ΕΝΑΣ ΔΙΔΑΚΤΙΚΟΣ ΜΙΚΡΟΚΟΣΜΟΣ ΓΙΑ ΜΙΑ ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ: ΤΟ ΡΟΜΠΟΤ KAREL

Μ. Σατρατζέμη, Επίκ. Καθηγήτρια , Τμ. Εφαρμοσμένης Πληροφορικής, Πανεπ. Μακεδονίας.
Β. Δαγδιλέλης, Λέκτορας, Παιδαγωγική Σχολή Φλώρινας, ΑΠΘ.
Κ. Μαργαρίτης, Καθηγητής, Τμ. Εφαρμοσμένης Πληροφορικής, Πανεπ. Μακεδονίας.
Στ. Μιχαήλ, πτυχιούχος τμ. Εφαρμοσμένης Πληροφορικής, Πανεπ. Μακεδονίας .
Στ. Παπανικολάου, πτυχιούχος τμ. Εφαρμοσμένης Πληροφορικής, Πανεπ. Μακεδονίας, Εγνατία 156, Τ.Θ. 1591, Θεσσαλονίκη, 54006.
{maya, dagdil, kmarg} @uom.gr

Λέξεις Κλειδιά: μικρόκοσμοι (microworlds), μικρές-γλώσσες προγραμματισμού (mini-languages), προγραμματισμός (programming).
Θέμα: Πληροφορική στην Εκπαίδευση: Σχεδιασμός και ανάπτυξη εκπαιδευτικού λογισμικού.
Επίπεδο εκπαίδευσης: Γυμνάσιο, Λύκειο, Μεταδευτεροβάθμια.
Κατηγορία: Εμπειρική – Πειραματική.

Περίληψη

Οι μικρο-γλώσσες προγραμματισμού αν και είναι ένας απλός και οπτικά διαισθητικός τρόπος για την εισαγωγή των μαθητών στον προγραμματισμό, αποτελούν ταυτόχρονα και μια ισχυρή μέθοδο διδασκαλίας του προγραμματισμού. Η εργασία αυτή διαπραγματεύεται το ρόλο των μικρο-γλωσσών προγραμματισμού και ιδιαίτερα τη μικρο-γλώσσα του ρομπότ Karel. Στηριζόμενοι σ' αυτή τη μικρο-γλώσσα κατασκευάσαμε ένα διδακτικό μικρόκοσμο με στόχο να διδάξουμε τη σχεδίαση αλγορίθμων, το σχεδιασμό προγράμματος και την αποσφαλμάτωση προγραμμάτων αλλά και να στηρίξουμε την αξιολόγηση των προγραμμάτων των μαθητών καθώς και την καταγραφή των ενεργειών τους όταν αυτοί αναπτύσσουν ένα πρόγραμμα.

Abstract

Even though mini-languages are a simple and a visually intuitive way to introduce students to programming, they are a powerful method for teaching programming. This paper deals with the role of mini-languages and especially with the robot's Karel mini-language. Based on this mini-language we develop a didactic microworld in order to teach algorithmic thinking, program development and debugging. Another important goal is to support the evaluation of students' programs by using this microworld. A possible extension of the environment proposed is to record students' actions during the face of program development.

Εισαγωγή

Ακόμη και σήμερα οι αρχές του προγραμματισμού συνεχίζουν να διδάσκονται με βάση την ίδια παλιά μέθοδο. Αυτή η μέθοδος βασίζεται στη χρήση: μιας γλώσσας προγραμματισμού γενικού σκοπού, όπως Pascal, C κτλ, ενός επαγγελματικού περιβάλλοντος προγραμματισμού και ενός συνόλου προβλημάτων που αναφέρονται σε επεξεργασία αριθμών και συμβόλων. Στην πλειονότητά της η ακαδημαϊκή κοινότητα έχει αποδεχθεί ότι αυτή η μέθοδος διδασκαλίας των αρχών προγραμματισμού δεν είναι αποτελεσματική και ιδιαίτερα όταν αυτή εφαρμόζεται σε μικρές ηλικίες μαθητών. Οι λόγοι της αναποτελεσματικότητας της μεθόδου θα μπορούσαν να συνοψιστούν στους εξής [Brusilovsky et al. 97]:

