

Μια πρόταση ταξινόμησης των προγραμματιστικών βρόχων στο Scratch

Α. Λαδιάς¹, Θ. Καρβουνίδης², Δ. Λαδιάς³, Χ. Δουληγέρης⁴

¹Εκπαιδευτικός Πληροφορικής
ladiastas@gmail.com

²Μεταδιδακτορικός Ερευνητής, Τμήμα Πληροφορικής Παν. Πειραιώς
tkarv@otenet.gr

³Φοιτητής Τμήματος Πληροφορικής ΕΚΠΑ
ladimitr@gmail.com

⁴Καθηγητής, Τμήμα Πληροφορικής Πανεπιστημίου Πειραιώς
cdoulig@unipi.gr

Περίληψη

Για το θέμα των δομών επανάληψης από εκπαιδευτική σκοπιά, υπάρχει μια εκτενής βιβλιογραφία, η οποία όμως τις περισσότερες φορές περιορίζεται σε στοχευμένα θέματα. Η παρούσα εργασία φιλοδοξεί να καταθέσει μια πρόταση για την ταξινόμηση των προγραμματιστικών βρόχων στο οπτικό περιβάλλον του Scratch με την προοπτική να ενταχθεί σε μια ευρύτερη έρευνα αξιολόγησης του κώδικα με τη βοήθεια της ταξινόμησης solo. Στην κατηγοριοποίηση των δυνατών εναλλακτικών τρόπων υλοποίησης βρόχων χρησιμοποιήθηκε το ρεπερτόριο των επαναληπτικών εντολών όπως επίσης και η χρήση μηνυμάτων, κλώνων και αναδρομικών διαδικασιών. Η οπτική αναπαράσταση που προέκυψε από αυτή την κατηγοριοποίηση οδήγησε αφενός σε σχήματα βρόχων και αφετέρου σε μοτίβα βρόχων που έτεμναν τα σχήματα. Επίσης, καταγράφηκαν και λανθάνοντες βρόχοι που δημιουργούνται αφενός από τις εντολές "περίμενε..." και αφετέρου από τις εντολές αντίχεισης συμβάντων "Όταν...". Η παρούσα πρόταση παρουσιάστηκε με μορφή σεμιναρίου σε εκπαιδευτικούς πληροφορικής. Από τα αποτελέσματα της αξιολόγησης του εν λόγω σεμιναρίου αναδεικνύεται η ανάγκη για εμβάθυνση ειδικότερα στους κλώνους.

Λέξεις κλειδιά: προγραμματιστικοί βρόχοι (loops), επανάληψη (iteration/repetition), αναδρομή (recursiveness), μηνύματα, κλώνοι, Scratch..

1. Εισαγωγή

Το άρθρο αυτό έρχεται να καλύψει μια ανάγκη η οποία προέκυψε κατά τη διαδικασία αξιολόγησης του κώδικα στα έργα του 4^{ου} Πανελληνίου Διαγωνισμού Εκπαιδευτικής Ρομποτικής, όπου εντοπίστηκαν σημεία που γινόταν επανάληψη με τη χρήση μηνυμάτων. Αυτό σε συνδυασμό με το γεγονός ότι επαναληπτικοί βρόχοι μπορεί να υλοποιηθούν με τη χρήση κλώνων (Manatakí & Kereki, 2015), μας οδήγησε στο να διερευνηθούν οι τρόποι με τους οποίους μπορεί να γίνουν επαναληπτικοί βρόχοι -

πέραν των εντολών επανάληψης και των αναδρομικών διαδικασιών, με χρήση μηνυμάτων και κλώνων.

Ο προγραμματισμός των έργων στον Διαγωνισμό γινόταν στο προγραμματιστικό περιβάλλον του Scratch. Το Scratch (<https://scratch.mit.edu/>) είναι ένα εκπαιδευτικό περιβάλλον οπτικού προγραμματισμού με πλακίδια το οποίο βασίζεται σε αντικείμενα και χρησιμοποιεί διεργασίες οι οποίες εκτελούνται εκτός από σειριακά και παράλληλα. Οι προγραμματιστικοί βρόχοι στο Scratch υλοποιούνται εκ πρώτης όψεως, αφενός μεν με τις τρεις προγραμματιστικές δομές επανάληψης "επανάλαβε τόσες φορές", "επανάλαβε ώσπου" και "για πάντα", αφετέρου δε με την αναδρομή -η οποία αν και συμπεριλαμβάνεται στα Προγράμματα Σπουδών Πληροφορικής της Δευτεροβάθμιας Εκπαίδευσης δεν διδάσκεται μετά από οδηγίες του Υπουργείου Παιδείας- κατά την οποία στο εσωτερικό του ορισμού μιας διαδικασίας χρησιμοποιείται η ίδια η διαδικασία (Papert, 1991). Σε αυτό το πλαίσιο και έχοντας υπόψη μας μερικές επιπλέον προτάσεις, όπως τη χρήση αντικειμένων και κλώνων, τις παράλληλες διεργασίες με χρήση μηνυμάτων και τις εντολές που υποκρύπτουν επαναληπτικές διεργασίες, επιχειρήθηκε να κατηγοριοποιηθούν οι διάφοροι τρόποι δημιουργίας βρόχων, εντάσσοντάς τους σε μια περισσότερο ολοκληρωμένη οπτική. Επειδή το Scratch, ως οπτικός προγραμματισμός, διευκολύνει την απεικόνιση των διεργασιών (με όσα προβλήματα αναφύονται καθώς οι βρόχοι εξελίσσονται στη χρονική διάσταση ενώ η οπτική αναπαράστασή τους γίνεται στο χώρο των δύο διαστάσεων) έτσι ώστε να αναδεικνύονται περιοχές με συγκεκριμένα σχήματα και μοτίβα. Για την κατανόηση των εννοιών αυτών, στη συνέχεια θα αναπτυχθούν αφενός οι διαφορετικοί τρόποι υλοποίησης, βρόχων που τους ονομάζουμε "**σχήματα βρόχων**" και αφετέρου οι βρόχοι που υλοποιούνται με διαφορετικούς τρόπους (σχήματα) αλλά έχουν κοινά χαρακτηριστικά,, που τα ονομάζουμε "**μοτίβα βρόχων**".

