

Μουσική Πληροφορική



Δ. Πολίτης, Τμήμα Πληροφορικής ΑΠΘ, 2015

[Άδεια Χρήσης]



ανοιτά μαθήματα
opencourses

Άδεια Χρήσης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων, π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».

[1] <http://creativecommons.org/licenses/by/4.0/>



[Άδεια Χρήσης]



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Ανοιχτός κώδικας και Γλώσσες Προγραμματισμού στη Μουσική Πληροφορική

Δ. Πολίτης - 8^ο Μάθημα

Το Λογισμικό Ανοικτού Κώδικα

Λογισμικό Ανοικτού Κώδικα:

- Τι είναι, τι σημαίνει, τι πρεσβεύει, πώς προέκυψε
- Γνωστά προϊόντα ανοιχτού κώδικα
- Ανοιχτός Κώδικας -vs- Κλειστός Κώδικας
- Pros & Cons

Εφαρμογές στη Μουσική Πληροφορική:

- Software Synthesis, CSound
- CSound & GUI, CSound-AV
- Artsbuilder
- Pure Data

Το Λογισμικό Ανοικτού Κώδικα

- Οι εφαρμογές που συνήθως συναντάμε στον κόσμο του ανοικτού κώδικα γίνονται ανάπτυξη από διάφορους προγραμματιστές οι οποίοι έχουν ανάγκη από μια συγκεκριμένη εφαρμογή που να έχει και συγκεκριμένες λειτουργίες. Λόγω όμως της φύσης της αρχιτεκτονικής του open source και ανάλογα από την ποιότητα του κώδικα μπορεί να ενδιαφερθούν και άλλοι προγραμματιστές οδηγώντας κτίζοντας έτσι μια εφαρμογή που έχει όλες τις δυνατότητες που μπορεί να έχει ανάγκη ο χρήστης, να είναι απλή στη χρήση και γρήγορη στην εκτέλεση. Αυτό είναι ό,τι θα μπορούσαμε να ονομάσουμε commercial quality εφαρμογή

Το Λογισμικό Ανοικτού Κώδικα

- Λέξεις κλειδιά:
 - Open Source
 - Free Software
 - Closed Source
 - Closed Format
 - Licenses
 - Free Software Foundation (FSF)
 - GNU (Gnu is Not Unix)
 - Open Source Initiative
- Εφαρμογές στη Μουσική Πληροφορική:
 - Ελευθερία στον τρόπο χρήσης
 - Ελευθερία στη διανομή
 - Ελευθερία στην τροποποίηση

Το Λογισμικό Ανοικτού Κώδικα

Λογισμικό Ανοικτού Κώδικα:

- Τι είναι, Τι σημαίνει, Τι πρεσβεύει, Πώς προέκυψε
- Γνωστά προϊόντα ανοικτού κώδικα
- Ανοιχτός Κώδικας -vs- Κλειστός Κώδικας
- Pros & Cons

Εφαρμογές στη Μουσική Πληροφορική:

- Software Synthesis, CSound
- CSound & GUI, CSound-AV
- Artsbuilder
- Pure Data

Το Λογισμικό Ανοικτού Κώδικα

- Λογισμικό Ανοικτού Κώδικα:

Όπως γενικά συμβαίνει στην Πληροφορική, έτσι και στη Μουσική Πληροφορική έχει ενδιαφέρον να εξετάσουμε τι σχέση μεταξύ του Κινήματος Ελευθέρου Λογισμικού και του χώρου της Μουσικής.

Αρχικά θα εξετάσουμε τι ακριβώς σημαίνει:

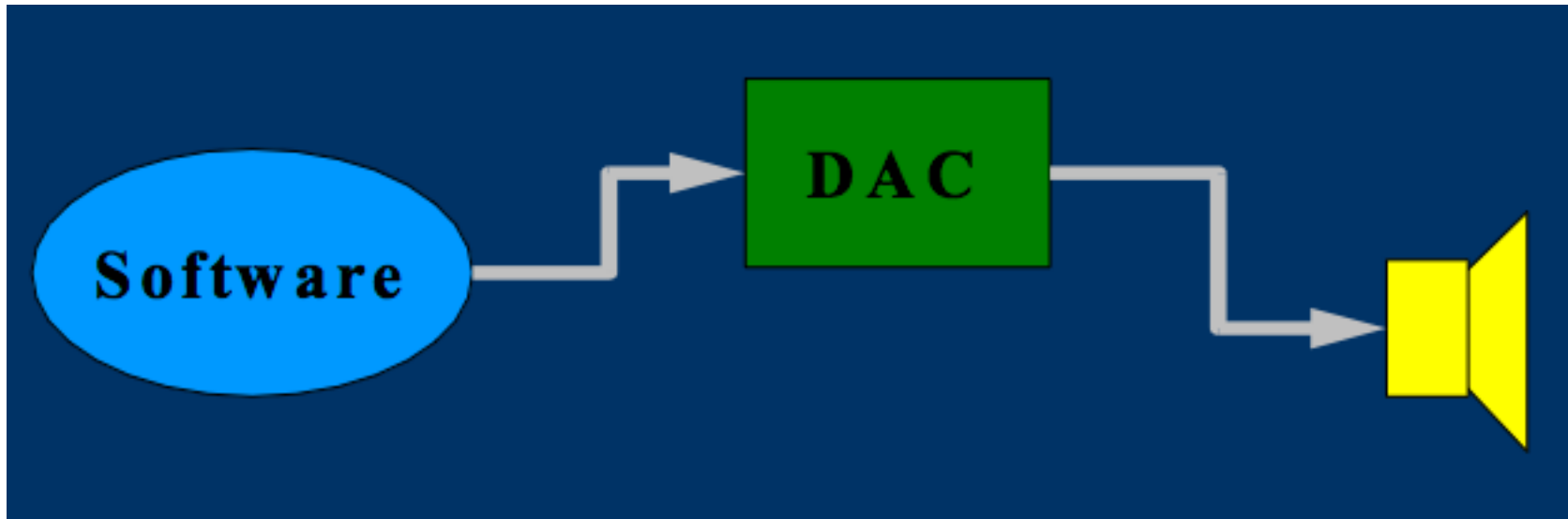
- η ελεύθερη, δωρεάν και χωρίς περιορισμούς διακίνηση μουσικών clips
- η δωρεάν διανομή λογισμικού που υποβοηθάει την παραγωγή της μουσικής
- η μάθηση με Λογισμικού Ανοικτού Κώδικα

