

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2009)

1ο Εκπαιδευτικό Συνέδριο «Ένταξη και χρήση των ΤΠΕ στην εκπαιδευτική διαδικασία»



Μάθηση προγραμματισμού ΗΥ από εκκολαπτόμενους εκπαιδευτικούς με το SCRATCH

Γιώργος Φεσάκης , Κυριακή Σεραφείμ

Βιβλιογραφική αναφορά:

Φεσάκης Γ., & Σεραφείμ Κ. (2024). Μάθηση προγραμματισμού ΗΥ από εκκολαπτόμενους εκπαιδευτικούς με το SCRATCH . *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, 1, 403–409. ανακτήθηκε από <https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/6537>

Μάθηση προγραμματισμού ΗΥ από εκκολλημένους εκπαιδευτικούς με το SCRATCH

Φεσάκης Γιώργος¹, Σεραφείμ Κυριακή²

¹Λέκτορας, ΤΕΠΑΕΣ, Παν/μιο Αιγαίου

gfsakis@rhodes.aegean.gr

²Φοιτήτρια του ΠΜΣ: “Παιδικό βιβλίο και παιδαγωγικό υλικό”, Παν/μιο Αιγαίου

pse04178@rhodes.aegean.gr

Περίληψη

Η εκμάθηση του προγραμματισμού ΗΥ παρουσιάζει συνήθως μεγάλες δυσκολίες. Η εργασία αυτή αφορά στην εισαγωγή στον προγραμματισμό εκκολλημένων εκπαιδευτικών. Η συνήθης διδακτική προσέγγιση του προγραμματισμού εξαντλείται στην παρουσίαση συντακτικών δομών της γλώσσας προγραμματισμού και την εφαρμογή αυτών σε μαθηματοποιημένα προβλήματα. Αντίθετα οι φοιτητές της έρευνας χρησιμοποιούν το περιβάλλον Scratch για την ανάπτυξη εκπαιδευτικού υλικού, αφού εξοικειωθούν με αυτό με μια προσέγγιση αναδυόμενου γραμματισμού στον προγραμματισμό. Τα αποτελέσματα δείχνουν ότι σχετικά σύντομα οι φοιτητές μπορούν να παράγουν ενδιαφέρουσες πολυπλοκότητας και ποιότητας εφαρμογές στις οποίες επιδεικνύουν κατανόηση βασικών εννοιών του προγραμματισμού.

Λέξεις κλειδιά: Scratch, αρχάριοι προγραμματιστές, διδακτική προγραμματισμού ΗΥ

1. Εισαγωγή

Ο προγραμματισμός των ΗΥ θεωρείται γενικά μια σημαντική ικανότητα. Το εκπαιδευτικό ενδιαφέρον του προγραμματισμού ΗΥ δεν βασίζεται μόνο στην οικονομική του σημασία αλλά και στη μαθησιακή του αξία. Ο προγραμματισμός αποτελεί δεξιοότητα κλειδί για τη κατανόηση πολλών άλλων περιοχών της Πληροφορικής. Επιπλέον, ορισμένα συστήματα ανάπτυξης προγραμμάτων μπορούν να αποτελέσουν κonstrouκτιβιστικά μαθησιακά περιβάλλοντα γενικού σκοπού (Papert, 1980). Ο προγραμματισμός ΗΥ μπορεί να αξιοποιηθεί στην ανάπτυξη προηγμένων δεξιοτήτων όπως: επίλυση προβλημάτων, δημιουργική σκέψη, λογικό συλλογισμό, συστηματικό πειραματισμό κ.α. Πιο πρόσφατα ο προγραμματισμός ΗΥ αναφέρεται και ως εγγραμμισμός για τη σύγχρονη κοινωνία, ο οποίος επιτρέπει στους πολίτες να είναι ενεργοί παραγωγοί διαδραστικού ψηφιακού περιεχομένου στο web 2.0 (Monroy-Hernandez, 2007; Peppler & Kafai, 2007). Για τους λόγους αυτούς ο προγραμματισμός ΗΥ εμφανίζεται στα αναλυτικά προγράμματα όλων των βαθμίδων και διδάσκεται σε πολλά τμήματα της τριτοβάθμιας εκπαίδευσης, που δεν είναι κατά ανάγκη άμεσα σχετικά με την Πληροφορική ή την Τεχνολογία. Μεταξύ των τμημάτων αυτών συγκαταλέγονται και τα παιδαγωγικά τμήματα, στα οποία εκκολλημένοι εκπαιδευτικοί έχουν μια πρώτη εξοικείωση με τον προγραμματισμό ΗΥ είτε για την άμεση εκπαιδευτική του αξιοποίηση είτε για να μπορούν να παράγουν διαδραστικό, πολυμεσικό, μαθησιακό υλικό.

