

# Προγραμματισμός ρομπότ με απτική διεπαφή: ανάλυση επιδόσεων παιδιών σε σύγκριση με τη γραφική διεπαφή

Θεοδόσιος Σαπουνίδης, Σταύρος Δημητριάδης, Ιωάννης Σταμέλος  
teo@edlit.auth.gr, sdemetri@csd.auth.gr, stamelos@csd.auth.gr  
Τμήμα Πληροφορικής, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

## Περίληψη

Αυτό το άρθρο διερευνά τις επιδόσεις 109 παιδιών από πέντε ηλικιακές ομάδες τα οποία προγραμματίσαν ένα Lego Mindstorms NXT ρομπότ, χρησιμοποιώντας απτική και γραφική διεπαφή. Οι μετρήσεις ανέλυσαν την επίδοση των παιδιών σε προκαθορισμένες ασκήσεις και είχαν σαν σκοπό να φωτίσουν τα πλεονεκτήματα – μειονεκτήματα των απτικών διεπαφών σε σχέση με αντίστοιχες γραφικές. Με βάση τις αποστολές που δόθηκαν τα αποτελέσματα έδειξαν ότι τα μικρότερα παιδιά χρειάστηκαν λιγότερο χρόνο για να ολοκληρώσουν τις προγραμματιστικές αποστολές με το απτικό υποσύστημα. Αντιθέτως, τα μεγαλύτερα παιδιά χρειάστηκαν σχεδόν τον ίδιο χρόνο για να ολοκληρώσουν τις αποστολές τους και με τις δύο διεπαφές. Επιπλέον τα παιδιά συνολικά υπέπεσαν σε λιγότερα προγραμματιστικά λάθη και καταγράφηκε καλύτερη αποσφαλμάτωση στην απτική συνθήκη.

**Λέξεις κλειδιά:** απτικές διεπαφές χρήστη, απτικός προγραμματισμός, εκπαιδευτική ρομποτική, εκπαίδευση

## Εισαγωγή

Οι απτικές διεπαφές χρήστη (tangible user interfaces TUIs) είναι διεπαφές με τις οποίες οι χρήστες μπορούν να έχουν διαδραστικές εμπειρίες με ψηφιακές πληροφορίες δια μέσου φυσικών αντικειμένων. Αυτές οι διεπαφές έχουν αναπτυχθεί και χρησιμοποιηθεί σε διάφορες εφαρμογές και για διάφορες ηλικιακές ομάδες (Orit & Eva, 2009). Ειδικά για παιδιά, οι απτικές διεπαφές έχουν επανειλημμένα προταθεί σαν εργαλεία τα οποία υποστηρίζουν μαθησιακές διαδικασίες μέσα από παιχνίδι σε συνεργατικά κυρίως περιβάλλοντα (Price et al., 2003). Αρκετές έρευνες έχουν διερευνήσει διάφορα χαρακτηριστικά της αλληλεπίδρασης με απτικές διεπαφές με σκοπό να παρέχουν πληροφορίες για την αποδοτικότητα τέτοιων συστημάτων όταν χρησιμοποιούνται από μικρά παιδιά (Falcão & Price, 2009; Price et al., 2003). Παρά το αυξανόμενο θεωρητικό πλαίσιο το οποίο υποστηρίζει τη χρήση απτικών διεπαφών στην εκπαίδευση παιδιών, υπάρχει περιορισμένη έρευνα η οποία συστηματικά διερευνά τα προτερήματα των απτικών διεπαφών συγκρινόμενα με τυπικά γραφικά περιβάλλοντα (Xie et al., 2008).

Η παρούσα έρευνα ενέπλεξε εκατόν εννέα παιδιά από πέντε ηλικιακές ομάδες τα οποία προγραμματίσαν ένα Lego Mindstorms NXT ρομπότ, με βάση προγραμματιστικές αποστολές (ασκήσεις) που δόθηκαν, χρησιμοποιώντας ταυτόχρονα απτική και γραφική διεπαφή. Οι μετρήσεις που έγιναν ανέλυσαν την επίδοση των παιδιών.

Τα αποτελέσματα έδειξαν ότι τα μικρότερα παιδιά χρειάστηκαν λιγότερο χρόνο για να ολοκληρώσουν τις προγραμματιστικές αποστολές με το απτικό υποσύστημα. Αντιθέτως, τα μεγαλύτερα παιδιά χρειάστηκαν σχεδόν τον ίδιο χρόνο για να ολοκληρώσουν τις αποστολές τους και με τις δύο διεπαφές. Επιπλέον, λιγότερα προγραμματιστικά λάθη και καλύτερη αποσφαλμάτωση καταγράφηκε στην απτική συνθήκη.