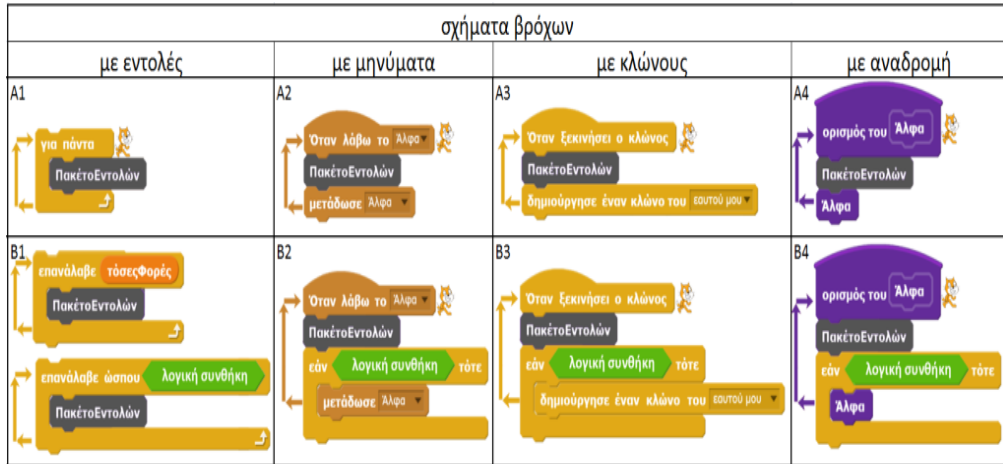
2. Σχήματα βρόχων

Τα σχήματα βρόχων που θα περιγραφούν παρακάτω είναι αφενός οι απλοί κυκλικοί βρόχοι και οι βρόχοι Möbius και αφετέρου οι εμφωλευμένοι βρόχοι.

Κυκλικοί βρόχοι. Σε πρώτη φάση έγινε η καταγραφή των διαφορετικών σχημάτων με τα οποία μπορεί να δημιουργηθούν προγραμματιστικοί βρόχοι στο περιβάλλον του Scratch. Έτσι πέραν (α) της χρήσης των εντολών επανάληψης των προγραμματιστικών δομών (Σχήμα 1, A1) και (β) της χρήσης αναδρομικών διαδικασιών (Σχήμα 1, A4), υιοθετήθηκε (γ) η χρήση κλώνων ως δομικό στοιχείο επανάληψης, όπου ένας κλώνος δημιουργεί ένα κλώνο του εαυτού του (Σχήμα 1, A3) και (δ) η χρήση των μηνυμάτων ως εργαλεία για τη δημιουργία βρόχων (αντίστοιχο του βρόχου με χρήση του "go to"), όπου μια διεργασία που εξυπηρετεί κάποιο μήνυμα στέλνει μήνυμα στον εαυτό της (Σχήμα 1, A2). Όλοι οι προαναφερθέντες τρόποι δημιουργούν ατέρμονες βρόχους.

Βρόχοι που τερματίζονται μπορεί να δημιουργηθούν με τη χρήση εντολών πεπερασμένου αριθμού επαναλήψεων ("επανάλαβε τόσες φορές" / for) ή

