

# Δυσκολίες μαθητών Δημοτικού με τη γλώσσα προγραμματισμού Scratch

Νικολός Δημήτρης<sup>1</sup>, Κόμης Βασίλης<sup>2</sup>  
dnikolos@upatras.gr, komis@upatras.gr

<sup>1</sup> Υποψήφιος Διδάκτορας, Τμήμα Επιστημών της Εκπαίδευσης και της Αγωγής στην Προσχολική Ηλικία, Πανεπιστήμιο Πατρών

<sup>2</sup> Καθηγητής, Τμήμα Επιστημών της Εκπαίδευσης και της Αγωγής στην Προσχολική Ηλικία, Πανεπιστήμιο Πατρών

## Περίληψη

Η εργασία αυτή περιγράφει τις δυσκολίες μαθητών της Ε' Δημοτικού κατά την εφαρμογή ενός εκπαιδευτικού σεναρίου για την εκμάθηση του προγραμματισμού με τη γλώσσα προγραμματισμού Scratch. Κάποιες από αυτές τις δυσκολίες υπήρχαν και στις παραδοσιακές γλώσσες προγραμματισμού, ενώ άλλες εμφανίζονται στις σύγχρονες γλώσσες με εκπαιδευτικό προσανατολισμό. Οι δυσκολίες που αντιμετώπισαν οι μαθητές κατηγοριοποιήθηκαν σε αρχικές δυσκολίες, δυσκολίες στην βασική δομή «για πάντα εάν» και δυσκολίες στις νέες δυνατότητες της γλώσσας Scratch 2.0. Τέλος, προτείνονται κάποιες λύσεις για την διαχείριση αυτών των δυσκολιών στο πλαίσιο του εκπαιδευτικού σεναρίου.

**Λέξεις κλειδιά:** προγραμματισμός, Scratch, δυσκολίες

## Εισαγωγή

Η τεχνολογία αποτελεί αναπόσπαστο μέρος της καθημερινής μας ζωής, είτε στο σπίτι ή στη δουλειά. Για να μπορούν οι μαθητές να είναι έτοιμοι να συμμετέχουν τόσο στην αγορά εργασίας όσο και στον ψηφιακό κόσμο που διαμορφώνεται θα χρειαστούν ένα νέο σύνολο δεξιοτήτων που ονομάστηκε «Υπολογιστική Σκέψη» (Wing, 2006; Berry, 2013). Η υπολογιστική σκέψη είναι υπερσύνολο του προγραμματισμού, όμως, αναμφίβολα ο προγραμματισμός είναι μια δεξιότητα που θα πρέπει να καλλιεργηθεί (Brennan & Resnick, 2012; Lye & Koch, 2014).

Η γλώσσα προγραμματισμού Scratch έχει προτιμηθεί από πολλούς εκπαιδευτικούς για την διδασκαλία του προγραμματισμού στην Πρωτοβάθμια Εκπαίδευση. Στο πρόγραμμα σπουδών για τις ΤΠΕ στην Πρωτοβάθμια Εκπαίδευση για το «ΝΕΟ ΣΧΟΛΕΙΟ», η γλώσσα προγραμματισμού Scratch ήταν μία από τις εναλλακτικές γλώσσες για την ενότητα «Προγραμματίζω τον υπολογιστή». Η ενότητα αυτή θα διδάσκονταν για 10 ώρες στην Ε' Δημοτικού και για 12 ώρες στην Στ' Δημοτικού. Στις οδηγίες διδασκαλίας για το Σχολικό Έτος 2016-17 (Φ20/1835/222136/Δ1 - 28-12-2016) η Scratch εισάγεται στην ενότητα «Προγραμματίζω τον υπολογιστή» η οποία καταλαμβάνει τέσσερις (4) διδακτικές ώρες στην Ε' και την Στ' Δημοτικού με ενδεικτικές δραστηριότητες από την πλατφόρμα Αίσωπος.

Στην παρούσα εργασία χρησιμοποιήθηκε ένα εκπαιδευτικό σενάριο δέκα (10) διδακτικών ωρών για την διδασκαλία του προγραμματισμού. Το εκπαιδευτικό σενάριο είναι ένα εργαλείο που χρησιμοποιήθηκε στην Επιμόρφωση Εκπαιδευτικών για την Αξιοποίηση και Εφαρμογή των Ψηφιακών Τεχνολογιών στην Διδακτική Πράξη (Επιμόρφωση Β' επιπέδου στις Τ.Π.Ε.). Κατά το σχεδιασμό του εκπαιδευτικού σεναρίου είναι σημαντικό να γίνεται διερεύνηση των προηγούμενων αντιλήψεων των μαθητών, των

παρανοήσεων και των δυσκολιών που μπορεί να εμπλέκονται στην εκμάθηση του νέου αντικειμένου (Komis, et al., 2013).

