Διδακτική - γνωστική ανάλυση περιβαλλόντων προγραμματισμού προσχολικής και πρώτης σχολικής ηλικίας

Βασίλης Κόμης
komis@upatras.gr
Τμήμα Επιστημών της Εκπαίδευσης και της Αγωγής στην Προσχολική Ηλικία, Πανεπιστήμιο Πατρών

Περίληψη

Το άρθρο επιχειρεί μια διδακτική - γνωστική ανάλυση περιβαλλόντων προγραμματισμού προσχολικής και πρώτης σχολικής ηλικίας. Αρχικά παρουσιάζεται η εν γένει προβληματική για τη θέση των περιβαλλόντων αυτών στην εκπαίδευση των μικρών παιδιών, στη συνέχεια αναπτύσσεται ένα πλαίσιο διδακτικής και γνωστικής ανάλυσης των δύο κύριων κατηγοριών για τη μάθηση του προγραμματισμού και την ανάπτυξη της υπολογιστικής σκέψης (ρομποτική και οπτικός προγραμματισμός) και τέλος αναλύονται γνωστά περιβάλλοντα ρομποτικής και οπτικού προγραμματισμού.

Λέξεις κλειδιά: εκπαιδευτική ρομποτική, οπτικό περιβάλλον προγραμματισμού, γνωστική και διδακτική ανάλυση, προσχολική εκπαίδευση, υπολογιστική σκέψη

Εισαγωγή

Οι σύγχρονες προσεγγίσεις στα προγράμματα σπουδών προσχολικής και πρώτης σχολικής ηλικίας αντιλαμβάνονται πλέον τις ψηφιακές τεχνολογίες τόσο ως αντικείμενο νέων γραμματισμών (γνωριμία με τις τεχνολογίες, δημιουργική έκφραση μέσω των τεχνολογιών, κατανόηση της θέσης τους στην κοινωνία και τον πολιτισμό), όσο και ως γνωστικό εργαλείο με εγκάρσιες χρήσεις (επικοινωνία, συνεργασία, διερεύνηση, πειραματισμός και ανακάλυψη, επίλυση προβλήματος, ανάπτυξη δημιουργικότητας, κριτική σκέψη) σε όλα τα γνωστικά αντικείμενα. Ο ψηφιακός γραμματισμός συνιστά πλέον εγγενές τμήμα των προγραμμάτων σπουδών από τα πρώτα στάδια του σχολείου. Μάλιστα, με μια ευρεία προσέγγιση, ο γραμματισμός αυτός ολοκληρώνεται όταν εντάσσει στην προβληματική του όχι μόνο δεξιότητες κατανόησης και χρήσης των ψηφιακών εργαλείων αλλά και την ανάπτυξη της αλγοριθμικής σκέψης και της προγραμματιστικής ικανότητας που αποτελούν βασικές συνιστώσες της υπολογιστικής σκέψης (Wing, 2006). Η πρόσφατη έρευνα αναφέρει ότι στο πλαίσιο της προσχολικής εκπαίδευσης η υπολογιστική σκέψη μπορεί να αποτελέσει κτήμα των παιδιών και αφορά κατά κύριο λόγο την ανάπτυξη αλγοριθμικής ικανότητας και της γνώσης βασικών αρχών προγραμματισμού (Manches & Plowman, 2015; Μισιρλή, 2015). Η έννοια του ψηφιακού γραμματισμού συνεπώς δεν εμπερικλείει απλώς τη γνωριμία των μαθητών με τις τεχνολογίες αλλά αφορά κυρίως την ανάπτυξη ικανοτήτων δημιουργίας και έκφρασης με τις τεχνολογίες αυτές, οι οποίες αποτελούν εγγενές μέρος του κόσμου μέσα στον οποίο αναπτύσσονται τα παιδιά. Η δημιουργία και η έκφραση θεωρούνται ικανότητες υψηλού επιπέδου και συνδυάζουν χρήσεις ψηφιακών εργαλείων παραγωγής και μετασχηματισμού ψηφιακού υλικού (κείμενα, εικόνες, ήχους κλπ.) και χρήσεις τεχνολογιών ελέγχου και αλγοριθμικής προσέγγισης, όπως η ρομποτική και ο προγραμματισμός. Το παρόν άρθρο ασχολείται με τα ψηφιακά περιβάλλοντα που αφορούν μικρές ηλικίες και μπορούν να ευνοήσουν την ικανότητα της υπολογιστικής σκέψης με τη χρήση τεχνολογιών

Τ. Α. Μικρόπουλος, Α. Τσιάρα, Π. Χαλκή (επιμ.), *Πρακτικά 8ου Πανελλήνιου Συνεδρίου «Διδακτική της Πληροφορικής»*, Ιωάννινα: ΕΤΠΕ. 23-25 Σεπτεμβρίου 2016. ISSN 2529-0908, ISBN 978-960-88359-9-3.

ελέγχου και αυτομάτων (εκπαιδευτική ρομποτική) και οπτικών περιβαλλόντων προγραμματισμού.

