

**Επισκόπηση του χώρου των εκπαιδευτικών περιβαλλόντων
προγραμματισμού υπολογιστών: τεχνολογικές και παιδαγωγικές
προβολές**

Γιώργος Φεσάκης¹ και Αγγελική Δημητρακοπούλου²

¹ Διδάσκων ΠΔ407/80, ΤΕΠΑΕΣ, Πανεπιστήμιο Αιγαίου

Λ. Δημοκρατίας, 1, 85100 Ρόδος

Τηλ: 2241069675, 6937901299

E-mail: gfsakis@rhodes.aegean.gr

² Αναπληρώτρια Καθηγήτρια, ΤΕΠΑΕΣ, Παν/μιο Αιγαίου

E-mail: adimitr@aegean.gr

Επισκόπηση του χώρου των εκπαιδευτικών περιβαλλόντων προγραμματισμού υπολογιστών: τεχνολογικές και παιδαγωγικές προβολές

Περίληψη

Για την εισαγωγή στον προγραμματισμό των υπολογιστών συχνά χρησιμοποιούνται επαγγελματικά περιβάλλοντα ανάπτυξης εφαρμογών που δεν έχουν σχεδιαστεί για διδασκαλία (ιδιαίτερα σε νεαρές ηλικίες). Στην παρούσα εργασία γίνεται επισκόπηση περιβαλλόντων εκπαίδευσης στον προγραμματισμό υπολογιστών και προτείνεται ένα σύστημα ταξινόμησής τους. Οι προτεινόμενες κατηγορίες προβάλλονται σε διάφορες διαστάσεις του χώρου όπως το επίπεδο αφαίρεσης της υπολογιστικής μηχανής, το συντακτικό της γλώσσας, τα υποδείγματα προγραμματισμού και οι ηλικιακές περίοδοι εφαρμογής. Στόχος της εργασίας είναι η υποστήριξη των εκπαιδευτικών στο διδακτικό σχεδιασμό για τον προγραμματισμό και τις διαθεματικές εφαρμογές του καθώς και η επικαιροποίηση του προσανατολισμού των σχετικών ερευνών.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Διδασκαλία προγραμματισμού υπολογιστών, Εκπαιδευτικά υπολογιστικά περιβάλλοντα

Εισαγωγή

Ο προγραμματισμός υπολογιστών δεν αποτελεί απλά μια ικανότητα μεγάλης οικονομικής σημασίας αλλά έχει και αυξημένο εκπαιδευτικό ενδιαφέρον. Ο προγραμματισμός υπολογιστών θεωρείται δραστηριότητα με την οποία καλλιεργούνται ανώτερες μορφές σκέψης όπως η αναλυτική, η συνθετική, η αναγνώριση προτύπων, κ.α. Με τον προγραμματισμό είναι δυνατό να βελτιωθεί η ικανότητα επίλυσης προβλημάτων (Papert 1991). Ο προγραμματισμός θεωρείται επίσης δεξιότητα - κλειδί για την προσέγγιση και κατανόηση άλλων θεμάτων των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών (ΤΠΕ). Ειδικά θέματα της διδακτικής του προγραμματισμού υπολογιστών έχουν απασχολήσει έλληνες και ξένους ερευνητές (Pea 1986, Spohrer and

Soloway 1986, Κόμης 2005; Τζιμογιάννης και Κόμης 2000; Stephenson et al. 2005, Φεσάκης και Δημητρακοπούλου 2005). Οι περισσότερες έρευνες βασίζονται στο διαδικαστικό-δομημένο (imperative) υπόδειγμα του προγραμματισμού σύμφωνα με το οποίο ο προγραμματιστής σχηματίζει προγράμματα βάζοντας στη σειρά ή εμφωλεύοντας εντολές με βάση τρεις βασικές δομές ελέγχου (ακολουθία, επιλογή και επανάληψη). Τα προγράμματα ελέγχουν και καθοδηγούν τον υπολογιστή για την επίλυση ενός προβλήματος. Στα εισαγωγικά μαθήματα στον προγραμματισμό, τα προβλήματα επιλέγονται συχνά από τον χώρο των μαθηματικών (Knuth 1968). Το υπόδειγμα αυτό του προγραμματισμού είναι ελάχιστα τροποποιημένο από τη μορφή που είχε όταν πρωτοεμφανίστηκαν οι δομημένες γλώσσες υψηλού επιπέδου.

Ο επαγγελματικός προγραμματισμός των υπολογιστών έχει εξελιχθεί με το πέρασμα του χρόνου και έχει πάρει διάφορες μορφές που καλούνται συχνά και υποδείγματα (programming paradigms). Τα υποδείγματα διαφοροποιούνται ανάλογα με το σύστημα που θα εκτελέσει το πρόγραμμα ή/και το χρησιμοποιούμενο σύστημα αναπαράστασης για τα προγράμματα. Για παράδειγμα, ο προγραμματισμός διακρίνεται σε παράλληλο (πολλοί επεξεργαστές με κοινή μνήμη), κατανεμημένο (πολλοί επεξεργαστές με ξεχωριστή μνήμη σε δίκτυο), λογικός (π.χ. με την γλώσσα PROLOG), αντικειμενοστρεφής, καθοδηγούμενος από τα γεγονότα, εικονικός, ενσωματωμένος, προγραμματισμός δυναμικών σελίδων στο διαδίκτυο, κλπ. Η εξοικείωση των μαθητών με διάφορα υποδείγματα προγραμματισμού προτείνεται ως παιδαγωγική προσέγγιση από τον Stephenson (Stephenson et al. 2005), ενώ παράλληλα τονίζεται η ανάγκη επιπρόσθετων ερευνών για την τεκμηρίωσή της.

Παράλληλα ο επαγγελματικός προγραμματισμός ως διαδικασία έχει μεταβληθεί πολύ από τα πρώτα χρόνια της COBOL, της ADA, της FORTRAN, PASCAL, κ.λ.π. Η

μεταβολή αντικατοπτρίζεται χαρακτηριστικά στην εξέλιξη των επαγγελματικών περιβαλλόντων ανάπτυξης εφαρμογών. Ο συνδυασμός απλού συντάκτη κειμένου και μεταφραστή ή διερμηνευτή έχει περάσει ανεπιστρεπτί, τα σύγχρονα περιβάλλοντα αποτελούνται από πληθώρα εργαλείων και κυρίως από προκαθορισμένες ιεραρχίες κλάσεων αντικειμένων, οι οποίες δυσχεραίνουν την εξοικείωση των προγραμματιστών παρέχοντάς τους ένα αφηρημένο περιβάλλον μέσα στο οποίο θα εκτελεστούν τα προγράμματα που αναπτύσσουν. Οι προκαθορισμένες ιεραρχίες κλάσεων αντικειμένων βελτιώνουν βέβαια την παραγωγικότητα του εξοικειωμένου προγραμματιστή επειδή υποστηρίζουν την επαναχρησιμοποίηση κώδικα αλλά παράλληλα είναι δυνατό να τον δεσμεύουν με το συγκεκριμένο περιβάλλον ανάπτυξης.

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

περιβαλλόντων προγραμματισμού τα καθιστά εφαρμόσιμα μόνο στις τελευταίες τάξεις της δευτεροβάθμιας εκπαίδευσης.

Στην διεθνή βιβλιογραφία καταγράφονται συχνά προτάσεις περιβαλλόντων υποστήριξης της εισαγωγικής μάθησης του προγραμματισμού που αφορούν κυρίως στην τριτοβάθμια εκπαίδευση. Χαρακτηριστικά τέτοια περιβάλλοντα περιγράφονται στο (Truong et al. 2005) ενώ μια επισκόπηση περιβαλλόντων καταγράφεται στο (Deek and McHugh 1998). Οι εργασίες αυτές σηματοδοτούν το ενδιαφέρον του ζητήματος αλλά δεν καλύπτουν τις ανάγκες της γενικής εκπαίδευσης πριν την Τριτοβάθμια. Έτσι στην επισκόπηση των Deek και McHugh η κατηγοριοποίηση των περιβαλλόντων γίνεται με τεχνολογικά κριτήρια στις εξής κατηγορίες: *προγραμματιστικά περιβάλλοντα, βοηθήματα εκσφαλμάτωσης, νοήμονα συστήματα διδασκαλίας* (Intelligent Tutoring Systems) και *νοήμονα προγραμματιστικά περιβάλλοντα* (Intelligent programming environments). Η κατηγοριοποίηση αυτή δεν αναδεικνύει τις σύγχρονες επιλογές που έχει ο εκπαιδευτικός στη διάθεσή του. Στην παρούσα εργασία θα αναλύσουμε περισσότερο την πρώτη κατηγορία «προγραμματιστικά περιβάλλοντα» ορίζοντας άξονες που θα λαμβάνουν υπόψη εμπειρία και γνώση που έχει αρχίσει να αναπτύσσεται από τις προσπάθειες εφαρμογής του προγραμματισμού υπολογιστών στην εκπαίδευση. Από ευρύτερη εκπαιδευτική άποψη ενδιαφέρει μια κατηγοριοποίηση που θα δίνει πληροφορίες για ζητήματα όπως: ποια πεδία προβλημάτων μπορούν να αξιοποιηθούν για την παραγωγή μαθησιακών δραστηριοτήτων, ποιες οι γνωστικές απαιτήσεις του περιβάλλοντος, το εύρος των ηλικιών για το οποίο το περιβάλλον είναι κατάλληλο, πόσο προσανατολισμένο στην υπολογιστική μηχανή ή στον χώρο προβλημάτων είναι το σύστημα αναπαράστασης των προγραμμάτων που υιοθετείται, ποια συντακτικά γλωσσών και ποια προγραμματιστικά υποδείγματα υποστηρίζονται, αν υπάρχει

δυνατότητα υλοποίησης διαθεματικών δραστηριοτήτων, η συνέπεια με σύγχρονες διδακτικές και μαθησιακές αντιλήψεις κ.α.

Λαμβάνοντας υπόψη τα παραπάνω και την ομαδοποίηση υπαρχόντων περιβαλλόντων με κανόνες συνάφειας και ομοιότητας προτείνεται το επόμενο σύστημα αξόνων ταξινόμησης:

A. Άξονας διαβάθμισης της ‘αφαίρεσης’ του υπολογιστικού συστήματος: Η τοποθέτηση ενός περιβάλλοντος στον άξονα αυτό μας δίνει μια ιδέα για το πόσο κοντά θα είναι το πρόγραμμα στον χώρο του προβλήματος σε σχέση με τον χώρο της υπολογιστικής μηχανής που θα εκτελεστεί. Με άλλα λόγια ενδιαφέρει για κάθε περιβάλλον αν το παρεχόμενο σύστημα αναπαράστασης των προγραμμάτων χρησιμοποιεί όρους που εξαρτώνται από την αρχιτεκτονική της μηχανής (π.χ. καταχωρητής, αθροιστής, θέση μνήμης, κλπ) ή από τον χώρο του προβλήματος (π.χ. μεταβλητή της ταχύτητας, κ.α). Το αφαιρετικό επίπεδο που προβάλλει ένα εκπαιδευτικό περιβάλλον καθορίζει σημαντικά τη δυνατότητα χρήσης του σε μαθησιακές δραστηριότητες γενικού τύπου και στο πλαίσιο διαφόρων μαθημάτων ή το αν μπορεί να αξιοποιηθεί κυρίως για τον προγραμματισμό υπολογιστών. Τέλος το *επίπεδο αφαίρεσης από τη μηχανή* καθορίζει συχνά και τα είδη μαθησιακών δραστηριοτήτων που υποστηρίζονται.

