



**Τ.Ε.Ι. Κρήτης, Παράρτημα Ρεθύμνου
Τμήμα Μουσικής Τεχνολογίας Και Ακουστικής**

Πτυχιακή Εργασία

Θέμα :

**“Δημιουργία προγράμματος καλλιέργειας της μουσικής
ακοής με τη χρήση του Macromedia Director”**



**Σπουδαστής :
Μετήλλιας Ορέστης**

**Επιβλέποντες Καθηγητές :
Παπαρρηγόπουλος Κώστας
Ανδρουλάκη Μαρία**

**Ρέθυμνο
Ιούνιος 2011**

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	3
ΚΕΦΑΛΑΙΟ 1 ΜΟΥΣΙΚΗ ΚΑΙ ΕΚΠΑΙΔΕΥΣΗ	4
1.1. ΚΑΛΛΙΕΡΓΕΙΑ ΤΗΣ ΜΟΥΣΙΚΗΣ ΑΚΟΗΣ	4
1.2. ΕΚΠΑΙΔΕΥΣΗ	4
1.3. ΑΠΟΛΥΤΗ ΑΚΟΗ	5
1.3.1. Σχετική ακοή	5
1.3.2. Επίκτητη ή εγγενής ικανότητα;	5
1.3.3. Απόλυτη ακοή, ενασχόληση με τη μουσική και μουσικοπαιδαγωγική πράξη	6
1.4. ΛΟΓΙΣΜΙΚΟ ΕΚΠΑΙΔΕΥΣΗΣ – ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	7
ΚΕΦΑΛΑΙΟ 2 ΥΠΟΛΟΓΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΑ	9
2. 1. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	9
2.1.1. Ορισμός	9
2.1.2. Ιστορία	10
2.2. ΑΛΓΟΡΙΘΜΟΣ	11
2.2.1. Ορισμός - λειτουργικότητα	11
2.2.2. Διαγράμματα ροής	12
2.3. GRAPHICAL USER INTERFACE (GUI)	15
2.4. MACROMEDIA DIRECTOR	16
2.4.1. Ταινία Director	16
2.4.2. Cast	16
2.4.3. Η σκηνή (Stage)	17
2.4.4. Score	17
2.4.5. Event Handlers	17

ΚΕΦΑΛΑΙΟ 3	ΑΝΑΛΥΣΗ ΛΟΓΙΣΜΙΚΟΥ	19
3.1.	Το Κεντρικό Μενού	20
3.2.	Dictée	22
3.3.	Αναγνώριση Συγχορδίων	28
3.4.	Αναγνώριση Διαστημάτων	30
3.5.	Ρυθμική Αγωγή	32
3.6.	Ήχοι	34
ΕΠΙΛΟΓΟΣ		35
ΠΑΡΑΡΤΗΜΑΤΑ		37
ΒΙΒΛΙΟΓΡΑΦΙΑ		84

ΕΙΣΑΓΩΓΗ

Με αφορμή τη χρήση του προγράμματος EarMaster PRO στο εργαστήριο των "Στοιχείων Θεωρίας & Σημειογραφίας της Μουσικής" και το ιδιαίτερο ενδιαφέρον μου προς τη μουσική, τον προγραμματισμό και την εκπαίδευση, θεώρησα σωστό να δημιουργήσω ένα αντίστοιχο πρόγραμμα καλλιέργειας της μουσικής ακοής. Μετά από σχετική έρευνα στο διαδίκτυο, διαπίστωσα ότι στο εμπόριο κυκλοφορούν ελάχιστα παρόμοια προγράμματα, καθώς και ότι κανένα δεν υποστηρίζει την Ελληνική γλώσσα. Έχοντας υπόψιν τα παραπάνω, ξεκίνησα να σχεδιάζω την παρούσα εργασία.

Το εκπαιδευτικό λογισμικό χωρίζεται σε τέσσερα τμήματα, το κάθε ένα μια διαφορετική άσκηση πάνω στην καλλιέργεια της ακοής και της μουσικής κατανόησης.

Το πρώτο τμήμα, που είναι και το κυριότερο, είναι το Dictée, μία άσκηση που χρησιμοποιείται κατά κόρον από ωδεία και μουσικές σχολές. Ο στόχος σε αυτήν την άσκηση είναι να καταγράψει στο πεντάγραμμο ο εκπαιδευόμενος μια άγνωστη μελωδία ακούγοντάς τη.

Το δεύτερο τμήμα είναι η αναγνώριση συγχορδιών, κατά την οποία παίζονται στον εκπαιδευόμενο διάφορες συγχορδίες και αυτός πρέπει να τις αναγνωρίσει με τη χρήση πολλαπλής επιλογής.

Η τρίτη άσκηση είναι η αναγνώριση διαστημάτων, όπου πρέπει να βρεθούν τα διαστήματα που ακούγονται με τη χρήση της πολλαπλής επιλογής.

Τέλος, είναι η ρυθμική αγωγή, η οποία διαφέρει από τα υπόλοιπα τμήματα στον τρόπο που δίνεται το ρυθμικό σχήμα. Εδώ η άσκηση δίνεται γραμμένη και ο εκπαιδευόμενος πρέπει να την αναπαραγάγει σε πραγματικό χρόνο.

Το λογισμικό δημιουργήθηκε στο Macromedia Director 2004 και τα στοιχεία που το χαρακτηρίζουν είναι η αποκλειστική χρήση της Ελληνικής γλώσσας, η ευκολία στον χειρισμό και το ευχάριστο στυλ.

ΚΕΦΑΛΑΙΟ 1

ΜΟΥΣΙΚΗ ΚΑΙ ΕΚΠΑΙΔΕΥΣΗ

1.1. ΚΑΛΛΙΕΡΓΕΙΑ ΤΗΣ ΜΟΥΣΙΚΗΣ ΑΚΟΗΣ

Η μουσική ακοή είναι η ικανότητα που αναπτύσσει κάποιος για να αναγνωρίζει εξ ακοής μελωδίες, μουσικά διαστήματα, συγχορδίες, ρυθμικά σχήματα, χροιές και άλλα στοιχεία της μουσικής

1.2. ΕΚΠΑΙΔΕΥΣΗ

Στα ωδεία και τις μουσικές σχολές, η καλλιέργεια της μουσικής ακοής καλύπτεται από το μάθημα Dictée κατά το οποίο ο καθηγητής παίζει μια μελωδία στο πιάνο και οι μαθητές πρέπει με δοσμένο το μέτρο, την κλίμακα και την πρώτη νότα της μελωδίας, να την καταγράψουν στο πεντάγραμμο. Κατά την διάρκεια των σπουδών τους οι μαθητές ξεκινούν με απλές ασκήσεις που περιλαμβάνουν μικρά διαστήματα και απλά ρυθμικά σχήματα. Καθώς οι νεαροί μουσικοί εξοικειώνονται με τις απλές μελωδίες, εισάγονται μεγαλύτερα και δυσκολότερα διαστήματα και πιο περίπλοκοι ρυθμοί όπως τρίηχα, συγχοπές και άλλα.

Άλλη άσκηση είναι η αναγνώριση συγχορδιών, όπου ο καθηγητής τις παίζει στο πιάνο και οι μαθητές πρέπει να τις αναγνωρίσουν, δηλαδή να βρουν εάν η συγχορδία είναι μείζονα, ελάσσονα, αυξημένη, ελαττωμένη, μεθ'εβδόμης, μεθ'ενάτης, ανεστραμμένη και ούτω καθεξής. Οι συγχορδίες μπορεί να είναι μόνες τους ή σε διαδοχές.

Μια λιγότερο διαδεδομένη άσκηση είναι η αναγνώριση της χροιάς. Το κάθε όργανο έχει ξεχωριστή χροιά και είναι χρήσιμο για έναν μουσικό να μπορεί να διακρίνει τα όργανα σε μια ορχήστρα εξ ακοής. Για εξάσκηση μπορεί να χρησιμοποιηθεί οποιοδήποτε μέσο αναπαραγωγής.

1.3. ΑΠΟΛΥΤΗ ΑΚΟΗ

Υπάρχει στη μουσική αυτό που λέμε *απόλυτος τόνος* που είναι η ικανότητα να αναγνωρίζει κάποιος τη νότα που ακούει χωρίς να την συσχετίζει με κάποια άλλη. Το ανθρώπινο αυτί μπορεί εύκολα να αναγνωρίσει αν ένας τόνος είναι πιο χαμηλός η ψηλός συχνοτικά από έναν άλλο και κατά πόσο αλλά, συνήθως, δεν μπορεί να ξέρει ακριβώς πια νότα ακούει αν δεν του δοθεί σε συνδυασμό με κάποια άλλη, γνωστή νότα.

1.3.1. Σχετική ακοή

Σε αντιδιαστολή με την απόλυτη ακοή βρίσκεται η σχετική ακοή, η οποία αναφέρεται στην ικανότητα να μπορεί κανείς να παράγει, να αναγνωρίζει ή να κατονομάζει μουσικούς φθόγγους βασιζόμενος στην αναγνώριση των μουσικών διαστημάτων που σχηματίζουν με έναν ήχο αναφοράς. Για παράδειγμα, εάν παίζουμε σε έναν κάτοχο σχετικής ακοής δύο διαδοχικές νότες με απόσταση της πρώτης από την δεύτερη μια τρίτη μικρή προς τα πάνω λέγοντάς του ότι η πρώτη είναι η Λα και του ζητήσουμε να βρει τη δεύτερη, βασιζόμενος στην αναγνώριση του διαστήματος της τρίτης μικρής θα μας πει ότι η δεύτερη είναι η Ντο. Εάν του λέγαμε ότι η πρώτη νότα ήταν Σολ θα ονόμαζε τη δεύτερη νότα Σι ύφεση, πάλι μια τρίτη μικρή προς τα πάνω, άσχετα με το τι πραγματικά του είχαμε παίξει και χωρίς να μπορεί να αντιληφθεί ο ίδιος εάν τελικά οι ονομασίες των φθόγγων που αποτελούσαν το διάστημα που του παίζαμε ήταν πραγματικά αυτές.

1.3.2. Επίκτητη ή εγγενής ικανότητα;

Οι νεύρο-ανατομικές διαφορές που παρατηρούνται ανάμεσα στους μουσικούς με απόλυτη ακοή και στους μουσικούς χωρίς απόλυτη ακοή δημιουργούν το ερώτημα για το κατά πόσο είναι επίκτητες ή εγγενείς. Για τον Levitin (2005) οι σχέσεις αιτίου και αποτελέσματος ανάμεσα στην επίδραση της φύσης και της ανατροφής στην εμφάνιση της απόλυτης ακοής στον άνθρωπο δεν είναι εύκολο να διερευνηθούν. Φαίνεται ότι τα

βρέφη γεννιούνται με την ικανότητα να μπορούν να χρησιμοποιούν στοιχεία απόλυτου τονικού ύψους κατά την αντίληψη μελωδιών αλλά στη συνέχεια, η φυσιολογική ανάπτυξη ή η μουσική παιδεία στην οποία εκτίθενται μπορεί να μετατρέψουν τη στρατηγική για την αντίληψη των μελωδιών σε αυτή της σχετικής ακοής. Οι μουσικοί οι οποίοι εμφανίζουν απόλυτη ακοή μπορεί να ξεκίνησαν τη συστηματική τους ενασχόληση με τη μουσική πριν από την κρίσιμη ηλικία των 9 ετών, αλλά ίσως είχαν και τις ανάλογες γενετικές ή νευρολογικές προδιαθέσεις για να το καταφέρουν (Keenan et al, 2001).

1.3.3. Απόλυτη ακοή, ενασχόληση με τη μουσική και μουσικο-παιδαγωγική πράξη

Συχνά οι μουσικοί θεωρούν την απόλυτη ακοή ως μια πολύτιμη ικανότητα γιατί μπορεί να βοηθήσει κάποιον να αναγνωρίζει αμέσως και με το αυτί το τονικό ύψος των ήχων ή την τονικότητα ενός έργου, να τραγουδά *prima vista* με ακρίβεια ή να συνθέτει και να γράφει νότες χωρίς τη βοήθεια κάποιου μουσικού οργάνου. Όμως, εκτός από την άμεση αναγνώριση και κατονομασία του τονικού ύψους των ήχων η απόλυτη ακοή δεν συσχετίζεται με καμία από τις υπόλοιπες μουσικές δραστηριότητες. Για παράδειγμα, δεν μπορούμε να πούμε ότι συνθέτες με απόλυτη ακοή τονικού ύψους (όπως για παράδειγμα ο Mozart, ο Skryabin, ο Messiaen, ή ο Boulez) έγραψαν καλύτερα ή χειρότερα μουσικά έργα από συνθέτες που δεν είχαν απόλυτη ακοή (όπως για παράδειγμα ο Wagner, ο Tchaikovsky, ο Ravel ή ο Stravinsky) (Parncutt & Levitin, 2000).

Αντιθέτως, η απόλυτη ακοή μπορεί να δημιουργήσει προβλήματα στους κατόχους της στην αναγνώριση των διαστηματικών σχέσεων ανάμεσα στις νότες ή στην αναγνώριση της μεταφοράς μιας μελωδίας σε άλλες τονικότητες, κάτι που οι μουσική με σχετική ακοή κάνουν με ιδιαίτερη ευκολία (Miyazaki & Rakowski, 2002). Μπορεί επίσης να εμποδίσει τους κατόχους της από το να ευχαριστηθούν τη μουσική που ακούν αν δεν πληρούνται στην εντέλεια οι προϋποθέσεις του σωστού, σύμφωνα με την απόλυτη ακοή, τονικού ύψους. Τέλος, έχει παρατηρηθεί, ότι με το πέρασμα της ηλικίας η απόλυτη ακοή μπορεί να αλλάξει και να διαφοροποιηθεί μέχρι ως και δύο ημιτόνια,

δημιουργώντας συχνά στους κατόχους της μια δυσφορία κατά την ακρόαση γνωστών, πριν της αλλαγής της, μουσικών έργων (Wynn, 1992).

Από την άλλη πλευρά, το φαινόμενο της κρίσιμης ηλικίας μέχρι τα 9 έτη για την απόκτηση της απόλυτης ακοής υπογραμμίζει και τη μεγάλη σημασία που έχει η σωστή μουσική εκπαίδευση που πρέπει να λαμβάνουν τα παιδιά της προσχολικής και πρωτοσχολικής ηλικίας ώστε οι εσωτερικές μουσικές αναπαραστάσεις που θα αναπτύξουν να είναι αυτές που ταιριάζουν στη φυσική ανάπτυξή τους και θα τα βοηθήσουν να μάθουν να τραγουδούν σωστά και να αντιλαμβάνονται τη μουσική πιο ολοκληρωμένα. Για παράδειγμα, οι Sergeant & Roche (1973) παρατήρησαν ότι τα παιδιά μικρής ηλικίας είναι ιδιαίτερα ευαίσθητα ως προς την απόλυτη ακοή για μουσικά κομμάτια από ότι τα μεγαλύτερα παιδιά. Εάν διδάσκονται ένα τραγούδι σε μια συγκεκριμένη τονικότητα μπορούν στη συνέχεια από μόνα τους να συνεχίσουν να το τραγουδούν στην ίδια τονικότητα.

Από τα παραπάνω γίνεται φανερό ότι ο ρόλος του εξειδικευμένου μουσικοπαιδαγωγού στη διδασκαλία της μουσικής σε παιδιά προσχολικής και πρωτοσχολικής ηλικίας είναι ιδιαίτερα σημαντικός. Οι μουσικές εμπειρίες στις οποίες εκτίθενται τα παιδιά σε αυτές τις ηλικίες μετατρέπονται σε μουσικές βάσεις πάνω στις οποίες θα χτίσουν όλες τις μετέπειτα στρατηγικές τους προσέγγισης και ενασχόλησης με τη μουσική. Μόνο ο εξειδικευμένος μουσικοπαιδαγωγός γνωρίζει με ποιες μουσικές δραστηριότητες πρέπει να προσεγγίσει τα παιδιά αυτής της ηλικίας ώστε να ενδυναμώσει και να ενθαρρύνει την ανάπτυξη του μουσικού δυναμικού τους με έναν φυσικό, σύμφωνα με το ρυθμό της ανάπτυξης τους, τρόπο. Το παρών εκπαιδευτικό λογισμικό κατατάσσεται στα εργαλεία του μουσικοπαιδαγωγού και τείνει περισσότερο στην καλλιέργεια της σχετικής ακοής.

1.4. ΛΟΓΙΣΜΙΚΟ ΕΚΠΑΙΔΕΥΣΗΣ – ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Η ακριβής αναγνώριση και αναπαραγωγή μουσικών διαστημάτων, κλιμάκων, συγχορδιών, ρυθμών και άλλων πτυχών της καλλιέργειας της μουσικής ακοής συχνά

απαιτεί πολλή εξάσκηση. Ασκήσεις που περιλαμβάνουν αναγνώριση συνήθως χρειάζονται ένα δεύτερο άτομο που να γνωρίζει μουσική για να παίζει τις ερωτήσεις σε κάποιο όργανο και να επαληθεύει τις απαντήσεις. Λογισμικό εξειδικευμένο στη μουσική θεωρία μπορεί να άρει την ανάγκη για παρτενέρ και να προσαρμόσει τις ασκήσεις στις ανάγκες του χρήστη. Πανεπιστημιακά τμήματα μουσικής συχνά χρησιμοποιούν λογισμικό του εμπορίου όπως τα EarMaster, Auralia (Ear Training Software) και MacGAMUT για την εξάσκηση των φοιτητών. Υπάρχει επίσης μικρός αριθμός δωρεάν προγραμμάτων. Το μεγαλύτερο προσόν της παρούσας εργασίας είναι ότι θα παραχθεί ένα ακόμα δωρεάν λογισμικό εκπαίδευσης με την καινοτομία ότι θα είναι αποκλειστικά στα Ελληνικά. Άλλα χαρακτηριστικά του προγράμματος είναι η απλότητα στη χρήση και το κωμικό θέμα που χρησιμοποιεί στην παρουσίαση του. Όλα αυτά το καθιστούν ιδανικό για χρήση από παιδιά μικρής ηλικίας που μόλις έχουν αρχίσει να εξοικειώνονται με τη θεωρία της μουσικής, ενώ ταυτόχρονα είναι πρακτικό και για πιο απαιτητικούς χρήστες.

Η πλειοψηφία των λογισμικών καλλιέργειας της μουσικής ακοής βασίζονται σε πρωτόκολλα midi, επιτρέποντας στον χρήστη να προσαρμόσει τα όργανα αναπαραγωγής και ακόμα και να εισάγει πληροφορίες με συσκευές συμβατές με midi. Λόγω ασυμβατότητας της πλατφόρμας στην οποία κατασκευάστηκε το πρόγραμμα και των πρωτοκόλλων midi, χρησιμοποιήθηκαν ηχογραφημένοι ήχοι από τέσσερα όργανα και δεν υπάρχει η δυνατότητα αλλαγής των χαρακτηριστικών τους ή εισαγωγής εφέ.

ΚΕΦΑΛΑΙΟ 2

ΥΠΟΛΟΓΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΑ

2.1. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Προγραμματισμός είναι η διαδικασία του σχεδιασμού, γραψίματος, δοκιμής, επίλυσης λαθών και διατήρησης του κώδικα προγραμμάτων υπολογιστή. Ο κώδικας γράφεται σε ειδικά λογισμικά που λέγονται γλώσσες προγραμματισμού. Παραδείγματα είναι η “C”, η “COBOL”, η “Java” και η “Lingo”, στην οποία γράφτηκε το παρόν πρόγραμμα. Σκοπός του προγραμματισμού είναι να δημιουργηθεί ένα πρόγραμμα που επιδεικνύει μια συγκεκριμένη συμπεριφορά. Η συγγραφή κώδικα συχνά απαιτεί πραγματογνωμοσύνη σε διάφορους τομείς, συμπεριλαμβανομένης της γνώσης πάνω στον τομέα εφαρμογής του προγράμματος, των εξειδικευμένων αλγόριθμων και της λογικής.

2.1.1. Ορισμός

Ο ορισμός του προγραμματισμού δόθηκε από τους Hoc και Nguyen-Xuan ως “η διαδικασία του να μεταμορφώνεις ένα πνευματικό σχέδιο σε ένα συμβατό με τον υπολογιστή.”

Είναι θέμα συνεχούς συζήτησης ο βαθμός στον οποίο ο προγραμματισμός είναι τέχνη, δεξιότητα ή αρχή της μηχανικής. Γενικά, καλός προγραμματισμός θεωρείται η μετρημένη εφαρμογή και των τριών, με στόχο την παραγωγή ενός αποδοτικού και εξελίξιμου λογισμικού (τα κριτήρια για “αποδοτικό” και “εξελίξιμο” ποικίλλουν). Ένα ακόμα θέμα υπό συζήτηση είναι το κατά πόσο η γλώσσα προγραμματισμού επηρεάζει την τελική μορφή που θα πάρει το πρόγραμμα. Το θέμα αυτό είναι αντίστοιχο με την υπόθεση Sapir-Whorf στη γλωσσολογία, η οποία λέει ότι η φύση μίας γλώσσας επηρεάζει τη σκέψη στις συνήθειες αυτών που τη μιλάνε. Διαφορετικά μοτίβα γλώσσας παράγουν διαφορετικά μοτίβα σκέψης. Η ιδέα αυτή είναι μια πρόκληση στην

προσπάθεια να περιγράψουμε τον κόσμο τέλεια με τη γλώσσα, γιατί αναγνωρίζει ότι οι μηχανισμοί μιας γλώσσας διέπουν τις σκέψεις της κοινότητας των ομιλητών της.