Το Λογισμικό Ανοικτού Κώδικα

- Η διαδρομή του Λογισμικό Ανοικτού Κώδικα:
 - Στην αρχή ήταν το UNIX
 - Ο δρόμος του BSD προς τον ανοιχτό κώδικα και το GNU project
 - Η δημιουργία του Linux Open Source Initiative
- Εφαρμογές στη Μουσική Πληροφορική:
 - Ο τελευταίο κρίκος στην αλυσίδα αυτή, με πολλαπλές επισημάνσεις: το λειτουργικό σύστημα Android
 - Η **έννοια του Open Source** δεν αφορά μονάχα την πρόσβαση που έχει κάποιος χρήστης στον κώδικα, αλλά και την ίδια τη χρήση της εφαρμογής, των δεδομένων που αποθηκεύει και τον τρόπο με τον οποίο εκτελεί κάποιες λειτουργίες. Συχνά συναντάμε την έννοια του Open Source σε συνδυασμό με κάποιο Open Format.

Η Γλώσσα Μουσικού Προγραμματισμού Csound

- Η προγραμματιστική αυτή γλώσσα υλοποιεί την παρακάτω δομή:



- Στις μέρες μας έχει διαφαινεται μία ισοδυναμία υλικού (hardware) και λογισμικού (software). Μπορούμε να συμπεράνουμε πως ότι κι αν υλοποιήσει κανείς κάτι με τη χρήση υλικού, τότε θα μπορούσε κάλλιστα να το υλοποιήσει και με τη χρήση λογισμικού. Με βάση αυτή τη διαπίστωση μπορεί κανείς να αποπειραθεί να αντικαταστήσει τα ηλεκτρονικά κυκλώματα που παράγουν ήχο με προγράμματα τα οποία υπολογίζουν τα δείγματα του ήχου με χρήση επεξεργαστών. Τα δείγματα αυτά με τη σειρά τους αποστέλλονται σε έναν Digital to Analog Converter (D/A-C) με αποτέλεσμα την παραγωγή ήχου.

Η Γλώσσα Μουσικού

Προγραμματισμού Csound

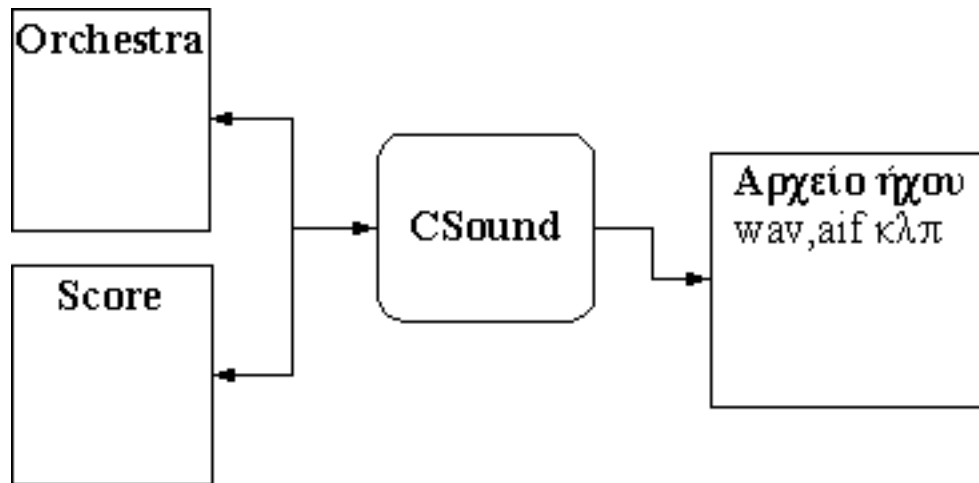
Η δημοτικότητα αυτή της CSound οφείλεται κυρίως σε **τρεις λόγους**:

- Πρόκειται ουσιαστικά για ένα από τα πρώτα, αν όχι το πρώτο εργαλείο αυτού του τύπου. Η θεμελιώδης συμβολή της CSound στη δημιουργία και εξάπλωση αυτού που σήμερα είναι ευρύτερα γνωστό ως Μουσική Πληροφορική παραλληλίζεται από πολλούς με αυτή με της C στον κλασικό προγραμματισμό (κάτι τέτοιο μαρτυρά άλλωστε και ίδιο της το όνομα).
- Πολύ γρήγορα υπήρξαν εκδόσεις της CSound για σχεδόν όλα τα δημοφιλή λειτουργικά συστήματα όπως UNIX, Linux, Irix, OSX, Windows, ακόμη και για πλατφόρμες λιγότερο διαδεδομένες σήμερα όπως Atari. Αυτό και μόνο το γεγονός, που σήμερα μπορεί να μη φαντάζει ιδιαίτερα σημαντικό, την εποχή εκείνη υπήρξε ένα σημαντικό πλεονέκτημα καθώς ο χώρος της Μουσικής πληροφορικής δεν ήταν ευρέως θεμελιωμένος.
- Κυριότερο ίσως λόγο αποτελεί το γεγονός ότι η CSound διανέμεται δωρεάν, μέσω του Διαδικτύου. Παράλληλα αποτελεί ένα πολύ δυνατό εργαλείο τόσο για τον μουσικό που είναι διατεθειμένος να δαπανήσει λίγο χρόνο για την εκμάθησή της, όσο και γενικότερα για οποιονδήποτε την προσεγγίσει σαν ένα μέσο διασκέδασης, δημιουργίας αλλά και σαν ένα ερευνητικό εργαλείο.

