Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Πτυχιακή εργασία

Επεξεργασία Βίντεο με το πρόγραμμα Max/MSP/Jitter

Βασίλειος Γεννάρης (ΑΜ: 1079)
E-mail: sporbill_1@yahoo.gr

Ηράκλειο – 10/09/2010

Επόπτης Καθηγητής: ΠΑΛΑΜΑΣ ΓΕΩΡΓΙΟΣ

***<u>Υπεύθυνη Δήλωση</u>**: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων του Τ.Ε.Ι. Κρήτης.*

Ευχαριστίες

Η παρούσα πτυχιακή εργασία εκπονήθηκε κατά το χρονικό διάστημα Σεπτέμβριος 2009 – Σεπτέμβριος 2010 στο εργαστήριο ευφυών συστημάτων υπό την επίβλεψη του Καθηγητή Γεωργίου Παλαμά τον οποίο θέλω να ευχαριστήσω θερμά για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο πρωτοποριακό θέμα, με το οποίο δεν έχει ασχοληθεί ξανά κάποιος στο τμήμα του Ε.Π.Π., και μέσω αυτού κατάφερα να κατανοήσω καλύτερα κάποια θέματα πάνω στα πολυμέσα τα οποία με ενδιέφεραν ιδιαίτερα.

Επίσης θα ήθελα να ευχαριστήσω τους γονείς μου Κώστα και Σοφία, τον αδερφό μου Γιάννη και την νύφη μου Μαρίζα για την πολύτιμη βοήθεια, συμπαράσταση και υπομονή που έδειξαν τόσο καιρό.

Περίληψη

Η πτυχιακή εργασία αναφέρεται στην δημιουργία ενός προγράμματος μέσω του λογισμικού Max/MSP/Jitter της εταιρίας cycling'74, το οποίο μπορούμε να βρούμε και να αγοράσουμε μέσα από την ιστοσελίδα της εταιρίας http://cycling74.com/ ,το οποίο θα μπορεί είτε μέσω καταγραφής βίντεο πραγματικού χρόνου μέσω δύο καμερών είτε σε προϋπάρχοντα βίντεο που έχουμε αποθηκευμένα στον υπολογιστή μας ή ακόμη και σε ένα βίντεο πραγματικού χρόνου και ένα προϋπάρχον, να τα συνδυάσει μεταξύ τους μεταβαίνοντας από το ένα στο άλλο έχοντας την δυνατότητα ρύθμισης του χρόνου μετάβασης από το ένα στο άλλο βίντεο. Στην συνέχεια πάνω στο αποτέλεσμα των βίντεο θα έχουμε την δυνατότητα να κάνουμε την χρήση κάποιων βασικών φίλτρων, ακόμη να φτιάξουμε τα δικά μας φίλτρα με την βοήθεια της μεθόδου της ελίκωσης "convolution" και των πινάκων kernel. Το αποτέλεσμα όλων των παραπάνω θα μπορούμε να το βλέπουμε σε ένα ξεχωριστό παράθυρο προβολής. Έχουμε επίσης την δυνατότητα χειρισμού κάποιων αντικείμενων, όπως της μετάβασης από το ένα βίντεο στο άλλο ή την ρύθμιση των τιμών των φίλτρων, του προγράμματος μας μέσω οποιασδήποτε εξωτερικής συσκευής (στο πρόγραμμά μας χρησιμοποιούμε ένα joystick και το πληκτρολόγιο). Ακόμη έχουμε και την δυνατότητα καταγραφής του αποτελέσματος σε .mov αρχείο. Στην συνέχεια δημιουργήσαμε ένα δεύτερο κομμάτι στο πρόγραμμά το οποίο έχει την δυνατότητα να εμφανίζει τα καρέ "frame" του αποτελέσματος των προηγούμενων εργασιών που αναφέραμε με τυχαία σειρά σε ένα δεύτερο παράθυρο προβολής, και ακόμη να ρυθμίζουμε τον ρυθμό και το μέγεθος παραγωγής των τυχαίων frame με την βοήθεια επίσης εξωτερικής συσκευής.

Abstract

The project refers on the development of a programme via the software of Max/MSP/Jitter of the company cycling'74, who we can find and buy it through the website of the company http://cycling74.com/ , that it could combine two videos either through recording real-time video from two cameras or from already recorded videos or even from one real-time and one recorded. Also, during mixing (combining) them together there is the ability to adjust the transition time. After that we can use some basic filters in the previous mixing video result and also create ours through the method of convolution using 'kernel' matrices. The result of all the above we see it in a separate projection window. We also have the ability to control some of our objects with external devices, like the transition of the two videos and the adjustment of some values of the filters, (in our programme we use a joystick and the keyboard). Furthermore we can record the result from the whole programme into a .mov file. Ensuant we create a second part in our programme which can present the frames, of our previous work, with random order in a second projection window, and furthermore we can adjust the rythm and the largeness of the creation random frames with a joystick.

Πίνακας Περιεχομένων

Ευχαριστίες iii

Περίληψη iv

Abstract v

Πίνακας Περιεχομένων vi

Κεφάλαιο 1 – Εισαγωγή 1

1.1 Σκοπός της εργασίας. 1

Κεφάλαιο 2 - Ιστορική Αναδρομή 2

2.1 Περιγραφή του Max/MSP/Jitter 2

2.1.1 Λίγα λόγια για το Max/Msp/Jitter. 2

2.1.2 Ιστορία του Max/Msp/Jitter. 2

2.1.3 Γλώσσα του Max/Msp/Jitter. 4

Κεφάλαιο 3 - Επεξήγηση και ανάλυση των χαρακτηριστικών του Max/MSP/Jitter 6

3.1 Χαρακτηριστικά του Max. 6

3.1.1 Χρόνος 6

3.1.2 Διαδραστικότητα 7

3.1.3 Χειρισμός 7

3.1.4 Επεκτασιμότητα 8

3.2 Χαρακτηριστικά του MSP. 8

3.2.1 Σχεδιασμός συνθετών μπλοκ 8

3.2.2 Δειγματισμός, Ηχογράφηση και Αναπαραγωγή 9

3.2.3 Συμβατότητα 10

3.2.4 Διεπαφή χρήστη 10

3.2.5 Πολυφωνία 11

3.2.6 Επεκτασιμότητα 11

3.3 Χαρακτηριστικά του Jitter 12

3.3.1 Βίντεο 12

3.3.2 2D/3D Γραφικά 12

3.3.3 Ευκολία Χρήσης 13

3.3.4 Matrices 14

3.3.5 Περισσότερες λεπτομέρειες 15

3.3.5.1 Εκτέλεση 16
3.3.5.2 Δικτύωση 17
3.3.5.3 Προγραμματισμός 17
3.3.5.4 Διαλειτουργικότητα 18
Κεφάλαιο 4 - Αποτελέσματα του προγράμματός μας, χρήσεις του Max/MSP/Jitter και Φίλτρα. 19
4.1 Αποτελέσματα προγράμματός πτυχιακής 19
4.2 Περιγραφή και χρήσεις του Max/MSP/Jitter. 20
4.3 Φίλτρα 21
4.3.1 Εισαγωγή 21
4.3.2 Χρήση φίλτρων στο πρόγραμμα μου. 22
Κεφάλαιο 5 - Επεξήγηση και ανάλυση του προγράμματος επεξεργασίας βίντεο σε πραγματικό χρόνο με το Max/MSP/Jitter. 23
5.1 Ενεργοποίηση προγράμματος και καταγραφή βίντεο. 23
5.1.1 Παράθεση και επεξήγηση των αντικειμένων ένα προς ένα. 23
5.1.2 Παράθεση και επεξήγηση του 1ου μέρους του προγράμματος του οποίου τα αντικείμενα αναλύσαμε παραπάνω. 26
5.2 Αναπαραγωγή και συνδυασμός βίντεο. 28
5.2.1 Παράθεση και επεξήγηση των αντικειμένων ένα προς ένα. 28
5.2.2 Παράθεση και επεξήγηση του 2ου μέρους του προγράμματος του οποίου τα αντικείμενα αναλύσαμε παραπάνω. 31
5.3Χρησιμοποίηση διασυνδετικών συσκευών αλληλεπίδρασης χρήστη με μηχανή(Human Interface Devices). 33
5.3.1 Παράθεση και επεξήγηση των αντικειμένων που χρησιμοποιούμε για την χρήση HID συσκευών. 33
5.3.2 Παράθεση και επεξήγηση του 3ου μέρους του προγράμματος του οποίου τα αντικείμενα αναλύσαμε παραπάνω. 37
5.4 Χρήση φίλτρων στο πρόγραμμα μας με την μέθοδο του Convolution. 39
5.4.1 Επεξήγηση των αντικειμένων ένα προς ένα. 39
5.4.2 Παράθεση και επεξήγηση του 4ου μέρους του προγράμματος του οποίου τα αντικείμενα αναλύσαμε παραπάνω. 43

5.5 Διάφορες λειτουργίες παρουσίασης του αποτελέσματος του προγράμματος μας. 45

5.5.1 Παράθεση και επεξήγηση των αντικειμένων ένα προς ένα. 45

5.5.2 Παράθεση και επεξήγηση του 5ου μέρους του προγράμματος μας του οποίου τα αντικείμενα αναλύσαμε παραπάνω. 47

5.6 Εμφάνιση των frames με τυχαία σειρά στο παράθυρο προβολής. 48

5.6.1 Παράθεση και επεξήγηση των αντικειμένων ένα προς ένα. 48

5.6.2 Παράθεση και επεξήγηση του 6ου μέρους του προγράμματος μας του οποίου τα αντικείμενα αναλύσαμε παραπάνω. 53

5.6 Επεξήγηση για τον τρόπο εμφάνισης, και διεπαφής του Max/MSP/Jitter. 55

Βιβλιογραφία 59

Κεφάλαιο 1
Εισαγωγή

1.1 Σκοπός της εργασίας.

Ο Σκοπός της εργασίας είναι κυρίως εκπαιδευτικός. Δηλαδή η δημιουργία ενός εγγράφου το οποίο θα μπορεί να χρησιμοποιηθεί και σαν εκπαιδευτικό σύγγραμμα ‘’tutorial’’ το οποίο θα βοηθάει όσο είναι δυνατόν, στην επεξήγηση και κατανόηση του λογισμικού του Max/MSP/Jitter, του τρόπου λειτουργίας του προγράμματος, κάποιων βασικών και μη εργαλείων και αντικειμένων και τον τρόπο σύνδεσης και επικοινωνίας αυτών, κάποιον τον οποίο δεν γνωρίζει το συγκεκριμένο λογισμικό και έτσι θα μπορεί να έχει μια γενική εικόνα και επίσης θα μπορεί να δημιουργήσει τα δικά του προγράμματα μέσω του Max/MSP/Jitter. Επίσης θα παραθέσουμε και λίγα λόγια για την χρήση εξωτερικών συσκευών και τον τρόπο λειτουργίας τους έτσι ώστε να μπορέσουμε να τις χρησιμοποιήσουμε στο πρόγραμμα μας. Ακόμη μέσα στην εργασία υπάρχει και μια απλή επεξήγηση και με παραδείγματα για το convolution και το πως εφαρμόζεται σε μια εικόνα μέσω των πινάκων kernel.

Κεφάλαιο 2
Ιστορική Αναδρομή

2.1 Περιγραφή του Max/MSP/Jitter

2.1.1 Λίγα λόγια για το Max/Msp/Jitter.

Το Max είναι μια οπτικά “Visual” προγραμματιζόμενη γλώσσα για μουσική και πολυμέσα, η οποία αναπτύσσεται και συντηρείται από την εταιρία, με βάση το San Francisco, την Cycling ’74. Κατά την εικοσάχρονη πορεία του στην ιστορία, έχει χρησιμοποιηθεί ευρέως από συνθέτες, ερμηνευτές, σχεδιαστές λογισμικού, ερευνητές και καλλιτέχνες για την δημιουργία καινοτόμων συνθέσεων, ερμηνειών και εφαρμογών.

Το πρόγραμμα Max αποτελείται από πρότυπα συναρτησιακά στοιχεία, με τις περισσότερες ρουτίνες αυτών να υπάρχουν με την μορφή διαμοιραζόμενων βιβλιοθηκών. Μια εφαρμογή επιτρέπει την δημιουργία νέων ρουτινών που ονομάζονται ¨εξωτερικά αντικείμενα¨. Σαν αποτέλεσμα, το Max έχει μια μεγάλη βάση χρηστών-προγραμματιστών που δεν εντάσσονται στην εταιρία Cycling’74 και που ενισχύουν το λογισμικό με εμπορικό και μη κώδικα. Λόγω του επεκτατικού του σχεδιασμού και της γραφικής διεπαφής του, το Max το λαμβάνουνε υπόψιν, οι άνθρωποι που ασχολούνται με τον χώρο αυτόν, ως το πλέον διαδεδομένο λογισμικό για την δημιουργία διαδραστικών μουσικών και πολυμεσικών εφαρμογών.

2.1.2 Ιστορία του Max/Msp/Jitter.

Το Max αρχικά γράφτηκε από τον Miller Puckette σαν τον επιπρόσθετο editor για τα Macintosh στο IRCAM(Ινστιτούτο έρευνας και συνεργασίας για την ακουστική και την μουσική) στα μέσα της δεκαετίας του ’80 για να δώσει στους συνθέτες πρόσβαση σε ένα συγγραφικό σύστημα για την δημιουργία διαδραστικής μουσικής μέσω υπολογιστή. Πρώτα χρησιμοποιήθηκε σε ένα πιάνο μαζί με μια πλακέτα υπολογιστή η οποία ονομάζεται Pluton, συγχρονίζοντας τον υπολογιστή στο πιάνο και ελέγχοντας μέσω μιας κάρτας, η οποία έκανε την επεξεργασία του ήχου.

Το 1989, το IRCAM ανέπτυξε και συντήρησε μια έκδοση του Max που την εισήγαγε στον IRCAM σταθμό επεξεργασίας σήματος που ονομάστηκε Max/FTS.

Το 1989, αδειοδοτήθηκε η εταιρία Opcode Systems από την IRCAM, η οποία πούλησε μια εμπορική έκδοση του προγράμματος που ονομαζόταν Max (αναπτύχθηκε και συντηρήθηκε από τον David Zicarelli). Στα μέσα της δεκαετίας του ΄90 εξελίχθηκε αρκετά το λογισμικό. Η εμπορική έκδοση του Max από το 1999 διανέμεται από την εταιρία του Zicarelli, την Cycling ’74.

Ο Puckette κυκλοφόρησε ένα ολοκληρωτικά ξανά-σχεδιασμένο πρόγραμμα το 1996 που ονομαζόταν Pd (¨Pure Data¨), το οποίο εκτός από έναν αριθμό από βασικές διαφορές από της IRCAM το αρχικό πρόγραμμα, είναι επιφανειακά παρόμοιο και παραμένει ένα ανοικτού κώδικα εναλλακτικό πρόγραμμα ως προς το Max/MSP.

Το Max έχει έναν μεγάλο αριθμό από προεκτάσεις, από τις πιο αξιοσημείωτες είναι ένα σετ από audio προεκτάσεις για το λογισμικό όπου εμφανίστηκε το 1997, προερχόμενο από ένα κομμάτι της μεταγενέστερης δουλειάς του Puckette πάνω στο Pure Data. Αυτό ονομάστηκε MSP (το όνομα του MSP προέρχεται είτε από το Max Singal Processing ή από τα αρχικά του Miller S. Puckette), αυτό το πρόσθετο πρόγραμμα για το Max επέτρεπε την διαχείριση ψηφιακών ηχητικών σημάτων σε πραγματικό χρόνο ”real-time”, επιτρέποντας στους χρήστες να δημιουργήσουν τα δικά τους synthesizers και τους δικούς τους επεξεργαστές εφέ (το MSP είχε σχεδιαστεί να συνεργάζεται με hardware synthesizers, samplers κ.ο.κ. σαν μια γλώσσα ελέγχου χρησιμοποιώντας MIDI ή κάποιο άλλο πρωτόκολλο).

Το 1998, ένας απόγονος του Max/FTS δημιουργήθηκε σε Java (jMax) και εκδόθηκε σαν open-source.

Το 1999 είδαμε την έκδοση του nato.0+55, μια σουίτα από προεκτάσεις που δημιουργήθηκε από την Netochka Nezvanova η οποία σουίτα έφερε στο Max εκτεταμένο έλεγχο του real-time βίντεο. Το nato έγινε πολύ γρήγορα γνωστό στον χώρο των καλλιτεχνών των πολυμέσων, η δημιουργία του ολοκληρώθηκε το 2001. Υπήρξε και το SoftVNS, ένα πακέτο για επεξεργασία γραφικών στο Max το οποίο δημιουργήθηκε από έναν Καναδό καλλιτέχνη των media και κυκλοφόρησε το 2002.

Εν τω μεταξύ, η Cycling ’74 ανέπτυξε τις δικές του προσθήκες για την επεξεργασία του βίντεο. Ένα πρωτεύον πακέτο για το Max/MSP που ονομαζόταν Jitter κυκλοφόρησε το 2003, παρέχοντας real-time βίντεο, 3-D, και δυνατότητα επεξεργασίας πινάκων πλέγματος “matrix”.

Σε αντίθεση, ένας αριθμός από προγράμματα σαν το Max δημιουργήθηκαν τα οποία μοιράζονταν το ίδιο σκεπτικό για τον visual προγραμματισμό και την real-time επεξεργασία όπως το Quartz Composer (από την Apple) και το vvvv όπου και τα δύο συγκεντρώνονται στην real-time βίντεο σύνθεση και επεξεργασία.

Μια μεγάλη αναβάθμιση για το Max/MSP/Jitter, είναι το Max 5, το οποίο βγήκε στην αγορά το 2008.

2.1.3 Γλώσσα του Max/MSP/Jitter.

Το όνομα του το Max το πήρε από τον Max Mathew και μπορεί να θεωρηθεί ως ένας απόγονος του προγράμματος MUSIC, είναι το γραφικό περιβάλλον το οποίο μπορεί να εξηγήσει το γεγονός αυτό. Όπως στις περισσότερες MUSIC-N γλώσσες, όπου MUSIC-N είναι μια οικογένεια από μουσικά προγράμματα για υπολογιστές και γλώσσες προγραμματισμού προερχόμενες και εμπνευσμένες από το πρόγραμμα MUSIC όπου έχει γραφτεί από τον Max Mathew το 1957 στα εργαστήρια της Bell. το Max/MSP/Jitter ξεχωρίζει δύο επίπεδα για τον χρόνου: ο ένας για τα γεγονότα που είναι καθορισμένα, και ο άλλος για τα DSP(Digital Signal Processing).

Η βασική γλώσσα του Max και των αδελφικών προγραμμάτων είναι αυτή της ροής δεδομένων: Τα προγράμματα του Max που ονομάζονται patches φτιάχνονται στήνοντας και ενώνοντας μπλοκ από αντικείμενα μέσα στο patcher, όπου patcher είναι η επιφάνεια δημιουργίας των προγραμμάτων μέσω του Max/MSP/Jitter. Αυτά τα αντικείμενα δρουν σαν αυτόνομα προγράμματα (στην πραγματικότητα είναι δυναμικά ενωμένες βιβλιοθήκες), κάθε μια αυτές μπορεί να δέχεται είσοδο (μέσω μιας ή περισσότερων εικονικών εισόδων), και να δημιουργεί έξοδο (μέσω οπτικών εξόδων), ή και τα δύο. Τα αντικείμενα περνάνε τις εξόδους τους στις εισόδους των συνδεδεμένων αντικειμένων.

Το Max υποστηρίζει έξι βασικούς τύπους δεδομένων, οι οποίοι μπορούν να μεταδοθούν σαν μηνύματα από αντικείμενο σε αντικείμενο: int, float, list, symbol, bang, signal (για MSP audio συνδέσεις). Ένας αριθμός πολυπλοκότερων δομών δεδομένων υπάρχει μέσα στο πρόγραμμα για να διαχειρίζεται αριθμητικά σύνολα (table data), πίνακες με διάφορα αντικείμενα (coll data), και XML πληροφορίες (pattr data). Μια MSP δομή δεδομένων (buffer~) μπορεί να κρατήσει ψηφιακή πληροφορία ήχου μέσα στην μνήμη του προγράμματος. Απ’ την άλλη το πακέτο Jitter προσθέτει μια πολυδιάστατη δομή δεδομένων, η οποία μπορεί να προσαρμοστεί στον όγκο των

πληροφοριών που έχουμε κάθε φορά, για την διαχείριση μεγάλων σετ αριθμών την αποθήκευση βίντεο και άλλων δεδομένων (matrix data).

Το Max έχει ένα λεξικό από αντικείμενα όπου το όνομά τους ταιριάζει και με την λειτουργία τους μέσα στο patcher. Για παράδειγμα, του αντικείμενου metro η λειτουργία του είναι μετρονόμος, και το αντικείμενο random δημιουργεί τυχαίους ακεραίους. Τα περισσότερα αντικείμενα είναι μη γραφικά, αποτελούμενα μόνο από το όνομα του αντικειμένου και έναν αριθμό από δεδομένα και ιδιότητες γραμμένα μέσα στο κουτί του αντικειμένου. Άλλα αντικείμενα είναι γραφικά, συμπεριλαμβανομένου τους sliders, number boxes, dials, table editors, pull-down menus, buttons και άλλα αντικείμενα ώστε να μπορεί να τρέχει το πρόγραμμα διαδραστικά. Το Max/MSP/Jitter κυκλοφορεί με περίπου 600 από αυτά τα αντικείμενα στην βασική έκδοση. Επεκτάσεις για το πρόγραμμα μπορούν να γραφτούν από τρίτους προγραμματιστές “third part programmers” σαν Max patches, ή σαν αντικείμενα γραμμένα σε C, C++, Java ή Javascript.

