

ΕΡΓΑΣΤΗΡΙΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

<http://etl.ppp.uoa.gr/>



# Εγχειρίδιο Δευτερογενούς Ανάπτυξης Λογισμικού με το Αβάκιο

---

## Εργαστήριο εκπαιδευτικής Τεχνολογίας

1	Πρόλογος.....	5
2	Πώς να χρησιμοποιήσετε αυτό το εγχειρίδιο.....	6
2.1	Δουλεύοντας με το Αβάκιο.....	7
2.2	Κατασκευάζοντας ένα μικρόκοσμο με το Αβάκιο.....	8
3	Προϋποθέσεις –χρήσιμα εργαλεία.....	8
3.1	Ποιο περιβάλλον χρησιμοποιούμε για την κατασκευή –επεξεργασία ενός μικρόκοσμου.....	8
3.2	Πώς ανοίγουμε έναν μικρόκοσμο στο περιβάλλον Αβάκιο;.....	9
3.3	Γραμμή εργασίας: Προβολή των ψηφίδων του μικρόκοσμου και εμφάνιση των μπαρών τους.....	11
3.3.1	Τι σημαίνει ότι μία ψηφίδα είναι ενεργή;.....	12
3.3.2	Γιατί είναι σημαντικό να εμφανίσουμε τη μπάρα των ψηφίδων;.....	13
3.4	Πώς αποκτάμε πρόσβαση στο «μηχανισμό» του μικρόκοσμου.....	14
3.4.1	Ψηφίδα Logo και Γεγονότα ή Διαχείριση προγραμμάτων ενεργειών ψηφίδων 14	
3.4.2	Γιατί είναι χρήσιμο το παράθυρο του Διαχειριστή προγραμμάτων ενεργειών ψηφίδων;.....	18
3.4.2.1	Σημαντικές διευκρινήσεις για τα γεγονότα των ψηφίδων.....	18
3.4.3	Logo, Γεγονότα ή και τα δύο: πώς ξέρουμε πού να ψάξουμε;.....	19
3.4.3.1	Ανάλυση παραδείγματος: Από τα Γεγονότα μίας ψηφίδας στην ψηφίδα Logo	20
3.4.3.2	Γλώσσα, Εντολές, διαδικασίες, κώδικας.....	24
3.5	Μερικά βασικά στοιχεία για τον προγραμματισμό με Logo.....	25
3.5.1	Σύνταξη εντολών στη Logo.....	25
3.5.1.1	Εντολές ψηφίδων του Αβακίου.....	29
3.5.1.2	Δουλεύοντας με τα μηνύματα λάθους της Logo.....	32
3.5.2	Ορισμός διαδικασιών στη Logo: φτιάχνοντας φωτεινές επιγραφές.....	33
3.5.2.1	Ισχύς διαδικασιών.....	37
3.5.3	Ορισμός υπερδιαδικασιών.....	37

3.5.4	Δημιουργία διαδικασιών με μεταβλητές .....	38
3.5.5	Σημαντικές διευκρινήσεις τις διαδικασίες και τις εντολές.....	39
3.5.6	Ιδέες για αξιοποίηση –επέκταση των παραδειγμάτων που παρουσιάστηκαν	40
3.5.6.1	Προοδευτική αλλαγή χρώματος στη φωτεινή επιγραφή .....	40
3.5.6.2	Δημιουργία θερμομέτρου .....	43
3.5.6.3	Δημιουργία χρωμο-ιστοριών .....	43
4	Ενσωματώνοντας στο μικρόκοσμο μας έτοιμα κομμάτια από άλλον μικρόκοσμο ...	44
4.1	Παίρνοντας αυτούσιες ορισμένες ψηφίδες .....	45
4.2	Ενσωματώνοντας τις ψηφίδες και τις λειτουργικότητές τους στο μικρόκοσμο μας	46
4.2.1	Ενσωμάτωση των αποθηκευμένων ψηφίδων .....	46
4.2.2	Ανοίγοντας τις ψηφίδες για να βρούμε τις λειτουργικότητές τους ή πότε δεν φτάνει η αποθήκευση των ψηφίδων μόνο. ....	47
4.2.3	Ερωτήματα και θέματα που χρειάζεται να προσέξει κανείς.....	49
5	Μαστορεύοντας μισοψημένους μικρόκοσμους.....	50
5.1	Από το Ζογκλέρ στο Arcanoid.....	51
5.1.1	Ανάλυση κώδικα.....	52
5.1.2	Ιδέες για επεκτάσεις – τροποποιήσεις .....	52
5.2	Αντιστοιχίσεις: Από τα Ιστορικά Γεγονότα στις Διαφημίσεις .....	52
5.2.1	Πώς λειτουργεί η αντιστοίχιση .....	53
5.2.2	Ανάλυση κώδικα.....	55
5.2.3	Κριτική και επεκτάσεις της αρχικής ιδέας .....	59
5.2.3.1	Κριτική – ή κατευθύνσεις για αλλαγή .....	59
5.2.3.2	Αντιστοίχιση των γραμμάτων του Αλφαβήτου με εικόνες.....	60
5.2.3.3	Αντιστοίχιση φωτογραφιών για διαφημίσεις με λεζάντες .....	60
5.2.3.4	Παροιμίες .....	61
5.2.3.5	Αριάδνη: Αντιστοίχιση Αρχαίων Ελληνικών με τη μετάφρασή τους.....	62
6	Συνήθη προβλήματα και απαντήσεις.....	63

## Εργαστήριο εκπαιδευτικής Τεχνολογίας

6.1	Γιατί δεν είναι ορατό το μενού του μικρόκοσμου; (Μικρόκοσμος, Ψηφίδα, Εργαλεία) .....	63
6.2	Γιατί όταν κάνω κλικ στο μικρόκοσμο ταξινομούμε που έχω αποθηκεύσει δεν ανοίγει;.....	63
6.3	Πώς οργανώνω τις ψηφίδες στο μικρόκοσμό μου ιδιαίτερα όταν είναι πολλές	63
6.3.1	Χρήση κουμπιών: Κουμπί, Διακόπτης, Κουμπί Ελέγχου και Κουμπί Επιλογής	64
6.3.2	Χρήση όψεων.....	65
6.3.3	Δημιουργία σύνθετων ψηφίδων με χρήση της λειτουργικότητας φιλοξενίας	65
6.4	Πώς εμφανίζω στο Αβάκιο κείμενα Αρχαίων Ελληνικών .....	65
6.4.1	Προβάλλοντας το κείμενο από το Internet.....	66
6.4.2	Προβάλλοντας το κείμενο από το δικό σας υπολογιστή: .....	66
7	Βιβλιοθήκη μαύρων κουτιών .....	67
7.1	Shuffle : Δημιουργεί μία λίστα με όσους αριθμούς ορίσουμε εμείς, σε τυχαία σειρά	67
8	Βιβλιογραφικές αναφορές.....	67
9	Index.....	68
10	Πίνακας Βασικών ενεργειών .....	68

## 1 Πρόλογος

«Το Αβάκιο δεν είναι για όλους!» Παραφράζοντας τη φράση με την οποία ξεκινάει το βιβλίο του ο Brian Harvey (Harvey, 1997) επιδιώκουμε να δείξουμε ότι η λογική της ενασχόλησης με το Αβάκιο έχει τα εξής χαρακτηριστικά:

- Βρίσκεται στον αντίποδα της κατανάλωσης έτοιμων προϊόντων λογισμικού (και της λογικής «click and don't think»)
- Η έμφαση είναι στα εργαλεία έκφρασης και κατασκευής που δίνει στο χρήστη και όχι στα θεαματικά γραφικά, στην απόθεση υλικού ή απλώς στην προσθήκη περιεχομένου
- Η έμφαση είναι στην περιγραφή της λογικής βάσει της οποίας σχεδιάζεται ένα εκπαιδευτικό λογισμικό αλλά και της κατασκευής του μηχανισμού που κρύβεται πίσω από αυτό το λογισμικό και το κάνει να λειτουργεί.

Εδώ τίθεται συχνά το ερώτημα: Γιατί να φτιάξω το δικό μου λογισμικό όταν υπάρχουν τόσα έτοιμα που κυκλοφορούν και μάλιστα πολύ πιο ελκυστικά (κυρίως από πλευράς γραφικών). Συγκεκριμένα στο πλαίσιο ενός μαθήματος η ίδια λογική διατυπώθηκε ως εξής: «Μπήκα σε μία σελίδα και βρήκα 15 εργαλεία web 2.0 που κάνουν ό,τι κάνει και το Αβάκιο. Γιατί λοιπόν να μπω στη διαδικασία να φτιάξω κάτι με το Αβάκιο;» Πέρα από τη γενικότητα και την ανακρίβεια της έκφρασης (επειδή η λειτουργία του Αβακίου δεν περιορίζεται στις δυνατότητες που έχει η κάθε ψηφίδα ξεχωριστά αλλά στη μεταξύ τους επικοινωνία) υπάρχει ένα πρόβλημα στη σύγκριση που γίνεται: για να χρησιμοποιήσουμε μία επίκαιρη μεταφορά «είναι σαν να συγκρίνει κανείς 10 γλυκά που μπορεί να βρει στο ζαχαροπλαστείο με ένα μίξερ που δίνει τη δυνατότητα να φτιάξει κάποιος ό,τι γλυκό θέλει και ακριβώς όπως το θέλει (με περισσότερη ή λιγότερη ζάχαρη για παράδειγμα ή με την τάδε μαρμελάδα και όχι κάποια άλλη κ.ο.κ)». Η σύγκριση κατά συνέπεια αφορά διαφορετικά πράγματα.

Ένα άλλο επιχείρημα που ακούγεται είναι το εξής: «τα γραφικά του Αβακίου δεν είναι ελκυστικά και άρα οι μαθητές δεν θα ενδιαφερθούν να χρησιμοποιήσουν το λογισμικό γιατί είναι συνηθισμένοι σε περιβάλλοντα που έχουν εντυπωσιακά γραφικά και μάλιστα συχνά τρισδιάστατα. Πριν επιχειρήσουμε να δούμε τις απαντήσεις ας σκεφτούμε λίγο περισσότερο το συγκεκριμένο επιχείρημα: μοιάζει με έναν τρόπο σαν να λέμε ότι τα παιδιά δεν θα ενδιαφέρονταν να ζωγραφίσουν ή να δουν μια ζωγραφιά επειδή υπάρχουν οι φωτογραφίες ή ότι μπορεί να μην διασκέδαζαν πια με το κυνηγητό επειδή υπάρχει το world of Warcraft σήμερα και αύριο κάτι άλλο. Αυτό που προσπαθούμε να πούμε εδώ είναι πως είναι μεν σημαντικό ένα λογισμικό να έχει ελκυστικά και ενδιαφέροντα γραφικά αλλά αυτό δεν είναι ούτε ο μόνος ούτε ο κυρίαρχος πόλος έλξης των μαθητών. Αυτό στο οποίο επενδύουμε με το Αβάκιο –και αυτό το οποίο καθορίζει τελικά και την επιτυχία στα παιχνίδια όπως το παραπάνω περισσότερο από τα γραφικά- είναι η ιδέα: τι ζητάμε δηλ. από τους μαθητές να κάνουν με τα λογισμικά που φτιάχνουμε εμείς και τι εργαλεία τους δίνουμε για αυτό το σκοπό.

Τέλος πριν ξεκινήσουμε να δουλεύουμε με το Αβάκιο καλό θα είναι να εξηγήσουμε ορισμένα σημαντικά πράγματα:

Το Αβάκιο είναι – και αντιμετωπίζεται σε αυτό το εγχειρίδιο- ως συγγραφικό εργαλείο, ως εργαλείο δηλ. που μας δίνει τη δυνατότητα να φτιάξουμε δικά μας λογισμικά<sup>1</sup>. Ως εκ τούτου αυτό που εξετάζουμε εδώ δεν είναι η σύγκριση του δικού μας τελικού προϊόντος με ένα εμπορικό λογισμικό αλλά τα μέσα που μας δίνονται για να εκφράσουμε και να υλοποιήσουμε τις παιδαγωγικές μας ιδέες. Επίσης για να μπορέσει κανείς να κρίνει το Αβάκιο θα πρέπει να αντιληφθεί ότι μέσα στο αβάκιο συνεργάζονται και επικοινωνούν (interoperability) λογισμικά (που έχουν υλοποιηθεί με συνδυασμούς ψηφίδων) τα οποία περιλαμβάνουν από απλό κειμενογράφο, φυλλομετρητή, μέχρι βάσεις δεδομένων, προσομοιώσεις και χάρτες. Το σημαντικό είναι πως όλα αυτά μέσα στο Αβάκιο δεν υπάρχουν ως αυτόνομες οντότητες αλλά μη τεχνικοί χρήστες μπορούν να τα κάνουν να επικοινωνήσουν μεταξύ τους<sup>2</sup>

## 2 Πώς να χρησιμοποιήσετε αυτό το εγχειρίδιο

Αυτό το εγχειρίδιο έχει γραφτεί με τρόπο που να απαιτεί παράλληλα την ανάγνωσή του και τη χρήση του Αβακίου. Αυτό σημαίνει ότι καθώς διαβάσετε το εγχειρίδιο θα πρέπει να έχετε ανοιχτό και το Αβάκιο ή να ανοίγετε τους μικρόκοσμους στους οποίους σας παραπέμπει το εγχειρίδιο και να δοκιμάζετε να κάνετε όλα όσα περιγράφονται εδώ.

Η προσπάθεια που γίνεται εδώ είναι να αναλυθεί ως επί το πλείστον η λογική μέσα από την οποία εμπλέκεται κανείς στο μαστόρεμα με ένα εργαλείο όπως το Αβάκιο (π.χ. πώς αποκτάμε πρόσβαση στο μηχανισμό που υπάρχει πίσω από κάθε μικρόκοσμο, πώς μαθαίνουμε από και πώς αξιοποιούμε άλλους μικρόκοσμους, κλπ). Ως εκ τούτου ο στόχος του συγκεκριμένου εγχειριδίου είναι διπλός α) να παρέχει τα απαραίτητα εργαλεία για το μαστόρεμα με το Αβάκιο και β) να αναλύσει τον τρόπο σκέψης, πώς δηλ. σκεφτόμαστε για να χαλάσουμε, να αλλάξουμε ή να φτιάξουμε από την αρχή ένα μικρόκοσμο με το Αβάκιο. Ο δεύτερος στόχος φιλοδοξούμε ότι θα σας υποστηρίξει να αναπτύξετε – αν δεν τον έχετε ήδη – έναν τρόπο σκέψης που θα σας επιτρέπει να «μαστορέψετε» και άλλα τεχνολογικά εργαλεία.

Αυτό το εγχειρίδιο μπορείτε να το χρησιμοποιήσετε με δύο τρόπους: είτε γραμμικά ξεκινώντας από την αρχή και προχωρώντας είτε επιλεκτικά αναζητώντας στα περιεχόμενα τα θέματα που σας ενδιαφέρουν. Μπορείτε ωστόσο να συνδυάσετε τους δύο τρόπους να ξεκινήσετε δηλ. από την αρχή και στη συνέχεια να επιλέξετε πού θέλετε

---

<sup>1</sup> Είναι σημαντικό να σημειωθεί εδώ ότι η επιστημονική κοινότητα πλέον έχει στρέψει το ενδιαφέρον της όχι στα χαρακτηριστικά μεμονωμένων λογισμικών αλλά στη δυνατότητα ενδυνάμωσης των τελικών χρηστών με την παροχή εργαλείων που επιτρέπουν επέμβαση στο σχεδιασμό των λογισμικών. Ας ληφθεί υπ' όψιν ότι το Αβάκιο με τις δυνατότητες αυτές που τώρα αποτελούν τεχνολογία αιχμής έχει εμφανιστεί από το 1999.

<sup>2</sup> Σκεφτείτε αν αυτό ακόμη και σήμερα συμβαίνει με τα εκπαιδευτικά ή τα γενικής χρήσης λογισμικά που χρησιμοποιείτε

να δώσετε έμφαση. Οδηγός σας θα είναι α) το επίπεδο εξοικείωσής σας με το Αβάκιο –για μη εξοικειωμένους χρήστες ίσως η γραμμική προσέγγιση να είναι πιο κατάλληλη και β) το τι θέλετε να κάνετε με το Αβάκιο<sup>3</sup>. Για να γίνει πιο εύχρηστο το εγχειρίδιο έχει προστεθεί στο τέλος του εγχειριδίου ένα Index με τους βασικούς όρους καθώς και ένας πίνακας με τις βασικές ενέργειες στο Αβάκιο (π.χ. πώς αλλάζουμε το μέγεθος ή το όνομα μίας ψηφίδας, τι σημαίνει ορίζουμε μία διαδικασία και τι σημαίνει εκτελούμε μία διαδικασία) καθώς επίσης και τις σελίδες στις οποίες θα βρείτε οδηγίες για αυτές τις ενέργειες.

## **2.1 Δουλεύοντας με το Αβάκιο**

Πριν ξεκινήσετε να δουλεύετε με το Αβάκιο χρειάζεται να έχετε υπ' όψη σας το εξής: το Αβάκιο είναι ένα δυναμικό και όχι στατικό εργαλείο πράγμα που σημαίνει ότι ανταποκρίνεται στις δικές σας ενέργειες. Επίσης είναι πρωτίστως συγγραφικό εργαλείο (authoring tool) πράγμα που σημαίνει ότι κρίνεται από τις δυνατότητες που δίνει στο χρήστη να φτιάχνει δικά του πράγματα. Αν κάτι δείχνει να μην έχει το επιθυμητό αποτέλεσμα θα πρέπει να προσπαθείτε να καταλάβετε τι δεν έχει πάει καλά αναλύοντας τη συμπεριφορά του Αβακίου και ποια από τις ενέργειές σας προκάλεσε το πρόβλημα (η ενότητα 3.5.1.2 : για παράδειγμα είναι ενδεικτική αυτής της λογικής). Ενδεχόμενες δυσκολίες με αυτή την προσέγγιση πηγάζουν από την αντιμετώπιση του Αβακίου σαν ένα κλειστό λογισμικό που καλύπτει συγκεκριμένα –μικρή - γκάμα ενεργειών και άρα οι ενέργειες αυτές είναι εύκολο να μαθευτούν και να επαναλαμβάνονται χωρίς παρεκκλίσεις. Τα λογισμικά αυτά όμως είναι πλέον παρελθόν. Αυτό που συμβαίνει με τα λογισμικά που κυκλοφορούν σήμερα είναι πως βρίσκονται σε μία διαρκή διαδικασία εξέλιξης και βελτίωσης (π.β. τα fora συζητήσεων που σχετίζονται με νέες λειτουργικότητες, απαιτήσεις χρηστών ή ακόμη και σφάλματα που αφορούν ευρείας χρήσης λογισμικά ξεκινώντας από το Mozilla μέχρι τη Microsoft τα λογισμικά της οποίας τα πληρώνουμε) η οποία διαδικασία υποστηρίζεται σε μεγάλο βαθμό από τους ίδιους τους χρήστες. Αυτή η στάση είναι στοιχείο μίας ολόκληρης κουλτούρας και για να είναι κανείς σε θέση να γίνει μέρος αυτής της κουλτούρας χρειάζεται την αναλυτική δεξιότητα που του/της επιτρέπει να βλέπει σε λεπτομέρεια τόσο την ανταπόκριση του λογισμικού όσο και τις δικές του ενέργειες.

Θα θέλαμε τέλος να επισημάνουμε, ως κατακλείδα για αυτή την ενότητα, ότι η εξοικείωση με το Αβάκιο εμπλέκει τις εξής διαδικασίες: χρήση του εγχειριδίου ως οδηγό, «λέρωμα των χεριών» με το ίδιο το Αβάκιο, και συζήτηση με τους άλλους συναδέλφους – συμφοιτητές – συμμαθητές, για τα προβλήματα που συναντάτε και τι κάνετε για να τα λύσετε. Επειδή συχνά δεν δίνεται ιδιαίτερη σημασία στο τρίτο μέρος της διαδικασίας θα θέλαμε να σημειώσουμε πως το θεωρούμε εξίσου σημαντικό με τα άλλα δύο και ότι αν το παραλείψει κάποιος χάνει πάρα πολλά σε επίπεδο γνώσης και εμπειρίας.

---

<sup>3</sup> Έχοντας βέβαια υπόψη ότι το τι θέλει κανείς να φτιάξει με ένα εργαλείο εμπλουτίζεται καθώς γνωρίζει καλύτερα το ίδιο το εργαλείο άρα χρήσιμο είναι το τι θέλετε να φτιάξετε να μη λειτουργήσει ως παρωπίδα αλλά να το θεωρήσετε ως μία διαδικασία διαλόγου με το εργαλείο και τους συναδέλφους σας.

## **2.2 Κατασκευάζοντας ένα μικρόκοσμο με το Αβάκιο**

Η διαδικασία κατασκευής ενός μικρόκοσμου μπορεί να είναι σύνθεση μίας σειράς από διαφορετικές διαδικασίες –διαδρομές και πρακτικές. Η μία διαδρομή είναι να πάρει κάποιος έτοιμα κομμάτια από έναν μικρόκοσμο που ήδη υπάρχει και να τα προσαρμόσει στο δικό του, η άλλη διαδικασία –διαδρομή είναι να αλλάξει ή να τροποποιήσει υπάρχοντες μικρόκοσμους και η τρίτη διαδικασία είναι να επιχειρήσει να συνθέσει ένα μικρόκοσμο από την αρχή. Η κατασκευή ενός νέου μικρόκοσμου δεν απαιτεί να ακολουθήσει κανείς αυστηρά τη μία ή την άλλη διαδικασία καθώς όπως αναφέρθηκε νωρίτερα είναι δυνατόν να συνδυάσει κανείς πρακτικές που υπάρχουν και στις τρεις διαδικασίες.

## **3 Προϋποθέσεις –χρήσιμα εργαλεία**

Στην ενότητα αυτή θα περιγράψουμε τι είναι απαραίτητο να ξέρουμε και τι εργαλεία θα πρέπει να χρησιμοποιήσουμε προκειμένου να είμαστε σε θέση να μαστορέψουμε ένα μικρόκοσμο. Τα εργαλεία και οι επεξηγήσεις θα παρουσιαστούν μέσα από συγκεκριμένα παραδείγματα και γι' αυτό καλό θα είναι να έχετε ανοιχτό το Αβάκιο καθώς θα δουλεύετε με το εγχειρίδιο και να ακολουθείτε τα ίδια βήματα όπως αυτά που παρουσιάζονται στο παράδειγμα.

Υπόθεση εργασίας: Ας υποθέσουμε ότι θέλουμε να ανοίξουμε το μικρόκοσμο `paroimies-manual.mwd`<sup>4</sup> που βρίσκεται στο φάκελο «Συνοδευτικό υλικό» και στον υποφάκελο μικρόκοσμοι» προκειμένου να μπορέσουμε στη συνέχεια είτε να πάρουμε κάποια κομμάτια από αυτόν (π.χ. τη λειτουργικότητα: εμφάνιση οδηγιών χρήσης με το πάτημα στο κουμπί οδηγίες) είτε να αλλάξουμε το υλικό του, είτε να προσθέσουμε λειτουργικότητες.

### **3.1 Ποιο περιβάλλον χρησιμοποιούμε για την κατασκευή – επεξεργασία ενός μικρόκοσμου**

**Λέξεις κλειδιά:** Αβάκιο ή Αβάκιο Composer, Αβάκιο Viewer

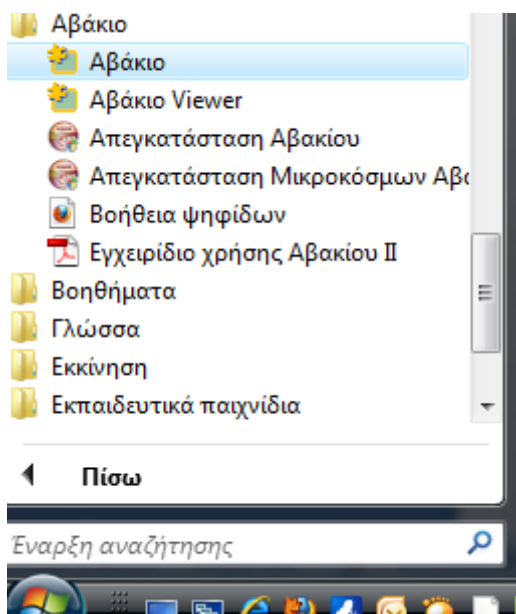
**Μικρόκοσμος:** `paroimies-manual.mwd`

Για να δημιουργήσουμε – αλλάξουμε - τροποποιήσουμε ένα μικρόκοσμο πρέπει να βρισκόμαστε στο περιβάλλον Αβάκιο και όχι στο περιβάλλον Αβάκιο Viewer. Για να ανοίξουμε το περιβάλλον Αβάκιο θα πρέπει να πάμε στο μενού Έναρξη του υπολογιστή μας, να επιλέξουμε Προγράμματα -> Αβάκιο -> Αβάκιο.

---

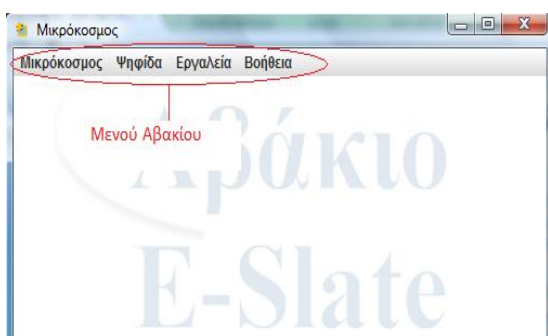
<sup>4</sup> Ο μικρόκοσμος `paroimies-manual.mwd` προέκυψε από το μικρόκοσμο `paroimies.mwd` που δημιουργήθηκε από τη Δρα Ιωάννα Ηλιοπούλου στο πλαίσιο της συνεργασίας της με το Εργαστήριο Εκπαιδευτικής Τεχνολογίας στο ερευνητικό έργο *Seeding cultural change in the school system through the generation of communities engaged in integrated educational and technological innovation* (SEED), European Community, IST, School of Tomorrow, IST-2000-25214. (2001-2004) Μεταφέρθηκε και προσαρμόστηκε στο Αβάκιο 2.0 από τον υποψ. Δρ. Μάριο Ξένο



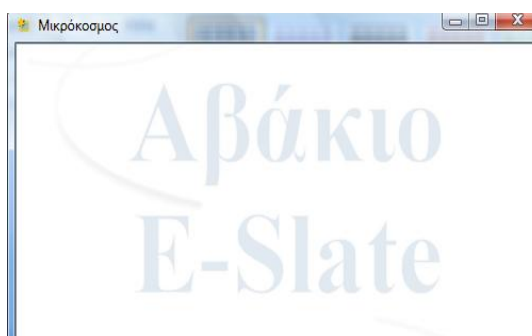


Εικόνα 3.1 Εκκίνηση Αβακίου σε κατάσταση σύνθεσης μικρόκοσμου (Αβάκιο composer)

Ποια είναι η όμως η διαφορά ανάμεσα στο περιβάλλον Αβάκιο και στο Περιβάλλον Αβάκιο Viewer; Όταν το Αβάκιο έχει ανοίξει στο περιβάλλον Viewer τότε ο χρήστης μπορεί να χρησιμοποιήσει μόνο όσες λειτουργικότητες έχουν ενσωματωθεί στο μικρόκοσμο (π.χ. να προσθέσει κείμενο, να αλλάξει τις τιμές κάποιας μεταβλητής κλπ) αλλά δεν μπορεί να κάνει τίποτα παραπάνω (π.χ. να αποθηκεύσει τις αλλαγές που έκανε δίνοντας στο μικρόκοσμο ένα όνομα της επιλογής του). Αυτό οφείλεται στο γεγονός ότι στην κατάσταση Αβάκιο Viewer δεν είναι εμφανές το μενού Αβακίου (βλ. Εικόνα 3.3.) ενώ είναι εμφανές στην κατάσταση Αβάκιο (βλ. Εικόνα 3.2).



Εικόνα 3.2 Περιβάλλον Αβάκιο



Εικόνα 3.3 Περιβάλλον Αβάκιο Viewer

Η διαδικασία κατασκευής λοιπόν όπως περιγράφεται σε αυτό το εγχειρίδιο αφορά πάντοτε το Περιβάλλον Αβάκιο και όχι το περιβάλλον Αβάκιο Viewer εκτός και δηλώνεται το αντίθετο.

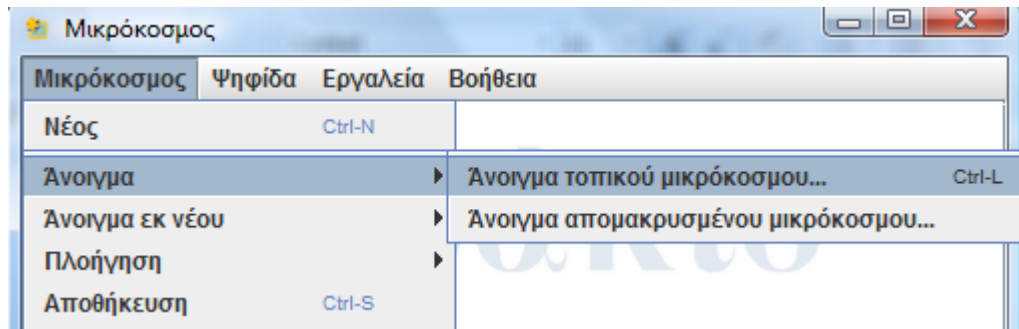
### **3.2 Πώς ανοίγουμε έναν μικρόκοσμο στο περιβάλλον Αβάκιο;**

Στην πολύ συχνή περίπτωση που θέλουμε να επεξεργαστούμε ένα μικρόκοσμο που υπάρχει ήδη, στην προκειμένη περίπτωση το μικρόκοσμο raqoimies -manual.mwd, για να τον ανοίξουμε στο περιβάλλον Αβάκιο πρέπει οπωσδήποτε να κάνουμε τα εξής βήματα:

## Εργαστήριο εκπαιδευτικής Τεχνολογίας

Πάμε στο μενού έναρξη του υπολογιστή μας και επιλέγουμε -> Προγράμματα -> Αβάκιο -> Αβάκιο.

Στη συνέχεια πηγαίνουμε στο μενού Μικρόκοσμος επιλέγουμε Άνοιγμα -> Μικρόκοσμος -> Τοπικός μικρόκοσμος (βλ. Εικόνα 3.4) και στη συνέχεια στο παράθυρο που εμφανίζεται επιλέγουμε το αρχείο που θέλουμε να ανοίξουμε (στην προκειμένη επιλέγουμε το αρχείο ragoimies-manual.mwd από τον υποφάκελο Μικρόκοσμοι που βρίσκεται μέσα στο φάκελο συνοδευτικό υλικό)



Εικόνα 3.4 Άνοιγμα μικρόκοσμου στο περιβάλλον Αβάκιο

ΠΡΟΣΟΧΗ: είναι δυνατόν ένας μικρόκοσμος να ανοίξει κάνοντας διπλό κλικ πάνω στο αρχείο. Σε αυτή την περίπτωση όμως ανοίγει στο περιβάλλον Αβάκιο Viewer και λείπει το μενού του Αβακίου που αναφέρθηκε στην προηγούμενη ενότητα (βλ. Εικόνα 3.3.) με τους περιορισμούς που περιγράφηκαν εκεί.

### Συμβουλές

#### Πρόσφατα αρχεία στο Αβάκιο

Μπορείτε να χρησιμοποιήσετε την επιλογή Άνοιγμα εκ νέου για να επιλέξετε κάποιον μικρόκοσμο που έχετε πρόσφατα ανοίξει, έτσι αποφεύγετε να ψάχνετε στους φακέλους του υπολογιστή σας (λειτουργεί όπως τα πρόσφατα αρχεία στο word).

