

## Themes in Science and Technology Education

Vol 2, No 3 (2009)



**Εκπαιδευτικά περιβάλλοντα οπτικοποίησης αλγορίθμων: Μια επισκόπηση των τεχνικών και παιδαγωγικών χαρακτηριστικών**

*Ευρυπίδης Βραχνός, Αθανάσιος Τζιμογιάννης*

### To cite this article:

Βραχνός Ε., & Τζιμογιάννης Α. (2009). Εκπαιδευτικά περιβάλλοντα οπτικοποίησης αλγορίθμων: Μια επισκόπηση των τεχνικών και παιδαγωγικών χαρακτηριστικών. *Themes in Science and Technology Education*, 2(3), 215–245. Retrieved from <https://ejournals.epublishing.ekt.gr/index.php/thete/article/view/44660>

# Εκπαιδευτικά περιβάλλοντα οπτικοποίησης αλγορίθμων: Μια επισκόπηση των τεχνικών και παιδαγωγικών χαρακτηριστικών

Ευριπίδης Βραχνός και Αθανάσιος Τζιμογιάννης  
evrachnos@gmail.com, ajimoyia@uop.gr  
Τμήμα Κοινωνικής και Εκπαιδευτικής Πολιτικής, Πανεπιστήμιο Πελοποννήσου

## Περίληψη

Η κατανόηση και η χρήση των αλγοριθμικών δομών από αρχάριους στον προγραμματισμό αποτελεί μια ιδιαίτερα δύσκολη και πολύπλοκη διαδικασία. Τα τελευταία χρόνια έχουν αναπτυχθεί πολλά εκπαιδευτικά περιβάλλοντα οπτικοποίησης αλγορίθμων, τα οποία έχουν ως στόχο την ανάδειξη των βασικών αλγοριθμικών χαρακτηριστικών και την υποστήριξη μαθητών και φοιτητών, ώστε να οικοδομήσουν επαρκείς αναπαραστάσεις για τις αλγοριθμικές και τις προγραμματιστικές δομές. Στην εργασία αυτή γίνεται μια επισκόπηση των σημαντικότερων συστημάτων οπτικοποίησης που αναφέρονται στη διεθνή βιβλιογραφία. Η μελέτη επικεντρώνεται στα τεχνικά και παιδαγωγικά χαρακτηριστικά τους, με στόχο την ανάδειξη των βασικών σχεδιαστικών αρχών που πρέπει να διέπουν ένα αποτελεσματικό σύστημα οπτικοποίησης αλγορίθμων για τη διδασκαλία των εισαγωγικών μαθημάτων στο προγραμματισμό.

## Εισαγωγή

Η διδασκαλία του προγραμματισμού και της αλγοριθμικής επίλυσης προβλημάτων σε μαθητές της δευτεροβάθμιας εκπαίδευσης ή φοιτητές, συνιστά ένα δύσκολο αλλά εξαιρετικά ενδιαφέρον έργο, τόσο από διδακτική όσο και από γνωστική πλευρά. Κατά την επίλυση αλγοριθμικών προβλημάτων χρησιμοποιούνται έννοιες και δομές, οι οποίες είναι δύσκολο να οικοδομηθούν με τα παραδοσιακά διδακτικά μέσα (αλγόριθμος, μεταβλητή, δομή επιλογής, δομή επανάληψης, δομή δεδομένων κλπ.). Τα συνηθισμένα προγραμματιστικά περιβάλλοντα και οι γλώσσες προγραμματισμού που χρησιμοποιούνται για εκπαιδευτικούς σκοπούς έχουν σχεδιαστεί για την ανάπτυξη εφαρμογών και όχι για τη διδασκαλία του προγραμματισμού. Είναι, συνεπώς, προσαρμοσμένα στο πλαίσιο γνώσεων και δεξιοτήτων των έμπειρων προγραμματιστών, γεγονός που ενισχύει τις δυσκολίες και τα εμπόδια που συναντούν οι μαθητές και οι αρχάριοι στον προγραμματισμό (Κόμης & Τζιμογιάννης, 2006).

Την τελευταία δεκαετία, αναπτύσσεται μεγάλο ερευνητικό και εκπαιδευτικό ενδιαφέρον για τη χρήση ειδικών περιβαλλόντων προγραμματισμού, όπως περιβάλλοντα Logo (γλώσσες Logo, MicroWolds Pro), περιβάλλοντα εκπαιδευτικής ρομποτικής (π.χ. LEGO/LOGO, JKarelRobot), μικρογλώσσες, προγραμματιστικοί μικρόκοσμοι (π.χ. Karel, Karel++, KarelJ, JEROO), περιβάλλοντα προσομοίωσης και οπτικοποίησης αλγορίθμων, ολοκληρωμένα εκπαιδευτικά προγραμματιστικά περιβάλλοντα κ.λ.π. Τα περιβάλλοντα αυτά παρέχουν νέες δυνατότητες για την οικοδόμηση γνώσεων και την ανάπτυξη δεξιοτήτων στον προγραμματισμό. Παράλληλα, διαπιστώνεται μια έντονη τάση για το σχεδιασμό μαθησιακών δραστηριοτήτων που βασίζονται στις αρχές του εποικοδομισμού (Ramadhan, 2000; Kolikant & Pollack, 2004; Γρηγοριάδου κ.α., 2004; Κόμης, 2005; Τσέλιος κ.α., 2006).

Κύριοι άξονες των εποικοδομιστικών διδακτικών προσεγγίσεων είναι η εκτίμηση των προϋπαρχουσών γνώσεων και αντιλήψεων των μαθητών και η οργάνωση διδακτικών-μαθησιακών δραστηριοτήτων που να ευνοούν τη *διερευνητική, ανακαλυπτική και συνεργατική* μάθηση. Οι προσεγγίσεις αυτές δίνουν έμφαση στον παιδαγωγικό σχεδιασμό της διδασκαλίας του προγραμματισμού και στη μετατόπιση από το συντακτικό στην καλλιέργεια δεξιοτήτων επίλυσης προβλημάτων (αναλυτική-συνθετική σκέψη, αφαιρετική ικανότητα, μοντελοποίηση λύσεων). Στο πλαίσιο αυτό, έχει προταθεί ο σχεδιασμός κατάλληλων μαθησιακών δραστηριοτήτων με χρήση εκπαιδευτικών περιβαλλόντων προσομοίωσης-οπτικοποίησης αλγορίθμων και προγραμμάτων (Stasko, 1997; Hundhausen & Douglas, 2002; Sajaniemi & Kuittinen, 2003; Vrachnos & Jimoyiannis, 2008).

Μια εφαρμογή οπτικοποίησης στοχεύει στη σωστή *αναπαραγωγή της εκτέλεσης* ενός αλγορίθμου και στον *πειραματισμό* του μαθητή, ώστε να ανακαλύψει σημαντικά χαρακτηριστικά της συμπεριφοράς του υπό μελέτη αλγορίθμου. Επίσης, μπορεί να αποτελέσει ένα εργαλείο επιβεβαίωσης της ορθότητας του αλγορίθμου που έχει σχεδιάσει ο μαθητής, μέσα από την δυναμική οπτικοποίησή του.

Ένας σημαντικός αριθμός περιβαλλόντων οπτικοποίησης αλγορίθμων έχουν αναπτυχθεί και είναι διαθέσιμα για εκπαιδευτική χρήση στα εισαγωγικά μαθήματα προγραμματισμού, κυρίως σε πανεπιστημιακό επίπεδο (Moreno et al., 2004; Cross & Hendrix, 2006; Hundhausen & Brown, 2007). Οι εφαρμογές αυτές έχουν σημαντικές διαφορές στον τρόπο υλοποίησης τους, και γενικά χαρακτηρίζονται από μεγάλη ετερογένεια. Οι διαφορές εντοπίζονται στον τρόπο παρουσίασης, στο βαθμό αλληλεπίδρασης του μαθητή με το περιβάλλον, στον τρόπο ανάπτυξης της οπτικοποίησης και σε άλλα σημεία.

Στις επόμενες ενότητες παρουσιάζονται τα σημαντικότερα εργαλεία οπτικοποίησης αλγορίθμων που χρησιμοποιούνται σήμερα διεθνώς. Αναφέρονται επίσης, κάποια παλαιότερα συστήματα τα οποία είχαν σημαντική συμβολή στην ανάπτυξη του συγκεκριμένου χώρου. Για την κατάταξη των συστημάτων χρησιμοποιείται η ταξινόμια του Price (1993) σε συνδυασμό με την ταξινόμια των επιπέδων διαδραστικότητας (αλληλεπίδρασης) που έχει προταθεί από τους Naps et al. (2003).

Οι στόχοι της παρούσας εργασίας είναι

- Η συγκριτική παρουσίαση και ταξινόμηση των διαθέσιμων εκπαιδευτικών περιβαλλόντων οπτικοποίησης, σε τρόπο ώστε να διακρίνονται και να αναδεικνύονται τα τεχνολογικά και τα παιδαγωγικά χαρακτηριστικά τους.
- Η περιγραφή των βασικών χαρακτηριστικών και προδιαγραφών που πρέπει να έχει ένα ιδανικό σύστημα οπτικοποίησης αλγορίθμων, ώστε να είναι αποτελεσματικό στην εκπαιδευτική πρακτική.

## Θεμελιώδεις έννοιες

Με τον όρο *οπτικοποίηση (visualization)* αλγορίθμου περιγράφεται μια διαδραστική οπτικοποιημένη παρουσίαση της λογικής του αλγορίθμου, βασισμένη σε μια σειρά εικόνων και αναπαραστάσεων που αναδεικνύουν τα βασικά χαρακτηριστικά της συμπεριφοράς του (Hundhausen & Brown, 2007). Τα συνήθη λογισμικά προσομοίωσης-οπτικοποίησης δεν αντικαθιστούν το προγραμματιστικό περιβάλλον, αφού δεν εκτελούν κάποιον αλγόριθμο, αλλά παρουσιάζουν, οπτικοποιούν και προσομοιώνουν την εκτέλεσή του για προκαθορισμένα δεδομένα εισόδου.

Υπάρχουν πολλοί λόγοι που συνηγορούν στη χρήση κατάλληλα σχεδιασμένων περιβαλλόντων προσομοίωσης-οπτικοποίησης στα εισαγωγικά μαθήματα του προγραμματισμού. Οι προσομοιώσεις αλγορίθμων αποτελούν δυναμικά μαθησιακά περιβάλλοντα τα οποία (Τζιμογιάννης κ.α., 2006):

- επιτρέπουν την οπτικοποίηση της λειτουργίας ενός αλγορίθμου, υποστηρίζοντας τη διάλεξη του εκπαιδευτικού και την εργαστηριακή εξάσκηση των μαθητών
- επιτρέπουν στους μαθητές να απομονώσουν τις μεταβλητές και το ρόλο τους στο πρόγραμμα, με στόχο την κατανόηση σύνθετων υπολογιστικών δομών και διαδικασιών
- βοηθούν τους μαθητές να εμβαθύνουν στη λογική του προγραμματισμού και να κατανοήσουν δύσκολες υπολογιστικές έννοιες και διαδικασίες
- ενεργοποιούν το ενδιαφέρον των μαθητών, παρέχοντας δυνατότητες να εκφράσουν τις δικές τους αναπαραστάσεις για αλγορίθμους και διαδικασίες
- διευκολύνουν την ενεργητική μάθηση μέσα από διαδικασίες παράθεσης υποθέσεων, μεταβολής των τιμών εισόδου και άμεσου ελέγχου των αποτελεσμάτων στην οθόνη
- βοηθούν τους μαθητές να μάθουν μέσα από διερευνητικές δραστηριότητες.

Τα λογισμικά προσομοίωσης αλγορίθμων θα πρέπει να προωθούν τη διερευνητική μάθηση και να ευνοούν την ενεργητική συμμετοχή και όχι την παθητική 'συμμόρφωση' των μαθητών (Hundhausen et al., 2002; Hundhausen & Brown, 2007). Ένα

σύγχρονο εκπαιδευτικό περιβάλλον οπτικοποίησης θα πρέπει να παρέχει στο μαθητή δυνατότητες να πειραματιστεί με την εκτέλεση του αλγορίθμου, να αλλάξει παραμέτρους, να ελέγξει ιδέες και αναπαραστάσεις, έτσι ώστε να ανακαλύψει τις διάφορες πτυχές του και να οικοδομήσει επαρκείς και λειτουργικές αναπαραστάσεις με στόχο την εφαρμογή τους για την επίλυση νέων προβλημάτων.

Σε κάθε εκπαιδευτική προσομοίωση διακρίνονται τρεις βασικές συνιστώσες (Jimoyiannis, 2008):

- το σενάριο της προσομοίωσης (simulation scenario)
- το υποκείμενο μοντέλο του συστήματος (model)
- η διδακτική συνιστώσα (instructional overlay), που καθορίζεται από τα αναπαραστατικά εργαλεία, την παιδαγωγική φιλοσοφία και τις χρησιμοποιούμενες δραστηριότητες.

Οι παραπάνω συνιστώσες ορίζουν τον τρόπο με τον οποίο προσεγγίζεται και υποστηρίζεται η νέα γνώση από το εργαλείο προσομοίωσης.

Στη συνέχεια παρουσιάζονται τα σημαντικότερα εκπαιδευτικά περιβάλλοντα οπτικοποίησης στον προγραμματισμό, τα οποία διακρίνονται σε δύο κατηγορίες με βάση τα χαρακτηριστικά και τις δομές του αλγορίθμου που οπτικοποιούν: α) συστήματα οπτικοποίησης προγραμμάτων και β) συστήματα οπτικοποίησης αλγορίθμων.