## Διεπαφές στον προγραμματισμό

Η δημιουργία προγραμματιστικών εργαλείων για παιδιά είναι ένα ευρύ πεδίο έρευνας από τις αρχές της δεκαετίας του '60. Βασισμένο κυρίως σε γραφικές διεπαφές και αξιοποιώντας τις θεωρίες του εποικοδομητισμού του Papert (1980), ένας μεγάλος αριθμός από γλώσσες προγραμματισμού δημιουργήθηκαν για παιδιά και αρχάριους χρήστες (Kelleher & Pausch, 2005). Πιο πρόσφατα, κάποιες ενδεικτικές γλώσσες προγραμματισμού όπως οι ToonTalk, Alice, Scratch και ROBOLAB ενσωμάτωσαν απλή σύνταξη, φωλιασμένες επαναλήψεις και δομές ελέγχου με γραφική αναπαράσταση. Αυτές οι προσεγγίσεις οι οποίες βασίστηκαν σε γραφική διεπαφή, επέτρεψαν σε παιδιά να προγραμματίζουν μετακινώντας και συνδέοντας εικονίδια στην οθόνη υπολογιστών. Η διαδικασία του προγραμματισμού μέσα από γραφική διεπαφή απαιτεί την ικανότητα να μεταφραστούν οι συμβολικές αναπαραστάσεις πάνω στην οθόνη στις δράσεις - ενέργειες που αυτές προκαλούν. Από την άλλη ο προγραμματισμός με απτική διεπαφή μπορεί να μειώσει αυτό το γνωστικό κενό μεταφέροντας τις προγραμματιστικές ενέργειες διαμέσου του χειρισμού απτικών αντικειμένων (Nusen & Sirotkiat, 2011).

Στον τομέα του απτικού προγραμματισμού πολλά απτικά συστήματα σχεδιάστηκαν για παιδιά και είχαν σκοπό να συνδέσουν τις προγραμματιστικές δραστηριότητες με το φυσικό κόσμο. Τα περισσότερα από αυτά τα συστήματα επικεντρώθηκαν στον εισαγωγικό προγραμματισμό δηλαδή στην εκμάθηση εισαγωγικών και απλών εννοιών που στόχευαν σε αρχάριους χρήστες και παιδιά. Τα απτικά συστήματα αυτού του τύπου μπορούν να ταξινομηθούν σε δυο κατηγορίες: (α) Τα ενεργά συστήματα (Suzuki & Kato, 1993) και (β) τα παθητικά συστήματα (Horn, 2009). Τα ενεργά συστήματα είναι συστήματα κατά κύριο λόγο με ενσωματωμένα ηλεκτρονικά κυκλώματα, ενώ τα παθητικά συστήματα είναι βασισμένα κυρίως σε αναγνώριση εικόνας ή RFIDs (Radio Frequency Identification). Μια ιδιαίτερα αξιοσημείωτη εργασία στο χώρο παρουσιάστηκε από τους Suzuki και Kato (1993) οι οποίοι κατασκεύασαν το σύστημα AlgoBlocks. Το σύστημα αυτό ήταν η πρώτη ενεργή απτική γλώσσα προγραμματισμού η οποία επιχειρούσε να προωθήσει τη συνεργασία κατά τη διάρκεια του προγραμματισμού μέσα από κοινωνική αλληλεπίδραση και συζήτηση.

Ένα από τα πλεονεκτήματα των απτικών διεπαφών εμφανίζεται να είναι η μείωση της δυσκολίας του να μάθει κάποιος πως λειτουργεί ένα προγραμματιστικό περιβάλλον. Δεδομένου ότι οι χρήστες δεν απαιτείται να μάθουν να χρησιμοποιούν πληκτρολόγιο και ποντίκι, μια και το μόνο που χρειάζεται είναι η φυσική ικανότητα του να χειρίζονται πραγματικά αντικείμενα της καθημερινότητας, όπως κύβους και πάζλ. Κατά συνέπεια, εκτιμάται ότι τα απτικά συστήματα μειώνουν το γνωστικό φορτίο που απαιτείται προκειμένου να μάθει κάποιος πως δουλεύει ένα σύστημα και έτσι η προσοχή του χρήστη εστιάζεται στο πεδίο εκμάθησης του προγραμματισμού (Marshall, 2007).

## Το σύστημα PROTEAS

Το σύστημα PROTEAS αναπτύχθηκε για τη συγκριτική αξιολόγηση της επίδρασης που μπορεί να έχει στον προγραμματισμό η χρήση απτικής διεπαφής σε σχέση με την ισομορφική γραφική διεπαφή. Αποτελείται από ένα απτικό υποσύστημα T\_ProRob (Sapounidis & Demetriadis, 2011) και ένα ισομορφικό γραφικό υποσύστημα (V\_ProRob). Για το σκοπό της παρούσης έρευνας χρησιμοποιήθηκαν το T\_ProRob και το V\_ProRob τα οποία περιγράφονται εν συντομία παρακάτω.