Η σειρά εκτέλεσης για τα μηνύματα που περιέχονται μέσα στο γράφημα ‘patch’ αντικείμενων ορίζεται από την οπτική οργάνωση μέσα στο patcher. Στις νεότερες εκδόσεις του Max προσφέρεται και ένας αριθμός τεχνολογιών για πιο στάνταρ GUI σχεδιασμό.

Ένας μεγάλος αριθμός ανθρώπων χρησιμοποιούν το Max, ακόμα και αν δεν είναι γνώστες αυτού. Αρχεία του Max που ονομάζονται patchers μπορούν να χρησιμοποιηθούν σε αυτόνομες εφαρμογές και να διανεμηθούν δωρεάν ή να πουληθούν εμπορικά. Συμπληρωματικά το Max μπορεί να χρησιμοποιηθεί για συγγραφή audio επιπρόσθετου “plug-in” λογισμικού για πρωτεύοντα συστήματα παραγωγής ήχου.

Με την αυξανόμενη χρήση των φορητών υπολογιστών σε ζωντανές παραγωγές μουσικής (στην ηλεκτρονική μουσική και οπουδήποτε αλλού), το Max/MSP και το Max/Jitter έλαβε ιδιαίτερη προσοχή ως αναπτυσσόμενο περιβάλλον σε αυτούς που ενδιαφέρονται για την δημιουργία ζωντανών βίντεο παρουσιάσεων και ζωντανής ηλεκτρονικής μουσικής.

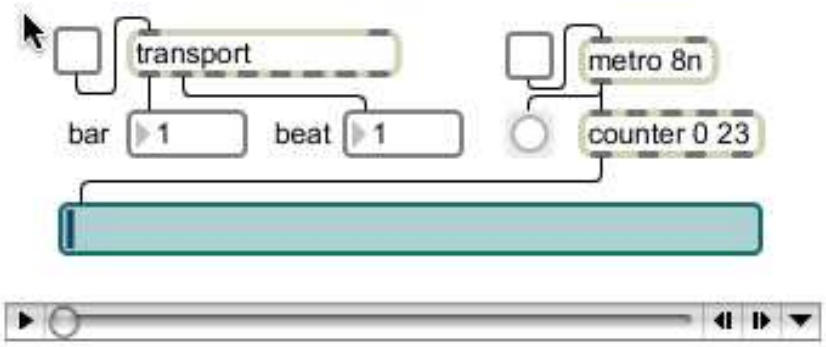
Κεφάλαιο 3
Επεξήγηση και ανάλυση των χαρακτηριστικών του Max/MSP/Jitter

3.1 Χαρακτηριστικά του Max.

3.1.1 Χρόνος

Δουλεύοντας με τον χρόνο είναι ευκολότερο με το Max από ότι με τον παραδοσιακό προγραμματισμό επειδή χρησιμοποιείς αντικείμενα για να δημιουργήσεις οπτικές ¨μηχανές χρόνου¨ των οποίων τις συμπεριφορές μπορείς να δεις, να ακούσεις και να μετατρέψεις καθώς λειτουργούν.

Στο Max, η βασική μονάδα χρόνου είναι το millisecond, άλλα μπορείς να χρησιμοποιήσεις και τον μετρικό χρόνο (για παράδειγμα, μπάρες και ρυθμούς). Στο Max, ότι δημιουργείς μπορεί να τρέξει την ίδια στιγμή, κάτι το οποίο είναι πολύ δύσκολο να κάνεις με μια παραδοσιακή γλώσσα. Επιπρόσθετα με το αντικείμενο metro που βλέπουμε παρακάτω, τα αντικείμενα του Max που έχουν σχέση με τον χρόνο, όπως το delay, μετράνε τα χρονικά διαστήματα και εφαρμόζουν φίλτρα βασισμένα στον χρόνο. Η γενική ιδέα του Max βασίζεται στους αριθμούς και στα γεγονότα που σου επιτρέπουν να δουλεύεις με τον χρόνο με ένα διαισθητικό και συνεπή τρόπο.



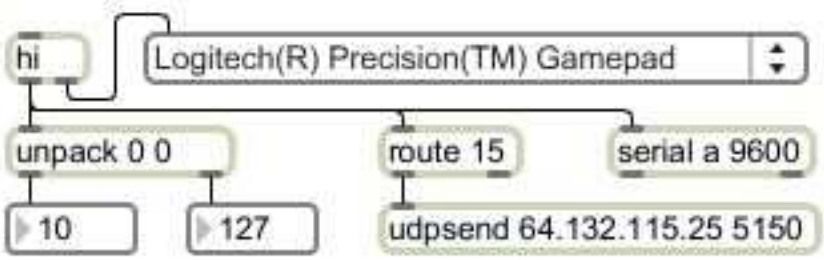
3.1.2 Διαδραστικότητα

Δουλεύοντας με διαδραστικότητα είναι ευκολότερο στο Max γιατί μπορείς να σχεδιάσεις οπτικές διεπαφές οι οποίες είναι κομμάτι του ίδιου του προγράμματος. Για παράδειγμα, ο έλεγχος του χρόνου και της μετάδοσης του παραδείγματος παραπάνω για τον χρόνο.

Το Max έχει ένα πλούσιο σετ από εργαλεία δημιουργίας διεπαφής για τον χρήστη, συμπεριλαμβανομένου πληκτρολόγια, κουμπιά, μενού. Σε ένα πιο χαμηλό επίπεδο, είναι πιθανό να ελέγχεις ακόμα και HI συσκευές, Quicktime βίντεο, κ.α.. Μπορείς ακόμη να γράψεις script στο Max για να δημιουργήσεις δυναμικές διεπαφές.

3.1.3 Χειρισμός

Στο Max τα πάντα μπορούν να συνδεθούν μεταξύ τους, γιατί όλα τα αντικείμενα επικοινωνούν με αριθμούς. Ένας άλλος τρόπος να εξηγήσεις το προηγούμενο είναι ότι η προτυποποίηση είναι η βασική αρχή του σχεδιασμού στο λογισμικό αυτό.



Το Max μπορεί να ελέγξει MIDI συσκευές, σειριακές συσκευές, να στείλει δεδομένα μέσω ενός δικτύου και να χειριστεί συσκευές εισόδου. Κάθε πηγή ελέγχου εύκολα συνδέεται σε οτιδήποτε θέλουμε να ελέγξουμε. Το πιο κοινά, παρόλα αυτά, δυναμικά χαρακτηριστικά ελέγχου εφαρμόζονται στον ήχο μέσω του MSP, και στα οπτικά μέσα με το Jitter. Και το MSP και το jitter έχουν τις δικές τους δυνατότητες ελέγχου, από ηχητικούς αναλυτές σήματος μέχρι και αναλυτές βίντεο και υποστήριξη μηχανών.

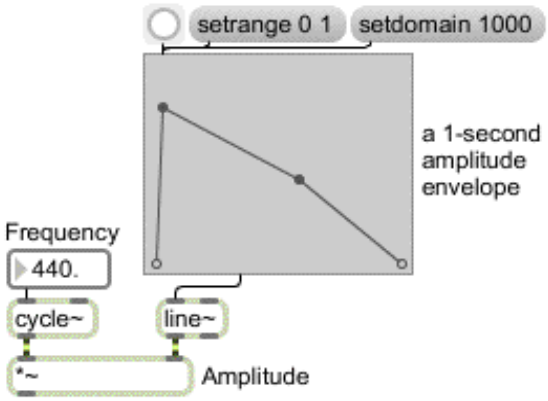
3.1.4 Επεκτασιμότητα

Οι χρήστες του Max μπορούν να επεκτείνουν το περιβάλλον του Max σε κάθε πιθανή κατεύθυνση. Οι παθιασμένοι χρήστες του Max έχουν δημιουργήσει χιλιάδες αντικείμενα γραμμένα σε C, Java ή Javascript τα οποία υπάρχουν στην κοινότητα χρηστών.

3.2 Χαρακτηριστικά του MSP.

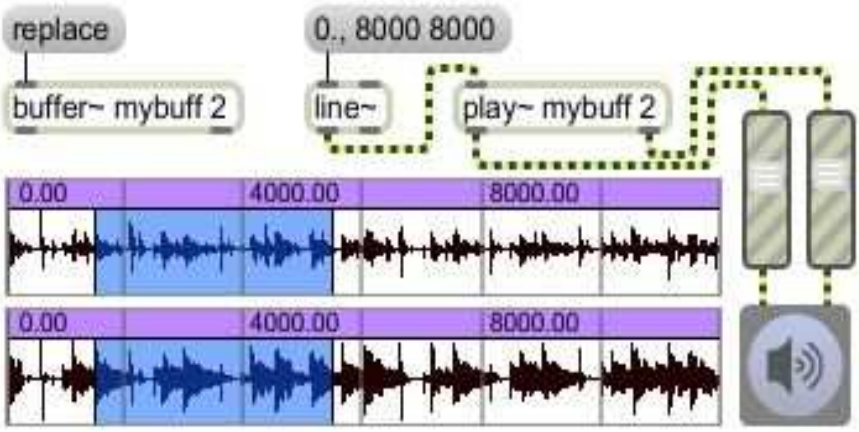
3.2.1 Σχεδιασμός συνθετών μπλοκ

Το MSP προσφέρει έτοιμα τα μπλοκ τα οποία χρησιμεύουν για δημιουργία και επεξεργασία ηχητικών σημάτων: π.χ. ταλαντωτές, φίλτρα, καθυστερήσεις κ.α. Μπορείς να καθορίσεις τις τιμές του χρόνου για τις περιβάλλουσες και την φάση με προσωρινές τιμές. Η ικανότητα να συνδυάζεις χαμηλού και υψηλού επιπέδου μέρη είναι μια από τις μοναδικότητες της εργασίας με το MSP.



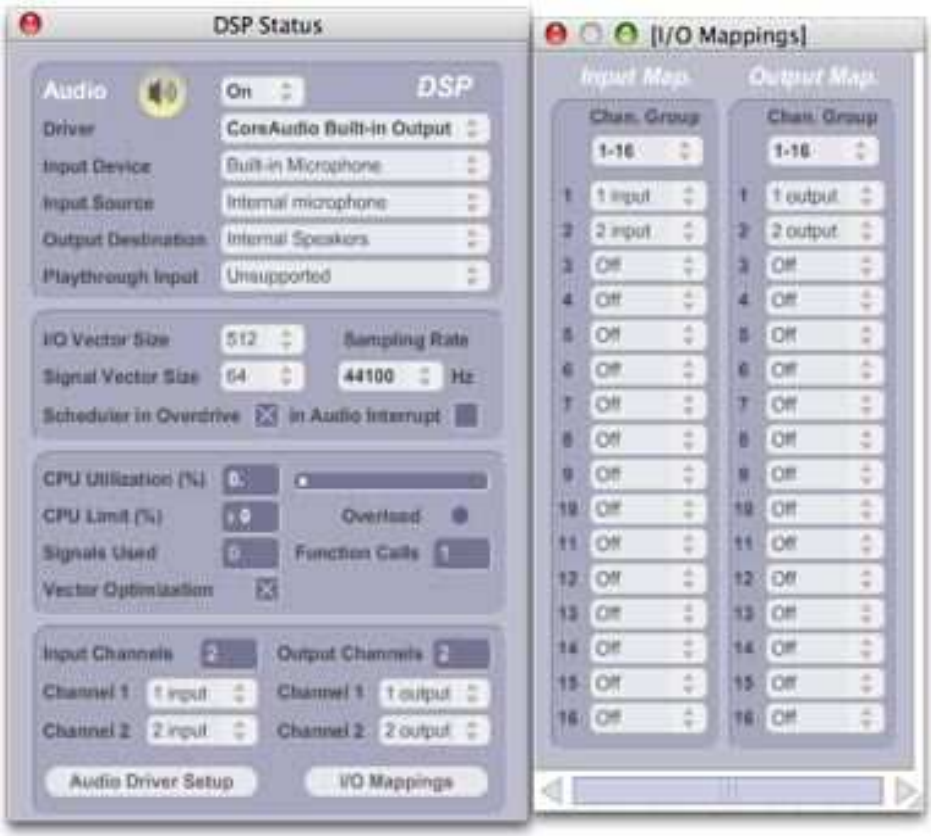
3.2.2 Δειγματισμός, Ηχογράφηση και Αναπαραγωγή

Το MSP προσφέρει ένα γκρουπ από αντικείμενα που δουλεύουν μεταξύ τους για δειγματισμό και αναπαραγωγή. Όλοι χρησιμοποιούν το αντικείμενο buffer~ το οποίο μπορεί να ελέγξει δείγματα από την μνήμη buffer έως και τεσσάρων καναλιών. Μπορείς να αναπαράγεις δείγματα από την buffer μνήμη με διάφορους τρόπους, να ηχογραφήσεις στην buffer, ή να χρησιμοποιήσεις ένα δείγμα buffer σαν την διεγερτική παρόρμηση για ένα φίλτρο. Καθορισμένοι τύποι αρχείων audio υποστηρίζονται για αναπαραγωγή και ηχογράφηση δειγμάτων στην buffer. Όπου buffer~ μια δομή δεδομένων η οποία μπορεί αποθηκεύσει ψηφιακή πληροφορία ήχου μέσα στην μνήμη του προγράμματος. Το MSP παρέχει ηχογράφηση και αναπαραγωγή σε σκληρό δίσκο, υποστηρίζοντας αρχεία μέχρι και οχτώ καναλιών. Οι δυνατότητες του υπολογιστή σου είναι το όριο στον αριθμό των αναπαραγωγών και των ηχογραφήσεων που μπορείς να χρησιμοποιήσεις ταυτόχρονα. Επιπρόσθετα είναι και ευκολότερο να συλλάβεις τον ήχο που παράγεις σε ένα πολυκαναλικό αρχείο.



3.2.3 Συμβατότητα

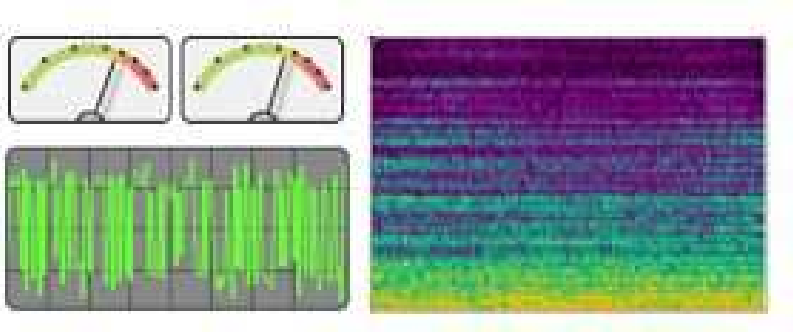
Το MSP υποστηρίζει μέχρι και 512 εισόδους και εξόδους καναλιών. Στα MACINTOSH, το MSP παρέχει υποστήριξη για Core Audio και στα Windows, υπάρχει μια ASIO, DirectSound, και MME υποστήριξη. Το MSP μπορεί να είναι ένας ReWire host ή client, και είναι πολύ ευέλικτος host για VST plug-ins. Το Max 5 έχει προσθέσει έναν βασισμένο σε host συντονισμό για VST plug-ins επισυναπτόμενο στο νέο σύστημα χρόνου που είναι βασισμένο στον ρυθμό.



3.2.4 Διεπαφή χρήστη

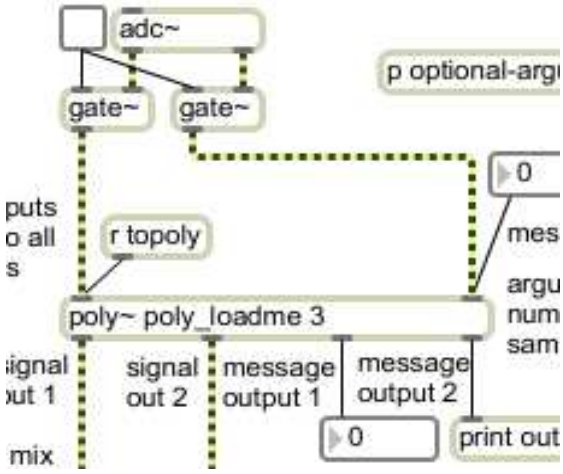
Τα αντικείμενα του MSP για την παρουσίαση ηχητικών σημάτων περιλαμβάνουν ταλαντοσκόπια, οθόνες παρουσίασης φάσματος, μετρητές επιπέδων, οθόνες παρουσίασης κυματομορφών, και ένα αντικείμενο παρουσίασης της ενέργειας των συχνοτήτων. Μπορείς να χρησιμοποιήσεις ρυθμιστές αυξομείωσης, να σχεδιάσεις γραφικά φίλτρα πολλών συχνοτήτων και να επεξεργαστείς συναρτήσεις τοποθετώντας αυθαίρετο αριθμό από breakpoints. Συν, ότι είναι ευκολότερο να κάνεις μετατροπή μεταξύ audio σημάτων και μονόμετρων τιμών, τα πάντα στην

πλούσια συλλογή εργαλείων διεπαφής του Max μπορούν να χρησιμοποιηθούν για τον έλεγχο και την εμφάνιση του ήχου. Στο Max 5, είναι εύκολο να μάθεις τα επίπεδα κάθε ηχητικού σήματος με το νέο εργαλείο επεξεργασίας σήματος.



3.2.5 Πολυφωνία

Το αντικείμενο poly~ σου επιτρέπει να χρησιμοποιείς πολλά αντίγραφα από κάθε patcher που δημιουργείς. Υποστηρίζει έξοδο μίξης, κατανομή για τις νότες, επίσης δυναμικά εισάγει patches και κάνει κατανομή ήχων, και μπορεί να "τρέξει" patchers χωρισμένα σε κομμάτια για την υποστήριξη της πολλαπλής επεξεργασίας.



3.2.6 Επεκτασιμότητα

Εάν θέλετε να δημιουργήσετε τους δικούς σας audio αλγόριθμους, το Max/MSP προσφέρει μια ιδανική πλατφόρμα βασισμένη στην C API για την δημιουργία νέων μονάδων. Το Max φροντίζει για τα I/O αρχεία και τις κατασκευές διεπαφών χρήστη. Μια άλλη επιλογή για χαμηλότερου επιπέδου δημιουργίες είναι το mxj~, ένα σύστημα βασισμένο στην Java για την ανάπτυξη συστημάτων audio.

3.3 Χαρακτηριστικά του Jitter

3.3.1 Βίντεο

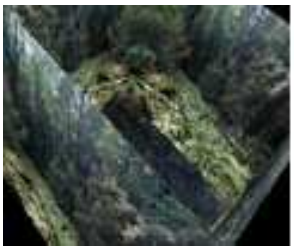
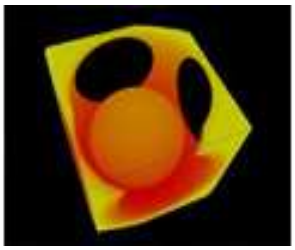
Παρόλο που η αρχιτεκτονική του Jitter είναι γενική, είναι βελτιστοποιημένη για χρήση με δεδομένα βίντεο, και εκτελείται σε υπερβολικά γρήγορες ταχύτητες. Ένα εύρωστο σετ από μαθηματικούς τελεστές, κουμπιά που εκτελούν κάποιες εντολές/συνθέσεις, ανάλυση, διορθώσεις και αλλαγές χρωμάτων, alpha channel επεξεργασίες, χωρικές ρυθμίσεις, φίλτρα βασισμένα στο convolution, και ειδικά εφέ τα οποία μπορείς να τα χειριστείς όπως εσύ θέλεις για να τα εφαρμόσεις στο βίντεο.

Το Jitter περιλαμβάνει ευρεία υποστήριξη για την αρχιτεκτονική του Quicktime της Apple, όπως αναπαραγωγή όλων των υποστηριζόμενων από το QT φόρμα αρχείων, real και non real-time δημιουργία αρχείων, λειτουργίες επεξεργασίας, εισαγωγή και εξαγωγή δυνατοτήτων, εφαρμογή real-time QT εφέ, ψηφιοποίηση βίντεο, QTVR, αλλαγή τύπου αρχείου, και άλλα. Το Quicktime audio μπορεί να δρομολογηθεί μέσα στο MSP για να εκμεταλλευτεί τις απίστευτες δυνατότητες του. Όσο αναφορά το περιβάλλον παραγωγής, το Jitter παρέχει δυνατότητες για έλεγχο ψηφιακού βίντεο (DV) κάμερας μέσω εισόδου και εξόδου FireWire, και πολλαπλές συσκευές παρουσίασης υποστηρίζονται για περιπτώσεις παρουσιάσεων.

3.3.2 2D/3D Γραφικά

Το πακέτο γραφικών 2D/3D του Jitter παρέχει τα εργαλεία , για να μπορείς να χρησιμοποιήσεις το hardware των ανεπτυγμένων γραφικών του OpenGL μαζί με βίντεο, συμπεριλαμβάνοντας την δυνατότητα να πλέκεις 3D γεωμετρίες με κινούμενο βίντεο σε πραγματικό χρόνο, να μετατρέπεις ήχο και βίντεο κομμάτια απευθείας σε γεωμετρικά δεδομένα, και μοντέλα render, NURBS, 2D/3D κείμενο, και άλλα παρόμοια σχήματα. Υπάρχει ακόμη χαμηλού επιπέδου πρόσβαση σε γεωμετρικά

δεδομένα και στην πλειοψηφία των OpenGL API για αυτούς που χρειάζονται να είναι πιο κοντά στον υπολογιστή.