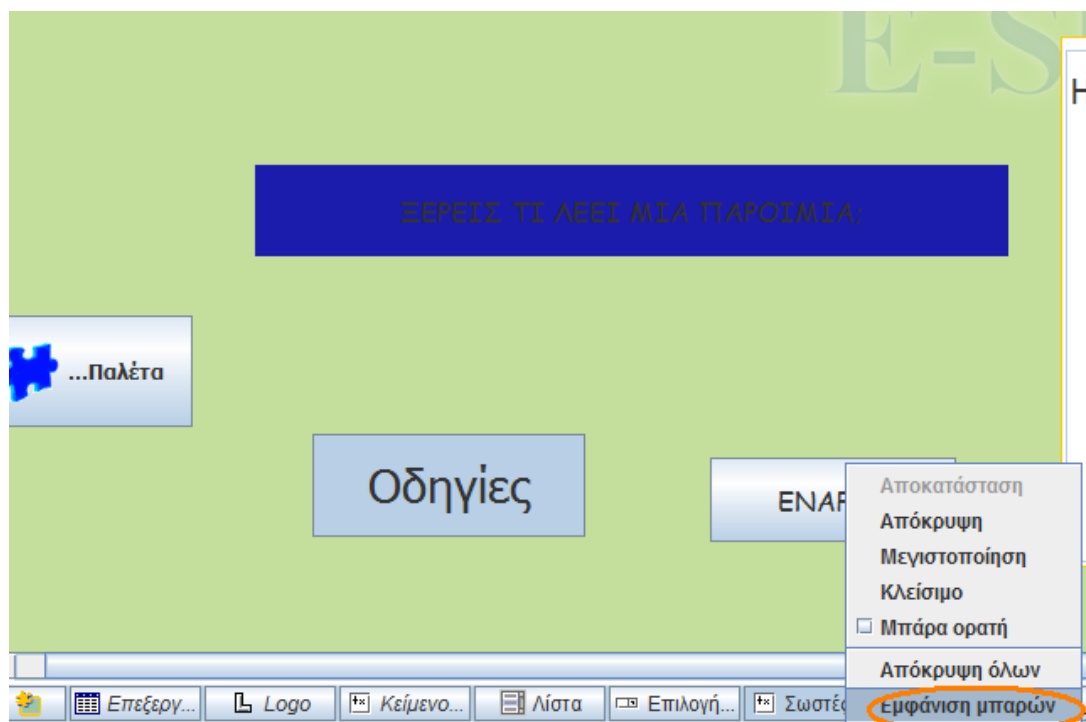
#### Άνοιγμα αρχείων Αβακίου πάντοτε με το Αβάκιο Composer

Μπορείτε να ανοίγετε πάντοτε τα αρχεία του Αβακίου (.mwd αρχεία) με το E-Slate composer αν ακολουθήσετε την εξής διαδικασία: Κάνετε δεξί κλικ σε ένα αρχείο του αβακίου (π.χ. στο μικρόκοσμο ragoimies-manual.mwd) στο μενού που εμφανίζεται επιλέξτε «Άνοιγμα με» και επιλέξτε «Επιλογή Προεπιλεγμένου Προγράμματος». Στο παράθυρο που θα εμφανιστεί κάνετε κλικ ώστε να επιλεγεί το «Άνοιγμα αυτού του είδους του αρχείου πάντα με το επιλεγμένο πρόγραμμα» στη Συνέχεια επιλέξτε Αναζήτηση και πηγαίνετε στο φάκελο που έχετε εγκαταστήσει το Αβάκιο. Αν δεν το έχετε αλλάξει ο φάκελος αυτός βρίσκεται στα «Αρχεία Εφαρμογών» ή Program Files και στο φάκελο E-Slate2. Από το φάκελο αυτό επιλέξτε το αρχείο E-SlateComposer.exe. Στη συνέχεια πιέστε άνοιγμα. Εις το εξής όλα τα αρχεία Αβακίου ανοίγουν με το «Αβάκιο Composer»

### 3.3 Γραμμή εργασίας: Προβολή των ψηφίδων του μικρόκοσμου και εμφάνιση των μπαρών τους

Για να μπορέσουμε να μαστορέψουμε ένα μικρόκοσμο χρειάζεται σε πρώτη φάση να καταφέρουμε να δούμε από ποια κομμάτια – στην προκειμένη ψηφίδες - αποτελείται. Συχνά σε ένα μικρόκοσμο οι ψηφίδες που είναι ορατές δεν είναι απαραίτητα και οι μόνες ψηφίδες που έχουν χρησιμοποιηθεί για την κατασκευή του. Ανοίγουμε λοιπόν το μικρόκοσμο `ragoimies manual.mwd` με τον τρόπο που αναφέρθηκε στην προηγούμενη ενότητα. Ωστόσο οι οδηγίες που δίνονται εδώ μπορούν να λειτουργήσουν για οποιονδήποτε μικρόκοσμο.

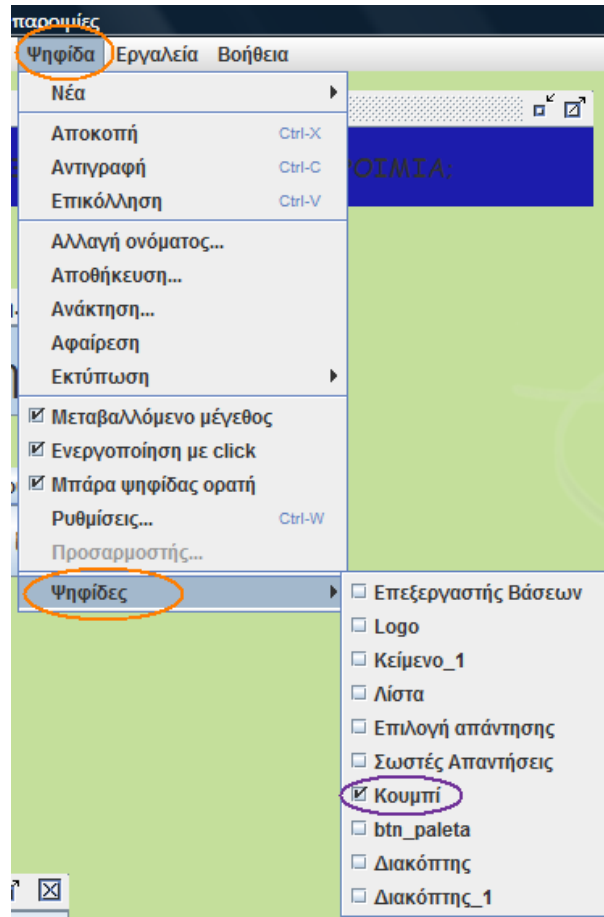
Πηγαίνουμε στο μενού «Εργαλεία» και επιλέγουμε «Γραμμή Εργασίας». Μόλις επιλεγθεί η γραμμή εργασίας στο κάτω μέρος της επιφάνειας του Αβακίου εμφανίζεται μία γραμμή με όλες τις ψηφίδες που έχουν χρησιμοποιηθεί στο συγκεκριμένο μικρόκοσμο (βλ. Εικόνα 3.5). Να σημειωθεί ότι στη γραμμή εργαλείων μπορούμε να δούμε και τις ψηφίδες που έχουν ελαχιστοποιηθεί και δεν εμφανίζονται στην επιφάνεια εργασίας του Αβακίου. Έτσι στην προκειμένη περίπτωση οι ψηφίδες που βρίσκονται στην επιφάνεια εργασίας του Αβακίου είναι έξι, ενώ οι ψηφίδες που έχουν χρησιμοποιηθεί είναι 10. Στη συνέχεια κάνουμε δεξί κλικ σε οποιοδήποτε σημείο στη γραμμή εργασίας και από το μενού που εμφανίζεται επιλέγουμε «εμφάνιση μπαρών» και παρατηρούμε ότι όλες οι ψηφίδες του μικρόκοσμου έχουν στο πάνω μέρος μία μπάρα.



Εικόνα 3.5: Γραμμή Εργασίας του Αβακίου και προβολή των μπαρών των ψηφίδων

Ένας άλλος τρόπος για να δούμε όλες τις ψηφίδες (ορατές και κρυφές) που έχουν χρησιμοποιηθεί στο μικρόκοσμο είναι να πάμε στο μενού «Ψηφίδα» και να επιλέξουμε «Ψηφίδες» (βλ. Εικόνα 3.6). Από το μενού που προβάλλεται επιλέγουμε την ψηφίδα που

μας ενδιαφέρει και με αυτό τον τρόπο την κάνουμε ενεργή. Επίσης στο ίδιο μενού φαίνεται και η ψηφίδα που είναι ήδη ενεργή τη συγκεκριμένη στιγμή διότι έχει ένα ν δίπλα στο όνομά της (στην προκειμένη περίπτωση ενεργή είναι η ψηφίδα που έχει το όνομα «Κουμπί»)



Εικόνα 3.6: Εμφάνιση των ψηφίδων του μικρόκοσμου από το μενού Ψηφίδα

### 3.3.1 Τι σημαίνει ότι μία ψηφίδα είναι ενεργή;

Όταν κάνουμε μία ψηφίδα ενεργή είναι σαν να την καλούμε να ξεχωρίσει από τις άλλες ή για να χρησιμοποιήσουμε μία μεταφορά είναι σαν να της ζητάμε να κάνει ένα βήμα μπροστά από τις υπόλοιπες χωρίς όμως να εκτελέσει την ή τις λειτουργίες για τις οποίες έχει προγραμματιστεί. Μία από όλες τις ψηφίδες του μικρόκοσμου είναι κάθε φορά ενεργή και μας δίνει τη δυνατότητα να δούμε και να επεξεργαστούμε τις ιδιότητές της χωρίς να ενεργοποιήσουμε απαραίτητα και το μηχανισμό της (π.χ. πατώντας ένα κουμπί να κρύβεται μία ψηφίδα ή να εμφανίζεται ένα κείμενο με πληροφορίες).

### 3.3.2 Γιατί είναι σημαντικό να εμφανίσουμε τη μπάρα των ψηφίδων;

Διότι έτσι θα μάθουμε το όνομά της και θα μπορέσουμε να βρούμε το «μηχανισμό» που είναι συνδεδεμένος με την ενεργοποίησή της. Επίσης η μπάρα μας επιτρέπει α) να μετακινούμε την ψηφίδα και β) μπορούμε κάνοντας κλικ πάνω στη μπάρα να κάνουμε ενεργή την ψηφίδα. [Αν κάνουμε κλικ πάνω στην ψηφίδα και όχι πάνω στη μπάρα της, το πιθανότερο είναι ότι θα θέσουμε σε λειτουργία το μηχανισμό της. Φανταστείτε λοιπόν την περίπτωση όπου με το κλικ σε μία ψηφίδα (στο μικρόκοσμο παροιμίες αυτό συμβαίνει με την ψηφίδα που γράφει πάνω της Παλέτα) να κλείνει ο συγκεκριμένος μικρόκοσμος και να εμφανίζεται νέος. Σε αυτή την περίπτωση δεν θα μπορούσαμε ποτέ να «ανοίξουμε» τη συγκεκριμένη ψηφίδα].

#### ΠΡΟΣΟΧΗ:

##### Όνομα ψηφίδας vs κείμενο ψηφίδας

Το όνομα της ψηφίδας είναι εκείνο που φαίνεται πάνω στη μπάρα της ψηφίδας και όχι το κείμενο που μπορεί να φαίνεται πάνω στην επιφάνεια της ψηφίδας. Έτσι για παράδειγμα η ψηφίδα που γράφει πάνω της ΕΝΑΡΞΗ στο μικρόκοσμο paroimies-manual.mwd ονομάζεται κουμπί. Ακολουθήστε τη διαδικασία που αναφέρθηκε πιο πάνω με την εμφάνιση των μπαρών των ψηφίδων για να το διαπιστώσετε.

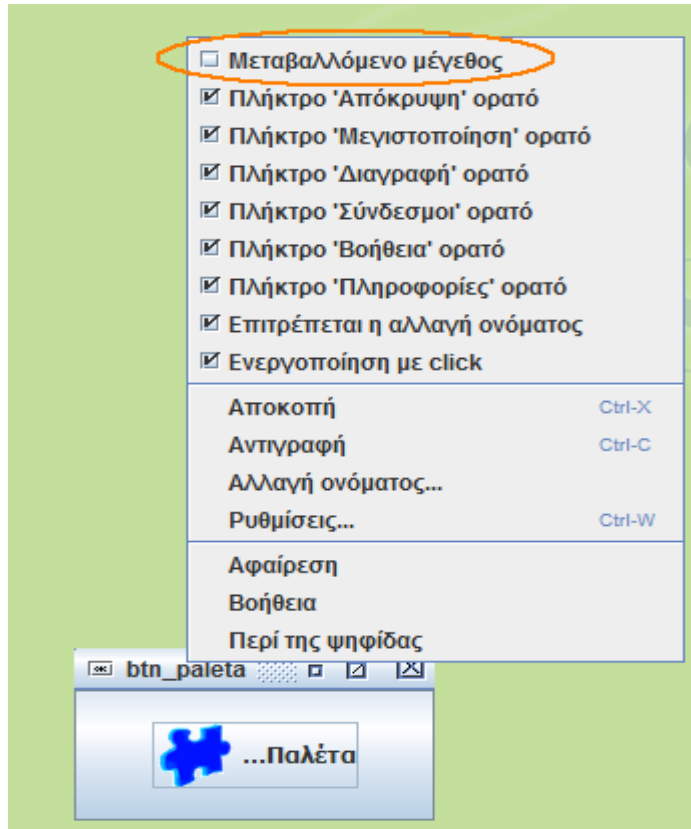
##### Όνομα ψηφίδας: Αλλαγή

Αλλάζουμε το όνομα μίας ψηφίδας κάνοντας κλικ πάνω στη μπάρα της και επιλέγουμε αλλαγή ονόματος. Στη συνέχεια στο παράθυρο που εμφανίζεται γράφουμε το όνομα που θέλουμε βλ. ενότητα 4.5.1 για το πώς καλούμε τις ψηφίδες με το όνομά τους.

##### Αλλαγή μεγέθους ψηφίδας

Ενδέχεται κάποια ψηφίδα να έχει μικρύνει πολύ και να μην φαίνεται το όνομά της. Σε αυτή την περίπτωση αλλάζουμε το μέγεθος της ψηφίδας πηγαίνοντας το ποντίκι μας σε κάποια από τις πλευρές της ή τις γωνίες της ώστε να μεταφερθεί σε ένα βέλος με δύο άκρες.

Δοκιμάστε να αλλάξετε το μέγεθος της ψηφίδας που γράφει πάνω της (δεν είναι το όνομά της αυτό) Παλέτα. Αν πάτε στην άκρη της ψηφίδας θα διαπιστώσετε ότι δεν είναι δυνατόν να της αλλάξετε το μέγεθος. Αυτό συμβαίνει γιατί έχει απενεργοποιηθεί η ιδιότητα «Μεταβαλλόμενο μέγεθος» Για να το αλλάξετε πάλι θα πρέπει να κάνετε δεξί κλικ πάνω στη μπάρα της ψηφίδας και στο μενού που θα εμφανιστεί κάνετε κλικ στο κουτάκι που είναι δίπλα στη φράση Μεταβαλλόμενο μέγεθος (βλ. Εικόνα 3.7)



Εικόνα 3.7: Ενεργοποίηση της δυνατότητας αλλαγής μεγέθους της ψηφίδας.

Για αναλυτική περιγραφή σχετικά με τις ιδιότητες των ψηφίδων τη μεμονωμένη ή τη συνολική διαχείρισή τους βλ. «Εγχειρίδιο Χρήσης Αβακίου II» ενότητες 3.10 Διαχείριση μπάρας ψηφίδας και ενότητα 3.11 Διαχείριση της Μπάρας όλων των ψηφίδων του μικρόκοσμου σελ. (25-29).

### **3.4 Πώς αποκτάμε πρόσβαση στο «μηχανισμό» του μικρόκοσμου.**

Λέξεις κλειδιά Ψηφίδα Logo, Γεγονότα, Διαχείριση Ψηφίδων – Παράθυρο Διαχειριστή προγραμμάτων ενεργειών ψηφίδων.

Μικρόκοσμος: paroimies\_manual.mwd

#### **3.4.1 Ψηφίδα Logo και Γεγονότα ή Διαχείριση προγραμμάτων ενεργειών ψηφίδων**

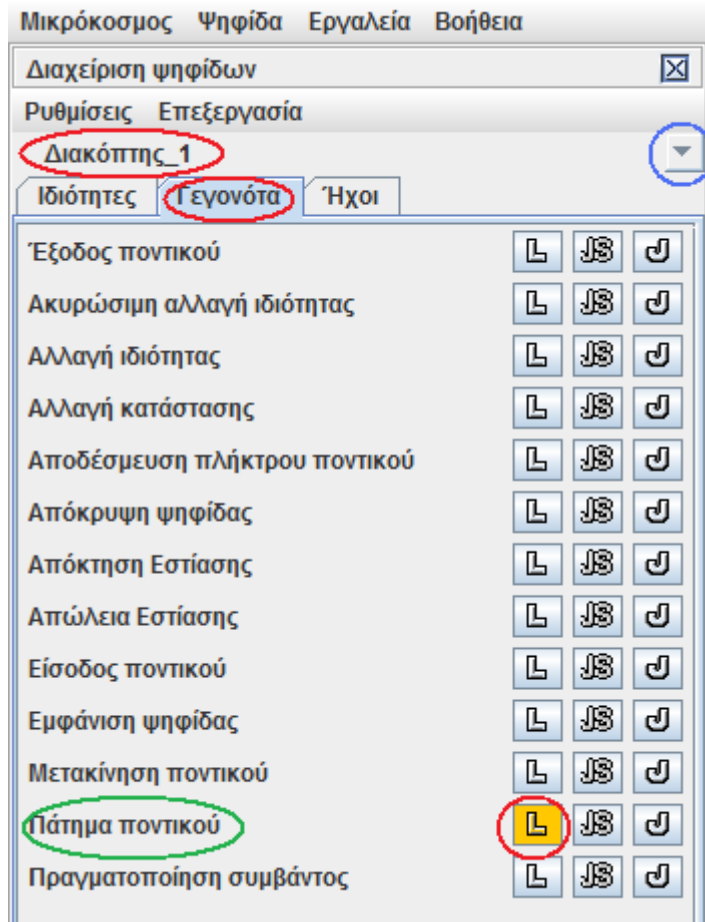
Για να μπορέσουμε να μαστορέψουμε ένα μικρόκοσμο (να πάρουμε μέρη του μικρόκοσμου, να προσθέσουμε, να αλλάξουμε ή να προσαρμόσουμε) είναι σημαντικό να μπορούμε να δούμε ποιος είναι ο μηχανισμός που κάνει κάποιες ψηφίδες του μικρόκοσμου να λειτουργούν με το συγκεκριμένο τρόπο π.χ. με το πάτημα ενός κουμπιού να ανοίγει ένα νέο μικρόκοσμο, ή να εμφανίζει και να κρύβει ένα κείμενο ή να κάνει πιο σύνθετα πράγματα π.χ. να αποθηκεύει ή να επεξεργάζεται δεδομένα κλπ.

Οι ψηφίδες του Αβακίου έχουν ήδη κάποιες λειτουργικότητες έχουν ήδη δηλ. προγραμματιστεί να κάνουν κάποια πράγματα και να συνεργάζονται με άλλες ψηφίδες αλλά το Αβάκιο δίνει τη δυνατότητα στους χρήστες να επεκτείνουν και να εμπλουτίσουν αυτές τις λειτουργικότητες χρησιμοποιώντας γι' αυτό το σκοπό εντολές που μπορούν να γράψουν είτε στην ψηφίδα **Logo** είτε στα **γεγονότα** της κάθε ψηφίδας (Για αναλυτική παρουσίαση αυτής της λογικής που έχει ονομαστεί "Black and white box approach" βλ. (Kynigos, 2004)).

**Τι είναι τα γεγονότα των ψηφίδων:** Τα γεγονότα είναι μία καθορισμένη σειρά από ενέργειες για κάθε ψηφίδα (πχ. Πάτημα ποντικού βλ. Εικόνα 3.8 ) τα οποία μπορούμε να χρησιμοποιήσουμε για να προσθέσουμε λειτουργικότητες στις ψηφίδες περιγράφοντας τι θα συμβαίνει όταν ενεργοποιείται το συγκεκριμένο γεγονός πάνω στην ψηφίδα (π.χ. **όταν κάνω κλικ με το ποντίκι πάνω στην ψηφίδα διακόπτης [Γεγονός: πάτημα ποντικού ] να εμφανίζεται ένα κείμενο με πληροφορίες [κώδικας που προσθέτει ο χρήστης] [βλ. ενότητα ppp].**

Κατά συνέπεια αν θέλουμε να δούμε ποιες είναι οι λειτουργικότητες που έχει προσθέσει ο κατασκευαστής του μικρόκοσμου `paroiimies_manual.mwd` ή οποιουδήποτε μικρόκοσμου γενικότερα, θα πρέπει να αναζητήσει την ψηφίδα Logo από τη «Γραμμή Εργασίας» και/ή να ανοίξει τα **γεγονότα** της ψηφίδας που εκτελεί τη λειτουργικότητα που μας ενδιαφέρει: Στην προκειμένη περίπτωση θέλουμε να δούμε με ποιόν τρόπο πιέζοντας το Διακόπτη\_1 που γράφει Οδηγίες, εμφανίζεται το κείμενο με τις οδηγίες χρήσης του μικρόκοσμου. Για να δούμε τις εντολές που περιγράφουν τη λειτουργία αυτή θα πρέπει να κάνουμε τα ακόλουθα βήματα:

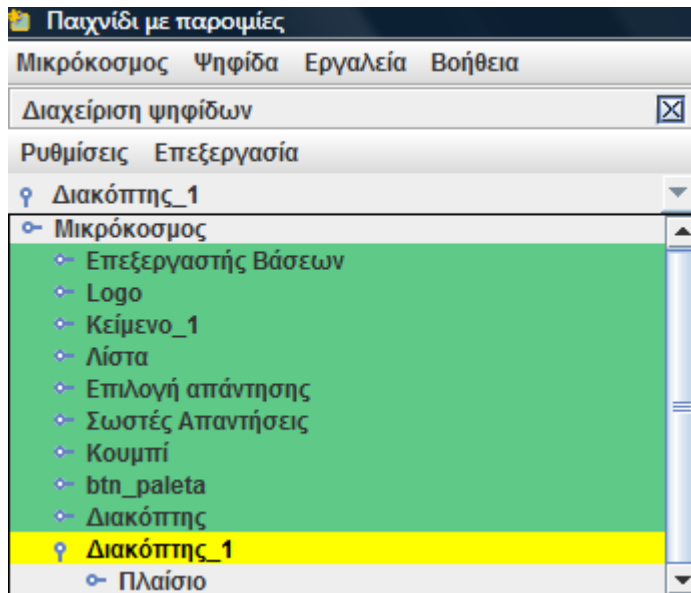
Κάνουμε ενεργή την ψηφίδα (με κλικ πάνω στη μπάρα της) που μας ενδιαφέρει στην προκειμένη πρόκειται για την ψηφίδα με το όνομα «Διακόπτης\_1» που γράφει πάνω της «ΟΔΗΓΙΕΣ», πάμε στο μενού Εργαλεία και επιλέγουμε Διαχείριση Ψηφίδων. Θα εμφανιστεί στο αριστερό μέρος της επιφάνειας εργασίας του Αβακίου ένα παράθυρο όπως αυτό που φαίνεται στην Εικόνα 3.8



Εικόνα 3.8: Προβολή των γεγονότων της ψηφίδας Διακόπτης\_1

Προσοχή: βεβαιωνόμαστε ότι έχουμε τη σωστή ψηφίδα κοιτάζοντας το όνομα που προβάλλεται στον πρώτο κόκκινο κύκλο (στην προκειμένη περίπτωση «Διακόπτης\_1»). Αν για οποιονδήποτε λόγο στο σημείο αυτό προβάλλεται το όνομα μίας άλλης ψηφίδας τότε πιέζουμε το βελάκι που βρίσκεται στο μπλε κύκλο και προβάλλεται όλη η λίστα με τις ψηφίδες που χρησιμοποιούνται στο μικρόκοσμο (βλ. Εικόνα 3.9) και από κει διαλέγουμε την ψηφίδα της οποίας της ιδιότητες θέλουμε να επεξεργαστούμε. Από τις τρεις καρτέλες που βρίσκονται κάτω από το όνομα της ψηφίδας (Ιδιότητες, Γεγονότα, Ήχοι) επιλέγουμε «Γεγονότα»

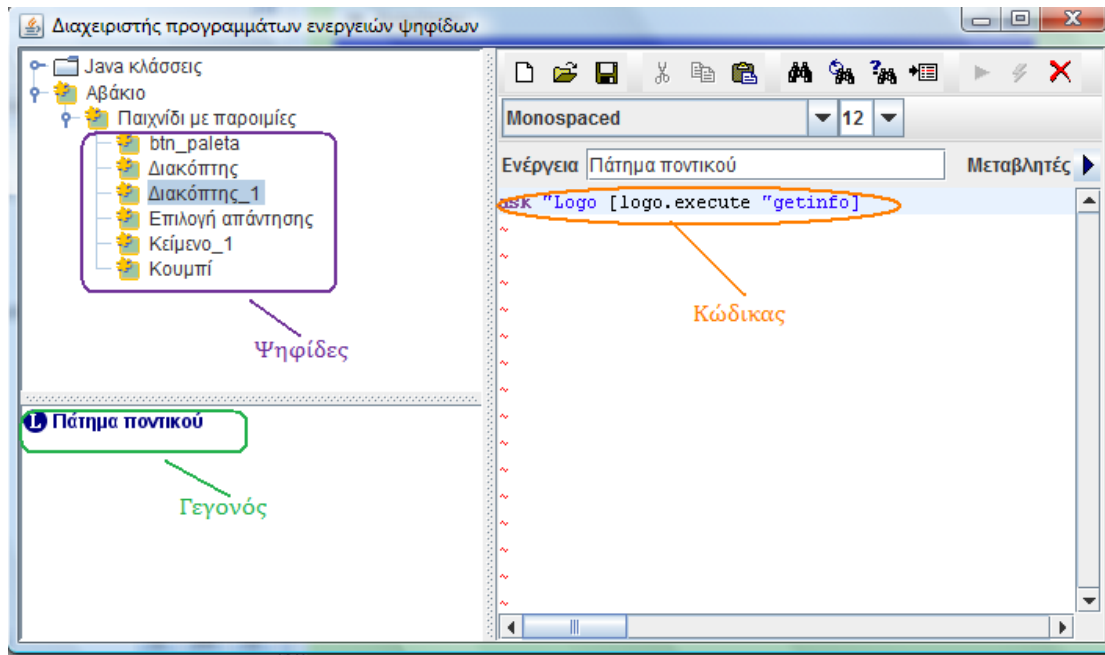




Εικόνα 3.9 Λίστα με τις ψηφίδες του μικρόκοσμου μέσα από το παράθυρο της διαχείρισης ψηφίδων

Αν η ψηφίδα που επιλέξαμε έχει πρόσθετες λειτουργικότητες σε κάποιο από τα γεγονότα της, τότε οι εντολές σε γλώσσα Logo είναι καταχωρημένες πίσω από το κουμπί L το οποίο έχει πορτοκαλί χρώμα (βλ. Εικόνα 3.8 τελευταίος κόκκινος κύκλος). Στην προκειμένη περίπτωση ο κώδικας αφορά το γεγονός « Πραγματοποίηση Συμβάντος» που σημαίνει πως αν γίνει κάτι πάνω στην ψηφίδα όπως π.χ. κλικ τότε θα ενεργοποιηθούν οι εντολές που είναι καταχωρημένες σε αυτό το γεγονός. Τα υπόλοιπα γεγονότα που τα κουμπιά τους είναι γκριζα δεν έχουν εντολές από πίσω τους.

Για να δούμε τις εντολές που σχετίζονται με τον γεγονός «Πραγματοποίηση συμβάντος» κάνουμε κλικ πάνω στο πορτοκαλί «L» και εμφανίζεται το παράθυρο με τίτλο «Διαχειριστής προγραμμάτων ενεργειών ψηφίδων» (βλ. Εικόνα 3.10)



Εικόνα 3.10 Το παράθυρο του διαχειριστή προγραμμάτων ενεργειών ψηφίδων

Όπως φαίνεται στην Εικόνα 3.10 ο κώδικας που περιγράφει τη λειτουργία της ψηφίδας βρίσκεται στο δεξί μέρος του παραθύρου, η ψηφίδα η οποία «περιέχει» αυτό το κομμάτι του κώδικα είναι στο πάνω αριστερό μέρος του παραθύρου και είναι επιλεγμένη (εδώ η ψηφίδα Διακόπτης\_1), ενώ το γεγονός με το οποίο ενεργοποιείται ο συγκεκριμένος κώδικας βρίσκεται στο κάτω αριστερά μέρος του παραθύρου.

Από το συγκεκριμένο παράθυρο του Διαχειριστή προγραμμάτων ενεργειών ψηφίδων μπορούμε να έχουμε πρόσβαση στον κώδικα που συνδέεται με τα γεγονότα όλων των ψηφίδων του μικρόκοσμου. Αυτό σημαίνει ότι αν για παράδειγμα κάνω κλικ στην ψηφίδα Διακόπτης στο πάνω αριστερό μέρος του παραθύρου του Διαχειριστή προγραμμάτων ενεργειών ψηφίδων τότε στο δεξί μέρος του ίδιου παραθύρου θα προβληθεί – αν υπάρχει – ο κώδικας που είναι συνδεδεμένος με τα γεγονότα της ψηφίδας Διακόπτης μαθητών.

### 3.4.2 Γιατί είναι χρήσιμο το παράθυρο του Διαχειριστή προγραμμάτων ενεργειών ψηφίδων;

Γιατί μας δίνει τη δυνατότητα να έχουμε πρόσβαση στον κώδικα που είναι συνδεδεμένος με τα γεγονότα όλων των ψηφίδων του κάθε μικρόκοσμου χωρίς να χρειάζεται κάθε φορά να κάνουμε τη διαδικασία που περιγράφηκε νωρίτερα για κάθε μία ψηφίδα. Να ενεργοποιούμε δηλ. μία-μία τις ψηφίδες, να πηγαίνουμε, στο μενού εργαλεία, να επιλέγουμε Διαχείριση Ψηφίδων να βλέπουμε αν κάποιο από τα γεγονότα της ψηφίδας έχει κώδικα και να συνεχίζουμε έτσι για όλες τις ψηφίδες του μικρόκοσμου.

