

Βρόχοι υλοποιούμενοι με κλώνους στο Scratch και ταξινόμια SOLO

Θεόδωρος Καρβουνίδης¹, Δημήτριος Λαδιάς²

tkarv@otenet.gr, ladimitr@gmail.com

¹Μεταδιδακτορικός Ερευνητής, Τμήμα Πληροφορικής Παν. Πειραιώς

²Φοιτητής Τμήματος Πληροφορικής ΕΚΠΑ

Περίληψη. Για το θέμα των προγραμματιστικών δομών επανάληψης από εκπαιδευτική σκοπιά, υπάρχει μια εκτενής βιβλιογραφία, η οποία όμως τις περισσότερες φορές περιορίζεται στην υλοποίησή τους με τις εντολές επανάληψης. Από σεμινάριο που έγινε για εκπαιδευτικούς σχετικά με τους διάφορους τρόπους που μπορούν να δημιουργηθούν προγραμματιστικοί βρόχοι στο Scratch (με επαναληπτικές εντολές, με χρήση μηνυμάτων, κλώνων και αναδρομικών διαδικασιών) αναδείχθηκε η ανάγκη για περαιτέρω εμβάθυνση στους βρόχους που υλοποιούνται με τη βοήθεια κλώνων. Η παρούσα εργασία φιλοδοξεί να καταθέσει μια πρόταση για την ταξινόμηση των προγραμματιστικών βρόχων που υλοποιούνται με χρήση κλώνων στο οπτικό περιβάλλον του Scratch με την προοπτική να ενταχθεί σε μια ευρύτερη έρευνα αξιολόγησης του κώδικα με τη βοήθεια της ταξινόμιας SOLO.

Λέξεις κλειδιά: προγραμματιστικοί βρόχοι, επανάληψεις, κλώνοι, Scratch

Εισαγωγή

Η παρούσα εργασία εντάσσεται σε ένα ευρύτερο πλαίσιο για την αξιολόγηση του κώδικα του οπτικού προγραμματισμού με χρήση πλακιδίων (Karvounidis, Argyriou, Ladias & Douligeris, 2017). Τα κριτήρια είναι η αποτελεσματικότητα, η αναγνωσιμότητα, η βελτιστοποίηση, η μεταβλητότητα, η ασφάλεια, η δοκιμαστικότητα, η τμηματοποίηση, η αλληλεπίδραση με τον χρήστη, η αναπαράσταση των δεδομένων, η επικοινωνία/ο συγχρονισμός και η παραλληλία/σειριακότητα (Αργυρίου, Καρβουνίδης, Λαδιάς & Δουλγιέρης, 2016). Έχουν προηγηθεί μελέτες για την προγραμματιστική δομή επανάληψης (Λαδιάς, Καρβουνίδης, Λαδιάς & Δουλγιέρης, 2018), την επανάληψη με χρήση μηνυμάτων (Karvounidis, Ladias, Ladias. & Douligeris, 2019) και την προγραμματιστική δομή ακολουθίας (Λαδιάς, Μικρόπουλος, Πλεσιώτης & Λαδιάς, 2019), για την τμηματοποίηση του κώδικα (Λαδιάς, Πλεσιώτης & Λαδιάς, 2018) και για την ενδοδόμηση των εντολών σε περιβάλλοντα οπτικού προγραμματισμού (Λαδιάς, Μικρόπουλος, Πλεσιώτης & Λαδιάς, 2018). Το πεδίο της παρούσης εργασίας σε σχέση με τον όλο σχεδιασμό της έρευνας φαίνεται στο Σχήμα 1.