### T\_ ProRob

Το υποσύστημα T\_ ProRob αποτελείται από 28 κύβους εντολών και 16 μικρότερους κύβους παραμέτρων. Οι χρήστες αυτού του συστήματος μπορούν να διατάξουν τους κύβους – εντολές και να προγραμματίσουν ένα Lego Mindstorms ρομπότ να εκτελέσει την αλληλουχία των εντολών που έχει σχηματιστεί από τους κύβους. Ένα ενδεικτικό πρόγραμμα φαίνεται στο Σχήμα 1.

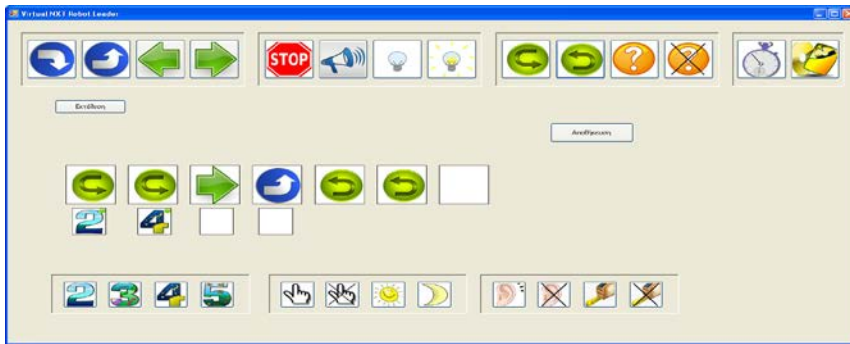


Σχήμα 1. Οι ενεργοί κύβοι στο σύστημα απτικής διεπαφής T\_ ProRob

Το ρομπότ θα εκτελέσει 3 φορές την αλληλουχία: 2 βήματα εμπρός, άναψε φως, στάση, σβήσε φως. Μετά το πέρας της ρουτίνας επανάληψης το ρομπότ θα κάνει έλεγχο με τον αισθητήρα υπερήχων και αν δεν υπάρχει εμπόδιο θα κάνει άλλο ένα βήμα προς τα εμπρός. Οι ενέργειες προγραμματισμού του ρομπότ σχετίζονται με τον έλεγχο αισθητήρων και κινητήρων, όπως: «άναψε/σβήσε φως», «κάνε ήχο», «βήμα εμπρός/πίσω», «στρίψε δεξιά/αριστερά». Το σύστημα επιπλέον διαθέτει πολύπλοκους συνδυασμούς όπως φωλιασμένες επαναλήψεις και ελέγχους. Ένας ειδικός κύβος, στον οποίο ο χρήστης μπορεί να υποθηκεύσει τον κώδικα του, συμπληρώνει το σύνολο των εντολών. Το σύνολο των παραμέτρων του T\_ ProRob είναι μικρότεροι κύβοι οι οποίοι προσαρμόζονται στις εντολές αλλάζοντας τους την λειτουργία. Για παράδειγμα η εντολή κίνησης ενός βήματος εμπρός, με την προσθήκη της παραμέτρου δυο μετατρέπεται σε κίνηση δυο βημάτων προς την ίδια κατεύθυνση. Για την κατασκευή του συστήματος έχουν χρησιμοποιηθεί δυο μικροελεγκτές με υψηλές επιδόσεις, ο 18F4550 και ο 18F2620 της Microchip. Ο χρήστης συνδέει πάνω σε ένα κουτί-βάση τις εντολές, προκειμένου να δημιουργήσει το πρόγραμμα. Στη συνέχεια με το πάτημα του κουμπιού εκτέλεσης, που βρίσκεται στο κουτί-βάση, αρχίζει η επικοινωνία με τα κουτιά - εντολές με σκοπό να αποσταλούν οι εντολές ελέγχου και ανάγνωσης του προγράμματος. Η επόμενη εργασία που αναλαμβάνει το κουτί-βάση είναι να επικοινωνήσει μέσω ασύρματης σύνδεσης Bluetooth (ή θύρας RS232) με έναν υπολογιστή ο οποίος καταγράφει σε μια βάση δεδομένων μεταξύ άλλων και στατιστικά στοιχεία σχετικά με το πρόγραμμα που έχει δημιουργήσει ο χρήστης. Μετά την καταγραφή ο υπολογιστής στέλνει μέσω Bluetooth το πρόγραμμα στο ρομπότ προκειμένου να εκτελεστεί.

### V\_ ProRob

Ο σχεδιασμός του V\_ ProRob βασίστηκε στο αντίστοιχο T\_ ProRob προκειμένου να αποτελέσει ένα αξιόπιστο γραφικό ομοίωμα του. Το υποσύστημα αυτό έχει τις ίδιες εντολές – παραμέτρους με αυτές που προσφέρει το T\_ ProRob.



