

Α.Τ.Ε.Ι. ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΟΥΣΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
ΚΑΙ ΑΚΟΥΣΤΙΚΗΣ

**ΜΕΛΕΤΗ – ΚΑΤΑΣΚΕΥΗ ΔΕΙΓΜΑΤΩΝ ΑΠΟ
ΚΑΝΟΝΑΚΙ
ΓΙΑ ΕΦΑΡΜΟΓΕΣ SAMPLER – SYNTHESISER**



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΤΟΥ ΣΠΟΥΔΑΣΤΗ
ΑΓΓΕΛΑΚΟΠΟΥΛΟΥ ΣΤΑΘΗ**

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΚΕΦΑΛΟΓΙΑΝΝΗΣ ΝΙΚΟΣ

ΡΕΘΥΜΝΟ 2007

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ

2. ΚΑΝΟΝΑΚΙ

2.1 Ιστορικά στοιχεία

2.2 Κατασκευή

2.3 Τρόπος παιξίματος - Εκφραστικότητα

3. AUDIO SAMPLING

3.1 Ιστορικά στοιχεία - Χρηση

4. ΔΗΜΙΟΥΡΓΙΑ ΒΙΒΛΙΟΘΗΚΗΣ ΔΕΙΓΜΑΤΩΝ

4.1 Διαδικασία Ηχογράφησης του οργάνου

4.2 Ψηφιακή Επεξεργασία

4.3 Δημιουργία Βιβλιοθήκης δειγμάτων

4.4 Προβλήματα που παρουσιάστηκαν - Τρόποι αντιμετώπισης

5. ΗΧΗΤΙΚΟ ΑΠΟΤΕΛΕΣΜΑ

5.1 Σύγκριση ηχητικού αποτελεσματος του sampler με τον αυθεντικό ήχο του οργάνου

5.2 Σύγκριση των τρόπων χρήσης του sampler – Midi προγραμματισμός και Keyboard Controller

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

7. ΒΙΒΛΙΟΓΡΑΦΙΑ

1. ΕΙΣΑΓΩΓΗ

Η εργασία αυτή πραγματεύεται την δημιουργία μίας βιβλιοθήκης δειγμάτων ενός οργάνου που χρησιμοποιεί μικροδιαστήματα και την αναπαραγωγή τους μέσω ενός προγράμματος sampler με πειστικό τρόπο.

Το όργανο το οποίο επιλέχθηκε για αναπαραγωγή είναι το κανονάκι. Θα παραθέσω στη συνέχεια ιστορικά στοιχεία για το κανονάκι και θα επεκταθώ στον τρόπο κατασκευής και παιξίματος του οργάνου.

Θα παραθέσω επίσης ιστορικά στοιχεία σχετικά με την διαδικασία του sampling στην οποία βασίζεται η συγκεκριμένη εργασία.

Θα αναλύσω διεξοδικά τον τρόπο ηχογράφησης των δειγμάτων καθώς και την μετέπειτα ψηφιακή τους επεξεργασία. Θα παρουσιάσω τον τρόπο δημιουργίας της βιβλιοθήκης δειγμάτων μέσα από το πρόγραμμα sampler και θα αναφερθώ στον τρόπο επίλυσης των προβλημάτων που παρουσιάστηκαν.

Στο τέλος θα συγκρίνω ηχητικά το αποτέλεσμα ώστε να προβώ στα απαραίτητα συμπεράσματα.

Σκοπός της εργασίας αυτής είναι η αναπαραγωγή του οργάνου μας μέσω του sampler να γίνεται με όσο το δυνατόν περισσότερη αληθοφάνεια.

2. ΚΑΝΟΝΑΚΙ

2.1 Ιστορικά Στοιχεία

Το κανονάκι παλιότερα, κυρίως στους μεσαιωνικούς χρόνους, ήταν γνωστό με την ονομασία "Ψαλτήριο". Οι αρχές του ψαλτηρίου ανιχνεύονται στον ασιατικό χώρο, πολλούς αιώνες πριν από τους αρχαιοελληνικούς κλασικούς χρόνους.

Στην αρχαία Ελλάδα, από πολλούς συγγραφείς έχουμε μαρτυρίες για μουσικά όργανα πιθανόν του τύπου του ψαλτηρίου, με διάφορες ονομασίες. Θα τις συναντήσουμε σε κείμενα του Αριστοτέλη (Πρόβλημα, 19.23,2), του Θεόφραστου (Περί φυτικών ιστοριών, 5.7,6), του Αθηναίου, αλλά και πολλών άλλων συγγραφέων. Δεν υπάρχουν όμως εικονογραφημένες μαρτυρίες, για το λόγο αυτό μόνο υποθέσεις έχουν γίνει έως σήμερα για τη σχέση του αρχαιοελληνικού ψαλτηρίου με το κανονάκι. Αντίθετα, στους βυζαντινούς και μεταβυζαντινούς χρόνους, τα ιστορημένα χειρόγραφα και οι τοιχογραφίες των εκκλησιών έχουν πολλές πληροφορίες για το ψαλτήριο σε σχήμα τριγώνου ή τραπεζίου, και τον τρόπο με τον οποίο κρατιέται και παίζεται.



Εκτεταμένες αναφορές στο ψαλτήριο έχουν υπάρξει και σε ένα σκανδιναβικό χρονικό του 12ου αιώνα, που αναφερόταν σε μια γιορτή που είχε γίνει στον Ιππόδρομο της Κωνσταντινούπολης, όπως επίσης και σε γιορτές πλουσίων οικογενειών, όπως για παράδειγμα στα κείμενα του Γάλλου Φιλίπ Ντι Φρεσν-Καναί (Philippe du Fresne-Canaye) που μνημόνευε τον υπέροχο ήχο μιας «ελληνικής άρπας» στο γάμο πλούσιου εμπόρου στο Πέραν της Κωνσταντινούπολης, το 1573.

Στους μεσαιωνικούς χρόνους, το κανονάκι ήταν γνωστό περισσότερο ως Ψαλτήριο ή Ψαλτήρι, αν και αργότερα, αντικαταστάθηκε από τον όρο «Qanun», μια λέξη που κατά πάσα πιθανότητα προκύπτει από την εξάπλωση του Ισλαμισμού προς τη Δύση και την επαφή των Αράβων και των Τούρκων με τους ευρωπαϊκούς πολιτισμούς. Από τον 12ο ως τον 15ο αιώνα παιζόταν όρθιο, όπως η άρπα, και αργότερα οριζόντια όπως σήμερα.

Κατά τον Galpin, το πολύχορδο ψαλτήριο πέρασε από την Μικρά Ασία στη Μεσοποταμία και την Αίγυπτο και επανήλθε στην Ελλάδα. (Text Book of European musical Instruments, London 1937).

Το κανονάκι γνώρισε ποικίλες ονομασίες, ανάλογα με τους τόπους που το φιλοξένησαν:

«Qanun», «Kanun», «Canon», «Micanon», «Κανονάκι», «Kin», «Kinnor», «Τρίγωνον», «Ψαλτήριο», «Μάγαδης», «Σιμίκιον», «Νόμος», «Σύστημα Κανόνων Θεωρίας», «Σύστημα Κανόνων Δεξιότητων», «ο Μονόχορδος Κανόνας του Πυθαγόρα και του Ευκλείδη»... είναι λίγες μόνο από τις γνωστές μέχρι σήμερα ονομασίες του.

Για την ονομασία qanun υπάρχουν δύο εκδοχές:

- Κατά τον H.G. Farmer το κανονάκι είναι αραβικό όργανο (The mediaeval psaltery in the orient).

- Κατά τον A.H.Stranways η ονομασία προέρχεται από την ελληνική λέξη κανών, ονομασία που ο Ευκλείδης χρησιμοποιεί για ένα μονόχορδο μουσικό όργανο το 300 πχ.

Σημειώνω πως ο Farmer θεωρεί τον κανόνα μονόχορδο μουσικό όργανο με το οποίο ο Πυθαγόρας έκανε πειράματα, με σκοπό να εξακριβώσει τους μουσικούς φθόγγους ανάλογα με το μήκος και το πάχος της χορδής. Δέχεται επίσης την ελληνική καταγωγή της λέξης qanun στο λεξιλόγιο των Αράβων τον 10ο αιώνα.

Οι παλαιοί Άραβες ιστορικοί θεωρούν ως εφευρέτη του οργάνου τον Hakim-i-sehir (Ebu Nasir Farabi) που έζησε από το 870 - 950 μ.Χ. Το πρώτο αυτό όργανο δεν διέθετε κινητούς καβαλάρηδες, δηλαδή μαντάλια (μανταλάκια).

Ο αντίχειρας του αριστερού χεριού χειριζόταν τη ζητούμενη χορδή αλλάζοντας το μήκος της για την παραγωγή της ζητούμενης νότας. Από καταγραφή του γάλλου αξιωματικού Villoteau στην Αίγυπτο το 1792, αναφέρεται σύστημα από μικρά μεταλλικά εξαρτήματα που αλλάζουν το κούρντισμα και ονομάζονται σιλλετ. Ωστόσο το κανονάκι απαντάται και χωρίς μανταλάκια εκατό χρόνια αργότερα όπου ο Τούρκος συνθέτης Hdji Arif Bey έπαιζε χωρίς αυτά. Αναφορά πρέπει να γίνει και στον Νίκο Στεφανίδη: γεννημένος στο Ικόνιο της Μικράς Ασίας. Τοποθετεί σε διαδρόμους ξύλινα βαρελάκια που με την μετακίνησή τους αυξομειώνουν την οξύτητα. Το 1900 εμφανίζονται και τα πρώτα μανταλάκια για να μην φθείρεται το δέρμα των δακτύλων.

Τα τελευταία χρόνια το όργανο διατηρήθηκε στην Μικρά Ασία και μεταφέρθηκε στην Ελλάδα το 1922. Στην Αμερική υπήρχε από τις αρχές του αιώνα όπου διασώθηκαν ηχογραφήσεις του Θεόδωρου Κάππου και του Γιώργου Γαρμπή. Αργότερα στην Ελλάδα ηχογραφούνται οι Λάμπρος Σαββαΐδης, Βασίλης Σαχινίδης, Βασίλης Νόνης και τελευταία ο Πέτρος Ταμπούρης.

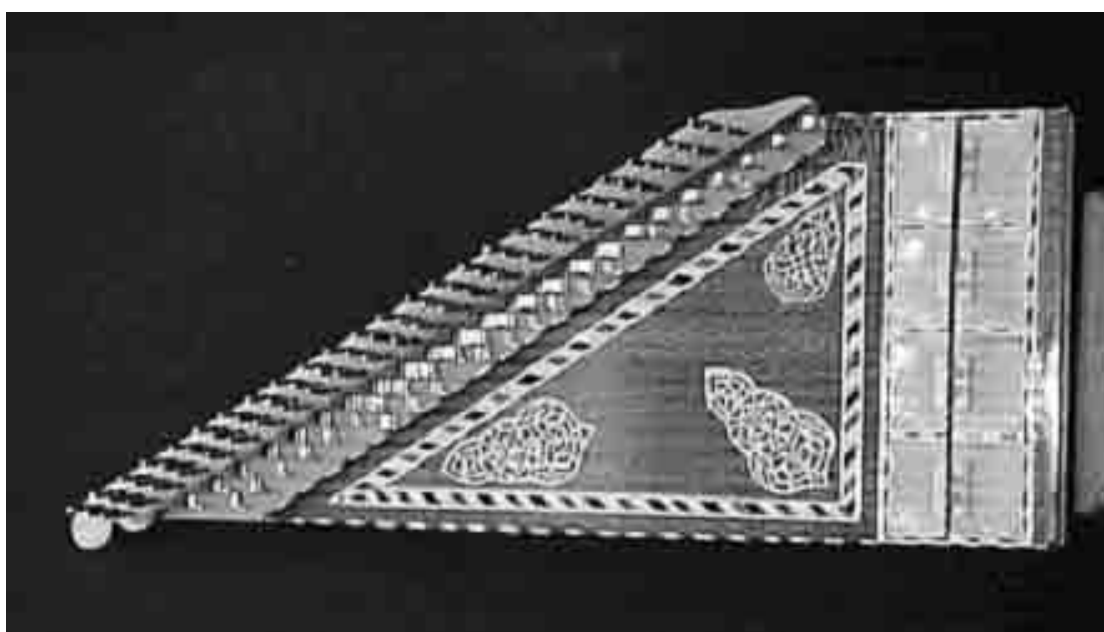
2.2 Κατασκευή

Το κανονάκι έχει σχήμα τραπέζιο. Το ηχείο του έχει μήκος 1μ στην μεγάλη βάση και πλάτος 40-50 εκατοστά. (Η δεξιά πλευρά είναι κάθετη προς τις δύο βάσεις). Οι χορδές (εντέρινες ή πλαστικές) αναπτύσσονται παράλληλα προς τις βάσεις.

Κατασκευάζεται από ξύλο, κυρίως σφένδαμο, ενώ το καπάκι από πλάτανο. Ένα τμήμα του, περίπου 15 εκατοστά πλάτος στην δεξιά πλευρά είναι κατασκευασμένο από δέρμα. Στο καπάκι επίσης ανοίγονται μια ή περισσότερες τρύπες, συχνά διακοσμημένες, σε σχήμα αυγού για να εισέρχεται ο ήχος στο τύμπανο.

Το κούρδισμά του είναι διατονικό, και κάθε φθόγγος αποτελείται από τριπλές χορδές σε ταυτοφωνία. Τα κλειδιά κουρδίσματος είναι ξύλινα στην αριστερή πλευρά και χρησιμοποιούμε ένα μεταλλικό κλειδί για το σφίξιμο ή το λασκάρισμα.

Στην αριστερή πλευρά βρίσκονται τα "μανταλάκια", ή μαντάλια, ένα είδος από κινητούς καβαλάρηδες, που με το ανέβασμα ή το κατέβασμά τους υψώνουν ή χαμηλώνουν το ύψος των φθόγγων κατά 1/4 του τόνου (ισούται περίπου με 2-3 μόρια ή κόμματα). Κάθε φθόγγος μπορεί να έχει από ένα μέχρι 5-6 περίπου μανταλάκια. Στην δεξιά πλευρά και πάνω στο δερμάτινο τμήμα βρίσκεται η γέφυρα (καβαλάρης) όπου πατούν οι χορδές πριν τερματίσουν για δέσιμο.



2.3 Τρόπος παιξίματος - Εκφραστικότητα

Το κανονάκι παίζεται κρατημένο πάνω στα πόδια του οργανοπαίκτη (όταν είναι καθιστός φυσικά), με δυο πένες ή νύχια, όπως λέγονται, που προσδένονται στους δείκτες των δυο χεριών με μεταλλικές δακτυλήθρες. Τα νύχια, κατασκευασμένα από πλαστική ύλη, είναι τεχνητά και αποτελούν προέκταση των φυσικών νυχιών (το κανονάκι είναι νυκτό όργανο), ο εκτελεστής με αυτά τσιμπάει με μεγαλύτερη ευκολία και σταθερότητα τις χορδές, συνήθως με το αριστερό χέρι τις χαμηλές και το δεξί τις ψηλότερες. Οι φθόγγοι αποτελούνται από τρεις χορδές ο καθένας, κουρδισμένες όλες στον ίδιο τόνο. Η μελωδική έκταση του οργάνου είναι περίπου τρεισήμισι οκτάβες.