Η Γλώσσα Μουσικού

Προγραμματισμού Csound

Η CSound λοιπόν λειτουργεί σαν ένας μεταγλωττιστής (Compiler) όχι πολύ διαφορετικός από τον μεταγλωττιστή μιας κλασικής γλώσσας προγραμματισμού όπως η C ή Pascal. Η διαδικασία που ακολουθείται από τον χρήστη παρουσιάζεται στο σχήμα που ακολουθεί. Όπως φαίνεται, ο χρήστης δίνει σαν είσοδο δυο αρχεία. Το αρχείο Orchestra και το αρχείο Score. Από τα αρχεία αυτά η CSound παράγει ένα αρχείο ήχου το οποίο μπορεί να είναι κάποιου από τους γνωστούς τύπους (wav, aiff κλπ).



Η Γλώσσα Μουσικού

Προγραμματισμού CSound

- **Το αρχείο orchestra:** περιλαμβάνει μια σειρά από δηλώσεις που ορίζουν ένα σύνολο από όργανα (instruments). Ένα όργανο είναι στην πράξη ένας αλγόριθμος που υπαγορεύει στην CSound πως να παράξει έναν ήχο ή πως να τροποποιήσει έναν ήδη υπάρχοντα ήχο.
- **Το αρχείο score:** υπαγορεύει στην CSound πότε να ενεργοποιήσει το κατάλληλο όργανο, καθώς και για πόσο.

Θα μπορούσαμε να παραλληλίσουμε το αρχείο orchestra με μια ορχήστρα κλασικής μουσικής ενώ το αρχείο score με το μάεστρο που τη διευθύνει.

Η Γλώσσα Μουσικού

Προγραμματισμού CSound

Το **αρχείο orchestra** ορίζει ένα σύνολο από όργανα, καθένα από τα οποία είναι στην ουσία συναρτήσεις οι οποίες υπαγορεύουν στον υπολογιστή πως να "παίξει" κάποιον ήχο. Στην πραγματικότητα κάθε όργανο είναι σε θέση εκτελέσει μια από τις εξής λειτουργίες:

- Να παράξει κάποιον ήχο κυριολεκτικά από το μηδέν, για παράδειγμα κάνοντας δειγματοληψία πάνω σε μια ημιτονοειδή κυματομορφή η οποία δημιουργείται με βάση τον κώδικα και τα δεδομένα που εισήγαγε ο χρήστης.
- Να τροποποιήσει κάποιο ηχητικό σήμα, σαν ένα από αυτά που δημιουργούνται με την προηγούμενη διαδικασία, δρώντας πάνω στο ηχητικό σήμα σαν ένα φίλτρο.

Η Γλώσσα Μουσικού

Προγραμματισμού CSound

Ένα αρχείο orchestra χωρίζεται σε **δύο μέρη**: τις **δηλώσεις κεφαλίδας (Header Statements)** και τα **όργανα (Instrument Block Statements)**. Στις δηλώσεις της κεφαλίδας ορίζονται διάφορες global παράμετροι που θα παραμείνουν ανέπαφες καθ' όλη τη διάρκεια του compile, και εκτελούνται μόνο μια φορά κατά την χρόνο εγκατάστασης της "ορχήστρας". Από την άλλη πλευρά τα blocks των οργάνων μπορούν να περιλαμβάνουν μια σειρά από εκφράσεις οι οποίες μπορούν να θέτουν τιμές, να ελέγχουν τη ροή ή να καλούν διάφορες υπό-ρουτίνες επεξεργασίας σήματος.

Οι **βασικοί κανόνες σύνταξης** ενός τέτοιου αρχείου είναι οι εξής:

- Σε κάθε γραμμή μπορεί να υπάρχει μόνο μια δήλωση.
- Ο ορισμός κάθε οργάνου ξεκινά με τη δήλωση `instr <x>`, όπου `<x>` κάποιος αριθμός εκτός από το 0 καθώς ως όργανο `instr 0` θεωρείται η κεφαλίδα. Το `x` μπορεί να πάρει οποιαδήποτε θετική ακέραια τιμή, οσοδήποτε μεγάλη, αν και καλά είναι οι ιδιαίτερα μεγάλες τιμές να αποφεύγονται.
- Το πέρας του ορισμού κάθε οργάνου δηλώνεται με την δήλωση `endin`.
- Δεν υπάρχει περιορισμός στο πλήθος των οργάνων που περιλαμβάνονται σε ένα αρχείο, αλλά δεν είναι επιτρεπτή η ενφώλευση ενός οργάνου σε ένα άλλο.