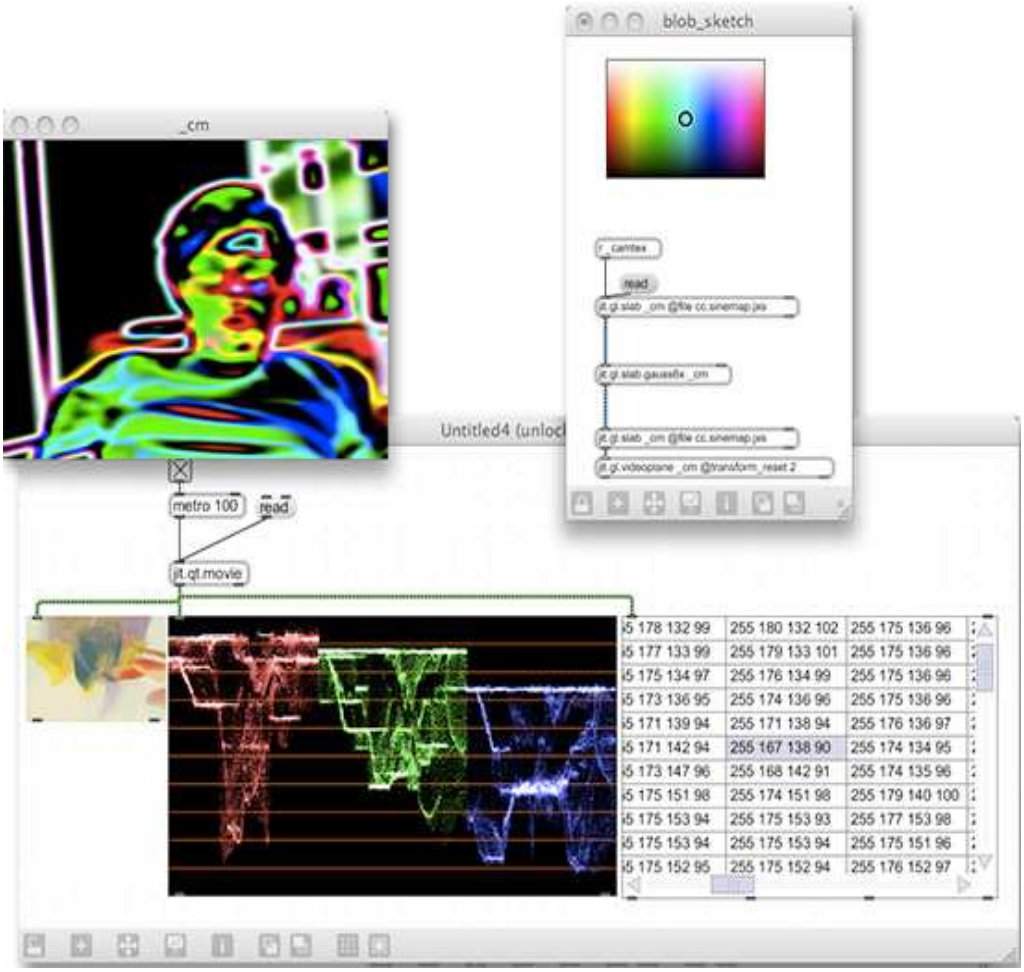



3.3.3 Ευκολία Χρήσης

Το Jitter ολοκληρώνεται απόλυτα με το γραφικό περιβάλλον του Max/MSP το οποίο σου επιτρέπει να συνδέσεις οπτικά αντικείμενα επεξεργασίας δεδομένων μεταξύ τους με εικονικά καλώδια για να δημιουργήσεις τυπικές εφαρμογές που είναι παρόμοιες με τα αναλογικά synthesizer.

Αυτό το οπτικό πλαίσιο παρέχει την δυνατότητα να φτιάξεις τα δικά σου μοναδικά εφέ για βίντεο, μίκτες βίντεο πραγματικού χρόνου, οπτικούς παρουσιαστές ηχητικών σημάτων, σύνθεση εικόνας και ήχου, αλγόριθμους δημιουργίας εικόνων, επεξεργασία και μετατροπή αρχείων, ή ότι άλλο επιθυμούμε. Μπορείς να μοιραστείς τα προγράμματα που δημιουργείς με άλλους χρήστες του Max/MSP και να δημιουργήσεις ολοκληρωμένες εφαρμογές μόνο και μόνο επειδή είναι δυνατό να το κάνεις με το Max/MSP. Μια δωρεάν Runtime έκδοση είναι διαθέσιμη η οποία τρέχει κάθε εφαρμογή που έχει δημιουργηθεί με το Max/MSP/Jitter σε οποιονδήποτε υπολογιστή αρκεί να την έχει εγκατεστημένη.

Το Jitter περιλαμβάνει διαδραστικά αρχεία βοήθειας για κάθε ένα από τα αντικείμενα του, αναλυτική τεκμηρίωση, 37 έγγραφα επεξήγησης, και μια μεγάλη ποικιλία από χρήσιμα παραδείγματα.

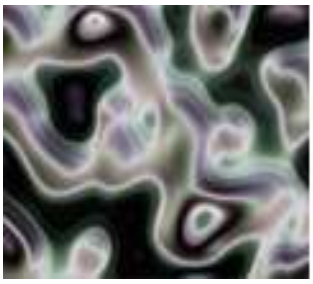
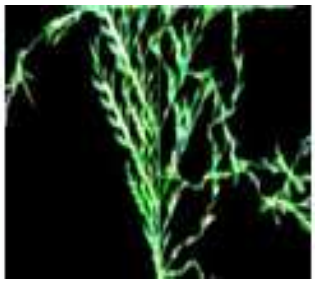


3.3.4 Matrices

Οι δυνατότητες και η ευελιξία του Jitter προέρχεται από την χρήση ενός μονού γενικευμένου πλέγματος δεδομένων όταν δουλεύουμε με βίντεο, 3D γεωμετρίες, audio, κείμενο, ή οποιοδήποτε άλλο είδος δεδομένων. Τα matrices του Jitter μπορούν να σχηματιστούν με έναν από τους 4 παρακάτω τύπους δεδομένων: char (8 bit insigned int), long (32 bit signed int), float 32 (32 bit floating point), ή float 64 (64 bit floating point). Τα πλέγματα μπορούν να είναι μέχρι 32 διαστάσεων, και μπορεί να έχουν μέχρι και 32 επιφάνειες “planes”.

Αυτή η κοινή παραδοχή κάνει την μετατροπή της πληροφορίας πολλή πιο εύκολη. Μπορείτε να πειραματιστείτε με το να παρουσιάσετε ένα κείμενο σαν μια εικόνα, να μετατρέψετε εικόνες βίντεο σε 3D γεωμετρία, να αναπαράγεται βίντεο δεδομένα σαν audio. Οι δυνατότητες είναι απεριόριστες.

Το Jitter έχει όλα τα βασικά μαθηματικά εργαλεία που χρειάζονται για να δουλέψουμε με αριθμητική αναπαράσταση δεδομένων. Το jit.op αντικείμενο από μόνο του παρέχει πάνω από 60 αριθμητικούς, δυαδικούς, λογικούς, εκθετικούς και τριγωνομετρικούς τελεστές. Το πλήθος των τελεστών στο jit.op είναι ιδιαιτέρως χρήσιμο για τον πειραματισμό με την σύνθεση βίντεο. Το jitter υποστηρίζει ακόμη γραμμική άλγεβρα, μοριακά συστήματα, ανάλυση και ανασύνθεση Fourier, κυτταρικούς αυτοματισμούς, και συστήματα Lindenmeyer όπου δίνουν την δυνατότητα για πιο εκτενή πειραματισμό.

3.3.5 Περισσότερες λεπτομέρειες

Τα αντικείμενα του Jitter έχουν διαθέσιμα πολλά χαρακτηριστικά της εσωτερικής τους κατάστασης με τρόπους που είναι καινούργιοι ακόμη και για τους έμπειρους βετεράνους του Max/MSP. Το Jitter παρουσιάζει την έννοια των γνωρισμάτων του, στις εσωτερικές μεταβλητές οι οποίες μπορούν να ρυθμιστούν και να τακτοποιηθούν, και έτσι σαν αποτέλεσμα έχουμε την ευκολότερη διαχείριση της κατάστασης του αντικειμένου. Σαν μέσο διευκόλυνσης, τα αντικείμενα του Jitter μπορούν να διαμορφωθούν με τιμές ιδιοτήτων της μορφής “@” μειώνοντας σημαντικά την ανάγκη για υπερβολική χρήση του αντικειμένου loadbang.

Τα αντικείμενα του Jitter μπορούν να δουλέψουν με matrices αυθαίρετου μεγέθους και είναι σχεδιασμένα έτσι ώστε να προσαρμόζονται στο τύπο και στο μέγεθος των δεδομένων των οποίων λαμβάνουν. Ένα μόνο πρόγραμμα μπορεί να έχει αντικείμενα τα οποία δουλεύουν με διαφορετικούς τύπους και μεγέθη δεδομένων κάθε φορά, και για αυτό υπάρχουν εργαλεία για εύκολη μετατροπή από τον ένα τύπο ή μέγεθος στον άλλο.

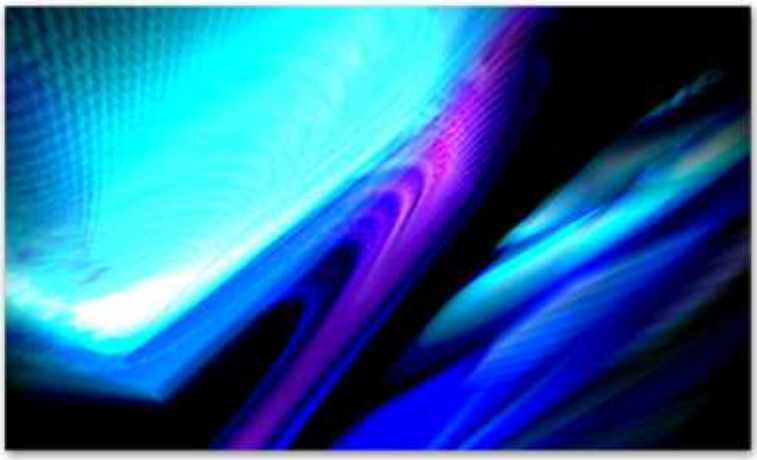
Όλα τα matrices τα οποία περνάμε μεταξύ των αντικειμένων ονομάζονται οντότητες παρόμοιες με το αντικείμενο buffer~ του MSP. Αναφερόμενοι κατά όνομα, ένα μονό πλέγμα μπορεί να είναι προσβάσιμο από περισσότερα από ένα αντικείμενα,

επιτρέποντας έτσι την δημιουργία διαδραστικών δικτύων, δουλεύοντας επεξεργασίες, και συμβάλλοντας στην συντήρηση μνήμης.

Υπάρχει δημοσίως διαθέσιμο ένα Jitter SDK με τον πηγαίο κώδικα πάνω από 30 αντικείμενων προερχόμενο απευθείας από την εργαλειοθήκη του Jitter, έτσι διάφοροι προγραμματιστές έχουν την δυνατότητα να επεκτείνουν τις ήδη σαρωτικές δυνατότητες που παρέχει το Jitter. Αυτό το είδος επεκτασιμότητας είναι ήδη γνωστό μέσω του Max/MSP.

3.3.5.1 Εκτέλεση

- Αρχιτεκτονική πολλαπλών πλατφορμών για επιτάχυνση του GPU (graphics possecive unit) hardware. Αυτή η βασική αλλαγή στην επεξεργασία προσφέρει μια δραματική αύξηση στην εκτέλεση της επεξεργασίας εικόνων και γενικότερα για σκοπούς επεξεργασίας, όπως για τις σκιές των υλικών και άλλων GPU εφέ.
- Γρήγορη YUV 4:2:2 (UYVY) μεταφορά βίντεο στην GPU. Χρησιμοποιώντας το μισό εύρος ζώνης, τις μισές χρωματικές πληροφορίες παρουσίασης κάτι το οποίο είναι κοινό σε πολλά από τα σημερινά βίντεο format, το Jitter είναι ικανό να αποσυμπιέσει και να μεταφέρει πιο γρήγορα YUV 4:2:2 δεδομένα στην κάρτα γραφικών σε πραγματικό χρόνο. Μαζί με την GPU επεξεργασία, αυτό μας επιτρέπει μέχρι και HD ανάλυση βίντεο για επεξεργασία σε πραγματικό χρόνο.
- Υποστηρίζεται επίσης παράλληλη επεξεργασία. Πολλά βασιζόμενα στην CPU jitter αντικείμενα κατορθώνουν να χρησιμοποιούν όλα τα πλεονεκτήματα των πολύ-επεξεργαστών και πολύ-πύρηνων συστημάτων που είναι διαθέσιμα σήμερα.

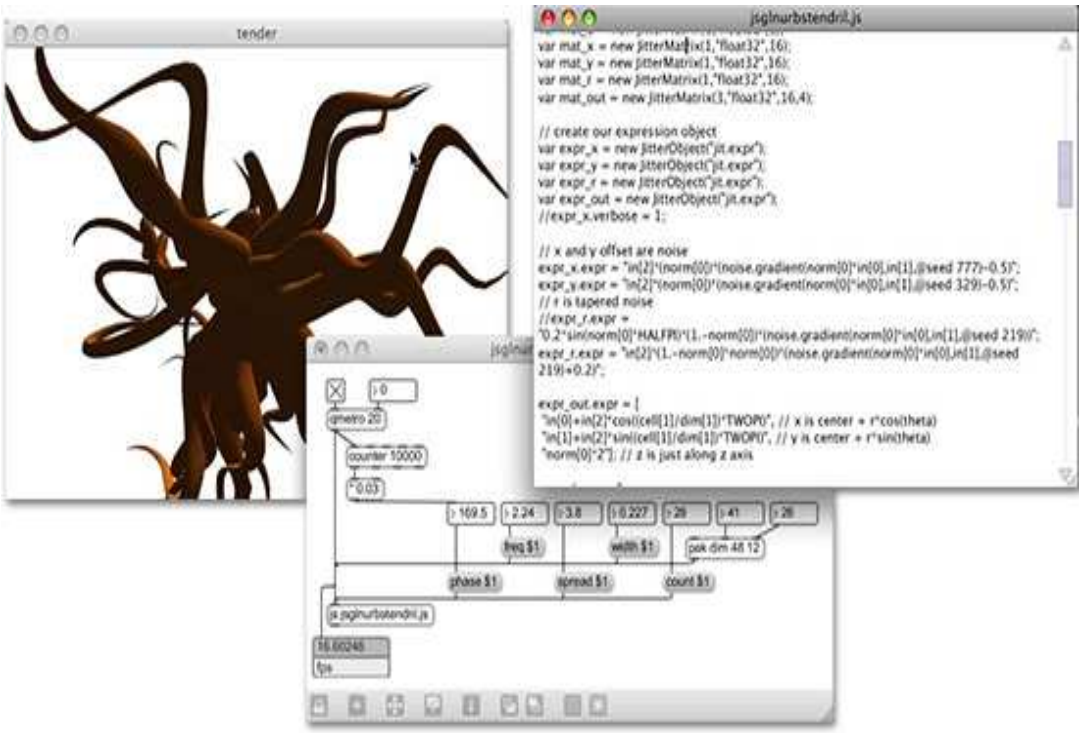


3.3.5.2 Δικτύωση

- Συμπιεσμένα με RTSP δεδομένα. Το Jitter 1.7 χρησιμοποιεί είτε την αρχιτεκτονική πλατφόρμας LiveMedia, ή την QT αρχιτεκτονική σε Macintosh για να κάνει μετάδοση μια ποικιλία από κωδικοποιήσεις βίντεο σε πραγματικό χρόνο.
- Μη συμπιεσμένα Jitter πλέγματα. Για μεταφορά με την ελάχιστη απώλεια των Jitter πλεγμάτων αυθαίρετου τύπου, των επιπέδων, και των διαστάσεων, το Jitter 1.7 επιτρέπει απευθείας διαδικτυακή επικοινωνία με ένα χαρακτηριστικό Jitter format.

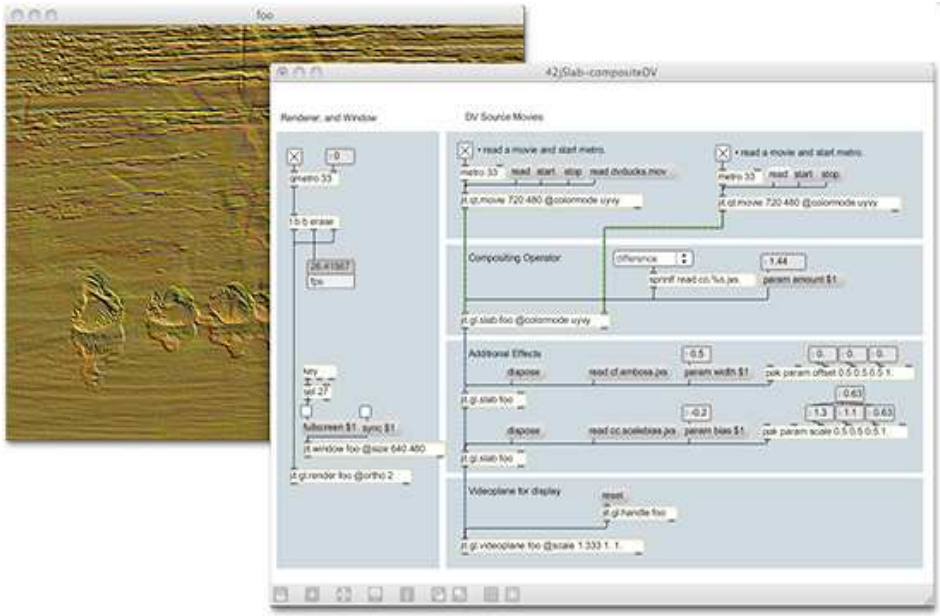
3.3.5.3 Προγραμματισμός

- Υποστηρίζεται Java και Javascript. Συγκεκριμενοποιώντας και ελέγχοντας τα αντικείμενα του Jitter απευθείας από τις βασισμένες σε κείμενο γλώσσες προγραμματισμού οι οποίες προσφέρουν μεγαλύτερο έλεγχο των περίπλοκων έργων.
- Εκφράσεις. Συνοπτικά ορίζουν μαθηματικές εκφράσεις για τον υπολογισμό δεδομένων πλέγματος.



3.3.5.4 Διαλειτουργικότητα

- Υποστήριξή Direct-X βίντεο είσοδο και έξοδο. Οι είσοδοί και έξοδοι βίντεο των Windows προσφέρουν υψηλότερη απόδοση και πιο αξιόπιστες I/O, χωρίς να χρειάζονται QT εμπλεκόμενα μέρη από τρίτους κατασκευαστές.
- Υποστήριξη του FreeFrame plug-in. Ένα ελεύθερο πρότυπο για επεξεργασία βίντεο. Τα FreeFrame plug-in παρέχουν ένα πλούσιο σετ επιλογών για την δημιουργία τυπικών δικτύων εφέ.
- MSP audio ενσωμάτωση. Συμπεριφορά των MSP buffer αντικειμένων σαν δεδομένα πλεγμάτων του Jitter, και μετατροπή διανυσμάτων audio προς και από Jitter πλέγματα για επεξεργασία βασισμένη σε frame και audio οπτικές αναπαραστάσεις.



Κεφάλαιο 4
Αποτελέσματα του προγράμματός μας, χρήσεις του Max/MSP/Jitter και Φίλτρα.

4.1 Αποτελέσματα προγράμματός πτυχιακής

Παράθεση των αποτελεσμάτων που είχαμε από την χρήση του προγράμματός μας. Έχουμε πρώτα από όλα την δυνατότητα καταγραφής και επεξεργασίας real-time βίντεο από δυο πηγές (κάμερες), επίσης έχω την δυνατότητα αναπαραγωγής υπαρχόντων βίντεο από τον σκληρό μου δίσκο. Και στην συνέχεια ανά δυο τα παραπάνω είτε real-time βίντεο με real-time βίντεο, real-time με ήδη υπάρχον, είτε ήδη υπάρχον με ήδη υπάρχον βίντεο να συνδυαστούν μεταξύ τους έχοντας την δυνατότητα να μεταβαίνουμε από το ένα βίντεο στο άλλο σταδιακά. Στην συνέχεια έχουμε περάσει την δυνατότητα ελέγχου κάποιων λειτουργιών του προγράμματος μας μέσω εξωτερικών συσκευών όπως είναι το πληκτρολόγιο και το joystick. Με το joystick έχουμε για παράδειγμα τον έλεγχο μετάβασης από το ένα βίντεο στο άλλο. Έχουμε επίσης την δυνατότητα εισαγωγής κάποιων προκαθορισμένων φίλτρων στο τελικό αποτέλεσμα του βίντεο ακόμη και παραγωγή και χρήση δικών μας με την μέθοδο convolution, το οποίο τελικό αποτέλεσμα μπορεί να καταγραφεί σε .mov αρχείο. Κάποια βασικά φίλτρα μπορούμε να τα εισάγουμε μέσω του πληκτρολογίου με τα κουμπιά των αριθμών από το 1 μέχρι το 6 και έχουμε ακόμη την δυνατότητα ρύθμισης της κεντρικής τιμής των φίλτρων μέσω του joystick . Υπάρχει και ένα δεύτερο κομμάτι του προγράμματος το οποίο εμφανίζει τα frame του αποτελέσματος από το πρώτο κομμάτι με τυχαία σειρά στην παρουσίασή τους σε ένα παράθυρο προβολής και έχει και την δυνατότητα ρύθμισης των παραμέτρων του, δηλαδή του ρυθμού παρουσίασης και τον τυχαίο ρυθμό παραγωγής των frame. Το όλο αποτέλεσμα θα μπορούμε να το παρακολουθήσουμε μέσω ενός παραθύρου προβολής.