Τεχνολογία ελέγχου και ρομποτική στην προσχολική εκπαίδευση

Οι τεχνολογίες ελέγχου και ειδικότερα οι εφαρμογές ρομποτικής καθώς και ο προγραμματισμός αυτομάτων και υπολογιστών συνιστούν ένα κατάλληλο εκπαιδευτικό πλαίσιο μέσα στο οποίο είναι δυνατόν να αναπτυχθούν υψηλού επιπέδου ικανότητες, όπως η επίλυση προβλήματος, η κριτική σκέψη, η αλγοριθμική προσέγγιση κ.α. Η παιδαγωγική προσέγγιση που ακολουθείται στην περίπτωση των τεχνολογιών ελέγχου στηρίζεται στην ανάπτυξη και την περιγραφή τεχνικών καταστάσεων που βασίζονται σε στοιχειώδεις γλώσσες εντολών, οι οποίες σχετίζονται με τη χρήση απλών (π.χ. χειριστήρια) ή και πιο σύνθετων (π.χ. προγραμματιζόμενα ρομπότ) συσκευών. Η εμφάνιση αναπτυξιακά κατάλληλων ρομποτικών συσκευών (π.χ. BeeBot ή Thymio) καθώς και οπτικών γλωσσών προγραμματισμού προσαρμοσμένων στις ικανότητες των παιδιών (π.χ. ScratchJr) εγγράφεται σε αυτή την προοπτική. Με την ένταξη των ρομπότ στο πρόγραμμα σπουδών οι μαθητές εισάγονται στην τεχνολογία ελέγχου και προσεγγίζουν βασικές έννοιες προγραμματισμού. Στη συνέχεια, κάνοντας χρήση προγραμματισμού εισάγονται σε πιο σύνθετες έννοιες και προσεγγίζουν την αλγοριθμική προσέγγιση ως βασικό συστατικό της υπολογιστικής σκέψης.

Τα περιβάλλοντα εκπαιδευτικής ρομποτικής

Η προσέγγιση της εκπαιδευτικής ρομποτικής προέρχεται απευθείας από τη θεωρία της Logo και κάνει χρήση κατάλληλων μικρόκοσμων (κατασκευή ή χειρισμός αυτομάτων όπως η «χελώνα» εδάφους), οι οποίοι χρησιμοποιούνται μέσα σε διάφορες παιδαγωγικές καταστάσεις με σημασία και νόημα για τους μαθητές.

Πίνακας 1. Ρομποτικά περιβάλλοντα προσχολικής και πρώτης σχολικής ηλικίας

Ρομποτική Συσκευή	Έννοιες	Ικανότητες	Ηλικία
Bee-Bot / BlueBot	Ακολουθία, κίνηση, αλγόριθμος	Αλγοριθμική, απτικός προγραμματισμός	4+
Probot	Ακολουθία, κίνηση, επανάληψη, αλγόριθμος, διαδικασία	Αλγοριθμική, απτικός προγραμματισμός	6+
Thymio	Ακολουθία, κίνηση, κατάσταση, μηχανή πεπερασμένης κατάστασης, αισθητήρες, γεγονότα	Αλγοριθμική, απτικός προγραμματισμός, υπολογιστική σκέψη, μαθηματική σκέψη	6+
Cubelets	Αισθητήρες, κίνηση, είσοδοι – έξοδοι, ενέργειες, παραλληλισμός	Αλγοριθμική, απτικός προγραμματισμός, κατασκευή	5+
Kibo	Ακολουθία, κίνηση, επανάληψη, επιλογή, αισθητήρες, αλγόριθμος	Απτικός προγραμματισμός, υπολογιστική σκέψη, κατασκευή	4+
LEGO® WeDo™	Ακολουθία, επανάληψη, επιλογή, αισθητήρες, μοτέρ	Υπολογιστική σκέψη, προγραμματισμός, κατασκευή	7+

Ειδικότερα, οι παιδαγωγικοί στόχοι της ρομποτικής μπορούν να διαιρεθούν σε δύο βασικές κατηγορίες: το χειρισμό και την κατασκευή ρομπότ. Η εκπαιδευτική ρομποτική στην προσχολική αγωγή αφορά κυρίως χειρισμό και όχι κατασκευή και συνιστά εναλλακτικό τρόπο εκμάθησης του προγραμματισμού (με έμφαση στην αλγοριθμική σκέψη) μέσα από το πρίσμα της ανάπτυξης και οργάνωσης της σκέψης μέσω πρόβλεψης για τη μετακίνηση αντικειμένων μέσα στο χώρο. Τα ρομπότ επιτρέπουν, μεταξύ άλλων, την εξερεύνηση του

χώρου «από απόσταση» χωρίς παρέμβαση του σώματος, την ακριβή και λογική γλώσσα εντολών μέσω μιας κωδικοποίησης, την πρόβλεψη των πράξεων και την αλγοριθμική οικοδόμηση της λύσης.
Δύο είναι οι κύριες κατηγορίες περιβαλλόντων εκπαιδευτικής ρομποτικής προσχολικής και πρώτης σχολικής ηλικίας (Πίνακας 1): περιβάλλοντα κατασκευαστικής ρομποτικής τύπου Lego-Logo (π.χ. Kibo και LEGO® WeDo™) και προγραμματιζόμενα ρομπότ τύπου Logo (π.χ. Bee-Bot και Pro-Bot).