Η Γλώσσα Μουσικού

Προγραμματισμού CSound

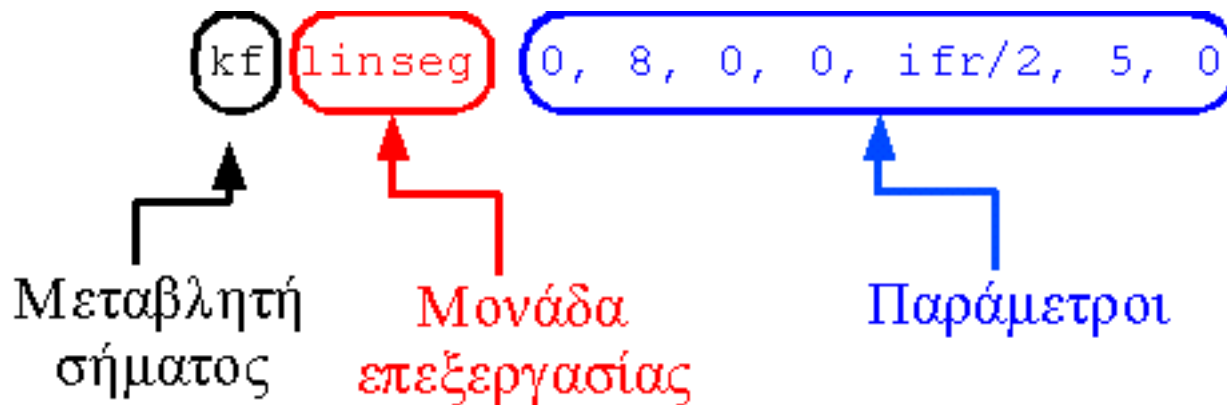
■ Σε γενικές γραμμές κάθε δήλωση μπορεί να έχει την εξής μορφή:

<μεταβλητή σήματος> <μονάδα επεξεργασίας> <παράμετρος 1> <παράμετρος 2>
<παράμετρος n>

το πλήθος των παραμέτρων μπορεί να ποικίλει.

πχ: α1 oscil 10000, 2.5, 1

kf linseg 0, 8, 0, 0, ifr/2, 5, 0



Η Γλώσσα Μουσικού

Προγραμματισμού CSound

- Για την καλύτερη κατανόηση των όσων ειπώθηκαν, να ένα παράδειγμα αρχείου τύπου orchestra

```
sr = 22050  
kr = 1470  
ksmps = 15  
nchnls = 1
```

Δηλώσεις
Κεφαλίδας

```
instr 1  
a1 oscil p4, p5, 1  
out a1  
endin
```

Δήλωση
Οργάνου

Η Γλώσσα Μουσικού

Προγραμματισμού CSound

```
sr = 22050
kr = 1470
ksmps = 15
nchnls = 1
```

Δηλώσεις
Κεφαλίδας

```
instr 1
a1 oscil p4, p5, 1
   out a1
endin
```

Δήλωση
Οργάνου

- Στις δηλώσεις κεφαλίδας έχουμε αρχικά τον ορισμό της συχνότητας δειγματοληψίας την οποία ορίζουμε να έχει τιμή 22050 ($sr = 22050$). Ακολουθεί ο ορισμός της μεταβλητής kr , η οποία ορίζει τη συχνότητα με την οποία θα αποστέλλονται συγκεκριμένα σήματα ελέγχου τα οποία ορίζουν παραμέτρους οι οποίες ελέγχου πράγματα όπως το vibrato και το tremolo. Η μεταβλητή $ksmps$ είναι ο λόγος sr / kr . Τέλος έχουμε το πλήθος των καναλιών που θα χρησιμοποιήσουμε που στην περίπτωση μας είναι ένα (μονοφωνικό).
- Στη συνέχεια ακολουθεί ο ορισμός του μοναδικού στο παράδειγμά μας οργάνου. Όπως φαίνεται και στο σχήμα το ορίζουμε ως $instr 1$, θα μπορούσαμε αν το επιθυμούσαμε να το είχαμε ονομάσει $instr 264$ ή όπως αλλιώς επιθυμούμε εκτός όμως από $instr 0$ καθώς αυτό θεωρείται από τη CSound ότι είναι ολόκληρο το αρχείο orchestra. Όπως παρατηρούμε ο ορισμός του οργάνου κλείνει με τη δήλωση $endin$. Μέσα στο block του οργάνου ορίζεται μια μεταβλητή $a1$ η οποία θα πάρει την τιμή που θα επιστρέψει η μονάδα επεξεργασίας $oscil$, με παραμέτρους $p4, p5$, και 1 . Τέλος το όργανό μας θα επιστρέψει την τιμή της $a1$.

Η Γλώσσα Μουσικού

Προγραμματισμού CSound

Για την καλύτερη κατανόηση των όσων ειπώθηκαν, να ένα παράδειγμα αρχείου τύπου score το οποίο ορίζει μια συνάρτηση-πίνακα, την f1. Στη δήλωση αυτή έχουμε τα εξής:

Ένα πίνακα-συνάρτηση (f1)

Ο οποίος θα ενεργοποιηθεί τη στιγμή t=0 sec

Θα διαθέτει 512 στοιχεία

Τέλος, το 10 αποτελεί μια

κωδικοποιημένη λειτουργία της CSound υπαγορεύει ότι στον πίνακα αυτό θα αποθηκευτεί ένα ημιτονοειδές σήμα.