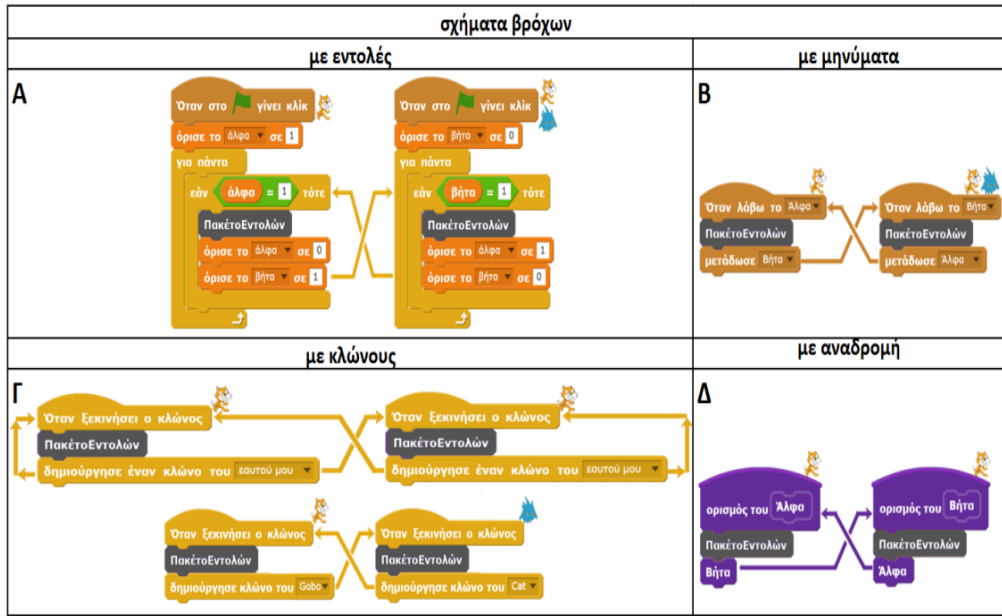
απροσδιόριστου (αλλά πεπερασμένου) πλήθους επαναλήψεων ("επανάλαβε ώσπου να συμβεί κάτι" / while & repeat until), όπως φαίνεται στο B1 του Σχήματος 1. Ανάλογα με τα προηγούμενα, έχουμε δομές βρόχων που ελέγχουν πριν κληθεί αναδρομικά η ίδια διαδικασία (Σχήμα 1, B4) ή πριν δημιουργηθεί ένας κλώνος του εαυτού του (Σχήμα 1, B3) ή πριν αποσταλεί ένα μήνυμα στον εαυτό του (Σχήμα 1, B2).



Σχήμα 1. Μη εμφωλευμένα κυκλικά σχήματα προγραμματιστικών βρόχων.

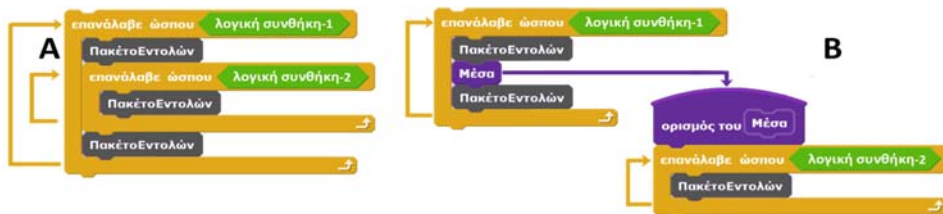
Βρόχοι Möbius. Ένας άλλος τρόπος σχηματισμού βρόχων -εν προκειμένω ατερμόνων που η απεικόνισή τους παραπέμπει στην κίνηση κατά μήκος μιας ταινίας Möbius- βασίζεται στον έμμεσο αναδρομικό τρόπο κλήσης διαδικασιών (Μικρόπουλος & Λαδιάς 2000) κατά τα τον οποίο μια διαδικασία A καλεί μια άλλη διαδικασία B, η οποία (η B) με τη σειρά της καλεί την αρχική διαδικασία A (Σχήμα 2, Δ). Με δεδομένο αυτό το αναδρομικό σχήμα, διερευνήθηκε η ύπαρξη ανάλογων τρόπων δημιουργίας βρόχων υλοποιούμενων με μηνύματα, κλώνους και εντολές. Αυτή η αναζήτηση οδήγησε: (α) στην υλοποίηση με μηνύματα (Σχήμα 2, Β) όπου το σενάριο που εξυπηρετεί το μήνυμα A στέλνει μήνυμα στο σενάριο που εξυπηρετεί το μήνυμα Β και αυτό με τη σειρά του στέλνει μήνυμα στο αρχικό σενάριο που εξυπηρετεί το μήνυμα Α (αξίζει να παρατηρηθεί ότι τα δύο αυτά σενάρια μπορεί και να μην ανήκουν στο ίδιο αντικείμενο), (β) στην υλοποίηση με κλώνους όπου υπάρχουν δύο σενάρια δημιουργίας κλώνου που εάν ανήκουν σε διαφορετικά αντικείμενα (Σχήμα 2, Γ κάτω) δημιουργείται ένας βρόχος τύπου Möbius, ενώ εάν ανήκουν στο ίδιο αντικείμενο (Σχήμα 2, Γ πάνω) δημιουργούνται ταυτόχρονα με τον βρόχο τύπου Möbius και δύο βρόχοι όπως αυτοί του σχήματος 1, A3 και, (γ) στην υλοποίηση με εντολές (Σχήμα 2, Α) όπου χρειάζονται δύο σενάρια που τρέχουν ταυτόχρονα και ενεργοποιούνται εναλλακτικά με τη χρήση μεταβλητών σε ρόλους σηματορών (αξίζει να παρατηρηθεί ότι τα δύο αυτά σενάρια μπορεί και να μην ανήκουν στο ίδιο αντικείμενο). Αντίστοιχα με τα κυκλικά σχήματα όπου μπορεί να υπάρξουν και τερματισμένες δομές που

ελέγχονται από εντολές ελέγχου, μπορεί να υλοποιηθούν και τερματίσιμες δομές σε σχήματα Möbius.



Σχήμα 2. Μη εμφωλευμένα σχήματα προγραμματιστικών βρόχων "Möbius". Τα εικονίδια δίπλα σε κάθε σενάριο δηλώνουν το αντικείμενο στο οποίο ανήκει το σενάριο.