2.1.2. Ιστορία

Ο μηχανισμός των Αντικυθήρων από την αρχαία Ελλάδα ήταν ένας υπολογιστής που χρησιμοποιούσε γρανάζια διαφόρων μεγεθών και σχηματισμών για να προβλέπει τον Μετωνικό Κύκλο, ο οποίος χρησιμοποιείται ακόμα σε κάποιους τύπους ημερολογίων και που είναι αξιόπιστος στον υπολογισμό των ημερομηνιών των Ολυμπιάδων. Ο Αλ Τζαζάρι έφτιαξε Αυτόματα τα οποία επιδέχονταν προγραμματισμό το 1206. Ένα από τα συστήματα που χρησιμοποιούσε στις συσκευές αυτές ήταν το εξής : σφήνες και δόντια ήταν τοποθετημένα σε ένα ξύλινο τύμπανο σε συγκεκριμένες τοποθεσίες, το οποίο ενεργοποιούσε μοχλούς που με τη σειρά τους λειτουργούσαν κάποια κρουστά όργανα. Η μηχανή αυτή ήταν σαν ένας μικρός τυμπανιστής που έπαιζε διάφορα ρυθμικά μοτίβα.

Στα τέλη του 1880, ο Χέρμαν Χόλλεριθ εφηύρε την καταγραφή στοιχείων σε ένα μέσο που μπορούσε ύστερα να διαβαστεί από μια μηχανή. Προηγούμενες χρήσεις αναγνώσιμων μέσων αποσκοπούσαν σε έλεγχο κάποιου μηχανικού μέρους και όχι καταγραφή στοιχείων. Σαν μέσο χρησιμοποιούσε διάτρητες κάρτες, γνωστές αρχικά σαν “κάρτες Χόλλεριθ”. Για να επεξεργαστεί αυτές τις κάρτες εφηύρε επίσης τον Πινακοποιητή και τον Πληκτροτυπητήρα. Οι τρεις αυτές συσκευές ήταν τα θεμέλια πάνω στα οποία στηρίχτηκε η σύγχρονη βιομηχανία πληροφορικής. Το 1896 ίδρυσε την Tabulating Machine Company (που αργότερα έγινε ο πυρήνας της IBM). Η πρόσθεση ενός πίνακα ελέγχου στον Τύπου I Πινακοποιητή του επέτρεψε να κάνει διαφορετικές δουλειές χωρίς να αναγκάζεται να συναρμολογείται διαφορετικά. Μέχρι τα τέλη του 1940, υπήρχε πληθώρα



Εικ. 1.1. Στοιχεία και εντολές αποθηκεύονταν σε εξωτερικές διάτρητες κάρτες.

προγραμματιζόμενων μηχανών, οι οποίες λέγονταν Εξοπλισμός Καταγραφής Μονάδας και εκτελούσαν εργασίες επεξεργασίας στοιχείων. Πρώιμοι προγραμματιστές υπολογιστών χρησιμοποιούσαν αυτά τα εργαλεία για διάφορους περίπλοκους υπολογισμούς.

Η εφεύρεση της αρχιτεκτονικής von Neumann επέτρεψε την αποθήκευση προγραμμάτων σε μνήμη του υπολογιστή. Τα πρώτα προγράμματα γραφόντουσαν με δυσκολία χρησιμοποιώντας τις οδηγίες του κάθε συγκεκριμένου μηχανήματος, συχνά και σε δυαδική μορφή.

Με την πάροδο του χρόνου δημιουργήθηκαν καλύτερες, πιο εξελιγμένες γλώσσες προγραμματισμού. Από την Assembly πήγαμε στην FORTRAN, στην COBOL και σε πολλές ακόμα. Η Lingo είναι μια γλώσσα που δημιουργήθηκε για τις ανάγκες του Macromedia Director από τον John H. Thompson. Η Lingo χρησιμοποιεί συντακτικό που είναι πολύ κοντά στον ανθρώπινο λόγο με πολλά προαιρετικά στοιχεία όπως κάποια άρθρα. Χρησιμοποιείται επίσης στο Adobe Shockwave, σε desktop εφαρμογές, σε CD-ROMs και άλλα.

2.2. ΑΛΓΟΡΙΘΜΟΣ

Ένα πολύ σημαντικό κομμάτι της εργασίας είναι ο αλγόριθμος ο οποίος θα παράγει τυχαίες μελωδίες διαφορετικές κάθε φορά και προσδιορισμένες από τις ανάγκες του χρήστη. Ο αλγόριθμος αυτός καθώς και όλος ο προγραμματισμός και η μορφοποίηση του προγράμματος θα γίνει στη γλώσσα προγραμματισμού Lingo.

2.2.1. Ορισμός - λειτουργικότητα

Στα μαθηματικά και στους υπολογιστές, αλγόριθμος είναι μια αποτελεσματική μέθοδος με τη μορφή μιας πεπερασμένης λίστας καλά-προσδιορισμένων οδηγιών για τον

υπολογισμό μιας λειτουργίας. Αλγόριθμοι χρησιμοποιούνται για υπολογισμό, επεξεργασία δεδομένων και αυτοματοποιημένη λογική (τεχνητή νοημοσύνη).

Η λειτουργία ενός αλγόριθμου είναι να πάρει κάποιο στοιχείο που εισάγει ο χρήστης και να το τροποποιήσει σε μια τελική, θεμιτή μορφή, αφού πρώτα έχει περάσει από άλλες διαδοχικές και συγκεκριμένες μορφές. Η μετάβαση από τη μια κατάσταση (μορφή) στην άλλη δεν είναι απαραίτητα ντετερμινιστική. Κάποιοι αλγόριθμοι ενσωματώνουν και τυχαία στοιχεία.

Για να πούμε ότι ένας αλγόριθμος είναι «καλοφτιαγμένος» πρέπει :

1. Να δίνει το αποτέλεσμα για το οποίο δημιουργήθηκε. Με άλλα λόγια εάν φτιάξαμε έναν αλγόριθμο που να παίρνει τα στοιχεία μίας λίστας και να μας λέει πιο είναι το μεγαλύτερο αριθμητικά πρέπει καταρχήν να πετυχαίνει αυτό!
2. Να είναι ευέλικτος και να μπορεί να ανταποκριθεί σε οποιαδήποτε περίπτωση. Παραδείγματος χάριν να μπορεί να αντεπεξέλθει στην περίπτωση που ένα στοιχείο της προαναφερθείσας λίστας δεν είναι αριθμητικό.
3. Να ολοκληρώνει τη λειτουργία του στο μικρότερο δυνατό χρόνο. Συνεπάγεται ότι δεν πρέπει να «φορτώνει» πολύ τη μνήμη του συστήματος στο οποίο τρέχει.
4. Ο κώδικάς πρέπει να είναι απλός και συμπυκνωμένος όσο δυνατόν περισσότερο, χωρίς αυτό να επηρεάζει τα τρία προηγούμενα κριτήρια.

2.2.2. Διαγράμματα ροής

Διάγραμμα ροής (flowchart) είναι ένα κοινού τύπου διάγραμμα που αναπαριστά έναν αλγόριθμο ή μια διαδικασία, δείχνοντας τα βήματα ως κουτιά διαφόρων ειδών που συνδέονται μεταξύ τους με βέλη. Αυτή η διαγραμματική παρουσίαση μπορεί να δώσει λύση βήμα προς βήμα σε ένα γνωστό πρόβλημα. Τα δεδομένα αναπαριστώνται σε κουτιά και τα βέλη δείχνουν τη ροή τους. Τα διαγράμματα ροής χρησιμοποιούνται στην ανάλυση, το σχεδιασμό, την τεκμηρίωση ή τον έλεγχο μιας διαδικασίας ή ενός προγράμματος σε διάφορα πεδία.

Ένα τυπικό διάγραμμα ροής από παλαιότερα εγχειρίδια της επιστήμης των υπολογιστών μπορεί να περιλαμβάνει τα ακόλουθα είδη συμβόλων:

Σύμβολα Έναρξης και Λήξης

Αναπαριστώνται ως κύκλοι, οβάλ ή στρογγυλεμένα ορθογώνια παραλληλόγραμμα που περιέχουν τη λέξη «έναρξη», «λήξη» ή μια φράση που να δείχνει την αρχή ή το τέλος της διαδικασίας.

Βέλη

Δείχνουν αυτό που ονομάζεται στην επιστήμη των υπολογιστών «ροή ελέγχου». Ένα βέλος που έρχεται από ένα σύμβολο και καταλήγει σε ένα άλλο δείχνει ότι ο έλεγχος ακολουθεί την ίδια πορεία.

Στάδια Επεξεργασίας

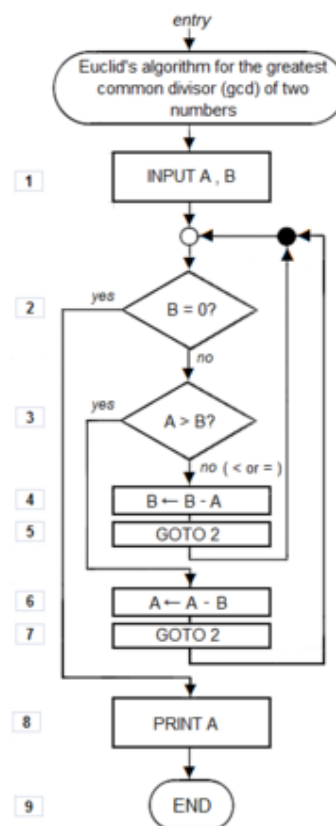
Αναπαριστώνται από ορθογώνια παραλληλόγραμμα, πχ. «πρόσθεσε 1 στο X», «αποθήκευσε τις αλλαγές» κλπ.

Είσοδος/Εξοδος

Παρουσιάζονται ως παραλληλόγραμμα, πχ. «δείξε X», «πάρε το X από τον χρήστη» κλπ.

Υποθέσεις/Αποφάσεις

Αναπαριστώνται από ρόμβους. Τυπικά περιέχουν ερώτηση «ΝΑΙ/ΟΧΙ» ή «ΑΛΗΘΕΣ/ΨΕΥΔΕΣ». Αυτό το σύμβολο συνήθως έχει δύο βέλη να βγαίνουν από αυτό, ένα από το πλάι που αντιστοιχεί στο ΟΧΙ/ΨΕΥΔΕΣ και ένα από κάτω που αντιστοιχεί στο ΝΑΙ/ΑΛΗΘΕΣ. Τα βέλη πρέπει πάντα να σημειώνονται. Μια απόφαση είναι απαραίτητη σε ένα διάγραμμα ροής. Μπορούν να χρησιμοποιηθούν παραπάνω από δύο βέλη, αλλά αυτό δείχνει ότι πρόκειται για μια σύνθετη απόφαση και σε αυτή την



Εικ. 1.2. Διάγραμμα ροής ενός αλγόριθμου (Ευκλείδειος Αλγόριθμος).

περίπτωση μπορεί να χρειαστεί να αναλυθεί περαιτέρω ή να αντικατασταθεί από το σύμβολο «προ-καθορισμένη διαδικασία».

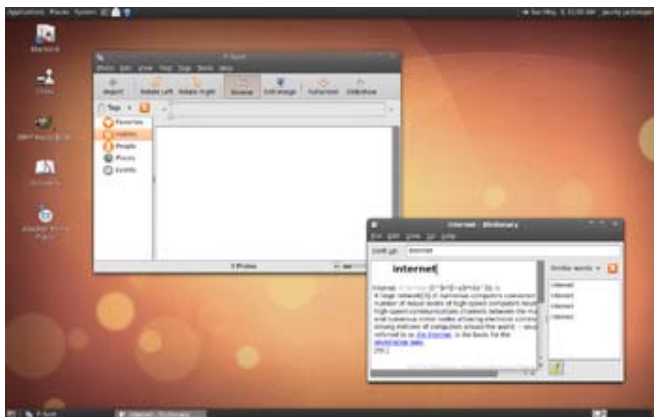
Μια σειρά από σύμβολα που είναι λιγότερο διαδεδομένα διεθνώς είναι:

- Ένα έγγραφο αναπαριστάται από ένα ορθογώνιο παραλληλεπίπεδο με κυματιστή βάση
- Μια Manual είσοδος αναπαριστάται από ένα παραλληλόγραμμο με το πάνω μέρος να γέρνει δεξιά και αριστερά
- Μια Manual λειτουργία αναπαριστάται από ένα τραπέζιο με τη μεγαλύτερη παράλληλη πλευρά στην κορυφή και δείχνει μια διαδικασία που μπορεί να γίνει μόνο με το χέρι
- Ένα αρχείο δεδομένων αναπαριστάται από έναν κύλινδρο

Τα διαγράμματα ροής μπορεί να περιέχουν και άλλα σύμβολα, όπως συνδέσμους για να δείξουν διαδρομές που συγκλίνουν, οι οποίοι συνήθως αναπαριστώνται από κύκλους. Οι κύκλοι θα έχουν περισσότερα από ένα βέλη ερχόμενα προς αυτούς, αλλά μόνο ένα εξερχόμενο. Κάποια διαγράμματα μπορεί να έχουν ένα βέλος, το οποίο να καταλήγει σε ένα άλλο. Αυτό είναι χρήσιμο στην αναπαράσταση μιας επαναληπτικής διαδικασίας (loop). Μια τέτοιου είδους διαδικασία για παράδειγμα, μπορεί να περιέχει έναν σύνδεσμο στον οποίο εισέρχεται πρώτα ο έλεγχος, μετά να ακολουθούν τα στάδια επεξεργασίας, μια απόφαση με ένα βέλος να εξέρχεται και ένα ακόμη βέλος να γυρνάει πίσω στο σύνδεσμο. Σύνδεσμοι «εκτός-σελίδας» χρησιμοποιούνται συνήθως για να εκφράσουν μια σύνδεση σε μια διαδικασία που συνεχίζεται σε άλλη σελίδα ή άλλη οθόνη. Είναι σημαντικό να διατηρείται μια λογική σειρά σε αυτές τις συνδέσεις. Όλες οι διαδικασίες θα πρέπει να ξεκινούν από την κορυφή προς το τέλος και από τα δεξιά προς τα αριστερά.

2.3. GRAPHICAL USER INTERFACE (GUI)

Για τις ανάγκες της εργασίας δημιουργήθηκε ένα ευχάριστο GUI με συγκεκριμένο στυλ και προσανατολισμό στα χρώματα. GUI είναι ένα μέσο με το οποίο αλληλεπιδρά ο άνθρωπος με τον υπολογιστή. Συνήθως είναι ένα γραφικό περιβάλλον το οποίο είναι κατασκευασμένο έτσι ώστε να είναι κομψό, εύχρηστο και κατανοητό στο χρήστη. Το GUI αντιπροσωπεύει τις διαθέσιμες πληροφορίες και δράσεις που μπορεί να πάρει ο χρήστης. Αυτό επιτυγχάνεται μέσω γραφικών εικονιδίων και οπτικοακουστικών ενδείξεων όπως κουμπιά, βέλη ή ακόμα και κείμενο, τα οποία ο χρήστης μπορεί να μεταχειριστεί απευθείας.



Εικ. 1.3. Φωτογραφία του GNOME GUI.

Ο σχεδιασμός της οπτικής σύνθεσης και τροπικής συμπεριφοράς ενός GUI είναι σημαντικό στοιχείο του προγραμματισμού εφαρμογών. Ο στόχος του είναι να ενισχύσει την αποδοτικότητα και ευκολία χρήσης για το υποβόσκον λογικό σχέδιο του προγράμματος. Τεχνικές ανθρωποκεντρικού σχεδιασμού χρησιμοποιούνται για να διασφαλίσουν ότι η οπτική επικοινωνία είναι ταιριαστή στο έργο που πρέπει να παραχθεί.

2.4. MACROMEDIA DIRECTOR



Εικ. 1.4. Εξώφυλλο του εγχειριδίου του Macromedia Director 2004.

Το Macromedia Director είναι μία πολυμεσική εφαρμογή κατασκευασμένη αρχικά με σκοπό τη δημιουργία animations. Η προσθήκη μίας ισχυρής γλώσσας, της Lingo, η οποία γράφτηκε για τις ανάγκες του Director, το έκανε δημοφιλή επιλογή για την δημιουργία CD-ROM και εφαρμογές web με τη χρήση του Shockwave. Το Director υποστηρίζει δισδιάστατες και τρισδιάστατες εφαρμογές και η Lingo επιτρέπει την μεταχείριση στοιχείων σε επίπεδο millisecond. Οι κύριες δυσκολίες της εργασίας ήταν δύο. Η πρώτη να ακούγονται οι θεμιτές νότες σε συγκεκριμένους χρόνους και η δεύτερη να υπάρχει δυνατότητα

ευελιξίας στα γραφικά, και πιο συγκεκριμένα στις νότες οι οποίες πρέπει να αλλάζουν ανάλογα με το πλήθος και την ποιότητά τους.

2.4.1. Ταινία Director

Αυτό που δημιουργεί κάποιος με το Macromedia Director είναι μία Ταινία Director, ή απλά ταινία. Μία ταινία έχει αρχή, μέση και τέλος. Μπορούμε να παίζουμε την ταινία, να την σταματήσουμε ή να την μεταχειριστούμε με όποιο τρόπο θέλουμε.

2.4.2. Cast

Αναφερόμαστε σε όλα τα στοιχεία της ταινίας ως cast members. Συμπεριλαμβάνονται όλες οι εικόνες οποιασδήποτε μορφής, κείμενα και κεφαλίδες, κουμπιά και ήχοι. Ακόμα και τα scripts που γράφουμε είναι cast members. Όλα τα cast members. Μπορούμε να αναφερθούμε σε ένα cast member χρησιμοποιώντας τον αριθμό που αντιστοιχεί στη θέση του στο cast ή με το όνομα που του έχουμε δώσει. Παρόλο που ένα στοιχείο πρέπει να είναι cast member για να χρησιμοποιηθεί στην ταινία, τα cast members εμφανίζονται μόνο αφού τα τοποθετήσουμε στην σκηνή (Stage).

2.4.3. Η σκηνή (Stage)

Όλη η δράση της ταινίας γίνεται στη σκηνή. Δημιουργούμε την ταινία μας τοποθετώντας τα στοιχεία της στην σκηνή. Αυτό που βλέπουμε στη σκηνή είναι στην πραγματικότητα ένα αντίγραφο του cast member που ονομάζουμε sprite. Το κάθε sprite έχει τις ιδιότητες του όπως η θέση του στη σκηνή και το μέγεθός του. Εάν θέλουμε να δημιουργήσουμε ένα animation, ή εάν η εφαρμογή μας έχει παραπάνω από μία σελίδες (οθόνες), θα πρέπει να κάνουμε διαφορετικές διαρρυθμίσεις στη σκηνή που η κάθε μία θα αντιστοιχεί σε διαφορετικά σημεία στο χρόνο (καρέ). Αυτό καθορίζεται στο score.

2.4.4. Score

Το κύριο μέρος του score είναι διαιρεμένο σε γραμμές και στήλες. Η κάθε στήλη αντιστοιχεί σε διαφορετικό σημείο στο χρόνο (καρέ). Κάθε στήλη αντιστοιχεί σε διαφορετικό κανάλι. Στα κανάλια μπαίνουν τα sprites που τοποθετούμε στη σκηνή. Ο αριθμός των καναλιών είναι πεπερασμένος για κάθε καρέ. Υπάρχουν επίσης ειδικά κανάλια για οτιδήποτε άλλο συμβαίνει στην ταινία, συμπεριλαμβανομένων και οχτώ καναλιών ήχου, ένα κανάλι χρονισμού και ένα κανάλι για script στο κάθε καρέ. Όταν παίζουμε την ταινία, το Director δείχνει το ένα καρέ μετά το άλλο με ρυθμό που τον ορίζουμε εμείς. Μπορούμε επίσης να μεταφερθούμε σε οποιοδήποτε καρέ θέλουμε και να αλλάξουμε οτιδήποτε ανά πάσα στιγμή με τη βοήθεια των Event Handlers.

