

Διδασκαλία Προγραμματισμού και Μαθηματικών στο Δημοτικό Σχολείο

Εισαγωγή

Ο Seymour Papert εισήγαγε τον όρο *Υπολογιστική Σκέψη* ΥΣ (Computational Thinking, CT) στο βιβλίο του *Mindstorms* του 1980: *Παιδιά, υπολογιστές και ισχυρές ιδέες* (Children, computers, and powerful ideas). Σε αυτό, υποστήριξε ότι οι μαθητές μαθαίνουν πιο αποτελεσματικά όταν ασχολούνται με την κατασκευή από κάτι που έχει νόημα για αυτούς και το οποίο μπορούν να μοιραστούν. Ως μαθηματικός, ο Papert ενδιαφέρθηκε ιδιαίτερα για τη διδασκαλία των μαθηματικών και οραματίστηκε ένα μαθησιακό περιβάλλον (το οποίο ανέφερε ως «Mathland») το οποίο οι μαθητές θα μπορούσαν να χρησιμοποιήσουν για να διερευνήσουν αφηρημένες μαθηματικές έννοιες με έναν πιο συγκεκριμένο τρόπο. Αυτή η ιδέα τον οδήγησε να αναπτύξει τη γλώσσα προγραμματισμού Logo, την οποία περιέγραψε ως «ένα εργαλείο σχεδιασμένο για να αλλάξει τον τρόπο που μιλάει, σκέφτεται και γράφει κανείς τα μαθηματικά και τη σχέση μεταξύ τους» (Papert 1990, σ. 7). Ο Papert πίστευε ότι οι μαθητές που προγραμματίζουν στην Logo θα είναι σε θέση να αναπτύξουν την κατανόησή τους στην εκμάθηση και στη σκέψη μέσα από τη διαδικασία της δοκιμής και της διόρθωσης των λαθών των ιδεών τους στον κώδικα. Αυτό το όραμα επέδρασε σε ορισμένους δασκάλους και ερευνητές, που είδαν τη Logo και τη φιλοσοφία της εκπαίδευσης του Papert ως μια «εναλλακτική λύση στις επικρατούσες τεχνοκεντρικές και συμπεριφοριστικές έννοιες της εκπαίδευσης της καθοδηγούμενης από υπολογιστή» (Agalianos, Whitty, & Noss, 2006, σελ. 241), έννοιες που ήταν κοινές στη δεκαετία του '80.

Κατά τη διάρκεια αυτής της περιόδου, ορισμένοι εκπαιδευτικοί εμπνεύστηκαν από το βιβλίο *Mindstorms* του Papert για να αλλάξουν τις διδακτικές προσεγγίσεις τους και σύντομα ο προγραμματισμός Logo έγινε μέρος των εθνικών εκπαιδευτικών μεταρρυθμίσεων τόσο στις ΗΠΑ όσο και στο Ηνωμένο Βασίλειο (Agalianos et al., 2006).

Αρχικά, ορισμένες τέτοιες αλλαγές ήταν στο επίπεδο του απλού κόσμου, αλλά τα επόμενα χρόνια έγινε μέρος εθνικών εκπαιδευτικών μεταρρυθμίσεων και στις δύο χώρες. Οι μικροϋπολογιστές εισήχθησαν επίσης σε πολλές αίθουσες διδασκαλίας στο Ηνωμένο Βασίλειο κατά τη διάρκεια της δεκαετίας του 1980, την οποία όπως υποστήριζαν ο Agalianos και οι συνεργάτες του, σε μεγάλο βαθμό παρακινήθηκε, από τους βιομηχάνους και όχι από ένα πνεύμα πραγματικής εκπαιδευτικής μεταρρύθμισης. Η άποψη ότι οι υπολογιστές θα μπορούσαν να χρησιμοποιηθούν από τους μαθητές για να διερευνήσουν μαθηματικές ιδέες όπως το οραματίστηκε ο Papert ήταν σε μεγάλο βαθμό απύσχα. Επιπλέον, οι εκδόσεις της Logo που ήταν συμβατές με τον πιο συνηθισμένο μικροϋπολογιστή που χρησιμοποιούταν στις αίθουσες διδασκαλίας (το BBC Micro) ήταν δύσκολο να εγκατασταθούν και είχαν περιορισμένη λειτουργικότητα. Τελικά, η Logo "απογυμνώθηκε από τις ριζοσπαστικές της δυνατότητες" (σελ. 241), και για πολλούς εκπαιδευτικούς η Logo έγινε συνώνυμο με τα «γραφικά της χελώνας». Κατά συνέπεια, τόσο στο Ηνωμένο Βασίλειο όσο και στις ΗΠΑ, οι εκπαιδευτικοί χρησιμοποίησαν κυρίως τη Logo για δραστηριότητες που θα έπρεπε να πραγματοποιηθούν στην τάξη και όχι ως περιβάλλον στο οποίο οι μαθητές θα μπορούσαν να αναπτύξουν τη σκέψη τους.

Σχεδόν τριάντα χρόνια μετά τη δημοσίευση του *Mindstorms*, η Jeanette Wing (2006) έγραψε ένα άρθρο για την υπολογιστική σκέψη, την οποία χαρακτήρισε ως “σκέφτομαι όπως ένας επιστήμονας πληροφορικής” (σελ. 35). Υποστήριξε ότι η υπολογιστική σκέψη ήταν μια δεξιότητα που θα μπορούσε να ωφελήσει όλους, όχι μόνο τους επιστήμονες υπολογιστών, και ότι θα έπρεπε να διδάσκεται παράλληλα με “την ανάγνωση, τη γραφή και την αριθμητική” (σελ. 33). Κατά τη δεκαετία που ακολούθησε το άρθρο της, η διδασκαλία της επιστήμης των υπολογιστών (Computer Science, CS) στο σχολείο έγινε πιο διαδεδομένη και άρχισε να εισάγεται στα προγράμματα σπουδών της υποχρεωτικής K-12 εκπαίδευσης (Wing 2016). Στις χώρες που έχουν πρόσφατα εισαγάγει τέτοια προγράμματα σπουδών, με έμφαση στην υπολογιστική σκέψη, περιλαμβάνεται η Αυστραλία (Falkner, Vivian, & Falkner, 2014), η Αγγλία (Brown, Sentance, Crick, & Humphreys, 2014), οι Ηνωμένες Πολιτείες (Fisher, 2016) και η Νέα Ζηλανδία (Bell, Newton, Andreae, & Robins, 2012a).

Ένας πιο πρόσφατος ορισμός για την υπολογιστική σκέψη δίνεται από τους Shute, Sun, & Asbell-Clarke (2017): “η εννοιολογική βάση που απαιτείται για την ενεργητική και αποτελεσματική επίλυση προβλημάτων (δηλαδή, αλγοριθμικά, με ή χωρίς τη βοήθεια υπολογιστών) με λύσεις που μπορούν να επαναχρησιμοποιηθούν σε διαφορετικά περιβάλλοντα” (Shute et al., 2017, σελ.142).

Αυτός ο ορισμός υπογραμμίζει ότι η ΥΣ είναι πρωτίστως ένας τρόπος σκέψης και δράσης, ο οποίος μπορεί να εκδηλωθεί μέσω της χρήσης συγκεκριμένων δεξιοτήτων, οι οποίες στη συνέχεια μπορούν να γίνουν η βάση για αξιολογήσεις που βασίζονται στις επιδόσεις των δεξιοτήτων της ΥΣ.