#### 3.4.2.1 Σημαντικές διευκρινήσεις για τα γεγονότα των ψηφίδων

1. Δεν είναι απαραίτητο όλες οι ψηφίδες να έχουν κάποιο κώδικα στα γεγονότα τους που να τους προσθέτει λειτουργικότητες

2. Ενδέχεται κάποιοι μικρόκοσμοι να μην έχουν κώδικα γραμμένο στην ψηφίδα Logo, αυτό δεν σημαίνει ότι δεν μπορούν να λειτουργήσουν.
3. Ενδέχεται κάποιοι μικρόκοσμοι να μην έχουν κώδικα γραμμένο στα γεγονότα κάποιων από τις ψηφίδες τους.
4. Δεν είναι απαραίτητο να προσθέσει κανείς κώδικα σε όλα τα γεγονότα μίας ψηφίδας, συνήθως χρησιμοποιούνται ένα ή δύο γεγονότα
5. Τα γεγονότα είναι συγκεκριμένα για κάθε ψηφίδα (δεν μπορούμε να προσθέσουμε νέα) εμείς προσθέτουμε τον κώδικα που λέει στην ψηφίδα τι να κάνει αν ενεργοποιηθεί το γεγονός
6. Ο κώδικας που γράφουμε σε κάποιο από τα γεγονότα μίας ψηφίδας μπορεί να αφορά είτε την ίδια την ψηφίδα (π.χ. αν κάνω κλικ με το ποντίκι στην ψηφίδα διακόπτης να αλλάζει το χρώμα του φόντου της ψηφίδας και από κόκκινο γίνει πορτοκαλί) είτε άλλες ψηφίδες (π.χ. αν κάνω κλικ με το ποντίκι στην ψηφίδα διακόπτης εμφάνισε την ψηφίδα κείμενο) είτε τον ίδιο ή κάποιον άλλο μικρόκοσμο (π.χ. αν κάνω κλικ στην ψηφίδα διακόπτης άνοιξε το μικρόκοσμο τάδε).
7. Ο κώδικας γράφεται πάντοτε πάνω στην ψηφίδα στην οποία συμβαίνει το γεγονός και όχι στην ψηφίδα η οποία είναι το αποτέλεσμα του γεγονότος: έτσι, όταν κάνοντας κλικ με το ποντίκι πάνω στην ψηφίδα «Διακόπτης» εμφανίζεται ένα κείμενο με οδηγίες, ο κώδικας είναι γραμμένος πάνω στην ψηφίδα διακόπτης και όχι στην ψηφίδα κείμενο γιατί πάνω στην ψηφίδα διακόπτης γίνεται το κλικ με το ποντίκι.:

#### **Συνοπτικά:**

Για να αποκτήσουμε πρόσβαση σε αυτό που ονομάσαμε μηχανισμό του μικρόκοσμου ψάχνουμε α) για την ψηφίδα Logo από τη Γραμμή εργασίας των ψηφίδων και β) τα γεγονότα της ψηφίδας που εκτελεί μία λειτουργία που μας ενδιαφέρει (π.χ. εμφάνιση πληροφοριών με το κλικ του ποντικού πάνω στην ψηφίδα διακόπτης). Από το παράθυρο του Διαχειριστή προγραμμάτων ενεργειών ψηφίδων μπορούμε να δούμε τον κώδικα που είναι γραμμένος σε όλα τα γεγονότα των ψηφίδων.

**Μικρή συμβουλή:** Οι μικρόκοσμοι που έχουν κουμπιά ή διακόπτες συνήθως έχουν κώδικα στα γεγονότα αυτών των ψηφίδων.

### **3.4.3 Logo, Γεγονότα ή και τα δύο: πώς ξέρουμε πού να ψάξουμε;**

Για να μην ψάχνουμε στα τυφλά όταν επιχειρούμε να ανοίξουμε ένα μικρόκοσμο χρειάζεται να γνωρίζουμε ποια είναι η λογική της χρήσης της ψηφίδας Logo και ποια η λογική της χρήσης των Γεγονότων καθώς και τις ιδιαιτερότητές τους. Όπως αναφέρθηκε ήδη τόσο στην ψηφίδα Logo όσο και στο Διαχειριστή προγραμμάτων ενεργειών ψηφίδων (εις το εξής ΔΠΕΨ) μπορούμε να γράψουμε εντολές που εμπλουτίζουν τις λειτουργικότητες των ψηφίδων και του μικρόκοσμου γενικότερα.

## Εργαστήριο εκπαιδευτικής Τεχνολογίας

Όταν γράφουμε κώδικα στο ΔΠΕΨ, ουσιαστικά γράφουμε εντολές πάνω στην κάθε ψηφίδα ξεχωριστά. Επίσης οι εντολές που γράφουμε είναι χύμα, η μία κάτω από την άλλη χωρίς δομή. Αντίθετα η ψηφίδα Logo μας επιτρέπει να γράφουμε τις εντολές που θέλουμε να εκτελεστούν δομημένες και οργανωμένες σε διαδικασίες (βλ. ενότητα 3.4.3.2):

Έτσι λοιπόν όταν θέλουμε να βρούμε πώς λειτουργεί ένας μικρόκοσμος χρειάζεται να σκεφτούμε ως εξής: α)προσδιορίζουμε τη λειτουργία που μας ενδιαφέρει π.χ. το πάτημα ενός κουμπιού ή η επιλογή ενός στοιχείου από τη λίστα combo box, η εμφάνιση επεξηγηματικού κειμένου με το πέρασμα του ποντικού πάνω από μία ψηφίδα β)προσδιορίζουμε ποια είναι η ψηφίδα που εκτελεί τη συγκεκριμένη λειτουργία (π.χ. ένα κουμπί, μία ψηφίδα combo box κλπ. γ) ψάχνουμε να βρούμε αν η ψηφίδα αυτή έχει στα γεγονότα της κώδικα που προσδιορίζει τη λειτουργικότητά της ή αν παραπέμπει στην ψηφίδα Logo (τότε στον κώδικα που καταγράφεται στα γεγονότα χρησιμοποιείται η ακόλουθη εντολή (ask "Logo [logo.execute βλ. ανάλυση παραδείγματος) στη δεύτερη περίπτωση πηγαίνουμε στην ψηφίδα Logo και ψάχνουμε να βρούμε τη διαδικασία που περιγράφει τη συγκεκριμένη λειτουργία (βλ. ανάλυση παραδείγματος). Συχνά ο κώδικας που είναι γραμμένος στα γεγονότα μίας ψηφίδας παραπέμπει σε διαδικασία που είναι γραμμένη στην ψηφίδα Logo.

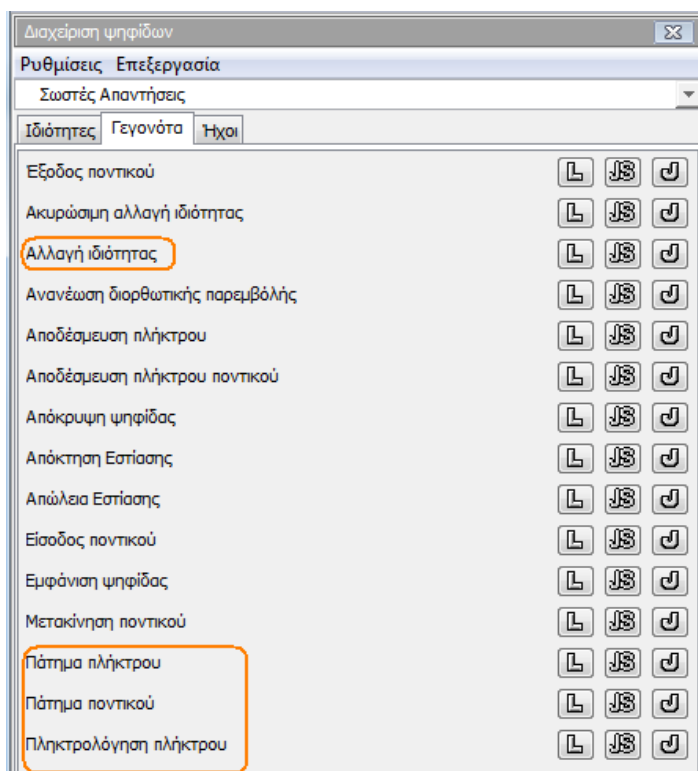
Για να γίνει αυτό πιο ξεκάθαρο ας το δούμε μέσα από ένα παράδειγμα:

### 3.4.3.1 Ανάλυση παραδείγματος: Από τα Γεγονότα μίας ψηφίδας στην ψηφίδα Logo

Ανοίγουμε το μικρόκοσμο «paroiimies\_manual.mwd» και ακολουθούμε τις οδηγίες που δίνονται στο μικρόκοσμο για να παίξουμε το παιχνίδι. Παίζοντας το παιχνίδι παρατηρούμε πως όταν επιλέγουμε μία λάθος παροιμία, αυτή δεν καταγράφεται στις σωστές απαντήσεις και το ερώτημα είναι πώς γίνεται αυτό. Για να μπορέσουμε να το απαντήσουμε χρειάζεται κατ' αρχάς να γνωρίζουμε σε γενικές γραμμές ποιες είναι οι δεδομένες λειτουργικότητες των ψηφίδων: στη συγκεκριμένη περίπτωση της ψηφίδας Λίστα combo box (ή λίστα επιλογής): Η συγκεκριμένη ψηφίδα μας δίνει τη δυνατότητα να καταχωρήσουμε μία σειρά από στοιχεία και στη συνέχεια να μπορούμε να επιλέγουμε όποιο θέλουμε από αυτά. Παρατηρούμε εδώ ότι δεν υπάρχει κάποια λειτουργικότητα που να δείχνει ποια επιλογή είναι σωστή και ποια όχι. Άρα υποθέτουμε ότι αυτή είναι μία λειτουργικότητα που προστέθηκε από τον κατασκευαστή του μικρόκοσμου και την περιγραφή της θα πρέπει να την ψάξουμε ή στα γεγονότα ή στην ψηφίδα Logo. Επειδή η επιλογή ενός στοιχείου από τα υπάρχοντα είναι ένα γεγονός (π.β. «πραγματοποίηση συμβάντος») που θα μπορούσε να ενεργοποιεί μία διαδικασία ελέγχου αν το στοιχείο που επιλέχθηκε είναι σωστό ή λάθος κατευθυνόμαστε στο να ελέγξουμε τα γεγονότα της ψηφίδας.

Αλλά ποιός ψηφίδας; Αρχικά εμφανίζουμε τις μπάρες των ψηφίδων του μικρόκοσμου με τον τρόπο που έχουμε αναφέρει ήδη (βλ. ενότητα 3.3). Οι υποψήφιες ψηφίδες είναι οι ακόλουθες: «Λίστα», «Επιλογή απάντησης», «Σωστές απαντήσεις». Ας δούμε τι κάνει κάθε μία από αυτές: Η ψηφίδα Λίστα προβάλλει τις 5 εναλλακτικές της παροιμίας. Στην

ψηφίδα «Επιλογή απάντησης» ο χρήστης επιλέγει εκείνη που θεωρεί σωστή και στην ψηφίδα «Σωστές απαντήσεις» προβάλλονται μόνο οι σωστές παροιμίες. Από δω θα μπορούσε κανείς να υποθέσει ότι το φιλτράρισμα της σωστής απάντησης γίνεται κατά την επιλογή της παροιμίας (με το που επιλέγεται η παροιμία ενεργοποιείται ο έλεγχος) ή πριν την προβολή της στην ψηφίδα «Σωστές απαντήσεις» (π.χ. Αν το κείμενο είναι σωστό προβάλλεται αν όχι δεν προβάλλεται). Ας ξεκινήσουμε από την τελευταία υπόθεση: Ανοίγουμε το διαχειριστή ψηφίδων, επιλέγουμε την ψηφίδα «Σωστές Απαντήσεις» (βλ. ενότητα 3.4.1) και πηγαίνουμε στα γεγονότα της ψηφίδας. Παρατηρούμε ότι δεν υπάρχει κώδικας σε κανένα από τα γεγονότα αυτής της ψηφίδας. Για να καταλάβουμε γιατί η λειτουργικότητα του ελέγχου δεν έχει συνδεθεί με τα γεγονότα της συγκεκριμένης ψηφίδας χρειάζεται να δούμε ποια είναι τα διαθέσιμα γεγονότα και να ελέγξουμε αν θα μπορούσαν να ενεργοποιήσουν μία διαδικασία ελέγχου:

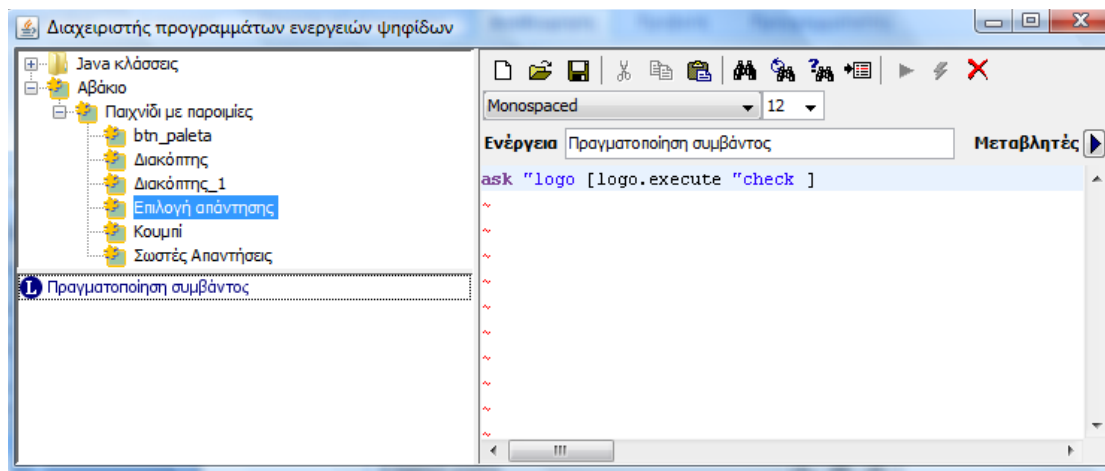


**Εικόνα 3.11: Τα γεγονότα της ψηφίδας «Σωστές Απαντήσεις»**

Από τα διαθέσιμα γεγονότα έχουν τοποθετηθεί σε πορτοκαλί κάποια γεγονότα που χρησιμοποιούνται ευρέως. Στην περίπτωση όμως που εξετάζουμε ούτε πάτημα κάποιου πλήκτρου γίνεται (γιατί το κείμενο μπαίνει αυτόματα στην ψηφίδα) ούτε πάτημα ποντικού ούτε πληκτρολόγηση πλήκτρου, αλλά ούτε και κάποια αλλαγή της ιδιότητας (π.χ. αλλαγή του χρώματος των γραμμάτων) της ψηφίδας συμβαίνει. Ως εκ τούτου, τα γεγονότα της ψηφίδας αυτής, έτσι όπως χρησιμοποιείται σε αυτό το μικρόκοσμο δεν είναι κατάλληλα να ενεργοποιήσουν μία διαδικασία ελέγχου.

Η άλλη επιλογή που έχουμε είναι η ψηφίδα «Επιλογή Απάντησης». Κάνουμε κλικ πάνω στη μπάρα της ψηφίδας και στη συνέχεια επιλέγουμε Διαχείριση Ψηφίδων για να δούμε τα Γεγονότα της συγκεκριμένης ψηφίδας. Βλέπουμε λοιπόν ότι στο γεγονός

«Πραγματοποίηση συμβάντος» (στην προκειμένη είναι η επιλογή ενός από τα στοιχεία της λίστας combo box) δείχνει ότι έχει μέσα του κώδικα:



Εικόνα 3.12: Ο κώδικας για την πραγματοποίηση συμβάντος στην ψηφίδα «Επιλογή απάντησης»

Ο κώδικας λοιπόν που φαίνεται στην εικόνα είναι ο εξής: **Ask "logo [logo.execute "check]**. Η συγκεκριμένη γραμμή κώδικα θα μπορούσε να μεταφραστεί στη φυσική γλώσσα ως εξής: Κάθε φορά που θα πραγματοποιείται ένα συμβάν στην ψηφίδα «Επιλογή απάντησης» **ζήτησε από την ψηφίδα logo να εκτελέσει τη διαδικασία με το όνομα «check»**. Αυτό σημαίνει ότι στην ψηφίδα Logo υπάρχει μία διαδικασία με το όνομα check η οποία πραγματοποιεί τον έλεγχο για το αν η παροιμία που επιλέχθηκε είναι σωστή ή όχι και αντίστοιχα δίνει την εντολή για να καταχωρηθεί ή όχι η παροιμία στις σωστές απαντήσεις.

```

Logo
Αρχείο Επεξεργασία Αναζήτηση Μηχανή LOGO
Monospaced 12 E II

make "fb (DB.CELL :pos "mia)
LIST.ADDITEM (word :fa " | | :fb)
COMBOBOX.ADDITEM (word :fa " | | :fb)
]

end

*****

to check
repeat 100[
  make "a DB.CELL recount "paroi
  make "al DB.CELL recount "mia
  make "b ask "|Επιλογή απάντησης| [COMBOBOX.SELECTEDITEM]
  make "c (word :a " | | :al)
  if :c = :b [ask "|Σωστές Απαντήσεις| [AREA.APPEND (word :b "\n)]]
]
end

```

Εικόνα 3.13:Το περιεχόμενο της διαδικασίας check στην ψηφίδα Logo

Όπως φαίνεται λοιπόν από την παραπάνω εικόνα η οποία απεικονίζει την ψηφίδα Logo, υπάρχει μία διαδικασία με το όνομα check το περιεχόμενο της οποίας έχει σημειωθεί με το πράσινο πλαίσιο (περεταιίρω ανάλυση του κώδικα θα γίνει σε επόμενες ενότητες). Το παραπάνω παράδειγμα δόθηκε εδώ για να παρουσιαστεί κυρίως ο τρόπος με τον οποίο σκεφτόμαστε όταν θέλουμε να αποκτήσουμε πρόσβαση στο μηχανισμό ενός μικρόκοσμου. Στη συνέχεια δίνονται μερικές πρακτικές συμβουλές - οδηγίες

### Πρακτικές συμβουλές

Πολύ συχνά οι πρόσθετες λειτουργικότητες ενός μικρόκοσμου βρίσκονται και στα γεγονότα των ψηφίδων και στην ψηφίδα Logo. Σκεφτόμαστε όπως στο παράδειγμα για να βρούμε από πού θα ξεκινήσουμε.

Όταν βλέπετε την εντολή "logo.execute" στον κώδικα των γεγονότων μίας ψηφίδας αυτό σημαίνει ότι παραπέμπει σε διαδικασία που έχει οριστεί στην ψηφίδα logo.

### 3.4.3.2 Γλώσσα, Εντολές, διαδικασίες, κώδικας

Η γλώσσα προγραμματισμού Logo αποτελείται από εντολές και συντακτικούς κανόνες συνδυασμού τους

**Εντολές:** Είναι – κατ’ αντιστοιχία με τη φυσική γλώσσα- το λεξιλόγιο της γλώσσας Logo.

**Διαδικασίες:** Τις διαδικασίες τις φτιάχνουμε εμείς χρησιμοποιώντας τις εντολές. Μία διαδικασία περιγράφει ουσιαστικά το περιεχόμενο μίας νέας εντολής που δημιουργήσαμε εμείς. Με αυτόν τον τρόπο εμπλουτίζουμε έτσι το λεξιλόγιο της γλώσσας Logo. Είναι δηλ. σαν να ορίζουμε μία νέα λέξη στη φυσική γλώσσα όπου από τη στιγμή που την περιγράφουμε αντί να χρησιμοποιούμε στην επικοινωνία μας την περιγραφή χρησιμοποιούμε τη νέα λέξη. (αντί για παράδειγμα να λέμε ότι «η μάζα μιας ποσότητας υγρού μετατρέπεται σε αέριο με φουσαλίδες αερίου να δημιουργούνται σε όλο τον όγκο του υγρού (Wikipedia) λέμε ότι το υγρό βράζει) Το νέο λεξιλόγιο που δημιουργούμε ισχύει τοπικά, για ένα συγκεκριμένο μικρόκοσμο. Αν θέλουμε να το χρησιμοποιήσουμε και σε άλλους μικρόκοσμους πρέπει να το ορίσουμε από την αρχή.

Στο μικρόκοσμο *raoimies-manual* και στην ψηφίδα Logo υπάρχει μία διαδικασία με το όνομα «start» η οποία αποτελείται από μία σειρά εντολές του λεξιλογίου της Logo οι οποίες κάνουν τα εξής: επιλέγουν ό,τι έχει καταγραφεί στην ψηφίδα «Σωστές απαντήσεις» και το αποκόπτουν (cut), στη συνέχεια αφαιρούν ό,τι υπάρχει στην ψηφίδα «Λίστα» και κάνουν το ίδιο και στην ψηφίδα «Επιλογή Απάντησης». Με αυτό τον τρόπο, καθαρίζουν οι συγκεκριμένες ψηφίδες από προηγούμενες χρήσεις και μπορούμε να ξεκινήσουμε νέο παιχνίδι. Έτσι, κάθε φορά που θέλουμε να ξεκινήσουμε ένα καινούριο παιχνίδι, αντί να πληκτρολογούμε και να εκτελούμε όλες αυτές τις εντολές μία - μία, αρκεί να εκτελέσουμε την εντολή start. Εκτελώ σημαίνει ότι ζητάω από το Αβάκιο να κάνει ό,τι λέει η εντολή.

#### Σημαντικές διευκρινήσεις.

- Μία διαδικασία έχει πάντοτε ένα όνομα που δίνουμε εμείς και πρέπει να είναι αντίστοιχο της λειτουργίας που εκτελεί η διαδικασία (η νέα εντολή)
- Ο ορισμός μίας διαδικασίας είναι διαφορετικό πράγμα από την εκτέλεσή της. Όταν ορίζω μία διαδικασία περιγράφω ποιες ενέργειες περιλαμβάνει (ξεκινάω με το «για» (ή «το») και τελειώνω με το «τέλος» (ή end)). Όταν την εκτελώ, αρκεί να γράψω το όνομα της διαδικασίας (όχι όλες τις ενέργειες) και τότε ο υπολογιστής κάνει ότι περιγράφει η εντολή.



- Οι εντολές που δημιουργήσαμε μπορούν να χρησιμοποιηθούν ως λεξιλόγιο άλλων εντολών που δημιουργούμε (βλ. στο μικρόκοσμο Paroimies π.χ. ότι η εντολή shuffle (που σημαίνει «ανακάτεμα») χρησιμοποιείται στην διαδικασία paroimia)
- Ισχύς των εντολών που δημιουργούμε στη Logo Οι εντολές που δημιουργούμε ισχύουν για το μικρόκοσμο στον οποίο κατασκευάστηκαν. Αν δηλ. ανοίξει κάποιος το μικρόκοσμο paleta\_etl\_vo2 που βρίσκεται στους μικρόκοσμους του συνοδευτικού υλικού, και πληκτρολογήσει την εντολή shuffle τότε η Logo δεν θα ξέρει τι να κάνει στη συγκεκριμένη περίπτωση. Για να μπορέσει να χρησιμοποιηθεί και εκεί η εντολή shuffle θα πρέπει να αντιγράψουμε τη διαδικασία από το μικρόκοσμο paroimies-manual.mwd για να οριστεί και σε αυτό το μικρόκοσμο και στη συνέχεια μπορούμε να την εκτελέσουμε. Προσοχή: Αν η διαδικασία που αντιγράφουμε περιέχει ονόματα άλλων διαδικασιών που ορίσαμε εμείς και δεν ανήκουν στο αρχικό λεξιλόγιο της logo, τότε θα πρέπει να αντιγράψουμε εκ νέου αυτές τις διαδικασίες.

### **3.5 Μερικά βασικά στοιχεία για τον προγραμματισμό με Logo**

Πριν προχωρήσουμε στην περιγραφή ορισμένων μόνο στοιχείων για το προγραμματισμό του Αβακίου με τη γλώσσα προγραμματισμού Logo είναι χρήσιμο να αναφερθούμε στη λογική που υπαγορεύει αυτή τη διαδικασία: «ο προγραμματισμός είναι το μέσον για να χρησιμοποιήσουμε τους υπολογιστές έτσι ώστε να κάνουν ό,τι θέλουμε εμείς και όχι να κάνουν ό,τι θέλουν αυτοί!» (Harvey 1997)

Επειδή συχνά η Logo ταυτίζεται με τα γραφικά της χελώνας και τη διδασκαλία της στις πρώτες τάξεις του Γυμνασίου θεωρείται αφενός παρωχημένη και αφετέρου απλοϊκή δηλ. κάτι που αφορά μαθητές του γυμνασίου ή και του δημοτικού και όχι μεταπτυχιακούς φοιτητές ή εκπαιδευτικούς για παράδειγμα. Στην πραγματικότητα η Logo έχει πολύ ευρύτερη χρήση από τα γραφικά της χελώνας και εδώ χρησιμοποιείται κυρίως ως εργαλείο επικοινωνίας και δημιουργικής έκφρασης με το Αβάκιο. Οι λόγοι που έχει η επιλεχθεί η γλώσσα προγραμματισμού Logo για αυτή την επικοινωνία και όχι κάποια άλλη είναι δύο α) πως πρόκειται για μία γλώσσα που είναι ενώ είναι εύκολο να τη μάθει κανείς την ίδια στιγμή μπορεί να φτιάξει ιδιαίτερα σύνθετα προγράμματα (low threshold and no ceiling) και β) είναι μία ισχυρή γλώσσα η οποία- σε σύγκριση με άλλες γλώσσες- επιτρέπει να εστιάσει ο χρήστης στο πρόβλημα που επιχειρεί να λύσει χωρίς να ανησυχεί ιδιαίτερα για τους περιορισμούς της γλώσσας (Harvey 1997)

#### **3.5.1 Σύνταξη εντολών στη Logo**

**Μικρόκοσμοι:** Welcome.mwd, welcome2.mwd

**Λέξεις κλειδιά:** εντολή, Input, μηνύματα λάθους, λίστα ενεργειών, χρήση εισαγωγικού

Οι εντολές της Logo (το λεξιλόγιό της) συνδυάζονται μεταξύ τους με βάση κάποιους κανόνες (π.β. τους κανόνες συντακτικού για τη φυσική γλώσσα) πράγμα που σημαίνει ότι δεν ταιριάζουν όλα με όλα. Σε αυτή την ενότητα θα αναφερθούμε μόνο ενδεικτικά σε

## Εργαστήριο εκπαιδευτικής Τεχνολογίας

κάποιες εντολές που θεωρούμε βασικές για την κατανόηση των όσων θα αναφέρουμε αργότερα

Ανοίξτε το μικρόκοσμο welcome. Ο μικρόκοσμος αυτός αποτελείται από δύο ψηφίδες: την ψηφίδα Logo και την ψηφίδα Κείμενο

Πληκτρολογήστε στη Logo την ακόλουθη εντολή:

```
ask "Κείμενο [KEIMENO.ΘΕΣΕΚΕΙΜΕΝΟ "ΚΑΛΩΣΟΡΙΣΑΤΕ]
```

Γραμμή κώδικα 1

και στη συνέχεια πιέστε το πλήκτρο **INSERT** (όχι Enter) ή πιέστε το κουμπί με το play που βρίσκεται στο μενού εργαλείων της ψηφίδας Logo. Αυτό που θα συμβεί αν ακολουθήσατε σωστά τις οδηγίες είναι να προβληθεί ψηφίδα Κείμενο η λ. ΚΑΛΩΣΟΡΙΣΑΤΕ.

### Εντολές: η σημασία των κενών μεταξύ εντολών

Καθώς πληκτρολογείτε την παραπάνω γραμμή κώδικα<sup>5</sup> χρειάζεται να προσέξετε τα κενά μεταξύ των λέξεων που απαρτίζουν την εντολή. Τα κενά που υπάρχουν είναι τρία: ένα μετά την εντολή ask ένα μετά τη λ. Κείμενο και ένα μετά την εντολή KEIMENO.ΘΕΣΕΚΕΙΜΕΝΟ. Αν έχετε κάνει λάθος και έχετε παραλείψει κάποιο κενό ή έχετε προσθέσει κάποιο άλλο τότε αφενός δεν θα εμφανιστεί η λ. ΚΑΛΩΣΟΡΙΣΑΤΕ και στο κάτω μέρος της ψηφίδας Logo (στο δεύτερο μισό) θα παρουσιαστεί σφάλμα όπως το επόμενο: ***I don't know how to ask"Κείμενο***. Το σφάλμα αυτό παρουσιάστηκε διότι έχει σβηστεί το κενό μετά την εντολή ask. Το κενό στη Logo σηματοδοτεί ότι ακολουθεί νέα λέξη (εντολή ή είσοδος (Input)). Έτσι λοιπόν στην περίπτωση μας η Logo εκλαμβάνει ως μία λέξη το ***ask"Κείμενο*** και δεν το αναγνωρίζει ως εντολή. Ο λόγος που δεν το αναγνωρίζει είναι διότι ξέρει τι σημαίνει ή τι πρέπει να κάνει όταν βρίσκει την εντολή ask αλλά δεν ξέρει τι σημαίνει η λ. ask"Κείμενο την οποία όπως αναφέρθηκε νωρίτερα τη θεωρεί ως μία ενιαία λέξη. Για να κατανοήσει κανείς αυτή τη συμπεριφορά θα πρέπει να έχει υπ' όψιν του ότι οι υπολογιστές κάνουν ακριβώς αυτό που τους λέμε ή που γράφουμε, όχι αυτό που εννοούμε ή που έχουμε στο μυαλό μας και συχνά οι συνομιλητές μας στη φυσική γλώσσα μπορούν να το συνάγουν. Άλλο ένα στοιχείο όταν δουλεύουμε όχι μόνο με τη γλώσσα προγραμματισμού αλλά και γενικότερα με τους υπολογιστές είναι να μάθουμε να δίνουμε προσοχή και να ερμηνεύουμε τα μηνύματα λάθους. Έτσι όταν η Logo εμφανίζει ένα μήνυμα όπως το προηγούμενο το πιθανότερο είναι πως έχουμε κάνει κάποιο συντακτικό λάθος.

### Εντολές: Χρήση εισαγωγικών

Άλλο ένα σημείο που χρειάζεται προσοχή είναι ότι τα εισαγωγικά πριν από τις λ. "Κείμενο ή "ΚΑΛΩΣΟΡΙΣΑΤΕ δεν κλείνουν.

---

<sup>5</sup> Με τη λ. κώδικα εννοούμε το συνδυασμό ενός ή περισσότερων εντολών οι οποίες όμως υλοποιούν μία ή περισσότερες λειτουργίες (κάνουν δηλ. κάτι, δεν είναι γραμμένες τυχαία).

Ερμηνεία του κώδικα: στο ερώτημα τι κάνει αυτή η γραμμή κώδικα μία πιθανή απάντηση θα ήταν ότι γράφει καλωσορίσατε. Αυτή η απάντηση όμως είναι αρκετά ελλιπής σε σχέση με αυτό που πραγματικά γίνεται. Μία ακριβής ερμηνεία του κώδικα αυτού είναι πως η Logo ζητάει από την ψηφίδα με το όνομα κείμενο να θέσει ως κείμενο (ή αλλιώς να γράψει) τη λ. ΚΑΛΩΣΟΡΙΣΑΤΕ. Γιατί η πρώτη ερμηνεία είναι προβληματική στην επικοινωνία με τον υπολογιστή; Η απάντηση είναι διότι δεν δηλώνει ποιος πρέπει να υλοποιήσει την εμφάνιση της λ.ΚΑΛΩΣΟΡΙΣΑΤΕ

Ας εξηγήσουμε λίγο τη λογική των εντολών. Η πρώτη εντολή του κώδικα που γράψαμε είναι η εντολή "Ask (δεν έχει σημασία αν το όνομα της εντολής είναι κεφαλαίο ή μικρό). Οι εντολές στις γλώσσες προγραμματισμού συχνά για να συμπληρωθούν χρειάζονται κάποια δεδομένα (input)<sup>6</sup> και σε ορισμένες περιπτώσεις εξάγουν και δεδομένα (output).

Έτσι λοιπόν ο χρήστης δεν αρκεί να γνωρίζει απλώς το όνομα της εντολής, πρέπει να ξέρει αν χρειάζεται να της προσθέσει κάποια δεδομένα για να λειτουργήσει και τι είδους δεδομένα είναι αυτά: π.χ. λέξεις, αριθμοί, λίστα κλπ.