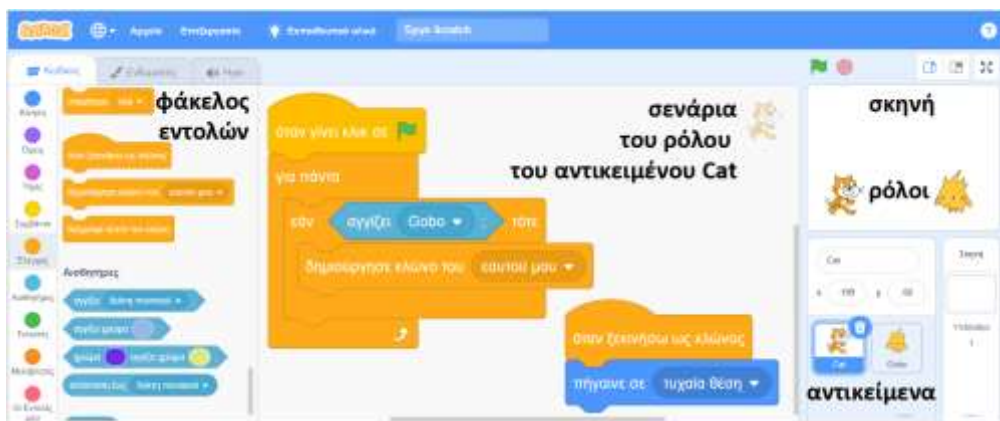


Σχήμα 1. Η θέση της παρούσης εργασίας στο πλαίσιο της ευρύτερης έρευνας για την αξιολόγηση του κώδικα στο Scratch

Στην έρευνα αυτή ως γλώσσα προγραμματιστικό και περιβάλλον οπτικού προγραμματισμού με πλακίδια έχει επιλεγεί το Scratch.

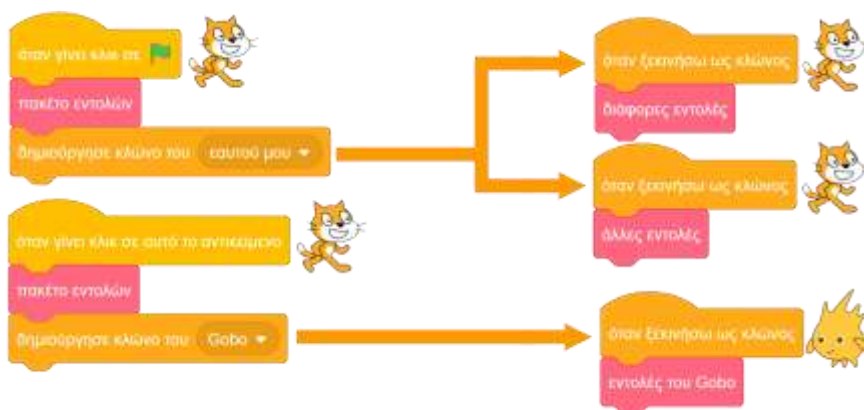
Η γλώσσα προγραμματισμού Scratch

Το Scratch (<http://scratch.mit.edu>) δημιουργήθηκε από το Lifelong Kindergarten Group στο Media Laboratory του MIT. Σύμφωνα με τους Resnick et al. (2009) τα προγράμματα στο Scratch αποτελούνται από αυτόνομα τμήματα κώδικα, που είναι προγραμματιστικά σενάρια και τα οποία ελέγχουν τη συμπεριφορά ρόλων (sprites) αντικειμένων (objects) που εμφανίζονται σε μια σκηνή.



Σχήμα 2. Αντικείμενα/objects, ρόλοι τους/sprites στη σκηνή και σενάρια που καθορίζουν τη συμπεριφορά τους.

Η ενεργοποίηση των σεναρίων προκαλείται από συμβάντα τα οποία στο πρόγραμμα δηλώνονται και ξεκινούν με μια εντολή-καπελάκι "όταν..". Μια περίπτωση δημιουργίας σεναρίου μπορεί να προκληθεί από την εντολή "δημιούργησε κλώνο του ...". Αυτή η εντολή ενεργοποιεί κάθε σενάριο του προγράμματος που ξεκινά με την εντολή-καπελάκι "όταν ξεκινήσω ως κλώνος" (Σχήμα 2). Ο μηχανισμός δημιουργίας κλώνου και απόδοσης συμπεριφοράς σε αυτόν αναπαριστάται στο σχήμα 3 με τους συνδέσμους-βέλη μεταξύ των σεναρίων.



Σχήμα 3. Μηχανισμός δημιουργίας κλώνων στο Scratch και απόδοση συμπεριφοράς σε αυτούς.

