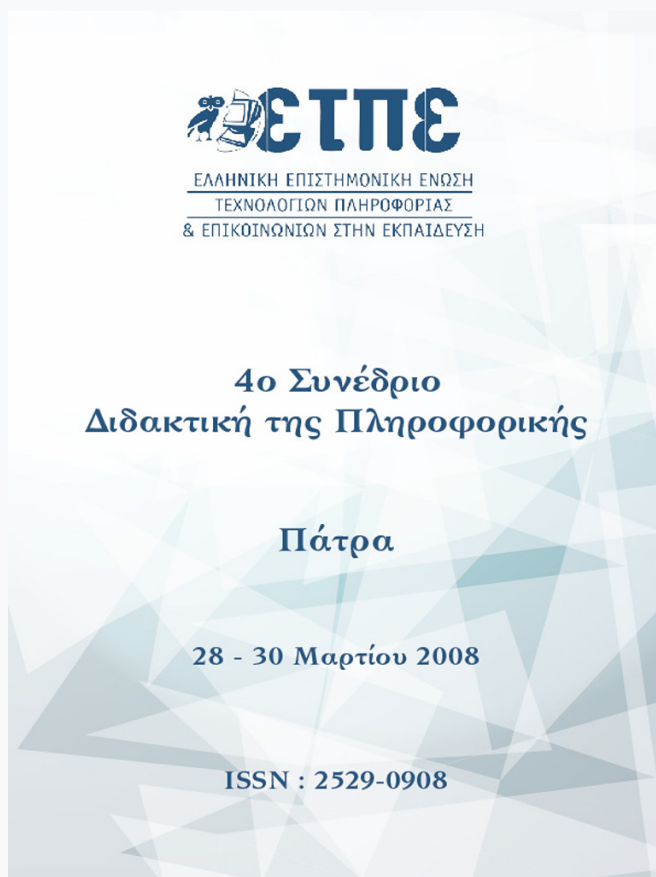


Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2008)

4ο Συνέδριο Διδακτική Πληροφορικής



Περιβάλλοντα απτού προγραμματισμού:
Επισκόπηση της παρούσας κατάστασης και
ανοιχτά ερευνητικά ερωτήματα

Θ. Σαπουνίδης, Σ. Δημητριάδης, Ι. Βλαχάβας

Βιβλιογραφική αναφορά:

Σαπουνίδης Θ., Δημητριάδης Σ., & Βλαχάβας Ι. (2023). Περιβάλλοντα απτού προγραμματισμού: Επισκόπηση της παρούσας κατάστασης και ανοιχτά ερευνητικά ερωτήματα. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, 1, 061-070. ανακτήθηκε από <https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/5850>

Περιβάλλοντα απτού προγραμματισμού: Επισκόπηση της παρούσας κατάστασης και ανοιχτά ερευνητικά ερωτήματα

Θ. Σαπουνίδης, Σ. Δημητριάδης, Ι. Βλαχάβας

Τμήμα Πληροφορικής, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
{teo, sdemetri, vlavahas}@auth.gr

Περίληψη

Αυτό το άρθρο παρουσιάζει γλώσσες προγραμματισμού και εκπαιδευτικά περιβάλλοντα, που υιοθετούν απτές διεπαφές χρήστη προκειμένου να διευκολύνουν τους χρήστες, κάθε ηλικίας, στην εκμάθηση εννοιών του προγραμματισμού. Τα συστήματα παρουσιάζονται με τη χρονολογική σειρά εμφάνισής τους ενώ παράλληλα, όπου είναι δυνατόν, γίνεται αναφορά στα μειονεκτήματα – πλεονεκτήματα των προσεγγίσεων που ακολούθησαν οι δημιουργοί τους. Τέλος, παρουσιάζονται τα ανοιχτά ερευνητικά ερωτήματα, που σύμφωνα με πρόσφατες έρευνες, προτείνεται να διερευνηθούν.

Λέξεις κλειδιά: *Απτές διεπαφές χρήστη, Προγραμματισμός, Εκπαίδευση.*

Abstract

This paper presents the tangible programming languages and the educational environments which were designed to make programming concepts more accessible to novice programmers of all ages. All languages are presented according to the chronological order of their development. Additionally, the paper provides information about the advantages and disadvantages of the design approach. Finally we compile a table of open questions for future research in the domain of tangible programming environments.