B. Άξονας αναπτυξιακής-ηλικιακής καταλληλότητας: Ο συγκεκριμένος άξονας δίνει πληροφορίες για το ηλικιακό διάστημα που ενδείκνυται η χρήση κάθε περιβάλλοντος. Η ηλικιακή κατανομή των περιβαλλόντων προγραμματισμού μπορεί ως ένα βαθμό να υπαγορευθεί από τις γενικές

δεξιότητες που αυτά απαιτούν, αλλά η εμπειρική της επικύρωση αποτελεί συχνότερα ένα ανοικτό πρόβλημα.

Γ. Άξονας των υποστηριζόμενων υποδειγμάτων προγραμματισμού: Η υποστήριξη πολλαπλών υποδειγμάτων προγραμματισμού είναι συνηθισμένη στα σύγχρονα περιβάλλοντα και επιτρέπει εκφραστική ελευθερία στον προγραμματιστή. Σε ένα περιβάλλον προγραμματισμού η έλλειψη υποστήριξης ενός υποδείγματος μπορεί συχνά να αντιμετωπιστεί με προσομοίωση αυξάνοντας κατά πολύ το απαιτούμενο γνωστικό φορτίο. Το προγραμματιστικό υπόδειγμα είναι σημαντικό γιατί καθορίζει σε μεγάλο βαθμό το χώρο των προβλημάτων τα οποία μπορούμε να χρησιμοποιήσουμε στο σχεδιασμό μαθησιακών δραστηριοτήτων.

Δ. Άξονας υποστηριζόμενων συντακτικών γλωσσών προγραμματισμού: Τα περισσότερα περιβάλλοντα προγραμματισμού υιοθετούν και υποστηρίζουν τη σύνταξη προγραμμάτων σύμφωνα με τους κανόνες τουλάχιστον μιας γλώσσας προγραμματισμού. Οι διαφωνίες για το ποιες γλώσσες είναι καταλληλότερες δε φαίνεται να συμβιβάζονται εύκολα και επομένως η υποστήριξη πολλαπλών εναλλακτικών συντακτικών συνιστά ένα ενδιαφέρον χαρακτηριστικό.

Ε. Άξονας 'αφηρημένου σχήματος' της διαδικασίας προγραμματισμού: Στον άξονα αυτό τοποθετούμε τα εκπαιδευτικά περιβάλλοντα με βάση το σχήμα που υπαγορεύουν στον προγραμματιστή για τη διαδικασία του προγραμματισμού και την εκτέλεση των προγραμμάτων. Για παράδειγμα, στα περισσότερα διαδικαστικά περιβάλλοντα ο προγραμματιστής όταν προγραμματίζει μοιάζει σαν να ετοιμάζει μια σειρά «διαταγών» για τον

υπολογιστή–εκτελεστή. Αντίθετα στον λογικό προγραμματισμό, ο προγραμματιστής ορίζει ένα σύνολο γεγονότων και ένα σύνολο κανόνων ενώ η «εκτέλεση» του προγράμματος αρχίζει με την υποβολή ενός ερωτήματος το οποίο αφορά τη συνέπεια μιας πρότασης σε σχέση με τη βάση γνώσης–πρόγραμμα. Στον αντικειμενοστρεφή προγραμματισμό, ο προγραμματιστής μοιάζει με σεναριογράφο-σκηνοθέτη που καλείται να ορίσει τα αντικείμενα-χαρακτήρες που έχουν κάποιο ρόλο να παίξουν στο πλαίσιο ενός έργου. Η εκτέλεση του προγράμματος μπορεί να συγκριθεί με την τοποθέτηση των αντικειμένων σε μια σκηνή στην οποία μπορούν να αρχίσουν να αλληλεπιδρούν, με βάση τις προδιαγεγραμμένες αρχέτυπες συμπεριφορές τους, τόσο μεταξύ τους όσο και με τον χρήστη του προγράμματος. Ο άξονας του αφηρημένου σχήματος της διαδικασίας προγραμματισμού είναι εκπαιδευτικά σημαντικός επειδή καθορίζει σε μεγάλο βαθμό την πολυπλοκότητα των προβλημάτων που μπορούν να αντιμετωπιστούν αλλά και την αναπτυξιακή καταλληλότητα ενός προγραμματιστικού περιβάλλοντος.

Σκοποί της παρούσας εργασίας είναι:

1. Η παρουσίαση και ταξινόμηση ποικιλίας διαθέσιμων περιβαλλόντων, ώστε να διευκολύνεται η διαδικασία επιλογής του καταλληλότερου από τον ενδιαφερόμενο εκπαιδευτικό, ανάλογα με την εκπαιδευτική κατάσταση (εκπαιδευτικός σχεδιασμός).
2. Η ενημέρωση των ενδιαφερόμενων εκπαιδευτικών με σύγχρονες ιδέες για εκπαιδευτικά περιβάλλοντα και μαθησιακές δραστηριότητες σε διάφορες ηλικίες.

3. Η σκιαγράφιση νέων γενικών κατευθύνσεων και προβλημάτων για τη σχετική έρευνα της Διδακτικής της Πληροφορικής αλλά και της αξιοποίησης των ΤΠΕ στην εκπαίδευση γενικότερα.

Στα επόμενα, αρχικά παρουσιάζονται οι προτεινόμενες κατηγορίες εκπαιδευτικών περιβαλλόντων για προγραμματισμό υπολογιστών μαζί με χαρακτηριστικά παραδείγματα και γενικούς τύπους μαθησιακών δραστηριοτήτων. Στη συνέχεια, σχολιάζονται οι κατηγορίες όσον αφορά τους άξονες που αναφέρθηκαν παραπάνω. Τέλος, συνοψίζεται η παρουσίαση των κατηγοριών, παρουσιάζονται σχετικά συμπερασματικά σχόλια και προτείνονται ερευνητικές κατευθύνσεις.

Εκπαιδευτικά περιβάλλοντα προγραμματισμού υπολογιστών

Οι κατηγορίες που παρουσιάζονται στη συνέχεια σχηματίζονται από την ομαδοποίηση περιβαλλόντων που χρησιμοποιούνται για τη διδασκαλία του προγραμματισμού σε διάφορες ηλικίες. Η έμφαση δίδεται στη διάκριση των περιβαλλόντων που έχουν σχεδιαστεί με σκοπό τη διδασκαλία ενώ όλα τα επαγγελματικά περιβάλλοντα προγραμματισμού ενσωματώνονται σε ομώνυμη κατηγορία.

Η οικογένεια της LOGO

Βασισμένα στις ιδέες του Papert (Papert 1991) που εκφράζει η γλώσσα LOGO, εμφανίζονται διάφορα συστήματα εκπαίδευσης στον προγραμματισμό (Ράπτης και Ράπτη 2001). Τα συστήματα της κατηγορίας αυτής έχουν ιδιαίτερο ενδιαφέρον και εκτεταμένη εφαρμογή μεταξύ άλλων επειδή:

- έχουν τη δυνατότητα ηλικιακής προσαρμογής και υποστηρίζουν σταδιακή και βαθμιαία εισαγωγή στον προγραμματισμό,
- επιτρέπουν την υλοποίηση ποικιλίας μαθησιακών δραστηριοτήτων επίλυσης προβλημάτων,

- υποστηρίζουν διαθεματικές δραστηριότητες, καθώς και τη διδασκαλία διαφόρων αντικειμένων διευκολύνοντας την αξιοποίηση των ΤΠΕ στην εκπαίδευση.

Η κατηγορία αυτή είναι αρκετά ευρεία ώστε να περιλαμβάνει υποκατηγορίες όπως οι επόμενες: Περιπλανητές Roamers, Λογισμικοί περιπλανητές, Περιβάλλοντα προγραμματισμού LOGO.

Περιπλανητές – Roamers

Τα συστήματα της κατηγορίας αυτής αναφέρονται συχνά και ως χελώνες εδάφους (floor turtles). Πρόκειται για υλικές υλοποιήσεις της χελώνας που έχουν τη δυνατότητα να δέχονται απλές εντολές προσανατολισμού (εμπρός ένα βήμα, πίσω ένα βήμα, στρίψε δεξιά ή αριστερά κάποιες μοίρες, π.χ. 90°). Συχνά οι περιπλανητές διαθέτουν τη δυνατότητα να αφήνουν ίχνος κατά τη μετακίνησή τους με τη βοήθεια μαρκαδόρων. Οι περιπλανητές λαμβάνουν στην πράξη διάφορες μορφές όπως μέλισσες, ρομπότ, κλπ.



Σχήμα 1: Ο περιπλανητής Bee-Bot™,

Οι μαθησιακές δραστηριότητες που υλοποιούνται συνήθως σε τέτοια συστήματα περιλαμβάνουν: την έξοδο του περιπλανητή από λαβύρινθο που μπορεί να είναι ζωγραφισμένος σε κάποιο χαλί ή/και να δημιουργείται από εμπόδια (π.χ. φελιζόλ), μετακίνηση από σημείο σε σημείο (εκτέλεση διαδρομής), σχεδίαση σχημάτων, κ.α. (Σχήμα 1).

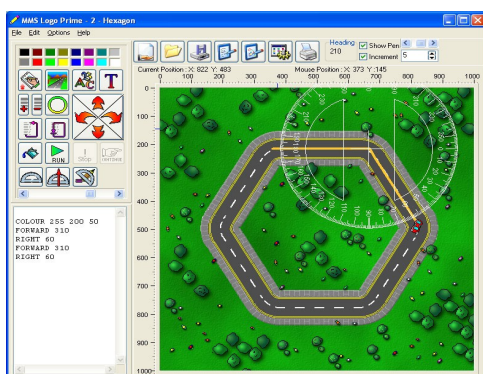
Είναι φανερό ότι τα συστήματα αυτά αποτελούν ιδανικά περιβάλλοντα προετοιμασίας μαθητών των πρώτων βαθμίδων της εκπαίδευσης για την εισαγωγή τους σε πολυπλοκότερα και ρεαλιστικότερα περιβάλλοντα προγραμματισμού της ίδιας τουλάχιστον οικογένειας. Η υπόθεση αυτή στηρίζεται στην ταύτιση των βασικών

εντολών της υλικής χελώνας με αυτές της λογισμικής-εικονικής και στην ομοιότητα που έχουν οι περιπλανητές με τα τηλεχειριζόμενα παιχνίδια που ενθουσιάζουν την πλειοψηφία των μικρών παιδιών. Εκτός από την ευκολία της χρήσης και της διαχείρισης οι περιπλανητές είναι δημοφιλείς λόγω της συμβατότητάς τους με το πρόγραμμα σπουδών και ειδικότερα με τις έννοιες του προσανατολισμού (εμπρός, πίσω, δεξιά αριστερά) και της απόστασης (μακριά, κοντά), καθώς και της μέτρησης, λόγω της διακριτής φύσης της κίνησης που βασίζεται σε σταθερής απόστασης βήματα και ορθές γωνίες. Για την τεκμηρίωση της ηλικιακής καταλληλότητας επισημαίνεται ότι οι μαθητές κατά τη διάρκεια της μαθησιακής δραστηριότητας χειρίζονται πραγματικό αντικείμενο, χρησιμοποιούν το σώμα τους και κίνηση στο χώρο, ενώ η συνεργασία μεταξύ τους επιτυγχάνεται με φυσικό τρόπο. Η έλλειψη απαίτησης γνώσης γραφής και ανάγνωσης καθώς και χρήσης υπολογιστών καθιστά τους περιπλανητές ιδανικούς για τις μικρές ηλικίες των 3-6 ετών. Η σκοπιμότητα της αξιοποίησης περιπλανητών στα νηπιαγωγεία υποστηρίζεται και από σχετικές έρευνες (Faculty of Education, Curtin University of Technology 1997).