## Συστήματα Οπτικοποίησης Προγραμμάτων

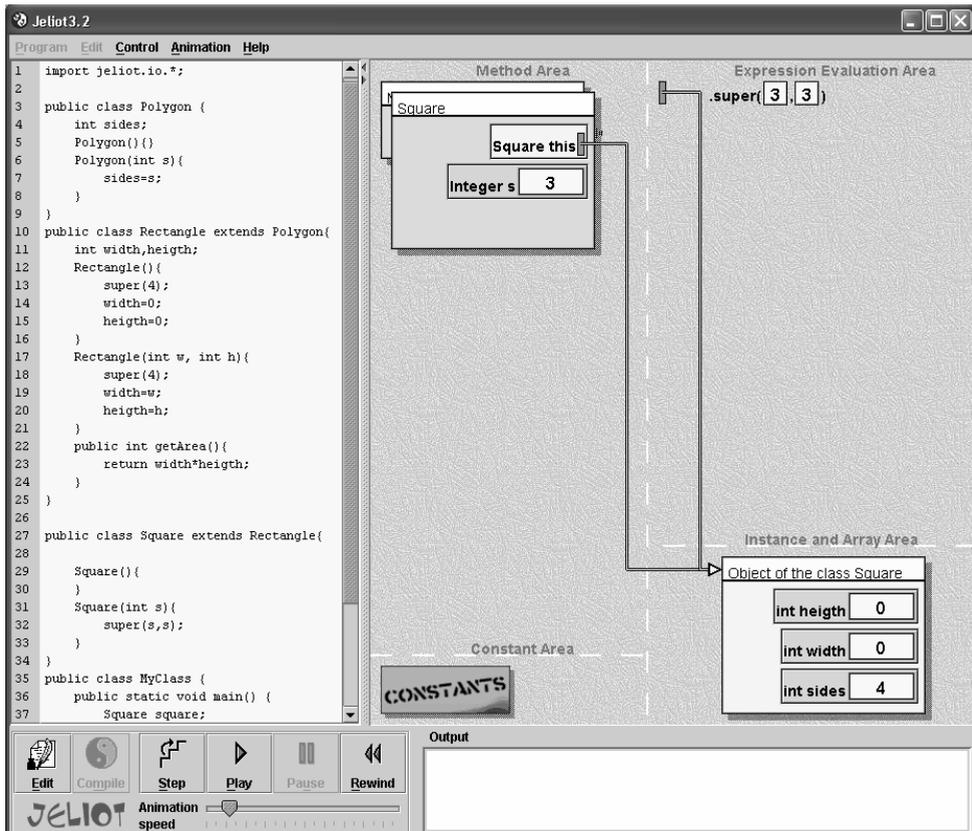
Τα *συστήματα οπτικοποίησης προγραμμάτων* αναπαριστούν γραφικά βασικές προγραμματιστικές δομές, όπως οι μεταβλητές, οι δείκτες ή οι πίνακες που είναι χρήσιμες στους προγραμματιστές, ειδικά κατά την εκσφαλμάτωση (debugging) των προγραμμάτων. Μέσω της οπτικοποίησης γίνεται παρακολούθηση των τιμών των μεταβλητών σε κάθε βήμα του αλγορίθμου, με στόχο την ανίχνευση και τη διόρθωση των λαθών του προγράμματος.

### *Το περιβάλλον αντικειμενοστρεφούς προγραμματισμού Jeliot*

Χαρακτηριστικός εκπρόσωπος της κατηγορίας αυτής είναι το σύστημα Jeliot που έχει ως στόχο την οπτικοποίηση προγραμμάτων που είναι γραμμένα στην γλώσσα προγραμματισμού Java. Ουσιαστικά πρόκειται για μια οικογένεια εφαρμογών, πρώτο μέλος της οποίας ήταν το Jeliot 2000 και τελευταίο μέλος το Jeliot 3 (Moreno et al., 2004). Η δυνατότητα που χαρακτηρίζει το Jeliot είναι η αυτόματη οπτικοποίηση των προγραμμάτων που είναι γραμμένα στη γλώσσα προγραμματισμού Java. Στο Σχήμα 1 απεικονίζεται μια οθόνη του περιβάλλοντος. Στο αριστερό τμήμα εμφανίζεται η τρέχουσα κατάσταση του προγράμματος (μεταβλητές, αντικείμενα και μέθοδοι) και στο δεξιό η δυναμική οπτικοποίηση του υπολογισμού σύνθετων εκφράσεων και των δομών επανάληψης και επιλογής. Κάτω αριστερά βρίσκονται τα πλήκτρα ελέγχου

της εκτέλεσης του προγράμματος. Η εφαρμογή εστιάζει στην οπτικοποίηση των βασικών αντικειμενοστρεφών δομών (κλάσεις, αντικείμενα, μέθοδοι κ.λπ.). Η τελευταία έκδοση του Jeliot μπορεί να προσαρτηθεί στο εκπαιδευτικό περιβάλλον BlueJ για Java (Kölling et al., 2003). Σε αυτή την περίπτωση κάποιος μπορεί να χρησιμοποιεί το περιβάλλον BlueJ και να βλέπει την οπτικοποίηση του προγράμματός του αυτόματα μέσω του Jeliot. Από μια επισκόπηση των ερευνών που έγιναν σχετικά με την εκπαιδευτική αξία του (Ben-Bassat Levy & Ben-Ari, 2009), προέκυψε ότι το Jeliot

- α) βελτιώνει τη μάθηση στον προγραμματισμό παρέχοντας στέρεες αναπαραστάσεις των δυναμικών πτυχών ενός προγράμματος,
- β) εστιάζει την προσοχή των μαθητών στις βασικές αντικειμενοστρεφείς δομές και
- γ) επηρεάζει το επίπεδο εμπλοκής των μαθητών με το λογισμικό, ειδικά αν αυτό χρησιμοποιηθεί σε ένα συνεργατικό πλαίσιο εφαρμογής.

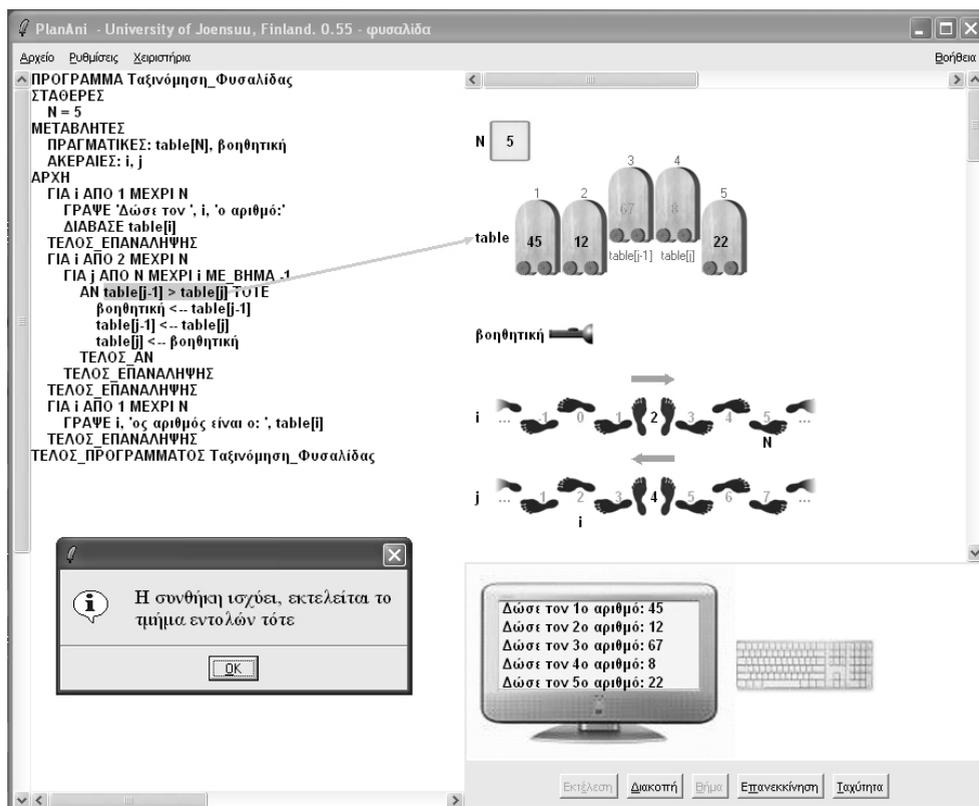


Σχήμα 1. Μέθοδοι και αντικείμενα στο Jeliot 3.2

## Το λογισμικό PlanAni

Το PlanAni αποτελεί ένα περιβάλλον προσομοίωσης της εκτέλεσης προγραμμάτων, το οποίο βασίζεται στην προσέγγιση της διδασκαλίας με ρόλους μεταβλητών (Sajaniemi & Kuittinen, 2003). Το λογισμικό αυτό χρησιμοποιεί διαφορετική αναπαράσταση για κάθε μεταβλητή ανάλογα με το ρόλο της στο πρόγραμμα. Ο μαθητής έχει τη δυνατότητα να παρακολουθεί την προσομοίωση της εκτέλεσης του προγράμματος ορίζοντας τα δεδομένα εισόδου αλλά είναι δύσκολο να κατασκευάσει την οπτικοποίηση του δικού του αλγορίθμου, γιατί θα πρέπει να προσθέσει τις κατάλληλες εντολές οπτικοποίησης ρόλων.

Στο Σχήμα 2 δίνεται μια οθόνη της ελληνικής έκδοσης του PlanAni με την οπτικοποίηση του αλγορίθμου ταξινόμησης της φουσαλίδας σε ψευδογλώσσα (Τζιμογιάννης κ.α., 2006).

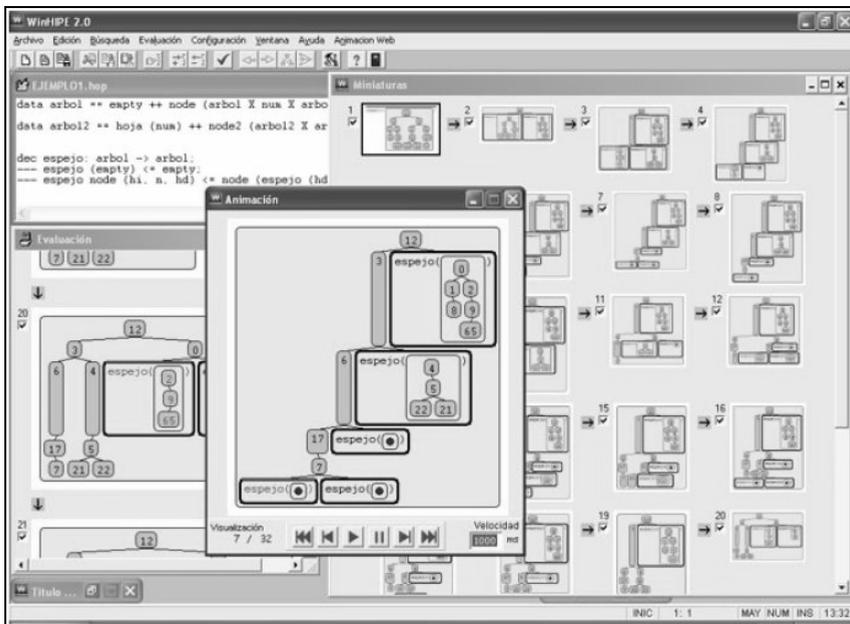


Σχήμα 2. Ταξινόμηση φουσαλίδας με την ελληνική έκδοση του PlanAni

## Τα Περιβάλλοντα WinHIPE και SRec

Το WinHIPE (Pareja-Flores et.al., 2007) δέχεται προγράμματα γραμμένα στην συναρτησιακή γλώσσα προγραμματισμού Hope, τα οποία οπτικοποιεί με σκοπό να αναδείξει τα ιδιαίτερα χαρακτηριστικά του συναρτησιακού προγραμματισμού, όπως είναι το μοντέλο αποτίμησης, οι λίστες και η αναδρομή. Το γραφικό περιβάλλον (διεπαφή) επιτρέπει τη σταδιακή εκτέλεση ή την παύση της οπτικοποίησης αλλά και την αντίστροφη εκτέλεση, μέσω ενός οπτικού (VCR-like) τηλεχειριστηρίου (Σχήμα 3). Η μόνη δυνατότητα που λείπει είναι η εκτέλεση της οπτικοποίησης βήμα-βήμα, δίπλα στον κώδικα του μαθητή, με παράλληλη επισήμανση της εντολής που εκτελείται.

Η ίδια ερευνητική ομάδα έχει αναπτύξει και το περιβάλλον οπτικοποίησης SRec (Velasquez Iturbide et al., 2008). Το σύστημα αυτό οπτικοποιεί την εκτέλεση αναδρομικών αλγορίθμων που έχουν σχεδιαστεί με την τεχνική *Διαίρει και Βασίλευε* και είναι κωδικοποιημένοι στη γλώσσα προγραμματισμού Java. Η οπτικοποίηση του αλγορίθμου παράγεται πολύ εύκολα από το χρήστη, ο οποίος πρέπει απλά να καθορίσει την *Διαίρει και Βασίλευε* αναδρομική δομή του αλγορίθμου. Δηλαδή η οπτικοποίηση δεν είναι πλήρως αυτοματοποιημένη και χρειάζεται κάποια παραμετροποίηση από την πλευρά του χρήστη, ώστε να γίνει σωστά η γραφική παρουσίαση του δέντρου των αναδρομικών κλήσεων.



Σχήμα 3. Οπτικοποίηση της αναδρομής με το WinHIPE

## ***Το περιβάλλον JIVE***

Το JIVE (Java Interactive Visualization Environment) είναι ένα περιβάλλον οπτικοποίησης των αντικειμενοστρεφών δομών δεδομένων για προγράμματα που είναι γραμμένα σε Java (Gestwicki & Jayaraman, 2002). Περιλαμβάνει τα εξής χαρακτηριστικά:

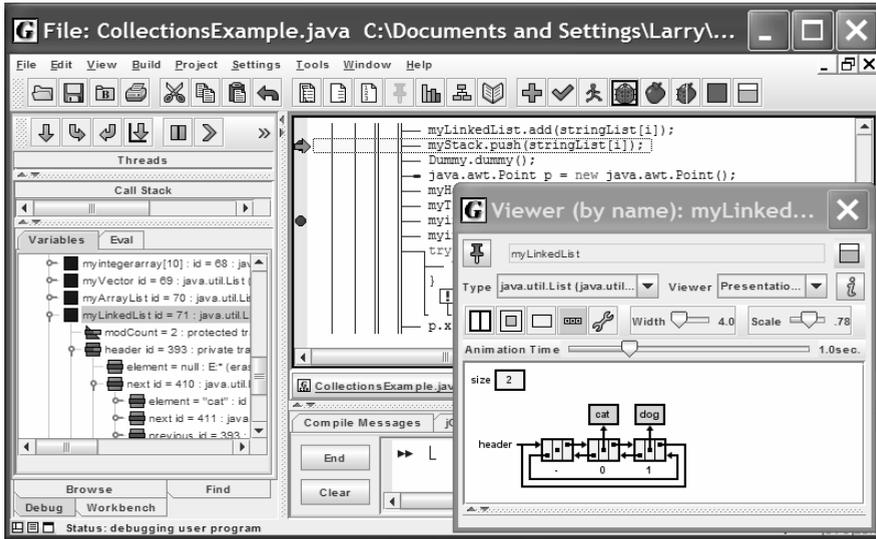
- Μια γλώσσα για την περιγραφή των καταστάσεων ενός προγράμματος Java κατά την εκτέλεση.
- Υψηλό βαθμό αλληλεπίδρασης με τον χρήστη κατά την εκτέλεση
- Αντίστροφη εκτέλεση (μετάβαση σε προηγούμενη κατάσταση)
- Αυτόματη κατασκευή διαγραμμάτων αντικειμένων και ακολουθίας (object and sequence diagrams), όπως στη γλώσσα UML.

