του προγραμματισμού μπορεί να προκαλέσει αλλαγές στο γνωστικό σύστημα των μαθητών:

- Αυστηρότητα στη σκέψη, ακρίβεια έκφρασης,
- Πρόσκτηση και κατανόηση γενικών εννοιών, όπως διαδικασία, μεταβλητή, συνάρτηση, μετασχηματισμός,
- Μάθηση τεχνικών αναζήτησης λαθών, που μπορεί να μεταφερθεί και σε άλλους εκτός προγραμματισμού χώρους,
- (4)Πρόσκτηση της γενικής ιδέας οικοδόμησης της λύσης με τη μορφή μικρών διαδικασιών ή τμημάτων, τα οποία μπορούν να χρησιμοποιηθούν συνδεδεμένα για την οικοδόμηση της λύσης σύνθετων προβλημάτων,
- Επέκταση της συνειδητοποίησης και της γνώσης πάνω σε τεχνικές επίλυσης προβλημάτων

Ο μαθητής που προγραμματίζει μαθαίνει ταυτόχρονα πολλά πράγματα. Μαθαίνει να κωδικοποιεί και να καθιστά αντικειμενικές τις πράξεις, όντας αναγκασμένος να περιγράψει με ακριβή τρόπο το αποτέλεσμα ενός μετασχηματισμού στον οποίο υπόκειται ένα αντικείμενο. Ο προγραμματισμός είναι μία εξελικτική δραστηριότητα. Η κατασκευή ενός προγράμματος μπορεί να δώσει ιδέες για την κατασκευή άλλων προγραμμάτων.

2.6 Διδακτικές προσεγγίσεις στον προγραμματισμό

Σύμφωνα με έρευνες οι σπουδαστές εκτίμησαν ότι μελετώντας μόνοι είναι πιο χρήσιμο από τις διαλέξεις. Η εκμάθηση με εξάσκηση θεωρήθηκε πιο αποτελεσματική επίσης, επειδή ασκήσεις θεωρήθηκαν πιο χρήσιμες από τις διαλέξεις και η πρακτική στα εργαστήρια εκτιμήθηκε ακόμα περισσότερο. Ομοίως, ο πρακτικός προγραμματισμός από τους ίδιους εκτιμήθηκε περισσότερο από ότι η προσωπική θεωρητική μελέτη. Τα παραδείγματα κώδικα προγράμματος θεωρήθηκαν ως πιο χρήσιμα από το υλικό και από τους σπουδαστές και από τους καθηγητές. Οι καθηγητές εκτίμησαν τις διαδραστικές απεικονίσεις περισσότερο από το υπόλοιπο υλικό του μαθήματος. Οι καθηγητές θεώρησαν ως αποτελεσματικότερο τρόπο εκμάθησης την πρακτική στα εργαστήρια, τις αναθέσεις εργασιών σε μικρές ομάδες και την μάθηση μέσω της καθορισμένης δομής του μαθήματος.

Ένα εισαγωγικό μάθημα προγραμματισμού πρέπει να είναι ρεαλιστικό ως προς τους στόχους που θα επιτευχθούν. Από παιδαγωγική σκοπιά ο καθηγητής πρέπει να δώσει βάση στα απλά και αρχικά πράγματα της γλώσσας και να επεκτείνει σε πιο πολύπλοκα μόνο όταν οι σπουδαστές έχουν αποκτήσει την κατάλληλη εμπειρία. Μια προτεινόμενη υπόδειξη είναι να μην δίνετε βάση μόνο στη διδασκαλία των χαρακτηριστικών και των δομών της γλώσσας αλλά και πως συνδυάζονται και χρησιμοποιούνται στο επίπεδο του σχεδιασμού ενός προγράμματος.

Προκειμένου να αντιμετωπιστούν οι αδυναμίες της κλασικής διδακτικής προσέγγισης, έχουν προταθεί εναλλακτικές διδακτικές προσεγγίσεις που παρουσιάζουν ένα ή περισσότερα από τα ακόλουθα χαρακτηριστικά[12]:

- *Έμφαση στη διαδικασία επίλυσης προβλημάτων*: οι μαθητές εξοικειώνονται με τα βασικά στάδια της διαδικασίας επίλυσης προβλημάτων τόσο σε μαθήματα του προγραμματισμού όσο και σε μαθήματα όπως η Τεχνολογία Λογισμικού, τα Πληροφοριακά Συστήματα, κ.λπ.
- *Έμφαση στη σχεδίαση προγραμμάτων*: οι μαθητές αποκτούν ικανότητες στη σχεδίαση προγραμμάτων και στην αξιολόγηση εναλλακτικών λύσεων στο πλαίσιο ενός προβλήματος
- *Χρησιμοποίηση εναλλακτικών μορφών αναπαράστασης της λύσης*: οι μαθητές χρησιμοποιούν ψευδοκώδικα ή άλλες μορφές για την αναπαράσταση της λύσης.
- *Χρησιμοποίηση παραδειγμάτων*: οι μαθητές κατανοούν χαρακτηριστικά των προγραμματιστικών δομών μέσα από σύντομα παραδείγματα.
- *Πειραματισμός και διερεύνηση*: οι μαθητές διερευνούν και πειραματίζονται με τις προγραμματιστικές δομές εκτελώντας προγράμματα στον υπολογιστή.
- *Συνεργατική μάθηση*: οι μαθητές συνεργάζονται στη σχεδίαση και υλοποίηση της λύσης προβλημάτων.
- *Αξιοποίηση εκπαιδευτικών εργαλείων*: αναπτύσσονται και αξιοποιούνται στη διδακτική διαδικασία εκπαιδευτικά εργαλεία που στοχεύουν κυρίως στην ελαχιστοποίηση των

συντακτικών λεπτομερειών, στον περιορισμό του υποστηριζόμενου ρεπερτορίου εντολών με απλή σύνταξη και σημασιολογία, στη στήριξη της διαδικασίας αποσφαλμάτωσης προγραμμάτων και στην κατανόηση των λαθών καθώς και στην οπτική/ηχητική προσομοίωση εκτέλεσης των προγραμμάτων.

Πολύ σημαντικό κομμάτι στην οργάνωση ενός εισαγωγικού μαθήματος προγραμματισμού είναι το *εργαστήριο*. Οι πληροφορίες που παρέχονται από τους compilers και τα άλλα εργαλεία που είναι διαθέσιμα είναι άμεσες και ιδιαίτερα λεπτομερείς. Η ενθάρρυνση που περνούν οι σπουδαστές από την δημιουργία ενός προγράμματος που λειτουργεί - όσο μικρό και αν είναι-επιδρά πολύ θετικά στην ψυχολογία τους. Στο εργαστήριο μπορούν να δουλέψουν, να πειραματιστούν και να μάθουν μόνοι τους. Αυτή η εμπειρία μπορεί να αποτελέσει εφελτήριο για την ανάπτυξη τεχνικών επίλυσης προβλημάτων. Δουλεύοντας πάνω σε μικρές και εύκολες εργασίες ειδικά σε προγράμματα που παρέχουν γραφική αναπαράσταση ως έξοδο τους αυξάνει το κίνητρο και τους δίνει ερεθίσματα. Εντούτοις πρέπει να δίνετε βάση στις προγραμματιστές αρχές που χρησιμοποιήθηκαν για να παραχθεί το αποτέλεσμα που βλέπουν.

Μια άλλη προσέγγιση για τη διδασκαλία του προγραμματισμού *η χρήση αυθεντικών παραδειγμάτων*. Η επιλογή των παραδειγμάτων-προβλημάτων πραγματοποιείται με τέτοιο τρόπο ώστε να προκαλούν το ενδιαφέρον των μαθητών και να υποστηρίζουν τη μη ακολουθιακή παρουσίαση των προγραμματιστικών δομών. Η συγκεκριμένη μέθοδος επιτρέπει την άμεση και σταδιακή εισαγωγή σε έννοιες του προγραμματισμού που κρίνονται δύσκολες π.χ. συναρτήσεις και διαδικασίες, αντικείμενα, κ.λπ. Απαιτεί όμως προσεκτικό σχεδιασμό των παραδειγμάτων και την παροχή έτοιμων προγραμμάτων/ διαδικασιών που να μπορούν να χρησιμοποιηθούν από τους μαθητές για μελέτη και επέκταση των λειτουργιών τους.

Η συγκεκριμένη προσέγγιση στοχεύει:

- στην αλλαγή του τρόπου διδασκαλίας του προγραμματισμού επικεντρώνοντας σε αυθεντικά παραδείγματα ως μέσο για την εισαγωγή των προγραμματιστικών εννοιών και δομών. Η παρουσίαση και ανάλυση των προγραμματιστικών εννοιών και δομών «υποκινείται» και πραγματοποιείται στο πλαίσιο παραδειγμάτων και δεν ακολουθεί την παραδοσιακή ακολουθιακή εισαγωγή των δομών που υποστηρίζει μια γλώσσα προγραμματισμού.
- στην ανάδειξη των δυνατοτήτων του υπολογιστή και της χρησιμότητας του προγραμματισμού. Τα παραδείγματα-προβλήματα που χρησιμοποιούνται δεν αφορούν κυρίως σε αριθμητικά προβλήματα αλλά προσεγγίζουν ενδιαφέροντα των μαθητών και καθημερινά αυθεντικά προβλήματα διαθεματικού χαρακτήρα, όπως για παράδειγμα υλοποίηση βάσεων δεδομένων σε πραγματικές εφαρμογές, προσομοιώσεις, κ.λπ. Με αυτό τον τρόπο, οι μαθητές αντιλαμβάνονται τη σημασία του προγραμματισμού και τις δυνατότητες αξιοποίησης του υπολογιστή, συνδέοντας την πληροφορική με άλλες επιστήμες.
- στη σταδιακή απόκτηση δεξιοτήτων στη σχεδίαση και ανάπτυξη προγραμμάτων. Οι μαθητές μέσω της μελέτης προγραμμάτων τα οποία χαρακτηρίζονται από καλή σχεδίαση, βέλτιστη χρήση των δομών της γλώσσας, κ.λπ., μαθαίνουν βασικούς κανόνες που τους βοηθούν στην ανάπτυξη προγραμμάτων.

Μια άλλη μέθοδος είναι η χρήση μελετών περίπτωσης. Οι μελέτες περίπτωσης μπορούν να χρησιμοποιηθούν είτε στο πλαίσιο της διδασκαλίας είτε στο πλαίσιο εργασιών που ανατίθενται στους μαθητές. Τα βασικά συστατικά μιας μελέτης περίπτωσης (case study) είναι [13]:

- η διατύπωση ενός προβλήματος
- μία περιγραφή της διαδικασίας για την επίλυση του προβλήματος που ακολουθείται από έναν έμπειρο προγραμματιστή
- το πρόγραμμα (κώδικας) που επιλύει το πρόβλημα
- ερωτήσεις που αξιολογούν κατά πόσο ο μαθητής έχει κατανοήσει τη λύση
- ερωτήσεις που στοχεύουν στην εξάσκηση του μαθητή στην ανάλυση προβλημάτων και στη σχεδίαση και υλοποίηση της λύσης τους.

Μέσα από τις μελέτες περίπτωσης, οι μαθητές καλούνται να σχολιάσουν τη λύση που προτείνεται, να διερευνήσουν αν η λύση μπορεί να εφαρμοστεί σε παρόμοια προβλήματα, να επεκτείνουν ή να τροποποιήσουν τη λύση στο πλαίσιο άλλων προβλημάτων, να κάνουν προβλέψεις για τα αποτελέσματα της προτεινόμενης λύσης, να διορθώσουν τυχόν λάθη ή ακόμη να κατασκευάσουν μελέτες περίπτωσης για συγκεκριμένα προβλήματα [13]. Η προσέγγιση αυτή δίνει ιδιαίτερη έμφαση στο να αποκτήσουν οι μαθητές δεξιότητες στη σχεδίαση της λύσης, στην υλοποίηση της λύσης βήμα-βήμα και στη διερεύνηση/αξιολόγηση εναλλακτικών λύσεων σε ένα πρόβλημα. Η πειραματική αξιολόγηση των μελετών περίπτωσης, έδειξε ότι ο σχολιασμός της λύσης από ειδικούς βοηθάει σημαντικά τους μαθητές στην αποτελεσματική επίλυση προβλημάτων.

Άλλη προσέγγιση είναι αυτή της χρήσης προτύπων. Η προσέγγιση αυτή αξιοποιεί τα «πρότυπα» σχεδίασης και προγραμματισμού (design and programming patterns) και στοχεύει στο να αποκτήσουν οι μαθητές δεξιότητες στη σχεδίαση προγραμμάτων, στην αξιολόγηση εναλλακτικών λύσεων και στη χρησιμοποίηση ή/και ενσωμάτωση υπαρχόντων λύσεων στα πλαίσια διαφόρων προβλημάτων. Η ιδέα των «προτύπων» χρησιμοποιείται σε διάφορους τομείς πολλά χρόνια. Στην περίπτωση του προγραμματισμού, τα βασικά συστατικά ενός «προτύπου» είναι [14].

- ένα όνομα
- το πλαίσιο του προβλήματος στο οποίο το «πρότυπο» βρίσκει εφαρμογή
- η λύση, που περιγράφει τα συστατικά στοιχεία του προβλήματος, τη λειτουργία τους και τη μεταξύ τους σχέση - η λύση εκφράζεται σε αφηρημένο επίπεδο, αν και είναι χρήσιμο να δίνεται συγκεκριμένο παράδειγμα σε κάποια γλώσσα προγραμματισμού
- οι συνέπειες της χρησιμοποίησης του συγκεκριμένου «προτύπου», δηλαδή πλεονεκτήματα και μειονεκτήματα που προκύπτουν από την εφαρμογή της συγκεκριμένης λύσης στο πλαίσιο του προβλήματος που έχει τεθεί.

Η προσέγγιση των «προτύπων» χρησιμοποιείται κυρίως στη διδασκαλία του αντικειμενοστραφούς προγραμματισμού. Υπάρχουν όμως πρότυπα διαθέσιμα για βασικές έννοιες και προγραμματιστικές δομές όπως η ανάγνωση δεδομένων, οι δομές επιλογής και επανάληψης.

Η πειραματική αξιολόγηση των «προτύπων» έδειξε ότι οι μαθητές βοηθούνται στην κατανόηση των προγραμματιστικών δομών και επιτυγχάνουν καλύτερα αποτελέσματα στην ανάλυση ενός προβλήματος, στη σχεδίαση και στην υλοποίηση της λύσης του. Η αποτελεσματικότητά τους εξαρτάται από παράγοντες όπως (i) η ποικιλία και ο αριθμός

των παραδειγμάτων που χρησιμοποιούνται, (ii) αν συνδέονται με ενδεικτικά προβλήματα και με ποιο τρόπο μπορούν να αξιοποιηθούν στο πλαίσιο διαφόρων προβλημάτων, (iii) αν εξετάζονται περιπτώσεις προβλημάτων για τα οποία δεν κρίνονται κατάλληλα.

Η επίλυση προβλημάτων έχει επίσης χαρακτηριστεί ως μια μέθοδος για τη διδασκαλία του προγραμματισμού. Σύμφωνα με αυτή ακολουθείται η προσέγγιση υπολογισμός μέσω της αλληλεπίδρασης. Τόσο οι έμπειροι όσο και οι αρχάριοι προγραμματιστές χρησιμοποιούν για την επίλυση προβλημάτων σχετικά παραδείγματα που είτε τα έχουν διδαχθεί/διαβάσει είτε τα έχουν αναπτύξει οι ίδιοι. Τα παραδείγματα, συνοδευόμενα από τις επεξηγήσεις τους, χρησιμοποιούνται ιδιαίτερα από τους διδάσκοντες στη διδασκαλία των μαθημάτων προγραμματισμού[15].

Μία διδακτική προσέγγιση που βασίζεται σε παραδείγματα αφορά στη χρήση «ελάχιστων» παραδειγμάτων (minimal examples). Καθένα από τα παραδείγματα επικεντρώνεται σε μία συγκεκριμένη έννοια του προγραμματισμού ή προγραμματιστική δομή και είναι μικρό σε έκταση. Τα βασικά συστατικά των «ελάχιστων» παραδειγμάτων είναι (Hoffman & Walsh, 1997)

- μία ερώτηση/περιγραφή που χαρακτηρίζει το παράδειγμα
- ο κώδικας του παραδείγματος που περιέχει τις σχετικές εντολές
- το αποτέλεσμα της εκτέλεσης του κώδικα για κάποια είσοδο τιμών
- σχόλια, επεξηγήσεις και επισημάνσεις σχετικά με την έννοια που παρουσιάζεται και τη λειτουργία των εντολών.

Το πλεονέκτημα των «ελάχιστων» παραδειγμάτων, σε σχέση με τα τυπικά παραδείγματα που υπάρχουν στα σχετικά συγγράμματα και ενδεχομένως χρησιμοποιούν οι διδάσκοντες στη διδασκαλία τους, είναι ότι δίνουν τη δυνατότητα στο μαθητή να επικεντρωθεί σε μία συγκεκριμένη έννοια, να εφαρμόσει γρήγορα και εύκολα το παράδειγμα και να πειραματιστεί με αυτό.

Τα χαρακτηριστικά στοιχεία της γλώσσας δεν διδάσκονται θεωρητικά αλλά ανακαλύπτονται στο πλαίσιο λύσεων προβλημάτων. Έτσι οι σπουδαστές μαθαίνουν να αξιολογούν μόνοι τους τις δυνατότητες τους και αποκτούν περισσότερη αυτοπεποίθηση από ότι στην κλασική μορφή διδασκαλίας. Εντούτοις πρέπει να αναφέρουμε ότι η επίλυση προβλημάτων είναι απαραίτητη συνθήκη αλλά όχι ικανή από μόνη της για τη διδασκαλία του προγραμματισμού καθώς το μεγαλύτερο πρόβλημα που αντιμετωπίζουν οι αρχάριοι προγραμματιστές είναι πως να γράψουν το πρόγραμμα που θα επιλύει ένα συγκεκριμένο πρόβλημα. Άρα η διδασκαλία των βασικών χαρακτηριστικών και δομών της γλώσσας και πως αυτά συνδυάζονται πρέπει να μείνει ως κύριο μέλημα του διδάσκοντα.

Τέλος εκτός από τις προσεγγίσεις που αναφέρθηκαν πάνω στη διδασκαλία του προγραμματισμού, πρέπει να δοθεί προσοχή από τον διδάσκοντα πάνω στα προτερήματα και τα μειονεκτήματα του κάθε σπουδαστή ξεχωριστά. Παραδείγματος χάριν, είναι χρήσιμο να προσδιοριστεί το γεγονός ότι ένας δεδομένος σπουδαστής έχει μια καλή γνώση των σχεδιαστικών αρχών αλλά δεν μπορεί να τις εφαρμόσει ή ίσως να μπορεί να είναι καλός στο σχεδιασμό αλλά όχι στην δοκιμή και αποσφαλμάτωση του κώδικα. Για αυτό οποιοδήποτε διαγνωστικό εργαλείο θα χρησιμοποιηθεί σε μια πραγματική εργαστηριακή κατάσταση θα πρέπει να είναι πλούσιο αρκετά ώστε να είναι χρήσιμο, αλλά αρκετά απλό ώστε να είναι εύχρηστο.

Τέλος πολύ σημαντική είναι η προσέγγιση βασισμένη στη συνεργασία. Η ανάθεση εργασιών σε ομάδες που πραγματοποιούνται στο πλαίσιο των μαθημάτων αποσκοπεί,

μεταξύ άλλων, στη δημιουργία κινήτρου στους μαθητές για ενασχόληση, στην απόκτηση ικανοτήτων από τους μαθητές για ανταλλαγή απόψεων και για έλεγχο παρακολούθησης της εργασίας, στην καλλιέργεια θετικής στάσης για συνεργασία με άλλα άτομα και για αμοιβαία συνεισφορά και ευθύνη στην επίτευξη ενός στόχου. Συχνά όμως οι μαθητές, κατά την εκπόνηση ομαδικών εργασιών, επικεντρώνονται στο δικό τους τμήμα εργασίας, δεν έχουν συνολική εικόνα και άποψη, εργάζονται μεμονωμένα και η συνεισφορά τους μπορεί να είναι άνιση [16].

Η προσέγγιση ανάπτυξης προγραμμάτων από ομάδες των δύο ατόμων (pair-programming) προτείνεται και εφαρμόζεται τόσο στη διαδικασία της διδασκαλίας όσο και στην εκπόνηση εργασιών. Σύμφωνα με αυτή την προσέγγιση, δύο άτομα συνεργάζονται στη σχεδίαση και ανάπτυξη ενός προγράμματος. Το ένα μέλος της ομάδας, παίζει το ρόλο του «οδηγού» (driver) και έχει τον έλεγχο του μολυβιού /ποντικιού /πληκτρολογίου στη σχεδίαση και ανάπτυξη του προγράμματος, ενώ το δεύτερο μέλος είναι ο «παρατηρητής» (observer) που διαρκώς ελέγχει το έργο του «οδηγού» θέτοντας ερωτήσεις, διερευνώντας εναλλακτικές λύσεις, παρατηρώντας ελλείψεις, κ.λπ. Οι ρόλοι του «οδηγού» και του «παρατηρητή» εναλλάσσονται μεταξύ των δύο ατόμων. Και τα δύο μέλη συμμετέχουν ενεργά στη διαδικασία και είναι εξίσου υπεύθυνα για την επίτευξη του στόχου.

Η εφαρμογή της μεθόδου, απαιτεί από την πλευρά του διδάσκοντα κατά τη διάρκεια των εργαστηρίων, να εξασφαλίσει ότι τηρούνται οι ρόλοι του «οδηγού» και του «παρατηρητή» και ότι υπάρχει ουσιαστική συνεισφορά και από τα δύο μέλη της ομάδας.

Η προσέγγιση ανάπτυξης προγραμμάτων σε ομάδες των δύο ατόμων, ακολουθείται χρόνια σε επαγγελματικό επίπεδο, τα τελευταία χρόνια όμως έχει προκαλέσει το ενδιαφέρον των διδασκόντων για την εφαρμογή της στη διδασκαλία των μαθημάτων προγραμματισμού. Η πειραματική εφαρμογή και αξιολόγηση της μεθόδου αναδεικνύει σημαντικά πλεονεκτήματα, όπως [16] :

- την ικανοποίηση των μαθητών: οι μαθητές αισθάνονται μεγαλύτερη αυτοπεποίθηση για το αποτέλεσμα της εργασίας επειδή έχουν κάποιον να τους βοηθήσει και να τους υποδείξει ελλείψεις, αισθάνονται λιγότερη απογοήτευση και μεγαλύτερη ικανοποίηση επειδή η συνεργασία οδηγεί σε πιο αποτελεσματικές λύσεις αφιερώνοντας λιγότερο χρόνο.
- την ανάπτυξη δεξιοτήτων στην επίλυση προβλημάτων: οι μαθητές ανταλλάσσουν ιδέες για την επίλυση ενός προβλήματος (είτε στη φάση της σχεδίασης είτε της υλοποίησης) και συζητώντας για τα πλεονεκτήματα/μειονεκτήματα της προτεινόμενης λύσης, μαθαίνουν να διερευνούν εναλλακτικές λύσεις και να επιλύουν προβλήματα που ενδεχομένως από μόνοι τους δε θα μπορούσαν.
- την ενίσχυση του μαθησιακού αποτελέσματος: οι μαθητές, επειδή αναγκάζονται να στοχάζονται (reflection), να εξηγούν τις ενέργειες τους (self-explanations) και να αναπτύσσουν μηχανισμούς παρακολούθησης/ελέγχου (monitoring) της διαδικασίας και της προόδου της εργασίας, μαθαίνουν αποτελεσματικότερα και αναπτύσσουν δεξιότητες αυτοαξιολόγησης και παρακολούθησης της προόδου τους.
- την ανάπτυξη ικανοτήτων συνεργασίας: οι μαθητές μαθαίνουν να επικοινωνούν και να συνεργάζονται αποτελεσματικά με άλλα άτομα αναπτύσσοντας την κοινωνικότητά τους.

Ενώ υπάρχει εκτενής αναφορά για τις διαφορές ανάμεσα σε έμπειρους και αρχάριους προγραμματιστές, υπάρχει μια διάκριση που δεν έχει λάβει μεγάλη προσοχή. Είναι η διάκριση μεταξύ των αποτελεσματικών και αναποτελεσματικών αρχαρίων. Οι αποτελεσματικοί αρχάριοι είναι εκείνοι που μαθαίνουν, χωρίς υπερβολική προσπάθεια ή βοήθεια, να προγραμματίζουν. Ακόμα και όταν συναντούν προβλήματα δε σταματούν την προσπάθεια αλλά συνεχίζουν να προσπαθούν, να πειραματίζονται και να αλλάζουν τον κώδικα τους. Έτσι χρησιμοποιούν τη γνώση από προηγούμενα λάθη και έχουν την προοπτική να λύσουν το πρόβλημα και να προχωρήσουν. Αντίθετα, οι αναποτελεσματικοί είναι εκείνοι που δεν μαθαίνουν, ή το κάνουν μόνο μετά από υπέρμετρη προσπάθεια και προσωπική προσοχή. Όταν αντιμετωπίσουν ένα πρόβλημα έχουν την τάση να σταματούν. Όταν δεν ξέρουν πως να προχωρήσουν απογοητεύονται και εγκαταλείπουν κάθε προσπάθεια επίλυσης.

Μπορεί να είναι παραγωγικό, σε μια εισαγωγική σειρά μαθημάτων προγραμματισμού, να εστιάσει ο διδάσκοντας ρητά στην προσπάθεια να δημιουργηθούν και να ενθαρρυνθούν οι αποτελεσματικοί αρχάριοι. Με άλλα λόγια, παρά να εστιάσει αποκλειστικά στο δύσκολο τελικό προϊόν της γνώσης του προγραμματισμού, μπορεί να είναι χρήσιμο να στραφεί τουλάχιστον εν μέρει στην προσπάθεια να κάνει τους σπουδαστές να λειτουργούν ως αποτελεσματικοί αρχάριοι. Η βασικότερη διαφορά ανάμεσα σε αυτούς τους τύπους αρχαρίων σχετίζονται με τις στρατηγικές και όχι με αυτή καθαυτή τη γνώση του αντικειμένου. Η θεωρητική γνώση της γλώσσας είναι διαθέσιμη από πολλές πηγές, και οι σειρές μαθημάτων και τα εγχειρίδια είναι χαρακτηριστικά σχεδιασμένα για να εισαγάγουν αυτήν την γνώση με έναν δομημένο τρόπο. Οι στρατηγικές για τη χρησιμοποίηση αυτής στην κατανόηση και την παραγωγή ενός προγράμματος, εντούτοις, είναι κρίσιμη για την έκβαση εκμάθησης, αλλά λαμβάνει πολύ λιγότερη προσοχή.

Συμπερασματικά το μεγαλύτερο πρόβλημα των αρχαρίων προγραμματιστών δεν να είναι η κατανόηση των βασικών εννοιών αλλά μάλλον η εκμάθηση το πως να εφαρμόζονται[17]. Προτείνετε ότι οι καθηγητές πρέπει να εστιάσουν περισσότερο στο συνδυασμό και τη χρήση αυτών των χαρακτηριστικών γνωρισμάτων. Ακόμα κι αν η θεωρία είναι πολύ σημαντική στην εκμάθηση του προγραμματισμού, οι σπουδαστές χρειάζονται επίσης την πρακτική εμπειρία για να καταλάβουν τις έννοιες. Όσο περισσότερο πρακτική χρησιμοποιείται και τα παραδείγματα είναι συγκεκριμένα τόσο πιο πολύ ευνοείται η μάθηση. Η εκμάθηση μέσω πρακτικής και επίδειξης πρέπει να είναι συνεχώς παρούσα.

2.7 Μαθησιακές δυσκολίες και παρανοήσεις σε βασικές προγραμματιστικές έννοιες και δομές

2.7.1 Μαθησιακές δυσκολίες – Εισαγωγή

Οι αρχάριοι προγραμματιστές αντιμετωπίζουν δυσκολίες στην ανάπτυξη προγραμμάτων. Δεν μπορούν να καταλάβουν τι ακριβώς πρέπει να κάνουν για λύσουν ένα ορισμένο πρόβλημα που τους έχει ανατεθεί και δυσκολεύονται να εξηγήσουν γιατί τα αποτελέσματα της εξόδου ενός προγράμματος δεν συμπίπτουν με αυτά που αναμένουν.

Οι μαθησιακές δυσκολίες μπορούν να ταξινομηθούν πιο συστηματικά στις ακόλουθες κατηγορίες [18]:

- Τι είναι προγραμματισμός (Orientation): οι μαθητές δεν κατανοούν τι είδη προβλημάτων μπορούν να επιλυθούν στον υπολογιστή και τι οφέλη αποκομίζουν από τις δεξιότητες που αποκτούν στην επίλυση προβλημάτων στον υπολογιστή.
- Λειτουργία του υπολογιστή: οι μαθητές δεν μπορούν να καταλάβουν τον τρόπο με τον οποίο λειτουργεί ο υπολογιστής. Για παράδειγμα δυσκολεύονται να κατανοήσουν τον τρόπο με τον οποίο χειρίζεται ο υπολογιστής τις μεταβλητές, με αποτέλεσμα να αντιμετωπίζουν προβλήματα σε ζητήματα όπως η αρχικοποίηση, η ανάθεση μιας τιμής καθώς και στην είσοδο-έξοδο του προγράμματος.
- Συντακτικό και σημασιολογία της γλώσσας προγραμματισμού: οι μαθητές αντιμετωπίζουν προβλήματα στο συντακτικό της γλώσσας που χρησιμοποιούν αλλά το μεγαλύτερο πρόβλημα είναι η κατανόηση της σημασιολογίας της γλώσσας, δηλαδή η λειτουργία των εντολών και το πως αυτές εκτελούνται από τον υπολογιστή.
- Σχέδια/πλάνα επίλυσης: όταν δίδεται στους μαθητές ένα προς επίλυση πρόβλημα, οι ενέργειες που ακολουθούν δεν εντάσσονται στα πλαίσια μιας ευρύτερης στρατηγικής και ενός σχεδίου/πλάνου για την επίτευξη του στόχου.
- Δοκιμή/αποσφαλμάτωση προγραμμάτων: οι μαθητές δεν δοκιμάζουν συχνά τον κώδικα του προγράμματος που γράφουν ως προς την ορθότητα του και δυσκολεύονται να εντοπίσουν τα λάθη στον κώδικα τους καθώς και να τα διορθώσουν.

Σε μια έρευνα[17] που έγινε πάνω στις δυσκολίες που αντιμετωπίζουν οι αρχάριοι προγραμματιστές, η αναδρομή, οι δείκτες και οι αναφορές, οι αφηρημένοι τύποι στοιχείων και η χρησιμοποίηση των γλωσσικών βιβλιοθηκών θεωρήθηκαν ως οι δυσκολότερες έννοιες να κατανοήσουν. Πρόκειται για αφηρημένες έννοιες και έτσι γνωστικά σύνθετες για να καταλάβουν χωρίς έναν παρόμοιο φαινόμενο στη καθημερινή ζωή για τη σύγκριση. Από τα αποτελέσματα της έρευνας προκύπτει πως ζητήματα σχετικά με την κατανόηση των δομών, της σύνταξης της γλώσσας, την κατανόηση πως να σχεδιαστεί ένα πρόγραμμα για να λυθεί ένας ορισμένος στόχος, έχουν ένα ισχυρό συσχετισμό μεταξύ τους. Ο μαθητής είτε τα μαθαίνει όλα εύκολα είτε έχει προβλήματα με όλα. Τα βασικά προβλήματα στον προγραμματισμό συσχετίζονται επίσης με τις υπόλοιπες έννοιες, αλλά όχι τόσο έντονα. Αυτές οι έννοιες περιλαμβάνουν τον χειρισμό εισόδου-εξόδου του προγράμματος, το χειρισμό λαθών και τη χρησιμοποίηση των γλωσσικών βιβλιοθηκών δηλαδή ζητήματα που δεν ανήκουν στα κύρια θέματα σε μια σειρά μαθημάτων προγραμματισμού.

Επίσης υπήρξαν μερικές σημαντικές στατιστικές διαφορές μεταξύ των γλωσσών στα περιεχόμενα μιας σειράς μαθημάτων προγραμματισμού. Η γλώσσα διδασκαλίας δεν φάνηκε να έχει επιπτώσεις στην εκμάθηση. Ωστόσο η C++ βρέθηκε να είναι δυσκολότερη από την Java. Οι πίνακες, οι δείκτες και οι αναφορές και οι παράμετροι ήταν δυσκολότερο αντιληπτοί κατά την εκμάθηση C++ απ' ότι στην Java. Η κατανόηση των δομών προγραμματισμού ήταν δυσκολότερη σε άλλες γλώσσες απ' ότι στην C++, την Java ή PASCAL. Η χρησιμοποίηση των γλωσσικών βιβλιοθηκών ήταν ευκολότερη στην Java απ' ότι σε PASCAL.

2.7.2 Μεταβλητές

Το σημαντικότερο μαθησιακό πρόβλημα που σχετίζεται με την έννοια της μεταβλητής, προέρχεται από το γεγονός ότι η έννοια της μεταβλητής στον προγραμματισμό δεν ταυτίζεται με αυτή των μαθηματικών. Έτσι οι μαθητές μεταφέρουν τις γνώσεις που έχουν

για την έννοια της μεταβλητής, στον προγραμματισμό με αποτέλεσμα να προκύπτουν διδακτικά και μαθησιακά προβλήματα.

Οι μαθησιακές δυσκολίες και οι παρανοήσεις που παρατηρήθηκαν και αφορούν στην έννοια της μεταβλητής σχετίζονται με[19]:

- Την αρχικοποίηση της μεταβλητής: οι μαθητές δεν κατανοούν το λόγο αρχικοποίησης της τιμής μιας μεταβλητής και θεωρούν ότι αφού δεν έχει δοθεί μια τιμή σε αυτή, δεν χρειάζεται να αρχικοποιηθεί αφού εκ των προτέρων θεωρούν ότι είναι ίση με μηδέν.
- Τον τύπο των μεταβλητών: οι μαθητές αντιμετωπίζουν δυσκολίες στον προσδιορισμό και καθορισμό του τύπου των μεταβλητών. Δε μπορούν να προσδιορίσουν πότε μία μεταβλητή πρέπει να δηλωθεί ως τύπου ακεραίου ή πραγματικού. Ιδιαίτερη δυσκολία παρουσιάζει η διαχείριση λογικών μεταβλητών.
- Την εντολή ανάθεσης τιμής: οι μαθητές συγχέουν την εντολή ανάθεσης τιμής σε μια μεταβλητή με τη μαθηματική ισότητα. Έτσι χρησιμοποιώντας τις πρότερες γνώσεις που έχουν αποκτήσει στα μαθηματικά, δεν μπορούν να κατανοήσουν πως είναι δυνατόν να είναι σωστή η εντολή εκχώρησης $x=15$, ενώ παρότι είναι μαθηματικά εύλογο, δε μπορεί να γραφεί $15=x$. Επίσης δεν κατανοούν ότι η εντολή ανάθεσης καταχωρεί δεδομένα στη θέση της προϋπάρχουσας τιμής, η οποία χάνεται. Έτσι θεωρούν ότι η μεταβλητή διατηρεί περισσότερες από μία τιμές και μάλιστα ότι έχει τη δυνατότητα να «θυμάται» την ιστορία των αναθέσεων.
- Την αντιμετάθεση τιμών μεταβλητών: οι μαθητές θεωρούν ότι με την εντολή ανάθεσης $a=b$ ισχύει και $b=a$ και με αυτό τον τρόπο μπορούν να αντιμεταθέσουν τις τιμές δύο μεταβλητών. Κάτι τέτοιο όμως δεν ισχύει και έτσι δυσκολεύονται να κατανοήσουν πως χρειάζεται η χρήση μιας βοηθητικής μεταβλητής για να γίνει η αντιμετάθεση.

2.7.3 Οι δομές ελέγχου και η έννοια της επιλογής

Η δομή ελέγχου χρησιμοποιείται για τη λήψη απόφασης μεταξύ δύο διαφορετικών καταστάσεων εκ των οποίων η μία είναι αληθής και η άλλη ψευδής. Η δομή αυτή επιτρέπει σε ένα αλγόριθμο να αλλάξει την εκτέλεση του ανάλογα με τις περιστάσεις, δηλαδή ο αλγόριθμος επιλέγει ή ελέγχει τις εντολές που πρόκειται να εκτελεστούν και ανάλογα με τα αποτελέσματα των ελέγχων επιλέγονται οι εντολές που θα ακολουθήσουν. Η διατύπωση των ελέγχων γίνεται με τη χρήση λογικών προτάσεων που ονομάζονται συνθήκες.

Η δομή ελέγχου είναι μια δομή που οικοδομείται με ιδιαίτερη δυσκολία από τους μαθητές που μαθαίνουν προγραμματισμό και από τους αρχάριους προγραμματιστές [9]. Στη μάθηση της παρεμβαίνουν διάφοροι παράγοντες, οι οποίοι και πρέπει να λαμβάνονται υπόψη στο πλαίσιο της διδασκαλίας του προγραμματισμού. Οι παράγοντες αυτοί έχουν σχέση τόσο με τις γενικότερες δυσκολίες που έχουν οι μαθητές στην οικοδόμηση των διαφόρων εννοιών του προγραμματισμού, όσο και με τις εγγενείς ιδιαιτερότητες που χαρακτηρίζουν τη συγκεκριμένη δομή.

Οι μαθητές αντιμετωπίζουν δυσκολίες στον προσδιορισμό της τιμής (true/false) της λογικής έκφρασης καθώς και στον καθορισμό της απαιτούμενης λογικής έκφρασης στο πλαίσιο ενός προβλήματος. Συγκεκριμένα, οι μαθητές δεν είναι σε θέση να εφαρμόζουν επιτυχώς τους λογικούς τελεστές σε λογικές εκφράσεις και να εφαρμόζουν τους κανόνες προτεραιότητας που ισχύουν σε αυτούς. Οι σημαντικότερες μαθησιακές δυσκολίες στις

δομές επιλογής σχετίζονται με την κατανόηση της λειτουργίας τους. Για παράδειγμα μαθητές θεωρούν ότι σε μια συνθήκη ελέγχου τύπου «if-then-else» ακόμα και αν δεν αληθεύει η συνθήκη ελέγχου στο «if», το σώμα των εντολών που ακολουθεί το «then» θα εκτελεστεί ούτως ή άλλως, ενώ άλλοι μαθητές θεωρούν ότι σε περίπτωση που δεν αληθεύει η συνθήκη ελέγχου στο «if», το πρόγραμμα θα σταματήσει εμφανίζοντας κάποιο μήνυμα λάθους, έστω και αν δεν περιλαμβάνεται σχετικό μήνυμα στο σώμα εντολών του «else» (ή ακόμη και αν δεν υπάρχει «else») [20]. Εκτός από τις μαθησιακές δυσκολίες που εμφανίζονται στο λογικό και σημασιολογικό περιεχόμενο της συνθήκης, οι μαθητές αντιμετωπίζουν προβλήματα και στη συντακτική αναπαράσταση με αποτέλεσμα να καταλήγουν σε μη ορθά προγράμματα. Τέλος οι μαθητές δεν μπορούν να διακρίνουν τη λειτουργία των εμφωλευμένων δομών επιλογής(συνδυασμός της απλής(«if-then-end») και της πλήρους δομής επιλογής(«if-then-else-end»)) και της πολλαπλής επιλογής(«if-then-else if-else-end»)[21]. Θεωρούν ότι οι δύο περιπτώσεις λειτουργούν ισοδύναμα και δεν μπορούν να εφαρμόσουν την πιο κατάλληλη ανάλογα με το προς επίλυση πρόβλημα.

Οι δυσκολίες αυτές οφείλονται στον τρόπο με τον οποίο οι μαθητές εκφράζουν τη «λογική» στην καθημερινή τους ζωή. Συχνά, χρησιμοποιούν γενικές περιπτώσεις ακολουθούμενες από εξαιρέσεις – δεν αναφέρουν ξεχωριστά μία – μία τις περιπτώσεις όπως απαιτείται κατά τη χρησιμοποίηση των προγραμματιστικών δομών επιλογής. Επίσης οι μαθητές δεν διαθέτουν εμπειρία στην εφαρμογή των βασικών κανόνων της λογικής καθώς η διδασκαλία σε έννοιες όπως οι λογικές εκφράσεις και μεταβλητές δεν κρίνεται επαρκής. Σημαντικό ρόλο στην πρόσκτηση της έννοιας της δομής ελέγχου διαδραματίζουν και οι πρότερες γνώσεις μαθηματικών και λογικής. Σε γενικό επίπεδο, οι έρευνες δείχνουν ότι οι μαθητές που έχουν ένα ανεπτυγμένο μαθηματικό υπόβαθρο, οικοδομούν πιο γρήγορα τις δομές αυτού του τύπου.

Παρακάτω παραθέτουμε ένα παράδειγμα για την επίλυση του προβλήματος σύγκρισης και εκτύπωσης του μεγαλύτερου από δύο πραγματικούς αριθμούς, με 3 διαφορετικές περιπτώσεις ώστε ο αλγόριθμος να είναι πιο κοντά στην φυσική γλώσσα[22]:

1 ^η περίπτωση	2 ^η περίπτωση	3 ^η περίπτωση
program megalyteros1;	program megalyteros2;	program megalyteros3;
var	var	var
X , Y, Meg: Real;	X , Y, Meg: Real;	X , Y, Meg: Real;
begin	begin	begin
Writeln('Dwse dyo arithmous');	Writeln('Dwse dyo arithmous');	Writeln('Dwse dyo arithmous');
ReadLn(X,Y);	ReadLn(X,Y);	ReadLn(X,Y);
Meg:=Y;	if X<=Y then Meg:=Y;	If X>Y then Meg:=X else Meg:=Y;
if X>Y then Meg:=X;	if X>Y then Meg:=X;	
Writeln('Megalyteros einai o:' Meg)	Writeln('Megalyteros einai o :' Meg)	Writeln('Megalyteros einai o :' Meg)
end.	end.	end.

Στην πρώτη περίπτωση γίνεται χρήση μιας εντολής εκχώρηση $Mem:=Y$ που κάνει την υπόθεση ότι ο μεγαλύτερος αριθμός είναι ο δεύτερος. Στην περίπτωση που η υπόθεση δεν αληθεύει, τότε εκτελείται η εσωτερική εντολή της IF...THEN η οποία διορθώνει την αρχική τιμή της μεταβλητής Mem.

Στη δεύτερη περίπτωση δεν γίνεται αυθαίρετη χρήση της εντολής $Mem:=Y$ αλλά χρησιμοποιείται ως εσωτερική της εντολής IF...THEN. Άρα αντίθετα με την πρώτη περίπτωση που απαιτείται μία σύγκριση και μιάμιση εκχώρηση εδώ θα γίνουν δύο συγκρίσεις και μία μόνο εντολή εκχώρησης.

Στην τελευταία περίπτωση ο αλγόριθμος υλοποιείτε με τη χρήση της εντολής IF...THEN...ELSE και εκτός από ότι είναι πιο κοντά στη φυσική γλώσσα, είναι και πιο οικονομικός αφού απαιτεί ακριβώς μία σύγκριση και μία εκχώρηση.

2.7.4 Δομές Επανάληψης

Η επαναληπτική διαδικασία συνιστά μια από τις βασικές προγραμματιστικές δομές η οποία επιτρέπει την επανάληψη ενεργειών για ορισμένο αριθμό φορών. Ο αριθμός των επαναλήψεων μπορεί να είναι εκ των προτέρων γνωστός ή να εξαρτάται από κάποιον παράγοντα(π.χ. μια τιμή εισόδου) κατά τη διάρκεια εκτέλεσης της επανάληψης.

Υπάρχουν τρεις μορφές της επαναληπτικής διαδικασίας που διδάσκονται οι μαθητές. Κάθε μία από αυτές έχει ιδιαίτερα χαρακτηριστικά και εξυπηρετεί διαφορετικό σκοπό, με αποτέλεσμα να μην εμφανίζονται τα ίδια διδακτικά προβλήματα. Οι δομές επανάληψης είναι οι εξής:

- «ΕΠΑΝΕΛΑΒΕ ... ΜΕΧΡΙΣ_ΟΤΟΥ συνθήκη» ή αλλιώς «Repeat...until»
- «ΟΣΟ συνθήκη ΕΠΑΝΕΛΑΒΕ...» ή αλλιώς «While...do»
- «ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ n ΜΕ_ΒΗΜΑ m ...ΕΠΑΝΑΛΑΒΕ» ή αλλιώς «For».

Οι πρώτες δύο μορφές αποτελούν την ακαθόριστη επανάληψη κατά την οποία, η επανάληψη ελέγχεται από μια συνθήκη, η οποία καθορίζει την έξοδο από την επαναληπτική διαδικασία. Στην πρώτη περίπτωση, αυτή η συνθήκη ελέγχεται μετά την εκτέλεση του σώματος της επανάληψης, ενώ στη δεύτερη η συνθήκη ελέγχεται εκ των προτέρων πριν εκτελεστεί το σώμα της επανάληψης. Η τρίτη μορφή επαναληπτικής χρησιμοποιείται όταν ο αριθμός των επαναλήψεων είναι εκ των προτέρων γνωστός και απαιτεί τη χρήση μιας μεταβλητής, η οποία παίζει το ρόλο του μετρητή για το πλήθος επαναλήψεων.

Το πιο σημαντικό διδακτικό πρόβλημα τίθεται από την ίδια τη διατύπωση του σώματος του βρόγχου. Ο μαθητής οφείλει να προσδιορίσει τόσο τις μεταβλητές όσο και τη σχέση ανάμεσα τους, η οποία πρέπει να διατηρείται σε όλη τη ζωή του βρόγχου.

Πιο συγκεκριμένα όσον αφορά στις δύο πρώτες επαναληπτικές δομές, οι μαθητές δυσκολεύονται να καθορίσουν τη συνθήκη ελέγχου της επαναληπτικής δομής στο πλαίσιο ενός συγκεκριμένου προβλήματος. Ιδιαίτερα, αν η συνθήκη ελέγχου απαιτεί συνδυασμό λογικών εκφράσεων ή τη χρήση λογικών μεταβλητών τότε κάνουν αρκετά λάθη. Άλλες μαθησιακές δυσκολίες σχετίζονται με την αρχικοποίηση και ανανέωση της τιμής των μεταβλητών ελέγχου. Ιδιαίτερη δυσκολία παρουσιάζει η αρχικοποίηση της τιμής των μεταβλητών όταν πρόκειται για μεταβλητές οι οποίες παίζουν το ρόλο του μετρητή ή αθροιστή και απαιτείται στην περίπτωση αυτή να χρησιμοποιηθεί εντολή ανάθεσης τιμής. Έτσι συχνά οδηγούνται σε ατέρμονους βρόγχους επειδή δεν ανανεώνεται η τιμή των μεταβλητών. Τέλος μεγαλύτερη δυσκολία αντιμετωπίζουν στην

επαναληπτική δομή «While...do» από την «Repeat...until». Αυτό οφείλεται στο ότι η συνθήκη ελέγχου βρίσκεται πριν το σώμα εντολών της επανάληψης κάτι που σημαίνει ότι οι εντολές της επανάληψης μπορεί να μην εκτελεστούν καμία φορά αν η αρχική τιμή της συνθήκης ελέγχου είναι ψευδής. Αυτό το γεγονός δε συνάδει με τον τρόπο σκέψης των μαθητών για την επαναληπτική διαδικασία.

Όσον αφορά στην επαναληπτική δομή «For» οι σημαντικότερες δυσκολίες σχετίζονται με τον καθορισμό του βήματος ανανέωσης της μεταβλητής μετρητή και στον προσδιορισμό της λειτουργίας των εμφωλευμένων «For». Στην πρώτη περίπτωση οι μαθητές δυσκολεύονται να καθορίσουν την τιμή βάσει της οποίας θα αυξάνεται ή θα μειώνεται η τρέχουσα τιμή της μεταβλητής που παίζει το ρόλο του μετρητή, ενώ στη περίπτωση των εμφωλευμένων «For», οι μαθητές θεωρούν ότι η εσωτερική «For» εκτελείται μόνο μία φορά σε κάθε επανάληψη της εξωτερικής «For» και όχι σύμφωνα με την αρχική/τελική τιμή και το βήμα ανανέωσης που καθορίζονται στη δομή της.

Τα διδακτικά προβλήματα που σχετίζονται με την επαναληπτική διαδικασία οφείλονται στο γεγονός ότι η επαναληπτική διαδικασία σαν έννοια δε συνάδει με τον τρόπο σκέψης των μαθητών. Οι μαθητές αντιλαμβάνονται την επανάληψη ως μια διαδοχή ενεργειών και όχι συναρτήσει ορισμένων παραγόντων που στον προγραμματισμό δηλώνονται ως λογικές εκφράσεις[21]. Αυτό έχει ως αποτέλεσμα οι μαθητές να δυσκολεύονται στον καθορισμό των ενεργειών που απαιτούνται να επαναληφθούν καθώς και στον καθορισμό της συνθήκης ελέγχου που δηλώνει πόσες φορές θα εκτελεστεί η επανάληψη. Τέλος οι μαθητές θα προτιμούσαν μία επαναληπτική δομή που να δίνει τη δυνατότητα τερματισμού της εκτέλεσής μέσω μιας σχετικής εντολής, δηλαδή να είναι δυνατή η διακοπή της εκτέλεσης της επανάληψης χωρίς να απαιτείται να ολοκληρωθεί η εκτέλεση του σώματος εντολών.

2.7.5 Αναδρομή

Με τον όρο αναδρομικές διαδικασίες περιγράφονται οι διαδικασίες που καλούν (άμεσα ή έμμεσα) τον εαυτό τους. Μια αναδρομική διαδικασία δεν περιέχει καμία εντολή εκχώρησης και δεν τροποποιεί κάποια μεταβλητή. Η αναδρομή είναι ένας τρόπος επίλυσης προβλήματος με ανάλυση του σε ένα ή περισσότερα υποπροβλήματα, τα οποία είναι πιο απλά στη επίλυση τους από το αρχικό πρόβλημα. Σαν έννοια είναι δυσνόητη τόσο στη σύλληψη όσο και στην εφαρμογή και η χρήση της θέτει μια σειρά από διδακτικά προβλήματα.

Η κυριότερη μαθησιακή δυσκολία που αντιμετωπίζουν οφείλεται στην παρανόηση ότι η αναδρομικότητα είναι μια κλασική επαναληπτική διαδικασία ενώ η αναδρομή πρέπει να θεωρηθεί ως μια λειτουργία που καλεί πολλαπλά αντίγραφα, τα οποία εξελίσσονται σε διαφορετικά επίπεδα του προγράμματος. Άλλες παρανοήσεις έχουν σχέση με τον τρόπο με τον οποίο λειτουργούν οι τοπικές μεταβλητές στο πλαίσιο αναδρομικών διαδικασιών. Τέλος δυσκολίες αντιμετωπίζουν επίσης με τον τερματισμό της αναδρομικής, κάτι που πρέπει να προβλεφθεί μέσα στο σώμα της αναδρομικής διαδικασίας.

Για αντιμετώπιστούν οι μαθησιακές δυσκολίες που εμφανίζονται πρέπει οι μαθητές να κατανοήσουν πως οικοδομείται μία αναδρομική λύση. Δηλαδή πρέπει να παραμετροποιήσουν το πρόβλημα ώστε να φανούν τα διάφορα στοιχεία από τα οποία εξαρτάται η λύση, να σπάσουν το πρόβλημα σε μικρότερα υπό-προβλήματα και να αναζητήσουν τη λύση ενός ήδη γνωστού προβλήματος η οποία θα δίνει και λύση σε κάποιο μέρος του ήδη υπάρχοντος.

2.7.6 Πίνακες

Οι πίνακες αποτελούν μία στατική δομή δεδομένων που χρησιμοποιείται για τη διαχείριση πολλών δεδομένων του ίδιου τύπου. Χαρακτηριστικό των πινάκων αποτελεί η διάστασή τους. Η αναφορά σε στοιχεία του πίνακα γίνεται μέσω του ονόματος του πίνακα ακολουθούμενο από ένα δείκτη.

Μια από τις σημαντικότερες δυσκολίες που αντιμετωπίζουν οι μαθητές κατά τη διαχείριση των πινάκων αφορά στη διάκριση ανάμεσα στο δείκτη (π.χ. i) και στο αντίστοιχο στοιχείο του πίνακα. Συχνά, οι μαθητές συγχέουν το δείκτη με το αντίστοιχο στοιχείο θεωρώντας ότι η χρησιμοποίηση του δείκτη (π.χ. $i+5$) έχει ως αποτέλεσμα την αναφορά στο αντίστοιχο στοιχείο [18].

Ιδιαίτερη δυσκολία παρουσιάζει στους μαθητές η χρησιμοποίηση διδιάστατων πινάκων. Σε αυτή την περίπτωση, οι μαθητές δυσκολεύονται να διακρίνουν το δείκτη που αντιστοιχεί στις γραμμές από αυτόν που αντιστοιχεί στις στήλες – συχνά αντιμεταθέτουν τους δύο δείκτες με αποτέλεσμα να μη διατρέχουν σωστά τα στοιχεία του πίνακα. Επιπλέον, αντιμετωπίζουν δυσκολία στην επεξεργασία των στοιχείων ενός διδιάστατου πίνακα κατά γραμμές ή κατά στήλες (π.χ. εύρεση του αθροίσματος των στοιχείων κατά στήλες). Η δυσκολία αυτή οφείλεται στο γεγονός ότι απαιτείται η χρήση εμφωλευμένων επαναληπτικών δομών προκειμένου να διατρέξουν τα στοιχεία του πίνακα, και στην οπτική αναπαράσταση που συνήθως χρησιμοποιείται για τους πίνακες – επειδή ένας μονοδιάστατος πίνακας παριστάνεται ως ένα σύνολο από συνεχόμενα «κουτιά» ζωγραφισμένα είτε κάθετα είτε οριζόντια, στην περίπτωση των διδιάστατων πινάκων δε μπορούν να αναγνωρίσουν ποιες σειρές «κουτιών» αντιστοιχούν στις στήλες και ποιες στις γραμμές, με αποτέλεσμα να συγχέουν τις γραμμές με τις στήλες [18].

2.7.7 Εντολές εισόδου-εξόδου

Οι εντολές εισόδου-εξόδου είναι από τις βασικές προγραμματιστικές δομές που διδάσκονται οι μαθητές στα αρχικά εισαγωγικά μαθήματα του προγραμματισμού. Οι κυριότερες δυσκολίες που αντιμετωπίζουν οι μαθητές αφορούν [23]:

- Στη σύνταξη των εντολών εισόδου-εξόδου: Οι μαθητές, συχνά, δεν είναι σε θέση να συνδυάσουν μεταβλητές με μηνύματα, που επιθυμούν να εμφανιστούν στην οθόνη του υπολογιστή, κατά την είσοδο ή κατά την εμφάνιση της τιμής των μεταβλητών. Οι δυσκολίες οφείλονται σε μεγάλο βαθμό στο συντακτικό των γλωσσών προγραμματισμού – συνήθως απαιτείται η χρήση του συμβόλου των εισαγωγικών για την εμφάνιση μηνυμάτων - και στη χρησιμοποίηση διαφορετικού τύπου μορφοποίησης για την είσοδο-εξόδο διαφορετικού τύπου μεταβλητών. Για παράδειγμα, οι μαθητές πολλές φορές περικλείουν το όνομα της μεταβλητής σε εισαγωγικά ή χρησιμοποιούν λάθος μορφοποιήσεις ή και καθόλου
- Στην αντιστοίχιση των τιμών εισόδου και των μεταβλητών: Κατά την εισαγωγή τιμών, οι μαθητές θεωρούν ότι η σειρά των μεταβλητών δεν αντιστοιχεί στις τιμές εισόδου. Συχνά, όταν εισάγουν τιμές εισόδου παρεμβάλλουν κενά, κόμματα και άλλους χαρακτήρες με αποτέλεσμα οι τιμές των μεταβλητών να μην είναι οι αναμενόμενες.

Επιπλέον, οι μαθητές αντιμετωπίζουν δυσκολία στην κατανόηση και στον προσδιορισμό του αποτελέσματος εκτέλεσης της εντολής εισόδου. Θεωρούν δηλαδή ότι η εντολή εισόδου δεν έχει ως αποτέλεσμα την εκχώρηση της τιμής εισόδου σε μία μεταβλητή.

Συχνά, συνδυάζουν την εντολή εισόδου με την εντολή ανάθεσης τιμής προκειμένου να ανατεθεί η επιθυμητή τιμή εισόδου στην αντίστοιχη μεταβλητή.

3. Προγραμματιστικά περιβάλλοντα για αρχάριους προγραμματιστές

3.1 Ταξινόμηση

Τα προγραμματιστικά περιβάλλοντα για αρχάριους μπορούν να διακριθούν σε δύο μεγάλες κατηγορίες: αυτά που διδάσκουν προγραμματισμό αποκλειστικά για την εκμάθηση του ίδιου του προγραμματισμού και αυτά που μέσω του προγραμματισμού επιχειρούν να διδάξουν ένα διαφορετικό γνωστικό αντικείμενο. Έτσι, η ταξινόμηση των συστημάτων γίνεται αρχικά με κριτήριο τους σκοπούς του συστήματος (διδασκαλία ή χρήση του προγραμματισμού) και δευτερευόντως με κριτήριο την πλευρά του προγραμματισμού που το σύστημα επιχειρεί να υλοποιήσει.