Κατασκευαστική ρομποτική

Ένα περιβάλλον τύπου Lego-Logo, στο οποίο το ρομπότ κατασκευάζεται από δομικά στοιχεία τύπου Lego, αποτελεί το πιο χαρακτηριστικό παράδειγμα εκπαιδευτικής ρομποτικής αφού συνδυάζει τόσο το κατασκευαστικό όσο και το προγραμματιστικό τμήμα της προσέγγισης. Το περιβάλλον συνιστά ένα μικρόκοσμο, ο οποίος αποτελείται από φυσικές διασυνδέσεις και μια συμβολική γλώσσα ελέγχου και επιτρέπει τη μελέτη και τη δημιουργία αρθρωτών ρομπότ κατευθυνόμενων από υπολογιστή. Δημιουργούνται συνεπώς κατασκευές που περιέχουν κινητήρα, ο οποίος λειτουργεί είτε μέσω ειδικής διασύνδεσης με τον υπολογιστή, όπως στο περιβάλλον LEGO® WeDo™, είτε αυτόνομα, όπως στο Kibo ή το Cubelets. Οι κινήσεις των κατασκευών γίνονται είτε με άμεσο τρόπο (με χειριστήριο), είτε από προγράμματα που ο χρήστης δημιουργεί με τη βοήθεια λογισμικού. Τα ρομπότ μπορεί να είναι εφοδιασμένα με αισθητήρες και να αποστέλλουν μηνύματα στον υπολογιστή. Η κατασκευαστική ρομποτική παρέχει ένα πλαίσιο εισαγωγής σε έννοιες ποικίλων γνωστικών περιοχών, όπως φυσικές επιστήμες, τεχνολογία, μηχανική και μαθηματικά (Science, Technology, Engineering, Mathematics).

Προγραμματιζόμενα ρομπότ

Τα προγραμματιζόμενα ρομπότ τύπου Logo είναι προκατασκευασμένα ρομπότ δαπέδου, τα οποία προγραμματίζονται από τον χρήστη για να εκτελέσουν κάποια κίνηση στον χώρο (Misirli & Komis, 2014). Ο χρήστης σχεδιάζει και καθορίζει το σύνολο των εντολών που εισάγονται στο ρομπότ, κατά περίσταση, χρησιμοποιώντας τις εντολές μιας γλώσσας, η οποία αποτελεί υποσύνολο της γλώσσας προγραμματισμού Logo. Πρόκειται για έτοιμα ρομπότ που διαθέτουν απλή διεπιφάνεια με πλήκτρα, τα οποία αναπαριστούν βασικές εντολές της Logo και επιτρέπουν απτικό προγραμματισμό (tangible programming), γεγονός που καθιστά τη χρήση τους εύκολη ακόμα και από παιδιά προσχολικής ηλικίας. Δύο είναι οι κύριες κατηγορίες προγραμματιζόμενων ρομπότ προσχολικής ηλικίας: α) τύπου Logo με τρία τουλάχιστον ρομπότ: BeeBot, BlueBot (μετεξέλιξη του BeeBot με προσθήκη τον προγραμματισμό μέσω φορητής συσκευής) και Probot και β) μια ειδική ομάδα ρομπότ (π.χ. Thymio και Cubelets), η οποία αξιοποιεί τις δυνατότητες αισθητήρων (ανίχνευση κίνησης, απόστασης, έντασης ήχου, κλπ.) για πραγματοποίηση ενεργειών στον χώρο. Σε τεχνικό επίπεδο λειτουργούν κυρίως μέσω απτικού προγραμματισμού ενώ σε εννοιολογικό επίπεδο, η πρώτη κατηγορία ακολουθεί τις αρχές του δομημένου προγραμματισμού και η δεύτερη κατηγορία του προγραμματισμού χειρισμού γεγονότων. Τα πιο διαδεδομένα ρομπότ είναι το BeeBot και το Thymio.
Το Bee-Bot ενσαρκώνει τη χελώνα Logo με μορφή μέλισσας ενώ βασίζεται σε αρχές προγραμματισμού της Logo για τον έλεγχο του ρομπότ δαπέδου. Τα παιδιά μπορούν έτσι να προγραμματίσουν σύνθετες διαδρομές στο δάπεδο για να επιλύσουν ανοικτού τύπου προβλήματα (Μισιρλή, 2015). Ο προγραμματισμός των κινήσεων γίνεται από ένα σύνολο πλήκτρων στο πάνω μέρος του ρομπότ. Το Bee-Bot επιτρέπει τη χρήση βασικών εντολών της

Logo και ειδικά τη δομή της ακολουθίας. Το ProBot, με μορφή αυτοκινήτου, διαθέτει πλήρες αριθμητικό πληκτρολόγιο, υποστηρίζει τις δομές ακολουθίας και επανάληψης (Repeat), τη δημιουργία διαδικασιών (procedures) και μέσω αισθητήρα αφής μπορεί να υλοποιήσει τη δομή ελέγχου. Το Thymio εισάγει μια νέα προγραμματιστική προσέγγιση, αυτή του προγραμματισμού χειρισμού γεγονότων (event driven programming). Το ρομπότ είναι προ-προγραμματισμένο με έξι διαφορετικές συμπεριφορές: ακολουθεί ένα αντικείμενο, διερευνά το χώρο αποφεύγοντας εμπόδια, ανιχνεύει κτυπήματα και την κατεύθυνση της βαρύτητας, εντοπίζει πίστες και χρωματικές διαφορές, υπακούει σε χειριστήρια, ακούει και διακρίνει ήχους. Επίσης, μέσω μιας οπτικής γλώσσας με χαρακτηριστικά προγραμματισμού χειρισμού γεγονότων, ο προγραμματιστής οργανώνει συμβάντα (events) και δράσεις (actions), ορίζει δηλαδή τι θα γίνει όταν λάβει χώρα ένα γεγονός. Κάθε ζεύγος «συμβάν – δράση», ορίζει μια μετάβαση από μια κατάσταση του ρομπότ σε μια άλλη και μας βοηθά έτσι να εισάγουμε μια πολύ ισχυρή έννοια της Πληροφορικής, αυτή των μηχανών πεπερασμένης κατάστασης.

