

Προγραμματιστικά Περιβάλλοντα Προσχολικής και Πρώτης Σχολικής Ηλικίας

Θεόδωρος Χ. Κασκάλης

Επίκουρος Καθηγητής, Τμήμα Νηπιαγωγών,
Παιδαγωγική Σχολή Φλώρινας, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Θεσσαλονίκη, Ελλάδα
kaskalis@uom.gr

ΠΕΡΙΛΗΨΗ

Η διδασκαλία γενικών προγραμματιστικών αρχών σε ηλικίες προσχολικών και πρώτων σχολικών τάξεων συγκεντρώνει όλο και μεγαλύτερη προσοχή από την ερευνητική κοινότητα. Η εργασία αυτή έχει σαν σκοπό να παρουσιάσει την υπάρχουσα κατάσταση και τις προσφερόμενες δυνατότητες των διαθέσιμων αυτήν τη στιγμή πακέτων, που ενσωματώνουν παρόμοιες ιδέες. Αφού παρουσιαστούν οι επιδιωκόμενοι στόχοι και τα επιθυμητά χαρακτηριστικά από ένα τέτοιο προγραμματιστικό περιβάλλον, προχωρούμε στην εξερεύνηση των υπάρχοντων προσεγγίσεων, διακρίνοντας συγκεκριμένες κατηγορίες λογισμικού. Έμφαση δίνεται σε εκείνες τις προτάσεις, που καλύπτουν όσο γίνεται γενικότερα τη διδασκαλία του προγραμματισμού, χωρίς να θυσιάζουν την ευκολία χρήσης και την ενσωμάτωση σε ένα ρεαλιστικό σχολικό περιβάλλον. Μέσα από κατασκευασμένα παραδείγματα, αναπαριστώνται οι δυνατότητες των πακέτων και εξάγονται συμπεράσματα, σχετικά με τις προγραμματιστικές ιδέες που υλοποιούν. Η συγκριτική παρουσίαση των ποικίλων λύσεων επιτρέπει τον εντοπισμό των χρήσιμων χαρακτηριστικών, που κατά περίπτωση μπορούν να εξυπηρετήσουν ένα συγκεκριμένο αναλυτικό πρόγραμμα ή να οδηγήσουν στη δημιουργία ενός νέου περιβάλλοντος.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Γραφικά Περιβάλλοντα, Διδασκαλία Προγραμματισμού

ΕΙΣΑΓΩΓΗ

Οι ηλεκτρονικοί υπολογιστές αποτελούν πλέον ένα σημαντικό μέρος της σύγχρονης, καθημερινής ζωής, όχι μόνον σαν αυτόματα αντικείμενα που αντιμετωπίζουν καταστάσεις και δέχονται δεδομένα, αλλά σαν προγραμματιζόμενα εργαλεία. Ο προγραμματισμός τους, συγκεκριμένα, είναι ίσως ένα από τα πιο γοητευτικά τους χαρακτηριστικά, αφού επιτρέπει την προσαρμογή τους στις ανάγκες ενός συγκεκριμένου χρήστη. Πόσοι χρήστες όμως αξιοποιούν αυτήν την προσφερόμενη δυνατότητα; Σίγουρα όχι πολλοί. Οι απλοί χρήστες Η/Υ σπανίως έχουν τον καιρό ή τη διάθεση να εκπαιδευθούν στα εργαλεία και να γνωρίσουν τις δεξιότητες ενός επαγγελματία προγραμματιστή. Στην προσπάθεια να γίνει η διαδικασία του προγραμματισμού πιο προσιτή στον απλό χρήστη, συμβάλλουν τρεις διαφορετικές προσεγγίσεις: οι οπτικές γλώσσες προγραμματισμού (Visual Programming Languages), τα οπτικά περιβάλλοντα προγραμματισμού (Visual Programming Environments) και ο προγραμματισμός με παραδείγματα (PBE: Programming By Example) ή προγραμματισμός με επίδειξη (PBD: Programming By Demonstration) (Cypher, 1993), (Lieberman, 2001), (Pane & Myers, 1996).

Η επίσημη εισαγωγή της διδασκαλίας της Πληροφορικής στο Νηπιαγωγείο (ΔΕΠΠΣΝ, 2001) ανοίγει νέους ερευνητικούς ορίζοντες στον τομέα της χρήσης του υπολογιστή στην προσχολική και πρώτη σχολική ηλικία. Η εκπαίδευση του Νηπιαγωγείου γενικά επιδιώκει την καθοδήγηση των παιδιών προς την εκμάθηση νέων δεξιοτήτων και τη δημιουργία μιας προσωπικής αντίληψης για τον κόσμο, κάνοντας τη μάθηση ευκολότερη και πιο φυσική. Η χρήση υπολογιστών στα Νηπιαγωγεία (όπου υφίσταται) χρησιμοποιεί το νέο αυτό μέσο για ασκήσεις και δραστηριότητες

που δε συνδέονται άμεσα με την ίδια την επιστήμη της Πληροφορικής (π.χ. ζωγραφική, σχεδιασμός). Είναι άραγε δυνατόν να εισαχθούν τα παιδιά από τόσο μικρή ηλικία στο πιο δυναμικό χαρακτηριστικό των υπολογιστών: την προσαρμογή και καθοδήγησή τους σύμφωνα με τις ανάγκες του εκάστοτε χρήστη; Και αν ναι, ποια είναι τα αναμενόμενα εκπαιδευτικά αποτελέσματα;

Ο σχεδιαστής της γλώσσας προγραμματισμού Logo, S. Papert, περιέγραψε το εγχείρημά του σαν την προσπάθεια προσαρμογής των καλύτερων ιδεών της επιστήμης των υπολογιστών στην εκπαιδευτική διαδικασία, με σκοπό την καλλιέργεια ποικίλων δεξιοτήτων σε διαφορετικούς τομείς (Papert, 1980). Αντιμετωπίζοντας τον προγραμματισμό σαν μια δεξιότητα επικοινωνίας (Morgado et al, 2001), (Smith & Cypher, 1998), εμφανίζεται λογικό να περιμένουμε οφέλη από την εκμάθηση καθορισμού κανόνων (rules), τη διαπίστωση του τρόπου που άλλοι (π.χ. ο υπολογιστής) ακολουθούν τους κανόνες αυτούς και τον τρόπο που οι κανόνες αλληλεπιδρούν μεταξύ τους. Οι έννοιες του καθορισμού και της εξέλιξης συγκεκριμένων "συμπεριφορών" που δίνονται προγραμματιστικά σε συγκεκριμένα αντικείμενα, επιτρέπει τη μελέτη και αλλαγή των συμπεριφορών αυτών, με σκοπό την εκάστοτε αξιοποίησή τους σε διαφορετικά περιβάλλοντα. Οι δραστηριότητες αυτές, σε συνδυασμό με τη δημιουργία συνεργατικών ομάδων και την ανταλλαγή εμπειριών με μαθητές, που βρίσκονται μακριά και διαφέρουν γλωσσικά και πολιτισμικά, ανοίγει νέους δρόμους στην κοινωνικοποίηση των παιδιών (Chiocciariello et al, 2000).