Ένα σημαντικό χαρακτηριστικό είναι η δυνατότητα που έχει ο χρήστης να δει το χρονογράφημα όλης της εκτέλεσης (execution history), μέσα από τα διαγράμματα ακολουθίας της UML, όπου φαίνονται όλες οι καταστάσεις της εκτέλεσης του προγράμματος. Ο χρήστης μπορεί να επιλέξει μια από αυτές και να δει την οπτικοποίηση του διαγράμματος των αντικειμένων (UML).

## ***Το περιβάλλον προγραμματισμού jGrasp***

Το jGrasp είναι ένα εκπαιδευτικό περιβάλλον προγραμματισμού με σκοπό την εκμάθηση βασικών εννοιών του αντικειμενοστρεφούς προγραμματισμού, όπως είναι η κλάση, το αντικείμενο, η κληρονομικότητα και ο πολυμορφισμός (Cross & Hendrix, 2006). Παρέχει τη δυνατότητα της αυτόματης οπτικοποίησης των περισσότερων δομών δεδομένων της Java, όπως είναι οι λίστες, οι πίνακες και οι συλλογές (collections), με στόχο την καλύτερη εκσφαλμάτωση των προγραμμάτων. Δεν πρόκειται για δυναμική οπτικοποίηση με κίνηση των αντικειμένων, αλλά απλά για ενημέρωση των περιεχομένων των αντίστοιχων δομών. Ωστόσο οι γραφικές αναπαραστάσεις των δομών δεδομένων συγκλίνουν στη νοερή δομή που έχουμε για αυτές, όπως για παράδειγμα η δομή της λίστας (Σχήμα 4).

Το λογισμικό χρησιμοποιείται σε περισσότερα από 300 εκπαιδευτικά ιδρύματα για τη διδασκαλία της γλώσσας προγραμματισμού Java. Υποστηρίζει πολλαπλές όψεις του ίδιου τμήματος κώδικα, όπως είναι το UML διάγραμμα και ο πάγκος εργασίας (workbench) για τα αντικείμενα που δημιουργούνται κατά την εκτέλεση. Έχει τη δυνατότητα εκτέλεσης μεμονωμένων εντολών με ταυτόχρονη ενημέρωση όλων των όψεων που έχει ανοίξει ο χρήστης, όπως ο πάγκος εργασίας (workbench) και η οπτικοποίηση της δομής στη μνήμη (viewer).



Σχήμα 4. Οπτικοποίηση μιας διπλά συνδεδεμένης λίστας σε Java μέσω jGrasp

## Συστήματα Οπτικοποίησης Αλγορίθμων

Τα συστήματα οπτικοποίησης αλγορίθμων είναι περιβάλλοντα που παράγουν μια γραφική παρουσίαση της εκτέλεσης του αλγορίθμου με σκοπό να αναδειχθούν τα ιδιαίτερα χαρακτηριστικά του, τα οποία είναι δύσκολο να εντοπίσουν οι μαθητές (Hundhausen & Brown, 2007, Rössling & Freisleben, 2002).

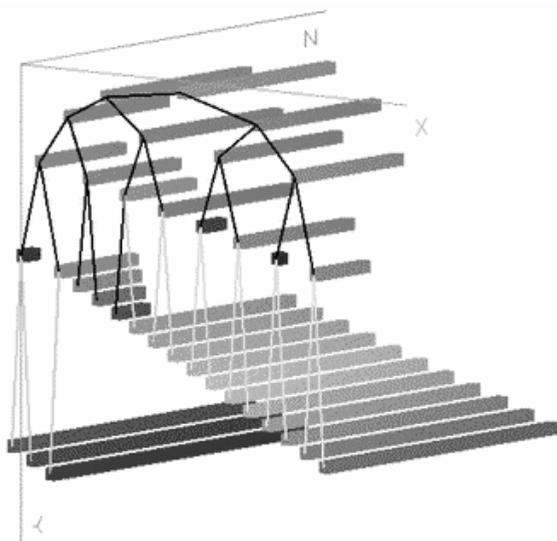
Η πρώτη σημαντική αναφορά οπτικοποίησης αλγορίθμων είναι το γνωστό πλέον βίντεο *Sorting Out Sorting* (Baecker, 1981), το οποίο θεωρείται η πρώτη σημαντική αναφορά στην οπτικοποίηση αλγορίθμων. Παρουσιάζει την οπτικοποίηση διάφορων αλγορίθμων ταξινόμησης πινάκων με προκαθορισμένα δεδομένα ώστε να κατανοήσουν οι σπουδαστές τη λειτουργία τους. Από τότε έχουν αναπτυχθεί πολλά συστήματα οπτικοποίησης αλγορίθμων με σημαντικές εφαρμογές στην εκπαιδευτική πρακτική (Brown, 1991; Rössling & Freisleben, 2002; Hundhausen & Brown, 2007). Στην παράγραφο αυτή θα παρουσιαστούν τα πιο ενδιαφέροντα (σε τεχνολογικό και παιδαγωγικό επίπεδο) και δημοφιλή συστήματα οπτικοποίησης αλγορίθμων.

### Το σύστημα Zeus

Το σύστημα Zeus (Brown, 1991) προέρχεται από την οικογένεια συστημάτων BALSΑ και BALSΑ II. Το σύστημα BALSΑ (Brown Algorithm Simulator and Animator) θεωρείται ως ο πρόγονος των σύγχρονων εφαρμογών οπτικοποίησης αλγορίθμων. Με το σύστημα αυτό ο Brown εισήγαγε τη έννοια του *ενδιαφέροντος γεγονότος* (*interesting event*) κατά την εκτέλεση ενός αλγορίθμου. Ένα ενδιαφέρον γεγονός μπορεί να είναι η

αλλαγή της τιμής μιας μεταβλητής ή η αφαίρεση ενός αντικειμένου από τη στοίβα. Με κατάλληλη επιλογή των ενδιαφερόντων γεγονότων ενός αλγορίθμου, επιλέγουμε ουσιαστικά τις πτυχές του αλγορίθμου που έχουν διδακτικό-μαθησιακό ενδιαφέρον ή δυσκολία και πρέπει να οπτικοποιηθούν. Έτσι μπορούμε να παρακολουθούμε πολλαπλές όψεις της εκτέλεσης του αλγορίθμου να εξελίσσονται ταυτόχρονα.

Για παράδειγμα μια δημοφιλής όψη ενός αλγορίθμου ταξινόμησης, παρουσιάζει τα στοιχεία του πίνακα προς ταξινόμηση σαν ένα σύνολο ιστογραμμάτων (Σχήμα 5). Το ύψος κάθε ιστογράμματος είναι ανάλογο με την τιμή του στοιχείου του πίνακα που παριστάνει.



Σχήμα 5. Αναπαράσταση της ταξινόμησης HeapSort στο σύστημα Zeus

Το Zeus είναι το πιο σύγχρονο σύστημα αυτής της κατηγορίας το οποίο μάλιστα δίνει τη δυνατότητα χρήσης τρισδιάστατων γραφικών. Το μειονέκτημα αυτής της οικογένειας συστημάτων είναι ότι η πολυπλοκότητα της κατασκευής μια νέας οπτικοποίησης, καθιστά αυτή τη δυνατότητα αποτρεπτική όχι μόνο για τους μαθητές αλλά και για τους εκπαιδευτικούς.

### ***Η οικογένεια συστημάτων οπτικοποίησης Tango, Polka, Samba***

Μια πολύ σημαντική οικογένεια εφαρμογών για το χώρο των συστημάτων οπτικοποίησης αλγορίθμων ξεκίνησε με τη δημιουργία του συστήματος Tango (Stasko, 1990). Το χαρακτηριστικό αυτού του συστήματος είναι η τεχνική της πορείας μετάβασης (path transition) μεταξύ των καταστάσεων. Εκεί οφείλεται και το όνομα του συστήματος (**T**ransition-**A**nimation **G**eneration). Δηλαδή η μετάβαση των

προγραμματιστικών αντικειμένων (μεταβλητές) σε μια άλλη κατάσταση παρουσιάζεται στην οθόνη με τη βαθμιαία κίνηση των αντικειμένων.

Ο αλγόριθμος που θα οπτικοποιηθεί πρέπει να είναι γραμμένος στη γλώσσα C. Ο χρήστης καθορίζει τα ενδιαφέροντα γεγονότα που θέλει να οπτικοποιήσει, εισάγοντας εντολές οπτικοποίησης στα κατάλληλα σημεία του αλγορίθμου. Μετά το Tango ακολούθησε το XTango μια έκδοση με φιλική γραφική διεπαφή για το περιβάλλον των X-Windows στο Unix. Ωστόσο, οι δυναμικές παρουσιάσεις που μπορούσαν να δημιουργηθούν με τα εργαλεία αυτά ήταν σχετικά απλές, χωρίς πολλαπλές μεταβάσεις. Δεν υποστήριζαν δηλαδή την εκτέλεση παράλληλων διεργασιών. Για το σκοπό αυτό αναπτύχθηκε λίγο αργότερα το σύστημα POLKA (**P**arallel program-focused **O**bject-oriented **L**ow **K**ey **A**nimation) (Stasko & Kraemer, 1993).

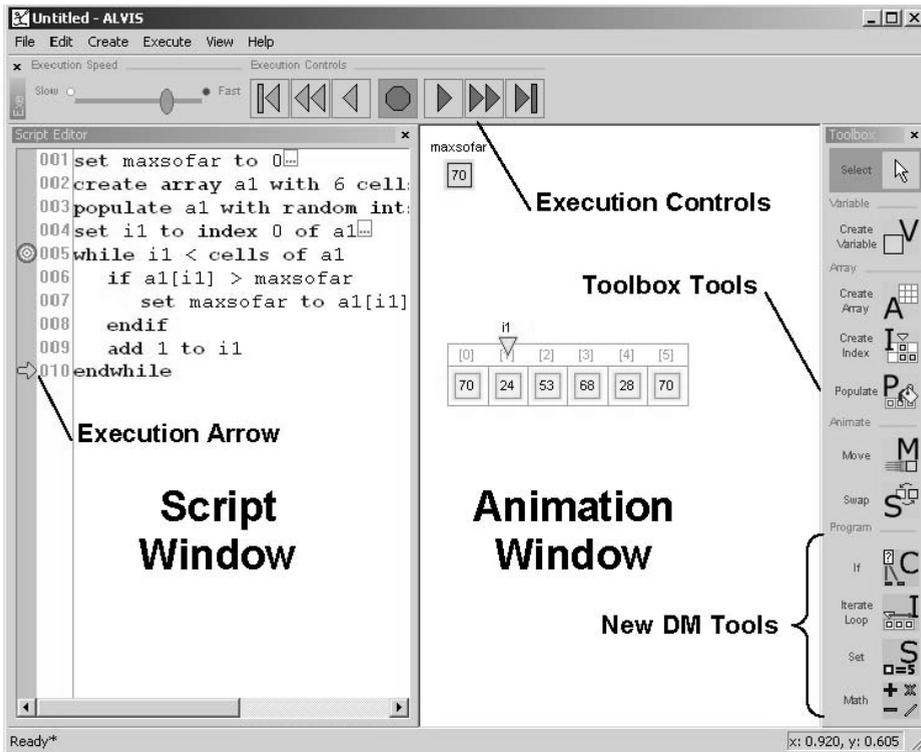
Ένα από τα τελευταία συστήματα που ανέπτυξε η ομάδα του Stasko, είναι το Samba (1998). Πρόκειται ουσιαστικά για μια γραφική διεπαφή (front-end) και ένα διερμηνευτή, ο οποίος τροφοδοτεί με εντολές οπτικοποίησης το σύστημα Polka. Για να δημιουργήσουν οι μαθητές τις δικές τους οπτικοποιήσεις, θα πρέπει να μάθουν να χρησιμοποιούν την ειδική γλώσσα του Samba, κάτι το οποίο είναι πολύ δύσκολο για όσους παρακολουθούν ένα εισαγωγικό μάθημα αλγορίθμων.

Επειδή το Samba δεν μπορεί να λειτουργήσει σε όλες τις πλατφόρμες, αναπτύχθηκε μια νέα έκδοσή του γραμμένη σε Java, το JSamba, το οποίο είναι προσπελάσιμο από το Διαδίκτυο (JSamba 1998). Ωστόσο δε μπορεί ο χρήστης να ορίσει τη δική του οπτικοποίηση, αλλά να δει μόνο τα έτοιμα παραδείγματα που προσφέρονται. Λειτουργεί δηλαδή, σαν ένα περιβάλλον παρουσίασης οπτικοποιήσεων που είναι ήδη γραμμένες στη γλώσσα σεναρίων του Samba.

## Alvis

Η εφαρμογή Alvis Live! (**A**lgorithm **V**isualization **S**toryboarder) (Hundhausen & Brown, 2007) είναι ένα διαδραστικό σύστημα οπτικοποίησης αλγορίθμων, το οποίο αποτελεί μετεξέλιξη του συστήματος οπτικοποίησης Alvis (Hundhausen & Douglas, 2002). Το Alvis εκτελεί έναν αλγόριθμο για προκαθορισμένες εισόδους ή για μικρό πλήθος δεδομένων, χωρίς να χρησιμοποιεί υψηλής ποιότητας γραφικά.