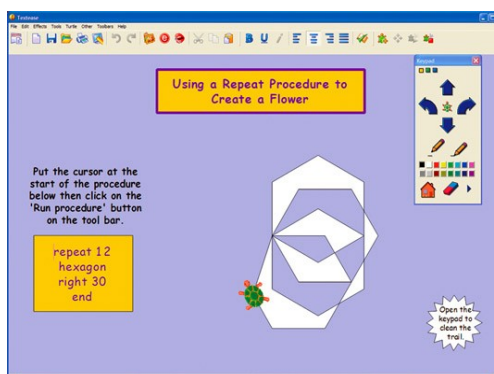
Λογισμικοί περιπλανητές

Οι λογισμικοί περιπλανητές δημιουργούνται ως προσομοιώσεις φυσικών περιπλανητών με ένα πρόγραμμα σε υπολογιστή. Τον περιπλανητή υποκαθιστά μια λογισμική οντότητα που απεικονίζεται με διάφορες μορφές όπως χελώνα, όχημα, κλπ. Τα περιβάλλοντα αυτά επιτρέπουν σταδιακή απομάκρυνση των παιδιών από τις χελώνες εδάφους προσεγγίζοντας προς την τυπική LOGO. Για παράδειγμα, οι λογισμικοί περιπλανητές παράλληλα με τη βασισμένη σε κείμενο σύνταξη προγραμμάτων παρέχουν λογισμικό «πληκτρολόγιο» στην οθόνη παρόμοιο με αυτό που έχει στην πλάτη του ένας φυσικός περιπλανητής ώστε να μπορεί άμεσα ο μαθητής να εφαρμόσει

αυτά που ήδη γνωρίζει. Για την εισαγωγή στην έννοια του προγράμματος με τη μορφή ακολουθίας εντολών γραμμένων σε κείμενο το περιβάλλον παρέχει τη δυνατότητα αυτόματης εγγραφής των εντολών που πληκτρολογεί ο χρήστης ώστε να είναι εύκολη η απόδοση ονόματος και η επανάληψή τους. Οι βασικές εκπαιδευτικές δραστηριότητες παραμένουν οι ίδιες αλλά είναι εφικτή και η υλοποίηση πιο συνηθισμένων υπολογιστικών προβλημάτων όπως αυτά ενός τυπικού μαθήματος προγραμματισμού.



Σχήμα 2: Διεπαφή χρήστη του λογισμικού Micro Mount, Logo Prime



Σχήμα 3: Διεπαφή χρήστη του λογισμικού Softease, Textease Turtle



Σχήμα 4: Διεπαφή χρήστη του λογισμικού LCSJ Microworlds JR

Το βασικό πλεονέκτημα της άμεσης παροχής μη λεκτικής πληροφορίας ανάδρασης παραμένει. Σε ορισμένες περιπτώσεις τα περιβάλλοντα της κατηγορίας αυτής παρέχουν χρήσιμα εργαλεία, όπως το εικονικό μοιρογνωμόνιο, που διευκολύνουν τον προγραμματισμό και τη σύνδεση με άλλα μαθήματα όπως τα μαθηματικά. Οι λογισμικοί περιπλανητές καλύπτουν το διάστημα μεταξύ των παιδιών που έχουν μάθει

γραφή και ανάγνωση και των παιδιών που δεν γνωρίζουν ή είναι στη διαδικασία της κατάκτησης των βασικών αυτών δεξιοτήτων (ηλικίες 5-7 ετών).

Χαρακτηριστικά παραδείγματα από την κατηγορία αυτή είναι τα συστήματα: Micro Mount Software Logo Prime [<http://www.micromount.co.uk>] (σχήμα 2), το Textease Turtle CT της softease [<http://www.softease.com>] (σχήμα 3) και το MicroWorlds JR [<http://www.microworlds.com>] (σχήμα 4) της γνωστής επιτυχημένης σειράς των μικρόκοσμων της εταιρείας LCSi. Τα περιβάλλοντα αυτά συνοδεύονται συχνά από εκτεταμένη τεκμηρίωση και εκπαιδευτικό υλικό.

Περιβάλλοντα προγραμματισμού LOGO

Πρόκειται για περιβάλλοντα που ακολουθούν βασικά το συνηθισμένο πρότυπο προγραμματισμού σε LOGO. Διαθέτουν επεξεργαστή κειμένου για τη σύνταξη των προγραμμάτων και μια περιοχή-παράθυρο για την προβολή της εκτέλεσης των προγραμμάτων. Τα περιβάλλοντα αυτά συχνά είναι αρκετά λιτά και διαθέσιμα δωρεάν όπως το KTurtle του Cies Breijs που αναπτύχθηκε στο πλαίσιο του έργου KDE Edutainment και διανέμεται με βάση την GPL V2 [<http://edu.kde.org/kturtle/>] ή το σύστημα MSWLogo [<http://www.softronix.com/logo.html>]. Υπάρχουν όμως και εκδόσεις εμπλουτισμένες με χαρακτηριστικά και εργαλεία δημιουργίας και έρευνας που χρησιμοποιούν τον προγραμματισμό ως δεξιότητα - κλειδί για μάθηση. Βασικός αντιπρόσωπος στην κατηγορία αυτή είναι το περιβάλλον microworlds της LCSi στις διάφορες εκδόσεις του, με τρέχουσα την πιο πρόσφατη έκδοση EX [<http://www.microworlds.com/solutions/mwex.html>] (Σχήμα 5). Το συγκεκριμένο περιβάλλον στην έκδοση Pro του 1999 έχει εξελληνιστεί το 2001 στο πλαίσιο του προγράμματος Κίρκη και έχει διανεμηθεί στα Δημοτικά σχολεία (Δαπόντες κ.α. 2003).

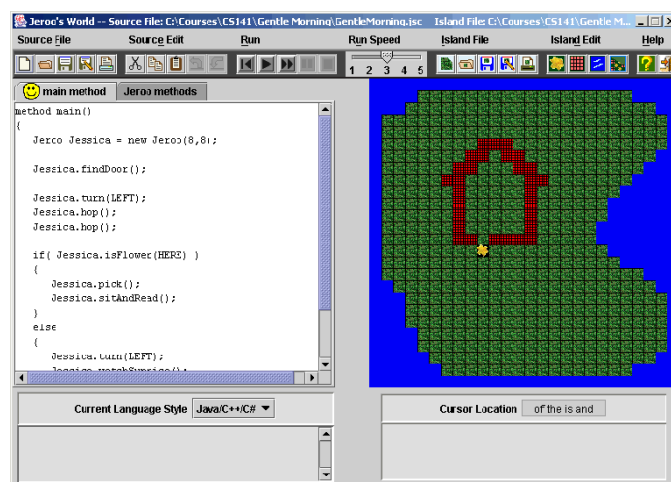
Γενικευμένοι χελωνόκοσμοι για τη διδασκαλία διαφόρων γλωσσών

Η ιδέα της εισαγωγής στον προγραμματισμό μέσω της χελώνας και του κόσμου της, που παρέχει άμεση οπτική ανάδραση και επιτρέπει το σχεδιασμό ποικιλίας δραστηριοτήτων εκτός των συνηθισμένων μαθηματικών προβλημάτων που χρησιμοποιούνται συνήθως στην εισαγωγή στον προγραμματισμό, ενέπνευσε το 1981 τον Richard E. Pattis ώστε να επινοήσει το περιβάλλον «Karel the robot» (Pattis 1995). Το περιβάλλον Karel περιλαμβάνει ένα ρομπότ (γενίκευση της χελώνας), με δυνατότητες ελαφρώς περισσότερες από αυτές της χελώνας, το οποίο “ζει” μέσα σε ένα μικρόκοσμο (δομημένο ως ορθογώνια ρυμοτομημένη περιοχή) και διαμορφούμενο ως ένα βαθμό από εμπόδια και άλλα απλά αντικείμενα (π.χ. ηχητικοί σηματοδότες – beepers). Ο υποψήφιος προγραμματιστής «επικοινωνεί» με τον Karel μέσω κάποιας γλώσσας προγραμματισμού. Στην αρχική έκδοση η γλώσσα αυτή ήταν η Pascal, η οποία την εποχή εκείνη θεωρείτο προσφορότερη για εκπαίδευση στον προγραμματισμό. Ο προγραμματιστής που χρησιμοποιεί το δεδομένο περιβάλλον, όπως και στη LOGO, χρησιμοποιεί τον προγραμματισμό για να μάθει σε μια τεχνητή – νοητή οντότητα να κάνει κάτι ή να λύνει ένα πρόβλημα. Ο προγραμματιστής δηλαδή δεν ελέγχει άμεσα τον ίδιο τον υπολογιστή ή έστω την αφαιρετική του αναπαράσταση που παρέχει το Λειτουργικό Σύστημα αλλά ένα φανταστικό αντικείμενο και τον κόσμο του.

Οι βασικές υποθέσεις του Pattis για την παιδαγωγική αξία του Karel εκφράζονται στο επόμενο απόσπασμα της εισαγωγής του βιβλίου του (Patis 1995): *“The first few weeks of a programming course are crucial to the students' perception of the subject; it is during this period that they briefly glimpse the aesthetics of the discipline and are most receptive to new ideas. By starting with Karel the Robot, it will be easy for students to absorb a large number of useful, important, and sophisticated concepts quickly”*. Σε

ελεύθερη απόδοση: «Οι πρώτες εβδομάδες ενός μαθήματος προγραμματισμού είναι κρίσιμες για τη διαμόρφωση των αντιλήψεων των φοιτητών/τριών για το αντικείμενο, στο διάστημα αυτό παίρνουν μια συνοπτική άποψη της αισθητικής του γνωστικού αντικειμένου και είναι περισσότερο δεκτικοί σε νέες ιδέες. Ξεκινώντας με το Karel, θα είναι εύκολο για τους φοιτητές να αφομοιώσουν γρήγορα, ένα μεγάλο πλήθος χρήσιμων, σημαντικών και σύνθετων εννοιών».

Ως φυσική εξέλιξη και προσαρμογή του Karel εμφανίσθηκαν παρόμοια περιβάλλοντα για να καλύψουν τις ανάγκες νέων υποδειγμάτων προγραμματισμού που διαδέχθηκαν τον δομημένο διαδικαστικό (imperative) προγραμματισμό που πρεσβεύει η PASCAL. Ειδικότερα, με χρονολογική σειρά, εμφανίσθηκαν τα περιβάλλοντα Karel++ (Bergin et al. 1997) (κατά αναλογία με τα ονόματα των γλωσσών C και C++), JKarelRobot (Buck and Stucki 2001), KarelJ (Bergin et al. 2002) και το JEROO (Sanders and Dorn 2003).