4.2 Περιγραφή και χρήσεις του Max/MSP/Jitter.

Το Max/MSP/Jitter είναι ένα διαδραστικό εικονικό προγραμματιστικό περιβάλλον για μουσική, ήχο και βίντεο. Το Max είναι το γραφικό προγραμματιστικό περιβάλλον το οποίο παρέχει τις διεπαφές χρήστη, τα εργαλεία για τον χρόνο τις επικοινωνίες, και την υποστήριξη MIDI. Το MSP προσθέτει την πραγματικού χρόνου σύνθεση audio και DSP, και το jitter επεκτείνει το Max με βίντεο και επεξεργασία δεδομένων matrices.

Κάποια παραδείγματα χρήσεων του Max/MSP/Jitter είναι: Μια διαδραστική εφαρμογή ήχου και βίντεο όπου ο παρατηρητής βλέπει τον εαυτό του σαν έναν άγγελο που χορεύει σε έναν μαγικό καθρέπτη, με την χρήση μια οθόνης και αισθητηρίων κίνησης και απόστασης μπορούμε να κατασκευάσουμε την εφαρμογή αυτή http://parmerud.com/MediaArtist/LostAngel.html. Εδώ βλέπουμε και την καλλιτεχνική πλευρά του Max/MSP/Jitter.

Ένα δεύτερο παράδειγμα είναι η παρακάτω διαδραστική εφαρμογή στην οποία έχουμε ένα μηχάνημα οπίσθιας προβολής εικόνων και μια κάμερα πάνω από αυτό. Το μηχάνημα προβάλει μια εικόνα την οποία την χωρίζει σε κομμάτια, εμείς ακουμπώντας πάνω στην επιφάνεια προβολής μπορούμε να μετακινήσουμε τα κομμάτια αυτά να τα ανακατέψουμε και να τα ξαναβάλουμε στην αρχική τους θέση κάτι σαν πάζλ και όλα αυτά με την βοήθεια της κάμερας η οποία διαβάζει την επιφάνεια προβολής σε συντεταγμένες και καταλαβαίνει τις κινήσεις του χεριού μας όταν ακουμπάμε πάνω σε αυτήν και πράττει ανάλογα http://vimeo.com/11906339.

4.3 Φίλτρα

4.3.1 Εισαγωγή

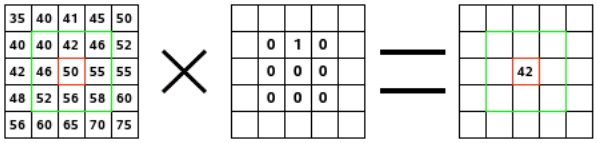
Τα φίλτρα είναι ηλεκτρονικά κυκλώματα τα οποία εκτελούνται από συναρτήσεις επεξεργασίας σήματος, για την απομάκρυνση μη επιθυμητών συχνοτήτων από το σήμα της εικόνας, ή για την ανάκτηση άλλων συχνοτήτων ή ακόμη και για τα δυο μαζί.

Τα περισσότερα φίλτρα στην επεξεργασία εικόνας χρησιμοποιούν convolution matrix. Με το Convolution Matrix, μπορείς να φτιάξεις ένα συνηθισμένο φίλτρο.
Μπορούμε να εξηγήσουμε χωρίς μαθηματικά εργαλεία την κεντρική ιδέα του τι είναι το convolution matrix. Convolution είναι η μεταχείριση ενός matrix από ένα άλλο το οποίο ονομάζεται “kernel”.

Το convolution matrix φίλτρο πρώτα χρησιμοποιεί ένα matrix το οποίο είναι η εικόνα που θα δεχθεί την τροποποίηση. Η εικόνα είναι μια δυσδιάστατη συλλογή από pixel με ορθογώνιες συντεταγμένες. Οι τιμές του χρησιμοποιούμενου πίνακα kernel εξαρτούνται από το εφέ που θέλουμε να χρησιμοποιήσουμε.

Παρόλο που μπορούμε να χρησιμοποιήσουμε διαφόρων μεγεθών πίνακες τύπου kernel, εμείς θα χρησιμοποιήσουμε τους πιο κοινούς που είναι οι 3*3 πίνακες, και είναι αρκετοί για όλα τα εφέ που θα χρησιμοποιήσουμε. Εάν οι τιμές των συνόρων του πίνακα kernel είναι 0, το σύστημα θα τον υπολογίσει πάλι σαν έναν 3*3 πίνακα.
Το φίλτρο εξετάζει διαδοχικά κάθε pixel της εικόνας. Για το κάθε ένα από αυτά, τα οποία θα ονομάζουμε κεντρικά pixel, θα πολλαπλασιάζουν την τιμή τους και τις τιμές των 8 pixel που το πλαισιώνουν με τις αντίστοιχες τιμές του πίνακα kernel. Στην συνέχεια προσθέτει το αποτέλεσμα, όλων των πολλαπλασιασμένων pixel και τοποθετεί την καινούργια τιμή στην θέση της κεντρικής τιμής κάθε φορά.

Ένα απλό παράδειγμα:



Στα αριστερά είναι ο πίνακας της εικόνας μας στην οποία θέλουμε να εφαρμόσουμε ένα φίλτρο με την χρήση πίνακα kernel, κάθε pixel είναι σημειωμένο με την τιμή του. Το κεντρικό pixel έχει ένα κόκκινο σύνορο. Ο χώρος δράσης του kernel έχει ένα πράσινο σύνορο. Στην μέση του παραδείγματος είναι ο πίνακας kernel και, στην δεξιά πλευρά βλέπουμε το αποτέλεσμα του convolution.

Εδώ θα εξηγήσουμε τι συνέβη: το φίλτρο διάβασε διαδοχικά από αριστερά στα δεξιά και από πάνω προς τα κάτω, όλα τα pixel της περιοχής δράσης του kernel. Στην συνέχεια πολλαπλασίασε την τιμή κάθε pixel με τις αντίστοιχες τιμές του kernel και πρόσθεσε τα αποτελέσματα. Το κεντρικό pixel πήρε την τιμή 42:(40*0)+(42*1)+(46*0) + (46*0)+(50*0)+(55*0) + (52*0)+(56*0)+(58*0) = 42. Σαν γραφικό αποτέλεσμα, το κεντρικό pixel μετακινήθηκε ένα pixel προς τα κάτω.

4.3.2Χρήση φίλτρων στο πρόγραμμα μου.

Έχουμε φτιάξει στο πρόγραμμα μας έναν πίνακα 3*3 αντίστοιχο του kernel, για να μπορούμε να περνάμε τις τιμές των φίλτρων που θέλουμε. Στον πίνακα μπορούμε να περνάμε τιμές είτε πληκτρολογώντας απευθείας από το πληκτρολόγιο μέσα σε κάθε κελί του πίνακα ξεχωριστά, είτε χρησιμοποιώντας τα έτοιμα μηνύματα με τα έξη βασικά φίλτρα των οποίων οι τιμές περνούν απευθείας με την επιλογή τους στον πίνακα, επίσης έχουμε φτιάξει και δυο αντικείμενα εισαγωγής αριθμών, ένα το οποίο αλλάζει τις τιμές όλων των κελιών του φίλτρου μας τοποθετώντας την ίδια τιμή σε όλα τα κελία και ένα το οποίο αλλάζει μόνο την κεντρική και βασικότερη ταυτόχρονα τιμή ενός φίλτρου με την βοήθεια του joystick. Κάθε φορά που γίνεται μια αλλαγή στον πίνακα στέλνεται απευθείας στο αντικείμενο που κάνει την εφαρμογή του φίλτρου στο βίντεο μας.

Κεφάλαιο 5

Επεξήγηση και ανάλυση του προγράμματος επεξεργασίας βίντεο σε πραγματικό χρόνο με το Max/MSP/Jitter.

Το πρόγραμμα μας αποτελείται από έξη διαφορετικά μέρη τα οποία συνδέονται μεταξύ τους και μας παράγουν το τελικό αποτέλεσμα, τα οποία θα τα αναλύσουμε κομμάτι κομμάτι.

5.1 Ενεργοποίηση προγράμματος και καταγραφή βίντεο.

5.1.1 Παράθεση και επεξήγηση των αντικειμένων ένα προς ένα.

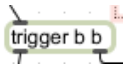
Ένα ένα θα εξηγήσουμε τα αντικείμενα που χρησιμοποιούμε στο 1ο μέρος του προγράμματος μας, με παραθέσεις και επεξηγήσεις και στο τέλος θα δείξουμε όλα τα αντικείμενα ενωμένα και τι παράγουν όλα μαζί.

Ξεκινάμε με το αντικείμενο Toggle:

Το αντικείμενο toggle είναι ένας διακόπτης ο οποίος όταν είναι απενεργοποιημένος στέλνει σαν έξοδο ένα 0 και όταν είναι ενεργοποιημένος ένα 1, έχει μια είσοδο και μια έξοδο, όταν δεχθεί στην είσοδο ένα μη μηδενικό αριθμό θα ανοίξει όταν δεχθεί ένα μηδενικό θα κλείσει, επίσης ένα bang αλλάζει την κατάσταση του toggle, όπου bang μπορεί να είναι οποιοδήποτε σήμα το οποίο περνάει στην είσοδο του προερχόμενο από ένα αντικείμενο ή ένα μήνυμα και προκαλεί το αντικείμενο να παράγει την εργασία για την οποία είναι φτιαγμένο (πυροδοτεί το αντικείμενο) επίσης το αντικείμενο toggle μπορεί να ενεργοποιηθεί και απλά κάνοντας κλικ με το ποντίκι πάνω του. Το αντικείμενο αυτό είναι ο βασικός διακόπτης που ενεργοποιεί όλο το πρόγραμμα μας.

Metro:

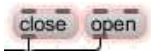
Λειτουργεί σαν μετρονόμος ο οποίος βγάζει έξοδο ένα μήνυμα bang κατά s τακτά και ορισμένα χρονικά διαστήματα. Όπου s είναι ο χρόνος σε millisecond ο οποίος μπαίνει δεξιά του metro, στο παράδειγμά μας το 30 σημαίνει ότι κάθε 0,03sec στέλνεται ένα μήνυμα bang από την έξοδό του. Μπορεί τα διαστήματα αυτά να είναι προσδιορισμένα εξ αρχής ή όχι, έχοντας την δυνατότητα να ρυθμίζονται περνώντας από την πάνω δεξιά είσοδο του αντικειμένου χρονικές τιμές μέσω ενός αντικείμενου που λέγεται “number box”, ελέγχοντας έτσι την συχνότητα εξόδου του metro, αλλάζοντας τον χρόνο.

Trigger: 

Το αντικείμενο trigger που συμβολίζεται και με σκέτο “t” μέσα στο κουτί του αντικείμενου βγάζει σαν έξοδο κάθε είσοδο που λαμβάνει από πάνω αριστερά, με σειρά από δεξιά προς τα αριστερά, η οποία έξοδος έχει την μορφή με την οποία έχει οριστεί μέσα στο αντικείμενο. Στην εικόνα μας το αντικείμενο βγάζει δυο εξόδους με την μορφή bang και οι δυο, δηλαδή ότι μορφή και αν έχει η είσοδος του trigger η έξοδος θα είναι bang. Το πόσες εξόδους θα έχει το αντικείμενο αυτό καθορίζεται από το πόσα ορίσματα είναι περασμένα δεξιά του ονόματος του αντικειμένου trigger.

Gswitch:

Το αντικείμενο gswitch δρομολογεί τα στοιχεία που λαμβάνει από την πάνω δεξιά είσοδο στην κάτω δεξιά ή αριστερή έξοδό του ανάλογα με το που βρίσκεται κάθε φορά ο σύνδεσμος που ενώνει την είσοδο με την έξοδο. Η πάνω αριστερή είσοδος του αντικείμενου είναι για τον έλεγχο της δρομολόγησης του αντικειμένου, δηλαδή μπορούμε συνδέοντας εκεί ένα μήνυμα bang πατώντας το κάθε φορά να μεταβαίνουμε από την μια έξοδο στην άλλη.

Open, Close: 

Τα open και close είναι μηνύματα τα οποία στέλνονται σε αντικείμενα για την εκτέλεση των μηνυμάτων αυτών. Τα συγκεκριμένα ενεργοποιούν το άνοιγμα, και το κλείσιμο μιας εργασίας που κάνει το αντικείμενο που δέχεται τα μηνύματα αυτά. Τα μηνύματα αυτά μπορούν να σταλθούν στο αντικείμενο είτε πατώντας τα είτε ενεργοποιώντας τα από την πάνω δεξιά είσοδο τους, π.χ. με ένα μήνυμα bang όπως είδαμε και στο προηγούμενο αντικείμενο gswitch.

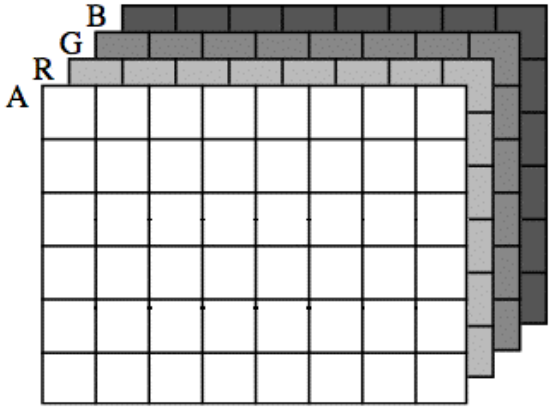
Jit.qt.grap:

Το αντικείμενο αυτό επιτρέπει σε κάποιον να κάνει εγγραφή και ψηφιοποίηση βίντεο από οποιοδήποτε Quicktime συμβατό ψηφιοποιητή, και να αποσυμπιέσει το σήμα σε ένα Jitter matrix.

Όπου Jitter matrix είναι ένα matrix όπου κάθε pixel του, περιέχει κάποια πληροφορία και έχει και μοναδικό όνομα εύρεσης μέσα στο matrix. Για παράδειγμα μια οθόνη προβολής βίντεο είναι φτιαγμένη από μοναδικά pixels, (στοιχεία εικόνας), κάθε ένα από τα οποία απεικονίζει ένα μοναδικό χρώμα. Υπάρχουν πολλοί τρόποι απεικόνισης του χρώματος σε κάθε pixel, ένας τρόπος στους υπολογιστές είναι να αναλύσεις το χρώμα σε κόκκινο, πράσινο, μπλε (RGB) και μια επιπρόσθετη πληροφορία της διαφάνειας (γνωστό ως alpha channel). Τα περισσότερα προγράμματα υπολογιστών αποθηκεύουν το χρώμα ενός συγκεκριμένου pixel σαν τέσσερις διαφορετικούς αριθμούς, οι οποίοι αναπαριστούν το alpha, το κόκκινο, το πράσινο και το μπλε στοιχείο. Η τετρακάναλη αυτή παρουσίαση του θέματος συχνά ονομάζεται ARGB ή RGBA.

Το Jitter δεν αποτελεί εξαίρεση από αυτή την αποδοχή. Για να αποδώσει κάθε κελί του matrix ένα χρωματιστό pixel, πρέπει στην πραγματικότητα να συγκρατεί 4 αριθμητικές τιμές(alpha, κόκκινο, πράσινο, και μπλε) και όχι μόνο μια. Έτσι ένα matrix που αποθηκεύει δεδομένα για ένα frame βίντεο στην πραγματικότητα περιέχει 4 τιμές σε κάθε κελί.

π.χ.



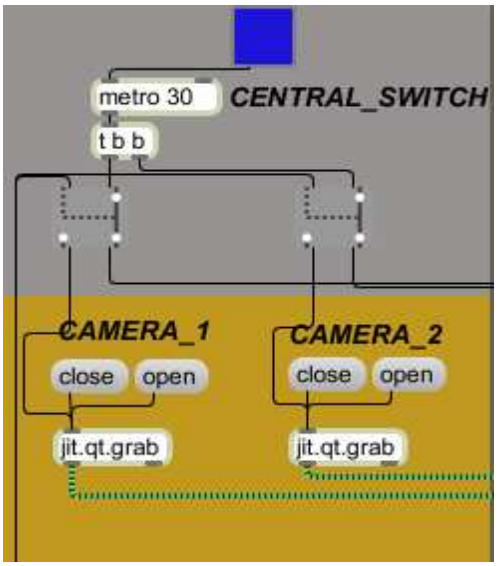
Εδώ βλέπουμε ότι κάθε pixel απεικονίζεται από τους τέσσερις αριθμούς όπως προαναφέραμε. Έτσι στην πάνω εικόνα βλέπουμε ότι το matrix χωρίζεται σε τέσσερα επίπεδα “planes” ένα για κάθε τιμή και όλα μαζί συνδυάζονται μετά για να βγεί το τελικό αποτέλεσμα στο matrix όπως βλέπουμε στην παρακάτω εικόνα.

A: 255 R: 218 G: 111 B: 218	A: 255 R: 218 G: 111 B: 218	A: 254 R: 218 G: 112 B: 217	etc.
A: 255 R: 218 G: 111 B: 217	A: 255 R: 218 G: 111 B: 217	A: 254 R: 218 G: 112 B: 217	etc.
etc.	etc.	etc.	etc.

5.1.2 Παράθεση και επεξήγηση του 1ου μέρους του προγράμματος του οποίου τα αντικείμενα αναλύσαμε παραπάνω.

Τώρα όλα αυτά τα αντικείμενα που εξετάσαμε και παρουσιάσαμε έχουν την παρακάτω διάταξη και διασύνδεση στο πρόγραμμα μας. Πατώντας τον κεντρικό διακόπτη toggle ξεκινάει η λειτουργία του προγράμματος μας στέλνοντας ένα μήνυμα 1 στο αντικείμενο metro το οποίο με την σειρά του στέλνει ένα μήνυμα bang κάθε 30millisecond στο αντικείμενο trigger, το trigger ενεργοποιείται και στην συνεχεία στέλνει δυο μηνύματα bang, όπως βλέπουμε από τα δύο b δεξιά του trigger, στα αντικείμενα gswitch όπου από εκεί το καθένα μήνυμα bang μπορεί να στέλνεται είτε στο αντικείμενο που κάνει την καταγραφή και ψηφιοποίηση με την κάμερα για να

ενεργοποιηθεί είτε σε ένα άλλο αντικείμενο που κάνει την αναπαραγωγή βίντεο από τον δίσκο μας και το οποίο θα δούμε στην συνέχεια, το σε ποιο αντικείμενο θα σταλθεί εξαρτάτε από το με ποια έξοδο είναι συνδεδεμένη η είσοδος του gswitch. Τα αντικείμενα gswitch λαμβάνουν μια είσοδο πάνω αριστερά η οποία είναι για τον έλεγχο της εξόδου τους μέσω μιας εξωτερικής συσκευής κάτι το οποίο θα δούμε και θα αναλύσουμε στην συνέχεια. Με την σειρά τους τα μηνύματα open και close ανοίγουν και κλείνουν τις κάμερες αντίστοιχα, ξεκινάει δηλαδή η καταγραφή από τις κάμερες. Το jit.qt.grab καταγράφει και ψηφιοποιεί το βίντεο σε default μέγεθος 320*240 και στην έξοδό του βγαίνει με την μορφή matrices.



5.2 Αναπαραγωγή και συνδυασμός βίντεο.

5.2.1 Παράθεση και επεξήγηση των αντικειμένων ένα προς ένα.

jit.qt.movie:

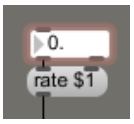
Το jit.qt.movie αντικείμενο χρησιμεύει για την αναπαραγωγή βίντεο σε μια οθόνη προβολής, και περιέχει μια σουίτα η οποία εμπεριέχει όλες τις υπηρεσίες που χρειάζονται για τις Quicktime ταινίες: όπως η αναπαραγωγή, η επεξεργασία, η εισαγωγή, η εξαγωγή, η δημιουργία εφέ και η απευθείας έξοδος για βίντεο μετάδοση. Το 320 240 σημαίνει ότι η quicktime ταινία που θα αναπαράγει θα είναι 320*240 pixel. Το αντικείμενο αυτό έχει πολλές παραμέτρους τις οποίες μπορεί να ελέγξει κάποιος όπως για παράδειγμα είναι το “vol” το οποίο έχει να κάνει με την ένταση της φωνής του βίντεο και η default τιμή του είναι το 1 και οι usedstrect και η dstrect οι οποίες προσδιορίζουν την περιοχή μέσα στο παράθυρο προβολής που θα εμφανιστεί η εικόνα μας. Η κάθε παράμετρος μπαίνει μετά το όνομα του αντικειμένου με το σύμβολο @ στην αρχή της όπως και σε όλα τα αντικείμενα του jitter, πχ

.

Έχει ακόμη πολλές ιδιότητες τις οποίες αν τις χρειαστούμε θα τις αναλύσουμε αργότερα. Επίσης η έξοδος του αντικειμένου jit.qt.movie είναι με την μορφή των matrices.

Start, stop, read:

Τα start stop read είναι μηνύματα τα οποία συνδέονται με αντικείμενα τα οποία μπορούν να δεχθούν τέτοια μηνύματα και ενεργοποιούν κάποιες λειτουργίες τους. Όπως ένα αντικείμενο σαν το jit.qt.movie το οποίο με το start ξεκινάει την αναπαραγωγή ενός βίντεο με το stop την σταματάει και με την read ανοίγει ένα παράθυρο διαλόγου και από εκεί επιλέγουμε το βίντεο προς αναπαραγωγή.