3.2 Διδακτικά συστήματα προγραμματισμού

Αυτά τα συστήματα έχουν σχεδιαστεί με σκοπό να βοηθήσουν στην εκμάθηση του προγραμματισμού. Τα περισσότερα συστήματα σε αυτή την κατηγορία είναι (ή περιέχουν) απλά προγραμματιστικά εργαλεία που εκθέτουν στους αρχάριους ορισμένες από τις βασικές προγραμματιστικές έννοιες. Μετά την απόκτηση εμπειρίας από ένα εκπαιδευτικό σύστημα, οι μαθητές οφείλουν να μεταβούν σε βασικότερες και εμπορικά δημοφιλέστερες γλώσσες. Λίγα είναι τα συστήματα που επιχειρούν να διδάξουν μια τέτοια γλώσσα απ' την αρχή ώστε να μην υπάρξει καμία μετάβαση. Αντιθέτως, τα εισαγωγικά προγραμματιστικά εργαλεία συνήθως κατασκευάζονται ώστε να θυμίζουν τη γλώσσα που αποσκοπεί να μάθει ο χρήστης, έτσι ώστε η μετάβαση να είναι ομαλότερη. Οι εμπορικά γνωστές γλώσσες όμως δεν σχεδιάζονται έχοντας ως κύριο γνώμονα τον αρχάριο, έτσι οι σχεδιαστές εκπαιδευτικών συστημάτων ταλαντεύονται κατά την κατασκευή τους ανάμεσα σε δύο στόχους: την δημιουργία ενός εύκολου προγραμματιστικού περιβάλλοντος για αρχάριους και την δημιουργία ενός περιβάλλοντος που θα κάνει όσο το δυνατόν ομαλότερη τη μετάβασή τους σε ένα γενικό προγραμματιστικό εργαλείο.

Τα εκπαιδευτικά εργαλεία επικεντρώνονται σε προγραμματιστικά ζητήματα που παραδοσιακά είναι δύσκολα για αρχάριους. Ενδιαφέρονται κυρίως για τον μηχανισμό με τον οποίο δουλεύει ολόκληρη η προγραμματιστική διαδικασία, δηλαδή τον τρόπο με τον οποίο ο χρήστης εκφράζει τις προθέσεις του στον υπολογιστή, αλλά και την κατανόηση των ενεργειών αυτού. Άλλα συστήματα επιχειρούν να παρουσιάσουν τον προγραμματισμό με τέτοιο τρόπο ώστε να απευθύνεται σε ένα ευρύτερο κοινό, με μεθόδους όπως π.χ. η συνεργατική μάθηση.

3.3 Μηχανισμοί του προγραμματισμού

Τα συστήματα σε αυτή την κατηγορία είναι σχεδιασμένα γύρω από την υπόθεση πως για να κατασκευαστεί ένα πρόγραμμα το βασικό εμπόδιο που προβάλλεται είναι η γνώση των μηχανισμών του. Συγκεκριμένα οι μαθητές οφείλουν να εξοικειωθούν με

- το πώς να εκφράζουν ένα πρόγραμμα σε μια γλώσσα,
- πώς να οργανώνουν τις εντολές και
- το πώς ο υπολογιστής εκτελεί αυτό το πρόγραμμα.

3.3.1 Εκφράζοντας ένα πρόγραμμα

Οι αρχάριοι συνήθως δυσκολεύονται στο να εκφράσουν τις προθέσεις τους στον υπολογιστή με τη βοήθεια μια προγραμματιστικής γλώσσας, Τα συστήματα σε αυτή την κατηγορία εξετάζουν δυο πιθανές οδούς ώστε να κάνουν την διαδικασία ευκολότερη για τους χρήστες: τη βελτίωση της γλώσσας ώστε να είναι προσιτή στο χρήστη και την ανάπτυξη διαφορετικών μεθόδων με τις οποίες ο χρήστης θα επικοινωνεί με τον υπολογιστή.

Απλοποίηση του εισαγομένου κώδικα

Πολλές γενικού σκοπού προγραμματιστικές γλώσσες κατά διαστήματα επιχειρούν να βελτιώσουν τη δομή τους έτσι ώστε ο κώδικάς τους να γίνει απλούστερος και πιο κατανοητός. Με οδηγό αυτή τη βελτίωση προσανατολίστηκαν στις εξής κατευθύνσεις:

- την **απλοποίηση της γλώσσας**
- τον **περιορισμό της γλώσσας** στην επίλυση συγκεκριμένων προβλημάτων
- την **παρεμπόδιση συντακτικών λαθών**.

Απλοποίηση της γλώσσας - Περιορισμός γλώσσας στην επίλυση συγκεκριμένων προβλημάτων

Με αυτή τη μέθοδο χρησιμοποιούνται ορισμένες βασικές παρατηρήσεις ώστε να μειωθεί ο αριθμός των συντακτικών δομών που ενδεχομένως θα δυσκόλευαν τον αρχάριο. Τέτοιες παρατηρήσεις-διορθώσεις συνήθως είναι αυστηρές συντακτικές δομές, εντολές με διαφορετική σημασία στα αγγλικά σε σχέση με τη γλώσσα κ.ά. Αποτελέσματα τέτοιων προσπαθειών είναι η BASIC, η SP/k, η Turing, η Blue Language, η JJ και η GRAIL.

BASIC, Dartmouth College, 1963[24]: Η BASIC σχεδιάστηκε έτσι ώστε να περιλαμβάνει ένα μικρό αριθμό εντολών ενώ αφαιρέθηκε από αυτήν και το άχρηστο συντακτικό. Το περιβάλλον θυσίαζε τον χρόνο που απαιτεί ο υπολογιστής προς όφελος του χρήστη, ενώ ο χρόνος απόκρισής του ήταν ιδιαίτερα μικρός

Οι δηλώσεις στην Basic αποτελούνται από 3 μέρη: έναν αριθμό γραμμής, ένα τελεστή κι ένα τελεστέο. Όλες οι εντολές ξεκινούν με μια αγγλική λέξη έτσι ώστε να είναι ευκολότερη η κατανόησή της από τον χρήστη (οι σχεδιαστές πίστευαν πως το LET S=S+1 θα ήταν πιο ευνόητο για τους αρχάριους). Οι δηλώσεις είναι παρόμοιες με τις περισσότερες γλώσσες, αφαιρέθηκαν όμως στοιχεία όπως το συντακτικό νόημα των κενών ανάμεσα από τις εντολές.

FORTRAN: do 30 i = 1, 10 m = m + I 30 continue	BASIC: 100 FOR I = 1 TO 10 110 LET S = S + I 120 NEXT I
---	--

SP/k, University of Toronto, 1977[25]: Η SP/k είναι ένα υποσύνολο της PL/1 επιλεγμένο έτσι ώστε να είναι ιδανικό για εισαγωγική διδασκαλία στον προγραμματισμό.

Απομακρύνθηκαν όλες οι *περιττές δομές*, οι *ασυνέπειες που οδηγούσαν σε σύγχυση* τους αρχάριους (π.χ. στην PL/1 η έκφραση $25+1/3$ ισούται με 5,33333), οι *δομές που συχνά χρησιμοποιούνταν λανθασμένα* (π.χ. οι δείκτες) και άλλες που *απευθύνονταν σε προχωρημένους χρήστες* (π.χ. δομές ταυτόχρονου προγραμματισμού). Το αποτέλεσμα όλης αυτής της διαδικασίας ήταν μια γλώσσα σαφώς πιο απλή και προτιμώμενη σε σχέση με την Fortran για διδασκαλία σε αρχάριους.

Blue Language, University of Monash, 1996[26]: Η Blue είναι μια αντικειμενοστραφής γλώσσα σχεδιασμένη ώστε να μπορεί να διδαχθεί ως πρώτη γλώσσα προγραμματισμού. Μετά την χρήση της για ένα χρόνο, οι μαθητές συνίσταται να μετακινηθούν σε μια άλλη, βιομηχανική γλώσσα όπως η C++. Οι σχεδιαστές της βασίστηκαν σε τέσσερις κανόνες για την κατασκευή της: 1) να υπάρχει μόνο ένας τρόπος για να κάνει ο χρήστης οτιδήποτε, 2) η γλώσσα να αντικατοπτρίζει ευκρινώς το θεωρητικό μοντέλο, 3) η γλώσσα να είναι αναγνώσιμη (σε κοινά Αγγλικά) για τους μαθητές έτσι ώστε να μπορούν να διαβάζουν σε λυμένα παραδείγματα, 4) και τέλος να υποστηρίζει διαδεδομένους μηχανισμούς ανάπτυξης λογισμικού (π.χ. pre and post conditions- δηλαδή καταστάσεις με ιδιότητα μνήμης).

JJ, California state university and California Institute of Technology, 1998[27]: Είναι μια Java-based γλώσσα σχεδιασμένη ώστε να μειώνει την πολυπλοκότητα του συντακτικού επιτρέποντας στους μαθητές να επικεντρωθούν στη φιλοσοφία του προγραμματισμού. *Αφαιρεί το μεγαλύτερο μέρος των σημείων στίξης ενώ υπάρχει μόνο ένας τρόπος για να κάνει ο χρήστης οτιδήποτε: ένα τύπος ακεραίων, ένας τρόπος να σχολιάσεις κτλ. Μετά από ένα εξάμηνο οι μαθητές μπορούν να ασχοληθούν με την Java.*

Computing weekly pay in JJ:	The same code in Java:
<pre> if (hours <= 40) then Set pay = 10 * hours Else Set pay = 400 + 15*(hours - 40) EndIf Output "The pay is " Outputln pay </pre>	<pre> if (hours <= 40) { pay = 10 * hours; } else { pay = 400 + 15 * (hours - 40); } // EndIf System.out.print ("The pay is "); System.out.println(pay); </pre>

GRAIL, University of Monash, 1999[28]: Είναι μια γλώσσα βασισμένη στην υπόθεση πως η έλλειψη εξοικείωσης των αρχάριων με το συντακτικό και την θεωρία είναι τα κύρια εμπόδια προς την κατάκτηση της γνώσης. Όπως και σε άλλα συστήματα, έτσι και σε αυτό ακολουθήθηκε μια φιλοσοφία κατά την κατασκευή του, αποτελούμενη από τρεις κανόνες:

- την διατήρηση ενός *συνεπούς συντακτικού* (χρησιμοποίηση όρων με τους οποίους ακόμη και οι αρχάριοι είναι εξοικειωμένοι)
- την αποφυγή τυπικών προγραμματιστικών όρων που έχουν όμως διαφορετικό νόημα στα κοινά Αγγλικά
- και την *αυστηρή χρησιμοποίηση απλών δομών* που έχουν ένα κοινό και προφανές τρόπο σύνταξης.

Δύο παραδείγματα της τελικής μορφής που είχε η γλώσσα σύμφωνα με τα παραπάνω είναι π.χ. η χρησιμοποίηση του συμβόλου «X» για πολλαπλασιασμό, αντί του κλασσικού προγραμματιστικού «*» και η αντικατάσταση του «=» με «->» για μια εκχώρηση τιμής. Ακόμη, δομές όπως οι δείκτες αφαιρέθηκαν.

Παρεμπόδιση συντακτικών λαθών

Η κατανόηση του συντακτικού είναι ίσως η μεγαλύτερη και δυσκολότερη πρόκληση κατά την εκμάθηση του προγραμματισμού. Τα συστήματα σε αυτή την κατηγορία είναι εργαλεία για ήδη διαδεδομένες γλώσσες προγραμματισμού, όπως Pascal και Fortran, τα οποία μέσω ιεραρχικής δόμησης των προγραμμάτων αποτρέπουν συντακτικά λάθη.

Cornell Program Synthesizer, Cornell University, 1981[28]: Ήταν ένα σύστημα σχεδιασμένο να αποτρέπει τους χρήστες να κάνουν συντακτικά λάθη. Χρησιμοποιώντας τον Synthesizer οι μαθητές δημιουργούσαν δομημένα προγράμματα *επιλέγοντας ήδη από ορισμένα πρότυπα (templates)*. Ένα πρότυπο περιλάμβανε δεσμευμένες κενές θέσεις για δηλώσεις, συνθήκες και φράσεις τις οποίες ο χρήστης καλούνταν να γεμίσει. Η πλοήγηση ανάμεσα σε αυτές τις θέσεις αλλά και στο συντακτικό δέντρο του προγράμματος υλοποιούνταν με τα βελάκια του πληκτρολογίου. Το σύστημα όμως δεν επέτρεπε έστω και μια προσωρινή λάθος σύνταξη ενός προγράμματος, χαρακτηριστικό που αποδείχτηκε μη λειτουργικό και μετατράπηκε στην επόμενη έκδοση σε οδηγό που μάκκαρε τα πιθανά λάθη του χρήστη με σκοπό να τραβήξει την προσοχή του.

An If-statement template in the Cornell Program Synthesizer:

```
IF (condition)
  THEN statement
  ELSE statement
```

Gnome, Carnegie Mellon University, 1984[28]: Τα συστήματα Gnome αποτέλεσαν μια προσπάθεια να κατασκευαστεί ένας επεξεργαστής δομών (structure editor) για αρχάριους, πιο ευπροσάρμοστος όμως από τον Cornell Program Synthesizer. Το Gnome *πρόβαλλε τα προγράμματα ιεραρχικά*, ωθώντας τους μαθητές να τα σκεφτούν ως μια ιεραρχική δομή διαδικασιών. Οι μαθητές με το Gnome είχαν την δυνατότητα να πλοηγηθούν στο πρόγραμμα τους χρησιμοποιώντας βέλη που αντιπροσώπευαν τις κινήσεις σε ένα συντακτικό δέντρο. Το Gnome απεικόνιζε τα τμήματα του κώδικα στην κοινή κειμενική μορφή των προγραμμάτων, όμως μετά από μια διόρθωση (ή κατ' απαίτηση) είχε τη δυνατότητα να αναλύει όλο το πρόγραμμα, αναφέρει όλα τα πιθανά συντακτικά λάθη και εμποδίζει τον προγραμματιστή να προχωρήσει αν το πρόγραμμά του δεν είναι σωστό. Όλοι αυτοί οι προληπτικοί μηχανισμοί όμως συνήθως έκαναν τον προγραμματιστή να ασχοληθεί περισσότερο απ' όσο χρειάζεται με το συντακτικό μέρος του προγράμματος, ενώ ο μηχανισμός που τον απέτρεπε από το να συνεχίσει συχνά καταντούσε εκνευριστικός.

McGnome, Carnegie Mellon University, 1994[28]: Το McGnome προσπάθησε να συνδυάσει το μοντέλο διόρθωσης της δομής του προγράμματος που υλοποιούσε το Gnome με το μοντέλο διόρθωσης του κειμένου του προγράμματος που υλοποιείται από τα τυπικά προγραμματιστικά εργαλεία. Η χρήση του Gnome οδήγησε τους ερευνητές στο συμπέρασμα πως οι μαθητές αντιμετωπίζουν προβλήματα κατά την διάρκεια της πλοήγησης στο συντακτικό δέντρο, έτσι το McGnome ενσωμάτωσε ένα νέο point & click σύστημα πλοήγησης με χρήση του mouse. Ακόμη, το McGnome δεν απαιτούσε διαρκώς το πρόγραμμα να μην παρουσιάζει λάθη, αλλά μετέτρεπε το συντακτικό δέντρο σε κείμενο στο οποίο επιτρεπόταν χωρίς περιορισμούς οι διορθώσεις του χρήστη. Έτσι δεν υποστηριζόταν κάποιος μηχανισμός αποτροπής λαθών, μετά την ανάλυση του προγράμματος, όμως εντοπιζόταν οποιαδήποτε λάθη δίνοντας στον χρήστη τη δυνατότητα να τα διορθώσει πριν μεταβεί σε κάποιο άλλο μέρος του κώδικα.

Εύρεση εναλλακτικών τρόπων πληκτρολόγησης προγραμμάτων

Παράλληλα με τις προσπάθειες να γίνει πιο απλή μια γλώσσα προγραμματισμού, εκδηλώθηκαν και άλλες που σκοπό είχαν να δώσουν μια ακόμη μεγαλύτερη βοήθεια στον αρχάριο κατά την πιο κρίσιμη στιγμή, αυτήν της πληκτρολόγησης. Αυτές οι

προσπάθειες μπορούν να χωριστούν σε τρία είδη: τη δημιουργία αντικειμένων που αναπαριστούν κώδικα, τη χρήση «έτοιμων» ενεργειών του προγραμματιστή στο interface της εφαρμογής και την παροχή πολλαπλών μηχανισμών για την δημιουργία προγραμμάτων.

Κατασκευή προγραμμάτων χρησιμοποιώντας γραφικά ή φυσικά αντικείμενα

Αυτά τα αντικείμενα αναπαριστούν συνήθως εντολές, μεταβλητές και δομές ελέγχου. Μπορούν να συνδυαστούν κατάλληλα ώστε να δημιουργήσουν ένα πρόγραμμα.[28]

Tortis slot Machine, MIT Artificial Intelligence Lab, 1976: Το Tortis slot Machine είναι ένα φυσικό περιβάλλον που επιτρέπει σε μικρά παιδιά να ελέγχουν μια ρομποτική χελώνα τύπου Logo. Επειδή η ρομποτική χελώνα είναι πολύ αργή, παρέχεται επιπρόσθετα μια γραφική προσομοίωση στην οθόνη για προχωρημένους χρήστες. Η Slot Machine αποτελείται από ένα σύνολο καρτών εντολών και τετράγωνων κουτιών που αναπαριστούν διαδικασίες και περιέχουν υποδοχές για αυτές τις κάρτες. Τα παιδιά δημιουργούν προγράμματα τοποθετώντας κάρτες στις υποδοχές των κουτιών σε σειρά, ενώ στη συνέχεια η ρομποτική χελώνα τα εκτελεί.

Pict, University of Washington, 1984: Το Pict επιτρέπει στους αρχάριους προγραμματιστές να δημιουργήσουν απλά προγράμματα συνδέοντας εικονίδια που αναπαριστούν εντολές. Τα προγράμματα αυτά μπορούν να επιτελέσουν απλούς αριθμητικούς υπολογισμούς, χρησιμοποιώντας την πρόσθεση και την αφαίρεση ακεραίων, την ανάθεση τιμών σε μεταβλητή και Boolean ελέγχους. Για την τοποθέτηση των εικονιδίων οι μαθητές χρησιμοποιούν ένα joystick, ενώ μετά την τοποθέτησή τους μπορούν να τα συνδέσουν επιλέγοντας απλά τις δύο άκρες τους. Η εκτέλεση προγράμματος αναπαρίσταται σειριακά, και ο χρήστης μπορεί να την εκκινήσει από οποιοδήποτε σημείο του προγράμματος επιθυμεί. Τέλος, σε περίπτωση λάθους η εκτέλεση σταματά, παραθέτοντας στον χρήστη το λάθος τμήμα του κώδικα.

Play, University of Washington, 1986: Είναι ένα σύστημα σχεδιασμένο να επιτρέπει σε παιδιά προσχολικής ηλικίας να σχεδιάζουν γραφικά παιχνίδια χρησιμοποιώντας μια εικονική γλώσσα. Οι ιστορίες αποτελούνται από μια σειριακή ακολουθία ενεργειών που εμφανίζονται στην κορυφή της οθόνης, πάνω από την ίδια την ιστορία, και μοιάζουν με στιγμιότυπα φωτογραφικού φιλμ. Ο χαρακτήρας, το τι πρέπει να κάνει και μια επιπρόσθετη πληροφορία (συνήθως η κατεύθυνση της κίνησης) προσδιορίζουν κάθε ενέργεια μιας ιστορίας. Οι χρήστες μπορούν να δημιουργήσουν τους δικούς τους χαρακτήρες.

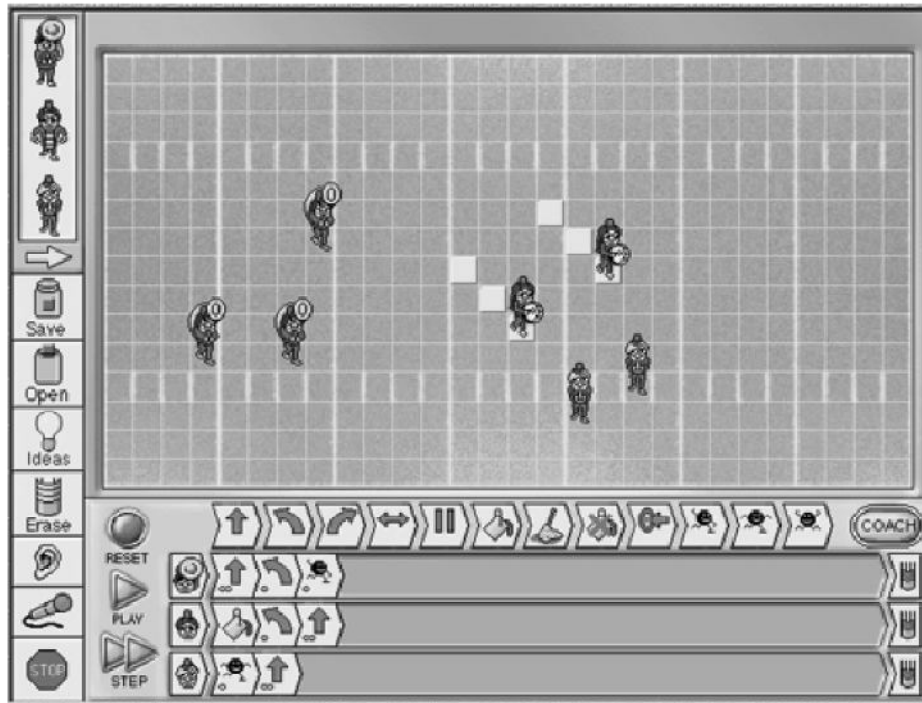
Show and Tell, Washington University and Bell Labs 1990: Είναι μια οπτική γλώσσα βασισμένη σε διαγράμματα ροής, σχεδιασμένη για παιδιά. Ένα πρόγραμμα στην Show and Tell αποτελείται από μια σειρά συνδεδεμένων κουτιών. Ένα κουτί μπορεί να αναπαριστά μια τιμή ή μια πράξη πάνω σε τιμές. Το πρόγραμμα περιλαμβάνει κουτιά που αναπαριστούν βασικές αριθμητικές λειτουργίες, είσοδο και έξοδο συστήματος και ορισμένα κουτιά για ιδιαίτερες λειτουργίες όπως αναπαραγωγή ήχων ή χρονόμετρα κ.ά. Τα παιδιά μπορούν να δημιουργήσουν διαδικασίες σχεδιάζοντας το δικό τους εικονίδιο και προσδιορίζοντας τι θα συμβαίνει μέσα σε αυτό χρησιμοποιώντας άλλα κουτιά.

My Make Believe Castle, Logo computer Systems Incorporated, 1995: Είναι ένα πρόγραμμα που απευθύνεται σε παιδιά ηλικίας 4-7 ετών και περιέχει δραστηριότητες σχεδιασμένες να βοηθούν στην ανάπτυξη ικανοτήτων επίλυσης προβλημάτων, στην

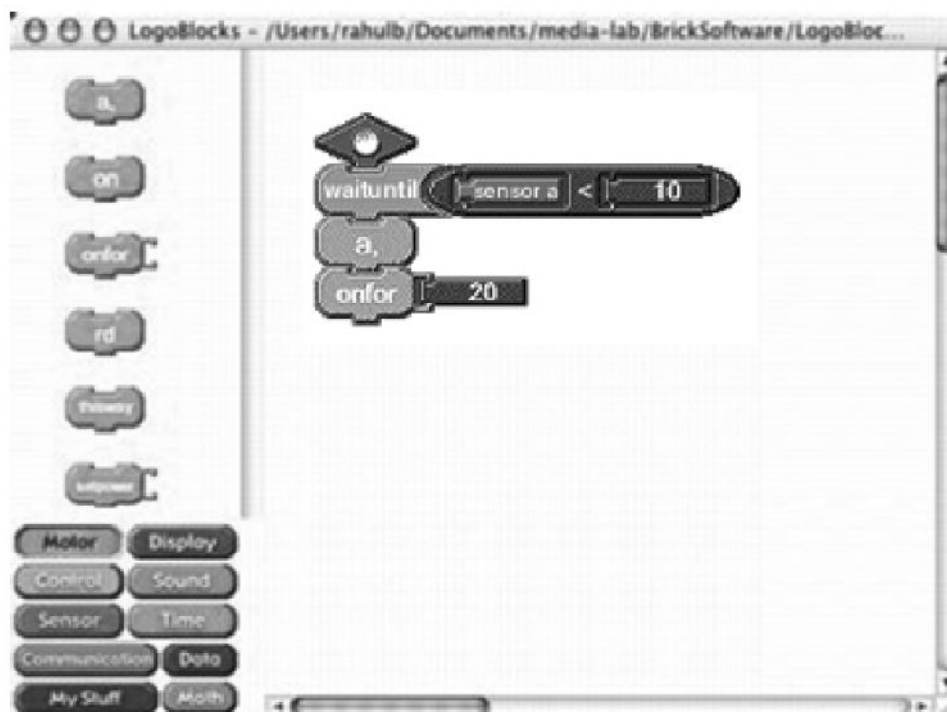
κριτική τους σκέψη, στην κατάστρωση σχεδίων και στην απομνημόνευση. Η κεντρική ιδέα εκτυλίσσεται γύρω από ένα κάστρο. Το κάστρο αποτελείται από έναν αριθμό δωματίων, κάθε ένα από τα οποία περιέχει μια δραστηριότητα. Στην αυλή του κάστρου κινούνται χαρακτήρες όπως ο πρίγκιπας, η πριγκίπισσα, ο δράκος και τα άλογα. Όταν ο χρήστης κάνει κλικ πάνω τους με κάποιο από τα διαθέσιμα εργαλεία, οι χαρακτήρες αυτοί χορεύουν, γλιστράνε σε μπανανόφλουδες κ.ά. Μετά την εξοικείωση των παιδιών με το περιβάλλον, μπορούν πλέον να συντάξουν απλά προγράμματα, καθορίζοντας τι θα κάνει καθένας από τους παραπάνω χαρακτήρας όταν συναντά κάποιον άλλο (π.χ. «Nicky dances when she meets the horse»).



Thinking Things Collection 3-Half time, Edmark Corporation, 1995: Το Half Time είναι ένα μόνο παιχνίδι από τη συλλογή Thinking things collection, στο οποίο ο χρήστης καλείται να δημιουργήσει ένα σόου με χορευτές (cheerleader) και μια μουσική μπάντα. Οι χρήστες μπορούν να επιλέξουν σχετικούς χαρακτήρες από το πάνω αριστερά μέρος της οθόνης και να τους τοποθετήσουν στο γήπεδο. Οι κινήσεις αυτών των χαρακτήρων είναι παρόμοιες με τις κινήσεις της χελώνας Logo, ενώ μπορούν να χρησιμοποιηθούν και να δοθούν εντολές το πολύ σε 30 χαρακτήρες.



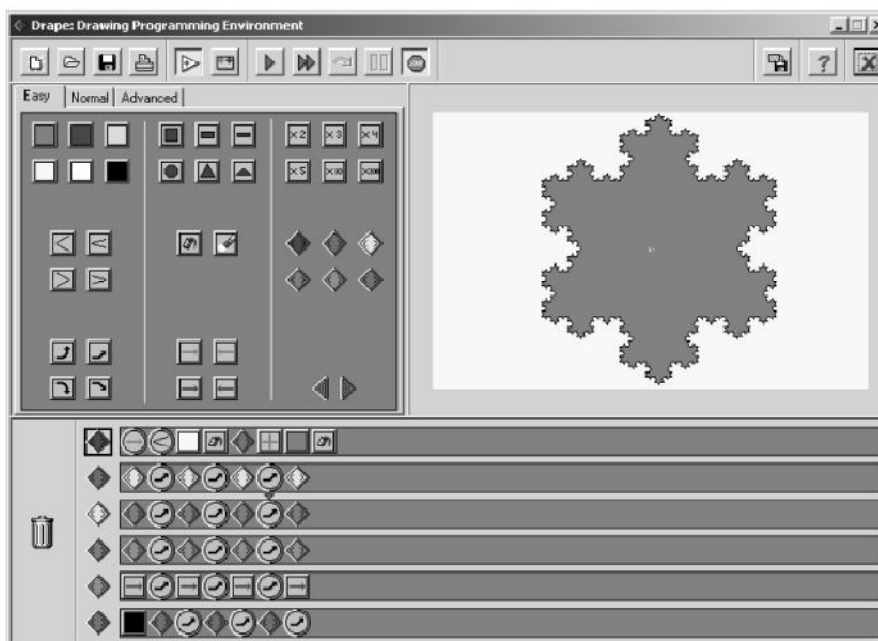
Logoblocks, MIT Media Lab, 1996: Είναι ένα γραφικό προγραμματιστικό περιβάλλον σχεδιασμένο για το Programmable Brick, τον προπομπό του RXC των Lego Mindstorms. Αυτά τα γραφικά κουτιά μπορούν να συρθούν από μια παλέτα εργαλείων που βρίσκεται στην οθόνη στην περιοχή που είναι ορισμένη για τη δημιουργία προγραμμάτων. Τα κουτιά στην παλέτα εργαλείων αναπαριστούν περισσότερο είδη αντικειμένων παρά όλα τα διαθέσιμα αντικείμενα.



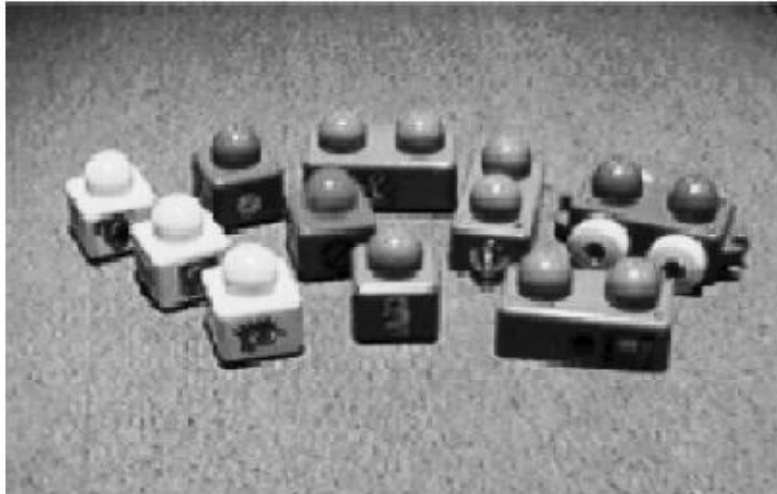
Pet Park Blocks, MIT Media Lab, 1998: Το Pet Park Blocks είναι μια γραφική προγραμματιστική γλώσσα, εμπνευσμένη από το Logoblocks, ενώ αναπτύχθηκε για το

συνεργατικό περιβάλλον Pet Park. Τα animations προβάλλονται ως τετράγωνα συνδεδεμένα μεταξύ τους και οι συνθήκες ως ρόμβοι όπου στις γωνίες τους προστίθενται τα στοιχεία. Όπως στο Logoblocks, οι πρότυπες προγραμματιστικές δομές είναι διαθέσιμες μέσω μιας παλέτας από την οποία οι χρήστες μπορούν να τις σύρουν σε μια ενεργή περιοχή. Τέλος, είναι διαθέσιμη και η απεικόνιση του προγράμματος ως κείμενο ώστε οι μαθητές να μπορούν σταδιακά να μεταβούν στον κλασικό προγραμματισμό.

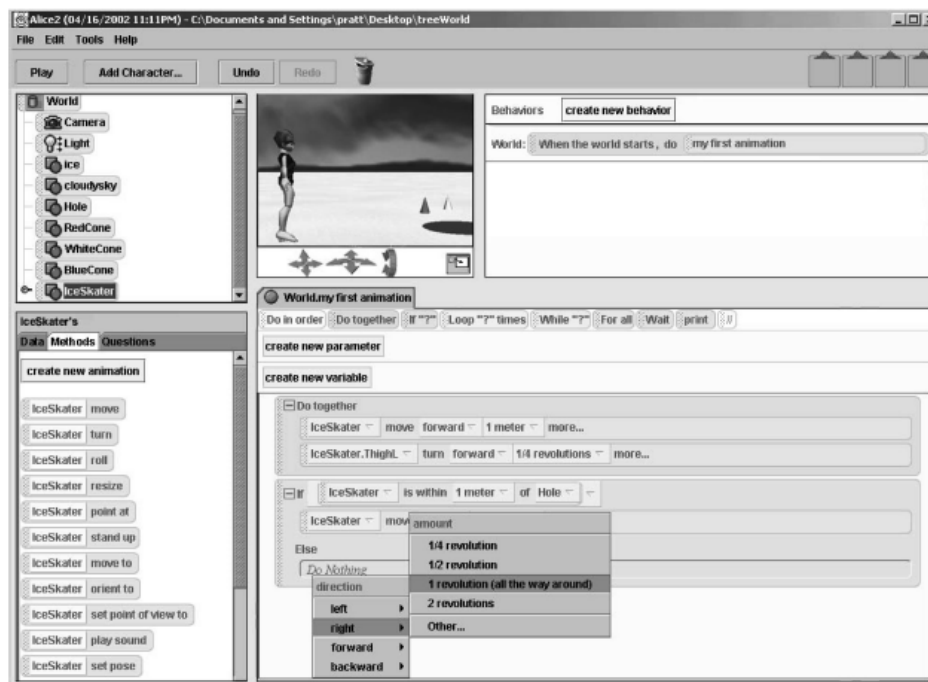
Drape, University of Utrecht, 2000: Είναι ένα προγραμματιστικό περιβάλλον που επιτρέπει στους χρήστες να ζωγραφίζουν εικόνες. Διατίθεται μια συλλογή από έτοιμα εικονίδια στο αριστερό μέρος της διεπαφής, τα οποία αντιπροσωπεύουν ενέργειες ανάλογες με αυτές μιας ρομποτικής χελώνας Logo. Για την δημιουργία προγράμματος ο χρήστης σέρνει αυτές εντολές στο κάτω μέρος της οθόνης του προγράμματος, ενώ αυτό εκτελείται από αριστερά προς τα δεξιά.



Electronic Blocks, University of Queensland, 2000: Είναι ένα σύνολο από πραγματικά Lego τουβλάκια, σχεδιασμένα να επιτρέπουν σε χρήστες (κυρίως παιδιά από 3-8 ετών) να δημιουργούν κατασκευές με ενδιαφέρουσες συμπεριφορές. Τα παιδιά μπορούν να κατασκευάσουν για παράδειγμα πύργους τούβλων που φωτίζουν όταν μιλούν ή αυτοκινητάκια που κινούνται όταν ένα φως πέφτει πάνω τους. Παρέχονται τρία είδη από τουβλάκια: 1) τουβλάκια αισθητήρων που αναγνωρίζουν ως είσοδο μια δέσμη φωτός, ήχο ή αφή 2) λογικά τουβλάκια (πράξεις AND, NOT, TOGGLE και DELAY) και τουβλάκια ενεργειών που μπορούν να παράγουν ήχο, φως και κίνηση. Η σύνταξη είναι πολύ απλή: αρκεί κάθε στοίβα από τουβλάκια να περιέχει τουλάχιστον ένα τουβλάκι ενεργειών και ένα αισθητήρα, με το πρώτο να βρίσκεται στην βάση της στοίβας.



Alice2, Carnegie Mellon University, 2003: Η Alice είναι ένα προγραμματιστικό σύστημα για κατασκευή 3D εικονικών κόσμων, μικρών ταινιών αλλά και παιχνιδιών. Χρησιμοποιείται πάλι η μέθοδος drag 'ή' drop για τη δημιουργία του προγράμματος, ενώ μετά την τοποθέτηση ενός στοιχείου, το σύστημα το αναγνωρίζει και προβάλλει όλα τα δυνατά μενού για εισαγωγή παραμέτρων σε αυτό. Διατίθενται ακόμη δομές ελέγχου και επαναληπτικές δομές. Η Alice αποτελεί ένα ασφαλές, χωρίς επιτρεπόμενα συντακτικά λάθη περιβάλλον για την ομαλή εισαγωγή σε βασικές προγραμματιστικές έννοιες.



Magic Forest, Logotron, 2002: Το Magic Forest δίνει τη δυνατότητα σε παιδιά άνω των 4 ετών να παίξουν, να αλλάξουν και να δημιουργήσουν δραστηριότητες αποτελούμενες από δυσδιάστατα sprites που μπορούν να κινούνται να αλλάζουν εμφάνιση και να αντιδρούν σε απλά γεγονότα. Σε κάθε sprite μπορεί να αποδοθεί ένα σύνολο κανόνων, δηλαδή τα γεγονότα που μπορεί να συμβούν και το πότε θα συμβαίνουν. Οι ενέργειες αλλά και τα αντικείμενα αναπαρίστανται γραφικά με πέτρες, ενώ οι πρώτες μπορούν να αλλάξουν την ταχύτητα ή την κατεύθυνση ενός αντικειμένου, να στείλουν ένα μήνυμα ή να αναπαράγουν ήχους. Ένα πρόγραμμα είναι μια ακολουθία από τέτοιες πέτρες.



Δημιουργία προγραμμάτων με τη χρήση πρότυπων ενεργειών διεπαφής

Τα συστήματα στην προηγούμενη κατηγορία δημιουργούσαν προγράμματα «τακτοποιώντας» κατά κάποιον τρόπο μια σειρά από γραφικά ή φυσικά αντικείμενα, ενώ σε αυτή την κατηγορία χρησιμοποιούνται ενέργειες διεπαφής (όπως το πάτημα ενός κουμπιού ή κίνηση σε χώρο). Τέτοια συστήματα είναι:[28]

Tortis-Button Box, MIT AI Lab, 1976: Είναι ένα φυσικό περιβάλλον που επιτρέπει στους μαθητές να ελέγχουν μια ρομποτική χελώνα τύπου Logo. Διαθέτει τεσσάρων ειδών κουτιά τα οποία έχουν τα εξής χαρακτηριστικά:

- Κουτιά με κουμπιά που μετακινούν τη χελώνα, κρατάνε ή αφήνουν το στυλό , ανάβουν ή σβήνουν ένα φως και ηχούν μια κόρνα.
- Κουτιά που διαθέτουν αριθμούς ώστε ο μαθητής να μπορεί να επαναλαμβάνει μια διαδικασία όσες φορές θέλει.
- Κουτιά με περιοχή προγραμματισμού, με τη βοήθεια των οποίων η χελώνα μπορεί να «θυμάται» εντολές και να τις αναπαράγει.
- Κουτιά που δημιουργούν τέσσερις διαδικασίες, διακρινόμενες από χρώματα, που μπορούν να καλούν η μία την άλλη.

Legosheets, University of Colorado, 1995: Το Legosheets επιχειρεί την ομαλή εισαγωγή των αρχάριων στον προγραμματισμό με τη βοήθεια του MIT Programmable Brick. Αρχικά οι μαθητές ασχολούνται χειρωνακτικά με τον έλεγχο από τα τουβλάκια και σταδιακά προχωρούν στην σύνταξη προγραμμάτων. Ξεκινώντας από μια προσομοιωμένη έκδοση, οι μαθητές εξοικειώνονται με τον χειρισμό τους και των μηχανημάτων που μπορούν να τους προσαρτηθούν και όταν στη συνέχεια έχουν αντιληφθεί επαρκώς όλες αυτές τις διαδικασίες, τους παρέχεται ένας κειμενογράφος στον οποίο μπορούν να εισάγουν ολοκληρωμένα προγράμματα.

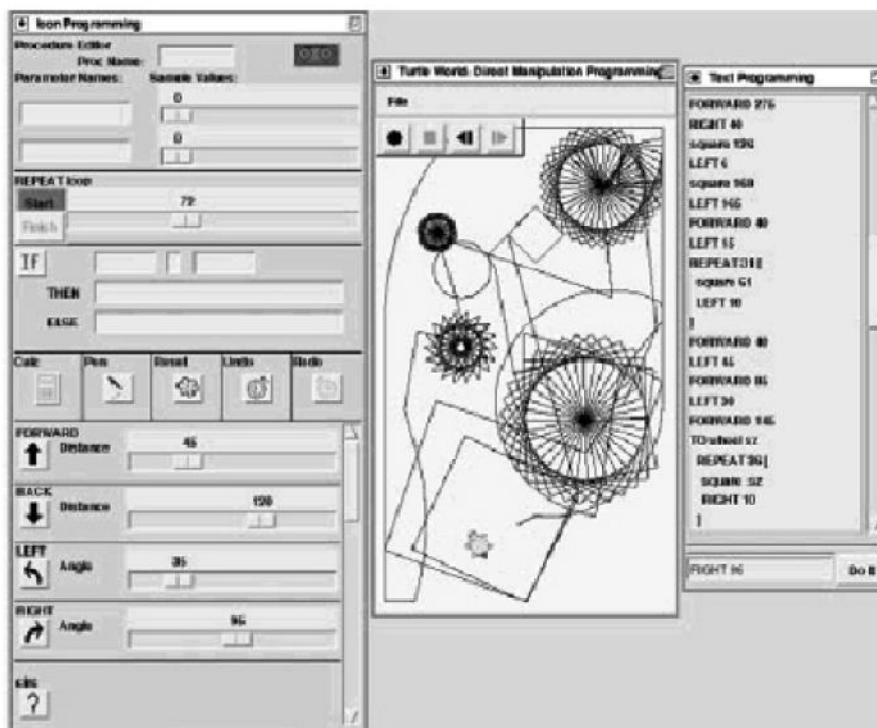
Curlybot, MIT Media Lab, 2000: Είναι ένα εκπαιδευτικό παιχνίδι για παιδιά άνω των τεσσάρων ετών. Αποτελείται από ένα δίτροχο όχημα με κατάλληλο εξοπλισμό που του

επιτρέπει να καταγράφει και να επαναλαμβάνει τις κινήσεις του. Διαθέτει ακόμη ένα και μοναδικό LED καθώς και ένα κουμπί. Τα δύο αυτά στοιχεία είναι υπεύθυνα για την καταγραφή ή την αναπαραγωγή μιας κίνησης. Το Curlybot δεν παρέχει την πολυπλοκότητα μιας προγραμματιστικής γλώσσας, εξοικειώνει όμως παιδιά μικρής ηλικίας με την έννοια των επαναλήψεων.

Παροχή πολλαπλών μηχανισμών για δημιουργία προγραμμάτων

Η εισαγωγή ενός προγράμματος ως κείμενο μπορεί να αποδειχτεί πολύ δυσκολότερη από μια εναλλακτική μέθοδο (π.χ. συμπλήρωση φόρμας), αλλά συνήθως δίνει μεγαλύτερη προγραμματιστική δύναμη. Σ' ένα σύστημα παροχής πολλαπλών μηχανισμών εισόδου προγράμματος και αναπαράστασης αποτελεσμάτων σε διάφορες μορφές, ο μαθητής μπορεί να χρησιμοποιήσει μια εναλλακτική μορφή εισόδου ως προθάλαμο για την χρησιμοποίηση μιας πιο παραδοσιακής και ισχυρότερης μορφής. Τα περισσότερα απ' αυτά τα συστήματα παρέχουν ταυτόχρονα και αναπαράσταση του προγράμματος ως κείμενο ώστε ο μαθητής να έρχεται οποιαδήποτε στιγμή σε επαφή με την παραδοσιακή προγραμματιστική μορφή. Ακολουθεί ένα παράδειγμα τέτοιου συστήματος[28]

LEOGO, Canterbury, 1997: Το Leogo παράγει σχήματα παρόμοια με αυτά της χελώνας Logo. Δεν περιορίζεται σε μία μέθοδο εισαγωγής προγράμματος, αλλά διαθέτει τρεις: 1) εισαγωγή κειμένου με μια γλώσσα βασιζόμενη στη Logo 2) απευθείας χειρισμός με τη βοήθεια μιας διεπαφής 3) και τέλος μια εικονική γλώσσα με πρότυπα προγράμματα και κινήσεις της χελώνας. Αξίζει να σημειωθεί πως είναι δυνατή η ταυτόχρονη απεικόνιση του προγράμματος και στις 3 μεθόδους.



3.3.2 Δομώντας προγράμματα

Αυτά τα συστήματα επικεντρώνονται στον τρόπο δημιουργίας της συνολικής δομής του κώδικα παρά στην σύνταξη μικρών τμημάτων του. Ως επί το πλείστον χρησιμοποιούν νέες προγραμματιστικές μεθόδους ενώ χωρίζονται σε 2 κατηγορίες : αυτά που αλλάζουν την δομή ενός προγράμματος και αυτά που κάνουν την δομή πιο κατανοητή.

Νέα προγραμματιστικά μοντέλα

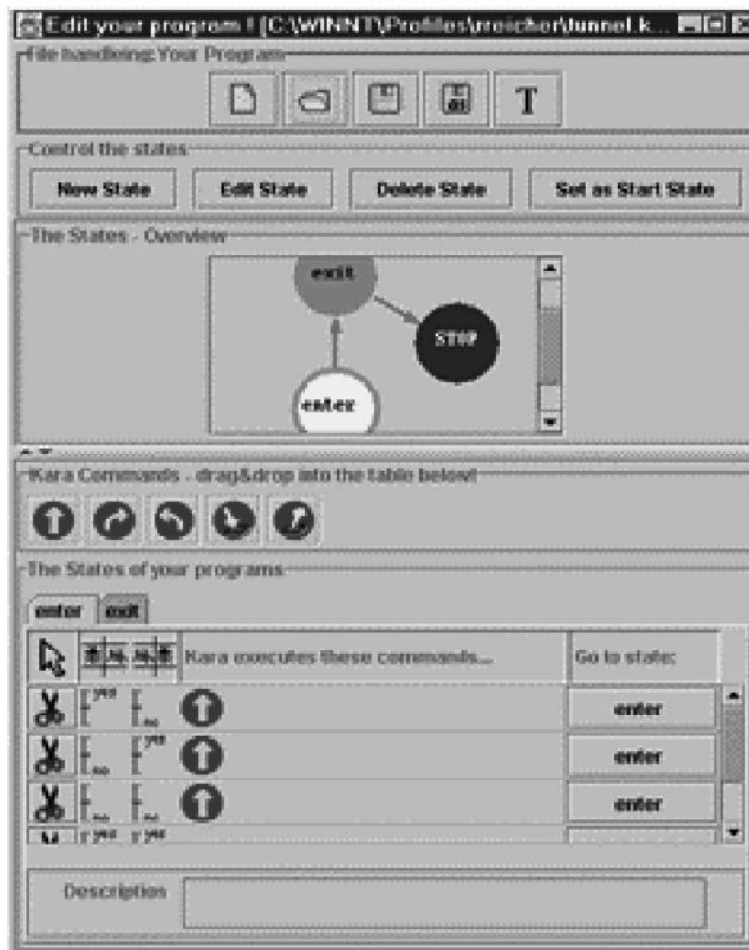
Είναι τα μοντέλα που εξετάζουν νέες μεθόδους οργάνωσης του κώδικα. Ακολουθούν ορισμένα τέτοια μοντέλα.[28]

Pascal, Institut Fur Computer system, 1970: Η πρώτη έκδοση της είχε ως επίκεντρο τη διδασκαλία προγραμματισμού, κυρίως δε για προγραμματισμό συστημάτων. Ήταν η μόνη γλώσσα που υποστήριζε τις νεοεισαχθέντες στην επιστήμη έννοιες του δομημένου προγραμματισμού από τον Dijkstra. Διδάσκεται από το 1971, ενώ οι προσθήκες που έγιναν κατά καιρούς την βελτίωσαν ως γλώσσα γενικού προγραμματισμού, επιτρέποντας την αποσυσχέτισή της από την ταμπέλα του διδακτικού εργαλείου.

Smalltalk, Xerox PARC, 1971: Δημιουργήθηκε ως γλώσσα για το KiddyKomputer, ένα φορητό υπολογιστή που απευθυνόταν σε παιδιά. Η ομάδα στο Xerox PARC ήθελε να δημιουργήσει μια εφαρμογή με απλή μέθοδο εκτέλεσης και μια μέθοδο προγραμματισμού που θα μπορούσε να ικανοποιεί διαφορετικά προγραμματιστικά στυλ. Το Smalltalk βασίστηκε σε τρεις κεντρικές ιδέες: 1) κάθε στοιχείο της γλώσσας είναι ένα αντικείμενο, 2) τα αντικείμενα έχουν μνήμη με τη μορφή άλλων αντικειμένων, 3) τα αντικείμενα μπορούν να επικοινωνούν μεταξύ τους μέσω μηνυμάτων

Playground, Apple Computer, 1989: Είναι μια αντικειμενοστραφής γλώσσα που δίνει τη δυνατότητα στους μαθητές να δημιουργούν γραφικά αντικείμενα και να τους προσδίδουν συμπεριφορά. Η κεντρική ιδέα ήταν πως κάθε αντικείμενο αποτελεί «ζωντανό οργανισμό» με δική του συμπεριφορά. Κάθε αντικείμενο διαθέτει τους δικούς του αισθητήρες, συσκευές επίδρασης και στοιχεία επεξεργασίας ώστε να είναι αυτόνομο. Ο προγραμματισμός στο Playground βασίζεται σε κανόνες οι οποίοι περιγράφουν τι πρέπει να συμβεί αλλά και κάτω από ποιες συνθήκες. Η γλώσσα του είναι σαφώς επηρεασμένη από τη φυσική γλώσσα.

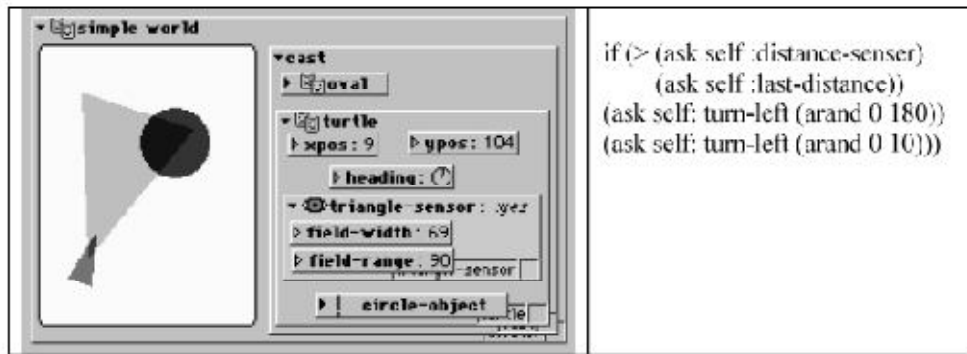
Kara, ETH Zurich, 2001: Είναι η γλώσσα προγραμματισμού του διάσημου Karel the robot. Το απλό πρότυπο περασμένων καταστάσεων που χρησιμοποιεί δίνει τη δυνατότητα στο χρήστη να βλέπει εξολοκλήρου τη δομή του προγράμματος σε οποιοδήποτε βήμα καθώς και να αλληλεπιδρά με την παρούσα αλλά και με τις επόμενες καταστάσεις. Το Kara μπορεί να κινείται όπως και να σηκώνει και να τοποθετεί τριφύλλια, με τις παραπάνω εντολές να είναι διαθέσιμες προς υλοποίηση με γραφικό τρόπο. Σε κάθε κατάσταση, ο χρήστης μπορεί να ρωτήσει την ακριβή θέση του Kara και στη συνέχεια να του δώσει μια σειρά εντολών και το όνομα της επόμενης κατάστασης της μηχανής. Το πεπερασμένο διάγραμμα μηχανής παρέχεται ώστε να εμφανίζεται η δομή του προγράμματος και να επιτρέπει στον χρήστη να αλλάξει οποιαδήποτε κατάσταση προϋπάρχει. Ακόμη, με αυτό το διάγραμμα ο προγραμματισμός μπορεί να έχει αποκλειστικά γραφική μορφή. Για μετάβαση σε παραδοσιακά προγραμματιστικά εργαλεία είναι διαθέσιμα τα Javakara, Mutikara και Turingkara, για εξοικείωση με Java, ταυτόχρονο προγραμματισμό και μηχανές Turing αντίστοιχα.



Καθιστώντας τα νέα μοντέλα προσβάσιμα

Ορισμένα προγραμματιστικά στυλ, όπως π.χ. ο αντικειμενοστραφής προγραμματισμός, είναι αρκετά δύσκολα για τους αρχάριους αλλά ταυτόχρονα ιδιαίτερα αποτελεσματικά για την οργάνωση της δομής ενός προγράμματος. Τα συστήματα που ανήκουν σε αυτή την κατηγορία προσπαθούν να κάνουν προσβάσιμες τις προγραμματιστικές τεχνικές αυτές σε μια ευρύτερη ομάδα ανθρώπων[28].

Liveworld, MIT Lab, 1994: Αναπτύχθηκε ως ένα αντικειμενοστραφές σύστημα με σκοπό να βελτιώσει το Playground. Το Liveworld επιχειρεί να αναπαραστήσει γραφικά όχι μόνο τα αντικείμενα, αλλά και τους κανόνες και τα χαρακτηριστικά που το διέπουν. Το interface του Liveworld είναι παρόμοιο με έναν ιεραρχικό browser: μέρη αντικειμένων μπορούν να ανοιχθούν, αποκαλύπτοντας λεπτομέρειες για αυτά. Ο χρήστης μπορεί να κάνει παρεμβάσεις και σε χαμηλότερο προγραμματιστικό επίπεδο αλλάζοντας των κώδικα Lisp που ελέγχει την συμπεριφορά των αντικειμένων ή απλά μπορεί να χρησιμοποιήσει τα αντικείμενα με βάση το μέγεθος των λεπτομερειών που επιθυμεί να βλέπει ο μαθητής. Έτσι γίνεται εύκολη η χρησιμοποίηση «μαύρων κουτιών» και η υλοποίηση πιο σύνθετων προγραμμάτων, στοιχεία που δεν διέθετε το Playground.



Blue environment and Bluej, University of Sydney, 1996: Είναι αντικειμενοστραφή συστήματα που υποστηρίζουν την δημιουργία κλάσεων ως αυτόνομα, εκτελέσιμα κομμάτια κώδικα σε Blue και Java αντίστοιχα. Οι δημιουργοί τους πίστευαν πως τα ενσωματωμένα περιβάλλοντα ανάπτυξης (IDE's) για αντικειμενοστραφείς γλώσσες θα έπρεπε να ενθαρρύνουν τους χρήστες να αναπτύσσουν και να δοκιμάζουν ανεξάρτητες κλάσεις παρά ολοκληρωμένα προγράμματα, όπως απαιτούν τα περισσότερα προγραμματιστικά περιβάλλοντα τέτοιου είδους. Έτσι τα Blue και Bluej παρέχουν στους χρήστες δυνατότητες δοκιμής των κλάσεων που δημιουργούν, κλήσης των μεθόδων και επιθεώρησης των εσωτερικών δεδομένων τους, μέσω των οποίων οι χρήστες μπορούν να δοκιμάζουν ξεχωριστά προγράμματα, εκτός του πλαισίου εκτέλεσης του προγράμματος που «τρέχει», υποστηρίζοντας καλύτερα έναν αντικειμενοστραφή σχεδιασμό. Είναι ακόμη διαθέσιμος ο προγραμματισμός με αναπαράσταση των αντικειμένων σε ένα δέντρο.

Karel++, Page University, 1997: Ενώ τα υπόλοιπα συστήματα για το Karel the robot επικεντρώνονται εκπαιδευτικά στην προετοιμασία των μαθητών για την κατανόηση του αντικειμενοστραφούς προγραμματισμού, η Karel++ εξοικειώνει τους μαθητές με τις διαδικασίες και τη δομή του προγράμματος. Για παράδειγμα, δεν δημιουργείται μια διαδικασία που στρέφει το Karel δεξιά αλλά ένα αντικείμενο δεξιόστροφα κινούμενου ως υποκλάση ενός ρομπότ. Η σύνταξη των προγραμμάτων είναι παρόμοια με αυτή της C++. Η Karel++ είναι ένα ακόμη σύστημα που εκμεταλλεύομενο την επιτυχία του Karel the robot, προσπαθεί να μυήσει αρχάριους στον αντικειμενοστραφή προγραμματισμό.

3.3.3 Κατανοώντας την εκτέλεση ενός προγράμματος

Συχνά ένα συντακτικά ορθό πρόγραμμα δεν παράγει το αποτέλεσμα που είχε ο αρχάριος χρήστης στο μυαλό του όταν το συνέτασσε. Τα συστήματα σε αυτή την κατηγορία προσπαθούν να δώσουν στους χρήστες τη δυνατότητα να καταλάβουν τι συμβαίνει κατά την εκτέλεση του προγράμματος.

Ένα τέτοιο παράδειγμα συστήματος είναι η ATARI 2600 BASIC (Robinett, 1979). Με αυτήν τα παιδιά μάθαιναν να γράφουν μικρά προγράμματα σε μια παραλλαγή της Basic και μπορούσαν να παρακολουθούν ζωντανά την διαδικασία εκτέλεσής τους. Η ATARI χώριζε την οθόνη σε 6 τμήματα, έτσι είχαμε τις περιοχές Προγράμματος, που εμφάνιζαν τον κώδικα, Στοιβάς, που εμφάνιζαν τις εντολές καθώς αυτές έπαιρναν τιμές, Μεταβλητών, που εμφάνιζαν τις μεταβλητές και τις τιμές τους, Εξόδου, που εμφάνιζαν όλα τα αποτελέσματα του προγράμματος, Γραφικών, μια δυσδιάστατη περιοχή με sprites και Κατάσταση, περιοχές που επεδείκνυαν την ταχύτητα εκτέλεσης του μεταγλωττιστή καθώς και την διαθέσιμη μνήμη.