Σχήμα 6: Διεπαφή χρήστη του λογισμικού JEROO

Τα περιβάλλοντα αυτά διαθέτουν τον Karel αλλά η επικοινωνία με αυτόν γίνεται με τη βοήθεια σύγχρονων αντικειμενοστρεφών γλωσσών προγραμματισμού όπως η C++, η JAVA και η C#. Με τα προσαρμοσμένα περιβάλλοντα υποστηρίζεται η διδασκαλία εννοιών όπως η κλάση αντικειμένων, το στιγμιότυπο, η μέθοδος, η ανταλλαγή

μηνυμάτων, η ιεραρχία κλάσεων, η κληρονομικότητα, ο πολυμορφισμός, κλπ. Το περιβάλλον Karel έχει απασχολήσει ερευνητές και στη χώρα μας (Ξυνόγαλος κ.α. 2005).

Όπως φαίνεται από τα προηγούμενα, τα περιβάλλοντα τύπου Karel απευθύνονται κυρίως στην τριτοβάθμια εκπαίδευση. Για το επίπεδο της δευτεροβάθμιας εκπαίδευσης τουλάχιστον ένας γενικευμένος χελωνόκοσμος είναι υπόψη των συγγραφέων, και πρόκειται για το περιβάλλον CeeBot [<http://www.ceebot.com/>].



Σχήμα 7: Διεπαφή χρήστη του λογισμικού CeeBot-3

Το CeeBot έχει διάφορες εκδόσεις όπως τις CeeBot-Teen και CeeBot-3 για ηλικίες 10-15 ετών και τις CeeBot-A και CeeBot-4 για ηλικίες από 15 ετών και άνω. Η βασική ιδέα είναι και εδώ ο προγραμματισμός λογισμικών ρομπότ-μινιατούρων για την εκτέλεση αποστολών πάνω σε ένα γραφείο, ή σε ένα υπόγειο, κλπ. Οι διάφορες εκδόσεις υποστηρίζουν διάφορες γλώσσες όπως Java και C#. Όσο αφορά στην εκπαιδευτική τεκμηρίωση, σε κάθε προϊόν υπάρχει ένα σύνολο από προβλήματα για επίλυση, τα οποία είναι οργανωμένα σε ενότητες. Μεταξύ άλλων περιλαμβάνονται δραστηριότητες όπως αγώνες ταχύτητας των ρομπότ, ποδόσφαιρο και απόδραση από λαβύρινθο. Οι εκδόσεις που απευθύνονται σε ευρύτερες ηλικίες περιγράφονται ως ένα

είδος παιχνιδιού σε υπολογιστή. Το CeeBot επιτρέπει την εισαγωγή εννοιών όπως μεταβλητή, επιλογή, επανάληψη, πίνακες, συναρτήσεις-υποπρογράμματα, κλάση, κ.α.

Περιβάλλοντα εκπαιδευτικής ρομποτικής

Οι χελώνες εδάφους, οι διάφορες LOGO, ο Karel και τα CeeBot έχουν ως βασικό κοινό στοιχείο την ύπαρξη ενός ρομπότ (ηλεκτρομηχανική διάταξη γενικότερα), προκαθορισμένων περιορισμένων δυνατοτήτων που επικοινωνεί με τον χρήστη με μια γλώσσα προγραμματισμού. Με τη φυσική υλοποίηση της γενικευμένης λογισμικής χελώνας σχηματίζεται η παρούσα κατηγορία περιβαλλόντων εκπαίδευσης στον προγραμματισμό στην οποία ανήκουν τα συστήματα εκπαιδευτικής ρομποτικής.



Σχήμα 8: Το εκπαιδευτικό ρομπότ IntelliBrain-Bot της RidgeSoft



Σχήμα 9: Ρομπότ κατασκευασμένο με το σύστημα LEGO Mindstorms

Τα περιβάλλοντα αυτά περιλαμβάνουν συνήθως κάποιο προγραμματιζόμενο μικροελεγκτή-μικροεπεξεργαστή συνδεδεμένο με εισόδους και εξόδους σε πλακέτα τυπωμένου κυκλώματος, ο οποίος είναι δυνατό να προγραμματιστεί με κάποιο περιβάλλον προγραμματισμού σε υπολογιστή. Το πρόγραμμα μεταφέρεται από τον υπολογιστή στον μικροελεγκτή και μπορεί να εκτελείται αυτόνομα μέχρι την εγκατάσταση νέου. Στο βασικό κύκλωμα του μικροεπεξεργαστή συνδέονται συσκευές όπως κινητήρες, λαμπτήρες, διακόπτες, κλπ. καθώς και αισθητήρες (θερμοκρασίας, φως, κλπ.) για τη σύνθεση συστημάτων αυτομάτου ελέγχου που έχουν συνήθως κινούμενα μέρη ή συνιστούν και ολόκληρα αυτοκινούμενα. Η κατασκευή ενός ρομπότ

με τον τρόπο αυτό είναι μια πλούσια, διαθεματική δραστηριότητα που περιλαμβάνει προβλήματα Φυσικής, Μαθηματικών, Ηλεκτρονικής, Μηχανολογίας και Πληροφορικής.

Τα συστήματα αυτά δεν είναι τόσο διαδεδομένα λόγω του κόστους τους αλλά και των δυσκολιών διαχείρισης που έχουν ως συλλογές πολλών και σχετικά ευαίσθητων συσκευών. Υπάρχουν αρκετοί ερασιτέχνες που διατηρούν μια ομάδα-πυρήνα με αρκετό ενδιαφέρον. Συχνά οι ομάδες αυτές αποτελούνται από φοιτητές οι οποίοι συμμετέχουν και σε σχετικούς διαγωνισμούς [π.χ. Northeast Indiana Robot Games Rules 2006, <http://www.sciencecentral.org/2006rules.htm>].

Χαρακτηριστικά παραδείγματα στην κατηγορία αυτή αποτελούν τα επόμενα περιβάλλοντα:

- **LEGO Mindstorms for Schools:** πρόκειται για προϊόν αποτέλεσμα της συνεργασίας του Papert με την εταιρεία παιχνιδιών LEGO (LEGO Mindstorms 2005). Ο μικροεπεξεργαστής έχει ενσωματωθεί σε ένα τουβλάκι Lego και λέγεται LEGO RCX. Διαθέτει τρεις εισόδους και τρεις εξόδους στις οποίες συνδέονται κινητήρες, λαμπτήρες και διάφοροι αισθητήρες που επίσης έχουν τη μορφή τούβλων της LEGO. Με τη βοήθεια διαφόρων δομικών στοιχείων (μεταξύ των οποίων γρανάζια, ιμάντες και άξονες) είναι δυνατό να κατασκευαστούν πολύπλοκες ρομποτικές διατάξεις, ο προγραμματισμός των οποίων αποτελεί αυθεντική πρόκληση. Υπάρχει ακόμα η δυνατότητα επικοινωνίας των RCX μεταξύ τους επεκτείνοντας ακόμα περισσότερο τη γκάμα και το ενδιαφέρον των εφαρμογών.

Το RCX είναι δυνατό να προγραμματιστεί με διάφορους τρόπους. Η LEGO προτείνει το περιβάλλον ROBO-LAB (Τσοβόλας και Κόμης 2005) ενώ υπάρχουν και διάφορα άλλα βασισμένα στη JAVA, τη C (π.χ. NQC), κ.α. Όπως έχει άλλωστε αναφερθεί

παραπάνω, υπάρχει ειδική έκδοση του microworlds EX της LCSi για τον προγραμματισμό του RCX. Στην Ελλάδα το συγκεκριμένο προϊόν αντιπροσωπεύεται από την εταιρεία «Διερευνητική Μάθηση» [<http://www.why.gr>].

- **Cricket & GoGo:** Παρόμοια περιβάλλοντα αναπτύσσονται στο MIT με ονόματα Cricket (MIT Cricket 2005), και GoGo (MIT GoGo 2005).

- **Ερασιτεχνικά ρομποτικά περιβάλλοντα από εμπορικές εταιρείες:** προϊόντα εκπαιδευτικής ρομποτικής προτείνονται και από το χώρο των εμπορικών επιχειρήσεων όπως το IntelliBrain-Bot της RidgeSoft (IntelliBrain™-Bot 2005), ενώ διατίθεται και μια μεγάλη ποικιλία μικροελεγκτών, αισθητήρων, και συσκευών βιομηχανικών προδιαγραφών της Parallax (Parallax 2005) που συνοδεύονται από εκτεταμένη τεκμηρίωση και ιδέες για εκπαιδευτικές εφαρμογές.

Τύποι και χαρακτηριστικά μαθησιακών δραστηριοτήτων εκπαιδευτικής ρομποτικής

Οι δραστηριότητες που προτείνονται στο πλαίσιο της εκπαιδευτικής ρομποτικής παρουσιάζουν μεγάλη ποικιλία και περιλαμβάνουν από την ‘έξοδο από λαβύρινθο’ μέχρι τη δημιουργία ‘ρομπότ κατοικίδιων’ που παίζουν επιτραπέζια παιχνίδια (Ferrari et al. 2001). Στο διαδίκτυο υπάρχει πληθώρα ιδεών για κατασκευές, παράλληλα με την ύπαρξη κοινοτήτων από ενθουσιώδεις ερασιτέχνες που μπορούν να κρατήσουν αμείωτο το ενδιαφέρον για τεχνολογίες αιχμής.

Το εκπαιδευτικό ενδιαφέρον της ρομποτικής εδράζεται και στις επόμενες ιδιότητες:

- εισαγωγή στον ενσωματωμένο (embedded) προγραμματισμό, η σημασία του οποίου είναι διαρκώς αυξανόμενη, λόγω π.χ. των κινητών τηλεφώνων, των επιταλάμιων υπολογιστών, των έξυπνων οικιακών συσκευών, κλπ.

- παροχή ενός περιβάλλοντος για εισαγωγή στη γλώσσα Assembly,

- παροχή περιβάλλοντος για εισαγωγή σε αντικείμενα όπως η επεξεργασία σήματος και ο αυτόματος έλεγχος,
- παροχή περιβάλλοντος για εισαγωγή σε υποδείγματα προγραμματισμού όπως ο ταυτόχρονος, παράλληλος, κατανεμημένος και λογικός,
- εισαγωγή σε μεθόδους τεχνητής νοημοσύνης και προγραμματισμού παιχνιδιών.

Τα περιβάλλοντα εκπαιδευτικής ρομποτικής που είναι κατάλληλα σχεδιασμένα επιτρέπουν την ενασχόληση παιδιών από 7 ετών. Το κόστος τους μπορεί να αιτιολογηθεί από την πληθώρα των εφαρμογών τους και το ενδιαφέρον που προκαλούν σε πολλά παιδιά, ενώ τα τελευταία χρόνια η παρουσία τους στα σχολεία αυξάνει εξαιτίας προγραμμάτων όπως το ΔΑΙΔΑΛΟΣ του ΕΙΝ. Τα περιβάλλοντα εκπαιδευτικής ρομποτικής τοποθετούν τον προγραμματισμό στο ευρύτερο πεδίο εφαρμογής του και παρέχουν ευκαιρίες για εμπλοκή των μαθητών σε αυθεντικά και διαθεματικά προβλήματα.