2.4.5. Event Handlers

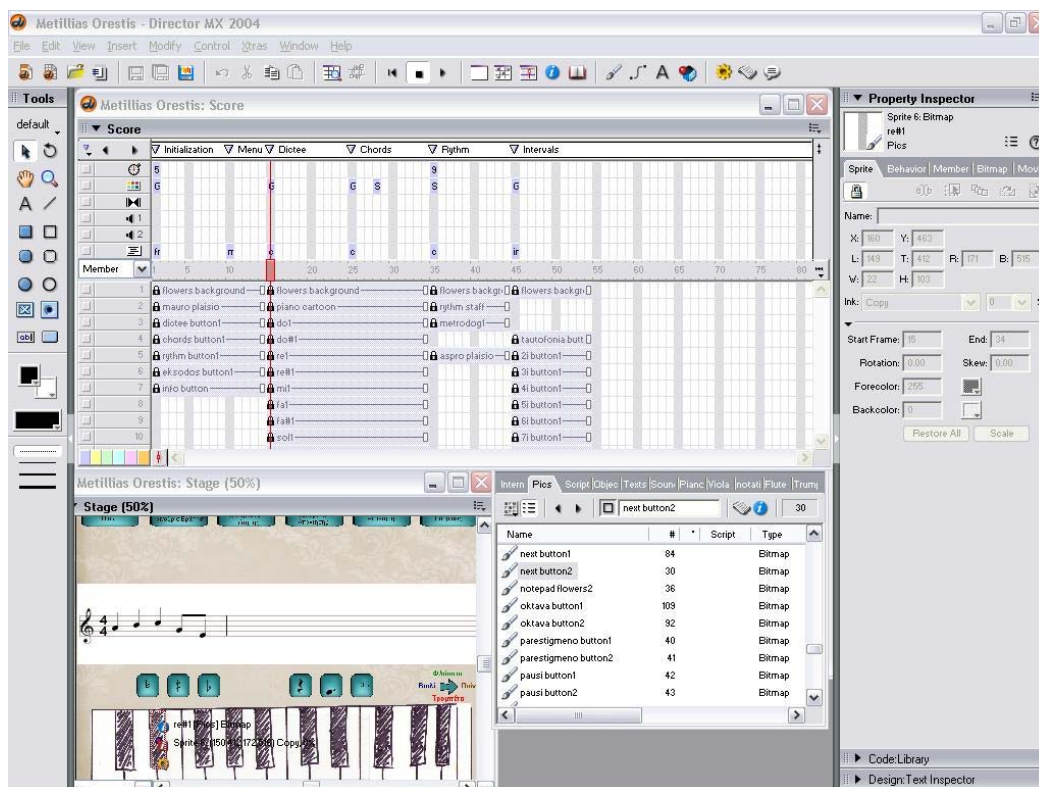
Η αναπαραγωγή της ταινίας θεωρείται σαν μία διαδοχή γεγονότων (events). Η ταινία ξεκινάει, προβάλλεται το κάθε καρέ, η ταινία σταματάει. Το Director αναγνωρίζει επίσης διάφορα άλλα γεγονότα : το κλικ του ποντικιού, το πάτημα ενός πλήκτρου στο πληκτρολόγιο, τη λήξη ενός χρονιστή και άλλα. Τα γεγονότα αυτά περνάνε απαρατήρητα εκτός αν έχουμε δημιουργήσει event handlers που υποδεικνύουν ότι θα συμβεί κάτι άλλο. Ένα event handler είναι μία διαδοχή εντολών του υπολογιστή. Όταν έρθει η στιγμή του συγκεκριμένου event, ο υπολογιστής ακολουθεί τις οδηγίες στο handler. Τα handlers γράφονται σε scripts με τη χρήση της Lingo. Η Lingo είναι μια ισχυρή γλώσσα

προγραμματισμού που υποστηρίζει οτιδήποτε μπορεί να χρειαστεί κάποιος για να γράψει ένα περίπλοκο πρόγραμμα.

ΚΕΦΑΛΑΙΟ 3

ΑΝΑΛΥΣΗ ΛΟΓΙΣΜΙΚΟΥ

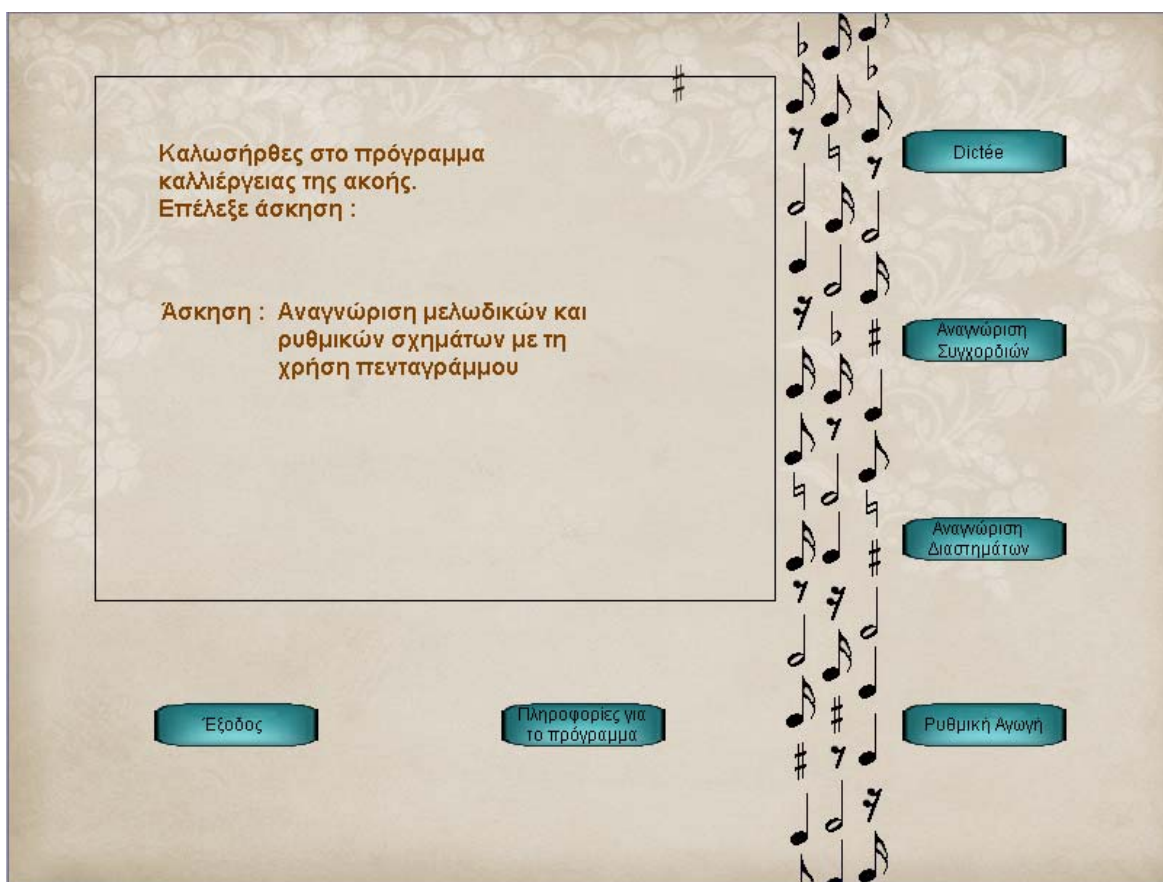
Η εργασία χωρίζεται σε τέσσερα επιμέρους κομμάτια, το καθένα με τη σειρά του μια άσκηση για την εκπαίδευση μίας συγκεκριμένης ιδιότητας της μουσικής ακοής. Πρακτικά, έχουν δημιουργηθεί στο Director πέντε διαφορετικές «σελίδες». Η μία αφορά στην καλλιέργεια της ρυθμικής αγωγής, η δεύτερη στην εύρεση μουσικών διαστημάτων, η τρίτη στην εύρεση συγχορδιών και η τέταρτη, που είναι και το μεγαλύτερο κομμάτι της εργασίας, είναι η εκπαίδευση στην αναγνώριση μελωδιών και περιλαμβάνει στοιχεία από την πρώτη και την δεύτερη άσκηση. Υπάρχει ακόμα μια αρχική σελίδα που λειτουργεί σαν υποδοχή, κεντρικό μενού και συνδετικός κρίκος ανάμεσα στις ασκήσεις. Όλες οι σελίδες έχουν το ίδιο φόντο και χρησιμοποιούν το ίδιο στυλ στα χρώματα, κουμπιά και



Εικ. 2.1. Το score του προγράμματος στο Director.

γραμματοσειρές. Για τη δημιουργία του πενταγράμμου, των νοτών, των κουμπιών και ορισμένων εικόνων χρησιμοποιήθηκαν τα προγράμματα Adobe Photoshop και Ms Paint. Παρακάτω θα εξεταστούν όλα τα μέρη του προγράμματος με τη σειρά κατά την οποία δημιουργήθηκαν.

3.1. Το Κεντρικό Μενού



Εικ. 2.2. Το κεντρικό μενού.

Η αρχική σελίδα είναι κυρίως ένα γραφικό περιβάλλον το οποίο μας εισάγει στο στυλ του προγράμματος, δίνει λίγες πληροφορίες για την εργασία και μας μεταφέρει στα επιμέρους τμήματα. Επίσης, από εδώ έχουμε την επιλογή να τερματίσουμε την εφαρμογή.

Η οθόνη αυτή αποτελείται από ένα πλαίσιο, μέσα στο οποίο εμφανίζονται πληροφορίες για την κάθε επιλογή καθώς μετακινούμε τον κέρσορα πάνω από τα κουμπιά (rollover), έξι κουμπιά, και μερικά γραφικά στοιχεία για την τέρψη του χρήστη. Τα κουμπιά είναι τα εξής :

- Dictée. Μας μεταφέρει στην άσκηση αναγνώρισης μελωδιών.
- Αναγνώριση Συγχορδιών. Μας μεταφέρει στην άσκηση αναγνώρισης συγχορδιών.
- Αναγνώριση Διαστημάτων. Μας μεταφέρει στην άσκηση αναγνώρισης διαστημάτων.
- Ρυθμική Αγωγή. Μας μεταφέρει στην άσκηση αναγνώρισης ρυθμικών σχημάτων.
- Πληροφορίες για το πρόγραμμα. Εμφανίζει πληροφορίες για το πρόγραμμα εντός του πλαισίου στην πάνω αριστερή γωνία της εικόνας όταν περνάμε τον κέρσορα από πάνω του. Ουσιαστικά δεν είναι κουμπί γιατί δεν εκτελεί κάποια λειτουργία όταν το πατάμε.
- Έξοδος. Τερματίζει την εφαρμογή.

Οι πληροφορίες για το κάθε κουμπί είναι ξεχωριστές εικόνες που εμφανίζονται στη σκηνή μόλις ο κέρσορας περάσει πάνω από το αντίστοιχο κουμπί. Ο βέλτιστος τρόπος να γίνει αυτή η λειτουργία θα ήταν να έχουμε ένα cast member και να αλλάζουμε το κείμενό του, κάτι το οποίο δεν γίνεται γιατί το Macromedia Director δεν υποστηρίζει Ελληνικά. Το εμπόδιο υπερπηδήθηκε με την δημιουργία ενός jpeg για κάθε κείμενο στο Ms Paint.

Τα προαναφερθέντα γραφικά στοιχεία είναι δύο. Το πρώτο είναι ένας «καταρράκτης» από νότες, οι οποίες μετακινούνται συνεχώς από το πάνω μέρος της οθόνης στο κάτω. Ο καταρράκτης είναι στην πραγματικότητα τρεις μακρόστενες εικόνες που εμφανίζονται κάθετα στο πάνω μέρος της σκηνής και εκτός του ορατού πλαισίου. Η εικόνες κινούνται με ρυθμό 2-3-4 pixels αντιστοίχως σε κάθε frame και αυτό καθορίζει την ταχύτητά τους. Το framerate είναι σταθερό στα 60fps (frames per second). Μόλις η κάθε μία από τις τρεις εικόνες φτάσει σε ένα προκαθορισμένο μέρος της οθόνης (εκτός του ορατού πλαισίου) μεταφέρεται στην αρχική της θέση και ο κύκλος επαναλαμβάνεται.

Το δεύτερο γραφικό στοιχείο είναι τυχαίες νότες που εμφανίζονται σε τυχαίο μέρος της οθόνης με fade in και εξαφανίζονται με fade out. Ο ρυθμός με τον οποίο γίνονται οι μεταβάσεις είναι προκαθορισμένος.

3.2. Dictée

Εικ. 2.3. Dictée.

Το ζητούμενο σε αυτήν την άσκηση είναι να γράψουμε τη μελωδία που μας παίζει ο υπολογιστής. Κάθε φορά που ζητάμε να μας παιχτεί μια καινούρια άσκηση, δημιουργείται μια τυχαία μελωδία σύμφωνα με τις προδιαγραφές που έχουμε καθορίσει στις ρυθμίσεις. Έπειτα, τη γράφουμε στο πεντάγραμμο και στο τέλος ο υπολογιστής μας λέει εάν τη βρήκαμε και μας δείχνει την σωστή για να τη συγκρίνουμε με αυτή που γράψαμε.

Τα πρώτα πράγματα που τραβάνε την προσοχή μας σε αυτή τη σελίδα είναι το πεντάγραμμο και το πιάνο. Το πεντάγραμμο δημιουργήθηκε στο MS Paint. Έπρεπε να

προσεχθεί οι γραμμές να ισαπέχουν μεταξύ τους ώστε να διευκολυνθεί η μετατόπιση νοτών στο πρόγραμμα. Παραδείγματος χάριν εάν έχουμε μία νότα στο Σολ και θέλουμε να την πάμε στο Λα ακριβώς από πάνω το μόνο που έχουμε να κάνουμε είναι να μετατοπίσουμε την νότα κατά 5 pixels στον άξονα Y και αυτό ισχύει για όλες τις νότες. Τελευταίο προστέθηκε το κλειδί του Σόλ στο πεντάγραμμο. Όταν μπαίνει ο κέρσορας μέσα στο πεντάγραμμο παίρνει τη μορφή της νότας που είναι επιλεγμένη. Σαν προεπιλογή είναι η νότα τετάρτου και αλλάζει κάνοντας δεξί κλικ. Οι διαθέσιμες αξίες είναι τα ολόκληρο, μισό, τέταρτο, όγδοο και δέκατο έκτο. Αν είναι επιλεγμένη η παύση ο κέρσορας παίρνει τη μορφή παύσης με την επιλεγμένη αξία. Υπάρχει ακόμα μια γόμα που χρησιμεύει για να σβήνουμε την τελευταία νότα που γράψαμε. Η εικόνα της γόμας πάρθηκε από το MS Paint όπου υπάρχει σαν επιλογή. Τέλος, αναλόγως με το ύψος που έχουμε τον κέρσορα στο πεντάγραμμο, εμφανίζεται στην πάνω αριστερά γωνία το όνομα της νότας που θα ακουστεί-γραφτεί. Αυτό γίνεται με τη βοήθεια ενός cast member του οποίου κάθε φορά αλλάζουμε το περιεχόμενο. Το πεντάγραμμο είναι το μέσο με το οποίο γράφουμε τη μελωδία που έχουμε ακούσει. Αναλόγως με το ύψος στο οποίο θα πατήσουμε, την αξία που είναι επιλεγμένη και εάν έχουμε επιλέξει κάποια αλλοίωση, γράφεται η συγκεκριμένη νότα με οποιαδήποτε τροποποίηση έχει υποστεί (αλλοίωση, τρίηχο, παρεστιγμένο, παύση). Όταν ολοκληρώνεται ένα μέτρο μπαίνει αυτόματα η διαχωριστική γραμμή. Εάν οι αξίες που έχουμε βάλει αθροίζουν παραπάνω από ένα μέτρο, μπαίνει το διαχωριστικό αλλά κοκκινίζει όλο το μέτρο για να μας δείξει ότι έχει γίνει λάθος. Στα όγδοα και στα δέκατα έκτα ενώνονται οι ουρές όπου χρειάζεται, όπως και στα τρίηχα. Αν τα τρίηχα δεν είναι σωστά γιατί έχουμε χρησιμοποιήσει διαφορετικές αξίες κοκκινίζει το ενδεικτικό τρίηχου για να μας δείξει το λάθος μας. Η απόσταση μεταξύ των νοτών είναι μεταβαλλόμενη και ανάλογη με το πόσες νότες έχουμε γράψει. Αν γραφτούν παραπάνω νότες από όσες χωράνε στην οθόνη με τη μικρότερη δυνατή απόσταση μεταξύ τους συνεχίζουν να γράφονται εκτός οθόνης. Ένα ακόμα χαρακτηριστικό είναι ότι αν παραδείγματος χάριν έχουμε Φα δίεση στον οπλισμό και γράψουμε Φα αναίρεση και μετά απενεργοποιήσουμε τις αλλοιώσεις και γράψουμε Φα θα μπει και μία δίεση αυτόματα. Στις επόμενες νότες δεν θα εμφανιστεί αλλοίωση και θα είναι Φα δίεση. Η γόμα σβήνει την τελευταία νότα που γράψαμε μαζί με οποιαδήποτε

τροποποίηση, καθώς και διαχωριστικό μέτρου, εάν ήταν η τελευταία νότα ενός μέτρου και αλλάζει την απόσταση μεταξύ των νοτών. Ο οπλισμός και το μέτρο μπαίνουν αυτόματα κάθε φορά που πατάμε το κουμπί «Καινούρια Ερώτηση». Όταν γράφουμε μια νότα, αν δεν είναι επιλεγμένη η παύση, την ακούμε παιγμένη από το όργανο που έχουμε επιλέξει. Προεπιλογή είναι το πιάνο. Η χαμηλότερη νότα που μπορεί να παιχτεί είναι η Σι ύφεση και η ψηλότερη η Ντο δίεση δύο οκτάβες πιο ψηλά.

Το πιάνο σαν εικόνα βρέθηκε στο ίντερνετ και τροποποιήθηκε στο Adobe Photoshop. Στη συνέχεια κατασκευάστηκαν 25 σχήματα γκριζού χρώματος στο MS Paint, το κάθε ένα κομμένο στο σχήμα του πλήκτρου που αντιστοιχεί. Κάθε φορά που ο χρήστης πατάει ένα πλήκτρο, εμφανίζεται από πάνω του το αντίστοιχο σχήμα, δίνοντας έτσι την αίσθηση του βάθους. Επίσης, όπως και στο πεντάγραμμο, εμφανίζεται στην πάνω αριστερά γωνία το όνομα της νότας που θα παιχτεί εάν πατήσουμε το πλήκτρο. Όταν πατήσουμε κάποια νότα, εκτός από το ηχητικό αποτέλεσμα, το οποίο εξαρτάται από το όργανο που είναι επιλεγμένο, εμφανίζεται η συγκεκριμένη νότα πάνω στο πεντάγραμμο με οποιαδήποτε αλλοίωση μπορεί να χρειάζεται. Παραδείγματος χάριν εάν έχουμε οπλισμό τρεις διέσεις και πατήσουμε το πλήκτρο Φα φυσικό θα εμφανιστεί και μια αναίρεση. Όταν αφήσουμε το πλήκτρο η νότα θα εξαφανιστεί. Πρέπει να πούμε ότι το πιάνο είναι καθαρά βοηθητικό και δεν αποτελεί αναπόσπαστο μέρος της άσκησης.

Το βέλος στο δεξί μέσο της οθόνης επιλέγει το όργανο που θα χρησιμοποιείται για να παίζονται οι μελωδίες, οι κλίμακες και οι νότες. Κάθε φορά που το πατάμε μετακινείται δεξιόστροφα και επιλέγεται το όργανο στο οποίο το όνομα δείχνει. Οι επιλογές είναι πιάνο, τρομπέτα, βιολί και φλάουτο.

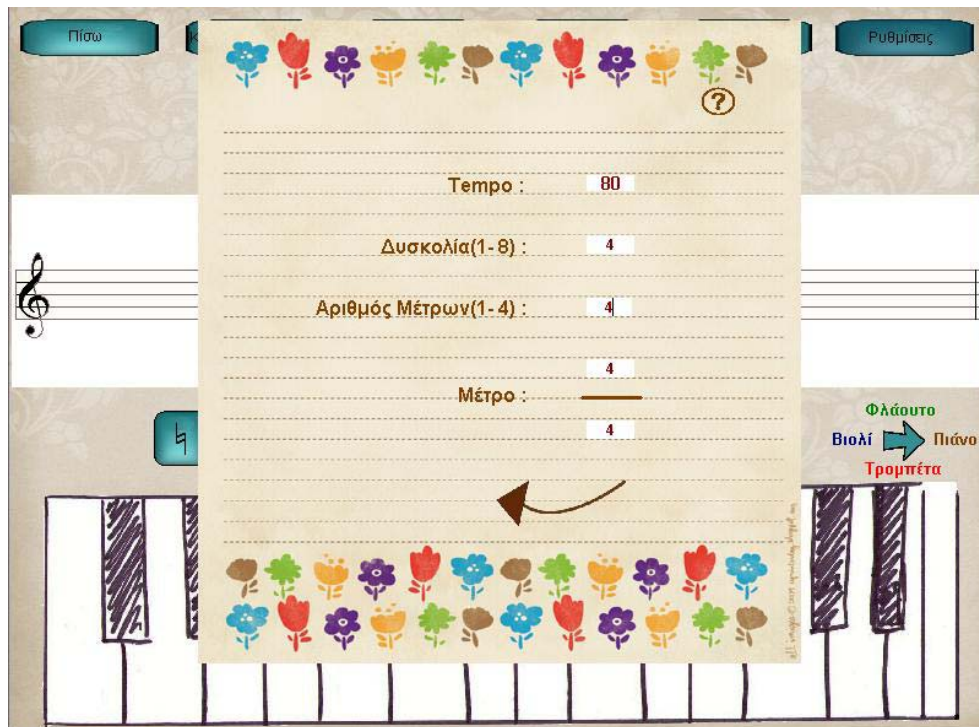
Στη σελίδα αυτή υπάρχουν δώδεκα κουμπιά: έξι αποτελούν το μενού και έξι τροποποιούν την νότα που θα γράψουμε. Οι τροποποιήσεις είναι οι εξής :

- Αναίρεση. Ανάλογα με τον οπλισμό, βάζει αναίρεση στη νότα.
- Δίεση. Ανάλογα με τον οπλισμό, βάζει δίεση στη νότα.
- Ύφεση. Ανάλογα με τον οπλισμό, βάζει ύφεση στη νότα.
- Παύση. Οι νότες γράφονται σαν παύσεις της επιλεγμένης αξίας.
- Παρεστιγμένο. Κάνει τη νότα παρεστιγμένη.

- Τρίηχο. Κάνει τη νότα τρίηχο. Τα τρίηχα γράφονται σε τριπλές αλλιώς βγαίνει λάθος.