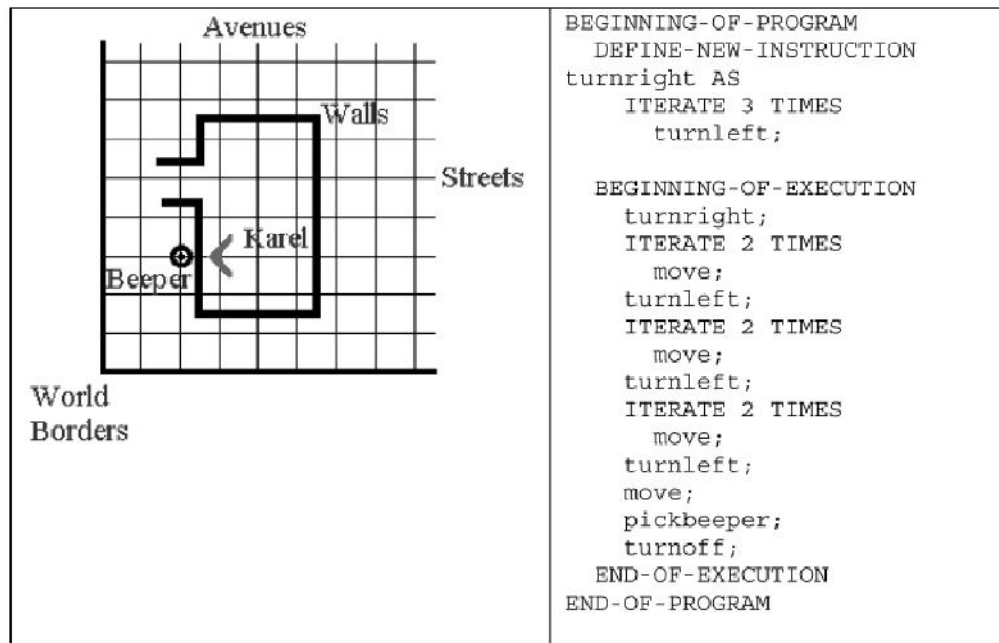
Έτσι, σε κάθε βήμα εκτέλεσης οι παραπάνω περιοχές διαμορφωνόταν ανάλογα με το πρόγραμμα ώστε να κάνουν κατανοητή στον μαθητή τη διαδικασία εκτέλεσης.[28]



Ισχυροποίηση του προγραμματισμού

Τα περισσότερα συστήματα που εισάγουν τον μαθητή σε γλώσσες γενικού περιεχομένου είναι αρκετά αόριστα. Αν επιθυμούμε να πετύχουμε ένα μεγαλύτερο βαθμό κατανόησης πρέπει να προσομοιώσουμε με ένα φυσικό τρόπο τις έννοιες που δύσκολα γίνονται κατανοητές. Δίνοντας ισχυρές θεωρητικές βάσεις στον μαθητή στο εισαγωγικό στάδιο, θωρακίζουμε τη γνώση του. Τέτοιου είδους προσπάθειες είναι το Karel , το Josef the Robot και το Turingal.[28]

Karel, Carnegie Mellon University, 1981: Η Karel είναι ίσως η πιο διαδεδομένη μικρο-γλώσσα, σχεδιασμένη για να χρησιμοποιείται εισαγωγικά στη διδασκαλία του προγραμματισμού. Βασίζεται σε ένα ρομπότ που κινείται πάνω σ' ένα πλέγμα αποτελούμενο από δρόμους και λεωφόρους που κατευθύνονται από ανατολικά προς δυτικά και από βόρεια προς νότια αντίστοιχα. Ο κόσμος της Karel διαθέτει επίσης αμετακίνητους τοίχους, και βομβητές. Το ίδιο το ρομπότ μπορεί να κινείται, να στρίβει, να αντιλαμβάνεται την παρουσία τοίχων στο ίδιο τετράγωνο με αυτό καθώς και την παρουσία βομβητών στην ίδια γωνία. Η προσομοίωση της Karel επιτρέπει στους μαθητές να παρακολουθούν την πρόοδο των προγραμμάτων τους βήμα προς βήμα. Ακόμη ένα θετικό στοιχείο για την Karel είναι η ύπαρξη συνοδευτικού εγχειριδίου. Οι μαθητές μπορούν να δημιουργούν διαδικασίες, όχι όμως μεταβλητές και δομές δεδομένων που δεν υποστηρίζονται από τη γλώσσα. Η σύνταξη της γλώσσας είναι παρόμοια με αυτή της Pascal, ώστε μετά από μερικές εβδομάδες εκπαίδευσης να γίνεται εύκολα η μετάβαση σε αυτήν.



Josef the Robot, Acadia University, 1983: Όπως η Karel, έτσι και η Josef the Robot αποσκοπεί στο να εισάγει αρχάριους στον προγραμματισμό με τη βοήθεια ενός ρομπότ, εν προκειμένω του Josef, σ' ένα περιβάλλον προσομοίωσης. Ο Josef «ζει» σ' ένα περιβάλλον, το Wolfville, το οποίο αναπαρίσταται με την βοήθεια ενός ASCII χάρτη (οι χρήστες μπορούν να τον αλλάξουν αν επιθυμούν). Σε αυτό το περιβάλλον κινείται, ενώ ακούει και διαβάζει σύνολα χαρακτήρων (strings) επιτρέποντας την είσοδο-έξοδο προγράμματος. Ολόκληρο το σύστημα προορίζεται για διδασκαλία σε ένα ολόκληρο εξάμηνο, έτσι υποστηρίζει περισσότερες δομές σε σχέση με την Karel.

Πρότυπα εκτέλεσης προγραμμάτων

Τα συστήματα σε αυτή την κατηγορία χρησιμοποιούν ένα φυσικό περιβάλλον βοηθώντας έτσι τους μαθητές να παραλληλίσουν μια διαδικασία που τους είναι ήδη γνωστή με κάποια αντίστοιχη ενός προγράμματος κι έτσι να αντιληφθούν ευκολότερα τα εννοιολογικά λάθη ενός προγράμματος. Τέτοια συστήματα είναι :

ToonTalk, Animated Programs, 1996[29]: Χρησιμοποιεί μια φυσική μεταφορά για την εκτέλεση ενός προγράμματος. Στο ToonTalk, πόλεις, πλάσματα και αντικείμενα παριστάνουν προγράμματα. Το μεγαλύτερο μέρος της υπολογιστικής διαδικασίας λαμβάνει μέρος μέσα στα σπίτια όπου ζουν εκπαιδευσιμα ρομπότ. Τα ρομπότ μπορούν να επικοινωνούν από σπίτι σε σπίτι με την βοήθεια πουλιών που μεταφέρουν μηνύματα και αντικείμενα. Χρησιμοποιώντας τεχνικές που μοιάζουν με αυτές των βιντεοπαιχνιδιών, ο χρήστης μπορεί να πλοηγηθεί στην πόλη, να διαλέξει εργαλεία και να τα χρησιμοποιήσει για να αλληλεπιδράσει με αντικείμενα. Τα προγράμματα συντάσσονται προσθέτοντας στα ρομπότ μικρά σύννεφα που περιγράφουν τι πρέπει ακριβώς να κάνουν.



Prototype 2, Victoria University, 1998[28]: Αυτό το σύστημα προσωποποιεί την ροή ελέγχου σε έναν υπολογιστή με τη βοήθεια ενός υπαλλήλου που ακολουθεί εντολές. Ο υπάλληλος μπορεί να αλληλεπιδρά με αριθμομηχανές, συσκευές εισόδου-εξόδου, φύλλα εργασίας και το δικό του πρόχειρο κατά την εκτέλεση ενός προγράμματος. Οι αριθμομηχανές αναπαριστούν τον μαθηματικό επεξεργαστή του υπολογιστή, το πρόχειρο την στοίβα του, οι συσκευές εισόδου-εξόδου την επικοινωνία και τα φύλλα εργασίας τις εντολές σε υπορουτίνες. Έτσι, ο μαθητής κατανοεί την λειτουργία του υπολογιστή πολύ πιο εύκολα, έχοντας στο μυαλό του την γραφική αναπαράσταση του υπαλλήλου.

3.4 Υποστήριξη μάθησης

Τα συστήματα στην προηγούμενη κατηγορία απλοποιούσαν τους μηχανισμούς με τους οποίους ο χρήστης εισάγει ένα πρόγραμμα. Στην παρούσα κατηγορία όμως τα συστήματα επικεντρώνονται στην προσπάθεια ενίσχυσης της ίδιας της διαδικασίας της μάθησης. Δύο τέτοιες μέθοδοι είναι η συνεργατική (κοινωνική) μάθηση και η διαμόρφωση του περιεχομένου έτσι ώστε να προσδίδει επιπλέον κίνητρο στους μαθητές.

3.4.1 Συνεργατική μάθηση

Όταν οι εκπαιδευόμενοι συνεργάζονται, τότε αποδεδειγμένα η διαδικασία της μάθησης γίνεται ευκολότερη και αποτελεσματικότερη. Στην περίπτωση μας, μιας και είναι δεδομένο πως το αντικείμενο έχει μεγάλη δυσκολία, υπάρχουν συστήματα που μελετούν διάφορους τρόπους έτσι ώστε οι μαθητές να εργάζονται μαζί πάνω σε ένα πρόβλημα.

Η πρώτη μέθοδος ονομάζεται *side by side learning* (μάθηση δίπλα-δίπλα σε μια κάπως ανορθόδοξη μετάφραση) και τα συστήματα της έχουν τέτοιες διεπαφές που επιτρέπουν στους μαθητές να δουλεύουν μαζί, κάτι που ένα παραδοσιακό υπολογιστικό περιβάλλον αποκλείει. Δύο τέτοια παραδείγματα συστημάτων είναι τα:[28]

Algoblock, Nec Information Technology Research laboratories, 1995: Οι συγγραφείς του Algoblock θέλησαν να δημιουργήσουν μια ενεργή διδακτική κοινότητα προγραμματισμό, στην οποία τα παιδιά καθώς μάθαιναν θα μπορούσαν να μοιράζονται σημειώσεις και

τεχνικές, καθώς και να μαθαίνουν το ένα απ' το άλλο. Δημιούργησαν αυτό το σύστημα, μια συλλογή από κουτιά, όπου το κάθε ένα από αυτά αναπαριστά μια εντολή της Logo. Τα κουτιά αυτά συνδέονται για να δημιουργήσουν ένα σύνολο κινήσεων για ένα υποβρύχιο μέσα σε ένα λαβύρινθο. Τα κουτιά αυτά είναι απτά και αρκετά μεγάλα ώστε να μπορούν να τοποθετηθούν πάνω σε ένα γραφείο γύρω από το οποίο θα εργάζονται οι μαθητές. Έτσι οι μαθητές μαθαίνουν συνεργαζόμενοι, ενώ η απτή φύση των κουτιών διευκολύνει την κατανόηση των δύσκολων εννοιών.

Tangible, Programming Bricks MIT Media Lab, 2000: Αποτελείται από απτά Lego τουβλάκια που μπορούν να τοποθετηθούν μαζί ώστε να σχηματίσουν προγράμματα. Η πρόθεση των σχεδιαστών ήταν να δημιουργήσουν ένα απλό σύστημα που θα επέτρεπε στους μαθητές να κατανοούν έννοιες μέσω της συνεργασίας. Για την εκπλήρωση αυτού του στόχου, δημιουργήθηκαν τρία πρωτότυπα περιβάλλοντα, χρησιμοποιώντας τουβλάκια που αναπαριστούσαν εντολές. Για μεγαλύτερη γκάμα εντολών ο χρήστης είχε τη δυνατότητα να τοποθετήσει ένα μικροτσιπ σε κάθε τούβλο. Αυτή η προσθήκη επέτρεπε στα τούβλα να επικοινωνούν μεταξύ τους μέσω υπερύθρων, να περνούν παραμέτρους σε σεντ εντολών, να λαμβάνουν ερεθίσματα από το περιβάλλον μέσω υπερύθρων και να απεικονίζουν μεταβλητές. Τα τρία περιβάλλοντα που αναφέρθηκαν πιο πάνω διευκόλυναν τους μαθητές στο να προγραμματίζουν μια σειρά από παιχνίδια (π.χ. τρενάκια που αντιδρούν σε σήματα στις ράγες)

Η δεύτερη μέθοδος αποκαλείται δικτυακή διαδραστικότητα (networked interaction) και επιτρέπει τη συνεργασία των μαθητών όχι δίπλα-δίπλα, αλλά μέσω ενός δικτύου υπολογιστικών συστημάτων. Τέτοια συστήματα είναι τα:

Moose Crossing, MIT Media Lab, 1997: Είναι ένα δικτυακό, προγραμματιστικό περιβάλλον σχεδιασμένο για παιδιά στο οποίο τα παιδιά μπορούν να χρησιμοποιήσουν μια αντικειμενοστραφή γλώσσα για να δημιουργήσουν χώρους και χαρακτήρες που κατοικούν σε έναν κόσμο, μέσω μιας εφαρμογής κειμένου. Αυτές οι δημιουργίες των μαθητών είναι παρόμοιες των στοιχείων των text adventure παιχνιδιών, π.χ. κάστρα με μυστικά περάσματα που οι μαθητές πρέπει να εξερευνήσουν. Όταν τελειώσουν αυτές οι εργασίες οι μαθητές μπορούν να παίξουν μαζί τους αλλά και να συνομιλήσουν με άλλους μαθητές που βρίσκονται στο σύστημα. Συνήθως κάθε μαθητής δουλεύει μόνος του, αλλά μπορεί να χρησιμοποιήσει την εργασία ενός άλλου για παράδειγμα ή να του ζητήσει βοήθεια και συμβουλές.

```
on pet this
  tell player "You pet Rover."
  if player member_of my friends
    emote "wags his tail."
end
```

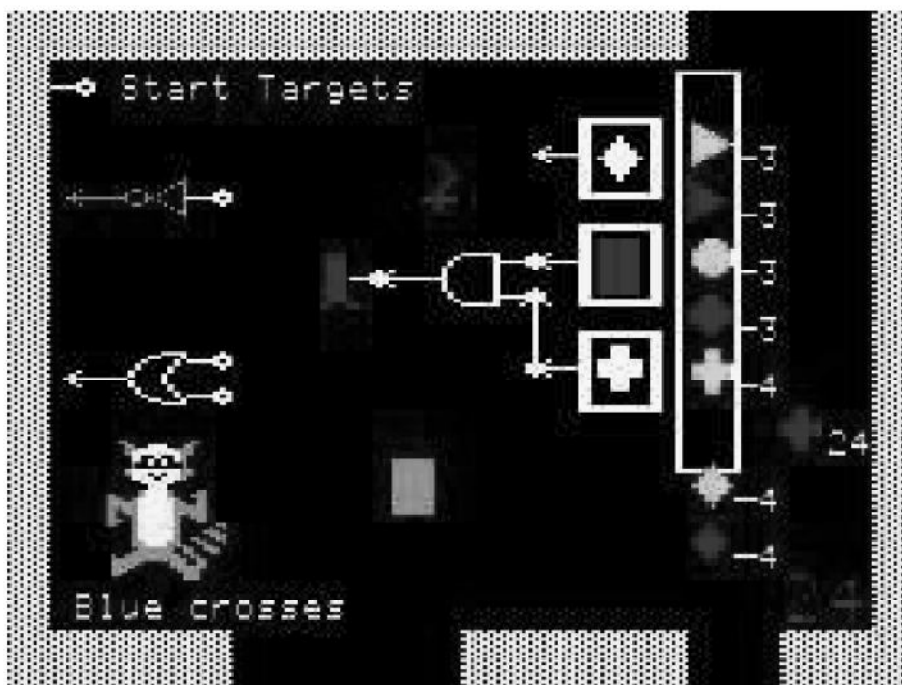
Pet Park, MIT Media Lab, 1998: Είναι μια 2D μεταφορά των ιδεών του προηγούμενου συστήματος. Οι μαθητές μπορούν να διαλέξουν ένα από 5 διαθέσιμα σκυλιά ως κατοικίδιο τους και να φτιάξουν animations δίνοντας σε αυτό βασικές εντολές όπως το κούνημα της ουράς και διάφορες βασικές κινήσεις και συνθέτοντας τες ανάλογα. Για την εισαγωγή του κώδικα ο μαθητής μπορεί να επιλέξει ανάμεσα σε έναν text editor και ένα γραφικό περιβάλλον. Διατηρούνται όλες οι συνεργατικές ιδιότητες της Moose που αναφέρθηκαν πριν.

Cleogo, University Of Canterbury, 1998: Είναι μια δικτυακή έκδοση της Leogo που επιτρέπει στους μαθητές να δουλέψουν συνεργατικά με το περιβάλλον της. Αντί να δημιουργηθεί μια κοινότητα προγραμματιστών, στην Cleogo κατασκευάζεται ένα κοινό περιβάλλον το οποίο μοιράζονται οι μαθητές. Δεν παρέχονται μέθοδοι επικοινωνίας μεταξύ τους, με την προϋπόθεση ότι οι μαθητές θα βρίσκονται στην ίδια αίθουσα ή ότι η επικοινωνία μπορεί να γίνει με εναλλακτικά μέσα.

3.4.2 Παροχή περιεχομένου με επιπλέον κίνητρο

Το κίνητρο για μάθηση είναι ένα στοιχείο κλειδί γενικότερα στην εκπαίδευση. Αυτά τα συστήματα προσπαθούν να εστιάσουν την εκμάθηση του προγραμματισμού σε προβλήματα η λύση των οποίων μπορεί να θεωρηθεί ενδιαφέρουσα για τους διδασκόμενους. Τέτοια συστήματα είναι τα:[28]

Rocky's Boots/Robot Odyssey, The Learning Company, 1982: Το Rocky's Boots αποτέλεσε ένα από τα πρώτα εκπαιδευτικά λογισμικά για προσωπικού υπολογιστές που χρησιμοποίησε μια διαδραστική γραφική προσομοίωση. Δίνει στα παιδιά τη δυνατότητα να δημιουργήσουν κυκλώματα συνδέοντας με τη βοήθεια ενός joystick λογικές πύλες μεταξύ τους. Όταν τα συστήματα ήταν ενεργά, οι μαθητές παρακολουθούσαν τα καλώδια να παίρνουν πορτοκαλί χρώμα, καθώς τα διαπερνούσε ηλεκτρισμός. Διέθετε ένα σύνολο γρίφων με αυξανόμενη δυσκολία. Από την άλλη, το Robot Odyssey, ενώ ακολουθούσε την ίδια φιλοσοφία, διέθετε μεγαλύτερο εύρος αντικειμένων προς χρήση στα κυκλώματα και περισσότερα animations.



Algoarena, Nec Information Technology Research laboratories, 1995: Στο περιβάλλον αυτό, οι μαθητές συντάσσουν προγράμματα ώστε να χειριστούν αθλητές του sumo σε ένα τουρνουά. Η γλώσσα θυμίζει τη Logo, ενώ με το πέρας της σύνταξης του προγράμματος ο κώδικας ανεβαίνει σε μια ιστοσελίδα όπου οι αθλητές των μαθητών μπορούν να διαγωνιστούν. Μέσα από τα παιχνίδια, ο μαθητής σκέφτεται τρόπους βελτίωσης των κινήσεων του αθλητή, κάνει δηλαδή πρώιμες προγραμματιστικές σκέψεις.

Robocode, IBM Advanced Technology, 2001: Σχεδιάστηκε με σκοπό να βοηθήσει τους αρχάριους να διδαχθούν Java προγραμματίζοντας ένα ρομπότ-τανκ και επεκτείνοντας τις δυνατότητές του. Χρησιμοποιείται μια ειδική έκδοση της Java, εμπλουτισμένη με νέες υποκλάσεις, ενώ ο κώδικας μπορεί να χρησιμοποιηθεί για τη διενέργεια τουρνουά σε μια ιστοσελίδα (όπως στο προηγούμενο λογισμικό). Αν μη τι άλλο το κίνητρο για τους αρχάριους προγραμματιστές είναι ισχυρό και ψυχαγωγικό.

3.5 Συστήματα ενδυνάμωσης

Τα συστήματα σε αυτή την κατηγορία έχουν κατασκευαστεί με την πεποίθηση πως ο προγραμματισμός οφείλει να βοηθά τους ανθρώπους να κατασκευάζουν λογισμικό που εξυπηρετεί αλλά και ακολουθεί τις ανάγκες τους. Συνεπώς, οι κατασκευαστές τέτοιων συστημάτων δεν ενδιαφέρονται κατά πόσο θα είναι εύκολη η μετάβαση από την γλώσσα που κατασκεύασαν σε μια παραδοσιακή γλώσσα προγραμματισμού, αλλά ενδιαφέρονται αποκλειστικά να δημιουργήσουν ένα πρόγραμμα που κατασκευάζει εφαρμογές εύκολα και γρήγορα.

3.5.1 Ο κώδικας είναι πολύ δύσκολος

Παρά τις προσπάθειες να κατασκευαστούν συστήματα ακόμη πιο εύκολα για αρχάριους, η δυσκολία της κατανόησης του κώδικα από αυτούς παραμένει πολλές φορές μεγάλη. Έτσι εκδηλώθηκαν νέες προσπάθειες, οι οποίες προσανατολίστηκαν στο να δίνουν στο χρήστη τη δυνατότητα να προσδιορίζει τις ενέργειες ενός προγράμματος είτε μέσω επίδειξης μιας σωστής προγραμματιστικής συμπεριφοράς και κατόπιν υιοθέτησης της σε ένα πρόγραμμα, είτε μέσω επιλογής μιας ενέργειας από ένα σύνολο προκαθορισμένων,

Επίδειξη ενεργειών στη διεπαφή

Είναι συστήματα στα οποία ο χρήστης επιδεικνύει στο πρόγραμμα ποια θα πρέπει να είναι η συμπεριφορά του, με τη βοήθεια μιας διεπαφής, χωρίς να καθίσταται αναγκαία η χρήση κώδικα. Ένα τέτοιο σύστημα είναι το παρακάτω. [28]

XPygmalion, Stanford University, 1975: Ήταν το πρώτο σύστημα προγραμματισμού μέσω επίδειξης, μόνο που δεν βασίστηκε σε γραφικά αντικείμενα, αλλά προσπάθησε να κάνει τους μαθητές να γράψουν αόριστα προγράμματα,, το γνωστότερο εκ των οποίων ήταν ένα πρόγραμμα υπολογισμού του παραγοντικού. Οι μαθητές δεν ήταν αναγκασμένοι να συντάξουν το πρόγραμμα από την αρχή γιατί προσφερόταν πρότυπα προγράμματα πάνω στα οποία μπορούσαν να βασιστούν και να εμπνευστούν. Εξετάζοντας το παράδειγμα του παραγοντικού, για να το λύσει ένας μαθητής έπρεπε να χρησιμοποιήσει ένα εικονίδιο και δύο υποεικονίδια, ένα για την είσοδο και ένα για την έξοδο, καθώς και ένα σύμβολο για να αναπαραστήσει το παραγοντικό. Στη συνέχεια ο μαθητής όφειλε να χρησιμοποιήσει την επιλογή απομνημόνευσης ώστε να αποθηκεύονται αναδρομικά και να εκτελούνται ορθώς οι υπολογισμοί.

Προγραμματισμός με δοκιμαστική εκτέλεση

Είναι μια υποκατηγορία της επίδειξης ενεργειών στην διεπαφή, σύμφωνα με την οποία ο αρχάριος διευκολύνεται στην κατανόηση του κώδικα δοκιμάζοντας την εκτέλεση των προγραμμάτων του και γνωρίζοντας εκ των προτέρων την συμπεριφορά του και τα πιθανά αποτελέσματα. Τέτοια συστήματα είναι τα:[28]

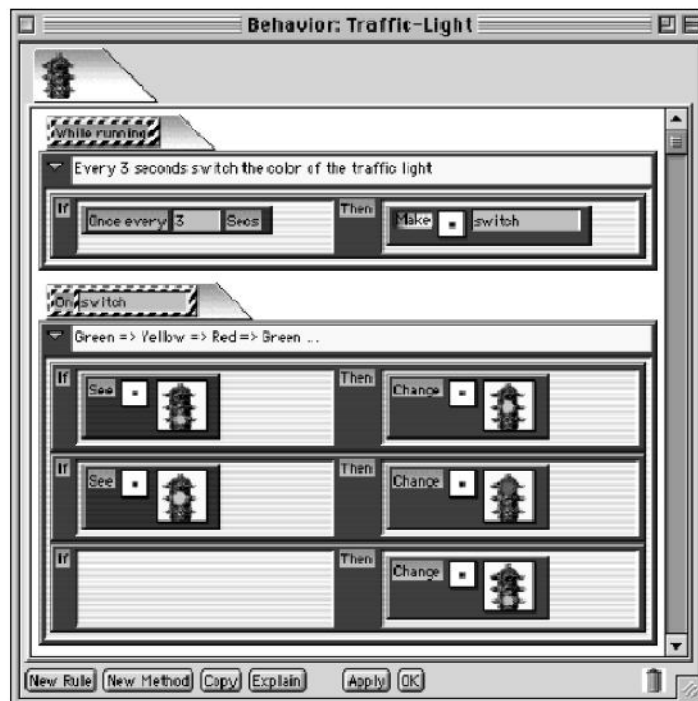
XEROX, Parc, 1984: Το περιβάλλον είναι σχεδιασμένο ως μια φυσική μεταφορά ενός θεάτρου και τη μετατροπή του σε προγραμματιστικό εργαλείο. Σε αυτό οι ηθοποιοί (προγραμματιστικές μονάδες) αλληλεπιδρούν ανταλλάσσοντας μηνύματα (πληροφορίες ή παράμετροι). Ένας χρήστης του συστήματος ξεκινά να δημιουργεί ένα τμήμα εκπαιδευτικού λογισμικού κάνοντας ένα είδος «οντισιόν» στους ηθοποιούς (επιλογή αντικειμένων που θα χρησιμοποιήσει), επιλέγοντας τις ατάκες τους (ενέργειες) από αναδυόμενα μενού και παρατηρώντας τις αντιδράσεις τους σε αυτές (αποτελέσματα δοκιμαστικής εκτέλεσης). Μετά την επιλογή, ο χρήστης αντιγράφει και τοποθετεί τους ηθοποιούς του στη σκηνή του περιβάλλοντος και τους εκπαιδεύει για τους τρόπους αντίδρασής τους στην είσοδο των χρηστών ή στις ενέργειες άλλων ηθοποιών (δηλαδή σύνταξη τελικού προγράμματος). Με το τέλος και αυτής της διαδικασίας ο χρήστης μπορεί να εκκινήσει την κατάσταση επιτήρησης, στην οποία βλέπει τους ηθοποιούς να αλληλεπιδρούν σε ήδη γνωστά ερεθίσματά αλλά και σε νέα που εισάγει ο ίδιος οποιαδήποτε στιγμή το επιθυμεί.

Mondrian, MIT, 1982: Είναι ένα σύστημα προγραμματισμού μέσω δοκιμαστικής εκτέλεσης για ζωγραφική αλλά και επεξεργασία γραφικών, στο οποίο οι εντολές απεικονίζονται ως εικονίδια ντόμινο που αναπαριστούν τις πρωθύστερες και μεταγενέστερες καταστάσεις τους. Για την εκτέλεση μιας εντολής, οι χρήστες επιλέγουν το αντίστοιχο εικονίδιο και το αντικείμενο ή την περιοχή που θα τοποθετηθεί, ενώ μπορούν να δημιουργήσουν και νέες εντολές. Τα βήματα παρουσιάζονται στο κάτω μέρος της οθόνης με τη μορφή κόμικς και μια μικρή ετικέτα για κάθε εντολή. Έτσι αν για παράδειγμα ο χρήστης επιθυμεί να ζωγραφίσει ένα πολύγωνο στην οθόνη τότε θα πρέπει να επιλέξει το αντίστοιχο κουτί εντολής «rectangle». Στη συνέχεια αν μετακινήσει αυτό το πολύγωνο τότε στο κουτί «rectangle» που ήδη υπάρχει, θα εμφανιστεί η εντολή «move».

Προγραμματισμός με επίδειξη συνθηκών και ενεργειών

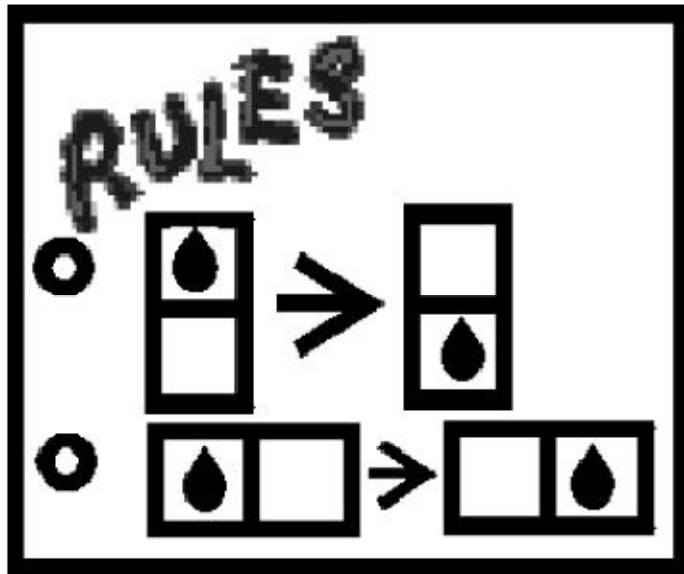
Όπως και στη κατηγορία της επίδειξης ενεργειών στην διεπαφή, έτσι κι εδώ ο κατασκευαστής αποφεύγει να αναγκάσει τον αρχάριο να συνθέσει προγράμματα με τη βοήθεια κώδικα. Σε αντίθεση όμως με την προηγούμενη κατηγορία όπου γινόταν χρήση της επίδειξης ενεργειών και των αποτελεσμάτων τους, τώρα τα συστήματα αυτά επιτρέπουν στους χρήστες να απεικονίσουν τις καταστάσεις αλλά και τους όρους σύμφωνα με τους οποίους επιθυμούν να εκτελεστεί το πρόγραμμά τους καθώς και τα επικείμενα αποτελέσματα. Μερικά από αυτά τα συστήματα είναι:

Agentsheets, University of Colorado, 1991[30]: Σε αυτό το σύστημα οι χρήστες μπορούν δημιουργούν προσομιώσεις προσδιορίζοντας την συμπεριφορά διάφορων sprites σε έναν δυσδιάστατο κόσμο. Τα σύνολα των sprites (πράκτορες-agents) μπορούν να μετακινούνται σε νέες θέσεις πάνω σε ένα βοηθητικό πλέγμα που βρίσκεται στο υπόβαθρο, να αναπαράγουν ήχους και να αλλάζουν εμφάνιση. Οι χρήστες μπορούν να δημιουργούν προγράμματα χρησιμοποιώντας γραφικούς κανόνες: επιλέγουν δηλαδή συνθήκες ελέγχου και δείχνουν στο σύστημα τι πρέπει να συμβεί όταν αυτές ισχύουν, μετακινώντας τους πράκτορες του συστήματος σε νέα θέση. Δημοφιλές είναι ακόμη το εργαλείο δημιουργίας αναλογιών μεταξύ των πρακτόρων, που δημιουργεί πανομοιότυπες κινήσεις με κάποιες που ήδη έχουν οριστεί.



ChemTrains, US West Advanced Technologies, 1993[31]: Είναι ένα εικονικό, βασισμένο σε κανόνες περιβάλλον που επιχειρεί να κάνει ευκολότερη την δημιουργία εικόνων που επιδεικνύουν συμπεριφορές. Μοιάζει με το Stagecast (ακολουθεί) γιατί και στα δύο συστήματα οι χρήστες δείχνουν της συνθήκες εκτέλεσης που πρέπει να ισχύουν κατά την εκτέλεση του προγράμματος αλλά και τα αποτελέσματα αυτών με τη βοήθεια εικόνων. Στο ChemTrains δίνεται μεγαλύτερη βάση στη διασύνδεση μεταξύ έτοιμων εικόνων παρά στη συλλογή ορισμένων pixels που τις αποτελούν. Ένα μεγάλο πλεονέκτημα του έναντι του Stagecast είναι και η χρήση μεταβλητών ως έτοιμων εικονιδίων. Για παράδειγμα, στην προσομοίωση μιας πύλης AND, αν υπάρχει ένα κουτί μεταβλητής που περιέχει το μηδέν και συνδέεται με αυτή, τότε η έξοδος αυτής της πύλης πρέπει υποχρεωτικά να είναι μηδέν.

StageCast, Apple Computer, 1995[28]: Είναι μια εμπορική έκδοση της KidSim και χρησιμεύει στην δημιουργία προσομοιώσεων. Παρουσιάζεται στα παιδιά ένα περιβάλλον ορισμένο με τη βοήθεια πλέγματος σχεδιασμού, στο οποίο μπορούν να δημιουργήσουν τις δικές τους ενέργειες. Οι χρήστες καθορίζουν κανόνες για την προσομοίωση επιλέγοντας μια αρχική κατάσταση ενός εικονιδίου από τον πλεγματοκόσμο και δείχνοντας στο σύστημα πώς αυτή η κατάσταση θα μεταβληθεί, δηλαδή που θα βρίσκεται το εικονίδιο μετά την επίδραση του κανόνα. Όταν η προσομοίωση ξεκινά, περιλαμβάνει όλες τους κανόνες που ο χρήστης έχει ορίσει. Αμέσως μετά την έναρξή της, ελέγχονται όλα τα εικονίδια του πλέγματος, και σε περίπτωση που ικανοποιούνται οι συνθήκες κάποιου από τους κανόνες που ο χρήστης έχει ορίσει, τότε αυτός εφαρμόζεται.

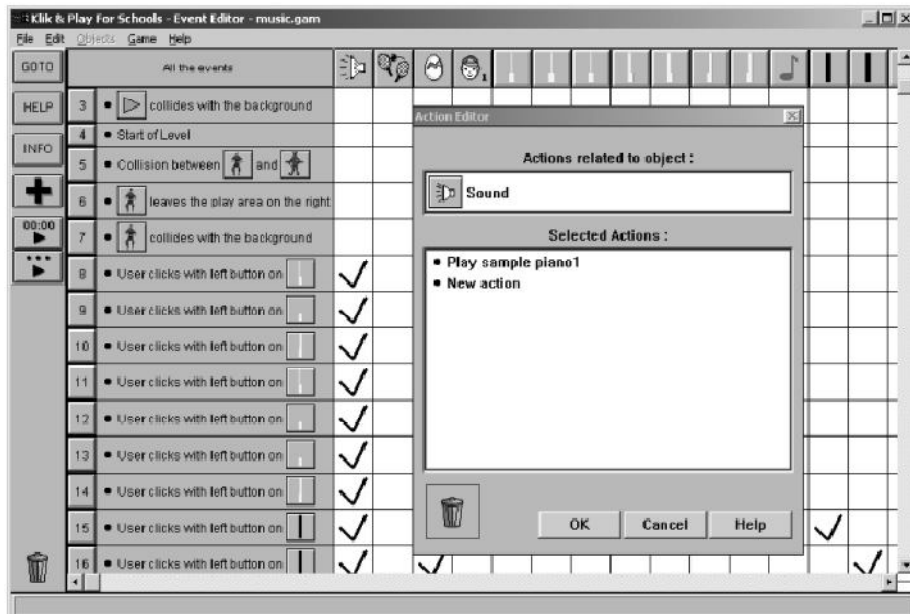


Προσδιορίζοντας τις ενέργειες

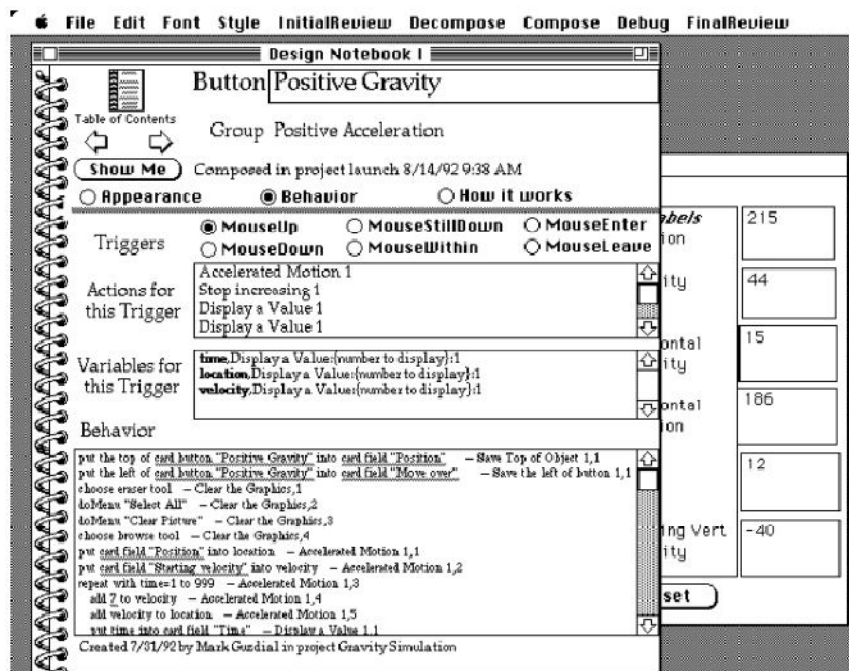
Σε αυτή την κατηγορία ο χρήστης δημιουργεί προγράμματα χρησιμοποιώντας τη διεπαφή ώστε να προσδιορίσει την επιθυμητή συμπεριφορά. Ο κώδικας δεν είναι ορατός σε αυτόν όπως και στις 2 προηγούμενες κατηγορίες, σε αντίθεση με αυτές όμως ο χρήστης δεν υποδεικνύει στο πρόγραμμα ποια πρέπει να είναι η συμπεριφορά του, αλλά διαλέγει τις ενέργειες. Τέτοια συστήματα είναι τα [28]:

Alternate Reality Kit, Xerox PARC, 1987: Είναι ένα περιβάλλον στο οποίο οι χρήστες μπορούν να κατασκευάσουν διαδραστικές (interactive) προσομοιώσεις. Οι χρήστες αλληλεπιδρούν με αντικείμενα ενός μεταφορικού κόσμου που αποτελεί και το περιβάλλον εκτέλεσης: κάθε αντικείμενο έχει χαρακτηριστικά όπως εικόνα, θέση, ταχύτητα και μπορεί να επηρεαστεί από δυνάμεις, ενώ διατίθεται και μια σειρά από ενέργειες επί των αντικειμένων όπως ανύψωση και μετακίνηση που υλοποιούνται μέσω μηνυμάτων (με τη μορφή κουμπιών). Για τον αρμονικό συνδυασμό όλων των παραπάνω χρησιμοποιούνται διαδομένες τεχνικές όπως drag η drop και κατάλληλες κινήσεις του ποντικιού του υπολογιστή. Ένα τελικό πρόγραμμα σε αυτή την γλώσσα προσομοίωσης αποτελείται από ένα σύνολο αυτών των αντικειμένων, τις ενέργειες που ο χρήστης καθορίζει για αυτά καθώς και τις επιδράσεις από δυνάμεις του ίδιου του περιβάλλοντος.

Klik N Play, Europress, 1994: Σχεδιάστηκε έτσι ώστε να επιτρέπει την δημιουργία απλών αλλά πολυεπίπεδων παιχνιδιών. Χωρίζεται σε τρία μέρη: έναν editor συγγραφής του προγράμματος που δείχνει όλα τα επίπεδα σε προεπισκόπηση, έναν editor επιπέδων και έναν editor αντικειμένων. Ο editor επιπέδων επιτρέπει στον χρήστη να διαλέξει φόντο, να προσθέσει προκαθορισμένα αντικείμενα σε ένα επίπεδο ενώ παράλληλα είναι διαθέσιμη και η δυνατότητα δημιουργίας νέων αντικειμένων και animation για αυτά τα αντικείμενα. Ο editor αντικειμένων χρησιμοποιεί μια μορφή πινάκων και επιτρέπει στο χρήστη να προσδιορίζει δραστηριότητες για μια πληθώρα προκαθορισμένων αντικειμένων. Αργότερα κυκλοφόρησε και μια νέα έκδοση από την Corel που επέτρεπε στους χρήστες να πουλούν τα παιχνίδια τους με την υποχρεωτική ταμπέλα όμως Click and Create στην ονομασία τους.



Emile, University of Michigan, 1995: Είναι ένα προγραμματιστικό περιβάλλον γραμμένο σε Hypercard (Goodman, 1987), το οποίο επιτρέπει σε μαθητές λυκείου να δημιουργούν προσομοιώσεις φυσικής. Παρέχει υποστήριξη και επίπεδα σταδιακής μάθησης, στοιχεία που κάνουν την μάθηση ευκολότερη για αρχάριους προγραμματιστές και μπορούν να αφαιρεθούν όταν ο μαθητής εξοικειωθεί πλήρως. Με τη βοήθεια του Emile οι αρχάριοι δημιουργούν προγράμματα συγκεντρώνοντας στοιχεία όπως κουμπιά, κουτιά κειμένου και προκαθορισμένες ενέργειες. Χρησιμοποιώντας τα μενού γίνεται δυνατή και η επιλογή μιας ή περισσότερων ενεργειών που πρέπει να πραγματοποιηθούν όταν πιεστεί ένα κουμπί καθώς και οι παράμετροι που θα περαστούν. Με την πάροδο του χρόνου και την κατανόηση του περιβάλλοντος, οι μαθητές μπορούν να χρησιμοποιούν σύνθετες μαθηματικές εκφράσεις και να δημιουργήσουν νέες ενέργειες με αυτές.



3.5.2 Βελτιώνοντας τις γλώσσες προγραμματισμού

Σκοπός τις περισσότερες φορές είναι μετά τη διδασκαλία με ένα ενδιάμεσο, βοηθητικό εργαλείο είναι ο εκπαιδευόμενος να μεταβεί σε μια γενικού σκοπού και περιεχομένου γλώσσα προγραμματισμού. Ορισμένοι σχεδιαστές λογισμικού όμως, όντας παράτολμοι προσέθεσαν κάποια στοιχεία σε αυτές τις κλασσικές γλώσσες προγραμματισμού ώστε να τις κάνουν πιο ενδιαφέρουσες για τους μαθητές, ερχόμενοι πολλές φορές σε σύγκρουση με τη γνώμη των διδασκόντων. Οι βελτιώσεις αυτές έγιναν αρχικά για να κάνουν τις ίδιες τις γλώσσες προγραμματισμού πιο κατανοητές στους αρχάριους αλλά και να ενθαρρύνουν τη γνώση προσφέροντας τους ενδιαφέροντα ζητήματα με τα οποία θα έπρεπε να ασχοληθούν. Έτσι δημιουργήθηκαν γλώσσες όπως οι παρακάτω, οι οποίες μπορεί να μην έκαναν εύκολη τη μετάβαση σε κλασσικές γλώσσες προγραμματισμού, εισήγαγαν όμως πολλές ριζοσπαστικές και πρωτότυπες ιδέες για την διδασκαλία του προγραμματισμού.

Κάνοντας την γλώσσα περισσότερο κατανοητή

Αυτή η υποκατηγορία συστημάτων αναπτύχθηκε δίνοντας βάση στη γλώσσα και τις λέξεις που χρησιμοποιούσαν οι αρχάριοι για να περιγράψουν καταστάσεις. Μέχρι εκείνη τη στιγμή τα περισσότερα προγραμματιστικά εργαλεία λάμβαναν υπόψη μόνο την συνάφεια ανάμεσα σε αυτά και σε απλές μαθηματικές δομές ή σε άλλα δημοφιλή εργαλεία. Τα νέα συστήματα όμως προσπάθησαν να χρησιμοποιήσουν ως δομικά τους στοιχεία, λέξεις που θα ήταν απόλυτα κατανοητές για τους αρχάριους και δεν θα χρειαζόταν να τις αντιστοιχίσουν με άλλες, καθημερινές λέξεις που χρησιμοποιούσαν, Παραδείγματα τέτοιων προσπαθειών είναι τα παρακάτω συστήματα[28].

COBOL, Department of Defense, 1960: Τα αρχικά της σημαίνουν Common Business Oriented Language και σχεδιάστηκε για να υποστηρίξει την ανάπτυξη επιχειρηματικών εφαρμογών. Κατασκευάστηκε έτσι ώστε να είναι χρήσιμη για αρχάριους και αναγνώσιμη από managers. Τα κοινά Αγγλικά επηρέασαν πολλές προγραμματιστικές δομές εκτός από την COBOL. Οι σχεδιαστές της όμως προσθέσανε και κάποιες λέξεις «θορύβου» για να διευκολύνουν ακόμη περισσότερο την ανάγνωση της π.χ. η έκφραση ADD X,Y έγινε ADD X TO Y.

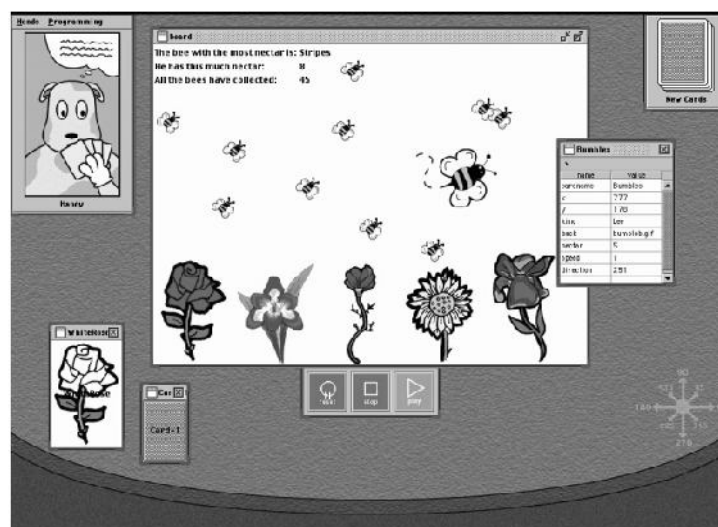
```
IF X = Y <...>
IF GREATER <...>
OTHERWISE <...>
```

Logo, MIT Papert, 1967: Είναι μια διάλεκτος της LISP στην οποία έχει αφαιρεθεί το μεγαλύτερο κομμάτι των απαραίτητων σημείων στίξης ώστε η σύνταξή της να είναι προσβάσιμη από παιδιά. Πρόθεσή της ήταν να επιτρέπει σε αυτά την εξερεύνηση ενός μεγάλου αριθμού θεματικών ενοτήτων, από μαθηματικά και επιστήμες έως μουσική. Το πιο γνωστό μέρος της Logo είναι η χελώνα, η οποία ξεκίνησε ως μια ρομποτική εφαρμογή που ζωγράφιζε στο πάτωμα αλλά αργότερα αντικαταστάθηκε από μια προσομοιωμένη έκδοση. Οι κινήσεις της χελώνας είναι αντικειμενοκεντρικές, δηλαδή αν δοθεί μια εντολή "forward 10", η χελώνα θα προχωρήσει προς την δική της κατεύθυνση κατά 10 και όχι σε αυτή που ορίζεται στην οθόνη. Από την δημιουργία της Logo μέχρι σήμερα, πάρα πολλά παιδιά εισήχθησαν στις έννοιες του προγραμματισμού με τη βοήθειά της, κάνοντας

δηλαδή την χελώνα να ζωγραφίζει απλές εικόνες, αν και διαθέτει μεγαλύτερο εύρος δυνατοτήτων. Ορισμένες τάξεις παιδιών έχουν γράψει μουσική, έχουν κατασκευάσει εργαλεία για μετάφραση από τα Αγγλικά στα Γαλλικά κ.ά.

Alice98, Carnegie Mellon University, 1998: Είναι ένα προγραμματιστικό 3D εργαλείο σχεδιασμένο να κάνει την κατασκευή διαδραστικών τρισδιάστατων γραφικών κόσμων προσβάσιμη σε φοιτητές που δεν έχουν επιλέξει την πληροφορική ως επιστήμη τους. Αποτελείται από έναν editor σκηνής με τον οποίο ο χρήστης μπορεί να δημιουργήσει την αρχική σκηνή και μια ετικέτα κειμένου στην οποία ο χρήστης μπορεί να προσδιορίσει τη συμπεριφορά του εικονικού κόσμου. Η γλώσσα πάνω στην οποία βασίστηκε είναι η Pythοn με ορισμένες αλλαγές (π.χ. δεν είναι case sensitive) ενώ οι κινήσεις των αντικειμένων είναι παρόμοιες με αυτές της Logo. Ακόμη, τα ονόματα των εντολών είναι εμπνευσμένα από τη φυσική γλώσσα π.χ. το scale γίνεται resize και το rate γίνεται speed. Η Alice προσφέρεται στον χρήστη σε διάφορα επίπεδα δυσκολίας, από το πιο απλό, ένα κουνέλι δηλαδή που δέχεται βασικές εντολές κίνησης ως τον προσδιορισμό σύνθετων χαρακτηριστικών της συμπεριφοράς του (π.χ. ταχύτητα κίνησης και κίνηση υπό συνθήκη) αλλά και την κατασκευή νέου εικονικού περιβάλλοντος.

Hands, Carnegie Mellon University, 2001: Αυτό το σύστημα επιτρέπει σε παιδιά άνω της πέμπτης τάξης του δημοτικού να δημιουργήσουν παιχνίδια και προσομοιώσεις παρόμοιες με αυτές που παίζουν. Ο σχεδιασμός του συστήματος βασίστηκε σε μελέτες της γλώσσας που χρησιμοποιούν για την επίλυση προβλημάτων τα παιδιά που δεν έχουν προγραμματιστική εμπειρία. Το περιβάλλον διαθέτει ένα συμπαγές υπολογιστικό μοντέλο που αναπαρίσταται για τα παιδιά από τον Handy τον σκύλο, ο οποίος χειρίζεται ένα σύνολο καρτών. Όλες οι πληροφορίες που χρησιμοποιούνται σε ένα πρόγραμμα αποθηκεύονται και στις δύο πλευρές αυτών των καρτών. Το μπροστινό μέρος των καρτών περιέχει δεδομένα σχετικά με αντικείμενα ενώ το πίσω εμφανίζει μια εικόνα του αντικειμένου. Ο χρήστης μπορεί να τοποθετήσει αυτές τις κάρτες στην επιφάνεια ενός τραπέζιου το οποίο αναπαριστά την οπτική του ίδιου του χρήστη για την δομή του προγράμματος.



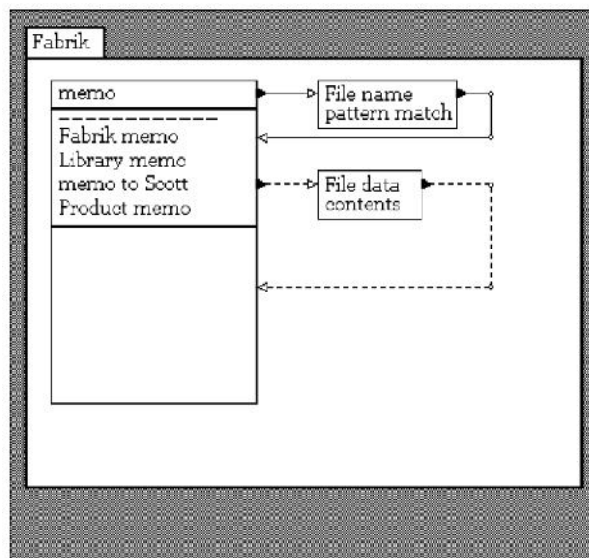
Βελτίωση αλληλεπίδρασης με τη γλώσσα

Επιπρόσθετα με τις αλλαγές στη γλώσσα και τη σύνταξή της, ένας τομέας που θα μπορούσε να βελτιωθεί ήταν οι τρόποι με τους οποίους αλληλεπιδρά η γλώσσα με τον άνθρωπο. Τα συστήματα σε αυτή την υποκατηγορία γεννήθηκαν ώστε να διευκολύνουν

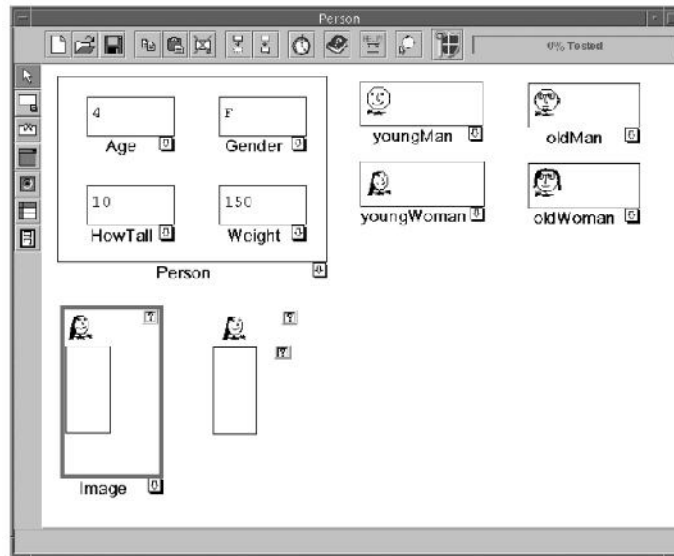
τους αρχάριους κατά την συγγραφή των προγραμμάτων, να βοηθήσουν στην κατανόηση τους και να αποτρέψουν οποιαδήποτε λάθη τους πριν αυτά συμβούν. Στον παρακάτω πίνακα παρουσιάζονται ορισμένα από αυτά τα συστήματα, συνοδευμένα από την τεχνική την οποία χρησιμοποίησαν οι κατασκευαστές τους για να τελειοποιήσουν την διαδραστικότητα γλώσσας-ανθρώπου[28].

Body electric, VPL 1990: Σχεδιάστηκε ως ένα συγγραφικό εργαλείο για ένα σύστημα εικονικής πραγματικότητας 2 ατόμων. Τα προγράμματα σε αυτό βασίζονται πάνω σε δεδομένα: ακατέργαστα δεδομένα από αισθητήρες μπορούν να περαστούν στην αναπαράσταση του εικονικού κόσμου μέσω προτύπων που έχουν την δυνατότητα να τα μεταμορφώνουν ή να παράγουν γεγονότα από αυτά. Τα πρότυπα αυτά αναπαρίστανται στο περιβάλλον με κουτιά συνδεδεμένα με βελάκια σε διάγραμμα ροής. Οι χρήστες μπορούν να δημιουργήσουν προγράμματα που τροποποιούν και αντιδρούν σε δεδομένα στέλνοντας στους αισθητήρες άλλα μέσω μιας αλληλουχίας προτύπων.

Fabrik, Apple Computer, 1988: Είναι ένα σύστημα υπολογιστικής δομής στο οποίο τμήματα διαδικασιών εμφανίζονται ως διασυνδεδεμένα κουτιά. Αυτά τα κουτιά μπορούν να συνδυαστούν και να δημιουργήσουν μια πληθώρα προγραμμάτων. Οι χρήστες εξοπλίζονται με ένα σύνολο απλών υπολογιστικών εργαλείων (π.χ. χειρισμού ακεραίων και γραμματοσειρών) και στοιχείων διεπαφών (όπως κουμπιά, εικόνες και λίστες). Με την τοποθέτηση και την σύνδεση κουτιών σε μια επιφάνεια εργασίας, ο χρήστης δημιουργεί προγράμματα, τα οποία παραμένουν πάντα ενεργά ώστε να μπορεί να τα ελέγχει καθώς τα κατασκευάζει.

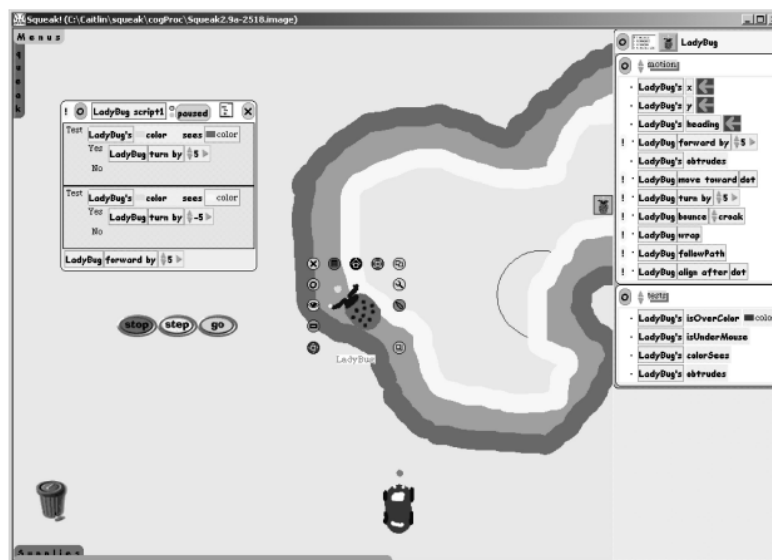


Forms/3, Oregon State University, 1995: Είναι μια οπτική προγραμματιστική γλώσσα βασισμένη στη φιλοσοφία των φύλλων εργασίας και σχεδιασμένη να δίνει περισσότερες δυνατότητες στο χρήστη, ενώ παράλληλα να διατηρεί την εύκολη χρήση τους. Σε αυτήν οι χρήστες δημιουργούν κελιά και δίνουν μαθηματικές εκφράσεις που το σύστημα θα χρησιμοποιήσει για να υπολογίσει την τιμή των πρώτων. Για την επέκταση των ειδών των προγραμμάτων που οι χρήστες μπορούν να γράψουν, το σύστημα παρέχει στους χρήστες τη δυνατότητα να δημιουργήσουν τους δικούς τους τύπους δεδομένων (περιλαμβανομένων των γραφικών) και να χρησιμοποιήσουν ένα ρολόι συστήματος για να πραγματοποιηθούν εν ευθέτω χρόνω υπολογισμοί.



Tangible Programming with Trains, MIT Media Lab, 1996: Είναι ένα σύστημα αποτελούμενο από τρενάκια και εξαρτήματα αυτών και σχεδιάστηκε ώστε να επιτρέπει σε παιδιά να εξερευνούν πρώιμες προγραμματιστικές έννοιες. Τα εξαρτήματα και τα τρενάκια επικοινωνούν μέσω υπερύθρων ώστε όταν πλησιάζουν αρκετά κοντά να αλλάζουν εγκαίρως συμπεριφορά. Ένα παράδειγμα συνδυασμού των παραπάνω δομικών στοιχείων του προγράμματος είναι ένα τρενάκι που σταματά σε ένα σταθμό.

Squeak Etoys, Disney, 1997: Σχεδιάστηκαν ώστε να επιτρέπουν σε παιδιά να μαθαίνουν έννοιες παίζοντας και κατασκευάζοντας με αυτές, είτε μέσω αλληλεπίδρασης με προσομοιωτές που άλλοι έχουν κατασκευάσει είτε δημιουργώντας νέες προσομοιώσεις. Το περιβάλλον αυτό παρέχει στους μαθητές μια ποικιλία προκαθορισμένων αντικειμένων και ένα απλό σχεδιαστικό εργαλείο με το οποίο οι μαθητές μπορούν να δημιουργήσουν τα δικά τους αντικείμενα. Τα προγράμματα που συντάσσονται μπορούν να αλλάζουν τη θέση, το μέγεθος και την εμφάνιση των αντικειμένων αλλά μια να αναπαράγουν ήχους. Οι χρήστες μπορούν να δημιουργήσουν απλές δομές ελέγχου.



Alice99, Carnegie Mellon University, 1999: Οι δημιουργοί της Alice98 παρατήρησαν πως η πληκτρολόγηση ήταν αρκετά δύσκολη για πολλούς χρήστες. Αυτό το σύστημα είναι ο διάδοχος της Alice98 και επικεντρώνεται σε τρόπους αναζήτησης μείωσης του μεγέθους του κειμένου που υποχρεώνονται να πληκτρολογήσουν οι χρήστες. Έτσι

χρησιμοποιήθηκε κυρίως η μέθοδος drag 'n' drop : ο χρήστης επιλέγει ένα χαρακτήρα από ένα δέντρο αποτελούμενο από αυτούς και μόλις το εναποθέσει στην επιφάνεια εργασίας του προγράμματος αναδύονται μενού, με τη βοήθεια των οποίων γίνεται δυνατός ο ορισμός παραμέτρων. Σε περίπτωση που ο χρήστης επιθυμεί να δημιουργήσει πιο σύνθετα προγράμματα, υπήρχε η δυνατότητα επιλογής του κλασσικού συντάκτη κειμένου.