Keywords: *Tangible user interfaces, Programming, Education.*

1. Εισαγωγή

Η πρόσφατη έρευνα στις απτές διεπαφές χρήστη (tangible user interfaces), όπως ορίστηκαν από τους Ishii και Ullmer, (Ishii et al., 1997) δημιούργησε εξαιρετικές ευκαιρίες για καινοτόμα εφαρμογή της τεχνολογίας μέσα σε τάξεις (π.χ. Ichida et al., 2004; Itoh Y et al., 2004). Μια περιοχή που φαίνεται να έχει επωφεληθεί από αυτό του είδους την τεχνολογία είναι αυτή των περιβαλλόντων απτού προγραμματισμού με εφαρμογή στην εκπαίδευση και όχι μόνο (McNerny 2004; Blackwell 2003). Ένα περιβάλλον απτού προγραμματισμού μπορεί να έχει όμοια αποτελέσματα χρήσης με μια text-based ή visual-based γλώσσα. Η ιδιαιτερότητα αυτών των περιβαλλόντων έγκειται στο ότι αντί να χρησιμοποιούνται visual αντικείμενα ή γραπτές εντολές στην οθόνη του υπολογιστή, αξιοποιούνται πραγματικά αντικείμενα του φυσικού κόσμου (π.χ. Wyeth et al., 2003; Smith et al., 2007; Horn et al., 2007).

2. Συστήματα απτών διεπαφών για τον προγραμματισμό

Η Radia Perlman, ερευνήτρια του MIT media lab, στα τέλη του 1970 διαπίστωσε ότι τα περισσότερα παιδιά, κάτω των 11 - 14 χρονών, δεν ήταν έτοιμα να αρχίσουν να προγραμματίζουν με τον παραδοσιακό τρόπο (Kelleher et al., 2005) (πληκτρολογώντας κώδικα logo σε έναν υπολογιστή με πληκτρολόγιο). Ένα από τα σημαντικότερα προβλήματα των παιδιών στον προγραμματισμό δεν ήταν μόνο η σύνταξη του κώδικα αλλά η διεπαφή χρήστη. Η Perlman τότε άρχισε να σχεδιάζει κάποιες διεπαφές που θα επέτρεπαν ακόμα και στα παιδιά προσχολικής ηλικίας να μάθουν να προγραμματίζουν μια χελώνα. Προϊόν αυτών των προσπαθειών ήταν η πρώτη διεπαφή αυτού του είδους το Tortis - Slot machine (Kelleher et al., 2005). Από εκείνη την ημέρα μέχρι και σήμερα ακολούθησαν διάφορες σχεδιαστικές προτάσεις οι οποίες θα παρουσιαστούν συνοπτικά σ' αυτό το άρθρο.

2.1 Tortis - Slot machine

Το Slot machine (Kelleher et al., 2005; McNerny, 2004), ήταν το πρώτο σύστημα που κατασκευάστηκε και εισήγαγε τις πλαστικές κάρτες που μπορούσαν να μπουν σε τρεις χρωματιστές στοίβες (κόκκινη, πράσινη και μπλε). Στο αριστερό άκρο κάθε στοίβας υπήρχε ένα 'Do it' κουμπί. Όταν κάποιος πατούσε το κουμπί, μια virtual χελώνα εκτελούσε την ενέργεια που φαινόταν στη φωτογραφία της κάθε κάρτας της πρώτης στοίβας. Όταν μια ενέργεια εκτελούνταν, η λάμπα κάτω από τη συγκεκριμένη κάρτα άναβε. Το Slot machine προσέφερε μεταξύ άλλων κάποιες σημαντικές δυνατότητες στον χρήστη όπως η άμεση δυνατότητα χειρισμού του προγράμματος είτε προσθέτοντας είτε αναδιατάσσοντας είτε ακόμα και βγάζοντας κάρτες. Για να δοθεί η δυνατότητα της κλήσης διαδικασίας η Perlman εισήγαγε ειδικές κάρτες (McNerny, 2004).

2.2 AlgoBlock

Τα AlgoBlock (Suzuki et al., 1993) είναι ένα εκπαιδευτικό περιβάλλον προγραμματισμού, με απτά στοιχεία, για το δημοτικό και τη μέση εκπαίδευση. Το σύστημα είναι μια συλλογή από φυσικούς κύβους που μπορούν να συνδεθούν μεταξύ τους προκειμένου να σχηματίσουν ένα πρόγραμμα. Αυτοί οι κύβοι στη συνέχεια συνδέονται με έναν υπολογιστή για να εκτελεστεί το πρόγραμμα. Ο κάθε κύβος αντιστοιχεί σε μια εντολή που μοιάζει με τις εντολές της Logo. Τα AlgoBlock κατασκευάστηκαν για να προωθήσουν τη συνεργασία. Δουλεύουν σαν ένα εργαλείο για συνεργατικές δραστηριότητες και βοηθούν τους εκπαιδευόμενους να χτίσουν προγράμματα μέσα από κοινωνικές αλληλεπιδράσεις και συζήτηση. Οι εκπαιδευόμενοι έχουν σαν στόχο να καθοδηγούν ένα υποβρύχιο που απεικονίζεται σε μια CRT οθόνη με Logo-like εντολές. Τέτοιες εντολές μπορούν να είναι πήγαινε μπροστά, στρίψε αριστερά - δεξιά, γύρνα ξεδίπλωσε τα πόδια για να απογειωθείς κ.λπ. Η κάθε μια από αυτές τις εντολές έχει ένα αντίστοιχο κουτί που την εκφράζει.