Παρά την σημασία του ο προγραμματισμός ΗΥ παρουσιάζει δυσκολίες στην εκμάθηση του (Pea, 1986), (Wiedenbeck, 2005). Πολλοί μαθητές αντιμετωπίζουν δυσκολίες στην κατανόηση κεντρικών εννοιών και στη σύνθεση προγραμμάτων με συγκεκριμένες προδιαγραφές. Είναι χαρακτηριστικό ότι τα ποσοστά αποτυχίας ή διαρροής των φοιτητών σε μαθήματα εισαγωγής στον προγραμματισμό υπολογίζονται συχνά σε (15-30)% (Guzdial & Soloway, 2002). Για να αυξηθεί η συμμετοχή και να αμβλυνθούν τα εμπόδια εκμάθησης του προγραμματισμού ΗΥ έχουν προταθεί διάφορες προσεγγίσεις όπως: η ανάπτυξη ειδικών εκπαιδευτικών γλωσσών προγραμματισμού (Φεσάκης & Δημητρακοπούλου 2006) που καθιστούν τον προγραμματισμό ελκυστικότερο, η έρευνα της Διδακτικής του προγραμματισμού (Wiedenbeck, 2005) και η μελέτη περιβαλλόντων άτυπης μάθησης του προγραμματισμού (Maloney et al, 2008). Ακολουθώντας την προσέγγιση των ειδικά σχεδιασμένων εκπαιδευτικών περιβαλλόντων προγραμματισμού, στην εργασία αυτή εστιάζουμε στην μάθηση προγραμματιστικών εννοιών, τις δυσκολίες και την ανάπτυξη προγραμματιστικής ικανότητας φοιτητών ενός παιδαγωγικού τμήματος με το εκπαιδευτικό περιβάλλον Scratch.

Στα επόμενα παρουσιάζεται σύντομη επισκόπηση των δυσκολιών των μαθητών στην εκμάθηση προγραμματισμού ΗΥ, ακολουθεί συνοπτική αναφορά στο Scratch, κατόπιν περιγράφονται οι ερευνητικές συνθήκες, στην συνέχεια παρουσιάζονται τα ερευνητικά δεδομένα και τέλος ακολουθεί σύνοψη και συμπεράσματα.

2. Θεωρητικό πλαίσιο

2.1 Δυσκολίες στην εκμάθηση προγραμματισμού ΗΥ

Η Διδακτική μελέτη του προγραμματισμού έχει από νωρίς προσελκύσει το ενδιαφέρον των ερευνητών στη χώρα μας και διεθνώς. Ο Pea (1986) μετά από μακροχρόνια έρευνα σε παιδιά ηλικιών 8-12 και 14-17 που μάθαιναν να προγραμματίζουν με την γλώσσα LOGO περιγράφει τις παρανοήσεις του «*κρυμμένου νου*» και του «*παραλληλισμού*» στην κατανόηση του τρόπου εκτέλεσης των προγραμμάτων. Οι Spohrer και Soloway (1986) εξετάζουν δύο διαισθητικές αρχές που διατυπώνουν συχνά όσοι διδάσκουν εισαγωγικά μαθήματα προγραμματισμού. Η πρώτη αρχή αναφέρει ότι τα σφάλματα δεν συμβαίνουν ισοπίθανα-κάποια εμφανίζονται συχνότερα από άλλα. Από την έρευνα προέκυψε πράγματι ότι στο 20% των τύπων σφαλμάτων οφειλόταν το 55% του συνόλου των καταγεγραμμένων προγραμματιστικών λαθών. Η δεύτερη διαισθητική αρχή υποστηρίζει ότι τα περισσότερα σφάλματα των νέων προγραμματιστών οφείλονται στην ελλιπή κατανόηση της σημασιολογίας των βασικών δομών των γλωσσών προγραμματισμού. Άλλες έρευνες αφορούν σε δυσκολίες στη κατανόηση βασικών εννοιών όπως η προγραμματιστική μεταβλητή (Τζιμογιάννης & Κόμης, 2000; Φεσάκης & Δημητρακοπούλου 2005) και οι βασικές δομές ελέγχου: δομή επιλογής, επανάληψης και αναδρομής (Εφόπουλος κ.α., 2005).

Από την παραπάνω επισκόπηση βλέπουμε ότι οι περισσότερες έρευνες αφορούν φοιτητές που λόγω ειδικότητας μαθαίνουν προγραμματισμό ΗΥ ή μαθητές που παρακολουθούν αντίστοιχο μάθημα. Επιπλέον οι έρευνες εστιάζουν κυρίως στις βασικές δομές και έννοιες των διαδικαστικών γλωσσών στο πλαίσιο συστηματικής διδασκαλίας με τη βοήθεια προβλημάτων μαθηματικών υπολογισμών. Στο Wiedenbeck (2005) αναφέρεται ότι οι παραδοσιακές προσεγγίσεις της διδασκαλίας του προγραμματισμού αποτυγχάνουν γενικά και περισσότερο στις περιπτώσεις των φοιτητών τμημάτων που δεν είναι αμιγώς Πληροφορικής. Στην εργασία αυτή εστιάζουμε σε φοιτητές παιδαγωγικού τμήματος που ενδιαφέρονται στην ανάπτυξη πολυμεσικού διαδραστικού εκπαιδευτικού υλικού. Η διδακτική προσέγγιση δεν εστιάζει στη συστηματική παρουσίαση των συντακτικών κανόνων της γλώσσας αλλά στην ολιστική και αναδύομενη ανάπτυξη ικανότητας προγραμματισμού μέσα σε πλαίσιο αυθεντικών προβλημάτων. Για να γίνει εφικτή η προσέγγιση επιλέχτηκε το εκπαιδευτικό περιβάλλον προγραμματισμού Scratch για λόγους που αναλύονται στην επόμενη ενότητα.