Ενώ το Alvis χρησιμοποιούσε αρχικά τη γλώσσα συγγραφής σεναρίων SALSA (Spatial Algorithmic Language for Storyboarding) για την περιγραφή του αλγορίθμου και της οπτικοποίησής του, το Alvis Live! υποστηρίζει στην τελευταία έκδοσή του τη γλώσσα Cb. Πρόκειται για ένα υποσύνολο της γλώσσας C εφοδιασμένο με κάποια στοιχεία της γλώσσας σεναρίων SALSA που χρησιμοποιούσε η προηγούμενη έκδοση. Στο Σχήμα 6 φαίνεται μια οθόνη του Alvis. Ο μαθητής θα πρέπει να κωδικοποιήσει τον αλγόριθμό του στη γλώσσα Cb (script window). Το σύστημα υποστηρίζει τη δυναμική οπτικοποίηση του αλγορίθμου (animation window) με παράλληλη επισήμανση της τρέχουσας εντολής που εκτελείται.



Σχήμα 6. Το περιβάλλον Alvis Live!

Ένα άλλο χαρακτηριστικό του Alvis είναι ο υψηλός βαθμός αλληλεπίδρασης με το χρήστη, δηλαδή η δυνατότητα σταδιακής (βήμα προς βήμα) και αντίστροφης εκτέλεσης του αλγορίθμου (σε προηγούμενη κατάσταση).

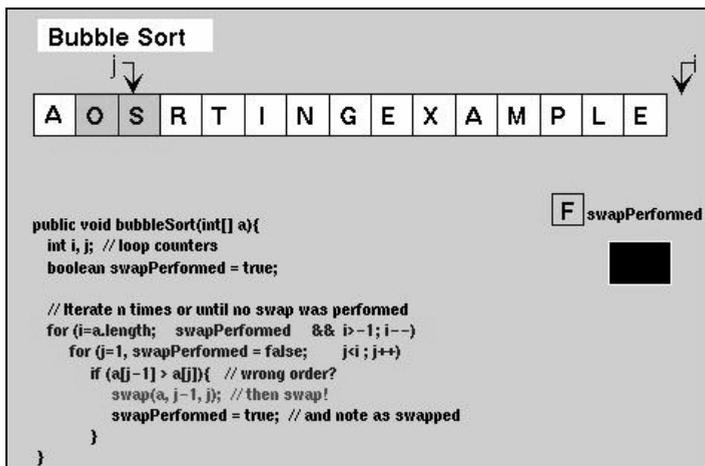
### Animal

Το Animal (Rössling & Freisleben, 2002) είναι ένα ισχυρό εργαλείο δημιουργίας οπτικοποιήσεων αλγορίθμων. Περιλαμβάνει έναν γραφικό συντάκτη οπτικοποιήσεων, μια γλώσσα συγγραφής σεναρίων οπτικοποίησης και μια βιβλιοθήκη γραφικών (Σχήμα 7). Ο γραφικός συντάκτης χρησιμοποιείται όπως σε μια γλώσσα οπτικού προγραμματισμού. Ο χρήστης καλείται να σχεδιάσει οπτικά το περιβάλλον και να καθορίσει τα βασικά χαρακτηριστικά της οπτικοποίησης. Σε κάποιες περιπτώσεις, μπορεί να ορίσει πλήρως την οπτικοποίηση με γραφικό τρόπο. Στις περισσότερες όμως θα χρειαστεί να γράψει κώδικα, ώστε να συνδέσει τον αλγόριθμο με την οπτικοποίηση. Το Animal προσφέρει δυο τρόπους για να γίνει η οπτικοποίηση:

- α) με τη χρήση μιας γλώσσας σεναρίων που ονομάζεται AnimalScript. Η γλώσσα αυτή υποστηρίζει και δομές δεδομένων, όπως οι πίνακες και οι λίστες.

β) με τη χρήση της βιβλιοθήκης που παρέχει το εργαλείο. Πρόκειται για μια βιβλιοθήκη σε Java (Java API). Αφού υλοποιήσει κάποιος τον αλγόριθμο (σε Java), στη συνέχεια προσθέτει στον κώδικα του αλγορίθμου τις κατάλληλες κλήσεις συναρτήσεων της βιβλιοθήκης, που έχουν ως αποτέλεσμα την παραγωγή κώδικα στη γλώσσα AnimalScript, ο οποίος στη συνέχεια θα εκτελεστεί από το Animal. Πρόσφατα παρουσιάστηκε και ένα πρόσθετο (Animalipse plugin) της γλώσσας AnimalScript για το περιβάλλον Eclipse (Rössling & Schroeder, 2009).

Επειδή οι οπτικοποιήσεις που παράγονται είναι γραμμένες σε Java μπορούν να εκτελεστούν σε οποιοδήποτε υπολογιστή και λειτουργικό σύστημα, δηλαδή είναι ανεξάρτητες πλατφόρμας. Το σημαντικό όμως μειονέκτημα είναι ότι δεν απευθύνεται τόσο σε μαθητές/φοιτητές όσο σε διδάσκοντες, αφού για να μπορεί να τη χρησιμοποιήσει κάποιος θα πρέπει ήδη να γνωρίζει προγραμματισμό σε Java, μαζί με τη γλώσσα AnimalScript. Υπάρχει φυσικά ο γραφικός συντάκτης, ο οποίος όμως δεν έχει την πληρότητα και τη δύναμη έκφρασης που έχει η γλώσσα σεναρίων οπτικοποίησης AnimalScript.



Σχήμα 7. Η ταξινόμηση της φυσαλίδας στο Animal

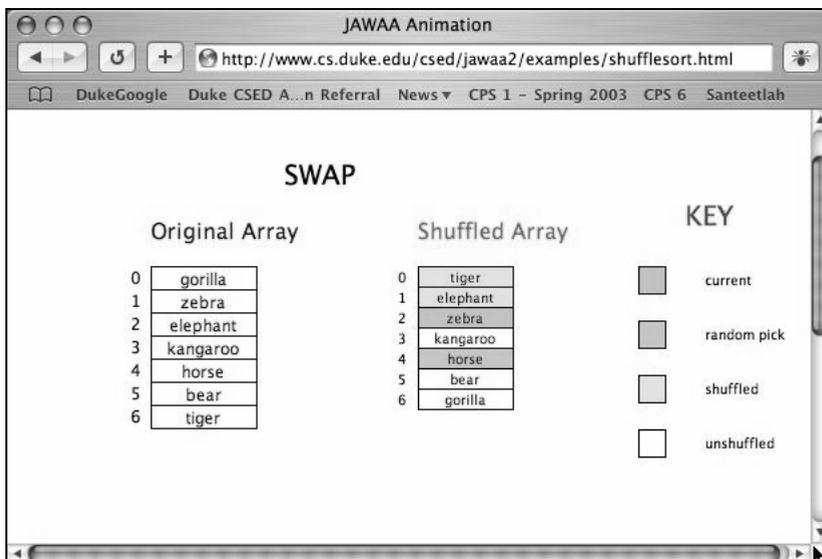
## JAWAA

Το JAWAA (*Java and Web-based Algorithm Animation*) είναι ένα σύστημα οπτικοποίησης αλγορίθμων και δομών δεδομένων που αποτελείται από τρία τμήματα (Pierson & Rodger, 1998):

- α) τη γλώσσα περιγραφής οπτικοποιήσεων,
- β) το γραφικό συντάκτη και
- γ) ένα applet, το οποίο εκτελεί τις οπτικοποιήσεις (Σχήμα 8). Μοιάζει σε αρκετά σημεία με το JSamba και το Animal αφού ο χρήστης μπορεί να κατασκευάσει

μια οπτικοποίηση είτε με τη βοήθεια της γραφικής διεπαφής ή με την ειδική γλώσσα σεναρίων του συστήματος.

Το μεγάλο πλεονέκτημα είναι η ανεξαρτησία πλατφόρμας, αφού το σύστημα είναι γραμμένο στη γλώσσα προγραμματισμού Java και η πρόσβαση σε αυτό μέσω του Παγκόσμιου Ιστού γίνεται εύκολα και γρήγορα, αφού αρκεί να αναρτήσουμε το applet στην ιστοσελίδα που θέλουμε. Το σύστημα υποστηρίζει την οπτικοποίηση δομών δεδομένων, όπως είναι η στήβα, η ουρά, το δέντρο, ο γράφος και η λίστα, μαζί με τις αντίστοιχες λειτουργίες τους. Ωστόσο, αν θέλουμε να κατασκευάσουμε την οπτικοποίηση του αλγορίθμου από τον κώδικά του, θα πρέπει να χρησιμοποιήσουμε τη γλώσσα συγγραφής σεναρίων της εφαρμογής, κάτι το οποίο θα ήταν αρκετά δύσκολο για αρχάριους στον προγραμματισμό.

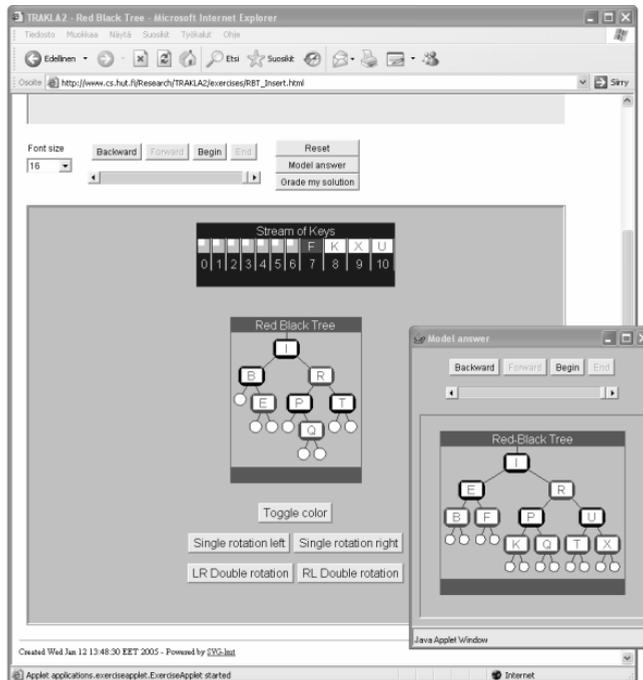
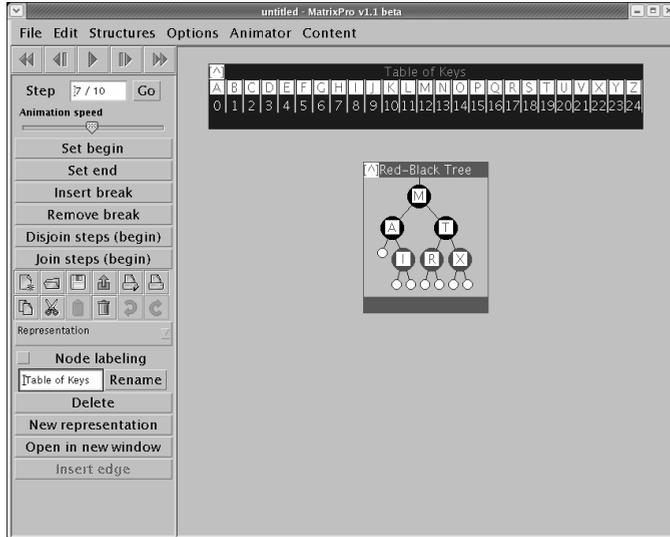


Σχήμα 8. Οπτικοποίηση του αλγορίθμου ταξινόμησης μέσω της εφαρμογής Jawaaw

### **MatrixPro και Trakla2**

Το MatrixPro είναι ένα σύστημα οπτικοποίησης αλγορίθμων και δομών δεδομένων (Karavirta et al., 2004). Πρόκειται για μετεξέλιξη του συστήματος οπτικοποίησης Matrix (Korhonen & Malmi, 2002). Οι μαθητές μπορούν να πειραματιστούν με τις οπτικοποιήσεις διάφορων αλγορίθμων πάνω σε έτοιμες δομές δεδομένων που παρέχει το λογισμικό. Εκτός από τις βασικές δομές δεδομένων, υπάρχουν και υλοποιήσεις πιο πολύπλοκων δομών, όπως αυτή του ισοζυγισμένου δέντρου (AVL). Ο χρήστης μπορεί να ορίσει ένα αντικείμενο αυτού του τύπου και με διαδοχικές εισαγωγές και διαγραφές κόμβων να έχει μια άμεση οπτικοποίηση όχι μόνο της δομής του αλλά και των αλγορίθμων εισαγωγής και διαγραφής δεδομένων.

Οι σχεδιαστές του συστήματος έχουν αναπτύξει μια βάση ασκήσεων για τους μαθητές πάνω στο MatrixPro με στόχο την κατανόηση των βασικών λειτουργιών των δομών δεδομένων (Σχήμα 9). Οι ασκήσεις αυτές αξιολογούνται αυτόματα από το



Σχήμα 9. Η εισαγωγή σε μελανέρυθρα δέντρα στα MatrixPro και Trakla2

σύστημα αξιολόγησης Trakla2 (Korhonen et al., 2003). Το σύστημα αυτό είναι σχεδιασμένο πάνω από το MatrixPro και παρέχει υποστηρικτικό υλικό, όπως σημειώσεις, διαλέξεις, επεξηγήσεις που εμπλουτίζονται συνεχώς. Έχει χρησιμοποιηθεί για την υποστήριξη ενός συνεργατικού διδακτικού μοντέλου με ασκήσεις οπτικοποίησης αλγορίθμων (Laakso et al., 2009). Τα MatrixPro και Trakla2 υποστηρίζουν μεγάλη ποικιλία δομών δεδομένων. Στο παρακάτω σχήμα δίνεται ένα παράδειγμα οπτικοποίησης μελανέρυθρων (red-black) δέντρων (Guibas & Sedgewick, 1978). Πρόκειται για μια ειδική περίπτωση ισοζυγισμένου δυαδικού δέντρου αναζήτησης που παρουσιάζει πολύ καλή απόδοση στις βασικές λειτουργίες του.