Τρόπος παιξίματος στο κανονάκι με δακτυλήθρες

Οι νότες βρίσκονται σε αλληλουχία μεταξύ τους, δηλ. η μια μετά την άλλη (κάτω οι χαμηλές και προς τα πάνω οι ψηλότερες) και είναι κουρδισμένες στη διατονική κλίμακα. Έτσι ο οργανοπαίκτης έχει μπροστά του την κλίμακα της μελωδίας που παίζει. Ανάλογα πάντα με το είδος της κλίμακας ή και την βάση της μελωδίας, μπορεί να αλλάξει με τα μανταλάκια την οξύτητα των φθόγγων κατά μερικά μόρια ή κατά ένα ημιτόνιο κι έτσι μπορεί να σχηματίζει δεκάδες διαφορετικές κλίμακες. Οι ανατολίτικες κλίμακες που χρησιμοποιούν μόρια και συναντάμε συνήθως στο κανονάκι είναι οι Ουσάκ , Χιτζάζ και Χιτζαζκιάρ.

Το γεγονός ότι το κανονάκι μπορεί να παίζει χρησιμοποιώντας μόρια είναι και το μεγαλύτερο πλεονέκτημα του σε σχέση με τα συγκερασμένα όργανα της δυτικής μουσικής. Μπορεί να παίζει όλα τα διαστήματα και τις υποδιαίρεσεις του τόνου και γι'αυτό το λόγο είναι αρκετά χρήσιμο και στην εκμάθηση της Βυζαντινής Μουσικής.

Ο εκτελεστής μπορεί να παίζει βυζαντινές κλίμακες ή ανατολίτικα μακάμια με μεγάλη ευκολία καθώς κουρδίζει από πριν το όργανο σύμφωνα με την κλίμακα που θέλει να παίζει. Έχει όμως και την δυνατότητα να κατεβάσει τον τόνο την ώρα που παίζει, κάτι που απαιτεί αρκετή δεξιοτεχνία αλλά προσδίδει έναν ξεχωριστό τρόπο έκφρασης και συναντάται συχνά σε πολλά κομμάτια.

Μια ακόμα ξεχωριστή χρήση για το μανταλάκι που συναντάμε συχνά είναι η πολύ γρήγορη εναλλαγή της θέσης του πάνω και κάτω ενώ χτυπάμε την χορδή, κάτι που δίνει στην νότα έναν «τρεμουλιαστό» ήχο που παραπέμπει σε vibrato.

3. AUDIO SAMPLING

3.1 Ιστορικά στοιχεία - Χρήση

Η χρήση των samplers κινείται σε δύο κατευθύνσεις. Προς την αναπαραγωγή ήχων φυσικών μουσικών οργάνων και προς την αναπαραγωγή αυτούσιων μουσικών φράσεων που προέρχονται από πρωτότυπες ηχογραφήσεις. Στην πρώτη περίπτωση που μας απασχολεί ιδιαίτερα το sampler είναι ένα εργαλείο στα χέρια του μουσικού, καθώς μπορεί πλέον να παίζει σ' ένα κλαβιέ το μέρος ενός κόντρα μπάσου με τον ίδιο σχεδόν ήχο με αυτόν του φυσικού οργάνου.

Οι Περισσότεροι έχουν την εντύπωση ότι το sampling είναι μια «εφεύρεση» της δεκαετίας του '90. Στην πραγματικότητα στην δεκαετία αυτή τα samplers εδραίωσαν την παρουσία τους στη μουσική βιομηχανία, καθώς η χρήση τους έγινε ιδιαίτερα διαδεδομένη.

Η αληθινή ιστορία του sampling αρχίζει πολλές δεκαετίες πίσω, ουσιαστικά στη δεκαετία του 1950. Εκείνη την εποχή η λέξη “sampling” δεν υπήρχε καν ως λήμμα στα λεξικά της αγγλικής γλώσσας. Όμως κάποιοι πρωτοπόροι, σαν τον Χάρυ Τσάμπερλιν από την Καλιφόρνια, είχαν αρχίσει ήδη τις πρώτες προσπάθειες με στόχο να προσφέρουν στους μουσικούς ένα ηλεκτρονικό μουσικό όργανο με πλήκτρα που θα παρήγαγε ήχους αυθεντικών φυσικών οργάνων και «έτοιμους» ρυθμούς, από προηχογραφημένες ταινίες μαγνητοφώνου.

Όσο κι αν για τα σημερινά δεδομένα της ψηφιακής τεχνολογίας το εγχείρημα του Τσάμπερλιν, φαντάζει παλαιολιθικό, τα όργανα που έφτιαξε αυτός ο πρωτοπόρος Καλιφορνέζος, το Rhythmate M1, M2 και το Musicmaster και τα οποία βγήκαν τελικά στο εμπόριο κάπου στις αρχές της δεκαετίας του 1960, όχι μόνο λειτούργησαν αλλά απετέλεσαν τη βάση για ένα άλλο πολύ επιτυχημένο ηλεκτρονικό μουσικό όργανο εκείνης της εποχής, προάγγελο των σημερινών samplers: το Mellotron.

Τα Mellotrons κατασκευάζονταν σε όλη τη διάρκεια της δεκαετίας του '60 ως και τις αρχές περίπου του '80 από την αγγλική φίρμα Streetly Electronics, την εταιρία που είχαν κληρονομήσει από τον πατέρα τους Λέσλι τα αδέρφια, Λες, Φρανκ και Νόρμαν Μπράντλεϋ.

Τα Mellotrons που έβαλαν τη σφραγίδα τους στον ήχο του '60 στην pop και τη rock μουσική, βασίζονταν στην αρχική ιδέα του Τσάμπερλιν, δηλαδή ένα κλαβιέ οργάνου που τα πλήκτρα του ενεργοποιούν τους μηχανισμούς αναπαραγωγής μαγνητοταινιών στις οποίες έχουν ηχογραφηθεί «κρατημένες» νότες φυσικών οργάνων, ή ρυθμικά patterns κρουστών.



Ένα Melotron Mark II του 1965

Η μεγάλη καινοτομία του Mellotron, σε αντίθεση με τα κλασικά συνθεσάιζερ, ήταν ότι εκτός από τους ήχους φυσικών οργάνων που αναπαρήγαγε, ο μουσικός είχε τη δυνατότητα ν' αλλάζει όλους του ήχους του οργάνου, αλλάζοντας απλά τις μαγνητοταινίες.

Το sampling δεν ήταν ακόμη εφικτό, αλλά ο χρήστης μπορούσε να στείλει στην κατασκευάστρια εταιρία τις ταινίες με τους ήχους που είχε ηχογραφήσει στο σπίτι του και οι τεχνικοί να του τις μεταφέρουν στο format των ταινιών του οργάνου.

Το μεγάλο μειονέκτημα των Mellotrons ήταν το μοτέρ που έδινε κίνηση στους οδηγούς των ταινιών. Όταν ο μουσικός προσπαθούσε να παίξει ένα «μεγάλο» ακόρντο, με έξι η επτά νότες, το μοτέρ δεν είχε την απαιτούμενη ροπή για να δώσει ικανοποιητική ισχύ σε όλους τους οδηγούς, με συνέπεια να πέφτει η ταχύτητα αναπαραγωγής των ταινιών με τα γνωστά καταστροφικά αποτελέσματα στην τονικότητα των μουσικών φθόγγων να κάνουν την εμφάνιση τους,

Τα Mellotrons ήταν ένας σημαντικός κρίκος στην εξελικτική διαδικασία των samplers. Όντας βαριά, δύσχρηστα, με αναλογικά κυκλώματα και με έναν πολύπλοκο μηχανισμό ηλεκτρομηχανικών διατάξεων που ήθελε συχνή συντήρηση, εξαφανίστηκαν μόλις ανέτειλε η ψηφιακή εποχή και η Streetly Electronics, αφού πέρασε πρώτα σε αμερικανικά χέρια, έκλεισε οριστικά το 1986.

Η σύγχρονη εποχή για τα samplers - με τα σημερινά τεχνικά χαρακτηριστικά τους - άρχισε το 1975 στο Rushcutters Bay της Νέας Νότιας Ουαλίας, στην Αυστραλία. Είναι η χρονιά που ο Peter Vogel και ο Kim Ryrie, ιδρύουν μια εταιρία με στόχο να κατασκευάσουν ένα ψηφιακά ελεγχόμενο ηλεκτρονικό μουσικό όργανο που θα ανταγωνίζονταν τα αναλογικά συνθεσάιζερ της εποχής.

Η αρχική ιδέα των δύο Αυστραλών ήταν να δημιουργήσουν ένα συνθεσάιζερ που θα παρήγαγε αληθοφανείς ήχους φυσικών οργάνων από ηχητικές κυματομορφές, τις οποίες θα δημιουργούσε σε πραγματικό χρόνο ένας μικροεπεξεργαστής, με βάση μία μαθηματική ακολουθία. Τα αποτελέσματα του σχεδίου δεν ήταν καθόλου ικανοποιητικά και έτσι η έρευνα στράφηκε προς νέα κατεύθυνση, με την χρησιμοποίηση «έτοιμων» ηχητικών δειγμάτων (samples) από τα οποία αναπαράγονταν πολύπλοκες κυματομορφές.

Το 1979 οι Vogel και Ryrie ήταν έτοιμοι να παρουσιάσουν το Fairlight CMI (Computer Musical Instrument), το πρώτο πληκτροφόρο ηλεκτρονικό μουσικό όργανο, στο οποίο μπορούσαν να ηχογραφηθούν και να αναπαραχθούν ψηφιακά ήχοι φυσικών οργάνων ή αυτούσιες μουσικές φράσεις και να γίνει περαιτέρω επεξεργασία τους με την βοήθεια κατάλληλου λογισμικού.

Τα πρώτα Fairlight είχαν κακό ήχο, λόγω της 8μπιτης ανάλυσης του ψηφιακού σήματος και της συχνότητας δειγματοληψίας που έφθανε ως τα 24kHz. Η 16μπιτη ανάλυση και δειγματοληψία ως τα 50kHz ήρθαν με τα μοντέλα της σειράς III που παρουσιάστηκαν το 1985.

Τα Fairlight ήταν (και παραμένουν ακόμη και σήμερα) πρωτοποριακά μηχανήματα. Μια από τις μοναδικές καινοτομίες τους ήταν το γεγονός ότι ο χρήστης μπορούσε να σχεδιάσει με το lightpen στο μόνιτορ μια κυματομορφή και το μηχανήμα να την αναπαράγει ως ήχο.



Ένα Fairlight CMI του 1979

Αλλά ούτε και αυτά στάθηκαν ικανά να καθιερώσουν το sampling και να το κάνουν ευρέως γνωστό. Κόστιζαν μια ολόκληρη περιουσία (η τιμή της σειράς III ξεκινούσε από τα 40.000 δολάρια!) και με μια τέτοια τιμή φυσικά δεν απευθύνονταν στον μέσο μουσικό. Το τελευταίο Fairlight κατασκευάστηκε το 1991.

Το 1981 παρουσιάστηκε από την E-mu Systems ο Emulator, ο οποίος επικεντρώθηκε στο sampling (δεν χρησιμοποιούσε Synthesizer και sequencing λειτουργίες για να κρατήσει χαμηλά το κόστος του). Ο Emulator διέυρνε τον κύκλο των μουσικών που είχαν πρόσβαση στην τεχνολογία sampling, αλλά παρέμεινε το προνόμιο των pop αστέρων, των μεγάλων στούντιο και των καλοπληρωμένων session μουσικών.

Αυτά τα πρώτα samplers είχαν και την δυνατότητα looping αλλά παρείχαν ελάχιστες παραμέτρους προς ρύθμιση στον χρήστη. Η εύρεση κάποιας loop που θα μπορούσε να χρησιμοποιήσεις ήταν περισσότερο θέμα τύχης παρά καλής ρύθμισης.

Ομοίως η δυνατότητα multisampling, δηλαδή να τοποθετηθούν τα διαφορετικά δείγματα σε διαφορετικά κλειδιά, ήταν εξαιρετικά περιορισμένη. Το μέγιστο αποδεκτό εύρος που μπορούσε να καλύψει ένα δείγμα μέσω της διαδικασίας του pitch transpose περιοριζόταν σε μερικά ημιτόνια ώστε να μην υπάρχουν μεγάλες αλλοιώσεις στον αρχικό ήχο.

Ένα από τα σημαντικότερα χαρακτηριστικά γνωρίσματα των πιο προσιτών samplers που άρχισαν να εμφανίζονται το 1985-1986 ήταν η χρήση στοιχείων της αναλογικής σύνθεσης όπως τα φίλτρα και τα envelopes. Αυτά επέτρεπαν στον χρήστη να αναδιαμορφώνει την αναπαραγωγή του ήχου και προσέφεραν ποικιλία χρήσεων. Έτσι στα περισσότερα προσιτά Samplers ανεξάρτητα από το bitrate τους (κάποια ακόμα χρησιμοποιούσαν οκτάμπιτο, αν και τα περισσότερα είχαν φτάσει μέχρι τα 12-bit), ανεξάρτητα από το μέγεθος της μνήμης τους (τα πιο πολλά είχαν μισό megabyte μνήμης), και ανεξάρτητα από την πολυφωνία τους, όλα άρχισε να τα χαρακτηρίζει η χρήση φίλτρων και envelopes.

Το αποτέλεσμα ήταν μηχανές που ήταν πολύ πιο ευπροσάρμοστες από τους ακριβότερους αδελφούς τους που μπορεί να είχαν samplerates των 100kHz αλλά των οποίων τα χαρακτηριστικά για την τροποποίηση του ήχου ήταν ακόμα αρκετά περιορισμένα.

Το μεγάλο κόστος της μνήμης RAM ήταν πάντα ένας σημαντικός παράγοντας στο κόστος και τις δυνατότητες των samplers .

Στα τέλη της δεκαετίας του 80 η τιμή της άρχισε να πέφτει εντυπωσιακά. Συγχρόνως, οι ψηφιακές τεχνικές για filtering έγιναν ευρέως γνωστές, και η αναλογική τεχνολογία φίλτρων και envelopes αντικαταστάθηκε από πιο φθηνές και λειτουργικές ψηφιακές λύσεις .

Η πραγματική επανάσταση για το sampling άρχισε τελικά το 1986.

Ήταν η χρονιά που η AKAI, η γιαπωνέζικη φίρμα home electronics, παρουσίασε στην αγορά το S612, το πρώτο sampler για το ευρύ κοινό. Ήταν μικρό, εύχρηστο με 12-bit ανάλυση και κόστιζε μόλις 995 δολάρια. Σχεδόν δύο χρόνια μετά η AKAI παρουσιάζει το S700, αργότερα το S950, το S1000 και καθιερώνει για την sampling ηχογράφηση και αναπαραγωγή τη στάνταρ ανάλυση του CD, 16bit/44,1kHz.

Το παράδειγμα της AKAI ακολούθησαν γρήγορα και άλλες εταιρίες, που κατασκεύαζαν samplers ήδη από τις αρχές του '80 (όπως η αμερικάνικη E-mu) και πολύ σύντομα, καθώς μάλιστα η τεχνολογία αναπτύσσεται, το sampling γίνεται προσιτό στο ευρύ κοινό. Φτάνοντας στην δεκαετία του 90 αυτά που θεωρούνταν κάποτε state of the art μηχανήματα και απευθύνονταν σε μια πολύ ξεχωριστή ομάδα ανθρώπων, έγιναν τελικά εργαλεία καθημερινής χρήσης για τον μέσο μουσικό.