2.2 Τι είναι το Scratch;

Το Scratch (MIT Media Lab, 2008) είναι ένα εκπαιδευτικό περιβάλλον προγραμματισμού που αναπτύχθηκε από το ομώνυμο έργο στο MIT Media Lab (Maloney et al, 2004). Το Scratch διαθέτει γραφική γλώσσα προγραμματισμού με την οποία καθιστά πιο προσιτό τον προγραμματισμό στα παιδιά (από 8 ετών και άνω), τους εφήβους και άλλους αρχάριους προγραμματιστές. Οι βασικές δομές ελέγχου (ακολουθία, επιλογή και επανάληψη) συντίθενται με το σύρσιμο και το στοιβάγμα γραφικών πλακιδίων που αντιστοιχούν σε εντολές. Ο προγραμματισμός με δομικά στοιχεία εξαλείφει τα συνήθη προβλήματα των συντακτικών λαθών επειδή τα πλακίδια μπορούν να τοποθετηθούν μαζί μόνο με συντακτικά ορθό τρόπο. Επιπλέον των βασικών δομών ελέγχου, η γλώσσα παρέχει πυροδότες γεγονότων (event trigger: when-clicked, when-key-pressed) για καθοδηγούμενο από τα γεγονότα προγραμματισμό και εκπομπή ανοικτής ακρόασης ονοματισμένων μηνυμάτων (named broadcasts) για πολυνηματικό προγραμματισμό. Με το Scratch οι αρχάριοι προγραμματιστές εξοικειώνονται επίσης και με άλλες προγραμματιστικές έννοιες όπως μεταβλητές, λίστες-πίνακες, δίτιμη λογική, σχεδιασμό διεπαφής χρήστη κ.α. Τα έργα που αναπτύσσει κανείς με το Scratch μπορούν να είναι πλούσια σε μέσα και να χρησιμοποιούν γραφικά, κινούμενα σχέδια, μουσική και ήχους. Το Scratch δίνει τη δυνατότητα δημιουργίας ηλεκτρονικών παιχνιδιών, κινουμένων σχεδίων, αλληλεπιδραστικών ιστοριών κ.α. Επιτρέπει στον χρήστη να μοιραστεί τις δημιουργίες του στο διαδίκτυο (<http://scratch.mit.edu>). Η κοινότητα δίνει την ευκαιρία να ανταλλάξει κανείς ιδέες και απόψεις με άλλους δημιουργούς και να εμπλακεί ενεργά σε μια κοινότητα πρακτικής και μάθησης. Η ύπαρξη της κοινότητας σε συνδυασμό με τα παραπάνω χαρακτηριστικά αποτελούν το βασικό λόγο για την επιλογή του Scratch στο πλαίσιο της παρούσας εργασίας μεταξύ άλλων εκπαιδευτικών περιβαλλόντων προγραμματισμού.

3. Έρευνα

Στην παρούσα έρευνα φοιτητές του Τμήματος των Επιστημών της Προσχολικής Αγωγής και του Εκπαιδευτικού Σχεδιασμού (ΤΕΠΑΕΣ) του Πανεπιστημίου Αιγαίου εξοικειώνονται με το Scratch προκειμένου να παράγουν διαδραστικό ψηφιακό εκπαιδευτικό υλικό. Η διδακτική προσέγγιση έχει

χαρακτηριστικά αναδυόμενου γραμματισμού στον προγραμματισμό. Η διδασκαλία δεν εξαντλείται στη συστηματική παράθεση των συντακτικών κανόνων της γλώσσας διανθισμένη με αποσπασματικά παραδείγματα, αντίθετα δίνει έμφαση σε ολοκληρωμένες μικροεφαρμογές. Αρχικά γίνεται παρατήρηση παραδειγμάτων, στη συνέχεια τροποποίηση υπαρχόντων και τέλος παραγωγή νέων από τους φοιτητές. Η προσέγγιση αυτή είναι εμπνευσμένη από την άτυπη μάθηση του προγραμματισμού που παρατηρείται στην κοινότητα του Scratch.