Σχήμα 2. Το υποσύστημα γραφικής διεπαφής V\_ProRob

Οι χρήστες προκειμένου να δημιουργήσουν ένα πρόγραμμα σύρουν τις διαθέσιμες εντολές και παραμέτρους και τις τοποθετούν στη σειρά προκειμένου να φτιάξουν την αλληλουχία των εντολών. Η ύπαρξη αμφίδρομης επικοινωνίας (με Bluetooth) του υποσυστήματος με το ρομπότ επιτρέπει στο υποσύστημα να αλληλεπιδρά με τους χρήστες εμφανίζοντας βοηθητικές ενδείξεις πάνω στα εικονίδια των εντολών και των παραμέτρων (όπως συμβαίνει και με το απτικό σύστημα). Τέλος το σύστημα συνδέεται με μια βάση δεδομένων στην οποία καταγράφονται στοιχεία του προγράμματος που έχει φτιάξει ο χρήστης. Ένα ενδεικτικό πρόγραμμα το οποίο προγραμματίζει το ρομπότ να σχηματίσει κινούμενο δύο τετράγωνα φαίνεται στο Σχήμα 2.

## Ερευνητικό πλαίσιο

### Συμμετέχοντες

Η έρευνα διεξήχθη σε ένα δημόσιο σχολείο στην περιοχή της Θεσσαλονίκης. Εκατόν εννέα παιδιά από πέντε ηλικιακές ομάδες, 6-7 (N1=20), 7-8 (N2=25), 9-10 (N3=14), 10-11 (N4=25) και 11-12 (N5=25), συμμετείχαν στην έρευνα. Όλα τα παιδιά συμμετείχαν εθελοντικά στις δραστηριότητες και επιλέχθηκαν τυχαία να συμμετέχουν σε δυάδες. Όλα τα παιδιά που συμμετείχαν δεν είχαν καμία προηγούμενη επαφή με τα συστήματα και μιλούσαν τα ελληνικά σαν μητρική τους γλώσσα. Όλα τα παιδιά είχαν εξοικείωση με το ποντίκι.

### Διαδικασία

Κατά την είσοδο των παιδιών στις κατάλληλα διαμορφωμένες αίθουσες συμπλήρωσαν το ερωτηματολόγιο με στοιχεία όπως ηλικία, φύλο, επίπεδο γνώσης υπολογιστών και επίπεδο γνώσης προγραμματισμού. Εν συνέχεια, τους παρουσιάσαμε το NXT ρομπότ και ακολουθώντας ένα απλό προγραμματιστικό σενάριο, ο ερευνητής τους παρουσίασε πως μπορούν να προγραμματίσουν το ρομπότ και με τα δύο συστήματα. Για να αποφύγουμε οποιαδήποτε πιθανότητα επηρεασμού των μετρήσεων από την σειρά παρουσίασης των συστημάτων, ακολουθήσαμε ισορροπημένη διαδικασία παρουσίασης.

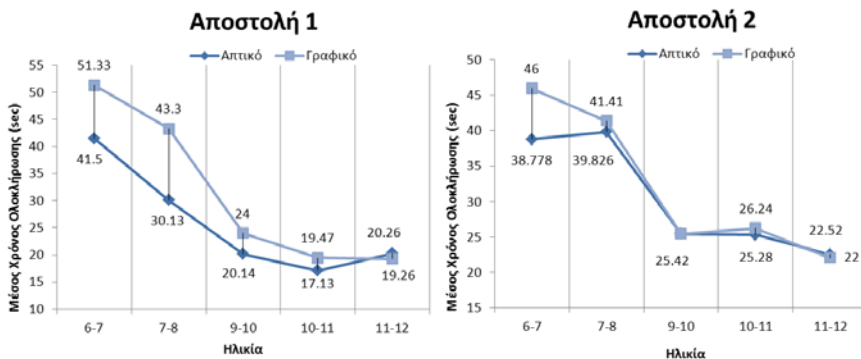
Εν συνέχεια ζητήθηκε από τα παιδιά να προγραμματίσουν το ρομπότ να εκτελέσει αποστολές (αποστολή 1 και αποστολή 2), χρησιμοποιώντας ένα σύστημα (τα μισά ζευγάρια ξεκίνησαν προγραμματίζοντας με το γραφικό σύστημα και τα άλλα μισά ξεκίνησαν προγραμματίζοντας με το απτικό σύστημα). Εν συνέχεια, τα παιδιά έπρεπε να προγραμματίσουν το ρομπότ με βάση ίδιας δυσκολίας αποστολές χρησιμοποιώντας όμως το άλλο υποσύστημα (αποστολή 1' και αποστολή 2').