2.3 Music Block

Το σύστημα Music Block (McNerny, 2004), το οποίο μάλιστα είναι και εμπορικό, είναι ένα από τα συστήματα που μπορούν να παίξουν μουσική χρησιμοποιώντας απτές εντολές. Το σύστημα έχει πέντε χρωματιστούς κύβους, ο καθένας από αυτούς αντιπροσωπεύει μια μουσική φάση του Eine Kleine Nachtmusic. Ο χρήστης βάζει κουτάκια σε ένα σύστημα αναπαραγωγής. Όταν τα κυβάρια μπουν στις τρύπες τότε ακούγεται η φάση του κομματιού που αντιπροσωπεύει το κάθε κουτί. Αν πατηθεί βέβαια το play (το μοναδικό κουμπί) τότε ακούγεται ολόκληρη η αλληλουχία. Βέβαια είναι φτιαγμένο έτσι ώστε οποιαδήποτε και να είναι η σειρά τοποθέτησης των κουτιών το αποτέλεσμα να μην είναι αποκρουστικό. Για μεγαλύτερη ποικιλία η κάθε πλευρά των κύβων παίζει και διαφορετικό μουσικό κομμάτι.

2.4 Block Jam

Το σύστημα Block Jam (Newton et al., 2003) αποτελείται από μια απτή διεπαφή και ταυτόχρονα είναι μια οπτική γλώσσα η οποία επιτρέπει σε αρχάριους και μη να εκμεταλλευτούν τις δυνατότητές της. Το σύστημα αυτό αποτελείται από 26 αντικείμενα τα οποία ανάλογα με την τοπολογία σύνδεσης τους μπορούν να παράγουν μουσικές 'φράσεις'. Υπάρχουν δύο είδη κουτιών, τα play blocks και τα path blocks. Τα play blocks είναι αυτά που ξεκινούν, σταματούν και ελέγχουν την ταχύτητα της cue ball ενώ τα path blocks ελέγχουν τη διαδρομή της νοητής cue ball. Όταν τα κουτιά συνδεθούν μεταξύ τους δημιουργούν ένα σύμπλεγμα. Το σύμπλεγμα αυτό συνδέεται με τον υπολογιστή μέσω ενός καλωδίου και ενός κουτιού βάσης (mother block). Το καλώδιο δίνει ρεύμα στο σύμπλεγμα καθώς και έναν δίαυλο επικοινωνίας με τον υπολογιστή. Η cue ball κινείται νοητά από το ένα κουτί στο άλλο ανάλογα με τους κανόνες που επιβάλλουν τα path blocks. Το κάθε κουτί περιέχει, μεταφορικά βέβαια, κάποιον ήχο και έτσι καθώς η cue ball ή οι cue balls κινούνται από κουτί σε κουτί παράγουν ήχους δημιουργώντας μουσική.

2.5 Απτός προγραμματισμός για τρένα

Ο Genee Lyn Colobong (McNerny, 2004) δημοσίευσε το 1999 μια γλώσσα προγραμματισμού για τρένα η οποία μάλιστα έγινε εμπορική από κάποιες εταιρίες. Αποτέλεσμα αυτής της έρευνας ήταν και το σύστημα Lego intelli - train του οποίου η συμπεριφορά προγραμματιζόταν με την τοποθέτηση κύβων - εντολών στη γραμμή. Για παράδειγμα, όταν το τρένο περνούσε από το 'toot' τουβλάκι ακουγόταν η κόρνα του. Όταν το τρένο περνούσε από το τουβλάκι 'αναστροφή' άλλαζε κατεύθυνση. Όταν έφτανε στο τουβλάκι στάση σταματούσε και περίμενε από το παιδί να πατήσει το κουμπί 'go'. Προσθέτοντας, αφαιρώντας, αναδιατάσσοντας τα τουβλάκια στη γραμμή τα παιδιά μπορούσαν να αλλάζουν το πρόγραμμα, ακόμα και όταν το τρένο βρισκόταν σε κίνηση. Μετά από κάποια πειράματα παρατηρήθηκε ότι η κατασκευή της γραμμής και η τοποθέτηση εντολών που θα εκτελούσε αργότερα το τρένο δεν ήταν διαισθητική διαδικασία για τα τετράχρονα παιδιά. Προκειμένου να δουν τι

κάνουν τα τουβλάκια τα έβαζαν ακριβώς μπροστά στο τρένο πράγμα που προκαλούσε εκτροχιασμούς.