Από τη άλλη μεριά, το πρόβλημα της προσέγγισης του προγραμματισμού στους απλούς χρήστες, εμφανίζεται τώρα πολλαπλασιαζόμενο, καθώς αναφερόμαστε σε ιδιαίτερα μικρές ηλικίες (Mayer, 1989), (Pane & Myers, 1996), (Smith & Cypher, 1998). Σε προσχολικό επίπεδο, ένα γενικό πακέτο λογισμικού θεωρείται ικανοποιητικό όταν: περιέχει γραφικό μενού κατευθυνόμενο με απλές κινήσεις του ποντικιού, περιέχει γραφικά και ήχους υψηλής ποιότητας που είναι ελκυστικά αλλά όχι υπερβολικά ερεθιστικά, κατευθύνεται από πολλαπλές εισόδους, αναφέρεται σε ποικίλους τομείς ευφύας, δεν είναι περιοριστικό αλλά "ανοικτό", αντικατοπτρίζει δραστηριότητες του πραγματικού κόσμου, δεν απαιτεί γνώσεις ανάγνωσης, είναι φιλικό προς τον χρήστη, είναι διαδραστικό, προσφέρει μεγάλο πεδίο επιλογών, "απαντά" με τρόπο ευγενικό και θετικό, προσφέρει ομαλή και διαισθητική καθοδήγηση (οπτική και ηχητική), έχει μεγάλες γραφικές περιοχές δράσης (π.χ. κουμπιά), περιλαμβάνει ομιλία υψηλού επιπέδου και προσφέρει διαφορετικά επίπεδα δυσκολίας. Αναφερόμενοι σε προγραμματιστικά περιβάλλοντα, μπορούμε να προσθέσουμε και τα παρακάτω χαρακτηριστικά (Chiocciariello et al, 2000):

- Απλότητα, χωρίς περιορισμό των προσφερόμενων δυνατοτήτων προγραμματισμού.
- Οπτική διαχείριση των βασικών αντικειμένων ελέγχου.
- Πολλαπλή αναπαράσταση των βασικών αντικειμένων (π.χ. εικόνα, σύμβολο, νέο σχέδιο). Πρέπει να επιτρέπεται ο επαναπροσδιορισμός των αντικειμένων.
- Υποστήριξη ομαδικής εργασίας και ανταλλαγής απόψεων – συζητήσεων, κατά τη φάση ανάπτυξης ή αντιμετώπισης ενός προβλήματος.
- Εφικτή εφαρμογή σε υπάρχοντα σχολικά περιβάλλοντα (π.χ. κόστος απόκτησης, συντήρηση υποδομής, γνωστικές απαιτήσεις από τους διδάσκοντες).

Λαμβάνοντας υπόψη τα παραπάνω χαρακτηριστικά, θα επιχειρήσουμε μια καταγραφή των υπάρχοντων περιβαλλόντων προγραμματισμού, που αναφέρονται σε μικρές ηλικίες, και θα εμβυθύνουμε σε εκείνες τις περιπτώσεις, που συγκεντρώνουν τα πιο σημαντικά από τα αναφερόμενα επιθυμητά στοιχεία. Ολοκληρώνοντας, θα αναφερθούμε σε ορισμένα συμπεράσματα, που έχουν καταγραφεί από την εφαρμογή τέτοιων περιβαλλόντων σε προσχολικές και πρώτες σχολικές ηλικίες.

ΥΠΑΡΧΟΝΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ

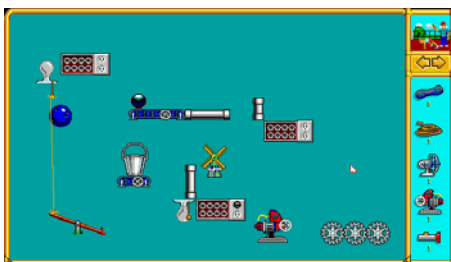
Η προσέγγιση των αρχών του προγραμματισμού στις πρώτες σχολικές ηλικίες, εμφανίζεται να ακολουθεί τρεις διαφορετικές προσεγγίσεις:

- Λογισμικό γενικών προγραμματιστικών στόχων.
- Προγραμματισμός ρομποτικών κατασκευών.
- Περιβάλλοντα προσομοίωσης.

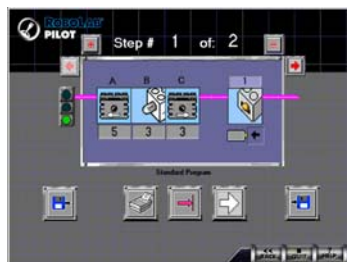
Στην περίπτωση που εξετάζουμε, οι δύο τελευταίες κατηγορίες εμφανίζουν συγκεκριμένους περιορισμούς, που κάνουν τη χρησιμοποίησή τους (προς το παρόν τουλάχιστον) αναποτελεσματική.

Συγκεκριμένα, τα περιβάλλοντα προσομοίωσης, αν και καλύπτουν μεγάλο μέρος των προαναφερθέντων απαιτήσεων, τουλάχιστον σε ό,τι αφορά την όψη και τον τρόπο διαχείρισής τους, εντούτοις δεν ενσωματώνουν τις γενικές βασικές αρχές του προγραμματισμού. Δεν επιτρέπουν την ανάπτυξη γενικών προγραμμάτων, ούτε ενσωματώνουν πλήρεις δομές. Συνήθως παραμένουν στο επίπεδο του εποικοδομητικού παιχνιδιού, επιτρέποντας ικανοποιητικό βαθμό αυτόνομων περιηγήσεων και δοκιμών, επιβάλλοντας όμως συγκεκριμένους περιορισμούς. Σαν αντιπροσωπευτικό παράδειγμα της κατηγορίας αυτής, μπορούμε να παρουσιάσουμε το προϊόν της Sierra On-Line, με όνομα *The Incredible Machine* (Σχήμα 1.α). Οι χρήστες καλούνται να χρησιμοποιήσουν (διασυνδέσουν) μοχλούς, πλατφόρμες, ψαλίδια, ανεμιστήρες, κινητήρες, πηγές ενέργειας κ.α. με σκοπό να λύσουν το πρόβλημα κάθε επιπέδου. Χρησιμοποιείται μία άμεση μέθοδος χειρισμού από τον χρήστη: κάθε αντικείμενο της εργαλειοθήκης, που βρίσκεται στη δεξιά πλευρά, σύρεται στην κεντρική σκηνή. Ανά πάσα στιγμή ο χρήστης δοκιμάζει τη λειτουργικότητα της τρέχουσας διάταξης ενεργοποιώντας την προσομοίωση. Η επίδραση της βαρύτητας σε συγκεκριμένα αντικείμενα, σε συνδυασμό με την κίνηση άλλων υπάρχοντων αντικειμένων (π.χ. μπαλονιών ή ζώων) εξελίσσει το σύστημα που προσομοιώνεται.