4. ΔΗΜΙΟΥΡΓΙΑ ΒΙΒΛΙΟΘΗΚΗΣ ΔΕΙΓΜΑΤΩΝ

4.1 Διαδικασία ηχογράφησης του οργάνου

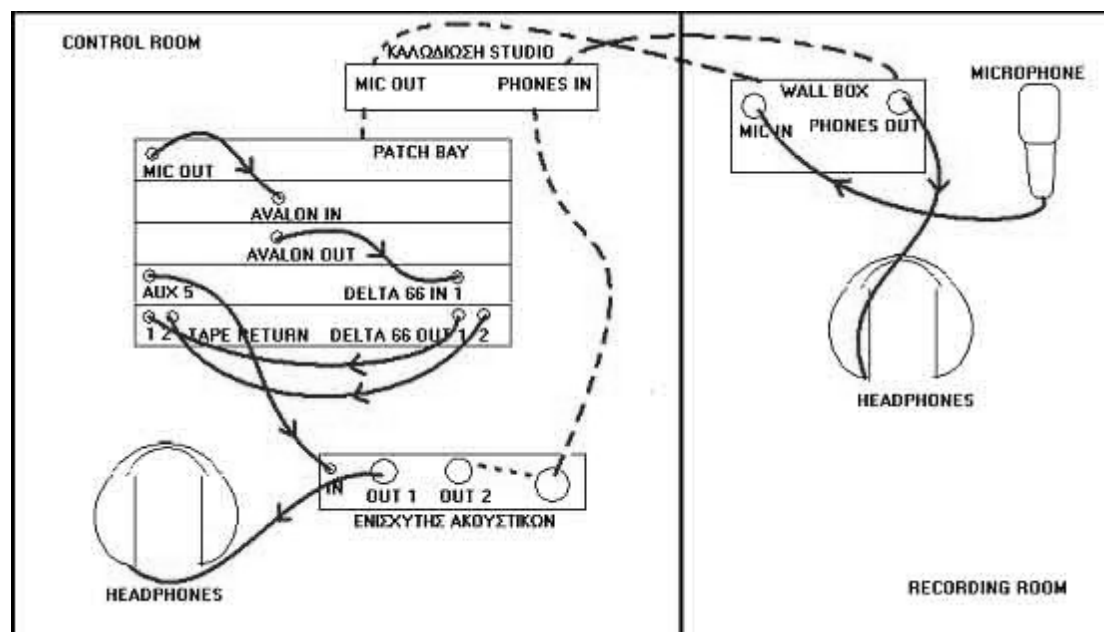
Η ηχογράφηση του οργάνου έγινε στο στούντιο του Τμήματος Μουσικής Τεχνολογίας και Ακουστικής. Τα δείγματα ηχογραφήθηκαν σε ποιότητα 16 bit και sample rate 44100 για να έχουν μικρό μέγεθος και κυρίως για να υπάρχει συμβατότητα των δειγμάτων μας με όλα τα Sampler της αγοράς.

Χρησιμοποίησα ένα υψηλής ευαισθησίας και μεγάλου δυναμικού εύρους πυκνωτικό μικρόφωνο μεγάλου διαφράγματος με πολύ καλή συχνοτική απόκριση που παρέχει και την επιλογή πολικών διαγραμμάτων. Επέλεξα να ηχογραφήσω το όργανο με ένα μικρόφωνο αντί για μια στερεοφωνική διάταξη κυρίως για να παραμείνουν τα δείγματα μικρά σε μέγεθος ώστε να καταναλώνουν λιγότερη μνήμη από τον υπολογιστή που θα καλείται να τα διαχειριστεί. Η στερεοφωνική εικόνα του οργάνου θα δίνεται ευκολότερα πλέον από το πρόγραμμα που θα αναπαράγει τα δείγματα. Δεν χρησιμοποίησα καθόλου compression ή οποιαδήποτε άλλη επεξεργασία στο σήμα καθώς κάτι τέτοιο θα ήταν αντίθετο με την βασική αρχή μιας multisampling ηχογράφησης που θέλει να παίρνουμε όσο το δυνατόν πιο φυσικό τον ήχο από το όργανο που ηχογραφούμε.

Το μικρόφωνο τοποθετήθηκε μπροστά ακριβώς στο κέντρο του οργάνου και μισό μέτρο πιο ψηλά από αυτό με την κάψα του να στοχεύει το όργανο με γωνία περίπου 45 μοιρών. Χρησιμοποίησα το καρδιοειδές πολικό διάγραμμα του μικροφώνου για να μειώσω την ηχογράφηση των ανεπιθύμητων αντηχήσεων από τους πίσω τοίχους και το ταβάνι. Χρησιμοποιώντας το patch bay συνέδεσα το μικρόφωνο στην είσοδο του προενισχυτή μικροφώνου Avalon ο οποίος είναι τάξης A. Ύστερα συνέδεσα την έξοδο του προενισχυτή στην πρώτη είσοδο της κάρτας ήχου που χρησιμοποίησα για την μετατροπή του σήματος από αναλογικό σε ψηφιακό ώστε να ηχογραφηθεί στον υπολογιστή.

Στη συνέχεια συνέδεσα τις εξόδους της κάρτας ήχου στην κονσόλα όπου έστειλα το σήμα για monitoring μέσω των nearfield monitor ηχείων. Για να καλύψω την ανάγκη του μουσικού για monitoring χρησιμοποίησα τον ενισχυτή ακουστικών από όπου έστειλα ένα ποσοστό του σήματος στα ακουστικά του μουσικού. Χρησιμοποίησα και εγώ ένα ζευγάρι ακουστικών που συνέδεσα στον ενισχυτή στο control room για χρήση εναλλακτικού monitoring.

Εδώ τελειώνει η συνδεσμολογία που παρατίθεται σχηματικά παρακάτω.



Συνδεσμολογία. (με διακεκομμένες γραμμές φαίνονται οι καλωδιώσεις που προϋπάρχαν)

Ξεκινώντας την ηχογράφιση ο μουσικός αρχίζει να παίζει δυνατά τις νότες. Ρυθμίζω το Gain στον προενισχυτή ώστε να παίρνω ένα δυνατό σήμα που να μην παραμορφώνει.

Χρησιμοποιώ στο PC στο οποίο θα ηχογραφήσω το Digital Audio Workstation πρόγραμμα Sonar 3 Producer Edition της Cakewalk. Δημιουργώ ένα νέο Project με ρυθμίσεις 16 bit και sample rate 44100 με ένα μονοφωνικό κανάλι και προσέχω να έχω σήμα στην είσοδο αλλά και στην stereo έξοδο του προγράμματος. Ύστερα ρυθμίζω την ένταση monitoring στην κονσόλα και στον ενισχυτή των ακουστικών. Αφού οπλίσω το κανάλι στο DAW τα πάντα είναι έτοιμα για να ξεκινήσει η ηχογράφιση.

Ηχογραφώ κάθε νότα ξεχωριστά περιμένοντας κάθε φορά μέχρι να σβήσει τελείως η ουρά της με τρεις διαφορετικούς τρόπους : με δυνατό χτύπημα , με σιγανό χτύπημα και με χτύπημα κοντά στον καβαλάρη του οργάνου (CD track 1 – 2). Η ηχογράφιση τελειώνει αφού γράψω και ένα μικρό μουσικό παράδειγμα που θα χρησιμεύσει σαν αναφορά στον ήχο του οργάνου.

Τέλος κάνω από το πρόγραμμα εξαγωγή όλων των ήχων του καναλιού με την σειρά που ηχογραφήθηκαν σε ένα ασυμπίεστο αρχείο ήχου τύπου wav που θα με βοηθήσει αργότερα στο να επεξεργαστώ τα δείγματα καλύτερα σε ένα audio editing πρόγραμμα.

Στο σημείο αυτό θέλω να ευχαριστήσω πολύ τον μουσικό Μανόλη Κανακάκη που έπαιξε το κανονάκι για την ηχογράφιση.

Παρακάτω παραθέτω αναλυτικά τον εξοπλισμό που χρησιμοποιήθηκε.

Πυκνωτικό μικρόφωνο : NEUMANN U89 Condenser Mic
Κονσόλα : Audient ASP 8024 HIGH RESOLUTION Mixing Console
Προενισχυτής μικροφώνου : Avalon Pure CLASS A Analog Preamplifier
Ενισχυτής ακουστικών : BEHRINGER POWERPLAY PRO XL
Ακουστικά : Beyer Dynamic DT 990
Ηχεία Monitor : Mayer Sound HM-1 Studio Monitors
Κάρτα ήχου : M-audio Delta 66 PCI
PC ηχογράφησης : Intel Pentium 4 3 GHz - 1 GB RAM
Software : Cakewalk Sonar 3 Producer Edition

4.2 Ψηφιακή επεξεργασία δειγμάτων

Μετά την ηχογράφηση των δειγμάτων είχα στην διάθεση μου ένα μεγάλο αρχείο ήχου wav που περιείχε όλες τις ηχογραφήσεις.

Το επόμενο βήμα μου είναι να επεξεργαστώ κατάλληλα τα δείγματα αυτά ώστε να τα φέρω σε μία μορφή που θα χρησιμοποιήσω ευκολότερα για την αναπαραγωγή τους από το sampler.

Για την επεξεργασία των δειγμάτων χρησιμοποίησα ένα γνωστό audio editor πρόγραμμα της Steinberg , το Wavelab 5.

Κατ αρχήν έπρεπε να προσαρμόσω ξανά την στάθμη των δειγμάτων καθώς κάποια είχαν γραφτεί με μεγαλύτερη και κάποια με μικρότερη ένταση (λόγω αδυναμίας του μουσικού να αναπαράγει πάντα το ίδιο χτύπημα στις χορδές).

Χρησιμοποίησα την λειτουργία Normalize του προγράμματος για να ισοροπήσω τις στάθμες των δειγμάτων στο ανώτατο όριο της πιο δυνατά παιγμένης νότας περίπου στα -5 db. Έτσι οι νότες που πριν ακούγονταν χαμηλότερα τώρα είχαν όλες την σωστή ένταση. Αυτή η διαδικασία έγινε ξεχωριστά για τις τρεις περιπτώσεις χτυπήματος της χορδής.

Μετά έπρεπε να διορθώσω το τονικό ύψος μερικών δειγμάτων. Λόγω της διαφοράς θερμοκρασίας στο στούντιο το κανονάκι ξεκούρδιζε μερικά cents σε κάποιες από τις χορδές του. Για να διορθώσω αυτά τα δείγματα χρησιμοποίησα την λειτουργία Pitch Correction του editor.

Επιλέγοντας κάθε νότα ξεχωριστά χρησιμοποίησα την λειτουργία Find Current Pitch που με πληροφορούσε σχετικά με την συχνότητα και την διαφορά σε cents που υπήρχε από την κοντινότερη νότα. Έτσι γνώριζα το pitch correction που έπρεπε να χρησιμοποιηθεί κάθε φορά και το εφάρμοζα με ενεργοποιημένη την επιλογή Preserve original time , ώστε να μην αλλοιώνεται από την μετατροπή η διάρκεια του δείγματος.

Τελική διαδικασία ήταν το κόψιμο των δειγμάτων και η δημιουργία ξεχωριστών αρχείων ήχου για κάθε νότα.

Στα δείγματα έγινε editing ξεχωριστά όπου κόπηκε το κενό που υπήρχε στην αρχή , ώστε να ξεκινά αμέσως η ατάκα του ήχου από μηδενική ένταση και το κάθε δείγμα τελείωνε όταν η ουρά του ήχου σταματούσε σε μηδενική ένταση.

Δημιουργήθηκαν έτσι 43 αρχεία ήχου wav (ένα για κάθε νότα) για κάθε έναν από του τρεις τρόπους παιξίματος και ονομάστηκαν κατάλληλα ώστε να ξεχωρίζουν μεταξύ τους.

4.3 Δημιουργία βιβλιοθήκης δειγμάτων

Το πρόγραμμα Sampler που επέλεξα για την αναπαραγωγή των δειγμάτων είναι το Kontakt 2 της Native Instruments. Το επέλεξα μεταξύ των άλλων δύο μεγάλων ανταγωνιστών του , Gigastudio 3 και Halion 3 για αρκετούς λόγους. Πρώτον έχει πολύ λιγότερες απαιτήσεις σε υπολογιστική ισχύ από το Gigastudio και είναι πιο απλό στην χρήση του. Άλλος ένας βασικός λόγος που απόρριψα το Gigastudio είναι η απαίτηση ύπαρξης ειδικών GSIF drivers στην κάρτα ήχου του συστήματος για να δουλέψει το πρόγραμμα , κάτι που δεν υποστηρίζουν όλες οι κάρτες ήχου. Τελικά επέλεξα το Kontakt αντί του Halion για δυο λόγους , καλύτερο και πιο ξεκάθαρο περιβάλλον εργασίας και περισσότερες δυνατότητες παραμετροποίησης του κάθε οργάνου και για τον ενσωματωμένο script editor του Kontakt όπου μέσω γραπτού κώδικα μπορεί κανείς να δώσει λύσεις σε οποιοδήποτε πρόβλημα.

Αρχίζοντας την διαδικασία δημιουργίας της βιβλιοθήκης δειγμάτων στο sampler το πρώτο μου βήμα είναι η τοποθέτηση όλων των δειγμάτων στα αντίστοιχα πλήκτρα του εικονικού κλαβιέ και η πρώτη ακρόαση τους χρησιμοποιώντας ένα Midi keyboard για τον έλεγχο του sampler. Από το αποτέλεσμα επιλέγω τα καλύτερα σε ήχο δείγματα (χωρίς τριξίματα κτλ.) που απέχουν 3-4 περίπου ημιτόνια μεταξύ τους.

Αυτά είναι τα δείγματα που θα χρησιμοποιήσω τελικά στο όργανο μου, έτσι ώστε να έχω το καλύτερο ηχητικό αποτέλεσμα χωρίς να χρειαστεί να καταναλώσω μεγάλο μέρος από την μνήμη του υπολογιστή.

Επέλεξα περίπου 16 δείγματα για κάθε τρόπο παιξίματος ώστε να καλύψω όλη την έκταση του οργάνου που είναι 43 ημιτόνια (περίπου 3,5 οκτάβες). Με τον τρόπο αυτό χρησιμοποίησα 4 δείγματα για να καλύψω μία οκτάβα , μια αρκετά καλή τιμή για τα δεδομένα του sampling καθώς ο ήχος της αρχικής νότας δεν θα υποστεί μεγάλες αλλαγές όταν γίνει μετατροπή του τονικού του ύψους για 3 μόνο ημιτόνια.

Για να καταφέρω να κάνω τα δείγματα να καλύπτουν αυτή την έκταση τα ανάθεσα στα αρχικά τους πλήκτρα μέσα από τον Map editor και μετά επέλεξα την περιοχή του κλαβιέ όπου ήθελα να ηχούν , δημιουργώντας έτσι μια ζώνη. Η κάθε ζώνη λοιπόν περιέχει ένα δείγμα αλλά αντιστοιχεί σε 2-3 πλήκτρα πάνω στο κλαβιέ . Δημιουργώντας λοιπόν 16 ζώνες έχω καλύψει όλη την έκταση του οργάνου μου καθώς το sampler αυτόματα θα εφαρμόσει το ανάλογο Pitch Shift στα δείγματά.

Επέλεξα για όλα τα δείγματα τον σωστό τρόπο αναπαραγωγής τους από το sampler. Ο τρόπος αναπαραγωγής ορίστηκε σαν One Shot που σημαίνει ότι με το πάτημα του πλήκτρου θα παίζεται το αντίστοιχο δείγμα μέχρι τέλους, ανεξάρτητα με το πότε θα αφήσω ελεύθερο το πλήκτρο.