Διδακτικές και γνωστικές προεκτάσεις εκπαιδευτικής ρομποτικής

Τα περιβάλλοντα κατασκευαστικής ρομποτικής παρέχουν ένα περιβάλλον εισαγωγής σε έννοιες από ποικίλες γνωστικές περιοχές όπως κίνηση, αισθητήρες εισόδου ή εξόδου πληροφορίας, μηχανική κατασκευή, αυτοματισμός, υλοποίηση αλγορίθμων κλπ. Ένα τέτοιο περιβάλλον ευνοεί την ανάπτυξη μεταγνωστικής ικανότητας, κατά την οποία τα παιδιά αναστοχάζονται σχετικά με τις διαδικασίες σκέψης που έχουν ακολουθήσει, βελτιώνεται η ικανότητα επίλυσης προβλήματος και προάγεται η ικανότητα χωρικού προσανατολισμού και η ευαισθητοποίηση των παιδιών στην κίνηση, στα σχήματα και στις γωνίες. Επιπρόσθετα, δεδομένου ότι διαθέτουν μια γλώσσα προγραμματισμού, παρέχουν στους μαθητές ένα κατάλληλο περιβάλλον εισαγωγής σε έννοιες αλγοριθμικής και υπολογιστικής σκέψης.

Απτικός προγραμματισμός/ οπτικός προγραμματισμός και δομές προγραμματισμού

Το βασικό χαρακτηριστικό όλων των ρομποτικών συστημάτων που αναλύονται στο άρθρο αυτό, εκτός από το LEGO® WeDo™, του οποίου ο προγραμματισμός γίνεται μέσω υπολογιστή, είναι η λειτουργία μέσω (ή και μέσω) απτικού προγραμματισμού. Το Beebot λειτουργεί μόνο με απτικό προγραμματισμό, το Bluebot και το Probot βασίζονται στον απτικό προγραμματισμό ενώ μπορούν να προγραμματιστούν και ψηφιακά, το Thymio προγραμματίζεται μέσω λογισμικού (γλώσσα VPL/Aseba) αλλά στις βασικές του λειτουργίες λειτουργεί απτικά κλπ. Ο απτικός προγραμματισμός χαρακτηρίζεται από τη φυσική / απτική σχέση του προγραμματιστή με το ρομπότ, με όλες τις γνωστικές διαστάσεις που η σχέση αυτή εμπεριέχει στο επίπεδο της προσχολικής εκπαίδευσης (Bers & Horn, 2009). Παράλληλα, όλες οι γλώσσες προγραμματισμού που συνοδεύουν τα προγραμματιζόμενα ρομπότ ακολουθούν το παράδειγμα του οπτικού προγραμματισμού, που επιτρέπει στο παιδί τη δημιουργία προγραμμάτων μέσα από τον εικονικό χειρισμό προγραμματιστικών στοιχείων. Όλα τα ρομπότ υποστηρίζουν τη δομή της ακολουθίας (sequence). Δεδομένου ότι η ροή του προγράμματος αναπαρίσταται εικονικά, η συγγραφή αλλά και η κατανόηση / διόρθωση των προγραμμάτων είναι σχετικά εύκολη διαδικασία. Κάποια περιβάλλοντα, όπως το Probot, υποστηρίζουν και την επαναληπτική δομή (repeat) ενώ άλλα, μέσω αισθητήρων, μπορούν να προσομοιώσουν δομές ελέγχου. Το προγραμματιστικό περιβάλλον του LEGO® WeDo™ είναι πιο σύνθετο αφού εκτός των βασικών δομών προγραμματισμού

επιτρέπει τον χειρισμό κινητήρα και αισθητήρων που μετατρέπουν το ρομπότ σε ένα πολλαπλών χρήσεων αυτόματο.

Αισθητήρες / είσοδοι - έξοδοι
Οι αισθητήρες, που ανιχνεύουν φυσικά μεγέθη και παράγουν από αυτά μετρήσιμες εξόδους, αποτελούν μια από τις κύριες λειτουργίες του προγραμματιζόμενου ρομπότ. Είναι το βασικό στοιχείο που διακρίνει μια ρομποτική συσκευή από ένα σύνηθες υπολογιστικό σύστημα. Εξάλλου, οι αισθητήρες ενδημούν στο περιβάλλον των παιδιών και συνιστούν για αυτά μέρος της καθημερινότητάς τους, όπως αισθητήρες κίνησης, φωτός, αφής, απόστασης, ήχου κλπ. Σε κάποια συστήματα, όπως το Thymio, οι αισθητήρες αποτελούν το κύριο μέσο επικοινωνίας με το ρομπότ δεδομένου ότι λαμβάνουν εισόδους / πληροφορίες από τους χρήστες ή το περιβάλλον και τις μετατρέπουν σε ενέργειες της ρομποτικής συσκευής. Σε άλλα ρομποτικά συστήματα, όπως τα LEGO® WeDo™, οι αισθητήρες αποτελούν πρόσθετα εξαρτήματα, τα οποία επεκτείνουν το πεδίο δράσης ή ενεργειών της ρομποτικής συσκευής. Η κατανόηση του ρόλου, της λειτουργίας και των ρυθμίσεων βασικών αισθητήρων για τον χειρισμό αυτομάτων αποτελεί πλέον μέρος της εγκύκλιας γνώσης που πρέπει να αποκτήσουν οι μαθητές, και στο πλαίσιο αυτό, σημαντικός είναι ο ρόλος της εκπαιδευτικής ρομποτικής.