Rate: 

Το rate είναι μια παράμετρος του αντικειμένου jit.qt.movie η οποία καθορίζει τον ρυθμό με τον οποίο αναπαράγεται το βίντεο. Η default τιμή του είναι 1. και η τιμές που δέχεται είναι δεκαδικές. Αυτός είναι ο δεύτερος τρόπος να περάσουμε παραμέτρους στο αντικείμενο, θα μπορούσαμε να περάσουμε αυτήν την παράμετρο με τον τρόπο που είδαμε πριν δηλαδή με το @ απευθείας μέσα στο αντικείμενο μας, αυτό που βλέπουμε συνδέεται στην πάνω δεξιά είσοδο του αντικειμένου. Το αριθμητικό κουτί πάνω από το rate ονομάζεται number box το οποίο δέχεται τιμές τύπου float και είναι αυτό που μας επιτρέπει την οποιαδήποτε αλλαγή στον ρυθμό αναπαραγωγής του βίντεο τοποθετώντας αριθμητικές τιμές μέσα του πληκτρολογώντας ή απλά σέρνοντας μέσα του κρατώντας πατημένο το mouse.

Slider:

Το slider είναι μια διεπαφή χρήστη που έχει την μορφή κυλιόμενου ποτενσιόμετρου. Έχει σαν έξοδο αριθμούς περιορισμένους από ένα προσδιορισμένο σύνολο αριθμών τους οποίους καθορίζουμε εμείς μέσω του inspector το οποίο είναι ένα εργαλείο ελέγχου όλων των παραμέτρων για όλα τα αντικείμενα, ο αριθμός από τον οποίο ξεκινάει και αυτός στον οποίο τελειώνει επίσης καθορίζεται από εμάς μέσω του inspector. Το slider μπορεί να έχει όποιο σχήμα και μέγεθος θέλουμε χωρίς πάραυτα να αλλάζει το εύρος τιμών του slider. Μπορεί να έχει σαν έξοδο από δύο έως άπειρες τιμές.

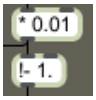
Pack:

Το αντικείμενο pack δέχεται διαφορετικά πράγματα από τις εισόδους του, τα συνδυάζει μεταξύ τους και τα βγάζει στην έξοδο του με την μορφή λίστας. Το μήκος της λίστας καθορίζεται από τον αριθμό των εισόδων του αντικειμένου. Η default

είσοδοι του pack είναι δύο. Μπορούμε επίσης να ρυθμίσουμε τις παραμέτρους όπως είναι π.χ. οι είσοδοι του μέσω του inspector.

Line:

Το line αντικείμενο χρησιμοποιείται για να δημιουργεί ράμπες σύνδεσης από μια τιμή σε μια άλλη μέσα σε μια συγκεκριμένη ποσότητα χρόνου. Δέχεται σαν ορίσματα δυο τιμές, η πρώτη δηλώνει την αρχική τιμή που αποθηκεύεται στο line και τον τύπο εξόδου του αντικειμένου. Αν δεν έχουμε τιμή, η αρχική είναι 0 και ο τύπος δεδομένων εξόδου είναι ακέραιος. Η δεύτερη τιμή ρυθμίζει το χρονικό διάστημα με το οποίο οι αριθμοί στέλνονται από την έξοδο του αντικειμένου. Εάν δεν υπάρχει όρισμα ο default χρόνος είναι 20millisecond. Στο παράδειγμα μας σαν αρχική τιμή έχουμε το 0 δηλαδή αρχική τιμή 0 και τα δεδομένα μας είναι τύπου ακεραίου και στέλνεται ένας αριθμός κάθε 50millisecond. Δηλαδή μεταβαίνει από μια τιμή σε μια άλλη σταδιακά μέσα σε ένα ορισμένο διάστημα χρόνου 0,05sec. Στο πρόγραμμα μας στην αριστερή είσοδο του line στέλνουμε μια λίστα από την έξοδο του αντικειμένου pack η οποία περιέχει δύο αριθμούς ο πρώτος είναι ο αριθμός που θέλουμε να μεταβούμε και ο δεύτερος καθορίζει τον χρόνο μετάβασης από την μια τιμή στην άλλη.

Τελεστές * !: 

Εδώ έχουμε δύο αντικείμενα, με το πρώτο πολλαπλασιάζουμε την είσοδο που έχουμε από την έξοδο του slider με το 0.01 τον οποίο αριθμό βγάζει στην έξοδο του, και το δεύτερο αντικείμενο δέχεται την έξοδο αυτή και αφαιρεί με -1 και βγάζει σαν έξοδο τον ανάστροφο του (το βλέπουμε από το !).

Jit.scalebias:

Το jit.scalebias δέχεται σαν είσοδο τις τιμές από matrix τεσσάρων επιπέδων του τύπου char (ARGB) από το αντικείμενο jit.qt.movie ή από το αντικείμενο jit.qt.grab και τις συνδυάζει σε μια μοναδική τιμή στην έξοδό του. Η ιδιότητα scale που έχουμε

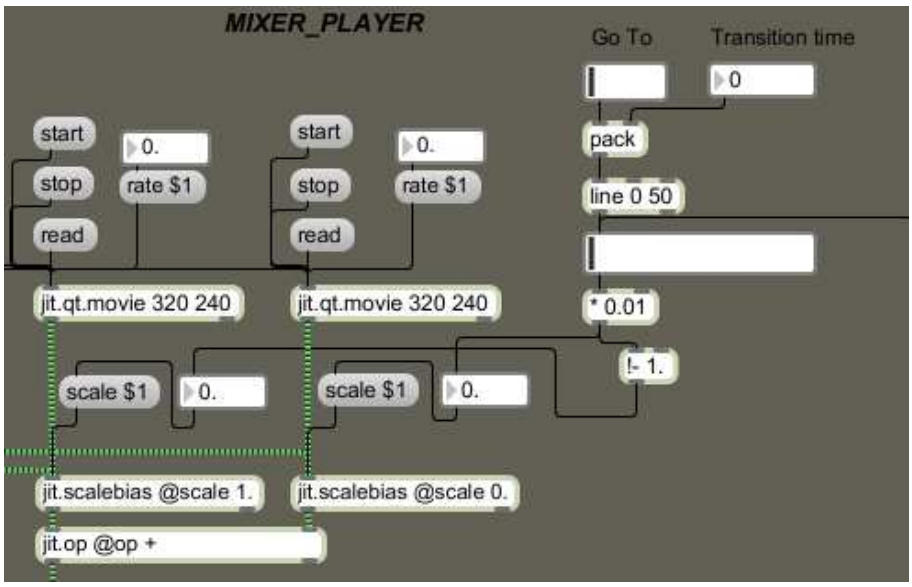
στο παράδειγμά μας καθορίζει το ποσοστό διαφάνειας οπού θα εμφανίζεται το άθροισμα των τιμών του ARGB στην οθόνη μας. Όταν η τιμή του είναι 1. τότε θα εμφανίζεται κανονικά η εικόνα μας και όσο πηγαίνουμε προς το 0 τόσο θα χάνεται η εικόνα μας. Το αντικείμενο jit.scalebias δουλεύει με δεδομένα της μορφής matrix και στην είσοδο του και στην έξοδο του.

Jit.op:

Το jit.op δέχεται σαν είσοδο δυο matrices και μπορεί να εφαρμόσει σε αυτά δυαδική πράξη δηλαδή είτε πρόσθεση είτε αφαίρεση, ή μπορούμε να εφαρμόσουμε, αν έχουμε είσοδο μόνο από δεξιά του αντικειμένου, μοναδιαία πράξη π.χ. Άρνηση (!). Η ιδιότητα op που έχει το αντικείμενο χρησιμεύει για να μπορέσουμε να χρησιμοποιήσουμε οποιονδήποτε τελεστή για να τον εφαρμόσει στα matrices των εισόδων του.

5.2.2 Παράθεση και επεξήγηση του 2ου μέρους του προγράμματος του οποίου τα αντικείμενα αναλύσαμε παραπάνω.

Και τώρα θα δούμε όλα τα αντικείμενα του 2ου κομματιού του προγράμματος μας συνδυασμένα και θα εξηγήσουμε τι κάνει όλο μαζί.



Στο 2ο κομμάτι του προγράμματός μας έχουμε την αναπαραγωγή βίντεο από το αντικείμενο jit.qt.movie σε μέγεθος 320*240 pixel, το αντικείμενο ελέγχεται από τα μηνύματα start για την έναρξη αναπαραγωγής του βίντεο, το stop για το σταμάτημα της αναπαραγωγής του και το read για το άνοιγμα ενός παραθύρου διαλόγου για την επιλογή ενός βίντεο μέσα από τα μέσα αποθήκευσης. Η ιδιότητα rate, η οποία κάλλιστα θα μπορούσε να είχε τοποθετηθεί και μέσα στο αντικείμενο jit.qt.movie, ελέγχει τον ρυθμό αναπαραγωγής του βίντεο μέσο μιας δεκαδικής τιμής που του περνάμε από το αριθμητικό κουτί που συνδέεται με το rate, όταν αυξάνουμε ή μειώνουμε τον ρυθμό αναπαραγωγής τότε αντίστοιχα αυξάνουμε ή μειώνουμε τον χρόνο διάρκειάς του βίντεο. Στην έξοδο του το jit.qt.movie βγάζει ένα matrix τεσσάρων επιπέδων τύπου ARGB, τρία επίπεδα για τα χρώματα κόκκινο, κίτρινο, μπλε και ένα για το alpha channel.

Δεξιά στην εικόνα βλέπουμε το κομμάτι που χειρίζεται τον συνδυασμό των δύο βίντεο. Το πρώτο αντικείμενο είναι μια κυλιόμενη μπάρα (slider) η οποία έχει μόνο δυο τιμές 0 και 100 χωρίς ενδιάμεσες, και πατώντας την πηγαίνει είτε στην αριστερή πλευρά δηλαδή στο 0 εμφανίζοντας στην οθόνη το ένα βίντεο είτε στην δεξιά δηλαδή στο 100 όπου εμφανίζεται το δεύτερο. Δίπλα της έχουμε ένα κουτί οπού μπορούμε να αλλάξουμε τον χρόνο μετάβασης από το ένα βίντεο στο άλλο. Ο χρόνος μετάβασης σε συνδυασμό με μια από τις δύο τιμές (0 ή 100) μπαίνουν στο αντικείμενο pack και βγαίνουν με την μορφή λίστας δηλαδή συνδυασμένα από την έξοδο του. Στην συνέχεια οι τιμές μπαίνουν στο αντικείμενο line το οποίο στέλνει συνεχόμενες ακέραιες τιμές από το 0 έως το 100 ή το αντίθετο για την διάρκεια χρόνου που δώσαμε στο κουτί του χρόνου μετάβασης. Οι τιμές αυτές περνάνε στην δεύτερη κυλιόμενη μπάρα, η οποία ορίζεται από 100 τιμές, και με την σειρά της μεταβαίνει στις τιμές που δέχεται σαν είσοδο, πάντα στην διάρκεια χρόνου που έχουμε δώσει στο κουτί του “Transition time”. Για την μετάβαση με την δεύτερη μπάρα έχουμε προσθέσει την δυνατότητα ελέγχου της μέσω του joystick. Κάθε τιμή που βγαίνει στην συνέχεια από τον slider πολλαπλασιάζεται με το 0.01 και στην συνέχεια κάθε τιμή είτε χρησιμοποιείται όπως είναι είτε αφαιρείται με το -1 και στην συνέχεια παίρνουμε τον ανάστροφο της(!), το άθροισμα της εισόδου με το άθροισμα της εξόδου στο αντικείμενο που κάνει την πράξη αυτή είναι πάντα 1. Οι τιμές αυτές στην συνέχεια περνάνε στις ιδιότητες scale δυο αντικειμένων jit.scalebias, στο ένα περνάει η κανονική και στο άλλο η συμπληρωματική του, όπου το scale στο jit.scalebias ρυθμίζει το ποσοστό εμφάνισης της διαφάνειας του matrix, δηλαδή εφόσον το ένα

αντικείμενο jit.scalebias παίρνει την μια τιμή η οποία κυμαίνεται από το 0.0 έως το 1.0 και το άλλο αντικείμενο παίρνει την άλλη τιμή 'όπου πάντα το άθροισμα τους θα είναι 1 τότε σε όλη την μετάβαση θα έχουμε ένα αθροιστικό matrix με άθροισμα ένα. Δηλαδή όσο θα μεταβαίνουμε από την μια τιμή στην άλλη τόσο θα σβήνει το ένα βίντεο και θα εμφανίζεται το άλλο μέχρι που θα εμφανίζεται μόνο το τελευταίο. Στην συνέχεια οι τιμές των matrix που βγαίνουν από το jit.scalebias περνάνε στο αντικείμενο jit.op, το οποίο έχει και σαν ιδιότητα περασμένη την πρόσθεση, και αθροίζονται ώστε να εμφανιστεί το τελικό αποτέλεσμα από τον συνδυασμό των ιδιοτήτων scale στο αντικείμενο προβολής βίντεο.

5.3 Χρησιμοποίηση διασυνδετικών συσκευών αλληλεπίδρασης χρήστη με μηχανή(Human Interface Devices).

Μια “συσκευή ανθρώπινη διεπαφής” Human Interface Device ή HID είναι ένας τύπος συσκευής υπολογιστή η οποία αλληλεπιδρά απευθείας, και συχνά παίρνει είσοδο, από ανθρώπους και επίσης μπορεί να επιστρέψει και είσοδο σε ανθρώπους. Ο όρος “HID” πιο συχνά αναφέρεται στις USB-HID συσκευές.

5.3.1 Παράθεση και επεξήγηση των αντικειμένων που χρησιμοποιούμε για την χρήση HID συσκευών.

Hi:

Το hi είναι το αντικείμενο διεπαφής του Max με τις HI συσκευές. Το αντικείμενο αυτό παρέχει όλες τις πληροφορίες που χρειαζόμαστε για να αλληλεπιδρούμε με joysticks, πλακέτες ζωγραφικής, πληκτρολόγια, mouse και άλλες συσκευές. Σαν είσοδο δέχεται μια περιφερειακή και όχι μόνο συσκευή αλληλεπίδρασης και σαν έξοδο βγάζει όλες τις πληροφορίες για την συσκευή όταν δεχθεί ένα bang ή ένα χρονικό μήνυμα poll. Το αντικείμενο hi έχει δύο εξόδους από την πρώτη βγαίνουν δεδομένα σαν λίστα ακεραίων δύο στοιχείων της συσκευής που έχει επιλεχθεί, ο πρώτος ακέραιος παρουσιάζει τα στοιχεία της συσκευής στην οποία

τα δεδομένα εφαρμόζονται, και ο δεύτερος παρουσιάζει τις τιμές των δεδομένων, και από την δεύτερη έξοδο βγαίνουν απαριθμημένες οι συσκευές, με την μορφή μενού, οι οποίες είναι συνδεδεμένες στον υπολογιστή και μπορούν να χρησιμοποιηθούν σαν HID συσκευές.

Info:

Το μήνυμα info περνάει στην είσοδο του αντικειμένου hi και αναγκάζει το αντικείμενο hi να βγάλει σαν έξοδο τις διαθέσιμες συσκευές, οι οποίες είναι συνδεδεμένες στον υπολογιστή μας και μπορούμε να χρησιμοποιήσουμε, και τις πληροφορίες αυτών στο παράθυρο της κονσόλας του Max.

Menu:

Το μήνυμα menu περνάει και αυτό στην είσοδο του αντικειμένου hi και το προκαλεί να βγάλει από την δεξιά του έξοδο σε ένα umenu αντικείμενο μια λίστα με αριθμημένες τις διαθέσιμες συσκευές αλληλεπίδρασης.

Poll:

Το μήνυμα poll περνάει και αυτό στην είσοδο του αντικειμένου hi ακολουθούμενο από ένα αριθμό που συμβολίζει χρόνο σε millisecond, και είναι υπεύθυνο για τον χρόνο μεταξύ των γεγονότων από τις εξόδους. Το μήνυμα poll 0 απενεργοποίει το αντικείμενο hi.

Route:

Στο αντικείμενο route περνάμε κάποια ορίσματα, τα οποία μπορεί να είναι είτε ακέραιοι είτε δεκαδικοί ή σύμβολα ή ακόμη και συνδυασμός των προηγούμενων, και έχει πάντα μια παραπάνω έξοδο από όσα είναι τα ορίσματα του. Όταν του περνάμε μια είσοδο τότε το αντικείμενο ελέγχει αν η είσοδος ταιριάζει με κάποιο από τα ορίσματα του, αν ναι τότε βγαίνει σαν έξοδο με την μορφή είτε ενός bang είτε του ίδιου του μηνύματος που μπήκε από την είσοδο, αν όχι τότε το μήνυμα βγαίνει από

την τελευταία έξοδο του αντικείμενου αυτούσιο, η οποία έχει αυτή την χρησιμότητα. Επίσης πρέπει να αναφέρουμε ότι όταν ένα μήνυμα που μπαίνει από την είσοδο ταιριάζει π.χ. Με το τέταρτο όρισμα του αντικείμενου τότε βγαίνει και από την τέταρτη έξοδο του.

Metro:

Εδώ έχουμε οχτώ αντικείμενα metro τα οποία κάθε 50millisecond βγάζουν από ένα μήνυμα bang και με την σειρά τους αυτά ενεργοποιούν δυο μηνύματα οπού το ένα βγάζει σαν έξοδο τον αριθμό 1 και το άλλο τον -1 ή τους δεκαδικούς αριθμούς 0.01 και -0.01 με συχνότητα 50 millisecond και αυτά.

Trigger:

Εδώ έχουμε πάλι δύο αντικείμενα trigger που όπως είπαμε μπορούν να αναφερθούν και με το γράμμα 't' μόνο, εδώ και τα δυο έχουν από δύο εξόδους. Όπως είχαμε πει το αντικείμενο αυτό χρησιμεύει για την δρομολόγηση των εισόδων του. Δηλαδή το πρώτο από την δεξιά έξοδο βγάζει μόνο ακεραίους και την αριστερή βγάζει σαν έξοδο ένα μήνυμα bang και το δεύτερο οπού από την δεξιά έξοδο βγάζει δεκαδικούς και την αριστερή ένα μήνυμα bang και αυτό. Δηλαδή αν έχουμε μια είσοδο ενός ακεραίου στο πρώτο trigger τότε από την δεξιά έξοδο θα έχουμε τον ακέραιο και στην αριστερή ένα bang.

Accum:

Το accum αντικείμενο αποθηκεύει μια τιμή (ακέραια ή δεκαδική) και πάνω σε αυτή κάνει προσθέσεις ή/και πολλαπλασιασμούς, κάθε φορά που δέχεται μια τιμή σαν είσοδο κάνει την πράξη ανάλογος από ποιά είσοδο δέχθηκε την τιμή. Αν η τιμή εισόδου είναι ακέραια, ο πολλαπλασιασμός γίνεται σαν να είναι δεκαδική και μετά μετατρέπεται σε ακέραια. Το μήνυμα set στην αριστερή είσοδο του accum ακολουθούμενο από ένα αριθμό, τον οποίο στο πρόγραμμά μας παίρνει από την έξοδο του clip, αλλάζει την αποθηκευμένη τιμή με αυτόν τον αριθμό χωρίς να βγάζει κάποια έξοδο το αντικείμενο accum. Οι αριθμοί που μπαίνουν στην μεσαία είσοδο

του accum προσθέτονται στον αρχικό αριθμό, ενώ οι τιμές που μπαίνουν στην δεξιά είσοδο πολλαπλασιάζονται με τον αποθηκευμένο αριθμό. Στην πρώτη είσοδο μπαίνει ο αριθμός ο οποίος αποθηκεύεται και γίνονται οι πράξεις πάνω σε αυτόν και μπαίνει επίσης σαν είσοδος ένα μήνυμα bang το οποίο ενεργοποιεί την διαδικασία της πράξης και της εξόδου(πρόσθεσης ή πολλαπλασιασμού). Έχουμε δύο αντικείμενα accum στο πρόγραμμα μας ένα το οποίο διαχειρίζεται ακεραίους και ένα που διαχειρίζεται δεκαδικούς.