Όταν πατάμε μία αλλοίωση από-επιλέγεται οποιαδήποτε άλλη αλλοίωση ήταν επιλεγμένη. Για να από-επιλεγθεί οποιαδήποτε επιλογή αρκεί να ξαναπατήσουμε το αντίστοιχο κουμπί. Τα κουμπιά του μενού είναι τα εξής :

- Πίσω. Μας μεταφέρει στην αρχική σελίδα.
- Καινούρια ερώτηση. Αυτό είναι το πιο σημαντικό κουμπί στη σελίδα. Όταν το πατάμε συμβαίνουν πολλά πράγματα. Κατ' αρχήν οτιδήποτε είναι γραμμένο στο πεντάγραμμο σβήνεται. Έπειτα, αναλόγως με τη δυσκολία που έχουμε επιλέξει επιλέγεται τυχαία η κλίμακα της επόμενης άσκησης και γράφεται στο πεντάγραμμο ο οπλισμός της. Επίσης γράφεται το μέτρο που έχουμε επιλέξει. Προεπιλογή είναι τα 4 / 4. Αμέσως μετά μπαίνει σε λειτουργία ο αλγόριθμος δημιουργίας μελωδίας. Μόλις η μελωδία ολοκληρωθεί, ξεκινάει και παίζει η κλίμακα που έχει επιλεγθεί από την τονική μέχρι την οκτάβα της τονικής. Στο τέλος αρχίζει να παίζει η νέο-δημιουργημένη μελωδία από το όργανο που έχουμε επιλέξει ταυτόχρονα με τον μετρονόμο.
 - Αναπαραγωγή ερώτησης. Παίζει την τελευταία μελωδία που δημιούργησε ο υπολογιστής ταυτόχρονα με το μετρονόμο.
 - Αναπαραγωγή Απάντησης. Παίζει τη μελωδία που έχουμε γράψει στο πεντάγραμμο ταυτόχρονα με το μετρονόμο
 - Απάντηση. Μεταφέρει το πεντάγραμμο χαμηλότερα κοντά στο πιάνο και εμφανίζεται από πάνω ένα καινούριο πεντάγραμμο με τη σωστή μελωδία ώστε να μπορέσουμε να τη συγκρίνουμε με αυτή που έχουμε γράψει εμείς. Επίσης βγάζει την ένδειξη «Σωστό» με πράσινα γράμματα αν η μελωδία που γράψαμε είναι ίδια με αυτή του υπολογιστή και «Λάθος» με κόκκινα γράμματα αν έχει λάθη.
 - Ρυθμίσεις. Εμφανίζει την καρτέλα των ρυθμίσεων. Εδώ είναι οι επιλογές που καθορίζουν τα χαρακτηριστικά της μελωδίας που θα δημιουργήσει ο υπολογιστής. Η επιλογή γίνεται κάνοντας κλικ με τον κέρσορα στην ρύθμιση που θέλουμε, εισάγοντας ένα νούμερο από το πληκτρολόγιο και πατώντας enter. Σε αυτή την άσκηση οι ρυθμίσεις είναι οι εξής :



Εικ 2.4. Ρυθμίσεις της άσκησης Dictée.

- Tempo. Ορίζει το tempo με στο οποίο θα παίζονται οι μελωδίες και οι κλίμακες και δίνεται σε b.p.m. (beats per minute). Μπορεί να πάρει τιμές από 1 έως 999.
- Δυσκολία(1-8). Από εδώ ορίζουμε τη δυσκολία της άσκησης. Η δυσκολία καθορίζει αν η άσκηση θα έχει παύσεις, τρίχα, παρεστιγμένα, αλλοιώσεις καθώς και το μέγιστο διάστημα που μπορεί να έχει η άσκηση. Για να υπάρχουν παρεστιγμένα πρέπει να είναι από 2 και πάνω. Για να υπάρχουν τρίχα από 3 και πάνω. Σε δυσκολία 1-2 υπάρχουν μόνο διαστήματα δευτέρας και ταυτοφωνίες. Στο 3 υπάρχουν και τρίτες, στο 4 τέταρτες, στο 5 πέμπτες, στο 6 έκτες, στο 7 έβδομες και στο 8 όγδοες. Η κλίμακα του κομματιού είναι τυχαία και αντίστοιχη της δυσκολίας. Δηλαδή σε δυσκολία 1 χρησιμοποιούνται μόνο η Ντό Μείζονα και η σχετική της (Λά Ελάσσονα). Στο 2 η κλίμακα μπορεί να περιέχει και μια δίεση ή μια ύφεση, στο 3 δύο διέσεις η δυο υφέσεις κ.ο.κ. ως το 8 που χρησιμοποιεί όλες τις κλίμακες ελάσσονες και μείζονες(μέχρι 7 διέσεις ή υφέσεις).

- Αριθμός μέτρων(1-4). Καθορίζει αν η άσκηση θα έχει ένα, δύο, τρία ή τέσσερα μέτρα.
- Μέτρο. Στην πραγματικότητα δύο επιλογές. Η πάνω ορίζει τον αριθμό των χτύπων σε κάθε μέτρο και η κάτω την αξία του κάθε χτύπου. Οι τιμές που μπορούν να πάρουν είναι συγκεκριμένες. Αν η τιμή που δώσουμε δεν υποστηρίζεται γυρνάει στην προηγούμενη τιμή. Ο πάνω αριθμός μπορεί να πάρει τιμές από 1 έως 12 και ο κάτω τις τιμές 1, 2, 4, 8 και 16 για ολόκληρα, μισά, τέταρτα, όγδοα και δέκατα έκτα αντιστοίχως.
- Ερωτηματικό. Όταν πατηθεί, εμφανίζεται ένα κείμενο με πληροφορίες για την κάθε ρύθμιση. Αν ξανά-πατηθεί εξαφανίζεται το κείμενο και έρχονται πάλι στην οθόνη οι επιλογές.
- Βέλος. Διώχνει την καρτέλα των ρυθμίσεων. Μόλις γίνει αυτό ό,τι είναι γραμμένο στο πεντάγραμμο σβήνεται και η προηγούμενη άσκηση διαγράφεται. Για να συνεχίσουμε πρέπει να πατήσουμε τα κουμπιά «Καινούρια Ερώτηση» ή «Πίσω».

3.3. Αναγνώριση Συγχορδιών

Εικ.2.5. Αναγνώριση συγχορδιών.

Σε αυτή την άσκηση, ο υπολογιστής παίζει συγχορδίες και ο χρήστης προσπαθεί να τις αναγνωρίσει με τη χρήση πολλαπλής επιλογής. Όπως και στην προηγούμενη άσκηση (Dictée) υπάρχει το πεντάγραμμο με τη διαφορά ότι εδώ η λειτουργία του είναι περιορισμένη στο να μας εμφανίζει τις νότες της σωστής συγχορδίας όταν επιλέξουμε κάποια απάντηση. Το πιάνο εξακολουθεί να είναι βοηθητικό και προαιρετικό και η επιλογή οργάνου είναι ακριβώς η ίδια, όπως σε όλες τις ασκήσεις.

Τα κουμπιά του μενού είναι λιγότερα σε σχέση με την προηγούμενη άσκηση. Υπάρχουν πάλι τα κουμπιά «Πίσω», «Καινούρια Ερώτηση» και «Αναπαραγωγή Ερώτησης» και εκτελούν τις ίδιες λειτουργίες. Η ερώτηση αυτή τη φορά δεν είναι κάποια μελωδία αλλά μία τετράφωνη συγχορδία. Το κουμπί των ρυθμίσεων έχει αντικατασταθεί

από τη μοναδική επιλογή που υπάρχει σε αυτήν την άσκηση : «Δυσκολία (1-4)». Πάλι η εισαγωγή γίνεται με το πληκτρολόγιο και πατώντας enter.

Η απάντηση επιλέγεται από τα κουμπιά απάντησης που είναι ανάμεσα στο πεντάγραμμο και το πιάνο, τα οποία εμφανίζονται και εξαφανίζονται ανάλογα με τη δυσκολία που έχουμε επιλέξει. Σε δυσκολία 1 έχουμε διαθέσιμα τα κουμπιά «Μείζονα»και «Ελάσσονα». Στο 2 έχουμε επιπλέον το κουμπί «Ελαττωμένη», στο 3 το «Αυξημένη»και στο 4 το «Μεθ' Εβδόμης». Τα κουμπιά εμφανίζονται-εξαφανίζονται μόλις πατηθεί το «Καινούρια Ερώτηση». Αντιστοίχως, η ερώτηση που δίνεται επιλέγεται αναλόγως με τη δυσκολία. Όταν πατηθεί κάποιο από τα κουμπιά απάντησης εμφανίζονται στο πεντάγραμμο οι νότες και το όνομα της σωστής συγχορδίας, καθώς και το όνομα της συγχορδίας που επιλέξαμε εμείς, με πράσινο χρώμα αν ήταν σωστή και με κόκκινο αν ήταν λάθος.

Ένα ακόμα χαρακτηριστικό αυτής της άσκησης είναι η επιλογή να παιχτεί η συγχορδία σε αρμονική ή μελωδική μορφή, το οποίο επιτυγχάνεται μέσω των δύο κουμπιών που βρίσκονται ακριβώς πάνω από το πεντάγραμμο. Όταν είναι επιλεγμένη η αρμονική μορφή (προεπιλογή) οι νότες της παίζονται ταυτόχρονα ενώ στην μελωδική μορφή παίζονται η μία μετά την άλλη.

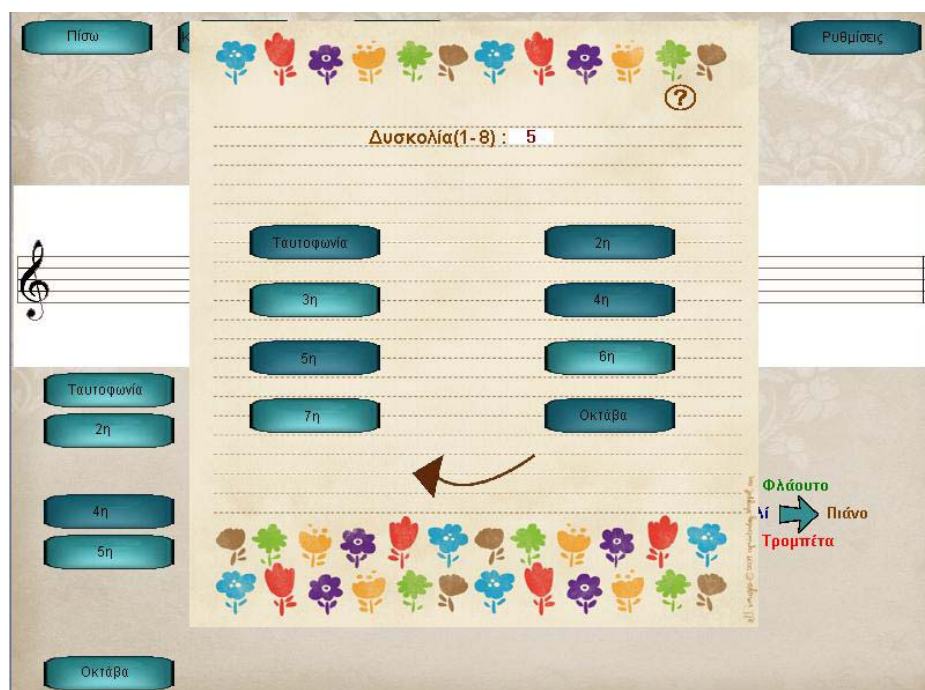
3.4. Αναγνώριση Διαστημάτων

Εικ. 2.6. Αναγνώριση διαστημάτων.

Στην τρίτη άσκηση του προγράμματος ο υπολογιστής παίζει μουσικά διαστήματα και ο χρήστης πρέπει να τα αναγνωρίσει με τη χρήση πολλαπλής επιλογής. Υπάρχει πάλι το πεντάγραμμο και λειτουργεί βοηθητικά, εμφανίζοντας τις νότες του σωστού διαστήματος, αφού κάνουμε μία επιλογή. Η επιλογή οργάνων είναι ίδια με τις προηγούμενες ασκήσεις.

Υπάρχουν πάλι τα κουμπιά «Πίσω», «Καινούρια Ερώτηση», «Αναπαραγωγή Ερώτησης» και «Ρυθμίσεις» και εκτελούν τις ίδιες λειτουργίες. Η καρτέλα των ρυθμίσεων έχει την επιλογή «Δυσκολία(1-8)» η οποία επηρεάζει την κλίμακα στην οποία θα παιχτεί η άσκηση. Στο 1 θα παιχτεί διάστημα από την Ντό μείζονα ή τη σχετική της (Λά ελάσσονα). Στο 2 η κλίμακα μπορεί να έχει και μία δίεση ή ύφεση, στο 3 δύο διέσεις ή υφέσεις και ούτω καθεξής. Υπάρχουν ακόμα οχτώ κουμπιά που αντιστοιχούν σε οχτώ μουσικά διαστήματα : Ταυτοφωνία, 2^η, 3^η, 4^η, 5^η, 6^η, 7^η και Οκτάβα. Όποιο από αυτά

επιλέξουμε μπορεί να χρησιμοποιηθεί στην άσκηση. Το κουμπί με το ερωτηματικό εμφανίζει πληροφορίες για τις ρυθμίσεις και το βέλος διώχνει την καρτέλα των ρυθμίσεων. Φεύγοντας από τις ρυθμίσεις, η προηγούμενη άσκηση διαγράφεται. Για να συνεχίσουμε πρέπει να πατήσουμε τα κουμπιά «Καινούρια Ερώτηση» ή «Πίσω».



Εικ. 2.7. Ρυθμίσεις της άσκησης Αναγνώριση διαστημάτων.

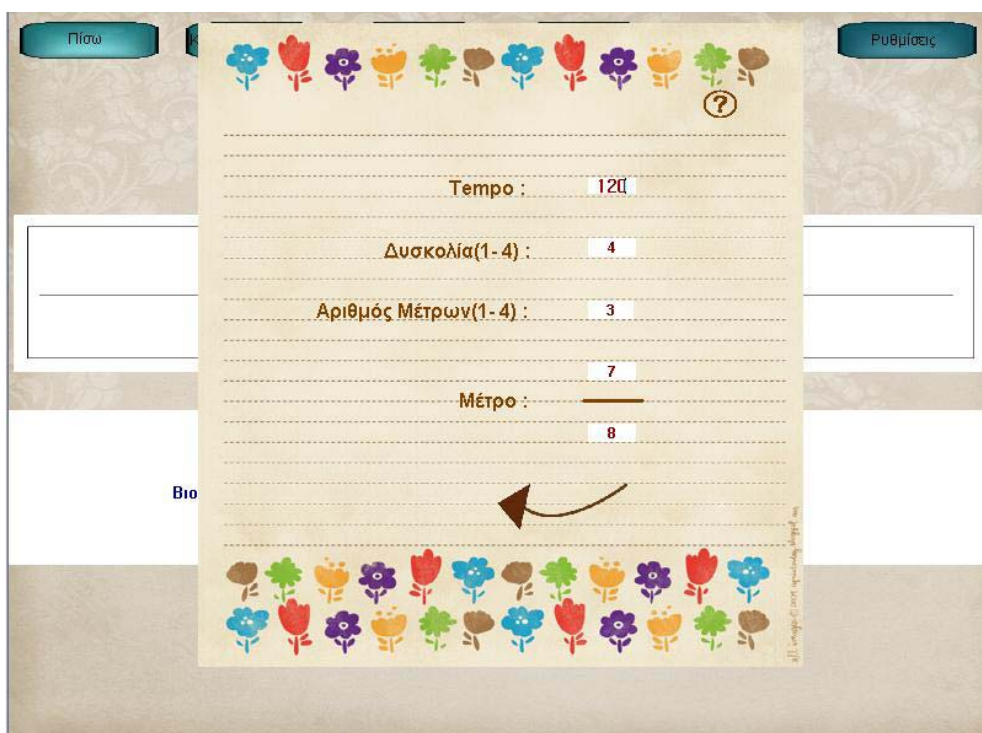
Ανάλογα με τα διαστήματα που επιλέξαμε στις ρυθμίσεις, θα μας εμφανιστούν κάτω από το πεντάγραμμο τα αντίστοιχα κουμπιά. Η προεπιλογή είναι «Ταυτοφωνία» και «2^η». Εάν επιλέξουμε κάποιο διάστημα εκτός των «Ταυτοφωνία» και «Οκτάβα» εμφανίζεται ένα δεύτερο σεντ κουμπιών που αφορά στο γένος του διαστήματος. Στο «2^η» υπάρχουν τα «Μικρή», «Μεγάλη» και «Τριημιτόνιο». Στα «3^η», «6^η» και «7^η» τα «Μικρή» και «Μεγάλη». Στα «Τέταρτη» και «Πέμπτη» υπάρχουν τα «Ελαττωμένη», «Καθαρή» και «Αυξημένη». Μόλις επιλέξουμε ένα διάστημα, εμφανίζονται οι σωστές νότες στο πεντάγραμμο, το όνομα του σωστού διαστήματος και, εάν επιλέξαμε σωστά, η λέξη «Σωστό» με πράσινα γράμματα ή «Λάθος» με κόκκινα γράμματα αν επιλέξαμε λάθος.

3.5. Ρυθμική Αγωγή

Εικ. 2.8. Ρυθμική αγωγή.

Η τελευταία άσκηση του προγράμματος. Είναι διαφορετική από τις υπόλοιπες ως προς το ότι δεν χρειάζεται να γράψει ο χρήστης τον ρυθμό που του δίνεται αλλά να τον παίζει με τη βοήθεια του πληκτρολογίου. Το πεντάγραμμα έχει αντικατασταθεί από ένα μονόγραμμα. Με το που πατάμε «Καινούρια Άσκηση» ο ρυθμός εμφανίζεται στην οθόνη και το πρόγραμμα παροτρύνει τον χρήστη να πατήσει ένα κουμπί και μπαίνει σε κατάσταση αναμονής. Η άσκηση ξεκινάει με το πάτημα οποιουδήποτε πλήκτρου. Ο μετρονόμος μετράει ένα μέτρο προετοιμασίας, στο τέλος του οποίου ο χρήστης πρέπει να παίζει το ρυθμό που βλέπει με το πληκτρολόγιο. Για βοήθεια, υπάρχει και οπτική ένδειξη του μετρονόμου στη μορφή ενός σκύλου που κουνάει την ουρά του ρυθμικά. Στον πρώτο χτύπο κάθε μέτρου υπάρχει μια επιπρόσθετη οπτική ένδειξη. Η άσκηση τελειώνει όταν ο χρήστης πατήσει τόσες φορές το πλήκτρο όσες νότες υπάρχουν στην οθόνη. Σε αυτό το σημείο ο υπολογιστής θα δώσει μία εκτίμηση για την επιτυχία της εκτέλεσης με πράσινο,

κόκκινο και πορτοκαλί χρώμα για καλώς, κακώς ή μέτρια αντιστοίχως. Επιπλέον, κάθε φορά που ο χρήστης «παίζει» μια νότα, εμφανίζεται από κάτω της ένα βέλος που δείχνει την επιτυχία στην κάθε νότα ξεχωριστά. Εάν η νότα παιχτεί πρόωρα, το βέλος θα είναι στα αριστερά της ενώ αν παιχτεί καθυστερημένα θα είναι στα δεξιά της. Υπάρχουν τρεις χρωματικές διαβαθμίσεις - πράσινο, κόκκινο και πορτοκαλί – και επηρεάζουν τη μετατόπιση του βέλους. Παραδείγματος χάριν, το βέλος μίας νότας που παίχτηκε 210ms πιο νωρίς απ’ ότι έπρεπε μπορεί να είναι πιο κοντά στη νότα από το βέλος μίας νότας που παίχτηκε 70ms πιο νωρίς γιατί ανήκουν σε διαφορετική κατηγορία και αυτό φαίνεται από το χρώμα. Μόλις ολοκληρωθεί η άσκηση τουλάχιστον μία φορά, ο χρήστης μπορεί να ακούσει το σωστό ρυθμό και, αν θέλει, να ξαναπροσπαθήσει.



Εικ. 2.9. Ρυθμίσεις της άσκηση Ρυθμική αγωγή.

Τα κουμπιά που αποτελούν το μενού σε αυτήν την άσκηση είναι τα εξής :

- Πίσω :
- Καινούρια ερώτηση :

- Αναπαραγωγή Ερώτησης : Αναπαραγάγει το σωστό ρυθμό. Εμφανίζεται μόνο αφού ολοκληρωθεί η άσκηση τουλάχιστον μία φορά.
- Επόμενη Προσπάθεια : Ξεκινάει την ίδια άσκηση από την αρχή.
- Ρυθμίσεις

Οι επιλογές στις ρυθμίσεις είναι ίδιες με αυτές του Dictée, εκτός από το επίπεδο δυσκολίας που παίρνει τιμές από 1 έως 4. Το κουμπί με το ερωτηματικό εμφανίζει πληροφορίες για τις ρυθμίσεις και το βέλος διώχνει την καρτέλα των ρυθμίσεων. Το βέλος επιλογής οργάνου παραμένει ίδιο όπως και στις άλλες ασκήσεις.