Από τα πρώτα χρόνια της διδασκαλίας του προγραμματισμού είχε εμφανιστεί η ανάγκη για τη διερεύνηση των δυσκολιών που αντιμετωπίζουν οι αρχάριοι προγραμματιστές. Κάποιες από αυτές τις δυσκολίες προκύπτουν από το συντακτικό της γλώσσας ενώ άλλες μπορεί να προκύπτουν από την ερμηνεία των λέξεων της γλώσσας προγραμματισμού με όρους φυσικής γλώσσας (Mayer, 1988). Στις συνηθισμένες δυσκολίες περιλαμβάνονται, η ελλιπής κατανόηση της μηχανής που εκτελεί το πρόγραμμα (notional machine) και η χρήση αναλογιών που δεν ανταποκρίνονται στην υπολογιστική έννοια αλλά σε παρόμοιες στην καθημερινότητα ή σε άλλα γνωστικά αντικείμενα, όπως η μεταβλητή στα μαθηματικά (duBoulay, 1986). Μια σημαντική δυσκολία είναι η αντίληψη πως ο υπολογιστής είναι αρκετά «ανθρώπινος» καθώς διερμηνεύει τα προγράμματα (Putnam et al., 1986). Κάθε έννοια που εμπλέκεται στον προγραμματισμό εμφανίζει παρανοήσεις και δυσκολίες που σχετίζονται ειδικά με αυτή (Κόμης, 2005).

Τα σύγχρονα εκπαιδευτικά περιβάλλοντα περιλαμβάνουν οπτικό προγραμματισμό με τον οποίο αποφεύγονται τα συντακτικά λάθη. Αυτή η προσέγγιση είναι και ένα από τα βασικά πλεονεκτήματα της γλώσσας Scratch (Resnick et al., 2009). Όμως, πολλά από τα υπόλοιπα προβλήματα συνεχίζουν να εμφανίζονται. Ακόμη και μεγαλύτεροι σε ηλικία μαθητές δυσκολεύονται στην κατανόηση προγραμματιστικών δομών (Σαρημπαλίδης, 2012) και στην αρχικοποίηση μεταβλητών (Κατσούλας κ.α., 2013). Στο νέο περιβάλλον προκύπτουν και νέες δυσκολίες. Οι αρχάριοι προγραμματιστές εμφανίζουν τη συνήθεια του κατακεραματισμού του κώδικά τους σε πολλά τμήματα που εκτελούνται ταυτόχρονα (Meerbaum-Salant et al., 2011) και δυσκολεύονται να διαχειριστούν τον ταυτόχρονο χαρακτήρα του περιβάλλοντος (Maloney et al., 2010). Επίσης, εμφανίζουν την τάση να διπλασιάζουν εντολές επαναλαμβάνοντας κώδικα που επιτελεί την ίδια λειτουργία (Moreno & Robles, 2014).

Οι δυσκολίες και οι παρανοήσεις που εμφανίζονται όταν μαθητές Δημοτικού διδάσκονται τη γλώσσα προγραμματισμού Scratch είναι το αντικείμενο της εργασίας. Στην επόμενη ενότητα παρουσιάζεται η μεθοδολογία της έρευνας. Στην τρίτη ενότητα παρουσιάζονται οι δυσκολίες και οι παρανοήσεις των μαθητών όπως αυτές εμφανίστηκαν. Στην τελευταία ενότητα αναφέρονται τρόποι αντιμετώπισης αυτών των δυσκολιών.

## Μεθοδολογία

Η έρευνα διεξήχθη κατά την διδασκαλία του μαθήματος «Τεχνολογίες της Πληροφορίας και των Επικοινωνιών» σε δύο τμήματα της Ε' Τάξης. Για τα αντικείμενα της έρευνας χρησιμοποιήθηκε βολική δειγματοληψία (Cresswell, 2002). Η δύναμη των τμημάτων φαίνεται στον Πίνακα 1.

**Πίνακας 1. Δύναμη τμημάτων**

Τμήμα	Αγόρια	Κορίτσια	Σύνολο
E1	15	8	23
E2	12	11	23
<b>Σύνολο</b>	<b>27</b>	<b>19</b>	<b>46</b>

Κανείς από τους μαθητές δεν είχε ασχοληθεί με κάποιου είδους προγραμματισμό, ενώ είχαν διαφορετικό βαθμό εξοικείωσης με τη χρήση του υπολογιστή όπως είχε φανεί στις

δραστηριότητες που προηγήθηκαν πριν τη διδασκαλία του προγραμματισμού. Κατά τη διάρκεια του σεναρίου σχημάτισαν ομάδες δύο ή τριών ατόμων, ενώ ένας μικρός αριθμός μαθητών εργάστηκε ατομικά.