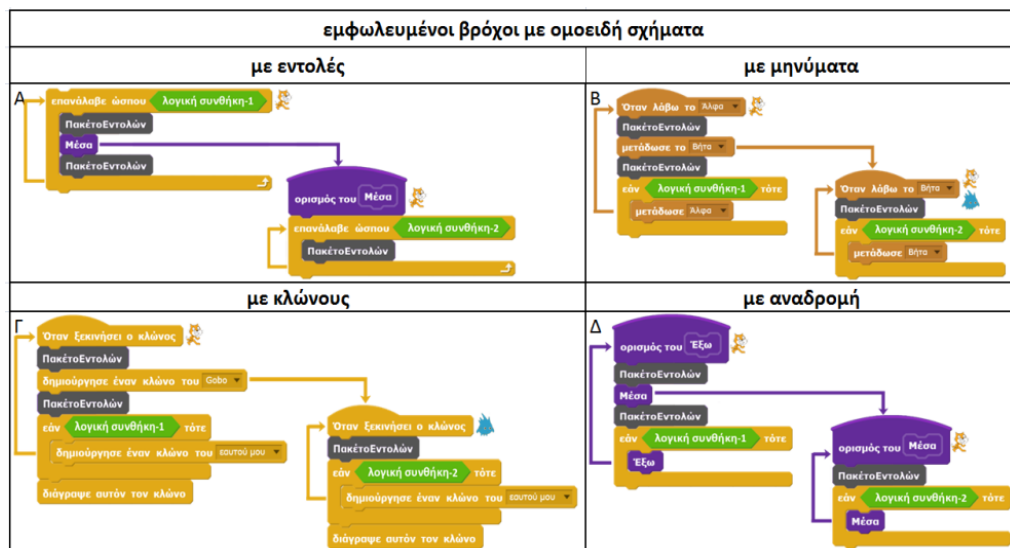
Εμφωλευμένοι βρόχοι. Η εμφώλευση βρόχου μέσα σε βρόχο (εμφώλευση 1ου βαθμού) με τη χρήση εντολών επανάληψης μπορεί να απεικονιστεί στο Scratch όπως φαίνεται στο Σχήμα 3A. Ο κώδικας του Σχήματος 3B, ο οποίος είναι ισοδύναμος με τον κώδικα 3A, προκύπτει μεταφέροντας τον εσωτερικό βρόχο του πρώτου κώδικα σε διαδικασία στο δεύτερο.



Σχήμα 3. Απεικονίσεις ισοδύναμων κωδίκων εμφωλευμένου βρόχου.

Αν και η εμφώλευση βρόχου μέσα σε βρόχο με τη χρήση εντολών επανάληψης (Σχήμα 3A) είναι μια ευρέως διαδεδομένη τεχνική που εφαρμόζεται στον προγραμματισμό, όπως επίσης και η αναδρομική κλήση διαδικασίας (Σχήμα 1, A4, B4) είναι γνωστή

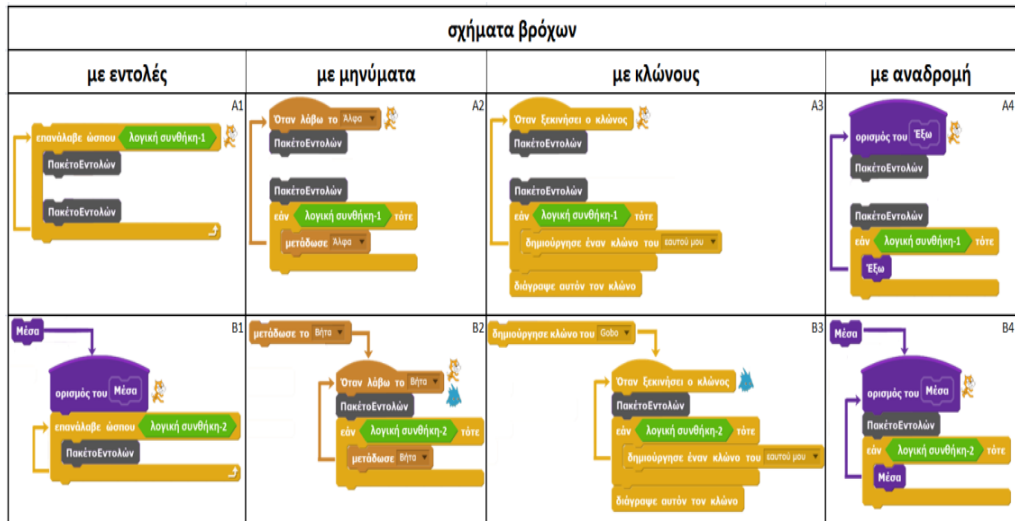
στους εκπαιδευτικούς πληροφορικής, σε ερώτηση που έγινε σε εκπαιδευτικούς κατά τη διάρκεια σεμιναρίου που παρακολούθησαν 55 εκπαιδευτικοί (Λαδιάς, 2018) για το αν έχουν διανοηθεί ποτέ να υλοποιήσουν εμφωλευμένη αναδρομή (αναδρομή μέσα σε αναδρομή), η απάντηση ήταν αρνητική.