3.6. Ήχοι

Οι ήχοι που χρησιμοποιήθηκαν στην εργασία είναι σε μορφή WAV μονοφωνικοί, 16 bit και με συχνότητα δειγματοληψίας 11025Hz. Η ποιότητα επιλέχθηκε έχοντας υπόψιν ότι οι ήχοι πρέπει να παίζονται και να εναλλάσσονται με μεγάλη ταχύτητα και, έπειτα από πολλές δοκιμές, διαπιστώθηκε ότι αυτή είναι η καλύτερη αναλογία ποιότητας-απαιτήσεων. Οι ήχοι των οργάνων είναι midi ήχοι ηχογραφημένοι στο Steinber Nuendo 3 και επεξεργασμένοι με το Wavelab 5, όπως και οι ήχοι του μετρονόμου. Το αρχείο «Knob Change.wav» πάρθηκε από τους ήχους των Microsoft Windows. Δημιουργήθηκε επίσης ένα αρχείο με το όνομα «Silence.Wav», το οποίο είναι 10 δευτερόλεπτα σιωπής και παίζεται κυρίως όταν υπάρχει παύση. Για την αναπαραγωγή των ήχων χρησιμοποιήθηκαν οι εντολές της Lingo «queue» και «play».

ΕΠΙΛΟΓΟΣ

Η καλλιέργεια της μουσικής ακοής επιτυγχάνεται με τη συνεργασία δύο ατόμων - τον εκπαιδευτή και τον εκπαιδευόμενο. Η χρήση λογισμικού για την δημιουργία μελωδιών, συγχορδιών, διαστημάτων και ρυθμικών σχημάτων άγνωστων στο χρήστη καθιστά δυνατή την εξάσκηση σε αυτά εντελώς ανεξάρτητα και σε ρυθμό που μπορεί να καθορίσει ο καθένας μόνος του. Το παρόν πρόγραμμα το πετυχαίνει αυτό με τρόπο ευχάριστο και, παρά τις δυσκολίες που αντιμετωπίστηκαν κατά τη δημιουργία του, με ακρίβεια.

Οι τέσσερις ασκήσεις συνεργάζονται με τέτοιο τρόπο ώστε όταν ο χρήστης έχει εκπαιδευτεί αρκετά σε όλες, έχει συνολική κατανόηση της μελωδίας και του ρυθμού. Αυτό θα φανεί στην πρώτη άσκηση, η οποία είναι σαν περίληψη των επιμέρους κομματιών.

Στην δεύτερη άσκηση ο υπολογιστής παίζει συγχορδίες και ο χρήστης προσπαθεί να τις αναγνωρίσει με τη χρήση πολλαπλής επιλογής. Επιτυγχάνεται με αυτόν τον τρόπο η εξάσκηση στην ανάλυση πολυφωνικών ήχων και τη διάσπασή τους στα ξεχωριστά στοιχεία τους.

Η τρίτη άσκηση θα μπορούσε να παρομοιαστεί με εξειδίκευση πάνω στην άσκηση του Dictée. Το αποτέλεσμά της περιλαμβάνεται εξ ολοκλήρου στην πρώτη άσκηση με τη διαφορά ότι εδώ έχουμε απόλυτο έλεγχο στα διαστήματα πάνω στα οποία θα εξασκηθούμε.

Η τέταρτη και τελευταία άσκηση είναι ένα «παιχνίδι» πάνω στους ρυθμούς. Τα λάθη μας εμφανίζονται με ακρίβεια millisecond και έχουμε απεριόριστες προσπάθειες για να πετύχουμε το καλύτερο δυνατό αποτέλεσμα. Η άσκηση αυτή δεν ασχολείται καθόλου με το τονικό ύψος αφήνοντας το χρήστη να επικεντρώσει το ενδιαφέρον του στην κατανόηση του κάθε ρυθμικού συμβόλου, ώστε, όταν τα χρειαστεί, να τα χρησιμοποιήσει κατάλληλα.

Συμπερασματικά, το παρόν λογισμικό έχει τη δυνατότητα να εκπαιδεύσει σφαιρικά έναν ενδιαφερόμενο και να τον προετοιμάσει για οποιαδήποτε κατάσταση στην οποία μπορεί να επικαλεστεί την μουσική ακοή του για να αναλύσει τα ακούσματά του.

ΠΑΡΑΡΤΗΜΑΤΑ

A) Οι μεταβλητές που χρησιμοποιήθηκαν με τις επεξηγήσεις τους :

<code>global n loops)</code>	<code>--Joker (used in</code>
<code>global n2 loops)</code>	<code>--Joker (used in</code>
<code>global Vvolume</code>	<code>--Volume</code>
<code>global Vnote played</code>	<code>--Name of note being</code>
<code>global Vinstrument being used</code>	<code>--Name of instrument</code>
<code>global Vrotate rotation of instrument knob</code>	<code>--Int used for the</code>
<code>global VinstrumentINT change of instrument</code>	<code>--Int used for the</code>
<code>global VcursorINT cursor being used</code>	<code>--Int used for the</code>
<code>global VcursorX X(left-right)</code>	<code>--Cursor location</code>
<code>global VcursorY Y(up-down)</code>	<code>--Cursor location</code>
<code>global Vcursorname being used</code>	<code>--Name of cursor</code>
<code>global VcursonOn 0 if cursor is off</code>	<code>--1 if cursor is on, 0 if cursor is off</code>
<code>global Vshadenumbr blink</code>	<code>--number of shade to</code>

```

global Vscriptedspritechannel          --Used for counting
channels for written notes

global Vnoteused                       --The number of the
sprite being written.First note used is 2 and not 1

global VnoteX                          --[] Location of note
written (X)

global VnoteY                          --[] Location of note
written (Y)

global Vpositionofnoteused            --The X position of
the note being modified

global Vapostasinoton                 --Distance between
notes.Changes according to how many notes are currently
being used

global VpliktroON                    --The number of the
note 1 being C1,2 C#1..and 15 C3.(0 = C1flat)

global Vnoumerpliktrou               --
F=1,C=2,G=3,D=4,A=5,E=6,B=7,F#=-1,C#=-2,G#=-3,D#=-4,A#=-
5,Eflat=-6,Bflat=-7

global Voplismos                     --Key signature.-7=1
flat,-6=2 flat etc,0=nothing,1=1sharp,2=2sharp etc

global Vzografianotas               --The name of the
cast member being used for scriptedsprite

global Vxrisimopoioumenasprites     --[] The number of
sprites being used for one note(sharps,flats etc)

global VparestigmenoOn              --1 if parestigmeno
is on,0 if parestigmeno is off

global Valliosion                   --1 if aneresi key is
pressed,2 if sharp,3 if flat,0 if none pressed

global Vpausi                        --1 if the Rest key
is on,0 if off

global Vpausicarry                   -- = Vpausi.Used in
correct key press

```

```

global Vpausiname          --" pausi" if Vpausi
= 1,"" if Vpausi = 0

global VtriixoON          --1 if triixo key is
pressed,0 if not

global Vsystema          --The number of the
system(system = a note with all its intonations,voithitikes
grammes ktl).Like Vnoteused

global Vaxianotas        --[] The rythmic
value of the note

global Vpcaxianotas      --[] The rythmic
value of the note in question playback

global Vdiaxoristikometrou --[] Sum of
Vaxianotas to know when to use diaxoristikometrou

global Vloopsystema      --Used in note
distance loop

global Vloopspritechannel --Used in note
distance loop

global Vloopsprite       --Used in note
distance loop

global Vloopnoteused     --Used in note
distance loop

global Vtypossprite      --[] The type of
every Vnoteused : 1 = Full note , 2 = Half note , 3 =
Quarter note , 4 = Eighth note , 5 = 16th note , 6 = Full
rest , 7 = Half rest , 8 = Quarter rest , 9 = Eighth rest ,
10 = 16th rest  11 = Aneresi , 12 = Diesi , 13 = Yfesi ,
14 = Mia voithitiki , 15 = Dyo voithitikes , 16 =
Parestigmeno , 17 = Diaxoristiko metrou , 18 = Triixo

global VnoteYmemory      --Used to determine
NoteY in Intonations after note

global Vnoteinsystem     --[] The Vnoteused of
the note in a system

global Vymax             --Used to determine
the Y of note beams

```



```
global Vymax2                --Used to determine
the Y of triplet beams

global Vchannelbeam1         --The channel number
of note beams for 8ths(500-550)

global Vchannelbeam2         --The channel number
of note beams for 16ths(550+)

global Vchannelbeam3         --The channel number
of note beams for triplets

global Vloopbeams           --Used in Vymax loop
in beam creation

global Vloopbeams2          --Used in Vymax loop
in 16th beam creation

global Vloopbeams3          --Used in triplet
creation

global Vstartofbeam1        --The beginning of a
beam system for 8ths

global Vstartofbeam2        --The beginning of a
beam system for 16ths

global Vstartofbeam3        --The beginning of a
beam system for triplets

global Vychange              --Used to change 16th
beam heights

global Vtempo                --Master Tempo

global Vtonikotita           --Pitch in playback
answer

global Vpctonikotita        --Pitch in playback
question

global VplaybackA           --Playback answer 1 =
On ,0 = Off

global VplaybackQ           --Playback question 1
= On ,0 = off

global Vlooptriixo          --Triplet counter
```

```

global Vtriixosystem          --Used in triplet-
8th-16th systems

global Vdrawoplismos         --1 if Key signature
is to be drawn,0 if not

global Vscriptedchanneloplismos --The sprite channels
of key signature. 700+

global Vmetro1               --Time signature
numerator

global Vmetro2               --Time signature
denominator

global VpanokoubiaON        --Used in button
mouseleave

global Vlathosdiaxoristikos --[] Used for
changing color of sprites

global Vdyskolia             --Difficulty of
exersice.1 = no key.2-8 = 1 to 7 sharps/flats

global Vdimiourgia          --1 if a new question
is to be generated

global Vgenos                --If its a major or
minor key

global Vnotenumberloop      --Used in question
generation note number loop

global Vpcloop               --Used in question
generation master loop

global Vpcdiaxoristikometrou --[] Sum of Vpcaxia
to know when to use diaxoristikometrou

global Vpcmetra              --The number of bars
created by new question button

global Vcarry                --Used to carry
variables around loops

global Vpcplayscale         --1 when scale is
being played

```

```

global Vpcscalenumber          --[] Variation of
Vpcnotenumber used in scale loop

global Vmetro2clickvalue      --The value of metro
2 for metronome

global Vclickcounter          --The count for
metronome

global Vclickend              --1 when Vclicktimes
is about to change

global Vclicktimes            --How many bars the
metronome will count

global Vapotelesma            --
1,0,meizona,ellasona,auksimeni,elattomeni,methevdomis

global Vapadisi                --If the programme is
in "apadisi" mode

global Vpcnotacarry            --[] Carries the note
as it was created so it can be written correctly when in a
scale with flats

global Vtotalmetra            --The number of bars
in a question

global Vpcnotenumbercarry      --Carries
Vpcnotenumber[3] in chords so Vapadisi can be sort out

global Vintitiate              --Becomes 1 when
kainouria erotisi is pressed for the first time and stays
until piso button is pressed

global VrythmiseisON          --1 if rythmiseis
button is pressed

global Vdog                    --1 if dog metronome
is on

global Vdogtime                --Time for dog in
milliseconds

global Vpcnumnotes            --The number of
keydowns needed in rythm

```

```

global Vnumberkeys          --The number of
keydowns in rythm

global Vkeytimes            --[] The time a rythm
key is pressed

global Vpckeytimes         --[] The times of pc
notes in rythm

global VrythmNoteX         --[] The X for rythm
triangles

global Vrythmapotelesma    --How well you did in
rythm test

global Vrythmtime          --Changes with
arithmitis in rythm

global VinfoON             --1 if mouse is in
info button in menu

global Vmenublink          --[] 1 if ascending ,
2 if descending

global Vintervall1        --Number of
interval(1-8)

global Vinterval2         --Type of
interval(Me,Mi,E,K,A,Tri)

global Vimitonia          --The number of
semitones in the interval

global Vintervchoice      --[] Which intervals
are used 0 or 1.1=tautofonia,2=2is ... 8=oktava

global Vmorfichords        --0 for armoniki
morfi, 1 for melodiki

```

β) Οι αλλαγές του κέρσορα :

```
on exitFrame me
```

```

--CURSOR CHANGE

--Cursor X
if VcursorINT = 4 or VcursorINT = 5 or VcursorINT = 6
then
    VcursorX = _mouse.mouseH + 4
else
    VcursorX = _mouse.mouseH
end if
--Cursor Y
if Vpausi = 0 then
    if VcursorINT = 1 then
        VcursorY = _mouse.mouseV + 1
    else if VcursorINT = 0 then
        VcursorY = _mouse.mouseV - 4
    else
        VcursorY = _mouse.mouseV - 12
    end if
else if Vpausi = 1 then
    if VcursorINT = 2 then
        VcursorY = _mouse.mouseV - 3
    else
        VcursorY = _mouse.mouseV - 12
    end if
end if

--WRITE CURSOR

sprite(74).locH = VcursorX
sprite(74).locV = VcursorY

if VcursorOn = 1 then
    channel(74).makeScriptedSprite(member(Vcursorname),
point(VcursorX, VcursorY))
    sprite(74).ink = 1
else
    channel(74).removeScriptedSprite()
end if

if Vpcplayscale = 1 then
    if (sound(5).isBusy()) then
        nothing
    else
        Vpcplayscale = 0
        sound(1).play()
        sound(2).play()
    end if
end if

```

```

--FRAME LOOP

go to the frame

end

On rightmousedown me

--CURSOR CHANGE

if VcursorON = 1 then
  VcursorINT = VcursorINT + 1
  case VcursorINT of
    1 : Vcursorname = "olokliro" & Vpausiname
    2 : Vcursorname = "miso" & Vpausiname
    3 : Vcursorname = "tetarto" & Vpausiname
    4 : Vcursorname = "ogdoo" & Vpausiname
    5 : Vcursorname = "dekatoekto" & Vpausiname
    6 : Vcursorname = "goma"
      VcursorINT = 0
  end case
end if

```

γ) Οι αλλαγές της οπτικής ένδειξης στην «Ρυθμική Αγωγή» :

```

if Vdog <> 0 then
  --Metrodog
  if _system.milliseconds >
    ((60000/Vtempo)*Vmetro2clickvalue) + Vdogtime then
    if Vclickcounter = Vmetro1 then
      channel(4).makeScriptedSprite(member("metrodog
beat"),point(300, 300))
      sprite(4).locZ = 1000000
      sprite(4).rect = rect(414, 335, 487, 435)
      Vclickcounter = 1
    else
      channel(4).removescriptedsprite()
      Vclickcounter = Vclickcounter + 1
    end if
    Vdogtime = _system.milliseconds
    if Vdog = 1 then
      Vdog = 2
      sprite(3).member = member("metrodog2")
    else
      Vdog = 1
      sprite(3).member = member("metrodog1")
    end if
  end if
end if
end if

```

δ) Η διαδικασία της καινούριας ερώτησης στην άσκηση Dictée με τη σχεδίαση του οπλισμού στο πεντάγραμμο και την εκτέλεση της κλίμακας για κούρδισμα :

```
--CREATE QUESTION

--Key Signature
Voplismos = random(Vdyskolia)
Voplismos = Voplismos - 1
n = random(2)
if n = 2 then
  if Voplismos <> 0 then
    Voplismos = Voplismos-8
  end if
end if
n = random(2)
if n = 2 then
  Vgenos = "minor"
else
  Vgenos = "major"
end if
case Voplismos of                                --Name of scale
-1:
  if Vgenos = "major" then
    Vkey = "Cb major"
  else
    Vkey = "Ab minor"
  end if
-2:
  if Vgenos = "major" then
    Vkey = "Gb major"
  else
    Vkey = "Eb minor"
  end if
-3:
  if Vgenos = "major" then
    Vkey = "Db major"
  else
    Vkey = "Bb minor"
  end if
-4:
  if Vgenos = "major" then
```

```
    Vkey = "Ab major"
else
    Vkey = "F minor"
end if
-5:
  if Vgenos = "major" then
    Vkey = "Eb major"
  else
    Vkey = "C minor"
  end if
-6:
  if Vgenos = "major" then
    Vkey = "Bb major"
  else
    Vkey = "G minor"
  end if
-7:
  if Vgenos = "major" then
    Vkey = "F major"
  else
    Vkey = "D minor"
  end if
0:
  if Vgenos = "major" then
    Vkey = "C major"
  else
    Vkey = "A minor"
  end if
1:
  if Vgenos = "major" then
    Vkey = "G major"
  else
    Vkey = "E minor"
  end if
2:
  if Vgenos = "major" then
    Vkey = "D major"
  else
    Vkey = "B minor"
  end if
3:
  if Vgenos = "major" then
    Vkey = "A major"
  else
    Vkey = "F# minor"
  end if
4:
  if Vgenos = "major" then
    Vkey = "E major"
  else
    Vkey = "C# minor"
```



```

    end if
5:   if Vgenos = "major" then
      Vkey = "B major"
    else
      Vkey = "G# minor"
    end if
6:   if Vgenos = "major" then
      Vkey = "F# major"
    else
      Vkey = "D# minor"
    end if
7:   if Vgenos = "major" then
      Vkey = "C# major"
    else
      Vkey = "A# minor"
    end if
end case
n = random(2)  --2 octaves

--First Note

Vpcnota[2] = chars(Vkey,1,1) & n--Name of sound sample
case Vpcnota[2].word[1] of
  "C1": Vpcnotenumber[2] = 1
  "D1": Vpcnotenumber[2] = 2
  "E1": Vpcnotenumber[2] = 3
  "F1": Vpcnotenumber[2] = 4
  "G1": Vpcnotenumber[2] = 5
  "A1": Vpcnotenumber[2] = 6
  "B1": Vpcnotenumber[2] = 7
  "C2": Vpcnotenumber[2] = 8
  "D2": Vpcnotenumber[2] = 9
  "E2": Vpcnotenumber[2] = 10
  "F2": Vpcnotenumber[2] = 11
  "G2": Vpcnotenumber[2] = 12
  "A2": Vpcnotenumber[2] = 13
  "B2": Vpcnotenumber[2] = 14
  "C3": Vpcnotenumber[2] = 15
end case
Vpcnota[2] = chars(Vpcnota[2],1,1) & n
Vpcnotacarry[2] = Vpcnota[2]
--Scale Creation
Vpctonikotita[1] = chars(Vpcnota[2],1,1) & "1"
Vpcscalenum[1] = Vpcnotenumber[2]
if Vpcnotenumber[2] > 8 then
  Vpcscalenum[1] = Vpcscalenum[1] - 7
end if
repeat with n = 1 to 8

```

```

Vpcscalenumber[n] = Vpcscalenumber[1] + (n-1)
case Vpcscalenumber[n] of
  1: Vpctonikotita[n] = "C1"
  2: Vpctonikotita[n] = "D1"
  3: Vpctonikotita[n] = "E1"
  4: Vpctonikotita[n] = "F1"
  5: Vpctonikotita[n] = "G1"
  6: Vpctonikotita[n] = "A1"
  7: Vpctonikotita[n] = "B1"
  8: Vpctonikotita[n] = "C2"
  9: Vpctonikotita[n] = "D2"
  10: Vpctonikotita[n] = "E2"
  11: Vpctonikotita[n] = "F2"
  12: Vpctonikotita[n] = "G2"
  13: Vpctonikotita[n] = "A2"
  14: Vpctonikotita[n] = "B2"
  15: Vpctonikotita[n] = "C3"
end case
case chars(Vpctonikotita[n],1,1) of
  "F" : Vnoumeropliktrou = 1
  "C" : Vnoumeropliktrou = 2
  "G" : Vnoumeropliktrou = 3
  "D" : Vnoumeropliktrou = 4
  "A" : Vnoumeropliktrou = 5
  "E" : Vnoumeropliktrou = 6
  "B" : Vnoumeropliktrou = 7
end case
if (Voplismos >= 0 and Vnoumeropliktrou <= Voplismos)
then
  case chars(Vpctonikotita[n],1,1) of
--sharp
    "C" : Vpctonikotita[n] = "C" & "#" &
chars(Vpctonikotita[n],2,2)
    "D" : Vpctonikotita[n] = "D" & "#" &
chars(Vpctonikotita[n],2,2)
    "E" : Vpctonikotita[n] = "F" &
chars(Vpctonikotita[n],2,2)
    "F" : Vpctonikotita[n] = "F" & "#" &
chars(Vpctonikotita[n],2,2)
    "G" : Vpctonikotita[n] = "G" & "#" &
chars(Vpctonikotita[n],2,2)
    "A" : Vpctonikotita[n] = "A" & "#" &
chars(Vpctonikotita[n],2,2)
    "B" :
      if chars(Vpctonikotita[n],2,2) = 1 then
        Vpctonikotita[n] = "C" & "2"
      else
        Vpctonikotita[n] = "C" & "3"
      end if
  end case
end case