Ο προγραμματισμός και η υπολογιστική σκέψη θεωρούνται γενικά ξεχωριστές δεξιότητες, αλλά ο προγραμματισμός απαιτεί τη χρήση της υπολογιστικής σκέψης και συχνά χρησιμοποιείται για να την διδάξει (Lye and Koh, 2014). Ο προγραμματισμός είναι η πράξη της σύνταξης κώδικα που καθοδηγεί τον υπολογιστή να εκτελέσει κάποιες ενέργειες, ενώ η υπολογιστική σκέψη είναι μια «μεθοδολογία επίλυσης προβλημάτων» (Barr and Stephenson, 2011, σελ. 48). Η διδασκαλία της υπολογιστικής σκέψης δεν απαιτεί απαραίτητως από τους μαθητές να δημιουργούν προγράμματα: για παράδειγμα, το CS Unplugged (Bell, Rosamond, & Casey, 2012b) περιλαμβάνει ένα σύνολο πόρων που αναπτύχθηκαν για να διδάξουν αυτή την ικανότητα χωρίς τη χρήση υπολογιστή.

Η εισαγωγή του προγραμματισμού και της υπολογιστικής σκέψης στις σχολικές τάξεις K-12 παρουσίασε μερικές προκλήσεις για τους εκπαιδευτικούς, καθώς η εκμάθηση αυτών των δεξιοτήτων δεν ήταν συνήθως μέρος της επίσημης εκπαίδευσης του εκπαιδευτικού. Αυτό ισχύει ιδιαίτερα για τους δασκάλους της πρωτοβάθμιας εκπαίδευσης, καθώς είναι απίθανο να έχουν ολοκληρώσει έναν τεχνολογικό κύκλο και συνήθως είναι εκπαιδευτικοί γενικής εκπαίδευσης (Vivian Falkner, K., & Falkner, N., 2014). Μια από τις προτεινόμενες προσεγγίσεις για την προετοιμασία των εκπαιδευτικών για μια τέτοια διδασκαλία είναι η ενσωμάτωση της υπολογιστικής σκέψης στα θέματα που διδάσκουν επί του παρόντος, για παράδειγμα, τα μαθηματικά (Barr and Stephenson, 2011).

I. Ιστορία του προγραμματισμού και της μάθησης των μαθηματικών για τα παιδιά

Κατά τη διάρκεια της δεκαετίας του 1970 και του '80, τα ερευνητικά ενδιαφέροντα εστιάζονταν βαθμιαία στην εκμάθηση των μαθηματικών μέσω γραπτών αλγορίθμων και στο δυναμικό πλαίσιο για την εκμάθηση των μαθηματικών που θα μπορούσε να προσφέρει ο προγραμματισμός της Logo (Noss and Hoyles 1996). Οι πρώτες έρευνες υπογράμμισαν επίσης τη δυνατότητα έκφρασης των μαθηματικών ιδεών ως προγραμμάτων ηλεκτρονικών

υπολογιστών (Feurzeig et al. 1969; Hatfield and Kieren 1972). Οι Noss και Hoyles (1996, σελ. 55) υποστήριζαν ότι η σύνταξη ενός προγράμματος ηλεκτρονικού υπολογιστή "παρέχει ένα ευρύ πλαίσιο στο οποίο ο μαθητής μπορεί να σχεδιάσει ημι-κατανοητές ιδέες και να συγκεντρώσει στην οθόνη μια ημι-συγκεκριμένη εικόνα των μαθηματικών δομών που αυτός κτίζει διανοητικά". Πολλές αλλά όχι όλες αυτές οι μελέτες ακολούθησαν ένα κατασκευαστικό παράδειγμα, στο οποίο η μάθηση θεωρείται ως κατασκευή δομών γνώσης και συμβαίνει πιο αποτελεσματικά όταν ο εκπαιδευόμενος ασχολείται ενεργά με την "οικοδόμηση μιας κοινής οντότητας" (Papert and Harel 1991). Ακολουθώντας αυτή τη λογική, ο Papert (1980, 1993), για παράδειγμα, διευκρίνιζε πώς οι κατάλληλα σχεδιασμένες εμπειρίες της Logo θα μπορούσαν να τονώσουν τη γνωστική ανάπτυξη των παιδιών και να τους προσφέρουν την ευκαιρία να εξερευνήσουν «ισχυρές ιδέες» που σχετίζονται με τα μαθηματικά. Ανέφερε πολλά πιθανά οφέλη από τον προγραμματισμό της Logo από την εκμάθηση των μαθηματικών που περιλάμβαναν συγκεκριμένες έννοιες όπως γωνίες, μοίρες και μεταβλητές, καθώς και γενικότερες στρατηγικές επίλυσης προβλημάτων όπως η ανάλυση και ο εντοπισμός σφαλμάτων (Miller Kelly, & Kelly, 1988).

Οι Clements και Sarama (1997) τόνισαν επίσης τις ευκαιρίες που θα μπορούσε να παράσχει στα παιδιά η Logo ώστε να χειριστούν συγκεκριμένες περιπτώσεις μαθηματικών ιδεών, να διευκολύνουν τις συνδέσεις μεταξύ συγκεκριμένων εμπειριών και πιο αφηρημένων μαθηματικών, καθώς και να επιτρέψουν στα παιδιά να χρησιμοποιούν τα μαθηματικά με τρόπους που έχουν μεγαλύτερη προσωπική σημασία για αυτούς.

Ωστόσο, τα ευρήματα από την έρευνα σχετικά με τον "αντίκτυπο" του προγραμματισμού ηλεκτρονικών υπολογιστών στην εκμάθηση των μαθηματικών των παιδιών ήταν ασαφή. Η ποικιλομορφία των ερευνητικών παραδειγμάτων, οι πειραματικές συνθήκες και τα απογοητευτικά χαμηλά μέτρα "μεταφοράς" καθιστούν δύσκολη τη σύγκριση. Επιπλέον, το ερώτημα για την επίπτωση του προγραμματισμού αγνοεί την πολυπλοκότητα των ζητημάτων που διακυβεύονται και την πλούσια αλληλεπίδραση του πλαισίου. Επιπλέον, μεγάλο μέρος αυτής της πρώιμης έρευνας επικεντρώθηκε σε εξωσχολικές ή επιλεκτικές ρυθμίσεις και όχι σε τυπικά περιβάλλοντα της τάξης, γεγονός που θα απαιτούσε να ληφθούν υπόψη περισσότερο τα προγράμματα σπουδών και οι εκπαιδευτικοί (Lye and Koh 2014). Υπό το πρίσμα των ανησυχιών σχετικά με την εφαρμοσιμότητα των αποτελεσμάτων της έρευνας στον ευρύτερο πληθυσμό, πολλοί ερευνητές σε αυτόν τον τομέα χρησιμοποίησαν μεθόδους έρευνας σχεδιασμού, απαιτώντας μια σαφή θεωρητική βάση από την οποία τα αποτελέσματα θα μπορούσαν να είναι γενικεύσιμα (Prediger, Gravemeijer, & Confrey, 2015).

Σε αντίθεση με ένα μεγάλο μέρος αυτής της πρώιμης έρευνας, ένας βασικός στόχος του προγράμματος ScratchMaths, το οποίο θα παρουσιάσουμε στη συνέχεια, ήταν ότι η παρέμβαση θα ήταν προσιτή σε όλους τους μαθητές ανεξάρτητα το υπόβαθρό τους, το φύλο τους ή το επίπεδο επίδοσης.

Οι ερευνητές προσπάθησαν να παράσχουν εξηγήσεις για τη διακύμανση των αποτελεσμάτων σε διάφορες μελέτες. Αυτές περιλαμβάνουν τον πιθανό αντίκτυπο που μπορεί να έχουν οι υπάρχουσες γνώσεις και ικανότητες των παιδιών καθώς και το στάδιο ανάπτυξης, με τον Clements (1985, σελ. 59) να ισχυρίζεται ότι για να μάθουν τη Logo "απαιτούνται ορισμένες ικανότητες μαθηματικές, χωρικές και επίλυσης προβλήματος" και οι οποιεσδήποτε πιθανές επιδράσεις χρειάζονται χρόνο για να αναδυθούν (Clements and Sarama 1997). Επιπλέον, οι Clements και Sarama (1997) προτείνουν την ανάγκη για ειδικά σχεδιασμένες και διαδοχικές δραστηριότητες, με τους Hoyles και Noss (1992) να εκφράζουν ανησυχίες για τα παιδιά τα οποία παρακάμπτουν τις μαθηματικές ιδέες μέσα στις λιγότερο

δομημένες μαθησιακές δραστηριότητες και χωρίς την καθοδήγηση από τους εκπαιδευτικούς (βλέπε, Clements 1999).