Η επόμενη δήλωση του αρχείου score καλεί το όργανο 1 (i1), να παίξει από τη χρονική στιγμή 0 και ως τη στιγμή 3 (πρόκειται για δευτερόλεπτα). Οι υπόλοιπες 2 παράμετροι ορίζουν τη νότα που θα παίξει το όργανο. Τέλος, κάθε αρχείο score έχει στην τελευταία του γραμμή τη **δήλωση e**.

```
f1 0 512 10 1
i1 0 3 3000 440
e
```

Το εργαλείο Artsbuilder

- Το **Arts** (**A**nalog **R**eal-Time **S**ynthesizer) είναι ένα αρθρωτό σύστημα σύνθεσης ήχου και μουσικής. Η βασική του νοοτροπία είναι η χρήση μικρών μονάδων (τα λεγόμενα **modules**) σαν δομικά στοιχεία για τη δημιουργία πιο σύνθετων μονάδων επεξεργασίας ήχου. Τα διάφορα modules παρέχουν λειτουργίες όπως γεννήτριες ήχου, φίλτρα, εφέ, μίκτες και αναπαραγωγή ήχου από διάφορα formats. Η καρδιά του όλου συστήματος είναι ο δαίμονας `artsd`, ο οποίος αναλαμβάνει να "mix-άρει" τον ήχο που παράγεται από διάφορες πηγές, σε πραγματικό χρόνο. Με τον τρόπο αυτό επιτρέπεται στις διάφορες εφαρμογές ήχου να χρησιμοποιούν το διαθέσιμο hardware κατά τρόπο διαφανή, αφού ο μόνος που έχει πρόσβαση ανά πάσα στιγμή σε αυτό είναι ο `artsd`.
- Ο **Artsbuilder** είναι ένα εργαλείο σύνθεσης με μια τελείως διαφορετική προσέγγιση από αυτή της CSound. Είναι άρρηκτα συνδεδεμένο με το σύστημα Arts του οποίου αποτελεί αναπόσπαστο κομμάτι, και χωρίς αυτό δεν είναι δυνατό να υπάρξει. Απαραίτητη προϋπόθεση για να εκτελέσει κανείς τον Artsbuilder είναι να εκτελείται την ίδια στιγμή και ο `Atrsd`.

Το εργαλείο Artsbuilder

- Το όλο σύστημα ακολουθεί την **αρχιτεκτονική πελάτη-εξυπηρετητή (c/s)**. Στην ουσία, ο artsd είναι ο server που διαχειρίζεται το hardware και χρησιμοποιώντας αυτό είναι σε θέση να παρέχει υπηρεσίες στους πελάτες. Οι clients δεν είναι τίποτα άλλο από τις διάφορες εφαρμογές ήχου. Όλη αυτή η διαδικασία λαμβάνει χώρα με τη χρήση του πρωτοκόλλου MCOP (Multimedia Communication Protocol) χάρη στο οποίο, επιτυγχάνεται κατ' αρχή διαφάνεια ως προς τη χρήση του υλικού και την παραγωγή του ήχου καθώς και η ασφάλεια/πιστοποίηση επιτρέποντας τη χρήση του υλικού μόνο σε διεργασίες και εφαρμογές που πρέπει. Τέλος, λόγω ακριβώς του τρόπου με τον οποίο σχεδιάστηκε το σύστημα επιτυγχάνεται μιας μορφής ανεξαρτησία από το υλικό καθώς ο μόνος άμεσα εμπλεκόμενος με αυτό είναι ο artsd, συνεπώς τυχόν αλλαγές στο υλικό αφήνουν παντελώς αδιάφορες όλες τις επιμέρους εφαρμογές που χρησιμοποιούν το σύστημα arts.

Το εργαλείο Artsbuilder

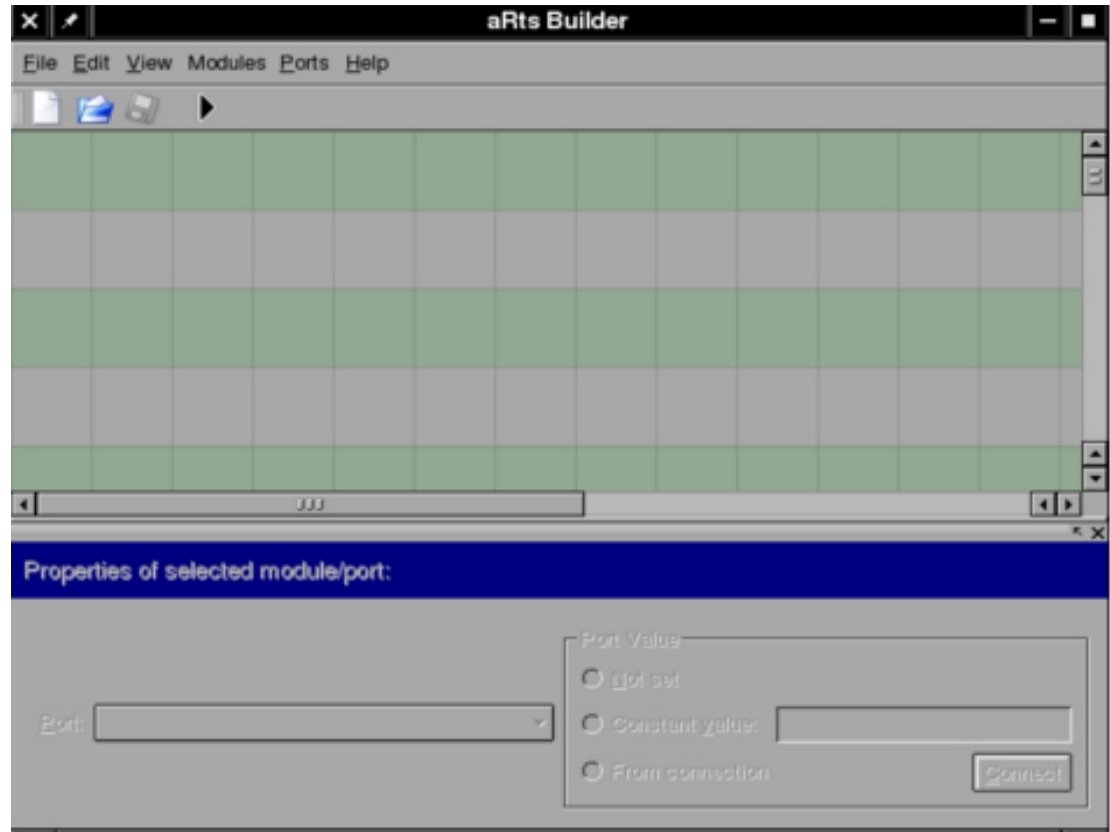
- Το **γραφικό περιβάλλον του Artsbuilder** χωρίζεται σε **τρεις περιοχές**. Στο επάνω μέρος εντοπίζουμε την κλασική μπάρα με τα **menu**, μέσω των οποίων ο χρήστης αλληλοεπιδρά με την εφαρμογή. Κάτω από το μενού συναντάμε το **χώρο εργασίας (workspace)**, μέσα στον οποίο ο χρήστης εισάγει τα modules της επιλογής του. Τέλος στο κάτω μέρος του παραθύρου έχουμε την **"περιοχή ιδιοτήτων" (Properties of selected Module/Port)** στην οποία ο χρήστης έχει τη δυνατότητα τόσο να δει τις τιμές των ιδιοτήτων των διαφόρων modules όσο και να τις αλλάξει.