### Μετρήσεις

Η συλλογή δεδομένων περιελάμβανε καταγραφές υπολογιστή (logs), τις παρατηρήσεις και καταγραφές των τριών ερευνητών. Χρησιμοποιώντας τα παραπάνω δεδομένα μετρήσαμε τρεις μεταβλητές οι οποίες αφορούν την προγραμματιστική επίδοση των μαθητών με τα δύο συστήματα. Ειδικότερα, μετρήθηκαν: (1) ο χρόνος επίτευξης των αποστολών (TAT, ο χρόνος που απαιτήθηκε προκειμένου τα παιδιά να ολοκληρώσουν τις αποστολές), (2) τα λάθη (ER, ο αριθμός των λανθασμένων προγραμμάτων που εκτελέστηκαν) και (3) τα στάδια της αποσφαλμάτωσης (DS, τα στάδια αποσφαλμάτωσης που έφτασαν τα παιδιά μετά από μια λάθος εκτέλεση προγράμματος). Για να ορίσουμε τα στάδια της αποσφαλμάτωσης χρησιμοποιήσαμε το μοντέλο των Carver και Klahr (1986) και με αυτό δημιουργήθηκε μια μεταβλητή τριών βαθμίδων. Χρησιμοποιώντας κατάλληλα τα δεδομένα μας, καθορίσαμε εάν τα παιδιά μετά από μια λάθος εκτέλεση προγράμματος: (α) έφτασαν σε πλήρη επίλυση του σφάλματος (Πλήρης διόρθωση), (β) εντόπισαν ότι ένα συγκεκριμένο λάθος υπήρχε αλλά δεν μπόρεσαν να το εντοπίσουν και να το διορθώσουν (Μερική διόρθωση) και (γ) δεν εντόπισαν καμία διαφορά ανάμεσα στο αποτέλεσμα και στο πρόβλημα που τους δόθηκε, δηλαδή δεν έκαναν αποσφαλμάτωση (Καμία διόρθωση). Οι παραπάνω μεταβλητές εξετάστηκαν σαν συνάρτηση του φύλου και της ηλικίας.

### Αποτελέσματα

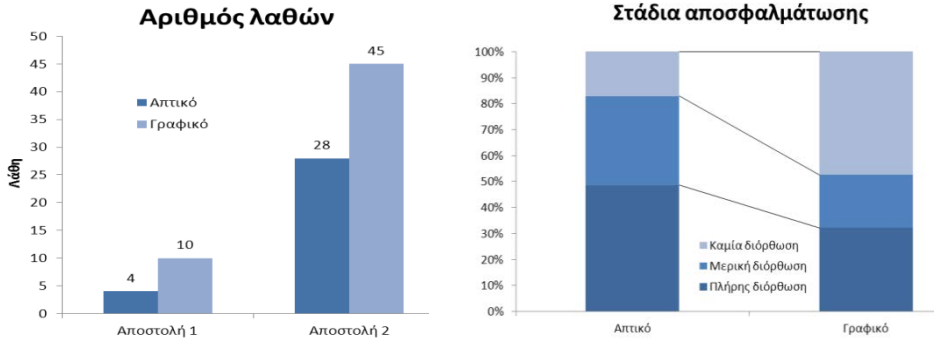
Σε ότι αφορά τους χρόνους ολοκλήρωσης (Σχήμα 3) της αποστολής 1, ο χρόνος ολοκλήρωσής της (Time to Accomplish Task1 - TAT1) ήταν αρνητικά συσχετισμένος με την ηλικία και για τις δύο συνθήκες (Pearson,  $r = - 0.63$ , ( $p < 0.001$ ) για το απτικό και  $r = - 0.68$  ( $p < 0.001$ ) για το γραφικό).



Σχήμα 3. Γράφημα TAT vs. Age για τις αποστολές 1 και 2

Για να εντοπίσουμε στις διαφορές ανάμεσα στα δύο συστήματα για την κάθε ηλικιακή ομάδα ξεχωριστά, προχωρήσαμε στην ανάλυση τόσο με t-test όσο και με non parametric Wilcoxon Signed Rank Test. Οι δύο αναλύσεις έδειξαν ταυτόχρονα ότι, για την περίπτωση της απτικής διεπαφής, ο χρόνος επίτευξης της λύσης ήταν στατιστικά σημαντικά μικρότερος για τις ηλικιακές ομάδες 6-7 ( $p=0.049$ ), 7-8 ( $p=0.030$ ) και 9-10 ( $p=0.012$ ), ενώ οι διαφορές για τις ηλικιακές ομάδες 10-11 και 11-12 δεν ήταν στατιστικά σημαντικές. Ο χρόνος ολοκλήρωσης της αποστολής 2 (Time to Accomplish Task2 - TAT2) έχει μια αρνητική συσχέτιση Pearson με την ηλικία και για τις δύο διεπαφές  $r = - 0.42$ , ( $p < 0.001$ ) για την απτική και  $r = - 0.44$  ( $p < 0.001$ ) για τη γραφική. Ενώ εάν επικεντρώσουμε στις δύο διεπαφές

για τις ίδιες ηλικιακές ομάδες τότε πραγματοποιώντας t-test ή και το μη παραμετρικό Wilcoxon Signed Rank Test δεν θα βρούμε στατιστικά σημαντικές διαφορές για καμία ηλικιακή ομάδα.