Τα ρομποτικά κατασκευαστικά σύνολα (Robotic Construction Kits), από την άλλη μεριά, έχουν μεγάλη απήχηση στην εκπαιδευτική κοινότητα και συγκεντρώνουν πολλά και αξιολόγια χαρακτηριστικά, σε ό,τι αφορά τον τομέα του προγραμματισμού (Miglino et al, 1999). Το επίπεδο χειρισμού τους όμως παραμένει σε αρκετά υψηλό επίπεδο, καθιστώντας δύσκολη την εφαρμογή τους σε εκπαιδευτικά πλαίσια προσχολικών και πρώτων σχολικών ηλικιών. Συγκεκριμένα, τα γραφικά περιβάλλοντα επικοινωνίας τους δεν έχουν καταφέρει ακόμη να απαλλαγούν από την χρήση κειμένου, εμποδίζοντας την περαιτέρω ενσωμάτωσή τους σε μαθήματα νηπιακών τάξεων.



(α)



(β)

Σχήμα 1: Περιβάλλοντα Προσομοίωσης και Προγραμματισμού Ρομποτικών Συνόλων

Το πιο γνωστό και διαδεδομένο προϊόν αυτής της κατηγορίας είναι με διαφορά τα LEGO® Mindstorms™. Το προϊόν αυτό μπορεί να χρησιμοποιηθεί με δύο διαφορετικά γραφικά περιβάλλοντα επικοινωνίας, ένα εκ των οποίων απεικονίζεται στο Σχήμα 1.β. Αν και το επίπεδο φιλικότητας προς τον χρήστη βρίσκεται σε ιδιαίτερα υψηλό επίπεδο, εντούτοις παρουσιάζονται

σοβαρές δυσκολίες στον χειρισμό από παιδιά μικρών ηλικιών, καθώς, αλλού περισσότερο και αλλού λιγότερο, απαιτείται η δυνατότητα της ανάγνωσης. Με τα δεδομένα αυτά, η χρήση των ρομποτικών κατασκευών σε νηπιακές τάξεις συνήθως καταλήγει στο να κατασκευάζει και προγραμματίζει ο ενήλικος εκπαιδευτής, ενώ τα παιδιά περιορίζονται στον καθορισμό της ιδέας προς υλοποίηση (Chiocciariello et al, 2000).

Αξίζει να παρατηρήσουμε, πάντως, ότι υπάρχουν αξιόλογες προσπάθειες απλοποίησης του περιβάλλοντος προγραμματισμού των ρομποτικών κατασκευαστικών συνόλων. Στο (Chiocciariello et al, 2000) αναφέρεται το περιβάλλον CAB-la, το οποίο στοχεύοντας στις ηλικίες 5–8, επιχειρεί να απλοποιήσει τον τρόπο διαχείρισης, χωρίς να μειώσει τις προσφερόμενες προγραμματιστικές δυνατότητες. Επίσης, στόχος αυτού του εγχειρήματος είναι η υποστήριξη του κοινωνικού περιεχομένου του προγραμματισμού ρομποτικών κατασκευαστικών συνόλων. Συγκεκριμένα, το περιβάλλον CAB-la σχεδιάστηκε στοχεύοντας στις ανάγκες μικρών ομάδων παιδιών, που συνεργάζονται για ένα μεγάλο χρονικό διάστημα στα πλαίσια έργων (projects). Έμφαση δίνεται στον τομέα της τεκμηρίωσης: μέσα από κάθε έργο είναι δυνατόν να προσπελαστούν διάφορες σελίδες τεκμηρίωσης, αποθηκευμένες τοπικά ή στο Διαδίκτυο. Δύο συγκεκριμένοι μηχανισμοί: το ιστορικό και ο κοινός δικτυακός χώρος αποθήκευσης, επιτρέπουν στα παιδιά να μοιραστούν τις σκέψεις, απόψεις, απορίες ή προβλήματά τους. Παράλληλα, έχουν τη δυνατότητα να γυρίσουν σε οποιοδήποτε στάδιο της ανάπτυξης του έργου επιθυμούν και να αναζητήσουν διαφορετικά μονοπάτια δράσης. Η έννοια της ευρείας συνεργασίας και συνδιαλλαγής στοχεύει στην ανάπτυξη ενός μέσου με καινούρια και χρήσιμα χαρακτηριστικά κοινωνικοποίησης. Αξίζει να παρατηρήσουμε, πάντως ότι και οι ίδιοι οι συγγραφείς, παραδέχονται ότι το τρέχον περιβάλλον προγραμματισμού παραμένει ακόμη αρκετά αφαιρετικό και δύσκολο για παιδιά μικρών ηλικιών.

Η τρίτη και πιο γενική κατηγορία από τις προαναφερθείσες, είναι το λογισμικό γενικών προγραμματιστικών στόχων, το οποίο φαίνεται να συγκεντρώνει τα περισσότερα από τα επιθυμητά χαρακτηριστικά. Εδώ ανήκει και η πιο "κλασική" εκπαιδευτική γλώσσα προγραμματισμού, η Logo. Αν και το σύνθημα της Logo είναι "χωρίς πάτωμα, χωρίς οροφή" ("no floor, no ceiling"), εντούτοις η γνώση της γραφής και ανάγνωσης μιας τουλάχιστον φυσικής γλώσσας, αποτελεί ένα είδος πατώματος. Αν και υπάρχουν προσπάθειες δημιουργίας φιλικών γραφικών περιβαλλόντων, το εμπόδιο αυτό δεν φαίνεται να έχει ξεπεραστεί. Αξίζει, πάντως να αναφέρουμε τις πιο ενδιαφέρουσες προσπάθειες, εκ των οποίων οι τρεις πρώτες προσφέρονται δωρεάν, ενώ η τέταρτη (Imagine Logo) επιδεικνύει σημαντικό και συνεχές επίπεδο εξέλιξης και ενσωμάτωσης ποικίλων ετερογενών χαρακτηριστικών (π.χ. ζωγραφική, animation, κατασκευή-ενσωμάτωση δικτυακών τόπων, χρήση πολυμέσων, δυνατότητες επικοινωνίας, συνεργατική ανάπτυξη κ.α.):