Σχήμα 4. Εμφωλευμένα ομοειδή σχήματα προγραμματιστικών βρόχων.

Το νοητικό αυτό άλμα μπορεί να διευκολυνθεί αν στο Σχήμα 3B ο εξωτερικός βρόχος αντικατασταθεί με τη δομή του κώδικα του σχήματος 1, A4 (με ονομασία της διαδικασίας "Έξω") ενώ ο εσωτερικός βρόχος αντικατασταθεί και πάλι με τη δομή του κώδικα του Σχήματος 1, Δ1 (με ονομασία της διαδικασίας "Μέσα"). Οι διαφορές των εμφωλευμένων βρόχων που υλοποιούνται με εντολές επανάληψης και με αναδρομή αναδεικνύονται συγκρίνοντας τα A και Δ στο Σχήμα 4. Η υλοποίηση εμφωλευμένων ομοειδών βρόχων με μηνύματα και με κλώνους μπορεί να γίνει με αντικατάσταση των αναδρομικών εξωτερικών και εσωτερικών βρόχων (Σχήμα 4, Δ) με βρόχους υλοποιούμενους με μηνύματα (Σχήμα 1, A2) και κλώνους (Σχήμα 1, A3), οπότε θα προκύψουν τα Β και Γ του Σχήματος 4.

Σε όλες τις προηγούμενες δομές εμφωλευμένων βρόχων ο τρόπος που υλοποιούνταν ο εξωτερικός και ο εσωτερικός βρόχος ήταν ο ίδιος. Εάν διαφοροποιηθεί ο τρόπος υλοποίησης των εμφωλευμένων βρόχων τότε θα προκύψουν δώδεκα συνδυασμοί (Σχήμα 5) όπου κάθε σχήμα βρόχου συνδυάζεται με τα άλλα τρία σχήματα βρόχων. Έτσι για παράδειγμα θα μπορεί να συνδυαστεί ως εξωτερικός βρόχος ένας βρόχος υλοποιημένος με μηνύματα (Σχήμα 5, A2) με εσωτερικό βρόχο ένα βρόχο υλοποιημένο με κλώνους (Σχήμα 5, B3).



Σχήμα 5. Δώδεκα εμφωλευμένα μη ομοειδή σχήματα προγραμματιστικών βρόχων δημιουργούνται ως συνδυασμοί όταν στη θέση της κενής εντολής καθενός αρθρώματος της πρώτης γραμμής εμφυτευτεί άρθρωμα από τη δεύτερη γραμμή αλλά διαφορετικής στήλης π.χ. A1 με B2 ή A1 με B3 ή A1 με B4.

3. Μοτίβα Βρόχων

Παρατηρείται ότι στην πρώτη γραμμή του Σχήματος 1, οι τέσσερις βρόχοι -που υλοποιούνται με διαφορετικά σχήματα- έχουν ως κοινό χαρακτηριστικό ότι είναι απλοί (μη εμφωλευμένοι), κυκλικοί, ατέρμονες βρόχοι. Αυτή η κοινότητα χαρακτηριστικών μπορεί να αποδοθεί ως ένα μοτίβο βρόχων. Ανάλογο μοτίβο δημιουργούν και οι βρόχοι της δεύτερης γραμμής του Σχήματος 1 που είναι απλοί, κυκλικοί, τερματίσιμοι βρόχοι. Με αφετηρία αυτή την παρατήρηση μπορεί να δημιουργηθεί ο Πίνακας 1 με τα μοτίβα των μη εμφωλευμένων βρόχων (κυκλικών και Möbius, ατέρμονων και τερματίσιμων, εφαρμοζόμενων σε ένα ή περισσότερα αντικείμενα) όπου ένα μοτίβο εμφανίζεται ως γραμμή με κελιά του ίδιου χρώματος.

Η ταξινόμηση αυτή σε πίνακα υποδεικνύει μοτίβα που θα πρέπει να διερευνηθούν (γκρίζες ζώνες στον Πίνακα 1), των οποίων τα χαρακτηριστικά μπορεί να περιγραφούν λόγω της θέσης τους στον πίνακα αλλά και κελιά που δηλώνεται η μη υλοποίηση συγκεκριμένων βρόχων λόγω περιορισμένων δυνατοτήτων του Scratch, π.χ. στο Scratch δεν μπορεί να δημιουργηθούν σχήματα βρόχων με αναδρομικό τρόπο σε διαφορετικά αντικείμενα γιατί το Scratch είναι object-based και όχι object-oriented (οι διαδικασίες στο Scratch διαβιούν στο εσωτερικό των αντικειμένων).