**Πίνακας 2. Η δομή του εκπαιδευτικού σεναρίου**

Διδακτική ώρα	Τίτλος	Περιγραφή
1 <sup>η</sup>	Εισαγωγή	Εισαγωγή στο περιβάλλον της γλώσσας Scratch
2 <sup>η</sup>	Όψεις	Εντολές που διαχειρίζονται την εμφάνιση των αντικειμένων. Αρχικοποίηση εμφάνισης.
3 <sup>η</sup>	Αλληλεπίδραση	Η αλληλεπίδραση υλοποιείται με την εντολή «εάν ... τότε...»
4 <sup>η</sup>	Μηνύματα	Αποστολή και παραλαβή μηνυμάτων για συγχρονισμό αντικειμένων.
5 <sup>η</sup>	Μεταβλητές	Υλοποίηση σκορ για ένα παιχνίδι. Αρχικοποίηση μεταβλητών.
6 <sup>η</sup>	Επανάληψη	Επανάληψη των εννοιών που έχουν διδαχθεί.
7 <sup>η</sup>	Δημιουργία νέων εντολών	Δημιουργία διαδικασιών από υπάρχουσες εντολές με χρήση γεωμετρίας χελώνας
8 <sup>η</sup>	Κλώνοι	Ανάπτυξη ενός παιχνιδιού με κλώνους
9 <sup>η</sup> - 10 <sup>η</sup>	Δημιουργήστε το δικό σας παιχνίδι	Οι μαθητές δημιουργούν τα δικά τους έργα με όσα γνωρίζουν.

Το σενάριο που χρησιμοποιήθηκε για τη διδασκαλία της γλώσσας Scratch βασίστηκε στη σειρά των εννοιών που προτάθηκε στο (Νικολός & Κόμης, 2010). Δύο λόγοι ωστόσο, επέβαλαν την αναθεώρηση αυτής της προσέγγισης. Ο πρώτος είναι πως οι μαθητές δεν μπορούν να ακολουθήσουν την προσέγγιση των φοιτητών. Ο δεύτερος λόγος είναι πως για την παρούσα έρευνα χρησιμοποιήθηκε η έκδοση Scratch 2.0 η οποία περιλαμβάνει τη δυνατότητα δημιουργίας εντολών και δημιουργίας κλώνων. Οι δύο αυτές δυνατότητες θα πρέπει να αποτελούν αντικείμενα διδασκαλίας ακόμη και στους μαθητές Δημοτικού αφού δίνουν τα εργαλεία για την αποφυγή του διπλασιασμού κώδικα. Η δομή του εκπαιδευτικού σεναρίου φαίνεται στον Πίνακα 2.

### Συλλογή δεδομένων

Για να διερευνηθούν οι δυσκολίες που αντιμετώπισαν οι μαθητές κατά την εφαρμογή του σεναρίου καταγράφηκαν τα εξής δεδομένα, στο πλαίσιο της έρευνας βασισμένης στο σχεδιασμό (Kafai, 2005):

1. Οι παρατηρήσεις του ερευνητή, στο πλαίσιο της συμμετοχικής παρατήρησης (Cresswell, 2002).
2. Τα έργα των μαθητών, τόσο στην πορεία των μαθημάτων όσο και τα τελικά.
3. Στιγμιότυπα οθόνης από διδακτικά επεισόδια που παρουσίασαν ενδιαφέρον.
4. Καταγραφές οθονών από ορισμένους μαθητές για την παρατήρηση του τρόπου με τον οποίο διαχειρίζονται τις δυσκολίες που συναντούν.

Αντικείμενο της παρούσας εργασίας αποτελούν οι δυσκολίες που αντιμετωπίζουν οι μαθητές κατά την ανάπτυξη προγραμμάτων. Άλλες δυσκολίες που παρατηρήθηκαν και

αφορούν το προγραμματιστικό περιβάλλον μέσα στο οποίο λειτουργεί ο μαθητής π.χ. περιστροφή αντικειμένων, δεν αναφέρονται στην παρούσα εργασία.

## Οι δυσκολίες των μαθητών

Στην πρώτη υποενότητα περιγράφονται κάποια αρχικά λάθη των μαθητών. Στην δεύτερη υποενότητα περιγράφονται οι δυσκολίες που διαπιστώθηκαν στην βασική δομή «για πάντα εάν». Στην τρίτη υποενότητα περιγράφονται οι δυσκολίες που εντοπίστηκαν στα νέα χαρακτηριστικά της γλώσσας Scratch δηλαδή στη δημιουργία νέων εντολών και στους κλώνους.