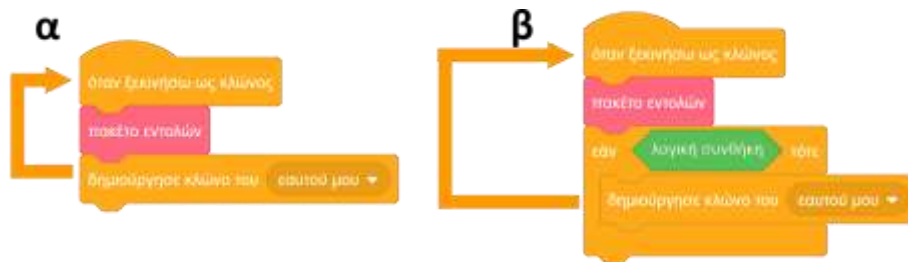
Είδη βρόχων που υλοποιούνται με κλώνους

Οι προγραμματιστικοί βρόχοι στο Scratch υλοποιούνται με τις εντολές επανάληψης, την αναδρομική κλήση διαδικασιών και τη χρήση μηνυμάτων. Στο Scratch υπάρχουν τρεις προγραμματιστικές δομές επανάληψης: Οι "επανάλαβε τόσες φορές", "επανάλαβε ώσπου" και "για πάντα" (Σχήμα 4).



Σχήμα 4. Βρόχοι υλοποιούμενοι με εντολές επανάληψης.

Ένας επιπλέον τρόπος δημιουργίας βρόχων είναι να χρησιμοποιηθεί για αυτό το σκοπό η δημιουργία κλώνου. Αυτό γίνεται στην περίπτωση που μέσα στο σενάριο "όταν ξεκινήσω ως κλώνος" υπάρχει η εντολή "δημιούργησε κλώνο του εαυτού μου" (Σχήμα 5).

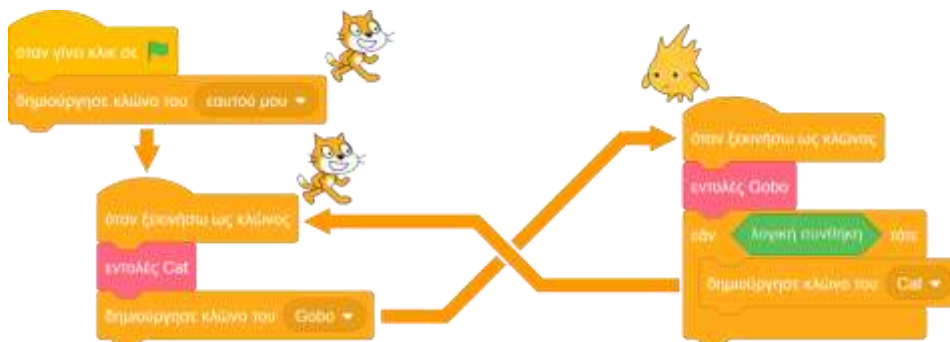


Σχήμα 5. Απλοί (α) ατέρμονες και (β) τερματίσιμοι βρόχοι υλοποιούμενοι με τη χρήση κλώνων.

Στη συνέχεια θα εξεταστούν ατέρμονες και τερματίσιμοι βρόχοι, υλοποιούμενοι με κλώνους, που διακρίνονται σε απλούς κυκλικούς, σε κυκλικούς τύπου Möbius, και σε εμφωλευμένους και σε διασταυρούμενους.

(α) **Απλοί κυκλικοί βρόχοι** δημιουργούμενοι με τη χρήση κλώνων μπορεί να είναι είτε ατέρμονες (Σχήμα 5α) είτε τερματίσιμοι (Σχήμα 5β).

(β) **Κυκλικοί βρόχοι τύπου Möbius.** Τους ονομάζουμε Möbius γιατί η απεικόνισή τους παραπέμπει στην κίνηση κατά μήκος της ομώνυμης ταινίας. Οι βρόχοι Möbius (Μικρόπουλος & Λαδιάς 2000) υλοποιούνται με χρήση κλώνων όταν τα δύο σενάρια ανήκουν σε διαφορετικούς ρόλους. Για τερματίσιμους βρόχους Möbius μπορεί να προστεθεί μια δομή ελέγχου σε κάποιο σημείο του κώδικα (Σχήμα 5).