Περιβάλλοντα αυτομάτου ελέγχου

Μια σειρά περιβαλλόντων αφορά σε συστήματα που δίνουν έμφαση στην έννοια του αυτόματου ελέγχου. Η έννοια αυτή εμπεριέχεται βέβαια στα συστήματα της εκπαιδευτικής ρομποτικής, όμως τα συστήματα της κατηγορίας αυτής αποφεύγουν συνήθως τα πολύπλοκα κατασκευαστικά προβλήματα των ρομπότ. Στις δραστηριότητες των συστημάτων αυτομάτου ελέγχου οι μαθητές εστιάζουν στη λογική-προγραμματισμό ενός συστήματος ελέγχου, δηλαδή ενός γενικευμένου συστήματος που αποτελείται από ένα σύνολο εισόδων και εξόδων. Οι μαθητές δεν προγραμματίζουν τον υπολογιστή, αλλά προδιαγράφουν ένα σύστημα ελέγχου ενός άλλου συστήματος. Πιο

διαδεδομένο σύστημα αναπαράστασης στα περιβάλλοντα της κατηγορίας φέρεται να είναι το διάγραμμα ροής. Τα συστήματα που ελέγχονται μπορούν να χωριστούν σε δύο κατηγορίες.

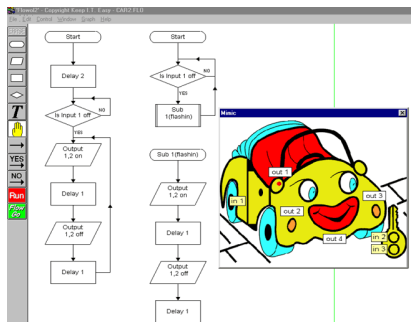
- **Πραγματικά:** Ο έλεγχος τους υλοποιείται μέσω κατάλληλων διατάξεων επικοινωνίας σημάτων με τον υπολογιστή.

- **Λογισμικά μοντέλα:** Οι λογισμικές αναπαραστάσεις των προς έλεγχο συστημάτων ονομάζονται συνήθως ‘mimics’ και μοιάζουν με τους μικρόκοσμους. Για παράδειγμα, αντί οι μαθητές να ελέγχουν ένα πραγματικό θερμοκήπιο, προκειμένου να πετύχουν ιδανικές συνθήκες ανάπτυξης των φυτών, μπορούν να ελέγξουν μια εικονική αναπαράστασή του στον υπολογιστή.

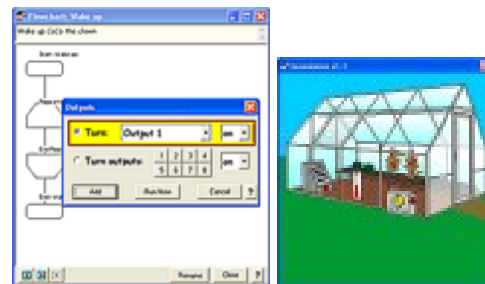
Μερικά δείγματα περιβαλλόντων της κατηγορίας αυτής παρουσιάζονται στις εικόνες των Σχημάτων 10, 11 και 12. Τα προϊόντα αυτά συνοδεύονται από μικρόκοσμοις-mimics, περιβάλλοντα ανάπτυξης μικρόκοσμων, καθώς και από εκπαιδευτική τεκμηρίωση με τη μορφή δραστηριοτήτων για τους μαθητές. Σε ορισμένες περιπτώσεις τα προϊόντα αυτά συνεργάζονται με άλλα προϊόντα για τον έλεγχο σημάτων εξωτερικών του υπολογιστή προκειμένου να υλοποιηθούν πιο αυθεντικές εξερευνήσεις (π.χ. το FlowGo της Data Harvest). Ορισμένα από τα περιβάλλοντα συνιστούν εκπαιδευτικές εκδόσεις επαγγελματικών λύσεων συστημάτων ‘Data Acquisition’ & ‘Automatic Control’ (DAQ/AC) που χρησιμοποιούνται στους βιομηχανικούς αυτοματισμούς και τα συστήματα τηλεματικής .

Τα περιβάλλοντα προγραμματισμού αυτομάτου ελέγχου είναι ιδιαίτερα ενδιαφέροντα για εφαρμογή του προγραμματισμού σε διαθεματικές δραστηριότητες. Επιπλέον δείχνουν στους μαθητές τις έννοιες που σχετίζονται με τον προγραμματισμό, ανεξάρτητα από το στενό πλαίσιο του προγραμματισμού του υπολογιστή. Η

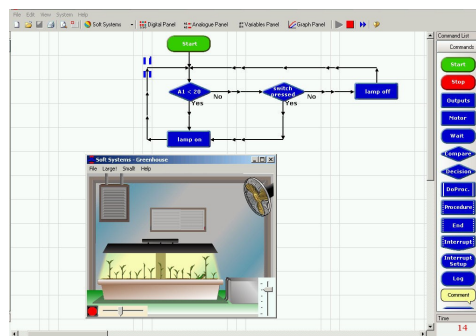
υποστήριξη οπτικών αναπαραστάσεων καθιστά τα περιβάλλοντα αυτά αξιοποιήσιμα από την ηλικία των 10 ετών, σύμφωνα με τους κατασκευαστές τους.



Σχήμα 10: Διεπαφή χρήστη του λογισμικού Flowol2 της KeepIT Easy



Σχήμα 11: Διεπαφή χρήστη του λογισμικού CoCo της MatrixMultimedia



Σχήμα 12: Διεπαφή χρήστη του λογισμικού Logicator της Economatics

Περιβάλλοντα προγραμματισμού παιχνιδιών

Η πλειονότητα των παιδιών έλκεται ισχυρά από τα παιχνίδια σε υπολογιστή ή τα παρόμοιά τους στις παιχνιδομηχανές. Τα παιδιά συχνά καταναλώνουν μεγάλο μέρος του χρόνου τους ασχολούμενα με τέτοια παιχνίδια. Είναι εύλογο να σκεφτεί κανείς ότι θα μπορούσε να δελεάσει τους νεαρούς μαθητές να ασχοληθούν με τον προγραμματισμό προκειμένου να αναπτύξουν τα δικά τους παιχνίδια. Η κατασκευή των παιχνιδιών απαιτεί την εξάσκηση ανώτερων μορφών σκέψης αλλά και την έκφραση δημιουργικότητας διαφόρων τύπων για την ανάπτυξη του σεναρίου, την κατασκευή των γραφικών και τη δημιουργία μουσικής επένδυσης (Kafai 2001, Habgood 2005).

Η ανάπτυξη των παιχνιδιών στην πραγματικότητα ξεφεύγει από τα στενά όρια του προγραμματισμού και αποτελεί ένα ξεχωριστό είδος που καθορίζει και μια διακριτή βιομηχανία. Για την περίπτωση των μαθητών υπάρχουν μερικά περιβάλλοντα που έχουν ενδιαφέρον, έχουν αναπτυχθεί για εκπαιδευτικούς σκοπούς και παρουσιάζονται σύντομα στη συνέχεια.

- **GameMaker** (GameMaker 2005): Το περιβάλλον αυτό αναπτύχθηκε από τον Mark Overmars στο πανεπιστήμιο Utrecht στην Ολλανδία, για τη διδασκαλία ενός μαθήματος για το σχεδιασμό παιχνιδιών σε υπολογιστή. Το GameMaker παρέχεται δωρεάν για εκπαιδευτικούς σκοπούς και χρησιμοποιεί ως βασικό υπόδειγμα προγραμματισμού τον αντικειμενοστρεφή, καθοδηγούμενο από γεγονότα, εικονικό προγραμματισμό με διαφάνεια στον ταυτόχρονο προγραμματισμό. Ο προγραμματιστής ορίζει ένα σύνολο από αντικείμενα με συμπεριφορά που προσδιορίζεται ως ένα σύνολο αντιδράσεων σε γεγονότα. Κατόπιν τοποθετεί όσα αντικείμενα θέλει σε δωμάτιο (θεατρική σκηνή) όπου αυτά αρχίζουν να αλληλεπιδρούν μεταξύ τους, αλλά και με τον χρήστη του υπολογιστή. Η εξέλιξη του παιχνιδιού καθορίζεται από την αλληλεπίδραση αυτή και συχνά δεν είναι προβλέψιμη εκ των προτέρων.

Η απλότητα και η εκφραστική δύναμη του GameMaker θα ξαφνιάσουν ευχάριστα ακόμα και έμπειρους προγραμματιστές. Οι δυνατότητες δραστηριοτήτων είναι ποικίλες και αγγίζουν παιδιά από την ηλικία των 10 ετών, ενώ υπάρχουν δικτυακοί τόποι αφιερωμένοι σε εκπαιδευτικές εφαρμογές του GameMaker στην υποχρεωτική και δευτεροβάθμια εκπαίδευση.

- **AgentSheets** (AgentSheets 2005): Πρόκειται για περιβάλλον ανάπτυξης προσομοιώσεων, παιχνιδιών και άλλων εφαρμογών που βασίζεται στο υπόδειγμα του λογικού προγραμματισμού με κανόνες. Η σύνταξη των κανόνων γίνεται με τη βοήθεια

εικονικού συντάκτη ιδιαίτερα εύχρηστου και για μικρές ηλικίες. Άλλο ενδιαφέρον χαρακτηριστικό του AgentSheets είναι η αυτόματη παραγωγή μικροεφαρμογής JAVA για κάθε πρόγραμμα που παράγει κανείς ώστε να είναι διαθέσιμο άμεσα μέσω του διαδικτύου. Τελευταία το περιβάλλον έχει μεταφραστεί στα Ελληνικά ανοίγοντας έτσι το δρόμο για εκπαιδευτικές εφαρμογές και στη χώρα μας (Σπυρόπουλος και Δαγδιλέλης 2005).

- **StageCast** (StageCast 2005): Της εταιρείας StageCast Software Inc, μοιάζει με το AgentSheets αλλά είναι προσανατολισμένο στην ανάπτυξη παιχνιδιών από παιδιά. Το υπόδειγμα προγραμματισμού είναι και πάλι ο λογικός προγραμματισμός με κανόνες ενώ η εκπαιδευτική του τεκμηρίωση είναι εκτεταμένη.

Περιβάλλοντα εκπαίδευσης σε προγραμματισμό χαμηλού επιπέδου

Στην κατηγορία αυτή εντάσσονται περιβάλλοντα που αφορούν τη διδασκαλία προγραμματισμού χαμηλού επιπέδου ενός ιδεατού-απλουστευμένου υπολογιστή. Ο μαθητής μπορεί να βλέπει το εσωτερικό του μικροεπεξεργαστή, τα περιεχόμενα των καταχωρητών, την αριθμητική και λογική μονάδα και τη μνήμη. Στα περιβάλλοντα αυτά η εκτέλεση των προγραμμάτων συνοδεύεται από την αναπαράσταση της ροής της πληροφορίας μέσα στον υπολογιστή με μεγάλη λεπτομέρεια. Περιβάλλοντα αυτού του τύπου είναι διδακτικά χρήσιμα για μαθήματα προγραμματισμού σε χαμηλό επίπεδο (Assembly, γλώσσα μηχανής) και μαθήματα αρχιτεκτονικής Η/Υ. Χαρακτηριστικό παράδειγμα συστήματος στην κατηγορία αυτή αποτελούν τα Beborputer της Maxfield & Montrose Interactive και το APOO των Rogério Reis, Nelma Moreira του Πανεπιστημίου του Πόρτο (Reis and Moreira 2001).