**Σχήμα 4. Γραφικές παραστάσεις για τον αριθμό λαθών και την αποσφαλμάτωση**

Το γράφημα του Σχήματος 4 που απεικονίζει τον αριθμό λαθών κατά τη διάρκεια των αποστολών δείχνει τον αριθμό των προγραμμάτων που εκτελέστηκαν με σφάλμα σε κάθε διεπαφή. Σε όλες τις περιπτώσεις περισσότερα λάθη εντοπίστηκαν με την γραφική διεπαφή. Η μελέτη των διαφορών έδειξε ότι μόνο στην περίπτωση της αποστολής 2 υπάρχει σημαντική διαφορά ανάμεσα στις δυο συνθήκες ( $\chi^2 = 3.96$ ,  $p < 0.05$ ). Εξετάζοντας τα στάδια αποσφαλμάτωσης των προγραμματιστικών λαθών (debugging stages) για τα δύο συστήματα, θα φτάσουμε σε δυο κατανομές οι οποίες θα περιγράφουν τι έκαναν οι χρήστες μετά από ένα προγραμματιστικό σφάλμα. Οι κατανομές αυτές απεικονίζονται στο γράφημα και μετά από στατιστική ανάλυση βλέπουμε ότι αυτές οι δύο κατανομές διαφέρουν μεταξύ τους ( $\chi^2 = 8.79$ ,  $df=2$ ,  $p < 0.05$ ). Έτσι στην περίπτωση της απτικής διεπαφής είναι πιο πιθανό ένα λάθος να διορθωθεί πλήρως ενώ στην περίπτωση της γραφικής διεπαφής είναι πιθανότερο το λάθος να μην εντοπιστεί. Σε ότι αφορά τις διαφορές ανάμεσα στη σύνθεση της ομάδος ή το φύλο δεν εντοπίστηκαν στατιστικά σημαντικές διαφορές.

## Συζήτηση

Σε αυτό το άρθρο παρουσιάστηκε εν συντομία το σύστημα PROTEAS το οποίο χρησιμοποιήθηκε για τη συγκριτική μελέτη που περιγράψαμε παραπάνω. Σε ό,τι αφορά τις αποστολές, η αποστολή 1 επιλέχθηκε σκοπίμως να είναι εύκολη προκειμένου να μπορέσει να εκπληρωθεί από όλα τα παιδιά. Η δεύτερη ομάδα αποστολών (αποστολή 2) ήταν πιο απαιτητική και για αυτό το λόγο δημιουργήθηκαν αρκετά προγραμματιστικά σφάλματα.

Σε ό,τι αφορά τον χρόνο ολοκλήρωσης των αποστολών, τα αποτελέσματά μας έδειξαν ότι ο χρόνος μειώνεται όσο αυξάνει η ηλικία και αυτό ισχύει για όλες τις αποστολές και για τις δύο συνθήκες (απτική, γραφική διεπαφή). Το συγκεκριμένο αποτέλεσμα ήταν αναμενόμενο, εάν λάβουμε υπόψη μας τις αυξημένες δυνατότητες και γνωστικές ικανότητες των μεγαλύτερων παιδιών. Εστιάζοντας στη σύγκριση ανάμεσα στα δύο συστήματα, ο χρόνος TAT1 για το απτικό σύστημα ήταν σημαντικά μικρότερος ειδικά για τα νεότερα παιδιά. Αυτό μπορεί, ενδεχομένως, να εξηγηθεί εάν αναλογιστούμε τις διαφορές στην οικειότητα των παιδιών με παρόμοια απτικά παιχνίδια (όπως τα τουβλάκια Lego, τα πάζλ κλπ) και την χρήση υπολογιστή. Στην περίπτωση των μεγαλύτερων παιδιών η εμπειρία τους με τους υπολογιστές έχει αυξηθεί και έτσι οι διαφορές των χρόνων TAT1 ανάμεσα στις δύο συνθήκες

είναι αναμενόμενο να μειωθούν. Τέλος, ο χρόνος TAT1 για τη γραφική διεπαφή φτάνει να είναι μικρότερος ακόμα από τον χρόνο της απτικής διεπαφής για την ηλικία 11-12 αλλά αυτό το φαινόμενο δεν είναι στατιστικά σημαντικό. Αυτή η τάση εντοπίστηκε σε όλες τις αποστολές και πιθανώς δείχνει ότι η συγκεκριμένη ηλικία είναι το κατώφλι, στο οποίο τα παιδιά αποκτούν το απαιτούμενο επίπεδο ικανοτήτων χειρισμού υπολογιστή, το οποίο τους επιτρέπει να μειώσουν περαιτέρω το χρόνο επίτευξης των αποστολών με το γραφικό υποσύστημα.