Στόχος της παρούσας έρευνας είναι να μελετηθούν οι δυσκολίες των εκκολλαπτόμενων εκπαιδευτικών στο Scratch καθώς και η εκτίμηση του βαθμού εκμάθησης του προγραμματισμού ΗΥ με την παραπάνω προσέγγιση. Πιο συγκεκριμένα, τα ερευνητικά ερωτήματα είναι τα εξής: **(i)** Ποιες δυσκολίες αναφέρουν οι φοιτητές στην ανάπτυξη εφαρμογών με το Scratch; **(ii)** Ποιες οι προγραμματιστικές έννοιες που κατάφεραν οι φοιτητές να χρησιμοποιήσουν; Τι πολυπλοκότητα έχουν τα προγράμματα που φτιάχνουν οι φοιτητές; **(iii)** Ποια τα είδη των εφαρμογών που επέλεξαν οι φοιτητές να αναπτύξουν;

Οι φοιτητές του δείγματος παρακολουθούσαν υποχρεωτικό μάθημα για την ανάπτυξη πολυμεσικών εκπαιδευτικών εφαρμογών. Ο αριθμός των φοιτητών στις διάφορες ερευνητικές δραστηριότητες είναι μεταβλητός. Στον πίνακα 1 εμφανίζονται ενδεικτικά ηλικιακά στοιχεία για το δείγμα. Η συντριπτική πλειοψηφία του δείγματος είναι γυναίκες.

Πίνακας 1: Ηλικιακή κατανομή των φοιτητών του δείγματος

ΕΤΟΣ ΓΕΝΝΗΣΗΣ	f	%
<1986	6	17.3t
1987	16	45.7
1988	11	31.4
1989	2	5.7
TOTAL	35	100

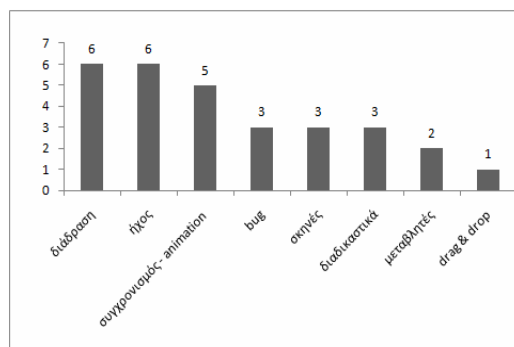
Για τη συλλογή των ερευνητικών δεδομένων επιλέχθηκαν οι παρακάτω πηγές: **(i)** Το περιεχόμενο του forum τεχνικής υποστήριξης που αναπτύχθηκε για τις δυσκολίες των φοιτητών στην ανάπτυξη των εφαρμογών με το Scratch. **(ii)** Οι εφαρμογές που σχεδιάστηκαν από τους φοιτητές, για την διερεύνηση των προγραμματιστικών εννοιών που χρησιμοποιήθηκαν και τα είδη εφαρμογών που έφτιαξαν από προγραμματιστικής και εκπαιδευτικής άποψης. Η έρευνα πραγματοποιήθηκε το εαρινό εξάμηνο του 2008. Αρχικά (2^η-4^η εβδομάδα) παρουσιάστηκε το περιβάλλον και η κοινότητα Scratch στους φοιτητές και αξιοποιήθηκε ειδικά σχεδιασμένο επιμορφωτικό υλικό για εξοικείωση (Φεσάκης κ.α. 2008). Κατόπιν (5^η-8^η εβδομάδα) οι φοιτητές έπρεπε να σχεδιάσουν και να υλοποιήσουν δικές τους εφαρμογές στο Scratch. Στο διάστημα αυτό υπήρχε, επιπλέον των συναντήσεων, διαθέσιμη τεχνική υποστήριξη στους φοιτητές μέσω ειδικού Forum.

3.1 Αποτελέσματα

Ακολουθεί η ανάλυση των ερευνητικών δεδομένων κατά ερευνητικό ερώτημα.

(i) Δυσκολίες των φοιτητών στην ανάπτυξη εφαρμογών με το Scratch με βάση το περιεχόμενο από το forum τεχνικής υποστήριξης.

Οι φοιτητές έγραφαν στο forum τα προβλήματα που αντιμετώπιζαν κατά την ανάπτυξη εφαρμογών με το Scratch, για να λάβουν βοήθεια από τους διδάσκοντες ή/και τους συμφοιτητές τους. Τα προβλήματα που αναφέρθηκαν χωρίζονται σε κατηγορίες, οι συχνότερες κάθε κατηγορίας εμφανίζονται στο διάγραμμα 1.