Βασικά οι Salomon και Perkins (1989) υποδηλώνουν ότι τα προγράμματα προγραμματισμού της Logo που έδειξαν πιο θετικές επιδράσεις στη “μεταφορά” προωθούσαν την προσεκτική αφαίρεση των μαθησιακών αποτελεσμάτων. Αυτό σχετίζεται βέβαια με το ρόλο του εκπαιδευτικού και πολλοί ερευνητές υπογραμμίζουν ότι αυτό είναι κρίσιμο, όχι μόνο για να κάνουν σαφείς τους συστηματικούς δεσμούς με τις υπάρχουσες μαθηματικές γνώσεις και εμπειρίες (Clements 1985, 1999; Clements and Sarama 1997) αλλά και για να διευκολύνουν την ανακάλυψη και κατανόηση στους μαθητές τους των ισχυρών ιδεών που συναντούν σε ένα περιβάλλον προγραμματισμού όπως η Logo (Yelland, 1995; McCoy 1996).

Πολλά νέα περιβάλλοντα προγραμματισμού έχουν αναπτυχθεί, καθώς πραγματοποιήθηκε μεγάλο μέρος της έρευνας στον προγραμματισμό και στη μάθηση των μαθηματικών: μερικά από αυτά είχαν εύλογα ευρεία διάδοση (Duncan et al., 2014). Μερικά έχουν σχεδιαστεί ειδικά για τους αρχάριους και τους νέους, πολλά από τα οποία είναι βασισμένα σε μπλοκ (Duncan, Bell, & Tanimoto, 2014; Weintrop and Wilensky 2015). Ένα από τα πιο δημοφιλή από αυτά είναι το Scratch (που χρησιμοποιείται από εκατομμύρια παιδιά παγκοσμίως, συνήθως σε εξωσχολικά περιβάλλοντα), που αναπτύχθηκε στο MIT και είναι απόγονος της Logo (Resnick et al., 2009). Οι Resnick et al. (2009) δηλώνουν ότι το Scratch παρέχει ευκαιρίες για εκμάθηση σημαντικών μαθηματικών και υπολογιστικών εννοιών, καθώς είναι χώρος δημιουργικής σκέψης, συστηματικής συλλογιστικής και συνεργατικής εργασίας.

II. Το σημερινό πρόγραμμα σπουδών του προγραμματισμού στην πρωτοβάθμια

Ο προγραμματισμός εντάσσεται στο Διαθεματικό Ενιαίο Πλαίσιο Προγράμματος Σπουδών Πληροφορικής στο δημοτικό και ειδικότερα διδάσκεται στις δύο τελευταίες τάξεις. Στόχος του μαθήματος είναι οι μαθητές να κατανοήσουν ότι ο υπολογιστής εκτελεί τις οδηγίες που δέχεται από τον χρήστη, οι οποίες εισάγονται σε κωδικοποιημένη μορφή. Τα παιδιά, χρησιμοποιώντας μία απλή γλώσσα προγραμματισμού (Logo-like περιβάλλοντα είναι τα μόνα που προτείνονται), μαθαίνουν να χρησιμοποιούν απλές εντολές με στόχο να δημιουργήσουν σχήματα ή να λύσουν κάποιο απλό πρόβλημα.

Σε γενικές γραμμές, η θεωρητική βάση του προγράμματος σπουδών μπορεί να χαρακτηριστεί ως φτωχή με βάση τα σύγχρονα δεδομένα. Μάλιστα, η αποτελεσματική διδασκαλία προϋποθέτει καταρτισμένο προσωπικό σε θέματα Πληροφορικής και προγραμματισμού, κατάλληλη υλικοτεχνική υποδομή, συνεχείς ενημερώσεις και βελτιώσεις του λογισμικού το οποίο θα χρησιμοποιηθεί, στοιχεία που σε αρκετές περιπτώσεις απουσιάζουν. Επιπρόσθετα, υπάρχουν και αρκετά άλλα προβλήματα στη διδασκαλία του προγραμματισμού στο δημοτικό σχολείο. Συνήθως ακολουθείται η παραδοσιακή προσέγγιση διδασκαλίας που στηρίζεται στην επίλυση μαθηματικών προβλημάτων και στη χρήση αριθμών και συμβόλων. Αυτή η προσέγγιση δεν μπορεί να ικανοποιήσει τις ανάγκες των μαθητών και είναι αναποτελεσματική στο να μεταφέρει τις αρχές του προγραμματισμού (Ξυνόγαλος, Σατρατζέμη, & Δαγδιδέλης, 2000). Όπως αναφέρεται από τον Κόμη (2005), κατά τη διάρκεια της διδασκαλίας του προγραμματισμού, οι μαθητές καλούνται να κατανοήσουν τις αρχές του προγραμματισμού και τη διαδικασία εισόδου και εξόδου των δεδομένων, χωρίς τη χρήση κατάλληλων διαγραμμάτων και οπτικού υλικού. Επίσης, οι μαθητές καλούνται να

προσεγγίσουν προβλήματα τα οποία τους είναι ξένα σε σχέση με τα προβλήματα που συναντούν στην καθημερινότητά τους.

Το Αναλυτικό Πρόγραμμα για τις ΤΠΕ προτείνει τη διδασκαλία του προγραμματισμού τόσο στην Ε΄ όσο και στην ΣΤ΄ τάξη του Δημοτικού, αναδεικνύοντας με τον τρόπο αυτό τη σημασία της Πληροφορικής και του προγραμματισμού στο σύγχρονο εκπαιδευτικό σύστημα. Η προσέγγιση αυτή συμφωνεί με τη θεώρηση του προγραμματισμού ως βασική συνιστώσα για την ανάπτυξη πληροφοριακού γραμματισμού και την εξίσωση της σημαντικότητάς του με αυτή της γραφής, της ανάγνωσης και των μαθηματικών ως θεμελιώδεις λίθους για τη γνωστική ανάπτυξη του ατόμου (Kelleher, 2012).

Ειδικότερα, η διδακτική ενότητα “Ελέγγω και Προγραμματίζω” που εντάσσεται στον άξονα “Διερευνώ, ανακαλύπτω και λύνω προβλήματα με ΤΠΕ” προτείνεται για τις δύο τελευταίες τάξεις, με τη διαφορά πως οι στόχοι που αναφέρονται για την Ε΄ τάξη αφορούν περισσότερο την εξοικείωση των μαθητών με το γραφικό περιβάλλον, με την κίνηση ενός αντικειμένου και τέλος με τον σχεδιασμό απλών γεωμετρικών σχημάτων. Αντίθετα, οι στόχοι που αναφέρονται για την ΣΤ΄ τάξη οδηγούν το μαθητή να μάθει βασικές έννοιες του προγραμματισμού όπως τι είναι μεταβλητή, τι ορίζουμε ως διαδικασία, τι είναι επανάληψη αλλά και πόσο χρήσιμες είναι όλες αυτές οι έννοιες για όποιον θέλει να ασχοληθεί με την επίλυση προβλημάτων και τον προγραμματισμό.