## **Ville**

Μια άλλη ερευνητική ομάδα από το Πανεπιστήμιο Turku της Φινλανδίας έχει αναπτύξει το σύστημα οπτικοποίησης Ville (Rajala et al., 2007). Το χαρακτηριστικό αυτού του εργαλείου είναι ότι υποστηρίζει την οπτικοποίηση προγραμμάτων (και όχι αλγορίθμων) σε διάφορες γλώσσες προγραμματισμού όπως C, C#, Python. Ο μαθητής επιλέγει τη γλώσσα προγραμματισμού στην οποία θα κωδικοποιήσει τον αλγόριθμό του. Στη συνέχεια μπορεί να δει την οπτικοποίηση των προγραμματιστικών δομών κατά την εκτέλεση. Το εργαλείο υποστηρίζει τη δημιουργία και αυτοματοποιημένη αξιολόγηση των ασκήσεων των μαθητών συνεργαζόμενο με το Trakla2. Για το λόγο αυτό, αναφέρεται σε αυτή την κατηγορία, παρόλο που δεν θεωρείται σύστημα 'καθαρής' οπτικοποίησης αλγορίθμων.

Η μελέτη του ίδιου παραδείγματος προγραμματισμού σε διάφορες γλώσσες, έχει το πλεονέκτημα ότι ο μαθητής αναγνωρίζει τις κοινές προγραμματιστικές δομές μεταξύ των γλωσσών, εντοπίζοντας τις ομοιότητες. Με τη μέθοδο της αφαίρεσης, ο μαθητής επικεντρώνεται στη μελέτη των βασικών χαρακτηριστικών του αλγορίθμου χωρίς να χάνει χρόνο με τις συντακτικές και τεχνικές λεπτομέρειες της δομής κάθε γλώσσας.

## **Alvie**

Το Alvie είναι ένα σύστημα οπτικοποίησης αλγορίθμων που περιέχει πολλές έτοιμες οπτικοποιήσεις, από απλούς αλγορίθμους ταξινόμησης μέχρι και αλγορίθμους γράφων (Crescenzi & Nocentini, 2007). Πρόσφατα οι δημιουργοί του, σχεδίασαν οπτικοποιήσεις των αποδείξεων για την NP-πληρότητα γνωστών προβλημάτων, όπως αυτό της κάλυψης κορυφών και των 3 χρωμάτων. Το σύστημα έχει μια πολύ απλή διεπαφή και αποτελεί μέρος ενός ηλεκτρονικού βιβλίου το οποίο περιέχει τις περιγραφές όλων των αλγορίθμων μαζί με τις οπτικοποιήσεις τους. Για τη δημιουργία νέων οπτικοποιήσεων χρησιμοποιείται μία XML γλώσσα σεναρίων.

## Leonardo Web

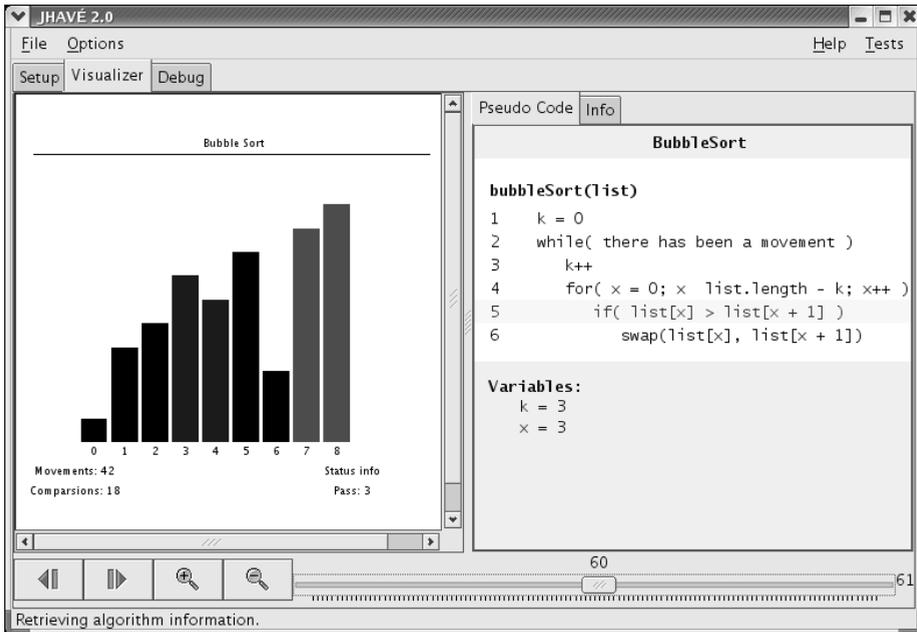
Το σύστημα Leonardo Web (Bonifaci et al., 2006) είναι γραμμένο σε Java και διατίθεται με τη μορφή applet. Έτσι μπορεί κανείς να το κατεβάσει και να το εκτελέσει στον υπολογιστή του. Η οπτικοποίηση του αλγορίθμου δεν γίνεται αυτόματα. Ο χρήστης είναι υποχρεωμένος να χρησιμοποιήσει τη γλώσσα συγγραφής σεναρίων Alpha, να ορίσει γραφικά αντικείμενα και να τα αντιστοιχίσει σε μεταβλητές. Οι αλλαγές των μεταβλητών οπτικοποιούνται μέσω των γραφικών αντικειμένων. Η αντιστοίχιση αυτή ολοκληρώνεται με την προσθήκη κατάλληλων εντολών της γλώσσας Alpha στις εντολές του κώδικα του προγράμματος. Ουσιαστικά η γλώσσα Alpha προσθέτει στον κώδικα τις εντολές οπτικοποίησης. Στη συνέχεια γίνεται η μείξη των εντολών οπτικοποίησης με τον κώδικα του προγράμματος, ώστε να προκύψει ένα είδος ενδιάμεσου-αντικείμενου κώδικα. Ο κώδικας αυτός εκτελείται από μια ιδεατή μηχανή υπολογισμού (virtual CPU). Η ιδεατή μηχανή προσφέρει τη δυνατότητα αντίστροφης εκτέλεσης του αλγορίθμου και της οπτικοποίησης του. Η δυνατότητα αυτή υλοποιείται σε πολύ λίγα περιβάλλοντα οπτικοποίησης προγραμμάτων.

## JHave

Το JHave (*Java Hosted Algorithm Visualization Environment*) (Naps et al., 2003) δεν θεωρείται ένα σύστημα οπτικοποίησης αλγορίθμων από μόνο του, αλλά ένα διαδραστικό περιβάλλον παρουσίασης οπτικοποιήσεων αλγορίθμων που έχουν δημιουργηθεί από άλλα συστήματα (Σχήμα 10). Το JHave βασίζεται στην αρχιτεκτονική πελάτη-εξυπηρετητή (client-server) και έχει υλοποιηθεί σε Java.

Ο εξυπηρετητής διαχειρίζεται τους διαθέσιμους αλγορίθμους οπτικοποίησης και δημιουργεί σενάρια που περιγράφουν την οπτικοποίηση του αλγορίθμου που ζητήθηκε από τον πελάτη. Η εφαρμογή-πελάτης εκτελείται στον υπολογιστή του χρήστη, ο οποίος επιλέγει έναν από τους διαθέσιμους αλγορίθμους. Στη συνέχεια ο πελάτης αποστέλλει μια αίτηση στον εξυπηρετητή και τον ενημερώνει για τον επιλεγμένο αλγόριθμο.

Η εφαρμογή – εξυπηρετητής παράγει την οπτικοποίηση του αλγορίθμου και την αποθηκεύει σε ένα αρχείο με τη μορφή μιας γλώσσα σεναρίων (script language). Στη συνέχεια το αρχείο αυτό αποστέλλεται στο java applet του πελάτη, το οποίο είναι έτσι σχεδιασμένο ώστε να οπτικοποιεί αρχεία γραμμένα στη γλώσσα αυτή. Με την αρχιτεκτονική αυτή το JHave μπορεί να υποστηρίξει πολλές γλώσσες σεναρίων, αφού η μηχανή παρουσίασης της οπτικοποίησης είναι ανεξάρτητη της εφαρμογής που κατασκευάζει το αρχείο περιγραφής της οπτικοποίησης του αλγορίθμου. Στην παρούσα μορφή του, το JHave υποστηρίζει τη γλώσσα GAIGS (*Generalized Algorithm Illustration through Graphical Software*), και τις γλώσσες Animal και AnimalScript (Rössling & Freisleben, 2002). Ένα σημαντικό πλεονέκτημα του συστήματος είναι η δυνατότητα δημιουργίας ερωτήσεων κλειστού τύπου κατά την εκτέλεση της οπτικοποίησης.



Σχήμα 10. Η οπτικοποίηση της ταξινόμησης φυσαλίδας στο JHave

## HalVis

Το σύστημα οπτικοποίησης αλγορίθμων HalVis (*Hypermedia Algorithm Visualization*), αναπτύχθηκε στο πανεπιστήμιο Auburn των Ηνωμένων Πολιτειών (Hansen et al., 2002). Πρόκειται για μια καθαρά υπερμεσική εφαρμογή, που υλοποιήθηκε με το λογισμικό συγγραφής εφαρμογών πολυμέσων Toolbook (Σχήμα 11).

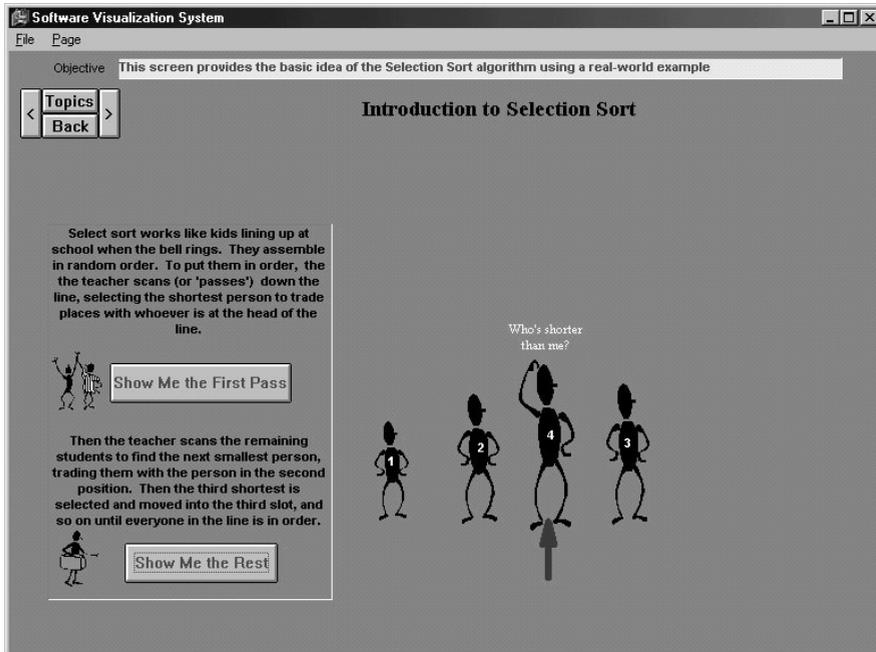
Το σύστημα παρέχει διαφορετικές όψεις της εκτέλεσης του ίδιου αλγορίθμου. Ο χρήστης μπορεί να έχει ανοικτά τέσσερα παράθυρα:

- α) το παράθυρο ψευδοκώδικα, στο οποίο επισημαίνεται η εντολή που εκτελείται,
- β) το παράθυρο με τη νοερή αναπαράσταση των δομών δεδομένων,
- γ) το παράθυρο στο οποίο φαίνονται οι τιμές των μεταβλητών, και
- δ) το παράθυρο με επεξηγήσεις για κάθε βήμα του αλγορίθμου.

Επίσης υιοθετούνται κατάλληλες αναπαραστάσεις των αντικειμένων, ώστε να τονίζονται τα ιδιαίτερα χαρακτηριστικά κάθε αλγορίθμου. Έτσι, κάθε αλγόριθμος ταξινόμησης μπορεί να έχει διαφορετική οπτικοποίηση.

Ένα άλλο σημαντικό πλεονέκτημα είναι η δημιουργία ερωτήσεων στους μαθητές κατά την εκτέλεση του αλγορίθμου. Το HalVis είναι ένα από τα πρώτα συστήματα που εισήγαγε αυτή την τεχνική των ερωταποκρίσεων κατά την εκτέλεση του αλγορίθμου. Παρόλο που δεν προσφέρει την δυνατότητα στον μαθητή να υλοποιήσει τον

δικό του αλγόριθμο, αποτελεί μια πολύ καλή επιλογή για μια πρώτη επαφή με τη λογική βασικών αλγορίθμων.