- Οι γλώσσες προγραμματισμού γενικού σκοπού είναι μεγάλου μεγέθους και αρκετά ιδιοσυγκρασιακές. Κατά συνέπεια η εκμάθηση των βασικών αρχών προγραμματισμού και των βασικών δομών της γλώσσας απαιτούν υψηλό διανοητικό κόστος.
- Τα προγραμματιστικά περιβάλλοντα αναπτύχθηκαν για να χρησιμοποιηθούν από επαγγελματίες και όχι από αρχάριους προγραμματιστές. Οι συντάκτες προγραμμάτων (editors), μεταφραστές (compilers) και αποσφαλματιστές (debuggers) έχουν δημιουργηθεί για επαγγελματίες προγραμματιστές και δημιουργούν επιπλέον εμπόδια στους αρχάριους προγραμματιστές.
- Τα περιβάλλοντα προγραμματισμού δεν διαθέτουν δυνατότητες οπτικοποίησης της εκτέλεσης ενός προγράμματος. Έτσι η διαδικασία της εκτέλεσης παραμένει «κρυμμένη» μ' αποτέλεσμα οι μαθητές να αναπτύσσουν μια αντίληψη για τη φάση της εκτέλεσης προσανατολισμένη περισσότερο σε Είσοδο / Έξοδο δεδομένων. Μ' αυτό τον τρόπο η έλλειψη οπτικής ανάδρασης δεν βοηθά στην κατανόηση της σημασιολογίας της γλώσσας.
- Οι γλώσσες προγραμματισμού γενικού σκοπού έχουν αναπτυχθεί για διαχείριση αριθμών και συμβόλων. Έτσι, αναγκαστικά και τα προβλήματα που επιλέγονται για να διδαχθούν στους μαθητές είναι πέραν της καθημερινότητας που βιώνουν και κατά συνέπεια δεν τους προκαλούν ενδιαφέρον.

Μια εναλλακτική προσέγγιση της διδασκαλίας των αρχών προγραμματισμού είναι η επινόηση *διδακτικών μικρόκοσμων* (*didactic microworlds*), οι οποίοι συνήθως διαθέτουν τη δική τους *μικρο-γλώσσα προγραμματισμού* (*mini-language*).

Η παρούσα εργασία πραγματεύεται το ρόλο των μικρο-γλωσσών προγραμματισμού και ιδιαίτερα της μικρο-γλώσσας του ρομπότ Karel [Pattis 81 & 95]. Στηριζόμενοι σ' αυτή τη μικρο-γλώσσα κατασκευάσαμε ένα διδακτικό μικρόκοσμο με στόχο να διδάξουμε τις αρχές προγραμματισμού αλλά και να στηρίξουμε την αξιολόγηση των προγραμμάτων των μαθητών καθώς και την καταγραφή των ενεργειών των μαθητών όταν αναπτύσσουν ένα πρόγραμμα.

Διδακτικοί Μικρόκοσμοι Προγραμματισμού

Η βασική ιδέα των μικρόκοσμων προγραμματισμού στηρίζεται στην κατασκευή μιας μικρής και απλής γλώσσας. Σε πολλές περιπτώσεις ο μικρόκοσμος έχει επικρατήσει να δηλώνει το συνδυασμό ενός *πρωταγωνιστή* (*actor*), ο οποίος ενεργεί στον μικρόκοσμο, και μιας γλώσσας, που συνήθως αναφέρεται σε μικρό σύνολο εντολών, οι οποίες ελέγχουν τη λειτουργία του πρωταγωνιστή. Στις πιο πολλές υπάρχουσες μικρο-γλώσσες ο πρωταγωνιστής είναι συνήθως μια χελώνα (όπως οι παραλλαγές της Logo), ένα ρομπότ (Karel, StageCreator) ή οποιαδήποτε άλλη οντότητα που μπορεί να λειτουργήσει μέσα στον μικρόκοσμο. Οι περισσότερες μικρο-γλώσσες περιλαμβάνουν όλες τις βασικές δομές μιας κλασσικής γλώσσας προγραμματισμού όπως οι δομές ελέγχου (δομές διακλάδωσης, επαναληπτικές δομές κτλ) και ένα μηχανισμό που επιτρέπει την κατασκευή νέων εντολών.

Η επιλογή μιας μικρο-γλώσσας, για μια εύκολη εισαγωγή στον προγραμματισμό, παρουσιάζει πολλά πλεονεκτήματα.

- Οι μικρο-γλώσσες προγραμματισμού έχουν περιορισμένο συντακτικό (syntax) και απλή σημασιολογία (semantics). Ο χρόνος για την εκμάθηση μιας μικρο-γλώσσας είναι μικρός και έτσι ο αρχάριος μπορεί να συγκεντρώσει τις προσπάθειές του στην ανάπτυξη του αλγόριθμου για τη λύση του προβλήματος και στην κατανόηση των βασικών αρχών σχεδιασμού ενός προγράμματος.
- Οι μικρο-γλώσσες προγραμματισμού στηρίζονται σε *μεταφορικές έννοιες* (*metaphors*), οι οποίες δίνουν τη δυνατότητα να επινοηθεί ένα μεγάλο σύνολο προβλημάτων που καλύπτουν τις βασικές αρχές προγραμματισμού αλλά και προκαλούν το ενδιαφέρον των μαθητών, αφού συνήθως έχουν σχέση με θέματα που είναι εξοικειωμένοι.
- Οι λειτουργίες που εκτελεί ο πρωταγωνιστής οπτικοποιούνται στο μικρόκοσμο στον οποίο μπορεί να ενεργήσει, όπως οι κινήσεις μιας χελώνας ή ενός ρομπότ, και έτσι γίνεται φανερή η σημασιολογία των κατασκευών καθώς και ή έννοια της εκτέλεσης ενός προγράμματος. Συνήθως, οι διδακτικοί μικρόκοσμοι συνοδεύονται και από ένα γραφικό ενδιάμεσο που προκαλεί το ενδιαφέρον του μαθητή.