AutoHAN, University of Cambridge, 2001: Δημιουργήθηκε από την ανάγκη ύπαρξης μιας απλής προγραμματιστικής διεπαφής για τις πολλές οικιακές συσκευές που διαθέτουν στοιχεία προγραμματισμού. Σκοπός του ήταν να «εκπαιδέσει» αυτές τις συσκευές ώστε να κάνουν απλές προγραμματισμένες εργασίες, όπως για παράδειγμα η εγγραφή μιας τηλεοπτικής εκπομπής ή η αυτόματη εκκίνηση της καφετιέρας. Οι σχεδιαστές του AutoHAN επέλεξαν να δημιουργήσουν μια ποικιλία φυσικών κύβων για να καλύψουν μεγάλο εύρος δραστηριοτήτων. Ένα παράδειγμα τέτοιας εφαρμογής με τη βοήθεια ενός κύβου είναι η αναπαραγωγή ενός CD, στην οποία ένας κύβος τοποθετείται πολύ κοντά στο CD player. Μόλις δημιουργηθεί η σύνδεση μεταξύ τους, ο χρήστης μπορεί να πιέσει το κουμπί του κύβου για να ακούσει το CD. Οι σχεδιαστές του AutoHAN πρότειναν δύο γλώσσες για τον χειρισμό των κύβων: μια βασισμένη σε οντολογική αφαίρεση και μία σε γλωσσολογική αφαίρεση. Στην πρώτη ανήκουν κύβοι γεγονότων και καναλιών που επιτρέπουν την σύνδεση και ομαδοποίηση τους, ενώ στην δεύτερη περιλαμβάνονται κύβοι που αντιστοιχούν σε λέξεις στα Αγγλικά, για παράδειγμα "Go" και "Play".

Physical Programming: University of Maryland, 2002: Το σύστημα αυτό περιγράφει μια μέθοδο για παιδιά 4-6 ετών με την οποία μπορούν να κατασκευάσουν διαδραστικές ιστορίες χρησιμοποιώντας το Story Room Kits, καθώς και αισθητήρες αλλά και ενεργοποιητές που χρησιμεύουν στον χειρισμό καθημερινών αντικειμένων όπως καρέκλες ή αρκουδάκια. Το Story Room Kits επιτρέπει στα παιδιά να δημιουργούν ιστορίες στις οποίες αντικείμενα του πραγματικού κόσμου αναπαριστούν χαρακτήρες ή στοιχεία σε αυτές. Η μέθοδος του Physical Programming δημιουργήθηκε χρησιμοποιώντας τεχνικές του Wizard of Oz και τα ακόλουθα εργαλεία: ένα «χέρι» που αναπαριστά την αφή, ένα φως ώστε να φωτίζονται τα αντικείμενα και να τραβούν την προσοχή τον χρηστών, ένα μουσικό κουτί που συσχετιζε κάθε πλευρά του με ένα διαφορετικό ήχο και μια μαγική ράβδο που έδειχνε πότε οι χρήστες προγραμματίζουν και πότε ήθελαν να αφηγηθούν μια ιστορία. Για την δημιουργία ενός προγράμματος, ένα παιδί συσχετιζε αισθητήρες, ενεργοποιητές και στηρίγματα με τη βοήθεια της μαγικής ράβδου.

Flogo, MIT Media Lab, 2001: Είναι μια οπτική γλώσσα ροής πληροφοριών που δίνει τη δυνατότητα σε παιδιά να κατασκευάσουν πιο σύνθετες ρομποτικές συμπεριφορές με τα Lego robotics kits. Οι σχεδιαστές του συστήματος πίστευαν πως η οπτικοποίηση της προσωρινής δομής ενός προγράμματος βοηθά στην κατανόηση του. Τα προγράμματα της Flogo είναι διαρκώς ζωντανά: μια πιθανή αλλαγή στις εισόδους τους θα φανεί άμεσα και στην έξοδο τους, γεγονός που την καθιστά μια αρκούντως φιλική γλώσσα.

Jive, University of Magdeburg, 2004: Εμπνευσμένο από το Squeak Etoys, σχεδιάστηκε με σκοπό να δώσει τη δυνατότητα σε μαθητές να κατασκευάσουν τρισδιάστατα διαδραστικά περιβάλλοντα, ενώ παράλληλα να κατανοήσουν και μαθηματικές έννοιες. Οι συγγραφείς του πίστευαν ότι αν ένα παιδί σχεδιάσει τους δικούς του χαρακτήρες, τότε θα αυξηθεί το ενδιαφέρον του για να τους προγραμματίσει. Παρέχονται δυσδιάστατες ζωγραφιές αυτών των χαρακτήρων, τις οποίες το σύστημα τοποθετεί σε τρισδιάστατα

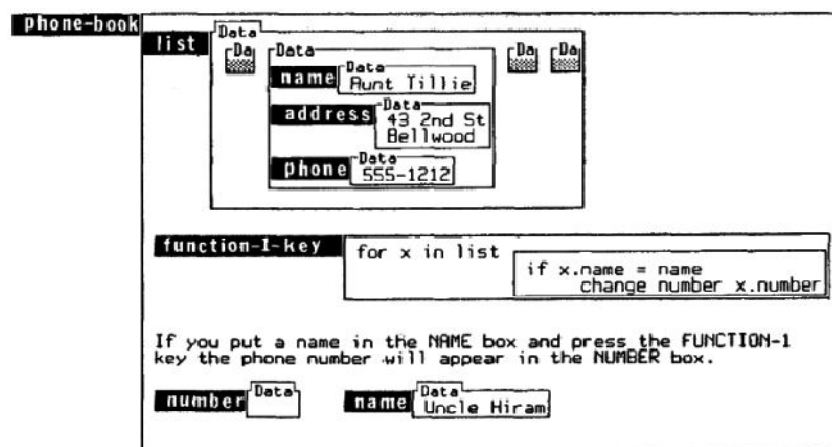
αντικείμενα χρησιμοποιώντας τον αλγόριθμο Teddy. Ενώ στο Etoys υπήρχε μόνο η δυνατότητα χρήσης ελέγχων με if, στο Jive διατίθενται δομές for και while.

Ενσωμάτωση στο περιβάλλον

Η διαδικασία για την δημιουργία ενός προγράμματος περιλαμβάνει την συγγραφή του κώδικα, την μεταγλώττιση του, την διόρθωση των λαθών, την κατασκευή ενός προγράμματος με τον παραπάνω κώδικα και τέλος την εκτέλεση του. Επειδή αυτά τα βήματα για έναν αρχάριο είναι αρκετά και συνήθως τον αποθαρρύνουν από το να ασχοληθεί, οι σχεδιαστές προσπάθησαν να δημιουργήσουν νέα συστήματα που θα ενσωματώνουν όλα τα παραπάνω στην διαδικασία συγγραφής του κώδικα.

Επιπροσθέτως ορισμένα συστήματα επέτρεπαν την εκτέλεση και προεπισκόπηση συγκεκριμένων τμημάτων κώδικα, ώστε να ενθαρρύνεται ο πειραματισμός και να διευκολύνεται περισσότερο η κατανόηση της γλώσσας Ακολουθούν ορισμένα παραδείγματα γλωσσών αυτής της υποκατηγορίας[28].

Boxer, University of California at Berkeley, 1986: Παρουσιάζει έναν ιεραρχικό κόσμο αποτελούμενο από κουτιά που μπορούν να περιέχουν άλλα κουτιά. Αντί να διαχωρίζεται μέσω του προγράμματος, η διαδικασία του προγραμματισμού ενσωματώνεται σε ένα περιβάλλον όπου ένας μαθητής μπορεί να χρησιμοποιήσει για σύνταξη κειμένου αλλά και δευτερευόντως για γραφικά σχεδιαγράμματα. Τα προγράμματα του Boxer περιλαμβάνουν τρία είδη κουτιών που μπορούν να περιέχουν κείμενο ή κώδικα, δεδομένα και γραφικές απεικονίσεις. Η σύνθεση αυτών των κουτιών έχει νόημα: δείχνει ότι οι υπο-διαδικασίες είναι μέρος των διαδικασιών και ότι οι καταχωρήσεις είναι μέρος των βάσεων δεδομένων. Γενικότερα, τα υπο-κουτιά είναι προσβάσιμα μόνο απ' το εσωτερικό των κουτιών. Τα κουτιά παρέχουν στους αρχάριους προγραμματιστές έναν απλό μηχανισμό για αυθαίρετα προγράμματα και στοιχεία δεδομένων, ενώ είναι δυνατή και η εμφάνιση δύσκολων τμημάτων κώδικα ως «μαύρα κουτιά», τα οποία εφόσον ο μαθητής εξοικειωθεί με το περιβάλλον μπορεί να ανοίξει και να κατανοήσει.



Hypercard, Apple Computer, 1987: Περιγράφεται από τον δημιουργό της Bill Atkinson ως ένα συγγραφικό εργαλείο και ένα είδος μηχανής αναπαραγωγής κασετών. Η εφαρμογή επιτρέπει στους χρήστες να δημιουργήσουν στοίβες από κάρτες που περιέχουν εικόνες, κείμενο και κουμπιά. Τα πιο απλά κουμπιά μπορούν να επιφέρουν οπτικές αλλαγές, να ηχήσουν μελωδίες ή να δείξουν μια νέα κάρτα. Διατίθεται ακόμα μια γλώσσα κειμένου που επιτρέπει στους χρήστες να δημιουργήσουν λειτουργικές εφαρμογές. Η γλώσσα

αυτή είναι σαφώς επηρεασμένη από τα ομιλούμενα Αγγλικά, καθώς ο Atkinson επικεντρώθηκε κατά την δημιουργία της στην πρώτη εμπειρία που θα έχει ο χρήστης μαζί της καθώς και στην υποστήριξή του κατά την διάρκεια της μάθησης. Η αποκάλυψη των στοιχείων της γλώσσας γινόταν σταδιακά. Αρχικά ο χρήστης μάθαινε να δημιουργεί κάρτες και να χρησιμοποιεί τα κουτιά μηνυμάτων και στη συνέχεια μάθαινε να χρησιμοποιεί γραφικές δομές και κουτιά μηνυμάτων, διαδικασίες δηλαδή πιο σύνθετες.

cT, Carnegie Mellon University, 1988: Αυτό το σύστημα επιχειρεί να απλοποιήσει τη διαδικασία δημιουργίας προγραμμάτων με βάση τα γραφικά παρέχοντας υψηλού επιπέδου εργαλεία. Τα προγράμματα δημιουργούνται μέσω ενός ενσωματωμένου περιβάλλοντος όπου οι χρήστες μπορούν να δουν τα αποτελέσματα τους άμεσα. Διατίθεται ακόμη και μια μέθοδος ώστε να μπορούν οι χρήστες να προσδιορίζουν σχήματα χρησιμοποιώντας τα κλικ του ποντικιού στην οθόνη.

Visual AgenTalk, University of Colorado, 1996: Είναι ένα προγραμματιστικό περιβάλλον βασισμένο στην προσέγγιση που οι σχεδιαστές αποκαλούν « Tactile Programming» και σύμφωνα με την οποία επιτρέπεται στους χρήστες να χειρίζονται κώδικα σε πολλαπλά πλαίσια ώστε να διευκολυνθούν στην κατανόηση, την κατασκευή πιο σύνθετων προγραμμάτων και την συνεργασία μεταξύ τους. Ακόμη πίστευαν πως οι χρήστες θα έπρεπε να έχουν τη δυνατότητα να «ρίχνουν» κομμάτια κώδικα σε 3 περιοχές εργασίας: τον συντάκτη προγραμμάτων, τον προγραμματιστικό κόσμο (τη βάση δηλαδή πάνω στην οποία το πρόγραμμα εκτελείται) και τον κόσμο συνεργασίας (διασύνδεση αντικειμένων και επικοινωνία).

Chart N Art, University of Colorado, 1996: Είναι ένα γραφικό περιβάλλον παρόμοιο με το McDraw. Καθώς οι σχεδιαστές χειρίζονται τη διεπαφή για να δημιουργήσουν σχέδια και πίνακες, οι αντίστοιχες προγραμματιστικές εντολές εμφανίζονται σε ένα κουτί που κρατά το ιστορικό στο κάτω μέρος της οθόνης. Αυτές οι εντολές μπορούν να αντιγραφούν και να εκτελεστούν. Η διεπαφή διαθέτει λειτουργίες για απλά αντικείμενα αλλά και ομάδες αυτών, επιτρέποντας στους σχεδιαστές να μάθουν πώς να τα προσδιορίζουν χρησιμοποιώντας εντολές κειμένου, ενώ ο σκοπός της είναι να επιτρέψει σε αυτούς την αυτοματοποιημένη δημιουργία πινάκων.

3.5.3 Δραστηριότητες που ενισχύει ο προγραμματισμός

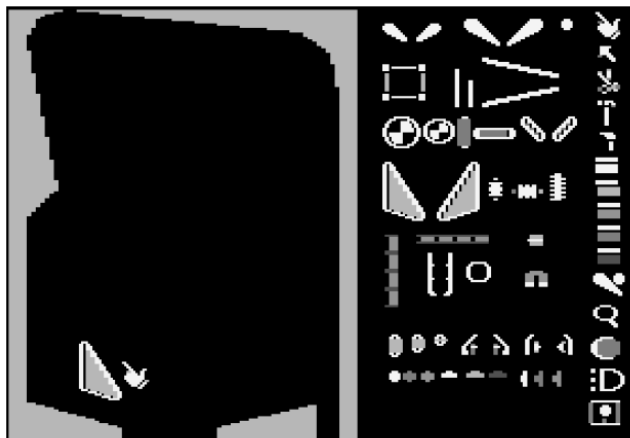
Τα συστήματα σε αυτή την κατηγορία βλέπουν τον προγραμματισμό ως ένα τρόπο να εμπλουτίζονται γενικότερες δραστηριότητες των αρχάριων, δίνοντας μεγαλύτερο έλεγχο ή παρέχοντας περισσότερες ευκαιρίες ώστε να εξερευνήσουν συγκεκριμένα πεδία. Οι κατασκευαστές τους δεν είχαν στο μυαλό τους να δημιουργήσουν ένα ολοκληρωμένο προγραμματιστικό περιβάλλον, αλλά ένα περιβάλλον κατάλληλα διαμορφωμένο έτσι ώστε να ενισχύονται παράπλευρες δραστηριότητες με την εκμάθησή τους, όπως η διασκέδαση των μαθητών και η γενικότερη εκπαίδευση.

Διασκέδαση

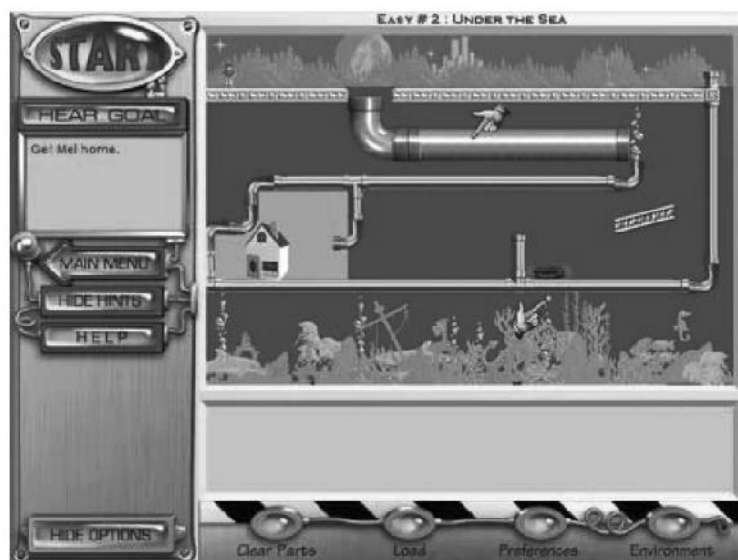
Τα συστήματα αυτής της υποκατηγορίας χρησιμοποιούν τον προγραμματισμό για δραστηριότητες διασκέδασης. Τα προγραμματιστικά μοντέλα που χρησιμοποιούν είναι απλά ώστε να γίνονται εύκολα κατανοητά από τους αρχάριους, ενώ βρίσκουν εφαρμογή σε πρωτότυπες, εναλλακτικές δραστηριότητες[28],

Pinball Construction Set, Exidy Software, 1983: Σχεδιάστηκε με σκοπό να επιτρέψει στους χρήστες να δημιουργούν εφαρμογές με φλιπεράκια. Διέθετε ένα χώρο εργασίας,

ένα σύνολο τμημάτων από ένα φλιπεράκι και δυνατότητες επεξεργασίας bitmaps εικόνων για προσωποποίηση των κατασκευών. Νόμοι της φυσικής και συμπεριφορές αποτελούσαν τα κομμάτια του κώδικα. Για παράδειγμα ο σχεδιαστής μπορούσε να ορίσει την αντίδραση μια μπίλιας όταν αυτή προσέκρουε σε μια ράμπα.



The Incredible Machine, Sierra Entertainment, 1993: Βασισμένο σε μηχανικές προκλήσεις του Goldberg, το Incredible Machine προσέφερε ένα εργαλείο υλοποίησης αυτών. Για παράδειγμα, μπορεί να ζητηθεί από τον παίχτη να κατασκευάσει ένα τρόπο ώστε μια μπάλα να μπαίνει μέσα σένα καλάθι. Κάθε τέτοια πρόκληση απαιτεί μια μικρή περιγραφή και όλα τα απαραίτητα μέρη που χρειάζεται για να δουλέψει. Οι παίχτες μπορούν να διαλέξουν αντικείμενα, να τα τοποθετήσουν στη μηχανή τους και να την βάλουν να δουλέψει. Κατά την προσομοίωση, τα τμήματα της μηχανής αντιδρούν όπως ακριβώς θα κάνανε και στον πραγματικό κόσμο.



Widget Workshop, Maxis, 1995: Διαθέτει μια σειρά έτοιμων γρίφων προς επίλυση, αλλά υποστηρίζει και την δημιουργία νέων. Κάθε τέτοιος γρίφος θέτει μια ερώτηση, για παράδειγμα το «Ποια χρώματα συνδυάζουμε ώστε να πάρουμε το άσπρο?» και παρέχει μια σειρά έτοιμων απαντήσεων με τις οποίες ο χρήστης πειραματίζεται, δηλαδή στη δική μας περίπτωση το σύνολο των χρωμάτων.

Bongo, MIT Media Lab, 1997 : Δίνει τη δυνατότητα στα παιδιά να κατασκευάσουν video games και να τα μοιραστού με άλλα μέσω του διαδικτύου. Η Bongo κατασκευάστηκε με βάση τη Starlogo (γενικότερα μπορούμε να πούμε ότι την μετατρέπει ώστε να

κατασκευάζονται με εύκολο τρόπο βιντεοπαιχνίδια), αλλά προσθέτει στοιχεία όπως αναπαραγωγή ήχων, αλλαγή σχημάτων και ανίχνευση συγκρούσεων των χαρακτήρων στην οθόνη.

Mindrover, Cognitoy , 2001: Είναι ένα εμπορικό παιχνίδι στο οποίο ο χρήστης υποδύεται έναν εξερευνητή στο διάστημα, ο οποίος στον ελεύθερο του χρόνο προγραμματίζει ρομποτικά οχήματα έτσι ώστε να διαγωνίζονται σε ταχύτητα. Χρησιμοποιείται πάλι η drag η drop τεχνική για τη δημιουργία προγραμμάτων από ένα σύνολο προκαθορισμένων στοιχείων. Τα οχήματα μπορούν να επιτελέσουν απλές λειτουργίες, με την εξοικείωση του χρήστη όμως παρέχονται και επιπλέον εντολές για δημιουργία πιο σύνθετων.

3.6 Σύνοψη κεφαλαίου

Κλείνοντας την εκτενή αναφορά στα προγραμματιστικά εργαλεία, θα ήταν θεμιτό να κάνουμε μια σύνοψη στους άξονες γύρω από τους οποίους αυτά κατασκευάστηκαν. Πιο συγκεκριμένα, μετά από την μελέτη όλων αυτών των προγραμματιστικών εργαλείων προκύπτουν εννιά κατασκευαστικοί άξονες, με αρκετά στοιχεία που υλοποιούν το καθένα και τα διαφοροποιούν μεταξύ τους. Είναι τα εξής:

- **Τα προγραμματιστικά στυλ :** Αποτελούνται από το ακολουθιακό, το αντικειμενοστραφές, το βασισμένο στα αντικείμενα , το συναρτησιακό, το βασισμένο σε γεγονότα καθώς και το βασισμένο σε καταστάσεις μηχανής στυλ.
- **Οι προγραμματιστικές δομές:** Αποτελούνται από γνωστές δομές όπως την υπό συνθήκη εκτέλεση, όλες τις επαναληπτικές δομές, μεταβλητές, παραμέτρους, διαδικασίες/ μεθόδους, τύπους δεδομένων του χρήστη και pre-post καταστάσεις.
- **Η αναπαράσταση του κώδικα:** Ο κώδικας μπορεί να αναπαρασταθεί με κείμενο, εικόνες, διάγραμμα ροής, animation, φόρμες, μηχανές τελικής κατάστασης και φυσικά αντικείμενα.
- **Ο τρόπος συγγραφής των προγραμμάτων:** Ο κώδικας μπορεί να εισαχθεί με πληκτρολόγηση, με αναπαράσταση γραφικών αντικειμένων, με τη επίδειξη ενεργειών, με τη συμπλήρωση ή την επιλογή φόρμας καθώς και με φυσικά αντικείμενα.
- **Η υποστήριξη για την κατανόηση ενός προγράμματος:** Οι αρχάριοι μπορούν να βοηθηθούν με ιστορικό λαθών, με αναλυτικό debugging, με φυσική διερμηνεία και γενικευμένα παραδείγματα
- **Η παρεμπόδιση συντακτικών λαθών:** Υλοποιείται συνήθως με μεθόδους όπως αναφορές σε φυσικά σχήματα, επιλογή του ορθού κώδικα από προτεινόμενες επιλογές, διόρθωση με βάση το συντακτικό, εισαγωγή συγκεκριμένου κώδικα μόνο στα κατάλληλα σημεία και πιο κατανοητά μηνύματα συντακτικών λαθών.
- **Ο σχεδιασμός προσβάσιμων γλωσσών:** Πραγματοποιείται με περιορισμό του domain, επιλογή λέξεων-κλειδιών προσφιλών στον χρήστη, αφαίρεση περιττών σημείων στίξης, χρήση φυσικής γλώσσας και αφαίρεση πλεοναζόντων στοιχείων
- **Η υποστήριξη επικοινωνίας:** Η επικοινωνία-συνεργασία μεταξύ των χρηστών μπορεί να πραγματοποιηθεί είτε όταν αυτοί κάθονται δίπλα δίπλα, είτε με χρήση δικτύου. Στη δεύτερη περίπτωση οι χρήστες μπορούν να επικοινωνούν είτε κατά τη συγγραφή του προγράμματος είτε κατά τη διαχείριση και σύνθεση των αποτελεσμάτων.

- **Το κίνητρο των εργασιών:** Οι χρήστες αποκτούν επιπλέον κίνητρο όταν οι εργασίες που καλούνται να διεκπεραιώσουν είναι χρηστικές, διασκεδαστικές ή εκπαιδευτικές σε κάποιο άλλο πεδίο γνώσης.

Η δημιουργία τόσων πολλών εργαλείων για την διδασκαλία του προγραμματισμού μαρτυρά από μόνη της πως είναι ένα σύνθετο και δύσκολο αντικείμενο. Τα κίνητρα και οι εμπνεύσεις των σχεδιαστών αυτών των συστημάτων μπορούν να αποτελέσουν φωτεινό οδηγό για οποιαδήποτε απόπειρα διδασκαλίας προγραμματισμού, συνδυασμένες όμως και εφαρμοσμένες ανάλογα με τις εκάστοτε συνθήκες. Επιπρόσθετα, επισημαίνουν έμμεσα ορισμένα από τα μεγαλύτερα προβλήματα της διαδικασίας της διδασκαλίας, τα οποία οφείλουν να λάβουν υπόψη τους υποχρεωτικά οι διδάσκοντες.

Το πρώτο από αυτά τα προβλήματα κάνει την εμφάνιση του από την πρώτη στιγμή και είναι οι τρόποι έκφρασης ενός προγράμματος. Οι αρχάριοι δεν μπορούν εύκολα να συντάξουν μια ολοκληρωμένη έκφραση. Οι προγραμματιστικές εφαρμογές που χρησιμοποιούν κώδικα, ακόμη και αυτές στις οποίες οι σχεδιαστές έχουν επιφέρει κάποιες βελτιώσεις, δεν συνίστανται για μαθητές με μηδενική εμπειρία. Ιδανικά για τέτοιες περιπτώσεις είναι τα εργαλεία που παρέχουν μια γραφική εφαρμογή αλλά παράλληλα και εναλλακτικό τρόπο προβολής του προγράμματος με κώδικα.

Ένα δεύτερο θέμα που προβληματίζει τους αρχάριους είναι η κατανόηση της δομής ενός προγράμματος. Ο διδάσκων οφείλει να δώσει μεγάλη έμφαση στην κατανόηση εννοιών όπως διαδικασία, υπορουτίνα, συνάρτηση, πίνακας, διαφορετικά οι διδασκόμενοι δεν θα καταφέρουν ποτέ να συντάξουν ένα ολοκληρωμένο πρόγραμμα. Το έργο του διδάσκοντα διευκολύνεται με εργαλεία που παρουσιάζουν και δομούν εναλλακτικά την δομή των προγραμμάτων, συνήθως με γραφικές απεικονίσεις. Εφόσον οι μαθητές εξοικειωθούν με αυτές τις έννοιες μπορούν χωρίς πολύ κόπο να τις εφαρμόσουν και σε νέα προγραμματιστικά εργαλεία.

Ακόμη ένα ζήτημα, ίσως το λιγότερο εμφανές με μια πρώτη ματιά αλλά και το πιο βασικό, είναι κατά πόσο οι μαθητές έχουν κατανοήσει την αντιστοίχιση μεταξύ ενός προγράμματος και των ενεργειών που πραγματοποιεί κατά την εκτέλεση του. Σε αυτή την περίπτωση οι αρχάριοι ισχυρίζονται ότι έχουν κατανοήσει ένα πρόγραμμα και τον κώδικά του, κατά την σύνταξη ενός δικού τους προγράμματος όμως τα αποτελέσματά του δεν ταιριάζουν στις προδιαγραφές του προβλήματος. Ο διδάσκων οφείλει μέσω παραδειγμάτων και αντίστοιχων γραφικών απεικονίσεων να δώσει στους μαθητές να καταλάβουν όλες τις εργασίες που εκτελούνται παρασκηνιακά από τον υπολογιστή. Εναλλακτικά μπορεί να χρησιμοποιήσει προγραμματιστικά εργαλεία που εμφανίζουν σε κάθε στάδιο του προγράμματος όλες τις αλλαγές που επιφέρει αυτό.

Αντίστοιχα με το παραπάνω, σε περιπτώσεις αρκετά δύσκολου ή μεγάλου κώδικα, αλλά και σε περιπτώσεις δύσκολων προγραμματιστικών εννοιών και ακόμη πιο δύσκολων μεταφορών τους σε μια προγραμματιστική γλώσσα, ο διδάσκων μπορεί με μια σειρά εργαλείων να παρουσιάσει στους μαθητές έτοιμα παραδείγματα ώστε η κατανόηση τους να γίνει αγγύστα. Τμήματα σύνθετου κώδικα μπορούν να χρησιμοποιηθούν ως μαύρα κουτιά, αν ο διδάσκων επιθυμεί να διευκολύνει τη ζωή των αρχάριων.

Ένα τόσο δύσκολο αντικείμενο όπως ο προγραμματισμός συχνά αποθαρρύνει τους αρχάριους από το να συνεχίσουν. Ο διδάσκων λοιπόν καλείται, εκτός από το να δώσει απαντήσεις σε εννοιολογικά και πρακτικά προβλήματα που αναφέρθηκαν πριν, να προσθέσει στην διδασκαλία του στοιχεία που σαφώς θα διευκολύνουν τους μαθητές

αλλά και θα κάνουν την διαδικασία πολύ πιο ευχάριστη και εποικοδομητική. Τέτοια στοιχεία μπορεί να δανειστεί από γενικότερες διδακτικές μεθόδους, όπως για παράδειγμα η συνεργατική μάθηση και η παροχή κινήτρου στους μαθητές. Μπορεί ακόμη να δώσει στους μαθητές μια σειρά από δραστηριότητες εκτός τάξης, να τους προτείνει εργαλεία (που χρησιμοποιούν προγραμματισμό) ψυχαγωγίας και εκπαίδευσης σε άλλα διδακτικά αντικείμενα, έτσι ώστε να ενισχύονται και να εμπλουτίζονται οι γνώσεις τους.

4. Διδακτικό πλάνο

4.1 Περιγραφή διδακτικού πλάνου

Η διδασκαλία αποφασίστηκε να οργανωθεί πάνω σε ένα διδακτικό πλάνο τριών δίωρων μαθημάτων. Στο πρώτο μάθημα ο συμμετέχων εξοικειώνεται με το ρομπότ, την γραφική διεπαφή και τις βασικές εντολές κίνησης του Scribbler, στη δεύτερη με την δομή της εντολής επανάληψης και στην τρίτη με την δομή της εντολής ελέγχου. Το διδακτικό πλάνο σχεδιάστηκε ώστε ο μαθητής να γνωρίσει μέσα από τις δραστηριότητες και άλλες δευτερεύουσας σημασίας εντολές (παύσης, αναπαραγωγής ήχου, χειρισμός φωτισμού), ενώ «χτίστηκε» έχοντας ως ιστορικό υπόβαθρο την αποστολή ενός ρομπότ στο διάστημα. Ακολουθεί τόσο το διδακτικό πλάνο, όσο και λεπτομερείς περιγραφές που αφορούν την διδακτική προσέγγιση του μαθήματος και όλα όσα θα χρειαστεί κάποιος για να το εφαρμόσει.

4.2 Πλαίσιο εκμάθησης: «Ερευνητική δραστηριότητα στον πλανήτη Άρη»

4.2.1 Στόχοι διδασκαλίας

Η συνολική δραστηριότητα της εργασίας καλύπτει γνωστικούς στόχους που αφορούν το αντικείμενο της Πληροφορικής και ειδικότερα του προγραμματισμού, σε βασικό πάντα επίπεδο. Παράλληλα, οι διδασκόμενοι έρχονται σε επαφή με διαφορετικά αλλά κλασσικά γνωστικά αντικείμενα, όπως η Φυσική, τα Μαθηματικά, αλλά και εξοικειώνονται με ερευνητικές δραστηριότητες όπως αυτές στο διάστημα.

Οι μαθητές, με το πέρας της διδασκαλίας, θα μπορούν να:

- **Γνωστικοί στόχοι**
 - Περιγράφουν τα βασικά χαρακτηριστικά ενός ρομπότ .
 - Περιγράφουν τις λειτουργίες που επιτελούν αυτά τα χαρακτηριστικά (πχ δράση αισθητήρων).
 - Χρησιμοποιούν βασικές προγραμματιστικές δομές και εντολές μέσω της γραφικής διεπαφής του ρομπότ ώστε να χειρίζονται τα χαρακτηριστικά του (κίνηση, αναπαραγωγή ήχων, λήψη ερεθισμάτων από το περιβάλλον και διενέργεια ελέγχων).
 - Μετρούν φυσικά μεγέθη όπως ο χώρος, ο χρόνος, η ένταση του φωτός και του ήχου, που επηρεάζουν τη λειτουργία του ρομπότ.
 - Συγκρίνουν και να αξιολογούν προτεινόμενες προγραμματιστικές λύσεις.
- **Δεξιότητες**
 - Να αξιοποιούν τεχνικές επίλυσης προβλήματος και να επιλέγουν τα ιδανικά μέσα για να την επιτύχουν.
 - Να διατυπώνουν υποθέσεις και να ελέγχουν την ορθότητά τους.
 - Να διατυπώνουν και να αξιολογούν επιχειρήματα που στηρίζονται στα δεδομένα τα οποία έχουν συλλέξει.

- **Στάσεις**

- Να γνωρίζουν και να αξιολογούν την προσφορά της επιστήμης και της τεχνολογίας στην ευημερία του σύγχρονου ανθρώπου.
- Να εργάζονται σε ομάδες και να λειτουργούν συνεργατικά αλλά και να συναγωνίζονται, με σεβασμό στην ιδιαιτερότητα του καθενός.

- **Απαιτούμενοι πόροι**

- Η συνολική διάρκεια της διδασκαλίας θα είναι 6 διδακτικές ώρες. Απαιτούνται ακόμη ένας προσωπικός υπολογιστής για κάθε μαθητή, οι κατασκευές που περιγράφονται στις αντίστοιχες δραστηριότητες και ένα μέσο προβολής οπτικοακουστικού υλικού.



α. Υλικό δραστηριοτήτων



β. Υλικό δραστηριοτήτων



γ. Υλικό δραστηριοτήτων



δ. Υλικό δραστηριοτήτων

- **Ηλικιακή ομάδα-Σύνδεση με το αναλυτικό πρόγραμμα**
 - Το διδακτικό πλάνο μπορεί να εφαρμοστεί σε αρχάριους σε θέματα προγραμματισμού, που διαθέτουν όμως βασικές γνώσεις χρήσης Η/Υ. Μπορεί ακόμη να εφαρμοστεί σε έμπειρους σε προγραμματιστικά περιβάλλοντα χρήστες, ώστε να συγκριθεί η παρούσα διδασκαλία με αυτές που ήδη έχουν βιώσει.
- **Λογισμικό/Υλικό**
 - Θα χρησιμοποιηθεί το Scribbler robot της Parallax και η γραφική διεπαφή που το συνοδεύει.

4.2.2 1η Διδακτική ώρα

Στόχοι του μαθήματος

Θα παραθέσω τους στόχους του μαθήματος βάσει της ταξινόμιας του Bloom: Οι μαθητές μετά το πέρας της διδασκαλίας θα πρέπει να είναι σε θέση:

Επίπεδο γνώσης

- Να γνωρίζουν τι πραγματεύεται το αντικείμενο του προγραμματισμού και ορισμένα βασικά στοιχεία για την εξέλιξη του.
- Να γνωρίζουν τι πραγματεύεται το αντικείμενο της ρομποτικής, την εξέλιξη της και την χρήση της ως διδακτικό μέσο.
- Να γνωρίζουν τη διδακτική ρομποτική στον προγραμματισμό ως συνδυασμό των παραπάνω αντικειμένων καθώς και τα πλεονεκτήματά της.
- Να γνωρίζουν τα βασικά χαρακτηριστικά του Scribbler robot και τις λειτουργίες που επιτελεί το καθένα.
- Να γνωρίζουν τα βασικά χαρακτηριστικά της γραφικής διεπαφής του Scribbler robot και την δομή των βασικών προγραμμάτων.
- Να γνωρίζουν την δομή και την φιλοσοφία των εντολών κίνησης του ρομπότ μέσω της γραφικής διεπαφής .

Επίπεδο κατανόησης

- Να μπορούν να κατανοήσουν τις λειτουργίες που επιτελούν τα επιμέρους εξαρτήματα του ρομπότ και πως επηρεάζουν τη συμπεριφορά του κατά την εκτέλεση ενός προγράμματος.
- Να μπορούν να αντιληφθούν την γενικότερη φιλοσοφία της γραφικής διεπαφής του Scribbler και την χρήση της ως μέσο για τον προγραμματισμό του.
- Να κατανοούν τις βασικές εντολές κίνησης του ρομπότ και τι αποτελέσματα έχουν σε αυτό, καθώς και να ξεπερνούν όποιες δυσκολίες προκύψουν κατά την εφαρμογή τους.
- Να έχουν τη δυνατότητα να κατανοούν την λειτουργικότητα των βασικών εντολών κίνησης και να επεξηγούν αυτή σε έτοιμα παραδείγματα.

Επίπεδο εφαρμογής

- Να δομούν βασικά προγράμματα στη γραφική διεπαφή του Scribbler.
- Να χρησιμοποιούν τις βασικές εντολές κίνησης με επιτυχία.
- Να μπορούν να τροποποιούν τις παραμέτρους των εντολών αυτών ώστε να επιτυγχάνουν ένα επιθυμητό αποτέλεσμα κίνησης.

Επίπεδο ανάλυσης

- Να μπορούν να αναλύσουν τα πλεονεκτήματα και την αξία της ρομποτικής κατά την διδασκαλία του προγραμματισμού.

Προσανατολισμός Μαθήματος

Σύντομη εισήγηση (5λεπτά)

Στη συγκεκριμένη φάση της διδασκαλίας, μέσω μιας σύντομης εισαγωγής, θα προσπαθήσουμε με προβολή ενός βίντεο να φέρουμε σε πρώτη επαφή τους μαθητές με

το αντικείμενο της ρομποτικής. Σε αυτό περιλαμβάνονται σύγχρονες ρομποτικές εφαρμογές με σκοπό να εγείρουν το ενδιαφέρον των μαθητών. Κατά τη διάρκεια της προβολής του βίντεο θα γίνουν ερωτήσεις στους μαθητές, όπως : «Τι κάνει το ρομπότ αυτή τη στιγμή;», «Πιστεύετε ότι εκτελεί τη συγκεκριμένη δραστηριότητα καλύτερα από τον άνθρωπο;». Ακόμη, οι μαθητές θα γνωρίσουν το ρομπότ που θα χρησιμοποιηθεί στη διδασκαλία μας, το Scribbler, σε κάποιες εκπαιδευτικές εφαρμογές.

Ερωταποκρίσεις(5 λεπτά)

Αμέσως μετά, μέσω ερωταποκρίσεων θα προσπαθήσουμε να εξάγουμε τα οφέλη χρήσης της ρομποτικής στη διδασκαλία του προγραμματισμού και να καταρρίψουμε το μύθο που θέλει τον προγραμματισμό ένα σύνθετο και δυσνόητο αντικείμενο. «Αναφέρετε ρομπότ που γνωρίζετε από την καθημερινή ζωή», «Τι είδους δραστηριότητες μπορούσαν να εκτελέσουν;», «Διευκολύνουν ή δυσκολεύουν αυτές τις ανθρώπινες δραστηριότητες;», «Ποιές από αυτές πιστεύετε ότι μπορεί να εκτελέσει το Scribbler;», «Πόσο δύσκολο πιστεύετε ότι θα είναι να υλοποιηθούν;». Αυτές οι ερωτήσεις θα διεγείρουν το ενδιαφέρον των μαθητών και θα ταυτίσουν την αναπαράσταση των δικών τους παραδειγμάτων με την αντίστοιχη προγραμματιστική, έξοδος της οποίας θα είναι το Scribbler.

Κύρια δραστηριότητα (100 λεπτά)

Στην κύρια δραστηριότητα της πρώτης διδακτικής ώρας δημιουργούμε μία μικρογραφία του κόσμου στον οποίο βρίσκεται το Scribbler. Σκοπός είναι οι μαθητές να αναγνωρίσουν τις δυνατότητες-λειτουργικότητες του ρομπότ και ποιες ενέργειες μπορεί αυτό να εκτελέσει. Οι μαθητές θα εξοικειωθούν με το ρομπότ και τον εξοπλισμό που φέρει και θα μάθουν πως μπορεί να κινηθεί σύμφωνα με την επιθυμία τους. Με το πέρας των διδακτικών δραστηριοτήτων θα πρέπει να μπορούν να εξάγουν κανόνες, να αναγνωρίζουν ποια είναι η είσοδος και ποια η έξοδος του προγράμματος και να έχουν αποκτήσει μια στοιχειώδη εμπειρία με το Scribbler και τις δυνατότητες του. Για να γίνουν πιο εύκολα κατανοητές οι διάφορες δραστηριότητες στους μαθητές, θα κάνουμε διηγηματικές περιγραφές

Φάση 1^η-[Διερεύνηση] (15 λεπτά):

Στόχος της πρώτης φάσης είναι να αντιληφθούν οι αρχάριοι την βασική διαδικασία εισόδου-εξόδου που συναντάμε στον προγραμματισμό και να τη συνδυάσουν με αυτή της λογικής οντότητας (Scribbler). Για την εκπλήρωση αυτού του στόχου, παρέχεται αρχικά στους μαθητές, η πρώτη δραστηριότητα του φύλλου εργασίας του πρώτου μαθήματος. Σε αυτήν, χρησιμοποιείται η διδακτική μέθοδος της διερεύνησης, ενώ υποθέτουμε πως το Scribbler δραστηριοποιείται σε έναν εικονικό κόσμο (επιφάνεια πλανήτη, εκτέλεση εργασιών για λογαριασμό ερευνητών). Πιο συγκεκριμένα, παρέχονται στους μαθητές όλες οι απαραίτητες οδηγίες ώστε να ενεργοποιήσουν το βασικό πρόγραμμα αποφυγής εμποδίων του Scribbler, χωρίς να γνωρίζουν τι πρόκειται να κάνει το ρομπότ. Μετά από ένα εύλογο χρονικό διάστημα (ως 1 min) στο οποίο το Scribbler αποφεύγει ορισμένα εμπόδια , ο εκπαιδευτικός ζητά από τους μαθητές να προχωρήσουν στις επόμενες δραστηριότητες. Αυτές περιλαμβάνουν τρεις ερωτήσεις και μια άσκηση αντιστοίχισης, με τη βοήθεια των οποίων επιτυγχάνεται σταδιακά ο διδακτικός στόχος που τέθηκε στην αρχή της φάσης.

Φύλλο Εργασίας 1**Αποφυγή εμποδίων**

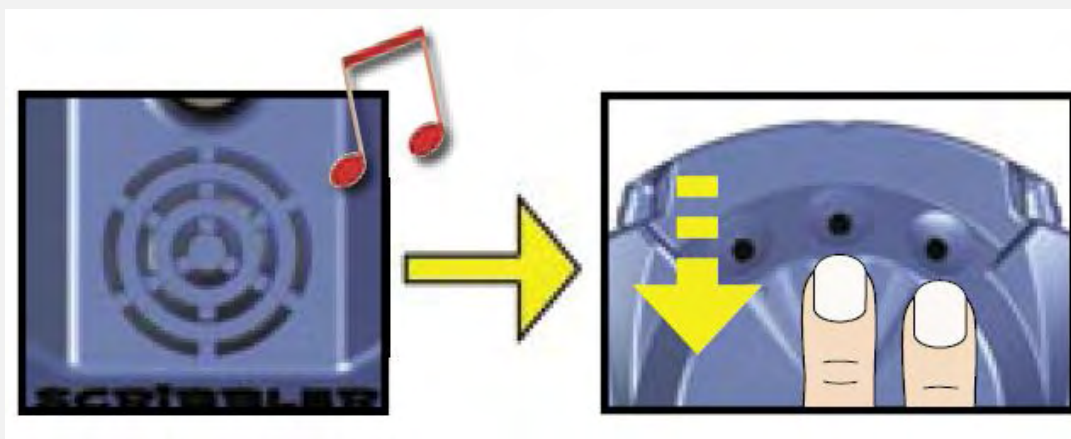
Η NASA έχει αποφασίσει να στείλει το ρομπότ μας σε μία αποστολή στον Άρη. Οι επιστήμονες δε ψάχνουν για συγκεκριμένα είδη ζωής αλλά για τυχόν ευρήματα που θα μαρτυρούν ότι οι συνθήκες που επικρατούσαν στο παρελθόν ήταν βιώσιμες. Σκοπός της αποστολής είναι η συλλογή στοιχείων από το χώμα και από τους βράχους του πλανήτη. Στην περιοχή που αφήνεται το ρομπότ μας, υπάρχει γκρεμός και φυσικά εμπόδια γι' αυτό μπορεί να κινηθεί μόνο σε συγκεκριμένα γεωγραφικά όρια αποφεύγοντας να πέσει στο γκρεμό και άλλα φυσικά εμπόδια που υπάρχουν στην περιοχή. Οι επιστήμονες έχουν προνοήσει και το ρομπότ είναι εξοπλισμένο έτσι ώστε να επιτυγχάνει αποφυγή εμποδίων. Στην μακέτα μας έχουμε σχεδιάσει μία περιοχή η οποία αποτελεί προσομοίωση της περιοχής του Άρη που βρίσκεται το ρομπότ μας.

Δραστηριότητες μαθητή

1.



Καθώς καλύπτετε με τα δάκτυλα σας την μεσαία και δεξιά οπή, πιέστε μία φορά το κουμπί reset όπως δείχνει η εικόνα.



Μόλις ακουστεί ο ήχος απελευθερώστε τα δάκτυλα σας από τις οπές.

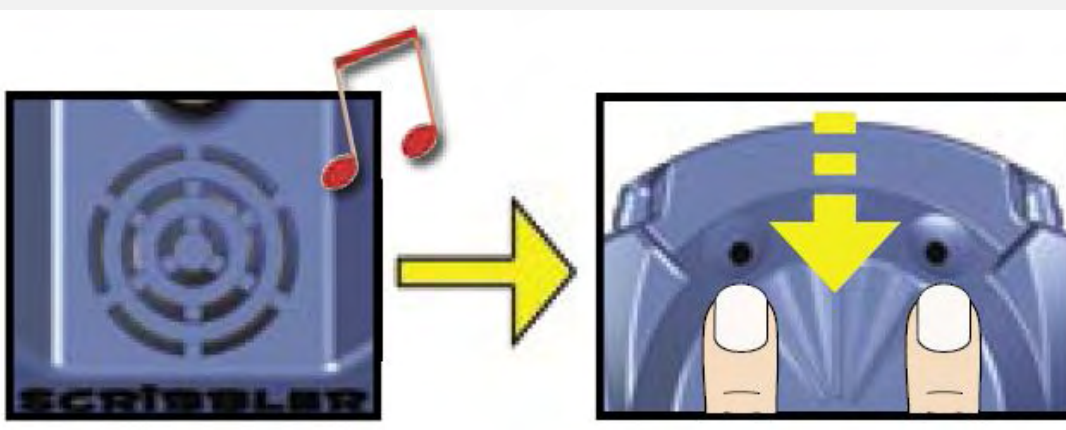
2. Ποια ενέργεια εκτελεί το ρομπότ μετά την πρώτη δραστηριότητα; Πώς πιστεύετε ότι επιτυγχάνεται αυτή;
3. Ποια λειτουργικότητα του ρομπότ πιστεύετε ότι υλοποιούν αυτές οι οπές;
4. Πώς βοήθησε η τοποθέτηση των δακτύλων στις οπές του ρομπότ; Η ενέργεια του ρομπότ θα ήταν ίδια αν η τοποθέτηση των δακτύλων ήταν διαφορετική;

Οι πληροφορίες για την ενεργοποίηση του ρομπότ είναι οι εξής:

Τοποθετήστε το ρομπότ στην αρχή της γραμμής. Έπειτα καθώς καλύπτετε με τα δάκτυλα σας τον δεξιό και αριστερό αισθητήρα πιέστε μία φορά το κουμπί reset.



Μόλις ακουστεί ο ήχος απελευθερώστε τα δάκτυλα σας από τους αισθητήρες.



Δραστηριότητες μαθητή

1. Ακολουθήστε τις οδηγίες ενεργοποίησης της νέα αποστολής. Ποια ενέργεια παρατηρείται να εκτελεί το Scribbler;

.....

2. Σε κάποιο σημείο, η γραμμή της πορείας σταματά για ένα διάστημα. Πώς παρατηρείται ότι αντιδρά το ρομπότ;

.....

3. Πώς πιστεύεται ότι καταφέρνει να εκτελεί τη συγκεκριμένη ενέργεια; (Υποδείξτε πάνω στο ρομπότ τα όργανα εκείνα που το βοηθούν). Είναι τα ίδια με αυτά της προηγούμενης αποστολής;

4. Ποια θα ήταν η συμπεριφορά του Scribbler αν το σχήμα ήταν τετράγωνο;

.....

Άσκηση εμπέδωσης

Γίνετε επιστήμονες και σχεδιάστε στο χαρτόνι την πορεία που θέλετε να ακολουθήσει το ρομπότ. Ακολουθήστε τα βήματα της προηγούμενης δραστηριότητας και δείτε αν το ρομπότ μπορεί να εκτελέσει την πορεία που έχετε σχεδιάσει.

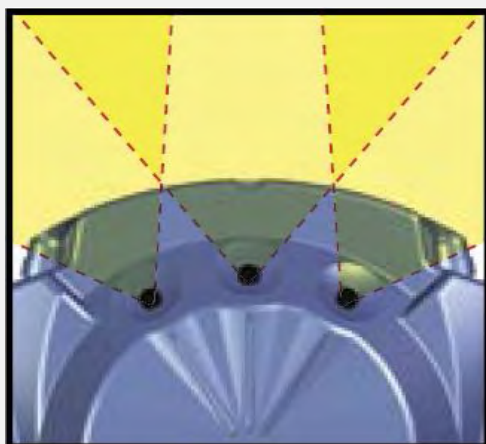
Φάση 3^η-[Επίδειξη] (15 λεπτά):

Με την τρίτη δραστηριότητα και με τη συμβολή της διδακτικής μεθόδου της επίδειξης, ο μαθητής καλείται να κατευθύνει το ρομπότ χρησιμοποιώντας ένα φακό και τους αισθητήρες του. Στόχος αυτής της φάσης είναι να μάθει ο αρχάριος πώς οι επιλογές του στην είσοδο (θέση φακού) επηρεάζουν σε πραγματικό χρόνο την έξοδο (κατεύθυνση κίνησης του ρομπότ). Χρησιμοποιώντας το τρίτο φύλλο εργασίας, ο μαθητής καθοδηγείται επαρκώς ώστε να ενεργοποιήσει το πρόγραμμα επιλογής φωτεινότερου σημείου του Scribbler. Έπειτα, καλείται να πειραματιστεί με το ρομπότ και τους αισθητήρες του κάνοντας χρήση του φακού.

Φύλλο Εργασίας 3

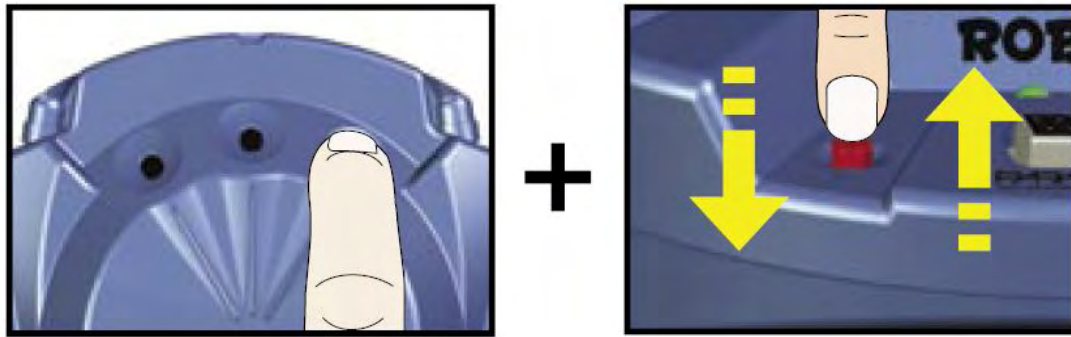
Ακολουθώντας το φως

Για τις περιπτώσεις μη επανδρωμένης αποστολής, οι επιστήμονες έχουν εξοπλίσει το ρομπότ με ακόμα μία λειτουργικότητα στους μπροστά αισθητήρες σύμφωνα με την οποία το ρομπότ μπορεί να ακολουθήσει μία πορεία και να βρει το φωτεινότερο σημείο. Έτσι, όταν τελειώσει η αποστολή, το διαστημόπλοιο που έχει ως ρόλο να πάρει το ρομπότ από τον Άρη πίσω στη Γη θα φωτίσει την επιφάνεια του πλανήτη και το ρομπότ θα επιστρέψει στο σκάφος.

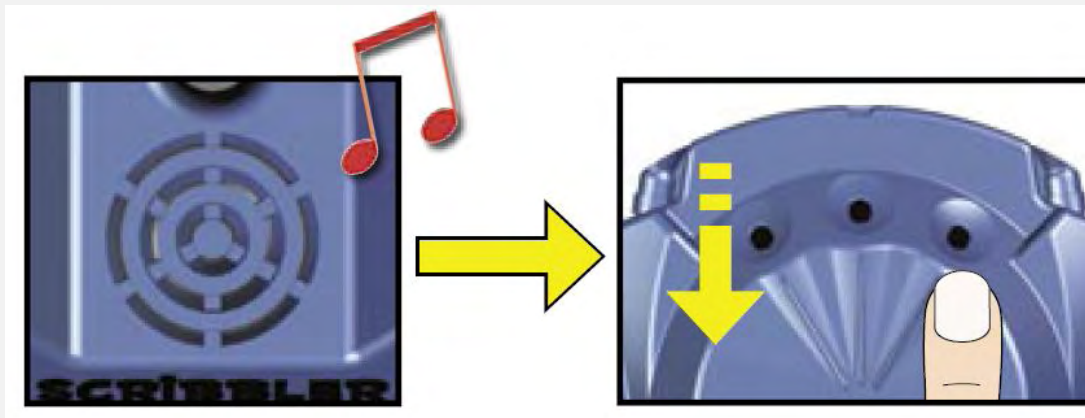


Οι τρεις αισθητήρες μπορούν να ανιχνεύσουν το φως όπως φαίνεται στην εικόνα. Ο μεσαίος ανιχνεύει ευθεία μπροστά και οι άλλοι δύο κατά 30° μοίρες δεξιά και αριστερά

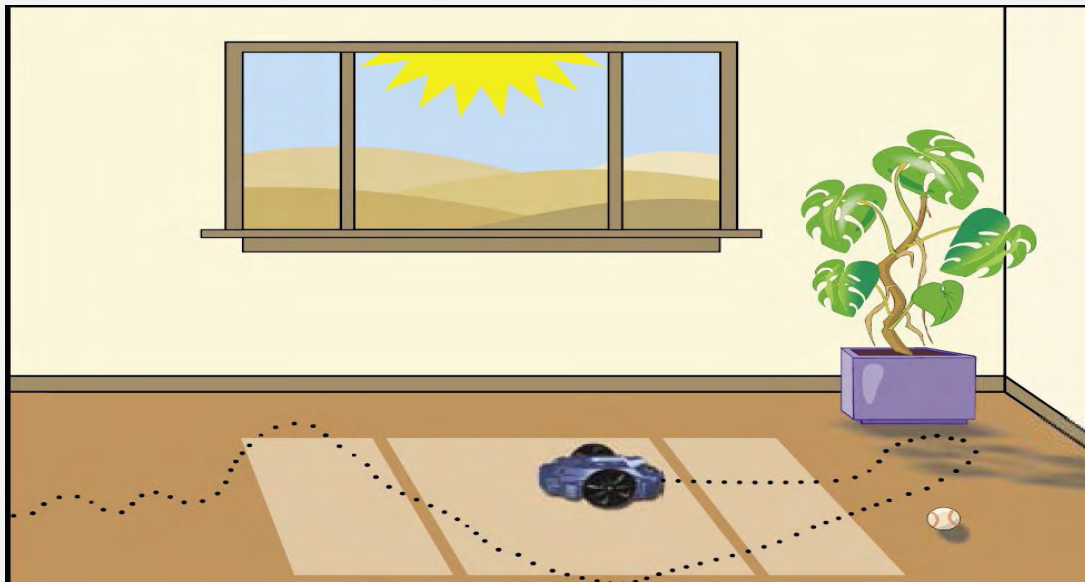
αντίστοιχα.



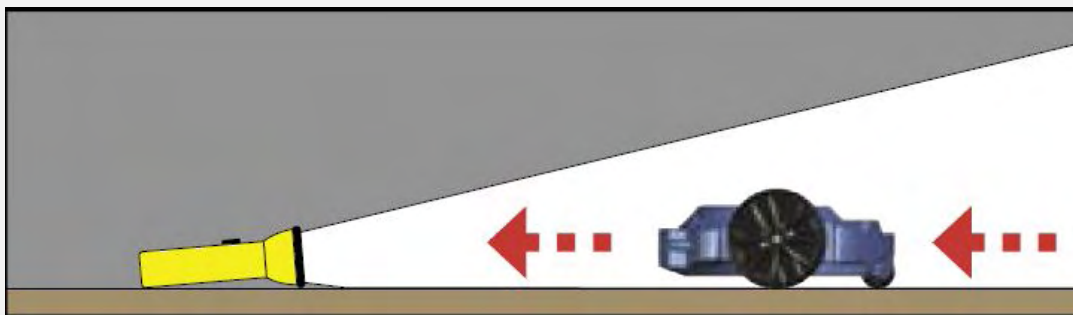
Καθώς καλύπτετε με το δάκτυλο σας τον δεξιό αισθητήρα πιέστε μία φορά το κουμπί reset όπως δείχνει η εικόνα.



Μόλις ακουστεί ο ήχος απελευθερώστε το δάκτυλο σας από τον αισθητήρα.



Τώρα το ρομπότ παίζοντας ήχο θα κινηθεί μέσα στο χώρο ψάχνοντας το φωτεινότερο σημείο.

Δραστηριότητες μαθητή

Σβήστε τα φώτα ώστε το δωμάτιο να είναι σκοτεινό και τοποθετήστε στο έδαφος ένα φακό ο οποίος παίζει το ρόλο του διαστημόπλοιου που ήρθε να μαζέψει το ρομπότ από τον Άρη. Μετακινήστε το φακό και δείτε αν το ρομπότ ακολουθεί το φως.