Το εργαλείο Artsbuilder

- Το γραφικό περιβάλλον του Artsbuilder χωρίζεται σε τρεις περιοχές. Στο επάνω μέρος εντοπίζουμε την κλασική μπάρα με τα **menu**, μέσω των οποίων ο χρήστης αλληλοεπιδρά με την εφαρμογή. Κάτω από το μενού συναντάμε το **χώρο εργασίας (workspace)**, μέσα στον οποίο ο χρήστης εισάγει τα modules της επιλογής του. Τέλος στο κάτω μέρος του παραθύρου έχουμε την "**περιοχή ιδιοτήτων (Properties of selected Module/Port)**" στην οποία ο χρήστης έχει τη δυνατότητα τόσο να δει τις τιμές των ιδιοτήτων των διαφόρων modules όσο και να τις αλλάξει.

Το εργαλείο Artsbuilder

Σε γενικές γραμμές ο Artsbuilder είναι ένα ιδιαίτερα εύχρηστο και απλό εργαλείο με πολλές δυνατότητες. Η μόνη δυσκολία που μπορεί να συναντήσει κανείς στην εκμάθηση και χρήση του έγκειται στη μη πλήρη τεκμηρίωση (documentation), ιδιαίτερα σε ότι αφορά τα διαθέσιμα modules και στο ποια ακριβώς λειτουργία καθένα από αυτά επιτελεί.



Η γλώσσα MAX/MSP

Η **Max** είναι μια Real - time αντικειμενοστρεφής γλώσσα προγραμματισμού, βασισμένη στα γραφικά. Οι όροι αυτοί αναλυτικότερα σημαίνουν:

- **Real-time** : Το πρόγραμμα το οποίο γράφουμε εκτελείται αμέσως, χωρίς να χρειαστεί να το περάσουμε από το ξεχωριστό στάδιο του compile
- **Αντικειμενοστρεφής:** Η Max χρησιμοποιεί προκαθορισμένα αντικείμενα (modules) με συγκεκριμένη συμπεριφορά, inputs και outputs, τα οποία συνδέονται από το χρήστη για να επιτελέσουν μια μεγαλύτερη λειτουργία.
- **Γλώσσα προγραμματισμού :** Με την έννοια ότι δεν είναι μια εφαρμογή χρήστη αλλά ένα ισχυρό προγραμματιστικό εργαλείο για τη δημιουργία των δικών μας «εικονικών μηχανών».
- **Βασισμένη στα γραφικά :** Τα αντικείμενα στην Max είναι γραφικά σχήματα τα οποία μετακινούμε στην οθόνη και συνδέουμε μεταξύ τους με εικονικά καλώδια.

Η γλώσσα MAX/MSP

Η Max αποτελεί το βασικό περιβάλλον το οποίο περιλαμβάνει πάνω από 250 αντικείμενα για MIDI, έλεγχο, διεπαφές χρήστη και συγχρονισμό. Αρχικά σχεδιάστηκε για να αποτελέσει ένα interface ελέγχου για άλλες εξωτερικές συσκευές (hardware synthesizers, samplers κ.λ.π.) χρησιμοποιώντας το MIDI ή κάποιο άλλο πρωτόκολλο.

Το 1997 προστέθηκε ένα σύνολο νέων αντικειμένων τα οποία είχαν τη δυνατότητα να χειρίζονται ψηφιακά σήματα ήχου, το οποίο ονομάστηκε MSP (Max Signal Processing). Έτσι ο χρήστης με τα 150 και πλέον καινούρια αντικείμενα μπορεί να κατασκευάσει τα δικά του ψηφιακά όργανα (software synthesizers, effects, filters κ.λ.π.) και να τα ενσωματώσει στα Max προγράμματα.

■ Τα **αντικείμενα στην Max/MSP** είναι ορθογώνια με τα εξής χαρακτηριστικά :

- Όνομα του αντικειμένου
- Είσοδοι για την πληροφορία που μπαίνει στο αντικείμενο
- Έξοδοι για την πληροφορία που βγαίνει από το αντικείμενο
- Resize handle για την αλλαγή μεγέθους

[Η γλώσσα MAX/MSP]

Αντικείμενα της γλώσσας MAX:

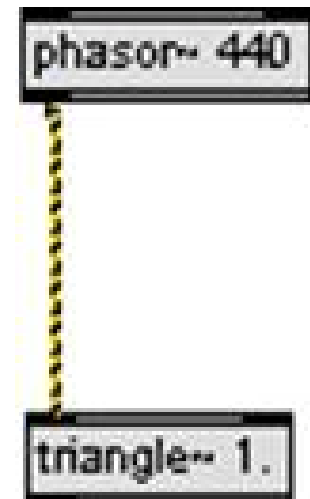


Το πρώτο είναι μετρονόμος, ενώ τα υπόλοιπα δύο είναι γεννήτριες και φίλτρα συνδεδεμένα με εικονικά καλώδια.