Οι πρώτες προσπάθειες για μικρο-γλώσσες ξεκίνησαν με τη χελώνα της Logo [Papert 80], αν και η Logo δεν σχεδιάστηκε για εισαγωγικά μαθήματα προγραμματισμού. Ακολούθησε το Ρομπότ Karel [Pattis 81 & 95]. Από τότε έχουν παρουσιαστεί και άλλοι μικρόκοσμοι, όπως: Josef the robot [Tomek 82, 83], Turingal [Brusolovsky 91], Karel-3D [Hvorecky 92], Karel++ [Bergin 96].

Όλες οι προσπάθειες που αναφέραμε παραπάνω επικεντρώθηκαν στην κατασκευή μικρο-γλωσσών προγραμματισμού και στην επινόηση ενός συνόλου προβλημάτων που στόχο έχουν να διδάξουν την ανάπτυξη αλγορίθμων, το σχεδιασμό προγράμματος και την αποσφαλμάτωση των προγραμμάτων, προκαλώντας το ενδιαφέρον του μαθητή. Ωστόσο, θα πρέπει να σημειωθεί ότι σε ελάχιστες μόνο περιπτώσεις έγινε κάποια προσπάθεια εκμετάλλευσης της τεχνολογίας ώστε να ενσωματωθούν στα αναπτυσσόμενα εκπαιδευτικά περιβάλλοντα λειτουργίες που θα διευκόλυναν τη *διδακτική χρήση τους*. Για παράδειγμα, σε λίγους σχετικά μικρόκοσμους προγραμματισμού υπάρχουν τμήματα για την αξιολόγηση των μαθητών ή την καταγραφή των ενεργειών τους, προκειμένου ο καθηγητής να διαπιστώσει τις φάσεις από τις οποίες πέρασε κάθε μαθητής στην πορεία λύσης ενός προβλήματος.

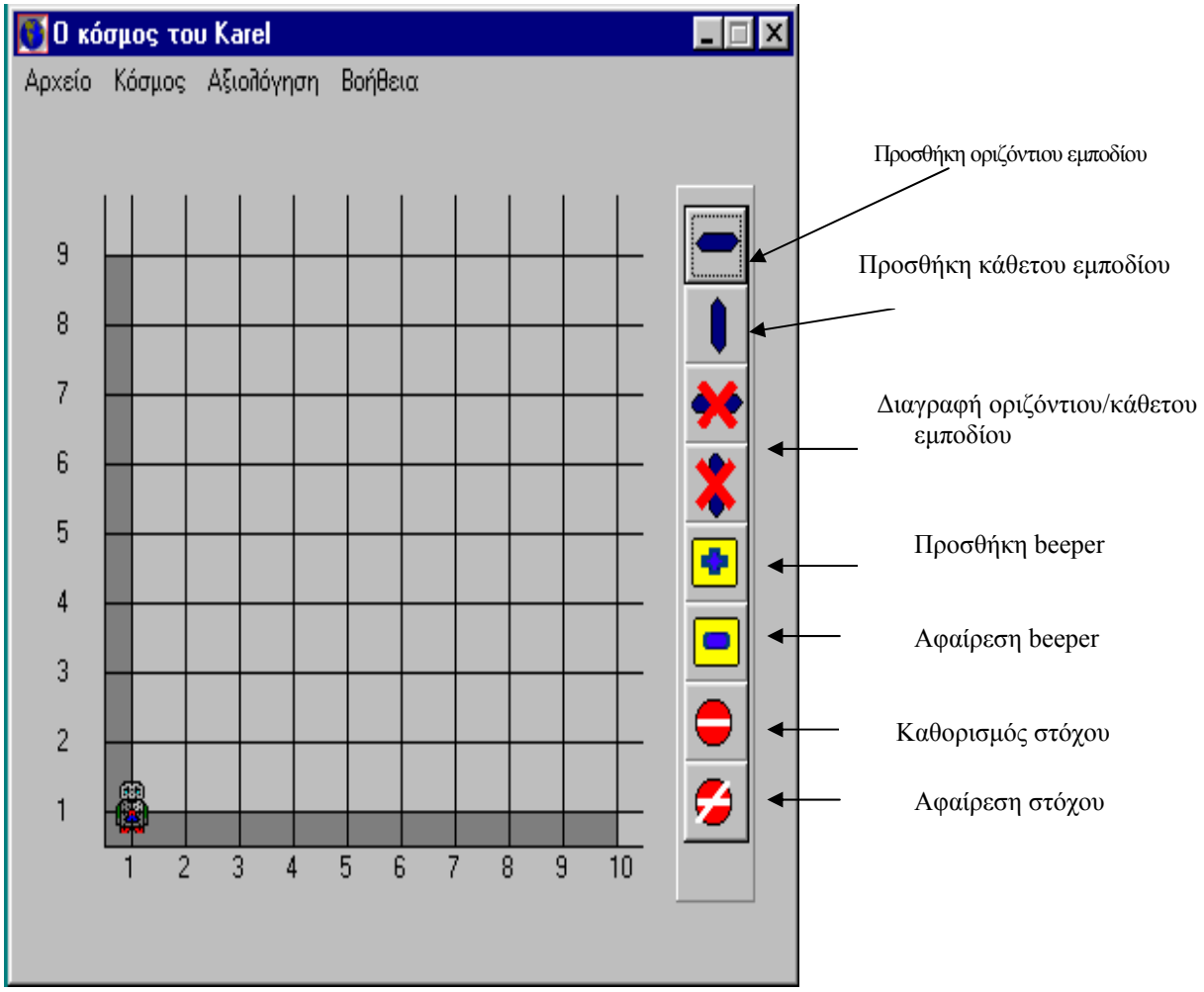
Στη συνέχεια της εργασίας αυτής παρουσιάζουμε ένα διδακτικό μικρόκοσμο ο οποίος στηρίζεται στο ρομπότ Karel δίνοντας έμφαση και σε μια άλλη συνιστώσα της διδακτικής πράξης, την αξιολόγηση. Ακόμη παρουσιάζονται τα άμεσα σχέδια μας για επέκταση του μικρόκοσμου που κατασκευάζουμε, ώστε να επιτρέψουμε την καταγραφή των ενεργειών των μαθητών, όταν αυτοί κατασκευάζουν ένα πρόγραμμα.