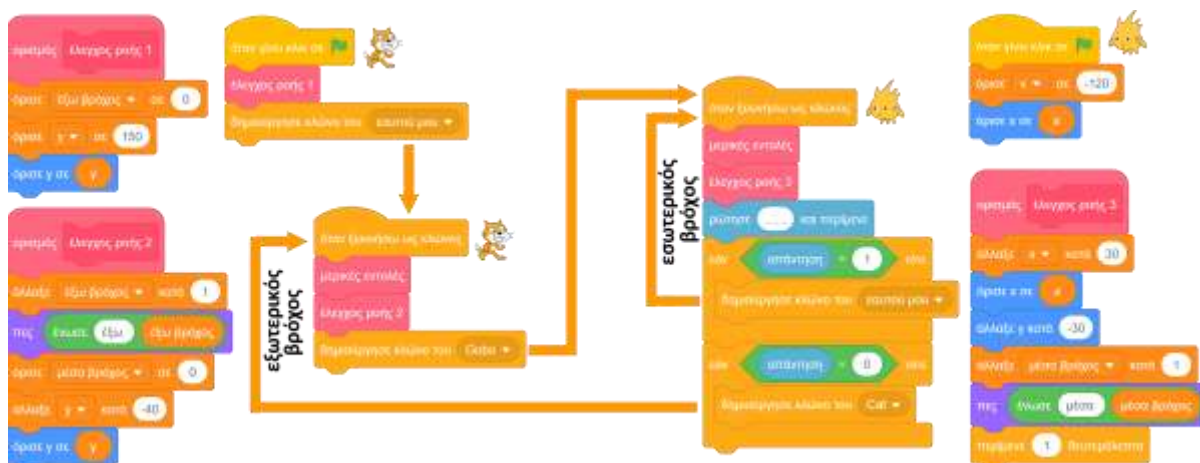


Σχήμα 6. Τερματίσιμος βρόχος τύπου Möbius που υλοποιείται με τη χρήση κλώνων.

(γ) **Οι εμφωλευμένοι βρόχοι** είναι κάτι σύνηθες στον προγραμματισμό με τη χρήση των εντολών επανάληψης. Αντίστοιχα μπορεί να δημιουργηθούν εμφωλευμένοι βρόχοι με τη χρήση κλώνων (Σχήμα 7). Σε αυτή την περίπτωση χρειάζεται μηχανισμός ώστε να μην υπάρχει παράλληλη / ταυτόχρονη εκτέλεση του εσωτερικού με τον εξωτερικό βρόχο.