Η επόμενη κίνηση ήταν να δημιουργήσω δύο επίπεδα δυναμικής που θα αντιστοιχούν σε διαφορετική σειρά από δείγματα (δυνατό και σιγανό χτύπημα). Έτσι επέλεξα για όλες τις ζώνες που αντιστοιχούν στο δυνατό χτύπημα να ηχούν όταν δέχονται Velocity με τιμές από 110 έως 127 από το Midi κλαβιέ, ενώ για τις ζώνες που αντιστοιχούν στο σιγανό χτύπημα έδωσα όριο τιμών Velocity από 0 έως 109. Έτσι χτυπώντας τα πλήκτρα δυνατά θα ακούω το δυνατό σετ δειγμάτων ενώ όταν χτυπάω με λιγότερη δύναμη θα ακούγονται τα δείγματα που έχουν παιχτεί σιγανά.

Έχει μείνει η επιλογή να ακούω τα δείγματα που έχουν παιχτεί κοντά στον καβαλάρη και έχουν διαφορετική χροιά από τα υπόλοιπα. Για να το πετύχω αυτό χρειάστηκε να επιλέξω όλες τις ζώνες κάθε τρόπου παιχνιδιού και να τις στείλω σε ένα Group. Έτσι δημιούργησα τρία διαφορετικά Group που τα ονόμασα χαρακτηριστικά High Vel, Low Vel (για δυνατό και σιγανό χτύπημα χορδής) και τέλος Close Vel (για χτύπημα της χορδής κοντά στον καβαλάρη). Σε καθένα από αυτά τα group όρισα τιμή πολυφωνίας τις 16 νότες με 400 ms χρόνο fade που σημαίνει ότι με γρήγορο παίξιμο περίπου 20 νότες μπορούν να ηχούν ταυτόχρονα.

Αφού τα δείγματα με χτύπημα κοντά στον καβαλάρη είχαν μόνο μια δυναμική και όριο τιμών velocity από 0 έως 127 έπρεπε να χρησιμοποιήσω έναν midi controller για να εναλλάσσω τους τρόπους παιχνιδιού. Χρησιμοποίησα για αυτόν το σκοπό τον Controller 1 που αντιστοιχεί στην λειτουργία Modulation στο Midi Keyboard. Αυτό έγινε μέσω του Group Start Options, όπου επέλεξα πότε θα ηχεί ή όχι το κάθε group.

Επέλεξα τα group High Vel και Low Vel να ηχούν μόνο όταν ο Controller 1 έχει τιμές από 0 έως 65, ενώ το group Close Vel να ηχεί όταν ο Controller 1 πάρει τιμές από 66 έως 127.

Επόμενη μου κίνηση ήταν να προσομειώσω την λειτουργία του damper pedal στο πιάνο για να έχω την δυνατότητα να κόψω την μεγάλη ουρά της κάθε νότας που θέλω να παίξω.

Για να το πετύχω αυτό δημιούργησα τρία ακριβώς ίδια group με τα παραπάνω και τα ονόμασα High Vel2 , Low Vel2 και τέλος Close Vel2. Σε καθένα από αυτά τα group όρισα τιμή πολυφωνίας τις 2 νότες με 400 ms χρόνο fade που σημαίνει ότι με γρήγορο παίξιμο περίπου 8 νότες μπορούν να ηχούν ταυτόχρονα.

Στα group αυτά όρισα από την επιλογή modulation να εφαρμόζεται σε κάθε νότα ένα envelope ADSR με τιμές attack : 1ms , decay : 750 ms , Sustain : -12db και Release : 1100ms. Έτσι ακούω τις περισσότερες νότες να σβήνουν σε 1200ms περίπου.

Για να επιλέξω ποια ομάδα group θα ηχεί κάθε φορά χρησιμοποίησα τον midi controller 11 που αντιστοιχεί στο damper pedal. Αυτό έγινε μέσω του Group Start Options , όπου επέλεξα τα group High Vel , Low Vel και Close Vel να ηχούν μόνο όταν ο Controller 11 έχει τιμή 0 (όταν δεν πατάω το damper) , ενώ τα group High Vel2 , Low Vel2 και Close Vel2 να ηχούν όταν ο Controller 11 πάρει τιμές από 1 έως 127 (όταν έχω πατημένο το damper pedal).

Στη συνέχεια δημιούργησα πέντε ξεχωριστά group για τα δείγματα που αντιστοιχούν στα πέντε διαφορετικά εφέ που έχω ηχογραφήσει. Τα εφέ αυτά χρησιμοποιούνται συχνά στο κανονάκι και είναι :

1. Το πολύ γρήγορο επαναλαμβανόμενο χτύπημα της χορδής που ονόμασα Tremolo.
2. Η γρήγορη εναλλαγή θέσης μπροστά και πίσω στο μανταλάκι ενώ χτυπάμε τη χορδή που ονόμασα Vibrato.
3. Το γρήγορο παίξιμο 3 νότων (I - III - I) με διαφορά τρίτης μεταξύ τους που ονόμασα Fast.
4. Το κατέβασμα ενός ημιτόνιου σε μια νότα ρίχνοντας σιγά τα μανταλάκια της χορδής που ονόμασα Long down.
5. Το κατέβασμα ενός ημιτόνιου σε μια νότα ρίχνοντας γρήγορα τα μανταλάκια της χορδής που ονόμασα Short down.

Έχοντας στην διάθεση μου τέσσερα δείγματα από κάθε εφέ χρειάστηκε να δημιουργήσω για κάθε group 4 ζώνες με την καθεμία να καλύπτει πέντε η έξι ημιτόνια για να καλύψω τελικά δύο οκτάβες στην κεντρική περιοχή της συνολικής έκτασης του οργάνου.

Για τα δύο πρώτα group , το tremolo και το vibrato , χρησιμοποίησα τον Loop editor για να ορίσω την κατάλληλη περιοχή όπου θα γίνεται το Loop σε κάθε δείγμα.

Χρησιμοποίησα δύο μέρη στο Loop section , στο πρώτο που περιείχε την Loop όρισα σαν τρόπο αναπαραγωγής από το Loop Mode το until release και στο δεύτερο που περιείχε το υπόλοιπο κομμάτι του δείγματος όρισα σαν τρόπο αναπαραγωγής το One Shot.

Έτσι όταν πατάω το πλήκτρο θα παίζει το δείγμα μέχρι το σημείο όπου έχω όρισει το loop , όσο το πλήκτρο παραμένει πατημένο θα συνεχίζει να παίζει το loop ενώ όταν αφήσω το πλήκτρο το loop θα σταματήσει και το δείγμα θα συνεχίσει μέχρι το τέλος του. Για τα τρία υπόλοιπα group όπου δεν χρειαζόταν η λειτουργία Loop επέλεξα ως τρόπο αναπαραγωγής των δειγμάτων το One Shot.

Για να επιλέξω το εφέ που θέλω να ηχεί κάθε φορά χρησιμοποίησα τους controllers 70 και 71. Μέσω του Group Start Options , επέλεξα ξεχωριστά για τα group Tremolo και Vibrato να ηχούν μόνο όταν controller 70 πάρει τιμές από 1 έως 64 και από 65 έως 127 αντίστοιχα. Με την προϋπόθεση ο controller 71 να έχει μηδενική τιμή.

Επέλεξα επίσης ξεχωριστά για τα group Fast , Long down και Short down να ηχούν μόνο όταν controller 71 πάρει τιμές από 1 έως 43 , από 44 έως 86 και από 87 έως 127 αντίστοιχα. Με την προϋπόθεση αυτή την φορά ο controller 70 να έχει μηδενική τιμή.

Για όλα τα group των εφέ επέλεξα μέγιστη πολυφωνία τις 2 φωνές. Σε όλα τα υπόλοιπα group που περιέχουν τις νότες έθεσα την προϋπόθεση να ηχούν μόνο όταν και οι δύο controllers 70 και 71 έχουν μηδενική τιμή.

Επόμενη κίνηση είναι η ανάθεση του Pitch Bend για κάθε group οπού υπάρχουν νότες μέσω της επιλογής modulation. Όρισα τον controller να αλλάζει το pitch και τη μέγιστη και την ελάχιστη διαφορά τονικού ύψους στο ένα ημιτόνιο για χρηστικούς λόγους και για πιο αληθοφανή αποτελέσματα.

Με την χρήση των controllers 7 (Volume) και 10 (Pan) που είναι ρυθμισμένοι από το πρόγραμμα μπορώ να ελέγχω την ένταση και την στερεοφωνική εικόνα του οργάνου μου.

4.4 Προβλήματα που παρουσιάστηκαν – Τρόποι αντιμετώπισης

Αφού ολοκλήρωσα τον σχεδιασμό του οργάνου , δοκίμασα εκτενώς την λειτουργία του χρησιμοποιώντας για τον έλεγχο του ένα midi κλαβιέ.

Ανακάλυψα ένα πρόβλημα στον τρόπο λειτουργίας του έπρεπε να λυθεί. Όταν πάταγα το ίδιο πλήκτρο συνεχόμενα η νότα που είχε αναπαραχθεί προηγουμένως δεν σταματούσε να παίζει με αποτέλεσμα να δημιουργεί ένα βουητό (ειδικά στις χαμηλές νότες που έχουν μεγάλη ουρά) και να καταναλώνει χωρίς λόγο φωνές από την πολυφωνία του οργάνου. Από την στιγμή που αυτό δεν είναι δυνατό να συμβαίνει σε ένα νυκτό όργανο έπρεπε να αντιμετωπιστεί άμεσα.

Ανακάλυψα επίσης πως είναι πρακτικά ακατόρθωτο να μπορέσει να παίζει κανείς μια ανατολίτικη κλίμακα πχ. Ουσάκ χρησιμοποιώντας μόνο το pitch bend για την ρύθμιση των μικροδιαστημάτων. Έπρεπε λοιπόν να βρεθεί ένας λειτουργικός τρόπος κουρδίσματος του οργάνου μας στις κλίμακες αυτές.

Την λύση στα προβλήματα αυτά την έδωσε ο ενσωματωμένος script editor που υπάρχει στο πρόγραμμα. Χρειάστηκε να γράψω μερικά κομμάτια κώδικα στην γλώσσα προγραμματισμού που καταλαβαίνει το πρόγραμμα και να τα εκτελέσω ώστε να πάρω τα επιθυμητά αποτελέσματα.

Ο πρώτος κώδικας που έγραψα σταμάτα την νότα που ήδη ακούγεται όταν η ίδια νότα πατηθεί για να παίζει ξανά. Η διακοπή της νότας που ακούγεται δεν είναι εντελώς απότομη για χάρη φυσικότητας στο άκουσμα αλλά σβήνει με fade out μέσα σε 200ms.

Ακολουθεί ο κώδικας τον οποίο θα εξηγήσω διεξοδικά παρακάτω.

```
on init
  declare $id45r
  declare $id46r
  declare $id47r
  declare $id48r
  declare $id49r
  declare $id50r
  declare $id51r
  declare $id52r
  declare $id53r
  declare $id54r
  declare $id55r
  declare $id56r
  declare $id57r
  declare $id58r
  declare $id59r
  declare $id60r
  declare $id61r
  declare $id62r
  declare $id63r
  declare $id64r
  declare $id65r
  declare $id66r
  declare $id67r
  declare $id68r
  declare $id69r
  declare $id70r
  declare $id71r
  declare $id72r
  declare $id73r
  declare $id74r
  declare $id75r
  declare $id76r
  declare $id77r
  declare $id78r
  declare $id79r
  declare $id80r
  declare $id81r
  declare $id82r
  declare $id83r
  declare $id84r
  declare $id85r
  declare $id86r
  declare $id87r
end on
```

```
on release
  if (%KEY_DOWN[45] = 0)
    $id45r := $EVENT_ID
  end if
  if (%KEY_DOWN[46] = 0)
    $id46r := $EVENT_ID
  end if
  if (%KEY_DOWN[47] = 0)
    $id47r := $EVENT_ID
  end if
  if (%KEY_DOWN[48] = 0)
    $id48r := $EVENT_ID
  end if
```



```
if (%KEY_DOWN[49] = 0)
$id49r := $EVENT_ID
end if
if (%KEY_DOWN[50] = 0)
$id50r := $EVENT_ID
end if
if (%KEY_DOWN[51] = 0)
$id51r := $EVENT_ID
end if
if (%KEY_DOWN[52] = 0)
$id52r := $EVENT_ID
end if
if (%KEY_DOWN[53] = 0)
$id53r := $EVENT_ID
end if
if (%KEY_DOWN[54] = 0)
$id54r := $EVENT_ID
end if
if (%KEY_DOWN[55] = 0)
$id55r := $EVENT_ID
end if
if (%KEY_DOWN[56] = 0)
$id56r := $EVENT_ID
end if
if (%KEY_DOWN[57] = 0)
$id57r := $EVENT_ID
end if
if (%KEY_DOWN[58] = 0)
$id58r := $EVENT_ID
end if
if (%KEY_DOWN[59] = 0)
$id59r := $EVENT_ID
end if
if (%KEY_DOWN[60] = 0)
$id60r := $EVENT_ID
end if
if (%KEY_DOWN[61] = 0)
$id61r := $EVENT_ID
end if
if (%KEY_DOWN[62] = 0)
$id62r := $EVENT_ID
end if
if (%KEY_DOWN[63] = 0)
$id63r := $EVENT_ID
end if
if (%KEY_DOWN[64] = 0)
$id64r := $EVENT_ID
end if
if (%KEY_DOWN[65] = 0)
$id65r := $EVENT_ID
end if
if (%KEY_DOWN[66] = 0)
$id66r := $EVENT_ID
end if
if (%KEY_DOWN[67] = 0)
$id67r := $EVENT_ID
end if
if (%KEY_DOWN[68] = 0)
$id68r := $EVENT_ID
end if
```

```

if (%KEY_DOWN[69] = 0)
$id69r := $EVENT_ID
end if
if (%KEY_DOWN[70] = 0)
$id70r := $EVENT_ID
end if
if (%KEY_DOWN[71] = 0)
$id71r := $EVENT_ID
end if
if (%KEY_DOWN[72] = 0)
$id72r := $EVENT_ID
end if
if (%KEY_DOWN[73] = 0)
$id73r := $EVENT_ID
end if
if (%KEY_DOWN[74] = 0)
$id74r := $EVENT_ID
end if
if (%KEY_DOWN[75] = 0)
$id75r := $EVENT_ID
end if
if (%KEY_DOWN[76] = 0)
$id76r := $EVENT_ID
end if
if (%KEY_DOWN[77] = 0)
$id77r := $EVENT_ID
end if
if (%KEY_DOWN[78] = 0)
$id78r := $EVENT_ID
end if
if (%KEY_DOWN[79] = 0)
$id79r := $EVENT_ID
end if
if (%KEY_DOWN[80] = 0)
$id80r := $EVENT_ID
end if
if (%KEY_DOWN[81] = 0)
$id81r := $EVENT_ID
end if
if (%KEY_DOWN[82] = 0)
$id82r := $EVENT_ID
end if
if (%KEY_DOWN[83] = 0)
$id83r := $EVENT_ID
end if
if (%KEY_DOWN[84] = 0)
$id84r := $EVENT_ID
end if
if (%KEY_DOWN[85] = 0)
$id85r := $EVENT_ID
end if
if (%KEY_DOWN[86] = 0)
$id86r := $EVENT_ID
end if
if (%KEY_DOWN[87] = 0)
$id87r := $EVENT_ID
end if
end on