- StarLOGO: <http://el.www.media.mit.edu/groups/el/Projects/starlogo>
- UCBLGO: <http://www.cs.berkeley.edu/~bh/logo.html>
- MSWLogo: <http://www.softronix.com/logo.html>
- Imagine Logo: <http://www.logo.com/cat/view/imagine.html>
- MicroWorlds: <http://www.microworlds.com>

Παρόμοια προβλήματα ενσωμάτωσης σε προσχολικές ηλικίες παρουσιάζουν και οι πολύ ενδιαφέρουσες ελληνικές προσεγγίσεις των προγραμμάτων Δ.Ε.Λ.Υ.Σ. και Αβάκιο E-Slate:

- Δ.Ε.Λ.Υ.Σ: <http://macedonia.uom.gr/~delys>
- Αβάκιο E-Slate: <http://e-slate.cti.gr>

Οι προγραμματιστικοί μικρόκοσμοι που ενσωματώνουν (Pascal-like και Logo, αντίστοιχα) αν και συγκεντρώνουν χαρακτηριστικά σημαντικής εκπαιδευτικής αξίας και τεχνικής-αισθητικής αρτιότητας, εντούτοις παραμένουν εξαρτώμενοι από γνώσεις μιας τουλάχιστον φυσικής γλώσσας.

Από την άλλη μεριά, αποδεικνύεται ότι υπάρχουν ικανοποιητικά περιβάλλοντα διδασκαλίας των αρχών του προγραμματισμού, που βασίζονται αποκλειστικά (ή τουλάχιστον σε πολύ μεγάλο βαθμό) σε γραφικά σύμβολα και εικόνες. Τα περιβάλλοντα, που προέκυψαν μετά την αναζήτησή μας, είναι τα ακόλουθα:

- Ηλεκτρονικά Παραδείγματα του Εθνικού Συμβουλίου Διδασκόντων Μαθηματικών (National Council of Teachers of Mathematics): <http://standards.nctm.org/document/eexamples>
- Το περιβάλλον HANDS: <http://www-2.cs.cmu.edu/~pane/research.html>
- Το περιβάλλον KidSim, που αργότερα εξελίχθηκε στο δωρεάν προσφερόμενο Cocoa <http://www.kidsdomain.com/games/cocoa.html>, ενώ μία πιο πλούσια, εμπορική εκδοχή του είναι το Stagecast Creator: <http://www.stagecast.com>
- Το περιβάλλον ToonTalk™: <http://www.toontalk.com>

Στα παραπάνω αξίζει να προσθέσουμε το περιβάλλον Agentsheets, το οποίο είναι εμπορικά διαθέσιμο στη διεύθυνση: <http://agentsheets.com>. Το περιβάλλον αυτό μοιάζει αρκετά στο προαναφερθέν Cocoa, ενώ η γλώσσα που χρησιμοποιεί (Visual AgentTalk) αυξάνει τις προσφερόμενες προγραμματιστικές δυνατότητες, θυσιάζοντας την ευκολία χειρισμού από μικρές ηλικίες. Το συγκεκριμένο περιβάλλον, δεν θα περιγραφεί αναλυτικότερα, διότι θεωρούμε ότι στα σημεία που διαφοροποιείται από το Cocoa, ξεπερνά το επιδιωκόμενο επίπεδο ευκολίας. Στην ακόλουθη ενότητα θα εμβαθύνουμε στα παραπάνω περιβάλλοντα προγραμματισμού, παρουσιάζοντάς τα και δίνοντας συγκεκριμένα αντιπροσωπευτικά παραδείγματα υλοποίησης, που αντικατοπτρίζουν τις προσφερόμενες δυνατότητες και τα σημαντικότερα χαρακτηριστικά τους.

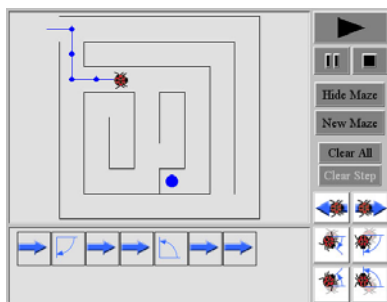
ΠΑΡΟΥΣΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ ΓΕΝΙΚΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΣΤΟΧΩΝ

Το Εθνικό Συμβούλιο Διδασκόντων Μαθηματικών (National Council of Teachers of Mathematics) στις δικτυακές του σελίδες περιέχει ένα σύνολο παραδειγμάτων, με στόχο την εκμάθηση μαθηματικών και προ-μαθηματικών εννοιών. Ανάμεσα στα παραδείγματα αυτά, όμως, συναντούμε ορισμένες περιπτώσεις, που μπορούν να αξιοποιηθούν σε ασκήσεις εκμάθησης προγραμματιστικών αρχών, καθώς παραπέμπουν άμεσα στην ιδέα της Logo, δοσμένης απλοποιημένα και με γραφικό τρόπο.

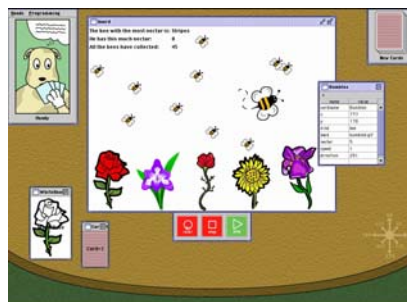
Στο Σχήμα 2.α βλέπουμε ένα από τα συγκεκριμένα παραδείγματα, όπου είναι φανερή η αναλογία της "πασχαλίστας" με την "χελώνα" της Logo. Το παράδειγμα αυτό, όπως και τα υπόλοιπα προσφερόμενα, είναι γραμμένο σε Java, με αποτέλεσμα να μην απαιτείται η προμήθεια ή εγκατάσταση άλλου λογισμικού πλην ενός κλασικού φυλλομετρητή. Τα έξι λευκά εικονίδια, που εμφανίζονται κάτω και δεξιά στο Σχήμα 2.α, αποτελούν τις βασικές κινήσεις που μπορεί να δηλώσει ο χρήστης. Πιέζοντάς τα διαδοχικά, εμφανίζεται η σειρά των επιλεγμένων κινήσεων στο κάτω μέρος του παραθύρου. Στο παράδειγμά μας έχουμε επιλέξει επτά διαδοχικές κινήσεις. Πιέζοντας το πλήκτρο, που φαίνεται επάνω και δεξιά, εκτελείται το σύνολο των επιλεγμένων κινήσεων. Στο Σχήμα 2.α έχουμε πιέσει το συγκεκριμένο πλήκτρο και βλέπουμε την τρέχουσα θέση του συμβόλου της πασχαλίστας. Προφανώς, απώτερος στόχος είναι η επιλογή εκείνων των κινήσεων, που θα μας οδηγήσουν στην "λύση" (κουκίδα) του λαβύρινθου. Μέσω απλών χειρισμών είναι δυνατή η διαγραφή συγκεκριμένων κινήσεων, που δεν οδηγούν στη λύση του προβλήματος, ενώ είναι ανά πάσα στιγμή δυνατός ο καθαρισμός όλων των επιλεγμένων βημάτων.