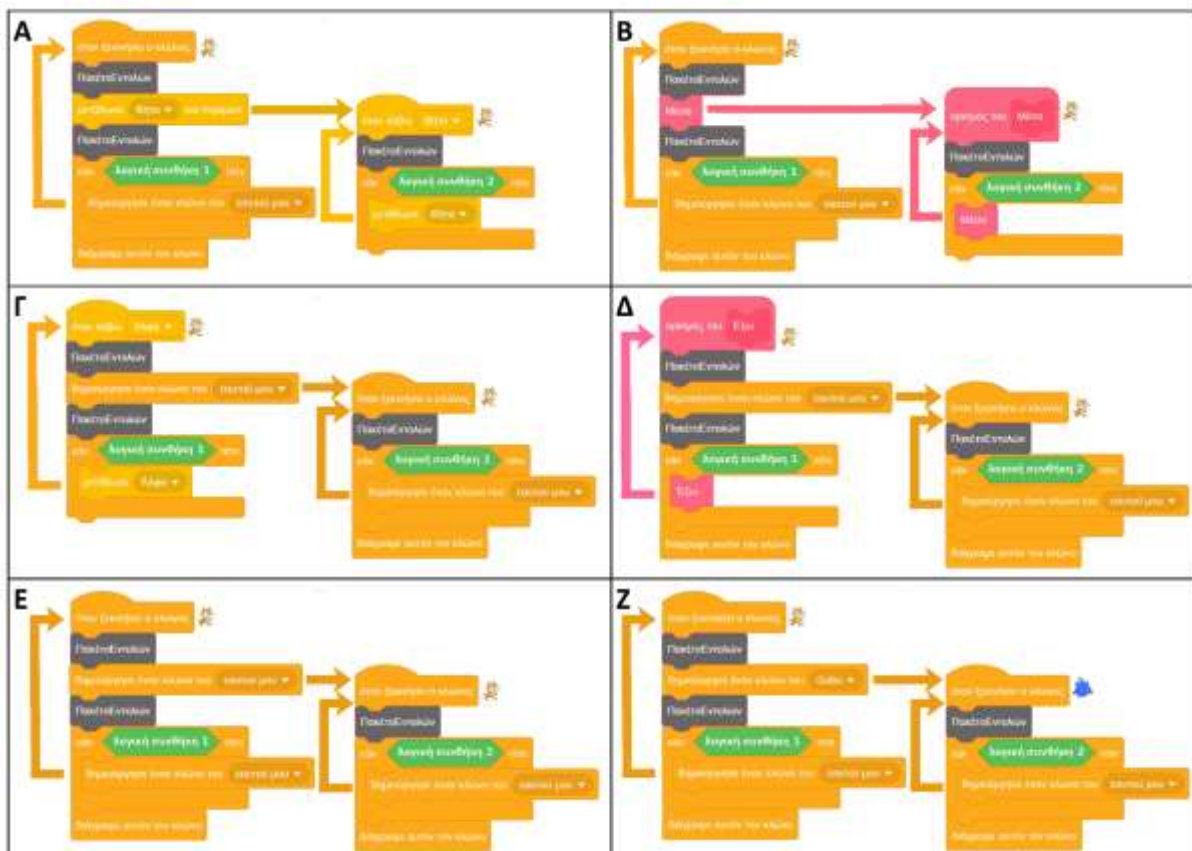
Επιπλέον μπορούμε να έχουμε μια ευρύτερη ποικιλία εμφωλευμένων βρόχων κατασκευάζοντας με κλώνο τον εξωτερικό βρόχο και τον εσωτερικό με εντολές επανάληψης ή με χρήση μηνυμάτων (Σχήμα 8Α) ή με αναδρομικές διαδικασίες (Σχήμα 8Β). Επίσης μπορεί ο εσωτερικός βρόχος να κατασκευαστεί με κλώνο και ο εξωτερικός βρόχος να υλοποιηθεί

με εντολές επανάληψης ή με χρήση μηνυμάτων (Σχήμα 8Γ) ή με αναδρομικές διαδικασίες (Σχήμα 8Δ).



Σχήμα 7. Εμφωλευμένοι βρόχοι με χρήση κλώνων.

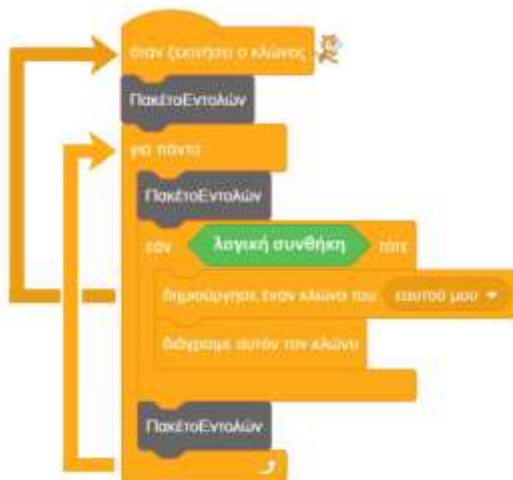
Πρέπει να σημειωθεί ότι η κατασκευή εμφωλευμένων βρόχων υλοποιούμενων -στο ίδιο αντικείμενο- με κλώνους εξωτερικά και εσωτερικά, οδηγεί σε ταυτόχρονη / παράλληλη εκτέλεση αυτών (Σχήμα 8Ε). Σε αυτή την περίπτωση είναι εμφανής η δυσκολία από τον προγραμματιστή να παρακολουθήσει τη ροή του προγράμματος. Αυτό το πρόβλημα δεν παρουσιάζεται αν οι εμφωλευμένοι βρόχοι γίνονται με κλώνους διαφορετικών αντικειμένων (Σχήμα 8Ζ).



Σχήμα 8. Εμφωλευμένοι βρόχοι υλοποιούμενοι με κλώνους σε συνδυασμό με μηνύματα και αναδρομή.