2.6 Tangible Programming Brick

Η έρευνα του McNerney (2001) στις απτές διεπαφές χρήστη άρχισε από το 2000 με την κατασκευή του συστήματος «Tangible Programming Brick» που ήταν ένα ακόμα περιβάλλον απτού προγραμματισμού. Οι κατασκευαστές αποφάσισαν να φτιάξουν ένα μονοδιάστατο σύστημα το οποίο βάζοντας το ένα τουβλάκι Lego πάνω στο άλλο θα έφτιαχναν μια αλληλουχία από εντολές. Για να γίνει το σύστημα πιο ισχυρό έφτιαζαν τα τουβλάκια έτσι ώστε να παίρνουν μια κάρτα στα πλάγια. Αυτή η μέθοδος αρχικά είχε σκοπό να δώσει τη δυνατότητα στο σύστημα να αποκτήσει παραμέτρους, αλλά στην πορεία φάνηκε ότι μπορούσαν να τοποθετήσουν και άλλα πράγματα όπως διακόπτες, αισθητήρια κ.λπ. Τελικά, τα αποτελέσματα περιορισμένης έρευνας, έδειξαν ότι το σύστημα ήταν αρκετά πολύπλοκο για χρήστες κάτω των 6 ετών.

2.7 Electronic blocks

Τα electronic block (Wyeth et al., 2003) έχουν σχεδιαστεί έτσι ώστε τα παιδιά να μπορούν να τα συνδέσουν όπως κάνουν και με οποιουδήποτε άλλους κύβους. Βάζοντας τα electronic block το ένα πάνω στο άλλο τα μικρά παιδιά φτιάχνουν προγράμματα που το κάθε ένα από αυτά εκτελεί και διαφορετική διεργασία. Τα electronic block έχουν εισόδους, εξόδους και όταν συνδεθούν η έξοδος του ενός κουτιού ελέγχει την είσοδο του άλλου. Υπάρχουν τρία είδη κουτιών:

- Τα κουτιά αισθητήρες (Sensor blocks)
- Τα κουτιά ενέργειες (Action blocks)
- Τα λογικά κουτιά (Logic blocks)

Τα λογικά κουτιά ανιχνεύουν φως, αφή και ήχο στο περιβάλλον. Τα κουτιά αισθητήρες είναι αυτά που δίνουν σήμα στα κουτιά ενέργειες να εκτελέσουν κάποια λειτουργία. Τα λογικά κουτιά έχουν ένα ενδιάμεσο ρόλο. Τοποθετώντας τα λογικά κουτιά ανάμεσα στα κουτιά αισθητήρες και στα κουτιά ενέργειες, έχουμε τη δυνατότητα να αλλάξουμε την αναμενόμενη ενέργεια. Ακόμα και πολύ μικρά παιδιά, σύμφωνα με τους Wyeth et al. (2003), μπορούν να χρησιμοποιήσουν μια συλλογή από electronic blocks. Ένα απλό παιχνίδι με τα action block μπορεί να παράγει κάποιου είδους ενέργειες που τα παιδιά θα βρουν εξαιρετικά ενδιαφέρονσα.

2.8 AutoHAN – Media Cubes

Το AutoHAN (Blackwell et al., 2001) είναι μια δικτυακή και προγραμματιστική αρχιτεκτονική που επιτρέπει τον προγραμματισμό ανάμεσα σε διάφορες συσκευές μιας οικίας. Ένα τμήμα του AutoHAN είναι οι mediaCubes. Οι mediaCubes είναι μια γλώσσα προγραμματισμού η οποία φιλοδοξεί να χρησιμοποιηθεί από απλούς χρήστες όπως αυτούς που μπορούν να χειριστούν ένα απλό τηλεχειριστήριο. Ο κάθε κύβος

του συστήματος είναι μια χειροπιαστή διαφανής διεπαφή η οποία είναι συσχετισμένη με μια λειτουργία μιας οικιακής συσκευής. Εάν για παράδειγμα ο κύβος αναπαριστά το play / pause, ο χρήστης μπορεί να συσχετίσει αυτόν τον κύβο με το CD player. Ο συνδυασμός περισσοτέρων του ενός κύβου, που αναπαριστά διαφορετικές λειτουργίες της ίδιας συσκευής είναι η διαδικασία που θα πρέπει να ακολουθηθεί προκειμένου να ολοκληρωθεί το 'γράψιμο' ενός προγράμματος.

Βέβαια, κάποιος μπορεί να πει ότι ο προγραμματισμός του VCR δεν είναι προγραμματισμός όπως αυτός του ηλεκτρονικού υπολογιστή. Αν όμως θέλουμε να δώσουμε την εντολή στο VCR να αρχίσει να γράφει από την κάμερα ασφαλείας της μπροστινής πόρτας για διάστημα 5 λεπτών από τη στιγμή που κάποιος ανιχνευτής εντόπισε κίνηση, τότε μιλάμε για προγραμματισμό.