Φάση 4^η-[Μελέτη περίπτωσης] (15 λεπτά):

Κατά τη διάρκεια της 4^{ης} φάσης του μαθήματος ο εκπαιδευτικός κάνει εισαγωγή στο γραφικό περιβάλλον του Scribbler και στις εντολές κίνησης του. Στόχος είναι οι μαθητές να δουν πως χρησιμοποιείτε ο προγραμματισμός για να λυθεί ένα συγκεκριμένο πρόβλημα και τι οφέλη αποκομίζουν από τις δεξιότητες που αποκτούν μέσω της επίλυσης προβλημάτων στον υπολογιστή. Επίσης θα μάθουν πως αποδομείται ένα πρόβλημα σε μικρότερα των οποίων η λύση είναι πιο απλή. Καλούνται επίσης να χρησιμοποιήσουν τις πρότερες γνώσεις που έχουν αποκτήσει από τις προηγούμενες δραστηριότητες για να αναγνωρίσουν την είσοδο και την έξοδο του προγράμματος. Επειδή είναι η πρώτη επαφή των μαθητών με το γραφικό περιβάλλον του Scribbler και είναι η πρώτη φορά που βλέπουν πρόγραμμα σε αυτό, δημιουργούμε μέσω διηγηματικής περιγραφής μία μελέτη περίπτωσης. Παρέχεται στους μαθητές βοηθητικό βίντεο, το οποίο παρουσιάζει σταδιακά την επίλυση του προβλήματος καθώς και στιγμιότυπο της τελικής λύσης του. Μετά το πέρας της δραστηριότητας με στοχευμένες ερωτήσεις οι μαθητές θα έχουν μάθει να χρησιμοποιούν και τεχνικές δοκιμής και αποσφαλμάτωσης του κώδικα. Δίνεται στους μαθητές το φύλλο εργασίας 4.

Φύλλο Εργασίας 4**Κίνηση**

Το ρομπότ ενώ βρίσκεται στον Άρη εμφανίζει κάποια δυσλειτουργία με αποτέλεσμα να μην δουλεύουν σωστά ούτε οι εμπρός ούτε οι κάτω αισθητήρες. Καθώς δεν υφίσταται ως επιλογή η ματαίωση της αποστολής οι επιστήμονες πρέπει να προγραμματίσουν μόνοι τους το ρομπότ ώστε να μαζέψει δείγματα από το χώμα του πλανήτη. Ιδιαίτερη προσοχή πρέπει να δοθεί στο σχεδιασμό ώστε το ρομπότ να βρίσκεται στα γεωγραφικά όρια της νοητής περιοχής που έχουμε σχεδιάσει καθώς στην αποφυγή φυσικών εμποδίων που θα συναντήσει. Το ενθαρρυντικό είναι ότι γνωρίζουμε εκ των προτέρων τη γεωγραφική περιοχή και τη μορφολογία του εδάφους και έτσι μπορούμε να σχεδιάσουμε τις απαραίτητες κινήσεις.



Καθώς δε μπορούμε να προγραμματίσουμε για πόσα εκατοστά θα κάνει μία κίνηση αλλά μόνο για πόσα δευτερόλεπτα θα κινείται, μας δίνεται από τις προδιαγραφές ότι το ρομπότ διανύει περίπου 15cm σε 1sec και κάνει μία περιστροφή 90° μοιρών σε χρόνο 2sec δίνοντας ως συντεταγμένες 50-20. Οι διαστάσεις του ρομπότ είναι 15x15cm.

Το πρόβλημα

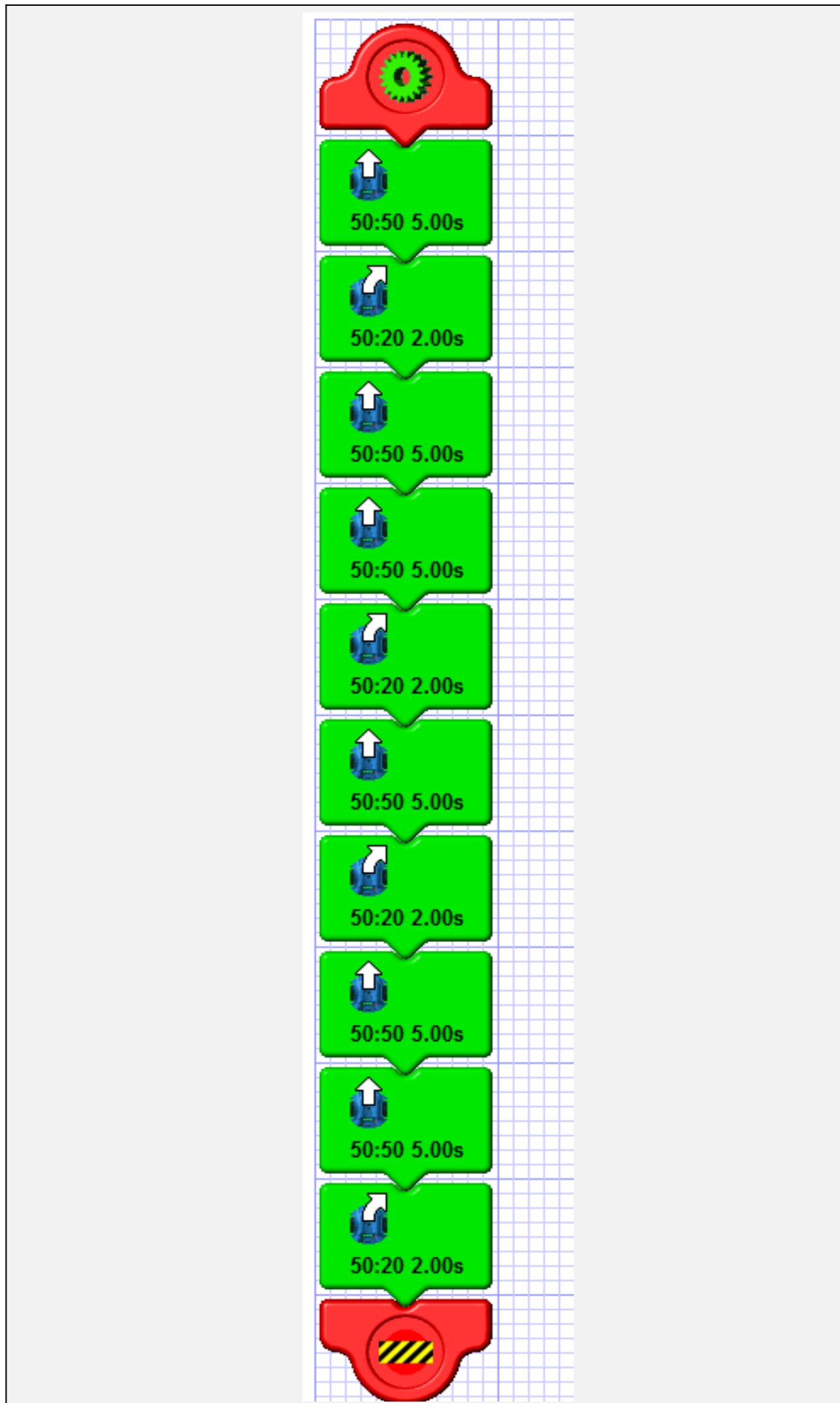
Οι επιστήμονες πριν ξεκινήσουν να χρησιμοποιούν όλες τις λειτουργικότητες του ρομπότ, θέλουν να υλοποιήσουν ένα δοκιμαστικό πρόγραμμα σύμφωνα με το οποίο το ρομπότ να διατρέχει από άκρη ως άκρη όλη την περιοχή και να επιστρέφει στη θέση που ξεκίνησε.

Ανάλυση του προβλήματος

Στην αρχή είναι προτιμότερο να σχεδιάσουμε τη λύση στο χαρτί και μετά να πάμε στην υλοποίηση της μέσα από το πρόγραμμα του ρομπότ. Από τα δεδομένα αυτό που καταλαβαίνουμε είναι ότι θέλουμε το ρομπότ να διαγράψει μία κίνηση σχηματίζοντας ένα παραλληλόγραμμο χωρίς να ξεπερνά τα όρια της παραπάνω περιοχής.

Τα ζητούμενα αφού το ρομπότ έχει μήκος 15cm να μην κινηθεί πάνω από 5sec ευθεία καθώς από τις προδιαγραφές ξέρουμε ότι 5 sec ισούται με 75cm. Το άλλο ζητούμενο είναι η κίνηση κατά 90° μοίρες που πρέπει να κάνει για να πάρει μία στροφή πρέπει να γίνει σε χρόνο 2 sec όπως γνωρίζουμε από τις προδιαγραφές.

Στον φάκελο First lesson->Video2.avi βρίσκεται μια ενδεικτική υλοποίηση του παραπάνω προβλήματος. Στην παρακάτω εικόνα παρέχεται ένα στιγμιότυπο της τελικής λύσης του προβλήματος.



Δραστηριότητες μαθητή

1. Ακολουθήστε τις οδηγίες του βίντεο και συμβουλευτείτε την εικόνα της τελικής λύσης για να φτιάξετε και να τρέξετε μόνοι σας το πρόγραμμα.

2. Ποια είναι τώρα η είσοδος και ποια η έξοδος του προγράμματος;

.....

.....

.....

.....

3. Τι θα συνέβαινε αν ανταλλάσσαμε θέσεις ανάμεσα στην 2^η και την 3^η κίνηση;

.....

.....

.....

.....

4. Τι θα συνέβαινε αν στην 5^η κίνηση αντί για χρόνο 0.5sec δίδαμε 2.0sec;

.....

.....

.....

.....

5. Προσθέστε μερικές κινήσεις ακόμα στο υπάρχον πρόγραμμα που έχετε φτιάξει ώστε το ρομπότ να προσεγγίσει οριακά τα γεωγραφικά όρια του παραλληλογράμμου.

Φάση 5^η-[Διερεύνηση] (20 λεπτά):

Κατά τη διάρκεια αυτής της φάσης ο εκπαιδευτικός, χρησιμοποιώντας τη μέθοδο της διερεύνησης, παρουσιάζει στους μαθητές ένα πρόβλημα και ένα στιγμιότυπο της τελικής του λύσης σε πρόγραμμα. Οι μαθητές καλούνται μέσω της διερεύνησης να προβλέψουν τη συμπεριφορά του ρομπότ. Στόχος της δραστηριότητας είναι οι μαθητές να αναπτύξουν γενικές τεχνικές επίλυσης προβλημάτων και να χρησιμοποιήσουν τις γνώσεις που έχουν χρησιμοποιήσει στις προηγούμενες φάσεις. Πρέπει δηλαδή μετά το πέρας της δραστηριότητας να έχουν κατανοήσει την τεχνική του διαίρει και βασίλευε σπάζοντας το πρόγραμμα σε μικρότερα υποπρογράμματα η λύση των οποίων είναι ήδη γνωστή. Λόγω της χρήσης της εντολής pause η οποία είναι άγνωστη στους μαθητές δίνεται επίσης ένα βοηθητικό βίντεο με τη χρήση της. Δίνεται στους μαθητές το φύλλο εργασίας 5.

Φύλλο Εργασίας 5**Κίνηση με παύση**

Αφού οι επιστήμονες έτρεξαν το δοκιμαστικό πρόγραμμα για τη σάρωση όλης της περιοχής από άκρη ως άκρη αποφεύγοντας τα εμπόδια τώρα είναι σε θέση προχωρήσουν στο κύριο μέρος της αποστολής, το οποίο είναι η συλλογή δειγμάτων από το έδαφος του πλανήτη. Η περιοχή που βρίσκεται το ρομπότ μας είναι χαρτογραφημένη όπως φαίνεται είδαμε πριν.

Οι επιστήμονες θέλουν τα ρομπότ κατά τη διάρκεια της κίνησης του στα όρια της χαρτογραφημένης περιοχής να μαζέψει δείγματα από 4 σημεία.

Ανάλυση προβλήματος

Για να μαζέψει τα δείγματα από το έδαφος το ρομπότ πρέπει να σταματήσει στο σημείο που βρίσκεται το δείγμα για διάστημα 2sec. Γι' αυτό το λόγο πρέπει εκτός από τις κινήσεις που ήδη έχουμε μάθει να προστεθεί η pause. Η χρήση της υπάρχει σε ένα παράδειγμα με δύο κινήσεις και δύο στάσεις στο φάκελο First lesson->Video3.avi.

Δραστηριότητες μαθητή

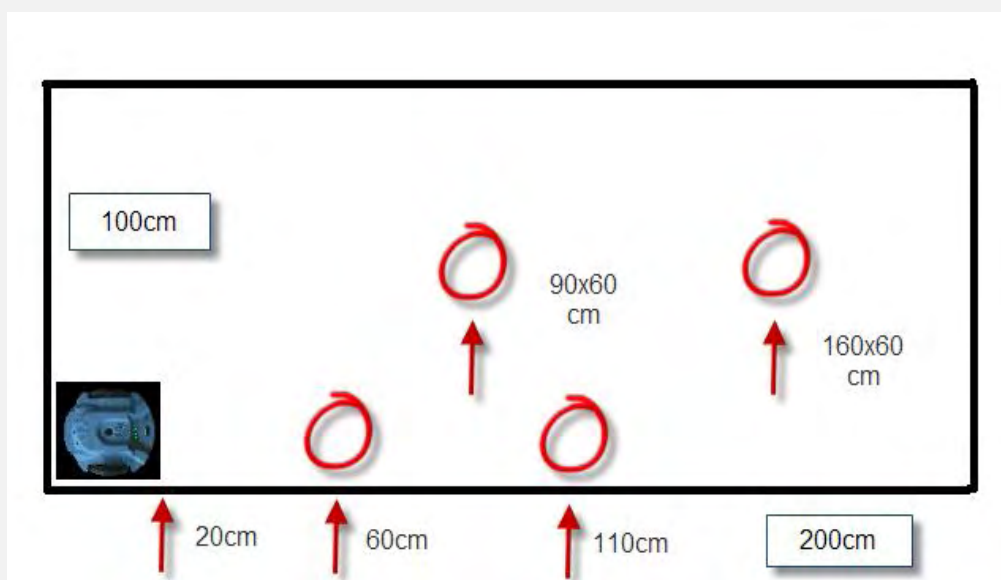
1. Προσθέστε στο προηγούμενο πρόγραμμα 4 τέσσερις εντολές παύσης σε όποια σημεία του προγράμματος επιθυμείτε. Πριν την εκτέλεση του προγράμματος σας σημειώστε στο χάρτη τα σημεία που πιστεύετε ότι θα σταματήσει για να συλλέξει δείγμα το ρομπότ. Τρέξτε το πρόγραμμα για να επιβεβαιώσετε την εκτίμηση σας.

Φάση 6^η-[Συνεργατική επίλυση προβλήματος] (20 λεπτά):

Σε αυτή τη φάση της διδασκαλίας οι μαθητές έχοντας μελετήσει τα παραδείγματα των προηγούμενων φάσεων καλούνται να δουλέψουν σε ομάδες για την επίλυση του συγκεκριμένου προβλήματος. Μέσω της δραστηριότητας ικανοποιούνται οι στόχοι στο επίπεδο εφαρμογής. Οι προηγούμενες δραστηριότητες και το βοηθητικό υλικό θα λειτουργήσουν υποστηρικτικά. Παρέχεται βίντεο με την υλοποίηση της εντολής αναπαραγωγής ήχου. Δίνεται στους μαθητές το φύλλο εργασίας 6.

Φύλλο Εργασίας 6

Όπως και στην προηγούμενη δραστηριότητα, απαιτείται ξανά η συλλογή δειγμάτων από την επιφάνεια του πλανήτη. Τώρα όμως η περιοχή που βρίσκονται τα δείγματα είναι χαρτογραφημένη. Τα κυκλωμένα σημεία στο χάρτη είναι τα μέρη που πρέπει να σκάψει το ρομπότ για συλλέξει τα δείγματα. Μόλις ολοκληρωθεί η λήψη, το ρομπότ ειδοποιεί τους επιστήμονες, αναπαράγοντας ένα μουσικό τόνο. Στον φάκελο First lesson->Video4.avi βρίσκεται ένα βοηθητικό βίντεο της εντολής αναπαραγωγής ήχου.



Δραστηριότητες μαθητή

1. Χωριστείτε σε ομάδες των δύο ατόμων. Γράψτε ένα πρόγραμμα χρησιμοποιώντας τη γραφική διεπαφή του Scribbler σύμφωνα με το οποίο το ρομπότ θα επισκέπτεται τα κυκλωμένα σημεία στο χάρτη. Σε κάθε σημείο το ρομπότ ακινητοποιείται για 2sec ώστε να συλλέξει το δείγμα και θα αναπαραγάγει ένα τόνο της επιλογής σας.
2. Η πρώτη ομάδα θα σημειώσει ακόμα ένα σημείο στο χάρτη και η δεύτερη θα επεκτείνει το πρόγραμμα ώστε να επισκεφτεί και να μαζέψει δείγμα το ρομπότ και από το νέο σημείο. Αλλάξτε ρόλους στις ομάδες και κάντε την ίδια δραστηριότητα.

Συνεδρία με τους μαθητές(10 λεπτά)

Κατά την διάρκεια της συνεδρίας οι μαθητές σχολιάζουν με τον διδάσκοντα το μάθημα που προηγήθηκε, διατυπώνουν κρίσεις, ερωτήσεις και απαντούν σε ερωτήσεις του που τον βοηθούν να το αξιολογήσει.

4.2.3 2η Διδακτική ώρα**Στόχοι μαθήματος**

Παραθέτουμε τους στόχους του μαθήματος βάσει της ταξινόμιας του Bloom:

Οι μαθητές μετά το πέρας της διδασκαλίας θα πρέπει να είναι σε θέση:

Επίπεδο γνώσης

- Να αναγνωρίζουν το block εντολών ελέγχου της γραφικής διεπαφής του Scribbler.
- Να γνωρίζουν πως μπορούν να ρυθμίσουν τον μετρητή ώστε να εκτελεστεί συγκεκριμένες φορές ο βρόγχος της επανάληψης.

Επίπεδο κατανόησης

- Να αναγνωρίζουν τις περιστάσεις στις οποίες είναι κατάλληλη η δομή επανάληψης «Για» και «Για πάντα» .
- Να μπορούν να περιγράψουν τη φιλοσοφία που διέπουν ορισμένα παραδείγματα δομών επανάληψης.
- Να αντιλαμβάνονται τη ροή εκτέλεσης του προγράμματος και να μην θεωρούν ότι αυτό σταματά ή αρχίζει από την αρχή.
- Να μπορούν να κατανοήσουν και να εξηγήσουν τη μέθοδο με την οποία εκτελείται η εντολή επανάληψης σε έτοιμα παραδείγματα κώδικα της γραφικής διεπαφής.
- Να κατανοούν ότι όταν ο μετρητής δεν παίρνει κάποια τιμή, τότε θα οδηγηθούν σε ατέρμονα βρόγχο.
- Να μπορούν να κατανοήσουν τις εμφωλευμένες δομές επανάληψης.

Επίπεδο εφαρμογής

- Να χρησιμοποιούν δομές επανάληψης της γραφικής διεπαφής για τον χειρισμό του ρομπότ.
- Να δημιουργούν μετρητές και να εκμεταλλεύονται τις επαναλήψεις για την επίλυση προβλημάτων.

- Να χρησιμοποιούν εμφωλευμένες δομές επανάληψης για την επίλυση προγραμματιστικών προγραμμάτων.

Ανάλυση

- Να μπορούν να αναλύσουν τα πλεονεκτήματα και την αξία των δομών επανάληψης.

Προσανατολισμός μαθήματος

Σύντομη εισήγηση (10 λεπτά):

Στη συγκεκριμένη φάση της διδασκαλίας μέσω μιας σύντομης εισαγωγής θα προσπαθήσουμε να υπενθυμίσουμε στους μαθητές την αξία των επαναλήψεων, εξάγοντας μέσω ερωτήσεων την ανάγκη ύπαρξης επανάληψης ενεργειών με ή χωρίς προκαθορισμένο αριθμό επαναλήψεων εκ των προτέρων. Θα χρησιμοποιήσουμε παραδείγματα από την καθημερινή τους ζωή όπως: «Μπορείτε να φανταστείτε περιστάσεις που μπορεί να θέλουμε να πραγματοποιηθεί μια ενέργεια πολλές φορές αλλά να μη γνωρίζουμε από πριν τον αριθμό των επαναλήψεων;», «Μπορείτε να μου πείτε αντίστοιχες περιστάσεις από παιχνίδια που παίζετε;». Οι ερωτήσεις θα μας οδηγήσουν σε πολλές περιστάσεις αντίστοιχων αναγκών οπότε διεγείρουμε το ενδιαφέρον των μαθητών για το πώς τα δικά τους παραδείγματα μπορούν να αναπαρασταθούν προγραμματιστικά.

Φάση 1^η-[Μελέτη Περίπτωσης] (20 λεπτά):

Μέσω της μελέτης περίπτωσης ο εκπαιδευτικός προσπαθεί να φέρει σε επαφή τους εκπαιδευόμενους με την έννοια της δομής επανάληψης. Παρέχεται ένα παράδειγμα κώδικα που περιέχει μια απλή δομή επανάληψης με μετρητή καθώς και συνοδευτικό βίντεο με τη συγγραφή του προγράμματος έτσι ώστε ο εκπαιδευόμενος να αντιληφθεί τη χρήση του block της. Οι μαθητές καλούνται να μελετήσουν αυτό το παράδειγμα, να δουν τα αποτελέσματά του, φορτώνοντας το στο ρομπότ και να απαντήσουν στις ερωτήσεις που τους δίνονται στο φύλλο εργασίας. Στόχος αυτής της φάσης είναι να κατανοήσουν οι μαθητές πως δουλεύει η δομή επανάληψης με μετρητή και μέσω στοχευμένων ερωτήσεων, να χτυπηθούν μαθησιακές δυσκολίες όπως ο ατέρμων βρόγχος και η λειτουργία του μετρητή. Δίνεται στους μαθητές το πρώτο φύλλο εργασίας.

Φύλλο Εργασίας 1

Η συλλογή δειγμάτων από την επιφάνεια του πλανήτη είναι μία πολύ σημαντική λειτουργία του ρομπότ για τους επιστήμονες. Ορισμένα από τα δείγματα που το ρομπότ καλείται να συλλέξει είναι κατά πολύ μεγαλύτερα από το μέγιστο βάρος που δύναται να μεταφέρει. Για να αντιμετωπίσουν αυτό το πρόβλημα, οι επιστήμονες πρόσθεσαν τη λειτουργικότητα της επανάληψης, ώστε το ρομπότ να συλλέγει μικρά δείγματα πολλές φορές από το ίδιο σημείο. Το πρόγραμμα που ακολουθεί εκτελεί την παραπάνω λειτουργικότητα.

Πρόβλημα

Σε απόσταση 2 δευτερολέπτων και μέσης ταχύτητας (50-50 για κάθε τροχό) βρίσκεται ένα μεγάλο δείγμα, το μέγεθος του οποίου είναι 5 φορές μεγαλύτερο από το ανώτατο βάρος

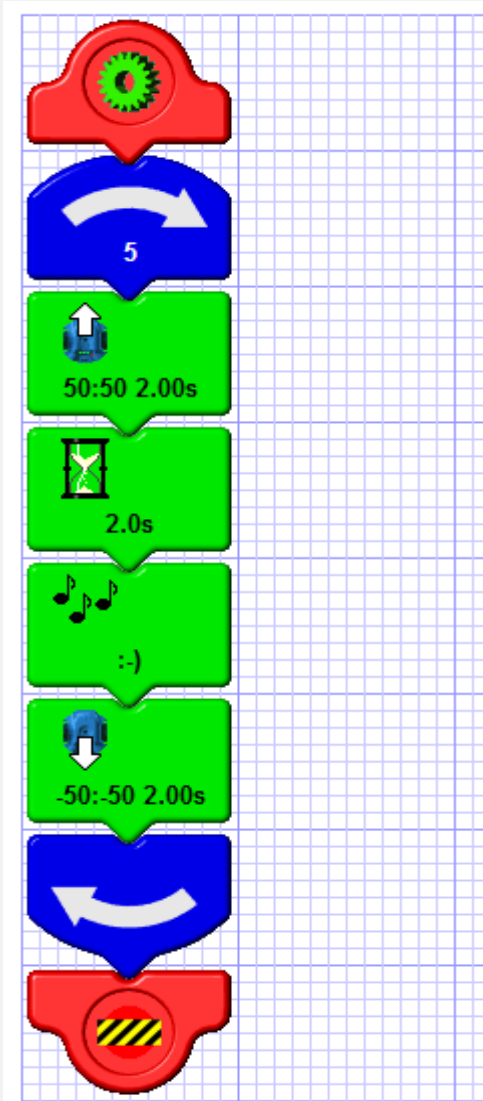
που μπορεί να μεταφέρει το ρομπότ. Οι επιστήμονες γράφουν ένα πρόγραμμα με τον ελάχιστο δυνατό αριθμό εντολών που υλοποιεί την παραπάνω διαδικασία.

Ανάλυση του προβλήματος

Η συλλογή δειγμάτων πραγματοποιείται με τον ίδιο τρόπο που διδάχθηκε στην προηγούμενη διδακτική ώρα, δηλαδή με παύση δύο δευτερολέπτων και αναπαραγωγή ενός μουσικού τόνου. Εφόσον η διατύπωση του προβλήματος ζητά η υλοποίηση να αποτελείται από τις λιγότερες δυνατές εντολές, θα γίνει χρήση της επαναληπτικής δομής. Το βάρος είναι πενταπλάσιο του μέγιστου επιτρεπτού βάρους που μπορεί να σηκώσει το ρομπότ, άρα η επανάληψη θα εκτελεστεί πέντε φορές.

Λύση

Μελετήστε και κατανοήστε την λύση του παραπάνω προβλήματος.



Δραστηριότητες μαθητή

1. Τι θα αλλάζατε στο πρόγραμμά σας αν το βάρος του δείγματος ήταν επτά φορές μεγαλύτερο από το βάρος που μπορεί να μεταφέρει το ρομπότ;

.....

.....

.....

2. Τι θα συνέβαινε στο πρόγραμμα αν η τιμή του μετρητή ήταν μηδέν; Θα εκτελεστεί καθόλου, και αν ναι πόσες φορές;

.....

3. Αν η τελευταία εντολή βρισκόταν εκτός της επανάληψης ποια θα ήταν η συμπεριφορά του ρομπότ;

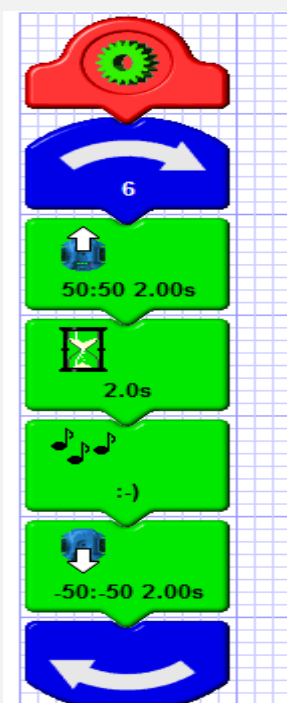
.....

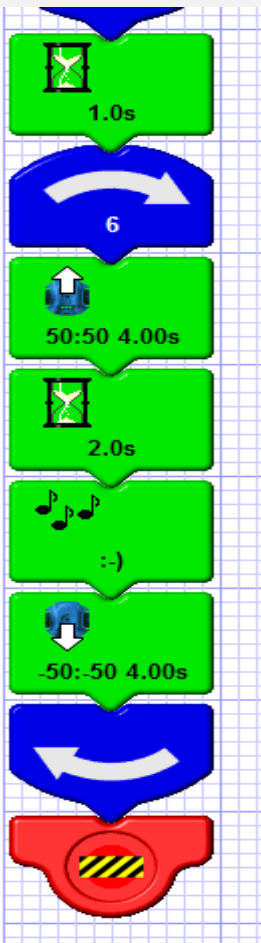
Φάση 2^η-[Διερεύνηση- Σύντομα παραδείγματα] (25 λεπτά):

Μέσω της φάσης διερεύνησης ο εκπαιδευτικός θα προσπαθήσει να δείξει στους μαθητές ότι με τη χρήση των δομών επανάληψης παύει να ισχύει η σειριακή εκτέλεση των προγραμμάτων που είχε διδαχθεί μέχρι πρότινος. Στόχος είναι να αντιληφθούν οι μαθητές ποιο είναι το σώμα του βρόγχου που εκτελείται επαναληπτικά, όπως επίσης την αρχή και το τέλος της επαναληπτικής διαδικασίας, στοιχεία που προβληματίζουν ιδιαίτερα τους εκπαιδευόμενους κατά την διδασκαλία του προγραμματισμού. Προς εκπλήρωση των ανωτέρω, δίνεται ένα παράδειγμα με δύο διαδοχικούς βρόγχους που εκτελούνται ανεξάρτητα καθώς και μια σειρά ερωτημάτων που ο μαθητής καλείται να απαντήσει. Δίνεται στους μαθητές το δεύτερο φύλλο εργασίας.

Φύλλο εργασίας 2

Μελετήστε και κατανοήστε το παρακάτω πρόγραμμα που συνέταξαν οι επιστήμονες και απαντήστε στις ερωτήσεις που ακολουθούν





Δραστηριότητες μαθητή

Δεδομένης της διαδικασίας λήψης δειγμάτων που είδατε σε προηγούμενα παραδείγματα, απαντήστε στα κάτωθι:

1. Από πόσο σημεία συλλέγει δείγματα με αυτό το πρόγραμμα το ρομπότ;

.....

2. Πόσο μεγαλύτερα είναι τα δείγματα από το μέγιστο βάρος που μπορεί να μεταφέρει το ρομπότ;

.....

3. Έστω ότι ο μετρητής της πρώτης επανάληψης ήταν μηδέν. Θα συνέλεγε το ρομπότ τα δείγματα από το δεύτερο σημείο;

.....

4. Αν επιθυμούσαμε η συλλογή δειγμάτων από τα δύο σημεία να γίνεται εναλλάξ κατά δύο (δύο δείγματα από το πρώτο-δύο δείγματα από το δεύτερο κοκ) πόσες δομές επανάληψης θα χρειαζόμασταν; Γράψτε ένα πρόγραμμα που να υλοποιεί το ζητούμενο.

5. Ποια θα ήταν η έξοδος του προγράμματος αν αντί για δύο δομές επανάληψης είχαμε μια μεγαλύτερη που θα περιείχε όλες τις εντολές με αριθμό μετρητή έξι; Αν ο μετρητής ήταν δώδεκα, τι θα συνέβαινε;

.....

.....

.....

.....

.....

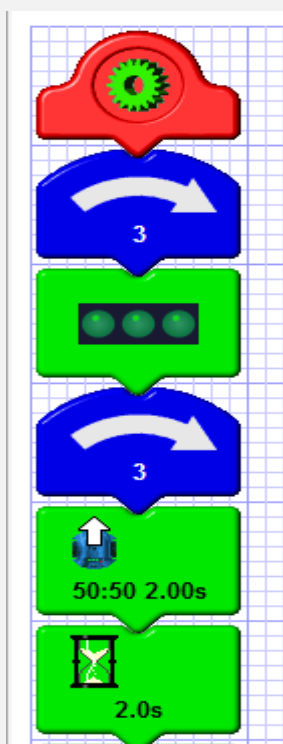
6. Οι ερευνητές διαπίστωσαν πως αν το ρομπότ συνέλεγε δείγματα εναλλάξ από τα δύο σημεία η κατανάλωση καυσίμων θα ήταν κατά πολύ μικρότερη. Γράψτε ένα πρόγραμμα χρησιμοποιώντας μια δομή επανάληψης, έτσι ώστε να υλοποιείται η παραπάνω λειτουργικότητα.

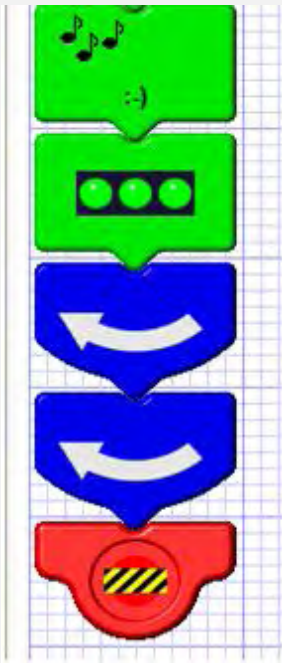
Φάση 3^η-[Τεχνική μαύρου κουτιού] (25 λεπτά):

Με την τεχνική του μαύρου κουτιού ο εκπαιδευτικός θα προσπαθήσει να εισάγει τους μαθητές στην έννοια των εμφωλευμένων δομών επανάληψης. Οι μαθητές καλούνται να τρέξουν ένα παράδειγμα που αποτελείται από δύο επαναληπτικές δομές, η μία από τις οποίες είναι εμφωλευμένη, ενώ στην εξωτερική περιέχεται και μία επιπλέον εντολή. Στόχος είναι να ξεπεραστούν οι δυσκολίες που αντιμετωπίζουν οι αρχάριοι του προγραμματισμού στις εμφωλευμένες δομές, όπως η λανθασμένη αντίληψη πως ο εσωτερικός βρόγχος εκτελείται μια φορά σε κάθε επανάληψη του εξωτερικού. Δίνεται στους μαθητές το τρίτο φύλλο εργασίας.

Φύλλο εργασίας 3

Τρέξτε το παρακάτω παράδειγμα και απαντήστε στις ερωτήσεις:





Δραστηριότητες μαθητή

1.α. Συμπληρώστε τον παρακάτω πίνακα.

Μετρητής πρώτης επανάληψης	Μετρητής δεύτερης επανάληψης
1	
	2
	3
2	2
3	1
3	3

1.β. Με βάση τις απαντήσεις που συμπληρώσατε στον πίνακα, πόσες φορές πιστεύετε ότι θα εκτελεστεί μια εντολή που βρίσκεται εντός του δεύτερου βρόγχου και πόσες μια που είναι εκτός;

.....

.....

.....

.....

1.γ. Πόση απόσταση σε δευτερόλεπτα θα έχει διανύσει το ρομπότ μετά το τέλος του προγράμματος;

.....

.....

.....

.....

2.α. Μετά το πέρας του προγράμματος, τα LED του ρομπότ θα είναι ανοιχτά ή κλειστά;

.....

2.β. Δεδομένης της απάντησης που δώσατε στο προηγούμενο υποερώτημα, αναδιατάξτε τις εντολές χειρισμού των LED ώστε να εκτελείται η αντίθετη λειτουργία.

3. α. Αλλάξτε τον μετρητή σε ένα από τους δύο βρόγχους έτσι ώστε να μην εκτελεστεί καμία φορά η δεύτερη επανάληψη.

3.β. Αντίστοιχα αλλάξτε το μετρητή έτσι ώστε η εξωτερική επανάληψη να εκτελεστεί μόνο μία φορά. Αναφέρετε και τους δύο τρόπους που αυτό μπορεί να γίνει.

Φάση 4^η-[Συνεργατική επίλυση προβλήματος] (30 λεπτά)

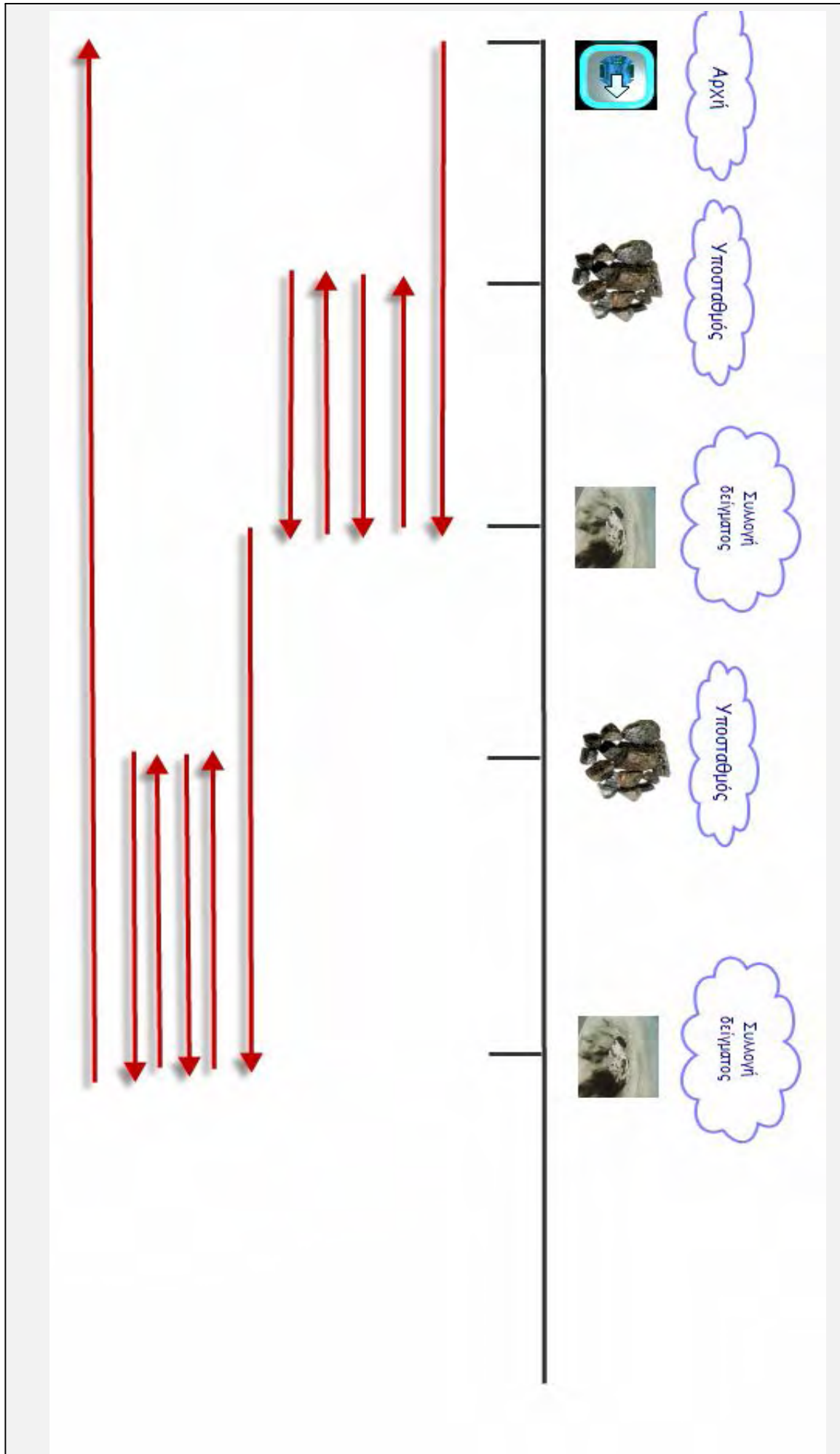
Μέσω αυτής της φάσης ο εκπαιδευτικός έχει ως στόχο να κατανοήσουν οι εκπαιδευόμενοι πληρέστερα τις εμφωλευμένες δομές επανάληψης καθώς και να δίνουν λύσεις σε προγραμματιστικά προβλήματα κάνοντας χρήση των ιδιοτήτων τους. Ζητείται η λύση ενός προβλήματος που χρησιμοποιεί την ίδια λογική με τη συλλογή δειγμάτων που χρησιμοποιήσαμε σε προηγούμενα παραδείγματα, μόνο που τώρα εισάγουμε την έννοια του υποσταθμού. Υπάρχουν δύο σημεία από τα οποία τα ρομπότ καλείται να συλλέξει δείγμα, με χρονική διαφορά 2 sec (δηλαδή αρχική θέση 0 sec, θέση λήψης πρώτου δείγματος -2 sec, θέση λήψης δεύτερου δείγματος -4 sec). Ανάμεσα από αυτές τις δύο θέσεις υπάρχουν δύο υποσταθμοί στο μισό της απόστασης (δηλαδή θέση πρώτου υποσταθμού -1 sec, θέση δεύτερου υποσταθμού -3 sec). Το ρομπότ καλείται να συλλέξει 2 δείγματα από την πρώτη θέση λήψης και να τα εναποθέσει στον υποσταθμό 1, ενώ στη συνέχεια θα συλλέξει δύο δείγματα από την θέση λήψης 2 και θα τα εναποθέσει στον υποσταθμό 2. Τέλος, θα επιστρέφει στην αρχική του θέση. Δίνεται στον μαθητή το φύλλο εργασίας 4, το οποίο καλείται να μελετήσει και να απαντήσει σε μια σειρά ερωτήσεων. Στο φύλλο παρέχεται αναλυτικά το σχήμα με τις λειτουργίες που πρέπει να εκτελέσει το ρομπότ.

Φύλλο εργασίας 4

Οι επιστήμονες ύστερα από μελέτες αποφάσισαν να δημιουργήσουν υποσταθμούς αποθήκευσης δειγμάτων ώστε να ελαχιστοποιήσουν την κατανάλωση καυσίμων και το χρόνο που απαιτείται για να ολοκληρωθεί η διαδικασία. Οι υποσταθμοί τοποθετούνται στο μέσο της διαδρομής, έτσι ώστε το ρομπότ να διανύει τη μισή απόσταση σε σχέση με πριν.

Πρόβλημα

Σε μια πλευρά του πλανήτη βρίσκονται δύο σημεία από τα οποία το ρομπότ καλείται να συλλέξει δείγματα. Τα σημεία αυτά είναι διατεταγμένα όπως φαίνεται στο παρακάτω σχήμα, ενώ το βάρος των δειγμάτων σε κάθε ένα από αυτά είναι διπλάσιο από το μέγιστο που μπορεί το ρομπότ να μεταφέρει. Είναι ακόμα διαθέσιμοι δύο υποσταθμοί στα σημεία 1 και 3 ενώ οι τοποθεσίες συλλογής δειγμάτων στα σημεία 2 και 4. Το ρομπότ ξεκινά από το σημείο 0. Γράψτε ένα πρόγραμμα που να υλοποιεί τα παραπάνω, βασιζόμενοι στη διαδικασία συλλογής δειγμάτων που έχετε διδαχθεί, χρησιμοποιώντας εμφωλευμένες δομές επανάληψης. Οι διαδοχικές κινήσεις που πρέπει να εκτελέσει το ρομπότ φαίνονται στα βελάκια του σχήματος.



Συνεδρία με τους μαθητές(10 λεπτά):

Κατά την διάρκεια της συνεδρίας οι μαθητές σχολιάζουν με τον διδάσκοντα το μάθημα που προηγήθηκε, διατυπώνουν κρίσεις, ερωτήσεις και απαντούν σε ερωτήσεις του που τον βοηθούν να το αξιολογήσει.

4.2.4 3η Διδακτική ώρα**Στόχοι μαθήματος**

Παραθέτουμε τους στόχους του μαθήματος βάσει της ταξινομίας του Bloom:

Οι μαθητές μετά το πέρας της διδασκαλίας θα πρέπει να είναι σε θέση:

Επίπεδο γνώσης

- Να γνωρίζουν τα διαφορετικά block εντολών ελέγχου της γραφικής διεπαφής του Scribbler.
- Να γνωρίζουν τους διαφορετικούς αισθητήρες του ρομπότ που μπορούν να αξιοποιήσουν σε block εντολών ελέγχου.
- Να γνωρίζουν πως μπορούν να συνδυαστούν λογικές συνθήκες με δομές επιλογής.

Επίπεδο κατανόησης

- Να αναγνωρίζουν τις περιστάσεις στις οποίες είναι κατάλληλη η δομή ελέγχου «αν» , «αν...τότε...αλλιώς».
- Να μπορούν να περιγράψουν τη σημασία των εντολών ελέγχου για τον έλεγχο των αισθητήρων
- Να μπορούν να περιγράψουν τη φιλοσοφία που διέπουν ορισμένα παραδείγματα δομών ελέγχου.
- Να μπορούν να ξεπεράσουν τυχόν παρανοήσεις για τη σειριακή εκτέλεση του προγράμματος.
- Να αντιλαμβάνονται τη ροή εκτέλεσης του προγράμματος και να μην θεωρούν ότι αυτό σταματά ή αρχίζει από την αρχή
- Να μπορούν να κατανοήσουν και να εξηγήσουν τη μέθοδο με την οποία εκτελούνται οι εντολές ελέγχου σε έτοιμα παραδείγματα κώδικα της γραφικής διεπαφής.

Επίπεδο εφαρμογής

- Να μπορούν να διαιρούν το πρόγραμμα σε μικρότερα τμήματα κώδικα και να αποσυνθέτουν τις δομές ελέγχου αλγοριθμικά.
- Να εφαρμόζουν εντολές ελέγχου της γραφικής διεπαφής για τον χειρισμό του ρομπότ και των αισθητήρων του.
- Να εφαρμόζουν εντολές ελέγχου για τον έλεγχο της ροής των προγραμμάτων τους.

Ανάλυση

- Να μπορούν να αναλύσουν τα πλεονεκτήματα και την αξία των δομών ελέγχου.

Προσανατολισμός μαθήματος**Σύντομη εισήγηση (10 λεπτά):**

Στη συγκεκριμένη φάση της διδασκαλίας μέσω μιας σύντομης εισαγωγής θα προσπαθήσουμε να δείξουμε στους μαθητές πως η λήψη αποφάσεων στη καθημερινή

ζωή μπορεί να απεικονιστεί με τη δομή ελέγχου στον προγραμματισμό, δηλαδή τη δυνατότητα του προγράμματος να αντιδρά διαφορετικά ανάλογα με την είσοδο που δέχεται. Προσπαθώντας να εξάγουμε από τους μαθητές τη λογική που βρίσκεται πίσω από τις δομές επιλογής, θα ζητήσουμε να αναφέρουν παραδείγματα που χρειάστηκε να λάβουν μια απόφαση. Για να καταλάβουν τη σύνταξη, θα αποδομήσουμε αυτά τα παραδείγματα, ώστε οι μαθητές να μπορούν να τα μετατρέψουν σε δομές επιλογής, κάνοντας χρήση των προτύπων «αν» , «αν...τότε...αλλιώς».

Φάση 1^η-[Τεχνική μαύρου κουτιού - 3 παραδείγματα] (20 λεπτά):

Οι μαθητές θα κληθούν να τρέξουν τρία παραδείγματα και να συμπληρώσουν το πρώτο φύλλο εργασίας. Στόχος των τριών δραστηριοτήτων είναι οι μαθητές να εξοικειωθούν με τη λογική των συνθηκών ελέγχου και μπορούν να δημιουργούν συνθήκες ελέγχου χρησιμοποιώντας την είσοδο των διάφορων αισθητήρων του ρομπότ. Για την επίτευξη αυτού του στόχου, παρέχεται αρχικά στους μαθητές, το φύλλο εργασίας με τις τρεις δραστηριότητες του δεύτερου μαθήματος. Χρησιμοποιώντας την τεχνική του μαύρου κουτιού, οι μαθητές καλούνται να τρέξουν τα προγράμματα, να παρατηρήσουν τη συμπεριφορά του ρομπότ και να συμπληρώσουν τα κενά στις προτάσεις που δίνονται.

Στην πρώτη δραστηριότητα ο εκπαιδευτικός καλείται να τοποθετήσει σε τρία σημεία εμπόδια με μέγεθος τουλάχιστον ίσο με το Scribbler, ώστε το ρομπότ χρησιμοποιώντας τη δομή ελέγχου να τα αποφύγει. Στη δεύτερη, τοποθετεί το ρομπότ σε ένα χαρτί πάνω στο οποίο έχουν σχεδιαστεί τρεις γραμμές που το ρομπότ πρέπει να ακολουθήσει, ενώ στην τρίτη κάνοντας χρήση ενός φακού, ζητά από τους μαθητές να φωτίσουν τους πάνω αισθητήρες του ρομπότ.

Δίνεται στους μαθητές βοηθητικό βίντεο με τη σύνταξη των δομών επιλογής στη γραφική διεπαφή του Scribbler, στο οποίο θα δίνεται έμφαση στη μη σειριακή εκτέλεση του προγράμματος όταν χρησιμοποιούνται δομές επιλογής. Επίσης δίνονται screenshots με την υλοποίηση των προγραμμάτων.

Φύλλο Εργασίας 1

Το ρομπότ επέστρεψε στα εργαστήρια της NASA μόλις ολοκληρώθηκε η πρώτη αποστολή. Η δεύτερη αποστολή λαμβάνει χώρα σε μια μη χαρτογραφημένη περιοχή του Άρη, γι αυτό οι επιστήμονες πρέπει να το προετοιμάσουν κατάλληλα, τρέχοντας κάποια πειραματικά προγράμματα που αφορούν την λειτουργικότητα των αισθητήρων του.

1η δραστηριότητα

Σε αυτήν, υλοποιείται ξανά η λειτουργία αποφυγής εμποδίων όπως και στην πρώτη αποστολή, μόνο που αυτή τη φορά οι επιστήμονες αποφασίζουν προς ποια κατεύθυνση θα κινηθεί το ρομπότ μετά την αποφυγή κι όχι αυθαίρετα όπως γινόταν στην πρώτη αποστολή.

Δραστηριότητες μαθητή

1. Οι επιστήμονες συνέταξαν την εφαρμογή “obstacle” η οποία υλοποιεί αποφυγή τριών εμποδίων. Φορτώστε στο ρομπότ την εφαρμογή “obstacle” (βρίσκεται στον φάκελο “Third lesson”), παρατηρήστε την συμπεριφορά του ρομπότ και συμπληρώστε τα κενά.

Αν το ρομπότ συναντήσει το πρώτο εμπόδιο, τότε _____.

Αν το ρομπότ συναντήσει το δεύτερο εμπόδιο, τότε _____.

Αν το ρομπότ συναντήσει το τρίτο εμπόδιο, τότε _____.

Εκφράσεις: στρίβει προς τα δεξιά , στρίβει προς τα αριστερά ,δεν κάνει τίποτα.

2. Αναγνωρίζεται στον κώδικά του προγράμματος τα μπλοκ που αποτελούν τις δομές επιλογής;

3. Τι θα συμβεί αν το ρομπότ δεν συναντήσει κανένα εμπόδιο κατά την πορεία του;

.....

4.Αν στην πρώτη δομή επιλογής αφαιρέσουμε την εντολή κίνησης προς τα αριστερά τι θα συμβεί;

.....

2η Δραστηριότητα

Αντίστοιχα με την πρώτη αποστολή όπου το ρομπότ έπρεπε να κινηθεί κατά μήκος των ορίων μιας χαρτογραφημένης περιοχής χρησιμοποιώντας τους κάτω αισθητήρες του, τώρα οι επιστήμονες επιθυμούν να εντοπίζει αν κάποια γραμμή βρίσκεται δεξιά ή αριστερά του και στη συνέχεια θα ανάβει κάποια από τα LED ειδοποιώντας τους.

Δραστηριότητες μαθητή

1.Οι επιστήμονες συνέταξαν την εφαρμογή “line” η οποία υλοποιεί την παραπάνω λειτουργικότητα. Φορτώστε στο ρομπότ την εφαρμογή “line” (βρίσκεται στον φάκελο “Third lesson”), παρατηρήστε την συμπεριφορά του ρομπότ και συμπληρώστε τα κενά.

Αν το ρομπότ συναντήσει την πρώτη γραμμή δεξιά του, τότε

_____.

Αν το ρομπότ συναντήσει την δεύτερη γραμμή αριστερά του, τότε

_____.

Αν το ρομπότ συναντήσει την τρίτη γραμμή δεξιά του , τότε

_____.

Εκφράσεις: ανάβει το δεξί LED, ανάβει το αριστερό LED, δεν κάνει τίποτα.

2. Αν η πρώτη γραμμή που συναντήσει το ρομπότ βρίσκεται αριστερά του, πως θα συμπεριφερθεί;

.....

3. Αν αλλάζατε θέσεις στις εντολές 1,2 τι θα συνέβαινε;

.....

.....

 4. Γράψτε ένα πρόγραμμα με μια δομή επιλογής, σύμφωνα με την οποία AN το ρομπότ συναντήσει μια γραμμή αριστερά του, TOTE θα ανάβει και τα τρία LEDs.

3η Δραστηριότητα

Οι επιστήμονες, έχουν υποψίες ότι το πρόγραμμα με το οποίο το ρομπότ ακολουθεί το φως του διαστημόπλοιου έχει εμφανίσει κάποια δυσλειτουργία γι αυτό και το επισκευάζουν.

Δραστηριότητες μαθητή

Φορτώστε στο ρομπότ το πρόγραμμα “light” (βρίσκεται στον φάκελο “Third lesson”) και σημαδέψτε με το φως του φακού τους πάνω αισθητήρες, ώστε να διαπιστώσετε την ορθότητα του προγράμματος των επιστημόνων.

1. Ποιά είναι τώρα η συμπεριφορά του ρομπότ;

.....

2. Διατυπώστε την λειτουργία που εκτελεί το ρομπότ με τη μορφή συνθήκης ελέγχου συμπληρώνοντας τα κενά.

AN _____, TOTE _____.

3. Ανοίξτε το πρόγραμμα και διορθώστε τα τυχόν λάθη των επιστημόνων, ώστε να εκτελείται η ίδια λειτουργικότητα με την πρώτη αποστολή.

Φάση 2^η-[Διερεύνηση- Σύντομα παραδείγματα] (30 λεπτά):

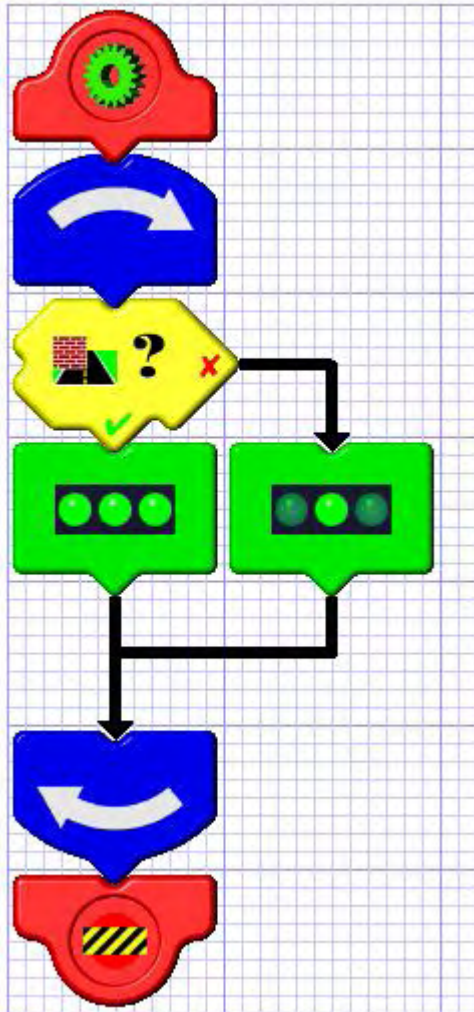
Μέσω της διερεύνησης σύντομων παραδειγμάτων, ο εκπαιδευτικός θα προετοιμάσει τους μαθητές ώστε να αναπτύξουν στη συνέχεια από μόνοι τους εφαρμογές που περιέχουν συνθήκες ελέγχου. Σκοπός των παραδειγμάτων είναι να αντιμετωπιστούν οι μαθησιακές δυσκολίες των μαθητών που μπορεί να οφείλονται είτε στις λογικές εκφράσεις που χρησιμοποιούνται στο σώμα του μπλοκ της εντολής είτε στην παρανόηση που έχουν για την σειριακή εκτέλεση του προγράμματος.

Ζητείται από το μαθητή να μελετήσει τα παραδείγματα που βρίσκονται στο δεύτερο φύλλο εργασίας. Με την προσέγγιση των πολλαπλών παραδειγμάτων οι μαθητές θα αποκτήσουν την απαιτούμενη γνωστική ευελιξία για την εφαρμογή των εντολών ελέγχου στην δημιουργία νέων προγραμμάτων.

Φύλλο εργασίας 2

Δραστηριότητα 1η

Μελετήστε και κατανοήστε το παρακάτω πρόγραμμα που συνέταξαν οι επιστήμονες και απαντήστε στις ερωτήσεις που ακολουθούν.



1. Ποια λειτουργία πιστεύετε ότι εκτελεί το ρομπότ με το παραπάνω πρόγραμμα;

.....

2. Ποια η διαφορά της συγκεκριμένης δομής ελέγχου σε σχέση με αυτές που έχετε συναντήσει ως τώρα;

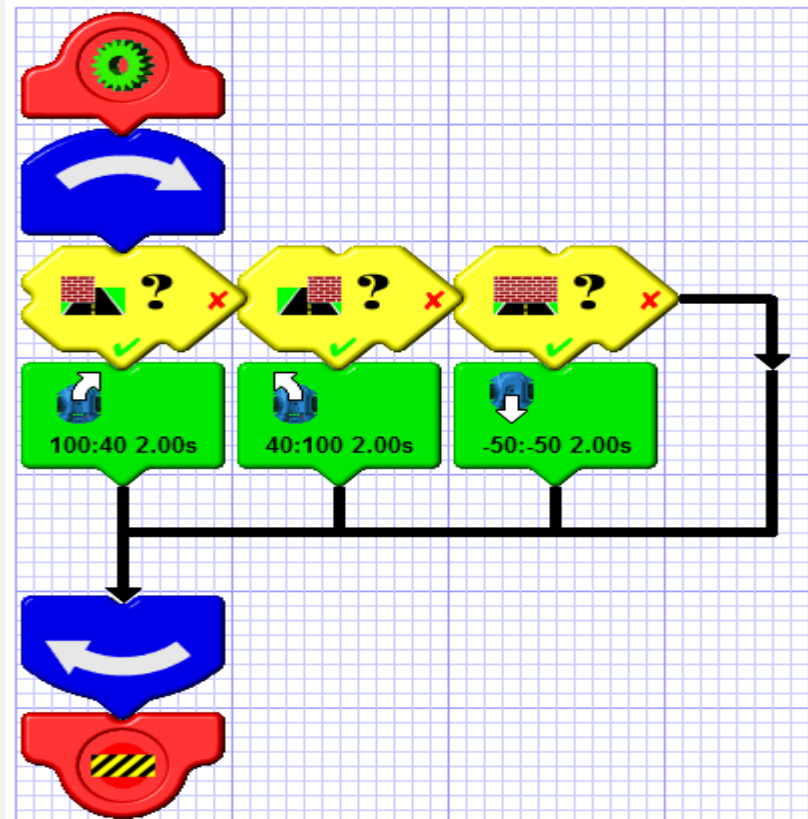
.....

3. Τι θα συμβεί αν το ρομπότ δε συναντήσει ποτέ στα αριστερά του εμπόδιο;

.....