Περιβάλλοντα επαγγελματικού προγραμματισμού

Ο πιο συνηθισμένος τρόπος εισαγωγής στον προγραμματισμό είναι η χρήση ενός επαγγελματικού περιβάλλοντος ανάπτυξης εφαρμογών για κάποια από τις γλώσσες όπως Pascal, C, C++, JAVA, C#, κλπ. Τα περιβάλλοντα αυτά, αν και ρεαλιστικά, δεν είναι τα καταλληλότερα για εισαγωγή στον προγραμματισμό, λόγω της πολυπλοκότητάς τους. Οι εξελλητισμένες απομιμήσεις διαδικαστικών γλωσσών που χρησιμοποιούνται σε μαθήματα του Γυμνασίου και του Λυκείου δε διαφέρουν ουσιαστικά από ένα τέτοιο περιβάλλον, που αν και αντιμετωπίζουν το πρόβλημα της χρήσης της Αγγλικής γλώσσας, στην πραγματικότητα δεν καινοτομούν όσον αφορά στη βοήθεια των νέων προγραμματιστών.

Τα σύγχρονα επαγγελματικά περιβάλλοντα προγραμματισμού αναμειγνύουν πολλά υποδείγματα προγραμματισμού ώστε να καλύψουν τις ανάγκες της ανάπτυξης των σύγχρονων πληροφοριακών συστημάτων. Η ανάμειξη των υποδειγμάτων δυσκολεύει ακόμα περισσότερο την εισαγωγή των μαθητών στον προγραμματισμό, επειδή για να εκτελέσουν μια απλή άσκηση θα πρέπει να δεχθούν ως δεδομένες πολλές παραδοχές και διαδικασίες. Πιο συγκεκριμένα, μπορεί να αναφερθεί η ιεραρχία προκαθορισμένων κλάσεων αντικειμένων που πρέπει κανείς να γνωρίσει για να μπορεί να προγραμματίσει στη JAVA ή τη Visual Basic.

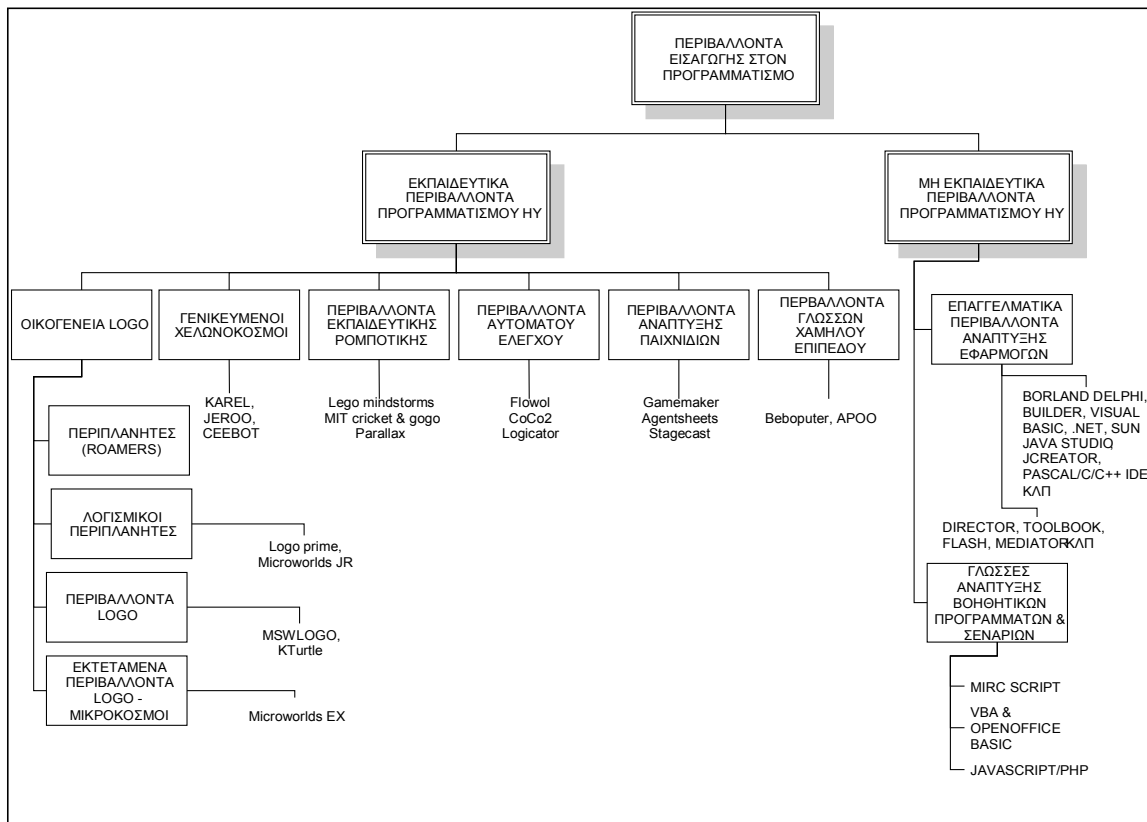
Σε κάποιες περιπτώσεις παρουσιάζει διδακτικό ενδιαφέρον η χρήση ειδικών περιβαλλόντων προγραμματισμού, όπως η δημιουργία βοηθητικών προγραμμάτων με τη γλώσσα συγγραφής σεναρίων σε περιβάλλοντα όπως το MIRC ή γλώσσες τύπου BASIC που περιέχουν οι διάφορες 'σουίτες' αυτοματισμού γραφείου (π.χ η VBA και η αντίστοιχη του Openoffice). Το ενδιαφέρον των περιβαλλόντων αυτών εδράζεται στην παροχή αυθεντικών προβλημάτων για προγραμματισμό που έχουν νόημα για τους

μαθητές. Ενδιαφέρον επίσης παρουσιάζουν περιβάλλοντα όπως το Mediator της Matchware που ενσωματώνει ευκολίες οπτικού προγραμματισμού. Σε κάθε περίπτωση πάντως τα περιβάλλοντα αυτά προσφέρονται μόνο για μαθητές των τελευταίων τάξεων Λυκείου.

Κοινό χαρακτηριστικό των σύγχρονων περιβαλλόντων ανάπτυξης εφαρμογών εκτός από την ανάμειξη διαφόρων υποδειγμάτων προγραμματισμού είναι και η παροχή μιας αφηρημένης απλοποιημένης εικόνας για το περιβάλλον μέσα στο οποίο εκτελούνται τα παραγόμενα προγράμματα. Ο προγραμματιστής με άλλα λόγια δεν οφείλει πλέον να προγραμματίσει άμεσα τη μηχανή αλλά ένα υπολογιστικό περιβάλλον, η αντίληψη του οποίου καθορίζεται σε μεγάλο βαθμό από το λειτουργικό σύστημα που θα χρησιμοποιηθεί αλλά και από τις προκαθορισμένες κλάσεις αντικειμένων (βιβλιοθήκες). Οι συνιστώσες αυτές είναι λογικό να επηρεάζουν τις αναπαραστάσεις που αναπτύσσουν οι μαθητές για τον προγραμματισμό και επομένως πρέπει να ελεγχθούν τα κριτήρια επιλογής τους.

Σύνοψη - συζήτηση

Στην ενότητα αυτή θα παρουσιάσουμε συνοπτικά τις κατηγορίες που αναλύθηκαν στα προηγούμενα, προβάλλοντάς τις παράλληλα στις διαστάσεις που ορίζουν οι άξονες που περιγράφηκαν στην εισαγωγή. Η εργασία ολοκληρώνεται με προτεινόμενες ερευνητικές κατευθύνσεις για την εκπαιδευτική αξιοποίηση των σύγχρονων προγραμματιστικών περιβαλλόντων.

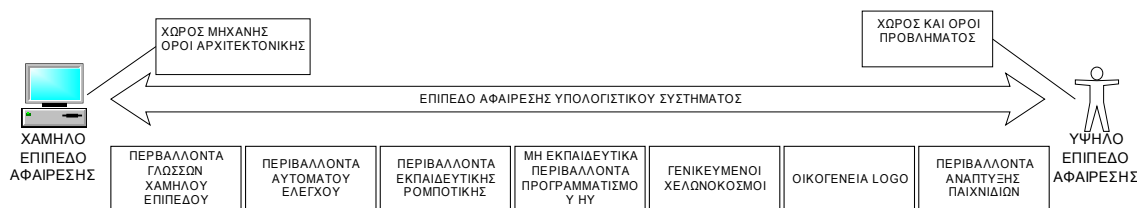


Σχήμα 13: Κατηγοριοποίηση των περιβαλλόντων εισαγωγής στον προγραμματισμό

Στο Σχήμα 13 εμφανίζεται διάγραμμα με τις κατηγορίες και τους χαρακτηριστικούς αντιπροσώπους για κάθε μια από αυτές. Το διάγραμμα εστιάζει κυρίως στα υπάρχοντα εκπαιδευτικά περιβάλλοντα και καθιστά φανερή την αυξημένη αντιπροσώπευση της οικογένειας LOGO. Τα περιβάλλοντα LOGO στις διάφορες εκφάνσεις τους μαζί με τα συγγενικά τους αποτελούν την πλειοψηφία των εκπαιδευτικών περιβαλλόντων.

Προβολή στον άξονα (Α) της διαβάθμισης αφαίρεσης του υπολογιστικού συστήματος: Τα εκπαιδευτικά περιβάλλοντα προγραμματισμού διαφοροποιούνται

ανάλογα με το κατά πόσο απαιτούν από τον προγραμματιστή να σκέφτεται με όρους μηχανής ή με όρους από το χώρο του προβλήματος. Στις περιπτώσεις που ενδιαφερόμαστε να αξιοποιήσουμε τον προγραμματισμό υπολογιστών σε δραστηριότητες επίλυσης προβλημάτων, μας ενδιαφέρουν τα συστήματα με υψηλό επίπεδο αφαίρεσης από την υπολογιστική μηχανή. Αντίθετα υπάρχουν περιπτώσεις που θέλουμε να εστιάσουμε σε λεπτομέρειες της υλοποίησης του προγραμματισμού, οπότε μας ενδιαφέρει διαφάνεια προς τη μηχανή. Στην ακραία περίπτωση χαμηλής αφαίρεσης, εντοπίζουμε τα περιβάλλοντα προγραμματισμού σε συμβολική γλώσσα ενώ στα υψηλά επίπεδα αφαίρεσης εντοπίζουμε τα περιβάλλοντα ανάπτυξης παιχνιδιών και τα περιβάλλοντα LOGO. Δημιουργείται έτσι η κατάταξη του Σχήματος 14.