Η εντολή ask αφορά κυρίως ψηφίδες του μικρόκοσμου και δέχεται ως είσοδο δύο είδη δεδομένων: α) **το όνομα της ψηφίδας** η οποία θέλουμε να εκτελέσει μία ενέργεια και β) **τις ενέργειες** θέλουμε να πραγματοποιήσει αυτή η ψηφίδα (δηλώνουμε ποιος πρέπει να κάνει τι). Η λειτουργία αυτή μοιάζει λίγο με τη σύνταξη των ρημάτων και τα αντικείμενά τους πβ. «λέω κάτι σε κάποιον» ). Έτσι λοιπόν πάντοτε μετά την εντολή **ask** ακολουθεί το καταρχάς το όνομα της ψηφίδας πριν από το οποίο προηγείται πάντοτε εισαγωγικό. Το εισαγωγικό χρησιμοποιείται για να δηλώσει ότι αυτό που ακολουθεί είναι όνομα (όνομα ψηφίδας ή διαδικασίας όπως είδαμε και νωρίτερα στην ενότητα 4.4.1 ή μεταβλητής) ή λέξη<sup>7</sup> την οποία η Logo προβάλλει όπως είναι (π.β. τη λ. ΚΑΛΩΣΟΡΙΣΑΤΕ) χωρίς να την αξιολογεί (π.χ. να προσπαθεί να αναγνωρίσει αν είναι όνομα εντολής για παράδειγμα).

Δοκιμάστε να αφαιρέσετε από την παραπάνω γραμμή κώδικα το εισαγωγικό πριν τη λ. ΚΑΛΩΣΟΡΙΣΑΤΕ. Τι συνέβη; Εμφανίστηκε πάλι το μήνυμα που λέει I don't know how to ΚΑΛΩΣΟΡΙΣΑΤΕ. Το μήνυμα αυτό σημαίνει πως χωρίς το εισαγωγικό η Logo θεωρεί πως η συγκεκριμένη λ. είναι όνομα εντολής και προσπαθεί να την εκτελέσει. Διαπιστώνει όμως ότι στο λεξιλόγιό της δεν υπάρχει τέτοια εντολή ούτε έχει οριστεί μέχρι τώρα κάποια τέτοια διαδικασία από εμάς γι' αυτό μας λέει ότι δεν γνωρίζει τι σημαίνει ΚΑΛΩΣΟΡΙΣΑΤΕ. Ας πάμε τώρα στο δεύτερο input που χρειάζεται η εντολή ask το οποίο είναι μία λίστα ενεργειών και δηλώνεται με την χρήση της αγκύλης [ ]. **ΠΡΟΣΟΧΗ: οι αγκύλες όταν ανοίγουν πρέπει και να κλείνουν** Αν ξεχάσετε να κλείσετε μία αγκύλη ο κώδικας που έχετε γράψει δεν θα εκτελεστεί. Επειδή σε σύνθετα προγράμματα μέσα σε μία αγκύλη μπορεί να υπάρχουν κι άλλες γι' αυτό καλό είναι μόλις πληκτρολογείτε το σύμβολο ανοίγματος της αγκύλης αμέσως μετά να πληκτρολογείτε και το σύμβολο

<sup>6</sup> Τα δεδομένα που εισάγονται από το χρήστη σε μία εντολή θα αναφέρονται εις το εξής με την αγγλική λ. Input. Το ίδιο θα ισχύσει και για τα δεδομένα που εξάγει μία εντολή, όπου θα χρησιμοποιείται η αγγλική λ. output.

<sup>7</sup> Για την εισαγωγή φράσης η σύνταξη διαφέρει κάπως και θα δούμε αργότερα πώς γίνεται

## Εργαστήριο εκπαιδευτικής Τεχνολογίας

κλεισίματος ([ ]) και μετά να πληκτρολογήσετε μέσα στις δύο αυτές αγκύλες ό,τι χρειάζεται. Μέσα στις αγκύλες μπορούμε να γράψουμε μία λίστα από πράγματα στην προκειμένη γράφουμε τη λίστα ενεργειών που θέλουμε να εκτελέσει η ψηφίδα κείμενο. Το παράδειγμα που συζητάμε εδώ, οι ενέργειες που εκτελεί η ψηφίδα κείμενο είναι μόνο μία εκείνη η οποία γράφει τη λ. ΚΑΛΩΣΟΡΙΣΑΤΕ αλλά θα μπορούσε να έχει και περισσότερες όπως η παρακάτω:

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "ΚΑΛΩΣΟΡΙΣΑΤΕ  
ΚΕΙΜΕΝΟ.ΘΕΣΕΔΙΑΦΑΝΕΣ]
```

### Γραμμή κώδικα 2

Σε αυτή τη γραμμή κώδικα η ψηφίδα κείμενο κάνει δύο ενέργειες πρώτα προβάλλει τη λ. «ΚΑΛΩΣΟΡΙΣΑΤΕ και στη συνέχεια γίνεται διαφανής (δηλ. παίρνει το χρώμα του υποβάθρου της επιφάνειας εργασίας του Αβακίου (αν το υποβάθρο σας ταυτίζεται με το χρώμα της γραμματοσειράς που χρησιμοποιήσαμε τότε δεν θα φαίνεται πια η λέξη. Ωστόσο, επιλέχθηκε χρώμα γραμματοσειράς που υποθέτουμε ότι θα φαίνεται στην πλειοψηφία των υποβάθρων)

Ας υποθέσουμε τώρα ότι στην ψηφίδα κείμενο δεν θέλουμε απλώς να γράφει τη λ. ΚΑΛΩΣΟΡΙΣΑΤΕ αλλά θέλουμε να μπει εκεί μία φράση όπου να υπάρχουν κανονικά κενά ανάμεσα στις λέξεις όπως π.χ. ΚΑΛΩΣΟΡΙΣΑΤΕ ΣΤΟ ΜΙΚΡΟΚΟΣΜΟ ΜΑΣ τότε η σύνταξη της εντολής θα πρέπει να έχει την ακόλουθη μορφή:

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "| ΚΑΛΩΣΟΡΙΣΑΤΕ ΣΤΟ ΜΙΚΡΟΚΟΣΜΟ ΜΑΣ|]
```

Δοκιμάστε εδώ να βάλετε το δικό σας κείμενο για να δείτε πώς θα δουλέψει<sup>8</sup>. Οι δύο κάθετες γραμμές χρησιμεύουν για να μην μπερδευτεί η Logo με τα κενά που μεσολαβούν μεταξύ των λέξεων.

### Όνομα ψηφίδας: Περισσότερες από μία λέξεις

Τις ίδιες κάθετες γραμμές χρησιμοποιούμε αν το όνομα της ψηφίδας μας αποτελείται από περισσότερες της μίας λέξεις: αν π.χ. αλλάζαμε το όνομα της ψηφίδας Κείμενο από Κείμενο σε Μήνυμα εισαγωγής τότε θα γράφαμε την παραπάνω εντολή ως εξής:

```
ask "|Μήνυμα εισαγωγής| [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "|ΚΑΛΩΣΟΡΙΣΑΤΕ ΣΤΟ  
ΜΙΚΡΟΚΟΣΜΟ ΜΑΣ|]
```

### Γραμμή κώδικα 3

Για το σύνολο των εντολών της Logo που χρησιμοποιούνται στο Αβάκιο μπορεί κανείς να συμβουλευτεί το Turtle Tracks του D. Azuma (να προστεθεί ppp).

<sup>8</sup> Η κάθετη γραμμή σε ορισμένα πληκτρολόγια εμφανίζεται αν πιέσετε ταυτόχρονα τα πλήκτρα shift και το πλήκτρο πάνω από το enter που έχει δύο διακεκομμένες γραμμές τη μία κάτω από την άλλη

Για να μπει κανείς στη φιλοσοφία του προγραμματισμού με τη Logo περισσότερο κατάλληλο είναι το βιβλίο Harvey, B. (1997). *Computer Science Logo Style* (2nd ed.). Cambridge, Massachusetts: MIT Press (Διαθέσιμο και ηλεκτρονικά στο <http://www.cs.berkeley.edu/~bh/pdf/v1choo.pdf>)

### 3.5.1.1 Εντολές ψηφίδων του Αβακίου

Εκτός από τις εντολές της Logo κάθε ψηφίδα του Αβακίου έχει ένα σύνολο από δικές της εντολές με τις οποίες μπορεί ο χρήστης να «της μιλήσει» και να προσθέσει λειτουργικότητες. Για να γίνει περισσότερο κατανοητό αυτό θα μπορούσαμε να παραλληλίσουμε τη Logo με τη φυσική γλώσσα και τις εντολές των ψηφίδων με την ορολογία των διαφόρων επιστημών που ακολουθούν τη λογική, τη σύνταξη και τη γραμματική της φυσικής γλώσσας.

Ας ξαναδούμε τώρα τη γραμμή κώδικα 1, που παρουσιάσαμε στην αρχή της προηγούμενης ενότητας δηλ.

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "ΚΑΛΩΣΟΡΙΣΑΤΕ]
```

Η εντολή ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ είναι εντολή της ψηφίδας Κείμενο. Φαίνεται εξ' άλλου από το πρώτο συνθετικό (ΚΕΙΜΕΝΟ). ΠΡΟΣΟΧΗ: Το πρώτο συνθετικό της εντολής δεν σχετίζεται με το όνομα της ψηφίδας αλλά με την κατηγορία της (ότι δηλ. πρόκειται για την ψηφίδα κείμενο). Δηλ. όταν πληκτρολογούμε την εντολή ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ το πρώτο συνθετικό δηλ. η λ. ΚΕΙΜΕΝΟ δεν αλλάζει ανάλογα με το όνομα της ψηφίδας.

Η λ. "ΚΑΛΩΣΟΡΙΣΑΤΕ είναι το Input στην εντολή ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ όπου στην πραγματικότητα γράφουμε τι θέλουμε να φαίνεται στην ψηφίδα Κείμενο. Το input στην εντολή αυτή μπορεί να είναι οτιδήποτε όπως φάνηκε και στην Γραμμή κώδικα 3. Δοκιμάστε να αλλάξετε το κείμενο που φαίνεται στην ψηφίδα και να το αντικαταστήσετε για παράδειγμα με το όνομά σας. Αν βάλετε μόνο το όνομά σας θα πρέπει να χρησιμοποιήσετε τη Γραμμή κώδικα 1. Αν βάλετε το ονοματεπώνυμό σας θα πρέπει να χρησιμοποιήσετε τη Γραμμή κώδικα 3. Προσοχή στο όνομα της ψηφίδας. Αν δεν έχετε κάνει μετονομασία στην ψηφίδα κείμενο τότε θα πρέπει να διαμορφώσετε την εντολή ως εξής:

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "[Όνομα Επίθετο]]
```

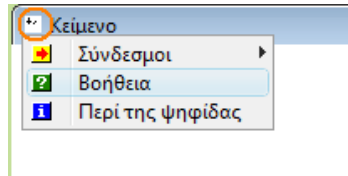
#### Γραμμή κώδικα 4

Αντίστοιχα, η εντολή ΚΕΙΜΕΝΟ.ΘΕΣΕΔΙΑΦΑΝΕΣ ανήκει στο σύνολο των εντολών της ψηφίδας Κείμενο. Μόνο που αυτή η εντολή δεν χρειάζεται input και γι' αυτό δεν έχουμε προσθέσει κάτι που να την ακολουθεί

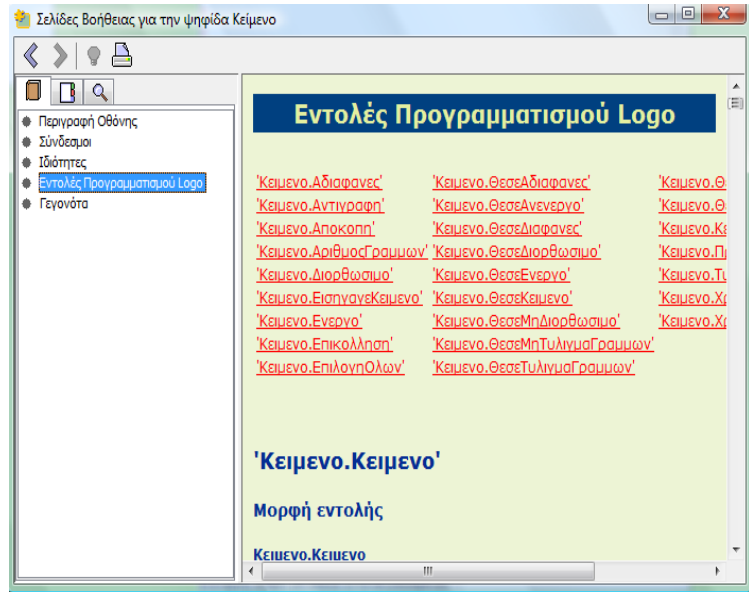
Εντολές Ψηφίδων: Πού τις βρίσκουμε

Τις εντολές των ψηφίδων μπορούμε να τις βρούμε σε δύο σημεία:

A) στη βοήθεια των ψηφίδων



Εικόνα 3.14 Προβολή βοήθειας των ψηφίδων



Εικόνα 3.15 Εντολές Προγραμματισμού Logo της ψηφίδας Κείμενο

Για να προβάλλουμε τις εντολές Logo μέσα από τη βοήθεια των ψηφίδων κάνουμε κλικ στο σύμβολο της ψηφίδας που βρίσκεται πριν το όνομά της πάνω στη Μπάρα (βλ. τον πορτοκαλί κύκλο στην Εικόνα Εικόνα 3.14) και κάνουμε κλικ πάνω στην επιλογή Βοήθεια. Στη συνέχεια, στο παράθυρο που εμφανίζεται επιλέξτε Εντολές Προγραμματισμού Logo (βλ. Εικόνα 3.15). Εκεί μπορείτε να δείτε το σύνολο των εντολών της κάθε ψηφίδας μαζί με μία μικρή επεξήγηση.

Β) Ο δεύτερος τρόπος προβολής των εντολών των ψηφίδων είναι μέσω της ψηφίδας Logo.

Πληκτρολογείτε στην ψηφίδα Logo την ακόλουθη εντολή:

`Ask "|Όνομα Ψηφίδας| [primitives]`

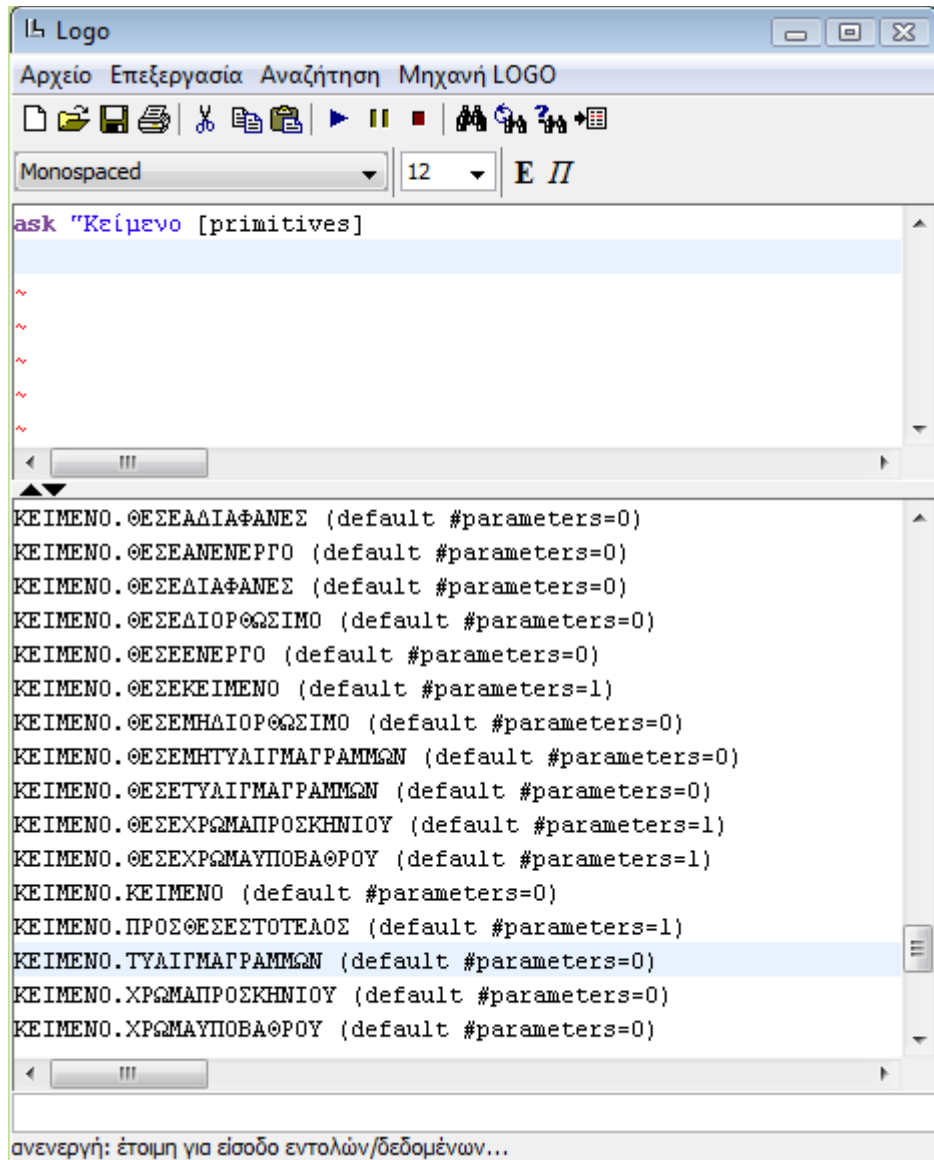
Γραμμή κώδικα 5

Μετά το Ask βάζουμε το όνομα της ψηφίδας που μας ενδιαφέρει στην περίπτωση της ψηφίδας Κείμενο διαμορφώνουμε την εντολή ως εξής: ask "Κείμενο [primitives]. Δοκιμάστε να εισάγετε στο μικρόκοσμο welcome1 μία ψηφίδα «Κουμπί Διακόπτης» και να εμφανίσετε μέσω της Logo τις εντολές της. Στο μικρόκοσμο welcome procedures θα βρείτε στην ψηφίδα Logo πώς έπρεπε να γράψετε την εντολή. Οι εντολές της ψηφίδας «Διακόπτης» βρίσκονται στο κάτω μισό της ψηφίδας Logo και τις έχουμε αντιγράψει και σε μία νέα ψηφίδα Κείμενο με το όνομα «Κείμενο\_1»

ΠΡΟΣΟΧΗ: αν έχουμε μετονομάσει την ψηφίδα ώστε το όνομά της να είναι «Μήνυμα εισαγωγής» τότε η ίδια εντολή διαμορφώνεται ως εξής: ask "|Μήνυμα εισαγωγής| [primitives]. Όπως αναφέρθηκε στην προηγούμενη ενότητα ο λόγος για τον οποίο

χρησιμοποιούμε τις κάθετες γραμμές γύρω από το όνομα της ψηφίδας είναι γιατί το όνομα της ψηφίδας αποτελείται από δύο λέξεις με κενό ανάμεσά τους.

Σε αυτή την περίπτωση οι εντολές της ψηφίδας εμφανίζονται στο κάτω μισό της ψηφίδας Logo (βλ. Εικόνα 3.15)



Εικόνα 3.16: Προβολή των εντολών της ψηφίδας Κείμενο στην ψηφίδα Logo

### Εντολές ψηφίδων: βοήθεια ψηφίδων vs Logo

Μεταξύ των εντολών που εμφανίζονται στη βοήθεια των ψηφίδων και των εντολών που εμφανίζονται στην ψηφίδα Logo υπάρχει η εξής σημαντική διαφορά: στην ψηφίδα Logo εκτός από τις εντολές εμφανίζεται η εξής πολύτιμη πληροφορία: αν η εντολή παίρνει input (αν δηλαδή χρειάζεται να προσθέσει ο χρήστης κάποια δεδομένα για λειτουργήσει εντολή ή όχι) και πόσα δεδομένα χρειάζεται. Έτσι λοιπόν η εντολή KEIMENO.ΘΕΣΕΚΕΙΜΕΝΟ βλέπουμε ότι συνοδεύεται από την εξής πληροφορία (default #parameters 1) που σημαίνει ότι έχει μία είσοδο δεδομένων δηλ. ο χρήστης θα πρέπει



μετά την εντολή KEIMENO.ΘΕΣΕΚΕΙΜΕΝΟ να πει στην ψηφίδα τι κείμενο θα βάλει (όπως φάνηκε στα προηγούμενα παραδείγματα).

Από την άλλη, εντολή KEIMENO.ΘΕΣΕΔΙΑΦΑΝΕΣ συνοδεύεται από την πληροφορία (default #parameters 0) που λέει ότι δεν χρειάζεται input από το χρήστη (το input εδώ είναι 0). Αυτό με έναν τρόπο είναι λογικό γιατί αν σκεφτούμε τι λέμε στην ψηφίδα να κάνει: να γίνει δηλ. διαφανής καταλαβαίνουμε ότι δεν υπάρχουν πολλά περιθώρια για διαφοροποιήσεις. Έτσι, οι εντολές που δέχονται input μπορούμε να τις αντιμετωπίσουμε ως πιο ανοιχτές και περισσότερο προσαρμόσιμες από το χρήστη (στην περίπτωση της εντολής KEIMENO.ΘΕΣΕΚΕΙΜΕΝΟ μπορούμε να βάλουμε ό,τι κείμενο θέλουμε).

### Εντολές Ψηφίδων: ποιόν τρόπο προβολής προτιμάμε

Από τους δύο τρόπους που συζητήσαμε νωρίτερα για την προβολή των εντολών των ψηφίδων, προτείνεται να προτιμήσετε τη Logo ακριβώς γιατί προβάλλει το input που χρειάζεται η ψηφίδα για να λειτουργήσει και στη συνέχεια χρησιμοποιείτε τις επεξηγήσεις που ενδεχομένως δίνει η βοήθεια.

#### 3.5.1.2 Δουλεύοντας με τα μηνύματα λάθους της Logo

Ας δούμε τώρα με περισσότερη λεπτομέρεια την εντολή KEIMENO.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ. Η εντολή αυτή αλλάζει το φόντο πάνω στο οποίο φαίνονται τα γράμματα του κειμένου. Σύμφωνα με τις εντολές της Logo βλέπουμε ότι έχει μία είσοδο άρα χρειάζεται ένα Input. Τι είδους όμως δεδομένο μπορεί να είναι αυτό; η λ. κίτρινο για παράδειγμα; (σε περίπτωση που θέλουμε να αλλάξουμε το υπόβαθρο σε κίτρινο). Δοκιμάστε να πληκτρολογήσετε λοιπόν στην ψηφίδα Logo την εντολή:

```
ask "Κείμενο [KEIMENO.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ "κίτρινο] τι θα συμβεί;
```

Δεν θα αλλάξει το χρώμα του υποβάθρου και στο κάτω μέρος της Logo κάτω από τις εντολές του κειμένου θα εμφανιστεί το εξής μήνυμα: «Integer expected ... while executing KEIMENO.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ». Το συγκεκριμένο μήνυμα σφάλματος μας δίνει μία πολύτιμη πληροφορία, ότι δηλ. η εντολή περιμένει ως input αριθμό. Ποιόν όμως αριθμό; Ας δοκιμάσουμε το 1. Πληκτρολογήστε στην ψηφίδα Logo την παραπάνω εντολή αντικαθιστώντας το κίτρινο με το 1. (προσοχή χωρίς εισαγωγικά πριν το 1 γιατί θέλουμε η logo να αξιολογήσει το input και να καταλάβει ότι πρόκειται για αριθμό) Τι θα συμβεί; Θα εμφανιστεί το ακόλουθο μήνυμα: Index out of bounds ... while executing KEIMENO.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ. Ας υποθέσουμε τώρα ότι δεν ξέρουμε να ερμηνεύσουμε ακριβώς το συγκεκριμένο μήνυμα. Το βασικό μας πρόβλημα είναι να βρούμε τι είδους δεδομένα πρέπει να δώσουμε ως Input στην εντολή KEIMENO.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ. Ας εξετάσουμε τώρα με ποιόν τρόπο θα μπορούσαμε να βγούμε από το αδιέξοδο. Συχνά οι ίδιες οι εντολές της ψηφίδας μπορούν να μας δώσουν μια απάντηση. Συγκεκριμένα, στο σύνολο των εντολών που φαίνονται ήδη στην προηγούμενη εικόνα (βλ. Εικόνα 3.16) θα παρατηρήσουμε ότι υπάρχει μία εντολή που ονομάζεται KEIMENO.ΧΡΩΜΑΥΠΟΒΑΘΡΟΥ και δεν χρειάζεται input (default #parameters 0). Τι κάνει αυτή η εντολή; Ουσιαστικά ζητάει από την ψηφίδα KEIMENO να



μας πει ποιο είναι το χρώμα υποβάθρου. Άρα η ψηφίδα αυτή θα δίνει output και το output αυτό θα είναι τα δεδομένα τα οποία περιγράφουν το χρώμα.

Πληκτρολογήστε λοιπόν στη Logo την ακόλουθη εντολή

```
ask "Κείμενο [print KEIMENO.ΧΡΩΜΑΥΠΟΒΑΘΡΟΥ].
```

**Γραμμή κώδικα 6**

Θα δείτε ότι στο κάτω μέρος της ψηφίδας Logo εμφανίζονται τρεις αριθμοί: 255 255 255 οι οποίοι περιγράφουν το άσπρο χρώμα. Εδώ λοιπόν έχουμε τρεις αριθμούς ενώ η εντολή KEIMENO.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ λέει ότι δέχεται ένα μόνο Input. Επίσης έχουμε ως δεδομένο το τελευταίο μήνυμα λάθους που μας έλεγε «Index out of bounds». Αυτό εν ολίγοις σημαίνει ότι οι αριθμοί που δηλώνουν το χρώμα μπαίνουν σε αγκύλη η οποία γίνεται αντιληπτή ως ένα πράγμα.

Δοκιμάστε λοιπόν να πληκτρολογήσετε στη Logo την εντολή

```
ask "Κείμενο [KEIMENO.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 0 204]]
```

**Γραμμή κώδικα 7**

Αυτό που θα συμβεί είναι να αλλάξει χρώμα το υπόβαθρο της ψηφίδας από λευκό σε φούξια.

**Πρόταση για δοκιμές:** Βρείτε τους αριθμούς που έχουν τα διάφορα χρώματα αλλάζοντας το φόντο του κειμένου με τον τρόπο που περιγράφεται στο εγχειρίδιο του Αβακίου σελ. 16-23 για την αλλαγή της μορφής των ψηφίδων και χρησιμοποιώντας τη **Γραμμή κώδικα 6**

### 3.5.2 Ορισμός διαδικασιών στη Logo: φτιάχνοντας φωτεινές επιγραφές

Ας υποθέσουμε τώρα ότι θέλουμε να κάνουμε κάτι πιο σύνθετο. Ότι θέλουμε να φτιάξουμε κάτι σαν φωτεινή επιγραφή με το φόντο να αναβοσβήνει για ένα διάστημα αλλάζοντας χρώμα από κίτρινο σε πορτοκαλί. Για να το κάνουμε αυτό θα στηριχτούμε στις γραμμές κώδικα που έχουμε παρουσιάσει ήδη και αλλάζουν το φόντο του κειμένου (γραμμή κώδικα 7)

```
ask "Κείμενο [KEIMENO.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 255 10]]
```

η πρώτη γραμμή κάνει το χρώμα κίτρινο και η επόμενη το κάνει πορτοκαλί

```
ask "Κείμενο [KEIMENO.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 98 20]]
```

Αντιγράψτε τις δύο εντολές στη Logo πηγαίνετε τον κέρσορα στο τέλος της πρώτης γραμμής (έξω από την αγκύλη) και πιάστε insert αμέσως μετά πηγαίνετε στο τέλος της δεύτερης γραμμής και πιάστε insert. Θα δείτε ότι όντως αλλάζει το χρώμα όπως αναφέρθηκε πριν αλλά όχι ακριβώς όπως σε μία φωτεινή επιγραφή που δουλεύει αυτόματα. Για να φτιάξουμε λοιπόν μία φωτεινή επιγραφή θα μπορούσαμε να δημιουργήσουμε στη logo μία νέα εντολή που θα την ονομάσουμε Φωτεινή\_επιγραφή.

Για να μπορέσει η Logo να εντάξει αυτή την εντολή στο λεξιλόγιό της θα πρέπει να την ορίσουμε, να περιγράψουμε δηλ. ποιες ενέργειες θα πρέπει να κάνει όταν θα εκτελεί την εντολή Φωτεινή\_επιγραφή. Επειδή λοιπόν ο ορισμός μίας νέας εντολής στη Logo ουσιαστικά περιλαμβάνει την περιγραφή μίας σειράς ενεργειών λέμε ότι ορίζουμε μία διαδικασία. Αφού οριστεί η διαδικασία και τη μάθει η Logo τότε το όνομα της διαδικασίας –στην προκειμένη η Φωτεινή\_επιγραφή - είναι η νέα εντολή ενταγμένη στο Λεξιλόγιο της Logo.

### Πώς ορίζουμε μία διαδικασία στη Logo;

Ξεκινάμε χρησιμοποιώντας τη λ. “για” ή στα Αγγλικά τη λ. “to” στη συνέχεια ακολουθεί το όνομα της διαδικασίας το οποίο είναι πάντοτε **ΜΙΑ ΛΕΞΗ** ενδεικτική της λειτουργίας που επιτελεί η διαδικασία. Στην προκειμένη το όνομα είναι Φωτεινή\_Επιγραφή που αποτελείται από δύο λέξεις ενωμένες με κάτω παύλα (underscore “\_”). Ο λόγος για τον οποίο χρησιμοποιούμε underscore εδώ σχετίζεται με το ρόλο των κενών που αναφέρθηκε νωρίτερα.