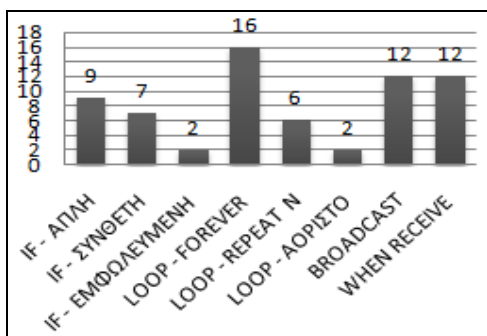


Διάγραμμα 1: Προβλήματα που ανέφεραν οι φοιτητές στο forum

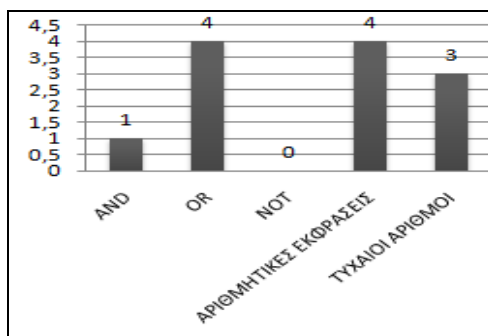
Εμφανίστηκαν συνολικά 6 προβλήματα που αφορούσαν στη διάδραση. Στη κατηγορία αυτή εντάσσονται περιπτώσεις στις οποίες ήθελαν να πετύχουν κάτι που περιέγραφαν με όρους διάδρασης μεταξύ προγραμματιστικών αντικειμένων και διεπαφής χρήστη. Π.χ. «πως θα γίνει όταν βρίσκει ο χρήστης κάρτες (κάρτες μνήμης) που έχουν το ίδιο αντικείμενο, να μένουν ανοιχτές». Άλλες 6 περιπτώσεις αφορούσαν προβλήματα σχετικά με τον ήχο Π.χ. «Πως γίνεται να ακούγεται ο ήχος κατά τη διάρκεια που εμφανίζονται τα λόγια γραπτά στην σκηνή;». Επίσης, υπήρξαν 5 προβλήματα με τον συγχρονισμό των αντικειμένων και την κίνηση (animation). Επιπλέον, 3 προβλήματα με τις σκηνές, 3 σχετικά με εξσφαλμάτωση (bug) και 3 αφορούσαν διαδικαστικά. Τέλος, αναφέρθηκαν 2 περιπτώσεις με προβλήματα στις μεταβλητές και μία με πρόβλημα στο σύρσιμο αντικειμένων με το ποντίκι (drag & drop). Το είδος των προβλημάτων που αναφέρθηκαν στο forum είναι πολύ διαφορετικό από τα προβλήματα που υπάρχουν συνήθως σε εκπαιδευτικά εγχειρίδια των γλωσσών προγραμματισμού. Αφορούν περιγραφές πιο κοντά στο τελικό αποτέλεσμα-εφαρμογή παρά στις γενικές δομές της γλώσσας. Τα προβλήματα αυτά δίνουν ένα ρεαλιστικό πλαίσιο για την εισαγωγή εννοιών και τεχνικών του προγραμματισμού, ενώ η χρήση του forum συμβάλει στην οικονομία της επικοινωνίας και την μέγιστη διάδοση των διδακτικών καταστάσεων.

(ii) Ποιες προγραμματιστικές έννοιες χρησιμοποίησαν οι φοιτητές στις εφαρμογές τους; Τι πολυπλοκότητας εφαρμογές έφτιαξαν;

Για τη μελέτη των προγραμματιστικών εννοιών που κατάφεραν οι φοιτητές να χρησιμοποιήσουν μελετήθηκαν οι 28 εφαρμογές που δημιούργησαν. Καταμετρήθηκαν οι εφαρμογές που εμφάνιζαν κάθε προγραμματιστική έννοια ξεχωριστά. Από τον πίνακα καταμέτρησης προκύπτουν τα επόμενα διαγράμματα, 2 και 3, που συνοψίζουν τα αποτελέσματα. Από τα διαγράμματα βλέπουμε ότι οι φοιτητές χρησιμοποίησαν απλή και σύνθετη επιλογή, δομές επανάληψης καθώς και τη δομή της αποστολής ονοματισμένων μηνυμάτων ευρείας εκπομπής (BROADCAST- WHEN RECEIVE), που χρησιμοποιείται για τον συγχρονισμό νημάτων. Η έννοια αυτή είναι προχωρημένη και δεν συναντάται συνήθως στα εισαγωγικά μαθήματα διαδικαστικού ή/και αντικειμενοστρεφούς προγραμματισμού.