Η γλώσσα MAX/MSP

Διαφορές MAX και MSP αντικειμένων:

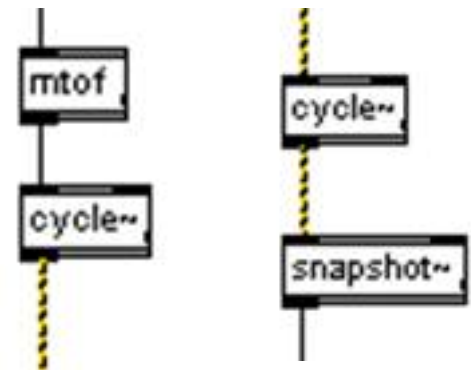
- Τα αντικείμενα της Max σχηματίζουν ένα «**μονοπάτι**» **μηνυμάτων (message path)** και επικοινωνούν μεταξύ τους στέλνοντας μηνύματα σε συγκεκριμένες χρονικές στιγμές. Π.χ. απόκριση σε γεγονός χρήστη (πάτημα νότας σε MIDI controller) ή σε προκαθορισμένο γεγονός (μετρονόμος). Σε αντίθεση, τα MSP αντικείμενα συνδεόμενα μεταξύ τους εγκαθιστούν μια σχέση μεταξύ των αντικειμένων η οποία χρησιμοποιείται για τον υπολογισμό της ηχητικής πληροφορίας σε κάθε χρονική στιγμή δημιουργώντας έτσι ένα δίκτυο σημάτων (signal network). Η διαφορά ανάμεσα στο message path και το signal network φαίνεται και γραφικά στον τρόπο με τον οποίο συνδέονται τα αντικείμενα. Στο μεν message path το εικονικό καλώδιο είναι μαύρο και λεπτό, ενώ στο signal network παρουσιάζεται παχύτερο και κιτρινόμαυρο



Η γλώσσα MAX/MSP

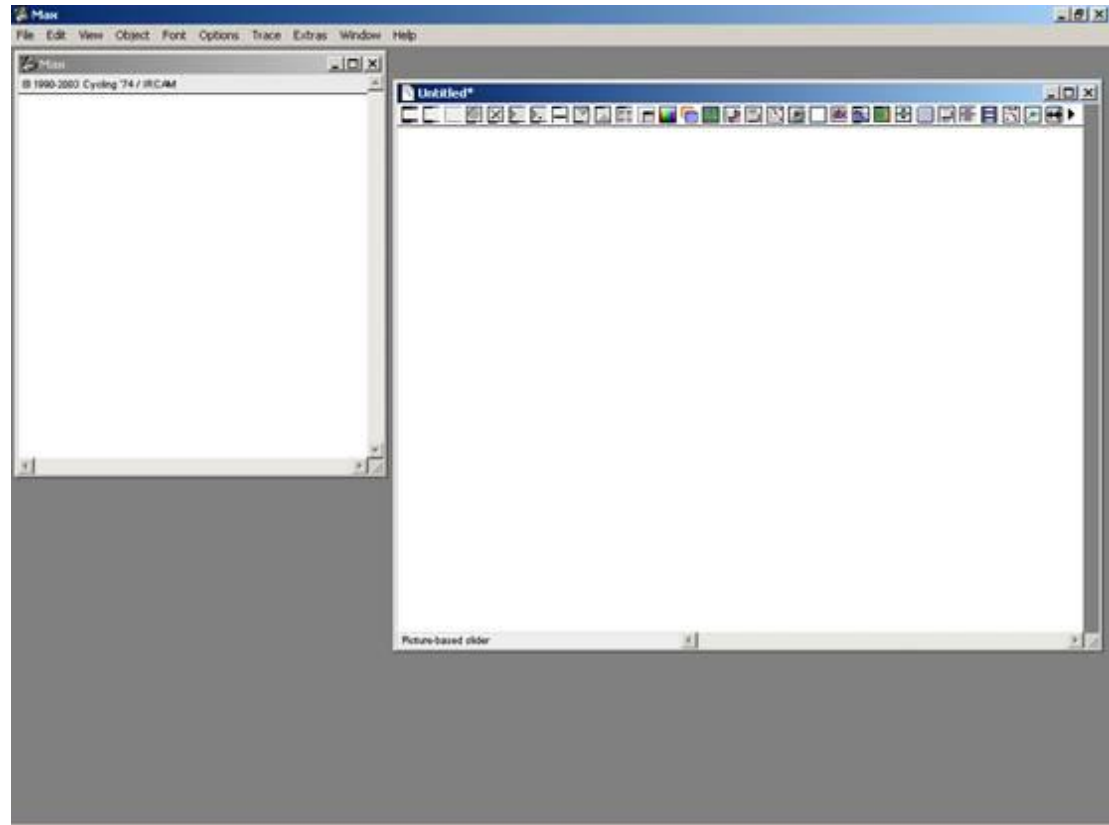
Διαφορές MAX και MSP αντικειμένων:

- Η μικρότερη τιμή για το προγραμματισμό γεγονότων στην Max είναι το millisecond, το οποίο είναι αρκετά γρήγορο για οποιοδήποτε τύπο ελέγχου συσκευής (control rate). Τα MSP αντικείμενα λόγω του ότι παράγουν ηχητικό σήμα πρέπει να επεξεργάζονται τα δεδομένα τους ταχύτερα, συνήθως 44100 φορές το δευτερόλεπτο (audio rate).
- Λόγω της διαφοράς των αντικειμένων αυτών, υπάρχουν κάποια MSP αντικείμενα τα οποία παρέχουν ένα σύνδεσμο στους δύο τύπους αντικειμένων. Τα αντικείμενα αυτά παίρνουν Max μηνύματα στην είσοδό τους και οι έξοδοί τους συνδέονται στο signal network ή αντίστροφα. Έχουμε δηλαδή μετάφραση μεταξύ του control και audio rate. Έτσι μπορούμε να επιτύχουμε τον άμεσο έλεγχο (Max objects) πάνω στο signal network (MSP objects).