4. Ποια η συμπεριφορά του αν βρει εμπόδιο στα δεξιά;

Δραστηριότητα 2η



1. Ποια λειτουργία πιστεύετε ότι εκτελεί το ρομπότ με το παραπάνω πρόγραμμα;

2. Αν το ρομπότ δε συναντήσει εμπόδιο στα αριστερά του, το πρόγραμμα θα τερματιστεί;

3. Υπάρχει πιθανότητα να εκτελεστούν και οι τρεις εντολές κίνησης;

4. Συμπληρώστε τα κενά:

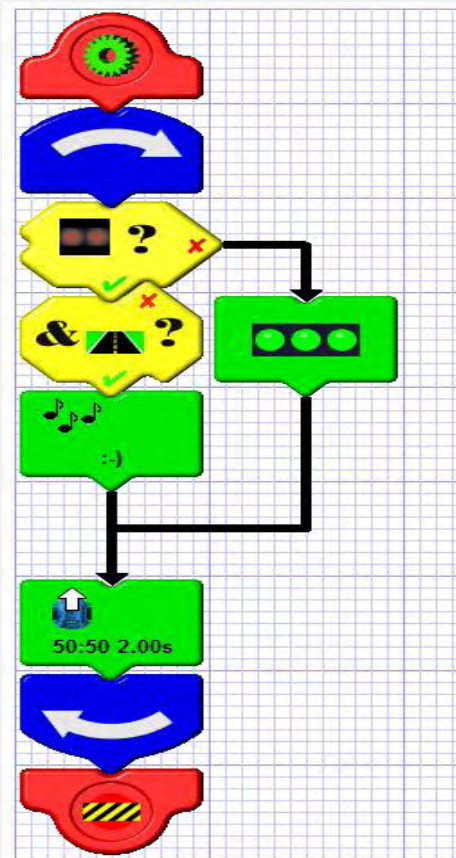
ΑΝ το ρομπότ συναντήσει εμπόδιο στα _____ ΤΟΤΕ θα κινηθεί

ΑΛΛΙΩΣ ΑΝ συναντήσει εμπόδιο στα _____ ΤΟΤΕ θα κινηθεί _____

ΑΛΛΙΩΣ ΑΝ συναντήσει εμπόδιο στα _____ ΤΟΤΕ θα κινηθεί _____

Εκφράσεις: αριστερά, δεξιά, πίσω.

Δραστηριότητα 3η



1. Ποια λειτουργία πιστεύετε ότι εκτελεί το ρομπότ με το παραπάνω πρόγραμμα;

.....

2. Συμπληρώστε τα παρακάτω πεδία ενεργειών για τις πιθανές αποτιμήσεις των συνθηκών ελέγχου.

1 ^η Συνθήκη	2 ^η Συνθήκη	Ενέργεια
ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	
ΑΛΗΘΗΣ	ΨΕΥΔΗΣ	
ΨΕΥΔΗΣ	ΑΛΗΘΗΣ	
ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	

3. Τροποποιήστε την ακολουθία εντολών, έτσι ώστε αν ικανοποιείται μία από τις δύο εντολές ελέγχου, να εκτελούνται και οι ενέργειες.

Φάση 3η-[Επίλυση προβλήματος] (40 λεπτά):

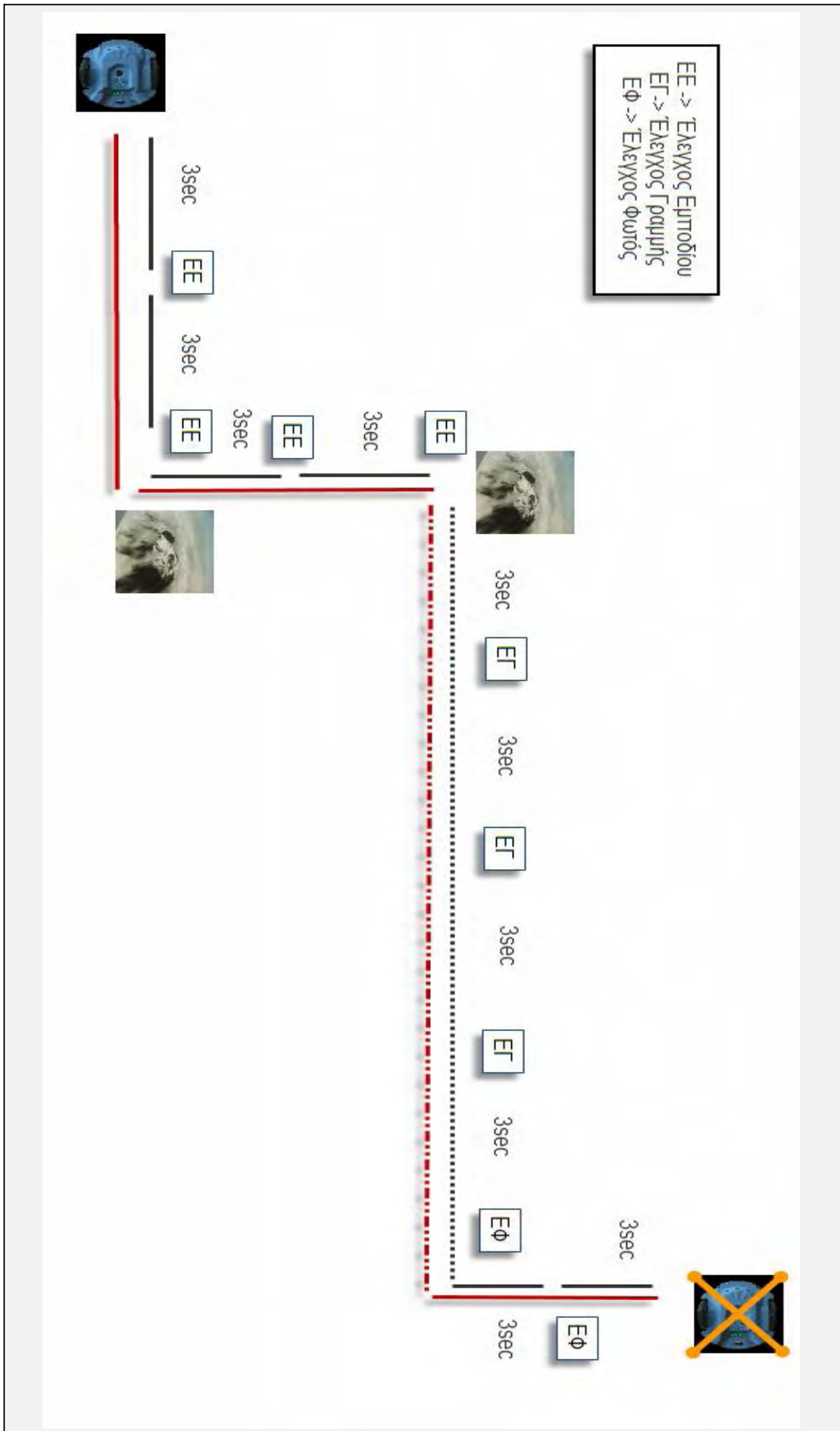
Μέσω αυτής της φάσης οι μαθητές καλούνται να συντάξουν ένα ολοκληρωμένο πρόγραμμα που περιλαμβάνει πολλές από τις δομές που έχουν διδαχθεί έως τώρα. Στόχος είναι οι μαθητές να αποδομήσουν το πρόβλημα, βλέποντας τις δομές ελέγχουν και επανάληψης από τις οποίες αποτελείται. Επίσης λόγω της σύνθετης δομής του προβλήματος θα δουν πως χωρίζεται ένα πρόγραμμα σε μικρότερα των οποίων η λύση είναι πιο απλή και ήδη γνωστή. Έτσι καλούνται να χρησιμοποιήσουν τις πρότερες γνώσεις που έχουν αποκτήσει κατά τη διάρκεια του παρόντος μαθήματος αλλά και του προηγούμενου. Δίνεται στους μαθητές μία λεπτομερής εξήγηση του προβλήματος (φύλλο εργασίας 3) και ένα σχεδιάγραμμα με τις ενέργειες που καλείται να εκτελέσει το ρομπότ. Ο εκπαιδευτικός πρέπει να διαθέτει ένα χαρτόνι μεγέθους 2Χ2 μέτρα, πάνω στο οποίο θα προσομοιωθεί το σχήμα της άσκησης.

Φύλλο εργασίας 3

Η τελευταία αποστολή του ρομπότ στον πλανήτη Άρη ήταν αποτυχημένη. Κατά την περιήγηση του στην επιφάνεια του πλανήτη, το ρομπότ προσέκρουσε σε δύο εμπόδια που του προκάλεσαν ανεπανόρθωτες βλάβες. Μετά την πρόσκρουση του στο πρώτο εμπόδιο, συνέχισε να κινείται χωρίς πρόβλημα, ενώ μετά την πρόσκρουση του στο δεύτερο προκλήθηκε διαρροή στα καύσιμα με αποτέλεσμα να σχηματιστεί στην επιφάνεια του πλανήτη μία γραμμή κατά μήκος της κατεύθυνσης που κινείται. Τέλος, ακινητοποιήθηκε και άναψε το φως έκτακτης ανάγκης ώστε να εντοπιστεί ευκολότερα.

Οι επιστήμονες αποφάσισαν να στείλουν ακόμα ένα Scribbler στον πλανήτη, ώστε να περισυλλέξει αυτό που καταστράφηκε. Η ακρίβεια της θέσης των εμποδίων, της γραμμής καυσίμων καθώς και του ίδιου του κατεστραμμένου ρομπότ δεν είναι μεγάλη, γι αυτό οι κινήσεις του ρομπότ-διασώστη πρέπει να είναι ιδιαίτερα προσεκτικές. Οι επιστήμονες θέλοντας να αποφύγουν την αποτυχία της προηγούμενης αποστολής, αλλά και προσπαθώντας να εντοπίσουν τη θέση του κατεστραμμένου αποφάσισαν να κάνουν χρήση των λειτουργιών αποφυγής εμποδίων, επακολούθησης γραμμής και εντοπισμού φωτεινού σημείου. Ένα σχεδιάγραμμα με όλα τα απαραίτητα στοιχεία και τις ενέργειες που πρέπει να κάνει το νέο ρομπότ ώστε να εντοπίσει το κατεστραμμένο, βρίσκονται στην εικόνα που ακολουθεί.

Γράψτε ένα πρόγραμμα ώστε το ρομπότ διασώστης να εντοπίζει το κατεστραμμένο. Κάντε χρήση των δομών επανάληψης όπου κρίνεται απαραίτητο, έτσι ώστε να χρησιμοποιηθεί ο ελάχιστος αριθμός εντολών. Όπως αναφέρθηκε, οι θέσεις δεν είναι απαραίτητα ακριβείς, γι αυτό και απαιτούνται τόσοι έλεγχοι.



Επίλογος μαθήματος (10 λεπτά):

Κατά την διάρκεια του επιλόγου προβάλλεται βίντεο με ρομποτικές εφαρμογές από τον κινηματογράφο και τις τέχνες γενικότερα. Οι μαθητές μετά την παρακολούθηση του καλούνται ,έχοντας αποκτήσει πλέον προσωπική εμπειρία, να καταθέσουν την διαμορφωθείσα άποψη τους για τις ρομποτικές εφαρμογές, τόσο στην εκπαίδευση όσο και στις εκφάνσεις της ανθρώπινης ζωής γενικότερα.

Συνεδρία με τους μαθητές(10 λεπτά):

Κατά την διάρκεια της συνεδρίας οι μαθητές σχολιάζουν με τον διδάσκοντα το μάθημα που προηγήθηκε, διατυπώνουν κρίσεις, ερωτήσεις και απαντούν σε ερωτήσεις του που τον βοηθούν να το αξιολογήσει.

5.Εφαρμογή διδασκαλίας και αξιολόγηση

5.1 Εφαρμογή διδασκαλίας

Το διδακτικό πλάνο που αναλύθηκε προηγουμένως, εφαρμόστηκε για τη διδασκαλία προγραμματισμού σε τρεις ομάδες τεσσάρων ενήλικων μαθητών, με διαφορετικό επίπεδο γνώσης στο αντικείμενο του προγραμματισμού (αρχάριων, στοιχειωδώς καταρτισμένων και έμπειρων). Χρειάστηκαν 2 διδακτικές ώρες για κάθε μάθημα του πλάνου, ενώ τα μαθήματα έλαβαν χώρα σε ιδιωτικό χώρο, διαμορφωμένο έτσι ώστε να προσομοιώνει ένα εργαστήριο προγραμματισμού, σε Βόλο, Λάρισα και Καστοριά.

Κάθε μαθητής ενός γκρουπ διέθετε προσωπικό υπολογιστή (laptop) εξοπλισμένο με το υλικό που περιγράφεται αλλά και απαιτείται για τη διεξαγωγή των μαθημάτων (γραφικό περιβάλλον της Parallax, εκφωνήσεις λύσεις και βοηθήματα, κτλ) ενώ στη διάθεση των διδασκόντων βρισκόταν ένας επιτραπέζιος υπολογιστής που διευκόλυνε την προβολή οπτικοακουστικού υλικού και την επίλυση τυχόν αποριών των μαθητών μέσω επίδειξης. Χρησιμοποιήθηκαν 2 ρομπότ Scribbler της Parallax για την εφαρμογή του διδακτικού πλάνου.



Καστοριά



Βόλος



Λάρισα

5.1.1 Προφίλ ομάδων

Αρχάριοι στον προγραμματισμό

Όνοματεπώνυμο (Αρχικά)	Φύλο	Ηλικία	Επίπεδο εκπαίδευσης	Γνώσεις σε προγραμματισμό
N.M.	A	23	Δευτεροβάθμια	Στοιχειώδεις γνώσεις χειρισμού Η/Υ

Δ.Π.	A	22	Δευτεροβάθμια	Στοιχειώδεις γνώσεις χειρισμού Η/Υ
M.N.	A	22	Τριτοβάθμια	Στοιχειώδεις γνώσεις χειρισμού Η/Υ
P.E.	Θ	21	Τριτοβάθμια	Στοιχειώδεις γνώσεις χειρισμού Η/Υ

Το τμήμα των αρχαρίων περιλάμβανε μαθητές που δεν είχαν έρθει ποτέ ξανά σε επαφή με προγραμματιστικές έννοιες, αλλά δεν αντιμετώπιζαν προβλήματα χειρισμού Η/Υ και Windows. Τα μαθήματα διεξήχθησαν στις 5, 6 και 7 Αυγούστου του 2009 στην Καστοριά. Αποτέλεσε την ομάδα εκείνη που έχοντας μηδαμινή εμπειρία στον προγραμματισμό, θα έπρεπε να οικοδομήσει λογική επίλυσης προγραμματιστικών προβλημάτων μέσω της εφαρμογής της διδασκαλίας.

Εξοικειωμένοι με τις βασικές έννοιες του προγραμματισμού

Όνοματεπώνυμο (Αρχικά)	Φύλο	Ηλικία	Επίπεδο εκπαίδευσης	Γνώσεις σε προγραμματισμό
N.M.	A	27	Τριτοβάθμια	Χρήση φύλλων εργασίας EXCEL
Δ.Π.	Θ	19	Τριτοβάθμια	Μαθήματα Matlab (στα πλαίσια οικονομικών σπουδών)
M.N.	A	24	Δευτεροβάθμια	Ερασιτεχνική ενασχόληση με C/C++
B.K.	Θ	20	Τριτοβάθμια	Μαθήματα HTML σε εργαστήριο σπουδών

Σε αυτή την ομάδα των μαθητών εντάχθηκαν άτομα με γνώσεις προγραμματισμού αρκετές ώστε να τους βοηθούν να χειρίζονται λογισμικό προς εκπλήρωση διάφορων καθηκόντων. Αυτές οι γνώσεις δεν είναι αρκετές ώστε να σχηματίσει κάποιος μια πλήρη προγραμματιστική λογική και συχνά τον οδηγεί σε σύγχυση σχετικά με βασικές έννοιες του προγραμματισμού. Τα μαθήματα διεξήχθησαν στις 24,25 και 26 Αυγούστου του 2009 στη Λάρισα, με σκοπό να διαπιστωθεί κατά πόσο μια τέτοια ομάδα ανθρώπων, με ελλιπή γνώση στον προγραμματισμό, μπορεί να αναδομήσει ορθά πλέον προγραμματιστική σκέψη.

Έμπειροι προγραμματιστές

Όνοματεπώνυμο (Αρχικά)	Φύλο	Ηλικία	Επίπεδο εκπαίδευσης	Γνώσεις σε προγραμματισμό
B. X.	A	24	Τριτοβάθμια	Φοιτητής στο Τμήμα Μηχ. Η/Υ του Παν. Θεσσαλίας

Θ.Ο.	A	22	Τριτοβάθμια	Φοιτητής στο Τμήμα Πληροφορικής του ΤΕΙ Λάρισας
Κ.Π.	A	26	Τριτοβάθμια	Μηχανικός Η/Υ
Π.Θ.	A	25	Τριτοβάθμια	Προγραμματιστής (απόφοιτος ΙΕΚ)

Τα μαθήματα για το τμήμα των έμπειρων χρηστών διεξήχθησαν στις 29,30 και 31 Ιουλίου του 2009 στον Βόλο. Οι μαθητές αυτού του τμήματος διέθεταν εμπειρία σε κλασικές προγραμματιστικές γλώσσες (C/C++, Java, Pascal) αλλά και μια στοιχειώδη εξοικείωση στις έννοιες της ρομποτικής. Πρωταρχικός στόχος της διεξαγωγής των μαθημάτων σε μια τέτοια έμπειρη ομάδα χρηστών δεν υπήρξε η μεταλαμπάδευση της γνώσης, αλλά η σύγκριση ενός τέτοιου πειραματικού διδακτικού πλάνου με την πρότερη εμπειρία τους σε μαθήματα προγραμματισμού που διεξάγονται με τον παραδοσιακό τρόπο εργαστηρίων-διαλέξεων και η ακόμη αυστηρότερη κριτική στον διδάσκοντα και στο διδακτικό πλάνο που κατάρτισε.

5.2 Μέθοδοι αξιολόγησης

Σε ένα εγχείρημα όπως η διδασκαλία του προγραμματισμού με τη χρήση ρομποτικών εφαρμογών, η αξιολόγηση της διδασκαλίας από τους μαθητές έχει βαρύνουσα σημασία. Λαμβάνοντας υπόψη το παραπάνω, επιχειρήθηκε μια όσο το δυνατόν πληρέστερη αξιολόγηση με εναλλακτικές μεθόδους, έτσι ώστε να εξαχθεί ένα σαφές και ακριβές συμπέρασμα, αλλά και να αξιοποιηθούν τα ιδιαίτερα χαρακτηριστικά κάθε ομάδας.

Τα θέματα προς αξιολόγηση που αποτέλεσαν τη βάση για την επιλογή των αντίστοιχων «εργαλείων» είναι:

- η επιτυχία ή μη των μαθησιακών στόχων και η συμπεριφορά των συμμετεχόντων στις τυπικές μαθησιακές δυσκολίες.
- η επιτυχία ή μη
 - των δραστηριοτήτων του διδακτικού πλάνου.
 - της ροής του και της εμπειρίας που προκαλεί .
- η αντιλαμβανόμενη (perceived) ικανοποίηση από τη συμμετοχή, την εμπειρία και το γνωστικό αποτέλεσμα και τη διάθεση συμμετοχής σε αντίστοιχες δραστηριότητες.
- η αντιλαμβανόμενη (perceived) από τους συμμετέχοντες διαφοροποίηση της συγκεκριμένης διδασκαλίας από τις παραδοσιακές προσεγγίσεις.

Με δεδομένα τα παραπάνω θέματα προς αξιολόγηση επιλέχθηκαν οι εξής μέθοδοι:

- Αναλυτική καταγραφή της συμπεριφοράς κάθε μαθητή κατά την διάρκεια των μαθημάτων. Τα στοιχεία αυτά συλλέχθηκαν ώστε να πραγματοποιηθεί μια ποιοτική αξιολόγηση της διδασκαλίας. Για την αποτίμηση της συμπεριφοράς των μαθητών ο διδάσκων κράτησε προσωπικές σημειώσεις σε όλη την διάρκεια των μαθημάτων και για κάθε μαθητή ξεχωριστά. Οι σημειώσεις αυτές συγκεντρώθηκαν και οργανώθηκαν έτσι ώστε να μπορεί αρχικά να περιγραφεί η εφαρμογή της διδασκαλίας και κατόπιν να εξαχθούν ποιοτικά συμπεράσματα τόσο για κάθε ώρα μαθήματος, όσο και για κάθε τμήμα ξεχωριστά.
- Συνεδρίες με τους συμμετέχοντες κάθε ομάδας μετά το τέλος της διδασκαλίας κάθε μαθήματος. Σε αυτές ο διδάσκων επιχείρησε σε μια

διάρκεια 10 λεπτών να αφουγκραστεί τυχόν προβλήματα και δυσκολίες των συμμετεχόντων χωρίς να τον περιορίζει κάποια συγκεκριμένη δομή αλλά με ελεύθερο θεματολόγιο. Δεδομένης της αυστηρής δομής των υπόλοιπων μορφών αξιολόγησης, με τις συνεδρίες επιδιώχθηκε όχι μόνο να προληφθεί οποιοδήποτε πρόβλημα των μαθητών πριν την συνέχιση της διδασκαλίας, αλλά και να δοθεί σε αυτούς ο χρόνος να μιλήσουν ελεύθερα για οτιδήποτε έχει να κάνει με το μάθημα.

- Γενικό ερωτηματολόγιο εμπειρίας (Βλ. Παράρτημα Β), το οποίο συντάχθηκε από τον διδάσκοντα και παραδόθηκε στους συμμετέχοντες όλων των ομάδων προς συμπλήρωση στο τέλος του τρίτου μαθήματος [32].
- Γενικό ερωτηματολόγιο αξιολόγησης του εκπαιδευτικού (Βλ. Παράρτημα Β), του περιβάλλοντος και των συνθηκών κάτω από τις οποίες διεξήχθησαν τα μαθήματα. Παραδόθηκε σε όλα τα γκρουπ των μαθητών μετά την συμπλήρωση του ερωτηματολογίου εμπειρίας [32].
- Ειδικό ερωτηματολόγιο (Βλ. Παράρτημα Β) που απευθύνθηκε αποκλειστικά στην ομάδα των έμπειρων μαθητών. Με αυτήν την αξιολογική δομή επιχειρήθηκε μια σύγκριση της εμπειρίας των συμμετεχόντων σε παραδοσιακά μαθήματα προγραμματισμού με αυτή που αποκόμισαν από την παρούσα διδασκαλία. Επιπρόσθετα, ζητήθηκε η άποψη τους για τη δομή, το ύψος, την έκταση της ύλης, την ευχρηστία των εργαλείων που χρησιμοποιήθηκαν αλλά και μια μικρή καταγραφή της πρότερης τυχόν άσχημης εμπειρίας τους με τον προγραμματισμό. Παραδόθηκε στο γκρουπ των πεπειραμένων χρηστών στο τέλος της τρίτης ώρας, μετά τη συμπλήρωση των υπόλοιπων ερωτηματολογίων.

5.3 Αποτελέσματα αξιολόγησης

5.3.1 Ποιοτική ανάλυση στην εφαρμογή της διδασκαλίας

Έμπειροι προγραμματιστές

1^ο μάθημα

Η διδασκαλία με τους έμπειρους προγραμματιστές αντιμετωπίστηκε εξ αρχής ως η δυσκολότερη των τριών γκρουπ, λαμβάνοντας υπόψη πως η συναναστροφή τους με το αντικείμενο του προγραμματισμού θα τους έκανε πιο απαιτητικούς. Με την έναρξη του πρώτου μαθήματος και όπως αναφέρεται στο διδακτικό πλάνο, προβλήθηκε ένα εισαγωγικό βίντεο με σύγχρονες ρομποτικές εφαρμογές. Οι συμμετέχοντες κλήθηκαν να απαντήσουν σε μια σειρά ερωτήσεων σχετικά με τη ρομποτική κατά τη διάρκεια της προβολής. Λόγω της επαγγελματικής ενασχόλησής τους με την πληροφορική, έδειξαν να γνωρίζουν πολλές από τις ρομποτικές εφαρμογές καθώς και τις δυνατότητες τους σήμερα. Η προβολή του βίντεο τους έξασε την περιέργεια για το ρομπότ που θα χρησιμοποιούσαν στη διδασκαλία, ενώ αποτέλεσε και μια ευχάριστη είσοδο στην επόμενη φάση, αυτή των ερωταποκρίσεων.

Σε αυτήν, οι έμπειροι προγραμματιστές έδειξαν πως εκτιμούν ως ιδιαίτερα χρήσιμη την ρομποτική, ειδικά ως συνοδευτικό μέσο σε διδακτικές προσεγγίσεις. Στη συνέχεια, ο διδάσκων έδωσε στους συμμετέχοντες τα δύο Scribbler που χρησιμοποιήθηκαν στη διδασκαλία ώστε να εξοικειωθούν με αυτά. Οι μαθητές, ανά δύο, κράτησαν για λίγα λεπτά στα χέρια τους το ρομπότ, ψηλαφίζοντας τα και δείχνοντας ενθουσιασμό και

περιέργεια. Είναι βέβαιο πως οι προσδοκίες τους για μια εξελιγμένη ρομποτική εφαρμογή δεν επαληθεύτηκαν, αλλά η προσδοκία τους για τις εφαρμογές στις οποίες επρόκειτο να χρησιμοποιηθεί το Scribbler ήταν εμφανής.

Συνεχίζοντας σύμφωνα με το διδακτικό πλάνο, οι μαθητές κάθισαν στους υπολογιστές του εργαστηρίου, όπου και τους παραδόθηκε το πρώτο φύλλο εργασίας και τους ζητήθηκε να ασχοληθούν με την πρώτη δραστηριότητα. Οι μαθητές των οποίων οι υπολογιστές ήταν τοποθετημένοι δίπλα-δίπλα θα αποτελούσαν από αυτή τη στιγμή μια ομάδα για την εκπλήρωση οποιωνδήποτε συνεργατικών ασκήσεων. Μετά την ενεργοποίηση της δυνατότητας αποφυγής εμποδίων του Scribbler, ζητήθηκε από τους μαθητές να προχωρήσουν στη συμπλήρωση των υπόλοιπων δραστηριοτήτων του φύλλου εργασίας. Κανείς από τους μαθητές δεν παρερμήνευσε αυτό που είδε από την ρομποτική εφαρμογή, έτσι οι δραστηριότητες 2-5 συμπληρώθηκαν σωστά στο σύνολο των μαθητών. Τρεις δε εξ αυτών, δήλωσαν πως έχουν έρθει σε επαφή και με αισθητήρες, σε άλλες επιστημονικές τους δραστηριότητες.

Η διδασκαλία συνεχίστηκε με την παράδοση του δεύτερου φύλλου εργασίας στους μαθητές και ζητώντας τους αρχικά να διαβάσουν την πρώτη. Ακολούθως, τοποθετήθηκαν δύο χαρτόνια σύμφωνα με την περιγραφή του διδακτικού πλάνου, ώστε πάνω σε αυτά να ενεργοποιηθεί η δυνατότητα του Scribbler να ακολουθεί γραμμές. Η διαδικασία αυτή ολοκληρώθηκε με επιτυχία και με τους μαθητές να ενθουσιάζονται που ανακαλύπτουν σταδιακά τις δυνατότητες της ρομποτικής εφαρμογής. Οι δραστηριότητες που ακολούθησαν δεν προβλημάτισαν κανέναν εκ των μαθητών, έτσι σε σύντομο χρονικό διάστημα ο διδάσκων τους κάλεσε να προχωρήσουν στην άσκηση εμπέδωσης του πλάνου. Ακόμη και στην ερώτηση με συγκαλυμμένη την μαθησιακή δυσκολία του ατέρμονα βρόγχου κανείς δεν απάντησε λανθασμένα, επαληθεύοντας έτσι το υψηλό επίπεδο της τάξης.

Η άσκηση εμπέδωσης που ακολούθησε, αντιμετωπίστηκε από τους μαθητές ως δημιουργικό παιχνίδι. Οι διαδρομές που σχεδίασαν στο χαρτί δυσκόλεψαν αρκετά το ρομπότ, ενώ αναπτύχθηκε και ένας άτυπος ανταγωνισμός μεταξύ των μαθητών, σχετικά με το ποιος θα καταφέρει να το αποπροσανατολίσει περισσότερο.

Το επίπεδο δυσκολίας του τρίτου φύλλου εργασίας δεν θα μπορούσε να αποτελέσει πρόβλημα για τους έμπειρους προγραμματιστές, όπως και συνέβη. Παρότι οι περισσότεροι είχαν ήδη υποψιαστεί τις δυνατότητες του ρομπότ, έδειξαν ενθουσιασμένοι τόσο με την εύρεση του φωτεινότερου σημείου στο δωμάτιο, όσο και με την κατεύθυνση του με τη βοήθεια ενός φακό. Το χαλαρό κλίμα που επικράτησε στην προηγούμενη άσκηση εμπέδωσης συνεχίστηκε και εδώ, με τους έμπειρους προγραμματιστές να δηλώνουν πως βρίσκουν το μάθημα μέχρι στιγμής διασκεδαστικό.

Ξεκινώντας την ενασχόληση με το τέταρτο φύλλο εργασίας, οι μαθητές επέστρεψαν στους υπολογιστές με μια μικρή απογοήτευση, ξέροντας πως τους περιμένει η πρώτη επαφή με το γραφικό περιβάλλον του Scribbler και αρκετή δουλειά. Μετά την ανάγνωση του φύλλου εργασίας, τρεις εξ αυτών ζήτησαν διευκρινιστικά σχόλια από τον διδάσκοντα, ώστε να αποσαφηνίσουν τυχόν προβλήματα με την κίνηση του ρομπότ. Πέραν αυτών των ερωτήσεων δεν εμφανίστηκε κανένα άλλο πρόβλημα. Οι έμπειροι προγραμματιστές απάντησαν σωστά και σύντομα στο σύνολο των ερωτήσεων του πρώτου φύλλου, αφήνοντας τον διδάσκοντα απόλυτα ικανοποιημένο. Τα σχόλια τους για τη γραφική διεπαφή, καθώς και οι αντιδράσεις τους την ώρα που δούλευαν πάνω σε

αυτή, οδηγούσαν στο συμπέρασμα πως περίμεναν μια πιο σύνθετη εφαρμογή και πως με την παρούσα δεν επρόκειτο να αντιμετωπίζουν πρόβλημα.

Η διαδικασία του πέμπτου φύλλου εργασίας κύλησε το ίδιο ομαλά. Σε ερώτηση του διδάσκοντα αν έχουν ξαναχρησιμοποιήσει εντολές παύσης τρεις μαθητές απάντησαν καταφατικά. Συνέταξαν σωστά τα προγράμματα τους, σημείωσαν αρκετά κοντά στο χαρτί τα σημεία που περίμεναν να σταματήσει το ρομπότ και μετά την εφαρμογή των προγραμμάτων στο ρομπότ επιβράβευσαν τον συμμαθητή τους που κατάφερε να είναι ακριβέστερος στην πρόβλεψη του.

Η συνεργασία των μαθητών για την επίλυση του έκτου φύλλου εργασίας κύλησε άψογα, με τον ομολογουμένως διασκεδαστικό ανταγωνισμό όμως μεταξύ των μαθητών να συνεχίζεται. Η ομάδα που ολοκλήρωσε πρώτη το ζητούμενο της πρώτης δραστηριότητας ζήτησε να έχει το προνόμιο να επιλέξει αυτή σημείο προς εξερεύνηση στο χαρτί, όπως και έγινε. Η δεύτερη ομάδα, ολοκλήρωσε επίσης σωστά το πρόγραμμά της, κλήθηκε όμως να εντοπίσει το σημείο που επέλεξε η πρώτη ομάδα, όχι και τόσο εύκολη δραστηριότητα, μιας και τοποθετήθηκε σε όσο το δυνατόν δυσκολότερη περιοχή. Μετά από τρεις προσπάθειες το σημείο εντοπίστηκε σωστά και η διδασκαλία του πρώτου μαθήματος είχε ολοκληρωθεί.

Η συνολική εντύπωση του διδάσκοντα από το πρώτο μάθημα ήταν πως οι συμμετέχοντες, διαθέτοντας ήδη αρκετές προγραμματιστικές γνώσεις, αντιμετώπισαν το πρώτο δίωρο ως μια διασκεδαστική εμπειρία, χωρίς αυτό βέβαια να έχει αρνητικές επιπτώσεις στο διδακτικό έργο. Οι μαθητές σε καμία περίπτωση δεν έδειξαν αγχωμένοι ή προβληματισμένοι, αντιθέτως αναπτύχθηκε άμυλα μεταξύ τους δίνοντας τους έτσι ένα παραπάνω κίνητρο να ολοκληρώσουν σωστά και γρήγορα τις υποχρεώσεις του μαθήματος. Έννοιες όπως δομή του προγράμματος(αρχή και τέλος), εντολές εκτέλεσης και παύσης έγιναν άμεσα αντιληπτές, ενώ στο σύνολό τους χειρίστηκαν την γραφική διεπαφή με χαρακτηριστική ευκολία.

2^ο μάθημα

Οι απαιτήσεις της σύντομης εισήγησης του δεύτερου μαθήματος με το τμήμα των εμπειριών προγραμματιστών. Ένας μικρός αριθμός παραδειγμάτων από την καθημερινή ζωή στάθηκε αρκετός να τους κάνει να καταλάβουν πως επρόκειτο να ασχοληθούμε με την δομή επανάληψης, την οποία όλοι γνώριζαν. Το μόνο που ισχυρίστηκαν πως περίμεναν να διευκρινιστεί ήταν η μορφή που θα έχει αυτή στην γραφική διεπαφή του Scribbler και κατά πόσο υποστηρίζονται όλες οι περιπτώσεις δομών που ήδη είχαν χρησιμοποιήσει προγραμματίζοντας.

Το πρώτο φύλλο εργασίας δεν θα μπορούσε να είναι δυσνόητο για προγραμματιστές τέτοιου επιπέδου. Ακολουθώντας τις δραστηριότητες, οι συμμετέχοντες άλλαξαν σωστά το όρισμα της επαναληπτικής δομής, δεν έπεσαν στην παγίδα του ατέρμονα βρόγχου στη δεύτερη δραστηριότητα αλλά απάντησαν με κάποια διστακτικότητα και σχολίασαν επαρκώς τα αποτελέσματα που θα απέφερε μια τροποποίηση του προγράμματος. Ζητήθηκε ένα πρώτο σχόλιο για την υλοποίηση της δομής επανάληψης, στο οποίο οι μαθητές απάντησαν πως την εκτιμούσαν ως ιδιαίτερα απλή και ευδιάκριτη μέσα σένα πρόγραμμα, αλλά με περιορισμένες επιλογές.

Η ενασχόληση τους με το δεύτερο φύλλο εργασίας αποκάλυψε μια μικρή αμηχανία στην τέταρτη δραστηριότητα. Ενώ όλοι τους είχαν ασχοληθεί με πολύ πιο σύνθετα

προγράμματα στο παρελθόν, χρειάστηκε η παρέμβαση του διδάσκοντα ώστε να αποσαφηνιστούν οι απαιτήσεις του ερωτήματος. Μόλις αυτές ολοκληρώθηκαν, όλοι τους συνέταξαν σωστά το πρόγραμμα-απάντηση που τους ζητήθηκε, ενώ επιλέχθηκε τυχαία μια από τις λύσεις και φορτώθηκε στο ρομπότ για να μπορέσουν οι μαθητές να δουν το αποτέλεσμα. Οι υπόλοιπες δραστηριότητες δεν προβληματίσαν κανένα μαθητή.

Η διδασκαλία συνεχίστηκε με το τρίτο φύλλο εργασίας. Ο διδάσκων ζήτησε από τους μαθητές να αναγνωρίσουν την σύνθετη επαναληπτική δομή που εμφανίστηκε σε αυτό το φύλλο και όλοι τους ανεξαιρέτως αντιλήφθηκαν πως επρόκειτο για μια εμφωλευμένη δομή επανάληψης. Έχοντας δουλέψει αρκετά πάν σε τέτοιες σύνθετες δομές, βρήκαν ιδιαίτερα εύκολες τις περισσότερες δραστηριότητες, με μόνη παραφωνία το τελευταίο ερώτημα του φύλλου. Μόνο δύο εκ των μαθητών απάντησαν σωστά στο συγκεκριμένο ερώτημα, κάνοντας την εξωτερική επανάληψη να εκτελεστεί μόνο μια φορά με δύο διαφορετικούς τρόπους. Οι άλλοι δύο μαθητές έκαναν χρήση μόνο του ατέρμονα βρόγχου και όταν έμαθαν τον δεύτερο τρόπο απογοητεύτηκαν με τον εαυτό τους που δεν απάντησαν το προφανές.

Η διδασκαλία του δεύτερου μαθήματος έκλεισε με την συνεργασία των μαθητών πάνω στο τέταρτο φύλλο εργασίας που απαιτούσε την ενσωμάτωση όλων όσων είχαν διδαχθεί μέχρι την δεδομένη στιγμή. Έχοντας πλέον εξοικειωθεί με τον τρόπο που το Scribbler χρησιμοποιεί επαναληπτικές δομές, δεν δυσκολεύτηκαν να συνεργαστούν άψογα και να ολοκληρώσουν επιτυχώς το ζητούμενο πρόγραμμα. Η εκτέλεση των προγραμμάτων στο ρομπότ έδωσε και πάλι τροφή για πειράγματα μεταξύ των ομάδων.

Ως ένα γενικότερο σχόλιο, θα μπορούσαμε να πούμε πως οι έμπειροι μαθητές εμφανίστηκαν στις διδακτικές υποχρεώσεις ακριβώς όπως αναμενόταν: ικανότατοι. Η παρέμβαση του διδάσκοντα χρειάστηκε μόνο για την επίλυση πρακτικών και διαδικαστικών ζητημάτων. Ο χρόνος για την ολοκλήρωση των δραστηριοτήτων αποδείχθηκε μεγάλος για αυτούς, ενώ το ρομπότ δεν έδειχνε πλέον να ικανοποιεί πλήρως τις απαιτήσεις τους.

3^ο μάθημα

Η εισήγηση κατά την έναρξη του τρίτου μαθήματος με την ομάδα των έμπειρων προγραμματιστών περιλάμβανε από απλά παραδείγματα με συνθήκες όπου πρέπει να ληφθούν αποφάσεις στην καθημερινή ζωή μέχρι και σύνθετες δομές ελέγχου που γνώριζαν οι μαθητές σε κλασικά προγραμματιστικά εργαλεία. Η προσμονή τους για την χρησιμοποίηση δομών ελέγχου φάνηκε αρκετά μεγάλη, καθώς όπως δήλωσε ένας συμμετέχων, με τη σύμφωνη γνώμη των υπολοίπων, αδημονούσαν να δώσουν στο ρομπότ μια στοιχειώδη νοημοσύνη. Ένα ακόμη ζήτημα για την ομάδα των έμπειρων προγραμματιστών υπήρξε και η απεικόνιση της δομής ελέγχου και κατά πόσο θα κάλυπτε τις απαιτήσεις τους, μιας και από την δομή επανάληψης δεν έμειναν πλήρως ικανοποιημένοι.

Η άριστη απόδοση των μαθητών στα δύο πρώτα φύλλα εργασίας δεν χωρά περαιτέρω σχολιασμό. Συμπλήρωσαν όλες τις δραστηριότητες με σχετική άνεση και διατύπωσαν ερωτήσεις μόνο για κάποιες δομές ελέγχου της γραφικής διεπαφής και ποια λειτουργικότητα του ρομπότ ελέγχουν αυτές. Αυθόρμητα σχολίασαν μεταξύ τους πως η επαλήθευση συνθηκών ελέγχου με την βοήθεια μιας ρομποτικής εφαρμογής θα μπορούσε να τους είχε γλυτώσει από αρκετές ώρες αποσφαλμάτωσης σε γλώσσες χαμηλού επιπέδου.

Η εμπειρία της συγκεκριμένης μαθητικής ομάδας φάνηκε κυρίως στο τελευταίο φύλλο εργασίας. Η συνδυαστική δραστηριότητα αυτού αντιμετωπίστηκε ως ένα ολοκληρωμένο project, αποδομήθηκε με χαρακτηριστική ευκολία σε μικρότερα υποπροβλήματα τις λύσεις των οποίων είχαν ήδη διδαχθεί και κατανοήσει. Ακόμη, εκδηλώθηκε ανοιχτά μια προτίμηση σε τέτοιου είδους δραστηριότητες, με παρόμοια δομή και αντίστοιχο επίπεδο δυσκολίας.

Κλείνοντας την διδασκαλία με τους έμπειρους προγραμματιστές και βασιζόμενοι στις αντιδράσεις τους αλλά και στην συνολική εντύπωση του διδάσκοντα θα λέγαμε πως παρά το ιδιαίτερα υψηλό επίπεδο κατάρτισής τους στον προγραμματισμό θα επιθυμούσαν να παρατείνουμε και άλλο την διδασκαλία με πολύ πιο σύνθετες εντολές. Η ενασχόληση τους με την ρομποτική εφαρμογή δεν έδειξε να τους κουράζει, αντιθέτως εκδήλωσαν την επιθυμία να ασχοληθούν και άλλο με ρομποτικές εφαρμογές, πολύ πιο εξελιγμένες όμως ώστε να εκμεταλλευτούν πλήρως τις γνώσεις τους σε αυτές.

Αρχάριοι στον προγραμματισμό

1^ο μάθημα

Το τμήμα των αρχάριων στον προγραμματισμό υπήρξε το πιο διδακτικά ευαίσθητο όλων των τμημάτων, έτσι αντιμετωπίστηκε εξ αρχής από τον διδάσκοντα με ιδιαίτερη προσοχή. Οι μαθητές κάθισαν στους υπολογιστές του εργαστηρίου και χωρίστηκαν σε ομάδες με τον ίδιο τρόπο που αυτά συνέβησαν στο τμήμα των έμπειρων που προηγήθηκε. Το μάθημα ξεκίνησε με την προβολή του εισαγωγικού βίντεο και τους μαθητές να απαντούν πως γνωρίζουν ρομποτικές εφαρμογές από την λογοτεχνία και τον κινηματογράφο, αλλά ελάχιστες υπαρκτές ρομποτικές εφαρμογές και καμία που να συμβάλλει στον τομέα της εκπαίδευσης.

Με την ολοκλήρωση της διαδικασίας των ερωταποκρίσεων και την εξαγωγή του συμπεράσματος πως ελάχιστα μπορούν να φανταστούν οι μαθητές γύρω από την διδασκαλία, παραδόθηκε σε αυτούς το πρώτο φύλλο εργασίας και τα δύο διαθέσιμα Scribbler. Αρχικά δόθηκε λίγος χρόνος στους μαθητές να περιεργαστούν τις ρομποτικές εφαρμογές, ο οποίος αποδείχθηκε αρκετός να κεντρίσει το ενδιαφέρον των μαθητών και να αναγκάσει έναν εξ αυτών να αναρωτηθεί πως θα συνδεθεί η διδασκαλία του προγραμματισμού με αυτές. Την ανάγνωση των απαιτήσεων του πρώτου φύλλου εργασίας ακολούθησε η εφαρμογή προκαθορισμένης δυνατότητας αποφυγής εμποδίων, δραστηριότητα που προκάλεσε ενθουσιασμό σχετικά με τη συμπεριφορά του ρομπότ. Οι υπόλοιπες δραστηριότητες του φύλλου συμπληρώθηκαν σωστά από τους μαθητές, ενώ σε επιπλέον ερωτήσεις του διδάσκοντα έδειξαν να γνωρίζουν ήδη την λειτουργία των αισθητήρων καθώς και την διαδικασία εισόδου-εξόδου από την καθημερινή ζωή.

Η ενασχόληση με το δεύτερο φύλλο εργασίας συνεχίστηκε στο ίδιο επίπεδο: το ρομπότ ακολούθησε την ζωγραφισμένη στο χαρτόνι διαδρομή και οι αρχάριοι έδειξαν να αντιλαμβάνονται αμέσως πως γνώρισαν άλλη μια δυνατότητα του ρομπότ και των αισθητήρων του. Στην τέταρτη δραστηριότητα του φύλλου εργασίας όλοι οι μαθητές απάντησαν λανθασμένα πως το ρομπότ θα ακολουθήσει μια και μόνο φορά το ζωγραφισμένο τετράγωνο, επισημαίνοντας έμμεσα στον διδάσκοντα πως η αντιμετώπιση των τυπικών μαθησιακών δυσκολιών του προγραμματισμού που θα ακολουθούσε στα επόμενα μαθήματα θα έκριζε ιδιαίτερης προσοχής. Εν τούτοις, με τις απαραίτητες

διευκρινήσεις του διδάσκοντα, όλοι τους έδειξαν να αντιλαμβάνονται το λάθος στο οποίο υπέπεσαν.

Ακολούθως, η άσκηση εμπέδωσης που τους ζητούσε να ζωγραφίσουν σε ένα χαρτί την διαδρομή που θα επιθυμούσαν να ακολουθήσει το ρομπότ φάνηκε να τους διασκεδάζει, όχι τόσο όμως ώστε να αποβάλλουν το άγχος τους για την εξέλιξη της διδασκαλίας. Εφάρμοσαν το πρόγραμμα επίδειξης του ρομπότ με ιδιαίτερη προσοχή, ανατρέχοντας στην διαδικασία ενεργοποίησης του και εφαρμόζοντας την πιστά για την αποφυγή οποιουδήποτε λάθους. Την όλη διαδικασία ακολούθησε σχολιασμός των μαθητών αναφορικά με τις περιπτώσεις που το ρομπότ δεν κατάφερε να ακολουθήσει μια διαδρομή και την αιτιολόγηση από μέρους τους για αυτή τη συμπεριφορά.

Ακόμη και στην εφαρμογή του τρίτου φύλλου εργασίας, παρά την χαλαρή φύση της δραστηριότητας, δύο εξ αυτών έδειξαν να μην έχουν διάθεση για ουσιαστικό παιχνίδι με το ρομπότ, αλλά παρέμεναν «σφιχτοί» και προσηλωμένοι στην παρακολούθηση της συμπεριφοράς του Scribbler.

Η επίλυση των δραστηριοτήτων του τέταρτου φύλλου εργασίας, παρουσίασε και τις πρώτες δυσκολίες για τους αρχάριους. Ενώ η πρώτη τους επαφή με τη γραφική διεπαφή δεν παρουσίασε προβλήματα, όλοι σχεδόν οι μαθητές ζήτησαν έστω και ελάχιστα την συμβολή του διδάσκοντα, παρότι τους δόθηκε αναλυτικό βίντεο για την κατασκευή του πρώτου τους προγράμματος, καθώς και εκτυπωμένη λύση. Από την ίδια την γραφική διεπαφή οι συμμετέχοντες δήλωσαν ικανοποιημένοι και πως την βρήκαν σχετικά εύχρηστη, αφήνοντας τον διδάσκοντα να ερμηνεύσει και να δικαιολογήσει αυτή τη συμπεριφορά ως έλλειψη εμπιστοσύνης στις δυνατότητές τους. Στις περισσότερες δραστηριότητες του φύλλου εργασίας που ακολούθησαν οι μαθητές απάντησαν σωστά δεσμεύοντας αρκετό χρόνο ωστόσο για να τις ολοκληρώσουν. Η ασάφεια δύο μαθητών να προσδιορίσουν τη συμπεριφορά του ρομπότ στις δραστηριότητες 3 και 4 ανάγκασε τον διδάσκοντα να φορτώσει στο ρομπότ το πρόγραμμα με τις τροποποιήσεις, με αποτέλεσμα οι μαθητές να λύσουν τις απορίες τους παρακολουθώντας το.

Οι μικρής κλίμακας δυσκολίες συνεχίστηκαν και στο πέμπτο φύλλο εργασίας για τους ίδιους δύο μαθητές. Τα τροποποιημένα προγράμματα τους με την εντολή παύσης δεν έτρεξαν σωστά, πείθοντας έτσι τον διδάσκοντα να ασχοληθεί μαζί με τους μαθητές κατά την αποσφαλμάτωση τους και να διαπιστώσει πως πρόβλημα για αυτούς τους δύο μαθητές αποτελούσε ο καθορισμός του προσανατολισμού του ρομπότ και μόνο. Μετά τις απαραίτητες εξηγήσεις πάνω σε αυτό, οι δύο αρχάριοι μαθητές αποσφαλμάτωσαν το πρόγραμμα τους και με φανερή ικανοποίηση παρακολούθησαν την εκτέλεση τους από το Scribbler.

Το πρώτο μάθημα με το τμήμα των αρχαρίων έκλεισε με μια ομαλή συνεργασία τους πάνω στο πρόβλημα του έκτου φύλλου εργασίας. Ο διδάσκων παρακολούθησε τα μέλη των ομάδων να αλληλοβοηθούνται και να ολοκληρώνουν τις δραστηριότητες με επιτυχία, παρότι το επίπεδο δυσκολίας του έκτου φύλλου ήταν σαφώς ανώτερο των προηγούμενων.

Διατυπώνοντας ένα συμπέρασμα με την ολοκλήρωση του πρώτου τμήματος της διδασκαλίας, θα λέγαμε πως οι αρχάριοι ανταπεξήλθαν αρκετά καλά στις απαιτήσεις του μαθήματος, ενώ τα προβλήματα που εμφανίστηκαν δεν αφορούσαν την κατανόηση του αντικειμένου της διδασκαλίας αλλά πρακτικά ζητήματα που ανέκυψαν λόγω απειρίας και άγχους. Η διδακτική τους εμπειρία έμοιαζε να είναι τόσο δημιουργική όσο και

διασκεδαστική, με πολλά περιθώρια βελτίωσης όμως αν δεν την αντιμετώπιζαν τόσο τυπικά.

2^ο μάθημα

Η εισήγηση του δεύτερου μαθήματος με το γκρουπ των αρχαρίων υπήρξε η αναλυτικότερη των τριών. Δεδομένου ότι ως αρχάριοι δεν είχαν έρθει ξανά σε επαφή με επαναληπτικές δομές, επιχειρήθηκε μια σταδιακή νοερή μεταφορά τους από παραδείγματα της καθημερινής ζωής, σε πιθανές επαναληπτικές διαδικασίες του Scribbler και την χρησιμότητα αυτών.

Αμέσως μετά παραδόθηκε στους μαθητές το πρώτο φύλλο εργασίας. Οι αρχάριοι, προς έκπληξη του διδάσκοντα, αναγνώρισαν αμέσως το μπλοκ της εντολής επανάληψης, παρατηρώντας το ιδιαίτερο σχήμα που έχει στη γραφική διεπαφή του Scribbler. Ωστόσο δεν απέφυγαν την παρερμηνεία του ατέρμονα βρόγχου στη δεύτερη δραστηριότητα. Ο διδάσκων, κατανοώντας πως οι έννοιες αυτές ήταν πρωτόγνωρες για τους μαθητές, επιχείρησε να αναλύσει σύντομα τι συμβαίνει κατά την εκτέλεση ενός ατέρμονα βρόγχου και πως θα μπορούσε να φανεί χρήσιμος σε ιδιαίτερες περιπτώσεις. Με την ίδια την δομή της εντολής δεν αναδείχθηκε κάποια παρερμηνεία, γεγονός που απέδειξε και η ορθή κρίση των αρχαρίων στο τρίτο ερώτημα.

Συνεχίζοντας με την διδασκαλία και πριν οι μαθητές προχωρήσουν στην εκπλήρωση των δραστηριοτήτων του δεύτερου φύλλου, ο διδάσκων ζήτησε από τους μαθητές της ομάδας να περιγράψουν την δομή του έτοιμου προγράμματος που αυτό περιείχε. Αρχικά υπήρξε ένα μικρό μούδιασμα, αλλά με την πρώτη απάντηση μαθητή και με τις απαραίτητες εξηγήσεις του διδάσκοντα το θέμα ύπαρξης δύο επαναληπτικών δομών σε ένα πρόγραμμα αποσαφηνίστηκε. Έτσι οι μόνες περιπτώσεις όπου χρειάστηκε μια παρέμβαση του διδάσκοντα εμφανίστηκαν στην δεύτερη δραστηριότητα όπου ένας μαθητής ζήτησε διευκρίνιση για την εκφώνηση και στην τέταρτη δραστηριότητα όπου όλοι οι μαθητές δίσταζαν να ξεκινήσουν τη συγγραφή του απαιτούμενου προγράμματος. Στη δεύτερη περίπτωση δόθηκαν ορισμένα επιπλέον στοιχεία για τη δομή που πρέπει να έχει το πρόγραμμα και τρεις εκ των μαθητών το ολοκλήρωσαν ορθά. Στον τέταρτο μαθητή οι οποιοσδήποτε απορίες λύθηκαν με την φόρτωση ενός εκ των ορθών προγραμμάτων στο ρομπότ, την εκτέλεση του και ορισμένα ακόμη επεξηγηματικά σχόλια του διδάσκοντα προς αυτόν.

Τα μεγαλύτερα προβλήματα των αρχαρίων στο τρίτο φύλλο εργασίας αφορούσαν την λογική συμπλήρωσης του πίνακα μετρητών για την εμφωλευμένη επαναληπτική δομή και γενικότερα την κατανόηση της δομής της. Με την αρωγή του διδάσκοντα στις πρώτες δραστηριότητες τα προβλήματα ξεπεράστηκαν. Η διδασκαλία ολοκληρώθηκε με μια υποδειγματική συνεργασία των μαθητών στο τέταρτο φύλλο και με τον διδάσκοντα να μένει απόλυτα ικανοποιημένος από τις επιδόσεις των αρχαρίων. Οι προβληματισμοί των μαθητών αυτού του γκρουπ αφορούσαν κυρίως δραστηριότητες που τους συνέστησαν νέες έννοιες (πχ δομή επανάληψης, εμφωλευμένες επαναλήψεις). Μόλις οι έννοιες αυτές έγιναν πλήρως αντιληπτές, οι επιδόσεις των μαθητών σε πιο σύνθετες εργασίες βελτιώθηκαν πάρα πολύ. Σαφώς και ο χρόνος που χρειάστηκαν για την πραγματοποίηση των δραστηριοτήτων ήταν μεγαλύτερος από αυτόν των αρχαρίων, όμως δεν θα μπορούσαμε να ισχυριστούμε πως δεν αντεπεξήλθαν άριστα.

3^ο μάθημα

Ξεκινώντας το τρίτο μάθημα με τους αρχάριους προγραμματιστές και επιπρόσθετα με την εισήγηση, ο διδάσκων εκμείωσε από αυτούς τον προβληματισμό τους σχετικά με την διαδικασία που το ρομπότ ακολουθεί για να πάρει αποφάσεις. Οι μαθητές έδειξαν ενθουσιασμένοι που θα μάθαιναν πώς να προγραμματίζουν το ρομπότ να αποφασίζει ανάλογα με τις συνθήκες του περιβάλλοντος, ενθουσιασμός που έγινε ευδιάκριτος καθ' όλη την διάρκεια του τρίτου μαθήματος.

Οι επιδόσεις των αρχάριων στο πρώτο φύλλο εργασίας όπου και ήρθαν για πρώτη φορά σε επαφή με την δομή της εντολής ελέγχου του Scribbler αλλά και με την διαδικασία παρακολούθησης της συμπεριφοράς του ρομπότ και καταγραφής της, ικανοποίησαν σε μεγάλο βαθμό των διδασκόντα. Καμία απορία δεν προέκυψε σε απλές δομές ελέγχου, εκτός των αναμενόμενων για την μεθοδολογία τοποθέτησης τους σε ένα πρόγραμμα. Σε πιο σύνθετες δομές επιλογής ο διδάσκων έκρινε ότι πρέπει να παρέμβει και να αποσαφηνίσει στους μαθητές τις ιδιαιτερότητες αυτών (πχ AND, OR, ακολουθιακή εκτέλεση προγράμματος) πριν αυτοί προχωρήσουν στη συγγραφή κώδικα. Παρόλα αυτά, λάθη και παρερμηνείες προέκυψαν και πάλι, αλλά από δύο μαθητές μόνο και σε τέτοια κλίμακα που δεν επέτρεψαν την δημιουργία κακής εντύπωσης στον διδάσκοντα. Σε γενικές γραμμές η επίδοσή τους στα δύο πρώτα φύλλα κρίθηκε ως αρκετά καλά, αν αναλογιστούμε πως πίνακες αληθείας και σύνθετες δομές ελέγχου ταλαιπωρούν για πολλά χρόνια αρχάριους του προγραμματισμού.

Στο project του τρίτου φύλλου εργασίας οι μαθητές αντέδρασαν κάπως νωχελικά. Η έκταση του προγράμματος που έπρεπε να συντάξουν μόνοι τους αυτή τη φορά και η σύνθεση σχεδόν όλων των εννοιών που είχαν διδαχθεί μέχρι στιγμής, τους προκάλεσε μια μικρή σύγχυση. Η καθοδήγηση από τον διδάσκοντα κρίθηκε επιβεβλημένη ώστε το πρόβλημα να αναλυθεί σε μικρότερα και οι μαθητές να συνειδητοποιήσουν πως η αμηχανία τους κατά την ανάγνωση της εκφώνησης δεν υπήρξε δικαιολογημένη. Έτσι λοιπόν κατάφεραν να ολοκληρώσουν και την πιο απαιτητική δραστηριότητα όλης της διδασκαλίας, δεσμεύοντας όμως χρόνο ως το τέλος του μαθήματος και με τις παρεμβάσεις του διδασκόντα να αποδεικνύονται σε πολλές περιπτώσεις καταλυτικές. Μία από τις λύσεις του προβλήματος φορτώθηκε ενδεικτικά στο Scribbler ώστε οι μαθητές να παρακολουθήσουν το αποτέλεσμα των προγραμμάτων τους. Όλοι τους έδειχναν σαφώς ενθουσιασμένοι που κατάφεραν να φέρουν εις πέρας ένα project που άνετα μπορεί να χαρακτηριστεί δύσκολο για το επίπεδό τους, ενώ παρακολουθούσαν το ρομπότ με ικανοποίηση.

Σχηματίζοντας μια ολοκληρωμένη κρίση για το τρίτο μάθημα με τους αρχάριους μπορούμε να πούμε πως ξεπέρασαν κατά πολύ τις προσδοκίες του διδασκόντα, ένα ακόμη μάθημα όμως με συνδυαστικές ασκήσεις από όλα τα κεφάλαια της διδασκαλίας θα έλυνε οποιοσδήποτε παρανοήσεις και θα τους εξοικείωνε ακόμη περισσότερο με την προγραμματιστική λογική επίλυσης προβλημάτων. Κρίνοντας όμως την παρούσα διδασκαλία, θα λέγαμε πως εισήχθησαν με ιδανικά σε βασικές έννοιες του προγραμματισμού και πλησίασαν το άριστα σε αρκετές περιπτώσεις.