Έτσι λοιπόν ξεκινάμε λέγοντας στη Logo τα εξής: **για** να κάνεις τη **Φωτεινή\_επιγραφή** θα πρέπει να εκτελέσεις τις **ακόλουθες ενέργειες**: και γράφουμε τη γραμμή κώδικα 1 (βλ. Διαδικασία 1) και τις δύο γραμμές κώδικα που αναφέρθηκαν νωρίτερα: δηλ. πρώτα θα ζητήσεις από την ψηφίδα κείμενο να γράψει τη λ. «ΚΑΛΩΣΟΡΙΣΑΤΕ, έπειτα θα ζητήσεις από την ψηφίδα κείμενο να κάνει το χρώμα του υποβάθρου κίτρινο και μετά πορτοκαλί. Αυτές είναι όλες κι όλες οι ενέργειες που θέλουμε προς το παρόν να εκτελέσει η Logo. Για να καταλάβει λοιπόν ότι δεν χρειάζεται να κάνει κάτι παραπάνω θα πρέπει μετά τις εντολές να πληκτρολογήσουμε τη λ. «τέλος» ή “end” στα Αγγλικά. Το τέλος λειτουργεί όπως η τελεία σε μία πρόταση. Έτσι λοιπόν η διαδικασία μας έχει την ακόλουθη μορφή

για Φωτεινή\_επιγραφή

ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "ΚΑΛΩΣΟΡΙΣΑΤΕ]

ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 255 10]]

ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 98 20]]

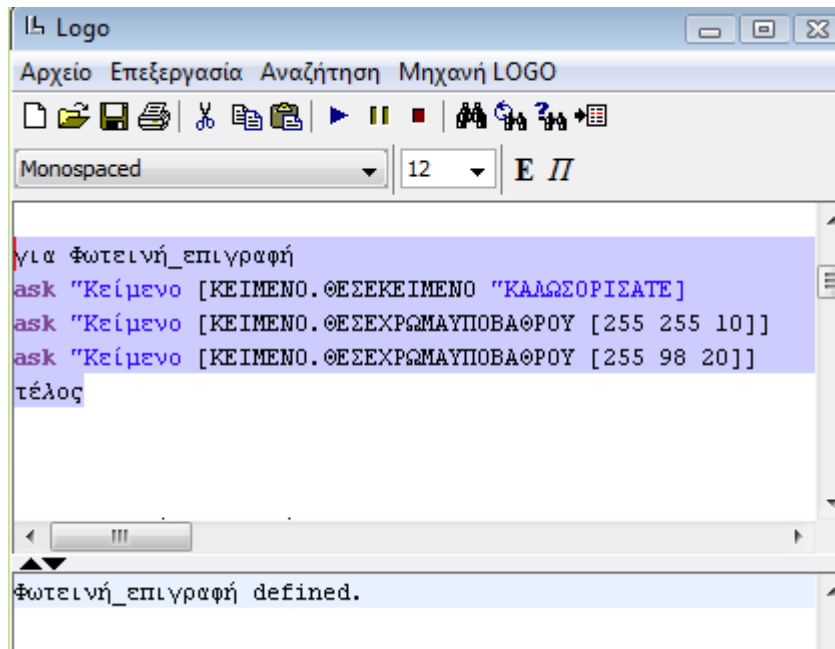
τέλος

#### Διαδικασία 1

Μέχρι τώρα έχουμε γράψει τη διαδικασία απλώς, δεν την έχουμε ορίσει όμως. Ο ορισμός είναι κάτι σαν δήλωση στη Logo ότι αυτή είναι μία νέα εντολή με το περιεχόμενο που περιγράψαμε στη διαδικασία: Αυτό γίνεται ως εξής:

- Πρώτα επιλέγουμε με το ποντίκι μας το περιεχόμενο της διαδικασίας από το «για» μέχρι και το «τέλος» (να περιλαμβάνει όλη τη λέξη βλ. Εικόνα 3.17)
- Πιέζουμε το πλήκτρο Insert ή το κουμπί run που βρίσκεται στη γραμμή εργαλείων της ψηφίδας Logo

Αν γίνει αυτό σωστά στο κάτω μισό της ψηφίδας Logo θα εμφανιστεί το μήνυμα **Φωτεινή\_επιγραφή defined**. Που σημαίνει ότι η διαδικασία Φωτεινή\_επιγραφή έχει οριστεί ως νέα εντολή (βλ. Εικόνα 3.17).



Εικόνα 3.17 Ορισμός της διαδικασίας Φωτεινή επιγραφή

Αντιγράψτε την παραπάνω διαδικασία στο μικρόκοσμο **welcome1** και ορίστε την με τον τρόπο που περιγράψαμε.

Αφού ορίσαμε τι σημαίνει στη Logo Φωτεινή\_Επιγραφή τώρα έχουμε προσθέσει μία νέα εντολή στο λεξιλόγιο της Logo την οποία μπορούμε να την καλούμε όποτε θέλουμε μόνο στο πλαίσιο του μικρόκοσμου που την έχουμε ορίσει.

#### Εντολές: εκτέλεση εντολών στη Logo

Για να εκτελέσουμε τη Φωτεινή\_επιγραφή, αρκεί να γράψουμε τώρα πια το όνομά της στη Logo – ΠΡΟΣΟΧΗ! Μόνο το όνομά της δηλ. Φωτεινή\_επιγραφή χωρίς τη συνοδεία του «για» - και να πατήσουμε το Insert ή το κουμπί run. [ΠΡΟΣΟΧΗ: Είμαστε στο μικρόκοσμο welcome 1]Τι συνέβη; Φαίνεται ότι το υπόβαθρο γίνεται κατ' ευθείαν πορτοκαλί παρακάμπτοντας το κίτρινο. Θα μπορούσε ίσως να υποθέσει κάποιος ότι υπάρχει κάποιο λάθος και παρακάμπτεται η πρώτη εντολή. Μάλλον όμως, δεν είναι αυτό το πρόβλημα, επειδή η Logo δεν βγάζει κάποιο σφάλμα. Στην πραγματικότητα αυτό που συμβαίνει είναι πως τα χρώματα αλλάζουν πάρα πολύ γρήγορα και δεν τα βλέπουμε άρα χρειάζεται να προσθέσουμε στη διαδικασία κάτι σαν καθυστέρηση να πούμε δηλ. στη Logo να αλλάξει πρώτα το υπόβαθρο σε κίτρινο, μετά να περιμένει λίγο και έπειτα να το αλλάξει σε πορτοκαλί. Θα χρησιμοποιήσουμε για αυτό το σκοπό την εντολή wait και θα προσθέσουμε ως input τον αριθμό 200 που λέει στη Logo πόσο να περιμένει μέχρι να προχωρήσει στην εκτέλεση της επόμενης εντολής. Για να αλλάξουμε λοιπόν τη διαδικασία Φωτεινή\_επιγραφή ώστε να περιλαμβάνει και τη νέα εντολή αρκεί να κάνουμε τα εξής:

- Να προσθέσουμε τη συγκεκριμένη εντολή μεταξύ των δύο εντολών που αλλάζουν το χρώμα υποβάθρου και έτσι η διαδικασία Φωτεινή\_επιγραφή θα διαμορφωθεί όπως φαίνεται στη συνέχεια

## Εργαστήριο εκπαιδευτικής Τεχνολογίας

για Φωτεινή\_επιγραφή

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "ΚΑΛΩΣΟΡΙΣΑΤΕ]
```

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 255 10]]
```

```
wait 200
```

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 98 20]]
```

τέλος

### Διαδικασία 2

- Να ενημερώσουμε τη Logo ότι το περιεχόμενο της εντολής Φωτεινή\_επιγραφή άλλαξε: ορίζουμε δηλαδή τη διαδικασία ξανά με τον τρόπο που αναφέραμε νωρίτερα (επιλέγοντάς την με τον κέρσορα από το «για» μέχρι το «τέλος» και να πιέσουμε το κουμπί Insert). Στο κάτω μισό της Logo θα εμφανιστεί το μήνυμα **Φωτεινή\_επιγραφή redefined** που σημαίνει ότι η εντολή Φωτεινή\_επιγραφή ορίστηκε εκ νέου

Δοκιμάστε στη συνέχεια να εκτελέσετε την εντολή Φωτεινή\_επιγραφή. Θα δείτε ότι το αποτέλεσμα της είναι πιο κοντά σε αυτό που μπορεί να θεωρηθεί φωτεινή επιγραφή. Για να το προχωρήσουμε ακόμη περισσότερο έτσι ώστε η φωτεινή επιγραφή να αλλάζει χρώμα κάμποσες φορές θα μπορούσαμε να αλλάξουμε πάλι την παραπάνω διαδικασία ως εξής:

για Φωτεινή\_επιγραφή

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "ΚΑΛΩΣΟΡΙΣΑΤΕ]
```

```
repeat 5 [
```

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 255 10]]
```

```
WAIT 200
```

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 98 20]]
```

```
WAIT 200 ]
```

Τέλος

### Διαδικασία 3

Η φωτεινή επιγραφή τώρα κάνει τα εξής: αζητάει από την ψηφίδα κείμενο να γράψει τη λ. Καλωσορίσατε στη συνέχεια επαναλαμβάνει πέντε φορές τις εξής ενέργειες: αλλάζει το χρώμα υποβάθρου σε κίτρινο, περιμένει 200 μονάδες χρόνου, αλλάζει το χρώμα υποβάθρου σε πορτοκαλί, περιμένει άλλες 200 μονάδες χρόνου

Για να δοκιμάσετε αυτή τη διαδικασία στη Logo θα πρέπει να κάνετε τις αλλαγές στον κώδικα όπως φαίνονται στη «Διαδικασία 3» και να ακολουθήσετε τα ίδια βήματα που έγιναν για τη διαδικασία 2 στο μικρόκοσμο welcome 1. Αν κάτι δεν δουλέψει για κάποιο λόγο, μπορείτε να δείτε πώς λειτουργεί αυτή η εντολή στο μικρόκοσμο welcome 2.

### 3.5.2.1 Ισχύς διαδικασιών

Ανοίξτε το μικρόκοσμο `paintings_manual`. Δοκιμάστε να πληκτρολογήσετε στη Logo την εντολή `Φωτεινή_επιγραφή` και να την εκτελέσετε. Θα δείτε ότι η Logo δεν θα εκτελέσει την εντολή και θα βγάλει ένα μήνυμα λάθους που θα λέει «I don't know how to Φωτεινή\_επιγραφή». Αυτό συμβαίνει διότι οι νέες εντολές που δημιουργούμε ισχύουν τοπικά, για το μικρόκοσμο δηλαδή που τις έχουμε ορίσει και όχι γενικά όπως οι βασικές εντολές της Logo. Άρα αν θέλουμε μία νέα εντολή να τη χρησιμοποιήσουμε και σε άλλον μικρόκοσμο θα πρέπει να την αντιγράψουμε και να την ορίσουμε ξανά στο μικρόκοσμο που θέλουμε να τη χρησιμοποιήσουμε.

### 3.5.3 Ορισμός υπερδιαδικασιών

Οι διαδικασίες που φτιάχνουμε και που εμπλουτίζουν το λεξιλόγιο της Logo μπορούν όπως και οι υπάρχουσες εντολές της Logo να ενταχθούν σε νέες διαδικασίες που τις ονομάζουμε υπερδιαδικασίες.

Ας ξεκινήσουμε από το προηγούμενο παράδειγμα της φωτεινής επιγραφής. Στο προηγούμενο παράδειγμα είδαμε ότι η φωτεινή επιγραφή αλλάζει μόνο δύο χρώματα. Ας υποθέσουμε ότι θέλουμε να τα εμπλουτίσουμε έτσι ώστε οι εναλλαγές των χρωμάτων να πλησιάζουν τα χρώματα του ουράνιου τόξου. Κάνουμε τα εξής βήματα

Ορίζουμε τις ακόλουθες έξι διαδικασίες που έχουν ονόματα χρωμάτων και κάθε μία ξεχωριστά θέτει το αντίστοιχο χρώμα στο φόντο του κειμένου

*για πράσινο*

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [ 0 204 0 ]]
```

*τέλος*

#### Διαδικασία 4

*για κίτρινο*

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [ 255 255 0 ]]
```

*τέλος*

#### Διαδικασία 5

*για πορτοκαλί*

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [ 255 102 0 ]]
```

*τέλος*

#### Διαδικασία 6

*για κόκκινο*

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [ 255 0 0 ]]
```

*τέλος*

#### Διαδικασία 7

*για μώβ*

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [ 204 0 204 ]]
```

τέλος

#### Διαδικασία 8

για γαλάζιο

*ask "Κείμενο [KEIMENO.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [ο 2ο4 2ο4 ]]*

τέλος

#### Διαδικασία 9

Όλες τις παραπάνω διαδικασίες τις εντάσσουμε σε μία υπερ-διαδικασία που την ονομάζουμε ουράνιο\_τόξο το οποίο ζητάει από τη Logo να εκτελέσει τις νέες διαδικασίες κόκκινο, πράσινο κ.ο.κ. Στην πραγματικότητα κάνει ό,τι έκανε και η Φωτεινή\_επιγραφή μόνο που χρησιμοποιεί περισσότερα χρώματα.

για ουράνιο\_τόξο

*επανάλαβε 5 [ κόκκινο wait 400 πορτοκαλί wait 400 κίτρινο wait 400 πράσινο wait 400*

*γαλάζιο wait 400 μώβ wait 400 ]*

τέλος

#### Διαδικασία 10

Μπορείτε να αντιγράψετε, να ορίσετε τις παραπάνω διαδικασίες και να εκτελέσετε το ουράνιο τόξο στο μικρόκοσμο. Οι παραπάνω διαδικασίες λειτουργούν μέσα στο μικρόκοσμο welcome 2. Σε περίπτωση που υπάρχει πρόβλημα μπορείτε να δείτε πώς δουλεύει το ουράνιο\_τόξο στο μικρόκοσμο welcome\_procedures

### 3.5.4 Δημιουργία διαδικασιών με μεταβλητές

Συζητώντας νωρίτερα, για τη σύνταξη των εντολών είδαμε ορισμένες από τις εντολές χρειάζονται συγκεκριμένο Input για να λειτουργήσουν. Οι εντολές οι οποίες χρειάζονται Input δίνουν κατά κάποιον τρόπο μεγαλύτερη ελευθερία στο χρήστη να τις προσαρμόσει σε αυτά που θέλει να κάνει (π.β. την εντολή KEIMENO.ΘΕΣΕΚΕΙΜΕΝΟ που επιτρέπει στο χρήστη να βάλει στην ψηφίδα κείμενο ό,τι κείμενο θέλει. Ενώ η εντολή KEIMENO.ΘΕΣΕΔΙΑΦΑΝΕΣ κάνει την ψηφίδα διαφανή και τίποτε άλλο).

Οι εντολές που φτιάξαμε μέχρι τώρα δεν χρειάζονταν Input. Βάζαμε απλώς το όνομά τους και τις εκτελούσαμε. Η Logo μας δίνει τη δυνατότητα να φτιάξουμε εντολές που δέχονται Input, έτσι γίνονται πιο ευέλικτες και δίνουν μεγαλύτερη ελευθερία στο χρήστη. Για να δούμε πώς μπορεί να γίνει αυτό, ας πάρουμε τη διαδικασία Φωτεινή\_επιγραφή (Διαδικασία 3), εκεί είδαμε πως ζητήσαμε από τη Logo να προχωρήσει σε εναλλαγή του χρώματος του υποβάθρου 5 φορές ωστόσο κάποιος μπορεί να θέλει να αλλάζει αυτόν τον αριθμό κατά βούλησιν. Αντί να αλλάζει και να ξανα-ορίζει τη Φωτεινή\_επιγραφή μπορεί να πει ότι ο χρήστης θα ορίζει πόσες φορές θα γίνεται η εναλλαγή του φόντου και να μην είναι καθορισμένο από πριν. Έτσι λοιπόν ορίζουμε τώρα μία νέα διαδικασία που θα μοιάζει με την τελική Φωτεινή\_επιγραφή αλλά το «αναβόσβημά» της θα ορίζεται από το χρήστη (βλ. Διαδικασία 4)

<p>για Φωτεινή_επιγραφή</p> <pre>ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΣΕΚΕΙΜΕΝΟ "ΚΑΛΩΣΟΡΙΣΑΤΕ] repeat 5 [ ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 255 10]] WAIT 200 ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 98 20]] WAIT 200 ] Τέλος <b>Διαδικασία 3</b></pre>	<p>Για ΝέαΦωτεινήΕπιγραφή <b>φορές</b></p> <pre>ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΣΕΚΕΙΜΕΝΟ "ΚΑΛΩΣΟΡΙΣΑΤΕ] repeat :φορές [ ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 255 10]] WAIT 500 ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ [255 98 20]] WAIT 500 ] τέλος <b>Διαδικασία 11</b></pre>
--	---

Αν αντιπαραβάλλουμε τις δύο διαδικασίες: Διαδικασία 3 - πάνω στην οποία στηριχτήκαμε- και Διαδικασία 11 θα δούμε ότι η μόνη διαφορά είναι η εξής: ο αριθμός 5 (σημειωμένος ) έχει αντικατασταθεί με το. **:φορές**. Επίσης το **:φορές** βλέπουμε ότι υπάρχει και στην αρχή αμέσως μετά όνομα της διαδικασίας (ξεκινήσαμε με το «για» αφήσαμε κενό, βάλαμε το όνομα Διαδικασίας αφήσαμε κενό και μετά βάλαμε το :φορές). Όπως διαπιστώνουμε πριν από το φορές υπάρχει άνω και κάτω τελεία. Το σημείο αυτό (η άνω και κάτω τελεία δηλ.) σηματοδοτεί στη Logo ότι ακολουθεί αυτό που ονομάζουμε μεταβλητή. Αυτό στην πραγματικότητα σημαίνει ότι τα συγκεκριμένα δεδομένα δεν θα δοθούν κατά τον ορισμό της διαδικασίας αλλά κατά την εκτέλεσή της. Έτσι για να εκτελέσουμε τη διαδικασία ΝέαΦωτεινήΕπιγραφή θα πρέπει να αφήσουμε κενό και να δώσουμε τον αριθμό που θα επαναληφθεί η αλλαγή του φόντου.

Αφού αντιγράψετε και ορίσετε τη διαδικασία 4, εκτελέστε την εντολή ΝέαΦωτεινήΕπιγραφή 7 (μετά το όνομα της διαδικασίας ακολουθεί κενό και έπειτα βάζουμε τον αριθμό 7). Αυτό που θα συμβεί είναι ότι το φόντο θα αλλάξει 7 φορές. Αν μας φαίνεται πολύ μπορούμε να εκτελέσουμε την εντολή ΝέαΦωτεινήΕπιγραφή 3 κ.ο.κ

### 3.5.5 Σημαντικές διευκρινήσεις τις διαδικασίες και τις εντολές

Συνοπτικά παρουσιάζουμε εδώ κάποια στοιχεία που έχουν ήδη αναφερθεί αλλά συχνά παρατηρούνται λάθη

- Χρησιμοποιούμε το «για» και το «τέλος» μόνο όταν θέλουμε να ορίσουμε μία διαδικασία (νέα εντολή) στη Logo. Αυτό το κάνουμε μόνο μία φορά
- Όταν θέλουμε να εκτελέσουμε μία εντολή, να ζητήσουμε δηλ. από τη Logo να κάνει ότι έχει περιγραφεί στην διαδικασία που ορίζει την εντολή, αρκεί να πληκτρολογήσουμε το όνομά της και να πατήσουμε το πλήκτρο Insert ή να πιέσουμε το κουμπί run στο μενού της Logo. Μία εντολή μπορούμε να την εντάξουμε σε νέες διαδικασίες και να την καλέσουμε εντός ή εκτός διαδικασιών όσες φορές θέλουμε
- Κάθε νέα εντολή ισχύει μόνο για το μικρόκοσμο στον οποίο ορίστηκε. Για να λειτουργήσει και σε κάποιον άλλο μικρόκοσμο θα πρέπει να ξανα-οριστεί.
- Το όνομα μίας νέας εντολής δεν θα πρέπει να περιέχει κενά, θα πρέπει να είναι γραμμένο σαν μία λέξη



- Ορίζουμε μία διαδικασία όταν θέλουμε να δημιουργήσουμε μία νέα εντολή (αυτό συμβαίνει μία φορά). Εκτελούμε μία εντολή όσες φορές θέλουμε
- Όταν κάνουμε αλλαγές σε μία διαδικασία χρειάζεται να δηλώσουμε στη Logo ότι θέλουμε η διαδικασία αυτή να έχει τις αλλαγές που περιγράψαμε. Γι' αυτό μετά από κάθε αλλαγή θα πρέπει να επιλέγουμε όλη τη διαδικασία με τον κέρσορα και να πιέσουμε το Insert ή το play. Η Logo έχει ενημερωθεί για τις αλλαγές στο περιεχόμενο της νέας εντολής μόνο όταν εμφανιστεί στο κάτω μισό της ψηφίδας Logo το μήνυμα: όνομαΔιαδικασίας redefined
- Για να βρούμε τις εντολές Logo που ακούει κάθε ψηφίδα χρειάζεται να πληκτρολογήσουμε την εξής εντολή: ask "ΌνομαΨηφίδας [primitives]

### 3.5.6 Ιδέες για αξιοποίηση -επέκταση των παραδειγμάτων που παρουσιάστηκαν

Στην παρουσίαση των βασικών στοιχείων του προγραμματισμού με Logo ξεκινήσαμε με κάποια απλά παραδείγματα προγραμματισμού που αφορούσαν την ψηφίδα κείμενο και ήταν ενδεικτικά για να γίνουν κατανοητές οι βασικές έννοιες που διαπραγματευόμασταν. Επειδή όμως αυτό το εγχειρίδιο έχει ως στόχο την υποστήριξη στη διαδικασία σχεδιασμού εκπαιδευτικού λογισμικού και όχι απλώς την παρουσίαση των τεχνικών χαρακτηριστικών και δυνατοτήτων, θα επιχειρήσουμε να συζητήσουμε εδώ πώς τα συγκεκριμένα παραδείγματα με την ψηφίδα Κείμενο θα μπορούσαν να επεκταθούν ώστε να έχουν κάποιου είδους εκπαιδευτικό ενδιαφέρον. Να επισημανθεί ότι οι επεκτάσεις που θα συζητηθούν εδώ δεν αποτελούν έτοιμες ιδέες για εκπαιδευτικό λογισμικό αλλά δείχνουν κυρίως κατευθύνσεις και τρόπο σκέψης.

#### 3.5.6.1 Προοδευτική αλλαγή χρώματος στη φωτεινή επιγραφή

Όπως είδαμε στα προηγούμενα παραδείγματα η εναλλαγή χρώματος βασιζόταν σε αποχρώσεις που είχαμε περιγράψει εμείς. Ενδεχομένως να είχαμε καλύτερο αισθητικό αποτέλεσμα αν για παράδειγμα ξεκινούσαμε από μία απόχρωση που εξασθενούσε ή σκούραινε σιγά-σιγά. Δεδομένου ότι το κάθε χρώμα ορίζεται από τρεις αριθμούς που ξεκινάνε από το 0 και καταλήγουν στο 255, φαίνεται ότι υπάρχει αρκετό πεδίο να δουλέψει κανείς με απλές ίσως μαθηματικές πράξεις και σχέσεις έως πιο σύνθετες.

Παρακάτω παραθέτουμε μία διαδικασία η οποία αλλάζει το χρώμα του φόντου ξεκινώντας από το κίτρινο περνάει από το πορτοκαλί και καταλήγει στο κόκκινο

για ΑλλαγήΧρώματος :a

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "|Αλλάζοντας χρώμα....|]
```

```
until [:a < 5]
```

```
[localmake "textlist fput 255 (fput :a [30])
```

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ :textlist]
```

```
make "a :a-10
```

```
wait 200]
```

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "|Αυτό ήταν όλο|]
```



τέλος

#### Διαδικασία 12

Η διαδικασία αυτή ουσιαστικά μειώνει κατά 10 τη δεύτερη τιμή που καθορίζει το χρώμα ενώ αφήνει τις άλλες δύο σταθερές η πρώτη είναι 255 και η τρίτη 30. Εκτελέστε τη διαδικασία δίνοντας ως αρχική τιμή για τη μεταβλητή :a το 255. Έτσι το αρχικό χρώμα θα είναι 255 255 30, το επομενο 255 245 30 κ.ο.κ Μόλις η δεύτερη τιμή του χρώματος γίνει μικρότερη από 5 η διαδικασία αλλαγής χρώματος σταματάει (. Έχει προβλεφθεί αυτός ο όρος επειδή η δεύτερη τιμή δεν μπορεί να πάρει τιμές μικρότερες από το 0 και αν γίνει αυτό η Logo θα βγάλει σφάλμα).

Στην επόμενη διαδικασία έχει γίνει προσπάθεια να ακολουθηθεί αντίθετη πορεία για τα γράμματα του κειμένου: όσο σκουραίνει το χρώμα του φόντου τόσο να ανοίγει το χρώμα των γραμμάτων.

για ΑλλαγήΧρώματος2 :a

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "|Αλλάζοντας χρώμα....|]
```

```
until [:a < 5]
```

```
[localmake "bglist fput 255 (fput :a [30])
```

```
localmake "tlist lput 45 (fput :a [255])
```

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ :bglist]
```