2.9 Απότς προγραμματισμός με 'σπάγκους'

Πρόκειται για ένα σύστημα το οποίο χρησιμοποιείται για τη δημιουργία απλών προγραμμάτων προκειμένου να ελεγχθούν ρομποτικά παιχνίδια (Patten et al., 2000). Για να δημιουργήσουν οι χρήστες ένα πρόγραμμα, πρέπει να συνδεθούν γεγονότα με ενέργειες που εκτελούνται ως ανταπόκριση σε αυτά τα γεγονότα. Αυτές οι σχέσεις παρουσιάζονται ως εικόνες που συνδέονται με κάποιο φυσικό 'σπάγκο' (πρόκειται για καλώδιο που μοιάζει με 'σπάγκο'). Ο χρήστης συνδέει γεγονότα που δημιουργούνται από τους αισθητήρες του ρομπότ με ενέργειες που εκτελούνται ως απόκριση σε αυτά τα ερεθίσματα. Το σύστημα έτσι, δημιουργεί ένα πρόγραμμα το οποίο μπορεί να φορτωθεί στη μνήμη του ρομπότ και να λειτουργεί σύμφωνα με τον τρόπο που περιγράφεται από τις συνδέσεις των 'σπάγκων'.

2.10 GameBlocks

Το σύστημα αυτό (Smith, 2007), αποτελείται από φυσικούς κύβους οι οποίοι τοποθετούνται πάνω σε ράγες και δημιουργούν έτσι μια ακολουθία από 'εντολές – κουτιά'. Η σχετική θέση των κύβων είναι σημαντική αφού εκφράζει μια λογική ακολουθία, η οποία είναι το πρόγραμμα. Το σύστημα διαθέτει συνολικά 6 εντολές για τον έλεγχο ενός ανθρωποειδούς ρομπότ. Οι εντολές που διατίθενται είναι: εμπρός, πίσω, σώμα αριστερά, σώμα δεξιά, κεφάλι αριστερά και κεφάλι δεξιά. Ένα ενδιαφέρον χαρακτηριστικό του συστήματος είναι ότι δεν απαιτούνται ενσωματωμένα ηλεκτρονικά κυκλώματα μέσα στο κάθε κουτί μια και οι εργασίες αναγνώρισης γίνονται από κυκλώματα στις ράγες. Αξίζει, τέλος να σημειωθεί ότι το σύστημα δεν απαιτεί τη χρήση υπολογιστή αφού η υπολογιστική ισχύς που χρειάζεται παρέχεται από μικροϋπολογιστές ενσωματωμένους στο κύκλωμα της βάσης.

2.11 Quetzal - Tern

Το Quetzal και το Tern (Horn et al., 2007) είναι δύο ακόμα περιβάλλοντα απτού προγραμματισμού με εκπαιδευτικό προσανατολισμό. Τα περιβάλλοντα αυτά,

αποτελούνται από συμπαγή αντικείμενα χωρίς ενσωματωμένα ηλεκτρονικά κυκλώματα και τροφοδοσίες. Τα προγραμματιστικά ‘κομμάτια’ των δυο περιβαλλόντων είναι πανομοιότυπα και μοιάζουν με τα κομμάτια ενός Puzzle. Το Quetzal είναι ένα σύστημα που χρησιμοποιείται για να ελέγχει ένα Lego Mindstorms ρομπότ ενώ το Tern ένα Virtual ρομπότ στην οθόνη υπολογιστή. Η φιλοσοφία κατασκευής και λειτουργίας των δυο αυτών περιβαλλόντων είναι κοινή αφού έχουν αναπτυχθεί από τους ίδιους ανθρώπους και μάλιστα την ίδια χρονική περίοδο. Οι μαθητές που χρησιμοποιούν τα περιβάλλοντα, προγραμματίζουν σε offline mode. Προκειμένου να γίνει η ανάγνωση του κώδικα χρησιμοποιείται ένα φορητό σύστημα σάρωσης. Το σύστημα σάρωσης αναγνωρίζει το πώς και ποιες εντολές έχουν συνδεθεί μεταξύ τους και έτσι δημιουργείται, μετά από αυτόματη αναγνώριση, το προς εκτέλεση πρόγραμμα. Είναι, τέλος αξιοσημείωτο ότι και τα δυο περιβάλλοντα επιτρέπουν στον χρήστη να εισάγει παραμέτρους στις εντολές.

3. Χαρακτηριστικά συστημάτων απτού προγραμματισμού

Μελετώντας τα άρθρα που έχουν δημοσιευτεί και αφορούν τα παραπάνω περιβάλλοντα απτού προγραμματισμού, μπορούμε να παραθέσουμε τα συνολικά χαρακτηριστικά των κατασκευαστικών προσεγγίσεων που ακολουθήθηκαν, από τους σχεδιαστές των συστημάτων (πίνακας 1).