Στοιχειωδώς καταρτισμένοι στον προγραμματισμό

1^ο μάθημα

Η εισαγωγή του πρώτου μαθήματος της τελευταίας ομάδας μαθητών πραγματοποιήθηκε ακριβώς με την ίδια διαδικασία που ακολουθήθηκε στις άλλες δύο. Οι μαθητές αυτού του γκρουπ έδειξαν με τις απαντήσεις τους κατά την διάρκεια προβολής του εισαγωγικού βίντεο πως γνώριζαν αρκετά περισσότερα από τους αρχάριους σχετικά με τις ρομποτικές εφαρμογές, ενώ είχαν ακούσει και για εναλλακτικά μαθήματα με χρήση εκπαιδευτικών ρομπότ. Η πρώτη επαφή τους με το ρομπότ υπήρξε ενδιαφέρουσα: αμέσως οι συμμετέχοντες άρχισαν να το παρατηρούν εξονυχιστικά, ενώ δύο από αυτούς εντόπισαν τους αισθητήρες του ρομπότ και ρώτησαν τον διδάσκοντα για να επαληθεύσουν την εκτίμησή τους. Ακόμη, σχολίασαν μεταξύ τους τα χαρακτηριστικά του Scribbler, μέχρι ο διδάσκων να τους διακόψει παραδίδοντάς τους το πρώτο φύλλο εργασίας.

Η συμπεριφορά των μαθητών κατά την ενασχόληση των τεσσάρων πρώτων φύλλων εργασίας δεν χρίζει ιδιαίτερης ανάλυσης. Εκτέλεσαν τις περισσότερες δραστηριότητες σωστά, απολαμβάνοντας όσες από αυτές αφορούσαν παιχνίδι με το ρομπότ, δείχνοντας όμως σαφώς λιγότερη άνεση σε σχέση με τους έμπειρους. Η παρέμβαση του διδάσκοντα κρίθηκε απαραίτητη μόνο στην πέμπτη δραστηριότητα του τέταρτου φύλλου εργασίας, όπου και βοήθησε έναν εκ των μαθητών να αντιληφθεί πως το ρομπότ με τις κινήσεις του μπορεί να προσεγγίσει μια προκαθορισμένη περιοχή.

Οι επιλογές των μαθητών στην δραστηριότητα του πέμπτου φύλλου εργασίας υπήρξαν αρκετά ακριβείς, ενώ οι εντολές παύσης προβλημάτισαν μόνο έναν εξ αυτών για τα σημεία όπου έπρεπε να τοποθετηθούν. Με μια πρώτη εκτίμηση θα μπορούσαμε να πούμε πως οι στοιχειώδεις γνώσεις αυτής της ομάδας επαρκούσαν ώστε να τους επιτρέψουν να εκτελέσουν απροβλημάτιστα τις πρώτες τους δραστηριότητες στην διδασκαλία. Ιδιαίτερα χρήσιμες φάνηκαν στον χειρισμό της γραφικής διεπαφής, καθώς μετά από τις πρώτες δραστηριότητες που απαιτούσαν την χρήση του και ένα μικρό χρονικό διάστημα εξοικείωσης, κανείς τους δεν αντιμετώπισε δυσκολία που να ευθύνεται στον χειρισμό της.

Μόνο ίσως πρόβλημα για το πρώτο μάθημα με τους στοιχειωδώς καταρτισμένους προγραμματιστές, αποτέλεσε η συνεργασία μεταξύ των μελών της μιας ομάδας κατά την επίλυση του έκτου φύλλου εργασίας, αλλά και αυτή γρήγορα ξεπεράστηκε. Πριν την διεξαγωγή του μαθήματος, η πεποίθηση του διδάσκοντα ήταν πως θα αντιμετώπιζε πολλές περισσότερες δύσκολες καταστάσεις με αυτό το γκρουπ, εκτίμηση που ευτυχώς διαψεύστηκε σε μεγάλο βαθμό.

2^ο μάθημα

Το δεύτερο μάθημα με το τμήμα των στοιχειωδώς καταρτισμένων στον προγραμματισμό μαθητών ξεκίνησε με μια μικρή εισήγηση στις επαναληπτικές δομές, όπως και στα υπόλοιπα τμήματα. Από τις πρώτες ερωτήσεις και παραδείγματα, δύο μαθητές αντιλήφθηκαν ποιο θα είναι το θέμα του δεύτερου μαθήματος, καθώς είχαν χρησιμοποιήσει και στο παρελθόν τέτοιες δομές. Παρά την εμπειρία των δύο αυτών μαθητών η διαδικασία συνεχίστηκε κανονικά, προσπαθώντας να δοθεί ένα επαρκές υπόβαθρο και για τους υπόλοιπους. Μόλις ο διδάσκων έκρινε πως αυτοί οι μαθητές εισήχθησαν επαρκώς στην γενικότερη δομή των επαναληπτικών εντολών, η διδασκαλία προχώρησε με το πρώτο φύλλο εργασίας.

Σε αυτήν την πρώτη επαφή με την εντολή επανάληψης του Scribbler, έγινε ορατή και η διαφορά εξοικείωσης των μαθητών με επαναληπτικές δομές. Οι δύο μαθητές που είχαν ξανασυναντήσει τέτοιες εντολές ασχολήθηκαν με το φύλλο εργασίας χωρίς να ανακύψει κάποιο πρόβλημα. Από την άλλη, οι δυο άλλοι μαθητές, χρειάστηκαν την συμβολή του διδάσκοντα ώστε να διακρίνουν την εντολή επανάληψης και τον τρόπο που αυτή δέχεται ορίσματα και εκτελείται, δείχνοντας όμως στην τρίτη δραστηριότητα πως συνειδητοποίησαν τα παραπάνω.

Το δεύτερο φύλλο εργασίας με τις συνεχόμενες επαναληπτικές δομές αποδείχθηκε δύσκολο μόνο για ένα μαθητή και κυρίως κατά την σύνταξη δικών του προγραμμάτων στην γραφική διεπαφή. Η περισσότερη προσπάθεια από την πλευρά του διδάσκοντα καταβλήθηκε προς αυτόν τον μαθητή, δεδομένου ότι οι υπόλοιποι δυσκολεύτηκαν μόνο κατά την συγγραφή του τελευταίου προγράμματος και μάλιστα σε μικρό βαθμό. Η παρανόηση για αυτόν τον μαθητή προέκυψε από την αδυναμία αντίληψης της έναρξης επόμενης επανάληψης μέσα στην δομή. Η προσοχή του διδάσκοντα στράφηκε σε αυτόν γιατί αν μια τέτοια λανθασμένη αντίληψη δεν διορθωνόταν, ο μαθητής θα αδυνατούσε να προχωρήσει συνολικότερα στη διδασκαλία. Με ανακούφιση διαπιστώθηκε πως στην συγγραφή του προγράμματος της έκτης δραστηριότητας δεν προβληματίστηκε, απλά όπως και οι συμμαθητές του δήλωσε πως θα επιθυμούσε να έχει λίγο περισσότερο χρόνο στην διάθεσή του.

Η ενασχόληση με το τρίτο φύλλο εργασίας απλά επιβεβαίωσε την εκτίμηση του διδάσκοντα ότι εκείνοι οι μαθητές που είχαν ξανασυναντήσει επαναληπτικές δομές θα περνούσαν ένα ευχάριστο δίωρο, σε αντίθεση με τους άλλους δύο όπου έπρεπε να επιδειχθεί ιδιαίτερη προσοχή. Γενεσιουργός αιτία των δυσκολιών που αντιμετώπισαν οι μαθητές ήταν η έλλειψη εμπιστοσύνης στην κρίση τους, καθώς μετά τις επεξηγήσεις του διδάσκοντα σε κάθε τι καινούριο που συναντούσαν, συνέχιζαν απρόσκοπτα στις υπόλοιπες δραστηριότητες.

Η συνεργατική δραστηριότητα του τετάρτου φύλλου επιβεβαίωσε τον φόβο της πρώτης ώρας για έλλειψη συνεργασίας στην μια ομάδα εκ των δύο. Οι διαφωνίες των μαθητών που την αποτελούσαν καθυστέρησαν αρκετά την εκπλήρωσή της, επιτρέποντας στην άλλη ομάδα που λειτούργησε άψογα να απολαύσει λίγο ελεύθερο χρόνο πειραματισμού με το ρομπότ.

Γενικότερα, το γκρουπ των στοιχειωδώς καταρτισμένων προγραμματιστών φάνηκε να χωρίζεται σε υποομάδες διαφορετικής δυναμικότητας, αυτούς που ήδη είχαν μια εμπειρία σε επαναληπτικές δομές και αυτούς που δεν είχαν. Αν οι δεύτεροι έδειχναν λίγη μεγαλύτερη αυτοπεποίθηση και λιγότερη ανασφάλεια (ίσως την προκαλούσε η χαρακτηριστική ευκολία με την οποία οι άλλοι μαθητές εκπλήρωναν τις δραστηριότητες), τα προβλήματά τους θα είχαν περιοριστεί κατά πολύ.

3^ο μάθημα

Στην εισήγηση του τρίτου μαθήματος με τους στοιχειωδώς καταρτισμένους στον προγραμματισμό μαθητές, επιχειρήθηκε και πάλι μια σταδιακή εισαγωγή των τους στην έννοια των δομών ελέγχου και της λήψης αποφάσεων. Δύο εκ των μαθητών της ομάδας δήλωσαν στο τέλος της εισήγησης πως έχουν χρησιμοποιήσει τέτοιες δομές στο παρελθόν, όπως και πριν εντολές επανάληψης. Όλοι τους συνέκλιναν στην διαπίστωση πως η λήψη αποφάσεων από το ρομπότ θα ήταν η εντυπωσιακή από τις διαθέσιμες λειτουργίες του.

Με την εμπειρία που απέκτησαν οι συμμετέχοντες στο δεύτερο μάθημα και την μεγάλη διάθεση συμμετοχής στη διδασκαλία, οι πρώτες δραστηριότητες που αφορούσαν απλές δομές ελέγχου κύλισαν ικανοποιητικά, με μόνο πρόβλημα την αδυναμία δύο μαθητών να παρακολουθήσουν την συμπεριφορά του ρομπότ και να την καταγράψουν ως λήψη απόφασης. Έτσι, η προσοχή του διδάσκοντα στράφηκε σε αυτούς και διαπιστώθηκε πως όταν οι μαθητές κλήθηκαν να παρατηρήσουν την σύνθεση του προγράμματος και να αναλύσουν μια ενδεχόμενη συμπεριφορά του ρομπότ δεν υπήρξε πρόβλημα.

Στις πιο σύνθετες δομές ελέγχου εμφανίστηκαν σε πρώτη οι ίδιες παρερμηνείες με το τμήμα των αρχαρίων. Αφού εντοπίστηκε πως το κύριο πρόβλημα ήταν η διακλάδωση σε ένα σύνθετο πρόγραμμα και η επιλογή ενός και μόνο μονοπατιού για να συνεχίσει η εκτέλεση του προγράμματος, ο διδάσκων αφιέρωσε περισσότερο χρόνο στα εισαγωγικά ερωτήματα των δραστηριοτήτων, που περιλάμβαναν τις δομές των προγραμμάτων-παραδειγμάτων. Στη συμπλήρωση των πινάκων αληθείας επιστρατεύτηκε και το Scribbler ώστε κανείς εκ των μαθητών να μην αμφιβάλλει για τις απαντήσεις του.

Η συνέχεια της διδασκαλίας με την ανάληψη του project στο τρίτο φύλλο εργασίας προβλημάτισε αρχικά τον διδάσκοντα, καθώς μέχρι στιγμής είχαν παρουσιαστεί αρκετές δυσκολίες που δεν αφορούσαν τα πρακτικά ζητήματα συγγραφής των προγραμμάτων αλλά την ίδια την προγραμματιστική λογική. Η βοήθεια του διδάσκοντα στους μαθητές του τμήματος των στοιχειωδώς καταρτισμένων προγραμματιστών υπήρξε σαφώς μεγαλύτερη απ' ό,τι στα άλλα τμήματα, αλλά αφορούσε αποκλειστικά τις δομές επιλογής και είχε τη μορφή επιβεβαίωσης των απαντήσεων που οι μαθητές είχαν δώσει. Το πρόβλημα αποσυντέθηκε σε μικρότερα σε ικανοποιητικό βαθμό, γεγονός που διευκόλυνε και τους ίδιους να ολοκληρώσουν την δραστηριότητα ταχύτερα από τους αρχάριους μαθητές. Η εκτέλεση ενδεικτικά ενός εκ των προγραμμάτων δεν βρήκε τους μαθητές αυτής της ομάδας απόλυτα ικανοποιημένους καθώς γνώριζαν πως θα μπορούσαν να είχαν αποδώσει αρκετά καλύτερα.

Εν τούτοις, αν λάβουμε υπόψη πως μόνο δύο από τους μαθητές είχαν εμπειρία σε αντίστοιχες προγραμματιστικές δομές και μάλιστα μικρή, η συνολική παρουσία των μαθητών μόνο ως καλή μπορεί να σχολιαστεί. Σαφέστατα και προέκυψαν αρκετά ζητήματα κατά την διάρκεια της διδασκαλίας και οι προσδοκίες του διδάσκοντα για ιδιαίτερα υψηλό προγραμματιστικό επίπεδο δεν επαληθεύτηκαν, αλλά αυτό οφείλεται περισσότερο στην υπερεκτίμηση των δυνατοτήτων των μαθητών από τον διδάσκοντα παρά σε λάθη των ίδιων.

5.3.2 Αποτελέσματα φύλλου αξιολόγησης έμπειρων χρηστών

Σκιαγραφώντας το προφίλ της ομάδας των έμπειρων χρηστών, θα μπορούσαμε να πούμε πως συγκροτήθηκε από αρκετά έμπειρους προγραμματιστές που έχουν δουλέψει πάνω σε βασικά προγραμματιστικά εργαλεία και εκτιμούν την προγραμματισμό ως πολύ ενδιαφέρον προγραμματιστικό αντικείμενο(ερωτήσεις 1,2,3, Εμπειρία στον προγραμματισμό). Παρόλα αυτά, απάντησαν πως δυσκολεύτηκαν πάρα πολύ κατά την πρώτη τους επαφή με το αντικείμενο, με την ευθύνη να επιρρίπτεται στη ίδια τη φύση του προγραμματισμού από δύο μαθητές, σε δυσκολία χρήσης της εκάστοτε εφαρμογής από έναν άλλο και σε υπερβολικό φόρτο εργασίας από τον τέταρτο. Λαμβάνοντας υπόψη τις απαντήσεις των χρηστών μπορούμε να πούμε πως ο δρόμος προς την κατάκτηση της γνώσης υπήρξε δύσβατος για όλους τους μαθητές, με σημαντικό μερίδιο ευθυνών να καταλογίζεται και στην παραδοσιακή μέθοδο διδασκαλίας, η οποία υπήρξε και η μόνη με

την οποία διδάχθηκαν όχι μόνο προγραμματισμό αλλά τα περισσότερα επιστημονικά αντικείμενα (ερωτήσεις 4,5,6,7,8,9, Εμπειρία στον προγραμματισμό). Η πρακτική εμπειρία τους σε ρομποτικές υποστηρίχθηκε πως ήταν ελάχιστη μέχρι πρότινος (ερώτηση 10, Εμπειρία στον προγραμματισμό).

Η κριτική των έμπειρων χρηστών για τη γραφική διεπαφή του Scribbler υπήρξε σκληρή. Η ευχρηστία της αλλά και οι απεικονίσεις των βασικών προγραμματιστικών δομών χαρακτηρίστηκαν συνολικά ως μέτριες, η δυνατότητα κάλυψης των προγραμματιστικών δομών που ήδη γνωρίζουν μικρή, ενώ συμπληρώνοντας αυθόρμητα την ελεύθερη ερώτηση για τη μεγαλύτερη αδυναμία της, τρεις μαθητές απάντησαν πως υπήρξε η περιορισμένη δυνατότητα δημιουργίας σύνθετων προγραμμάτων και ένας εξ αυτών η μη επιστημονική μορφή της. Αυτές οι δυσκολίες έπεισαν δύο εκ των συμμετεχόντων να απαντήσουν πως θα προτιμούσαν η διδασκαλία να βασιστεί σε ένα χαμηλότερου επιπέδου προγραμματιστικό εργαλείο, αφήνοντας ανεπηρέαστους τους άλλους δύο(ερωτήσεις 1,2,3,4,5, Γραφική διεπαφή).

Παρά την αρνητική κρίση των έμπειρων προγραμματιστών σχετικά με τη δυνατότητα να καλυφθεί ένα ολοκληρωμένο μάθημα προγραμματισμού μόνο με το Scribbler, η διδασκαλία αυτών των εισαγωγικών μαθημάτων κρίθηκε ως αρκετά πλήρης, με αρκετά ευφάνταστες δραστηριότητες και προφανή σύνδεση αυτών με τις βασικές προγραμματιστικές έννοιες. Οι βοηθητικές σημειώσεις με εναλλακτική μορφή βίντεο ικανοποίησαν και τους τέσσερις, σε αντίθεση με την άποψη τους για περισσότερο ελεύθερο χρόνο με την ρομποτική εφαρμογή, όπου είχαμε διχογνωμία(ερωτήσεις 1,2,3,4,5,6,7, Κρίνοντας την διδασκαλία).

Στη συνέχεια, με τη βοήθεια των αντίστοιχων ζητημάτων του ερωτηματολογίου, ο διδάσκων επιχείρησε να φέρει τους μαθητές στη θέση του, ώστε να διαπιστώσει πως θα οργάνωναν εκείνοι μια διδασκαλία προγραμματισμού. Οι μαθητές απάντησαν κατά πλειοψηφία τριών έναντι ενός πως θα επέλεγαν ως μέθοδο διδασκαλίας αυτή που προϋπόθετε χρήση εκπαιδευτικής ρομποτικής. Οι τρεις μαθητές δικαιολόγησαν αυτή την επιλογή τους με διαφορετικά επιχειρήματα («...γιατί τα σύγχρονα τεχνολογικά μέσα διευκολύνουν την εκπαίδευση», «γιατί οι μαθητές θα καταλάβαιναν ευκολότερα», «...γιατί με το ρομπότ τα αποτελέσματα ενός προγράμματος είναι ορατά»), ενώ ο τέταρτος μαθητής επέλεξε την παραδοσιακή μέθοδο εργαστηρίων-διαλέξεων γιατί αυτή είχε βιώσει και έβρισκε αναγνωρισμένη επιστημονικά.(«...γιατί με αυτήν έχω διδαχθεί και εγώ και εφαρμόζεται χρόνια τώρα») (ερωτήσεις 1,2, Στη θέση του διδάσκοντα).

Η επίδραση των παραδοσιακών μεθόδων διδασκαλίας στους μαθητές του έμπειρου γκρουπ ήταν προφανής, έτσι δύο εξ αυτών θα πρόσθεταν στην παρούσα διδασκαλία ένα εγχειρίδιο για τη γραφική διεπαφή του Scribbler, ενώ οι άλλοι δύο την εκπόνηση ενός project στο σπίτι. Στην ερώτηση για την αφαίρεση ενός στοιχείου από την διδασκαλία υπήρξε πλήρης διασπορά των απαντήσεων, με έναν εκ των μαθητών να συμπληρώνει σε επιλογή ελεύθερης απάντησης πως θα αφαιρούσε τις υποδομές για την εκτέλεση κάποιων δραστηριοτήτων, γιατί θα ήταν χρονοβόρες στην κατασκευή. Τέλος, οι έμπειροι προγραμματιστές, με τις τελευταίες απαντήσεις τους όντας στη θέση του διδάσκοντα, κατέδειξαν πως μπορεί από τη μία η παραδοσιακή μέθοδος διαλέξεων-εργαστηρίων να είναι ιδιαίτερα επίπονη για τον μαθητή, από την άλλη όμως η σύγχρονη μέθοδος με τη βοήθεια της ρομποτικής κρύβει περισσότερους κινδύνους και δυσκολίες για τον ίδιο τον διδάσκοντα (ερωτήσεις 3,4,5, Στη θέση του διδάσκοντα).

Στο τελευταίο μέρος του φύλλου αξιολόγησης αποκλειστικά για έμπειρους χρήστες, επιχειρήθηκε μια σύγκριση μεταξύ των δύο μεθόδων. Εξάγοντας ένα γενικό συμπέρασμα, οι μαθητές απάντησαν πως μια σύγχρονη μέθοδος διδασκαλίας με εκπαιδευτική ρομποτική υπερέχει ξεκάθαρα σε τομείς που ανέκαθεν αποτελούσαν πρόβλημα κατά την διδασκαλία του προγραμματισμού (στρατηγικές επίλυσης προβλημάτων, τυπικές μαθησιακές δυσκολίες, ανάπτυξη προγραμματιστικής λογικής), αλλά και σε γενικότερους διδακτικούς στόχους είναι όμως ιδιαίτερα χρονοβόρα τόσο για τον διδάσκοντα όσο και για τον μαθητή. Κλείνοντας, τρεις μαθητές υποστήριξαν την εφαρμογή εναλλακτικών μορφών διδασκαλίας έναντι ενός που διαφώνησε, βρίσκοντας όμως μια εναλλακτική διδασκαλία δύσκολα εφαρμόσιμη σε ένα ελληνικό πανεπιστήμιο. (ερωτήσεις 1-12, Σύγκριση διδακτικών μεθόδων).

Ακολουθούν οι πίνακες με τις απαντήσεις των μαθητών.

Ερώτηση	ΜΟ	Ερώτηση	ΜΟ	Ερώτηση	ΜΟ
Εμπειρία στον προγραμματισμό		Γραφική διεπαφή		Κρίνοντας την διδασκαλία	
2.Πόσο έμπειρο προγραμματιστή θα χαρακτηρίζατε τον εαυτό σας;	4,25	1.Βαθμολογήστε στην κλίμακα την ευχρηστία της γραφικής διεπαφής του Scribbler.	3,25	1. Δεδομένης της εμπειρίας σας και του εισαγωγικού χαρακτήρα των μαθημάτων, πόσο πλήρη θα τα χαρακτηρίζατε;	4
3.Πόσο ενδιαφέρον επιστημονικό αντικείμενο θα χαρακτηρίζατε τον προγραμματισμό;	4	2.α Κατά πόσο πιστεύετε ότι απεικονίζει κατανοητά τη δομή του προγράμματος;	3,25	2. Σε τι βαθμό πιστεύετε ότι μπορεί να καλυφθεί η ύλη ενός ολοκληρωμένου μαθήματος προγραμματισμού αποκλειστικά με το Scribbler;	2,75
4.Πόσο δυσκολευτήκατε κατά την πρώτη επαφή σας με μια κλασσική γλώσσα προγραμματισμού;	4,5	2.β Κατά πόσο πιστεύετε ότι απεικονίζει κατανοητά την εντολή ελέγχου;	3,75	3. Πόσο ευδιάκριτη θα χαρακτηρίζατε την σύνδεση των επιμέρους δραστηριοτήτων των μαθημάτων με τις αντίστοιχες προγραμματιστικές έννοιες που διδάχθηκαν;	4,75
6.Πόσο ενδιαφέρουσα θα χαρακτηρίζατε την παραδοσιακή μέθοδο διδασκαλίας προγραμματισμού, με διαλέξεις και εργαστήρια;	2,25	2.γ Κατά πόσο πιστεύετε ότι απεικονίζει κατανοητά την εντολή επανάληψης;	3,25	4. Σε ποιο βαθμό πιστεύετε ότι οι διδάσκοντες εκμεταλλεύθηκαν τις δυνατότητες που παρέχει στον εκπαιδευτικό μια τέτοια μορφή διδασκαλίας;	3,75
9.Βαθμολογήστε στην κλίμακα την εμπειρία σας σε εναλλακτικές μεθόδους διδασκαλίας προγραμματισμού.	2,25	5. Κατά πόσο πιστεύετε ότι οι εντολές της γραφικής διεπαφής του Scribbler καλύπτουν όλες τις περιπτώσεις αντίστοιχων εντολών σε	2,75	5. Πόσο ευφάνταστες θα χαρακτηρίζατε αυτές τις δραστηριότητες;	4,25

			κλασσικά προγραμματιστικά εργαλεία που έχετε χρησιμοποιήσει;				
10.Βαθμολογήστε στην κλίμακα την εμπειρία σας σε ρομποτικές εφαρμογές.	2,5						
						Σύγκριση διδακτικών μεθόδων	
						12. Πόσο αποτελεσματική πιστεύετε ότι θα ήταν μια τέτοια μέθοδος διδασκαλίας προγραμματισμού σε ένα ελληνικό εκπαιδευτικό ίδρυμα;	3,5

Αποτελέσματα σε ερωτήσεις βαθμολογικής κλίμακας

Ερώτηση	Επιλογή 1	Επιλογή 2
1.Καλύπτεται μεγαλύτερη ύλη σε μικρότερο χρονικό διάστημα.	4	0
2.Ο μαθητής κατανοεί ευκολότερα δυσνόητες προγραμματιστικές έννοιες.	1	3
3. Ο μαθητής θέτει στέρεες προγραμματιστικές βάσεις.	1	3
4.Το έργο του διδάσκοντα είναι ευκολότερο.	4	0
5. Ο μαθητής σφείλει να αφιερώσει στο μάθημα λιγότερες ώρες από τον ελεύθερό του χρόνο.	2	2
6. Εγείρεται ευκολότερα το ενδιαφέρον του μαθητή.	0	4
7. Η μετάβαση σε σύγχρονα προγραμματιστικά εργαλεία είναι ευκολότερη.	4	0
8. Ο μαθητής δομεί τρόπο σκέψης που θα του χρειαστεί σε παράπλευρα επιστημονικά πεδία.	0	4
9. Ο μαθητής κατανοεί και χρησιμοποιεί ευκολότερα στρατηγικές επίλυσης προβλημάτων.	1	3
10. Οι μαθητές συνεργάζονται	0	4

αρτιότερα για την επίλυση ενός προβλήματος.		
---	--	--

Σύγκριση διδακτικών μεθόδων (ο μαθητής επιλέγει 1 για τη μέθοδο διαλέξεων-εργαστηρίων και 2 για την εκπαιδευτική ρομποτική)

Ερώτηση	A	B	Γ	Δ
Εμπειρία στον προγραμματισμό				
	Αδυναμία κατανόησης της προγραμματιστικής λογικής.	Αδυναμία υλοποίησης της λύσης ενός προβλήματος παρά την κατανόηση της λογικής της.	Αδυναμία αφομοίωσης της γνώσης λόγω υπερβολικής ύλης σε μικρό χρονικό διάστημα	Άλλη
5. Ποια ήταν η αιτία της δυσκολίας που αντιμετωπίσατε;	2	1	1	0
	Έλλειψη σύνδεσης των εννοιών του προγραμματισμού με την πραγματική ζωή.	Έλλειψη κατανοητής οπτικοποίησης των αποτελεσμάτων ενός προγράμματος.	Αυστηρή προσέγγιση της δομής μιας γλώσσας	Άλλη
7. Ποιο πιστεύετε ότι είναι το μεγαλύτερο μειονέκτημα της παραδοσιακής μεθόδου διδασκαλίας;	0	3	1	0
	Μόνο με τη μέθοδο εργαστηρίων-διαλέξεων.	Με λογισμικό απεικόνισης των αποτελεσμάτων σε γραφικό περιβάλλον	Με κάποια ρομποτική εφαρμογή	Με τις μεθόδους
8. Έχετε διδαχθεί προγραμματισμό :	4	0	0	0
Γραφική διεπαφή	-	-	-	-
	Το rython based περιβάλλον	Τη γραφική διεπαφή	-	-
4. Γνωρίζοντας ότι το Scribbler robot μπορεί να προγραμματιστεί και σ ένα rython-based περιβάλλον και μορφή προγραμμάτων σε κώδικα, ποιο από τα δύο περιβάλλοντα θα προτιμούσατε να χρησιμοποιήσετε;	2	2	0	0
Κρίνοντας την διδασκαλία	-	-	-	-
	Ναι	Όχι	-	-
6. Θα προτιμούσατε το βοηθητικό υλικό των μαθημάτων να παρεχόταν σε μορφή σημειώσεων με επεξηγήσεις αντί των αρχείων βίντεο;	0	4	0	0

	Ναι	Όχι	-	-
7. Θα προτιμούσατε η μορφή των δραστηριοτήτων να μην ήταν τόσο αυστηρή αλλά να έδινε στον μαθητή περιθώρια ελευθερίας;	2	2	0	0
Στη θέση του διδάσκοντα	-	-	-	-
	Θα επέλεγα την παραδοσιακή μέθοδο διαλέξεων-εργαστηρίων	Θα επέλεγα την μέθοδο διδασκαλίας με χρήση εκπαιδευτικής ρομποτικής		
1. Αν σας είχε ζητηθεί να διδάξετε ένα ολοκληρωμένου μάθημα προγραμματισμού, ποια μέθοδο διδασκαλίας θα επιλέγατε και γιατί;	1	3	0	0
	Εγχειρίδιο χρήσης της γραφικής διεπαφής του Scribbler.	Εκπόνηση συνδυαστικού project στο σπίτι.	Βασικά στοιχεία της python based γλώσσας του Scribbler.	Άλλο
2. Αν βρισκόσασταν στη θέση του διδάσκοντα τι θα προσθέτατε στην διδασκαλία;	2	1	1	0
	Δραστηριότητες που αφορούν αποκλειστικά τη γραφική διεπαφή του Scribbler.	Δραστηριότητες με επίλυση στο χαρτί.	Εισαγωγή (multimedia και συζήτηση) στην ρομποτική.	Άλλο
3. Αν βρισκόσασταν στη θέση του διδάσκοντα τι θα αφαιρούσατε από την διδασκαλία;	1	1	1	ΆΛΛΟ
	Αδυναμία μετάδοσης γνώσης στους μαθητές.	Δημιουργία κατανοητών παραδειγμάτων προς μελέτη.	Προσέλκυση του ενδιαφέροντος των μαθητών.	Άλλο
4. Αν είχατε επιλέξει την παραδοσιακή μέθοδο διδασκαλίας με εργαστήρια και διαλέξεις, ποιο πιστεύετε ότι θα ήταν το μεγαλύτερο πρόβλημα που θα αντιμετωπίζατε;	2	1	1	0
	Υπερβολικός απαιτούμενος χρόνος για την οργάνωση ενός μαθήματος.	Δυσκολία σύνδεσης προγραμματιστικών εννοιών με λειτουργίες του ρομπότ.	Αποπροσανατολισμός των μαθητών από το ίδιο το ρομπότ.	Άλλο
5. Αν είχατε επιλέξει	1	1	2	0

<p>μια εναλλακτική μέθοδο διδασκαλίας με τη βοήθεια της εκπαιδευτικής ρομποτικής, ποιο πιστεύετε ότι θα ήταν το μεγαλύτερο πρόβλημα που θα αντιμετώπιζατε;</p>				
--	--	--	--	--

Αναλυτικά αποτελέσματα σε ερωτήσεις πολλαπλής επιλογής

5.3.3 Σύγκριση αποτελεσμάτων φύλλων αξιολόγησης

Σύγκριση αποτελεσμάτων φύλλου αξιολόγησης εμπειρίας

Τα αποτελέσματα του φύλλου αξιολόγησης εμπειρίας συγκεντρώθηκαν για κάθε τμήμα στους παραπάνω πίνακες. Η σύγκριση των αποτελεσμάτων ανά ομάδα μαθητών εξήγαγε αρκετά χρήσιμα συμπεράσματα γύρω από τη διδασκαλία.

Με ικανοποίηση διαπιστώθηκε πως οι μαθησιακοί στόχοι των δραστηριοτήτων του μαθήματος έγιναν σαφείς για όλους τους μαθητές με ολοκληρωτικό τρόπο (ερώτηση 1). Η άποψη των αρχαρίων, του πιο ευαίσθητου ίσως γκρουπ μαθητών, για τη δομή των δραστηριοτήτων και την περιγραφή αυτών ήταν πάλι η καλύτερη δυνατή (ερωτήσεις 2,3). Οι άλλες δύο μαθητικές ομάδες άγγιξαν επίσης το απόλυτο στις αντίστοιχες ερωτήσεις, με ένα μόνο μαθητή σε κάθε γκρουπ να μην μένει ικανοποιημένος.

Η δομή του διδακτικού πλάνου περιλάμβανε όλες τις παιδαγωγικές στρατηγικές που περιγράφονται στην ερώτηση 4, γι αυτό και επιλέχτηκαν από 4 αρχάριους, 2 έμπειρους και 3 στοιχειωδώς καταρτισμένους μαθητές. Η παραφωνία στα συγκεντρωτικά αποτελέσματα αφορούσε και πάλι έναν έμπειρο κι έναν στοιχειωδώς καταρτισμένο προγραμματιστή που απάντησαν πως η παιδαγωγική στρατηγική που χρησιμοποιήθηκε διέυρυνση της γνώσης των μαθητών στα πλαίσια ενός δοθέντος ζητήματος. Κατά την συγγραφή του φύλλου αξιολόγησης ο διδάσκων περίμενε ακόμη μεγαλύτερη απόκλιση σε απαντήσεις από αυτά τα δυο μαθητικά γκρουπ, καθώς η συμμετοχή τους σε αυτά προαπαιτούσε γνώσεις προγραμματισμού, άρα θα αντιμετώπιζαν την διδασκαλία ως μια προσπάθεια διέυρυνσης αυτής της γνώσης και μόνο. Όπως αποδείχθηκε, οι μαθητές που συμμετείχαν σε αυτές τις ομάδες δεν στην πλειοψηφία τους απάντησαν εγωιστικά, επιλέγοντας μια απάντηση που περιγράφει πλήρως τη διδασκαλία.

Προχωρώντας στην εξαγωγή συμπερασμάτων, διαπιστώθηκε για τρίτη φορά πως οι αρχάριοι μαθητές συμφώνησαν στο σύνολό τους σε μία απάντηση. Πιο συγκεκριμένα, υποστήριξαν πως όλες οι δεξιότητες που περιγράφονται στην ερώτηση 5 αποκτήθηκαν/υποστηρίχθηκαν από την διδασκαλία. Στις απαντήσεις των άλλων δύο ομάδων, παρουσιάστηκε η ίδια κατανομή επιλογών με πριν: τρεις έμπειροι και στοιχειωδώς καταρτισμένοι επέλεξαν το δ, ενώ ένας μόνο μαθητής από αυτά τα γκρουπ επέλεξε το β, δηλαδή την κατάλληλη χρήση των αισθητήρων και των γραφικών διεπαφών. Αυτή η επιλογή δεν μπορεί παρά να είναι προφανής κατά τη συμπλήρωση του φύλλου αξιολόγησης, συνεπώς οι υπόλοιπες δεξιότητες, που όντως υποστηρίχθηκαν κατά τη διάρκεια διδασκαλίας, δεν έγιναν άμεσα αντιληπτές από αυτούς τους δύο μαθητές.

Η ερώτηση που αφορούσε την αποσφαλμάτωση των προγραμμάτων (ερώτηση 6), απαντήθηκε από τους μαθητές ακριβώς όπως ο διδάσκων παρατήρησε κατά την

διεξαγωγή των μαθημάτων. Οι έμπειροι προγραμματιστές βρήκαν ιδιαίτερα αποτελεσματική την διαδικασία παρακολούθησης της συμπεριφοράς του ρομπότ και διόρθωσης του προγράμματος που αυτό «τρέχει» ώστε να επιδειχθεί η επιθυμητή συμπεριφορά από αυτό. Στις άλλες δύο ομάδες παρατηρήθηκε όμως μια σύγχυση κατά την διαδικασία αποσφαλμάτωσης, γεγονός που αποτυπώνεται και στα μοιρασμένα αποτελέσματα αυτής της ερώτησης.

Σχετικά με τους γενικούς στόχους της διδασκαλίας, η διασπορά των απαντήσεων των μαθητών παρουσίασε ενδιαφέρον(ερώτηση 8). Τρεις αρχάριοι μαθητές και ένας στοιχειωδώς καταρτισμένων προγραμματιστών θεώρησαν πως η εισαγωγή ενός νέου επιστημονικού θέματος υπήρξε ο κύριος στόχος της διδασκαλίας, με έναν εξ αυτών να απαντά πως η παρουσίαση μιας νέας προσέγγισης για αυτήν ήταν ο στόχος. Με την τελευταία επιλογή συμφώνησαν όλοι οι έμπειροι προγραμματιστές και τρεις εκ των στοιχειωδώς καταρτισμένων. Οι απαντήσεις των διαφορετικών ομάδων καταδεικνύουν πως η πλειοψηφία των μαθητών που διέθεταν κάποιες γνώσεις προγραμματισμού αντιμετώπισε το μάθημα ως έναν εναλλακτικό τρόπο να διδαχθεί αυτό το αντικείμενο, ενώ η πλειοψηφία των αρχάριων θεώρησε πως όλη η διδασκαλία αποτέλεσε την εξερεύνηση ενός νέου επιστημονικού ζητήματος. Αντίστοιχα, ο συνδυασμός των διδακτικών μεθόδων που χρησιμοποιήθηκαν από τον διδάσκοντα, κρίθηκε θετικά και από τις τρεις ομάδες μαθητών, με ένα μόνο μέλος κάθε μιας από αυτές να την χαρακτηρίζει μέτρια (ερώτηση 10).

Οι δραστηριότητες του διδακτικού πλάνου τοποθετήθηκαν από τον διδάσκοντα σε ένα φανταστικό πλαίσιο ιστορίας, μια επιλογή που ικανοποίησε όπως φαίνεται αρχάριους και στοιχειωδώς καταρτισμένους προγραμματιστές(ερώτηση 9). Στο σύνολό τους, οι μαθητές αυτών των ομάδων βρήκανε πολύ χρήσιμο αυτό το πλαίσιο. Από την άλλη, στην ομάδα των έμπειρων προγραμματιστών αναδείχθηκε διχογνωμία: ένας εξ αυτών χαρακτήρισε απλά χρήσιμο το ιστορικό υπόβαθρο των δραστηριοτήτων ενώ τους άλλους τρεις τους άφησε απλά αδιάφορους. Η επιλογή των έμπειρων μαθητών εύκολα μπορεί να δικαιολογηθεί, καθώς η επαφή τους με προγραμματιστικά εργαλεία και η συνήθως αυστηρή διατύπωση ερωτημάτων προς επίλυση, δεν αφήνει περιθώρια για ευφάνταστες εκφωνήσεις.

Οι μαθησιακοί στόχοι που υποστηρίχθηκαν κατά την διδασκαλία, αποδείχθηκε πως ικανοποίησαν κατά πολύ τους μαθητές των τριών ομάδων (ερώτηση 11) με λιγότερο δημοφιλή την πρόσκτηση στρατηγικών επίλυσης προβλήματος στο σύνολο των ομάδων και την συνεργατική επίλυση προβλήματος στην ομάδα των στοιχειωδώς καταρτισμένων, άποψη που επαληθεύεται ξανά από το συγκεκριμένο γκρουπ μαθητών με βάση τις απαντήσεις τους για την ποιότητα της συνεργασίας με το άλλο μέλος της ομάδας τους (ερώτηση 12).

Ερώτηση	Α			Β			Γ			Δ			Άλλο		
	Αρ.	Εμ.	Στ.	Αρ.	Εμ.	Στ.	Αρ.	Εμ.	Στ.	Αρ.	Εμ.	Στ.	Αρ.	Εμ.	Στ.
1. Είναι οι μαθησιακοί στόχοι των προτεινόμενων δραστηριοτήτων σαφείς;	Ναι			Όχι			-	-	-	-	-	-	-	-	-
	4	4	4	0	0	0	-	-	-	-	-	-	-	-	-
	Ναι			Όχι											
	4	3	3	0	1	1	-	-	-	-	-	-	-	-	-
2. Οι δραστηριότητες είναι καλά δομημένες;	Ναι			Όχι											
	4	3	3	0	1	1	-	-	-	-	-	-	-	-	-
3. Είναι η περιγραφή τους σαφής και κατανοητή;	Ναι			Όχι											
	4	3	3	0	1	1	-	-	-	-	-	-	-	-	-
4. Ποιές παιδαγωγικές στρατηγικές πιστεύετε ότι υποστηρίχθηκαν κατά τη διάρκεια του μαθήματος;	0	0	0	0	0	0	0	0	0	0	1	1	4	3	3
	Ανάπτυξη και εριμηνεία μέσω της γραφικής αναπαράστασης, των δεδομένων και της λύσης ενός προβλήματος			Κατάλληλη χρήση των αισθητήρων και των γραφικών διαπαφών			Ανάπτυξη δεξιοτήτων ή σχετικά με την διατύπωση υποθέσεων			Διεύρυνση της γνώσης των μαθητών στα πλαίσια ενός δοθέντος			Άλλες		
5. Υποθέτουμε ότι η διδασκαλία των εννοιών που διδασχθήκατε οδηγεί στην απόκτηση νέων δεξιοτήτων. Διαλέξτε από την παρακάτω λίστα τις δεξιότητες που αποκτήθηκαν ή υποστηρίχθηκαν από τη διδασκαλία:	0	0	0	0	1	1	0	0	0	4	3	3	-	-	-
	Πολύ ευκολότερη			Ευκολότερη			Δυσκολότερη			Πολύ			Δεν την		
6. Η χρησιμοποίηση του ρομπότ έκανε την αποσφαλμάτωση των προγραμμάτων:	0	4	0	2	0	2	2	0	2	0	0	0	-	-	-

Συγκριτικός πίνακας απαντήσεων μαθητών-1^ο μέρος

Σύγκριση αποτελεσμάτων φύλλου γενικής αξιολόγησης

Με βάση τις απαντήσεις των μαθητών όλων των ομάδων στο φύλλο γενικής αξιολόγησης, επιχειρήθηκε μια ποσοτική σύγκριση των απαντήσεων μεταξύ των τμημάτων, ώστε να εντοπιστούν διαφορετικές τάσεις σε μια σειρά από ζητήματα.

Ξεκινώντας με τις ερωτήσεις που αφορούν την φύση του μαθήματος(ερωτήσεις 1-7, «Το μάθημα...»), οι τρεις ομάδες έδειξαν να συμφωνούν με μέσο όρο άνω του ικανοποιητικού. Πιο συγκεκριμένα, ο απώτερος στόχος του μαθήματος, η ύλη, το υλικό, η προετοιμασία των μαθημάτων και η συνέπεια στο ωράριο του διδακτικού πλάνου αξιολογήθηκαν κατά πολύ πάνω του ικανοποιητικού, προσεγγίζοντας το άριστο. Η μόνη διαφωνία μεταξύ των ομάδων προέκυψε –όπως αναμενόταν άλλωστε- στη κρίση τους για το επίπεδο δυσκολίας των μαθημάτων. Οι ομάδες των έμπειρων, των στοιχειωδών καταρτισμένων και των αρχάριων έκριναν ως λίγο, φυσιολογικά και πολύ δύσκολο το μάθημα αντίστοιχα. Η διαφοροποίηση αυτή δεν προκάλεσε έκπληξη καθώς οι δύο πρώτες ομάδες είχαν ήδη μια επαφή με τον προγραμματισμό, ενώ για την ομάδα των αρχαρίων η παρούσα διδασκαλία αποτέλεσε και την παρθενική τους επαφή.

Η συνέχεια του φύλλου αξιολόγησης αφορούσε τις εργασίες που ανατέθηκαν στους μαθητές κατά την διάρκεια της διδασκαλίας.(ερωτήσεις 8-13). Η ομάδα των αρχαρίων δυσκολεύτηκε σαφώς περισσότερο σε σχέση με τις άλλες δύο ομάδες, να κατανοήσει τι ακριβώς ζητούσαν οι εκφωνήσεις. Μάλιστα θα επιθυμούσαν να έχουν και περισσότερο χρόνο στην διάθεση τους για να απαντήσουν, απαίτηση που δεν εκδηλώθηκε από τις άλλες ομάδες. Η κρίση σχετικά με το μάθημα ολοκληρώθηκε με τις τρεις ομάδες να συμφωνούν πως θα επιθυμούσαν μεγαλύτερη καθοδήγηση από τον διδάσκοντα, αλλά και πως η συμβολή του με σχόλια ,όπου αυτά έγιναν, υπήρξε διαφωτιστική. Το αποτέλεσμα αυτό θα μπορούσε να είναι οξύμωρο για τις ομάδες των έμπειρων και των στοιχειωδών καταρτισμένων προγραμματιστών, καθώς από τις απαντήσεις τους και τις εντυπώσεις που άφηναν κατά την διάρκεια του μαθήματος, μια περαιτέρω καθοδήγηση δεν κρίθηκε αναγκαία από την πλευρά του διδάσκοντα. Λαμβάνοντας υπόψη όμως την τυπική εργαστηριακή διδασκαλία, μπορούμε να πούμε πως οι δύο αυτές ομάδες περίμεναν η συμπεριφορά του διδάσκοντα να την προσεγγίζει. Από την άλλη, ένας ακόμη πιο ενεργός ρόλος του διδάσκοντα κατά την παράδοση εργασιών σε τμήμα αρχαρίων, είναι θεμιτός.

Πέρα από την κρίση του ρόλου του διδάσκοντα κατά την παράδοση εργασιών, οι μαθητές κλήθηκαν να αξιολογήσουν και τη γενικότερη συμπεριφορά του κατά την διάρκεια της διδασκαλίας (ερωτήσεις 14-23). Οι εντυπώσεις και των τριών γκρουπ για αυτόν συνέπιπταν ως θετικές. Η κοινή συνισταμένη αυτών των ερωτήσεων αναδεικνύει τον διδάσκοντα ως συνεπή, συγκαταβατικό, προσιτό και σαφή στις εξηγήσεις του, με μόνη παραφωνία την σχετικά χαμηλή εκτίμηση των έμπειρων προγραμματιστών για την ατμόσφαιρα εμπιστοσύνης και σεβασμού που δημιουργήσε στην τάξη. Μια πιθανή εξήγηση για αυτό το αποτέλεσμα είναι οι συνομιλίες του διδάσκοντα με μέλη της ομάδας των έμπειρων χρηστών, όταν αυτά είχαν ολοκληρώσει τις δραστηριότητες τους, και οι μεμονωμένες ερωτήσεις του για πιθανές βελτιώσεις των δραστηριοτήτων.

Ακόμη ένας τομέας όπου ζητήθηκε να κρίνουν οι συμμετέχοντες υπήρξε το περιβάλλον διδασκαλίας (ερωτήσεις 24-31). Ο εργαστηριακός εξοπλισμός και η ποιότητα του χώρου διδασκαλίας αποδείχθηκαν ικανοποιητικά επαρκείς για τους μαθητές και των τριών ομάδων. Διαφοροποίηση παρουσιάστηκε ανάμεσα στις ομάδες στο κατά πόσο τα

εργαστηριακά μέσα διευκόλυναν την εκπαιδευτική διαδικασία. Η ομάδα των έμπειρων χρηστών αποδείχτηκε και εδώ απαιτητική, δίνοντας και εδώ κατά μέσο όρο, βαθμολογία μικρότερη του ικανοποιητικού, σε αντίθεση με τις άλλες δύο ομάδες που δεν προβληματίστηκαν με τα εργαστηριακά μέσα. Γνωρίζοντας ήδη την άποψη κάθε ομάδας συμμετεχόντων για το Scribbler και τη γραφική του διεπαφή, λόγω των συνεδριών μετά από κάθε μάθημα αλλά και της ίδιας της εμπειρίας του διδάσκοντα κατά την διάρκεια της διδασκαλίας, τα αποτελέσματα στις ερωτήσεις κρίσεως του ρομπότ δεν αποτέλεσαν έκπληξη. Το γκρουπ των αρχάριων μαθητών, έμεινε πάρα πού ικανοποιημένο, τόσο από το ρομπότ όσο και από τη γραφική διεπαφή. Όσο όμως ανεβάνουμε σε επίπεδο εμπειρίας στον προγραμματισμό, τόσο η ικανοποίηση φθίνει. Έτσι, το τμήμα των στοιχειωδώς καταρτισμένων προγραμματιστών αποδείχθηκε ικανοποιημένο αισθητά άνω του μετρίου (στη σχετική κλίμακα), ενώ η μέτρηση στο τμήμα των έμπειρων προσέγγιζε την απόλυτη μετριότητα. Ένα τέτοιο αποτέλεσμα μόνο ως φυσιολογικό μπορεί να κριθεί, καθώς όσο πιο έμπειρος είναι ένας προγραμματιστής, τόσο μεγαλύτερες απαιτήσεις έχει και από τα εργαλεία που χρησιμοποιεί. Τέλος, παρά την διαφοροποίηση στις απόψεις των μαθητών για το Scribbler και τη γραφική του διεπαφή, κάποια μικρά προβλήματα που παρουσιάστηκαν κατά την κρίση των υπόλοιπων εργαστηριακών μέσων δεν τους έπεισαν να διαμορφώσουν αρνητική άποψη.

Το φύλλο αξιολόγησης έκλεισε με γενικές ερωτήσεις για την ακαδημαϊκή αξία και συνολικές εκτιμήσεις για μια τέτοιου είδους εναλλακτική διδασκαλία (ερωτήσεις 32-36). Οι ομάδες των στοιχειωδώς καταρτισμένων προγραμματιστών και των αρχαρίων, έδειξαν μέσω των απαντήσεων τους πως το μάθημα αποτέλεσε μια σημαντική εμπειρία για αυτούς και είναι πολύ πιθανό να τους βοηθήσει στο μέλλον σε κάποια απόπειρα ενασχόλησης τους με τον προγραμματισμό. Τα αποτελέσματα για τα δύο αυτά γκρουπ, δεν αφήνουν περιθώριο παρερμηνείας; η διδασκαλία τους φάνηκε ιδιαίτερα ευχάριστη και ο στόχος της εισαγωγής στις βασικές προγραμματιστικές έννοιες είχε επιτευχθεί. Παράλληλα, η ομάδα των έμπειρων προγραμματιστών, δεν έκρινε η διδασκαλία θα τους φανεί ιδιαίτερα χρήσιμη στην πορεία τους στον προγραμματισμό, ούτε βρήκε δύσκολο το αντικείμενο, αλλά είναι σίγουρο πως αποτέλεσε για αυτούς μια ενδιαφέρουσα διδακτική προσέγγιση σε έννοιες με τις οποίες ήταν ήδη εξοικειωμένοι.

Συγκριτικός πίνακας μέσων όρων για κάθε τμήμα μαθητών

Ερώτηση	ΜΟ Τμήματος Έμπειρων	ΜΟ Τμήματος Στοιχειωδώς καταρτισμένων	ΜΟ Αρχαρίων
1. Οι συνολικός στόχος του μαθήματος ήταν σαφής;	5	4,75	4,5
2. Η ύλη που καλύφθηκε ανταποκρινόταν στους στόχους του μαθήματος;	4,5	4,75	5
3. Τα μαθήματα ήταν καλά προετοιμασμένα και οργανωμένα;	4,75	4,75	5
4. Το υλικό που χρησιμοποιήθηκε βοήθησε στην κατανόηση του μαθήματος;	4,5	4,75	4,5
5. Πώς κρίνεται το επίπεδο δυσκολίας του μαθήματος;	2,5	3,25	4,5
6. Χρησιμοποιήθηκαν άλλα σύγχρονα μέσα διδασκαλίας (βίντεο, παρουσιάσεις ,κα.);	4,25	3,75	4
7. Τηρήθηκε το προβλεπόμενο	5	5	5

ωράριο που έγινε γνωστό πριν την έναρξη του μαθήματος;			
8. Οι εκφωνήσεις ήταν κατανοητές;	4,5	4,75	3,5
9. Η χρονική προθεσμία για την υποβολή των λύσεων ήταν λογική;	4	4	3,5
10. Υπήρχε καθοδήγηση από τον διδάσκοντα;	3,75	3,75	3,5
11. Τα σχόλια του διδάσκοντα ήταν εποικοδομητικά και αναλυτικά;	4,25	4,75	4,5
12. Δόθηκε η δυνατότητα βελτίωσης της εργασίας;	3	3,5	4
13. Η συγκεκριμένη εργασία σας βοήθησε να κατανοήσετε το συγκεκριμένο θέμα;	4,5	4,5	4,75
14. Οργάνωσε καλά την παρουσίαση της ύλης στα μαθήματα;	4,5	4,75	4,75
15. Πέτυχε να διεγείρει το ενδιαφέρον σας για το αντικείμενο του μαθήματος;	4,5	4,75	4,75
16. Ανάλυσε και παρουσίασε τις έννοιες με τρόπο απλό και ενδιαφέροντα χρησιμοποιώντας παραδείγματα;	4,75	4	4
17. Δημιούργησε μια ατμόσφαιρα σεβασμού και εμπιστοσύνης μέσα στην τάξη;	3,75	4,5	4,25
18. Συνδύασε την ύλη του μαθήματος με πραγματικά παραδείγματα/εφαρμογές;	5	4,75	4,5
19. Σας έδωσε να καταλάβετε πως οι διάφορες ενότητες του μαθήματος σχετίζονται μεταξύ τους;	4,5	5	4,75
20. Σας ενθάρρυνε να διατυπώσετε ερωτήσεις και απορίες ώστε να αναπτύξετε την κρίση σας;	3,75	3,75	4,5
21. Ήταν συνεπής στις υποχρεώσεις του;	5	5	5
22. Ήταν γενικά προσιτός κατά τη διάρκεια του μαθήματος;	4,75	4,75	4
23. Οι εξηγήσεις του ήταν ξεκάθαρες;	4,75	5	4,75
24. Ήταν επαρκής ο εξοπλισμός του εργαστηρίου;	3,75	3,75	4,5
25. Αξιολογήστε την ποιότητα του χώρου που έγινε η διδασκαλία του μαθήματος	4	3,75	3,75
26. Τα εργαστηριακά μέσα διευκόλυναν τις δραστηριότητες;	3,75	4	4
27. Πόσο ικανοποιημένοι μείνατε από το Scribbler;	3,25	3,5	4,75
28. Πως ανταποκρίθηκε στις εργασίες που του ζητήσατε;	3,75	4	4,75
29. Πόσο ικανοποιημένοι μείνατε από τη γραφική διεπαφή του Scribbler;	3,25	3,75	4,75
30. Πως ανταποκρίθηκαν οι υπολογιστές του εργαστηρίου;	4,5	5	4,75
31. Τα εργαστηριακά μέσα συνδυάστηκαν	4,5	4,75	4,75
32. Έχω μάθει και καταλάβει το αντικείμενο του μαθήματος	5	4,5	4
33. Το μάθημα είναι υψηλών απαιτήσεων διανοητικά και διεγείρει το ενδιαφέρον;	3,25	4,25	4,75
34. Το μάθημα θα σας φανεί σε κάποια στιγμή της ζωής σας;	3,75	5	4,5
35. Ο καθηγητής ήταν ικανός και αποτελεσματικός για τη διδασκαλία του συγκεκριμένου αντικειμένου;	4,5	4,5	4,75
36. Ο καθηγητής συνεισέφερε θετικά	4,5	4,25	5

στην προαγωγή της γνώσης μου για το αντικείμενο του μαθήματος;			
--	--	--	--

5.3.4 Αποτελέσματα αξιολόγησης με συνεδρίες

1^ο μάθημα

Με την ολοκλήρωση του πρώτου μαθήματος και κατά την διάρκεια της πρώτης συνεδρίας στις τρεις ομάδες μαθητών, ζητήθηκε από αυτούς να περιγράψουν αρχικά τις εμπειρίες που αποκόμισαν αλλά και τυχόν προβλήματα που αντιμετώπισαν. Στο σύνολο των τριών ομάδων, οι μαθητές ισχυρίστηκαν πως το μάθημα ήταν αρκετά ενδιαφέρον ενώ η πρώτη επαφή τους με το ρομπότ τους άφησε ενθουσιασμένους. Οι ομάδες των έμπειρων προγραμματιστών και αυτών με μια σχετική εμπειρία δεν αντιμετώπισαν κάποιο πρόβλημα κατά τη διάρκεια ολοκλήρωσης των δραστηριοτήτων. Αντιθέτως, δύο μέλη της ομάδας των αρχαρίων ισχυρίστηκαν πως δυσκολεύθηκαν σε μικρό βαθμό κατά την εκτέλεση του 4^{ου} και 5^{ου} φύλλου εργασίας, αλλά όχι τόσο στην 6^η. Αιτία αυτής της δυσκολίας φαίνεται πως ήταν η εξοικείωση με το γραφικό περιβάλλον του Scribbler και τη σύνταξη του προγράμματος. Το πρόβλημα αυτό ξεπεράστηκε στο 6^ο φύλλο εργασίας όπου συνεργαζόμενοι έλυσαν οποιαδήποτε διαδικαστικά προβλήματα είχαν παρουσιαστεί. Στην ερώτηση των διδασκόντων σχετικά με το αν πιστεύουν πως θα δυσκολευτούν περισσότερο στα επόμενα μαθήματα, η ομάδα έμπειρων προγραμματιστών απάντησε σύσσωμη πως όχι, εν αντιθέσει με τις δύο άλλες ομάδες που παρουσιάστηκαν επιφυλακτικές στο σύνολό τους.

Συμπερασματικά, μπορούμε να πούμε πως μετά την ολοκλήρωση της πρώτης συνεδρίας με τις τρεις ομάδες, οι έμπειροι προγραμματιστές έδειχναν να απολαμβάνουν τη διδασκαλία και να την αντιμετωπίζουν ως ένα δημιουργικό παιχνίδι. Από την άλλη, η ομάδα των στοιχειωδώς καταρτισμένων στον προγραμματισμό δεν φαινόταν και ούτε δήλωσε πως δυσκολεύτηκε, είχε όμως επιφυλάξεις για το αν θα κατάφερνε να ανταπεξέλθει στα επόμενα μαθήματα. Τέλος, η ομάδα των αρχαρίων φάνηκε πως διαθέτει ακόμη λιγότερη αυτοπεποίθηση, χωρίς όμως να μειώνεται ο ενθουσιασμός τους για το μάθημα.