Πίνακας 1. Μοτίβα μη εμφωλευμένων βρόχων

				Σχήματα βρόχων υλοποιούμενα				
				με εντολές	με μηνύματα	με κλώνους	με αναδρομή	
Μοτίβα βρόχων	μη εμφωλευμένα	κυκλικά	μοναδικό αντικείμενο	ατέρμονα	Σχήμα 1, A1	Σχήμα 1, A2	Σχήμα 1, A3	Σχήμα 1, A4
			ατέρμονα	Σχήμα 1, B1	Σχήμα 1, B2	Σχήμα 1, B3	Σχήμα 1, B4	
		Möbius	ατέρμονα	Σχήμα 2, A	Σχήμα 2, B	Σχήμα 2, Γ	Σχήμα 2, Δ	
			τερματίσιμα					
	περισσότερα αντικείμενα	ατέρμονα				μη υλοποιήσιμο		
		τερματίσιμα				μη υλοποιήσιμο		

Ανάλογη ταξινόμηση δημιουργείται για τους εμφωλευμένους βρόχους στον Πίνακα 2. Επιπλέον μοτίβο (πέραν των μοτίβων που δημιουργούνται ανά γραμμή ή στήλη με τους συνδυασμούς ανόμοιων τρόπων υλοποίησης του εξωτερικού με τον εσωτερικό βρόχο) προκύπτει όταν παρόμοια υλοποιείται ο εξωτερικός με τον εσωτερικό βρόχο και ο οποίος διατάσσεται κατά μήκος της κυρίας διαγωνίου του Πίνακα 2.

Πίνακας 2. Μοτίβα εμφωλευμένων βρόχων 1ου βαθμού

			Σχήματα βρόχων υλοποιούμενα με...				
			...εντολές	...μηνύματα	...κλώνους	...αναδρομή	
Μοτίβα βρόχων	εμφωλευμένα 1ου βαθμού	Ο εσωτερικός βρόχος υλοποιείται με...	εντολή	Σχήμα 4.A ή συνδυασμός A1+B1 Σχήματος 5	συνδυασμός A2+B1 Σχήματος 5	συνδυασμός A3+B1 Σχήματος 5	συνδυασμός A4+B1 Σχήματος 5
		μηνύμα	συνδυασμός A1+B2 Σχήματος 5	Σχήμα 4.B ή συνδυασμός A2+B2 Σχήματος 5	συνδυασμός A3+B2 Σχήματος 5	συνδυασμός A4+B2 Σχήματος 5	
		κλώνο	συνδυασμός A1+B3 Σχήματος 5	συνδυασμός A2+B3 Σχήματος 5	Σχήμα 4.Γ ή συνδυασμός A3+B3 Σχήματος 5	συνδυασμός A4+B3 Σχήματος 5	
		αναδρομή	συνδυασμός A1+B4 Σχήματος 5	συνδυασμός A2+B4 Σχήματος 5	συνδυασμός A3+B4 Σχήματος 5	Σχήμα 4.Δ ή συνδυασμός A4+B4 Σχήματος 5	

4. Λανθάνοντες βρόχοι

Εκτός από τα προαναφερθέντα σχήματα και μοτίβα βρόχων στο Scratch υπάρχουν επιπλέον προγραμματιστικές δομές που λειτουργούν ως βρόχοι. Μια πρώτη περίπτωση είναι η κατηγορία των τεσσάρων εντολών "περίμενε" (Σχήμα 6).



Σχήμα 6. Το ρεπερτόριο των εντολών "περίμενε" του Scratch.

Απόδειξη ότι οι εντολές "περίμενε" λειτουργούν ως βρόχοι είναι τα ισοδύναμα σενάρια του Σχήματος 7, στα οποία η λειτουργία της εντολής "περίμενε" αντιστοιχίζεται με την εντολή βρόχου "επανάλαβε ώσπου".



Σχήμα 7. Η υλοποίηση της εντολής "περίμενε ώσπου να ισχύσει μια λογική συνθήκη" με επαναληπτική δομή "επανάλαβε ώσπου να ισχύσει η λογική συνθήκη".

Μια άλλη περίπτωση είναι αυτή της εντολής "Όταν το πλήκτρο ... πατηθεί" που ανιχνεύει συμβάντα από το περιβάλλον του υπολογιστή (Σχήμα 8). Η αντιστοιχία αυτής της εντολής (τεχνική interrupt) είναι το σενάριο στο δεξιό μέρος του Σχήματος 8 που χρησιμοποιεί την εντολή "περίμενε ώσπου ..." για να ανιχνεύσει το ίδιο συμβάν (τεχνική polling).



Σχήμα 8. Λανθάνοντες βρόχοι για την ανίχνευση συμβάντων με τεχνικές interrupt και polling.

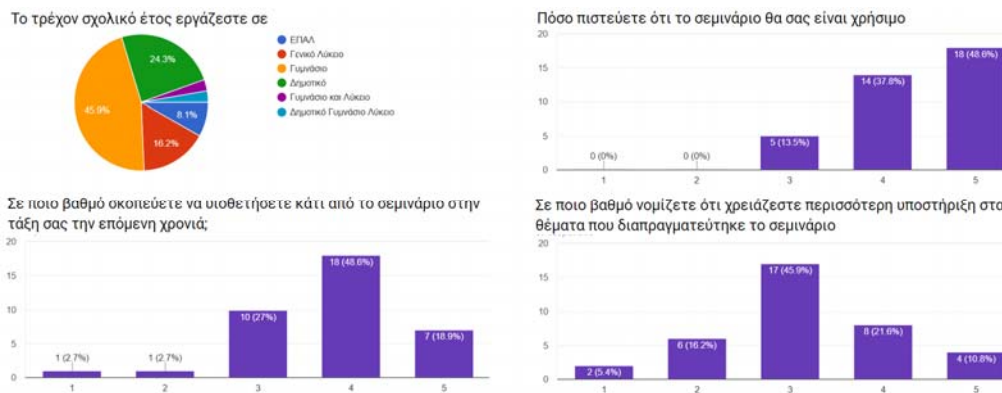
5. Πιλοτική παρουσίαση

Η παρούσα πρόταση ταξινόμησης των προγραμματιστικών βρόχων παρουσιάστηκε πιλοτικά σε εκπαιδευτικούς πληροφορικής, σε τρίωρο εργαστηριακό σεμινάριο, το σχολικό έτος 2017-18, σε σύνολο 44 εκπαιδευτικών (37 απάντησαν στο ερωτηματολόγιο αξιολόγησης).