Clip:

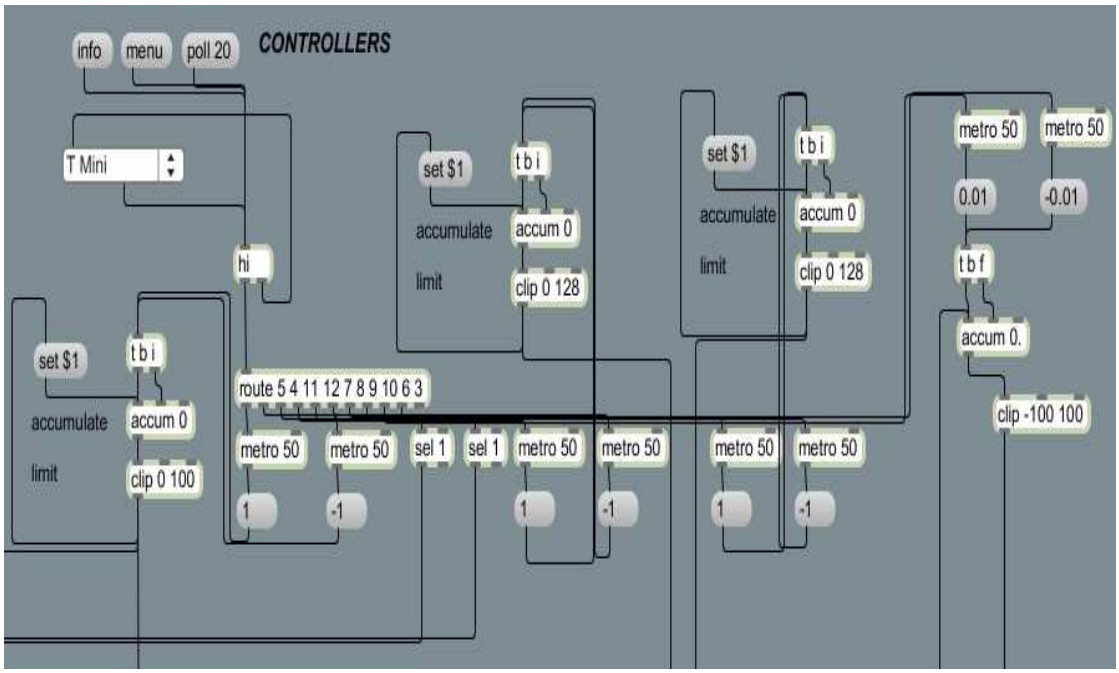
Η δουλειά των συγκεκριμένων αντικειμένων είναι να οριοθετούν τους αριθμούς που δέχονται σαν είσοδο, στο παράδειγμά μας τα όρια είναι ελάχιστο 0 και μέγιστο 100 στο πρώτο, και ελάχιστο -100 και μέγιστο 100 στο δεύτερο, αν κάποιος αριθμός από αυτούς που λαμβάνει σαν είσοδο το πρώτο αντικείμενο clip του προγράμματος μας ξεπεράσει το 100 σαν έξοδο το clip βγάζει πάλι 100 και δεν το αφήνει να ξεφύγει, το ίδιο γίνεται και με το αν η τιμή γίνει μικρότερη του 0 το clip δεν την αφήνει να περάσει και βγάζει σαν έξοδο το 0. Το αντίστοιχο συμβαίνει και στο 2ο clip του προγράμματος μας.

Select:

Η δουλεία του αντικειμένου select είναι επιλεκτικά να βγάζει μια έξοδο bang από την αριστερή έξοδο εφόσον η είσοδος που λαμβάνει από αριστερά ταιριάζει με την τιμή που είναι δεξιά του select, αν δεν ταιριάζει τότε η είσοδος βγαίνει από την δεξιά έξοδο χωρίς καμία αλλαγή. Από την δεξιά είσοδο μπορούμε να περάσουμε ένα number box για παράδειγμα και να αλλάξουμε την τιμή δεξιά του select.

5.3.2 Παράθεση και επεξήγηση του 3ου μέρους του προγράμματος του οποίου τα αντικείμενα αναλύσαμε παραπάνω.

Τώρα θα δούμε όλα μαζί τα αντικείμενα όπου αναλύσαμε παραπάνω.



Στο 3ο κομμάτι του προγράμματος μας έχουμε την εισαγωγή χρήσεις συσκευών ελέγχου, όπως είναι το joystick, το πληκτρολόγιο, το mouse, κ.α.. Ξεκινώντας από πάνω, το κουμπί info περνάει σαν είσοδο στο αντικείμενο hi και το αναγκάζει να βγάλει σαν έξοδο όλα τους διαθέσιμους controllers και τα στοιχεία αυτών στο παράθυρο πληροφοριών του MAX, το μήνυμα αυτό είναι απλά πληροφοριακό και δεν μας παρέχει κάτι στο πρόγραμμα μας. Περνώντας τώρα στο δεύτερο μήνυμα το menu περνάει και αυτό σαν είσοδο στο αντικείμενο hi και το αναγκάζει να βγάλει σαν έξοδο σε ένα umenu αντικείμενο όλες τις διαθέσιμες συνδεδεμένες συσκευές ελέγχου από όπου εκεί μπορούμε να επιλέξουμε πια συσκευή θα χρησιμοποιήσουμε. Στην συνέχεια και αφού έχουμε επιλέξει πια συσκευή θέλουμε να ενεργοποιήσουμε από το umenu έχουμε το μήνυμα poll20 το οποίο με την σειρά του κανονίζει τον χρόνο σε millisecond ανάμεσα στα γεγονότα τα οποία παράγονται από τις συσκευές εξόδου. Στην συνέχεια έχουμε το αντικείμενο hi το οποίο είναι η διεπαφή μας με τις HI συσκευές, και μας παρέχει όλες τις πληροφορίες για να μπορούμε να

αλληλεπιδρούμε με αυτές. Δέχεται στην είσοδο την επιλεγμένη HI συσκευή, και στην αριστερή του έξοδο βγάζει όλες τις πληροφορίες αυτής στο παράθυρο MAX, δηλαδή βγάζει τις τιμές που παράγονται από το πάτημα κάθε κουμπιού της συσκευής μας. Στην συνέχεια έχουμε το αντικείμενο route το οποίο έχει κάποιες τιμές δεξιά του ονόματος του, οι οποίες με την σειρά από αριστερά προς τα δεξιά έχουν και τις αντίστοιχες εξόδους τους από κάτω . Πατώντας τώρα κουμπιά της επιλεγμένης συσκευής μας παράγονται κάποιες τιμές οι οποίες περνάνε στο αντικείμενο route και συγκρίνονται με τις ήδη υπάρχουσες του αντικείμενου μας, αν κάποια τιμή είναι ίδια με κάποια του αντικειμένου route τότε έχουμε παραγωγή μιας τιμής 1 όταν είναι πατημένο το κουμπί και 0 όταν σταματήσει να είναι πατημένη. Αν η τιμή εισόδου δεν ταιριάζει με καμία του αντικειμένου τότε βγαίνει από την πιο δεξιά έξοδο.

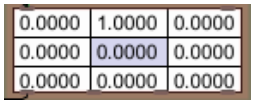
Τώρα αν η τιμή εισόδου μας είναι 5 ή 4 τότε έχουμε αποτέλεσμα μια τιμή 1 από την πρώτη έξοδο αν είναι 5 και μια τιμή 1 από την δεύτερη έξοδο αν είναι 4 και ενεργοποίηση του αντικειμένου metro50 και στις δύο περιπτώσεις το οποίο αντικείμενο παράγει ένα bang κάθε 50 millisecond και με την σειρά του ενεργοποιεί ένα μήνυμα το οποίο στέλνει έναν αριθμό 1 για την περίπτωση του 5 και έναν -1 για την περίπτωση του 4 στο αντικείμενο trigger, δηλαδή είτε στέλνεται ο ακέραιος 1 είτε ο -1, όπου με την σειρά του το trigger ενεργοποιείται και στέλνει από την αριστερή του έξοδο ένα bang και από την δεξιά του τον ακέραιο αριθμό που κάθε φορά λαμβάνει στην είσοδο του, το bang στην συνέχεια ενεργοποιεί το αντικείμενο accum περνώντας σε αυτό από την αριστερή του είσοδο και ο ακέραιος (είτε 1 είτε -1) περνάει μέσα στο αντικείμενο accum από την μεσαία είσοδο και προστίθεται στον αποθηκευμένο αριθμό που συγκρατεί το αντικείμενο accum, ο οποίος αριθμός εισάγεται στο accum μέσω της αριστερής εισόδου του με την βοήθεια του μηνύματος set $1 όπου με την σειρά του το set έχει πάρει τον αριθμό αυτό από την έξοδο του αντικειμένου clip. Το αντικείμενο clip λαμβάνει από το accum μια είσοδο και ελέγχει αν οι αριθμοί που λαμβάνει είναι μέσα στα όρια του 0 – 100 αν ναι τότε ο αριθμός βγαίνει από την έξοδο όπως είναι, αν τώρα ξεπεράσει το 100 τότε από την έξοδο συνεχίζει να βγάζει 100 και αν γίνει μικρότερος του 0 τότε βγάζει πάλι 0 σαν έξοδο, αυτή είναι και η δουλεία του να οριοθετεί τις τιμές εξόδου. Αυτό στο κομμάτι του προγράμματος μας που εξετάζουμε επαναλαμβάνεται άλλες τρείς φορές αλλά με άλλα όρια τιμών μέσα στο αντικείμενο clip. Επίσης έχουμε και τα αντικείμενα select ή σε συντομία sel τα οποία αν έχουμε είσοδο στο route είτε 11 είτε 12

ενεργοποιούνται οι αντίστοιχοι έξοδοί τους οι οποίοι βγάζουν ένα δεκαδικό το 1. Τώρα αν παρατηρήσουμε δεξιά του ονόματος sel του αντικειμένου υπάρχει μια τιμή το 1, οπότε αν ταιριάζει η είσοδος που θα λάβει από το αντικείμενο route με την τιμή αυτή στέλνει από την έξοδο του ένα bang, το οποίο προορίζεται για τον έλεγχο των διακοπτών gswitch που αναλύσαμε στο πρώτο μέρος του προγράμματος μας.

5.4 Χρήση φίλτρων στο πρόγραμμα μας με την μέθοδο του Convolution.

5.4.1 Επεξήγηση των αντικειμένων ένα προς ένα.

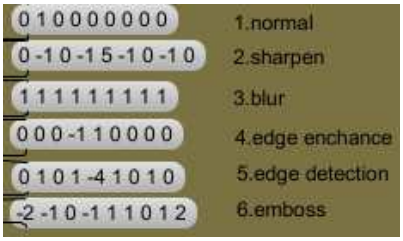
Jit.cellblock, select:



Το αντικείμενο jit.cellblock παρέχει αποθήκευση, εμφάνιση και επεξεργασία δεδομένων σε ένα πίνακα δύο διαστάσεων. Το format είναι παρόμοιο με αυτά των “grid” εργαλείων παρουσίασης που βρίσκονται σε άλλα προγραμματιστικά περιβάλλοντα. Η τρέχουσα θέση των κελίων, το format, η παρουσίαση και τα περιεχόμενα μέσα στο jit.cellblock μπορούν να ρυθμιστούν με το mouse ή χρησιμοποιώντας μηνύματα Max ή ακόμη και μέσω του inspector. Εμείς χρησιμοποιούμε το αντικείμενο αυτό για να μπορούμε να ρυθμίζουμε τις τιμές των φίλτρων που θα χρησιμοποιούμε στο πρόγραμμα μας με την μέθοδο του convolution οπού σε αυτήν την μέθοδο ο πίνακας μας ονομάζεται kernel. Το αντικείμενο μας για να εξάγει από την αριστερή του έξοδο το περιεχόμενο του πρέπει να δεχτεί ένα μήνυμα bang. Υπάρχουν επίσης και μηνύματα τα οποία μπορούν να ρυθμίσουν το περιεχόμενο των κελιών όλων μαζί ή του καθενός ξεχωριστά τα οποία θα δούμε στην συνέχεια. Εδώ έχουμε και το μήνυμα select το οποίο το περνάμε στην πάνω αριστερή είσοδο του πίνακα με τα ορίσματα 1, 1 οπού και αυτό το μήνυμα παίρνει είσοδο από το αντικείμενο loadbang, έτσι ώστε να ενεργοποιείται με το άνοιγμα του προγράμματος μας και να ορίζει ως προεπιλεγμένο κελί για τον πίνακα μας το κεντρικό, το γιατί θα το δούμε στην συνέχεια.

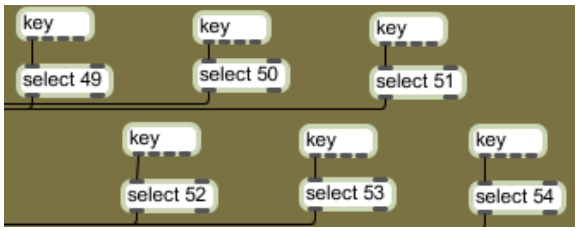
Jit.fill:

Το αντικείμενο αυτό μας χρησιμεύει στο γέμισμα ενός plane ενός matrix το οποίο έχει όνομα, με τις τιμές που περνάνε μέσω μιας λίστας από την αριστερή είσοδο. Οι offset και plane ιδιότητες καθορίζουν τα κελία που είναι για γέμισμα. Μόλις γίνει η συμπλήρωση, το jit.fill αντικείμενο στέλνει ένα bang μήνυμα από την αριστερή έξοδο. Εμείς εδώ το χρησιμοποιούμε για να περάσουμε κάποια μηνύματα με τις προκαθορισμένες τιμές έξη φίλτρων μέσα στο αντικείμενο jit.cellblock και στο αντικείμενο jit.matrix αυτός είναι και ο λόγος που έχουν το ίδιο όνομα. Το 0 είναι η default τιμή για το plane το οποίο θέλουμε να επηρεάσουμε. Το 0 σημαίνει ότι επηρεάζονται όλα τα plane.

Messages: 

Εδώ έχουμε 6 μηνύματα όπως προαναφέραμε παραπάνω με τις προκαθορισμένες τιμές έξη βασικών φίλτρων, πατώντας πάνω στα μηνύματα οι τιμές αυτές περνάνε μέσα στο αντικείμενο jit.cellblock με σειρά από πάνω προς τα κάτω και αριστερά προς τα δεξιά. Έχουμε με την σειρά τα εξής φίλτρα: 1. Είναι η κανονική εμφάνιση της εικόνας δηλαδή χωρίς αλλοιώσεις από κάποιο φίλτρο. 2. Είναι το φίλτρο “sharpen” το οποίο κάνει όξυνση των χρωμάτων. 3. Έχουμε το φίλτρο “blur” το οποίο θολώνει την εικόνα μας. 4. Έχουμε το φίλτρο “edge enhance” το οποίο τονίζει τις άκρες των διαφορών μέσα στη εικόνα μας σκουραίνοντας το υπόλοιπο κομμάτι. 5. Το φίλτρο “edje detection” το οποίο αναγνωρίζει τις ακμές μέσα στην εικόνα και της τονίζει. 6. Και τελευταίο το φίλτρο emboss το οποίο κάνει την εικόνα μας ανάγλυφη. Για τα φίλτρα αυτά έχουμε προσθέσει την δυνατότητα να ενεργοποιούνται και από το πληκτρολόγιο πατώντας τα κουμπιά με τους αριθμούς από 1 εις 6, κάτι το οποίο θα δούμε πως γίνεται ακριβώς παρακάτω.

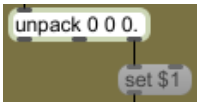
Key, select:



Εδώ έχουμε τα αντικείμενα key και select, το key αναγνωρίζει και βγάζει σαν έξοδο τους κωδικούς των κουμπιών του πληκτρολογίου τα οποία πατήθηκαν, σε τιμές ASCII. Το select βγάζει επιλεκτικά μια έξοδο bang, σε απόκριση κάθε εισόδου που ταιριάζει με την τιμή της μεταβλητής που έχει σαν όρισμα δίπλα από το όνομα select, από την αριστερή έξοδο του και από την δεξιά θα βγάλει τα μηνύματα εισόδου που δεν ταιριάζουν με το όρισμα του. Εδώ έχουμε το αντικείμενο key όπου βγάζει την τιμή ASCII ενός κουμπιού όταν πατηθεί από την αριστερή έξοδό του και περνάει την τιμή αυτή στην αριστερή είσοδο του αντικειμένου select, στην συνέχεια αν η τιμή που πήρε σαν είσοδο ταιριάζει με την τιμή που υπάρχει δεξιά του select τότε το select βγάζει από την αριστερή έξοδο ένα bang. Οι τιμές στα αντικείμενα select είναι οι τιμές ASCII των κουμπιών με τους αριθμούς από 1 έως 6 στο πληκτρολόγιο, και είναι συνδεδεμένα με τα μηνύματα για τα έξη φίλτρα που είδαμε παραπάνω.

Set:

Το μήνυμα set περνάει την τιμή που δέχεται από το αντικείμενο clip που αναλύσαμε στο προηγούμενο κομμάτι του προγράμματος μας με όριο τιμών από -100 μέχρι 100, τις οποίες τιμές τις ρυθμίζουμε από το joystick, απευθείας στο αντικείμενο jit.cellblock ρυθμίζοντας έτσι την τιμή του κελιού που βρίσκεται στην γραμμή και στήλη 1, 1 αυτό το βλέπουμε από τα δύο ορίσματα που έχουμε ακριβώς μετά το set, δηλαδή γεμίζει το κεντρικό κελί του cellblock. Επίσης στέλνεται ένα bang έτσι ώστε με το που αλλάξει η τιμή στο cellblock να βγει η αλλαγή στην έξοδο του αντικειμένου.

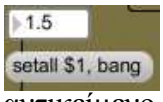
Unpack: 

Το αντικείμενο unpack δέχεται σαν είσοδο μια λίστα, χωρίζει τα στοιχεία της, και στέλνει το καθένα από ξεχωριστή έξοδο. Όσα ορίσματα έχει δεξιά του ονόματος τόσες και οι έξοδοι του. Στο πρόγραμμα μας τώρα σαν είσοδο παίρνει μια λίστα από την αριστερή έξοδο του αντικειμένου jit.cellblock, όπου η λίστα περιέχει τρείς αριθμούς οι δύο πρώτοι είναι οι αριθμοί του επιλεγμένου κελιού, και ο τρίτος είναι η τιμή που έχει το κελί αυτό. Επειδή όπως προανέφερα στην αρχή έχουμε βάλει το μήνυμα select να επιλέγει το μεσαίο κελί με την έναρξη του προγράμματος μας, αν δεν έχουμε αλλάξει αυτή την επιλογή τότε η έξοδος μας θα αφορά το κεντρικό κελί, του οποίου την τιμή θέλουμε να μπορούμε να αλλάζουμε μέσω του joystick. Έτσι η είσοδος μας στο unpack αναλύεται σε τρείς τιμές άρα το unpack θα έχει και τρείς εξόδους οι πρώτες δυο θα αφορούν την γραμμή και την στήλη του κελιού και από την τρίτη που είναι και αυτή που μας ενδιαφέρει θα βγαίνει η τιμή του κελιού. Η οποία τιμή θα περνά μέσα στο μήνυμα set $1, και εν συνεχεία θα περνάει μέσω του set στο αντικείμενο accum έτσι ώστε να γνωρίζουμε πάντοτε την τιμή του κεντρικού κελιού και όταν χρησιμοποιούμε το joystick για να αυξήσουμε ή να μειώσουμε την τιμή του να ξεκινάμε από την ήδη υπάρχουσα τιμή του κελιού. Δηλαδή με λίγα λόγια ολο αυτό γίνεται για να ενημερώνεται το αντικείμενο accum όταν η τιμή του κελιού αλλάζει.

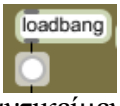
Jit.matrix:

Το αντικείμενο αυτό δημιουργεί ένα matrix με όνομα που μπορεί να χρησιμοποιηθεί για την αποθήκευση και ανάκτηση δεδομένων, επαναδειγματολειψία, και πράξεις μεταξύ matrix. Στο πρόγραμμα μας το αντικείμενο αυτό έχει κατασκευάσει ένα matrix 3*3 το οποίο το ονοματίζουμε named και τα δεδομένα του matrix είναι float32, παίρνει δεδομένα από το jit.fill οπού και έχουν το ίδιο όνομα γι αυτό τον λόγο, το matrix αυτό θα χρησιμοποιηθεί στην συνεχεία. Το jit.matrix κάθε

φορά που αλλάζει κάποια τιμή στο jit.cellbox δέχεται σαν είσοδο ένα bang από το αντικείμενο trigger και έτσι παράγει έξοδο.

Numberbox, setall: 

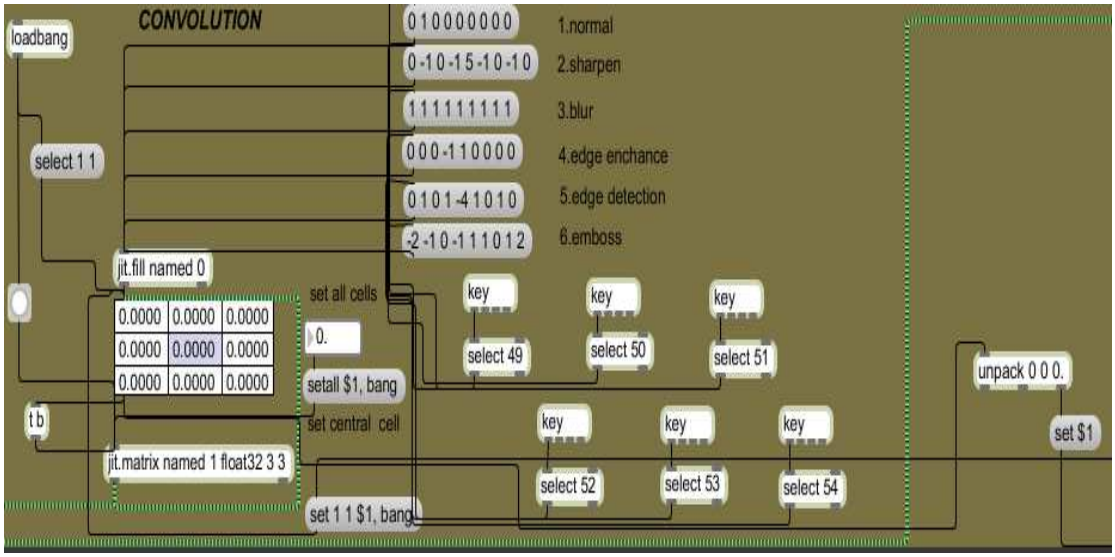
Εδώ έχουμε ένα αντικείμενο εισαγωγής δεκαδικών αριθμών το number box, και ένα μήνυμα setall το οποίο στέλνει την τιμή που λαμβάνει από το number box στο αντικείμενο jit.matrix μαζί με ένα μήνυμα bang για την ενεργοποίηση του αντικείμενου. Η τιμή που δέχεται το jit.matrix περνάει σε όλα τα κελία του jit.cellblock.