```

on note

```
    if (%KEY_DOWN[45] = 1)
        fade_out ($id45r, 200000,1)
    end if
    if (%KEY_DOWN[46] = 1)
        fade_out ($id46r, 200000,1)
    end if
    if (%KEY_DOWN[47] = 1)
        fade_out ($id47r, 200000,1)
    end if
    if (%KEY_DOWN[48] = 1)
        fade_out ($id48r, 200000,1)
    end if
    if (%KEY_DOWN[49] = 1)
        fade_out ($id49r, 200000,1)
    end if
    if (%KEY_DOWN[50] = 1)
        fade_out ($id50r, 200000,1)
    end if
    if (%KEY_DOWN[51] = 1)
        fade_out ($id51r, 200000,1)
    end if
    if (%KEY_DOWN[52] = 1)
        fade_out ($id52r, 200000,1)
    end if
    if (%KEY_DOWN[53] = 1)
        fade_out ($id53r, 200000,1)
    end if
    if (%KEY_DOWN[54] = 1)
        fade_out ($id54r, 200000,1)
    end if
    if (%KEY_DOWN[55] = 1)
        fade_out ($id55r, 200000,1)
    end if
    if (%KEY_DOWN[56] = 1)
        fade_out ($id56r, 200000,1)
    end if
    if (%KEY_DOWN[57] = 1)
        fade_out ($id57r, 200000,1)
    end if
    if (%KEY_DOWN[58] = 1)
        fade_out ($id58r, 200000,1)
    end if
    if (%KEY_DOWN[59] = 1)
        fade_out ($id59r, 200000,1)
    end if
    if (%KEY_DOWN[60] = 1)
        fade_out ($id60r, 200000,1)
    end if
    if (%KEY_DOWN[61] = 1)
        fade_out ($id61r, 200000,1)
    end if
    if (%KEY_DOWN[62] = 1)
        fade_out ($id62r, 200000,1)
    end if
    if (%KEY_DOWN[63] = 1)
        fade_out ($id63r, 200000,1)
    end if
    if (%KEY_DOWN[64] = 1)
        fade_out ($id64r, 200000,1)
    end if
```

```

end if
  if (%KEY_DOWN[65] = 1)
    fade_out ($id65r, 200000,1)
  end if
  if (%KEY_DOWN[66] = 1)
    fade_out ($id66r, 200000,1)
  end if
  if (%KEY_DOWN[67] = 1)
    fade_out ($id67r, 200000,1)
  end if
  if (%KEY_DOWN[68] = 1)
    fade_out ($id68r, 200000,1)
  end if
  if (%KEY_DOWN[69] = 1)
    fade_out ($id69r, 200000,1)
  end if
  if (%KEY_DOWN[70] = 1)
    fade_out ($id70r, 200000,1)
  end if
  if (%KEY_DOWN[71] = 1)
    fade_out ($id71r, 200000,1)
  end if
  if (%KEY_DOWN[72] = 1)
    fade_out ($id72r, 200000,1)
  end if
  if (%KEY_DOWN[73] = 1)
    fade_out ($id73r, 200000,1)
  end if
  if (%KEY_DOWN[74] = 1)
    fade_out ($id74r, 200000,1)
  end if
  if (%KEY_DOWN[75] = 1)
    fade_out ($id75r, 200000,1)
  end if
  if (%KEY_DOWN[76] = 1)
    fade_out ($id76r, 200000,1)
  end if
  if (%KEY_DOWN[77] = 1)
    fade_out ($id77r, 200000,1)
  end if
  if (%KEY_DOWN[78] = 1)
    fade_out ($id78r, 200000,1)
  end if
  if (%KEY_DOWN[79] = 1)
    fade_out ($id79r, 200000,1)
  end if
  if (%KEY_DOWN[80] = 1)
    fade_out ($id80r, 200000,1)
  end if
  if (%KEY_DOWN[81] = 1)
    fade_out ($id81r, 200000,1)
  end if
  if (%KEY_DOWN[82] = 1)
    fade_out ($id82r, 200000,1)
  end if
  if (%KEY_DOWN[83] = 1)
    fade_out ($id83r, 200000,1)
  end if
  if (%KEY_DOWN[84] = 1)
    fade_out ($id84r, 200000,1)
  end if

```

```

end if
  if (%KEY_DOWN[85] = 1)
    fade_out ($id85r, 20000,1)
  end if
  if (%KEY_DOWN[86] = 1)
    fade_out ($id86r, 20000,1)
  end if
  if (%KEY_DOWN[87] = 1)
    fade_out ($id87r, 20000,1)
  end if

end on

```

Η γλώσσα προγραμματισμού του Kontakt λειτουργεί χωρίζοντας τα κομμάτια του κώδικα σε callbacks.

Callback είναι ένα συγκεκριμένο κομμάτι του κώδικα που εκτελείται σε μια συγκεκριμένη στιγμή.

Ο κώδικας που έχω γράψει χωρίζεται σε τρία callbacks στο initiation callback , στο release callback και τέλος στο note callback. Ας τα πάρουμε από την αρχή.

Το initiation Callback ξεκινάει με τον ορισμό on init και τελειώνει με το end on και περιέχει οτιδήποτε έχω γράψει ενδιάμεσα. Αυτό είναι το αρχικό callback και εκτελείται μόνο μία φορά όταν πατήσω το apply στον script editor και εκείνος τρέχει για πρώτη φορά τον κώδικα αφού έχει πρώτα τελειώσει την ανάλυση του χωρίς να βρει λάθη. Στο init callback λοιπόν δηλώνω τις μεταβλητές που θα χρησιμοποιήσω παρακάτω στον κώδικα μου. Συγκεκριμένα δήλωσα 43 μεταβλητές , όσα είναι και τα πλήκτρα που χρησιμοποιώ για να παίζω τις νότες στο όργανο.

Το όνομα της κάθε μεταβλητής ξεκινά με id ακολουθούμενο από το νούμερο της νότας που της αντιστοιχεί πχ. 45 και στο τέλος της ένα r. Όλες οι αριθμητικές μεταβλητές ορίζονται μπροστά από το όνομα τους με το σύμβολο \$. Έτσι το όνομα της πρώτης μας μεταβλητής είναι \$id45r της επόμενης \$id46r και συνεχίζει με την τελευταία να ονομάζεται \$id87r. Η δήλωση στο πρόγραμμα ότι θα χρησιμοποιήσω μια μεταβλητή γίνεται με την εντολή declare ακολουθούμενη από το όνομα της μεταβλητής.

Έτσι λοιπόν δήλωσα τις μεταβλητές ως εξής :

```

declare $id45r
declare $id46r
.....
declare $id87r

```

Το callback αυτό που άρχισε με τον ορισμό on init τελειώνει με τον ορισμό end on.

Το δεύτερο callback του κώδικα μου ξεκινάει με τον ορισμό on release και τελειώνει με το end on και περιέχει επίσης οτιδήποτε έχω γράψει ενδιάμεσα. Το callback αυτό εκτελείται κάθε φορά που εκτελείται από το πρόγραμμα ένα μήνυμα note off , δηλαδή την στιγμή που θα αφήσω ελεύθερο το πλήκτρο για αυτό και αναφέρεται ως release callback. Στο callback αυτό χρησιμοποιώ μια ρουτίνα ελέγχου if για να διαπιστώσω αν αληθεύει μια ισότητα ώστε να προβώ σε μια συγκεκριμένη ενέργεια. Για παράδειγμα αν έχω μια μεταβλητή \$number με τιμή 5 και θέλω αν ισχύει αυτή η τιμή να δίνω στην μεταβλητή \$page την τιμή 1 τότε θα χρησιμοποιούσα την if ως εξής:

```
if ($number = 5)
    $page := 1
end if
```

Με τον ορισμό end if δηλώνω το τέλος μιας ρουτίνας if. Έτσι αν η ισότητα που ελέγχει η if ισχύει τότε εκτελούνται οι παρακάτω εντολές μέχρι να δοθεί το end if αλλιώς αγνοούνται οι επόμενες εντολές και το πρόγραμμα προχωρά κατευθείαν στο end if. Μια παρόμοια περίπτωση χρησιμοποιώ και στον κώδικα αυτόν αλλά η μεταβλητή που ελέγχω είναι η %KEY_DOWN[note number] (όπου note number ο αριθμός της νότας που παίζω).

Η %KEY_DOWN[note number] είναι μία μεταβλητή που προϋπάρχει στο πρόγραμμα και παίρνει την τιμή 1 όταν πατήσω το πλήκτρο που αντιστοιχεί στον αριθμό της νότας που έχω δηλώσει ως [note number] και την τιμή 0 όταν αφήσω το πλήκτρο αυτό. Στην συγκεκριμένη περίπτωση της ισότητας που θέλω να ελέγξω (%KEY_DOWN[45] = 0) θέλω η μεταβλητή αυτή για την νότα με το νούμερο 45 να έχει τιμή 0 , ή αλλιώς να έχω αφήσει ελεύθερο το πλήκτρο που παίζει την συγκεκριμένη νότα. Η εντολή if (%KEY_DOWN[45] = 0) ελέγχει λοιπόν την ισότητα αυτή και αν ισχύει τότε εκτελεί την παρακάτω εντολή \$id45r := \$EVENT_ID και τελειώνει με το end if.

Η μεταβλητή \$EVENT_ID επίσης προϋπάρχει στο πρόγραμμα και ισούται με ένα ξεχωριστό αριθμό ταυτοποίησης του event που ενεργοποίησε το callback. Στο συγκεκριμένο callback επειδή είπαμε ότι είναι ένα release callback το event που το ενεργοποιεί κάθε φορά είναι ένα note off μήνυμα. Άρα χρησιμοποιώ την μεταβλητή που όρισα πιο πάνω ως \$id45r για να αποθηκεύσω τον αριθμό ταυτοποίησης του note off της νότας με νούμερο 45 όταν αυτό δίνεται (όταν δηλ. αφήσω ελεύθερο το πλήκτρο που αντιστοιχεί στην νότα με το νούμερο 45).

Όταν η ισότητα δεν ισχύει γιατί έχω αφήσει ένα άλλο πλήκτρο τότε το πρόγραμμα προχωρά κατευθείαν στην end if και συνεχίζει από εκεί να εκτελέσει τις υπόλοιπες γραμμές του κώδικα. Το ίδιο ακριβώς έκανα και στις επόμενες ρουτίνες if μόνο που κάθε φορά έλεγχα ένα διαφορετικό %KEY_DOWN από το πλήκτρο 45 έως το 87 και αποθήκευα αντίστοιχα τους αριθμούς ταυτοποίησης στις ανάλογες μεταβλητές από την \$id45r ως την \$id87r.

Στο τέλος του το callback αυτό που άρχισε με το on release τελειώνει με τον ορισμό end on.

Το τρίτο callback του κώδικα αυτού ξεκινάει με τον ορισμό on note και τελειώνει με το end on και περιέχει και αυτό όλες τις γραμμές κώδικα που υπάρχουν ενδιάμεσα. Το callback αυτό εκτελείται κάθε φορά που εκτελείται από το πρόγραμμα ένα μήνυμα note on , δηλαδή την στιγμή που θα πατήσω ένα πλήκτρο για αυτό και αναφέρεται ως note callback.

Ο κώδικας του note callback είναι περίπου ίδιος με αυτόν του release callback μόνο που εδώ θέλω η μεταβλητή %KEY_DOWN να έχει τιμή 1 δηλαδή να έχω πατημένο το πλήκτρο που της αντιστοιχεί ώστε να ισχύει η ισότητα. Επίσης αν η ισότητα ισχύει αντί να δίνω τιμή στην μεταβλητή \$id45r εκτελείται μια άλλη εντολή fade out.

Η εντολή fade out συντάσσεται ως εξής : fade_out (\$EVENT_ID, 100000,1) και χαμηλώνει σταδιακά την ένταση ενός event που καθορίζεται από την τιμή της μεταβλητής \$EVENT_ID. Η επόμενη παράμετρος που στο παράδειγμα αυτό είναι 100000 είναι ο χρόνος που θα χρειαστεί για να ολοκληρωθεί το fade out σε microseconds. Η τελευταία παράμετρος παίρνει την τιμή 1 αν θέλω να σταματά το event όταν τελειώσει το fade out ή την τιμή 0 αν θέλω να συνεχίσει να υπάρχει με μηδενική ένταση.

Εγώ την χρησιμοποίησα ως εξής : fade_out (\$id45r, 200000,1) αντικαθιστώντας για κάθε περίπτωση το \$EVENT_ID με την αντίστοιχη μεταβλητή που είχα ορίσει στο προηγούμενο callback από την \$id45r έως την \$id87r.

Στο τέλος του και το callback αυτό που άρχισε με το on note τελειώνει με τον ορισμό end on.

Αφού ανάλυσα επιμέρους την λειτουργία του κάθε callback θα δοκιμάσω την γενική λειτουργία του κώδικα μου όταν πατάω μια ίδια νότα δύο φορές.

Όταν πατήσω για πρώτη φορά το πλήκτρο που αντιστοιχεί στην νότα με το νούμερο 45 εκτελείται το note callback αλλά χωρίς κάποιο αποτέλεσμα αφού η μεταβλητή \$id45r δεν έχει πάρει ακόμα τιμή.

Η νότα ξεκινά να ηχεί και συνεχίζει ακόμα και όταν αφήσω το πλήκτρο καθώς το note off μήνυμα συνεχίζει να υπάρχει με μια συγκεκριμένη ένταση.

Όταν αφήσω λοιπόν το πλήκτρο θα εκτελεστεί το release callback , η πρώτη ρουτίνα if θα ελέγξει την ισότητα (%KEY_DOWN[45] = 0) θα δει ότι ισχύει και θα προχωρήσει στην εκτέλεση της εντολής \$id45r := \$EVENT_ID η οποία δώσει στην μεταβλητή τον αριθμό ταυτοποίησης του note off μηνύματος που μόλις εκτελέστηκε.

Έτσι αν πατήσω ξανά το πλήκτρο που αντιστοιχεί στην νότα με το νούμερο 45 θα εκτελεστεί το note callback , η πρώτη ρουτίνα if θα ελέγξει την ισότητα (%KEY_DOWN[45] = 1) θα δει ότι ισχύει και θα προχωρήσει στην εκτέλεση της εντολής fade_out (\$id45r, 200000,1) η οποία θα χαμηλώσει σταδιακά μέσα σε 200ms την ένταση του note off μηνύματος από το προηγούμενο άφημα του πλήκτρου 45 και όταν η ένταση φτάσει στο μηδέν θα σταματήσει το note off μήνυμα με συνέπεια να σταματήσει να ηχεί η προηγούμενη νότα ενώ η καινούργια νότα ξεκινά. Συγκεκριμένα οι δύο νότες θα ακούγονται μαζί για όσο χρόνο χρειάζεται να ολοκληρωθεί το fade out δηλαδή μόνο για 200ms.

Η λύση στο δεύτερο πρόβλημα ήρθε πιο εύκολα καθώς χρησιμοποίησα ένα έτοιμο script που υπήρχε στο Kontakt και λέγεται Microtunning. Το Script αυτό είναι αρκετά πολύπλοκο και είναι εξαιρετικά δύσκολο να εξηγήσω επακριβώς πως δουλεύει ο κώδικας του καθώς το έχουν γράψει οι ίδιοι οι προγραμματιστές που δημιούργησαν το Kontakt αλλά μπορώ να εξηγήσω τι ακριβώς κάνει και ποιες μετατροπές χρειάστηκε να χρησιμοποιήσω για να λειτουργήσει όπως ηθέλα.

Το Script αυτό εμφανίζει ένα νέο περιβάλλον στο παράθυρο του Script editor όπου μέσα από drop down menus μπορείς να επιλέξεις μία από τις υπάρχουσες 11 κλίμακες που χρησιμοποιούν μικροδιαστήματα και από ποία νότα θα ξεκινά η κλίμακα αυτή. Οι αλλαγές φαίνονται άμεσα σε μία γραφική αναπαράστασή με 12 μπάρες για κάθε ημιτόνιο τις οποίες ο χρήστης μπορεί να μετακινήσει αφαιρώντας η δίνοντας cents αν θέλει. Η ανάλυση των μικροδιαστημάτων είναι μεγάλη και μπορώ να αυξήσω ή να μειώσω αντίστοιχα ένα ημιτόνιο από 0.001 μέχρι 50 cents.