Διάγραμμα 2: Δομές ελέγχου

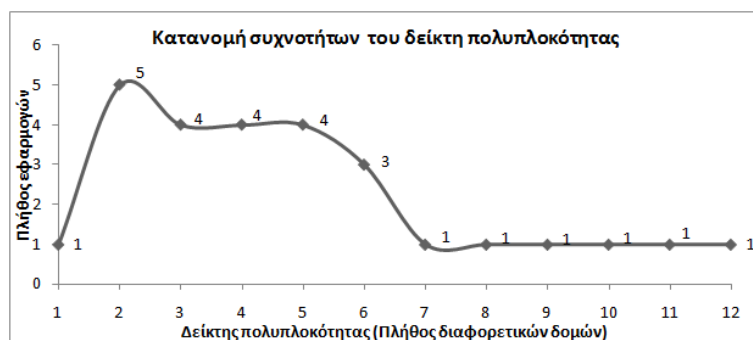


Διάγραμμα 3: Παραστάσεις

Η κατάκτηση της από τους φοιτητές είναι ιδιαίτερα σημαντική, επειδή τους επιτρέπει να αναπτύσσουν πολυνηματικές εφαρμογές δημιουργώντας έτσι πιο ενδιαφέροντα προγράμματα και προσλαμβάνουσες παραστάσεις για την τυπική εισαγωγή των εννοιών. Από το διάγραμμα 3 βλέπουμε ότι οι λογικοί τελεστές AND, OR και NOT χρησιμοποιούνται σποραδικά. Η εξοικείωση με τις λογικές εκφράσεις δεν μπορεί να εξαντληθεί από την αναδυόμενη προσέγγιση.

Η πολυπλοκότητα των εφαρμογών των φοιτητών

Για την εκτίμηση και τη σύγκριση της πολυπλοκότητας των εφαρμογών των φοιτητών ορίστηκε ο δείκτης πολυπλοκότητας. Ο δείκτης πολυπλοκότητας για μια συγκεκριμένη εφαρμογή εκφράζει το πλήθος των διαφορετικών βασικών προγραμματιστικών δομών που περιλαμβάνει. Για παράδειγμα, μια εφαρμογή με IF και LOOP θα έχει δείκτη 2 (ανεξάρτητα από το συνολικό πλήθος τους).



Διάγραμμα 4: Δείκτης Πολυπλοκότητας

Στο διάγραμμα 4 εμφανίζεται το πλήθος των εφαρμογών των φοιτητών για κάθε τιμή του δείκτη πολυπλοκότητας (οριζόντιος άξονας). Συνοπτικά μπορούμε να πούμε ότι οι φοιτητές κατάφεραν στην πλειοψηφία τους να αναπτύξουν σημαντικής πολυπλοκότητας εφαρμογές.

Ένας άλλος τρόπος, για να εκτιμηθεί η πολυπλοκότητα των εφαρμογών των φοιτητών, είναι η καταγραφή των δομικών στοιχείων του Scratch, που χρησιμοποιήθηκαν για την κατασκευή τους. Τα δομικά στοιχεία του Scratch δεν είναι πάντα γενικεύσιμα (π.χ. σκηνές). Όμως, σε γενικές γραμμές τα αντικείμενα έχουν το συνηθισμένο νόημα στον προγραμματισμό, ενώ τα scripts αντιστοιχούν σε υποπρογράμματα. Με όρους δομικών στοιχείων οι εφαρμογές των φοιτητών αναλύονται στον πίνακα 2.

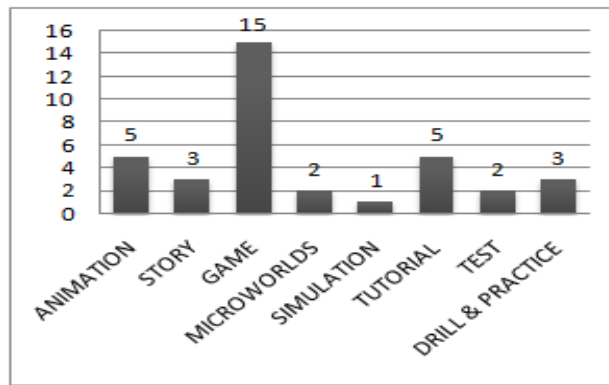
Πίνακας 2: Πολυπλοκότητα με όρους Scratch

	Ελάχιστη	Μέγιστη	Μέση
Σκηνές	1	9	1.81
Αντικείμενα	1	28	12.67
Scripts	2	97	29.59
Μεταβλητές	0	30	2.78

Από τον πίνακα φαίνεται ότι υπάρχει ποικιλία στην πολυπλοκότητα των εφαρμογών, η μέση εφαρμογή κρίνεται ενδιαφέρουσας πολυπλοκότητας. Είναι ενδιαφέρον να σημειώσει κανείς ότι υπάρχουν ολοκληρωμένες εφαρμογές χωρίς μεταβλητές.

(iii) Ποια τα είδη των εφαρμογών που επέλεξαν οι φοιτητές να αναπτύξουν;

Οι εφαρμογές των φοιτητών κατηγοριοποιήθηκαν σε είδη. Τα είδη εφαρμογών είναι: Animation (κινούμενα σχέδια), Story (αφήγηση), Game (Παιχνίδια), Microworlds (Μικρόκοσμοι), Simulation (Προσομοίωση), Tutorial (Εκπαιδευτική), Test (Αξιολόγηση), Drill & Practice (εφαρμογής και εξάσκησης) και Other (Λοιπά). Η κατανομή των εφαρμογών στις διάφορες κατηγορίες φαίνεται στο διάγραμμα 5. Οι περισσότερες εφαρμογές αφορούν σε παιχνίδια (15), κινούμενα σχέδια (5) και εκπαιδευτικές εφαρμογές (5). Τα παιχνίδια είναι συνήθως απαιτητικά στην κατασκευή τους. Τα στοιχεία δείχνουν ότι οι φοιτητές ασχολήθηκαν με ποικιλία ειδών και ότι η εύλογη επιλογή της κατασκευής παιχνιδιών μπορεί να αποτελέσει ένα ακόμα παράγοντα για την εκμάθηση του προγραμματιστικών εννοιών λόγω των απαιτήσεων τους.