Μηχανολογική κατασκευή - κινητήρας / κίνηση
Ο χειρισμός ενός ρομπότ επιτρέπει την ανάλυση της κίνησης που κάνει μέσα στον χώρο και τον χρόνο και του συγχρονισμού των επιμέρους κινήσεων. Τα ρομπότ εισάγουν έτσι την έννοια της λογικής του χειρισμού για την εκπλήρωση ενός έργου ή την επίτευξη ενός στόχου. Κατασκευάζοντας ένα ρομπότ, προσεγγίζουμε τα προβλήματα της μετάδοσης και του μετασχηματισμού των κινήσεων. Το παιδί αναπαράγει έτσι ένα δοσμένο μηχανισμό ή ανακαλύπτει έναν άλλο για να πραγματοποιήσει μια δεδομένη κίνηση. Ανάλογα με τον τύπο του ρομπότ, το παιδί το χειρίζεται ή σαν μια μαριονέτα και αποκρυσταλλώνει σε προφορικό λόγο τις πράξεις του (χειροκίνητη χρήση) ή το κατευθύνει από απόσταση με τη βοήθεια ενός χειριστηρίου (στην περίπτωση αυτή πρόκειται για αναλογικό χειρισμό) ή προγραμματίζει τις κινήσεις του στο πληκτρολόγιο ενός υπολογιστή ή απευθείας σε αυτό (πρόκειται για λογικό χειρισμό με την χρήση μιας κωδικοποιημένης γλώσσας).

Τα περιβάλλοντα οπτικού προγραμματισμού

Η μάθηση του προγραμματισμού και γενικότερα η ανάπτυξη της υπολογιστικής σκέψης εντάσσεται πλέον στα προγράμματα σπουδών ήδη από την προσχολική εκπαίδευση (Manches & Plowman, 2015).
Η εξέλιξη αυτή οφείλεται αφενός στις γενικότερες προσεγγίσεις που αφορούν τη θέση των ψηφιακών τεχνολογιών στην εκπαιδευτική διαδικασία και αφετέρου στην εξέλιξη των διεπιφανειών χρήσης των υπολογιστικών συστημάτων (οθόνες αφής & άμεση αλληλεπίδραση) και στην ανάπτυξη κατάλληλων περιβαλλόντων μάθησης προγραμματισμού. Τα περιβάλλοντα αυτά έχουν συνήθως παιγνιώδη μορφή, βασίζονται στις αρχές του οπτικού προγραμματισμού (εντολές με εικόνες) και απαιτούν απλές γνώσεις χρήσης ψηφιακών συσκευών, ενώ επιτρέπουν στους μαθητές να χειριστούν σημαντικές έννοιες προγραμματισμού, όπως ακολουθία, επανάληψη, επιλογή, κίνηση, αλγόριθμος, διαδικασία, παράμετρος, μεταβλητή (Πίνακας 2), ενώ κάποια περιβάλλοντα εισάγουν νέες έννοιες προγραμματισμού: συγχρονισμός, μηνύματα, συμβάντα.

Πίνακας 2. Περιβάλλοντα προγραμματισμού προσχολικής και πρώτης σχολικής ηλικίας

Λογισμικό οπτικού προγραμματισμού	Έννοιες	Ηλικία
ScratchJr	Ακολουθία, επανάληψη, παράμετρος, κίνηση, συνθήκη, αλγόριθμος, συμβάντα, μηνύματα, ήχος, έλεγχος, συγχρονισμός	5+
The foos	Ακολουθία, επανάληψη, συνθήκη επιλογής, κίνηση, παράμετρος, αλγόριθμος	5+
Run Marco!	Ακολουθία, επανάληψη, συνθήκη επιλογής, κίνηση, αλγόριθμος	5+
Move the Turtle	Ακολουθία, επανάληψη, συνθήκη επιλογής, κίνηση, παράμετρος, αλγόριθμος, διαδικασία, μεταβλητή, χωρικός προσανατολισμός	6+
Cato's Hike	Ακολουθία, επανάληψη, συνθήκη επιλογής, κίνηση, παράμετρος, αλγόριθμος	6+
Kodable	Ακολουθία, επανάληψη, συνθήκη επιλογής, κίνηση, παράμετρος, αλγόριθμος	
LightBot Junior	Ακολουθία, επανάληψη, κίνηση, αλγόριθμος, διαδικασία, χωρικός προσανατολισμός	5+
Bit by Bit	Ακολουθία, κίνηση, αλγόριθμος, διαδικασία, χωρικός προσανατολισμός, συγχρονισμός	5+

Βασικές δομές προστακτικού προγραμματισμού

Τα περιβάλλοντα της ενότητας αυτής λειτουργούν σε φορητές συσκευές (IOS ή Android). Όλα υποστηρίζουν δομές προστακτικού προγραμματισμού (ακολουθία και επανάληψη, και κατά περίπτωση επιλογή). Παράλληλα, βασίζονται στις αρχές του οπτικού προγραμματισμού και προσπαθούν να εφαρμόσουν απλό συντακτικό και διαφανή σημασιολογία στις εντολές τους.