Από το συνδυασμό των απαντήσεων του Σχήματος 9, φαίνεται ότι οι εκπαιδευτικοί εκτίμησαν ότι το περιεχόμενο του σεμιναρίου μπορεί να διδαχθεί στους μαθητές τους, αλλά και ότι οι ίδιοι χρειάζονται υποστήριξη σε αυτά τα θέματα. Τα στοιχεία που παρατίθενται (Λαδιάς, 2018) στοχεύουν στο να καταδείξουν την αποδοχή των εκπαιδευτικών πληροφορικής να εφαρμόσουν την προτεινόμενη ταξινόμηση στην

εκπαιδευτική πράξη. Αξίζει να παρατηρηθεί ότι η πλειοψηφία αυτών των εκπαιδευτικών (~70%) διδάσκει στο γυμνάσιο και το δημοτικό.

Κατά τη διεξαγωγή του σεμιναρίου οι επιμορφούμενοι κλήθηκαν να απαντήσουν σε ερωτήσεις μέσω του Διαδικτύου, σχετικές με την επίλυση προβλημάτων που αναφέρονταν στο αντικείμενο του σεμιναρίου. Από τις απαντήσεις τους αναδεικνύεται η ανάγκη για εμβάθυνση σε όλα τα προς διερεύνηση θέματα αλλά ειδικότερα στους κλώνους. Η ανάγκη αυτή για εμβάθυνση αποτελεί τη βάση για περαιτέρω έρευνα.

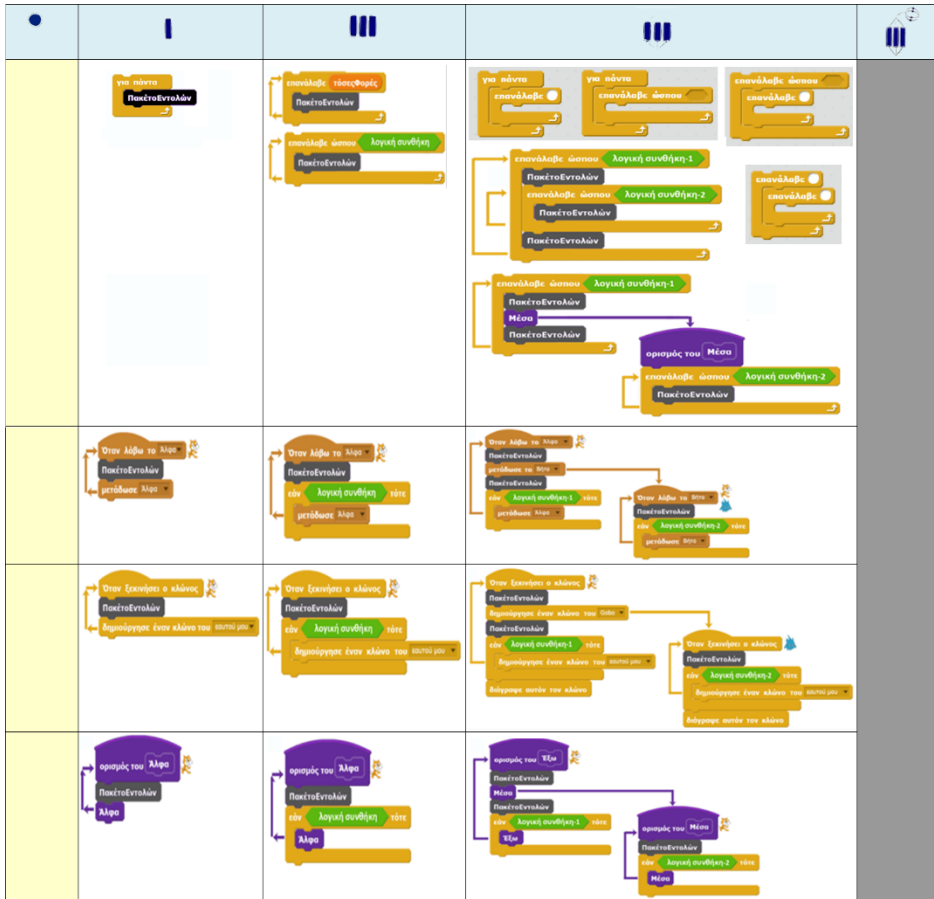


Σχήμα 9. Κατανομή των απαντήσεων στο ερωτηματολόγιο αξιολόγησης του σεμιναρίου.