Σχήμα 14: Κατάταξη των περιβαλλόντων προγραμματισμού κατά επίπεδο αφαίρεσης του υπολογιστικού συστήματος

Προβολή στον ηλικιακό άξονα (B): Στον Πίνακα 1 εμφανίζονται τα περιβάλλοντα εκπαίδευσης στον προγραμματισμό και οι αντίστοιχες ηλικίες που ενδείκνυται να εφαρμοστούν. Ο πίνακας 1 εμφανίζει την διάταξη των κατηγοριών στον άξονα (B). Η ηλικιακή αντιστοίχιση είναι ενδεικτική και βασίζεται στις προτάσεις των κατασκευαστών και στις εκτιμήσεις των συγγραφέων. Από τον πίνακα φαίνεται ότι η οικογένεια της LOGO καλύπτει όλες τις ηλικίες της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης.

Άλλες διαστάσεις που ενδιαφέρουν κατά την επιλογή ενός περιβάλλοντος εκπαίδευσης στον προγραμματισμό είναι το υπόδειγμα του προγραμματισμού που υποστηρίζει, τη

γλώσσα της οποίας το συντακτικό χρησιμοποιεί, καθώς και το υποτιθέμενο σχήμα του προγραμματισμού ως διαδικασία. Οι προβολές στους αντίστοιχους άξονες (Γ-Ε) αναλύονται στα επόμενα και εμφανίζονται συνοπτικά στον Πίνακα 2.

Πίνακας 1: Ηλικίες εφαρμογής των περιβαλλόντων εισαγωγής στον προγραμματισμό

	4-6	6-8	8-12	12-15	15-17	17+
ΠΕΡΙΗΓΗΤΕΣ	√					
Α. ΠΕΡΙΗΓΗΤΕΣ		√				
ΑΠΛΗ LOGO			√	√		
ΜΙΚΡΟΚΟΣΜΟΙ			√	√	√	
Γ. ΧΕΛΩΝΟΚΟΣΜΟΙ				√	√	√
Ε. ΡΟΜΠΟΤΙΚΗ			√	√	√	√
Α. ΕΛΕΓΧΟΣ				√	√	
Α. ΠΑΙΧΝΙΔΙΩΝ			√	√	√	√
ΧΑΜΗΛΟΥ ΕΠ/ΔΟΥ					√	√
ΜΗ ΕΚΠΑΙΔΕΥΤΙΚΑ					√	√

Πίνακας 2: Υποδείγματα και συντακτικό για κάθε περιβάλλον προγραμματισμού

	ΑΦΗΡΗΜΕΝΟ ΣΧΗΜΑ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	ΥΠΟΔΕΙΓΜΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ										ΣΥΝΤΑΚΤΙΚΟ	
		ΔΙΑΔΙΚΑΣΤΙΚΟΣ	ΟΠΤΙΚΟΣ	ΚΑΘΟΔΗΓΟΥΜΕΝΟΣ ΑΠΟ ΤΑ ΓΕΓΟΝΟΤΑ	ΑΝΤΙΚΕΙΜΕΝΟΣ ΤΡΕΦΗΣ	ΠΑΡΑΛΛΗΛΟΣ	ΤΑΥΤΟΧΡΟΝΟΣ	ΚΑΤΑΝΕΜΗΜΕΝΟΣ	ΛΟΓΙΚΟΣ	ΕΝΣΩΜΑΤΩΜΕΝΟΣ	ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ		
ΠΕΡΙΗΓΗΤΕΣ	ΚΑΝΩ ΤΟΝ ΠΕΡΙΗΓΗΤΗ ΝΑ ΚΑΝΕΙ ΚΑΤΙ	√											
ΛΟΓΙΣΜΙΚΟΙ ΠΕΡΙΗΓΗΤΕΣ	ΚΑΝΩ ΜΙΑ ΟΝΤΟΤΗΤΑ ΝΑ ΚΑΝΕΙ ΚΑΤΙ	√											LOGO
ΑΠΛΗ LOGO	ΚΑΝΩ ΜΙΑ ΟΝΤΟΤΗΤΑ ΝΑ ΚΑΝΕΙ ΚΑΤΙ	√											LOGO
ΜΙΚΡΟΚΟΣΜΟΙ	ΚΑΝΩ ΟΝΤΟΤΗΤΕΣ ΝΑ ΑΛΛΗΛΕΠΙΔΡΟΥΝ ΓΙΑ ΝΑ ΚΑΝΟΥΝ ΚΑΤΙ	√	√	√	√			√					LOGO
ΓΕΝΙΚΕΥΜΕΝΟΙ ΧΕΛΩΝΟΚΟΣΜΟΙ	ΚΑΝΩ ΟΝΤΟΤΗΤΕΣ ΝΑ ΑΛΛΗΛΕΠΙΔΡΟΥΝ ΓΙΑ ΝΑ ΚΑΝΟΥΝ ΚΑΤΙ	√	√	√	√								PASCAL, C++, JAVA, C#
ΕΚΠΑΙΔΕΥΤΙΚΗ ΡΟΜΠΟΤΙΚΗ	ΚΑΤΑΣΚΕΥΑΖΩ ΣΥΣΚΕΥΕΣ ΠΟΥ ΑΛΛΗΛΕΠΙΔΡΟΥΝ ΓΙΑ ΝΑ ΚΑΝΟΥΝ ΚΑΤΙ	√		√				√	√		√	√	ASSEMBLY, LOGO, C, JAVA
ΑΥΤΟΜΑΤΟΣ ΕΛΕΓΧΟΣ	ΕΛΕΓΧΩ ΕΝΑ ΣΥΣΤΗΜΑ	√		√				√			√		FLOW CHART
ΑΝΑΠΤΥΞΗ ΠΑΙΧΝΙΔΙΩΝ	ΟΡΙΖΩ ΟΝΤΟΤΗΤΕΣ ΠΟΥ ΑΛΛΗΛΕΠΙΔΡΟΥΝ ΓΙΑ ΝΑ ΓΙΝΕΙ ΚΑΤΙ	√	√	√	√			√		√		√	RULES, C
ΧΑΜΗΛΟΥ ΕΠ/ΔΟΥ	ΠΡΟΓΡΑΜΜΑΤΙΖΩ ΤΟΝ Η/Υ ΝΑ ΚΑΝΕΙ ΚΑΤΙ	√											ASSEMBLY
ΜΗ ΕΚΠΑΙΔΕΥΤΙΚΑ	ΕΛΕΓΧΩ ΤΟΝ Η/Υ Η ΛΟΓΙΣΜΙΚΟ ΠΕΡΙΒΑΛΛΟΝ	√	√	√	√	√	√	√	√	√	√	√	ΟΛΑ

Προβολή στον άξονα (Γ) των υποστηριζόμενων υποδειγμάτων προγραμματισμού:

Στα υποστηριζόμενα υποδείγματα του προγραμματισμού περιλαμβάνονται ο διαδικαστικός, ο οπτικός, ο καθοδηγούμενος από τα γεγονότα, ο αντικειμενοστρεφής, ο παράλληλος, ο ταυτόχρονος, ο κατανεμημένος, ο ενσωματωμένος, ο λογικός προγραμματισμός και οι τεχνικές της τεχνητής νοημοσύνης. Το προγραμματιστικό υπόδειγμα είναι σημαντικό και γιατί καθορίζει την εκφραστική δύναμη του περιβάλλοντος και τον χώρο προβλημάτων που μπορεί να αξιοποιηθεί για το σχεδιασμό δραστηριοτήτων. Επιπλέον κάθε υπόδειγμα έχει διαφορετικές απαιτήσεις από το μαθητή στην κατάκτηση και την εφαρμογή του ενώ από διδακτική πλευρά τα υποδείγματα είναι ελάχιστα μελετημένα ιδιαίτερα στο επίπεδο της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης.

Προβολή στον άξονα (Δ) των υποστηριζόμενων γλωσσών προγραμματισμού:

Πολλές φορές η γλώσσα προγραμματισμού που καλείται κανείς να χρησιμοποιήσει είναι δεδομένη και δεν αφήνεται ως επιλογή στον εκπαιδευτικό. Η επιλογή της γλώσσας θεωρείται συχνά σημαντική προκειμένου, για παράδειγμα, να αξιοποιηθεί ο χρόνος που θα αφιερώσει ο μαθητής σε μια γλώσσα που να θεωρείται ανταγωνιστική, ή να μπορεί χρησιμοποιηθεί και σε άλλα μαθήματα, κλπ. Από τις γλώσσες που χρησιμοποιούνται στα περιβάλλοντα, ευρύτερα γνωστές είναι οι ακόλουθες: LOGO, BASIC, PASCAL, C, C++, JAVA, C# και ASSEMBLY. Ιδιαίτερο ενδιαφέρον παρουσιάζουν τα συστήματα που υποστηρίζουν ταυτόχρονα περισσότερες από μία γλώσσες.

Προβολή στον άξονα (Ε) του σχήματος της προγραμματιστικής διαδικασίας:

Κάθε περιβάλλον υποθέτει και υποδεικνύει ένα αφηρημένο σχήμα για τον προγραμματισμό και αναμένεται να επηρεάσει και την αντίληψη των μαθητών για το περιεχόμενο της

αντίστοιχης διαδικασίας. Όπως φαίνεται και στον πίνακα 2, από την άποψη αυτή οι περιηγητές εμφανίζουν τον προγραμματισμό τους ως μια διαδικασία κατά την οποία *«κάνω τον περιηγητή να κάνει κάτι»*. Κατόπιν, ο μαθητής που αλληλεπιδρά με μια χελώνα της LOGO, καλείται να *κάνει μια αφηρημένη λογισμική οντότητα να κάνει κάτι*.

Το επόμενο υπόδειγμα εισάγει ορισμένους χελωνόκοσμους που υποστηρίζουν την χρήση πολλαπλών χελωνών (ή ρομπότ-μινιατούρων) και έτσι ο προγραμματισμός εμφανίζεται ως διαδικασία κατά την οποία *«κάνω οντότητες να αλληλεπιδρούν για να κάνουν κάτι»*. Στην περίπτωση της εκπαιδευτικής ρομποτικής, οι μαθητές ακολουθούν το προηγούμενο σχήμα, αλλά όχι με λογισμικές οντότητες παρά με συσκευές που κατασκευάζουν οι ίδιοι, έτσι ο προγραμματισμός μπορεί στην περίπτωση αυτή να περιγραφεί ως δραστηριότητα κατά την οποία *«κατασκευάζω συσκευές που αλληλεπιδρούν για να κάνουν κάτι»*. Στα περιβάλλοντα ανάπτυξης παιχνιδιών οι μαθητές *«ορίζουν λογισμικές οντότητες-αντικείμενα και χώρους μέσα στους οποίους τοποθετούνται για να αλληλεπιδράσουν τόσο μεταξύ τους όσο και με τον χρήστη ώστε να γίνει κάτι»*. Στα συστήματα αυτομάτου ελέγχου ο μαθητής *«ελέγχει ένα σύστημα»*. Στα περιβάλλοντα προγραμματισμού χαμηλού επιπέδου οι μαθητές *«δίνουν εντολές σε ένα υπολογιστή για να κάνει κάτι»*. Στα περιβάλλοντα επαγγελματικού προγραμματισμού, οι μαθητές καλούνται να ελέγξουν τη λειτουργία του υπολογιστή ή της λογισμικής του αφαίρεσης που εκφράζουν οι προκαθορισμένες βιβλιοθήκες προγραμματισμού του περιβάλλοντος και του ΛΣ.