Γενικά, τα συγκεκριμένα αποτελέσματα τα οποία αφορούν στους χρόνους επίτευξης των αποστολών είναι σύμφωνα με τα αποτελέσματα στην έρευνα (Sapounidis & Demetriadis, 2013) και ταυτόχρονα εμπλουτίζουν τα αποτελέσματα που αναφέρθηκαν από το Xie et al. (2008), ο οποίος ανέφερε ότι για παιδιά 7 έως 9 ετών ήταν ευκολότερο να αλληλεπιδράσουν με φυσικά αντικείμενα παρά με το ποντίκι του υπολογιστή. Συνοψίζοντας, κάποιος μπορεί να πει ότι η οικειότητα με παρόμοια απτικά συστήματα και από την άλλη μεριά η συσσωρευμένη εμπειρία με τους υπολογιστές είναι ο πιο πιθανός παράγοντας ο οποίος μπορεί να εξηγήσει τα αποτελέσματα μας (Sapounidis & Demetriadis, 2013).

Αν αναλογιστούμε τα προγραμματιστικά σφάλματα, τα παιδιά υπέπεσαν σε περισσότερα σφάλματα με το γραφικό υποσύστημα σε όλες τις περιπτώσεις. Ειδικότερα, στην αποστολή 2 η οποία ήταν δυσκολότερη, ο αριθμός των λαθών ήταν στατιστικά σημαντικά μικρότερος για την περίπτωση του απτικού υποσυστήματος. Αυτή η στατιστικά σημαντική διαφορά δεν παρατηρήθηκε στην περίπτωση της αποστολής 1. Η αποστολή 1 ήταν η ευκολότερη και πολύ μικρός αριθμός λαθών εντοπίστηκε και άρα είχαμε μικρό αριθμό περιπτώσεων για να εξαχθεί κάποια στατιστική σημαντικότητα.

Ο μικρότερος αριθμός λαθών, κατά τη διάρκεια του προγραμματισμού του ρομπότ με το απτικό σύστημα, πιθανώς να οφείλεται στην ενεργή συμμετοχή και στα κίνητρα των παιδιών. Τα παιδιά συμμετέχουν πιο ενεργά σε προγραμματιστικές δραστηριότητες με το απτικό σύστημα από ότι με αντίστοιχο γραφικό (Horn et al., 2012). Αρκετές έρευνες έχουν δείξει ότι η εργασία σε ζευγάρια σε συγκεκριμένες συνεργατικές δραστηριότητες μπορεί να αυξήσει την ευχαρίστηση, την εμπλοκή και τα κίνητρα για συμμετοχή (π.χ. Scott et al., 2003; Inkpen et al., 1995)). Στην περίπτωση της απτικής διεπαφής και οι δύο συμμετέχοντες μπορούν να είναι ταυτόχρονα ενεργοί. Έτσι, έχουν και οι δύο κίνητρα. Αντίθετα στην περίπτωση της γραφικής συνθήκης, μόνο ένα παιδί αλληλεπιδρά ενώ το άλλο είναι πιθανώς παθητικός παρατηρητής με λιγότερα κίνητρα (Stamovlasis, Dimos, & Tsapralis, 2006).

Η αλληλεπίδραση των παιδιών με το απτικό σύστημα φαίνεται να επιτρέπει την καλύτερη αποσφαλμάτωση μετά από ένα λάθος. Χρησιμοποιώντας την απτική διεπαφή, είναι πιο πιθανό να συμβεί πλήρεις αποσφαλμάτωση και διόρθωση του λάθους ενώ με το γραφικό υποσύστημα είναι πιο πιθανό το λάθος να παραβλεφθεί. Αυτό το εύρημα μπορεί ξανά να αποδοθεί στην ενεργή συμμετοχή των δύο μελών της ομάδος στην περίπτωση του απτικού συστήματος, ενώ στην περίπτωση της γραφικής διεπαφής το ένα μέλος χρησιμοποιεί το ποντίκι και προγραμματίζει σαν ενεργό μέλος ενώ το άλλο μέλος είναι, πολύ πιθανόν, ένας παθητικός παρατηρητής. Δουλεύοντας με τις απτικές διεπαφές οι χρήστες μπορούν ισάξια να συμμετέχουν (Rogers et al., 2009), (Falcão & Price, 2009) και πιθανώς νοιώθουν εξίσου υπεύθυνοι να επιβλέπουν την εκτέλεση του προγράμματος. Ακόμα, με τις απτικές διεπαφές τα άτομα μπορούν να αυξήσουν την ορατότητα τους προς το πεδίο δράσης (Stanton et al., 2001; Klemmer, Hartmann, & Takayama, 2006) και συνεπώς και τα δύο μέλη μπορούν ευκολότερα να εντοπίσουν πιθανά σφάλματα.