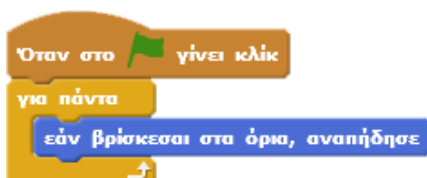
### Αρχικές δυσκολίες

Η πρώτη δυσκολία που αντιμετωπίζουν οι αρχάριοι προγραμματιστές στη συγγραφή προγραμμάτων στις παραδοσιακές γλώσσες προγραμματισμού είναι η εναρμόνισή τους με το συντακτικό της γλώσσας. Τα συντακτικά λάθη αναγνωρίζονται από τους μεταφραστές με αντίστοιχα μηνύματα λάθους, αλλά αυτή η δυσκολία δεν είναι εύκολο να ξεπεραστεί και τέτοια λάθη εμφανίζονται συχνά (McCall & Kolling, 2014). Η γλώσσα προγραμματισμού Scratch έχει ως βασικό πλεονέκτημα, ήδη από τις πρώτες εκδόσεις της, την αποφυγή συντακτικών λαθών με χρήση εντολών-μπλοκ που συνδυάζονται μόνο με σωστούς συντακτικά τρόπους (Maloney et al., 2004). Ωστόσο, στα πρώτα μαθήματα οι μαθητές βάζουν τις εντολές σε κοντινές θέσεις αλλά δεν γνωρίζουν τον τρόπο με τον οποίο αυτές συνδυάζονται (Σχήμα 1). Αυτή η δυσκολία αντιμετωπίστηκε εύκολα αλλά εμφανιζόταν ξανά στην περίπτωση που οι μαθητές χρειαζόταν να κάνουν συνδυασμούς που δεν είχαν ξανακάνει.

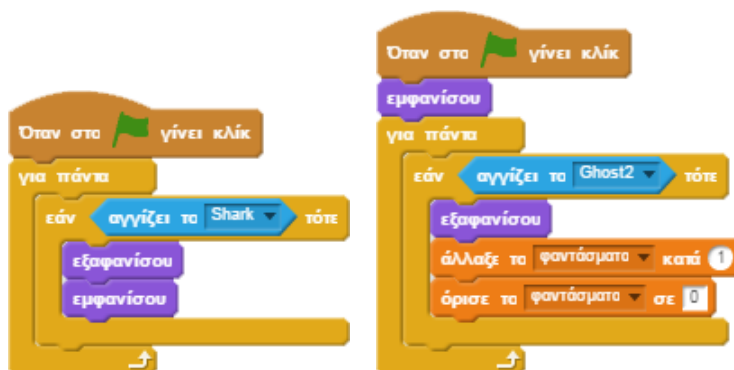


Σχήμα 1. Εντολές που δεν έχουν συνδυαστεί σωστά

Όπως και σε άλλες γλώσσες έτσι και στη Scratch, η κατανόηση εντολών με όρους καθημερινότητας δημιουργεί προβλήματα στους αρχάριους προγραμματιστές. Η εντολή «εάν βρίσκεσαι στα όρια, αναπήδησε» αλλάζει μόνο την κατεύθυνση του αντικειμένου και όχι τη θέση του. Ωστόσο, οι μαθητές περιμένουν ότι η εντολή αυτή προκαλεί κάποια κίνηση (Σχήμα 2).



## Σχήμα 2. Προσέγγιση εντολής με όρους καθημερινής γλώσσας



(α) Δύο απόλυτες εντολές

(β) Μία σχετική και μία απόλυτη εντολή

### Σχήμα 3. Εντολές που το αποτέλεσμά τους αναιρείται

Καθώς η Scratch εκτελεί τις συνεχόμενες εντολές γρήγορα, είναι πιθανό το αποτέλεσμα μιας εντολής να αναιρείται από την αμέσως επόμενη. Σε αυτή την περίπτωση οι μαθητές δυσκολεύονται να παρακολουθήσουν το πρόγραμμά τους και εκφράζουν απορίες για τον τρόπο με τον οποίο λειτουργεί. Παραδείγματα αυτής της συμπεριφοράς αποτελούν συνεχόμενες εντολές «εμφανίσου» και «εξαφανίσου» (Σχήμα 3α). Την κατάσταση δυσχεραίνουν ακόμη περισσότερο οι δύο τύποι εντολών της γλώσσας Scratch (Armoni & Ben-Ari, 2013, σελ. 17), οι απόλυτες εντολές των οποίων η εκτέλεση δεν εξαρτάται από την τρέχουσα κατάσταση (όπως η «όρισε τη μεταβλητή σε») και οι σχετικές εντολές των οποίων η εκτέλεση εξαρτάται από την τρέχουσα κατάσταση (όπως η «άλλαξε τη μεταβλητή κατά») (Σχήμα 3β).

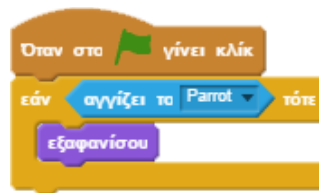
Στη γλώσσα προγραμματισμού Scratch μπορούν να δημιουργηθούν προγράμματα των οποίων η συμπεριφορά δεν μπορεί να προβλεφθεί μέχρι να εκτελεστεί το πρόγραμμα (Armoni & Ben-Ari, 2013, σελ. 155). Η ταυτόχρονη εκτέλεση εντολών δυσκολεύει τους μαθητές, ήδη από τις πρώτες προσπάθειες τους στην εκμάθηση προγραμματισμού. Ένα παράδειγμα είναι το πρόγραμμα του Σχήματος 4. Μάλιστα, υπάρχει το ενδεχόμενο το πρόγραμμα να λειτουργεί με τον επιθυμητό τρόπο μέχρι να συμβεί κάποια τροποποίηση σε κάποιο άλλο σημείο του κώδικα.