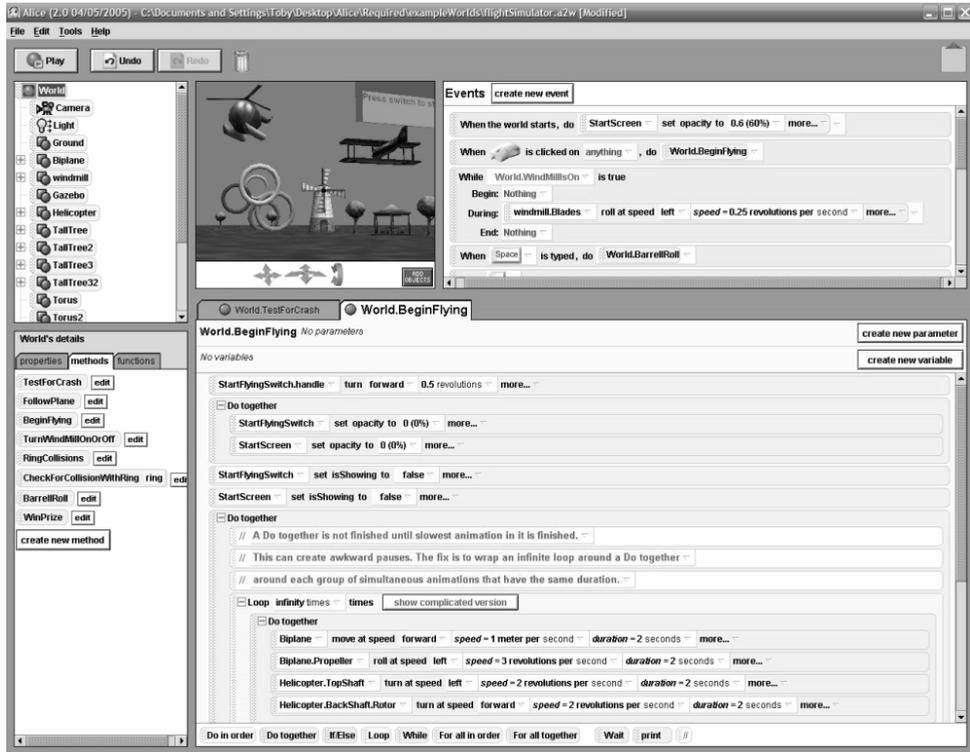


Σχήμα 11. Εισαγωγής στην ταξινόμηση με επιλογή στο HALVIS

## Alice

Το Alice είναι ένα περιβάλλον προγραμματισμού που επιτρέπει τη χρήση τρισδιάστατων γραφικών και τη δημιουργία κινούμενης εικόνας με στόχο την αφήγηση μιας ιστορίας ή τη δημιουργία ενός διαδραστικού παιχνιδιού ή ενός βίντεο (Pausch et.al., 1995). Επιτρέπει τη διαχείριση τρισδιάστατων αντικειμένων (π.χ. άνθρωποι, ζώα, οχήματα) σε έναν εικονικό κόσμο που δημιουργεί ο ίδιος ο μαθητής. Πρόκειται για ένα ελεύθερο διδακτικό εργαλείο, σχεδιασμένο για μια πρώτη επαφή με τον αντικειμενοστρεφή προγραμματισμό (Σχήμα 12), μέσα από τη δημιουργία ταινιών κινουμένων σχεδίων, καθιστώντας τη διαδικασία της μάθησης πολύ ελκυστική για τους μαθητές.

Αν και δεν πρόκειται για ένα περιβάλλον που σχεδιάστηκε με αποκλειστικό στόχο την οπτικοποίηση αλγορίθμων, μπορεί να χρησιμοποιηθεί για το σκοπό αυτό ακόμη και από μικρούς μαθητές. Ο μαθητής καλείται να ορίσει τα αντικείμενα του ιδεατού κόσμου και τον τρόπο, με τον οποίο αυτά αντιδρούν στα διάφορα ερεθίσματα. Για την κωδικοποίηση του αλγορίθμου παρέχονται όλες οι γνωστές αλγοριθμικές δομές (ακολουθίας, επιλογής, επανάληψης) με την μορφή μπλοκ, όπως στο Scratch (2009).



Σχήμα 12. Το περιβάλλον προγραμματισμού Alice

Ο μαθητής επιλέγει αντικείμενα με το ποντίκι, οπότε αποφεύγει συντακτικά λάθη αξιοποιώντας το χρόνο του στην ανάπτυξη του αλγορίθμου και στην εμβάθυνση σε βασικές προγραμματιστικές δομές και έννοιες.

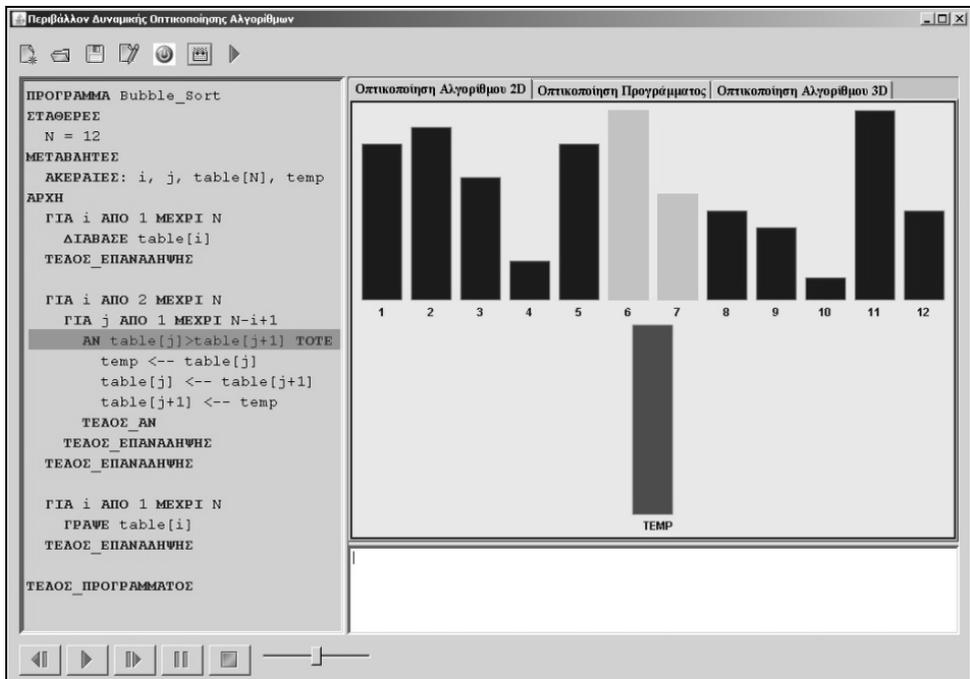
### ***Dynamic Algorithm Visualization Environment (DAVE)***

Το DAVE (Vrachnos & Jimoyannis, 2008) είναι ένα περιβάλλον δυναμικής οπτικοποίησης αλγορίθμων. Επιτρέπει την αυτόματη οπτικοποίηση ενός αλγορίθμου ο οποίος είναι γραμμένος σε ψευδογλώσσα. Η οπτικοποίηση ενός αλγορίθμου μέσω του DAVE δεν περιορίζεται στην απλή εμφάνιση των τιμών των μεταβλητών, κατά την εκτέλεση, αλλά περιλαμβάνει τη δυναμική οπτικοποίηση των δομών που προσομοιώνουν κατάλληλα τη λειτουργία του αλγορίθμου.

Το λογισμικό DAVE αφορά στην οπτικοποίηση βασικών αλγορίθμων ταξινόμησης πινάκων. Ο μαθητής μπορεί να αναπτύξει τον δικό του αλγόριθμο ταξινόμησης και να παρακολουθήσει την οπτικοποίησή του βήμα προς βήμα (Σχήμα 13). Ο μεταγλωττιστής εντοπίζει τα σημαντικά γεγονότα που φαίνονται στην οπτικοποίηση, για παράδειγμα τη σύγκριση ή την ανταλλαγή δύο στοιχείων του πίνακα, και εισάγει τις

κατάλληλες εντολές οπτικοποίησης. Η επέκταση του συστήματος σε άλλα είδη αλγορίθμων μπορεί να γίνει εύκολα με την προσθήκη και άλλων συμβάντων στον κώδικα του μεταγλωττιστή.

Η επιλογή των μέσων αναπαράστασης έγινε έτσι ώστε ο μαθητής να μπορεί εύκολα να παρατηρήσει τις διαφορές μεταξύ των αναμενόμενων και των δικών του αποτελεσμάτων. Οι χρησιμοποιούμενες αναπαραστάσεις είναι διαφορετικές για κάθε αλγόριθμο. Για παράδειγμα, στον αλγόριθμο της ταξινόμησης μονοδιάστατου πίνακα με τη μέθοδο της φυσαλίδας χρησιμοποιούμε για την αναπαράσταση των στοιχείων του πίνακα ράβδους μεταβλητού μεγέθους, ώστε να είναι εμφανής η σχέση διάταξης.



Σχήμα 13. Η ταξινόμηση της φυσαλίδας στο Dave

## Η Συνδυασμένη Ταξινόμια Price/Naps

Στη βιβλιογραφία αναφέρονται διάφορες ταξινομίες για την κατηγοριοποίηση λογισμικών οπτικοποίησης. Μια από τις πρώτες ήταν η ταξινόμια του Myers (1990), στην οποία τα συστήματα κατηγοριοποιούνται με βάση δύο συνιστώσες:

- α) τον τύπο της οπτικοποίησης, δηλαδή τι οπτικοποιείται (κώδικας, δεδομένα, αλγόριθμος) και
- β) τη μορφή εμφάνισης, δηλαδή αν πρόκειται για στατική ή δυναμική οπτικοποίηση (με κίνηση).

Οι Stasko και Patterson (1992) χρησιμοποιούν τέσσερα κριτήρια ταξινόμησης: την όψη/πτυχή (aspect), το επίπεδο αφάιρεσης (abstraction), την οπτικοποίηση (animation) και το επίπεδο αυτοματισμού (automation).

Σε μια άλλη δημοφιλή ταξινομία οι Roman και Cox (1993) χρησιμοποιούν πέντε κριτήρια κατηγοριοποίησης: την εμβέλεια (scope), το επίπεδο αφάιρεσης (abstraction), τη μέθοδο ανάπτυξης της οπτικοποίησης (specification method), και την τεχνική παρουσίασης (technique) που αναφέρεται στο σχεδιασμό της διεπαφής και στην αναπαράσταση της πληροφορίας.

Η ταξινομία του Price (1993) θεωρείται η πληρέστερη ταξινομία συστημάτων οπτικοποίησης και χρησιμοποιείται στην παρούσα συγκριτική μελέτη σε συνδυασμό με την ταξινομία του Naps (2003). Η ταξινομία του Price αναλύει τα συστήματα οπτικοποίησης αλγορίθμων-προγραμμάτων με βάση έξι κριτήρια-δείκτες: εμβέλεια (scope), περιεχόμενο (content), μορφή (form), μέθοδος (method), αποτελεσματικότητα (effectiveness), αλληλεπίδραση (interaction).

Στον Πίνακα 1 παρουσιάζονται συγκριτικά τα συστήματα οπτικοποίησης που μελετήθηκαν και τα βασικά τους χαρακτηριστικά, σύμφωνα με τους άξονες της ταξινομίας του Price.

**Πίνακας 1. Ταξινομία των συστημάτων οπτικοποίησης κατά Price**

Σύστημα	Εμβέλεια	Πλατφόρμα	Περιεχόμενο	Μορφή	Μέθοδος
Jeliot	Προγράμματα σε Java	Java	Αντικειμενο-στρεφείς δομές	Δομές Java, κώδικας	Αυτόματα
Leonardo	Προγράμματα στη γλώσσα Alpha	Java applet	Προγραμματιστικές δομές	Δομές Δεδομένων	Αυτόματα
PlanAni	Pascal, C, κ.α.	Tcl/Tk	Ρόλοι μεταβλητών	Πολλαπλές όψεις	Από τον χρήστη
WinHIPE	Συναρτησιακή γλώσσα Hope	Ανεξάρτητο πλατφόρμας	Συναρτησιακά χαρακτηριστικά	Αναδρομικές κλήσεις	Αυτόματα
JSamba	Ψευδο-γλώσσα	Java applet	Αλγοριθμικά χαρακτηριστικά	Αλγόριθμος, κώδικας	από τον χρήστη
Alvis Live	Cb/SALSA	.NET	Μεταβλητές, πίνακες	Αλγόριθμος, παράθυρο κώδικα	Αυτόματα
Animal	Script γλώσσα	Java applet	Αλγοριθμικά χαρακτηριστικά	Αλγόριθμος, παράθυρο κώδικα	από τον χρήστη

(συνεχίζεται)

Πίνακας 1. (συνέχεια)

Σύστημα	Εμβέλεια	Πλατφόρμα	Περιεχόμενο	Μορφή	Μέθοδος
Jawaa	Script γλώσσα	Java applet	Αλγοριθμικά χαρακτηριστικά	Αλγόριθμος, παράθυρο κώδικα	Από τον χρήστη
Matrix-Pro	Παρέχει έτοιμες τις περισσότερες δομές όπως λίστες, δέντρα	Java	Αλγοριθμικά χαρακτηριστικά	Δομές Δεδομένων	Έτοιμες οπτικοποιήσεις
Trakla2	Περιέχει έτοιμες ασκήσεις	Java	Δομές Δεδομένων	Δομές Δεδομένων	Έτοιμες οπτικοποιήσεις
Ville	Java, C++, ψευδο-γλώσσα	Java	Προγραμματιστικές δομές	Εξήγηση για κάθε εντολή	Αυτόματα
JHave	Πλατφόρμα οπτικοποίησης άλλων συστημάτων	Java Web Start	Αλγοριθμικά χαρακτηριστικά	Αλγόριθμος	Έτοιμες οπτικοποιήσεις
Halvis	Μόνο αλγόριθμοι Ταξινόμησης	Toolbook	Αλγοριθμικά χαρακτηριστικά	Υπερμεσική παρουσίαση πολλαπλών αναπαραστάσεων	Έτοιμες οπτικοποιήσεις
Jive	Προγράμματα σε Java	Java	Αντικειμενοστρεφή χαρακτηριστικά	Αντικειμενοστρεφή χαρακτηριστικά	Αυτόματα
jGrasp	Προγράμματα σε Java	Java	Αντικειμενοστρεφή χαρακτηριστικά	Πολλαπλές αναπαραστάσεις	Αυτόματα
Alice	Ψευδο-γλώσσα	Java	Προγραμματιστικές δομές	Εικονικοί κόσμοι	Αυτόματα
Dave	Ψευδογλώσσα/Γλώσσα	Java	Αλγοριθμικά χαρακτηριστικά	Αλγόριθμοι πινάκων	Αυτόματα

**Εμβέλεια (Scope):** Η εμβέλεια περιγράφει το εύρος των προγραμμάτων ή αλγορίθμων που το σύστημα μπορεί να οπτικοποιήσει. Αν δηλαδή έχει σχεδιαστεί για την οπτικοποίηση συγκεκριμένων προγραμμάτων/αλγορίθμων ή μπορεί να οπτικοποιήσει οποιοδήποτε αλγόριθμο, ο οποίος ανήκει σε μία κλάση αλγορίθμων. Όσο ευρύτερη είναι αυτή η κλάση τόσο μεγαλύτερη θεωρείται η εμβέλεια του συστήματος.

Το σύστημα που ξεχωρίζει σε αυτή την κατηγορία είναι το jGrasp, κυρίως, επειδή επιτρέπει επίσης την οπτικοποίηση των νημάτων όσων προγραμμάτων χρησιμοποιούν παράλληλο προγραμματισμό. Επίσης, το MatrixPro περιέχει πολλές κατηγορίες οπτικοποίησης δομών δεδομένων, όπως πίνακες, λίστες, δέντρα, γραφήματα κλπ.