Η απλοποιημένη αυτή εκδοχή των κινήσεων της χελώνας της Logo επιτρέπει την εισαγωγή σε απλές προγραμματιστικές αρχές. Η σειρά των κινήσεων αντικατοπτρίζει τη διαδοχή των εντολών. Η ίδια εντολή (π.χ. κίνηση εμπρός) επηρεάζεται από τις προηγούμενες (στροφή), αναπαριστώντας την αλληλεπίδραση των δηλωμένων κανόνων. Αναπαριστάται η έννοια των διαφορετικών φάσεων της δημιουργίας κώδικα και της εκτέλεσης. Οι φάσεις αυτές εναλλάσσονται με σκοπό τον ενδιάμεσο έλεγχο της πορείας προς τη λύση. Τέλος, η διαγραφή κινήσεων αποτελεί ένα είδος

εκσφαλμάτωσης του προγράμματος, επιτρέποντας την επιστροφή σε σημεία, που έχουν ήδη ελεγχθεί. Το φανερό πρόβλημα της πρότασης αυτής είναι η αδυναμία δημιουργίας γενικών προγραμμάτων. Ο χρήστης περιορίζεται στο να λύνει προβλήματα, η δομή των οποίων εμπεριέχει αρχές που συναντώνται στον προγραμματισμό. Αυτό όμως είναι αναμενόμενο, καθώς ο αρχικός σκοπός του συγκεκριμένου προγράμματος είναι η εξοικείωση με απλές γεωμετρικές έννοιες.



(α)



(β)

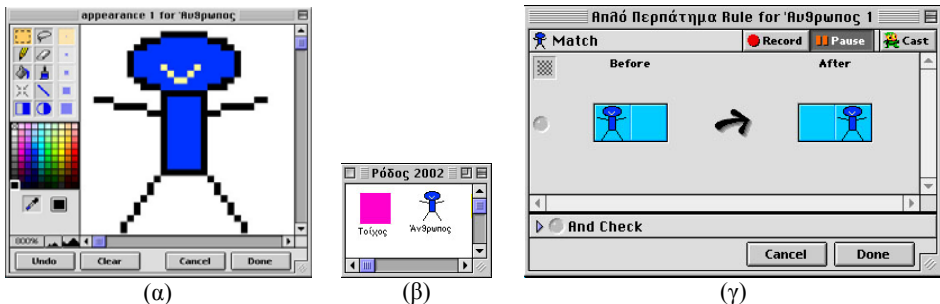
Σχήμα 2: Το Παράθυρο Ελέγχου της Πασχαλίσας και το Περιβάλλον HANDS

Προχωρώντας σε λύσεις, που παρουσιάζουν μεγαλύτερη πληρότητα σε ό,τι αφορά τις προγραμματιστικές δυνατότητες, συναντάμε το περιβάλλον προγραμματισμού HANDS (Σχήμα 2.β). Το συγκεκριμένο εγχείρημα αποτελεί ένα νέο προγραμματιστικό σύστημα για παιδιά, το οποίο δεν είναι ακόμη διαθέσιμο στο ευρύ κοινό. Προσφέρει ένα συμπαγές υπολογιστικό μοντέλο, που επιτρέπει σε χρήστες μικρών ηλικιών να εξοικειωθούν, με έναν εναλλακτικό τρόπο, με ιδέες όπως οι μεταβλητές, οι παράμετροι, η εμβέλεια εκτέλεσης, η ροή ελέγχου και οι υπορουτίνες. Ο συμβολισμός που χρησιμοποιείται είναι ένα παιχνίδι χαρτιών τράπουλας, που εμπεριέχει τα βασικά συστατικά στοιχεία του προγραμματισμού. Ένα τυπικό πρόγραμμα περιλαμβάνει ένα σύνολο παικτών, που συμβολίζουν τις διεργασίες ή τα αρθρώματα (modules). Κάθε παίκτης κρατά ένα σύνολο ενεργών και ανενεργών καρτών, οι οποίες είναι γνωστές μόνον στον ίδιο. Επίσης υπάρχουν κάρτες ανοικτές και ορατές από όλους τους παίκτες, υλοποιώντας το μηχανισμό της κεντρικής ή διαμοιραζόμενης μνήμης. Οι διαφορετικές κάρτες δίνουν υπόσταση σε διαφορετικούς τύπους πληροφορίας προγραμμάτων ή δεδομένων. Οι παίκτες εναλλακτικά "παίζουν" (εκτελούν) τις κάρτες τους σε μια μορφή συνεργατικής πολυ-επεξεργασίας.

Οι ιδέες, που ενσωματώνει το συγκεκριμένο περιβάλλον, παρουσιάζουν αρκετό ενδιαφέρον σε ό,τι αφορά "μοντέρνα" προγραμματιστικά μοντέλα, όπως για παράδειγμα ο αντικειμενοστραφής προγραμματισμός και η απουσία της έννοιας της κλασικής von Neumann μηχανής. Δυστυχώς, καθώς το συγκεκριμένο περιβάλλον δεν είναι ακόμη διαθέσιμο, δεν ήταν δυνατόν να αναπτύξουμε συγκεκριμένα παραδείγματα, ώστε να διαπιστωθεί το επίπεδο φιλικότητας προς το χρήστη. Τα επιμέρους χαρακτηριστικά, πάντως, του περιβάλλοντος HANDS συνιστούν μία αξιόλογη πρόταση για ηλικίες που έχουν κάποιο βαθμό εξοικείωσης με το γραπτό λόγο.