Σχήμα 4. Κώδικας του οποίου η συμπεριφορά δεν μπορεί να προβλεφθεί

### Δυσκολίες στην δομή «για πάντα εάν»

Μέχρι την έκδοση 1.4 της γλώσσας προγραμματισμού Scratch υπήρχε η εντολή «για πάντα εάν», η οποία αποτέλεσε πηγή έντονου προβληματισμού και στην έκδοση 2.0 καταργήθηκε ώστε οι χρήστες να χρησιμοποιούν την «για πάντα» με εμφωλευμένα ένα ή περισσότερα «εάν» ([https://wiki.scratch.mit.edu/wiki/Forever\\_If\\_\(block\)](https://wiki.scratch.mit.edu/wiki/Forever_If_(block))). Πρόκειται για μια δομή η οικοδόμηση της οποίας φαίνεται να δυσκολεύει τους μαθητές σε αρκετά σημεία.



Σχήμα 5. Εντολές χωρίς το απαραίτητο «για πάντα»

Σε κάποιες περιπτώσεις, οι μαθητές δεν εισάγουν καθόλου την εντολή «για πάντα» (Σχήμα 5), με αποτέλεσμα το πρόγραμμά τους να μην έχει την επιθυμητή συμπεριφορά. Αν και η έλλειψη του «για πάντα» μοιάζει με αβλεψία, τα λάθη αυτά θυμίζουν τα «parallelism bugs» που παρατηρήθηκαν ακόμη και σε γλώσσες προγραμματισμού που δεν χαρακτηρίζονταν από ταυτόχρονη εκτέλεση εντολών, όπως η Basic. Οι μαθητές φαίνεται να θεωρούν ότι η εντολή «εάν» περιμένει μέχρι η συνθήκη της να γίνει αληθής (Pea, 1986), μια προσέγγιση που έρχεται σε συμφωνία με τη φυσική γλώσσα, π.χ. «Αν χιονίσει, θα κλείσουν τα σχολεία». Κατά την εφαρμογή του σεναρίου σε πολλές περιπτώσεις το πρόβλημα επιλύθηκε με την συμβολή του εκπαιδευτικού αφού οι μαθητές δυσκολευόντουσαν να το εντοπίσουν μέσα στον κώδικα.



Σχήμα 6. Λάθος σειρά εντολών «για πάντα» και «εάν»

Οι μαθητές αντιμετωπίζουν προβλήματα με την εμφώλευση των εντολών «για πάντα» και «εάν». Αν και οι διαφορές στην εκτέλεση μεταξύ του «για πάντα» «εάν» σε σχέση με το «εάν» «για πάντα» είναι σημαντικές (Lewis, 2013), οι μαθητές δεν μπορούν εύκολα να εντοπίσουν το πρόβλημα που δημιουργείται (Σχήμα 6).

Στο σενάριο που εφαρμόστηκε, οι εμφωλευμένες δομές επιλογής δεν ήταν αντικείμενο διδασκαλίας, ωστόσο ένας μαθητής που θέλησε να ελέγξει πολλά ενδεχόμενα κατέληξε σε μια λύση εμφωλευμένων «εάν» τη ροή της οποίας είχε παρανοήσει (Σχήμα 7). Η παρακολούθηση της ροής εκτέλεσης του προγράμματος παρουσιάζει δυσκολίες παρά τα οπτικά βοηθήματα που παρέχει ο οπτικός τρόπος προγραμματισμού (Νικολός & Κόμης, 2011).



Σχήμα 7. Εμφωλευμένες «εάν» αντί για «εάν αλλιώς»

Στο Σχήμα 8 φαίνονται και άλλα παραδείγματα παρανόησης της ροής του προγράμματος. Στο παράδειγμα του Σχήματος 8α η εξαφάνιση είναι στιγμιαία και δεν είναι ορατή στον προγραμματιστή αφού στην επόμενη επανάληψη του «για πάντα» το αντικείμενο εμφανίζεται αμέσως ξανά. Στο παράδειγμα του Σχήματος 8β, η διαγραφή του κλώνου θα έπρεπε να γίνει μέσα στην εντολή «εάν» και όχι έξω από αυτή. Ειδικά σε αυτό το πρόγραμμα το αποτέλεσμα της εντολής «Όταν ξεκινήσει ο κλώνος» δεν είναι ορατό στον αρχάριο προγραμματιστή γιατί ο κλώνος δημιουργείται και διαγράφεται πολύ γρήγορα, δημιουργώντας προβλήματα στην κατανόηση της συμπεριφοράς του προγράμματος.