Πίνακας 1: Χαρακτηριστικά συστημάτων

Χαρακτηριστικά Συστημάτων	Συστήματα (παράγραφοι που περιγράφονται)										
	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	2.10	2.11
Υποστηρίζει Loop	X	X		X		X					X
Υποστηρίζει For						X					X
Υποστηρίζει Call	X										X
Υποστηρίζει παραμέτρους						X					X
Διάδραση πάνω στη διεπαφή (π.χ. φυσικός κύβος)		X			X	X	X				
Αποτέλεσμα του προγραμματισμού στον ίδιο φυσικό χώρο			X	X	X	X	X		X	X	X
Αξιοποίηση σχήματος διεπαφής							X				X
Εμπορική εφαρμογή			X		X			X			
Ανεξαρτησία Υπολογιστή			X		X	X	X	X		X	

Σε γενικές γραμμές, με εξαίρεση το σύστημα AutoHAN, τα περιβάλλοντα απτού προγραμματισμού απευθύνονται σε παιδιά του νηπιαγωγείου έως και γυμνασίου. Το κόστος ενός τέτοιου συστήματος, η κατασκευαστική του πολυπλοκότητα καθώς και η

έλλειψη εκπαιδευτικής αξιολόγησης περιόρισαν τη χρήση τέτοιων συστημάτων σε ερευνητικά κέντρα (Horn et al., 2006). Οι συνεχιζόμενες όμως κατασκευαστικές προσπάθειες από τους διάφορους ερευνητές μας δίνουν τη δυνατότητα να διατυπώσουμε ορισμένες γενικές υποδείξεις για την ανάπτυξη τέτοιων συστημάτων:

- Σύμφωνα με κάποιους ερευνητές τα συστήματα πρέπει να έχουν δυνατότητα εισαγωγής παραμέτρων (Caitlin et al., 2005; Cockburn et al., 1997; Sutphen et al., 2000).
- Ένα ενδιαφέρον στοιχείο που δε φαίνεται να καλύπτουν τα σημερινά συστήματα είναι η δυνατότητα αποθήκευσης του κώδικα (Kahn 1996).
- Ένας σύγχρονος τρόπος λειτουργίας (δηλαδή, άμεση εκτέλεση ενός προγραμματιστικού στοιχείου με την τοποθέτηση του στο πρόγραμμα) είναι μια ιδιότητα που μπορεί να αξιοποιηθεί στα αρχικά στάδια εξοικείωσης με τον απτό προγραμματισμό (Ichida et al., 2004).
- Μια πιθανή επέκταση των περιβαλλόντων απτού προγραμματισμού στις 3 διαστάσεις (3D) είναι ένα στοιχείο που παρουσιάζει ενδιαφέρον για συστήματα όπως το Flowblocks (Zuckerman et al., 2005) και το Gameblocks (Smith, 2006).
- Η δικτυακή επέκταση του συστήματος είναι ένα στοιχείο επίσης με ερευνητικό ενδιαφέρον (Frei et al., 2000).
- Προτείνεται να μην υπάρχει διαχωρισμός ανάμεσα σε είσοδο και έξοδο, δηλαδή το αποτέλεσμα του προγραμματισμού να παρουσιάζεται στον ίδιο φυσικό χώρο. (McNerny 2004; Kitamura et al., 2000; Zuckerman et al., 2005; Fernaeus et al., 2006).
- Η αυξημένη διάδραση με τον χρήστη είναι ένα από τα ζητούμενα σε τέτοιου είδους συστήματα (Cockburn et al., 1997; Zuckerman et al., 2005).
- Ενδιαφέρον παρουσιάζουν τα συστήματα που μπορούν να απεικονίσουν μεταξύ άλλων μια πληθώρα εννοιών και δυναμικών συμπεριφορών, όπως τα συστήματα: SystemBlocks, Flowblocks (Zuckerman et al., 2005; Zuckerman et al., 2003).
- Είναι σημαντικό να προστεθούν στις διεπαφές στοιχεία (σχήμα, χρώμα, υφή, ήχος κ.λπ.) που θα τις φέρουν πιο κοντά στις έννοιες και λειτουργίες που αναπαριστούν (Blackwell 2003; Fishkin 2004).