Η εμφώλευση ενός βρόχου μέσα σε άλλο θεωρείται ως εμφώλευση 1ου βαθμού. Είναι δυνατόν να υπάρξουν πολλαπλές εμφωλεύσεις βρόχων θεωρούμενες ως εμφωλεύσεις ν-ιστού βαθμού.

(δ) **Οι διασταυρούμενοι βρόχοι** είναι κάτι που συνήθως αποφεύγεται στον προγραμματισμό, ιδιαίτερα στο δομημένο προγραμματισμό, όμως είναι κάτι υπαρκτό. Στο Σχήμα 9 φαίνονται αφενός ο βρόχος της εντολής "για πάντα" και αφετέρου ο βρόχος που δημιουργείται όταν μεταξύ των εντολών του σεναρίου "όταν ξεκινήσει ο κλώνος" υπάρχει και η εντολή "δημιούργησε έναν κλώνο του εαυτού μου" που μπορεί να αντιστοιχεί στο "GOTO" της γλώσσας προγραμματισμού BASIC ή το "JMP" της Assembly του MC-6800.



Σχήμα 9. Διασταυρούμενοι βρόχοι υλοποιούμενοι με χρήση κλώνου και εντολής επανάληψης.

Η ταξινόμια SOLO

Η ταξινόμια SOLO (Structure of the Observed Learning Outcome) προτείνει την αξιολόγηση της γνώσης με βάση τη δομή του παρατηρούμενου μαθησιακού αποτελέσματος (Biggs & Collis, 1982). Σύμφωνα με τους Λαδιά κ.α. (2018), προτείνεται μια προσαρμογή της ταξινόμιας SOLO, στο αναφερόμενο πεδίο που ταξινομεί τα μαθησιακά αποτελέσματα σε πέντε επίπεδα με βάση τη δομή τους.

1. Το **προ-δομικό** επίπεδο, στο οποίο γίνεται αναφορά ή χρήση μη συνδεδεμένων και ανοργάνωτων πληροφοριών που δεν έχουν νόημα.
2. Το **μονο-δομικό** επίπεδο, όπου παρατηρείται μια περιορισμένη οπτική -κυρίως χρησιμοποιείται ή τονίζεται ένα στοιχείο ή μια πτυχή- ενώ παραλείπονται οι υπόλοιπες συνιστώσες και δεν πραγματοποιούνται σημαντικές συνδέσεις μεταξύ των μερών.
3. Το **πολύ-δομικό** επίπεδο, στο οποίο υπάρχει μια προοπτική πολλαπλών σημείων - χρησιμοποιούνται ή αναγνωρίζονται διάφορα σχετικά στοιχεία ή πτυχές- αλλά δεν υπάρχουν σημαντικές συνδέσεις και δεν έχει διαμορφωθεί ακόμη μια ολοκληρωμένη εικόνα.
4. Το **σχεσιακό** επίπεδο, στο οποίο υπάρχει μια ολιστική προοπτική όπου οι μετα-συνδέσεις μεταξύ των μερών γίνονται αντιληπτές και η σημασία των τμημάτων σε σχέση με το σύνολο αποδεικνύεται και εκτιμάται.

5. Το επίπεδο της **εκτεταμένης γενίκευσης**, στο οποίο επιπλέον των χαρακτηριστικών του προηγούμενου συσχετιστικού επιπέδου, το περιεχόμενο αντιμετωπίζεται ως ένα στιγμιότυπο μιας γενικότερης περίπτωσης».

Η ταξινόμια SOLO χρησιμοποιείται για την αξιολόγηση μαθησιακών αποτελεσμάτων αλλά και ολόκληρων μαθημάτων και προγραμμάτων σπουδών. Οι Μπέλλου & Μικρόπουλος (2008) προτείνουν το μοντέλο της Ιεραρχικής Αξιολόγησης Γνώσεων Προγραμματισμού (ΙΑΠ), που βασίζεται στην ταξινόμια SOLO και προτείνει πέντε ιεραρχικά επίπεδα που ορίζονται με άξονες αφενός την ανάπτυξη αλγοριθμικής σκέψης για την επίλυση προβλήματος, και αφετέρου τις δεξιότητες στη γλώσσα προγραμματισμού. Η ταξινόμια SOLO σε συνδυασμό με την αναθεωρημένη ταξινόμια Bloom έχει χρησιμοποιηθεί στην αξιολόγηση γνώσεων προγραμματισμού στο Scratch αναλύοντας σε βάθος τις γνώσεις των μαθητών (Meerbaum-Salant, Armoni & BenAri, 2013).