Ο Κόσμος του Karel- Προγραμματίζοντας με τον Karel

Ο Karel είναι ένα ρομπότ που ζει σε ένα απλό κόσμο στον οποίο ο χρήστης μπορεί να παρεμβαίνει με τη χρήση του ποντικιού. Οι εντολές που καθοδηγούν το ρομπότ για τις εργασίες που θα εκτελέσει συντάσσονται σε συγκεκριμένο συντάκτη. Ο διδακτικός μικρόκοσμος που αναπτύξαμε παρουσιάζεται στις εικόνες 1 και 2. Η εικόνα 1 παρουσιάζει τον κόσμο του Karel και η εικόνα 2 παρουσιάζει το περιβάλλον προγραμματισμού.

Ο κόσμος του Karel αποτελείται από *οριζόντιους* και *κάθετους δρόμους* που απεικονίζονται στο περιβάλλον που αναπτύξαμε ως ένα δισδιάστατο πλέγμα οριζόντιων και κάθετων γραμμών. Ο Karel μπορεί να κινηθεί πάνω στους δρόμους οριζόντια και κάθετα και όχι διαγώνια και να σταθεί στις γωνίες και όχι πάνω στους δρόμους. Οι *εξωτερικοί τοίχοι* που περιβάλλουν το πλέγμα, περιορίζουν την κίνηση του ρομπότ μέσα σε συγκεκριμένα όρια.

Στον κόσμο του Karel υπάρχουν δυο είδη αντικείμενων: οι *βομβητές* (beepers*)* και τα *εμπόδια*. Οι βομβητές είναι μικρά αντικείμενα που τοποθετεί ο χρήστης ή και ο Karel στις διασταυρώσεις των δρόμων και τα οποία επιδιώκει να μαζέψει το ρομπότ. Τα εμπόδια μπορούν να τοποθετηθούν οριζόντια και κάθετα πάνω στους δρόμους για να εμποδίσουν την πορεία του. Ο Karel είναι ελεύθερος να μετακινηθεί οπουδήποτε πάνω στους δρόμους αρκεί να μην προσπαθήσει να διασχίσει περιοχές όπου υπάρχουν εμπόδια. Αν πέσει πάνω σε κάποιο εμπόδιο τότε η λειτουργία του τερματίζεται. Στον κόσμο του Karel πρέπει να τοποθετηθεί ένα σημείο *τέλους* (stop) που καθορίζει το σημείο όπου τελειώνει η διαδρομή του. Προκειμένου να διευκολυνθεί ο χρήστης κατά την τοποθέτηση εμποδίου, βομβητή ή σημείου τέλους, ο δρομέας μετατρέπεται σε «χέρι» υποδεικνύοντας το σημείο του κόσμου που μπορεί να τοποθετηθεί το συγκεκριμένο αντικείμενο.