(α) Λάθος στην εντολή «εμφανίσει»

(β) Λάθος στην εντολή «διάγραψε αυτόν τον κλώνο»

Σχήμα 8. Προβλήματα στη ροή του προγράμματος

Λόγω της συνήθειας του κατακερματισμού του κώδικα (Meerbaum-Salant et al., 2011) οι μαθητές φαίνεται πως προβληματίζονται στο να παρακολουθήσουν τις εντολές που εκτελούνται ταυτόχρονα. Έτσι για παράδειγμα το «για πάντα» στο «Όταν το πλήκτρο [space] πατηθεί» αναιρεί το αποτέλεσμα του εξαφανίσου που γίνεται υπό συνθήκη (Σχήμα 9). Στην πραγματικότητα δεν χρειάζεται το πλήκτρο space (κενό) για αρχικοποίηση. Η εντολή «εμφανίσου» θα μπορούσε να εισαχθεί κάτω από την πράσινη σημαία. Παρόλα αυτά οι μαθητές δείχνουν να προτιμούν λύσεις με εξωτερικά γεγονότα σε αρκετές περιπτώσεις, όπως είχε παρατηρηθεί και στο (Nikolos & Komis, 2015).



Σχήμα 9. Εντολές που εκτελούνται ταυτόχρονα

### Δυσκολίες στις νέες εντολές και τους κλώνους

Όταν οι μαθητές διδάχθηκαν την έννοια των νέων εντολών που μπορούν να ορίσουν οι ίδιοι αντιμετώπισαν πολλά προβλήματα. Αυτό που εμφανίστηκε συχνότερα στα δεδομένα της παρούσας έρευνας ήταν η σύγχυση του μπλοκ της κλήσης της εντολής με το μπλοκ του ορισμού (Σχήμα 10).

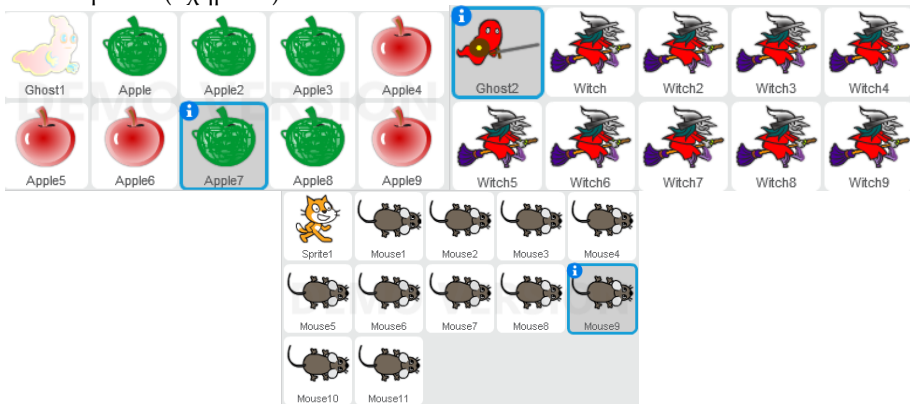


Σχήμα 10. Σύγχυση του μπλοκ κλήσης με το μπλοκ ορισμού

Όταν οι μαθητές διδάχθηκαν την έννοια των κλώνων, ήρθαν αντιμέτωποι με παιχνίδια που απαιτούσαν πολλά όμοια αντικείμενα. Η πρώτη τους αντίδραση ήταν να τα αντιμετωπίσουν όπως ήδη γνώριζαν, δημιουργώντας αντίγραφα των αντικειμένων. Στην πορεία, ενώ ήδη είχαν χρησιμοποιήσει κλώνους, για να δημιουργήσουν περισσότερα ίδια

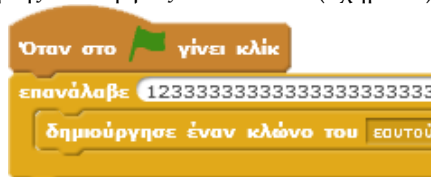


αντικείμενα αντί να αλλάζουν τον αριθμό των κλώνων επέμεναν να κάνουν διπλασιασμό των αντικειμένων (Σχήμα 11).



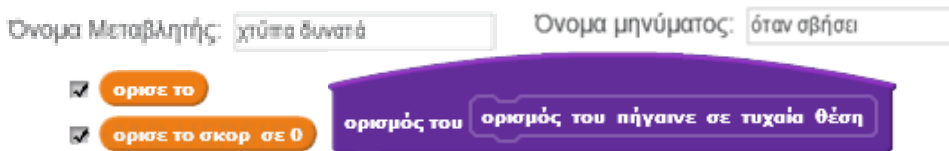
Σχήμα 11. Αντίγραφα αντικειμένων αντί κλώνων

Στην κατανόηση των προγραμμάτων με κλώνους αντιμετώπισαν επίσης μεγάλες δυσκολίες επειδή στη γλώσσα Scratch οι κλώνοι δημιουργούνται σχετικά αργά και ταυτόχρονα μπορεί κάποιος να διαγράφονται, π.χ. τα ποντίκια τα τρώει η γάτα. Έτσι στους μαθητές δίνεται η εντύπωση ότι οι κλώνοι δημιουργούνται και διαγράφονται με αυθαίρετο τρόπο. Το γεγονός πως δοκιμάζαν μεγάλο αριθμό από κλώνους για να πειραματιστούν δεν βοήθησε στην κατανόηση της λειτουργίας των κλώνων (Σχήμα 12).