Όποια αλλαγή κάνω σε μία κλίμακα μπορώ να την αποθηκεύσω μέσα από το drop down menu save σε μία από τις πέντε ελεύθερες θέσεις που παρέχονται.

Μια βασική αλλαγή που έκανα ήταν να αντικαταστήσω τις δύο πρώτες υπάρχουσες κλίμακες με τις κλίμακες που θέλω να χρησιμοποιήσω. Έτσι αντικατέστησα τις Pure και Overtone 16-32 με την Ousak και την Xitzaz που χρησιμοποιούνται συνήθως στο κανονάκι. Άλλαξα στον κώδικα το όνομα τους και το κούρδισμα τους.

```

on init
  {-----Menu Entries-----}
  add_menu_item ($tuning, "Select Tuning", 99)
  add_menu_item ($tuning, "-----",99)
  add_menu_item ($tuning,"Ousak",5)
  add_menu_item ($tuning,"Xitzaz",9)
  add_menu_item ($tuning, "-----",99)

on ui_control($tuning)
  select ($tuning)
    case 5 {Ousak}
      %tune[0] := 0
      %tune[1] := 0
      %tune[2] := -50000
      %tune[3] := 0
      %tune[4] := 0
      %tune[5] := 0
      %tune[6] := 0
      %tune[7] := 0
      %tune[8] := 50000
      %tune[9] := 0
      %tune[10] := 0
      %tune[11] := 0

    case 9 {Xitzaz}
      %tune[0]:= 0
      %tune[1] := 0
      %tune[2] := -50000
      %tune[3] := 0
      %tune[4] := 0
      %tune[5] := -20000
      %tune[6] := 0
      %tune[7] := 0
      %tune[8] := 0
      %tune[9] := 0
      %tune[10] := 0
      %tune[11] := 0

```

Αλλαγή ονόματος κλίμακας στο ορισμό του μενού

Αλλαγή κουρδίσματος κλίμακας στον ορισμό του κουρδίσματος για την χρήση του drop down menu

Επόμενη αλλαγή ήταν η χρήση των πλήκτρων για την αλλαγή των κλιμάκων που μας ενδιαφέρουν.

Με τον επόμενο κώδικα ορίσα αν πατάω το πλήκτρο 92 (ΣΟΛ#) να επιλέγεται αυτόματα η κλίμακα Ousak , αν πατάω το πλήκτρο 94 (ΛΑ#) να επιλέγεται αυτόματα η κλίμακα Xitzaz ενώ αν πατάω το πλήκτρο 90 (ΦΑ#) να χρησιμοποιείται καθαρό κούρδισμα χωρίς μικροδιαστήματα.

```

if (%KEY_DOWN[90] = 1)
$stuning := 0
else
    if (%KEY_DOWN[92] = 1)
        $stuning := 5
    else
        if (%KEY_DOWN[94] = 1)
            $stuning := 9
        end if
    end if
end if

select ($stuning)
case 0 {NO TUNE}

        %tune[0] := 0
        %tune[1] := 0
        %tune[2] := 0
        %tune[3] := 0
        %tune[4] := 0
        %tune[5] := 0
        %tune[6] := 0
        %tune[7] := 0
        %tune[8] := 0
        %tune[9] := 0
        %tune[10] := 0
        %tune[11] := 0

case 5 {Ousak}

        %tune[0] := 0
        %tune[1] := 0
        %tune[2] := -50000
        %tune[3] := 0
        %tune[4] := 0
        %tune[5] := 0
        %tune[6] := 0
        %tune[7] := 0
        %tune[8] := 50000
        %tune[9] := 0
        %tune[10] := 0
        %tune[11] := 0

case 9 {Xitzaz}

        %tune[0]:= 0
        %tune[1] := 0
        %tune[2] := -50000
        %tune[3] := 0
        %tune[4] := 0
        %tune[5] := -20000
        %tune[6] := 0
        %tune[7] := 0
        %tune[8] := 0
        %tune[9] := 0
        %tune[10] := 0
        %tune[11] := 0

end select

```

Ο τελευταίος κώδικας που παρέβαλλα φροντίζει έτσι ώστε να αλλάζει η τονικότητα κάθε κλίμακας πατώντας τα αντίστοιχα πλήκτρα στο κλαβιέ δύο οκτάβες μετά το τέλος της έκτασης του οργάνου.

```

if (%KEY_DOWN[96] = 1)
$tonart := 0
else
  if (%KEY_DOWN[97] = 1)
$tonart := 11
  else
    if (%KEY_DOWN[98] = 1)
$tonart := 10
    else
      if (%KEY_DOWN[99] = 1)
$tonart := 9
      else
        if (%KEY_DOWN[100] = 1)
$tonart := 8
        else
          if (%KEY_DOWN[101] = 1)
$tonart := 7
          else
            if (%KEY_DOWN[102] = 1)
$tonart := 6
            else
              if (%KEY_DOWN[103] = 1)
$tonart := 5
              else
                if (%KEY_DOWN[104] = 1)
$tonart := 4
                else
                  if (%KEY_DOWN[105] = 1)
$tonart := 3
                  else
                    if (%KEY_DOWN[106] = 1)
$tonart := 2
                    else
                      if (%KEY_DOWN[107] = 1)
$tonart := 1
                      end if
                    end if
                  end if
                end if
              end if
            end if
          end if
        end if
      end if
    end if
  end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if

```

Έτσι καταφέρα χρησιμοποιώντας τον τροποποιημένο αυτόν κώδικα να κουρδίζω εύκολα και ενώ παίζω το όργανο μου στους πιο βασικούς ανατολίτικους δρόμους που χρησιμοποιούν μικροδιαστήματα. Παρακάτω παραθέτω ολόκληρο τον τροποποιημένο κώδικα του microtunning.

```

{*****}
Microtuning
Author : Native Instruments
Written by: Josef Natterer (natterer-script@web.de), Nicki Marinic
Created: Sep 22, 2004
Modified: Apr 4, 2005
Version: 1.0.2
*****}

```

on init

```

declare $act_note
declare $old_note
declare $old_note2
declare $diff
declare $diff2
declare $intens := 100
declare %tune[12]
declare %myTune_1[12]
declare %myTune_2[12]
declare %myTune_3[12]
declare %myTune_4[12]
declare %myTune_5[12]
declare $count := 0
declare $last_tuning
declare $last_tonart

{-----UI Elements-----}

declare ui_value_edit $I(-50000,50000,1000)
declare ui_value_edit $II(-50000,50000,1000)
declare ui_value_edit $III(-50000,50000,1000)
declare ui_value_edit $IV(-50000,50000,1000)
declare ui_value_edit $V(-50000,50000,1000)
declare ui_value_edit $VI(-50000,50000,1000)
declare ui_value_edit $VII(-50000,50000,1000)
declare ui_value_edit $VIII(-50000,50000,1000)
declare ui_value_edit $IX(-50000,50000,1000)
declare ui_value_edit $X(-50000,50000,1000)
declare ui_value_edit $XI(-50000,50000,1000)
declare ui_value_edit $XII(-50000,50000,1000)

declare ui_menu $tuning
declare ui_menu $tonart
declare ui_menu $save_tuning

declare ui_table %tune_ui[12](3,4,-50000)

declare ui_label $label (3, 1)
set_text ($label, "Do      Re      Mi      Fa      So      La      Si")

declare ui_label $interval_label (2,1)
declare ui_label $interval2_label (2,1)
set_text ($interval_label, "Interval:")
set_text ($interval2_label, "")

{-----Menu Entries-----}

```

```

add_menu_item ($tuning, "Select Tuning", 99)
add_menu_item ($tuning, "-----",99)
add_menu_item ($tuning,"Ousak",5)
add_menu_item ($tuning,"Xitzaz",9)
add_menu_item ($tuning, "-----",99)
{add_menu_item ($tuning,"Pythagorean",6)}
add_menu_item ($tuning,"Pythag. (Mid)",7)
add_menu_item ($tuning,"Pythag. (Up)",8)
add_menu_item ($tuning, "-----",99)
add_menu_item ($tuning,"Mean Tone 1/4",1)
add_menu_item ($tuning,"Silbermann",10)
add_menu_item ($tuning, "-----",99)
add_menu_item ($tuning,"Werckmeister III",3)
add_menu_item ($tuning,"Kirnberger III",2)
add_menu_item ($tuning,"Neidhardt I",11)
add_menu_item ($tuning,"Valotti",4)
add_menu_item ($tuning,"Young",12)
add_menu_item ($tuning, "-----",99)
add_menu_item ($tuning,"No Tune",0)
add_menu_item ($tuning, "-----",99)
add_menu_item ($tuning,"User 1",101)
add_menu_item ($tuning,"User 2",102)
add_menu_item ($tuning,"User 3",103)
add_menu_item ($tuning,"User 4",104)
add_menu_item ($tuning,"User 5",105)
add_menu_item ($tuning, "-----",99)
add_menu_item ($tuning,"<edit>",98)

```

```

add_menu_item ($tonart, "Select Key", 99)
add_menu_item ($tonart, "-----",99)
add_menu_item ($tonart,"C",0)
add_menu_item ($tonart,"C#",11)
add_menu_item ($tonart,"D",10)
add_menu_item ($tonart,"Eb",9)
add_menu_item ($tonart,"E",8)
add_menu_item ($tonart,"F",7)
add_menu_item ($tonart,"F#",6)
add_menu_item ($tonart,"G",5)
add_menu_item ($tonart,"Ab",4)
add_menu_item ($tonart,"A",3)
add_menu_item ($tonart,"Bb",2)
add_menu_item ($tonart,"B",1)

```

```

add_menu_item ($save_tuning,"Save Tuning",98)
add_menu_item ($save_tuning, "-----",99)
add_menu_item ($save_tuning,"User 1",1)
add_menu_item ($save_tuning,"User 2",2)
add_menu_item ($save_tuning,"User 3",3)
add_menu_item ($save_tuning,"User 4",4)
add_menu_item ($save_tuning,"User 5",5)

```

```
{----Initialization----}
```

```

%tune[0] := 0
%tune[1] := -29328
%tune[2] := 3910
%tune[3] := 15641
%tune[4] := -13686

```

```
%tune[5] := -1955
%tune[6] := -31283
%tune[7] := 1955
%tune[8] := 13686
%tune[9] := -15641
%tune[10] := 17596
%tune[11] := -11731
```

```
$I := 0
$II := -29328
$III := 3910
$IV := 15641
$V := -13686
$VI := -1955
$VII := -31283
$VIII := 1955
$IX := 13686
$X := -15641
$XI := 17596
$XII := -11731
```

```
%tune_ui[0] := %tune[0]
%tune_ui[1] := %tune[1]
%tune_ui[2] := %tune[2]
%tune_ui[3] := %tune[3]
%tune_ui[4] := %tune[4]
%tune_ui[5] := %tune[5]
%tune_ui[6] := %tune[6]
%tune_ui[7] := %tune[7]
%tune_ui[8] := %tune[8]
%tune_ui[9] := %tune[9]
%tune_ui[10] := %tune[10]
%tune_ui[11] := %tune[11]
```

```
$tuning := 5
$last_tuning := 5
$tonart := 0
$last_tonart := 0
$save_tuning := 0
```

```
{-----Layout-----}
```

```
move_control ($label, 1,1)
move_control (%tune_ui, 1,2)
move_control ($I, 5,1)
move_control ($II, 5,2)
move_control ($III, 5,3)
move_control ($IV, 5,4)
move_control ($V, 5,5)
move_control ($VI, 5,6)
move_control ($VII, 6,1)
move_control ($VIII, 6,2)
move_control ($IX, 6,3)
move_control ($X, 6,4)
move_control ($XI, 6,5)
move_control ($XII, 6,6)
move_control($tonart,4,2)
move_control($tuning,4,3)
move_control ($save_tuning, 4,5)
```

```
move_control ($interval_label, 1,6)
move_control ($interval2_label, 3,6)
```

```
{----- Recall -----}
```

```
make_persistent (%mytune_1)
make_persistent (%mytune_2)
make_persistent (%mytune_3)
make_persistent (%mytune_4)
make_persistent (%mytune_5)
make_persistent ($last_tuning)
make_persistent ($last_tonart)
make_persistent ($tonart)
make_persistent ($tuning)
make_persistent (%tune_ui)
make_persistent ($I)
make_persistent ($II)
make_persistent ($III)
make_persistent ($IV)
make_persistent ($V)
make_persistent ($VI)
make_persistent ($VII)
make_persistent ($VIII)
make_persistent ($XI)
make_persistent ($X)
make_persistent ($XI)
make_persistent ($XII)
```

```
message (" ")
```

```
end on
```

```
on ui_control($tuning)
  select ($tuning)
```

```
  case 0 {No Tune}
```

```
    %tune[0] := 0
    %tune[1] := 0
    %tune[2] := 0
    %tune[3] := 0
    %tune[4] := 0
    %tune[5] := 0
    %tune[6] := 0
    %tune[7] := 0
    %tune[8] := 0
    %tune[9] := 0
    %tune[10] := 0
    %tune[11] := 0
```

```
  case 1 {MEAN TONE 1/4 comma}
```

```
    %tune[0] := 0
    %tune[1] := -23950
    %tune[2] := -6843
    %tune[3] := 10265
```

```
%tune[4] := -13686
%tune[5] := 3422
%tune[6] := -20529
%tune[7] := -3422
%tune[8] := -27372
%tune[9] := -10265
%tune[10] := 6843
%tune[11] := -17108
```

case 2 {KIRNBERGER III}

```
%tune[0] := 0
%tune[1] := -9775
%tune[2] := -6843
%tune[3] := -5865
%tune[4] := -13686
%tune[5] := -1955
%tune[6] := -11730
%tune[7] := -3421
%tune[8] := -7820
%tune[9] := -10264
%tune[10] := -3910
%tune[11] := -11731
```

case 3 {WERCKMEISTER III}

```
%tune[0] := 0
%tune[1] := -9775
%tune[2] := -7820
%tune[3] := -5865
%tune[4] := -9775
%tune[5] := -1955
%tune[6] := -11730
%tune[7] := -3910
%tune[8] := -7820
%tune[9] := -11730
%tune[10] := -3910
%tune[11] := -7820
```

case 4 {Valotti}

```
%tune[0] := 0
%tune[1] := -5865
%tune[2] := -3910
%tune[3] := -1955
%tune[4] := -7820
%tune[5] := 1955
%tune[6] := -7820
%tune[7] := -1955
%tune[8] := -3910
%tune[9] := -5865
%tune[10] := 0
%tune[11] := -9775
```

case 5 {Ousak}

```
%tune[0] := 0
%tune[1] := 0
%tune[2] := -50000
```



```
%tune[3] := 0
%tune[4] := 0
%tune[5] := 0
%tune[6] := 0
%tune[7] := 0
%tune[8] := 50000
%tune[9] := 0
%tune[10] := 0
%tune[11] := 0
```

case 6 {PYTHAGOREAN}

```
%tune[0]:= 0
%tune[1] := 13685
%tune[2] := 3910
%tune[3] := -5865
%tune[4] := 7820
%tune[5] := -1955
%tune[6] := 11730
%tune[7] := 1955
%tune[8] := 15640
%tune[9] := 5865
%tune[10] := -3910
%tune[11] := 9775
```

case 7 {Pythagorean Middle}

```
%tune[0]:= 0
%tune[1] := -9775
%tune[2] := 3910
%tune[3] := -5865
%tune[4] := 7820
%tune[5] := -1955
%tune[6] := 0
%tune[7] := 1955
%tune[8] := -7820
%tune[9] := 5865
%tune[10] := -3910
%tune[11] := 9775
```

case 8 {Pythagorean Up}

```
%tune[0]:= 0
%tune[1] := 13685
%tune[2] := 3910
%tune[3] := 17595
%tune[4] := 7820
%tune[5] := 21505
%tune[6] := 11730
%tune[7] := 1955
%tune[8] := 15640
%tune[9] := 5865
%tune[10] := 19550
%tune[11] := 9775
```

case 9 {Xitzaz}

```
%tune[0]:= 0
%tune[1] := 0
%tune[2] := -50000
%tune[3] := 0
%tune[4] := 0
%tune[5] := -20000
```

```

%tune[6] := 0
%tune[7] := 0
%tune[8] := 0
%tune[9] := 0
%tune[10] := 0
%tune[11] := 0

case 10 {Silbermann}
%tune[0]:= 0
%tune[1] := 8145
%tune[2] := -3258
%tune[3] := 4887
%tune[4] := -6515
%tune[5] := 1629
%tune[6] := -9774
%tune[7] := -1629
%tune[8] := 6515
%tune[9] := -4887
%tune[10] := 3258
%tune[11] := -8145

case 11 {Neidhardt I}
%tune[0]:= 0
%tune[1] := -5868
%tune[2] := -3910
%tune[3] := -3910
%tune[4] := -7820
%tune[5] := -1955
%tune[6] := -7820
%tune[7] := -1955
%tune[8] := -3910
%tune[9] := -5865
%tune[10] := -3910
%tune[11] := -7820

case 12 {Young 1/6 pC}
%tune[0]:= 0
%tune[1] := -9775
%tune[2] := -3910
%tune[3] := -5865
%tune[4] := -7820
%tune[5] := -1955
%tune[6] := -11730
%tune[7] := -1955
%tune[8] := -7820
%tune[9] := -5865
%tune[10] := -3910
%tune[11] := -9775

case 101
%tune[0] := %myTune_1[0]
%tune[1] := %myTune_1[1]
%tune[2] := %myTune_1[2]
%tune[3] := %myTune_1[3]
%tune[4] := %myTune_1[4]
%tune[5] := %myTune_1[5]
%tune[6] := %myTune_1[6]
%tune[7] := %myTune_1[7]
%tune[8] := %myTune_1[8]