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΠΡΟΣΚΗΝΙΟΥ :tlist]
```

```
make "a :a-10
```

```
wait 200]
```

```
ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΕΚΕΙΜΕΝΟ "|Αυτό ήταν ολο|]
```

τέλος

#### Διαδικασία 13

Θα μπορούσε κανείς να δουλέψει εδώ με τις σχέσεις μεταξύ των αριθμών π.χ. να δει τι χρώματα θα προκύψουν καθώς αυξάνεται η πρώτη τιμή του χρώματος ενώ ταυτόχρονα μειώνεται η δεύτερη με τον ίδιο ρυθμό κ.ο.κ.

Οι δύο παραπάνω διαδικασίες έχουν οριστεί και δουλεύουν στο μικρόκοσμο welcome procedures.mwd

#### Πρόταση για επέκταση:

Μικρόκοσμος: welcome\_slider.mwd

Στο μικρόκοσμο welcome\_slider και οι τρεις τιμές που καθορίζουν το χρώμα ελέγχονται από τρεις ολισθητές. Παρατηρήστε σε αυτό το μικρόκοσμο ότι η χρήση των ολισθητών, η οποία επιτρέπει αυτό που λέμε δυναμικό χειρισμό του χρώματος, φέρνει στο επίκεντρο το ρόλο της κάθε μίας τιμής στην αλλαγή του χρώματος. Ενώ κάτι τέτοιο δεν είναι τόσο κεντρικό στους προηγούμενους μικρόκοσμους. Οι διαδικασίες που έχουν χρησιμοποιηθεί είναι οι ακόλουθες:

## Εργαστήριο εκπαιδευτικής Τεχνολογίας

*το χρώμα\_Ολισθητής :a :b :c*

*Localmake "τιμέςΧρώματος fput :a (fput :b (fput :c []))*

*ask "Κείμενο [ΚΕΙΜΕΝΟ.ΘΕΣΧΡΩΜΑΥΠΟΒΑΘΡΟΥ :τιμέςΧρώματος]*

*end*

### Διαδικασία 14

*το ΤιμεςΑποΟλισθητή*

*localmake "a ask "Ολισθητής [slider.value]*

*localmake "b ask "Ολισθητής\_1 [slider.value]*

*localmake "c ask "Ολισθητής\_2 [slider.value]*

*χρώμα\_Ολισθητής :a :b :c*

*end*

### Διαδικασία 15

Η πρώτη διαδικασία δίνει τη δυνατότητα το χρώμα υποβάθρου του κειμένου να ορίζεται από μεταβλητές. Η δεύτερη διαδικασία με το όνομα ΤιμεςΑποΟλισθητή δίνει σε κάθε τιμή του χρώματος την τρέχουσα τιμή του ολισθητή και στη συνέχεια εκτελεί τη διαδικασία 14 με τις τρέχουσες τιμές των τριών Ολισθητών.

Για να πετύχουμε όμως να αλλάζει το χρώμα του φόντου του κειμένου μετακινώντας τον κέρσορα του ολισθητή χρησιμοποιήσαμε τα Γεγονότα (βλ. 3.4.1) της ψηφίδας ολισθητής και συγκεκριμένα το γεγονός Αλλαγή Κατάστασης όπου ζητήσαμε κάθε φορά που αλλάζει η κατάσταση του Ολισθητή η Logo να εκτελεί τη Διαδικασία 15. Αλλαγή κατάστασης για τον Ολισθητή δημιουργείται καθώς μετακινούμε τον κέρσορα του Ολισθητή. Το γεγονός αλλαγή κατάστασης είναι ενεργοποιημένο και στους τρεις ολισθητές έτσι ώστε όταν αλλάζει οι τιμή του ενός να εκτελείται η διαδικασία 15 και να παίρνει την τρέχουσα τιμή όλων των ολισθητών.

Ιδέες για τροποποίηση του μικρόκοσμου welcome\_slider.mwd: Δοκιμάστε οι δύο από τις τρεις τιμές που καθορίζουν το χρώμα να προκύπτει σε συνάρτηση με την τρίτη. Αυτό μπορεί να γίνει με δύο τρόπους είτε χρησιμοποιώντας μία μεταβλητή μόνο την :a και άρα μόνο έναν ολισθητή. Σε αυτή την περίπτωση αντικαταστήστε τη γραμμή 2 της διαδικασίας 14 με την παρακάτω γραμμή

*Localmake "τιμέςΧρώματος fput :a (fput :a/2 (fput :a\*2 []))*

### Γραμμή κώδικα 8

Προσοχή σε αυτή την περίπτωση μετά το όνομα της διαδικασίας θα βάλετε μόνο τη μεταβλητή :a και όχι τις :b και :c (τις δύο τελευταίες θα τις σβήσετε).

Ένας άλλος τρόπος να κάνετε κάτι αντίστοιχο είναι μέσω των τριών ολισθητών όπου ο δεύτερος και ο τρίτος ολισθητής θα παίρνουν τιμές που θα σχετίζονται με τον πρώτο.

### 3.5.6.2 Δημιουργία θερμόμετρου

Χρησιμοποιήσετε έναν ολισθητή για να φτιάξετε κάτι σαν θερμόμετρο, αλλάζοντας το φόντο του ολισθητή αυτή τη φορά αντί για το φόντο του κειμένου (αντί για ΚΕΙΜΕΝΟ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ χρησιμοποιήστε την εντολή ΟΛΙΣΘΗΤΗΣ.ΘΕΣΕΧΡΩΜΑΥΠΟΒΑΘΡΟΥ). Ο στόχος είναι μετακινώντας τον κέρσορα του ολισθητή το χρώμα θα πρέπει να ξεκινάει από μπλε για τις μικρές τιμές και να καταλήγει σε κόκκινο για τις μεγάλες.

Για να το κάνετε να μοιάζει πιο πολύ με θερμόμετρο θα πρέπει να χρησιμοποιήσετε το όριο τιμών που έχει ένα θερμόμετρο (τώρα οι ολισθητές έχουν ως όριο τιμών τις τιμές που παίρνει το χρώμα από 0 -255). Ο απλούστερος τρόπος είναι να μη χρησιμοποιήσετε μεταβλητές για τις τιμές του χρώματος αλλά να πείτε ότι αν η τιμή του ολισθητή είναι μικρότερη από μηδέν βαθμούς βάλτε χρώμα υποβάθρου άσπρο, αν είναι μικρότερη από 10 βάλτε μπλε κ.ο.κ. Στην περίπτωση αυτή όμως τα χρώματα δεν θα αλλάζουν ομαλά όπως με τη χρήση των ολισθητών και των μεταβλητών. (για να δείτε πως συντάσσεται η εντολή if συμβουλευτείτε τη λίστα εντολών της Logo του D.Azuma)

### 3.5.6.3 Δημιουργία χρωμο-ιστοριών

Μικρόκοσμος: colouredstories.mwd

Η δυνατότητα αλλαγής χρώματος στο φόντο και στα γράμματα έχει αξιοποιηθεί στο μικρόκοσμο colouredstories.mwd . Η λογική εδώ είναι η εξής: οι τρεις ολισθητές χρησιμεύουν για να αποτυπωθεί χρωματικά το ύφος ή η αίσθηση που δημιουργεί κάθε ένα από τα κείμενα που εμφανίζονται στην ψηφίδα κείμενο. Στο τέλος τα κείμενα μπορούν να προβληθούν διαδοχικά με τη χρωματολογία που έχει ορίσει ο κάθε χρήστης.

Ο μικρόκοσμος αυτός είναι ανοιχτός στην αλλαγή ή στην προσθήκη κειμένων με τη μορφή txt αρχείων. Για να προσθέσετε κείμενα: α) αντιγράψτε το κείμενο που θέλετε να ενσωματωθεί στο μικρόκοσμο στο σημειωματάριο (notebook) και αποθηκεύστε το. Επαναλάβετε τη διαδικασία για κάθε διαφορετικό κείμενο-ιστορία. Τοποθετήστε τα αρχεία σας στο φάκελο texts που βρίσκεται στο φάκελο Συνοδευτικό υλικό. Στη συνέχεια ανοίξτε το μικρόκοσμο colouredstories.mwd, βρείτε την ψηφίδα TextBase και βάλτε στο πεδίο ΚΕΙΜΕΝΑ τα ονόματα των αρχείων με τις ιστορίες που έχετε αποθηκεύσει ως εξής: texts/βοριάς-ήλιος.TXT

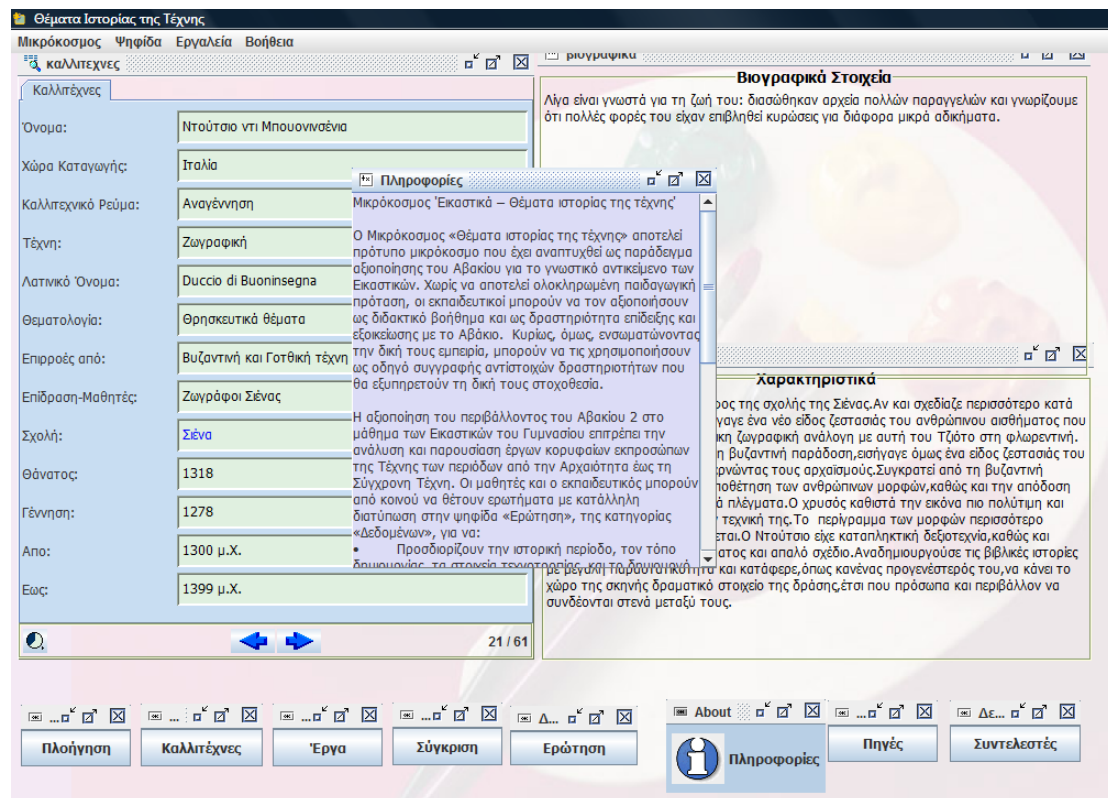
Ο συγκεκριμένος μικρόκοσμος παρουσιάζεται εδώ όχι για να δείξουμε ότι μπορούμε μόνοι μας να υλοποιήσουμε μία λειτουργικότητα που ήδη έχει ο κειμενογράφος, αλλά για να δείξουμε κυρίως πώς μία αρχική ιδέα στο Αβάκιο μπορεί να μετεξελιχθεί έτσι ώστε να δίνεται έμφαση σε διαφορετικές έννοιες- στοιχεία. Ειδικότερα ξεκινήσαμε από την αρχική ιδέα της φωτεινής επιγραφής που ο στόχος ήταν να την κάνουμε να δουλέψει, στη συνέχεια δώσαμε έμφαση στον τρόπο με τον οποίο αλλάζουν τα χρώματα στη Logo ειδικά αλλά και στους υπολογιστές ειδικότερα και εκεί εξετάστηκε το ενδεχόμενο της απεικόνισης των σχέσεων των αριθμών με χρώματα. Για το μικρόκοσμο των χρωμο-ιστοριών δώσαμε έμφαση όχι στο πως αλλάζουν τα χρώματα αλλά στο πως θα περάσουμε από μία λέξη ή φράση σε διαφορετικά κείμενα με στόχο να εμβαθύνουμε στη χρωματική αποτύπωση του ύφους του κειμένου (στο πλαίσιο της πολυτροπικότητας).

## 4 Ενσωματώνοντας στο μικρόκοσμό μας έτοιμα κομμάτια από άλλον μικρόκοσμο

Μικρόκοσμος: HistoryArt.mwd

Λέξεις κλειδιά: Αντιγραφή – επικόλληση ψηφίδων, εντολή logo.execute

Ας υποθέσουμε ότι θέλουμε να κατασκευάσουμε ένα δικό μας μικρόκοσμο που να περιέχει ένα κουμπί προβολής πληροφοριών όπως αυτόν που βρίσκεται στο μικρόκοσμο Θέματα Ιστορίας της Τέχνης (όνομα μικρόκοσμου HistoryArt.mwd στο φάκελο Συνοδευτικό Υλικό)<sup>9</sup>.



Εικόνα 4.1 : Εμφάνιση πληροφοριών με το πάτημα στο σημείο με την ένδειξη “i”

Για να μπορέσουμε να «ξεκλειδώσουμε» αυτό το μικρόκοσμο αφού τον έχουμε ανοίξει πρώτα στο περιβάλλον του Αβακίου και να δανειστούμε από αυτόν τα κομμάτια που μας ενδιαφέρουν έχουμε δύο επιλογές: Η μία είναι να πάρουμε αυτούσια τα κομμάτια αυτά και να τα αντιγράψουμε στο μικρόκοσμό μας και η άλλη είναι να πάρουμε την πληροφορία σχετικά με αυτά τα κομμάτια και να τα προσαρμόσουμε στο μικρόκοσμό μας. Είτε ακολουθήσουμε τη μία τακτική είτε την άλλη και στις δύο περιπτώσεις χρειάζεται να ξέρουμε τί ψηφίδα είναι αυτή που μας ενδιαφέρει και ποιο είναι το όνομά της. Για να το βρούμε αυτό χρειάζεται να κάνουμε τα ακόλουθα βήματα:

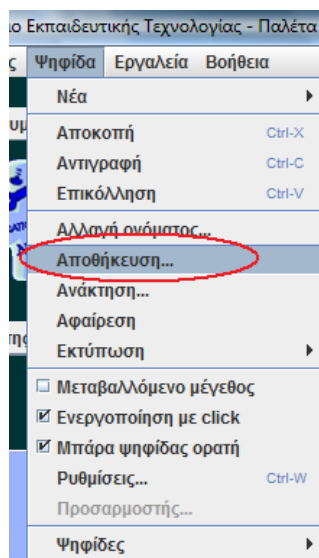
<sup>9</sup> Αρχείο History Art στο φάκελο του συνοδευτικού υλικού του εγχειριδίου.

## 4.1 Παίρνοντας αυτούσιες ορισμένες ψηφίδες

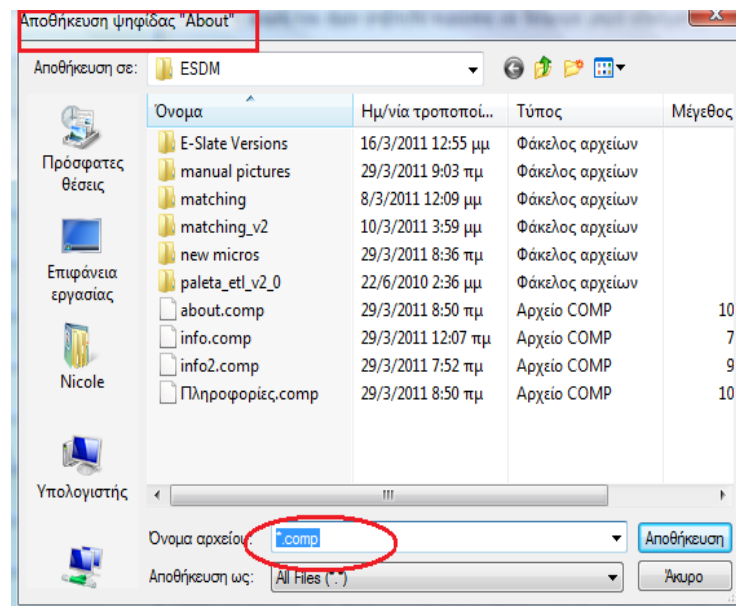
Όπως αναφέρθηκε νωρίτερα ένας τρόπος για να «πάρουμε» μία επιθυμητή λειτουργικότητα από έναν μικρόκοσμο και να τη βάλουμε στο δικό μας είναι να αποθηκεύσουμε τις ψηφίδες που εμπλέκονται στη λειτουργικότητα που μας ενδιαφέρει. Στη συγκεκριμένη περίπτωση, αυτό που θέλουμε είναι: κάνοντας κλικ στο κουμπί Πληροφορίες (ψηφίδα «about») να εμφανίζεται το πλαίσιο που παρουσιάζει αυτές τις πληροφορίες και έχει το όνομα «Πληροφορίες».

Άρα αυτό που μας ενδιαφέρει στην παρούσα περίπτωση είναι να πάρουμε την ψηφίδα “about” και την ψηφίδα «Πληροφορίες» και να τις εντάξουμε στο δικό μας μικρόκοσμο. Για να αντιγράψουμε λοιπόν την ψηφίδα κάνουμε κλικ πάνω στη μπάρα της ώστε να είναι ενεργή και στη συνέχεια πηγαίνουμε στο μενού Ψηφίδα και επιλέγουμε Αποθήκευση (βλ. Εικόνα 4.2) ΕΝΑΛΛΑΚΤΙΚΑ μπορούμε να ενεργοποιήσουμε την ψηφίδα κάνοντας κλικ πάνω της, στη γραμμή εργασίας.

Αφού επιλέξουμε Αποθήκευση ψηφίδας εμφανίζεται στην οθόνη μας το παράθυρο της . Εικόνα 4.3. Σημαντικό σε αυτό το σημείο είναι να προσέξουμε ότι στο πάνω μέρος του παραθύρου μας λέει **ποια ψηφίδα αποθηκεύει** (εδώ λέει Αποθήκευση Ψηφίδας “About”). Είναι πάντοτε πολύ σημαντικό να βλέπουμε σε αυτό το παράθυρο το όνομα της ψηφίδας που Αποθηκεύεται γιατί συχνά, μπορεί καθώς μετακινούμε το ποντίκι μας προς το μενού της ψηφίδας να ενεργοποιήσουμε άλλες ψηφίδες και αν πιάσουμε τότε το κουμπί αποθήκευση, να αποθηκεύσουμε κάποια άλλη ψηφίδα. Σε περίπτωση που διαπιστώσουμε ότι έχει ενεργοποιηθεί άλλη ψηφίδα, ακυρώνουμε την αποθήκευση και επιχειρούμε την ενεργοποίηση της ψηφίδας είτε ξανά μέσω της μπάρας είτε μέσω της γραμμής εργασίας, αποφεύγοντας να περάσουμε το ποντίκι μας πάνω από τις άλλες ψηφίδες.



Εικόνα 4.2: Αποθήκευση ψηφίδας



Εικόνα 4.3: Παράθυρο αποθήκευσης

Η κατάληξη της ψηφίδας που αποθηκεύεται είναι «.comp» (π.β. την κατάληξη .mwd που αφορά αρχεία μικρόκοσμου και την κατάληξη .cdb που αφορά τις βάσεις δεδομένων).

Πριν την κατάληξη γράφουμε το όνομα που μας ενδιαφέρει (π.χ. about ) και πιέζουμε το κουμπί Αποθήκευση. Η ψηφίδα έχει αποθηκευτεί με τη μορφή και τις λειτουργικότητες που είχε στον αρχικό μικρόκοσμο. Κάνουμε το ίδιο και με την ψηφίδα που έχει το όνομα Πληροφορίες (δώστε το όνομα info). Για διευκόλυνσή σας και σε περίπτωση που δεν πήγε κάτι καλά μπορείτε να βρείτε στο φάκελο components τις δύο παραπάνω ψηφίδες αποθηκευμένες με τα ονόματα about.comp και Plirofories.comp

### **4.2 Ενσωματώνοντας τις ψηφίδες και τις λειτουργικότητές τους στο μικρόκοσμό μας**

Ξεκινάμε με την εξής ερώτηση: αν βάλουμε την αποθηκευμένη ψηφίδα Info στο νέο μικρόκοσμο που θέλουμε να φτιάξουμε θα δουλέψει; Δηλ. αν κάνουμε κλικ πάνω της θα εμφανιστεί το κείμενο με τις οδηγίες;

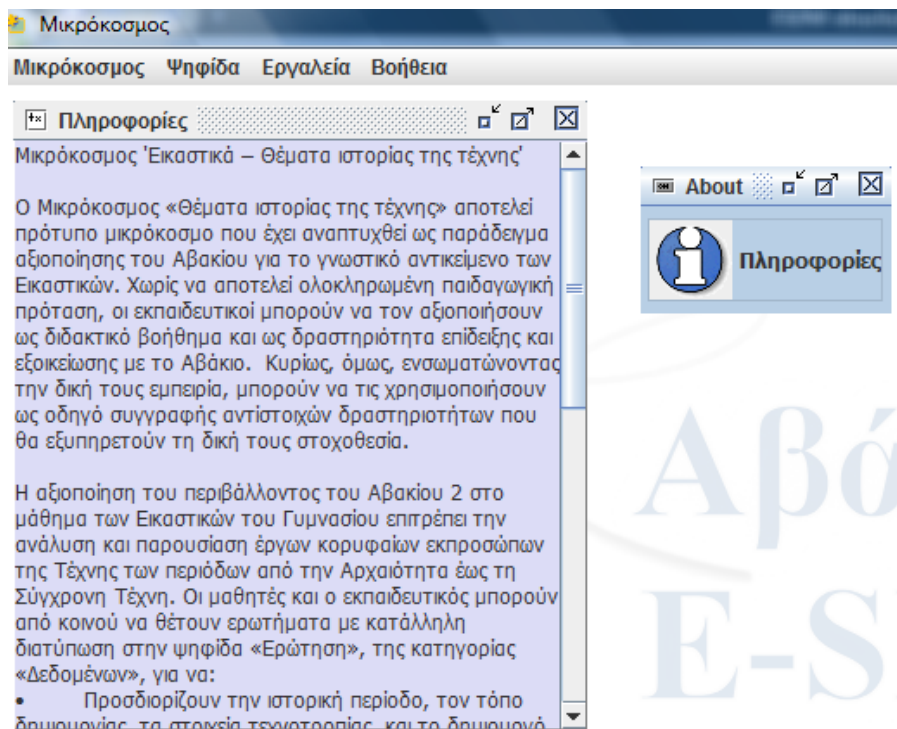
Η απάντηση δεν είναι ίδια σε όλες τις περιπτώσεις. Η επιλογή που έχουμε κάνει εδώ είναι τέτοια που η απάντηση στο συγκεκριμένο ερώτημα είναι «ναι» Υπάρχουν όμως περιπτώσεις που εισάγοντας τις ψηφίδες στο νέο μικρόκοσμο δεν έχουμε μεταφέρει και τη λειτουργικότητά τους και τις οποίες θα εξετάσουμε στη συνέχεια.

#### **4.2.1 Ενσωμάτωση των αποθηκευμένων ψηφίδων**

Κλείνουμε το μικρόκοσμο «Θέματα Ιστορίας Τέχνης» πηγαίνοντας στο Μενού Μικρόκοσμος -> Κλείσιμο. Από το ίδιο μενού επιλέγουμε «Νέος» για να δημιουργηθεί ο νέος μικρόκοσμος. Επιλέγουμε από το μενού Μικρόκοσμος «Αποθήκευση ως» και δίνουμε το όνομα «New-Info».

Στη συνέχεια πηγαίνουμε στο μενού Ψηφίδα και επιλέγουμε «Ανάκτηση». Από το παράθυρο που εμφανίζεται εντοπίζουμε το φάκελο στον οποίο αποθηκεύσαμε τις ψηφίδες μας επιλέγουμε πρώτα τη μία και μετά την άλλη. Βλέπουμε τώρα τις δύο ψηφίδες από το μικρόκοσμο « Θέματα Ιστορίας της Τέχνης» να έχουν ενταχθεί στο νέο μικρόκοσμο που φτιάξαμε.





Εικόνα 4.4 Ο νέος Μικρόκοσμος με τις ανακτημένες ψηφίδες

Κάνουμε κλικ πάνω στην ψηφίδα "About" και βλέπουμε ότι η λειτουργικότητα που είχε στο μικρόκοσμο Θέματα Ιστορίας της Τέχνης έχει μεταφερθεί και στο δικό μας μικρόκοσμο. ( δηλ. με ένα κλικ εξαφανίζεται και με δεύτερο κλικ εμφανίζεται το πλαίσιο με τις πληροφορίες)

Επειδή το πιθανότερο είναι πως οι πληροφορίες που θέλουμε να δείξουμε σε ένα μικρόκοσμο που κατασκευάσαμε εμείς να είναι διαφορετικές από αυτές που προβάλλονται στον αποθηκευμένο μικρόκοσμο ακολουθήστε τη διαδικασία που περιγράφεται στο «Εγχειρίδιο Χρήσης Αβακίου II<sup>10</sup>» στην ενότητα «3.7 Επεξεργασία της μορφής των ψηφίδων» σελ. 16 – 23.για να προσαρμόσετε την ψηφίδα στο δικό σας μικρόκοσμο. Αν θέλετε απλώς να αλλάξετε το κείμενο στις περισσότερες περιπτώσεις μπορείτε να επέμβετε απλώς μέσα στην ψηφίδα, να σβήσετε ό,τι υπάρχει και να γράψετε το δικό σας κείμενο. Σε περίπτωση που θέλετε να προστατέψετε το κείμενό σας και να μην μπορεί κάποιος να παρέμβει σε αυτό πηγαίνετε στις ιδιότητες της ψηφίδας κείμενο και από-επιλέξτε την δυνατότητα «Επέμβαση στο κείμενο». Με τον ίδιο τρόπο μπορείτε και να την ενεργοποιήσετε αν κάποιο κείμενο δεν σας επιτρέπει να επέμβετε.

<sup>10</sup> Το εγχειρίδιο Χρήσης Αβακίου μπορείτε να το βρείτε αν κάνετε κλικ στο μενού Έναρξη -> Προγράμματα ->Αβάκιο-> Εγχειρίδιο Χρήσης Αβακίου II. Για να το ανοίξετε χρειάζεστε το πρόγραμμα Acrobat της Adobe

#### 4.2.2 3.4 Ανοίγοντας τις ψηφίδες για να βρούμε τις λειτουργικότητές τους ή τότε δεν φτάνει η αποθήκευση των ψηφίδων μόνο.

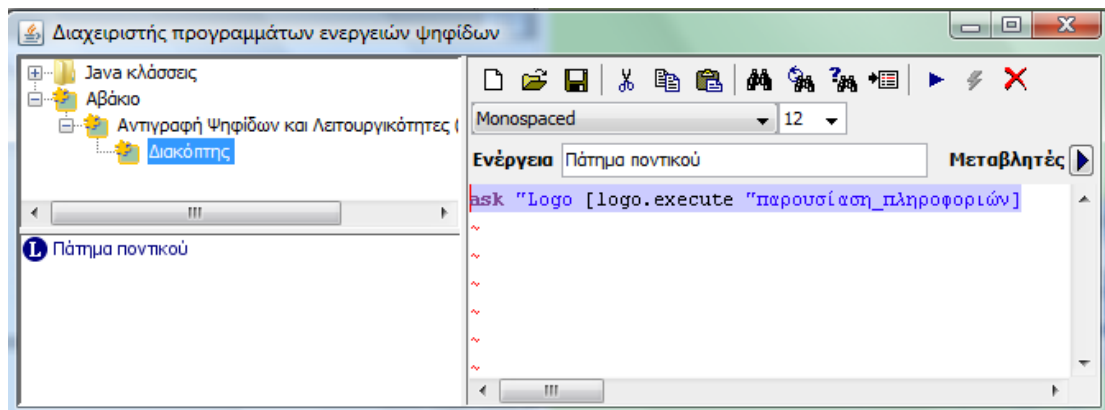
Αρχεία μικρόκοσμων: button-info.mwd, empty.mwd

Δοκιμάστε τώρα να κάνετε την διαδικασία που περιγράψαμε στην προηγούμενη ενότητα με το μικρόκοσμο button-info.mwd που βρίσκεται στο φάκελο Συνοδευτικό υλικό. Αντιγράψτε τις ψηφίδες «Διακόπτης» (έχει το σύμβολο i σε κύκλο και γράφει «ΕΜΦΑΝΙΣΗ ΠΛΗΡΟΦΟΡΙΩΝ») και την ψηφίδα που εμφανίζεται με τις πληροφορίες (είναι η ψηφίδα με το όνομα «Κείμενο»). Πιέστε πρώτα το κουμπί με τις πληροφορίες (i) και αφού εμφανιστεί το κείμενο κάνετε κλικ να εμφανιστεί η γραμμή εργασίας και στη συνέχεια επιλέξτε εμφάνιση μπαρών για όλες τις ψηφίδες. (Βλ. Ενότητα 3.3). Σε περίπτωση που δυσκολευτείτε θα βρείτε αυτές τις ψηφίδες αποθηκευμένες στο φάκελο components με τα ονόματα Keimeno.comp και Diakortis.comp. Στη συνέχεια εισαγάγετε αυτές τις ψηφίδες στο μικρόκοσμο «empty.mwd». Δοκιμάστε να πιέσετε το Διακόπτη και θα δείτε ότι δεν έχει καμία επίδραση πάνω στο Κείμενο και άρα δεν έχει μεταφερθεί η λειτουργικότητά του με την αποθήκευση. Αυτή λοιπόν είναι μία περίπτωση όπου η αποθήκευση μίας ψηφίδας δεν αρκεί για να μεταφερθεί το σύνολο των λειτουργικοτήτων της.

Το πρώτο πράγμα που θα προσπαθήσουμε να κάνουμε είναι να επιχειρήσουμε να διαπιστώσουμε τι έχει συμβεί εδώ. Γι' αυτό θα πάμε στις Διαχείριση ψηφίδων και από κει στα Γεγονότα της ψηφίδας «Διακόπτης» (βλ. Ενότητα 3.4) και θα κάνουμε κλικ στο κουμπί με το L που αφορά το γεγονός «Πάτημα Ποντικιού». Στο παράθυρο του Διαχειριστή Ενεργειών Ψηφίδων που εμφανίζεται βλέπουμε την εξής γραμμή κώδικα (βλ. και Εικόνα 4.5):

```
ask "Logo [logo.execute "παρουσίαση_πληροφοριών]
```

Γραμμή κώδικα 9



Εικόνα 4.5: Ο κώδικας στα γεγονότα της ψηφίδας Διακόπτης

Με βάση τα όσα είπαμε στην προηγούμενη ενότητα (βλ. ενότητα 3.5.1 κ.εξ) η συγκεκριμένη γραμμή κώδικα μας δείχνει τα εξής: Με το πάτημα του Ποντικιού η ψηφίδα



Διακόπτης ζητάει από την Ψηφίδα Logo να κάνει μία ενέργεια που βρίσκεται στην αγκύλη (να εκτελέσει τη διαδικασία «παρουσίαση\_πληροφοριών»). Εδώ ακριβώς βρίσκεται η αρχή του προβλήματος: Στο μικρόκοσμο empty δεν υπάρχει η ψηφίδα Logo. Ας βάλουμε λοιπόν στο μικρόκοσμο empty μία ψηφίδα Logo (Από το μενού Ψηφίδα->Νέα->Logo->Logo. Λύνεται το πρόβλημά μας; Η απάντηση είναι όχι. Την εξήγηση για την απάντηση αυτή μας τη δίνει η εντολή logo.execute. Η εντολή αυτή ουσιαστικά μπορεί να εκτελέσει οποιαδήποτε διαδικασία έχει οριστεί και είναι γραμμένη στην ψηφίδα Logo αρκεί μετά την εντολή Logo.execute να αφήσουμε κενό και να γράψουμε το όνομα της διαδικασίας με εισαγωγικό μπροστά (βλ Ενότητα.3.5.1 για τη χρήση των Εισαγωγικών και τη σημασία των κενών στις εντολές της Logo). Στην προκειμένη το όνομα της διαδικασίας είναι «παρουσίαση\_πληροφοριών» άρα αυτό που χρειάζεται να κάνουμε είναι α) να αποθηκεύσουμε το μικρόκοσμο με νέο όνομα π.χ. empty2.mwd να πάμε στο μικρόκοσμο απ' όπου πήραμε τις ψηφίδες δηλ. το μικρόκοσμο tbutton-info.mwd και να αντιγράψουμε την διαδικασία παρουσίαση πληροφοριών.

Ανοίξτε το μικρόκοσμο tbutton-info.mwd, εμφανίστε την ψηφίδα Logo, επιλέξτε τη διαδικασία από το "για" μέχρι το "τέλος" πηγαίνετε στο μενού Επεξεργασία της Logo και επιλέξτε αντιγραφή (προσοχή το ctrl+c δεν θα δουλέψει), κλείστε αυτό το μικρόκοσμο, επιλέγοντας άνοιγμα εκ νέου και ανοίξτε το μικρόκοσμο empty2.mwd