Διάγραμμα 5: Κατηγορίες εφαρμογών

4. Συζήτηση- Συμπεράσματα

Η εκμάθηση του προγραμματισμού είναι σημαντική ικανότητα αν και παρουσιάζει διάφορες δυσκολίες. Για την αντιμετώπιση τους έχουν προταθεί διάφορες παρεμβάσεις μεταξύ των οποίων και η χρήση ειδικών εκπαιδευτικών περιβαλλόντων προγραμματισμού. Στην εργασία αυτή παρουσιάζεται η χρήση του εκπαιδευτικού περιβάλλοντος Scratch σε συνδυασμό με ειδική διδακτική προσέγγιση για την εξοικείωση εκκολαπτόμενων εκπαιδευτικών με τον προγραμματισμό. Για την αποτίμηση της προσέγγισης διερευνήθηκαν οι δυσκολίες των φοιτητών, η πολυπλοκότητα και η ποιότητα των εφαρμογών που έφτιαξαν.

Οι φοιτητές περιγράφουν τα προβλήματα τους με όρους τελικού αποτελέσματος και διεπαφής χρήστη. Το γεγονός αυτό παρέχει ένα ρεαλιστικό πλαίσιο για την καλύτερη εισαγωγή των προγραμματιστικών εννοιών και τεχνικών ως εργαλεία για την επίλυση προβλημάτων. Από πλευράς προγραμματιστικών εννοιών οι φοιτητές με την συγκεκριμένη διδακτική προσέγγιση κατάφεραν σε σύντομο χρονικό διάστημα να χρησιμοποιήσουν τις βασικές δομές και αρκετές πιο σύνθετες (συγχρονισμό νημάτων, προγραμματισμό καθοδηγούμενο από τα γεγονότα κλπ) με επιτυχία. Εκτιμάται έτσι ότι είναι σε καλύτερη κατάσταση να προσεγγίσουν πολυπλοκότερες έννοιες με τυπικό τρόπο, όπως για παράδειγμα τη χρήση σύνθετων λογικών εκφράσεων, τις εμφωλευμένες δομές κ.α.

Οι περισσότερες από τις εφαρμογές που δημιουργήθηκαν από τους φοιτητές είναι παιχνίδια. Επίσης, οι δομές ελέγχου που χρησιμοποιήθηκαν κυρίως είναι το *forever*, *broadcast- when receive* και *if (απλή)*. Όσον αφορά στη διεπαφή με τον χρήστη στο μεγαλύτερο μέρος των εφαρμογών γίνεται με το ποντίκι (με κλικ και λιγότερο με σύρσιμο). Η δημιουργία εφαρμογών, όπως είναι οι μικρόκοσμοι και τα ηλεκτρονικά παιχνίδια, δείχνει ότι οι φοιτητές έχουν αρκετές δυνατότητες ανάπτυξης προγραμματιστικών δεξιοτήτων με το Scratch. Κατάφεραν να εξοικειωθούν και να χρησιμοποιήσουν προχωρημένες προγραμματιστικές έννοιες όπως: δομές ελέγχου, μεταβλητές, αντικείμενα, επικοινωνία μεταξύ αντικειμένων, ταυτόχρονος προγραμματισμός κ.α. Οι εφαρμογές των φοιτητών είναι ενδιαφέρουσες προγραμματιστικά και εκπαιδευτικά. Συνεπώς, η χρήση του scratch μπορεί να υποστηρίξει αρχάριους προγραμματιστές και κατά προέκταση τους εκπαιδευτικούς για την ανάπτυξη μαθησιακού υλικού.

Όσο αφορά στην πολυπλοκότητα των προγραμμάτων των φοιτητών τα περισσότερα είναι μέτρια, αν και υπήρξαν ορισμένα αρκετά πολύπλοκα, τόσο με όρους δομικών στοιχείων Scratch όσο και του δείκτη πολυπλοκότητας. Ο δείκτης πολυπλοκότητας που προτείνουμε στην παρούσα εργασία είναι σημαντικός, επειδή μπορεί να αξιοποιηθεί και για τη σύγκριση των προγραμμάτων σε διαφορετικά περιβάλλοντα.