Ερευνητικές κατευθύνσεις

Έχοντας υπόψη την παραπάνω ανάλυση είναι δυνατό να σκιαγραφηθούν νέες ερευνητικές κατευθύνσεις για τη Διδακτική του προγραμματισμού υπολογιστών, αλλά και γενικότερα για την αξιοποίηση των ΤΠΕ στην εκπαίδευση, όπως:

- μελέτη από τη σκοπιά της Διδακτικής
 - o της εξοικείωσης των μαθητών με τα διάφορα υποδείγματα προγραμματισμού (γνωστικές δυσκολίες, προτεινόμενες διδακτικές παρεμβάσεις),
 - o της εμπλοκής των μαθητών σε διάφορα αφηρημένα σχήματα για τη διαδικασία του προγραμματισμού,
 - o εννοιών του Αυτοματισμού, της Ρομποτικής και της Τεχνητής Νοημοσύνης που αποτελούν νέα γνωστικά πεδία για τις πρώτες βαθμίδες της εκπαίδευσης
- μελέτη και βελτίωση της κατανόησης των γνωστικών απαιτήσεων (γνώσεις και δεξιότητες) της εμπλοκής των μαθητών σε κάθε διαφορετικό περιβάλλον και των εννοιών που εμπλέκονται από συγγενικά γνωστικά πεδία όπως τα Μαθηματικά και οι Φυσικές επιστήμες,
- μελέτη της επίδρασης των περιβαλλόντων στη γενική ικανότητα επίλυσης προβλημάτων και στον επιτεύξιμο βαθμό ανάπτυξης της δεξιότητας του προγραμματισμού,
- τεκμηρίωση της προτεινόμενης ηλικιακής κατανομής,
- ανάδειξη των δυνατοτήτων και των ορίων των διαφόρων περιβαλλόντων ως μέσων κατασκευής ψηφιακών τεχνουργημάτων για έκφραση, σκέψη και νοερό πειραματισμό από μαθητές.

Από την επισκόπηση των εκπαιδευτικών περιβαλλόντων για τον προγραμματισμό διαπιστώνεται ποικιλία διαθέσιμων συστημάτων που μπορούν να καλύψουν όλο το φάσμα των μαθητικών ηλικιών καθώς και τα πιο συνηθισμένα υποδείγματα και τις γνωστές γλώσσες προγραμματισμού. Κάθε περιβάλλον υιοθετεί διαφορετικό σχήμα για

τη διαδικασία και το σκοπό του προγραμματισμού με απλούστερο αυτό των περιηγητών και πολύπλοκότερο αυτό της εκπαιδευτικής ρομποτικής. Η προτεινόμενη επισκόπηση και ταξινόμηση των περιβαλλόντων εκπαίδευσης στον προγραμματισμό αναμένεται να διευκολύνει το διδακτικό σχεδιασμό των εκπαιδευτικών και να προσανατολίσει τη σχετική εκπαιδευτική έρευνα.

Βιβλιογραφία

AgentSheets (2005), Τελευταία πρόσβαση: 13 ΙΑΝ 2006, Ιστοχώρος: <http://www.agentsheets.com>

Bergin J., Stehlik, M., Roberts, J. and Pattis, R., (2002), *Karel J. Robot: A gentle Introduction to the Art of Object-Oriented Programming in Java*, Available on the web at: <http://csis.pace.edu/~bergin/KarelJava2ed/Karel++JavaEdition.html>

Bergin, J., Stehlik, M., Roberts, J., Pattis, R., (1997), *Karel++: A Gentle Introduction to the Art of Object-Oriented Programming*, John Wiley & Sons, Inc., New York, NY,

Buck, D. and Stucki, D., (2001), JKarelRobot: A Case Study in sSupporting Levels levels of Cognitive cognitive Development development in the Computer computer Science science Curriculumcurriculum., *ACM SIGCSE Bulletin Proceedings of the Thirty-Second SIGCSE Technical Symposium*, ACM Press 33(1), 16-20.

Deek, F., and McHugh, J., (1998), A survey and critical analysis of Tools for Learning Programming., *Computer Science Education*, 8(2), 130-178

Faculty of Education, Curtin University of Technology, (1997), Children, robotics and problem solving: Technology in the early childhood classroom, *Australian Educational Computing*, 12 (2), 24-31

Ferrari M., Ferrari G., Hempel R. (2001), *Building Robots With Lego Mindstorms : The Ultimate Tool for Mindstorms Maniacs*, Syngress, Rockland, MASyngress

GameMaker (2005), Τελευταία πρόσβαση: 13 ΙΑΝ 2006, Ιστοχώρος: <http://www.gamemaker.nl>

Habgood, M.P.J. (2005), Zombie Division: Intrinsic Integration in Digital Learning Games. *Proceedings of the 2005 Human Centred Technology Workshop* 28-29 June 2005. Brighton, UK, Διαθέσιμο στη δνση: <http://hct.fcs.sussex.ac.uk/Submissions/12.pdf>

IntelliBrain™-Bot (2005), Τελευταία πρόσβαση: 13 ΙΑΝ 2006, Ιστοχώρος: <http://www.ridgesoft.com/intellibrainbot/intellibrainbot.htm>

Kafai, Y. B. (2001), The Educational Potential of Electronic Games: From Games-To-Teach to Games-To-Learn. Τελευταία πρόσβαση: Retrieved 1st January IAN, 20042006, from <http://culturalpolicy.uchicago.edu/conf2001/papers/kafai.html>

Karel the Robot Online. Internet. (2002), Τελευταία πρόσβαση: 13 ΙΑΝ 2006, Ιστοχώρος: <http://math.otterbein.edu/JKarelRobot>

Knuth D. (1997/1968), “The art of computer programming. Volume I (3rd ed.). Fundamental Algorithms”, Addison-Wesley Inc/Addison Wesley Longman Publishing Co., Inc., Redwood City, CA

LEGO Mindstorms (2005), Τελευταία πρόσβαση: 13 ΙΑΝ 2006, Ιστοχώρος: <http://www.lego.com/eng/education/mindstorms/default.asp>

MIT CRICKET (2005), Τελευταία πρόσβαση: 13 ΙΑΝ 2006, Ιστοχώρος: <http://llk.media.mit.edu/projects/cricket/>

MIT GOGO (2005), Τελευταία πρόσβαση: 13 ΙΑΝ 2006, Ιστοχώρος: <http://padthai.media.mit.edu:8080/cocoon/gogosite/home.xsp?lang=en>

Papert S. (1991), “Νοητικές Θύελλες: Παιδιά, ηλεκτρονικοί υπολογιστές και δυναμικές ιδέες”, Αθήνα: Εκδόσεις Οδυσσέας (Ελληνική μετάφραση)

Parallax (2005), Τελευταία πρόσβαση: 13 ΙΑΝ 2006, Ιστοχώρος: <http://www.parallax.com>

Pattis, R. E., (1995), Karel the Robot: A gentle Introduction to the Art of Programming, 2nd ed., John Wiley & Sons, Inc., New York, NY, John Wiley & Sons.

Pea R. (1986), “Language-Independent Conceptual ‘Bugs’ in Novice Programming”, *Journal of Educational Computing Research*, 2 (1), 25-36

Rogério Reis and Nelma Moreira., (2001), APOO: an environment for a firsts course in assembly language programming, *ACM SIGCSE Bulletin (ACM Special Interest Group on Computer Science Education)*, 33(24), 43-47

Sanders, D. and Dorn, B., (2003), Jerroo: A Tool for Teaching Object-Oriented Programming, *ACM SIGCSE Bulletin, Proceedings of the Thirty-Fourth SIGCSE Technical Symposium*, ACM Press 35(1), 201-204.

Spohrer J., & Soloway E., (1986), “Novice Mistakes: Are the Folk Wisdoms Correct?”, *Communications of the ACM*, 29 (7), 624-632

StageCast, (2005), Τελευταία πρόσβαση: 13 ΙΑΝ 2006, Ιστοχώρος: <http://www.stagecast.com>

Stephenson Chris et al, (2005), The New Educational Imperative: Improving High School Computer Science Education, Final Report of the CSTA Curriculum Improvement Task Force, ACM, Τελευταία πρόσβαση 23 ΙΑΝ 2006, Ιστοχώρος: http://csta.acm.org/Publications/White_Paper07_06.pdf

Truong N., Bancroft P., Roe P., (2005), Learning to program through the web, *ACM SIGCSE Bulletin, Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education ITiCSE '05*, ACM Press 37(3), 9-13

Δαπόντες Ν, Ιωάννου Σ, Μαστρογιάννης Ι, Τσοβόλας Σ, Τζιμόπουλος Ν, Αλπίας Α., (2003): *Ο Δάσκαλος Δημιουργός: Παιδαγωγική αξιοποίηση του Microworlds Pro στο Νηπιαγωγείο και Δημοτικό*, Αθήνα: Καστανιώτη

Κόμης Β., (2005), Εισαγωγή στην Διδακτική της Πληροφορικής, Αθήνα: Εκδόσεις Κλειδάριθμος

Ευνόγαλος Σ., Σατρατζέμη Μ., Δαγδιλέλης Β., Ευαγγελίδης Γ., (2005), Η Διδασκαλία της Κληρονομικότητας στον Προγραμματιστικό Μικρόκοσμο objectKarel, στο Α. Τζιμογιάννης (επιμ.), Πρακτικά 3ου Πανελληνίου Συνεδρίου, "Διδακτική της Πληροφορικής", Κόρινθος, 7-9 Οκτωβρίου, Κόρινθος

Ράπτης, Α. & Ράπτη, Α., (2001), *Μάθηση και διδασκαλία στην εποχή της Πληροφορίας. Παιδαγωγικές δραστηριότητες (τόμος Β')*, Αθήνα: Α. Ράπτη

Σπυρόπουλος, Χ., Δαγδιλέλης, Β., (2005), Η Διδασκαλία της Αντικειμενοστραφούς Λογικής σε Αρχάριους Προγραμματιστές, με το Περιβάλλον AgentSheets, στο Α. Τζιμογιάννης (επιμ.), Πρακτικά 3ου Πανελληνίου Συνεδρίου, "Διδακτική της Πληροφορικής", Κόρινθος, 7-9 Οκτωβρίου, Κόρινθος

Τζιμογιάννης Αθ., Κόμης Β., (2000), Η έννοια της μεταβλητής στον Προγραμματισμό: δυσκολίες και παρανοήσεις μαθητών του Ενιαίου Λυκείου, στο Α. Δημητρακοπούλου (επιμ.), Πρακτικά 2ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή "Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση", 103-114, Πάτρα

Φεσάκης Γ., Δημητρακοπούλου Α., (2005), Γνωστικές δυσκολίες μαθητών δευτεροβάθμιας εκπαίδευσης σχετικά με την έννοια της προγραμματιστικής μεταβλητής και προτεινόμενες παρεμβάσεις, στο Α. Τζιμογιάννης (επιμ.), Πρακτικά 3ου Πανελληνίου Συνεδρίου, "Διδακτική της Πληροφορικής", Κόρινθος, 7-9 Οκτωβρίου, Κόρινθος