4. Ερευνητικά ερωτήματα

Γενικά στο χώρο των απτών διαπαφών χρήστη, και ειδικότερα στα συστήματα απτού προγραμματισμού, υπάρχει περιορισμένη έρευνα που να εστιάζει στην κατανόηση των γνωστικών και κοινωνικών πλεονεκτημάτων, τα οποία προκύπτουν από την αξιοποίησή τους (Marshall, 2007; Horn et al., 2007). Η υπόθεση, για παράδειγμα, ότι τα συστήματα απτής διεπαφής εξασφαλίζουν ευκολότερη προσβασιμότητα σε άτομα με αναπηρίες και παιδιά, μειώνοντας ταυτόχρονα το ηλικιακό όριο συμμετοχής στις

διαδικασίες μάθησης, πρέπει να διερευνηθεί. Παράλληλα, πρέπει να εξεταστεί σε ποια ερευνητικά πεδία (προγραμματισμός, αστρονομία, μαθηματικά κ.λπ.) και κάτω από ποιες συνθήκες είναι αποτελεσματικότερη η μάθηση, η οποία στηρίζεται στον χειρισμό απτών αντικειμένων (Marshall, 2007).

Ειδικότερα, στα συστήματα απτού προγραμματισμού, οι πρόσφατες ερευνητικές προσεγγίσεις (π.χ. Horn et al., 2006; Horn et al., 2007), σε αντίθεση με παλαιότερες, θέτουν ως στόχους τη διερεύνηση θεμάτων όπως:

- Πόσο, αν βέβαια συμβαίνει, ένα τέτοιο περιβάλλον μπορεί να διευκολύνει την εκμάθηση προγραμματιστικών εννοιών;
- Βελτιώνεται και κατά πόσο, η συνεργατικότητα μεταξύ των εκπαιδευομένων;
- Μπορεί ένα τέτοιο σύστημα να διευκολύνει τον καθηγητή να αναπτύξει ένα θετικό περιβάλλον μάθησης μέσα στην τάξη;

Παράλληλα, άλλες έρευνες όπως των Kelleher et al. (2005) θέτουν ως στόχους, μεταξύ άλλων, τη δημιουργία περιβαλλόντων που με κατάλληλες δραστηριότητες θα προωθήσουν τον προγραμματισμό σε άτομα και κοινωνικές ομάδες που ‘παραδοσιακά’ δεν σχετίζονται με το αντικείμενο. Άλλους βέβαια ερευνητές, όπως ο Blackwell (2003), πριν ακόμα σχεδιάσουν κάποιο σύστημα, φαίνεται να απασχολεί η διερεύνηση των ιδιοτήτων των απτών διαπαφών, που θα εξασφαλίσουν πλεονεκτήματα, σε σχέση με τις κλασικές text-based ή visual-based γλώσσες, σε ένα σύστημα απτού προγραμματισμού. Η απάντηση τέτοιων ερωτημάτων καθώς και συμπληρωματικές έρευνες ευχρηστίας (όπως Smith, 2006) μπορούν να οδηγήσουν σε αρτιότερα κατασκευαστικά και αποδοτικότερα εκπαιδευτικά συστήματα στο μέλλον.

5. Συμπεράσματα

Στην εργασία αυτή παρουσιάσαμε τα βασικότερα συστήματα απτού προγραμματισμού παραθέτοντας συγκεντρωτικά τα ενδιαφέροντα λειτουργικά χαρακτηριστικά τους. Επίσης, παρουσιάσαμε γενικές υποδείξεις για την κατασκευή τέτοιων περιβαλλόντων και διατυπώσαμε τα θέματα εκείνα που συγκεντρώνουν σήμερα το ερευνητικό ενδιαφέρον. Το γενικό συμπέρασμα είναι ότι η έρευνα σε τέτοια συστήματα (πέρα από την όποια κατασκευαστική και σχεδιαστική καινοτομία) θα πρέπει να διερευνήσει θέματα αποδοτικότητας και εκπαιδευτικής αξίας προκειμένου να καθοδηγήσει τους μελλοντικούς σχεδιαστές αλλά και τους εκπαιδευτικούς στην αξιοποίησή τους.

Βιβλιογραφία

Blackwell, A. (2003). *Cognitive Dimensions of Tangible Programming Languages*. In Proceedings of the First Joint Conference of EASE and PPIG 2003, pp. 391-405.