```
για παρουσίαση_πληροφοριών
ifelse ask "Διακόπτης [tbutton.selected]
[Ask "Κείμενο [restore] ask "Διακόπτης [ΔΙΑΚΟΠΤΗΣ.ΘΕΣΕΚΕΙΜΕΝΟ "ΙΑΠΟΚΡΥΨΗ
ΠΛΗΡΟΦΟΡΙΩΝ]]
[Ask "Κείμενο [hide] ask "Διακόπτης [ΔΙΑΚΟΠΤΗΣ.ΘΕΣΕΚΕΙΜΕΝΟ "ΙΕΜΦΑΝΙΣΗ
ΠΛΗΡΟΦΟΡΙΩΝ]]
τέλος
```

#### Διαδικασία 16

Στη συνέχεια πηγαίνετε στην ψηφίδα Logo του μικρόκοσμου empty2.mwd πηγαίνετε στο μενού Επεξεργασία της Logo και επιλέξτε «Επικόλληση». Στη συνέχεια ορίστε τη διαδικασία (Επιλέγοντάς την και πιέζοντας το κουμπί insert βλ. και ενότητα 3.5.2). Δοκιμάστε τώρα να χρησιμοποιήσετε το διακόπτη που εμφανίζει και κρύβει τις πληροφορίες. Θα δείτε ότι τώρα δουλεύει.

### 4.2.3 Ερωτήματα και θέματα που χρειάζεται να προσέξει κανείς

**Αντιγραφή της ψηφίδας Logo αντί για τη συγκεκριμένη διαδικασία:** Ένα πιθανό ερώτημα θα ήταν αν θα δούλευε ο διακόπτης σε περίπτωση που αντιγράφαμε από τον αρχικό μικρόκοσμο (tbutton-info.mwd) την ψηφίδα Logo και όχι απλώς τη διαδικασία – στο νέο μικρόκοσμο empty2.mwd. Η απάντηση είναι πως ο Διακόπτης θα δούλευε. Ωστόσο υπάρχουν τα εξής ζητήματα: συνήθως στην ψηφίδα Logo ενός μικρόκοσμου δεν υπάρχει μόνο μία διαδικασία αλλά πολύ περισσότερες που δεν συνδέονται με τη λειτουργία του διακόπτη έτσι αντιγράφοντας όλη την ψηφίδα Logo θα ήταν σαν να χρειάζοσασταν μία παράγραφο από ένα κείμενο και να αντιγράφατε ολόκληρο το βιβλίο.

## Εργαστήριο εκπαιδευτικής Τεχνολογίας

**Προσοχή στα ονόματα των ψηφίδων που καλούν οι διαδικασίες:** Αν για παράδειγμα αποφασίζετε να μετονομάσετε την ψηφίδα Κείμενο που έχετε αντιγράψει και να της δώσετε το όνομα «Info» τότε θα πρέπει να αλλάξετε τη διαδικασία έτσι ώστε να αναφέρεται στην ψηφίδα με το νέο της όνομα (Μην ξεχάσετε να ορίσετε εκ νέου τη διαδικασία αν κάνετε αλλαγές):

*για παρουσίαση\_πληροφοριών*

*ifelse ask "Διακόπτης [tbutton.selected]*

*[Ask "Info [restore] ask "Διακόπτης [ΔΙΑΚΟΠΤΗΣ.ΘΕΣΕΚΕΙΜΕΝΟ "ΙΑΠΟΚΡΥΨΗ ΠΛΗΡΟΦΟΡΙΩΝ]]*

*[Ask " Info [hide] ask "Διακόπτης [ΔΙΑΚΟΠΤΗΣ.ΘΕΣΕΚΕΙΜΕΝΟ "ΙΕΜΦΑΝΙΣΗ ΠΛΗΡΟΦΟΡΙΩΝ]]*

*τέλος*

### Διαδικασία 17

Αντίστοιχες ενέργειες θα έπρεπε να κάνετε και για την ψηφίδα Διακόπτης ή για την ψηφίδα Logo σε περίπτωση που τα ονόματά τους ήταν διαφορετικά από αυτά που χρησιμοποιούνται στην διαδικασία ή στα γεγονότα των ψηφίδων.

**Προσοχή στις υπερδιαδικασίες:** Υπάρχει περίπτωση μία διαδικασία που ενεργοποιείται με τα γεγονότα μίας ψηφίδας να είναι υπερδιαδικασία, να καλεί δηλαδή άλλες διαδικασίες που έχουν οριστεί μέσα στην ψηφίδα Logo. Τότε σε αυτή την περίπτωση θα πρέπει να αντιγράψετε όχι μόνο την υπερδιαδικασία αλλά και τις διαδικασίες που καλεί. Ας το δούμε αυτό μέσα από ένα παράδειγμα.

## 5 Μαστορεύοντας μισοψημένους μικρόκοσμους

Στην ενότητα αυτή θα συζητήσουμε πώς θα αξιοποιήσουμε όσα αναφέρθηκαν στις προηγούμενες ενότητες για να ανοίξουμε, επεκτείνουμε, αλλάξουμε αυτό που ονομάζουμε «μισοψημένο μικρόκοσμο». Ένας μισοψημένος μικρόκοσμος διαθέτει τα χαρακτηριστικά του μικρόκοσμου (Edwards 1998, Hoyles 1995), δηλαδή ενσωματώνει ένα μέρος του προς διδασκαλία αντικείμενου και προσφέρει στο μαθητή τα εργαλεία για να αλληλεπιδράσει με το μικρόκοσμο προκειμένου να διερευνήσει το γνωστικό αντικείμενο. Το κύριο χαρακτηριστικό όμως των μισοψημένων μικρόκοσμων είναι ότι έχουν σχεδιαστεί ώστε να «προκαλούν» επιμορφούμενους εκπαιδευτικούς ή μαθητές να τους αλλάξουν (Kynigos 2004, Kynigos 2007). Δεν πρόκειται δηλαδή για λογισμικά τα οποία είναι έτοιμα για χρήση από τους μαθητές, αντίθετα πρόκειται για εργαλεία σχεδιασμένα να λειτουργήσουν ως γεννήτριες ιδεών και ως αφητηρία για αλλαγή και βελτίωση (Κυνηγός, Γιαννούτσου, & Φράγκου, 2006).

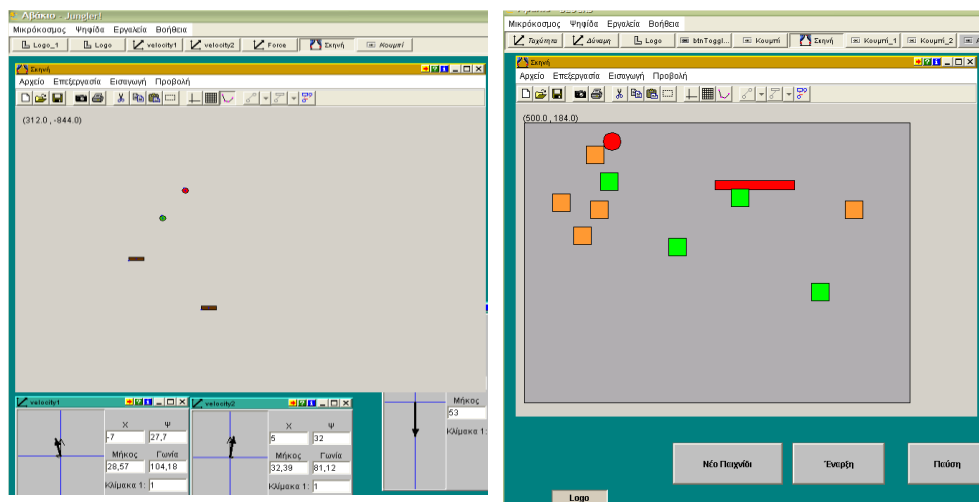
**ΠΡΟΣΟΧΗ:** μισοψημένος μικρόκοσμος δεν είναι οποιοσδήποτε μικρόκοσμος είναι ημιτελής με την έννοια ότι είναι άδειος για παράδειγμα από περιεχόμενο, ή έχει περιορισμένες λειτουργικότητες. Το αντίθετο μάλιστα ένας μισοψημένος μικρόκοσμος δείχνει με πληρότητα τη λογική του, το περιεχόμενο και τις λειτουργικότητές του τα οποία όμως έχουν σχεδιαστεί ώστε να προκαλούν για την αλλαγή τους αλλά και την ίδια

στιγμή να αποτελούν τη βάση για συζήτηση, ανταλλαγή απόψεων και διαπραγματεύση με στόχο την αλλαγή. Ένα ακόμη χαρακτηριστικό των μισοψημένων μικρόκοσμων είναι ότι διαθέτουν προσεκτικά σχεδιασμένα εργαλεία που επιτρέπουν στο χρήστη να παρέμβει και να αλλάξει το μικρόκοσμο. Έτσι, οι μικρόκοσμοι που προκύπτουν από ένα μισοψημένο μικρόκοσμο δεν είναι εκφάνσεις του αρχικού το οποίο αποτελεί μία αφαίρεση (όπως συμβαίνει με τα πρότυπα templates π.β. π.χ. το μικρόκοσμο ταξινομούμε) αλλά είναι διαφορετικοί μικρόκοσμοι είτε ως προς το περιεχόμενο, είτε ως προς τις λειτουργικότητες, είτε ως προς τη λογική που τους διέπει

## 5.1 Από το Ζογκλέρ στο Arcanoid

**Μικρόκοσμοι:** juggler.mwd, touvlakia.mwd

Σε αυτή την ενότητα παρουσιάζουμε ένα παράδειγμα μετατροπής του μισοψημένου μικρόκοσμου Ζογκλέρ σε ένα παιχνίδι Arcanoid (παιχνίδι με τουβλάκια). Ο Ζογκλέρ λειτουργεί ως εξής: ο παίκτης μπορεί να μετακινεί τα χέρια του ζογκλέρ (2 ρακέτες) προκειμένου να μην του πέσει κάποια από τις δύο μπάλες με τις οποίες παίζει (βλ. Εικόνα 5.1).



Εικόνα 5.1 Μετατροπή του ζογκλέρ (αριστερά) στο παιχνίδι με τα τουβλάκια (δεξιά)

Για να δείξουμε πώς έγινε η μετατροπή θα περιγράψουμε πρώτα τις αλλαγές σε επίπεδο λειτουργικότητας και στην επόμενη ενότητα θα γίνει ανάλυση του κώδικα. Ειδικότερα, η λειτουργικότητα της σύγκρουσης της μπάλας με τη ρακέτα διατηρήθηκε και στο παιχνίδι με τα τουβλάκια. Η ίδια λειτουργικότητα χρησιμοποιήθηκε ως βάση για να δημιουργηθεί μία νέα που να περιγράφει τη σύγκρουση της μπάλας με τα τουβλάκια η οποία έχει την εξής λογική: όταν η μπάλα συναντάει οποιοδήποτε άλλο αντικείμενο, που δεν είναι ρακέτα, να το εξαφανίζει από την οθόνη. Αφαιρέθηκε από το παιχνίδι η δεύτερη μπάλα και η δεύτερη ρακέτα, όπως επίσης και η δυνατότητα ελέγχου της ταχύτητας της μπάλας μέσω των διανυσμάτων. Οι προσθήκες που έγιναν στο παιχνίδι ήταν ένας αριθμός από χρωματιστά τουβλάκια (ο ίδιος κάθε φορά που ξεκινούσε το παιχνίδι) και αδυναμία εξόδου της μπάλας από το σκούρο γκρι πλαίσιο που φαίνεται στην εικόνα δεξιά. Η τελευταία αυτή προσθήκη βασίζεται στην ίδια λογική με εκείνη της σύγκρουσης μόνο που στη συγκεκριμένη περίπτωση η μπάλα αλλάζει κατεύθυνση όχι μόνο όταν

συγκρούεται με τη ρακέτα αλλά και όταν ξεπερνάει κάποιο συγκεκριμένο σημείο πάνω στο πλαίσιο που κινείται. Στην περίπτωση του ζογκλέρ η μπάλα μπορούσε να εξαφανιστεί από το προσκήνιο αν δεν την απέκρουε η ρακέτα.

### 5.1.1 Ανάλυση κώδικα

.Εδώ κανονικά πρέπει να δείξουμε πώς τροποποιήθηκε ο αρχικός κώδικας και προέκυψαν τα τουβλάκια δηλ. πώς μετατράπηκε η σύγκρουση με ρακέτα στη σύγκρουση με τούβλο για παράδειγμα. Αυτό με τον κώδικα έτσι όπως είναι γραμμένος τώρα δεν μπορεί να γίνει γιατί το Arcanoid δεν είναι βασισμένο στον κώδικα Logo Style και η λογική που ακολουθεί είναι διαφορετική.

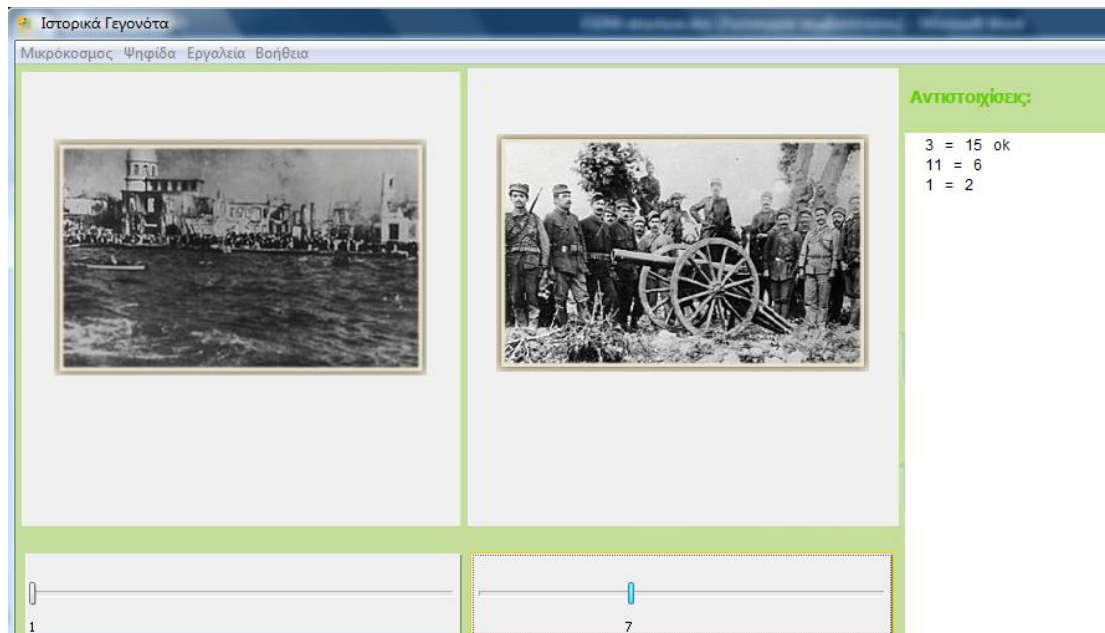
### 5.1.2 Ιδέες για επεκτάσεις – τροποποιήσεις

- Προσθήκη χρόνου: Αρχικά να μπει ένα χρονόμετρο που να ξεκινάει με το play που να δείχνει τη διάρκεια του παιχνιδιού. (προσθήκη ψηφίδας και έναρξη μετρήματος χρόνου μόλις πιέζεται το κουμπί «Έναρξη»)
- Να προστεθεί ένα script στον κώδικα έτσι ώστε να σταματάει το παιχνίδι σε συγκεκριμένο χρόνο και να προβάλλεται ένα σκορ ανάλογα με το πόσα τούβλα έχουν μείνει.
- Να τοποθετούνται τα τούβλα σε τυχαίες θέσεις στο χώρο και όχι το ένα πάνω στο άλλο.

## 5.2 Αντιστοιχήσεις: Από τα Ιστορικά Γεγονότα στις Διαφημίσεις

**Μικρόκοσμοι:** Istorika-gegonota.mwd, matching.mwd

Σε αυτή την ενότητα θα δείξουμε πώς από το μικρόκοσμο Ιστορικά Γεγονότα πήραμε ένα μέρος του (μία σειρά από λειτουργικότητες) τροποποιήσαμε λίγο κάποιες λειτουργικότητες για να δημιουργήσουμε κάτι άλλο. Ειδικότερα, ο μικρόκοσμος istorika-gegonota αποτελείται από δύο μέρη, στο πρώτο οι χρήστες δουλεύουν με το χάρτη και τις πληροφορίες που σχετίζονται με αυτόν ενώ στο δεύτερο παίζουν ένα παιχνίδι αντιστοίχισης εικόνων μεταξύ τους, εικόνων –χαρτών και εικόνων λεζαντών. Τα ιστορικά γεγονότα έχουν επίσης μία λειτουργία η οποία καταγράφει τις αντιστοιχήσεις που έχουν κάνει οι μαθητές και ελέγχει ποιες από αυτές είναι σωστές εμφανίζοντας ένα Ok δίπλα τους (βλ. Εικόνα 5.2). Αυτή ακριβώς τη λειτουργία που μας δίνει τη δυνατότητα να αντιστοιχίζουμε μεταξύ τους διαφορετικά πράγματα την ενσωματώσαμε σε ένα νέο μικρόκοσμο και επιχειρήσαμε να δούμε διαφορετικές εφαρμογές.



Εικόνα 5.2: Οι αντιστοιχίσεις στα ιστορικά Γεγονότα και ο έλεγχος των απαντήσεων

### 5.2.1 Πώς λειτουργεί η αντιστοίχιση

Στην ενότητα αυτή θα εξηγήσουμε το μηχανισμό της αντιστοίχισης για να γίνει καλύτερα κατανοητό πώς παρεμβαίνουμε σε αυτό το μικρόκοσμο και γιατί. Η αντιστοίχιση στα ιστορικά γεγονότα έχει τρεις μορφές: αντιστοίχιση εικόνων μεταξύ τους, αντιστοίχιση εικόνων κειμένου και αντιστοίχιση εικόνων – χαρτών. Κάθε ένα από αυτά τα είδη αντιστοιχίσεων έχει καταχωρηθεί σε έναν ξεχωριστό πίνακα σε μία βάση δεδομένων που μπορείτε να τη δείτε στο μικρόκοσμό σας αν εμφανίσετε από τα εργαλεία τη Γραμμή Εργασίας (μπορείτε να δείτε τη βάση και στην. Εικόνα 5.3).

Ας δούμε τώρα το περιεχόμενο της βάσης δεδομένων και το ρόλο της στο μικρόκοσμο: Όπως βλέπετε στην παρακάτω εικόνα η βάση αποτελείται από τρεις πίνακες (iconomap, iconoicon, iconotext). Κάθε πίνακας έχει δύο πεδία Imagefile1 και imagefile2. Στο πρώτο πεδίο (imagefile1) τοποθετούνται τα ονόματα αρχείων των εικόνων (μια και είμαστε στον πίνακα iconotext) και στο δεύτερο πεδίο (imagefile2) τοποθετούνται τα ονόματα των αρχείων των αντίστοιχων λεζαντών<sup>11</sup>. Για τα ιστορικά γεγονότα είναι σημαντικό η εικόνα να βρίσκεται στην ίδια εγγραφή με την αντίστοιχη της λεζάντα<sup>12</sup>, εικόνα, χάρτη γιατί σε αυτό ακριβώς το χαρακτηριστικό βασίζεται ο έλεγχος του αν η αντιστοίχιση που έχουν κάνει οι μαθητές είναι σωστή ή όχι. Η βάση δεδομένων σε αυτή την περίπτωση λοιπόν χρησιμεύει σαν ένας «Αποθηκευτικός χώρος» όπου έχουν τοποθετηθεί οι εικόνες που θα χρησιμοποιηθούν στο μικρόκοσμο.

<sup>11</sup> Το «image|post» που βλέπετε πριν το όνομα αρχείου της εικόνας είναι τα ονόματα των φακέλων μέσα στα οποία βρίσκονται όλες οι εικόνες.

<sup>12</sup> Προσοχή: για αυτό το μικρόκοσμο είναι σημαντικό οι λεζάντες και οι χάρτες να είναι αποθηκευμένοι ως αρχεία εικόνων.

* imagefile1	imagefile2
image\post\p9.jpg	image\post\t9.gif
image\post\p8.jpg	image\post\t8.gif
image\post\p7.jpg	image\post\t7.gif
image\post\p6.jpg	image\post\t6.gif
image\post\p5.jpg	image\post\t5.gif
image\post\p4.jpg	image\post\t4.gif
image\post\p3.jpg	image\post\t3.gif
image\post\p2.jpg	image\post\t2.gif
image\post\p16.jpg	image\post\t16.gif
image\post\p15.jpg	image\post\t15.gif
image\post\p14.jpg	image\post\t14.gif
image\post\p13.jpg	image\post\t13.gif
image\post\p12.jpg	image\post\t12.gif
image\post\p11.jpg	image\post\t11.gif
image\post\p10.jpg	image\post\t10.gif
image\post\p1.jpg	image\post\t1.gif
image\post\p0.jpg	image\post\t0.gif

Εικόνα 5.3: Η βάση δεδομένων της αντιστοίχισης

Το επόμενο ερώτημα που χρειάζεται να σκεφτούμε είναι σε ποια ψηφίδα εμφανίζονται οι εικόνες. Αν εμφανίσετε τις μπάρες των ψηφίδων του μικρόκοσμου (βλ.Ενότητα 3.3) θα δείτε ότι κάθε εικόνα εμφανίζεται πάνω σε μία ψηφίδα σκηνή. Συγκεκριμένα με τη μετακίνηση του πρώτου ολισθητή (slider 1 βλ. Γεγονότα ψηφίδας – Αλλαγή κατάστασης) ενεργοποιείται η διαδικασία της Logo με τίτλο showimage1. Η διαδικασία αυτή επιλέγει τυχαία μία εικόνα από τις εικόνες του πρώτου πεδίου (imagefiles1) και στη συνέχεια την τοποθετεί ως εικόνα (με την έννοια της εμφάνισης) ενός κουτιού της σκηνής<sup>13</sup>. Το ίδιο γίνεται και για τη δεύτερη σκηνή όπου επιλέγεται τυχαία μία εικόνα από το δεύτερο πεδίο. Ο μαθητής αναμένεται, μετακινώντας τους ολισθητές να βρίσκει ζευγάρια εικόνων που θεωρεί ότι ταιριάζουν. Μόλις βρει ένα τέτοιο ζευγάρι πιέζει το κουμπί «Αντιστοίχιση» το οποίο καταγράφει το ζευγάρι των τιμών των ολισθητών π.χ. 11 = 5 (βλ. γεγονότα ψηφίδας Κουμπί\_4 ) το οποίο ζευγάρι τιμών όπως αναφέρθηκε νωρίτερα αντιστοιχεί σε συγκεκριμένες εικόνες. Να σημειωθεί ότι το ανακάτεμα των εικόνων γίνεται με τέτοιο τρόπο ώστε σε κάθε τιμή των ολισθητών (π.χ. στην τιμή 2) να

<sup>13</sup> Η ψηφίδα σκηνή διαθέτει μία σειρά από αντικείμενα –κουτιά, μπάλες, κεκλιμένα επίπεδα, σκοινιά κλπ- τα οποία μπορεί να εισαγάγει ο χρήστης μέσα στη σκηνή και να τους δώσει τη μορφή που θέλει χρησιμοποιώντας για παράδειγμα εικόνες όπως στην περίπτωση αυτή,

αντιστοιχεί διαφορετική εικόνα από υπολογιστή σε υπολογιστή αλλά και από δοκιμή σε δοκιμή. Έτσι οι μαθητές δεν μπορούν να αντιγράψουν τις τιμές από τους συμμαθητές τους ούτε έχει νόημα να θυμούνται τους αριθμούς της αντιστοίχησης γιατί δεν θα είναι ίδιοι την επόμενη φορά που θα δοκιμάσουν να κάνουν τις αντιστοιχήσεις.

Στη συνέχεια γίνεται έλεγχος της ορθότητας της αντιστοίχησης: ο οποίος όπως είπαμε βασίζεται στο ότι κάθε εικόνα βρίσκεται στην ίδια εγγραφή με αυτή που θεωρείται σωστή αντίστοιχή της. Έτσι λοιπόν η διαδικασία check που ενεργοποιείται με τη μετακίνηση του κέρσορα του ολισθητή ελέγχει αν τα αρχεία των εικόνων που έχουν αντιστοιχηθεί από το χρήστη ανήκουν στην ίδια εγγραφή, αν ναι εμφανίζεται δίπλα στην αντιστοίχηση το ok. Αν όχι καταγράφεται απλώς η αντιστοίχηση.

### 5.2.2 Ανάλυση κώδικα

Ο κώδικας logo του μικρόκοσμου περιλαμβάνει τις παρακάτω διαδικασίες:

- Ανακάτεμα διαθέσιμων εικόνων (διαδικασία mix)
- Δημιουργία λίστας τυχαίων αριθμών (διαδικασία shuffle)
- Εμφάνιση επιλεγμένης εικόνας (διαδικασίες showImage1, showImage2)
- Έλεγχος αντιστοίχησης εικόνων (διαδικασία checkimages)
- Διαγραφή καταγεγραμμένων αντιστοιχίσεων (διαδικασία undo)

#### Ανακάτεμα διαθέσιμων εικόνων

```
to mix
make "numofrecords ask "base [db.recordcount]
make "eikonesa shuffle :numofrecords
make "eikonesb shuffle :numofrecords
ask "slider1 [slider.setvalue 1]
showimage1
ask "slider2 [slider.setvalue 1]
showimage2
ask "slider1 [slider.setmax :numofrecords]
ask "slider2 [slider.setmax :numofrecords]
undo
end
```

#### Διαδικασία 18

Αρχικά ανιχνεύεται το πλήθος των εγγραφών (άρα των εικόνων) στη βάση δεδομένων (base) `make "numofrecords ask "base [db.recordcount]`

Στη συνέχεια δημιουργούνται λίστες με ανακατεμένους τους διαθέσιμους αριθμούς εγγραφών π.χ. αν οι εικόνες είναι 3 μπορεί να δημιουργηθεί η λίστα [2 3 1]



```
make "eikonesa shuffle :numofrecords
```

```
make "eikonesb shuffle :numofrecords
```

Οι ολισθητές μεταβαίνουν στην τιμή «1» και εμφανίζουν την πρώτη εικόνα. Αυτό είναι απαραίτητο αν ζητείται η έναρξη νέου παιχνιδιού:

```
ask "slider1 [slider.setvalue 1]
```

```
showimage1
```

```
ask "slider2 [slider.setvalue 1]
```

```
showimage2
```

Τέλος τίθεται η μέγιστη τιμή στους ολισθητές, η οποία αντιστοιχεί με το πλήθος των εικόνων και διαγράφονται (undo) τυχόν προηγούμενες αντιστοιχήσεις.

```
ask "slider1 [slider.setmax :numofrecords]
```

```
ask "slider2 [slider.setmax :numofrecords]
```

```
undo
```

### Δημιουργία λίστας τυχαίων αριθμών

Για να είναι δυνατή η επιλογή τυχαίας εικόνας από ένα συγκεκριμένο πλήθος εικόνων, κατά την έναρξη του παιχνιδιού δημιουργούνται με τη βοήθεια της διαδικασίας shuffle λίστες τυχαίων αριθμών που περιλαμβάνουν τους αριθμούς από 1 έως το πλήθος των εικόνων.

Η shuffle δέχεται ως παράμετρο τον αριθμό των αντικειμένων που θα «ανακατέψει» (μεταβλητή items) και επιστρέφει (στο όνομά της) την ανακατεμένη λίστα.

```
to shuffle :items
```

```
  randomize
```

```
  make "list []
```

```
  make "nr 0
```

```
  while [:nr < :items] [
```

```
    make "b 1 + random :items
```

```
    make "c memberp :b :list
```

```
    if not :c [
```

```
      make "list fput :b :list
```

```
      make "nr :nr+1
```

```
    ]
```

```
  ]
```

```
  output :list
```

```
end
```

### Διαδικασία 19



Αρχικά με την εντολή `randomize` αρχικοποιείται η «εσωτερική γεννήτρια» που αργότερα θα δώσει τυχαίους αριθμούς. Στη συνέχεια δημιουργείται μια κενή λίστα (`list`) και μηδενίζεται ο βοηθητικός μετρητής (`nr`).

```
randomize
make "list []
make "nr 0
```

Με τον παρακάτω βρόχο (`while [ ]`) δημιουργείται η λίστα τυχαίων αριθμών. Αρχικά ζητείται από το σύστημα ένας τυχαίος αριθμός (`random`) μέχρι τον μέγιστο αριθμό αντικειμένων που έχει δοθεί στην `shuffle`. Στον αριθμό αυτόν προστίθεται η μονάδα (1) για να αποφευχθεί η τιμή (0).

Στη συνέχεια ελέγχεται (`memberp`) αν ο αριθμός υπάρχει ήδη στην υπάρχουσα λίστα και αν δεν υπάρχει προστίθεται σε αυτήν (`fput`). Τέλος ο βοηθητικός μετρητής (`nr`) μετράει πόσες φορές θα επαναληφθούν τα παραπάνω βήματα, ώστε να τερματιστεί ο βρόχος όταν ο αριθμός των επαναλήψεων φθάσει τον αριθμό των αντικειμένων (`nr < items`)

```
while [:nr < :items] [
    make "b 1 + random :items
    make "c memberp :b :list
    if not :c [
        make "list fput :b :list
        make "nr :nr+1
    ]
]
```

Η τελευταία εντολή της `shuffle` προκαλεί την εξαγωγή της λίστας από την διαδικασία.

```
output :list
```

### Εμφάνιση επιλεγμένης εικόνας

Με τις διαδικασίες `showImage1`, `showImage2` η εικόνα που επιλέχθηκε με έναν από τους δύο ολισθητές εμφανίζεται στο αντίστοιχο πλαίσιο. Οι δύο διαδικασίες διαφέρουν μόνο ως προς τα ονόματα των μεταβλητών και αντικειμένων.

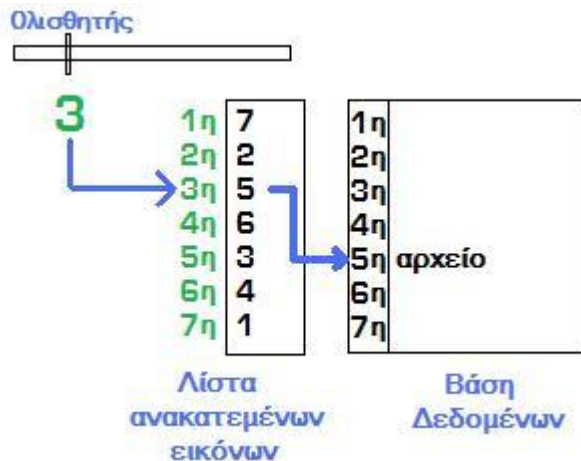
```
to showImage1
    make "svalue ask "slider1 [slider.value]
    make "svalue item :svalue :eikonesa
    make "image word microworldfolder (db.cell "imagefiles :svalue "imagefile1)
    ask "box1 [setImage :image]
end
```

### Διαδικασία 20

## Εργαστήριο εκπαιδευτικής Τεχνολογίας

Αρχικά λαμβάνεται ο αριθμός της εικόνας όπως αυτός επιλέχθηκε στον ολισθητή (slider.value) . Στη συνέχεια αναζητείται ο πραγματικός αριθμός της αντίστοιχης εγγραφής στη βάση δεδομένων (item).

Για παράδειγμα, μπορεί να έχει επιλεγθεί στον ολισθητή η εικόνα 3 και στην 3η θέση της λίστας ανακατεμένων εικόνων (eikonesa) βρίσκεται η εγγραφή με αριθμό 5 (βλ. παρακάτω σχήμα).



```
make "svalue ask "slider1 [slider.value]
```

```
make "svalue item :svalue :eikonesa
```

Τελικά, με βάση τον πραγματικό αριθμό εγγραφής (svalue), ανακτάται το όνομα του αρχείου της εικόνας από αντίστοιχο κελί (imagefile1) του πίνακα (imagefiles) της βάσης δεδομένων και εμφανίζεται στο πλαίσιο (αντικείμενο της σκηνης) box1.

```
make "image word microworldfolder (db.cell "imagefiles :svalue "imagefile1)
```

```
ask "box1 [setimage :image]
```

### Έλεγχος αντιστοίχισης εικόνων

Κατά των έλεγχο αντιστοίχισης, οι επιλεγμένες εικόνες ελέγχονται αν «ταιριάζουν», σύμφωνα με τις καταχωρημένες πληροφορίες στη βάση δεδομένων.

```
to checkimages
```

```
make "sva ask "slider1 [slider.value]
```

```
make "svb ask "slider2 [slider.value]
```

```
make "w (word "| |:sva "| = |:svb "| |)
```

```
ask "text [area.append :w]
```

```
make "sva item ask "slider1 [slider.value] :eikonesa
```

```
make "svb item ask "slider2 [slider.value] :eikoneseb
```

```
if :sva = :svb [ask "text [area.append "ok]]
```

```
ask "text [area.append "\n ]
```

*end*

#### **Διαδικασία 21**

Αρχικά λαμβάνονται οι θέσεις των δύο ολισθητών (*slider.value*) και με βάση αυτές δημιουργείται το κατάλληλο κείμενο (μεταβλητή *w*) ώστε η ενέργεια του χρήστη να καταχωρηθεί στη συνέχεια στον πίνακα (*text*) των προσπαθειών του.

```
make "sva ask "slider1 [slider.value]
make "svb ask "slider2 [slider.value]
make "w (word "| |:sva|" = |:svb|" |)
ask "text [area.append:w]
```

Στη συνέχεια αναζητείται από τις λίστες «ανακατεμένων εικόνων» (*eikonesa*, *eikonesb*), η πραγματική θέση των εικόνων στην βάση δεδομένων. Για να είναι το «ταίριασμα» των εικόνων επιτυχές οι εικόνες πρέπει να βρίσκονται στην ίδια εγγραφή της βάσης δεδομένων. Αν το «ταίριασμα» είναι επιτυχημένο (*sva=svb*) τότε δίπλα στην προσπάθεια του χρήστη εμφανίζεται «ok».

Τέλος γίνεται αλλαγή γραμμής, ώστε ο πίνακας (*text*) να δεχθεί την επόμενη προσπάθεια σε νέα γραμμή.

```
make "sva item ask "slider1 [slider.value] :eikonesa
make "svb item ask "slider2 [slider.value] :eikonesb
if :sva = :svb [ask "text [area.append "ok]]
ask "text [area.append "\n ]
```

#### **Διαγραφή καταγεγραμμένων αντιστοιχίσεων**

Για το ξεκίνημα νέου παιχνιδιού ή νέας προσπάθειας απαιτείται η διαγραφή των προηγούμενων προσπαθειών του χρήστη. Πρώτα επιλέγονται όλα τα στοιχεία από τον πίνακα προσπαθειών (*text*) και έπειτα διαγράφονται.