Loadbang, button: 

Βλέπουμε δύο αντικείμενα εδώ, ένα είναι το loadbang οπού η δουλειά του είναι με το που ανοίγει το αρχείο με το πρόγραμμα μας να στέλνει ένα bang αυτομάτως από την έξοδό του. Το οποίο bang στο πρόγραμμα μας ενεργοποιεί το αντικείμενο button, και στην συνέχεια ενεργοποιεί το αντικείμενο jit.matrix. Επίσης το αντικείμενο loadbang με την έναρξη του προγράμματος μας ενεργοποιεί και το μήνυμα select το οποίο επιλέγει το κεντρικό κελί του jit.cellblock.

5.4.2 Παράθεση και επεξήγηση του 4ου μέρους του προγράμματος του οποίου τα αντικείμενα αναλύσαμε παραπάνω.

Στην παρακάτω εικόνα βλέπουμε συνδεδεμένα όλα τα αντικείμενα που αναλύσαμε παραπάνω.



Στο 4ο κομμάτι του προγράμματος μας γίνεται η παραγωγή των φίλτρων με την μέθοδο του convolution. Έτσι εδώ έχουμε έναν πίνακα 3*3 ο οποίος ονομάζεται πίνακας kernel και είναι φτιαγμένος με το αντικείμενο jit.cellblock, στον οποίο μπορούμε να περνάμε τις τιμές κάποιων βασικών φίλτρων είτε με την βοήθεια κάποιων μηνυμάτων όπως βλέπουμε παραπάνω με την βοήθεια του αντικείμενου jit.fill, το οποίο αντικείμενο έχει και όνομα “named”, ακόμη αυτά τα έτοιμα φίλτρα μπορούμε να τα εισάγουμε στον πίνακα πατώντας κάποιο κουμπί από το πληκτρολόγιο με την βοήθεια του αντικείμενου key και select, είτε χειροκίνητα γράφοντας τις τιμές από μόνοι μας μέσα σε κάθε κελί, επίσης μπορούμε να βάλουμε τιμές με την βοήθεια των αντικείμενων number boxes από τα οποία το ένα αλλάζει το κεντρικό μόνο κελί και δέχεται τιμές από το joystick και το άλλο τις τιμές όλων των κελίων με την βοήθεια των μηνυμάτων setall και set, το πρώτο για την αλλαγή όλων των κελιών και το δεύτερο για την αλλαγή μόνο του μεσαίου κελιού. Επίσης έχουμε το αντικείμενο jit.matrix το οποίο έχει το ίδιο όνομα με το αντικείμενο jit.fill, ο λόγος που έχουν το ίδιο όνομα είναι γιατί αυτά τα δύο αντικείμενα ανταλλάζουν τιμές για το matrix, το αντικείμενο αυτό χρησιμεύει στην δημιουργία ενός matrix 3*3 με τα δεδομένα του να είναι του τύπου float32, το οποίο αποθηκεύει τις τιμές του πίνακα jit.cellblock οπού θα χρησιμοποιηθούν στην συνέχεια για την εφαρμογή των φίλτρων. Με το που ξεκινήσουμε την εφαρμογή μας το αντικείμενο αυτό ενεργοποιείται μέσω ενός κουμπιού το οποίο επίσης ενεργοποιείται μέσω του αντικείμένου loadbang. Επίσης το αντικείμενο loadbang ενεργοποιεί αυτομάτος και το μήνυμα select του οποίου η δουλεία είναι να επιλέγει το κεντρικό κελί του πίνακα ως default, έτσι ώστε από την πρώτη έξοδο του jit.cellblock να βγαίνει η τιμή του κελιού και να είναι πάντα ενημερωμένο το αντικείμενο accum το οποίο είναι υπεύθυνο για την αυξομείωση της τιμής του μέσω του joystick. Κάθε φορά που αλλάζει μια τιμή μέσα στο αντικείμενο jit.cellblock τότε από την αριστερή του έξοδο στέλνεται ένα μήνυμα bang μέσω του αντικειμένου trigger και μπαίνει στην είσοδο του jit.matrix έτσι ώστε να ενεργοποιηθεί και να στείλει τα καινούργια δεδομένα.

5.5 Διάφορες λειτουργίες παρουσίασης του αποτελέσματος του προγράμματος μας.

5.5.1 Παράθεση και επεξήγηση των αντικειμένων ένα προς ένα.

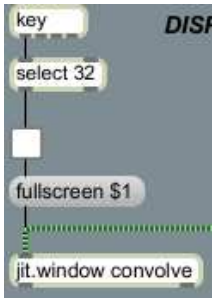
Jit.convolve:

Το αντικείμενο μας συνδυάζει ένα matrix με ένα άλλο με την μέθοδο του convolution kernel. Αυτό μπορεί να χρησιμοποιηθεί για απλή επεξεργασία εικόνας όπως είναι το “blurring”, “sharpening, και το “edge detection”. Εξορισμού, ο convolution kernel είναι ένας 3*3 float32 matrix. Στο πρόγραμμα μας από την αριστερή είσοδο του περνάμε την συνδυασμένη ή μη εικόνα μας και από την δεξιά το φίλτρο το οποίο θέλουμε να της εφαρμόσουμε. Δηλαδή κάνει όλη την διαδικασία με τις προσθέσεις και τους πολλαπλασιασμούς που αναφέραμε παραπάνω.

Jit.qt.record, write, stop:

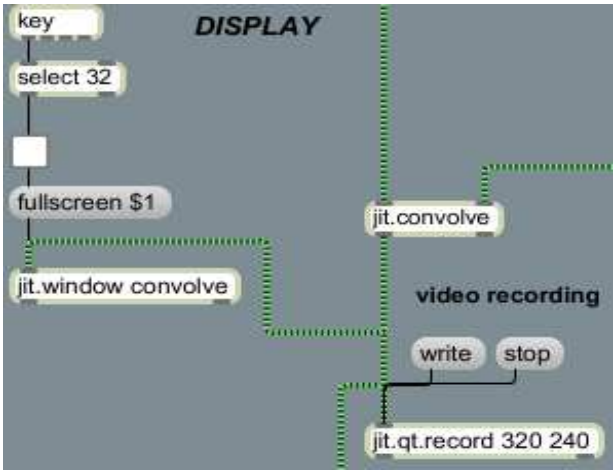
Εδώ βλέπουμε το αντικείμενο μας το οποίο λαμβάνει μια ακολουθία από Jitter matrices σαν είσοδο, σώζοντας αυτή την ακολουθία στον δίσκο μας σαν μια Quicktime ταινία. Το αντικείμενο μας είναι συνδεδεμένο με δύο μηνύματα το write το οποίο πατώντας το ανοίγει ένα παράθυρο διαλόγου ώστε να του περάσουμε το όνομα με το οποίο θέλουμε να αποθηκεύσουμε το αρχείο μας με την μορφή αρχείου .mov και επίσης για να καθορίσουμε το σημείο που θέλουμε να αποθηκευτεί το αρχείο. Έχουμε επίσης το μήνυμα stop το οποίο πατώντας το μας δίνει την δυνατότητα να σταματήσουμε την αποθήκευση του αρχείου. Το μέγεθος αποθήκευσης του αρχείου είναι 320*240 pixel.

Jit.window, key, select, toggle, fullscreen:



Ξεκινώντας από πρώτο στην σειρά αντικείμενο έχουμε, το key το οποίο αναγνωρίζει και βγάζει σαν έξοδο τους κωδικούς ASCII τον κουμπιών του πληκτρολογίου που πατιούνται, οι κωδικοί αυτοί περνάνε στο αντικείμενο select το οποίο δίπλα στο όνομά του έχει τον αριθμό 32 ο οποίος αντιστοιχεί στο κουμπί space του πληκτρολογίου. Τότε αν το κουμπί που πατήθηκε είναι το space από την αριστερή έξοδο του select βγαίνει ένα μήνυμα bang, το οποίο με την σειρά του ενεργοποιεί το αντικείμενο toggle, οπού toggle ένας διακόπτης ο οποίος έχει δύο καταστάσεις on και off. Με το που δεχτεί το bang από το πάτημα του κουμπιού space τότε ενεργοποιείται και στέλνει από την έξοδό του ένα 1 το οποίο ενεργοποιεί το μήνυμα fullscreen του οποίου η δουλεία είναι να ενεργοποιεί την πλήρη οθόνη παρουσίασης περνώντας στο αντικείμενο jit.window, όπου αυτό το αντικείμενο δημιουργεί μια οθόνη προβολής για matrices. Επίσης το jit.window χρησιμεύει για τον σχεδιασμό αντικειμένων, ακόμη αναφέρει την θέση του ποντικιού πάνω στην οθόνη μαζί με την χρήση κάποιων κουμπιών με σκοπό την διαδραστικότητα με τον χρήστη. Εδώ η χρήση του είναι για την εμφάνιση του αποτελέσματος του συνδυασμού των βίντεο με το convolution, η λέξη convolve δίπλα στο αντικείμενο είναι το όνομα του παραθύρου προβολής, το οποίο εμφανίζεται στο πάνω μέρος του παραθύρου του, με default μέγεθος 320*240 pixel.

5.5.2 Παράθεση και επεξήγηση του 5ου μέρους του προγράμματος μας του οποίου τα αντικείμενα αναλύσαμε παραπάνω.



Βλέπουμε εδώ τα αντικείμενα που αναλύσαμε παραπάνω συνδεδεμένα μεταξύ τους τα οποία κάνουν την παρουσίαση της εικόνας μας. Βλέπουμε πως μπορούμε με το πάτημα του κουμπιού space να έχουμε παρουσίαση σε πλήρη οθόνη με την βοήθεια των αντικειμένων key, select, toggle και του μηνύματος fullscreen το οποίο είναι ιδιότητα του αντικειμένου jit.window. Βλέπουμε επίσης ότι το αντικείμενο jit.window δέχεται είσοδο από το αντικείμενο jit.convolve το οποίο κάνει τον συνδυασμό του βίντεο με τα φίλτρα που του περνάμε με την μέθοδο του convolution kernel. Και βλέπουμε επίσης ότι και το αντικείμενο jit.qt.record δέχεται είσοδο από το jit.convolve για την εγγραφή σε .mov αρχείο του αποτελέσματος της ανάμιξης των βίντεο σε συνδυασμό με τα φίλτρα.

Στο 5ο κομμάτι του προγράμματος μας εμπεριέχεται και ένα αντικείμενο της μορφής

οπού p σημαίνει patcher και randplz είναι το όνομα του patcher. Εδώ έχουμε την δημιουργία ενός subpatch μέσα στο ήδη υπάρχον patch του προγράμματος μας το οποίο ονομάζεται ptixiaki. Το αντικείμενο patcher διευκολύνει την δημιουργία patches μέσα σε patches. Στο παράδειγμά το subpatch που έχουμε φτιάξει έχει τρείς εισόδους και καμία έξοδο, αυτό καθορίζεται από το patch randplz το οποίο χρειάζεται τρείς εισόδους για να λειτουργήσει τις οποίες τις παίρνει από το patch ptixiaki στο οποίο και εμπεριέχεται. Η έξοδος του randplz βγάνει μέσα στο ίδιο το patch οπότε επειδή δεν έχουμε κάποια επιστροφή δεδομένων στο patch ptixiaki γι

αυτό τον λόγο δεν έχουμε και έξοδο στο αντικείμενο p randplz. Οι είσοδοι που δέχεται είναι η έξοδος του jit.convolve και άλλοι δύο που έχουν να κάνουν με τον έλεγχο κάποιων στοιχείων του patch randplz μέσω συσκευών HID. Με αυτόν τον τρόπο μπορούμε να συνδέουμε patches για να μειώσουμε και την πολυπλοκότητα εμφάνισης τους.

5.6 Εμφάνιση των frames με τυχαία σειρά στο παράθυρο προβολής.

5.6.1 Παράθεση και επεξήγηση των αντικειμένων ένα προς ένα.

Loadbang:

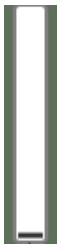
Το αντικείμενο loadbang ενεργοποιείται αυτόματα όταν ανοίξουμε το αρχείο μας ή όταν το patch που το περιέχει είναι κομμάτι ενός άλλου αρχείου το οποίο ανοίχτηκε. Βγάζει σαν έξοδο ένα bang. Μπορούμε επίσης να το κάνουμε να βγάλει ένα bang από την έξοδό του είτε στέλνοντάς του στην είσοδο του ένα bang είτε κάνοντας διπλό κλικ πάνω στο αντικείμενο.

Toggle:

Είναι ένας διακόπτης ο οποίος βγάζει στην έξοδο του ένα 0 όταν είναι κλειστός και ένα 1 όταν είναι ανοιχτός, (όταν δέχεται είσοδο ένα μη μηδενικό αριθμό θα ανοίξει και όταν δέχεται ένα μηδενικό θα κλείσει), επίσης ένα μήνυμα bang στην είσοδο του θα αλλάξει την κατάστασή του.

Metro:

Το αντικείμενο αυτό μπορεί να λειτουργήσει σαν μετρονόμος, ο οποίος βγάζει ένα bang κατά s προσδιορισμένα και τακτά χρονικά διαστήματα. Στο πρόγραμμά μας δέχεται δύο εισόδους η αριστερή είναι ένα bang που ενεργοποιεί το αντικείμενο και η δεξιά είναι αυτή που ρυθμίζει τον χρόνο s κατά τον οποίο το αντικείμενο μας θα βγάζει τα μηνύματα bang στην έξοδό του.

Slider: 

Το αντικείμενο αυτό είναι μια διεπαφή χρήστη με την μορφή ποτενσιόμετρου. Το slider αυτό έχει 128 τιμές, από 0 έως 128 τις οποίες βγάζει από την έξοδό του μια μια ανάλογα με την φάση της κατάστασής του.

Input:

Το input αυτό επιτρέπει την ανταλλαγή δεδομένων μεταξύ patcher. Όπως είδαμε στο προηγούμενο κομμάτι έχουμε το αντικείμενο p randplz με τρείς εισόδους. Το 2 μέσα στο input σημαίνει ότι αντιστοιχεί στην δεύτερη είσοδο του αντικειμένου μας. Η είσοδος που δέχεται εδώ είναι αριθμητικές τιμές από το joystick οι οποίες ρυθμίζουν την κατάσταση του slider. Οι τιμές που μπαίνουν στην είσοδο έχουν εύρος από 0 έως 128 όσες έχει και ο slider.

Τελεστής:

Το αντικείμενο αυτό προσθέτει σε οποίον αριθμό μπαίνει από την είσοδό του τον δεκαδικό αριθμό 24 και στην συνέχεια τον προωθεί στην έξοδό του.

Loadmess:

Το αντικείμενο αυτό ενεργοποιείται αυτόματα όταν ανοίξουμε το αρχείο που το περιέχει ή όταν το patch που εμπεριέχεται είναι κομμάτι κάποιου άλλου αρχείου που ανοίχτηκε. Με το που ενεργοποιηθεί βγάζει στην έξοδο του ότι έχει διπλά στο όνομά του. Στο παράδειγμά μας το ένα βγάζει τον ακέραιο 24 και το άλλο 50.

Qmetro:

Το αντικείμενο qmetro είναι παρόμοιο με τον συνδυασμό των αντικειμένων metro και jit.qball. Στο Jitter, οι περισσότερες εκτελέσεις συμβαίνουν σε χαμηλής

προτεραιότητας σειρά, για την αποφυγή εμφάνισης σχεδίων στην οθόνη σε περίπτωση καθυστερήσεων των matrix, χρησιμοποιούμε το qmetro αντί του metro. Αυτή είναι και η δουλειά του qmetro να στέλνει bang μήνυμα όχι ανά συγκεκριμένα και προσδιορισμένα χρονικά διαστήματα άλλα όποτε χρειάζεται.

Trigger:

Το trigger βγάζει στην έξοδό του οποιαδήποτε είσοδο έχει δεχθεί με σειρά από δεξιά προς τα αριστερά, σύμφωνα με το πως έχουν οριστεί οι τιμές τους μέσα στο αντικείμενο. Εδώ έχουμε μόνο μια έξοδο η οποία βγάζει ένα bang.

Input:

Το input1 είναι η πρώτη είσοδος του patch randplz, όπως προαναφέραμε και πριν, και δέχεται σαν είσοδο matrix από το patch ptixiaki.

Τελεστής:

Το αντικείμενο αυτό προσθέτει σε οποίον αριθμό μπαίνει από την είσοδό του τον αριθμό 10 και στην συνέχεια τον προωθεί στην έξοδό του.

Random:

Το αντικείμενο random λειτουργεί σαν μια μηχανή παραγωγής τυχαίων αριθμών, οπού εδώ το σύνολο των αριθμών από το οποίο θα παραχθεί ο τυχαίος αριθμός είναι το 0 έως 50. Αν δεν ήταν ορισμένο μέσα στο αντικείμενο θα ήταν από 0 έως 1. Επίσης το σύνολο των αριθμών μπορούμε να το προσδιορίσουμε εισάγοντας αριθμούς στην πάνω δεξιά είσοδο του αντικειμένου μας, στο πρόγραμμά μας περνάμε τιμές από το 10 έως το 137 μέσω του slider. Στην αριστερή είσοδο του αντικειμένου περνάμε την έξοδο του αντικειμένου metro για να έχουμε την δημιουργία εξόδου από το αντικείμενο random ανά τακτά και καθορισμένα χρονικά διαστήματα. Το αντικείμενο αυτό ενεργοποιείται με ένα μήνυμα bang.

Outputmatrix:

Περνάμε την τιμή της εξόδου του αντικειμένου random στο μήνυμα outputmatrix το οποίο με την σειρά του το περνάει στην είσοδο του αντικειμένου jit.matrixset, έτσι ώστε να ενεργοποιήσει το αντικείμενο αυτό να βγάλει στην έξοδο

του το matrix με τον συγκεκριμένο αριθμό που του περάσαμε. Αυτή είναι και η δουλεία αυτού του μηνύματος.

Gate:

Το gate θα προωθήσει ένα μήνυμα που θα λάβει από την δεξιά είσοδο του σε προκαθορισμένη έξοδο. Από την αριστερή είσοδο του καθορίζουμε αν θα είναι σε λειτουργία ή όχι το αντικείμενο μας, στο πρόγραμμά μας το ρυθμίζουμε αυτό με ένα αντικείμενο toggle, έτσι αν το αντικείμενο είναι ενεργοποιημένο τότε θα προωθήσει το μήνυμα από την έξοδο του αλλιώς όχι. Η λειτουργία του είναι παρόμοια όπως αυτή ενός διακόπτη.

Counter:

Το αντικείμενο counter βγάζει μια ακολουθία τιμών με την σειρά από την αριστερή έξοδο του, στο πρόγραμμά μας βγάζει την ακολουθία από 0 έως 50 με σειρά από το 0 προς το 50 αυτό γίνεται λόγο του πρώτου ορίσματος 0 που έχει το counter, αν είχε το 1 τότε οι τιμές θα έβγαιναν με την ανάποδη σειρά από το 50 προς το 0. Επίσης έχουμε την δυνατότητα να αλλάξουμε τον αριθμό 50 από την δεξιά είσοδο του αντικειμένου μας. Οι τιμές βγαίνουν από την έξοδο με τον ρυθμό που εισέρχονται τα μηνύματα bang από το αντικείμενο metro από την αριστερή είσοδο το οποίο και ενεργοποιεί το αντικείμενο μας.

Index:

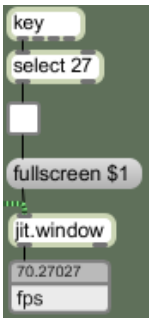
Το μήνυμα index παίρνει σαν είσοδο την τιμή της εξόδου του counter και την περνάει στην είσοδο του αντικειμένου jit.matrixset. Η δουλεία του μηνύματος index είναι να καθορίζει σε ποίο matrix του αντικειμένου jit.matrixset θα γραφτεί το εισερχόμενο matrix του input1 που προέρχεται από το patch ptixiaki.