Συντακτικό και σημασιολογία

Ο προγραμματισμός στα περιβάλλοντα αυτά αφορά στη μάθηση γλώσσας επικοινωνίας ώστε να εκτελεστούν συγκεκριμένες διεργασίες από τη φορητή συσκευή σχετικά με δράσεις ενός ή και περισσότερων αντικειμένων στη σκηνή (οθόνη εργασίας). Κάθε περιβάλλον διαθέτει το δικό του συντακτικό και σημασιολογία που συνδέεται με τα βασικά στοιχεία της γλώσσας. Οι μαθητές, μαθαίνοντας το περιβάλλον, αναπτύσσουν συντακτικές και σημασιολογικές γνώσεις, ενώ είναι δυνατόν να αναπτύξουν στρατηγικού τύπου γνώσεις, που συνδέονται με τη μεταφορά γνώσης σε άλλους γνωστικούς χώρους (Fay & Mayer, 1988).
Το συντακτικό είναι απλό με πολλά κοινά χαρακτηριστικά (λόγω οπτικού προγραμματισμού) ενώ η σημασιολογία διαφέρει σημαντικά από περιβάλλον σε περιβάλλον για τις περισσότερες εντολές. Η συντακτική δομή συνδέεται με τη μορφή της διεπιφάνειας χρήσης, η οποία διαθέτει τρεις συνιστώσες (Fay & Mayer, 1988): τα βασικά στοιχεία (εικονίδια εντολών), τους κανόνες σύνταξης αυτών των στοιχείων σε εντολές και τους κανόνες σύνταξης των εντολών σε προγράμματα. Η σύνταξη σε κάποιες από τις γλώσσες αυτές έχει τη μορφή παζλ (π.χ. ScartchJr και RunMarco!) ενώ σε άλλες κάθε εντολή τοποθετείται μέσα σε μία ταινία με τετράγωνα κουτιά, όπου κάθε κουτί επιδέχεται μία εντολή (The foos, Cato’s Hike, LightbotJr, Bit by Bit, Kodable). Μόνο στη γλώσσα Move the Turtle το περιβάλλον σύνταξης εντολών θυμίζει κλασικό παράθυρο σύνταξης, όπου οι εντολές γράφονται η μία κάτω από την άλλη και εισάγονται μέσα από ένα μενού εντολών. Στις άλλες γλώσσες η σύνταξη είναι περισσότερο διαισθητική, δεδομένου ότι λειτουργεί με την αρχή «σύρε κι άφησε» ενώ όλα τα προγράμματα ή τα σενάρια (τμήματα

προγραμμάτων) συναπαρτίζονται από εικονίδια εντολών τοποθετημένα χωρικά στον χώρο προγραμματισμού (σε μορφή παζλ ή μέσα σε κουτιά).
Η σημασιολογία μιας γλώσσας προγραμματισμού συνδέεται με τη σημασία που προσδίδουμε στις διαφορετικές συντακτικές κατασκευές που δημιουργούνται στο περιβάλλον της. Με την έννοια αυτή, η σημασιολογία είναι δύσκολο να οριστεί και σε κάθε περίπτωση καθορίζει εάν το περιβάλλον μπορεί να υποστηρίζει διαφορετικά προγραμματιστικά παραδείγματα, όπως ο προστακτικός ή ο αντικειμενοστραφής προγραμματισμός. Η σημασιολογία μιας γλώσσας δεν είναι αυθαίρετη αλλά διαθέτει λογική δομή που εξαρτάται από τις λειτουργικές συνιστώσες του χώρου προγραμματισμού. Οι συνιστώσες που περιγράφουν τη σημασιολογία των εντολών μιας γλώσσας είναι τρεις : α) οι πράξεις (operations), οι δράσεις δηλαδή που μπορούν να πραγματοποιηθούν (π.χ. κίνηση, στροφή, καταγραφή), β) τα αντικείμενα (objects), οι οντότητες δηλαδή πάνω στις οποίες επιδρούν οι πράξεις (π.χ. μορφές ηρώων) και γ) οι τοποθεσίες, ο χώρος (σκηνή) δηλαδή όπου διεξάγονται οι πράξεις. Κάθε πτυχή μιας εντολής προγραμματισμού μπορεί να περιγραφεί ως μια πράξη που εφαρμόζεται σε ένα αντικείμενο σε ένα συγκεκριμένο χώρο (Fay & Mayer, 1988). Η σημασιολογία των εντολών των γλωσσών που μελετώνται δεν είναι πάντα διαισθητική ούτε πάντα γνωστικά κατάλληλη. Τα εικονίδια που αναπαριστούν τις εντολές έχουν σχεδιαστεί ώστε να υπονοούν τις πράξεις που η εντολή θα εκτελέσει. Αυτό όμως δε συμβαίνει σε πολλές περιπτώσεις. Εξάλλου, η οικοδόμηση μιας εντολής δεν μπορεί να πραγματοποιηθεί παρά μόνο εάν η εντολή εφαρμοστεί πάνω σε αντικείμενο που επιδρά και στο πλαίσιο προβλημάτων που έχουν νόημα για τα παιδιά. Για το λόγο αυτό, είναι απαραίτητη η κατάλληλη διδακτική προσέγγιση των προς μελέτη γλωσσών προγραμματισμού, υπό το πρίσμα της Διδακτικής της Πληροφορικής.