6. Συμπεράσματα - επόμενα βήματα

Η παρούσα εργασία προτείνει μια ταξινόμηση των προγραμματιστικών βρόχων στο οπτικό περιβάλλον του Scratch, με την προοπτική να ενταχθεί σε μια ευρύτερη έρευνα αξιολόγησης του κώδικα. Μια πρώτη προσέγγιση του θέματος είναι η ένταξη της προτεινόμενης ταξινόμησης (του σχήματος 5), ως το τέταρτο και ανώτερο επίπεδο της εκτεταμένης αφαίρεσης της γνωστικής ταξινομίας μαθησιακών αποτελεσμάτων SOLO (Μπέλλου & Μικρόπουλος, 2008) όσον αφορά τη χρήση των προγραμματιστικών δομών υλοποίησης βρόχων (σχήμα 10). Τα υπόλοιπα επίπεδα της ταξινομίας είναι το πρώτο που καλύπτεται από τους ατέρμονες βρόχους, το δεύτερο που αποτελείται από απλούς βρόχους ενώ το τρίτο επίπεδο συνίσταται από συνδυασμούς εμφωλευμένων βρόχων ομοειδών σχημάτων. Στην κατηγοριοποίηση των δυνατών εναλλακτικών τρόπων υλοποίησης βρόχων χρησιμοποιήθηκε το ρεπερτόριο των επαναληπτικών εντολών όπως επίσης και η χρήση μηνυμάτων, κλώνων και αναδρομικών διαδικασιών. Οι συνδυασμοί σχημάτων και μοτίβων βρόχων που προέκυψαν μπορεί να αποτελέσουν έναν οδηγό αφενός για διδακτικές προσεγγίσεις και αφετέρου για εντοπισμό θεμάτων προς περαιτέρω μελέτη. Αυτό ενισχύεται από την ανατροφοδότηση που υπήρξε από τους εκπαιδευτικούς πληροφορικής οι οποίοι αφενός μεν προτίθενται να εντάξουν

πεδία του στη διδασκαλία τους το επόμενο σχολικό έτος και αφετέρου τη δήλωση της ανάγκης τους για επιπρόσθετες επιμορφωτικές δράσεις στα συγκεκριμένα θέματα.



Σχήμα 10. Τα δώδεκα εμφωλευμένα μη ομοειδή σχήματα προγραμματιστικών βρόχων αποτελούν το ανώτερο επίπεδο (της εκτεταμένης αφαίρεσης) της γνωστικής ταξινόμιας μαθησιακών αποτελεσμάτων SOLO όσον αφορά τη χρήση των προγραμματιστικών δομών υλοποίησης βρόχων.

Τέτοια θέματα μπορεί να είναι (α) η σειριακότητα ή η παραλληλία, της εκτέλεσης των διεργασιών των διαφόρων βρόχων, γιατί παρατηρείται ότι η εκτέλεση των εντολών των βρόχων των Πινάκων 1 και 2 γίνεται σειριακά σε όλα τα κελιά εκτός από αυτά με κίτρινο χρώμα (στο Σχήμα 2Γ αναφερόμαστε μόνο στον επάνω κώδικα), (β) ο συγχρονισμός των διεργασιών όπως αυτές του σχήματος 2Α ή όταν γίνεται με διαφορετικές εντολές ("μετάδωσε" και "μετάδωσε και περίμενε") και (γ) η διαχείριση αντίκρουσης συμβάντων με τεχνικές interrupt και polling.

7. Αναφορές

Λαδιάς, Α. (2018). *Αξιολόγηση σεμιναρίου για τα 4! σχήματα επανάληψης*. Πειραιάς: Γραφείο Σχολικών Συμβούλων Δευτεροβάθμιας Εκπαίδευσης (αρ. πρωτ. 62, 5/2/2018).

Μικρόπουλος, Α. & Λαδιάς, Α. (2000). *Η Logo στην εκπαιδευτική διαδικασία. Ιωάννινα*. Πανεπιστήμιο Ιωαννίνων.

Μπέλλου, Ι. & Μικρόπουλος, Α. (2008). Μέθοδος για την Ιεραρχική Αξιολόγηση Γνώσεων Προγραμματισμού. *4ο Συνέδριο Διδακτική Πληροφορικής*. ΕΤΠΕ. Αθήνα.

Manataki, A. & de Kereki, F. (2015). *Code Yourself! An Introduction to Programming*. MOOC: Coursera, Ανακτήθηκε στις 12 Απριλίου 2015 από <https://www.coursera.org/learn/intro-programming>

Papert, S. (1991). *Νοητικές θύελλες. Παιδιά, Ηλεκτρονικοί Υπολογιστές και Δυναμικές Ιδέες. Τα πάντα γύρω από τη Logo*. Αθήνα: Οδυσσέας.

Abstract

This paper proposes a classification of programming loops in Scratch's visual environment with the prospect of joining a wider, solo taxonomy-based, code evaluation. The categorization of the possible alternative ways of implementing loops was based on the study of the repertoire of iterative commands, the use of messages, clones, and recursive processes. The visual representation of this categorization led, on the one hand, to loop shapes and, on the other hand, to loop patterns that intersect these loop shapes. Moreover, latency loops were generated from the "expected ..." commands and the "When ..." event detection commands respectively. Finally, this proposal was presented in the form of a seminar in IT teachers. The outcome of this seminar, among others, highlighted the need to delve into particular aspects, such as the use of clones.

Keywords: Loops, iterations, recursiveness, clones, Scratch