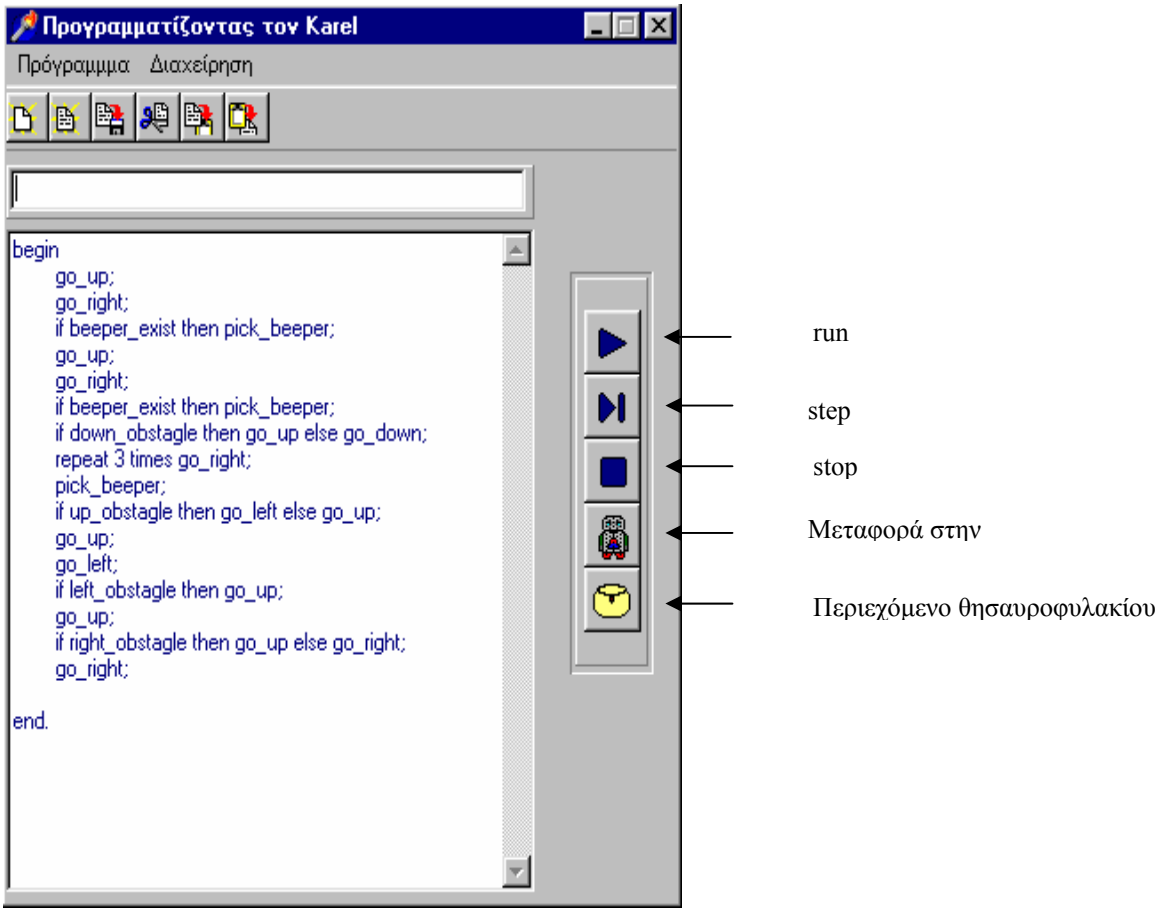
Εικόνα 1

Ο Karel μπορεί να εκτελέσει ορισμένες *εργασίες* (tasks). Η εργασία που θα εκτελέσει το ρομπότ καθοδηγείται από ένα σύνολο εντολών. Οι *βασικές εντολές* (primitive commands) που μπορεί να εκτελέσει το ρομπότ είναι:

go_up: Μετακίνηση μιας θέσης προς τα πάνω
go_down: Μετακίνηση μιας θέσης προς τα κάτω
go_l;eft: Μετακίνηση μιας θέσης προς τα αριστερά
go_right: Μετακίνηση μιας θέσης προς τα δεξιά
pick_beeper: Μαζεύει ένα beeper

Ακόμη υποστηρίζεται η δομή ελέγχου *if /then/else*, η επαναληπτική δομή *repeat (x) times <εντολή>* και τέλος υπάρχει η δυνατότητα να οριστούν νέες εντολές (υποπρογράμματα).

Το περιβάλλον σύνταξης του προγράμματος που θα εκτελέσει στη συνέχεια ο Karel είναι ένα ξεχωριστό κομμάτι από τον κόσμο του. Περιλαμβάνει ένα συντάκτη με τη βοήθεια του οποίου συντάσσεται το πρόγραμμα, επιλογές για την αποθήκευση ή την ανάκληση ενός ήδη υπάρχοντος προγράμματος και τέλος δυνατότητες για αντιγραφή/κόλληση/διαγραφή τμήματος του προγράμματος. Στην εικόνα 2 μπορούμε να δούμε το παράθυρο «Προγραμματίζοντας τον Karel», όπου φαίνεται το μενού μέσω του οποίου αποθηκεύουμε/ανακαλούμε προγράμματα στη γλώσσα του ρομπότ καθώς και τις δυνατότητες του συντάκτη. Στο αριστερό τμήμα του παράθυρου φαίνεται το τμήμα του συντάκτη προγράμματος. Στο δεξί τμήμα φαίνεται η εργαλειοθήκη



Εικόνα 2

που αναφέρεται στην εκτέλεση του προγράμματος. Με τη βοήθεια των εργαλείων αυτών επιτρέπεται η εκτέλεση του προγράμματος (run) ή η εκτέλεση του προγράμματος εντολή – εντολή (step). Τα άλλα τρία πλήκτρα διευκολύνουν στην εκτέλεση του προγράμματος και στη διαχείριση του ρομπότ. Για παράδειγμα το πλήκτρο «θησαυροφυλάκιο» εμφανίζει το σύνολο βομβητών που έχει συλλέξει το ρομπότ.