Οπτικός προγραμματισμός

Όλες οι γλώσσες διαθέτουν αναπτυξιακά κατάλληλη διεπιφάνεια και ιδιαίτερες λειτουργίες που επιτρέπουν στους μαθητές την εύκολη δημιουργία προγραμμάτων. Το σημαντικό παιδαγωγικό χαρακτηριστικό των γλωσσών αυτών είναι ο οπτικός προγραμματισμός, που γίνεται με χρήση εντολών διαφορετικών σχημάτων, συνήθως σε μορφή κομματιών παζλ που συνδυάζονται κατάλληλα μόνο με συντακτικά ορθούς τρόπους. Έτσι αποκλείονται τα συντακτικά λάθη, τα οποία αποθαρρύνουν τους μαθητές κατά τη δημιουργία κώδικα. Στη ScratchJr, η οποία είναι η πιο σύνθετη από όλες τις γλώσσες, όλα τα αλληλεπιδραστικά αντικείμενα (μορφές, σκηνές, ήχοι) μπορούν να εισαχθούν με απλό τρόπο στο πρόγραμμα και να συνδυαστούν ποικιλοτρόπως. Έτσι οι αρχάριοι προγραμματιστές μπορούν να έχουν γρήγορα αποτελέσματα και αποκτούν κίνητρο για να προσπαθήσουν περαιτέρω. Το γραφικό περιβάλλον βασίζεται σε ένα λεξιλόγιο εικόνων (οπτικές εντολές) και η λειτουργία γίνεται με κινήσεις «σύρε κι άφησε» επιτρέποντας έτσι σε παιδιά που δε γνωρίζουν ακόμα ανάγνωση και γραφή να εισαχθούν σε βασικές έννοιες της Πληροφορικής και του προγραμματισμού.

Εντολή / παράμετρος

Οι εντολές έχουν εικονική μορφή, όπου η εικόνα προσπαθεί να έχει σημασιολογική αναφορά στον τρόπο λειτουργίας της εντολής. Κάποιες φορές η εικόνα είναι εύκολα κατανοητή (π.χ. εντολές κίνησης, ηχογράφηση) ενώ άλλες φορές είναι εντελώς δυσνόητη για τα παιδιά (π.χ. η εντολή «Διαδικασία» (P1) στο Lightbot Jr και η εντολή «Repeat» στο ScratchJr). Όμως η σημασιολογία δεν είναι πάντα η ίδια σε παρόμοιες εντολές στις επιμέρους γλώσσες. Για παράδειγμα, οι εντολές κίνησης στο ScratchJr έχουν βήμα με παράμετρο, στο The Foos οι εντολές έχουν σταθερό βήμα και στο kodable η κίνηση

εξελίσσεται μέχρι το αντικείμενο να συναντήσει κάποιο εμπόδιο. Συνεπώς, η ορθή χρήση της εντολής απαιτεί κάθε φορά τη δημιουργία κατάλληλου νοητικού μοντέλου λειτουργίας της και δεν επαρκεί η εικονική μορφή για την κατανόησή της. Σε όσες γλώσσες, όπως τη ScratchJr και τη Move the Turtle, υποστηρίζεται η παράμετρος, η εντολή είναι ήδη αρχικοποιημένη για λόγους γνωστικής ευκολίας, γεγονός που οδηγεί σε διδακτικό πρόβλημα αν πρέπει να αλλαχθεί η παράμετρος.
Οι εντολές χωρίζονται σε δύο μεγάλες κατηγορίες διαφορετικής γνωστικής δυσκολίας: εντολές δράσης και εντολές ελέγχου (Olabe et al., 2015). Οι εντολές δράσης (π.χ. κίνηση, εμφάνιση) είναι εντολές που θέτουν σε λειτουργία τις δράσεις των μορφών ή των αντικειμένων του προγράμματος ενώ οι εντολές ελέγχου (δομές ελέγχου ή εκκίνηση δράσεων) καθορίζουν πότε οι δράσεις θα εκτελεστούν. Στην περίπτωση των εντολών ελέγχου είναι δυσκολότερη η οικοδόμηση κατάλληλων νοητικών μοντέλων σε σχέση με τις εντολές δράσης. Ειδικότερα, στον προγραμματισμό χειρισμού γεγονότων (όπως στη ScratchJr) κάποιες δράσεις δεν θα εκτελεστούν παρά μόνο εάν πληρωθούν κάποιες συνθήκες ελέγχου.