## Αναφορές

- Carver, S. M., & Klahr, D. (1986). Assessing children's LOGO debugging skills with a formal model. - *Journal of Educational Computing Research*, 2(4), 487-525. doi:- 10.2190/KRD4-YNHH-X283-3P5V.
- Falcão, T. P., & Price, S. (2009). What have you done! the role of 'interference' in tangible environments for supporting collaborative learning. Paper presented at the 9th CSCL, Rhodes, Greece. pp. 325-334.
- Horn, M. S. (2009). Tangible computer programming: Exploring the use of emerging technology in classrooms and science museums Tufts University.
- Horn, M., Crouser, R., & Bers, M. (2012). Tangible interaction and learning: The case for a hybrid approach. *Personal and Ubiquitous Computing*, 16(4), pp. 379-389.
- Inkpen, K., Booth, K. S., Gribble, S. D., & Klawe, M. (1995). Give and take: Children collaborating on one computer. Paper presented at the CHI '95 Conference Companion on Human Factors in Computing Systems, Denver, Colorado, United States. pp. 258-259.
- Ishii, H., & Ullmer, B. (1997). Tangible bits: Towards seamless interfaces between people, bits and atoms. Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Atlanta, Georgia, United States. pp. 234-241.
- Kelleher, C., & Pausch, R. (June 2005) Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37(2), 83-137.
- Klemmer, S. R., Hartmann, B., & Takayama, L. (2006). How bodies matter: Five themes for interaction design. Paper presented at the Proceedings of the 6th Conference on Designing Interactive Systems, University Park, PA, USA. pp. 140-149. doi:10.1145/1142405.1142429
- Marshall, P. (2007). Do tangible interfaces enhance learning? Paper presented at the Proceedings of the 1st International Conference on Tangible and Embedded Interaction, Baton Rouge, Louisiana, USA. pp. 163-170.
- McNerney, T. S. (2004). From turtles to tangible programming bricks: Explorations in physical language design. *Personal and Ubiquitous Computing*, 8(5), 326-337.
- Nusen, N., & Sipitakiat, A. (2011) Robo-blocks: A tangible programming system with debugging for children.
- Orit, S., & Eva, H. (2009). Tangible user interfaces: Past, present, and future directions. *Foundations and Trends® in Human-Computer Interaction*, 3(1-2), 1-137.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc.
- Price, S., Rogers, Y., Scaife, M., Stanton, D., & Neale, H. (April 2003). Using 'tangibles' to promote novel forms of playful learning. *Interacting with Computers*, 15(2), 169-185.
- Sapounidis, T., & Demetriadis, S. (2009). Tangible programming interfaces: A literature review. 4th Balkan Conference in Informatics, Thessaloniki, GREECE. pp. 70-75.
- Sapounidis, T., & Demetriadis, S. (2011). Touch your program with hands: Qualities in tangible programming tools for novice. Paper presented at the 15th Panhellenic Conference on Informatics (IEEE/PCI), pp. 363-367.
- Sapounidis, T., & Demetriadis, S. (2013). Tangible versus graphical user interfaces for robot programming: Exploring cross-age children's preferences. *Personal and Ubiquitous Computing*, 17(8), 1775-1786. doi:10.1007/s00779-013-0641-7
- Scott, S. D., Mandryk, R. L., & Inkpen, K. M. (2003). Understanding children's collaborative interactions in shared environments. *Journal of Computer Assisted Learning*, 19(2), 220-228. doi:10.1046/j.0266-4909.2003.00022.x
- Stamovlasis, D., Dimos, A., & Tsapalis, G. (2006). A study of group interaction processes in learning lower secondary physics. *Journal of Research in Science Teaching*, 43(6), 556-576. doi:10.1002/tea.20134
- Stanton, D., Bayon, V., Neale, H., Ghali, A., Benford, S., Cobb, S., et al. (2001). Classroom collaboration in the design of tangible interfaces for storytelling. Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. United States. pp. 482-489.
- Suzuki, H., & Kato, H. (1993). AlgoBlock: A tangible programming language, a tool for collaborative learning. Paper presented at the Proceedings of 4th European Logo Conference, Athens, Greece. pp. 297-303.



Xie, L., Antle, A. N., & Motamedi, N. (2008). Are tangibles more fun?: Comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces. Paper presented at the Proceedings of the 2nd International Conference on Tangible and Embedded Interaction (TEI '08), pp. 191-198.