Πιο συγκεκριμένα, στην Ε΄ τάξη στόχος του νέου προγράμματος σπουδών είναι οι μαθητές (ΔΕΠΠΣ 2003):

- Να γνωρίσουν τις βασικές αρχές του προγραμματισμού και τις ιδιότητες ενός αλγορίθμου μέσα από το παιχνίδι: να σχεδιάζουν και να επεξεργάζονται δικές τους ζωγραφιές σε ένα Logo προγραμματιστικό περιβάλλον, να εισάγουν και να επεξεργάζονται έτοιμα σχήματα, να εισάγουν έτοιμες εικόνες και ζωγραφιές, να εισάγουν τη χελώνα και να αλλάζουν τις διαστάσεις της, να αλλάζουν τη μορφή της χελώνας χρησιμοποιώντας τα έτοιμα σχήματα, να γνωρίζουν τις βασικές εντολές κίνησης της χελώνας (μπροστά, πίσω δεξιά, αριστερά, περίμενε κ.λπ.) , να σχεδιάζουν απλά γεωμετρικά σχήματα (τετράγωνο, παραλληλόγραμμο, κ.ά.) με τη γραφίδα της χελώνας.

Στην ΣΤ΄ τάξη, στόχος είναι οι μαθητές:

- να εφαρμόζουν τις γνώσεις που έχουν και να υλοποιούν απλά προγράμματα σε ένα Logo προγραμματιστικό περιβάλλον: να κατανοούν την έννοια της μεταβλητής, να χρησιμοποιούν διαδικασίες με συνθήκη, παραμετρικές και αναδρομικές διαδικασίες.

Στόχος της όλης διαδικασίας είναι οι μαθητές να αποκτήσουν αναλυτική και συνθετική σκέψη, να εξοικειωθούν με τεχνικές διόρθωσης σφαλμάτων (debugging) και βελτιστοποίησης προγραμμάτων ώστε να καταστούν ικανοί να δημιουργούν οι ίδιοι σύνθετα έργα βασιζόμενοι στη σύνθεση απλούστερων μερών σε ένα περιβάλλον οπτικού προγραμματισμού. Τα περιβάλλοντα οπτικού προγραμματισμού που προτείνονται είναι τα εξής: EasyLogo, Scratch, BYOB, Kodu, MicroWorldsPro, GameMaker, K-Turtle, TurtleArt, openStarlogo και Εκπαιδευτική Ρομποτική (Μπόκος, 2014).

III. Το έργο ScratchMaths

Οι Benton, Hoyles, Kalas, & Noss, (2016, 2017) δημιούργησαν το έργο ScratchMaths (SM), το οποίο περιλαμβάνει υλικά προγράμματος σπουδών και επαγγελματική ανάπτυξη (PD) για την υποστήριξη της μαθηματικής μάθησης μέσω του προγραμματισμού για μαθητές ηλικίας 9 έως 11 ετών. Το έργο πλαισιώθηκε από το ιδιαίτερο πλαίσιο της πληροφορικής στο αγγλικό εκπαιδευτικό σύστημα παράλληλα με τη μακρά ιστορία της έρευνας και την ανάπτυξη στον προγραμματισμό και τα μαθηματικά. Το έργο αυτό είναι σύμφωνο με ένα «πλαίσιο δράσης» (diSessa and Cobb, *Journal of the Learning Sciences*, 13, 77-103, 2004) μετά από έρευνα σχεδιασμού με στόχο να αναπτύξει μια τεκμηριωμένη παρέμβαση προγράμματος σπουδών γύρω από προσεκτικά επιλεγμένες μαθηματικές και υπολογιστικές έννοιες. Εστιάζει σε δύο βασικές θεμελιώδεις έννοιες μέσα στο πρόγραμμα σπουδών του SM: την έννοια του αλγορίθμου και της ολόκληρης στροφής των 360°. Διαπιστώνετε ότι η παρέμβασή στο σύνολό της επέτρεψε στους εκπαιδευτικούς με διαφορετικό υπόβαθρο και επίπεδα εμπιστοσύνης να προσαρμόσουν την παράδοση του SM με τρόπους που μπορούν να καταστήσουν πιο προσιτές αυτές τις προκλητικές έννοιες τόσο για τους ίδιους όσο και για τους μαθητές τους.

Λόγω της μαζικής προσέγγισής του, της ευκολίας πρόσβασης και της καταλληλότητάς του για τα παιδιά, το Scratch επελέγη ως το περιβάλλον προγραμματισμού που χρησιμοποιήθηκε στο έργο SM. Επίσης, από την ανασκόπηση της προηγούμενης έρευνας δημιουργήθηκε η ανάγκη να ληφθεί σοβαρά υπόψη, πρώτα ο σχεδιασμός ενός αναλυτικού προγράμματος (και όχι μεμονωμένες δραστηριότητες) και δεύτερον, ο ρόλος του εκπαιδευτικού και ο τρόπος με τον οποίο μπορούν να υποστηριχθούν οι εκπαιδευτικοί. Υπήρξε σημαντική επένδυση για τη στήριξη της εισαγωγής του νέου νόμιμου αναλυτικού προγράμματος σπουδών στα σχολεία της Αγγλίας. Κατά την άποψη των δημιουργών του έργου, εξακολουθεί να υπάρχει επείγουσα ανάγκη υποστήριξης αυτής της πρωτοβουλίας για τη σύνδεση με άλλα θέματα προς όφελος και των δύο. Για παράδειγμα, να διερευνηθεί ο τρόπος με τον οποίο ο προγραμματισμός μπορεί να αξιοποιηθεί ως εργαλείο μοντελοποίησης σε όλο το πρόγραμμα σπουδών και να εντοπιστούν τρόποι για να φανεί η αποτελεσματικότητα αυτής της προσέγγισης. Το έργο SM προωθεί αυτό στο πλαίσιο του προγραμματισμού και των μαθηματικών και επιδιώκει να βασιστεί σε προηγούμενες έρευνες καθώς και να επωφεληθεί από τις νέες γνώσεις προγραμματισμού και την ευρεία διασύνδεση.

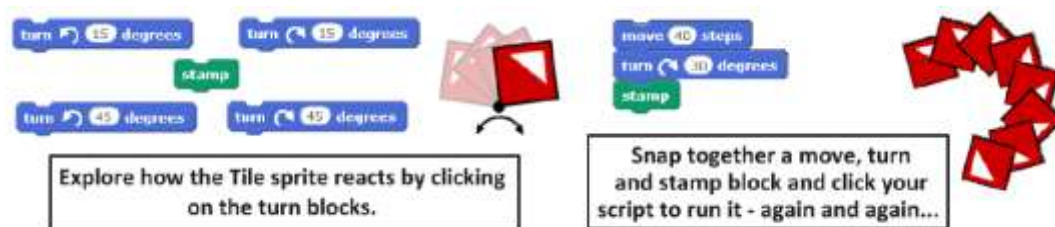
Τα 5E

Ένα βασικό αποτέλεσμα της διαδικασίας σχεδιασμού του προγράμματος σπουδών ήταν ένα «πλαίσιο δράσης» (DiSessa & Cobb 2004), το οποίο ονόμασαν οι Benton, et al (2016) "5Es". Αυτό το πλαίσιο (αποτελούμενο από πέντε ανεξάρτητα κατασκευάσματα) ήταν σαφώς πλαισιωμένο από μια σειρά από έρευνες ορθής πρακτικής στη διδασκαλία των μαθηματικών, αλλά επίσης προέκυψε από τα αρχικά εργαστήρια σχεδίασης και εξευγενίστηκε μέσω της έρευνας σχεδιασμού στα σχολεία. Έχει αναπτυχθεί για να παρέχει καθοδήγηση σχετικά με τις παιδαγωγικές στρατηγικές που μπορούν να υιοθετήσουν οι εκπαιδευτικοί για την επιτυχή εφαρμογή διαφορετικών πτυχών της παρέμβασης των SM. Το πλαίσιο αυτό περιγράφεται λεπτομερέστερα παρακάτω σε σχέση με τις δύο τάξεις που εντοπίστηκαν από την Brennan (2015).