Αντιστοίχιση των τρόπων υλοποίησης βρόχων με κλώνους στις κατηγορίες της SOLO

Στη συνέχεια παρουσιάζονται οι αντιστοιχίσεις των διαφόρων τρόπων υλοποίησης βρόχων με κλώνους με τα επίπεδα της ταξινόμιας SOLO, ως δειγματικές απαντήσεις που κατατάσσονται στα πέντε ιεραρχικά επίπεδα.

Το **μονοδομικό επίπεδο** περιλαμβάνει απλούς κώδικες ατέρμωνων βρόχων, οι οποίοι υλοποιούνται χρησιμοποιώντας κλώνους όπως αυτά του Σχήματος 5α. Η χρήση τέτοιων κωδικών από τους μαθητές υποδηλώνει περιορισμένη αντίληψη στην οποία χρησιμοποιείται κυρίως ένα στοιχείο, ενώ τα άλλα στοιχεία παραλείπονται και δεν γίνονται σημαντικές συνδέσεις μεταξύ των μερών.

Στο **πολυδομικό επίπεδο**, οι κώδικες εκτελούνται σειριακά/διαδοχικά, αφενός όπως οι απλοί πεπερασμένοι βρόχοι του Σχ. 5β και αφετέρου, όπως οι βρόχοι Möbius, άπειροι ή/και πεπερασμένοι (Εικ. 6), οι οποίοι καθορίζουν τη συμπεριφορά ενός ή περισσότερων ρόλων αντικειμένων. Οι μαθητές που αναπτύσσουν τέτοιους κώδικες δείχνουν μια πολύπλευρη προοπτική. Ωστόσο, δεν έχουν ακόμη μια συνολική άποψη του εν λόγω θέματος.

Σε **σχεσιακό επίπεδο**, υπάρχουν κώδικες με εμφωλευμένους βρόχους οι οποίοι υλοποιούνται αμφότεροι με χρήση κλώνων. Αυτοί οι κώδικες ενσωματώνουν μια ολιστική προοπτική. Οι συνδυασμοί/συσχετίσεις μεταξύ των επιμέρους μερών είναι εμφανείς και αναδεικνύεται η σημασία τους σε σχέση με το σύνολο. Οι κώδικες που προσεγγίζουν αυτό το επίπεδο δείχνουν ότι οι μαθητές είναι σε θέση να συσχετίσουν: εμφωλευμένους βρόχους που εκτελούνται σειριακά και μηχανισμούς ελέγχου της ροής του προγράμματος προς αποφυγήν ανεξέλεγκτων παράλληλων βρόχων.

Στο **επίπεδο εκτεταμένης γενίκευσης**, οι κώδικες που αναπτύσσονται από τους μαθητές μπορούν να περιλαμβάνουν εμφωλευμένους βρόχους (υλοποιούμενους αμιγώς με κλώνους) με βαθμό εμφώλευσης μεγαλύτερο από δύο, βρόχους που εκτελούνται παράλληλα είτε συνδυασμούς με βρόχους που υλοποιούνται και με εντολές επανάληψης, με αναδρομή ή με χρήση μηνυμάτων.

Συμπεράσματα

Ο συνδυασμός των διαφορετικών τρόπων δημιουργίας βρόχων με χρήση κλώνων δημιουργεί ένα εργαλείο αξιολόγησης για τον οπτικό κώδικα προγραμματισμού. Αυτό το εργαλείο επιτρέπει στον καθηγητή να έχει μετρήσιμα δεδομένα σχετικά με τον βαθμό ωριμότητας του κώδικα ενός μαθητή σχετικά με τον τρόπο δημιουργίας βρόχων χρησιμοποιώντας κλώνους. Ο δάσκαλος μπορεί να εφαρμόσει αυτό το εργαλείο για να εκτιμήσει την κατανόηση των διαφόρων παραγόντων που εμπλέκονται από τον μαθητή, όπως εάν ο βρόχος τερματίζεται, εάν είναι εμφωλευμένος, εάν οι βρόχοι εκτελούνται διαδοχικά ή ταυτόχρονα, και εάν χρησιμοποιεί μια ποικιλία τεχνικών επανάληψης.