```

```

else if (Voplismos < 0 and Vnoumeropliktrou*(-1) <=
Voplismos) then --flat
  case chars(Vpctonikotita[n],1,1) of
    "C" :
      if chars(Vpctonikotita[n],2,2) = 1 then
        Vpctonikotita[n] = "B" & "0"
      else if chars(Vpctonikotita[n],2,2) = 2 then
        Vpctonikotita[n] = "B" & "1"
      else if chars(Vpctonikotita[n],2,2) = 3 then
        Vpctonikotita[n] = "B" & "2"
      end if
    "D" : Vpctonikotita[n] = "C" & "#" &
chars(Vpctonikotita[n],2,2)
    "E" : Vpctonikotita[n] = "D" & "#" &
chars(Vpctonikotita[n],2,2)
    "F" : Vpctonikotita[n] = "E" &
chars(Vpctonikotita[n],2,2)
    "G" : Vpctonikotita[n] = "F" & "#" &
chars(Vpctonikotita[n],2,2)
    "A" : Vpctonikotita[n] = "G" & "#" &
chars(Vpctonikotita[n],2,2)
    "B" : Vpctonikotita[n] = "A" & "#" &
chars(Vpctonikotita[n],2,2)
  end case
end if
end repeat
--Queue scale
sound(1).stop()
sound(2).stop()
sound(3).stop()
sound(4).stop()
sound(5).stop()
sound(6).stop()
sound(7).stop()
sound(8).stop()
sound(1).setPlayList([])
sound(2).setPlayList([])
sound(3).setPlayList([])
sound(4).setPlayList([])
sound(5).setPlayList([])
sound(6).setPlayList([])
sound(7).setPlayList([])
sound(8).setPlayList([])
repeat with n = 1 to 8
  Vpctonikotita[n] = Vpctonikotita[n] & Vinstrument
sound(5).queue([#member:member(Vpctonikotita[n]),#endtime:(
((60000/Vtempo)*Vmetro2clickvalue)/2)])
end repeat

```

```

--REST OF NOTES
Vdimiourgia = 1
Vpcdiaxoristikometrou[Vpcloop] = 0
Vpctriixo[1] = 0
Vpcloop = 1

repeat while Vdimiourgia = 1 ---Vdimiourgia ginetai 0
otan oloklirothei o arithmos ton metron

    Vpcloop = Vpcloop + 1

--Note advancement
if Vpcloop > 2 then
    Vpcnotenumber[Vpcloop] = 50
    Vcarry = Vpcnotenumber[Vpcloop]
    repeat while Vpcnotenumber[Vpcloop] > 15 or
Vpcnotenumber[Vpcloop] < 1
        Vpcnotenumber[Vpcloop] = Vcarry
        Vnotenumberloop = random(Vdyskolia) -- 1 eos
9(8?)
        Vnotenumberloop = Vnotenumberloop - 1 --Mexri kai
diastimata 8is
        if Vnotenumberloop = 0 then
            n = random(10)
            if n <> 10 then
                Vnotenumberloop = 1
            end if
        end if
        n = random(2)
        if n = 1 then
            Vpcnotenumber[Vpcloop] = Vpcnotenumber[Vpcloop-
1]+Vnotenumberloop
        else
            Vpcnotenumber[Vpcloop] = Vpcnotenumber[Vpcloop-
1]-Vnotenumberloop
        end if
    end repeat

case Vpcnotenumber[Vpcloop] of
    1: Vpcnota[Vpcloop] = "C1"
    2: Vpcnota[Vpcloop] = "D1"
    3: Vpcnota[Vpcloop] = "E1"
    4: Vpcnota[Vpcloop] = "F1"
    5: Vpcnota[Vpcloop] = "G1"
    6: Vpcnota[Vpcloop] = "A1"
    7: Vpcnota[Vpcloop] = "B1"
    8: Vpcnota[Vpcloop] = "C2"
    9: Vpcnota[Vpcloop] = "D2"
    10: Vpcnota[Vpcloop] = "E2"

```

```

11: Vpcnota[Vpcloop] = "F2"
12: Vpcnota[Vpcloop] = "G2"
13: Vpcnota[Vpcloop] = "A2"
14: Vpcnota[Vpcloop] = "B2"
15: Vpcnota[Vpcloop] = "C3"
end case
Vpcnotacarry[Vpcloop] = Vpcnota[Vpcloop]
end if

Vpcaxia[Vpcloop] = 50 --Rythmic value
if Vpctriixo[Vpcloop-1] = 1 or Vpctriixo[Vpcloop-1] =
2 then
    Vpcaxia[Vpcloop] = Vpcaxia[Vpcloop-1]
    Vpctypossprite[Vpcloop] = Vpctypossprite[Vpcloop-1]
else
    repeat while
(Vpcaxia[Vpcloop]+Vpcdiaxoristikometrou[Vpcloop-1] >
Vmetrol) and
(Vpcaxia[Vpcloop]+Vpcdiaxoristikometrou[Vpcloop-1] <>
Vmetrol)
        n = random(5)
        n2 = random(2)
        if n = 1 then
            n = n + n2 - 1
        end if
        case Vmetro2 of
1:
            case n of
1 : Vpcaxia[Vpcloop] = 1
    Vpctypossprite[Vpcloop] = 1
2 : Vpcaxia[Vpcloop] = 0.5
    Vpctypossprite[Vpcloop] = 2
3 : Vpcaxia[Vpcloop] = 0.25
    Vpctypossprite[Vpcloop] = 3
4 : Vpcaxia[Vpcloop] = 0.125
    Vpctypossprite[Vpcloop] = 4
5 : Vpcaxia[Vpcloop] = 0.0625
    Vpctypossprite[Vpcloop] = 5
            end case
2:
            case n of
1 : Vpcaxia[Vpcloop] = 2
    Vpctypossprite[Vpcloop] = 1
2 : Vpcaxia[Vpcloop] = 1
    Vpctypossprite[Vpcloop] = 2
3 : Vpcaxia[Vpcloop] = 0.5
    Vpctypossprite[Vpcloop] = 3
4 : Vpcaxia[Vpcloop] = 0.25
    Vpctypossprite[Vpcloop] = 4
5 : Vpcaxia[Vpcloop] = 0.125
    Vpctypossprite[Vpcloop] = 5
            end case
        end case
    end repeat
end if
end case

```

```

end case
4:
case n of
  1 : Vpcaxia[Vpcloop] = 4
    Vpctypossprite[Vpcloop] = 1
  2 : Vpcaxia[Vpcloop] = 2
    Vpctypossprite[Vpcloop] = 2
  3 : Vpcaxia[Vpcloop] = 1
    Vpctypossprite[Vpcloop] = 3
  4 : Vpcaxia[Vpcloop] = 0.5
    Vpctypossprite[Vpcloop] = 4
  5 : Vpcaxia[Vpcloop] = 0.25
    Vpctypossprite[Vpcloop] = 5
end case
8:
case n of
  1 : Vpcaxia[Vpcloop] = 8
    Vpctypossprite[Vpcloop] = 1
  2 : Vpcaxia[Vpcloop] = 4
    Vpctypossprite[Vpcloop] = 2
  3 : Vpcaxia[Vpcloop] = 2
    Vpctypossprite[Vpcloop] = 3
  4 : Vpcaxia[Vpcloop] = 1
    Vpctypossprite[Vpcloop] = 4
  5 : Vpcaxia[Vpcloop] = 0.5
    Vpctypossprite[Vpcloop] = 5
end case
16:
case n of
  1 : Vpcaxia[Vpcloop] = 16
    Vpctypossprite[Vpcloop] = 1
  2 : Vpcaxia[Vpcloop] = 8
    Vpctypossprite[Vpcloop] = 2
  3 : Vpcaxia[Vpcloop] = 4
    Vpctypossprite[Vpcloop] = 3
  4 : Vpcaxia[Vpcloop] = 2
    Vpctypossprite[Vpcloop] = 4
  5 : Vpcaxia[Vpcloop] = 1
    Vpctypossprite[Vpcloop] = 5
end case
end case
end repeat
end if

Vpcalliosi[Vpcloop] = 0
if Vpcloop > 2 then
  n = random(15)
  if n = 15 then
    n = random(3)
    if n = 1 then
      --Intonation

```

```

        Vpcalliosi[Vpcloop] = 1
    else if n = 2 then
        Vpcalliosi[Vpcloop] = 2
    else if n = 3 then
        Vpcalliosi[Vpcloop] = 3
    end if
end if
end if

    case chars(Vpcnota[Vpcloop],1,1) of --
Vnumeropliktrou
        "F" : Vnumeropliktrou = 1
        "C" : Vnumeropliktrou = 2
        "G" : Vnumeropliktrou = 3
        "D" : Vnumeropliktrou = 4
        "A" : Vnumeropliktrou = 5
        "E" : Vnumeropliktrou = 6
        "B" : Vnumeropliktrou = 7
    end case

    Vpcparestigmeno[Vpcloop] = 0 --Parestigmeno
    if Vdyskolia > 1 then
        n = random(10)
        if n = 10 and (Vpctriixo[Vpcloop-1] = 0 or
Vpctriixo[Vpcloop-1] = 3) and (Vpcaxia[Vpcloop] > 0.2500)
then
            if ((Vpcaxia[Vpcloop] + (Vpcaxia[Vpcloop]/2.0)) +
Vpcdiaxoristikometrou[Vpcloop-1])<=Vmetrol then
                Vpcparestigmeno[Vpcloop] = 1
                Vpcaxia[Vpcloop] = Vpcaxia[Vpcloop] +
(Vpcaxia[Vpcloop]/2.0)
            else
                Vpcparestigmeno[Vpcloop] = 0
            end if
        end if
    end if

    Vpctriixo[Vpcloop] = 0 --Triplet 1,2 or 3
    if Vdyskolia > 2 then
        n = random(15)
        if (n = 15 and Vpcparestigmeno[Vpcloop] = 0) or
(Vpctriixo[Vpcloop-1] = 1 or Vpctriixo[Vpcloop-1] = 2) then
            Vpctriixo[Vpcloop] = Vpctriixo[Vpcloop-1] + 1
            case Vpctriixo[Vpcloop] of

```

```

1:
    if (((Vpcaxia[Vpcloop]*2)/3.0)*3) +
Vpcdiaxoristikometrou[Vpcloop-1])<=Vmetrol then
        Vpcaxia[Vpcloop] = (Vpcaxia[Vpcloop] * 2) /
3.0
    else
        Vpctriixo[Vpcloop] = 0
    end if
4:
    Vpctriixo[Vpcloop] = 1
    if (((Vpcaxia[Vpcloop]*2)/3.0)*3) +
Vpcdiaxoristikometrou[Vpcloop-1])<=Vmetrol then
        Vpcaxia[Vpcloop] = (Vpcaxia[Vpcloop] * 2) /
3.0
    else
        Vpctriixo[Vpcloop] = 0
    end if
end case
end if
end if

n = random(15)      --Rests
Vpcpausi[Vpcloop] = 0
if n = 15 and Vpcpausi[Vpcloop-1] = 0 then
    Vpcpausi[Vpcloop] = 1
    Vpctypossprite[Vpcloop] = Vpctypossprite[Vpcloop] +
5
    Vpcnotenumber[Vpcloop] = Vpcnotenumber[Vpcloop-1]
end if

--Correction of note acording to Key signature

if Vpcalliosi[Vpcloop] = 2 or (Voplismos >= 0 and
Vnnumeropliktrou <= Voplismos and Vpcalliosi[Vpcloop] <> 1
and Vpcalliosi[Vpcloop] <> 3) then
    case chars(Vpcnota[Vpcloop],1,1) of
--sharp
        "C" : Vpcnota[Vpcloop] = "C" & "#" &
chars(Vpcnota[Vpcloop],2,2)
        "D" : Vpcnota[Vpcloop] = "D" & "#" &
chars(Vpcnota[Vpcloop],2,2)
        "E" : Vpcnota[Vpcloop] = "F" &
chars(Vpcnota[Vpcloop],2,2)
        "F" : Vpcnota[Vpcloop] = "F" & "#" &
chars(Vpcnota[Vpcloop],2,2)
        "G" : Vpcnota[Vpcloop] = "G" & "#" &
chars(Vpcnota[Vpcloop],2,2)
        "A" : Vpcnota[Vpcloop] = "A" & "#" &
chars(Vpcnota[Vpcloop],2,2)

```



```

        "B" :
            if chars(Vpcnota[Vpcloop],2,2) = 1 then
                Vpcnota[Vpcloop] = "C" & "2"
            else
                Vpcnota[Vpcloop] = "C" & "3"
            end if
        end case
    else if Vpcalliosi[Vpcloop] = 3 or (Voplismos < 0 and
\
Vnnumeropliktrou*(-1) <= Voplismos and \
Vpcalliosi[Vpcloop] <> 1 and Vpcalliosi[Vpcloop] <>
2)then
        --flat
        case chars(Vpcnota[Vpcloop],1,1) of
            "C" :
                if chars(Vpcnota[Vpcloop],2,2) = 1 then
                    Vpcnota[Vpcloop] = "B" & "0"
                else if chars(Vpcnota[Vpcloop],2,2) = 2 then
                    Vpcnota[Vpcloop] = "B" & "1"
                else if chars(Vpcnota[Vpcloop],2,2) = 3 then
                    Vpcnota[Vpcloop] = "B" & "2"
                end if
            "D" : Vpcnota[Vpcloop] = "C" & "#" &
chars(Vpcnota[Vpcloop],2,2)
            "E" : Vpcnota[Vpcloop] = "D" & "#" &
chars(Vpcnota[Vpcloop],2,2)
            "F" : Vpcnota[Vpcloop] = "E" &
chars(Vpcnota[Vpcloop],2,2)
            "G" : Vpcnota[Vpcloop] = "F" & "#" &
chars(Vpcnota[Vpcloop],2,2)
            "A" : Vpcnota[Vpcloop] = "G" & "#" &
chars(Vpcnota[Vpcloop],2,2)
            "B" : Vpcnota[Vpcloop] = "A" & "#" &
chars(Vpcnota[Vpcloop],2,2)
        end case
    end if

        --Diaxoristiko metrou
        Vpcdiaxoristikometrou[Vpcloop] =
Vpcdiaxoristikometrou[Vpcloop-1]+Vpcaxia[Vpcloop]
        if Vpcdiaxoristikometrou[Vpcloop] = Vmetrol then
            Vpcdiaxoristikometrou[Vpcloop] = 0
            Vpcmetra = Vpcmetra + 1
        end if
        if Vpcmetra = Vtotalmetra then
            -----
            -----POSA METRA
            Vdimiourgia = 0
        end if
    end if
end if

```

```

    if Vpcpausi[Vpcloop] = 1 and Vpcloop > 2 then
        Vpcnota[Vpcloop] = Vpcnota[Vpcloop-1]
    end if
end repeat

--Key signature
if Vdrawoplismos = 1 then
    Vscriptedchanneloplismos = 699
    if Voplismos > 0 then
        repeat with n = 1 to Voplismos
            case n of
                1: VnoteY[1] = 219
                2: VnoteY[1] = 234
                3: VnoteY[1] = 214
                4: VnoteY[1] = 229
                5: VnoteY[1] = 244
                6: VnoteY[1] = 224
                7: VnoteY[1] = 239
            end case
            VnoteX[1] = VnoteX[1] + 10

channel(Vscriptedchanneloplismos+n).makeScriptedSprite(memb
er("diesi"), point(VnoteX[1], VnoteY[1]))
        sprite(Vscriptedchanneloplismos+n).ink = 1
        end repeat
    else if Voplismos < 0 then
        repeat with n = 1 to (8-(Voplismos.abs))
            case n of
                1: VnoteY[1] = 233
                2: VnoteY[1] = 218
                3: VnoteY[1] = 238
                4: VnoteY[1] = 223
                5: VnoteY[1] = 243
                6: VnoteY[1] = 228
                7: VnoteY[1] = 248
            end case
            VnoteX[1] = VnoteX[1] + 10

channel(Vscriptedchanneloplismos+n).makeScriptedSprite(memb
er("yfesi"), point(VnoteX[1], VnoteY[1]))
        sprite(Vscriptedchanneloplismos+n).ink = 1
        end repeat
    end if
    VnoteX[1] = VnoteX[1] - 31

--Time siganture
VnoteX[1] = VnoteX[1] + Vapostasinoton + 10 --
kanonika xoris to 10
case Vmetrol of

```

```

1: Vzografianotas = "ena"
2: Vzografianotas = "dyo"
3: Vzografianotas = "tria"
4: Vzografianotas = "tessera"
5: Vzografianotas = "pente"
6: Vzografianotas = "eksi"
7: Vzografianotas = "efta"
8: Vzografianotas = "oxto"
9: Vzografianotas = "enia"
10: Vzografianotas = "deka"
11: Vzografianotas = "edeka"
12: Vzografianotas = "dodeka"
end case
VnoteY[1] = 229

channel(709).makeScriptedSprite(member(Vzografianotas),
point(VnoteX[1], VnoteY[1]))
sprite(709).ink = 1
case Vmetro2 of
2: Vzografianotas = "dyo"
4: Vzografianotas = "tessera"
8: Vzografianotas = "oxto"
16: Vzografianotas = "dekaeksi"
end case
VnoteY[1] = 250

channel(710).makeScriptedSprite(member(Vzografianotas),
point(VnoteX[1], VnoteY[1]))
sprite(710).ink = 1
VnoteX[1] = VnoteX[1] - 18 --29
Vdrawoplismos = 0
end if

cursor 280

sound(2).stop()
sound(1).stop()
sound(2).setPlayList([])
sound(1).setPlayList([])
repeat with n = 2 to Vpcloop
if Vpcpausi[n] = 1 then
Vtonikotita[n] = "silence"
else
Vtonikotita[n] = Vpcnota[n] & Vinstrument
end if
Vaxianotas[n] = (60000/Vtempo)*Vpcaxia[n]

sound(2).queue([#member:member(Vtonikotita[n]),#endtime:Vax
ianotas[n]])
end repeat
Vclickcounter = Vmetrol

```

```

repeat with n = 1 to Vmetrol*Vtotalmetra
  if Vclickcounter = Vmetrol then
    Vclickcounter = 1
  sound(1).queue([#member:member("metronome1"),#endtime:((600
00/Vtempo)*Vmetro2clickvalue)])
  else
    Vclickcounter = Vclickcounter + 1
  sound(1).queue([#member:member("metronome2"),#endtime:((600
00/Vtempo)*Vmetro2clickvalue)])
end if
end repeat

```

ε) Η αλλαγή της απόστασης των νοτών ανάλογα με το συνολικό αριθμό τους :

```

--DISTANCE BETWEEN NOTES
if Vnoteused > 10 then
  if VcursorINT <> 0 then
    if Vapostasinoton > 14 then
      Vapostasinoton = Vapostasinoton - 2
    end if
  end if
end if
if Vnoteused < 21 then
  if VcursorINT = 0 then
    if Vapostasinoton < 42 then
      Vapostasinoton = Vapostasinoton +
2*(Vxrisimopoioiomenasprites[Vsystema-1])
    end if
  end if
end if

```

στ) Το παίξιμο μιας νότας :

```

-PLAY NOTE
if Vpausi = 0 and VcursorINT <> 0 then
  if Valliosion = 2 or (Voplismos >= 0 and
Vnoumeropliktrou <= Voplismos and Valliosion <> 1 and
Valliosion <> 3) then

```

```

--sharp
    case chars(Vpitch,1,1) of
        "C" : Vpitch = "C" & "#" & chars(Vpitch,2,2)
        "D" : Vpitch = "D" & "#" & chars(Vpitch,2,2)
        "E" : Vpitch = "F" & chars(Vpitch,2,2)
        "F" : Vpitch = "F" & "#" & chars(Vpitch,2,2)
        "G" : Vpitch = "G" & "#" & chars(Vpitch,2,2)
        "A" : Vpitch = "A" & "#" & chars(Vpitch,2,2)
        "B" :
            if chars(Vpitch,2,2) = 1 then
                Vpitch = "C" & "2"
            else
                Vpitch = "C" & "3"
            end if
        end case
    else if Valliosion = 3 or (Voplismos < 0 and \
Vnumeropliktrou*(-1) <= Voplismos and \
Valliosion <> 1 and Valliosion <> 2)then
flat
    case chars(Vpitch,1,1) of
        "C" :
            if chars(Vpitch,2,2) = 1 then
                Vpitch = "B" & "0"
            else if chars(Vpitch,2,2) = 2 then
                Vpitch = "B" & "1"
            else if chars(Vpitch,2,2) = 3 then
                Vpitch = "B" & "2"
            end if
        "D" : Vpitch = "C" & "#" & chars(Vpitch,2,2)
        "E" : Vpitch = "D" & "#" & chars(Vpitch,2,2)
        "F" : Vpitch = "E" & chars(Vpitch,2,2)
        "G" : Vpitch = "F" & "#" & chars(Vpitch,2,2)
        "A" : Vpitch = "G" & "#" & chars(Vpitch,2,2)
        "B" : Vpitch = "A" & "#" & chars(Vpitch,2,2)
    end case
    end if
    Vnote = Vpitch & Vinstrument
    sound(3).play(member(Vnote))
    end if

```

ζ) Η εμφάνιση μίας καινούριας νότας στο πεντάγραμμο μαζί με οποιοδήποτε σύμβολο αλλοίωσης :

```
--INTONATION BEFORE NOTE
```

```

if VcursorINT <> 0 then
  if Vpausi = 0 then

    if Valliosion = 0 then          --CHECK FOR
PASSED SAME NOTE INTONATION
      n = Vnoteused - 1
      repeat while n > 1 and Vtypossprite[n] <> 17
        if VnoteY[n] = VnoteY[Vnoteused] then
          if Vtypossprite[n-1] > 10 and
Vtypossprite[n-1] < 14 then
            if Voplismos = 0 or (Voplismos > 0 and
Vnoumeropliktrou > Voplismos) or (Voplismos < 0 and
(Vnoumeropliktrou*(-1)) > Voplismos) then
              if Vtypossprite[n-1] <> 11 then
--aneresi
                  Vzografianotas = "aneresi"
VnoteX[Vnoteused] = VnoteX[Vnoteused
- 1] + Vapostasinoton
VnoteY[Vnoteused]
                  VnoteY[Vnoteused + 1] =
VnoteY[Vnoteused]
+ 1] + 13
                  Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
                  Vtypossprite[Vnoteused] = 11
                  end if
                  else if Voplismos > 0 then
--diesi
                      if Vtypossprite[n-1] <> 12 then
Vzografianotas = "diesi"
VnoteX[Vnoteused] = VnoteX[Vnoteused
- 1] + Vapostasinoton
VnoteY[Vnoteused]
                      VnoteY[Vnoteused + 1] =
VnoteY[Vnoteused]
+ 1] + 13
                      Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
                      Vtypossprite[Vnoteused] = 12
                      end if
                      else
--yfesi
                          if Vtypossprite[n-1] <> 13 then