Εξερευνώ (Explore)

Ο Papert (1980) πιστεύει ότι τα παιδιά πρέπει να χρησιμοποιούν υπολογιστές για να διερευνήσουν τις διαδικασίες σκέψης τους, προτείνοντας με σεβασμό για τη Logo ότι η βασική εμπειρία μάθησης είναι να «γνωρίσεις τη χελώνα και να διερευνήσεις τι μπορεί και δεν μπορεί να κάνει μια χελώνα». Οι κατασκευαστικές προσεγγίσεις εκτιμούν την μάθηση με αυτόν τον τρόπο μέσω δραστηριοτήτων σχεδιασμού (Brennan, 2015), οι οποίες παρέχουν ευκαιρίες να εξερευνήσουν τρόπους αντιμετώπισης διαφορετικών περιορισμών και αμφιβολιών, χρησιμοποιώντας δεξιότητες όπως η επαναληπτική σκέψη, η επίλυση προβλημάτων και η δημιουργικότητα. Επομένως, αυτή η πρώτη κατασκευή υποδηλώνει τη σημασία της ανάπτυξης και της υποστήριξης δραστηριοτήτων που επιτρέπουν στους εκπαιδευόμενους να διερευνήσουν ιδέες, να δοκιμάσουν τα πράγματα για τον εαυτό τους και να διορθώσουν τα εννοιολογικά και τεχνικά λάθη όταν είναι απαραίτητο. Μέρος αυτής της προσπάθειας είναι να μετατοπιστούν οι μαθητές προς την «ανάληψη ελέγχου της δικής τους μάθησης» και να αναζητήσουν τους λόγους πίσω από διαφορετικά αποτελέσματα.

Στις αρχές του προγράμματος σπουδών του SM οι μαθητές εισάγονται στα εργαλεία εξερεύνησης μέσα στο περιβάλλον Scratch. Για παράδειγμα, η χρήση αυτού που περιγράφει ο Blackwell (2002) ως άμεσο χειρισμό - μια ενιαία ενέργεια με ένα μόνο ορατό αποτέλεσμα. Στο πρόγραμμα σπουδών του SM ο όρος αυτός θεωρείται ότι περιλαμβάνει τον χειρισμό όλων των αντικειμένων τόσο φυσικών (δηλ. BeeBots) όσο και ψηφιακών και επομένως χρησιμοποιούμε τον όρο 'άμεση οδήγηση' για να αναφερθούμε μόνο σε ψηφιακά αντικείμενα. Η άμεση οδήγηση χρησιμοποιείται για να εξερευνήσουν πρώτα τα μπλοκ από μόνοι τους (Σχήμα 1 αριστερά), κάνοντας κλικ σε αυτά και παρατηρώντας την αντίδραση, στη συνέχεια χρησιμοποιούμε αυτά για να χτίσουμε πιο σύνθετα σενάρια (Σχήμα 1 δεξιά). Αυτή η σταδιακή εξέλιξη από την άμεση οδήγηση των μπλοκ σε σχεδιασμό και κατασκευαστικές συμπεριφορές ενσωματωμένες σε σενάρια παρέχει μια δομημένη προσέγγιση για την εξερεύνηση που ενθαρρύνεται σε όλο το πρόγραμμα σπουδών του SM.



Σχήμα 1. Παράδειγμα δραστηριότητας άμεσης οδήγησης (αριστερά) προχωρώντας στην κατασκευή απλών σεναρίων (δεξιά)

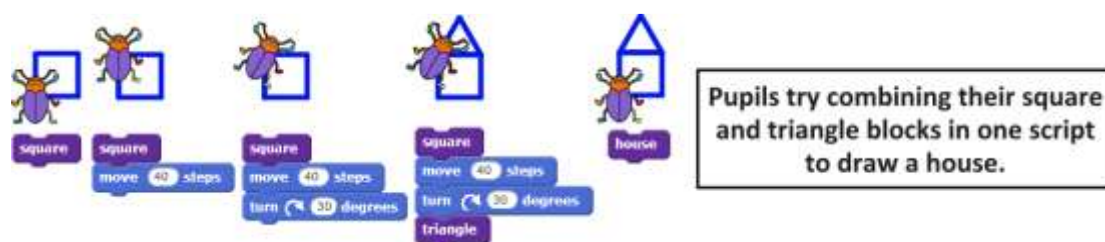
Εξηγώ (Explain)

Μια κρίσιμη πτυχή της κατανόησης των ιδεών είναι ο εκπαιδευόμενος να είναι ικανός να εξηγήσει τι έχει μάθει και να διατυπώνει τους λόγους για τους οποίους επέλεξε μια προσέγγιση (χρησιμοποιώντας διαφορετικούς τρόπους επικοινωνίας).

Αυτό βοηθάει στην αποσαφήνιση των ιδεών, εκφράζοντάς τες ρητά καθώς και απαντώντας σε ερωτήσεις από τους συνομηλίκους. Αρκετοί θεωρητικοί έχουν επισημάνει το γνωστικό πλεονέκτημα της δημιουργίας λεκτικών εξηγήσεων (Harel & Papert 1990). Για παράδειγμα, ο Brown (1988) έχει δείξει ότι η ενθάρρυνση να εξηγεί και να αναπαριστά κάποιος τη γνώση

με πολλούς τρόπους μπορεί να αυξήσει τα κίνητρα και τα επίπεδα κατανόησης, καθώς και την κατοχή του αντικειμένου. Σε σχέση με τα μαθηματικά, η Hoyles (1985) συζητά πώς η γλώσσα μπορεί να διευκολύνει το συλλογισμό και την εσωτερική ρύθμιση και μέρος αυτής της διαδικασίας είναι ο εντοπισμός των τμημάτων της μαθηματικής ιδέας που είναι σημαντικά και αυτά που δεν είναι. Αυτός ο προβληματισμός ή η σκέψη για τη δική του σκέψη είναι ένα βασικό συστατικό της κατασκευαστικής προσέγγισης (Han & Bhattacharya 2001), με την ίδια τη γλώσσα προγραμματισμού να γίνεται το εργαλείο «σκέψης με». Αυτή η δεύτερη δομή υποδηλώνει τη σημασία της ενσωμάτωσης συλλογιστικών ερωτήσεων και ευκαιριών για συζήτηση με συνομηλικούς καθώς και αλληλεπιδράσεων ολόκληρης τάξης που ενορχηστρώθηκαν από τον δάσκαλο.

Οι δραστηριότητες που σχεδιάζονται γύρω από αυτό το κατασκεύασμα προσπαθούν να αντιμετωπίσουν την αλληλεπίδραση μεταξύ του εργαλείου και της μάθησης. Μόλις οι μαθητές εξοικειωθούν με το εργαλείο, παρατηρήθηκε ότι η ευκολία δημιουργίας περιγραφών (σεναρίων), μπορεί να ενθαρρύνει τους μαθητές να δημιουργήσουν εξαιρετικά μεγάλες περιγραφές οι οποίες στη συνέχεια φαίνεται ότι απαιτούν πολλή «εργασία»! Ωστόσο, είναι δύσκολο να κατανοήσουν και να προβλέψουν τι θα κάνουν αυτές οι περιγραφές όταν κάνουν κλικ. Μια άλλη βασική ιδέα του προγράμματος σπουδών της SM είναι οι ορισμοί, ένα υπο-χρησιμοποιημένο στοιχείο του Scratch αλλά το οποίο συμβάλλει στη μείωση της πολυπλοκότητας και βοηθά στην αναγνωσιμότητα των σεναρίων. Το πρόγραμμα σπουδών του SM προάγει τη χρήση ορισμών μέσω της διαδικασίας δημιουργίας ενός σεναρίου και στη συνέχεια του προσδίδει ένα όνομα με νόημα (π.χ. Σχήμα 2). Αυτό με τη σειρά του υποστηρίζει τους μαθητές στο να εξηγούν βήμα προς βήμα τι κάνει το σενάριο τους και ποιο αποτέλεσμα θα συμβεί.