Βασικές δομές προγραμματισμού: Ακολουθία - Επανάληψη - Επιλογή
Η ακολουθιακή δομή ενυπάρχει σε όλα τα περιβάλλοντα αυτής της κατηγορίας. Εξάλλου, η ακολουθία αποτελεί την πιο βασική δομή κατά τη δημιουργία ενός προγράμματος. Ως γενικότερη έννοια αναπτύσσεται κατά την προσχολική ηλικία, και ο προγραμματισμός μπορεί να ευνοήσει αισθητά την οικοδόμησή της. Η απλή δομή επανάληψης, υποστηρίζεται σε όλα τα περιβάλλοντα ενώ το ScratchJr υποστηρίζει και την ατέρμονη επανάληψη. Η δομή ελέγχου (συνθήκη) υποστηρίζεται με διάφορους τρόπους αλλά το ScratchJr δεν υπάρχει με την έννοια της επιλογής. Όταν η επιλογή υποστηρίζεται από το περιβάλλον, αυτή εδράζεται πάνω σε χωρική ή χρωματική μεταφορά. Για παράδειγμα, η επιλογή ενός συνόλου εντολών γίνεται είτε στη διακλάδωση δύο διαδρομών είτε με βάση το χρώμα των εντολών.

Σύνοψη: προς σύγχρονες έννοιες προγραμματισμού και ρομποτικής

Στο άρθρο μελετήθηκαν τα περιβάλλοντα ρομποτικής και προγραμματισμού που μπορούν να χρησιμοποιηθούν στις πιο χαμηλές βαθμίδες της εκπαίδευσης με στόχο την ανάπτυξη ικανοτήτων υπολογιστικής σκέψης. Έγινε μια σύντομη διδακτική και γνωστική ανάλυση δεκαπέντε περιβαλλόντων (προγραμματιζόμενα ρομπότ, κιτ ρομποτικής και περιβάλλοντα οπτικού προγραμματισμού). Τα περιβάλλοντα αυτά έχουν ποικίλα χαρακτηριστικά, με κύριο γνώρισμα τη δυνατότητα να υποστηρίξουν βασικές ή και πιο σύνθετες έννοιες προγραμματισμού εάν συνοδεύονται με κατάλληλη διδακτική προσέγγιση. Ειδικότερα, κάποια περιβάλλοντα όπως το Thymio και η ScartchJr προωθούν σύγχρονες προγραμματιστικές έννοιες, οι οποίες αξίζει να μελετηθούν στην προσχολική και την πρώτη σχολική ηλικία. Η ScartchJr (Portelance et al., 2015) εγγράφεται στην παράδοση της Logo και συμπεριλαμβάνει πολλές σημασιολογικές λειτουργίες του προστακτικού προγραμματισμού ενώ ταυτόχρονα υποστηρίζει και άλλα προγραμματιστικά παραδείγματα: ο προγραμματισμός πολλών πρακτόρων (multi-agent programming) αφού μπορούν να προγραμματιστούν ταυτόχρονα πολλές μορφές (sprites) και να αλληλοεπιδράσουν μεταξύ τους με συγχρονισμό (concurrent programming). Στοιχεία συγχρονισμού επιτρέπουν και άλλες γλώσσες, όπως η Bit by Bit. Η ScratchJr είναι η μόνη που εισάγει στοιχεία οντοκεντρικού προγραμματισμού αφού ο χρήστης μπορεί να ορίσει πολλές μορφές που αλληλεπιδρούν ανταλλάσσοντας μηνύματα ή δημιουργώντας συμβάντα. Δύο μορφές στη ScratchJr μπορούν να επικοινωνήσουν άμεσα (ανταλλάσσοντας μηνύματα) ή έμμεσα (σε περίπτωση που η μία αγγίξει την άλλη). Στο πλαίσιο αυτό έχουμε

προγραμματισμό με συμβάντα, όπως και στην περίπτωση του ρομπότ Thymio. Είναι προφανές ότι η ερευνητική αυτή περιοχή απαιτεί περισσότερες έρευνες, είτε στο επίπεδο της θεωρητικής προσέγγισης, είτε στο επίπεδο της ανάπτυξης νέων εργαλείων, είτε στο επίπεδο της εφαρμογής στην τάξη.

Αναφορές

Fay, A. L., & Mayer, R. E. (1988). Learning Logo: A cognitive analysis. In R. E. Mayer (eds.), *Teaching and learning computer programming: Multiple researcher perspectives* (pp. 55-74), Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Manches, A., & Plowman, L. (2015). Computing education in children's early years: A call for debate. *British Journal of Educational Technology*. doi: 10.1111/bjet.12355.

Misirli, A., & Komis, V. (2014). Robotics and Programming Concepts in Early Childhood Education: A Conceptual Framework for Designing Educational Scenarios. In C. Karagiannidis, P. Politis, & I. Karasavvidis (eds.), *Research on e-Learning and ICT in Education* (pp. 99–118), New York, NY: Springer.

Olabe, J. C., Basogain, X., & Olabe, M. (2015). HelloScratchJr.org: Curricular Design and Assessment Tools to Foster the Integration of ScratchJr and Computational Thinking into K-2 Classrooms. In *Proceedings of 7th international Scratch conference* (pp. 31-41), Amsterdam.

Portelance, D. J., Strawhacker, A. L., & Bers, M. U. (2015). Constructing the ScratchJr programming language in the early childhood classroom. *International Journal of Technology and Design Education*, 1-16. http://doi.org/10.1007/s10798-015-9325-0.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

Μισιρλή, Α. (2015). Η ανάπτυξη ικανοτήτων αλγοριθμικής σκέψης και προγραμματισμού σε παιδιά προσχολικής ηλικίας με τη χρήση προγραμματιζόμενων ρομπότ (Μη δημοσιευμένη διδακτορική διατριβή), Πανεπιστήμιο Πατρών.