Ο μικρόκοσμος που κατασκευάσαμε έγινε με βάση το μικρόκοσμο και τη μικρο-γλώσσα που όρισε ο Pattis (1981). Η μικρο-γλώσσα του Karel περιέχει όλες τις δομές μιας Pascal-like γλώσσας και διδάσκει τις βασικές έννοιες της σειριακής εκτέλεσης, της εκτέλεσης υπό συνθήκη, της επανάληψης και της διαδικαστικής αφαίρεσης. Η έλλειψη των μεταβλητών και των εκφράσεων έχει ως αποτέλεσμα τη μείωση της πολυπλοκότητας που έχει μια γλώσσα υψηλού επιπέδου. Οι μαθητές γράφοντας προγράμματα στο περιβάλλον του Karel εξοικειώνονται με τις βασικές έννοιες του προγραμματισμού ενώ εργάζονται σ' ένα ευχάριστο και πειστικά «μεταφορικό» περιβάλλον.

Ενσωμάτωση ενός Συστήματος Αξιολόγησης & Καταγραφής Ενεργειών Μαθητών

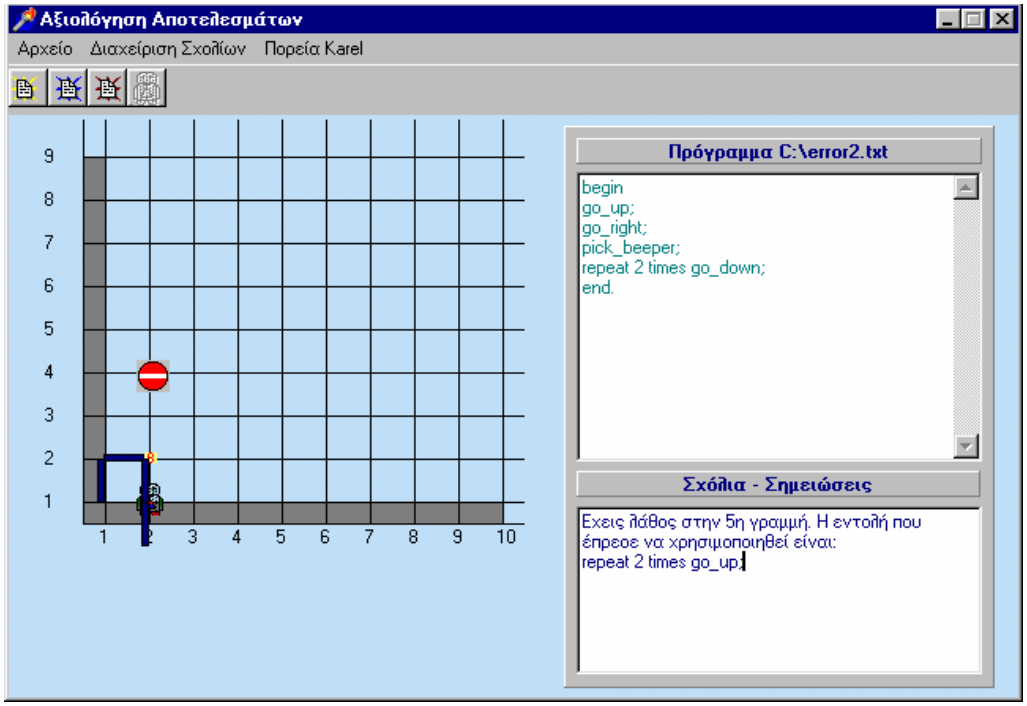
Τα μέχρι σήμερα προγραμματιστικά περιβάλλοντα, είτε πρόκειται για επαγγελματικά είτε για εκπαιδευτικά, δεν ενσωματώνουν λειτουργίες οι οποίες θα διευκόλυναν τη διδακτική τους λειτουργία, όπως, για παράδειγμα, συστήματα τα οποία θα συνεισέφεραν στην αξιολόγηση των μαθητών ή ακόμη περισσότερο δυνατότητες που θα επέτρεπαν την καταγραφή των ενεργειών των μαθητών. Συνήθως ο δάσκαλος εκείνο που αξιολογεί είναι το τελικό προϊόν, δηλαδή το πρόγραμμα. Οι ενδιάμεσες προσπάθειες των μαθητών συνήθως δεν καταγράφονται από τα υπάρχοντα προγραμματιστικά περιβάλλοντα, με αποτέλεσμα να μην γνωρίζει ο δάσκαλος την πορεία που ακολουθεί ο μαθητής όταν επεξεργάζεται τη λύση ενός προβλήματος. Επιπλέον, όταν χρησιμοποιούνται επαγγελματικά προγραμματιστικά περιβάλλοντα, δεν δίνεται η δυνατότητα της βήμα προς βήμα εκτέλεσης τους προγράμματος με ταυτόχρονη οπτικοποίηση της εκτέλεσης αυτής. Η χρήση των ενσωματωμένων αποσφαλματιστών στα επαγγελματικά προγραμματιστικά περιβάλλοντα απαιτεί προφανώς ένα υψηλό επίπεδο κατανόησης και γενικότερα γνώσης προγραμματισμού, το οποίο δε διαθέτουν οι αρχάριοι μαθητές. Εξάλλου μια οπτικοποίηση της εκτέλεσης ενός προγράμματος θα έπρεπε να διαχειρίζεται με ιδιαίτερο τρόπο στοιχεία του προγράμματος τα οποία συχνά αναπαριστάνονται με γραφικό τρόπο, όπως για παράδειγμα οι δείκτες, τα αλφαριθμητικά κτλ.