Γενικά είναι πεποίθηση των συγγραφέων ότι με την χρήση του Scratch οι φοιτητές των παιδαγωγικών τμημάτων μπορούν να παράγουν ενδιαφέρουσες πολυπλοκότητας και ποιότητας ψηφιακό εκπαιδευτικό υλικό, ενώ ταυτόχρονα να εξοικειωθούν με βασικές έννοιες του προγραμματισμού ΗΥ γενικότερα. Οι φοιτητές καταφέρνοντας να φτιάξουν ολοκληρωμένες εφαρμογές αναπτύσσουν την αυτοεκτίμησή τους στις ΤΠΕ γενικά και στον προγραμματισμό ΗΥ ειδικότερα. Με τον τρόπο αυτό αυξάνονται οι πιθανότητες να προοδεύσουν στον προγραμματισμό ΗΥ (Wiedenbeck, 2005) και να αξιοποιήσουν αποδοτικά τις ΤΠΕ μελλοντικά στην εκπαίδευση.

Η συγκεκριμένη εργασία μπορεί να συνεχιστεί στο μέλλον με πιο συστηματική μελέτη της προτεινόμενης διδακτικής προσέγγισης και τη σύγκριση διαφορών πληθυσμών.

Βιβλιογραφία

- Guzdial, M., & Soloway, E. (2002). Log on education: teaching the Nintendo generation to program. *Communications of the ACM*, 45 (4), pp. 17-21.
- Maloney, J. H., Pepler, K., Kafai, Y. B., Resnick, M., & Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. *ACM SIGCSE Bulletin*, 40 (1), pp. 367-371.
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). Scratch: A Sneak Preview. *Proceedings of the Second Int. Conference on Creating, Connecting, and Collaborating through Computing*, Jan 29-30, 2004, Kyoto, pp. 104-109
- MIT Media Lab. (2008). Scratch. Retrieved OCT 27, 2008, from <http://scratch.mit.edu>
- Monroy-Hernandez, A. (2007). ScratchR: a platform for sharing user-generated programmable media. In M. B. Skov (Ed.), *6th International Conference on Interaction Design and Children*. Aalborg, Denmark.
- Monroy-Hernández, A., & Resnick, M. (2008, March-April). Empowering kids to create and share programmable media. *Interactions*, pp. 50-53.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. NY: Basic Books.
- Pea R. (1986), Language-independent conceptual ‘bugs’ in novice programming, *Journal of Educational Computing Research*, 2(1), 25-36.
- Pepler, K. A., & Kafai, Y. B. (2007). Collaboration, Computation, and Creativity: Media Arts Practices in Urban Youth Cultures. In the proceeding of the Computer Supported Collaborative Learning conference, July 16 - 21, Rutgers, USA pp. 586-588
- Spohrer J., & Soloway E. (1986), “Novice Mistakes: Are the Folk Wisdoms Correct?”, *Communications of the ACM*, Volume 29, Issue 7, 624-632
- Wiedenbeck, S. (2005). Factors affecting the success of non-majors in learning to program. In the proceedings of the first international workshop on Computing education research, Seattle, WA, USA October 01 - 02, (pp. 13 - 24
- Εφόπουλος, Β., Ευαγγελίδης, Γ., Δαγδιλέλης, Β., και Κλεφτοδήμος Α., (2005), *Οι δυσκολίες των αρχάριων Προγραμματιστών*, στο Α. Τζιμογιάννης (επιμ), *Πρακτικά του 3^{ου} Πανελληνίου συνεδρίου «Διδακτική της Πληροφορικής»*, Κόρινθος 7-9 Οκτωβρίου 2005
- Τζιμογιάννης Αθ. & Κόμης Β. (2000), “*Η έννοια της μεταβλητής στον Προγραμματισμό: δυσκολίες και παρανοήσεις μαθητών του Ενιαίου Λυκείου*”, στο Α. Δημητρακοπούλου (επιμ.), *Πρακτικά 2ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή “Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση”*, 103-114, Πάτρα
- Φεσάκης Γ., Δημητρακοπούλου Α., (2006), «*Επισκόπηση του χώρου των εκπαιδευτικών περιβαλλόντων προγραμματισμού ΗΥ: Τεχνολογικές και Παιδαγωγικές προβολές*», στο *ΘΕΜΑΤΑ στην Εκπαίδευση*, 7(3), pp. 279-304
- Φεσάκης Γ., Δημητρακοπούλου Α., Σεραφείμ, Κ., Ζαφειροπούλου, Α., Ντούνη, Μ. & Τούκα, Β. (2008). *Γνωριμία με το εκπαιδευτικό περιβάλλον προγραμματισμού Scratch*. Πρακτικά 4ου συνεδρίου «Διδακτικής της Πληροφορικής», 26-28 Μαρτίου, Πάτρα
- Φεσάκης Γ., Δημητρακοπούλου Δ., (2005), «Γνωστικές δυσκολίες μαθητών δευτεροβάθμιας εκπαίδευσης σχετικά με την έννοια της προγραμματιστικής μεταβλητής και προτεινόμενες παρεμβάσεις.», 3ο πανελλήνιο συνέδριο “Διδακτική της Πληροφορικής”, Κόρινθος, 7-9 Οκτωβρίου 2005 (<http://www.etpe.gr>)