Σχήμα 2. Παράδειγμα δραστηριότητας όπου οι μαθητές ενθαρρύνονται να καθορίσουν μπλοκ που στη συνέχεια τους βοηθούν να εξηγήσουν πώς έχουν σχεδιάσει τα σπίτια τους

Αντιμετωπίζω (Envisage)

Είναι σημαντικό να έχετε στο μυαλό σας ένα στόχο κατά την κατασκευή ενός προγράμματος υπολογιστή και να προβλέψετε ποιο θα ήταν το αποτέλεσμα πριν το δοκιμάσετε. Ο Papert (1980) περιγράφει το ρόλο της δημιουργίας προγραμμάτων ηλεκτρονικών υπολογιστών στη διευκόλυνση του προβληματισμού σχετικά με τις διαισθητικές προσδοκίες και τις γνώσεις και επισημαίνει ότι η σύνδεση μεταξύ της ιδέας και της διαισθητικής γνώσης του παιδιού θεωρείται βασικό στοιχείο στην κατανόηση της δύναμης της ιδέας (Papert, 2000). Ωστόσο, οι Rader, Brand, & Lewis, (1997) διαπίστωσαν ότι στην εργασία τους χρησιμοποιώντας περιβάλλον προγραμματισμού για παιδιά, τα παιδιά μπορούν εύκολα να δημιουργήσουν προγράμματα χωρίς "πολύ γνώση των υποκείμενων μηχανισμών του προγράμματος" (όπως προαναφέρθηκε). Δεδομένου ότι συχνά τα εργαλεία προγραμματισμού για αρχάριους μπορούν τώρα να διαχειριστούν ένα μεγάλο μέρος του χειρισμού των συντακτικών σφαλμάτων, οι μαθητές αναγκάζονται να "εντοπίσουν τα σφάλματα" μόνο όταν έχουν σαφή

στόχο, με άλλα λόγια είναι αρκετά απλό να δημιουργηθεί ένα αποτέλεσμα, αλλά όχι απαραίτητα ένα συγκεκριμένο αποτέλεσμα που αποφασίστηκε προκαταβολικά.

Επομένως, για να κατανοήσουμε πραγματικά μια ιδέα, είναι απαραίτητο να πάρουμε χρόνο για να προβλέψουμε το αποτέλεσμα πριν χτίσουμε το πρόγραμμα και στη συνέχεια να συγκρίνουμε το πραγματικό αποτέλεσμα με αυτή την πρόβλεψη. Αυτό επιτρέπει τη διαπίστωση του εάν η αρχική διαίσθηση ήταν σωστή ή αν η γνώση αυτή πρέπει να αναδιαμορφωθεί (Papert, 1980). Αυτό το τρίτο κατασκεύασμα υποδηλώνει την ανάγκη να διεξαχθούν κάποιες μαθησιακές δραστηριότητες πριν από τη διερεύνηση με το εργαλείο προγραμματισμού, ώστε να δοθεί στους μαθητές η ευκαιρία να εξετάσουν το στόχο του προγράμματος και να προβλέψουν τα πιθανά αποτελέσματα της χρήσης διαφορετικών στρατηγικών. Είναι σημαντικό αυτό το κατασκεύασμα να είναι ισορροπημένο με τη διερεύνηση, παρέχοντας ευκαιρίες εξερεύνησης που επιτρέπουν την πραγματοποίηση ανακαλύψεων, αλλά και περιστάσεις για την πρόβλεψη του αποτελέσματος πρώτα.

Οι δραστηριότητες που σχεδιάζονται γύρω από αυτό το κατασκεύασμα έχουν σκοπό να αντιμετωπίσουν την ένταση μεταξύ του εργαλείου και της μάθησης. Πολλοί μαθητές που παρατηρήθηκαν κατά τη διάρκεια της δοκιμής ήταν πολύ χαρούμενοι και ενθουσιασμένοι με οποιοδήποτε αποτέλεσμα που οδήγησε σε ένα ελκυστικό σχέδιο που εμφανίζεται στην οθόνη ή σε μια διασκεδαστική κινούμενη εικόνα που παίζει, την οποία ήταν σε θέση να παράγουν χωρίς απαραίτητα να κατανοήσουν πλήρως πώς το δημιούργησαν ή ακόμα και έχοντας έναν σαφή στόχο σε ό, τι σκόπευαν να επιτύχουν. Το πρόγραμμα σπουδών του SM περιλαμβάνει μια σειρά από ασύνδετες δραστηριότητες, οι οποίες απαιτούν από τους μαθητές να εργάζονται στον υπολογιστή, ενθαρρύνοντάς τους να ασκούν δεξιότητες πρόβλεψης, συλλογισμού και εντοπισμού σφαλμάτων πριν δοκιμάσουν τον υπολογιστή τους και να προβληματιστούν σχετικά με τον τρόπο με τον οποίο μπορούν να ασχοληθούν με παρόμοιες δραστηριότητες σε άλλους τομείς του προγράμματος σπουδών (βλ. Γεφύρωσε τα E παρακάτω). Προωθεί επίσης το συντονισμό των ατόμων (Watt, 1998), ενθαρρύνοντας ταυτόχρονα τους δασκάλους και τους μαθητές να θεωρούν τους εαυτούς τους ως στοιχεία δραστηριοτήτων που απαιτούν να λειτουργήσουν τα σενάρια ή τα φυσικά αντικείμενα όπως το κόψιμο χαρτιού και τα παιχνίδια.

Ανταλλαγή (Exchange)

Η συνεργασία και η ανταλλαγή είναι ένας ισχυρός τρόπος μάθησης, με τις κατασκευαστικές προσεγγίσεις να υποστηρίζουν την ανάπτυξη ιδεών μέσω της αλληλεπίδρασης με άλλους (Han & Bhattacharya 2001). Αυτό επιτρέπει σε κάποιον να έρθει στη θέση του άλλου, ενώ προσπαθεί να δει ένα πρόβλημα από την πλευρά του άλλου, καθώς και να υπερασπιστεί τη δική του προσέγγιση και να τη συγκρίνει με τους άλλους. Επιπλέον, η Hoyles (1985) προτείνει ότι οι ιδέες των άλλων μπορούν δυνητικά να οδηγήσουν σε τροποποιήσεις των διαδικασιών σκέψης ενός ατόμου, ιδιαίτερα χρήσιμες στην αποσαφήνιση προβλέψεων ή στην εξήγηση ιδεών που δεν έχουν ακόμη διαμορφωθεί πλήρως.

Ο Bruckman (1998) στην έρευνα του καταδεικνύει τα γνωστικά, κοινωνικά και ψυχολογικά οφέλη που μπορούν να παράσχουν οι κατασκευαστικές δραστηριότητες στο πλαίσιο μιας online κοινότητας. Ωστόσο, τα παιδιά αναπτύσσουν ακόμα τις δεξιότητές της συνεργασίας και ίσως χρειαστούν βοήθεια για να συνεργαστούν, να επιλύσουν τις διαφωνίες τους και να αμφισβητήσουν το ένα το άλλο (Hoyles 1985). Επομένως, το τέταρτο κατασκεύασμα απαιτεί τη συμπερίληψη σημαντικών ευκαιριών για να μοιραστούν και να βασιστούν στις ιδέες των άλλων.

Οι δραστηριότητες που σχεδιάστηκαν γύρω από αυτό το κατασκεύασμα έχουν σκοπό να αντιμετωπίσουν την ένταση μεταξύ κατεύθυνσης και ανακάλυψης. Η συνεργατική μάθηση προσφέρει τη δυνατότητα να προωθηθεί η λιγότερο κατευθυνόμενη εξερεύνηση και ανακάλυψη. Κατά τη διάρκεια της δοκιμής στα σχολεία παρατηρήθηκε ότι η εργασία σε ζευγάρια θα μπορούσε να ενθαρρύνει τις συζητήσεις, απαιτώντας από τους μαθητές να εξηγήσουν τις στρατηγικές και τις ανακαλύψεις τους στον συμμαθητή τους. Μερικοί δάσκαλοι διευθετούν μικτές ικανότητες ζευγαριών, ενθαρρύνοντας τους πιο ικανούς μαθητές να υποστηρίξουν λιγότερο ικανούς μαθητές, «διδάσκοντας» τους ό,τι είχαν ήδη ανακαλύψει για τους εαυτούς τους. Οι μεμονωμένες ανακαλύψεις παρατηρήθηκαν επίσης γρήγορα διασκορπισμένες σε ολόκληρη την τάξη χωρίς την παρέμβαση των εκπαιδευτικών μέσω των μαθητών που παρακολουθούσαν αυτό στο οποίο εργάζονταν οι συνομήλικοί τους.