Σε μια προσπάθεια να υποστηριχθεί η διαδικασία της αξιολόγησης αλλά και να βοηθηθεί ο μαθητής στον έλεγχο του προγράμματος που παράγει, επεκτείναμε το διδακτικό μικρόκοσμο του Karel.

Το περιβάλλον του μικρόκοσμου που αναπτύξαμε περιλαμβάνει τη δυνατότητα αποθήκευσης του κόσμου του Karel καθώς και του αντίστοιχου προγράμματος μετά από κάθε εκτέλεσή του. Σε περίπτωση που το πρόγραμμα του μαθητή

περιλαμβάνει λογικά λάθη, η εκτέλεση διακόπτεται με την εμφάνιση σχετικού μηνύματος, παρέχοντας τη δυνατότητα να αποθηκευθεί ο κόσμος του Karel και το αντίστοιχο πρόγραμμα. Με την επιλογή της λειτουργίας «Αξιολόγηση» από το μενού που φαίνεται στην εικόνα 1 εμφανίζεται το παράθυρο της εικόνας 3.

Κατά την ενεργοποίηση της λειτουργίας της αξιολόγησης, οι τρεις περιοχές της εικόνας 3 είναι κενές. Μέσω της επιλογής «Αρχείο» ανακαλείται ο κόσμος του Karel, τον οποίο είχαμε αποθηκεύσει μαζί με το αντίστοιχο πρόγραμμα.



Εικόνα 3

Μέσω της επιλογής «Πορεία Karel» παρουσιάζεται με έντονη γραμμή η πορεία που ακολούθησε το ρομπότ όταν εκτέλεσε το αντίστοιχο πρόγραμμα. Ο δάσκαλος μπορεί να διατυπώσει κάποια σύντομα σχόλια σχετικά με το λανθασμένο πρόγραμμα, ώστε ο μαθητής να αντιληφθεί το λάθος που έκανε βοηθούμενος από τα 2 πλαίσια εκφοράς του προγράμματος, δηλαδή του κόσμου του Karel, όπου οπτικοποιούνται τα βήματα της εκτέλεσης και του συντάκτη του προγράμματος. Προφανώς ο δάσκαλος μπορεί να προσφέρει πολύ ουσιαστικότερη βοήθεια στους μαθητές αν τους καθοδηγήσει ώστε αυτοί μόνοι τους να βρουν τη σωστή λύση. Έτσι σχεδιάζεται η δημιουργία ενός συστήματος διαχείρισης δυσκολιών στο οποίο, για παράδειγμα, ο δάσκαλος υποδεικνύει την ορθή πορεία χρησιμοποιώντας έντονες γραμμές διαφόρων χρωμάτων και καλώντας το μαθητή να διορθώσει το πρόγραμμά του.

Συμπεράσματα

Θεωρούμε ότι οι μικρο-γλώσσες προγραμματισμού αν και είναι ένας απλός και οπτικά διαισθητικός τρόπος για την εισαγωγή των μαθητών στον προγραμματισμό, αποτελούν ταυτόχρονα και μια ισχυρή μέθοδο διδασκαλίας του προγραμματισμού. Με τη βοήθεια των μικρο-γλωσσών προγραμματισμού επιχειρείται μια εισαγωγή στον προγραμματισμό αλλά και μια πολύ καλή εισαγωγή στην Επιστήμη των ΗΥ, αφού οι βασικές αρχές του προγραμματισμού αποτελούν τη βάση για τη θεμελίωση του λογικού και του αφαιρετικού συλλογισμού που είναι θεμελιώδεις έννοιες στην επιστήμη της Πληροφορικής.

Πιστεύουμε ότι οι διδακτικοί μικρόκοσμοι ανοίγουν μια νέα προοπτική στην εκπαιδευτική διαδικασία όχι μόνο ως μέσο διδασκαλίας εισαγωγικών εννοιών της Πληροφορικής αλλά και ως μέσο αξιολόγησης και μελέτης των αντιλήψεων των μαθητών.

Για το σκοπό αυτό αναπτύξαμε το περιβάλλον προγραμματισμού βασισμένοι στον κόσμο του Karel. Η προσπάθεια μας αυτή συνίσταται στην ανάπτυξη ενός ολοκληρωμένου περιβάλλοντος το οποίο εκτός από την υποστήριξη που θα παρέχει στη φάση της αξιολόγησης των εργασιών των μαθητών αποβλέπει και στην καταγραφή των αντιλήψεων των μαθητών. Μέχρι τώρα ελάχιστες εργασίες έχουν δημοσιευθεί σχετικά με τις αντιλήψεις των μαθητών όταν αυτοί αναπτύσσουν προγράμματα πολύ δε περισσότερο δεν έχουν αναπτυχθεί αυτόματα περιβάλλοντα καταγραφής των ενεργειών των μαθητών. Στόχος μας είναι να ενσωματώσουμε λειτουργίες οι οποίες θα επιτρέπουν την καταγραφή των ενεργειών των μαθητών. Κάθε φορά που ο μαθητής θα ζητά την εκτέλεση του προγράμματος αυτόματα θα αποθηκεύεται ο κόσμος του Karel μαζί με το αντίστοιχο πρόγραμμα. Πιστεύουμε ότι όταν ο μαθητής ζητά την εκτέλεση του προγράμματος αυτή η ενέργεια του υποδηλώνει, ότι πιστεύει πως έχει οδηγηθεί στη λύση του προβλήματος ή πως έχει ολοκληρώσει μια σημαντική φάση της λύσης. Η δυνατότητα να ανακαλούμε και να βλέπουμε