```

```

        Vzografianotas = "yfesi"
        VnoteX[Vnoteused] = VnoteX[Vnoteused
- 1] + Vapostasinoton
        VnoteY[Vnoteused + 1] =
VnoteY[Vnoteused]
        VnoteY[Vnoteused] = VnoteY[Vnoteused
+ 1] + 7
        Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
        Vtypossprite[Vnoteused] = 13
        end if
        end if
        sprite(Vscriptedspritechannel).ink = 1
        n = 2
        end if
        end if
        n = n - 1
    end repeat
end if

if Voplismos < 0 then                                --For
flats
    case Valliosion of
        1 :                                          --
aneresi
            if (Vnoumeropliktrou*(-1)) <= Voplismos
then
                Vzografianotas = "aneresi"
                VnoteX[Vnoteused] = VnoteX[Vnoteused - 1]
+ Vapostasinoton
                VnoteY[Vnoteused + 1] = VnoteY[Vnoteused]
                VnoteY[Vnoteused] = VnoteY[Vnoteused + 1]
+ 13
                Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
                Vtypossprite[Vnoteused] = 11
                end if
            2 :                                      --diesi
                Vzografianotas = "diesi"
                VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] +
Vapostasinoton

```

```

VnoteY[Vnoteused + 1] = VnoteY[Vnoteused]
VnoteY[Vnoteused] = VnoteY[Vnoteused + 1] +
13
Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1
channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
Vtypossprite[Vnoteused] = 12
3 : --yfesi
if (Vnouveropliktrou*(-1)) > Voplismos then
Vzografianotas = "yfesi"
VnoteX[Vnoteused] = VnoteX[Vnoteused - 1]
+ Vapostasinoton
VnoteY[Vnoteused + 1] = VnoteY[Vnoteused]
VnoteY[Vnoteused] = VnoteY[Vnoteused + 1]
+ 7
Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1
channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
Vtypossprite[Vnoteused] = 13
end if
end case
end if
if Voplismos >= 0 then --For
sharp
case Valliosion of
1 : --
aneresi
if Vnouveropliktrou <= Voplismos then
Vzografianotas = "aneresi"
VnoteX[Vnoteused] = VnoteX[Vnoteused - 1]
+ Vapostasinoton
VnoteY[Vnoteused + 1] = VnoteY[Vnoteused]
VnoteY[Vnoteused] = VnoteY[Vnoteused + 1]
+ 13
Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1
channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
Vtypossprite[Vnoteused] = 11
end if
2 : --diesi
if Vnouveropliktrou > Voplismos then
Vzografianotas = "diesi"

```



```

VnoteX[Vnoteused] = VnoteX[Vnoteused - 1]
+ Vapostasinoton
VnoteY[Vnoteused + 1] = VnoteY[Vnoteused]
VnoteY[Vnoteused] = VnoteY[Vnoteused + 1]
+ 13
Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
Vtypossprite[Vnoteused] = 12
end if
3 : --yfesi
Vzografianotas = "yfesi"
VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] +
Vapostasinoton
VnoteY[Vnoteused + 1] = VnoteY[Vnoteused]
VnoteY[Vnoteused] = VnoteY[Vnoteused + 1] +
7
Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
Vtypossprite[Vnoteused] = 13
end case
end if
sprite(Vscriptedspritechannel).ink = 1
end if
end if

--WRITE NOTE

if Vtypossprite[Vnoteused]>10 and
Vtypossprite[Vnoteused]<14 then --Notex
Vscriptedspritechannel = Vscriptedspritechannel + 1
Vnoteused = Vscriptedspritechannel - 99
case VcursorINT of
4: VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] + 15
5: VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] + 15
otherwise: VnoteX[Vnoteused] = VnoteX[Vnoteused -
1] + 11
end case
else
case VcursorINT of
4: VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] +
Vapostasinoton + 4
5: VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] +
Vapostasinoton + 4

```

```

        otherwise: VnoteX[Vnoteused] = VnoteX[Vnoteused -
1] + Vapostasinoton
    end case
    if Vpausi = 1 then
        case VcursorINT of
            1 : VnoteY[Vnoteused] = 232 --rest + 13
              Vtypossprite[Vnoteused] = 6
            2 : VnoteY[Vnoteused] = 237
              Vtypossprite[Vnoteused] = 7
            3 : VnoteY[Vnoteused] = 240
              Vtypossprite[Vnoteused] = 8
            4 : VnoteY[Vnoteused] = 237
              Vtypossprite[Vnoteused] = 9
            5 : VnoteY[Vnoteused] = 241
              Vtypossprite[Vnoteused] = 10
        end case
    end if
end if

if Vpausi = 0 then
    case VcursorINT of
        1 : Vtypossprite[Vnoteused] = 1
          VnoteY[Vnoteused] = VnoteY[Vnoteused] + 13
        2 : Vtypossprite[Vnoteused] = 2
        3 : Vtypossprite[Vnoteused] = 3
        4 : Vtypossprite[Vnoteused] = 4
        5 : Vtypossprite[Vnoteused] = 5
    end case
end if

Vzografianotas = Vcursorname

if VcursorINT <> 0 then

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
    Vnoteinsystem[Vsystema] = Vnoteused
    Vlathosdiaxoristiko[Vnoteinsystem[Vsystema]] = 0
else
notes
--Erase

    if Vsystema - 1 > 1 then
notes for correct diaxoristiko metrou
--Black
        if Vtypossprite[Vnoteused-1] = 17 then
            n = Vnoteused-2
            repeat while Vtypossprite[n] <> 17 and n > 1
                sprite(Vscriptedspritechannel-(Vnoteused-
n)).color = rgb(0,0,0)
            end repeat
        end if
    end if
end if

```

```

        Vlathosdiaxoristiko[n-1] = 0
        n = n - 1
    end repeat
end if

Vsystema = Vsystema - 1
Vscriptedspritechannel = Vscriptedspritechannel -
1
Vnoteused = Vscriptedspritechannel - 99
0
repeat while Vxrisimopoioumenasprites[Vsystema] >
    if Vtypossprite[Vnoteused] = 17 then
        if Vclicktimes > 1 then
            Vclicktimes = Vclicktimes - 1
        end if
    end if

channel(Vscriptedspritechannel).removeScriptedSprite()
    Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] - 1
    Vscriptedspritechannel = Vscriptedspritechannel
- 1
    Vnoteused = Vscriptedspritechannel - 99
    end repeat
    channel(400+Vsystema).removeScriptedSprite()
    Vsystema = Vsystema - 1
else
    Vscriptedspritechannel = 100
    Vsystema = 1
end if
end if
sprite(Vscriptedspritechannel).ink = 1

--INTONATION AFTER NOTE
if VcursorINT <> 0 then
    if Vpausi = 0 then
        --Voithitikes grammes
        if VpliktroON = 3 or VpliktroON = 4 or VpliktroON
= 24 or \
VpliktroON = 25 then
            --mia voithitiki
            Vzografianotas = "mia voithitiki"
            Vscriptedspritechannel = Vscriptedspritechannel
+ 1
            Vnoteused = Vscriptedspritechannel - 99
            VnoteX[Vnoteused] = VnoteX[Vnoteused - 1]
            if VcursorINT = 1 then
                VnoteY[Vnoteused] = VnoteY[Vnoteused - 1]
            else if VcursorINT = 2 or VcursorINT = 3 then
                VnoteY[Vnoteused] = VnoteY[Vnoteused - 1] +
13
            else if VcursorINT = 4 or VcursorINT = 5 then

```

```

VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] - 4
VnoteY[Vnoteused] = VnoteY[Vnoteused - 1] +
13
    end if
    Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
    Vtypossprite[Vnoteused] = 14
    sprite(Vscriptedspritechannel).ink = 1
    else if VpliktroON = 26 then
    Vzografianotas = "mia voithitiki"
    Vscriptedspritechannel = Vscriptedspritechannel
+ 1
    Vnoteused = Vscriptedspritechannel - 99
    VnoteX[Vnoteused] = VnoteX[Vnoteused - 1]
    if VcursorINT = 1 then
        VnoteY[Vnoteused] = VnoteY[Vnoteused - 1] + 5
    else if VcursorINT = 2 or VcursorINT = 3 then
18
        VnoteY[Vnoteused] = VnoteY[Vnoteused - 1] +
18
        else if VcursorINT = 4 or VcursorINT = 5 then
            VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] - 4
            VnoteY[Vnoteused] = VnoteY[Vnoteused - 1] +
18
        end if
        Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
    Vtypossprite[Vnoteused] = 14
    sprite(Vscriptedspritechannel).ink = 1
    else if VpliktroON = 27 then
18
        Vzografianotas = "dyo voithitikes"
        Vscriptedspritechannel = Vscriptedspritechannel
+ 1
        Vnoteused = Vscriptedspritechannel - 99
        VnoteX[Vnoteused] = VnoteX[Vnoteused - 1]
        if VcursorINT = 1 then
            VnoteY[Vnoteused] = VnoteY[Vnoteused - 1] + 5
        else
18
            VnoteY[Vnoteused] = VnoteY[Vnoteused - 1] +
            end if
            Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1

```

```

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
    Vtypossprite[Vnoteused] = 15
    sprite(Vscriptedspritechannel).ink = 1
end if
end if
--Parestigmeno
if VparestigmenoON = 1 then
    Vzografianotas = "parestigmeno"
    Vscriptedspritechannel = Vscriptedspritechannel +
1
    Vnoteused = Vscriptedspritechannel - 99
    VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] + 10
    if Vtypossprite[Vnoteused - 1] = 14 or
Vtypossprite[Vnoteused - 1] = 15 then
        VnoteYmemory = VnoteYmemory + 1
    end if
    if Vpausi = 1 then
        case VcursorINT of
            1: VnoteY[Vnoteused] = VnoteY[Vnoteused -
VnoteYmemory]
            2: VnoteY[Vnoteused] = VnoteY[Vnoteused -
VnoteYmemory]
            3: VnoteY[Vnoteused] = VnoteY[Vnoteused -
VnoteYmemory] -7
            4: VnoteY[Vnoteused] = VnoteY[Vnoteused -
VnoteYmemory] -4
            5: VnoteY[Vnoteused] = VnoteY[Vnoteused -
VnoteYmemory] -8
        end case
    else if Vpausi = 0 then
        if VcursorINT = 1 then
            VnoteY[Vnoteused] = VnoteY[Vnoteused -
VnoteYmemory]
        else
            VnoteY[Vnoteused] = VnoteY[Vnoteused -
VnoteYmemory] + 12
        end if
    end if
end if
    Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
    Vtypossprite[Vnoteused] = 16
    sprite(Vscriptedspritechannel).ink = 1
end if
end if

```

```

--Diaxoristikometrou
if Vdiaxoristikometrou[Vsystema] = Vmetrol then
--Correct
  Vclickend = 1 --Used for metronome count
  Vzografianotas = "diaxoristikometrou"
  Vscriptedspritechannel = Vscriptedspritechannel + 1
  Vnoteused = Vscriptedspritechannel - 99
  VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] +
Vapostasinoton
  VnoteY[Vnoteused] = 239

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
  Vtypossprite[Vnoteused] = 17
  sprite(Vscriptedspritechannel).ink = 1
  Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1
  Vdiaxoristikometrou[Vsystema] = 0
  else if Vdiaxoristikometrou[Vsystema] > Vmetrol then
--Wrong
  Vzografianotas = "diaxoristikometrou"
  Vscriptedspritechannel = Vscriptedspritechannel + 1
  Vnoteused = Vscriptedspritechannel - 99
  VnoteX[Vnoteused] = VnoteX[Vnoteused - 1] +
Vapostasinoton
  VnoteY[Vnoteused] = 239

channel(Vscriptedspritechannel).makeScriptedSprite(member(V
zografianotas), point(VnoteX[Vnoteused],
VnoteY[Vnoteused]))
  Vtypossprite[Vnoteused] = 17
  sprite(Vscriptedspritechannel).ink = 1
  Vxrisimopoioumenasprites[Vsystema] =
Vxrisimopoioumenasprites[Vsystema] + 1
  Vdiaxoristikometrou[Vsystema] = 0
  sprite(Vscriptedspritechannel).color = rgb(255,0,0)
  n = Vnoteused - 1
  repeat while Vtypossprite[n] <> 17 and n > 1
    sprite(Vscriptedspritechannel-(Vnoteused-
n)).color = rgb(255,0,0)
    Vlathosdiaxoristikiko[n] = 1 --n-1 kanonika
    n = n - 1
  end repeat
end if

```

η) Η επαναγραφή όλων των προηγούμενων νοτών στην καινούρια τους θέση όταν γράφεται μία καινούρια νότα και η εμφάνιση των ουρών στις νότες 8^{00} και 16^{00} , καθώς και η εμφάνιση του ενδεικτικού για τρίηχα :

```

--Passed Notes Location

    repeat with Vloopbeams = 501 to Vchannelbeam1
        channel(Vloopbeams).removeScriptedSprite()  --
Erase 8th beams
    end repeat
    repeat with Vloopbeams = 551 to Vchannelbeam2
        channel(Vloopbeams).removeScriptedSprite()  --
Erase 16th beams
    end repeat
    repeat with Vloopbeams = 601 to Vchannelbeam3
        channel(Vloopbeams).removeScriptedSprite()  --
Erase triplets
    end repeat

Vchannelbeam1 = 500
Vchannelbeam2 = 550
Vchannelbeam3 = 600
Vloopspritechannel = 100
Vloopnoteused = Vloopspritechannel - 99
Vymax = 3000
Vymax2 = 3000
Vlooptriixo = 0

    repeat with Vloopssystema = 2 to Vsystema  --proto
systema
        repeat with Vloopsprites = 1 to
(Vxrisimopoioumenasprites[Vloopssystema])--proto sprite

            Vloopspritechannel = Vloopspritechannel + 1
            Vloopnoteused = Vloopspritechannel - 99  -- =
Vnoteused

            if Vloopssystema = 2 then
                case Vtypossprite[Vloopnoteused] of
                    1: Vzografianotas = "olokliro"
                    2: Vzografianotas = "miso"
                    3: Vzografianotas = "tetarto"
                    4: Vzografianotas = "ogdoo"
                    5: Vzografianotas = "dekatoekto"
                    6: Vzografianotas = "olokliro pausi"
                    7: Vzografianotas = "miso pausi"
                    8: Vzografianotas = "tetarto pausi"
                    9: Vzografianotas = "ogdoo pausi"
                    10: Vzografianotas = "dekatoekto pausi"
                    11: Vzografianotas = "aneresi"
                    12: Vzografianotas = "diesi"

```

```

13: Vzografianotas = "yfesi"
14: Vzografianotas = "mia voithitiki"
15: Vzografianotas = "dyo voithitikes"
16: Vzografianotas = "parestigmemo"
17: Vzografianotas = "diaxoristikometrou"
end case

channel(Vloopspritechannel).makeScriptedSprite(member(Vzografianotas), point(VnoteX[Vloopnoteused], VnoteY[Vloopnoteused]))
sprite(Vscriptedspritechannel).ink = 1

channel(400+Vloopsystema).removeScriptedSprite()
end if

if Vloopsystema > 2 then --NoteX
  if Vloopsprites = 1 then
    case Vtypossprite[Vloopnoteused] of
      4: VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1] + Vapostasinoton + 4
      5: VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1] + Vapostasinoton + 4
      otherwise: VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1] + Vapostasinoton
    end case
  else
    case Vtypossprite[Vloopnoteused] of
      4: VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1] + 15
      5: VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1] + 15
      14:
        if Vtypossprite[Vloopnoteused-1] = 4 or
Vtypossprite[Vloopnoteused-1] = 5 then
          VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1] - 4
        else
          VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1]
        end if
      15:
        if Vtypossprite[Vloopnoteused-1] = 4 or
Vtypossprite[Vloopnoteused-1] = 5 then
          VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1] - 4
        else
          VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1]
        end if
    end case
  end if
end if

```



```

        16: VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1] + 10
        17: VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1] + Vapostasinoton
        otherwise: VnoteX[Vloopnoteused] =
VnoteX[Vloopnoteused - 1] + 11
    end case
end if
case Vtypossprite[Vloopnoteused] of
  1: Vzografianotas = "olokliro"
  2: Vzografianotas = "miso"
  3: Vzografianotas = "tetarto"
  4: Vzografianotas = "ogdoo"
  5: Vzografianotas = "dekatoekto"
  6: Vzografianotas = "olokliro pausi"
  7: Vzografianotas = "miso pausi"
  8: Vzografianotas = "tetarto pausi"
  9: Vzografianotas = "ogdoo pausi"
 10: Vzografianotas = "dekatoekto pausi"
 11: Vzografianotas = "aneresi"
 12: Vzografianotas = "diesi"
 13: Vzografianotas = "yfesi"
 14: Vzografianotas = "mia voithitiki"
 15: Vzografianotas = "dyo voithitikes"
 16: Vzografianotas = "parestigmeno"
 17: Vzografianotas = "diaxoristikometrou"
end case

channel(Vloopspritechannel).makeScriptedSprite(member(Vzografianotas),
point(VnoteX[Vloopnoteused],
VnoteY[Vloopnoteused]))
    sprite(Vscriptedspritechannel).ink = 1

channel(400+Vloopsystem).removeScriptedSprite()
    end if
end repeat

--Triplets
Vtriixosystem = 0
if Vmemtriixo[Vloopsystem] = 1 then
    Vlooptriixo = Vlooptriixo + 1
    if Vloopsystem = 2 then
        Vlooptriixo = 1        ---For 1st system
    else if
(Vtypossprite[Vnoteinsystem[Vloopsystem]-1] = 17) or
(Vtypossprite[Vnoteinsystem[Vloopsystem]-1] > 10 and
Vtypossprite[Vnoteinsystem[Vloopsystem]-1] < 14 and
Vtypossprite[Vnoteinsystem[Vloopsystem]-2] = 17) then
        Vlooptriixo = 1        ---After diaxoristiko
    end if
end if
metrou

```

```

14         Vymax2 = VnoteY[Vnoteinsystem[Vloopsystema]] -
end if
if (VnoteY[Vnoteinsystem[Vloopsystema]]-14) <
Vymax2 then
14         Vymax2 = VnoteY[Vnoteinsystem[Vloopsystema]] -
end if
if Vlooptriixo = 1 then
Vtriixosystem = 1
Vstartofbeam3 = Vloopsystema
Vchannelbeam3 = Vchannelbeam3 + 1

channel(Vchannelbeam3).makeScriptedSprite(member("triixonum
ber"),point((VnoteX[Vnoteinsystem[Vloopsystema]]+2), Vymax2
- 16))
sprite(Vchannelbeam3).color = rgb(255,0,0)
sprite(Vchannelbeam3).ink = 1
Vchannelbeam3 = Vchannelbeam3 + 1

channel(Vchannelbeam3).makeScriptedSprite(member("epektasi"
),point((VnoteX[Vnoteinsystem[Vloopsystema]]),
VnoteY[Vnoteinsystem[Vloopsystema]]))
sprite(Vchannelbeam3).rect =
rect(VnoteX[Vnoteinsystem[Vstartofbeam3]]-4, Vymax2 -10,
VnoteX[Vnoteinsystem[Vstartofbeam3]]-3, Vymax2-4)
sprite(Vchannelbeam3).color = rgb(255,0,0)
Vchannelbeam3 = Vchannelbeam3 + 1

channel(Vchannelbeam3).makeScriptedSprite(member("triixobea
m"),point((VnoteX[Vnoteinsystem[Vloopsystema]]),
VnoteY[Vnoteinsystem[Vloopsystema]]))
sprite(Vchannelbeam3).rect =
rect(VnoteX[Vnoteinsystem[Vstartofbeam3]]-4, Vymax2 - 10,
VnoteX[Vnoteinsystem[Vstartofbeam3]]+8, Vymax2-9)
sprite(Vchannelbeam3).color = rgb(255,0,0)
Vchannelbeam3 = Vchannelbeam3 + 1

channel(Vchannelbeam3).makeScriptedSprite(member("epektasi"
),point((VnoteX[Vnoteinsystem[Vloopsystema]]),
VnoteY[Vnoteinsystem[Vloopsystema]]))
sprite(Vchannelbeam3).rect =
rect(VnoteX[Vnoteinsystem[Vstartofbeam3]]+8, Vymax2 - 10,
VnoteX[Vnoteinsystem[Vloopsystema]]+9, Vymax2-4)
sprite(Vchannelbeam3).color = rgb(255,0,0)
else if Vlooptriixo = 2 then
Vtriixosystem = 0
channel(Vchannelbeam3-
3).makeScriptedSprite(member("triixonumber"),point((VnoteX[
Vnoteinsystem[Vloopsystema]-