Σχήμα 12. Δημιουργία κλώνων

Οι μαθητές επίσης προβληματίζονται στα ονόματα που δίνουν στα μηνύματα (Nikolos & Komis, 2015), τις μεταβλητές και τις νέες εντολές (Σχήμα 13). Σε αυτό το σημείο σημειώνεται πως οι μαθητές σε καμία περίπτωση δεν αλλάζανε το όνομα κάποιου αντικειμένου που ήδη είχε όνομα ώστε να είναι πιο αντιπροσωπευτικό, όπως παρατηρήθηκε και από τους Moreno & Robles (2014).



Σχήμα 13. Μη αντιπροσωπευτικά ονόματα

Στην ενότητα αυτή περιγράφηκαν οι δυσκολίες που αντιμετώπισαν οι μαθητές της Ε' Δημοτικού ενός Δημοτικού Σχολείου κατά τη διάρκεια της εφαρμογής ενός σεναρίου εκμάθησης Scratch, 10 ωρών.

### Τρόποι αντιμετώπισης

Η παρούσα έρευνα έγινε στο πλαίσιο μιας έρευνας βασισμένης στο σχεδιασμό που αφορά την ανάπτυξη της προγραμματιστικής ικανότητας μαθητών με χρήση οπτικών γλωσσών προγραμματισμού. Οι δυσκολίες που εντοπίστηκαν σε αυτόν τον κύκλο υλοποίησης θα αποτελέσουν το εφαλτήριο για την προσέγγιση του επόμενου κύκλου (Barab & Squire, 2004).

Βασικό στοιχείο στον επόμενο κύκλο θα αποτελέσουν οι ομάδες εντολών (chunks). Τέτοιες ομάδες εντολών υπήρχαν και στο σενάριο που εφαρμόστηκε στην παρούσα έρευνα (Σχήμα 14), ωστόσο όταν οι μαθητές αντιμετώπιζαν πρόβλημα δινόταν έμφαση σε κάθε εντολή ξεχωριστά, ακολουθώντας την αναπαράσταση που αποκτούν οι αρχάριοι προγραμματιστές οι οποίοι διαβάζουν τον κώδικα γραμμή-γραμμή. Αντίθετα, οι έμπειροι προγραμματιστές προσεγγίζουν τον προγραμματισμό έχοντας οικοδομήσει ομάδες εντολών με σημασιολογική λειτουργία (Hoc, 2015, σελ. 107). Οι κάρτες Scratch (Scratch cards), ένα εργαλείο υποστήριξης της διδασκαλίας από την επίσημη ιστοσελίδα του λογισμικού, περιέχουν ομάδες εντολών που επιτελούν μια συγκεκριμένη λειτουργία. Σε αυτή την προσέγγιση έχει δοθεί ακόμη μεγαλύτερη έμφαση με την έκδοση αντίστοιχου βιβλίου (Rusk, 2016).



Σχήμα 14. Τυπικός κώδικας αντικειμένων παιχνιδιού

Η βασική δραστηριότητα του σεναρίου όπως αυτό σχεδιάστηκε ήταν η δημιουργία κώδικα από τους μαθητές με βάση φύλλα εργασιών. Ωστόσο, φαίνεται πως η προσέγγιση αυτή πρέπει να εμπλουτιστεί με άλλες δραστηριότητες, διαφόρων τύπων (Κορδάκη & Ψώμος, 2012).

Κάποια από τα προβλήματα που εντοπίστηκαν στην παρούσα μελέτη θα αντιμετωπιστούν στον επόμενο κύκλο εφαρμογής του εκπαιδευτικού σεναρίου. Ωστόσο, κάποια από τα εμπόδια που αναφέρθηκαν είναι πιθανόν να παραμείνουν και να χρειαστεί περαιτέρω διερεύνηση των εμποδίων στην εκμάθηση του προγραμματισμού και των στρατηγικών που εφαρμόζουν οι μαθητές προκειμένου να τα ξεπεράσουν.