Jit.matrixset:

Το jit.matrxiset αντικείμενο δημιουργεί κάποια matrices, και έτσι τα εισερχόμενα matrices που δέχεται από τη είσοδο του μπορούν να αποθηκευτούν σε αυτά και στην συνέχεια να διαβαστούν από αυτά με όποια σειρά του ορίσουμε εμείς. Αυτό μπορεί να φανεί χρήσιμο για αποθήκευση δεδομένων και/ή για τροποποιήσεις στην σειρά εξόδου των matrix. Το πρώτο όρισμα σημαίνει ότι δημιουργεί 750

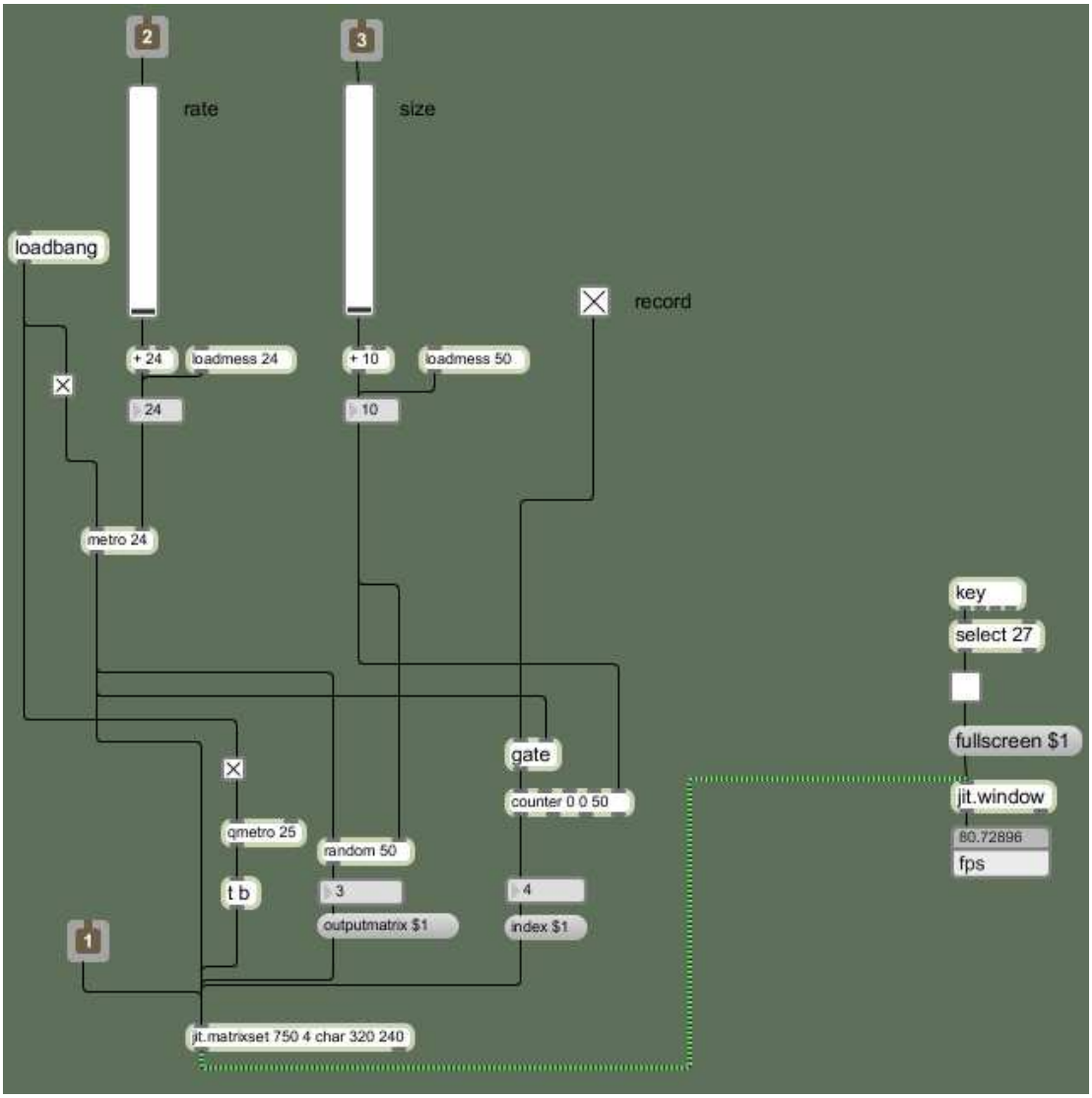
matrices, το δεύτερο ότι κάθε matrix θα έχει τέσσερα plane, το τρίτο ότι θα είναι τύπου char και οι τελευταίες τιμές είναι οι διαστάσεις του κάθε matrix.

Key, select, toggle, fullscreen, jit.window, jit.fpsgui:



Τα αντικείμενα key, select, toggle, jitwindow και το μήνυμα fullscreen τα έχουμε ξαναδεί με την ίδια ακριβώς διάταξη, το μόνο που αλλάζει είναι η τιμή του ορίσματος του select η οποία είναι η τιμή του κουμπιού escape του πληκτρολογίου σε κώδικα ASCII και το χρησιμοποιούμε για να φέρουμε σε πλήρη οθόνη το παράθυρο προβολής (jit.window) του patcher randplz πατώντας το. Το μόνο καινούργιο αντικείμενο είναι το jit.fpsgui το οποίο είναι μια διεπαφή χρήστη η οποία μας αναφέρει πληροφορίες για τον χρόνο και τα matrix που εμφανίζονται στο παράθυρο προβολής. Μπορεί να μας αναφέρει τα fps (frames ανά δευτερόλεπτο), ms (milliseconds μεταξύ των εισερχόμενων δεδομένων), τον τύπο των matrix, τις διαστάσεις, τον αριθμό και το όνομα των planes. Στο πρόγραμμά μας, βλέπουμε ότι ο αριθμός των frames ανά δευτερόλεπτο κυμαίνεται από 45 μέχρι 74, ενώ οι φυσιολογικές τιμές είναι περίπου στο 40fps, αυτό συμβαίνει γιατί έχουμε δυο αντικείμενα που ενεργοποιούν το jit.matrixset να στέλνει frames στην έξοδό του προς εμφάνιση, ένα είναι το metro του οποίου η τιμή είναι ρυθμιζόμενη μέσω του joystick και το δεύτερο είναι το qmetro του οποίου η τιμή είναι σταθερή.

5.6.2 Παράθεση και επεξήγηση του 6ου μέρους του προγράμματος μας του οποίου τα αντικείμενα αναλύσαμε παραπάνω.



Με το άνοιγμα του αρχικού μας patcher με όνομα ptyxiaki έχουμε και την απευθείας ενεργοποίηση και του δεύτερου patcher το οποίο βλέπουμε παραπάνω και ονομάζεται randplz. Το patcher αυτό κάνει παρουσίαση των matrices ενός βίντεο σε μια οθόνη προβολής αλλά με τυχαία σειρά. Θα εξηγήσουμε πως επικοινωνούν μεταξύ τους τα αντικείμενα για την παραγωγή του αποτελέσματος.

Με την ενεργοποίηση στέλνεται ένα bang μήνυμα από το αντικείμενο loadbang στα δύο toggle με τα οποία είναι συνδεδεμένα, στην συνέχεια αυτά ενεργοποιούνται και στέλνουν από τις εξόδους τους ένα 1. Το ένα toggle ενεργοποιεί το αντικείμενο qmetro, όπου αυτό στέλνει ένα bang κάθε 25ms στο trigger και αυτό μετά στέλνει ένα bang κάθε 25ms στο jit.matrixset. Το δεύτερο toggle ενεργοποιεί ένα αντικείμενο metro με αρχική τιμή 24ms την οποία μπορούμε να αλλάξουμε μέσω μιας κυλιόμενης μπάρας με την οποία συνδέεται στην δεξιά είσοδο του ή μέσω του joystick το οποίο στέλνει τιμές στην μπάρα. Η έξοδος bang του metro περνάει στις εισόδους των αντικειμένων jit.matrixset, random, και gate τα οποία και ενεργοποιεί. Μια δεύτερη κυλιόμενη μπάρα παράγει τιμές από 10 έως 137, οπού και αυτή ρυθμίζεται είτε μέσω joystick είτε απλά πατώντας και σέρνοντας με το mouse πάνω στην μπάρα, οι τιμές αυτές περνάνε στα αντικείμενα random και counter. Το random παράγει τυχαίους αριθμούς από ένα εύρος αριθμών από το 0 έως τον αριθμό που του έχουμε περάσει από την μπάρα, οι οποίοι στην συνέχεια περνάνε μέσω του μηνύματος outputmatrix στο αντικείμενο jit.matrixset και το ενεργοποιεί για να βγάλει σαν έξοδο το matrix που έχει αποθηκευμένο με τον ίδιο αριθμό εισόδου που πήρε από το μήνυμα. Στην συνέχεια έχουμε το αντικείμενο gate το οποίο παίρνει στην δεξιά του είσοδο την έξοδο του metro24 όπως προαναφέραμε, και στην αριστερή, παίρνει είσοδο από ένα διακόπτη toggle, έτσι όταν ενεργοποιηθεί το αντικείμενο gate βγάζει στην έξοδό του τα μηνύματα bang που του περνάνε από την δεξιά είσοδο, εν συνεχεία τα μηνύματα αυτά ενεργοποιούν το αντικείμενο counter το οποίο μπορεί να βγάλει στην αριστερή έξοδο ακολουθίες αριθμών από το 0 έως και το 10 που είναι το μικρότερο ή έως και το 137 όπου είναι το μεγαλύτερο, αυτό ορίζεται από την δεξιά είσοδο του counter και την τιμή που βγαίνει από την κυλιόμενη μπάρα. Στην συνέχεια αυτός ο αριθμός περνάει μέσω του μηνύματος index στο αντικείμενο jit.matrixset και καθορίζει σε ποιό matrix του αντικειμένου αυτού θα γραφτεί το εισερχόμενο matrix από το input1.

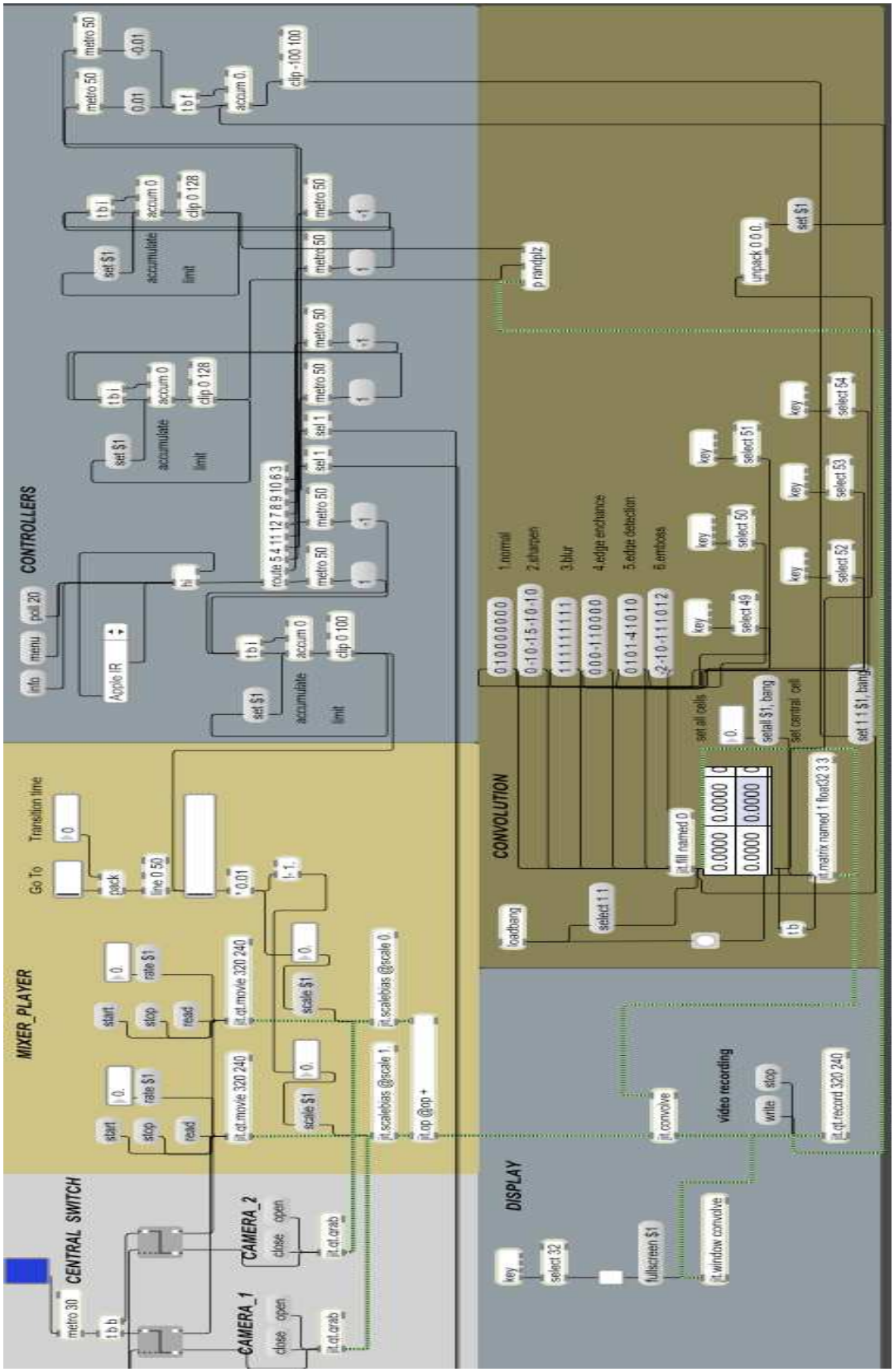
Τώρα έχουμε το αντικείμενο jit.matrixset με 5 εισόδους, το input1 που είναι η είσοδος των matrix από το patch ptixiaki, την είσοδο από το metro, την είσοδο από το qmetro,και τις εισόδους από τα μηνύματα outputmatrix και index. Στην έξοδό του βγάζει με τυχαία σειρά, λόγο του αντικειμένου random, matrices από την είσοδο του input1 τα οποία είχαν αποθηκευτεί στα matrices που είχε δημιουργήσει το

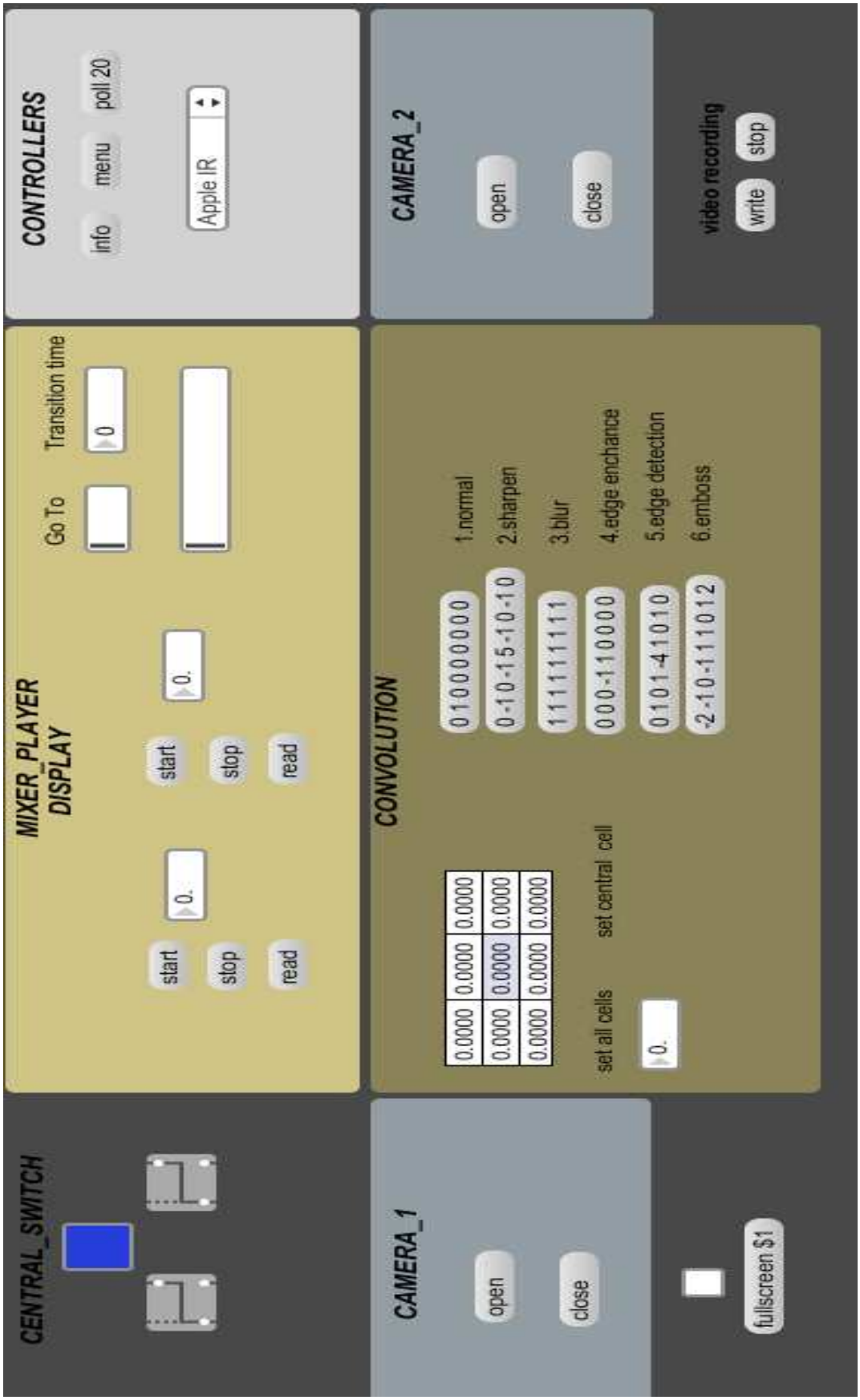
αντικείμενο jit.matrixset, και στην συνέχεια τα εμφανίζει σε ένα παράθυρο προβολής με την βοήθεια του αντικείμενου jit.window.

5.7 Επεξήγηση για τον τρόπο εμφάνισης, και διεπαφής του Max/MSP/Jitter.

Το Max/MSP/Jitter παρέχει δυνατότητες ρύθμισης για τον τρόπο εμφάνισης και παρουσίασης των αντικειμένων μέσα στο patcher. Δηλαδή όλα τα αντικείμενα που χρησιμοποιούμε παραμετρικοποιούντε ως προς το χρώμα τους το μέγεθος και το σημείο εμφάνισης μέσα στο patcher, όλα τα παραπάνω μπορούμε να τα ρυθμίσουμε μέσα από το παράθυρο του inspector κάθε αντικειμένου. Επίσης βλέπουμε ότι τα αντικείμενα έχουν τριών τύπων καλώδια διασύνδεσης μεταξύ τους, το συνεχόμενο καλώδιο που αφορά αντικείμενα του Max, το διακεκομμένο που αφορά αντικείμενα του MSP, και το διακεκομμένο αλλά πιο χοντρό και χρωματιστό που αφορά αντικείμενα του Jitter. Με αυτόν τον τρόπο μπορούμε να ξεχωρίσουμε και τον τύπο του αντικειμένου και τον τύπο τον δεδομένων που περνάει μέσα από αυτά τα καλώδια. Έχουμε επίσης το αντικείμενο panel το οποίο σου επιτρέπει να δημιουργήσεις ορθογώνια πλαίσια φόντου για την δημιουργία διεπαφών χρήστη. Μπορείς επίσης να δημιουργήσεις ορθογώνια με στρογγυλεμένες γωνίες και σκιές τα οποία μπορούν να χρησιμοποιηθούν σαν κουμπιά όταν χρησιμοποιηθούν σε συνεργασία με ένα ubutton αντικείμενο. Εμείς στο πρόγραμμά μας το έχουμε χρησιμοποιήσει για να κάνουμε πιο όμορφο στην παρουσίαση το patcher μας και να ξεχωρίζουμε τα διάφορα μέρη του τοποθετώντας διαφορετικό χρώμα στο καθένα. Υπάρχει επίσης το αντικείμενο comment το οποίο χρησιμοποιείται μόνο για την εμφάνιση κειμένου που πληκτρολογείται μέσα στο αντικείμενο για τον σκοπό της επικεφαλίδας μέσα στο patcher ή σαν επεξηγητικό κείμενο. Τελευταίο και πολύ βασικό στοιχείο είναι το presentation mode το οποίο χρησιμεύει για την απλούστευση ενός patch όσον αφορά την παρουσίαση του, στην ιδιότητα αυτή έχουμε την δυνατότητα να αφήσουμε εμφανή μέσα στο patch μόνο όσα αντικείμενα μας χρειάζονται για την παραγωγή αποτελέσματος και όλα τα άλλα να τα αποκρύψουμε μαζί και τα καλώδια σύνδεσης και ότι άλλο δεν θα μας φανεί χρήσιμο στην

λειτουργία του. Βλέπουμε στις εικόνες παρακάτω τι ακριβώς εννοούμε. Στην πρώτη εικόνα βλέπουμε πόσο πολύπλοκο εμφανίζεται το πρόγραμμα διότι φαίνονται όλα τα αντικείμενα που το αποτελούν και τα καλώδια σύνδεσης, στην δεύτερη εικόνα εμφανίζονται μόνο όποια αντικείμενα μας χρειάζονται για την παραγωγή αποτελέσματος και η διεπαφή μας δείχνει πολύ πιο απλή προς χρήση.


CONTROLLERS
MIXER_PLAYER
CENTRAL SWITCH
CONVOLUTION
DISPLAY
CAMERA_1
CAMERA_2
video recording
Go To
Transition time
1.normal
2.sharpen
3.blur
4.edge enchance
5.edge detection
6.emboss


CENTRAL_SWITCH
CONTROLLERS
info
menu
poll 20
Apple IR
MIXER_PLAYER
DISPLAY
Go To
Transition time
0
start
0.
stop
read
start
0.
stop
read
CAMERA_2
open
close
video recording
write
stop
CAMERA_1
open
close
fullscreen $1
CONVOLUTION
0.0000 0.0000 0.0000
0.0000 0.0000 0.0000
0.0000 0.0000 0.0000
set all cells
set central cell
0.
0 1 0 0 0 0 0 0 0
0 -1 0 -1 5 -1 0 -1 0
1 1 1 1 1 1 1 1 1
0 0 0 -1 1 0 0 0 0
0 1 0 1 -4 1 0 1 0
-2 -1 0 -1 1 1 0 1 2
1.normal
2.sharpen
3.blur
4.edge enchance
5.edge detection
6.emboss

Βιβλιογραφία

Wikipedia: http://en.wikipedia.org/wiki/Max_(software)

Ιστοσελίδα λογισμικού: CYCLING'74:http://cycling74.com/

Convolution matrix: http://docs.gimp.org/en/plug-in-convmatrix.html

Projects: http://cycling74.com/category/projects/

Help files του λογισμικού Max/MSP/Jitter 5.