2^ο μάθημα

Η συνεδρία του δεύτερου μαθήματος με το τμήμα των έμπειρων κύλησε όπως και το μάθημα : απροβλημάτιστα. Οι μαθητές αντιλήφθηκαν εξ αρχής τον τρόπο με τον οποίο δουλεύει η επαναληπτική δομή της γραφικής διεπαφής του Scribbler, ξεπερνώντας ακόμη και ερωτήσεις παγίδες των τυπικών μαθησιακών δυσκολιών αυτής. Ο ενθουσιασμός τους εξακολουθούσε να βρίσκεται σε υψηλά επίπεδα, ενώ ήδη έκαναν τις πρώτες σκέψεις για κατασκευή ενός προγράμματος δικιάς τους έμπνευσης και αντάλλασαν απόψεις σχετικά με αυτό. Μόνο ίσως μελανό σημείο της συνεδρίας ήταν η ένταση ενός μαθητή σχετικά με τις δυνατότητες του ρομπότ και η συναίνεση των υπολοίπων. Ανεξάρτητα με αυτή την παρατήρηση όμως, κανένας από αυτούς δεν ισχυρίστηκε πως το τρίτο μάθημα που ακολουθούσε θα τους φαινόταν βαρετό. Τέλος, δύο εκ των μαθητών θα προτιμούσαν λιγότερες δραστηριότητες και πιο συμπυκνωμένο διδακτικό χρόνο.

Με την ομάδα των στοιχειωδώς καταρτισμένων στον προγραμματισμό μαθητών, η συνεδρία δεν ήταν τόσο ομαλή. Ενώ αντιλήφθηκαν την λειτουργία της δομής επανάληψης και τον παραλληλισμό της με τις ενέργειες του Scribbler, τρεις εξ αυτών

δυσκολεύτηκαν να απαντήσουν σωστά στα ερωτήματα που ακολουθούσαν τις δραστηριότητες. Επιπρόσθετα, η μία εκ των δύο ομάδων που συστάθηκαν για την επίλυση του 4^{ου} φύλλου εργασίας δεν έμεινε ικανοποιημένη από την συνεργασία των μελών της και είχε έντονες διαφωνίες για την επίλυση. Σε ερώτηση των διδασκόντων για την αιτία της παρερμηνείας της επαναληπτικής δομής, οι μαθητές της ομάδας που αντιμετώπισε συνεργατικά προβλήματα απάντησαν πως ήταν βέβαιοι ότι είχαν αντιληφθεί πλήρως τις ιδιαιτερότητες της εντολής, γι αυτό και διαφώνησαν. Ο τρίτος μαθητής που αντιμετώπισε δυσκολίες ισχυρίστηκε πως αυτές παρουσιάστηκαν κυρίως στο δεύτερο και τρίτο φύλλο εργασίας, αλλά μόνο στα αρχικά ερωτήματα αυτών. Στη συνέχεια αντιλήφθηκε τις παγίδες των ερωτήσεων που αναφερόταν στις τυπικές μαθησιακές δυσκολίες και έχοντας άριστη συνεργασία με το μέλος της ομάδας του, διέλυσε οποιεσδήποτε αμφιβολίες κατά την επίλυση του 4^{ου} φύλλου εργασίας.

Παρόλα τα προβλήματα που παρουσιάστηκαν σε αυτό το γκρουπ των μαθητών, δήλωναν πλέον πολύ πιο αισιόδοξοι για την πορεία τους στη διδασκαλία. Το ενδιαφέρον αυτών φάνηκε αμείωτο, ενώ έμειναν και ικανοποιημένοι από την ροή του μαθήματος.

Η συνεδρία με την ομάδα των αρχάριων είχε απρόσμενα ευχάριστη εξέλιξη. Έκτος από την αρχική παρερμηνεία σχετικά με τον ατέρμονα βρόγχο που αντιμετώπισαν όλοι οι μαθητές αλλά εύκολα αντιμετωπίστηκε και διορθώθηκε, κανείς τους δεν αντιμετώπισε περαιτέρω προβλήματα. Τρεις εκ των μαθητών ισχυρίστηκαν πως η αφοσίωση τους στο μάθημα και το ευχάριστο αντικείμενο απλοποίησαν τη διαδικασία της κατανόησης, ενώ ένας εξ αυτών δήλωσε πως η χαρακτηριστική ευκολία των θεμάτων ήταν ο κύριος παράγοντας. Οι δραστηριότητες του μαθήματος φάνηκαν σε όλους ευφάνταστες, αλλά πιο ελκυστικό όλων χαρακτήρισαν το 4^ο φύλλο εργασίας. Η συνεργασία τους σε ομάδες υποστήριξαν πως υπήρξε άπταιστη, ενώ ο ρυθμός του μαθήματος αρκετά καλός. Στην ερώτηση των διδασκόντων για την αιτία της παρερμηνείας του ατέρμονου βρόγχου, οι μαθητές υποστήριξαν στο σύνολό τους πως δεν ήταν προφανής η υλοποίηση της γραφικής διεπαφής. Η εξέλιξη των μαθημάτων τους άφηνε απόλυτα ικανοποιημένους, κυρίως λόγω της ύπαρξης του Scribbler κατά δήλωση τριών μαθητών και λόγω του ιστορικού υποβάθρου των δραστηριοτήτων, όπως υποστήριξε ο τέταρτος μαθητής. Τέλος, στην ερώτηση κατά πόσο πιστεύουν πως θα ανταπεξέλθουν στις δραστηριότητες του τρίτου μαθήματος που βρισκόταν ενόψει, δύο εξ αυτών απάντησαν πως πεποίθησή τους ήταν ότι δεν θα αντιμετώπιζαν κάποιο πρόβλημα, ενώ οι άλλοι δύο εξακολουθούσαν να διατηρούν επιφυλάξεις, αλλά σαφώς λιγότερες σε σχέση με τη συνεδρία του τρίτου μαθήματος.

3^ο μάθημα

Η τελευταία συνεδρία με το τμήμα των έμπειρων στον προγραμματισμό μαθητών υπήρξε πιο ενδιαφέρουσα σε σχέση με τις προηγούμενες. Οι μαθητές αυτής της ομάδας ερωτήθηκαν τόσο σχετικά για το μάθημα που μόλις είχε ολοκληρωθεί, όσο και για τις συνολικές εντυπώσεις τους συγκριτικά με τις προηγούμενες μορφές διδασκαλίας στις οποίες είχαν συμμετάσχει.

Σχετικά με το τρίτο μάθημα, όπως ήταν αναμενόμενο άλλωστε, κανείς τους δεν αντιμετώπισε προβλήματα όσον αφορά τις δραστηριότητες της διδασκαλίας. Τα σχόλια γύρω από την εφαρμογή της εντολής ελέγχου διέφεραν, έχοντας όμως μια κοινή θετική συνισταμένη. Κύριο συμπέρασμα γύρω από αυτήν ήταν πως η εφαρμογή της εντολής ελέγχου, που όλους ανεξαιρέτως τους δυσκόλεψε κατά την πρώτη επαφή τους με τον

προγραμματισμό, απλοποιούσε κατά πολύ οποιεσδήποτε δυσκολίες κατανόησης της. Εν αντιθέσει με την εντολή επανάληψης, η εντολή ελέγχου του Scribbler τους άφηνε σε γενικές γραμμές ικανοποιημένους λόγω της δυνατότητας σύνδεσης πολλών τέτοιων εντολών και της δημιουργίας σύνθετων δομών. Κατά δήλωση και των τεσσάρων μαθητών που συμμετείχαν στις δύο ομάδες του 3^{ου} φύλλου εργασίας, αυτή η δραστηριότητα υπήρξε η πιο ενδιαφέρουσα όλων των μαθημάτων εξαιτίας του συνδυαστικού της χαρακτήρα. Δύο εκ των μαθητών χαρακτήρισαν περιττές τις υπόλοιπες δραστηριότητες του τρίτου φύλλου εργασίας, ενώ οι άλλοι δυο αναγκάιες για την πρόοδο του μαθήματος.

Στην ερώτηση των διδασκόντων για τις συνολικές εντυπώσεις από το μάθημα υπήρξε διάσταση απόψεων. Ένας εκ των μαθητών υποστήριξε πως το διδακτικό πλάνο υπήρξε άρτιο και ο εναλλακτικός τρόπος διδασκαλίας σαφώς καλύτερος των διαλέξεων-εργαστηρίων, αλλά θα προτιμούσε μια ρομποτική εφαρμογή που θα υποστήριζε σύνθετες προγραμματιστικές δομές. Ο δεύτερος μαθητής του γκρουπ, βρήκε το διδακτικό πλάνο αρκετά «σφιχτό» και θα προτιμούσε να δοθεί στους μαθητές περισσότερος χρόνος ελεύθερης ενασχόλησης με το ρομπότ. Ισχυρίστηκε ακόμη, ότι ο ρόλος των διδασκόντων ήταν αρκετά παθητικός και θα προτιμούσε πολλή μεγαλύτερη συμμετοχή τους στο μάθημα. Οι άλλοι δύο μαθητές βρήκανε το μάθημα ιδανικό για ένα τμήμα αρχαρίων και δεν επεσήμαναν κάποιο πρόβλημα είτε με τη δομή αυτού, είτε με το ρόλο των διδασκόντων.

Το γκρουπ με τους στοιχειωδώς καταρτισμένους προγραμματιστές αποδείχθηκε το πιο ενδιαφέρον από τα τρία. Οι μαθητές δεν δυσκολεύτηκαν κατά την αναγνώριση και τη σύνταξη των δομών επιλογής, ούτε κατά την σύνταξη αντίστοιχων εντολών στην γραφική διεπαφή του Scribbler. Αντίθετα, δύο εξ αυτών δήλωσαν πως μέχρι την συμπλήρωση του φύλλου εργασίας 3, η πιο δύσκολη δραστηριότητα για αυτούς ήταν η συμπλήρωση των κενών σε φυσική γλώσσα για σύνθετες δομές ελέγχου καθώς και των πινάκων αληθείας. Η σύνθεση ταυτόχρονων συνθηκών ελέγχου μπορούσε εύκολα να γίνει αντιληπτή σε πρόγραμμα του Scribbler αλλά όχι βλέποντας τις κινήσεις του ίδιου του ρομπότ και καταγράφοντας τη συμπεριφορά του. Ως αιτία αυτής της δυσκολίας προβλήθηκε η αδυναμία παρακολούθησης όλων των διαδοχικών λειτουργιών του ρομπότ και η μετατροπή αυτών σε έλεγχο. Για τους άλλους δύο μαθητές δεν εμφανίστηκε κάποια παρερμηνεία σχετικά με την εντολή ελέγχου, εκτός ίσως από μερικές ενστάσεις του ενός για τις απεικονίσεις του ελέγχου των διαφορετικών αισθητήρων της γραφικής διεπαφής του Scribbler.

Μετά την ενασχόληση με το 3^ο φύλλο εργασίας, οι μαθητές υποστήριξαν πως τα περισσότερα των παραπάνω προβλημάτων ξεπεράστηκαν. Η παρερμηνεία για τις εντολές ελέγχου αποσαφηνίστηκε πλήρως μέσα από το πίνακα της εκφώνησης και την βοήθεια του διδάσκοντα κατά την συγγραφή του τελικού προγράμματος. Οι μαθητές που δυσκολεύτηκαν με την συμπλήρωση των πινάκων αληθείας και την σύνταξη των δομών ελέγχου σε φυσική γλώσσα, δήλωσαν πως η καλώς ορισμένη ροή των ελέγχων του 3^{ου} φύλλου εργασίας, αλλά και η αποσφαλμάτωση των λανθασμένων προγραμμάτων, βελτίωσαν κατά πολύ την αντίληψη τους για τη φιλοσοφία της πραγματοποίησης των ελέγχων. Σε ερώτηση του διδάσκοντα, πως πιστεύουν ότι θα αντιδρούσαν τώρα κατά την συμπλήρωση των δραστηριοτήτων που πριν τους δυσκόλεψαν, και οι δύο μαθητές ισχυρίστηκαν πως θα τα πήγαιναν πολύ καλύτερα.

Τέλος, διατυπώθηκαν ερωτήσεις για τις συνολικές εντυπώσεις των μαθητών από τη διδασκαλία. Σε αντίθεση με το προηγούμενο, απαιτητικό γκρουπ των έμπειρων προγραμματιστών, τρεις μαθητές αυτού του γκρουπ δήλωσαν απόλυτα ικανοποιημένοι τόσο από το διδακτικό πλάνο όσο και από την εφαρμογή του. Σε συνεχόμενες ερωτήσεις των διδασκόντων για τον εντοπισμό οποιουδήποτε αρνητικού σημείου της διδασκαλίας, οι τρεις μαθητές επέμειναν στην αρχική τους άποψη. Ο τέταρτος μαθητής έκανε 2 παρατηρήσεις για το σύνολο των μαθημάτων, σχετικά με την αποτελεσματικότητα των συνεργατικών δραστηριοτήτων και την συμπεριφορά των διδασκόντων. Αναλυτικότερα, ο μαθητής ισχυρίστηκε πως θα προτιμούσε οποιαδήποτε δραστηριότητα να πραγματοποιείται ατομικά ώστε να μην αποσπάται η προσοχή του από το ίδιο το αντικείμενο αυτής και μπορεί να επιλύσει οποιοδήποτε πρόβλημα ανακύψει βασιζόμενος στις δικές του ικανότητες. Σχετικά με την συμπεριφορά των διδασκόντων, ο μαθητής τη χαρακτήρισε αρκετά καλή αλλά με πολλά περιθώρια βελτίωσης, ειδικά στον τομέα επικοινωνίας με τους μαθητές.

Η τελευταία συνεδρία με τους αρχάριους μαθητές μόνο ως πλούσια μπορεί να χαρακτηριστεί. Κανείς τους δεν δήλωσε πως αντιμετώπισε προβλήματα στο πρώτο φύλλο εργασίας, ενώ στη δραστηριότητα της παρατήρησης του ρομπότ και καταγραφής της συμπεριφοράς του με μορφή συνθήκης ελέγχου μόνο ένας μαθητής δυσκολεύτηκε. Το πρόβλημα δημιουργήθηκε γιατί ο μαθητής δεν αντιλήφθηκε την εκφώνηση της δραστηριότητας πριν το ρομπότ εκτελέσει το πρόγραμμα επίδειξης. Στο δεύτερο φύλλο εργασίας, διατυπώθηκαν αρκετοί προβληματισμοί σχετικά με τη δεύτερη δραστηριότητα. Στα πρώτα ερωτήματα αυτής, οι αρχάριοι ήρθαν για πρώτη φορά σε επαφή με μια σύνθετη δομή ελέγχου και όπως αναμενόταν τους δημιουργήθηκε μια μικρή σύγχυση. Απαντώντας διαδοχικά στα ερωτήματα της δεύτερης αλλά και της τρίτης δραστηριότητας, όλοι οι μαθητές απάντησαν πως κατανόησαν πλήρως τις σύνθετες δομές, 2 όμως εξ αυτών παραδέχτηκαν πως η τοποθέτηση των μπλοκ ελέγχου στη γραφική διεπαφή του Scribbler τους φάνηκε δυσνόητη. Παρότι αρκετά προβλήματα αναφέρθηκαν για τα δυο πρώτα φύλλα εργασίας, στον σχολιασμό του τρίτου εμφανίστηκαν ενθουσιασμένοι. Ο συνδυαστικός του χαρακτήρας σχολιάστηκε ως απαιτητικός αλλά σε φυσιολογικό βαθμό, ενώ αντίστοιχα, η προϋπόθεση της συνεργατικής επίλυσης χρήσιμη. Τέλος, το σενάριο εκτέλεσης αυτού του φύλλου υπήρξε διασκεδαστικό σύμφωνα με τους μαθητές. Σε ένα τελευταίο σχόλιο σχετικά με το τρίτο μάθημα, τρεις εκ των συμμετεχόντων το χαρακτήρισαν αρκετά πιο δύσκολο από τα προηγούμενα, αλλά σαφώς πιο ενδιαφέρον. Από την άλλη, ο τέταρτος συμμετέχων δήλωσε πως η δυσκολία του ήταν ίδιου βαθμού με το προηγούμενο και ότι το περίμενε σαφώς δυσκολότερο.

Κλείνοντας με την διδασκαλία του τρίτου μαθήματος, οι αρχάριοι κλήθηκαν να σχολιάσουν τις γενικότερες συνθήκες του μαθήματος, του περιβάλλοντος διδασκαλίας αλλά και τους ίδιους τους διδάσκοντες. Η άποψη όλων για τα παραπάνω παρέμεινε εξαιρετικά καλή, αν και όλοι συμφώνησαν πως δραστηριότητες με μικρότερη κλιμακούμενη δυσκολία θα απλοποιούσαν κατά πολύ την διδασκαλία.

Τέλος, σε ένα γενικότερο σχολιασμό της συνολικής εμπειρίας των μαθητών, τρεις εξ αυτών απάντησαν πως ήταν μια εξαιρετική εμπειρία και θα το επαναλάμβαναν στο μέλλον αν τους δοθεί η ευκαιρία, ενώ δεν τους άφηνε ασυγκίνητους η προοπτική να ασχοληθούν με ένα κλασικό προγραμματιστικό εργαλείο. Ο τέταρτος μαθητής βρήκε την εναλλακτική διδασκαλία με τη βοήθεια του Scribbler ικανοποιητική, αλλά δεν τον

έλκυε ο προγραμματισμός ως επιστημονικό αντικείμενο. Ο ίδιος μαθητής δήλωσε πως η ύπαρξη του Scribbler τον παρότρυνε να συνεχίσει με μεγάλο ενδιαφέρον τη διδασκαλία, αλλά με μια πιο σύνθετη ρομποτική εφαρμογή δεν θα μπορούσε να ανταπεξέλθει στις απαιτήσεις του μαθήματος.

6.Συζήτηση

Η εφαρμογή σύγχρονων τεχνολογιών στην εκπαίδευση αποτέλεσε εξ αρχής ένα ιδιαίτερα ενδιαφέρον αντικείμενο για τον γράφοντα, γι αυτό και επιλέχθηκε από αυτόν ως θέμα της διπλωματικής εργασίας. Η τάση σε πολλά αναγνωρισμένα εκπαιδευτικά ιδρύματα να χρησιμοποιείται η ρομποτική για την διδασκαλία του προγραμματισμού δεν φαίνεται να ακολουθείται σε ικανοποιητικό βαθμό και στην Ελλάδα, παρότι αντιμετωπίζεται πλέον ως ένα χρήσιμο εκπαιδευτικό εργαλείο. Η εκπόνηση αυτής της εργασίας αντιμετωπίστηκε από την ανάληψη της ως μια προσπάθεια προς την διάδοση αυτών των σύγχρονων πρακτικών, μιας και η δυσκολία εκμάθησης του προγραμματισμού είναι δεδομένη και στην χώρα μας.

Αντικειμενικός στόχος της εργασίας δεν υπήρξε μόνο η ίδια η διδασκαλία με τη χρήση μιας ρομποτικής εφαρμογής. Η ανάλυση των τυπικών μαθησιακών δυσκολιών του προγραμματισμού, η παρουσίαση και ανάλυση των εκπαιδευτικών εργαλείων που έχουν χρησιμοποιηθεί για την διδασκαλία του κατά καιρούς, η κατάστρωση ενός αναλυτικού διδακτικού πλάνου, η εφαρμογή του σε ομάδες με διαφορετικό επίπεδο γνώσης και η εξαγωγή συγκριτικών αποτελεσμάτων αποτέλεσαν εξίσου σημαντικά ζητήματα για αυτή την εργασία.

Το Scribbler robot της Parallax που χρησιμοποιήθηκε για την διεκπεραίωση της διδασκαλίας φάνταζε ως ιδανικό: οι απαιτήσεις του σε προγραμματιστικές γνώσεις είναι μηδαμινές, το κόστος του χαμηλό και οι απαιτήσεις του σε εργαστηριακό εξοπλισμό ελάχιστες. Μια διδασκαλία όμως σε ομάδες μόνο αρχάριων μαθητών δεν θα μπορούσε να εξάγει συμπεράσματα για την ποιότητα της γνώσης που προσφέρει τόσο το ρομπότ, όσο και ο διδάσκων. Έτσι, αποφασίστηκε η ίδια διδασκαλία να εφαρμοστεί σε τρεις ομάδες με διαφορετικό επίπεδο γνώσης και να εξαχθούν από κάθε ομάδα συμπεράσματα που σε άλλη περίπτωση θα ήταν αδύνατο να έχουμε.

Το διδακτικό πλάνο υπήρξε κοινό και για τις τρεις ομάδες, ώστε τα αποτελέσματα της εφαρμογής της διδασκαλίας να είναι εύκολα συγκρίσιμα. Κατά την δημιουργία του, ο γράφων επιχείρησε να ισορροπήσει ανάμεσα στην αυστηρή αντίληψη πραγματοποίησης μαθημάτων και στην παροχή όλων των εχεγγύων στους συμμετέχοντες ώστε να πειραματιστούν και να «παίξουν» δημιουργικά με την ρομποτική εφαρμογή.

Η προετοιμασία για την πραγματοποίηση των μαθημάτων δεν υπήρξε εύκολη. Η σύνταξη του διδακτικού πλάνου με ευφάνταστο τρόπο, η κατασκευή του βοηθητικού υλικού για την πραγματοποίηση των μαθημάτων και η εύρεση των πόρων για τον εξοπλισμό των εργαστηρίων δυσκόλεψαν αρκετά τον γράφοντα, αλλά ευτυχώς ξεπεράστηκαν.

Η πρώτη ομάδα μαθητών στην οποία εφαρμόστηκε το διδακτικό πλάνο ήταν αυτή των έμπειρων. Γνωρίζοντας το επίπεδο τους στον προγραμματισμό, ο διδάσκων προσδοκούσε μια σειρά μαθημάτων χωρίς ιδιαίτερα εκπαιδευτικά προβλήματα, αλλά με μια αυστηρή, κριτική ματιά στην διδασκαλία και με τον κίνδυνο να χαθεί το ενδιαφέρον τους σε αυτήν, λόγω της αναντιστοιχίας του επιπέδου του μαθήματος με το δικό τους. Ακόμη ένα στοίχημα που έπρεπε να κερδηθεί με την συγκεκριμένη ομάδα ήταν η δημιουργία τέτοιων συνθηκών, ώστε οι παρατηρήσεις τους και τα σχόλια τους να μπορούν να χρησιμοποιηθούν για μια πιο εξειδικευμένη αξιολόγηση της διδασκαλίας.

Στην πράξη, οι έμπειροι προγραμματιστές υπήρξαν όντως αυστηροί κριτές. Οι προσπάθειες του διδάσκοντα να μην χαθεί το ενδιαφέρον αυτών των μαθητών στην

πορεία της διδασκαλίας υπήρξαν επιτυχημένες, όμως ήταν φανερό πως τους ταίριαζε η χρήση μιας σύνθετης ρομποτικής εφαρμογής και η συγγραφή προγραμμάτων σε ένα προγραμματιστικό εργαλείο χαμηλού επιπέδου. Η άσχημη εμπειρία τους σε κλασικές διδακτικές προσεγγίσεις τους έκανε να απολαύσουν το όλο εγχείρημα και να δείξουν μια προτίμηση σε τέτοιες εναλλακτικές μεθόδους, αλλά και να συμπεριφερθούν πολλές φορές αμήχανα λόγω της συνήθειας τους σε πιο παραδοσιακές μεθόδους.

Η διδασκαλία με το τμήμα των στοιχειωδώς καταρτισμένων στον προγραμματισμό μαθητών, έκρυβε εξαρχής αρκετούς κινδύνους. Οι συμμετέχοντες διέθεταν κάποιες γνώσεις πάνω στον προγραμματισμό, αλλά οι περισσότερες από αυτές ήταν ημιτελείς και μπερδεμένες, δημιουργώντας σύγχυση πρωταρχικά στους ίδιους και ακολούθως στον διδάσκοντα. Καθήκον του τελευταίου με την συγκεκριμένη ομάδα ήταν να αποσαφηνίσει όποιες παρερμηνείες είχαν οι μαθητές γύρω από βασικές προγραμματιστικές έννοιες και να πραγματοποιήσει μια διδασκαλία που θα αναδομούσε, ορθά πλέον, μια προγραμματιστική λογική.

Η διδασκαλία με αυτή την ομάδα των μαθητών εξελίχθηκε ακριβώς όπως αναμενόταν. Σε πολλές περιπτώσεις οι γνώσεις των μαθητών διευκόλυναν τον διδάσκοντα, σε άλλες όμως των ενέπλεξαν σε μια επίπονη επεξηγηματική διαδικασία που καθυστερούσε την ροή του μαθήματος. Συνήγορος του διδάσκοντα σε αυτές τις περιπτώσεις υπήρξε το Scribbler, με τη βοήθεια του οποίου συνδυάστηκε η επίλυση των γνωστικών ζητημάτων, προς τέρψιν των μαθητών που έδειχναν να απολαμβάνουν τις λειτουργίες του.

Ως μια γενικότερη τοποθέτηση θα μπορούσαμε να πούμε η εξέλιξη της τεχνολογίας και η χρήση της για καθημερινές δραστηριότητες, έχει δημιουργήσει μια νέα ομάδα ανθρώπων που διαθέτουν σκόρπιες γνώσεις σε θέματα προγραμματισμού και σε οποιαδήποτε απόπειρά τους για εκμάθησή του, θα διευκολυνθούν από αυτές αλλά και θα δυσκολευτούν

εξίσου. Με τη βοήθεια μιας ρομποτικής εφαρμογής, η διδασκαλία μπορεί να αποδειχθεί ευκολότερη και αποτελεσματικότερη για διδάσκοντα και μαθητές, σε σχέση με μια αυστηρή διδακτική προσέγγιση εργαστηρίων και διαλέξεων.

Η μαθητική ομάδα των αρχαρίων δεν θα μπορούσε παρά να αποτελεί μια πρόκληση για τον διδάσκοντα. Ουσιαστικές προγραμματιστικές γνώσεις για τα μέλη αυτής της ομάδας δεν υπήρχαν και ο διδάσκων καλούταν να οικοδομήσει εξ αρχής μια ορθή προγραμματιστική εντολή, γεγονός που από τη μία τον γεμίζει ευθύνη και από την άλλη του δίνει του δυνατότητα να δράσει ελεύθερα και ανεμπόδιστα. Ο τρόπος διδασκαλίας, η γραφική διεπαφή και το ρομπότ «φωτογράφιζαν» το επίπεδο των αρχαρίων, συνεπώς ήταν στο χέρι του διδάσκοντα να διευκολύνει τους μαθητές και να τους μεταλαμπαδεύσει την απαιτούμενη γνώση.

Η μεγαλύτερη δυσκολία που αντιμετώπισε το γκρουπ των αρχαρίων δεν ήταν αναμενόμενη: οι μαθητές παρουσιάστηκαν αρκετά αγχωμένοι και με την αυτοπεποίθηση τους σε τέτοιο επίπεδο που δεν τους επέτρεπε να απολαύσουν τις δραστηριότητες της διδασκαλίας. Ένα μέρος αυτής της συμπεριφοράς αποβλήθηκε κατά την διάρκεια της διδασκαλίας αλλά σε αρκετές περιπτώσεις αποτέλεσε ανασταλτικό παράγοντα. Με λίγη περισσότερη πίστη στις δυνατότητές τους, οι μαθητές θα άφηναν κυριολεκτικά άναυδο τον διδάσκοντα. Παρότι η διδασκαλία ήταν μικρή σε διάρκεια και ύλη, οι μαθητές ανταποκρίθηκαν εντυπωσιακά, τηρουμένων των αναλογιών. Για τους ίδιους τους μαθητές, αυτή η εμπειρία αποτέλεσε κίνητρο να συνεχίσουν την ενασχόλησή τους με τον

προγραμματισμό, πράγμα που συνήθως φαντάζει ουτοπικό σε αρχάριους που διδάσκονται με την παραδοσιακή μέθοδο εργαστηρίων-διαλέξεων.

Βλέποντας λοιπόν πως η διδασκαλία λειτούργησε σχεδόν άψογα στους αρχάριους μαθητές, δεν υπάρχει λόγος να προβληθεί η παραμικρή ένσταση για την χρήση του Scribbler σε παρόμοιες διδασκαλίες. Ένα ολοκληρωμένο μάθημα με αυτό, καθώς και μια εισαγωγή στην rython based γλώσσα που χρησιμοποιεί εναλλακτικά, θα μπορούσε να εφαρμοστεί με μεγάλη επιτυχία σε οποιοδήποτε ελληνικό εκπαιδευτικό ίδρυμα, όπως άλλωστε γίνεται καιρό τώρα σε αντίστοιχα ξένα.

Στο κλείσιμο αυτής της έρευνας, μοιραία τίθεται το ερώτημα: Αξίζει να προτιμηθεί μια μέθοδος διδασκαλίας προγραμματισμού με τη χρήση ρομποτικής, έναντι μια παραδοσιακής μεθόδου με εργαστήρια και διαλέξεις;

Η απάντηση είναι σαφώς ότι αξίζει, αλλά με αρκετές παραμέτρους. Επιλέγοντας μια τέτοια μέθοδο διδασκαλίας πρέπει να διασφαλίζεται αρχικά ότι μπορούν να βρεθούν οι αντίστοιχοι εργαστηριακοί πόροι (ευρύχωρες αίθουσες, ρομποτικές εφαρμογές, υπολογιστές κ.ά.) ώστε η διδακτική διαδικασία να εφαρμοστεί ολοκληρωμένα και ανεμπόδιστα. Εφόσον έχουν εξασφαλιστεί τα παραπάνω, ο διδάσκων πρέπει να λάβει σοβαρά υπόψη πως το «χτίσιμο» ενός τέτοιου μαθήματος απαιτεί πολύ περισσότερο χρόνο απ ότι ένα παραδοσιακό μάθημα, αλλά και δημιουργική διάθεση και φαντασία ώστε οι δυνατότητες της ρομποτικής εφαρμογής να εκμεταλλευτούν στο έπακρο, κάνοντας τους μαθητές ευτυχισμένους που συμμετέχουν σε μια τέτοια διαδικασία. Ιδιαίτερης προσοχής χρίζει και η επιλογή της ρομποτικής εφαρμογής, με την βοήθεια της οποίας θα διεξαχθεί το μάθημα. Ο διδάσκων έχει τη δυνατότητα να επιλέξει ανάμεσα από μια πληθώρα ρομποτικών εφαρμογών, με γνώμονα ποια από αυτές μπορεί να ανταπεξέλθει πλήρως στις απαιτήσεις τόσο των μαθητών όσο και του ίδιου του μαθήματος. Με σεβασμό στις παραπάνω παραμέτρους και κυρίως με όρεξη και δημιουργικότητα, μια εναλλακτική διδασκαλία προγραμματισμού όχι μόνο προτείνεται ανεπιφύλακτα, αλλά είναι και κατά πολύ αυξημένες οι πιθανότητες επιτυχίας της.

ΠΑΡΑΡΤΗΜΑ

Φύλλα αξιολόγησης

Φύλλο αξιολόγησης εμπειρίας

Η συμπλήρωση του εμπιστευτικού αυτού ερωτηματολογίου είναι πολύ σημαντική. Συγκεντρώνει χρήσιμες πληροφορίες που χρησιμοποιούνται αποκλειστικά από τους διδάσκοντες για τον σχεδιασμό και την ανάπτυξη μελλοντικών μαθημάτων. Ιδιαίτερη αξία έχουν τα σχόλια που μπορείτε να συμπεριλάβετε στο τέλος του ερωτηματολογίου.

Απαντήστε στις παρακάτω ερωτήσεις:

1. Είναι οι μαθησιακοί στόχοι των προτεινόμενων δραστηριοτήτων σαφείς;
 - α. Ναι
 - β. Όχι
2. Οι δραστηριότητες είναι καλά δομημένες;
 - α. Ναι
 - β. Όχι
3. Είναι η περιγραφή τους σαφής και κατανοητή;
 - α. Ναι
 - β. Όχι
4. Ποιές παιδαγωγικές στρατηγικές πιστεύετε ότι υποστηρίχθηκαν κατά τη διάρκεια του μαθήματος;
 - α. Δημιουργία των καταστάσεων του προβλήματος
 - β. Ανάπτυξη δεξιοτήτων σχετικά με την ανάλυση του προβλήματος
 - γ. Ανάπτυξη δεξιοτήτων σχετικά με την διατύπωση υποθέσεων
 - δ. Διεύρυνση της γνώσης των μαθητών στα πλαίσια ενός δοθέντος ζητήματος
 - ε. Ανάπτυξη δεξιοτήτων σχετικά με την επαλήθευση των υποθέσεων
 - στ. Όλες οι παραπάνω
 - ζ. Άλλες(αναφέρετε ποιες).....
5. Υποθέτουμε ότι η διδασκαλία των εννοιών που διδαχθήκατε οδηγεί στην απόκτηση νέων δεξιοτήτων. Διαλέξτε από την παρακάτω λίστα τις δεξιότητες που αποκτήθηκαν ή υποστηρίχθηκαν από τη διδασκαλία:
 - α. Ανάπτυξη και ερμηνεία μέσω της γραφικής αναπαράστασης, των δεδομένων και της λύσης ενός προβλήματος
 - β. Κατάλληλη χρήση των αισθητήρων και των γραφικών διεπαφών
 - γ. Δημιουργία προτύπων ή χρησιμοποίηση προσομοιώσεων
 - δ. Όλες οι παραπάνω
 - ε. Άλλες(αναφέρετε ποιες).....
6. Η χρησιμοποίηση του ρομπότ έκανε την αποσφαλμάτωση των προγραμμάτων:

- α. Πολύ ευκολότερη
- β. Ευκολότερη
- γ. Δυσκολότερη
- δ. Πολύ δυσκολότερη
- ε. Δεν την επηρέασε

7. Η χρήση μιας εκπαιδευτικής ρομποτικής εφαρμογής θα μπορούσε να ήταν χρήσιμη σε διδασκαλία:

- α. Θετικών επιστημών
- β. Θεωρητικών επιστημών
- γ. Ελεύθερο, δημιουργικό παιχνίδι
- δ. Άλλο.....

8. Οι γενικοί στόχοι διδασκαλίας των δραστηριοτήτων στις ενότητες που διδαχθήκατε ήταν:

- α. Η εισαγωγή μιας νέας έννοιας, θέματος, προβλήματος
- β. Η παρουσίαση μιας νέας προσέγγισης για τη διδασκαλία και την εκμάθηση
- γ. Η επεξήγηση γνωστών επιστημονικών ιδεών
- δ. "Άλλο(αναφέρετε).....

9. Το πλαίσιο ιστορίας μέσα στο οποίο το ρομπότ εκτελεί τις δραστηριότητές του αποδείχθηκε:

- α. Ευφάνταστο
- β. Χρήσιμο
- γ. Αδιάφορο
- δ. Αχρειαστο
- ε. Δεν έχω καμία άποψη

10. Πως κρίνεται τον συνδυασμό των διδακτικών μεθόδων που χρησιμοποιήθηκαν;

- α. Θετικά
- β. Μέτρια
- γ. Αρνητικά

11. Προσδιορίστε ποιοι μαθησιακοί στόχοι υποστηρίχθηκαν κατά τη διδασκαλία (μπορείτε να επιλέξετε παραπάνω από έναν):

- α. Διατύπωση και αξιολόγηση επιχειρημάτων από δεδομένα που έχουν συλλεχθεί
- β. Διατύπωση υποθέσεων και έλεγχος της ορθότητας τους
- γ. Πρόσκτηση στρατηγικών επίλυσης προβλήματος
- δ. Ικανότητα συνεργατικής επίλυσης προβλήματος

12. Πώς θα χαρακτηρίζατε τη συνεργασία σας με το άλλο μέλος της ομάδας;

- α. Θετικά
- β. Μέτρια
- γ. Αρνητικά

13. Προσθέστε οποιοδήποτε σχόλιο επιθυμείτε το οποίο θα ήταν χρήσιμο για τη βελτίωση του μαθήματος

.....

.....

.....

Φύλλο γενικής αξιολόγησης

Η συμπλήρωση του εμπιστευτικού αυτού ερωτηματολογίου είναι πολύ σημαντική. Συγκεντρώνει χρήσιμες πληροφορίες που χρησιμοποιούνται αποκλειστικά από τους διδάσκοντες για τον σχεδιασμό και την ανάπτυξη μελλοντικών μαθημάτων. Ιδιαίτερη αξία έχουν τα σχόλια που μπορείτε να συμπεριλάβετε στο τέλος του ερωτηματολογίου.

Καθόλου	Λίγο	Μέτρια	Πολύ	Πάρα πολύ
1	2	3	4	5
Απαράδεκτη/ Απαράδεκτο/ Απαράδεκτα	Μη ικανοποιητική/ Μη ικανοποιητικό/ Μη ικανοποιητικά	Μέτρια/ Μέτριο	Ικανοποιητική/ ικανοποιητικό/ Ικανοποιητικά	Πολύ καλή/ Πολύ καλό

Αξιολογήστε τις ακόλουθες προτάσεις σημειώνοντας x στο αντίστοιχο τετραγωνάκι

Το μάθημα:	1	2	3	4	5
1. Οι συνολικός στόχος του μαθήματος ήταν σαφής;					
2. Η ύλη που καλύφθηκε ανταποκρινόταν στους στόχους του μαθήματος;					
3. Τα μαθήματα ήταν καλά προετοιμασμένα και οργανωμένα;					
4. Το υλικό που χρησιμοποιήθηκε βοήθησε στην κατανόηση του μαθήματος;					
5. Πώς κρίνεται το επίπεδο δυσκολίας του μαθήματος;					
6. Χρησιμοποιήθηκαν άλλα σύγχρονα μέσα διδασκαλίας (βίντεο, παρουσιάσεις ,κα.);					
7. Τηρήθηκε το προβλεπόμενο ωράριο που έγινε γνωστό πριν την έναρξη του μαθήματος;					

Στις περιπτώσεις που υπήρχαν εργασίες:	1	2	3	4	5
8. Οι εκφωνήσεις ήταν κατανοητές;					
9. Η χρονική προθεσμία για την υποβολή των λύσεων ήταν λογική;					
10. Υπήρχε καθοδήγηση από τον διδάσκοντα;					
11. Τα σχόλια του διδάσκοντα ήταν εποικοδομητικά και αναλυτικά;					
12. Δόθηκε η δυνατότητα βελτίωσης της εργασίας;					
13. Η συγκεκριμένη εργασία σας βοήθησε να κατανοήσετε το συγκεκριμένο θέμα;					

Ο διδάσκων:	1	2	3	4	5
14. Οργάνωσε καλά την παρουσίαση της ύλης στα μαθήματα;					
15. Πέτυχε να διεγείρει το ενδιαφέρον σας για το αντικείμενο του μαθήματος;					

16. Ανάλυσε και παρουσίασε τις έννοιες με τρόπο απλό και ενδιαφέροντα χρησιμοποιώντας παραδείγματα:					
17. Δημιούργησε μια ατμόσφαιρα σεβασμού και εμπιστοσύνης μέσα στην τάξη:					
18. Συνδύασε την ύλη του μαθήματος με πραγματικά παραδείγματα/εφαρμογές;					
19. Σας έδωσε να καταλάβετε πως οι διάφορες ενότητες του μαθήματος σχετίζονται μεταξύ τους;					
20. Σας ενθάρρυνε να διατυπώσετε ερωτήσεις και απορίες ώστε να αναπτύξετε την κρίση σας;					
21. Ήταν συνεπής στις υποχρεώσεις του;					
22. Ήταν γενικά προσιτός κατά τη διάρκεια του μαθήματος;					
23. Οι εξηγήσεις του ήταν ξεκάθαρες;					

Περιβάλλον εργασίας:	1	2	3	4	5
24. Ήταν επαρκής ο εξοπλισμός του εργαστηρίου;					
25. Αξιολογήστε την ποιότητα του χώρου που έγινε η διδασκαλία του μαθήματος					
26. Τα εργαστηριακά μέσα διευκόλυναν τις δραστηριότητες;					
27. Πόσο ικανοποιημένοι μείνατε από το Scribbler;					
28. Πως ανταποκρίθηκε στις εργασίες που του ζητήσατε;					
29. Πόσο ικανοποιημένοι μείνατε από τη γραφική διεπαφή του Scribbler;					
30. Πως ανταποκρίθηκαν οι υπολογιστές του εργαστηρίου;					
31. Τα εργαστηριακά μέσα συνδυάστηκαν					

Ακαδημαϊκή αξία/Συνολική εκτίμηση:	1	2	3	4	5
32.Έχω μάθει και καταλάβει το αντικείμενο του μαθήματος					
33.Το μάθημα είναι υψηλών απαιτήσεων διανοητικά και διεγείρει το ενδιαφέρον:					
34.Το μάθημα θα σας φανεί σε κάποια στιγμή της ζωής σας;					
35.Ο καθηγητής ήταν ικανός και αποτελεσματικός για τη διδασκαλία του συγκεκριμένου αντικειμένου;					
36.Ο καθηγητής συνεισέφερε θετικά στην προαγωγή της γνώσης μου για το αντικείμενο του μαθήματος;					

Φύλλο αξιολόγησης για έμπειρους στον προγραμματισμό

Η συμπλήρωση του εμπιστευτικού αυτού ερωτηματολογίου είναι πολύ σημαντική. Συγκεντρώνει χρήσιμες πληροφορίες που χρησιμοποιούνται αποκλειστικά από τους διδάσκοντες για τον σχεδιασμό και την ανάπτυξη μελλοντικών μαθημάτων. Ιδιαίτερη αξία έχουν τα σχόλια που μπορείτε να συμπεριλάβετε στο τέλος του ερωτηματολογίου.

Καθόλου	Λίγο	Μέτρια	Πολύ	Πάρα πολύ
1	2	3	4	5
Απαράδεκτη/ Απαράδεκτο/ Απαράδεκτα	Μη ικανοποιητική/ Μη ικανοποιητικό/ Μη ικανοποιητικά	Μέτρια/ Μέτριο	Ικανοποιητική/ ικανοποιητικό/ Ικανοποιητικά	Πολύ καλή/ Πολύ καλό

Αξιολογήστε τις ακόλουθες προτάσεις σημειώνοντας x στο αντίστοιχο τετραγωνάκι, κυκλώνοντας την απάντηση που σας ταιριάζει ή συμπληρώνοντας τα απαραίτητα σχόλια

Εμπειρία στον προγραμματισμό

1. Ποια προγραμματιστικά εργαλεία είχατε χρησιμοποιήσει πριν την παρούσα διδασκαλία (έως 3);

.....

2. Πόσο έμπειρο προγραμματιστή θα χαρακτηρίζατε τον εαυτό σας;

1	2	3	4	5

3. Πόσο ενδιαφέρον επιστημονικό αντικείμενο θα χαρακτηρίζατε τον προγραμματισμό;

1	2	3	4	5

4. Πόσο δυσκολευτήκατε κατά την πρώτη επαφή σας με μια κλασσική γλώσσα προγραμματισμού;

1	2	3	4	5

5. Ποια ήταν η αιτία της δυσκολίας που αντιμετωπίσατε;
- A. Αδυναμία κατανόησης της προγραμματιστικής λογικής.
 B. Αδυναμία υλοποίησης της λύσης ενός προβλήματος παρά την κατανόηση της λογικής της.
 Γ. Αδυναμία αφομοίωσης της γνώσης λόγω υπερβολικής ύλης σε μικρό χρονικό διάστημα.
 Δ. Άλλη

6. Πόσο ενδιαφέρουσα θα χαρακτηρίζατε την παραδοσιακή μέθοδο διδασκαλίας προγραμματισμού, με διαλέξεις και εργαστήρια;

1	2	3	4	5

7. Ποιο πιστεύετε ότι είναι το μεγαλύτερο μειονέκτημα της παραδοσιακής μεθόδου διδασκαλίας;
- A. Έλλειψη σύνδεσης των εννοιών του προγραμματισμού με την πραγματική ζωή.
 B. Έλλειψη κατανοητής οπτικοποίησης των αποτελεσμάτων ενός προγράμματος.
 Γ. Αυστηρή προσέγγιση της δομής μιας γλώσσας.
 Δ. Άλλη

8. Έχετε διδαχθεί προγραμματισμό :
- A. Μόνο με τη μέθοδο εργαστηρίων-διαλέξεων.
 B. Με λογισμικό απεικόνισης των αποτελεσμάτων σε γραφικό περιβάλλον.
 Γ. Με κάποια ρομποτική εφαρμογή.
 Δ. Με τις μεθόδους(συνδυασμός των παραπάνω)

 E. Άλλο.....

9. Βαθμολογήστε στην κλίμακα την εμπειρία σας σε εναλλακτικές μεθόδους διδασκαλίας προγραμματισμού.

1	2	3	4	5

10. Βαθμολογήστε στην κλίμακα την εμπειρία σας σε ρομποτικές εφαρμογές.

1	2	3	4	5

Γραφική διεπαφή

1. Βαθμολογήστε στην κλίμακα την ευχρηστία της γραφικής διεπαφής του Scribbler.

1	2	3	4	5

2. Κατά πόσο πιστεύετε ότι απεικονίζει κατανοητά τις βασικές προγραμματιστικές έννοιες που χρησιμοποίησατε;

A. Δομή προγράμματος

1	2	3	4	5

B. Δομή εντολής ελέγχου

1	2	3	4	5

Γ. Επαναληπτική δομή

1	2	3	4	5

3. Περιγράψτε σύντομα την κατά τη γνώμη σας μεγαλύτερη αδυναμία της γραφικής διεπαφής του Scribbler.

.....

.....

.....

4. Γνωρίζοντας ότι το Scribbler robot μπορεί να προγραμματιστεί και σ ένα rython-based περιβάλλον και μορφή προγραμμάτων σε κώδικα, ποιο από τα δύο περιβάλλοντα θα προτιμούσατε να χρησιμοποιήσετε;

A. Το rython based περιβάλλον

B. Τη γραφική διεπαφή

5. Κατά πόσο πιστεύετε ότι οι εντολές της γραφικής διεπαφής του Scribbler καλύπτουν όλες τις περιπτώσεις αντίστοιχων εντολών σε κλασσικά προγραμματιστικά εργαλεία που έχετε χρησιμοποιήσει;

1	2	3	4	5

Κρίνοντας την διδασκαλία

1. Δεδομένης της εμπειρίας σας και του εισαγωγικού χαρακτήρα των μαθημάτων, πόσο πλήρη θα τα χαρακτηρίζατε;

1	2	3	4	5

2. Σε τι βαθμό πιστεύετε ότι μπορεί να καλυφθεί η ύλη ενός ολοκληρωμένου μαθήματος προγραμματισμού αποκλειστικά με το Scribbler;

1	2	3	4	5

--	--	--	--	--

3. Πόσο ευδιάκριτη θα χαρακτηρίζατε την σύνδεση των επιμέρους δραστηριοτήτων των μαθημάτων με τις αντίστοιχες προγραμματιστικές έννοιες που διδάχθηκαν;

1	2	3	4	5

4. Σε ποιο βαθμό πιστεύετε ότι οι διδάσκοντες εκμεταλλεύθηκαν τις δυνατότητες που παρέχει στον εκπαιδευτικό μια τέτοια μορφή διδασκαλίας;

1	2	3	4	5

5. Πόσο ευφάνταστες θα χαρακτηρίζατε αυτές τις δραστηριότητες;

1	2	3	4	5

6. Θα προτιμούσατε το βοηθητικό υλικό των μαθημάτων να παρεχόταν σε μορφή σημειώσεων με επεξηγήσεις αντί των αρχείων βίντεο;

- A. Ναι
B. Όχι

7. Θα προτιμούσατε η μορφή των δραστηριοτήτων να μην ήταν τόσο αυστηρή αλλά να έδινε στον μαθητή περιθώρια ελευθερίας;

- A. Ναι
B. Όχι

Στη θέση του διδάσκοντα

1. Αν σας είχε ζητηθεί να διδάξετε ένα ολοκληρωμένου μάθημα προγραμματισμού, ποια μέθοδο διδασκαλίας θα επιλέγατε και γιατί;

A. Θα επέλεγα την παραδοσιακή μέθοδο διαλέξεων-εργαστηρίων γιατί

.....
.....
.....

B. Θα επέλεγα την μέθοδο διδασκαλίας με χρήση εκπαιδευτικής ρομποτικής γιατί

.....
.....
.....

2. Αν βρισκόσασταν στη θέση του διδάσκοντα τι θα προσθέτατε στην διδασκαλία;

A. Εγχειρίδιο χρήσης της γραφικής διεπαφής του Scribbler.

- B. Εκπόνηση συνδυαστικού project στο σπίτι.
 Γ. Βασικά στοιχεία της rython based γλώσσας του Scribbler.
 Δ. Άλλο
3. Αν βρισκόσασταν στη θέση του διδάσκοντα τι θα αφαιρούσατε από την διδασκαλία;
 Α. Δραστηριότητες που αφορούν αποκλειστικά τη γραφική διεπαφή του Scribbler.
 Β. Δραστηριότητες με επίλυση στο χαρτί.
 Γ. Εισαγωγή (multimedia και συζήτηση) στην ρομποτική.
 Δ. Άλλο
4. Αν είχατε επιλέξει την παραδοσιακή μέθοδο διδασκαλίας με εργαστήρια και διαλέξεις, ποιο πιστεύετε ότι θα ήταν το μεγαλύτερο πρόβλημα που θα αντιμετωπίζατε;
 Α. Αδυναμία μετάδοσης γνώσης στους μαθητές.
 Β. Δημιουργία κατανοητών παραδειγμάτων προς μελέτη.
 Γ. Προσέλευση του ενδιαφέροντος των μαθητών.
 Δ. Άλλο
5. Αν είχατε επιλέξει μια εναλλακτική μέθοδο διδασκαλίας με τη βοήθεια της εκπαιδευτικής ρομποτικής, ποιο πιστεύετε ότι θα ήταν το μεγαλύτερο πρόβλημα που θα αντιμετωπίζατε;
 Α. Υπερβολικός απαιτούμενος χρόνος για την οργάνωση ενός μαθήματος.
 Β. Δυσκολία σύνδεσης προγραμματιστικών εννοιών με λειτουργίες του ρομπότ.
 Γ. Αποπροσανατολισμός των μαθητών από το ίδιο το ρομπότ.
 Δ. Άλλο
1. Αν είχατε επιλέξει μια εναλλακτική μέθοδο διδασκαλίας με τη βοήθεια της εκπαιδευτικής ρομποτικής, ποιο πιστεύετε ότι θα ήταν το μεγαλύτερο πρόβλημα που θα αντιμετωπίζατε;
 Α. Υπερβολικός απαιτούμενος χρόνος για την οργάνωση ενός μαθήματος.
 Β. Δυσκολία σύνδεσης προγραμματιστικών εννοιών με λειτουργίες του ρομπότ.
 Γ. Αποπροσανατολισμός των μαθητών από το ίδιο το ρομπότ.
 Δ. Άλλο

Σύγκριση διδακτικών μεθόδων

Επιλέξτε 1 για την παραδοσιακή μέθοδο διαλέξεων-εργαστηρίων και 2 για την διδασκαλία του προγραμματισμού με τη βοήθεια ρομποτικής.

Με ποια μέθοδο διδασκαλίας πιστεύετε ότι:	1	2
1. Καλύπτεται μεγαλύτερη ύλη σε μικρότερο χρονικό διάστημα.		
2. Ο μαθητής κατανοεί ευκολότερα δυσνόητες προγραμματιστικές έννοιες.		
3. Ο μαθητής θέτει στέρεες προγραμματιστικές βάσεις.		

4. Το έργο του διδάσκοντα είναι ευκολότερο.		
5. Ο μαθητής οφείλει να αφιερώσει στο μάθημα λιγότερες ώρες από τον ελεύθερό του χρόνο.		
6. Εγείρεται ευκολότερα το ενδιαφέρον του μαθητή.		
7. Η μετάβαση σε σύγχρονα προγραμματιστικά εργαλεία είναι ευκολότερη.		
8. Ο μαθητής δομεί τρόπο σκέψης που θα του χρειαστεί σε παράπλευρα επιστημονικά πεδία.		
9. Ο μαθητής κατανοεί και χρησιμοποιεί ευκολότερα στρατηγικές επίλυσης προβλημάτων.		
10. Οι μαθητές συνεργάζονται αρτιότερα για την επίλυση ενός προβλήματος.		

11. Εν κατακλείδι, πιστεύετε ότι πρέπει να χρησιμοποιούνται εναλλακτικές μέθοδοι διδασκαλίας στα μαθήματα προγραμματισμού;

- A. Ναι
- B. Όχι

12. Πόσο αποτελεσματική πιστεύετε ότι θα ήταν μια τέτοια μέθοδος διδασκαλίας προγραμματισμού σε ένα ελληνικό εκπαιδευτικό ίδρυμα;

1	2	3	4	5

Βιβλιογραφία

- [1]. Μπαγιάτη Αικ., Δρομπούρας Χρ., Κοντογιάννης Δημ. (2008), "Συγκριτική μελέτη της εκπαιδευτικής ρομποτικής στην διδασκαλία του προγραμματισμού", Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχ. Η/Υ, Τηλ. Και Δικτύων.
- [2]. Δημητρίου Αν, Χατζηκρανιώτης Εύρ. (1998), " Η εκπαιδευτική ρομποτική ως εργαλείο ανάπτυξης δεξιοτήτων για τη λύση προβλήματος: Εφαρμογή με το περιβάλλον LegoDacta ".
- [3]. Wagner, P. S. (1998), "Robotics and children: Science achievement and problem solving", *Journal of Computing in Childhood Education*, 9(2), 149-192.
- [4]. Δαγδιλέλης, Β. Φαχαντίδης, Ν. & Γροπέτης, Γ. (1999), "Η επιμόρφωση των εκπαιδευτικών στην πληροφορική: Η περίπτωση της εκπαιδευτικής ρομποτικής". Πρακτικά συνεδρίου: Νέες παράμετροι στην εκπαίδευση: Εκπαίδευση από απόσταση και διά βίου εκπαίδευση. Πανεπιστήμιο Αιγαίου - Σχολή Ελληνικών και Μεσογειακών Σπουδών - Τμήμα Δημοτικής Εκπαίδευσης
- [5]. Ανθή Καρατράντου, Νικόλαος Τάχος, Δημήτρης Αλιμίσης, "Εισαγωγή σε βασικές αρχές και δομές προγραμματισμού με τις ρομποτικές κατασκευές Lego Mindstorms".
- [6]. <http://www.parallax.com/>
- [7]. Winslow, L.E. (1996), "Programming pedagogy – A psychological overview", *SIGCSE Bulletin*, 28, 17–22.
- [8]. Guindon, R. (1990), " Knowledge exploited by experts during software systems design. *International Journal of Man-Machine Studies*", 33, 182–279.
- [9]. Soloway, E., & Spohrer, J.C. (Eds.) (1989) "Studying the novice programmer", Hillsdale, NJ: Lawrence Erlbaum.
- [10]. Davies, S.P. (1993), " Models and theories of programming strategy", *International Journal of Man-Machine Studies*, 39, 237–267.
- [11]. Essi Lahtinen, Kirsti AlaMutka, Hannu-Matti Jarvinen, "A Study of the Difficulties of Novice Programmers".
- [12]. Ρίτσα Γόγουλου, "Εναλλακτικές διδακτικές προσεγγίσεις σε εισαγωγικά μαθήματα προγραμματισμού".
- [13]. Linn, M. & Clancy, M. (1992), "The case for case studies of programming problems", *Communications of the ACM*, 35(3), 121-132.
- [14]. Alexander C. & et al. (1977), "A Pattern Language", Oxford University Press.
- [15]. Brusilovsky, P. (2001), "WebEx: Learning from Examples in a Programming Course, In Fowler, W. & Hasebrook, J. (eds.)", *Proceedings of WebNet'2001, World Conference of the WWW and Internet*, 124-129, Orlando, Canada.
- [16]. Williams, L. & Upchurch, R.L. (2001), "In Support of Student Pair-Programming", *Proceedings of the ACM SIGCSE '01 Conference*, 327-331, Charlotte, USA.

- [17]. Robins, Anthony, Rountree, Janet and Rountree, Nathan(2003), "Learning and Teaching Programming: A Review and Discussion" , Computer Science Education,13:2,137 — 172
- [18]. Du Boulay, B. (1989), "Some difficulties of learning to program, In E. Soloway & J. C. Spohrer (Eds)", Studying the Novice Programmer, 283-299, Hillsdale, NJ, Lawrence Erlbaum Associates.
- [19]. Τζιμογιάννης Α. & Κόμης Β. (2000), "Η έννοια της μεταβλητής στον Προγραμματισμό: δυσκολίες και παρανοήσεις μαθητών του Ενιαίου Λυκείου".
- [20]. Τζιμογιάννης, Α. & Γεωργίου, Β. (1999), "Οι δυσκολίες μαθητών δευτεροβάθμιας εκπαίδευσης στην εφαρμογή της δομής ελέγχου για την ανάπτυξη αλγορίθμων. Μια μελέτη περίπτωσης".
- [21]. Ρίτσα Γόγουλου, "Μαθησιακές Δυσκολίες και Παρανοήσεις στις Βασικές Έννοιες και Δομές του Προγραμματισμού".
- [22]. Βασίλης Ι. Κόμης,(2005), " Εισαγωγή στη Διδακτική της Πληροφορικής" , 261-268.
- [23]. Ebrahimi, A. (1994), "Novice programmer errors: language constructs and plan composition", Int. J. Human-Computer Studies, 41, 457-480.
- [24]. Kurtz, T. (1981), " BASIC. In Wexelblat, R., Ed. History of Programming Languages", Academic Press, New York, 515–537.
- [25]. Holt, R., Wortman, D., Barnard, D., And Cordy, J. R. (1977), " SP/k: A system for teaching computer programming", Commun. ACM 20, 5, 301–309.
- [26]. Holt, R. And Cordy, J. (1988), "The turing programming language", Commun. ACM 31, 12, 1410–1423.
- [27]. Motil J. and Epstein D. (1998)," JJ: A language designed for beginners (less is more)"
- [28]. Caitlin Kelleher and Randy Pausch," Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers", Carnegie Mellon University, ACM Computing Surveys, Vol. 37, No. 2, June 2005, pp. 83–137.
- [29]. Kahn, K. (1996), " Drawings on napkins, video-game animation, and other ways to program computers", Commun. ACM 43, 3. 104–106.
- [30]. Repenning, A. (1993), " Agentsheets: A tool for building domain-oriented visual programming", In the Conference on Human Factors in Computing Systems, 142–143.
- [31]. Bell, B. And Lewis, C. (1993), " ChemTrains: A language for creating behaving pictures", In IEEE Symposium on Visual Languages, 188–195.
- [32]. Alimisis D. and partners (2007), "Robotics in school education, Review of research literature, A synthesis", Terecop, Aspete, Greece.