Γεφύρωσε τα E (bridgeE)

Οι ισχυρές ιδέες πρέπει να ενσωματωθούν σε κάθε καλοσχεδιασμένη κατασκευαστική δραστηριότητα (Bers, Flannery, Kazakoff, & Sullivan, 2014) και οι ιδέες θεωρούνται ισχυρές εν μέρει μέσω των συνδέσεών τους με άλλους κλάδους, όπως τα μαθηματικά (Papert, 2000) και εν μέρει μέσω της γλώσσας που εκφράζονται. Προκειμένου να αναπτυχθούν αυτές οι συνδέσεις, οι ιδέες πρέπει να επαναπλαισιωθούν και να ξαναχτιστούν στη γλώσσα του άλλου κλάδου. Επομένως, το τελικό κατασκεύασμα απαιτεί οι δραστηριότητες ή οι κινήσεις των εκπαιδευτικών να προτείνονται για να κάνουν ρητούς δεσμούς με ένα άλλο πλαίσιο (στην περίπτωση μας τα σχολικά μαθηματικά).

Οι δραστηριότητες που σχεδιάζονται γύρω από αυτό το κατασκεύασμα φαίνεται να αντιμετωπίζουν την ένταση ανάμεσα στο εργαλείο και τη μάθηση. Σε μία από τις τάξεις κατά τη διάρκεια μιας δραστηριότητας, η οποία απαιτούσε να εμφανιστούν κυκλικά επαναλαμβανόμενα μοτίβα, παρατηρήθηκε οι μαθητές να υπολογίζουν την τιμή του μπλοκ επανάληψης, διαιρώντας το 360 με οποιαδήποτε επιλεγμένη τιμή στο μπλοκ στροφής. Μερικές φορές αυτό είχε ως αποτέλεσμα ένα δεκαδικό αριθμό, π.χ. 5,5, τον οποίο έπειτα εισήγαγαν στο μπλοκ επανάληψης. Το Scratch αντιμετωπίζει αυτόματα την είσοδο για επανάληψη στρογγυλευμένη πριν από την εκτέλεση (καθώς δεν είναι δυνατή η απεικόνιση), η οποία στην περίπτωση αυτή έγινε με στρογγυλοποίηση προς τα πάνω. Για να διασφαλιστεί ότι αυτές οι σημαντικές μαθηματικές ευκαιρίες μάθησης δεν παραβλέπονται λόγω της συμπεριφοράς του εργαλείου, γίνονται σαφείς δεσμοί με το πρόγραμμα σπουδών των μαθηματικών και προτείνονται ερωτήσεις στους δασκάλους για την εκκίνηση της συζήτησης στο εσωτερικό των υλικών. Οι δραστηριότητες εκτός σύνδεσης αποσκοπούν επίσης στην περαιτέρω εδραίωση αυτών των συνδέσεων μακριά από τον υπολογιστή.

Κάποια τελικά συμπεράσματα

Παρακάτω συζητούνται τα ευρήματα των Benton et al. (2017) όσον αφορά τους αρχικούς ερευνητικούς τους στόχους. Ο πρώτος στόχος ήταν ο εξής: *Ποια είναι τα χαρακτηριστικά μιας τεκμηριωμένης παρέμβασης στο πρόγραμμα σπουδών τόσο στα μαθηματικά (π.χ. 360° πλήρης στροφή) όσο και στον προγραμματισμό (π.χ. αλγόριθμος) για μαθητές της πρωτοβάθμιας εκπαίδευσης (ηλικίας 9-11 ετών)*; Ο ερευνητικός σχεδιασμός τόνισε ότι η ιδέα του αλγορίθμου είναι μια ιδιαίτερα προκλητική ιδέα τόσο για τους μαθητές όσο και για τους εκπαιδευτικούς. Κατά συνέπεια, αυτός ήταν ένας συγκεκριμένος στόχος, ο οποίος παρουσιάστηκε με σαφήνεια και επεξηγηματικά, και για τον οποίο οι εκπαιδευτικοί ήταν εξοπλισμένοι με μια σειρά υποστηρίξεων για να τους βοηθήσουν να μεταδώσουν την έννοια στους μαθητές τους.

Συνήθως, στα βασικά μαθήματα υπολογιστών, ο αλγόριθμος εισάγεται μέσω καθημερινών ή σχολικών δραστηριοτήτων, οι οποίες απευθύνονται μόνο σε περιορισμένα χαρακτηριστικά ενός αλγορίθμου, όπως η διαδοχική σειρά και έχουν πολύ λίγη εξάρτηση από τα μεταφορικά παραδείγματα, όπως η κατασκευή φλιτζανιών τσαγιού ή σάντουιτς μαρμελάδας (sic). Μέσα στην αθροιστική μελέτη περίπτωσης, είδαμε ότι και οι δύο εκπαιδευτικοί χρησιμοποιούν πολλούς τρόπους για να αναπαραστήσουν την έννοια των εναλλακτικών αλγορίθμων, κυρίως χρησιμοποιώντας διαφορετικές μορφές απεικόνισης όπως η φυσική εφαρμογή ή οπτικοποίηση του μοτίβου που πρέπει να εξάγεται στην οθόνη.

Η έννοια της πλήρους στροφής 360° ήταν επίσης προκλητική τόσο για τους δασκάλους όσο και για τους μαθητές. Σημειώσαμε τη σημασία της επιστροφής σε δραστηριότητες εκτός υπολογιστή για την ρητή αντιμετώπιση αυτής της έννοιας και την παροχή δεικτών για τα βασικά χαρακτηριστικά, όπως η έκφραση του αριθμού των επαναλήψεων.

Τα ευρήματα επισήμαναν ότι αυτή η προσέγγιση επέτρεψε σε μερικούς μαθητές με χαμηλότερες επιδόσεις την επίτευξη του μαθησιακού στόχου με ελάχιστη υποστήριξη. Αυτό επομένως επιτρέπει στον δάσκαλο κατά τη διάρκεια των επόμενων μαθημάτων να παρέχει πρόσθετη υποστήριξη σε εκείνους τους μαθητές που την χρειάζονται πραγματικά και όχι να παρέχει περιττή υποστήριξη σε εκείνους που υποτίθεται ότι το χρειάζονται. Ακόμα περισσότερο αν και η έρευνα αυτή δεν επικεντρώθηκε στην επίδραση του μαθητή, στο πλαίσιο της κονστрукτιβιστικής θεμελίωσης της εργασίας αξίζει να σημειωθεί ότι παρατηρήθηκε υψηλό επίπεδο συμμετοχής των μαθητών στις δραστηριότητες και στις δύο τάξεις, καθώς το ενδιαφέρον των μαθητών είναι σημαντικό στην κονστрукτιβιστική προσέγγιση (Papert 1980, 1993).

Στις συνεντεύξεις μετά το μάθημα οι δάσκαλοι ήταν καθολικά θετικοί σχετικά με τη εμπλοκή των μαθητών τους με τις δραστηριότητες, με τη Δασκάλα 2 να υπογραμμίζει το όφελος αυτής της εμπλοκής για την εκμάθηση μαθηματικών των μαθητών της. «Ήταν πολύ καλό γιατί ενδιαφέρονται πραγματικά για το Scratch και είναι καλό να βάζεις γωνίες ειδικά σε ένα πλαίσιο που σημαίνει κάτι για αυτούς γιατί διαφορετικά μπορεί να είναι πραγματικά περίεργο και αόριστο».