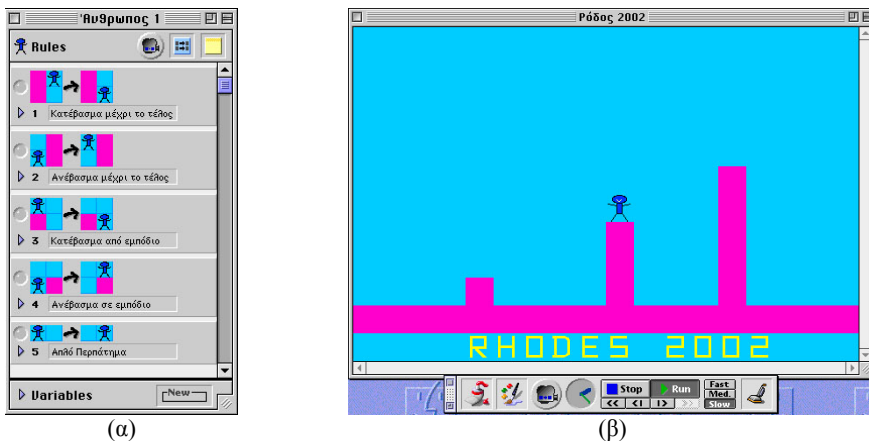
Προχωρώντας στην επόμενη πρόταση λογισμικού γενικών προγραμματιστικών στόχων, συναντάμε τα περιβάλλοντα Cocoa και Stagecast Creator. Το δεύτερο αποτελεί την εξέλιξη του πρώτου, με μόνη, αλλά σημαντική, διαφορά την εμπορική διαθεσιμότητά του. Εδώ θα ασχοληθούμε, με τη δωρεάν προσφερόμενη έκδοση του Cocoa (Gilmore et al, 1995), αφού το όφελος για τα σχολικά εργαστήρια αντισταθμίζει τα απουσιάζοντα χαρακτηριστικά, για τις απαιτήσεις των ηλικιών που εξετάζουμε. Το Cocoa αποτελεί ένα προγραμματιστικό περιβάλλον, στο οποίο τα παιδιά μπορούν να δημιουργήσουν απλά έργα (projects), ορίζοντας διαφορετικούς χαρακτήρες και τοποθετώντας τους σε ένα σκηνικό. Στη συνέχεια, και με πλήρως γραφικό τρόπο,

μπορούν να ορίσουν μεταβάσεις (κινήσεις) των χαρακτήρων αυτών, ανάλογα με την ικανοποίηση συγκεκριμένων προϋποθέσεων. Στο Σχήμα 3 αναπαριστώνται οι φάσεις δημιουργίας και καθορισμού κανόνων ενός απλού χαρακτήρα στο περιβάλλον Cocoa. Το Σχήμα 3.α επιδεικνύει τις δυνατότητες καθορισμού της όψης ενός χαρακτήρα, μέσα από απλά εργαλεία σχεδίασης. Για τις ανάγκες του παραδείγματος ζωγραφίσαμε δύο χαρακτήρες (τοίχος, άνθρωπος), που φαίνονται στο Σχήμα 3.β και θα τους χρησιμοποιήσουμε για να κατασκευάσουμε το μοντέλο ενός κινούμενου ανθρώπου μέσα σε ένα σκηνικό, που περιλαμβάνει εμπόδια. Ο πρώτος κανόνας, που πρέπει να ορίσουμε, είναι η κίνηση του ανθρώπου. Μέσα από το παράθυρο, που φαίνεται στο Σχήμα 3.γ μπορούμε να ορίσουμε, με κινήσεις του ποντικιού, τον εξής απλό κανόνα: "αν δεξιά του ανθρώπου υπάρχει κενό πλαίσιο, τότε ο άνθρωπος πρέπει να μετακινηθεί στο πλαίσιο αυτό". Η εικόνα αναπαριστά γραφικά αυτόν τον κανόνα, επιτρέποντας τη συνεχή δεξιά οριζόντια κίνηση του ανθρώπου όσο δεν συναντά εμπόδια.



Σχήμα 3: Ορισμός Χαρακτήρων και Κανόνων στο Περιβάλλον Cocoa

Δημιουργώντας ένα σκηνικό από εμπόδια και ανθρώπους, μπορούμε να δοκιμάσουμε την ισχύ του κανόνα μας "εκτελώντας" (προσομοιώνοντας) το περιβάλλον της κίνησης. Προφανώς, όταν ο χαρακτήρας μας θα συναντά εμπόδια, ο κανόνας, μη έχοντας ισχύ, θα τον ακινητοποιεί. Σε μια τέτοια περίπτωση, μπορούμε επιτόπου να ορίσουμε νέους κανόνες, που θα αντιμετωπίζουν τις εκάστοτε καταστάσεις. Στο Σχήμα 4.α έχουμε ήδη δηλώσει άλλους τέσσερις κανόνες, που εξασφαλίζουν το "προσπέρασμα" των εμποδίων, που συναντά ο άνθρωπός μας. Το Σχήμα 4.β αναπαριστά το συνολικό γραφικό περιβάλλον του Cocoa, με το παράδειγμά μας σε εξέλιξη.



Σχήμα 4: Σύνολο Κανόνων και Παράθυρο Εκτέλεσης του Περιβάλλοντος Cocoa

Οι προγραμματιστικές δομές, που εμπεριέχονται στους δηλωθέντες κανόνες, αν και φαινομενικά απλές, είναι πραγματικά πολύ ισχυρές. Ο κανόνας που φαίνεται στο Σχήμα 3.γ (όπως και οι κανόνες 1 και 2 του Σχήματος 4.α) αποτελεί μια αφαιρετική αναπαράσταση της δομής **While-Do**. Ταυτοχρόνως, πίσω από τους κανόνες αυτούς, υπονοείται η έννοια του **If-Then**, αφού αναπαριστούν περιπτώσεις, που ισχύουν υπό συνθήκη. Τέλος, η συνεχής κλήση του ίδιου κανόνα, όσο τίθεται σε ισχύ, μπορεί ακόμα και να επεκταθεί στην έννοια της αναδρομής. Αξίζει να παρατηρήσουμε ότι ένα παράδειγμα μπορεί να είναι όσο απλό ή όσο πολύπλοκο απαιτείται, για τον επιδιωκόμενο εκπαιδευτικό στόχο. Λόγω του γραφικού τρόπου ορισμού των κανόνων, όμως, η πολυπλοκότητα ενός μεγάλου προγράμματος γίνεται σύντομα υπερβολική. Για τον λόγο αυτό, το περιβάλλον Cocoa εμφανίζεται πιο ελκυστικό για τη διδασκαλία μικρών επιμέρους έργων, η διαχείριση των οποίων μπορεί να διατηρείται σε λογικό επίπεδο.