τις διαδοχικές φάσεις της ανάπτυξης του προγράμματος, θα βοηθήσει στην καταγραφή των λανθασμένων αντιλήψεων των μαθητών, στην καταγραφή των συστηματικών λαθών και συνεπώς στην ανάπτυξη διδακτικών καταστάσεων που θα διευκολύνουν τη διδασκαλία του προγραμματισμού.

Πιστεύουμε ότι το περιβάλλον αυτό μπορεί να χρησιμοποιηθεί για τη διδασκαλία του προγραμματισμού τόσο στη δευτεροβάθμια όσο και στην τριτοβάθμια εκπαίδευση. Το επίπεδο γνώσεων των χρηστών μπορεί να καθορίσει και το αντίστοιχο επίπεδο δυσκολίας των προτεινομένων προβλημάτων. Για παράδειγμα, μπορεί να ζητηθεί από τους σπουδαστές όταν είναι προχωρημένοι να προσδιορίσουν τη βέλτιστη διαδρομή του Karel, ή να επιλύσουν πιο πολύπλοκα προβλήματα. Ήδη σε πολλές χώρες έχουν από χρόνια χρησιμοποιήσει τη μικρο-γλώσσα του Karel, καθώς και άλλες μικρές-γλώσσες, για να διδάξουν προγραμματισμό στη δευτεροβάθμια και τριτοβάθμια εκπαίδευση [Brusilovsky et al. 97]. Σε πολλά πανεπιστήμια τα εισαγωγικά μαθήματα προγραμματισμού γίνονται με κάποια μικρο-γλώσσα ή με κάποια προγραμματιστικά περιβάλλοντα αναπτυγμένα ειδικά για εκπαιδευτικούς σκοπούς, όπως τα περιβάλλοντα Logo (π.χ. Πανεπιστήμιο J. Fourier, Grenoble), Pascal Genie 1.0 [Miller et. Al. 94], Thetis [Freund and Roberts 96] και άλλα.

Τελειώνοντας θα θέλαμε να επισημάνουμε ότι τα μέχρι τώρα δημοσιευμένα αποτελέσματα σχετικά με την αποτελεσματικότητα των μικρο-γλωσσών προγραμματισμού δεν στηρίζονται σε μεγάλη εμπειρική αξιολόγηση [Brusilovsky et al. 97]. Θεωρούμε ότι ο μικρόκοσμος που αναπτύσσουμε θα συμβάλλει προς αυτή την κατεύθυνση με τη δυνατότητα που ενσωματώνει για καταγραφή των ενεργειών των μαθητών.

Βιβλιογραφία

[Bergin 96] Bergin J., Stehlik M., Roberts J., Pattis R., (1996), *Karel ++ - A Gentle Introduction to the Art of Object-Oriented Programming*, John Wiley.

[Brusilovsky et al. 97] Brusilovsky P. et al, (1997), Mini-languages: a way to learn programming principle, *Education and Information Technologies*, 2, 65-83.

[Freund and Roberts 96] Freund S., Roberts E., (1996), Thetis: An ANSI C Programming Environment Designed for Introductory Use, *In Proceedings SIGCSE 96*, ACM Press.

[Hvorecky 92] Hvorecky J., (1992), Karel the Robot for PC, In *Proceedings of the east-West Conference on Emerging Computer Technologies in Education*, P. Brusilovsky and V. Stefanuk (eds.), Moscow, 6-9 April 1992, 157-160.

[Miller et al. 94] Miller P., Pane J. Meter G., Vorthmann S., (1994), Evolution of Novice Programming Environments: The structure editors of Carnegie Mellon University, *Interactive Learning Environment*, 4, 2, 140-158.

[Papert 80] Papert S., (1980), *Mindstorms, children, Computers and Powerful Ideas*, New York, Basic Books.

[Pattis 81] Pattis R., (1981), *Karel – The Robot, A Gentle Introduction to the Art of Programming*, London, Wiley.

[Pattis 95] Pattis R., Roberts J., Stehlik M., (1995) *Karel – The Robot, A Gentle Introduction to the Art of Programming*, 2nd edn., London, Wiley.

[Tomek 82] Tomek I., (1982), Josef the Robot, *Computers and Education*, 6(3), 287-293.

[Tomek 83] Tomek I., (1983), *The first Book of Josef, Englewood Cliffs*, Prentice-Hall.