```
to undo
    AREA.SELECTALL
    AREA.CUT
end
```

### **5.2.3 Κριτική και επεκτάσεις της αρχικής ιδέας**

#### **5.2.3.1 Κριτική – ή κατευθύνσεις για αλλαγή**

Η βασική κριτική που ασκήθηκε στη διαδικασία της σύγκρισης των ιστορικών γεγονότων είχε να κάνει με το είδος της ανατροφοδότησης που προσφέρει ο μικρόκοσμος και συγκεκριμένα με την προβολή του *ok* ως επιβεβαίωση της σωστής επιλογής. Το ζήτημα εδώ είναι κυρίως ότι λείπει από το μικρόκοσμο η πληροφορία που να βοηθάει το μαθητή να διατυπώσει το κριτήριο του συσχετισμού των εικόνων και να κάνει τη διατύπωση αυτή εργαλείο για τη συσχέτιση. Συγκεκριμένα, τώρα οι εικόνες συσχετίζονται στο μικρόκοσμο με μία εσωτερική λογική που είναι γνωστή στο δημιουργό του μικρόκοσμου

και ίσως και στους γνώστες της ιστορικής πληροφορίας. Αυτή η λογική είναι που καθορίζει το ok δίπλα στις σωστές επιλογές αλλά παραμένει κλειδωμένη για αυτούς που χρησιμοποιούν το μικρόκοσμο επιχειρώντας να μάθουν. Έτσι με τη συγκεκριμένη ανατροφοδότηση ο μαθητής μπορεί να καταλάβει ότι μία συσχέτιση είναι σωστή ή λάθος αλλά δεν έχει την ευκαιρία να ανακαλύψει το γιατί το οποίο μοιάζει να είναι και το πιο σημαντικό για αυτή τη διαδικασία. Αυτό είναι το στοιχείο το οποίο «θυμώνει» τους πιο πολλούς εκπαιδευτικούς ή φοιτητές γιατί ουσιαστικά καταργεί τη λογική του μικρόκοσμου. Ωστόσο στην πλειοψηφία των μικρόκοσμων που βασίστηκαν στη σύγκριση το στοιχείο αυτό διατηρήθηκε ή απλώς παρακάμφθηκε (βλ. Ενότητα 5.2.3.3). Το ερώτημα λοιπόν είναι πώς θα μπορούσε αυτή η λογική των συσχετίσεων να ξεκλειδωθεί και να γίνει εργαλείο σκέψης για τους μαθητές: Έτσι λοιπόν για κάθε εικόνα που εμφανίζεται στο μικρόκοσμο ο μαθητής θα καλείται να αποφασίσει για μία σειρά από ζητήματα που σχετίζονται με την εικόνα και τα έχουμε προκαθορίσει (πράγμα που σημαίνει ότι μπορεί να εμφανίζονται από μία ψηφίδα λίστα) π.χ. ποιο είναι το θέμα της φωτογραφίας, ποια χρονιά συμβαίνει ή ποια εποχή του χρόνου, ποιος ή τι πρωταγωνιστεί στη φωτογραφία κ.ο.κ. Μέσα από αυτού του είδους την επεξεργασία βοηθάμε τους μαθητές να αρχίσουν να αναλύουν τις φωτογραφίες με συγκεκριμένα κριτήρια και στη συνέχεια να χρησιμοποιήσουν αυτή την ανάλυση ως εργαλείο για να αποφασίσουν ποια εικόνα ταιριάζει με ποια.

### 5.2.3.2 Αντιστοίχιση των γραμμάτων του Αλφαβήτου με εικόνες

Η ιδέα του να εμπλακούν οι μαθητές σε διαδικασίες σύγκρισης κειμένων εικόνων χαρτών κλπ φάνηκε ενδιαφέρουσα ως διαδικασία και ως εφαρμογή. Έτσι λοιπόν βασισμένη σε αυτή ακριβώς την ιδέα και με αντικατάσταση απλώς των εικόνων μία φοιτήτρια του μεταπτυχιακού προγράμματος σπουδών Θεωρία Πράξη και Αξιολόγηση του Εκπαιδευτικού Έργου μετέφερε στο μικρόκοσμο μία πρακτική που γίνεται στην πρώτη δημοτικού με κάρτες. Έτσι στην πρώτη στήλη έβαλε τα γράμματα της Αλφαβήτου ως αρχεία Εικόνας και στη Δεύτερη στήλη έβαλε λέξεις που αρχίζουν από αυτό το γράμμα και οι μαθητές θα έπρεπε να βρουν τις αντιστοιχίες. Η εφαρμογή είχε γίνει σε προηγούμενη έκδοση του Αβακίου και έτσι δεν είναι διαθέσιμη τώρα. Ακόμη υπήρχε η ιδέα να προστεθεί και ήχος (να ακούγεται η φωνή του γράμματος που επιλέγει ο μαθητής) αλλά δεν είχε υλοποιηθεί.

### 5.2.3.3 Αντιστοίχιση φωτογραφιών για διαφημίσεις με λεζάντες

Μία άλλη πρόταση βασισμένη στην ίδια ιδέα ήταν να επιχειρήσουν οι μαθητές να συζητήσουν για τη διαφήμιση επιχειρώντας να αντιστοιχήσουν εικόνες με λεζάντες. Στην περίπτωση αυτή όμως κρίθηκε ότι έπρεπε να γίνουν κάποιες αλλαγές στο μικρόκοσμο και ειδικότερα να αφαιρεθεί ο έλεγχος του σωστού ή λάθους γιατί στην προκειμένη εφαρμογή περισσότερες από μία επιλογές ήταν εφικτή και το ενδιαφέρον ήταν να προβάλλουν οι μαθητές τις επιλογές τους και να τις δικαιολογήσουν. Για το σκοπό αυτό προτάθηκαν οι εξής αλλαγές στις λειτουργικότητες: α)αφαίρεση του ελέγχου ορθότητας της αντιστοίχισης και β)προσθήκη ενός νέου ολισθητή ο οποίος στη συνέχεια θα πρόβαλε το συνδυασμό εικόνων – λεζαντών (δεν θα κατέγραφε μόνο τους αριθμούς των

αντιστοιχήσεων) έτσι ώστε να είναι δυνατόν σε κάθε ομάδα να προβάλλει τους συνδυασμούς που έκανε και να τους συζητήσει – αιτιολογήσει στους υπόλοιπους

### 5.2.3.4 Παροιμίες

**Μικρόκοσμος:** paroimies-manual.mwd

Στην ίδια λογική με τη σύγκριση των ιστορικών γεγονότων βασίζονται και οι παροιμίες μόνο που εδώ αντί να συσχετίζονται δύο εικόνες συσχετίζονται τα μέρη παροιμιών (κάθε παροιμία είναι χωρισμένη στη μέση). Για να γίνει περισσότερο κατανοητή η ομοιότητα δείτε τη βάση δεδομένων στην οποία είναι καταχωρημένες οι παροιμίες και συγκρίνετέ τη με τη βάση δεδομένων των ιστορικών γεγονότων (βλ. Εικόνα 5.3)

*	paroi	mia
	Η γλώσσα κόκκαλα δεν έχει	και κόκκαλα τσακίζει.
	Κατά τα γόνατα που 'χω	για μετάνοιες είμαι;
	Τα δαχτυλίδια αν έπεσαν	τα δάχτυλα απομένουν.
	Θεωρία επισκόπου	και καρδιά μυλωνά.
	Κάλλιο κόμπο στο πουγγί	παρά κόμπο στην καρδιά.
	Με το κεφάλι που 'χω	μ' αυτό σε χαιρετώ.
	Όσα τραβάει το κορμί	τα φταίει το κεφάλι.
	Σηκώθηκαν τα πόδια	και χτυπούν το κεφάλι.
	Αλί που το 'χει η κούτρα του	να κατεβάζει ψείρες.
	Το γινάπι	βγάζει μάπι.
	Η μύτη του να πέσει	δε θα σκύψει να τη σηκώσει ...
	Όποιος δεν έχει μυαλό	έχει πόδια.
	Κάλλιο κόκκινο πρόσωπο	παρά μαύρη καρδιά.
	Η αρρώστια μπαίνει με το τσουβάλι	και φεύγει με τη δαχτυλήθρα.
	Από καλοσύνη	δεν πάει κανείς στο αναγκαίο.
	Ντράπου τον ένα, ντράπου τον άλλο,	κανένα παιδί με τον άντρα μου.
	Πάρε αυτό που δίνει η μαμή	και πες πως είν' αγόρι.
	Γεια σου	κι αλήθεια λέω.
	Όπου κοιμάται με στραβό,	το πρωί αλλοιθωρίζει.

Εικόνα 5.4: Η βάση δεδομένων με τα μέρη των παροιμιών

Όπως βλέπετε υπάρχουν και εδώ δύο πεδία: στο πρώτο πεδίο γράφεται το πρώτο μέρος της παροιμίας ενώ στο δεύτερο πεδίο γράφεται το (σωστό) συμπλήρωμά του. Τα κελιά του κάθε πεδίου ανακατεύονται και εδώ με τον ίδιο τρόπο που ανακατεύονται και στις αντιστοιχήσεις. Ίδιος είναι και ο έλεγχος του σωστού συνδυασμού των μερών των παροιμιών. Η μόνη διαφοροποίηση αυτού του μικρόκοσμου από την αντιστοίχιση είναι η εξής: Από το πρώτο πεδίο επιλέγεται τυχαία μία εγγραφή. Στη συνέχεια αυτή η εγγραφή

συνδυάζεται με δύο τρόπους α) με την αντιστοιχί της και β) τυχαία με μία άλλη έτσι ώστε να παράγει ακόμη και αστείες προτάσεις πράγμα το οποίο επαναλαμβάνεται τέσσερις φορές (βλ. Εικόνα 5.5 όπου εκτός από τη σωστή υπάρχουν και τέσσερις τυχαίοι συνδυασμοί).

Βήχα και συνάχι τον νηστικό δεν τον πιστεύει.  
 Βήχα και συνάχι πάνε οι πληγωμένοι.  
 Βήχα και συνάχι χαρά στον οπου τάχει.  
 Βήχα και συνάχι του χάρου μήνυμα.  
 Βήχα και συνάχι δεν χορεύει.

Εικόνα 5.5 Αντιστοιχίσεις παροιμιών

Από τις πέντε επιλογές ο μαθητής πρέπει να διαλέξει τη σωστή και το σύστημα απαντάει αν η επιλογή είναι σωστή ή όχι καταγράφοντας την σε ένα ξεχωριστό πεδίο.

### 5.2.3.5 Αριάδνη: Αντιστοιχίση Αρχαίων Ελληνικών με τη μετάφρασή τους

Μικρόκοσμος: Ariadne.mwd

Στην ίδια λογική με τις παροιμίες είναι βασισμένη και η Αριάδνη με κάποια παραλλαγή. Στην Αριάδνη ένα χωρίο γραμμένο στα αρχαία ελληνικά συνδυάζεται με τρεις εκδοχές της μετάφρασής τους (έτσι δεν επιλέγει τυχαία από όλες τις μεταφράσεις αλλά προβάλλει απλώς τις τρεις εναλλακτικές). Αποκαλυπτική και σε αυτή την περίπτωση είναι η βάση δεδομένων.

*	ΠΡΩΤΟΤΥΠΟ	ΜΕΤΑΦΡΑΣΗ	α
	Βωμός , το μέγεθος και τό της εργασίας ταύτη τη κατασκευη	Βωμός ως προς τὸ μέγεθος καὶ τὴν ἐργασία κατασκευῆς του	false
	Βωμός , το μέγεθος και τό της εργασίας ταύτη τη κατασκευη	Ανάλογος στο μέγεθος και με την εμφάνιση του περιβάλλοντος	false
	Βωμός , το μέγεθος και τό της εργασίας ταύτη τη κατασκευη	Βωμός για το μέγεθος καὶ τὴν πολυπλοκότητα της κατασκευῆς του	true
	Ἡδονη δὲ καὶ ἀρετὴ τῶν ὑδάτων πρὸς ἑκατέρου τὴν χρῆσιν θαυμαστοῦ πεφ...	Ὡς πρὸς τὴν εὐχάριστη ὄψη καὶ τὴν ομορφιά τοῦ νεροῦ θαυμαζό...	false
	Ἡδονη δὲ καὶ ἀρετὴ τῶν ὑδάτων πρὸς ἑκατέρου τὴν χρῆσιν θαυμαστοῦ πεφ...	Με τὰ ὁμορφα καὶ θεραπευτικὰ νερά τους ἐξυπηρετοῦσαν τους ...	true
	Ἡδονη δὲ καὶ ἀρετὴ τῶν ὑδάτων πρὸς ἑκατέρου τὴν χρῆσιν θαυμαστοῦ πεφ...	Για τὴν χαρὰ καὶ μετὰξὺ τους ομορφιά θαυμασὰ τα χρησιμοποιοῦ...	false
	Τὰς δὲ χειμερινὰς τοὶς θερμοὶς λουτροὶς υποστέγουσ περιθῆντες	Καὶ τὶς χειμερινῆς με ὑπόστεγα τὶς χρησιμοποιοῦσαν ὡς καλοκαίρι...	true
	Τὰς δὲ χειμερινὰς τοὶς θερμοὶς λουτροὶς υποστέγουσ περιθῆντες	Ὀρισμένες ἦταν ὑπαίθριες καὶ ἄλλες κλειστῆς για νὰ χρησιμοποιοῦ...	false
	Τὰς δὲ χειμερινὰς τοὶς θερμοὶς λουτροὶς υποστέγουσ περιθῆντες	Καὶ τὶς χειμερινῆς με θερμὰ λουτρά καὶ στέγαστρα τὶς περιέβαλλαν	true
	Καὶ ἐπὶ τοῦς ἔξω κύκλους δι' ὁσῶτων κατὰ τὰς γεφύρας ἐπωχέτευον	Καὶ κυκλικὰ ἐξωτερικὰ κατευθύνονταν πρὸς τὶς γέφυρες	false
	Καὶ ἐπὶ τοῦς ἔξω κύκλους δι' ὁσῶτων κατὰ τὰς γεφύρας ἐπωχέτευον	Καὶ ἀπὸ τοῦς ἔξω κύκλους με νερὸ ὀδηγοῦνταν σὶς γέφυρες	false
	Καὶ ἐπὶ τοῦς ἔξω κύκλους δι' ὁσῶτων κατὰ τὰς γεφύρας ἐπωχέτευον	Με σωλῆκια ἔφερναν τὸ νερὸ ἀπὸ τὶς δεξαμενῆς μέχρι τὶς γέφυρ...	true

Εικόνα 5.6: Η βάση δεδομένων για την αντιστοιχίση των αρχαίων με τη μετάφρασή τους

Εδώ όπως βλέπουμε υπάρχουν τρία πεδία. Στο πρώτο υπάρχει το χωρίο στα Αρχαία Ελληνικά το οποίο επαναλαμβάνεται το ίδιο και στις τρεις εγγραφές. Στο χωρίο αυτό αντιστοιχούν τρεις εκδοχές της μετάφρασης εκ των οποίων μόνο η μία είναι σωστή και

αυτή δηλώνεται στο τρίτο πεδίο με τον τίτλο α και παίρνει τιμές αληθές –ψευδές (true-false). Έτσι λοιπόν αρχικά επιλέγει τυχαία το χωρίο στα αρχαία ελληνικά, στη συνέχεια προβάλλει τις τρεις εκδοχές της μετάφρασης και επιβεβαιώνει την ορθότητα της επιλογής βασισμένο στο αν ο συνδυασμός που έχει επιλεγεί έχει δίπλα του στη βάση δεδομένων (στην ίδια δηλ. εγγραφή) true ή false.

## **6 Συνήθη προβλήματα και απαντήσεις**

### **6.1 Γιατί δεν είναι ορατό το μενού του μικρόκοσμου; (Μικρόκοσμος, Ψηφίδα, Εργαλεία)**

Διότι έχουμε ανοίξει το αρχείο του μικρόκοσμου κάνοντας διπλό κλικ πάνω του και δεν έχουμε ακολουθήσει τη διαδικασία που περιγράφεται στην ενότητα 3.2

### **6.2 Γιατί όταν κάνω κλικ στο μικρόκοσμο ταξινομούμε που έχω αποθηκεύσει δεν ανοίγει;**

Συχνά κατά την κατασκευή μικρόκοσμων που βασίζονται στο ταξινομούμε οι χρήστες θεωρούν ότι έχουν αποθηκεύσει το μικρόκοσμό τους αλλά το αρχείο που αποθήκευσαν δεν ανοίγει. Αυτό συμβαίνει διότι στην πλειοψηφία των περιπτώσεων αποθηκεύουν μόνο τη βάση δεδομένων και όχι όλο το μικρόκοσμο. Σε αυτή την περίπτωση η κατάληξη του αρχείου δεν είναι «mwd» αλλά «cdb». Χρειάζεται μεγάλη προσοχή ως προς το τι αποθηκεύουμε. Στην περίπτωση που πιάσαμε την αποθήκευση που βρίσκεται στη βάση δεδομένων τότε αποθηκεύεται μόνο το περιεχόμενο της βάσης δεδομένων και όχι όλος ο μικρόκοσμος (δηλ. αλλαγές που έχουμε κάνει π.χ. στην ψηφίδα Logo ή στα κουμπιά ή σε άλλες ψηφίδες δεν θα αποθηκευτούν). Αποθηκεύουμε ξεχωριστά τη βάση δεδομένων μόνο όταν θέλουμε να την εισαγάγουμε σε έναν άλλο μικρόκοσμο. Σε περίπτωση που έχουμε αποθηκεύσει όχι το μικρόκοσμο αλλά μόνο τη βάση δεδομένων (δηλ. έχουμε δημιουργήσει ένα αρχείο «cdb») για να επαναφέρουμε το περιεχόμενο της βάσης δεδομένων στο μικρόκοσμό μας κάνουμε τα εξής βήματα

- Ανοίγουμε το μικρόκοσμό μας
- Πηγαίνουμε στον επεξεργαστή βάσεων -> Στο μενού «Βάση» και επιλέγουμε «Άνοιγμα». Μέσα από το παράθυρο που ανοίγει επιλέγουμε το cdb αρχείο που δημιουργήσαμε και έτσι η βάση δεδομένων μπαίνει στο μικρόκοσμό μας.

Προσοχή! Εδώ δεν έχετε αποθηκεύσει την ψηφίδα (επεξεργαστής βάσεων) αλλά το περιεχόμενό της. Αυτό σημαίνει ότι όποιες αλλαγές έχετε κάνει στη μορφή της ψηφίδας δεν θα έχουν αποθηκευτεί.

### **6.3 Πώς οργανώνω τις ψηφίδες στο μικρόκοσμό μου ιδιαίτερα όταν είναι πολλές**

Συχνά σε έναν μικρόκοσμο μπορεί να χρειαστεί να χρησιμοποιήσουμε πολλές ψηφίδες οι οποίες όταν είναι όλες παρούσες στο μικρόκοσμο δημιουργούν ένα χάος στην επιφάνεια εργασίας. Υπάρχουν όμως τρόποι να τις οργανώσουμε. Πριν προχωρήσουμε να περιγράψουμε αυτούς τους τρόπους χρήσιμο είναι να πούμε ότι ορισμένες ψηφίδες



μπορεί να βρίσκονται συνεχώς σε απόκρυψη αν έχουμε αποφασίσει ότι δεν θέλουμε να τις χρησιμοποιήσουν οι μαθητές μας. Έτσι για παράδειγμα μία βάση δεδομένων (σε ένα μικρόκοσμο μπορεί να υπάρχουν περισσότερες από μία βάσεις δεδομένων) η οποία καταγράφει τις διάφορες τιμές που μπορεί να δίνει ένας ολισθητής ή ένας μεταβολέας και εξυπηρετεί αποκλειστικά τις ανάγκες του κώδικα που έχουμε γράψει δεν χρειάζεται να είναι ορατός στο μαθητή αν δεν τη χρησιμοποιήσει ποτέ. Αντίστοιχα το ίδιο μπορεί να συμβαίνει και με μία ψηφίδα κείμενο κ.ο.κ

### 6.3.1 Χρήση κουμπιών: Κουμπί, Διακόπτης, Κουμπί Ελέγχου και Κουμπί Επιλογής

Όταν στο μικρόκοσμο υπάρχουν ψηφίδες που δεν θέλω να φαίνονται συνέχεια αλλά να δίνω στο χρήστη τη δυνατότητα να τις καλεί όποτε υπάρχει ανάγκη τότε μπορώ να χρησιμοποιήσω τις ψηφίδες κουμπιά. Ένα τέτοιο παράδειγμα που έχουμε δει συχνά μέχρι τώρα έχει να κάνει με οδηγίες για το πώς να χρησιμοποιήσει κανείς το μικρόκοσμο ή πληροφορίες σχετικά με τους συντελεστές κ.ο.κ. Στην περίπτωση αυτή μπορεί κανείς να χρησιμοποιήσει μία ψηφίδα κουμπί ή μία ψηφίδα διακόπτη. Η διαφορά μεταξύ των δύο είναι η ακόλουθη: η ψηφίδα διακόπτη έχει δύο καταστάσεις α) στη μία κατάσταση ο διακόπτης είναι επιλεγμένος (π.β. την εντολή ΔΙΑΚΟΠΤΗΣ.ΕΠΙΛΕΓΜΕΝΟΣ) και β) στην άλλη κατάσταση ο διακόπτης είναι από-επιλεγμένος. Έτσι μπορούμε να διαφοροποιήσουμε τι θα συμβαίνει σε κάθε μία από αυτές τις δύο καταστάσεις: να εμφανίζεται μία ψηφίδα όταν πατάμε το κουμπί και επιλέγεται ο διακόπτης ή να εξαφανίζεται αυτή η ψηφίδα όταν πατάμε το κουμπί και από-επιλέγεται ο διακόπτης. Στην περίπτωση της ψηφίδας διακόπτη μπορούμε δηλ. με το πάτημα του κουμπιού να εμφανίζουμε τις οδηγίες και με το πάτημα του ίδιου κουμπιού να τις εξαφανίζουμε (βλ. Διαδικασία 16). Την ίδια λειτουργία μπορούν να επιτελέσουν και τα κουμπιά «Κουμπί Ελέγχου» και «Κουμπί Επιλογής» (π.β. τις εντολές `rbutton.selected` και `checkbox.selected` αντίστοιχα)

Με την ψηφίδα κουμπί όμως μπορούμε να κάνουμε μόνο τη μία από τις δύο ενέργειες ή πατώντας το κουμπί να εμφανίζεται μία ή περισσότερες ψηφίδες ή να εξαφανίζεται/ονται. Σε περίπτωση λοιπόν που θέλουμε να εμφανιστούν περισσότερες από μία ψηφίδες τότε ή να εμφανιστούν κάποιες και να εξαφανιστούν κάποιες άλλες τότε φτιάχνουμε μία διαδικασία που να περιλαμβάνει όλες αυτές τις ψηφίδες γράφοντας π.χ. μία διαδικασία όπως η ακόλουθη που κρύβει την ψηφίδα Κορνίζα και εμφανίζει τις ψηφίδες επεξεργασίας δεδομένων

*Για επεξεργασία\_δεδομένων*

*Ask "Βάση [restore]*

*Ask "Κορνίζα [hide]*

*Ask "Ερώτηση [restore]*

*Ask "Σύνολο [restore]*

*End*

**Διαδικασία 22**



Στην περίπτωση που χρησιμοποιήσουμε την ψηφίδα Κουμπί για να εμφανίσουμε οδηγίες ή πληροφορίες ο χρήστης θα πρέπει στη συνέχεια να κάνει κλικ στο κουμπί ελαχιστοποίησης της ψηφίδας για να την απομακρύνει από την επιφάνεια εργασίας του Αβακίου. Το κουμπί επίσης είναι συχνά πολύ χρήσιμο για να εκτελέσει μία διαδικασία στη Logo και να ξεκινήσει π.χ. να λειτουργεί ο μικρόκοσμος, ή να αρχικοποιήσει τις τιμές ενός μικρόκοσμου ή να καθαρίσει το περιεχόμενο του κειμένου (π.β. το μικρόκοσμο `paroiimies_mannual.mwd`)

### 6.3.2 Χρήση όψεων

Οι όψεις αποτυπώνουν μία διάταξη της επιφάνειας εργασίας και μπορούν να την επαναφέρουν όποτε εσείς το επιλέξετε. Διαμορφώστε λοιπόν την επιφάνεια εργασίας όπως θέλετε. Πηγαίνετε στα εργαλεία και επιλέξτε «Διαχείριση όψεων». Από το παράθυρο που εμφανίζεται επιλέξτε «Νέα» και στη συνέχεια επιλέξτε αλλαγή ονόματος για να δώσετε στην όψη του μικρόκοσμου το όνομα που θέλετε π.χ. `Επεξεργασία_δεδομένων` ή `Παρουσίαση_υλικού` ή ότι άλλο σχετικό κάνουν οι ψηφίδες στη συγκεκριμένη διάταξη). Στη συνέχεια μπορείτε να χρησιμοποιήσετε κατευθείαν στα γεγονότα ενός κουμπιού την εντολή `showview` "ΌνομαΌψης π.χ. `showview` "Επεξεργασία\_δεδομένων και να σας εμφανίζει αυτόματα τη συγκεκριμένη διάταξη. Μπορείτε να προσθέσετε όσες όψεις θέλετε σε ένα μικρόκοσμο.

### 6.3.3 Δημιουργία σύνθετων ψηφίδων με χρήση της λειτουργικότητας φιλοξενίας

Σε κάποιες περιπτώσεις και όχι μόνο όταν έχετε πολλές ψηφίδες σε έναν μικρόκοσμο αλλά και όταν μία σύνθετη λειτουργικότητα (που απαιτεί συνδυασμό ψηφίδων) θέλετε να τη χρησιμοποιήσετε σε περισσότερους από έναν μικρόκοσμος τότε είναι ίσως χρήσιμο να φτιάξετε μία σύνθετη ψηφίδα που να φιλοξενεί μέσα της άλλες. Αναλυτικά για τη διαδικασία δημιουργίας σύνθετης ψηφίδας μπορείτε να συμβουλευτείτε το εγχειρίδιο του Αβακίου (σελ. 61-72)

## 6.4 Πώς εμφανίζω στο Αβάκιο κείμενα Αρχαίων Ελληνικών

Μικρόκοσμος: `Archaia_Ellinika.mwd`

Για να εμφανίσετε διάφορα κείμενα που είναι γραμμένα στα αρχαία Ελληνικά και ως εκ τούτου θέλετε να εμφανίζονται τα πνεύματα, οι τόνοι και υπογεγραμμένες, μέσα στο Αβάκιο μπορείτε να το κάνετε με τους εξής τρόπους:

Χρειάζεστε έναν μικρόκοσμο που να έχει τουλάχιστον τις εξής ψηφίδες: «επεξεργαστή βάσεων» και «Πλοηγό»

Στη συνέχεια έχετε δύο επιλογές: Είτε να προβάλλετε το κείμενο κατ'ευθείαν από το Internet (την ψηφιακή βιβλιοθήκη Περσέας βλ. παρακάτω). Είτε να προβάλλετε το κείμενο από τον υπολογιστή σας.

### 6.4.1 Προβάλλοντας το κείμενο από το Internet

**Βήμα1:** Πηγαίνετε στον επεξεργαστή Βάσεων που έχετε στο μικρόκοσμό σας και ορίστε τις ιδιότητες ενός από τα πεδία να δέχεται δεδομένα url.

**Βήμα2:** Συνδέστε μέσα από τη διαχείριση συνδέσμων το πεδίο αυτό με τον Πλοηγό (σύνδεσμος Url)

**Βήμα3:** βρείτε το κείμενο που θέλετε στον Περσέα στην Ελληνορωμαϊκή Συλλογή (<http://www.perseus.tufts.edu/hopper/collection?collection=Perseus:collection:Greco-Roman>) προσοχή: Τα κείμενα σε κάποιες περιπτώσεις είναι στα Αρχαία Ελληνικά και στα Αγγλικά ή μόνο στο ένα από τα δύο: Διαλέξτε βεβαίως το κείμενο που είναι γραμμένο στα Αρχαία Ελληνικά. Να προσδιορίσετε το συγκεκριμένο απόσπασμα και να αντιγράψετε την ηλ. διεύθυνση url στην πρώτη εγγραφή του πεδίου που ορίσατε να παίρνει δεδομένα url (στο μικρόκοσμο Archaia\_Ellinika.mwd το πεδίο αυτό έχει τίτλο «Archaia\_internet» και έχει συνδεθεί με τον Πλοηγό\_1. Όπως θα δείτε μπορείτε να προσδιορίσετε συγκεκριμένα αποσπάσματα ενός συγγραφέα όπως όμως είναι χωρισμένα στον Περσέα. Δεν μπορείτε να σταματήσετε το κείμενο εκεί που θέλετε εσείς. Επαναλάβετε τη διαδικασία (βρείτε δηλ. το απόσπασμα στον Περσέα και αντιγράψτε το url) για όσα αποσπάσματα θέλετε να συμπεριλάβετε στο μικρόκοσμό σας.

### 6.4.2 Προβάλλοντας το κείμενο από το δικό σας υπολογιστή:

Σε περίπτωση που θέλετε να συμπεριλάβετε αποσπάσματα που έχετε ορίσει εσείς (πάντοτε βέβαια μπορείτε να τα αντιγράψετε από τον Περσέα όπως περιγράφηκε νωρίτερα). Θα πρέπει να ακολουθήσετε τα εξής βήματα

**Βήμα1:** Δημιουργήστε ένα πεδίο ορίζοντας από τις ιδιότητες τα δεδομένα του να είναι αλφαριθμητικά (πεδίο Archaia\_myFiles στο μικρόκοσμο Archaia\_Ellinika.mwd)

**Βήμα 2:** Συνδέστε στην ψηφίδα Πλοηγός το πεδίο στο σύνδεσμο «url ως αλφαριθμητικό»

**Βήμα3:** Αποθηκεύστε το κείμενό σας ως αρχείο html (Για τα windows vista: Αποθήκευση ως -> Άλλες μορφές -> Ιστοσελίδα) και επαναλάβετε τη διαδικασία για όσα κείμενα θέλετε να προβάλλετε στο αβάκιο (για κάθε κείμενο διαφορετική ιστοσελίδα)

**Βήμα4:** Βάλτε τις html σελίδες σας σε ένα φάκελο (μπορείτε να τον ονομάσετε html\_files) Τοποθετήστε το φάκελο μέσα στον ίδιο φάκελο που βρίσκεται και ο μικρόκοσμος

**Βήμα5:** Γράψτε το όνομα του φακέλου και το όνομα του αρχείου στα κελιά του πεδίου που έχετε ορίσει να δέχεται αλφαριθμητικά δεδομένα (Archaia\_myFiles) όπως στο μικρόκοσμο Archaia\_Ellinika.mwd. Το όνομα του κάθε αρχείου θα πρέπει να μπει σε διαφορετικό κελί.

Προσοχή για να μπορέσει να λειτουργήσει η σύνδεση του πεδίου με τον Πλοηγό θα πρέπει να έχετε αποθηκεύσει το μικρόκοσμό σας. Στο μικρόκοσμο *Archaia\_Ellinika.mwd* έχουν χρησιμοποιηθεί και οι δύο τρόποι που αναφέρθηκαν σε αυτή την ενότητα. Μπορείτε να χρησιμοποιήσετε αυτό το μικρόκοσμο ως βάση για να βάλετε τα δικά σας κείμενα και αφού επιλέξετε έναν από τους δύο τρόπους που αναφέρθηκαν σβήστε το αντίστοιχο πεδίο και τον αντίστοιχο πλοηγό από το μικρόκοσμο.

## 7 Βιβλιοθήκη μαύρων κουτιών

### 7.1 *Shuffle* : Δημιουργεί μία λίστα με όσους αριθμούς ορίσουμε εμείς, σε τυχαία σειρά

```
to shuffle :items
  randomize
  make "a []
  make "nr 0
  while [:nr < :items] [
    make "b 1 + random :items
    make "c memberp :b :a
    if not :c [make "a fput :b :a]
    make "nr length :a
  ]
  output :a
end
```

## 8 Βιβλιογραφικές αναφορές

Edwards, L. (1998). Embodying Mathematics and Science: Microworlds as representations. *Journal of Mathematical Behavior*, 17 (1), pp. 53-78.

Harvey, B. (1997). *Computer Science Logo Style* (2nd ed.). Cambridge, Massachusetts: MIT Press.

Hoyles, C. (1995). Thematic chapter: Exploratory software, exploratory cultures? In A. DiSessa, C. Hoyles, & R. Noss, *Computers and Exploratory Learning* (pp. 199 - 220). NATO ASI Series, Berlin Springer - Verlag.

Kynigos, C. (2004). A "Black-and-White Box" Approach to User Empowerment With Component Computing. *Interactive Learning Environments*, 12 (1-2), pp. 27-71.

Kynigos, C. (2007). Half - Baked logo microworlds as boundary objects in integrated design. *Informatics in Education*, 6 (2), pp. 1-24.

Κυνηγός, Χ., Γιαννούτσου, Ν., & Φράγκου, Σ. (2006). Μετατρέποντας "Μισοψημένους μικρόκοσμους" σε ηλεκτρονικά παιχνίδια: μία πρόταση για τη διδασκαλία του προγραμματισμού. Στο Β. Δαγδιδέλης, & Δ. Ψύλλος (Επιμ.), *Πρακτικά του 5ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή: Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση*, (σσ. 440-448). Θεσσαλονίκη, 5-8 Οκτωβρίου .

## 9 Index

Αβάκιο Composer.....	8, 11	Διαχειριστή προγραμμάτων ενεργειών ψηφίδων	15, 18, 19, 20, 21
Αβάκιο Viewer.....	8, 9, 10	Πραγματοποίηση Συμβάντος	18
γεγονότα .....	16, 18, 19, 20, 21, 22, 24	ορισμός διαδικασίας	25

## 10 Πίνακας Βασικών ενεργειών

### Συμβουλές

Αλλαγή μεγέθους ψηφίδας .....	14
Άνοιγμα αρχείων Αβακίου πάντοτε με το Αβάκιο Composer .....	11
<b>Εντολές Ψηφίδων: ποιόν τρόπο προβολής προτιμάμε</b> .....	33
Εντολές: showview .....	66
<b>Εντολές: εκτέλεση εντολών στη Logo</b> .....	36
<b>Εντολές: η σημασία των κενών μεταξύ εντολών</b> .....	27
Εντολές: Χρήση εισαγωγικών .....	27
Ισχύς των εντολών που δημιουργούμε στη Logo .....	26
Όνομα ψηφίδας vs κείμενο ψηφίδας.....	14
<b>Όνομα ψηφίδας: Αλλαγή</b> .....	14
<b>Όνομα ψηφίδας: Περισσότερες από μία λέξεις</b> .....	29
Πρόσφατα αρχεία στο Αβάκιο .....	11