Η πλατφόρμα στην οποία εκτελείται κάθε σύστημα είναι ένας ακόμα δείκτης της εμβέλειας. Τα περισσότερα συστήματα είναι γραμμένα στη γλώσσα προγραμματισμού Java και έτσι μπορούν να εκτελεστούν σε όλα τα λειτουργικά συστήματα αλλά και μέσω του Παγκόσμιου Ιστού με τη μορφή applets (Animal, Jawaaw).

**Περιεχόμενο (Content):** Το περιεχόμενο αναφέρεται στις πτυχές του προγράμματος ή του αλγορίθμου που θέλουμε να οπτικοποιήσουμε. Τέτοιες πτυχές είναι ο κώδικας του προγράμματος, η αναπαράσταση των δεδομένων και γενικότερα οι αναπαραστάσεις που αναδεικνύουν τα βασικά χαρακτηριστικά της λειτουργίας του αλγορίθμου. Τα συστήματα χωρίζονται ουσιαστικά σε δύο κατηγορίες: α) συστήματα που οπτικοποιούν τις προγραμματιστικές (μεταβλητή, δείκτης, υποπρόγραμμα) ή τις αντικειμενοστρεφείς δομές (κλάση, αντικείμενο, πολυμορφισμός) και β) συστήματα που οπτικοποιούν τις αλγοριθμικές δομές, δηλαδή επιλέγουν κατάλληλες αναπαραστάσεις ώστε να αναδεικνύεται η λογική που κρύβεται πίσω από κάθε αλγόριθμο.

**Μορφή (Form):** Το κριτήριο αυτό αφορά στα χαρακτηριστικά του αποτελέσματος της οπτικοποίησης, όπως το μέσο, τα γραφικά, οι αναπαραστάσεις και η κίνηση των αντικειμένων. Ένα πολύ σημαντικό χαρακτηριστικό είναι η εμφάνιση του κώδικα, ο οποίος εκτελείται συνήθως σε ένα παράθυρο με παράλληλη επισήμανση της τρέχουσα εντολής και του παραγόμενου γραφικού αποτελέσματος καθώς εκτελείται.

**Μέθοδος (Method):** Το κριτήριο αυτό εξετάζει τον τρόπο με τον οποίο παράγεται η οπτικοποίηση. Σε κάποια συστήματα η παραγωγή της οπτικοποίησης γίνεται με αυτοματοποιημένο τρόπο από τον κώδικα του αλγορίθμου του χρήστη (Jeliot, jGrasp, Dave). Στα περισσότερα όμως, ο μαθητής πρέπει να κατασκευάσει ο ίδιος την οπτικοποίηση σε κάποια ειδική γλώσσα (Animal, Jawaaw, JSamba). Αρκετά συστήματα παρέχουν τις δικές τους έτοιμες οπτικοποιήσεις προκαθορισμένων αλγορίθμων (MatrixPro, HalVis) με τις οποίες μπορεί να πειραματιστεί ο χρήστης.

**Αποτελεσματικότητα (Effectiveness):** Το κριτήριο αυτό εξετάζει πόσο αποτελεσματική είναι η οπτικοποίηση του αλγορίθμου τόσο σε τεχνολογικό όσο και σε εκπαιδευτικό ή μαθησιακό επίπεδο. Οι έρευνες που έχουν γίνει για αυτόν τον σκοπό είναι λίγες και δεν καλύπτουν όλα τα συστήματα, οπότε δεν μπορούν να βγουν ασφαλή συμπεράσματα. Για αυτό αποτελεί ένα ενδιαφέρον θέμα για μελλοντική έρευνα.

**Αλληλεπίδραση (Interaction):** Το κριτήριο αυτό αξιολογεί το βαθμό εμπλοκής και αλληλεπίδρασης του εκπαιδευόμενου με το λογισμικό, σε παιδαγωγικό και γνωστικό επίπεδο. Κάποια λογισμικά θέτουν ερωτήσεις στο χρήστη, κατά την εκτέλεση της οπτικοποίησης, επιτρέπουν την παύση τη βηματική ή την ανάστροφη εκτέλεση του

αλγορίθμου, τον έλεγχο της ταχύτητας εκτέλεσης της προσομοίωσης, ακόμη και την κατασκευή της οπτικοποίησης του αλγορίθμου από τον ίδιο το χρήστη.

Το κριτήριο της αλληλεπίδρασης αποτελεί τη σημαντικότερη παράμετρο για την παιδαγωγική αποτελεσματικότητα ενός συστήματος οπτικοποίησης, όπως δείχνουν πολλές έρευνες (για παράδειγμα Naps et al., 2003; Hundhausen et al., 2002). Για την ανάλυση του βαθμού αλληλεπίδρασης χρησιμοποιείται η ταξινομία του Naps (2003), η οποία διακρίνει έξι επίπεδα αλληλεπίδρασης του λογισμικού με τον χρήστη. Η κεντρική ιδέα της ταξινομίας είναι ότι το επίπεδο αλληλεπίδρασης που υποστηρίζεται από το λογισμικό, έχει άμεση σχέση με τα εκπαιδευτικά οφέλη που αποκομίζει ο μαθητής από την ενασχόλησή του με αυτό.

Η κλίμακα ταξινομίας, όσον αφορά τα επίπεδα αλληλεπίδρασης των συστημάτων οπτικοποίησης, που θα χρησιμοποιήσουμε αναλύεται στη συνέχεια:

1. *Παρακολούθηση της οπτικοποίησης (Viewing)*: Ο μαθητής απλά παρακολουθεί την οπτικοποίηση χωρίς να παρεμβαίνει. Όλα τα συστήματα οπτικοποίησης που εξετάσαμε υποστηρίζουν αυτή τη δυνατότητα.
2. *Ελεγχόμενη παρακολούθηση της οπτικοποίησης (Controlled viewing)*: Ο μαθητής μπορεί να ελέγχει την ταχύτητα εμφάνισης της οπτικοποίησης στην οθόνη ή να σταματάει εντελώς την εκτέλεση. Κάποια λογισμικά (Leonardo, jGrasp) παρέχουν ακόμη και τη δυνατότητα εκτέλεσης του αλγορίθμου με την ανάστροφη φορά (προς τα πίσω). Το επίπεδο της *ελεγχόμενης παρακολούθησης* αρχικά δεν υπήρχε στην ταξινομία του Naps αλλά προστέθηκε σε μια επέκτασή της που προτάθηκε πρόσφατα από τον Myller (2009).
3. *Απόκριση σε ερωτήσεις (Responding)*: Ο μαθητής απαντάει σε ερωτήσεις κατά την εκτέλεση του αλγορίθμου οι οποίες στις περισσότερες περιπτώσεις αξιολογούνται άμεσα (Trakla2, Ville). Έτσι ο μαθητής έχει τη δυνατότητα να διερευνήσει, να διορθώσει τα λάθη του και να προχωρήσει γνωστικά.
4. *Τροποποίηση (Changing)*: Ο μαθητής μπορεί να αλλάξει τα δεδομένα εισόδου του αλγορίθμου ή να τροποποιήσει κάποια χαρακτηριστικά της οπτικοποίησης, όπως η ταχύτητα εκτέλεσης, η γραφική αναπαράσταση των αντικειμένων κ.λπ.
5. *Δημιουργία νέας οπτικοποίησης (Constructing)*: Αφορά στις δυνατότητες του εκπαιδευόμενου να δημιουργήσει την δική του οπτικοποίηση. Υπάρχουν λογισμικά (π.χ. jGrasp, Jeliot, DAVE), όπου η οπτικοποίηση παράγεται αυτόματα από τον κώδικα του αλγορίθμου που συντάσσει ο ίδιος ο μαθητής. Στα περισσότερα συστήματα που μελετήθηκαν ο μαθητής είναι υποχρεωμένος να περιγράψει την οπτικοποίηση στην ειδική γλώσσα του συστήματος, η οποία είναι, συνήθως, μία γλώσσα σεναρίων (script language).

Από την ανάλυση που προηγήθηκε είναι προφανές ότι όσο υψηλότερο είναι το επίπεδο στην κλίμακα της ταξινομίας τόσο μεγαλύτερα εκπαιδευτικά οφέλη αναμένονται για το μαθητή. Τα τεχνολογικά και παιδαγωγικά χαρακτηριστικά, καθώς και τα

αναπαραστατικά εργαλεία ενός συστήματος προσομοίωσης αλγορίθμων καθορίζουν τους τρόπους εκπαιδευτικής αξιοποίησης και χρήσης του λογισμικού από εκπαιδευτικούς και μαθητές.

Στον Πίνακα 2 παρουσιάζονται τα επίπεδα διαδραστικότητας των συστημάτων, σύμφωνα με την ταξινομία του Naps.

**Πίνακας 2. Ταξινομία των συστημάτων οπτικοποίησης κατά Naps**

Σύστημα	Παρακολούθηση	Ελεγχόμενη παρακολούθηση	Σύστημα Ερωτήσεων	Τροποποίηση	Δημιουργία οπτικοποίησης
Jeliot	✓	✓	–	✓	✓
Leonardo	✓	✓		✓	–
PlanAni	✓	✓		–	–
WinHIPE	✓	✓		✓	✓
JSamba	✓	✓	–	✓	–
Alvis	✓	✓		✓	✓
Animal	✓	✓	✓	✓	–
Jawaa	✓	✓	✓	✓	–
MatrixPro/ Trakla2	✓	✓	✓	✓	
Ville	✓	✓	✓	✓	–
JHave	✓	✓	✓	-	
Halvis	✓	✓	✓	-	
Jive	✓	✓	✓	✓	✓
jGrasp	✓	✓		✓	✓
Alice	✓	✓		✓	✓
Dave	✓	✓		✓	✓

✓ πλήρης λειτουργικότητα του συστήματος στο επίπεδο αυτό

– περιορισμένη λειτουργικότητα του συστήματος στο επίπεδο αυτό

## Συμπεράσματα

Στην εργασία αυτή έγινε μια συγκριτική παρουσίαση και μελέτη των σημαντικότερων συστημάτων οπτικοποίησης αλγοριθμικών και προγραμματιστικών δομών, τα οποία αναφέρονται στη βιβλιογραφία και είναι διαθέσιμα για εκπαιδευτικούς σκοπούς. Χρησιμοποιήθηκε ο συνδυασμός των ταξινομιών Price και Naps και τα σχετικά κριτήρια περιγραφής για το βαθμό αλληλεπίδρασης που διαθέτει κάθε σύστημα. Έγινε μια ιστορική αναδρομή της εξέλιξης του επιστημονικού πεδίου με την παρουσίαση των πρώτων συστημάτων οπτικοποίησης (Tango, Polka, Samba) και, στη συνέ-

χεια, αναλύθηκαν τα χαρακτηριστικά σύγχρονων συστημάτων, όπως τα JSamba, Jawa, Animal κ.α.).

Τα συστήματα αυτά μπορούν να βοηθήσουν τον εκπαιδευτικό να δημιουργήσει την οπτικοποίηση ενός αλγορίθμου, ώστε να υποστηρίξει τη διδασκαλία του στην τάξη. Το ζητούμενο όμως είναι, ακολουθώντας και τις διεθνείς τάσεις στο χώρο του εκπαιδευτικού λογισμικού, να διαμορφωθούν περιβάλλοντα τα οποία παρέχουν στο μαθητή δυνατότητες να πειραματιστεί με οπτικοποιήσεις αλγορίθμων, να ελέγξει τις αναπαραστάσεις του για αντικείμενα και δομές και, τέλος, να δημιουργήσει τις δικές του οπτικοποιήσεις. Ο σχεδιασμός των νεότερων συστημάτων οπτικοποίησης αλγορίθμων εστιάζει στις απαιτούμενες παιδαγωγικές προδιαγραφές και στο βαθμό αλληλεπίδρασης του συστήματος, ώστε να ενεργοποιούν και θα ενισχύουν το μαθητή. Στην κατεύθυνση αυτή οδήγησαν πολλές έρευνες σχετικά με την αποτελεσματικότητα των συστημάτων οπτικοποίησης (Naps et al., 2003; Hundhausen et al., 2002) και έδειξαν ότι τα εκπαιδευτικά οφέλη για τους μαθητές καθορίζονται από τις χρησιμοποιούμενες *αναπαραστάσεις* και το *βαθμό αλληλεπίδρασης* που προσφέρει κάθε σύστημα.

Το ζήτημα των κατάλληλων γραφικών αναπαραστάσεων για τα υπολογιστικά αντικείμενα, ώστε να αναδεικνύονται για το μαθητή τα σημαντικά χαρακτηριστικά που διέπουν τη συμπεριφορά κάθε αλγορίθμου, αποτελεί την πρώτη παράμετρο μελέτης για το σχεδιασμό συστημάτων οπτικοποίησης. Το γενικό συμπέρασμα που προέκυψε από τη μελέτη των παραπάνω συστημάτων είναι ότι όσο περισσότερο αυτοματοποιημένη είναι η οπτικοποίηση ενός αλγορίθμου, τόσο λιγότερο προσαρμοσμένη είναι στα ιδιαίτερα χαρακτηριστικά του. Αντίθετα, τα συστήματα που προσφέρουν καλές αναπαραστάσεις έχουν χαμηλό βαθμό αυτοματοποίησης στην παραγωγή της οπτικοποίησης. Η δημιουργία μιας νέας οπτικοποίησης από τον ίδιο το χρήστη σε αυτά τα συστήματα, όταν αυτό επιτρέπεται, δεν είναι εύκολη υπόθεση για το μαθητή.