Ολοκληρώνοντας την παρουσίαση του λογισμικού γενικών προγραμματιστικών στόχων, θα αναφερθούμε στο περιβάλλον ToonTalk™ (Kahn, 2000), (Morgado et al, 2001). Το πακέτο αυτό είναι το πιο απλλαγμένο από κείμενα (αν και όχι ολοκληρωτικά) και έχει σαφή κατεύθυνση σε μικρές ηλικίες. Τόσο τα γραφικά του σύμβολα (μεγάλα και απλά), όσο και ο τρόπος χειρισμού του (ίδιες αποκρίσεις σε διαφορετικά πλήκτρα του ποντικιού), απευθύνονται σε παιδιά προσχολικών ή πρώτων σχολικών ηλικιών. Επιπροσθέτως, το "πεδίο δράσης" του ποντικιού γίνεται άμεσα σαφές, με ένα "τρεμπάιγμα" των αντικειμένων, που βρίσκονται υπό την επίδρασή του. Στο Σχήμα 5 βλέπουμε τα βασικά εργαλεία που μπορεί να χρησιμοποιήσει ο χρήστης. Συγκεκριμένα, υπάρχει μία εργαλειοθήκη με αντικείμενα προς χρήση (αριθμοί, γράμματα, κουτιά κ.α.) που χρησιμεύουν για ποικίλες εργασίες, από την απλή αναπαράσταση, μέχρι την εκτέλεση πράξεων, συγκρίσεων ή μεταφορών. Επίσης, ο χρήστης έχει στη διάθεσή του τροποποιητικά εργαλεία, όπως μία ηλεκτρική σκούπα (σβήσιμο αντικειμένων), ένα μαγικό ραβδί (αντιγραφή), ένα σημειωματάριο (αρχείο εικόνων και νέων αντικειμένων) ή μία τρόμπια ποδηλάτου (αλλαγή μεγέθους).



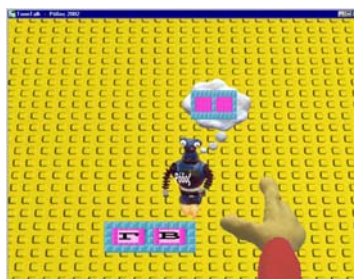
(α)



(β)



(γ)



(δ)

Σχήμα 5: Το Περιβάλλον Προγραμματισμού ToonTalk™

Σημαντικό στοιχείο της εργαλειοθήκης είναι το "ρομπότ". Το αντικείμενο αυτό περιλαμβάνει μία "φυσαλίδα σκέψης" (thought bubble), η οποία όταν τροφοδοτηθεί με ένα συγκεκριμένο αντικείμενο (π.χ. δύο κουτιά με γράμματα στο Σχήμα 5.α) μεταβαίνουμε στο παράθυρο "σκέψης" (προγραμματισμού) του ρομπότ (Σχήμα 5.β), όπου μπορούμε να ορίσουμε συγκεκριμένους κανόνες, με τρόπο γραφικό. Στο παράδειγμά μας κατευθύνουμε το ρομπότ να βγάλει το ένα γράμμα από τη θέση του και να το τοποθετήσει προσωρινά έξω από τα κουτιά. Στη συνέχεια αλλάζουμε θέση στο δεύτερο γράμμα και τέλος τοποθετούμε το πρώτο γράμμα στην κενή θέση, που δημιουργήθηκε. Βγαίνοντας από το παράθυρο σκέψης του ρομπότ, μπορούμε να το θέσουμε σε λειτουργία αν το τροφοδοτήσουμε με ένα αντικείμενο ίδιο με αυτό που "θυμάται" (στην περίπτωση μας δύο κουτιά με τα γράμματα Α και Β). Μπορούμε να γενικεύσουμε τον κανόνα, χρησιμοποιώντας το εργαλείο της ηλεκτρικής σκούπας και σβήνοντας τα γράμματα Α και Β από τη "σκέψη" του, ώστε να αντιμετωθεί οποιαδήποτε γράμματα βρίσκονται σε δύο κουτιά. Η ενέργεια αυτή αναπαριστάται στα Σχήματα 8.γ και 8.δ, με τα γράμματα Β και Γ.

Η προγραμματιστική αρχή, που περιέχεται στο παράδειγμα, περιλαμβάνει τις έννοιες των μεταβλητών και την ανταλλαγή των τιμών τους. Τα δύο κενά κουτιά στη σκέψη του ρομπότ, μπορούμε να πούμε ότι αναπαριστούν δύο μεταβλητές x και y , που περιέχουν γράμματα, ενώ ο κανόνας που "θυμάται" το ρομπότ είναι η γραφική αναπαράσταση της δομής:

$temp:=x$ $x:=y$ $y:=temp$

Από την άλλη μεριά, η όλη έννοια του ρομπότ αναπαριστά την ιδέα της υπορουτίνας, που δέχεται σαν ορίσματα συγκεκριμένες μεταβλητές, πάνω στις οποίες εφαρμόζει τον περιεχόμενο κώδικά της. Όπως είναι αναμενόμενο, ο ορισμός κανόνων για περισσότερα ρομπότ, που θα "εργάζονται" ταυτόχρονα και ενδεχομένως θα "συνεργάζονται", επιτρέπει την ανάπτυξη μεγαλύτερων παραδειγμάτων, επιτρέποντας τη δημιουργία πολύπλοκων προγραμμάτων, που θα περιλαμβάνουν την παράλληλη εκτέλεση πολλαπλών διεργασιών.

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Προσπαθήσαμε να καλύψουμε την περιοχή των προγραμματιστικών περιβαλλόντων, που απευθύνεται σε προσχολικές και πρώτες σχολικές ηλικίες. Είναι ενδιαφέρον να ανακαλύπτει κανείς το πλήθος των υπάρχοντων προσεγγίσεων, αλλά και δύσκολο να εμβαθύνει σε όλες τις επιμέρους προτάσεις ταυτόχρονα, χωρίς να υπερβαίνει τη συγκεκριμένη έκταση μιας εργασίας. Εδώ επιλέξαμε να εστιάσουμε την προσοχή μας στα περιβάλλοντα γενικών προγραμματιστικών στόχων, όπου σαν πιο πλήρεις και γενικές λύσεις, την παρούσα στιγμή, εμφανίζονται να είναι τα περιβάλλοντα Cocoa/Stagecast Creator και ToonTalk. Οι δύο αυτές προσεγγίσεις καλύπτουν μεγάλο μέρος από τα αρχικά επιθυμητά χαρακτηριστικά, που θέσαμε, χωρίς όμως να λύνουν ολοκληρωτικά το θέμα της διδασκαλίας του προγραμματισμού σε μικρές ηλικίες. Κανένα περιβάλλον δεν είναι πλήρως απαλλαγμένο από γλωσσικά σύμβολα, ενώ η ανάγκη χρήσης ποντικιού ή χειριστήριου (joystick) και οθόνης επιβάλλει συγκεκριμένους περιορισμούς.