## Αναφορές

- Κατούλας, Κ., Κατούλας, Σ., Κατούλας, Χ. (2013). *Διερεύνηση της μάθησης εννοιών προγραμματισμού 'ο με το Scratch από μαθητές Ε' και Στ' Δημοτικού με βάση μια μεικτή ταξινόμηση Bloom και SOLO*. 3<sup>ο</sup> Πανελλήνιο Συνέδριο «Ένταξη των ΤΠΕ στην Εκπαιδευτικής Διαδικασία». Πειραιάς.
- Κόμης, Β. (2005). *Εισαγωγή στην Διδακτική της Πληροφορικής*. Κλειδάριθμος.
- Κορδάκη, Μ., Ψώμος, Π. (2012). Scratch: 11 διαφορετικές κατηγορίες μαθησιακών δραστηριοτήτων. 6<sup>ο</sup> Πανελλήνιο Συνέδριο 'Διδακτική της Πληροφορικής', Διοργάνωση: ΕΠΥ και Παιδαγωγικό τμήμα Παν/μίου Δυτικής Μακεδονίας, (σελ.591-594), 20-22 Απριλίου 2012, Φλώρινα
- Νικολός, Δ., Κόμης, Β. (2010). *Μια διδακτική πρόταση για τη γλώσσα προγραμματισμού Scratch*. 5<sup>ο</sup> Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής». Αθήνα.
- Νικολός, Δ., Κόμης, Β. (2011). *Η Δομή Επιλογής στη Γλώσσα Προγραμματισμού Scratch: Μια Μελέτη Περίπτωσης με Μαθητές Γυμνασίου*. Πρακτικά 5ου Πανελληνίου Συνεδρίου Εκπαιδευτικού Συνεδρίου Καθηγητών Πληροφορικής. Απρίλιος 2011
- Σαρημαλιδής, Ι. (2012). *Μάθηση προγραμματισμού Η/Υ από μαθητές Α' Λυκείου με το Scratch*. 6<sup>ο</sup> Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής». Φλώρινα.
- Armoni, M., & Ben-Ari, M. (2013). Computer Science Concepts with Scratch. Retrieved from [http://stwwww.weizmann.ac.il/g-cs/scratch/scratch\\_en.html](http://stwwww.weizmann.ac.il/g-cs/scratch/scratch_en.html)
- Barab, S., & Squire, K. (2004). Design-Based Research: Putting a Stake in the Ground. *Journal of the Learning Sciences*, 13(1), 1–14. [http://doi.org/10.1207/s15327809jls1301\\_1](http://doi.org/10.1207/s15327809jls1301_1)
- Berry, M. (2013). *Computing in the national curriculum. A guide for primary teachers*. Bedford: Computing at School.
- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada (pp. 1-25).
- Creswell, J. W. (2002). *Educational research: Planning, conducting, and evaluating quantitative*. New Jersey: Upper Saddle River.
- duBoulay, U. (1986). Some difficulties of learning to program, *Journal of Educational Computing - Research*, 2(1), pp. 57-73.
- Hoc, J. M. (Ed.). (2014). *Psychology of programming*. Academic Press.
- Kafai, Y. B. (2005). The Classroom as "Living Laboratory": Design-Based Research for Understanding, Comparing, and Evaluating Learning Science Through Design. *Educational Technology*, 29–34. Retrieved from <http://www.gse.upenn.edu/~kafai/paper/pdfs/ClassroomLab.pdf>
- Komis, V., Tzavara, A., Karsenti, T., Collin, S., & Simard, S. (2013). Educational scenarios with ICT: An operational design and implementation framework. *Society for Information Technology & Teacher Education International Conference*, 2013(1), 3244–3251. Retrieved from [http://www.editlib.org/p/48594/proceeding\\_48594.pdf](http://www.editlib.org/p/48594/proceeding_48594.pdf)
- Lewis, C. (2013). *Video: Programming in Scratch: Forever-IF vs. IF-Forever*. Retrieved January 11, 2017, from <https://www.schooltube.com/video/fe012819d575489c8cc3>.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. <http://doi.org/10.1016/j.chb.2014.09.012>
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). Scratch: a sneak preview. *Second International Conference on Creating, connecting and collaborating through computing*, (pp. 104-109). IEEE.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, 10(4), 1–15. <http://doi.org/10.1145/1868358.1868363>.
- Mayer, R. E. (1988). Learning LOGO: A cognitive analysis. *Teaching and Learning Computer Programming: Multiple Research Perspectives*, 55.
- McCall, D., Kolling, M. (2014). *Meaningful Categorisation of Novice Programmer Errors*. In *Frontiers In Education Conference*.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011). *Habits of programming in scratch*. In Proceedings of the 16th annual joint conference on Innovation and technology in computer science education (pp. 168-172). ACM.

- Moreno, J., Robles, G. (2014). *Automatic detection of bad programming habits in scratch: A preliminary study*. IEEE Frontiers in Education Conference (FIE) Proceedings. IEEE.
- Nikolos, D., Komis, V. (2015). *Synchronization in Scratch: A Case Study with Education Science Students*. *Journal of Computers in Mathematics and Science Teaching*, 34(2), 223-241. Chesapeake, VA: Association for the Advancement of Computing in Education (AACE).
- Pea, R. D. (1986). Language-independent conceptual "bugs" in novice programming. *Journal of Educational Computing Research*, 2(1), 25-36. <http://doi.org/10.2190/689T-1R2A-X4W4-29J2>
- Putnam, R. T., Sleeman, D., Baxter, J. A., & Kuspa, L. K. (1986). A summary of misconceptions of high school Basic programmers. *Journal of Educational Computing Research*, 2(4), 459-472.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52, 60-67.
- Rusk, N. (2016). *The Scratch Coding Cards: Creative Coding Activities for Kids*. No Starch Press.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33. <http://doi.org/10.1145/1118178.1118215>

**Αλλαγές με βάση τα σχόλια των κριτών**

Αναφορά στα σχήματα 12 και 13 στο σώμα του κειμένου

Διόρθωση των βιβλιογραφικών αναφορών

Διόρθωση γραμματικών και συντακτικών λαθών