```

```
%tune[9] := %myTune_1[9]
%tune[10] := %myTune_1[10]
%tune[11] := %myTune_1[11]
```

case 102

```
%tune[0] := %myTune_2[0]
%tune[1] := %myTune_2[1]
%tune[2] := %myTune_2[2]
%tune[3] := %myTune_2[3]
%tune[4] := %myTune_2[4]
%tune[5] := %myTune_2[5]
%tune[6] := %myTune_2[6]
%tune[7] := %myTune_2[7]
%tune[8] := %myTune_2[8]
%tune[9] := %myTune_2[9]
%tune[10] := %myTune_2[10]
%tune[11] := %myTune_2[11]
```

case 103

```
%tune[0] := %myTune_3[0]
%tune[1] := %myTune_3[1]
%tune[2] := %myTune_3[2]
%tune[3] := %myTune_3[3]
%tune[4] := %myTune_3[4]
%tune[5] := %myTune_3[5]
%tune[6] := %myTune_3[6]
%tune[7] := %myTune_3[7]
%tune[8] := %myTune_3[8]
%tune[9] := %myTune_3[9]
%tune[10] := %myTune_3[10]
%tune[11] := %myTune_3[11]
```

case 104

```
%tune[0] := %myTune_4[0]
%tune[1] := %myTune_4[1]
%tune[2] := %myTune_4[2]
%tune[3] := %myTune_4[3]
%tune[4] := %myTune_4[4]
%tune[5] := %myTune_4[5]
%tune[6] := %myTune_4[6]
%tune[7] := %myTune_4[7]
%tune[8] := %myTune_4[8]
%tune[9] := %myTune_4[9]
%tune[10] := %myTune_4[10]
%tune[11] := %myTune_4[11]
```

case 105

```
%tune[0] := %myTune_5[0]
%tune[1] := %myTune_5[1]
%tune[2] := %myTune_5[2]
%tune[3] := %myTune_5[3]
%tune[4] := %myTune_5[4]
%tune[5] := %myTune_5[5]
%tune[6] := %myTune_5[6]
%tune[7] := %myTune_5[7]
%tune[8] := %myTune_5[8]
%tune[9] := %myTune_5[9]
%tune[10] := %myTune_5[10]
%tune[11] := %myTune_5[11]
```

end select

```

%tune_ui[0] := %tune[0]
%tune_ui[1] := %tune[1]
%tune_ui[2] := %tune[2]
%tune_ui[3] := %tune[3]
%tune_ui[4] := %tune[4]
%tune_ui[5] := %tune[5]
%tune_ui[6] := %tune[6]
%tune_ui[7] := %tune[7]
%tune_ui[8] := %tune[8]
%tune_ui[9] := %tune[9]
%tune_ui[10] := %tune[10]
%tune_ui[11] := %tune[11]

$I := %tune[0]
$II := %tune[1]
$III := %tune[2]
$IV := %tune[3]
$V := %tune[4]
$VI := %tune[5]
$VII := %tune[6]
$VIII := %tune[7]
$IX := %tune[8]
$X := %tune[9]
$XI := %tune[10]
$XII := %tune[11]

if ($stuning = 99 or $stuning = 98)
    $stuning := $last_tuning
else
    $last_tuning := $stuning
end if

```

end on

```

on ui_control(%tune_ui)
    $I := %tune_ui[0]
    $II := %tune_ui[1]
    $III := %tune_ui[2]
    $IV := %tune_ui[3]
    $V := %tune_ui[4]
    $VI := %tune_ui[5]
    $VII := %tune_ui[6]
    $VIII := %tune_ui[7]
    $IX := %tune_ui[8]
    $X := %tune_ui[9]
    $XI := %tune_ui[10]
    $XII := %tune_ui[11]

    %tune[0] := %tune_ui[0]
    %tune[1] := %tune_ui[1]
    %tune[2] := %tune_ui[2]
    %tune[3] := %tune_ui[3]
    %tune[4] := %tune_ui[4]
    %tune[5] := %tune_ui[5]
    %tune[6] := %tune_ui[6]
    %tune[7] := %tune_ui[7]
    %tune[8] := %tune_ui[8]
    %tune[9] := %tune_ui[9]
    %tune[10] := %tune_ui[10]
    %tune[11] := %tune_ui[11]

```

```

        $tuning := 98
        $last_tuning := 98

end on

on ui_control($I)
    %tune[0] := $I
    %tune_ui[0] := $I
    $tuning := 98
    $last_tuning := 98
end on

on ui_control($II)
    %tune[1] := $II
    %tune_ui[1] := $II
    $tuning := 98
    $last_tuning := 98
end on

on ui_control($III)
    %tune[2] := $III
    %tune_ui[2] := $III
    $tuning := 98
    $last_tuning := 98
end on

on ui_control($IV)
    %tune[3] := $IV
    %tune_ui[3] := $IV
    $tuning := 98
    $last_tuning := 98
end on

on ui_control($V)
    %tune[4] := $V
    %tune_ui[4] := $V
    $tuning := 98
    $last_tuning := 98
end on

on ui_control($VI)
    %tune[5] := $VI
    %tune_ui[5] := $VI
    $tuning := 98
    $last_tuning := 98
end on

on ui_control($VII)
    %tune[6] := $VII
    %tune_ui[6] := $VII
    $tuning := 98
    $last_tuning := 98
end on

on ui_control($VIII)
    %tune[7] := $VIII
    %tune_ui[7] := $VIII
    $tuning := 98
    $last_tuning := 98

```

```

end on

on ui_control($IX)
    %tune[8] := $IX
    %tune_ui[8] := $IX
    $tuning := 98
    $last_tuning := 98
end on

on ui_control($X)
    %tune[9] := $X
    %tune_ui[9] := $X
    $tuning := 98
    $last_tuning := 98
end on

on ui_control($XI)
    %tune[10] := $XI
    %tune_ui[10] := $XI
    $tuning := 98
    $last_tuning := 98
end on

on ui_control($XII)
    %tune[11] := $XII
    %tune_ui[11] := $XII
    $tuning := 98
    $last_tuning := 98
end on

on ui_control ($tonart)
    if ($tonart # 99)
        $last_tonart := $tonart
    else
        $tonart := $last_tonart
    end if
end on

on ui_control ($save_tuning)
    $count := 0
    select ($save_tuning)
        case 1
            while ($count < 12)
                %mytune_1[$count] := %tune[$count]
                inc ($count)
            end while
            $tuning := 101
        case 2
            while ($count < 12)
                %mytune_2[$count] := %tune[$count]
                inc ($count)
            end while
            $tuning := 102
        case 3
            while ($count < 12)
                %mytune_3[$count] := %tune[$count]
                inc ($count)
            end while
            $tuning := 103
        case 4
            while ($count < 12)

```

```

                                %mytune_4[$count] := %tune[$count]
                                inc ($count)
                            end while
                            $tuning := 104
                        case 5
                            while ($count < 12)
                                %mytune_5[$count] := %tune[$count]
                                inc ($count)
                            end while
                            $tuning := 105
                        end select

                    $save_tuning := 98
                end on

            on note

                if (%KEY_DOWN[90] = 1)
                    $tuning := 0
                else
                    if (%KEY_DOWN[92] = 1)
                        $tuning := 5
                    else
                        if (%KEY_DOWN[94] = 1)
                            $tuning := 9
                        end if
                    end if
                end if

                select ($tuning)
                    case 0 {NO TUNE}

                        %tune[0] := 0
                        %tune[1] := 0
                        %tune[2] := 0
                        %tune[3] := 0
                        %tune[4] := 0
                        %tune[5] := 0
                        %tune[6] := 0
                        %tune[7] := 0
                        %tune[8] := 0
                        %tune[9] := 0
                        %tune[10] := 0
                        %tune[11] := 0

                    case 5 {Ousak}

                        %tune[0] := 0
                        %tune[1] := 0
                        %tune[2] := -50000
                        %tune[3] := 0
                        %tune[4] := 0
                        %tune[5] := 0
                        %tune[6] := 0
                        %tune[7] := 0
                        %tune[8] := 50000
                        %tune[9] := 0
                        %tune[10] := 0
                        %tune[11] := 0

                    case 9 {Xitzaz}

```

```

        %tune[0]:= 0
        %tune[1] := 0
        %tune[2] := -50000
        %tune[3] := 0
        %tune[4] := 0
        %tune[5] := -20000
        %tune[6] := 0
        %tune[7] := 0
        %tune[8] := 0
        %tune[9] := 0
        %tune[10] := 0
        %tune[11] := 0
end select

```

```

%tune_ui[0] := %tune[0]
%tune_ui[1] := %tune[1]
%tune_ui[2] := %tune[2]
%tune_ui[3] := %tune[3]
%tune_ui[4] := %tune[4]
%tune_ui[5] := %tune[5]
%tune_ui[6] := %tune[6]
%tune_ui[7] := %tune[7]
%tune_ui[8] := %tune[8]
%tune_ui[9] := %tune[9]
%tune_ui[10] := %tune[10]
%tune_ui[11] := %tune[11]

```

```

$I := %tune[0]
$II := %tune[1]
$III := %tune[2]
$IV := %tune[3]
$V := %tune[4]
$VI := %tune[5]
$VII := %tune[6]
$VIII := %tune[7]
$IX := %tune[8]
$X := %tune[9]
$XI := %tune[10]
$XII := %tune[11]

```

```

if ($tuning = 99 or $tuning = 98)
    $tuning := $last_tuning
else
    $last_tuning := $tuning
end if

```

```

if (%KEY_DOWN[96] = 1)
    $tonart := 0
else
    if (%KEY_DOWN[97] = 1)
        $tonart := 11
    else
        if (%KEY_DOWN[98] = 1)
            $tonart := 10
        else
            if (%KEY_DOWN[99] = 1)
                $tonart := 9
            else
                if (%KEY_DOWN[100] = 1)

```



```

$tonart := 8
else
if (%KEY_DOWN[101] = 1)
$tonart := 7
else
if (%KEY_DOWN[102] = 1)
$tonart := 6
else
if (%KEY_DOWN[103] = 1)
$tonart := 5
else
if (%KEY_DOWN[104] = 1)
$tonart := 4
else
if (%KEY_DOWN[105] = 1)
$tonart := 3
else
if (%KEY_DOWN[106] = 1)
$tonart := 2
else
if (%KEY_DOWN[107] = 1)
$tonart := 1
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if
end if

if ($tonart # 99)
$last_tonart := $tonart
else
$tonart := $last_tonart
end if

if ($sold_note # $sact_note)
$sold_note := $sact_note
end if

$sact_note := $EVENT_NOTE + $tonart
$sact_note := $sact_note mod 12
set_event_mark($EVENT_ID,sh_left(2,$sact_note))
if (%KEY_DOWN[$sold_note2] = 1 and ($sold_note # $sact_note))
if ($sold_note2 < $EVENT_NOTE)
$difff := %tune[$sact_note] - %tune[$sold_note]
$difff2 := (((sact_note -$sold_note + 12) mod 12) *100000) + $difff
set_text($interval_label,"Interval: "&$sold_note+1&" to "&$sact_note+1&":
"&$difff&" mcent" )
else
$difff := %tune[$sold_note] - %tune[$sact_note]
$difff2 := (((sold_note -$sact_note + 12) mod 12) *100000) + $difff

```

```

set_text($interval_label,"Interval: "&$act_note+1&" to "&$sold_note+1&":
"&$diff&" mcent" )
end if
set_text($interval2_label,"")
select ($diff2)
  case 70672
    set_text($interval2_label,"25/24 (minor chroma)")
  case 104955
    set_text($interval2_label,"17/16 (17th harmonic)")
  case 203910
    set_text($interval2_label,"9/8 (major whole tone)")
  case 315641
    set_text($interval2_label,"6/5 (minor third)")
  case 297513
    set_text($interval2_label,"19/16 (19th harmonic)")
  case 386314
    set_text($interval2_label,"5/4 (major third)")
  case 407820
    set_text($interval2_label,"81/64 (pythagorean third)")
  case 470781
    set_text($interval2_label,"21/16 (narrow fourth)")
  case 498045
    set_text($interval2_label,"4/3 (perfect fourth)")
  case 551318
    set_text($interval2_label,"11/8")
  case 568717
    set_text($interval2_label,"25/18 (classic aug fourth)")
  case 701955
    set_text($interval2_label,"3/2 (perfect fifth)")
  case 737637
    set_text($interval2_label,"wolf fifth")
  case 813686
    set_text($interval2_label,"8/5 (minor sixth)")
  case 840528
    set_text($interval2_label,"13/8")
  case 884359
    set_text($interval2_label,"5/3 (major sixth)")
  case 905865
    set_text($interval2_label,"27/16 (pythag. major sixth)")
  case 968826
    set_text($interval2_label,"7/4 (harmonic seventh)")
  case 1017596
    set_text($interval2_label,"9/5 (just minor seventh)")
  case 1088269
    set_text($interval2_label,"15/8 (classic minor seventh)")
  case 0
    set_text($interval2_label,"2/1 (octave)")
end select
else
$diff := 0
set_text($interval2_label,"")
set_text($interval_label,"Interval: ")
end if
$sold_note2 := $EVENT_NOTE
select ($act_note)
  case 0
    change_tune(by_marks($MARK_1),(%tune[0] * $Sintens)/100,0)
  case 1
    change_tune(by_marks($MARK_2),(%tune[1] * $Sintens)/100,0)
  case 2