Το *πρώτο επίπεδο αλληλεπίδρασης* αναφέρεται στη δυνατότητα της ελεγχόμενης εκτέλεσης του αλγορίθμου με κάποιο γραφικό χειριστήριο, το οποίο επιτρέπει στο μαθητή να σταματά, να επιβραδύνει ή να επιταχύνει την εκτέλεση του αλγορίθμου, με βάση τις μαθησιακές ανάγκες του κάθε φορά. Στο *δεύτερο επίπεδο αλληλεπίδρασης*, το οποίο δεν υποστηρίζεται από όλα τα συστήματα, δίνεται η δυνατότητα στο μαθητή να απαντήσει σε ερωτήσεις που γίνονται κατά την εκτέλεση του αλγορίθμου, όπως για παράδειγμα 'τι θα συμβεί στο επόμενο βήμα?' (HalVis, JHave). Κάποια συστήματα προσφέρουν επίσης άμεση διόρθωση των απαντήσεων των μαθητών πριν προχωρήσουν στο επόμενο βήμα (Trakla2, Ville). Το *υψηλότερο επίπεδο αλληλεπίδρασης* αφορά στις δυνατότητες που έχει ο μαθητής να δημιουργήσει την οπτικοποίηση του δικού του αλγορίθμου. Δηλαδή, να μπορεί να κωδικοποιήσει έναν αλγόριθμο σε κάποια γλώσσα προγραμματισμού, να δει άμεσα το αποτέλεσμα της οπτικοποίησής του και να αξιοποιήσει στη συνέχεια μέσα από κατάλληλα σενάρια

(π.χ. 'τι θα συμβεί αν αλλάξει κάποια υπολογιστική παράμετρος'). Υπάρχουν συστήματα που οπτικοποιούν αυτόματα τον αλγόριθμο του μαθητή, όπως το Jeliot, το Alvis ή το jGrasp ενώ άλλα παρέχουν τη δυνατότητα στο μαθητή να περιγράψει τα βήματα του αλγορίθμου οπτικά με τη χρήση κάποιας γραφικής διεπαφής (π.χ. MatrixPro, Animal).

Ο βαθμός διαδραστικότητας και η επιλογή κατάλληλων αναπαραστάσεων για τις αλγοριθμικές δομές είναι τα δύο πιο σημαντικά χαρακτηριστικά, τα οποία καθορίζουν την αποτελεσματικότητα ενός συστήματος οπτικοποίησης αλγορίθμων. Ο κατάλληλος σχεδιασμός ενός συστήματος οπτικοποίησης για τη διδασκαλία αλγορίθμων θα πρέπει να έχει ως στόχο την ανάδειξη των σημαντικών στοιχείων (γεγονότων) κάθε αλγορίθμου και την παροχή στον μαθητή αυξημένων δυνατοτήτων αλληλεπίδρασης με το περιβάλλον αυτό. Τα κριτήρια αλληλεπιδραστικότητας και τα εργαλεία αναπαράστασης των εμπλεκόμενων υπολογιστικών δομών και αντικειμένων αποτέλεσαν τα βασικά στοιχεία που καθόρισαν το σχεδιασμό του πρωτοτύπου του περιβάλλοντος δυναμικής οπτικοποίησης αλγορίθμων DAVE (Vrachnos & Jimoyannis, 2008) για τη δυναμική οπτικοποίηση αλγορίθμων ταξινόμησης πινάκων σε ψευδογλώσσα. Το σύστημα DAVE βρίσκεται στη φάση της ανάπτυξης και πρόκειται να χρησιμοποιηθεί πιλοτικά για τη διδασκαλία μαθημάτων αλγοριθμικής σε μαθητές της Γ' Λυκείου Τεχνολογικής Κατεύθυνσης.

## Αναφορές

- Baecker, R. M. (1981). *Sorting out Sorting*. Narrated color videotape presented at ACM SIGGRAPH '81.
- Ben-Bassat Levy, R., & Ben-Ari., M. (2009). A survey of research on the Jeliot program animation system. In Y. Eshet-Alkalai, A. Caspi, S. Eden, N. Geri & Y. Yair (Eds.), *Proceedings of the Chais conference on instructional technologies research 2009: Learning in the technological era* (pp. 41-47). Raanana: The Open University of Israel.
- Bonifaci, V., Demetrescu, C., Finocchi, I., & Laura L. (2006). Visual editing of animated algorithms: The Leonardo Web Builder. *Proceedings of the Working Conference on Advanced Visual Interfaces* (pp. 476-479). Venezia, Italy: ACM.
- Brown, M.H. (1991). Zeus: A system for algorithm animation and multi-view editing. *1991 IEEE Workshop on Visual Languages* (pp. 4-9). Kobe, Japan.
- Crescenzi, P., & Nocentini, C. (2007). Fully Integrating Algorithm Visualization into a CS2 course. *12<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education* (pp.296-300). Dundee, Scotland: ACM.
- Cross, J. H., & Hendrix, T. D. (2006). jGrasp: A lightweight IDE with dynamic object viewers for CS1 and CS2. *12<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education* (pp.356-356). Bologna, Italy: ACM.
- Gestwicki, P., & Jayaraman, B. (2002). Interactive visualization of Java programs. *IEEE 2002 Symposium on Human Centric Computing Languages and Environments* (pp. 226-235). Arlington, USA: IEEE Computer Society.

- Guibas, L., & Sedgewick, R. (1978). A dichromatic framework for balanced trees. *IEEE Symposium on Foundations of Computer Science* (pp. 8-21). Michigan: IEEE Computer Society.
- Hansen, S., Narayanant, H., & Hegarty, M. (2002). Designing educationally effective algorithm visualizations, *Journal of Visual Languages & Computing*, 13(3), 291-317.
- Hundhausen, C., & Brown, J. L. (2007). What you see is what you code: a 'live' algorithm development and visualization environment for novice learners. *Journal of Visual Languages and Computing*, 18(1), 22-47.
- Hundhausen, C., & Douglas, S. (2002). Low-fidelity algorithm visualization. *Journal of Visual Languages and Computing*, 13(5), 449-470.
- Hundhausen, C., Douglas, S., & Stasko, J. (2002). A metastudy of algorithm visualization effectiveness. *Journal of Visual Languages and Computing*, 3(3), 259-290.
- Jimoyiannis A. (2008). Computer simulations and scientific knowledge construction, In A. Cartelli & M. Palma (Eds.), *Encyclopedia of Information Communication Technology* (pp. 106-120). Hershey, PA: IGI Global.
- Karavirta, V., Korhonen, A., Malmi, L., & Stalnacke, K. (2004). MatrixPro. A tool for on-the-fly demonstration of data structures and algorithms. *Third Program Visualization Workshop* (pp. 26-33). The University of Warwick, UK.
- Kolikant, Y. B.-D., & Pollack, S. (2004). Establishing computer science professional norms among high-school students. *Computer Science Education*, 14(1), 21-35
- Kölling, M., Quig, B., Patterson, A., & Rosenberg, J. (2003). The BlueJ system and its pedagogy, *Journal of Computer Science Education, Special issue on Learning and Teaching Object Technology*, 13(4), 249-268.
- Korhonen, A., & Malmi, L. (2002). Matrix - Concept animation and algorithm simulation system. *Working Conference on Advanced Visual Interfaces* (pp. 109-114). Trento, Italy: ACM.
- Korhonen, A., Malmi, L., & Silvasti, P. (2003). TRAKLA2: a framework for automatically assessed visual algorithm simulation exercises. *Third Annual Baltic Conference on Computer Science Education* (pp. 48-56). Joensuu, Finland: Institute of Mathematics and Informatics.
- Laakso, M., Myller, N., & Korhonen, A. (2009). Comparing Learning Performance of Students Using Algorithm Visualizations Collaboratively on Different Engagement Levels. *Educational Technology & Society*, 12(2), 267-282.
- Moreno, A., Myller, N., Sutinen, E., & Ben-Ari, M. (2004). Visualizing Program with Jeliot3. *International Working Conference on Advanced Visual Interfaces, AVI* (pp. 373-380). Bari, Italy: ACM.
- Myers, B.A. (1990). Taxonomies of visual programming and program visualisation. *Journal of Visual Languages of Computing*, 1(1), 97-123.
- Myller, N., Bednarik, R., Sutinen, E., & Ben-Ari, M., (2009). Extending the engagement taxonomy: software visualization and collaborative learning. *ACM Transactions on Computing Education*, 9(1), Article 7.
- Naps T., Eagan J.R., & Norton L.L. (2003). JHAVE. An environment to actively engage students in Web-based algorithm visualizations. *31<sup>st</sup> SIGCSE Technical Symposium on Computer Science Education* (pp. 109-113). Reno, Nevada, USA: ACM.
- Naps, T., Roessling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., & Velasquez Iturbide, J., A. (2003). Exploring the role of visualization and engagement in computer science education. *ACM SIGCSE Bulletin*, 35(2), 131-152.

- Pareja-Flores, C., Urquiza Fuentes J.A., & Velasquez Iturbide, J. (2007). WinHIPE: An IDE for functional programming based on rewriting and visualization, *ACM SIGPLAN Notices*, 43(2), 14-23.
- Pausch, R., Burnette, T., Capeheart, A.C., Conway, M., Cosgrove, D., DeLine, R., Durbin, J., Gossweiler, R., Koga, S., & White, J. (1995). Alice: Rapid prototyping system for virtual reality. *IEEE Computer Graphics and Applications*, 15(3), 8-11.
- Pierson, W., & Rodger S. (1998). Web-based animation of data structures using JAWAA. *29<sup>th</sup> SIGCSE Technical Symposium on Computer Science Education (SIGCSE '98)* (pp. 267-271). Atlanta, Georgia, USA, ACM.
- Price, B.A. Baecker, R. M., & I.S. Small, I. S. (1993). A principled taxonomy of software visualization. *Journal of Visual Languages and Computing*, 4(3), 211-266.
- Rajala, T., Laakso, M., Kaila, E., & Salakoski, T. (2007). VILLE – A language-independent program visualization tool. *Seventh Baltic Sea Conference on Computing Education Research* (Vol.88, pp. 151-159). Kolli National Park, Finland: ACS.
- Ramadhan, H. A. (2000). Programming by discovery. *Journal of Computer Assisted Learning*, 16, 83-93.
- Roman, G. C., & Cox K. C. (1993). A taxonomy for program visualisation systems. *Computer*, 26(12), 11-24.
- Rössling, G., & Freisleben, B. (2002). ANIMAL: A system for supporting multiple roles in algorithm animation. *Journal of Visual Languages & Computing*, 13(3), 341-354.
- Rössling, G., & Schroeder, P. (2009). Animalipse. An eclipse plugin for AnimalScript. *Electronic Notes in Theoretical Computer Science*, 224, 3-14.
- Sajaniemi, J., & Kuittinen, M. (2003). Program animation based on the roles of variables. *ACM Symposium on Software Visualization* (pp. 7-16). San Diego, USA: ACM.
- Scratch (2009). <http://scratch.mit.edu>.
- Stasko, J. & Patterson, C. (1992). Understanding and characterizing software visualization systems. *IEEE Workshop on Visual Languages* (pp. 3-10). Seattle, USA: IEEE Computer Society.
- Stasko, J. (1990). TANGO: A framework and system for algorithm animation. *IEEE Computer*, 23(9), 27-39.
- Stasko, J. (1997). Using student-built algorithm animations as learning aids. *28<sup>th</sup> SIGCSE Technical Symposium on Computer Science Education (SIGCSE 97)* (pp. 25-29), San Jose, USA: ACM.
- JSamba (1998). Retrieved 18 March 2009, from [www.cs.gatech.edu/gvu/softviz/algoanim/jsamba](http://www.cs.gatech.edu/gvu/softviz/algoanim/jsamba).
- Samba (1998). Retrieved 18 March 2009, from [www.cs.gatech.edu/gvu/softviz/algoanim/samba.html](http://www.cs.gatech.edu/gvu/softviz/algoanim/samba.html).
- Stasko, J., & Kraemer, E. (1993). The Visualization of Parallel Systems: An Overview, *Journal of Parallel and Distributed Computing*, 18(2), 105-117.
- Velasquez Iturbide, J., A., Perez-Carrasco, A., & Urquiza-Fuentes, J., A. (2008). SRec: An animation system of recursion for algorithm courses. *13<sup>th</sup> Annual Conference Innovation and Technology in Computer Science Education* (pp. 225-229). Madrid: ACM.
- Vrachnos, E., & Jimoyiannis, A. (2008). Dave: A Dynamic Algorithm Visualization Environment for Novice Learners. *8<sup>th</sup> IEEE International Conference on Advanced Learning Technologies* (pp. 319-323). Santander: IEEE Computer Society.
- Γρηγοριάδου, Μ., Γόγουλου, Α., Γούλη, Ε., & Σαμαράκου, Μ. (2004). Σχεδιάζοντας “Διερευνητικές+ Συνεργατικές” δραστηριότητες σε εισαγωγικά μαθήματα προγραμματισμού. Στο Π. Πολίτης

(επιμ.), Πρακτικά 2<sup>ης</sup> Διημερίδας με Διεθνή Συμμετοχή "Διδακτική της Πληροφορικής" (σ. 86-96). Βόλος.

Κόμης, Β. (2005). *Εισαγωγή στη Διδακτική της Πληροφορικής*. Αθήνα: Κλειδάριθμος.

Κόμης, Β., & Τζιμογιάννης, Α. (2006). Ο προγραμματισμός ως μαθησιακή δραστηριότητα: από τις εμπειρικές προσεγγίσεις στη γνώση παιδαγωγικού περιεχομένου. *Θέματα στην Εκπαίδευση*, 7(3), 229-255.

Τζιμογιάννης, Α., Τσιωτάκης, Π., & Sajaniemi, J. (2006). Μελετώντας το ρόλο των προσομοιώσεων αλγορίθμων στη διδασκαλία του προγραμματισμού στο Ενιαίο Λύκειο. Στο Ε. Σταυρίδου & Χ. Σολομωνίδου (επιμ.), *Πρακτικά Πανελληνίου Συνεδρίου "Ψηφιακό Εκπαιδευτικό Υλικό: Ζητήματα δημιουργίας, διδακτικής αξιοποίησης και αξιολόγησης"* (σ. 99-108). Βόλος.

Τσέλιος, Ν., Γεωργόπουλος, Α., Πολίτης, Π., Πύρζα, Φ., Φανίκος, Α., Κουμπούρη, Δ., & Κόμης, Β. (2006). Μελέτη της χρήσης πολλαπλών αναπαραστάσεων προγράμματος σε εκπαιδευτικό περιβάλλον προγραμματισμού, *Θέματα στην Εκπαίδευση*, 7(3), 251-275.