- Blackwell, A.F., & Hague, R. (2001). *AutoHAN: An Architecture for Programming the Home*. IEEE Symposia on Human-Centric Computing Languages and Environments, 2001, pp. 150-157.
- Cockburn, A., & Bryant, A. (1997). *Leogo: An equal opportunity user interface for programming*. Journal of Visual Languages and Computing, 8(5-6), 601-619.
- Fernaesus, Y., & Tholander, J. (2006). *Finding Design Qualities in a Tangible Programming Space*. In Proc. of CHI, Montréal, Québec, Canada, 22-27 April 2006, pp. 447-456.
- Fishkin, K. P. (2004). *A Taxonomy for and Analysis of Tangible Interfaces*. Personal and Ubiquitous Computing 8(5), 347-358.
- Frei, P., SU, V., Mikhak, B., & Ishii, H. (2000). *Curlybot: Designing a New Class of Computational Toys*. In Proceedings of CHI'00, pp. 129-136.
- Horn, M., & Jacob, R.J.K. (2007). *Tangible Programming in the Classroom with Tern*. CHI Interactivity, San Jose, CA, USA, 28April - 3 May, 2007, pp 1965-1970.
- Horn, M. & Jacob, R.J.K. (2006). *Tangible Programming in the Classroom: A Practical Approach*. Extended Abstracts CHI 2006, ACM Press, pp. 869-874.
- Ichida, H., Itoh, Y., Kitamura, Y., & Kishino, F. (2004). *ActiveCube and its 3D Applications*. IEEE VR2004 Workshop Beyond Wand and Glove Based Interaction, 28 March, Chicago, IL USA.
- Ishii, H., & Ullmer, B. (1997). *Tangible bits: towards seamless interfaces between people, bits, and atoms*. In Proceedings of the CHI'97 conference on human factors in computing systems, Atlanta, Georgia, March 1997, pp 234-241.
- Itoh, Y., Akinobu, S., Ichida, H., Watanabe, R., Kitamura, Y., & Kishino, F. (2004). *TSU.MI.KI: Stimulating Children's Creativity and Imagination with Interactive Blocks*. In proc. to the Second International Conference on Creating, Connecting and Collaborating through Computing (C5) 29 - 30January, 2004, pp. 62-70.
- Kahn, K. (1996). *Drawings on napkins, video-game animation, and other ways to program computers*. Communications ACM, 39(8), 49-59.
- Kelleher, C., & Pausch, R. (2005). *Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers*. ACM Computing Surveys, 37(2), 83-137.
- Kitamura, Y., Itoh, Y., Masaki, T. & Kishino, F. (2000). *ActiveCube: A Bi-directional User Interface using Cubes*. In Proc. of International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies 2000, pp. 99-102.
- Marshall, P. (2007). *Do tangible interfaces enhance learning?*. TEI'07, Baton Rouge, LA, USA, 15-17 Feb 2007, pp.163-170.
- McNerny, T. S. (2001). *Tangible computation bricks: building-blocks for physical microworlds* CHI 2001.
- McNerny, T. S. (2004). *From turtles to Tangible Programming Bricks: Explorations in Physical Language Design*. Personal and Ubiquitous Computing 8(5), 326-37.

- Newton-Dunn, H., Nakano, H., & Gibson, J. (2003). *Block Jam: A Tangible Interface for Interactive Music*. Proceeding of NIME-03, Montreal, Canada, 22-24 May 2003. pp.170-177.
- Patten, J., Griffith, L., & Ishii, H. (2000). *A Tangible Interface for Controlling Robotic Toys*. Presented at CHI '00, 1-6 April 2000. pp. 277-278.
- Rekimoto, J., Ullmer, B., & Oba, H. (2001). *DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions*. In: Proceedings of the CHI 2001 conference on human factors in computing systems, Seattle, Washington, USA, 31 March -4 April 2001, pp 276–283.
- Smith, A.C. (2007). *Using magnets in physical blocks that behave as programming objects*. In Proceedings of the 1st international conference on Tangible and embedded interaction. TEI'07, Baton Rouge, LA, USA, 15-17 Feb 2007, pp. 147 – 150.
- Smith, A.C. (2006). *Tangible cubes as programming objects*. In Proc. ICAT 2006, IEEE Computer Society (2006), pp. 157-161.
- Sutphen, E., Sharlin, B., Watson, J., & Frazer S. (2000). *Reviving a Tangible Interface Affording 3D Spatial Interaction*. Proceedings of 11th Western Canadian Computer Graphics 2000, pp. 155-166.
- Suzuki, H., & Kato, H., (1993). *AlgoBlock: a tangible programming language, a tool for collaborative learning*. In Proc. of 4th European Logo Conference Aug. 1993, Athens Greece, pp. 297-303.
- Wyeth, P., Purchase, H. (2002). *Designing Technology for Children: Moving from the Computer into the Physical World with Electronic Blocks*. Information Technology in Childhood Education Annual 2002 (1), 219-244.
- Wyeth, P., & Purchase, H. C. (2003). *Using Developmental Theories to Inform the Design of Technology for Children*. In proceedings of Interaction Design and Children, ACM Press, pp.93–100.
- Yonemoto, S., Yotsumoto, T., Taniguchi, R. (2007). *Virtual Object Manipulation Using Physical Blocks*. In 11th International Conference Information Visualization (IV'07) IEEE. pp. 781-785.
- Zuckerman, O., Arida, S., & Resnick M. (2005). *Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives*. Proceedings of CHI 2005. ACM Press (2005), pp.859-868.
- Zuckerman, O., Resnick, M. (2003). *System Blocks: A Physical Interface for System Dynamics Simulation*. In Proceedings of CHI '03, ACM Press, pp. 810-811.