Τι θα συνέβαινε όμως αν μπορούσαμε να αντικαταστήσουμε τις περιφερειακές συσκευές της οθόνης, του πληκτρολογίου και του ποντικιού; Τίθεται το πολύ ενδιαφέρον ερώτημα αν μπορούμε να μετατρέψουμε αντικείμενα της καθημερινής μας ζωής σε μονάδες εισόδου/εξόδου ενός ηλεκτρονικού υπολογιστή. Ο τομέας του "απτού προγραμματισμού" (tangible programming) αποτελεί ένα αξιόλογο αντικείμενο έρευνας, που αναμένεται να δώσει ικανοποιητικά αποτελέσματα στο μέλλον (McNerney, 2000), (Montemayor et al, 2001).

Σε ό,τι αφορά τις καταγραμμένες εμπειρίες από τη διδασκαλία προγραμματιστικών αρχών σε μικρές μαθητικές ηλικίες, διαπιστώνουμε ότι αν και ενθαρρυντικές, εντούτοις δεν στερούνται προβλημάτων. Η εμπειρία της διδασκαλίας της Logo σε μεγαλύτερες ηλικίες απέδειξε ότι οι βασικές προγραμματιστικές αρχές αφομοιώνονται όταν ο μαθητής λύνει τα δικά του προβλήματα και όχι ασκήσεις που θέτει ο εκπαιδευτής (Gilmore et al, 1995). Ο προγραμματισμός προκύπτει περισσότερο από την προσωπική ανακάλυψη και λιγότερο από την εκμάθηση ιδιοτήτων του

κώδικα και προτεινόμενων τεχνικών. Η χρήση του περιβάλλοντος KidSim, προγόνου του Cocoa, σε παιδιά ηλικιών 11-14, επιβεβαίωσε αυτόν τον κανόνα και για τα γραφικά περιβάλλοντα προγραμματισμού. Αξίζει όμως να παρατηρήσουμε ότι υπήρξε σημαντικό ενδιαφέρον από την πλευρά των παιδιών, τα οποία εξοικειώθηκαν γρήγορα με τη χρήση του πακέτου, δυσκολεύτηκαν όμως να αφομοιώσουν τις βασικές έννοιες των κανόνων (Gilmore et al, 1995).

Από την άλλη μεριά, η χρήση του ToonTalk σε ένα μαθητικό περιβάλλον προσχολικής ηλικίας (Morgado et al, 2001) έδωσε ενθαρρυντικά αποτελέσματα, αν συνυπολογιστεί η υπάρχουσα νοητική αδυναμία κατανόησης κανόνων και αφαιρέσεων. Τα παιδιά αντιμετώπισαν τον χειρισμό του περιβάλλοντος σαν μια ψυχαγωγική διαδικασία και συγκεκριμένες τεχνικές γενίκευσης αποδείχθηκαν ικανές να αφομοιωθούν. Σε κάθε περίπτωση πάντως, είναι ακόμη ανοικτό το ζήτημα της μεταφοράς των γνώσεων και της γενικότερης εκπαιδευτικής αξίας του αντικειμένου. Απαιτούνται περισσότερα δεδομένα πριν μπορέσουμε να καταλήξουμε σε ασφαλή συμπεράσματα, ενώ παραμένουν ασάφειες σε πολλά θέματα συμβολισμών και αναπαράστασης, που θα οδηγούσαν στην ευκολότερη κατανόηση των προγραμματιστικών τεχνικών. Είναι ενθαρρυντικό, πάντως, να διαπιστώνει κανείς ότι ο συγκεκριμένος τομέας είναι ερευνητικά ενεργός και αναμένεται να προκύψουν αρκετά αποτελέσματα στο άμεσο μέλλον.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Chiocciariello, A., Manca, S. & Sarti, L. (2000), Behaviour Construction Kit, *Proceedings of the i3 Annual Conference: Building Tomorrow Today: Community, Design and Technology*, 72-76
- Cypher, A. (Ed.) (1993), *Watch What I Do, Programming By Demonstration*, The MIT Press
- Διαθεματικό Ενιαίο Πλαίσιο Προγραμμάτων Σπουδών για το Νηπιαγωγείο, ΦΕΚ 1376 τ. Β' 18-10-2001, <http://www.pi-schools.gr/greek/depps/fek5.pdf>
- Gilmore, D.J., Pheasey, K., Underwood, J. & Underwood, G. (1995), Learning Graphical Programming: An Evaluation of KidSim, In K. Nordby, P.H. Helmersen, D.J. Gilmore, S.A. Arnesen (Eds.), *Human-Computer Interaction: Interact'95*, London: Chapman & Hall, 145-150
- Kahn, K. (2000), Generalizing by Removing Detail: How Any Program Can Be Created by Working with Examples, <http://www.animated-programs.com/PBD>
- Lieberman, H. (Ed.) (2001), *Your Wish Is My Command: Programming By Example*, Morgan Kaufmann Publishers
- Mayer, R.E. (1989), The Psychology of How Novices Learn Computer Programming, In E. Soloway, J.C. Spohrer (Eds.), *Studying the Novice Programmer*, Hillsdale, NJ: Lawrence Erlbaum Associates, 129-159
- McNerney, T.S. (2000), *Tangible Programming Bricks: An Approach to Making Programming Accessible to Everyone*, Master's Thesis, MIT Media Lab
- Miglino, O., Lund, H.H. & Cardaci, M. (1999), Robotics as an Educational Tool, *Journal of Interactive Learning Research*, 10(1), 25-48
- Montemayor, J., Druin, A., Farber, A., Simms, S., Churaman, W. & Damour, D. (2001), *Physical Programming: Designing Tools for Children to Create Physical Interactive Environments*, Univ. of Maryland, CS-TR-4288
- Morgado, L., Cruz, M.G.B. & Kahn, K. (2001), Working in ToonTalk with 4- and 5-year olds, *Playground International Seminar*, <http://www.utad.pt/~leonelm/TTon4-5.htm>
- Pane, J.F. & Myers, B.A. (1996), *Usability Issues in the Design of Novice Programming Systems*, TR CMU-CS-96-132
- Papert, S. (1980), *Mindstorms: Children, Computers and Powerful Ideas*, Basic Books NY
- Smith, D.C. & Cypher, A. (1998), Making Programming Easier for Children, In A. Druin (Ed.), *The Design of Children's Technology*, Morgan Kaufmann Publishers, 201-221