Το πρωταρχικό θέμα σε αυτό το άρθρο ήταν να διερευνηθεί η αποτελεσματικότητα μιας παρέμβασης στον προγραμματισμό και τα μαθηματικά από την άποψη της παιδαγωγικής ενός προγράμματος σπουδών αλλά και από την άποψη της μάθησης των μαθητών δύο βασικών εννοιών. Το ερώτημα ήταν αν η διδασκαλία με σωστή επιλογή μαθηματικών ιδεών μέσω του προγραμματισμού Scratch με προσεκτικά σχεδιασμένους τρόπους συνδέεται με τη μάθηση των παιδιών στο πώς μαθαίνουν καλύτερα να εκφράζονται και να αιτιολογούν με μαθηματικό τρόπο. Επιπλέον, ποια θα μπορούσαν να είναι τα εννοιολογικά και παιδαγωγικά εμπόδια και πώς θα μπορούσαν να αντιμετωπιστούν;

Στην έρευνα αυτή έχει δημιουργηθεί ένα πλαίσιο δράσης το οποίο, εστιάζοντας ρητά στα μαθηματικά (η περισσότερη έρευνα που αφορά τον προγραμματισμό αφορά στην πραγματικότητα τον προγραμματισμό), μπορεί να βοηθήσει πιο αποτελεσματικά στην κατανόηση της σχέσης μεταξύ υπολογιστικής και μαθηματικής σκέψης και, κατά συνέπεια, να σχεδιαστούν προγράμματα σπουδών (τόσο τυπικά όσο και άτυπα) που είναι πιο επιτυχημένα από αυτά που έχουμε τώρα.

Αναφορές

Agalianos, A., Whitty, G., & Noss, R. (2006). The social shaping of Logo. *Social studies of science*, 36(2), 241-267.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K–12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54.

Bell, T., Newton, H., Andreae, P., & Robins, A. (2012a). The introduction of computer science to NZ high schools: An analysis of student work. In *Proceedings of the 7th Workshop on Primary and Secondary Computing education* (pp. 5–15). Hamburg: WIPSCe.

Bell, T., Rosamond, F., & Casey, N. (2012b). Computer science unplugged and related projects in math and computer science popularization. In H. Bodlaender, R. Downey, F. Fomin, & D. Marx (Eds.), *The multivariate algorithmic revolution and beyond* (pp. 398–456). Berlin: Springer-Verlag Berlin Heidelberg.

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, 3(2), 115-138.

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2016, February). Building mathematical knowledge with programming: insights from the ScratchMaths project. Suksapattana Foundation.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.

Brennan, K. (2015). Beyond technocentrism: supporting constructionism in the classroom. *Constructivist Foundations*, 10, 289–296.

Brown, A.L., 1988. Motivation to learn and understand: On taking charge of one's own learning. *Cognition and Instruction*, 5, pp.311–322.

Brown, N., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Transactions on Computing Education*, 14(2), 1–22.

Bruckman, A., (1998). Community Support for Constructionist Learning. *CSCW (Computer Supported Collaborative Work: The Journal of Collaborative Computing)*, 7, pp.47–86.

Clements, D. H. (1985). Research on Logo in education: is the turtle slow but steady, or not even in the race? *Computers in the Schools*, 2, 55–71.

Clements, D. H. (1999). The future of educational computing research: the case of computer programming. *Information Technology in Childhood Education Annual*, 1, 147–179.

Clements, D. H., & Sarama, J. (1997). Research on Logo: a decade of progress. *Computers in the Schools*, 14, 9–46.

- DiSessa, A.A. & Cobb, P., 2004. Ontological Innovation and the Role of Theory in Design Experiments. *Journal of the Learning Sciences*, 13(1), pp.77–103.
- Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, pp. 60–69.
- Han, S. & Bhattacharya, K., (2001). Constructionism, Learning by Design, and Project Based Learning. In *Emerging perspectives on learning, teaching, and technology*.
- Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments*, 1, 1–32.
- Hatfield, L., & Kieren, T. E. (1972). Computer assisted problem solving in school mathematics. *Journal for Research in Mathematics Education*, 3, 99–112.
- Hoyles, C., (1985). What is the point of group discussion in mathematics? *Educational studies in mathematics*, 16(2), pp.205–214.
- Hoyles, C., & Noss, R. (1992). *Learning mathematics and Logo*. Cambridge: MIT Press.
- Falkner, K., Vivian, R., & Falkner, N. (2014). The Australian digital technologies curriculum: Challenge and opportunity. In J. Whalley & D. D'Souza (Eds.), *Proceedings of the Sixteenth Australasian Computing Education conference* (pp. 3–12). Auckland: Australian Computer Society.
- Fisher, L. (2016). A decade of ACM efforts contribute to computer science for all. *Communications of the ACM*, 59(4), 25–27.
- Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. (1969). Programming Languages as a Conceptual Framework for Teaching Mathematics. Final Report on the First Fifteen Months of the LOGO Project.
- Kelleher, C. (2012). Reading, Writing, Arithmetic and Programming.
- Κόμης Β. (2005). *Εισαγωγή στη Διδακτική της Πληροφορικής*. Αθήνα: Εκδόσεις Κλειδάριθμος.
- Lye, S., & Koh, J. (2014). Review on teaching and learning of computational thinking through programming: What is next for K–12? *Computers in Human Behavior*, 41, 51–61.
- McCoy, L. P. (1996). Computer-based mathematics learning. *Journal of Research on Computing in Education*, 28, 438–460.
- Miller, R. B., Kelly, G. N., & Kelly, J. T. (1988). Effects of Logo computer programming experience on problem solving and spatial relations ability. *Contemporary Educational Psychology*, 13, 348–357.
- Μπόκος, Α. (2014). *Ρομποτικά Εκπαιδευτικά Περιβάλλοντα: μελέτη της διαδικασίας προγραμματισμού* (Doctoral dissertation).
- Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings: learning cultures and computers*. London: Kluwer.

Ξυνόγαλος, Σ., Σατρατζέμη Μ. & Διαγδιδέλης, Β. (2000). Η εισαγωγή στον προγραμματισμό: Διδακτικές Προσεγγίσεις και Εκπαιδευτικά Εργαλεία. *Πρακτικά 2ου Συνεδρίου ΕΤΠΕ*.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc.

Papert, S. (1990). *Children, computers and powerful ideas*. BasicBooks.

Papert, S. (1993). *Mindstorms: Children, computers and powerful ideas*. New York: Basic Books.

Papert, S., (2000). What's the big idea? Toward a pedagogy of idea power. *IBM Systems Journal*, 39(3.4), pp.720–729.

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, 36, 1–11.

Prediger, S., Gravemeijer, K., & Confrey, J. (2015). Design research with a focus on learning processes: an overview on achievements and challenges. *ZDM*, 47, 877–891.

Rader, C., Brand, C. & Lewis, C., 1997. Degrees of comprehension: children's understanding of a visual programming environment. In *Proc. CHI '9*. ACM, pp. 351–358.

Resnick, M., Maloney, J., Monroy-Hernández, A. R., Eastmond, N. E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52, 60–67.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22(September), 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>

Vivian, R., Falkner, K., & Falkner, N. (2014). Addressing the challenges of a new digital technologies curriculum: MOOCs as a scalable solution for teacher professional development. *Research in Learning Technology*, 22. <https://doi.org/10.3402/rlt.v22.24691>.

Watt, S., (1998). Syntonicity and the psychology of programming. In *Proceedings of 10th Annual Meeting of the Psychology of Programming Interest Group*. pp. 75–86.

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. *Proceedings of the 14th International Conference on Interaction Design and Children, ACM*, pp. 199–208.

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

Wing, J. M. (2016). Computational thinking, 10 years later. *Microsoft Research Blog*, 23.

Yelland, N. (1995). Mindstorms or a storm in a teacup? A review of research with Logo. *International Journal of Mathematical Education in Science and Technology*, 26, 853–869.