Όπως αναφέρθηκε προηγουμένως, η παρούσα εργασία εντάσσεται σε ένα ευρύτερο έργο που στοχεύει στην ανάπτυξη ενός πλαισίου για την αξιολόγηση του Κώδικα Οπτικού Προγραμματισμού με πλακίδια, το οποίο στοχεύει στη διερεύνηση, την κατηγοριοποίηση και τη θέσπιση κατάλληλων κριτηρίων αξιολόγησης οπτικού προγραμματισμού με τη βοήθεια της ταξινόμησης SOLO.

Αναφορές

- Αργυρίου, Ι., Καρβουνίδης, Θ., Λαδιάς, Α., & Δουληγέρης, Χ. (2016). Κριτήρια και Μοντέλο Αξιολόγησης Κώδικα Οπτικού Προγραμματισμού. *8th Conference on Informatics in Education 'Η Πληροφορική στην Εκπαίδευση'* (σ.492-503). Πειραιάς.
- Biggs, J. B., & Collis, K. F. (1982). *Evaluating the quality of learning. The SOLO taxonomy*. NY: Academic Press.
- Μικρόπουλος, Α., & Λαδιάς, Α. (2000). *Η Logo στην εκπαιδευτική διαδικασία*. Πανεπιστήμιο Ιωαννίνων.
- Λαδιάς Αν., Καρβουνίδης Θ., Λαδιάς Δ., & Δουληγέρης Χρ. (2018). Μια πρόταση ταξινόμησης των προγραμματιστικών βρόχων στο Scratch. *10th Conference on Informatics in Education 'Η Πληροφορική στην Εκπαίδευση'*, Θεσσαλονίκη.
- Λαδιάς, Δ., Μικρόπουλος, Α., Πλεσιώτης, Η., & Λαδιάς, Α. (2018). Εφαρμογή της ταξινομίας SOLO στην αξιολόγηση της δόμησης των εντολών σε περιβάλλοντα οπτικού προγραμματισμού. *Έρκυνα*, 15, 43-52.
- Λαδιάς, Δ., Μικρόπουλος, Α., Πλεσιώτης, Η., & Λαδιάς, Α. (2019). Η ταξινομία SOLO στις προγραμματιστικές δομές ακολουθίας στο Scratch. *6ο Πανελλήνιο Συνέδριο Ένταξη και Χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία*. Αθήνα.
- Λαδιάς, Δ., Πλεσιώτης, Η., & Λαδιάς, Α. (2018). Εφαρμογή της ταξινομίας SOLO στην αξιολόγηση της τμηματοποίησης του κώδικα σε Scratch. *5ο Πανελλήνιο Συνέδριο eTwinning "Αξιοποίηση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας στα συνεργατικά σχολικά προγράμματα στην Πρωτοβάθμια και Δευτεροβάθμια Εκπαίδευση"*, Ιωάννινα.
- Karvounidis, T., Argyriou, I., Ladias, A., and Douligeris, C. (2017). A Design and Evaluation Framework for Visual Programming Codes. *The IEEE Global Engineering Education Conference (EDUCON)*. Athens.
- Karvounidis, T., Ladias, A., Ladias, D., & Douligeris, C. (2019) «Kinds of loops implemented with messages in Scratch and the SOLO Taxonomy». *SEEDA-CECNSM 2019*, University of Piraeus.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with Scratch. *Computer Science Education*, 23(3), 239-264.
- Μπέλλου, Ι., & Μικρόπουλος, Α. (2008). Μέθοδος για την Ιεραρχική Αξιολόγηση Γνώσεων Προγραμματισμού. *4ο Συνέδριο Διδακτική Πληροφορικής. ΕΤΠΕ*. Αθήνα.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y. (2009). Scratch: Programming for All, *Communications of the ACM*, 52(11), 60-67.