```

```

1]])+((VnoteX[Vnoteinsystem[Vloopsystema]]-
VnoteX[Vnoteinsystem[Vloopsystema-1]])/2), Vymax2 - 16))
    sprite(Vchannelbeam3-3).color = rgb(255,0,0)
    sprite(Vchannelbeam3-3).ink = 1
    channel(Vchannelbeam3-
2).makeScriptedSprite(member("epektasi"),point((VnoteX[Vnot
einsystem[Vloopsystema]]),
VnoteY[Vnoteinsystem[Vloopsystema]]))
    sprite(Vchannelbeam3-2).rect =
rect(VnoteX[Vnoteinsystem[Vstartofbeam3]]-4, Vymax2 -10,
VnoteX[Vnoteinsystem[Vstartofbeam3]]-3, Vymax2-4)
    sprite(Vchannelbeam3-2).color = rgb(255,0,0)
    channel(Vchannelbeam3-
1).makeScriptedSprite(member("triixobeam"),point((VnoteX[Vn
oteinsystem[Vloopsystema]]),
VnoteY[Vnoteinsystem[Vloopsystema]]))
    sprite(Vchannelbeam3-1).rect =
rect(VnoteX[Vnoteinsystem[Vstartofbeam3]]-4, Vymax2 - 10,
VnoteX[Vnoteinsystem[Vloopsystema]]+8, Vymax2-9)
    sprite(Vchannelbeam3-1).color = rgb(255,0,0)

channel(Vchannelbeam3).makeScriptedSprite(member("epektasi"
),point((VnoteX[Vnoteinsystem[Vloopsystema]]),
VnoteY[Vnoteinsystem[Vloopsystema]]))
    sprite(Vchannelbeam3).rect =
rect(VnoteX[Vnoteinsystem[Vloopsystema]]+8, Vymax2 - 10,
VnoteX[Vnoteinsystem[Vloopsystema]]+9, Vymax2-4)
    sprite(Vchannelbeam3).color = rgb(255,0,0)
    else if Vlooptriixo = 3 then
        channel(Vchannelbeam3-
3).makeScriptedSprite(member("triixonumber"),point((VnoteX[
Vnoteinsystem[Vloopsystema-1]]+2), Vymax2 - 16))
        sprite(Vchannelbeam3-3).color = rgb(0,0,0)
        sprite(Vchannelbeam3-3).ink = 1
        channel(Vchannelbeam3-
2).makeScriptedSprite(member("epektasi"),point((VnoteX[Vnot
einsystem[Vloopsystema]]),
VnoteY[Vnoteinsystem[Vloopsystema]]))
        sprite(Vchannelbeam3-2).rect =
rect(VnoteX[Vnoteinsystem[Vstartofbeam3]]-4, Vymax2 -10,
VnoteX[Vnoteinsystem[Vstartofbeam3]]-3, Vymax2-4)
        sprite(Vchannelbeam3-2).color = rgb(0,0,0)
        channel(Vchannelbeam3-
1).makeScriptedSprite(member("triixobeam"),point((VnoteX[Vn
oteinsystem[Vloopsystema]]),
VnoteY[Vnoteinsystem[Vloopsystema]]))
        sprite(Vchannelbeam3-1).rect =
rect(VnoteX[Vnoteinsystem[Vstartofbeam3]]-4, Vymax2 - 10,
VnoteX[Vnoteinsystem[Vloopsystema]]+8, Vymax2-9)
        sprite(Vchannelbeam3-1).color = rgb(0,0,0)

```

```

channel(Vchannelbeam3).makeScriptedSprite(member("epektasi"
),point((VnoteX[Vnoteinsystem[Vloopsystema]]),
VnoteY[Vnoteinsystem[Vloopsystema]]))
        sprite(Vchannelbeam3).rect =
rect(VnoteX[Vnoteinsystem[Vloopsystema]]+8, Vymax2 -10,
VnoteX[Vnoteinsystem[Vloopsystema]]+9, Vymax2-4)
        sprite(Vchannelbeam3).color = rgb(0,0,0)
        if
(Vaxianotas[Vloopsystema]<>Vaxianotas[Vloopsystema-1]) or
(Vaxianotas[Vloopsystema]<>Vaxianotas[Vloopsystema-2]) then
        sprite(Vchannelbeam3-3).color = rgb(255,0,0)
        sprite(Vchannelbeam3-2).color = rgb(255,0,0)
        sprite(Vchannelbeam3-1).color = rgb(255,0,0)
        sprite(Vchannelbeam3).color = rgb(255,0,0)
        end if
        Vlooptriixo = 0                --Second triixo system
in a row
        Vymax2 = 3000
        else
        Vymax2 = 3000
        Vlooptriixo = 0
        end if
        else
        Vymax2 = 3000
        Vlooptriixo = 0
        if Vmemtriixo[Vloopsystema-1] = 1 then
        Vtriixosystem = 1
        end if
        end if

        --8th and 16th Beams

        if Vtypossprite[Vnoteinsystem[Vloopsystema]] = 4 or
Vtypossprite[Vnoteinsystem[Vloopsystema]] = 5 then
        if Vloopsystema = 2 then
--Beam start
                Vstartofbeam1 = Vloopsystema
                Vchannelbeam1 = Vchannelbeam1 + 1
                Vstartofbeam2 = Vloopsystema
                Vchannelbeam2 = Vchannelbeam2 + 1
        else
                if (Vtypossprite[Vnoteinsystem[Vloopsystema-1]]
<> 4 and Vtypossprite[Vnoteinsystem[Vloopsystema-1]] <> 5)
or (Vtypossprite[Vnoteinsystem[Vloopsystema]-1] = 17) or
(Vtypossprite[Vnoteinsystem[Vloopsystema]-1] > 10 and
Vtypossprite[Vnoteinsystem[Vloopsystema]-1] < 14 and
Vtypossprite[Vnoteinsystem[Vloopsystema]-2] = 17) or
(Vtriixosystem = 1) then
                Vstartofbeam1 = Vloopsystema                --
for 8ths

```

```

Vchannelbeam1 = Vchannelbeam1 + 1
Vymax = VnoteY[Vnoteinsystem[Vloopsystema]] -
14
    --Beam height
    Vychange = 1
    end if
    if Vtypossprite[Vnoteinsystem[Vloopsystema-1]]
<> 5 or Vtypossprite[Vnoteinsystem[Vloopsystema]-1] = 17 or
(Vtypossprite[Vnoteinsystem[Vloopsystema]-1] > 10 and
Vtypossprite[Vnoteinsystem[Vloopsystema]-1] < 14 and
Vtypossprite[Vnoteinsystem[Vloopsystema]-2] = 17) or
(Vtriixosystem = 1) then
    Vstartofbeam2 = Vloopsystema
for 16ths
    Vchannelbeam2 = Vchannelbeam2 + 1
    end if
    end if
    end if
    if Vtypossprite[Vnoteinsystem[Vloopsystema]] = 4 or
Vtypossprite[Vnoteinsystem[Vloopsystema]] = 5 then
        if (VnoteY[Vnoteinsystem[Vloopsystema]]-14) <
Vymax then
14
            Vychange = 1
            end if
            end if
            if Vtypossprite[Vnoteinsystem[Vloopsystema]] <> 4
and Vtypossprite[Vnoteinsystem[Vloopsystema]] <> 5 then
                Vymax = 3000
                end if

                --Alagi grafikon

                if Vloopsystema > 2 then
                    if Vtypossprite[Vnoteinsystem[Vloopsystema-1]] =
4 or \
                        Vtypossprite[Vnoteinsystem[Vloopsystema-1]] = 5
then--an to proigoumeno einai 8o i 16o
                            if Vtypossprite[Vnoteinsystem[Vloopsystema]] =
4 or Vtypossprite[Vnoteinsystem[Vloopsystema]] = 5 then --
kai afto einai 8o i 16o
                                --Draw 8th Beams

channel(Vchannelbeam1).makeScriptedSprite(member("beam"),po
int((VnoteX[Vnoteinsystem[Vstartofbeam1]]+1),
VnoteY[Vnoteinsystem[Vstartofbeam1]]))
                                sprite(Vchannelbeam1).rect =
rect(VnoteX[Vnoteinsystem[Vstartofbeam1]]+1, Vymax,
VnoteX[Vnoteinsystem[Vloopsystema]]+2, Vymax-4)

```

```

        if
Vlathosdiaxoristiko[Vnoteinsystem[Vloopsystema]] = 1 then
            sprite(Vchannelbeam1).color = rgb(255, 0,
0)
        else
            sprite(Vchannelbeam1).color = rgb(0, 0, 0)
        end if
        --Draw 16th Beams
        if Vtypossprite[Vnoteinsystem[Vloopsystema]]
= 5 then
            if Vstartofbeam2 <> Vloopsystema then
--Long beam

channel(Vchannelbeam2).makeScriptedSprite(member("beam"),po
int((VnoteX[Vnoteinsystem[Vstartofbeam2]]+1),
VnoteY[Vnoteinsystem[Vstartofbeam2]]))
            sprite(Vchannelbeam2).rect =
rect(VnoteX[Vnoteinsystem[Vstartofbeam2]]+1, Vymax + 2,
VnoteX[Vnoteinsystem[Vloopsystema]]+2, Vymax+6)
            else
--Left short beam
                if Vstartofbeam1 < Vstartofbeam2 then

channel(Vchannelbeam2).makeScriptedSprite(member("beam"),po
int((VnoteX[Vnoteinsystem[Vstartofbeam2]]+1),
VnoteY[Vnoteinsystem[Vstartofbeam2]]))
                    sprite(Vchannelbeam2).rect =
rect(VnoteX[Vnoteinsystem[Vloopsystema]]-5, Vymax + 2,
VnoteX[Vnoteinsystem[Vstartofbeam2]]+1, Vymax+6)
                    end if
                end if
            else
--Right short beam
                if Vtypossprite[Vnoteinsystem[Vloopsystema-
1]] = 5 and Vloopsystema-1 = Vstartofbeam2 then

channel(Vchannelbeam2).makeScriptedSprite(member("beam"),po
int((VnoteX[Vnoteinsystem[Vstartofbeam2]]+1),
VnoteY[Vnoteinsystem[Vstartofbeam2]]))
                    sprite(Vchannelbeam2).rect =
rect(VnoteX[Vnoteinsystem[Vstartofbeam2]]+1, Vymax + 2,
VnoteX[Vnoteinsystem[Vstartofbeam2]]+8, Vymax+6)
                    end if
                end if
            if
Vlathosdiaxoristiko[Vnoteinsystem[Vloopsystema]] = 1 then
                sprite(Vchannelbeam2).color = rgb(255, 0,
0)
            else
                sprite(Vchannelbeam2).color = rgb(0, 0, 0)
            end if

```

```

--Draw tetarta & epektaseis
repeat with Vloopbeams = Vstartofbeam1 to
Vloopsystema
    if Vstartofbeam1 <> Vloopsystema then
        channel(99 +
Vnoteinsystem[Vloopbeams]).makeScriptedSprite(member("tetar
to"), point((VnoteX[Vnoteinsystem[Vloopbeams]]-4),
VnoteY[Vnoteinsystem[Vloopbeams]]))
        sprite(99 +
Vnoteinsystem[Vloopbeams]).ink = 1
        if Vychange = 1 then
            repeat with Vloopbeams2 = 551 to
Vchannelbeam2
                if (sprite(Vloopbeams2).LocH) <
(VnoteX[Vnoteinsystem[Vloopbeams]]) and
(sprite(Vloopbeams2).LocH) >
(VnoteX[Vnoteinsystem[Vloopbeams-1]]) then
                    sprite(Vloopbeams2).locV = Vymax +
4
                    end if
                end repeat
            end if
        end if
    end if

channel(400+Vloopbeams).makeScriptedSprite(member("epektasi
"),point((VnoteX[Vnoteinsystem[Vloopbeams]]+1),
VnoteY[Vnoteinsystem[Vloopbeams]]))
    sprite(400+Vloopbeams).rect =
rect(VnoteX[Vnoteinsystem[Vloopbeams]]+1, Vymax,
VnoteX[Vnoteinsystem[Vloopbeams]]+2,
VnoteY[Vnoteinsystem[Vloopbeams]])
    if
Vlathosdiaxoristiko[Vnoteinsystem[Vloopsystema]] = 1 then
        sprite(400+Vloopbeams).color = rgb(255,
0, 0)
    else
        sprite(400+Vloopbeams).color = rgb(0, 0,
0)
    end if
end repeat
Vychange = 0
end if
end if
end if
end repeat

```

η) Το κουμπί αλλαγής οργάνου :

```

on mouseUp me
  if VrythmiseisON = 0 then
    if Vapadisi = 0 or Vintitiate = 0 then
      sound(4).play(member("knob change"))

      Vrotate = Vrotate + 90
      sprite(53).rotation = Vrotate

      VinstrumentINT = VinstrumentINT + 1

      case VinstrumentINT of
        1: Vinstrument = " Piano"
           Vrotate = 0
        2: Vinstrument = " Trumpet"
        3: Vinstrument = " Violin"
        4: Vinstrument = " Flute"
           VinstrumentINT = 0
      end case

    end if
  end if
end

```

θ) Η αναπαραγωγή της μελωδίας που έχει γράψει στο πεντάγραμμο ο χρήστης στην άσκηση «Dictee» :

```

on mousedown me
  if VrythmiseisON = 0 and Vmemsystima > 1 then
    sprite(84).member = member("anaparagogi apadisis
button2")
    VpanokoubiaON = 1
    if Vpcplayscale = 0 then
      if Vmemsystima > 1 then
        sound(2).stop()
        sound(1).stop()
        sound(2).setPlayList([])
        sound(1).setPlayList([])
        repeat with n = 2 to Vmemsystima
          if Vmempausi[n] = 1 then
            Vtonikotita[n] = "silence"
          else

```



```

        Vtonikotita[n] = Vmemnota[n] & Vinstrument
    end if
    Vaxianotas[n] = (60000/Vtempo)*Vmemaxia[n]

    sound(2).queue([#member:member(Vtonikotita[n]),#endtime:Vaxianotas[n]])
    end repeat
    Vclickcounter = 4
    repeat with n = 1 to Vmetrol*Vclicktimes
        if Vclickcounter = Vmetrol then
            Vclickcounter = 1

    sound(1).queue([#member:member("metronome1"),#endtime:((60000/Vtempo)*Vmetro2clickvalue)])
        else
            Vclickcounter = Vclickcounter + 1

    sound(1).queue([#member:member("metronome2"),#endtime:((60000/Vtempo)*Vmetro2clickvalue)])
        end if
    end repeat
    end if
end if--VrythmiseisON
end

```

ι) Η αλλαγή του κέρσορα σε χέρι όταν περνάει πάνω από κάποιο κουμπί :

```

on mouseenter me
    if VrythmiseisON = 0 then
        cursor 280
    end if

end mousenter me

on mouseleave me
    cursor 0
end mouseleave me

```

ια) Η εμφάνιση των οπτικών εφέ στην σελίδα του κεντρικού μενού (καταρράχτης & νότες με fade in – fade out) :

```

n = random(11)
case n of
  1: Vzografianotas = "olokliro trans"
  2: Vzografianotas = "miso trans"
  3: Vzografianotas = "tetarto trans"
  4: Vzografianotas = "ogdoo trans"
  5: Vzografianotas = "dekatoekto trans"
  6: Vzografianotas = "tetarto pausi trans"
  7: Vzografianotas = "ogdoo pausi trans"
  8: Vzografianotas = "dekatoekto pausi trans"
  9: Vzografianotas = "aneresi trans"
  10: Vzografianotas = "diesi trans"
  11: Vzografianotas = "yfesi trans"
end case

if sprite(11).blend < 100 and Vmenublink[1] = 1 then
  sprite(11).blend = sprite(11).blend + 1
else if sprite(11).blend = 0 then
  n = random(100)
  if n = 100 then
    n = random(700)
    n2 = random(100)

channel(11).makeScriptedSprite(member(Vzografianotas),
point(n+20, n2+20))
  sprite(11).blend = 1
  Vmenublink[1] = 1
  end if
else
  sprite(11).blend = sprite(11).blend - 1
  Vmenublink[1] = 0
  end if

if sprite(12).blend < 100 and Vmenublink[2] = 1 then
  sprite(12).blend = sprite(12).blend + 1
else if sprite(12).blend = 0 then
  n = random(500)
  if n = 500 then
    n = random(700)
    n2 = random(500)

channel(12).makeScriptedSprite(member(Vzografianotas),
point(n+20, n2+20))
  sprite(12).blend = 1
  Vmenublink[2] = 1
  end if
else
  sprite(12).blend = sprite(12).blend - 1
  Vmenublink[2] = 0
  end if

```

```

if sprite(13).blend < 100 and Vmenublink[3] = 1 then
    sprite(13).blend = sprite(13).blend + 1
else if sprite(13).blend = 0 then
    n = random(500)
    if n = 500 then
        n = random(700)
        n2 = random(500)

channel(13).makeScriptedSprite(member(Vzografianotas),
point(n+20, n2+20))
    sprite(13).blend = 1
    Vmenublink[3] = 1
    end if
else
    sprite(13).blend = sprite(13).blend - 1
    Vmenublink[3] = 0
end if

if sprite(14).blend < 100 and Vmenublink[4] = 1 then
    sprite(14).blend = sprite(14).blend + 1
else if sprite(14).blend = 0 then
    n = random(500)
    if n = 500 then
        n = random(700)
        n2 = random(500)

channel(14).makeScriptedSprite(member(Vzografianotas),
point(n+20, n2+20))
    sprite(14).blend = 1
    Vmenublink[4] = 1
    end if
else
    sprite(14).blend = sprite(14).blend - 1
    Vmenublink[4] = 0
end if

if sprite(15).locV = 736 then
    channel(16).makeScriptedSprite(member("kataraktis1"),
point(547, -726))
    sprite(16).ink = 1
end if
if sprite(16).locV = 736 then
    channel(15).makeScriptedSprite(member("kataraktis2"),
point(547, -734))
    sprite(15).ink = 1
end if
sprite(15).locV = sprite(15).locV + 2
sprite(16).locV = sprite(16).locV + 2

```

```
go to the frame

if sprite(18).locV = 736 then
    channel(17).makeScriptedSprite(member("kataraktis2"),
point(572, -734))
    sprite(17).ink = 1
end if
if sprite(17).locV = 736 then
    channel(18).makeScriptedSprite(member("kataraktis1"),
point(572, -726))
    sprite(18).ink = 1
end if
sprite(17).locV = sprite(17).locV + 1
sprite(18).locV = sprite(18).locV + 1
go to the frame

if sprite(20).locV = 736 then
    channel(19).makeScriptedSprite(member("kataraktis2"),
point(597, -752))
    sprite(19).ink = 1
end if
if sprite(19).locV = 736 then
    channel(20).makeScriptedSprite(member("kataraktis3"),
point(597, -752))
    sprite(20).ink = 1
end if
sprite(19).locV = sprite(19).locV + 3
sprite(20).locV = sprite(20).locV + 3
go to the frame
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΞΕΝΟΓΛΩΣΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- Blass, Andreas; Gurevich, Yuri (2003). Algorithms A Quest for Absolute Definitions.
- Gurevich, Y. Sequential Abstract State Machines Capture Sequential Algorithms. *ACM Transactions on Computational Logic*, Vol 1, no 1 (July 2000), pages 77–111
- Iverson K.E., "Notation as a tool of thought", *Communications of the ACM*, 23: 444-465 (August 1980).
- Keenan, J.P., Thangaraj, V., Halpern, A.R. & Schlaug, G. (2001). Absolute pitch and planum temporal. *Neuroimage*, 14(6): 1402-1408
- Levitin, D.J. (2005). Absolute pitch: perception, coding, and controversies. *Trends in Cognitive Sciences*, 9(1): 26:33.
- Miyazaki, K. & Rakowski, A. (2002). Recognition of notated melodeiw by possessors and nonpossessors of absolute pitch. *Perception & Psychophysics*. 64: 1337-1345.
- Parncutt, R. & Levitin, D.J. (2000). Absolute pitch. *New Grove Dictionary of Music and Musicians*. Macmillan Publishers Ltd.
- Sergeant, D.C. & Roche, S. (1973). Perceptual Shifts in the Auditory Information Processing of Young Children. *Psychology of Music*, i/1: 39-48.
- Wynn, V.T. (1992). Absolute pitch revisited. *British Journal of Psychology*, 83(1): 129-131.

ΕΛΛΗΝΟΓΛΩΣΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- Κόνιαρη, Δ. (2005). Απόλυτη ακοή: σύντομη ανασκόπηση στον μύθο και την πραγματικότητα.. *Μουσικοπαιδαγωγικά*. 2: 42-49

ΕΓΧΕΙΡΙΔΙΑ ΛΟΓΙΣΜΙΚΟΥ

Macromedia, Inc. (2004). Macromedia Director MX 2004 : Getting Started With Director. San Francisco.

Macromedia, Inc. (2004). Macromedia Director MX 2004 : Scripting Reference” , San Francisco.

Macromedia, Inc. (2004). Macromedia Director MX 2004 : Using Director” , San Francisco.

ΑΝΑΦΟΡΕΣ ΙΣΤΟΣΕΛΙΔΩΝ

<http://www.earmaster.com/eartraining.htm>

http://en.wikipedia.org/wiki/Absolute_pitch#Genetics_and_absolute_pitch

http://en.wikipedia.org/wiki/Ear_training

http://en.wikipedia.org/wiki/Computer_programming

<http://en.wikipedia.org/wiki/Algorithm>

http://en.wikipedia.org/wiki/Lingo_%28programming_language%29

http://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%AC%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1_%CF%81%CE%BF%CE%AE%CF%82

<http://www.sci.brooklyn.cuny.edu/~lori/facdev/director/MDintro.htm>

http://en.wikipedia.org/wiki/Adobe_Director

http://en.wikipedia.org/wiki/Graphical_user_interface