```

```

        change_tune(by_marks($MARK_3),(%tune[2] * $intens)/100,0)
case 3
        change_tune(by_marks($MARK_4),(%tune[3] * $intens)/100,0)
case 4
        change_tune(by_marks($MARK_5),(%tune[4] * $intens)/100,0)
case 5
        change_tune(by_marks($MARK_6),(%tune[5] * $intens)/100,0)
case 6
        change_tune(by_marks($MARK_7),(%tune[6] * $intens)/100,0)
case 7
        change_tune(by_marks($MARK_8),(%tune[7] * $intens)/100,0)
case 8
        change_tune(by_marks($MARK_9),(%tune[8] * $intens)/100,0)
case 9
        change_tune(by_marks($MARK_10),(%tune[9] * $intens)/100,0)
case 10
        change_tune(by_marks($MARK_11),(%tune[10] * $intens)/100,0)
case 11
        change_tune(by_marks($MARK_12),(%tune[11] * $intens)/100,0)
    end select
end on
on release
    set_text ($interval_label, "Interval:")
    set_text ($interval2_label, " ")
end on

```

5. ΗΧΗΤΙΚΟ ΑΠΟΤΕΛΕΣΜΑ

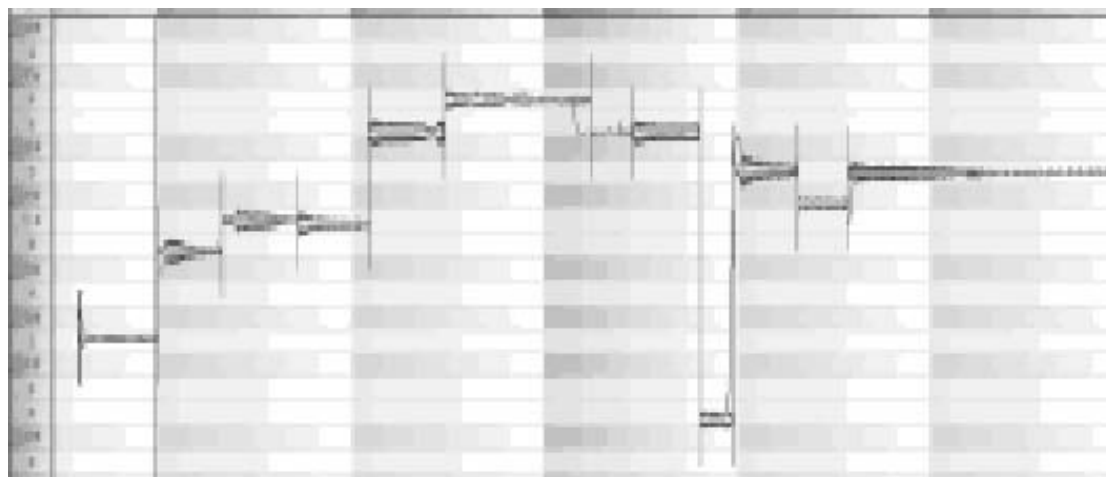
5.1 Σύγκριση ηχητικού αποτελεσματος του sampler με τον αυθεντικό ήχο του οργάνου

Αφού ολοκλήρωσα την δημιουργία του οργάνου μου ήρθε η ώρα να δοκιμάσω το πόσο αληθοφανές ακούγεται. Η δοκιμή αυτή βασίζεται σε μεγάλο βαθμό στην σύγκριση του φυσικού ήχου μέσα από ένα ηχογραφημένο ταξίμι και της αναπαραγωγής του ίδιου μουσικού παραδείγματος από το sampler.

Για να γίνει η σύγκριση όσο το δυνατόν καλύτερα χρειάστηκε να δημιουργήσω ένα midi αρχείο που να περιέχει ακριβώς τις ίδιες νότες με το ηχογραφημένο ταξίμι.

Για την μετατροπή του audio αρχείου σε midi χρησιμοποίησα το πρόγραμμα Melodyne 2.6 της Celemony software που ενδείκνυται για αυτή τη συγκεκριμένη δουλειά.

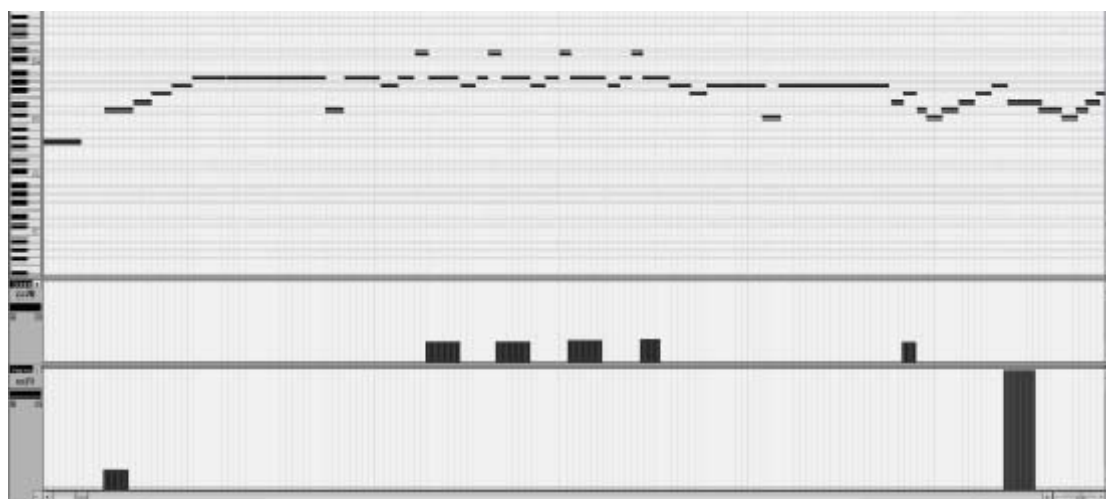
Εισήγαγα το αρχείο ήχου και το πρόγραμμα το επεξεργάστηκε έτσι ώστε να ξεχωρίσει τις νότες μεταξύ τους και να ανακαλύψει το pitch της κάθε νότας. Χρειάστηκε μετά από προσεκτική ακρόαση να επέμβω σε συγκεκριμένα σημεία ώστε να διορθώσω μόνος μου κάποιες λάθος προβλέψεις του προγράμματος στο τονικό ύψος και τον διαχωρισμό των νοτών. Το αποτέλεσμα ήταν να δημιουργήσω από το πρόγραμμα ένα midi αρχείο που περιείχε ακριβώς τις ίδιες νότες με το αρχικό αρχείο ήχου.



Εικόνα από την επεξεργασία του αρχείου ήχου μέσα από το πρόγραμμα Melodyne

Για την περαιτέρω παραμετροποίηση και χρήση του αρχείου midi χρησιμοποιήσα το δημοφιλές Audio production πρόγραμμα Cubase 3 της Steinberg.

Χρησιμοποιώντας τον midi editor του συγκεκριμένου προγράμματος κατάφερα να ορίσω τιμές για τους controllers 70 και 71 όπου χρειαζόταν να ακουστεί μια νότα με κάποιο εφέ όπως πχ. Vibrato και ανάθεσα την αναπαραγωγή του midi αρχείου στο Kontakt που είχε φορτωμένο το όργανο. Το αποτέλεσμα το εξήγαγα από το πρόγραμμα ως ένα νέο audio αρχείο που ήταν έτοιμο για να συγκριθεί ηχητικά με το ηχογραφημένο ταξίμι που έπαιξε ο μουσικός.



Εικόνα από τον ορισμό τιμών των controllers μέσα από τον midi editor του Cubase.

Ακούγοντας προσεκτικά και τα δύο αρχεία ήχου ξεχωρίζω εύκολα ποιο έχει δημιουργηθεί από τον υπολογιστή καθώς περιέχει έκτος των άλλων ένα πολύ γρήγορο παίξιμο των χορδών σε οκτάβα που είναι αδύνατον να προσομοιωθεί πειστικά από το sampler.

Το ηχογραφημένο ταξίμι (CD track 3) ακούγεται πιο συγκροτημένο με καλύτερη ροή και πιο δεμένο ήχο από το ταξίμι που δημιουργήσα μέσω του sampler (CD track 4).

Σε γενικές γραμμές ο ήχος του audio αρχείου που δημιουργήσα είναι αρκετά πειστικός καθώς προσομοιώνει αρκετά καλά τις δυναμικές του οργάνου με την εναλλαγή των velocity layers που είχα δημιουργήσει. Μεγάλο πλεονέκτημα στην πειστικότητα του είναι και η χρήση των εφέ όπως το vibrato και μάλιστα σε γρήγορα περάσματα (κάτι πρακτικά αδύνατο με την χρήση του κλαβιέ).

Σίγουρα το ηχογραφημένο κανονάκι έχει διαφορετικό feeling στο άκουσμα (στο οποίο συμβάλει τα μέγιστα και ο μουσικός) κάτι που δεν προσομοιώνεται από ένα sampler άλλα το όργανο μου καταφέρνει υπό συνθήκες (χωρίς εξαιρετικά γρήγορα παιξίματα σε πολλές χορδές) να πλησιάσει πάρα πολύ τον ήχο του αυθεντικού και σε συνδυασμό με την χρήση των εφέ που διαθέτει να προσφέρει αληθοφάνεια στο τελικό αποτέλεσμα.

5.2 Σύγκριση των τρόπων χρήσης του sampler – Midi προγραμματισμός - Keyboard Controller – Midi κιθάρα

Δοκίμασα επίσης τον ήχο του οργάνου μου χρησιμοποιώντας ένα midi κλαβιέ (Midi keyboard controller) για να διαπιστώσω τον καλύτερο τρόπο χρήσης του sampler.

Το ηχητικό αποτέλεσμα κινήθηκε πολύ κοντά στον ήχο του κανονικού οργάνου και η ύπαρξη δειγμάτων δύο διαφορετικών δυναμικών έκανε το αποτέλεσμα πιο αληθοφανές. Μπορούσα να εκμεταλλευτώ άμεσα τις δυνατότητες του midi keyboard μέσω των controllers για να αλλάξω την στερεοφωνική εικόνα του οργάνου, την συνολική ένταση του, η να χρησιμοποιήσω το pitch bend την ώρα που παίζω. Η χρήση του microtuning για αλλαγή του κουρδίσματος γίνεται χωρίς πρόβλημα αρκεί το keyboard να έχει παραπάνω από τέσσερις οκτάβες. Το μοναδικό πρόβλημα που αντιμετώπισα είναι η χρήση των εφέ του οργάνου σε μια νότα κατά το παίξιμο καθώς η γρήγορη αλλαγή των controllers καθιστά την συγκεκριμένη λειτουργία μη εφικτή.

Τελευταία δοκίμασα την midi κιθάρα ως controller για το sampler. Το αποτέλεσμα ήταν απογοητευτικό. Ο pitch to midi converter της κιθάρας δεν μπόρεσε να ανταπεξέλθει στα γρήγορα παιξίματα και στο ταυτόχρονο παίξιμο σε παραπάνω από μία χορδές με αποτέλεσμα να μην αποδίδει πάντα το σωστό τονικό ύψος και να μεταφράζει τα τριξίματα των χορδών σε νότες που δεν είχαν σχέση με αυτό που έπαιζα. Ένα ακόμα μειονέκτημα είναι η απουσία ενσωματωμένων midi controllers που θα μπορούσα να χρησιμοποιήσω για να παίζω κάποιο από τα εφέ του οργάνου. Τα προβλήματα αυτά που παρουσιάζει η midi κιθάρα δεν την καθιστούν ένα ικανοποιητικό controller για το όργανο μου σε αντίθεση με το midi keyboard που απέδωσε πολύ καλύτερα.

Γενικά το Midi keyboard φαίνεται να είναι το καλύτερο μέσο για την χρήση του sampler μετά από τον midi προγραμματισμό αρκεί να πληρεί κάποια συγκεκριμένα χαρακτηριστικά όπως δυναμικά πλήκτρα, ενσωματωμένους αναθέσιμους controllers και πέντε τουλάχιστον οκτάβες. Η χρήση του midi κλαβιέ προσθέτει επίσης στην εκφραστικότητα του οργάνου καθώς ο μουσικός που παίζει τα πλήκτρα θα ακουστεί πιο φυσικά από ένα αρχείο midi που έχουμε γράψει σε ένα πρόγραμμα, όσο καλά και αν είναι προγραμματισμένο.

Η αλήθεια είναι ότι αν θέλω να χρησιμοποιήσω τα διάφορα έφε που μου προσφέρει το όργανο πάνω σε ένα γρήγορο ταξίμι δεν μπορώ να βασιστώ στην χρήση του κλαβιέ καθώς η γρήγορη εναλλαγή των controllers για την χρήση του εφέ σε μια μόνο νότα καθώς παίζω είναι πρακτικά αδύνατη.

Σε αυτή λοιπόν την περίπτωση ο προγραμματισμός ενός midi αρχείου είναι μονόδρομος. Για τον λόγο αυτό ο midi προγραμματισμός είναι ο καλύτερος τρόπος χρήσης για ένα τόσο περίπλοκο όργανο , γιατί προσφέρει απόλυτο έλεγχο σε όλες τις παραμέτρους του οργάνου , κάτι που δεν μπορούν να προσφέρουν οι midi controllers που λειτουργούν σε πραγματικό χρόνο.

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Το ηχητικό αποτέλεσμα του οργάνου που δημιούργησα κρίνεται αποτελεσματικό. Είναι αρκετά αληθοφανές και μπορεί να χρησιμοποιηθεί σε αρκετές περιπτώσεις για να αντικαταστήσει ένα πραγματικό κανονάκι.

Τα μειονεκτήματα του οργάνου που δημιούργησα μέσω του sampler είναι η έλλειψη αληθοφάνειας σε πολύ γρήγορα παιξίματα και η ανάγκη midi προγραμματισμού για την σωστή χρήση των εφέ του οργάνου. Το μεγαλύτερο πλεονέκτημα του όμως είναι η προσφορά χρήσης του οπουδήποτε ως αντικαταστάτης του κανονικού οργάνου στις πολλές περιπτώσεις που είναι δύσκολο να βρεθεί ένα πλέον σπάνιο όργανο όπως το κανονάκι και ο μουσικός που θα μπορεί να το χρησιμοποιήσει.

Έτσι λοιπόν στην εποχή που πλέον η χρήση του ηλεκτρονικού υπολογιστή και της τεχνολογίας γενικότερα έχει αλλάξει τα δεδομένα παρέχοντας συνεχώς ευκολίες και νέες λύσεις στην μουσική παραγωγή , η απουσία δυνατότητας της ηχογράφησης ενός οργάνου δεν αποτελεί πια πρόβλημα καθώς υπάρχει πάντα διαθέσιμος ο καλύτερος αντικαταστάτης του , το Sampler.

7. ΒΙΒΛΙΟΓΡΑΦΙΑ

Παρακάτω παραθέτω την βιβλιογραφία πού χρησιμοποιήθηκε.

Στα αριστερά αναφέρεται η ημερομηνία της τελευταίας επίσκεψης στην συγκεκριμένη ιστοσελίδα

Κανονάκι:

www.geocities.com/canonaki/	16-11-2006
http://www.e-politismos.gr/php/articles.php?lng=gr&pg=126	12-11-2006
www.angelfire.com/music5/kanun/left.htm	22-11-2006
http://portal.kithara.gr/modules.php?name=News&file=article&sid=264	22-11-2006
www.mousikoergastiri.gr/kan-san.htm	25-11-2006

Audio Sampling :

www.bigtime.gr/articles/kitharistas.htm	5-2-2007
www.fortunecity.com/emachines/e11/86/synth7.html	6-2-2007
http://www.mellotron.com/	13-9-2007