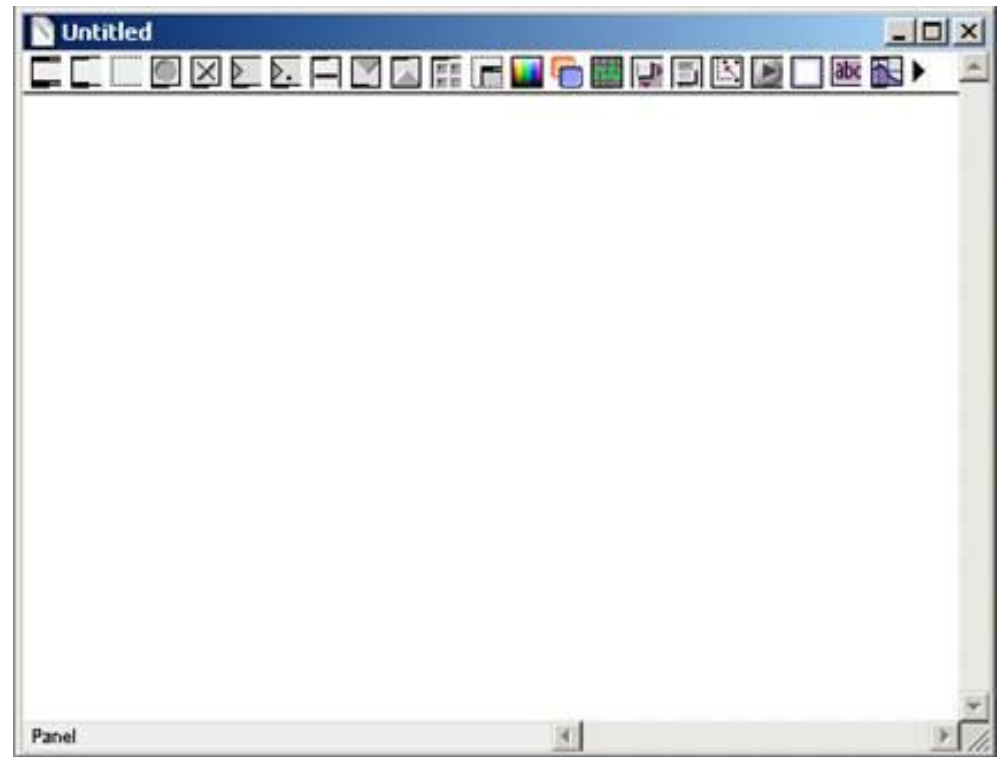
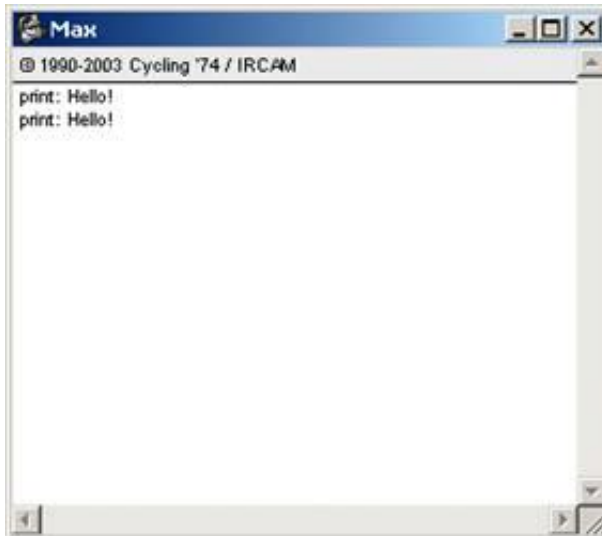
Η γλώσσα MAX/MSP

Το περιβάλλον εργασίας ξεκινά με μία εικόνα σαν την παρακάτω



Η γλώσσα MAX/MSP

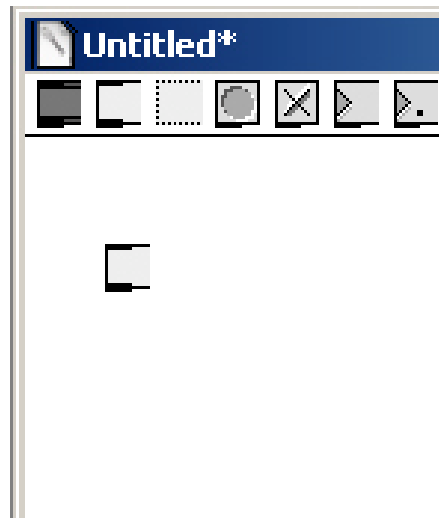
Ξεχωρίζουμε στα αριστερά το Max Window, το οποίο είναι ο τρόπος επικοινωνίας της Max με το χρήστη, είτε εμφανίζοντας μηνύματα της Max (errors κ.λ.π.), είτε αποτελέσματα του κώδικα που έχουμε γράψει. Όλος ο προγραμματισμός λαμβάνει χώρα στο λεγόμενο Patcher Window. Στην κορυφή του παραθύρου υπάρχει το Icon-bar το οποίο παρουσιάζει στο χρήστη με γραφικό τρόπο όλα τα διαθέσιμα αντικείμενα της Max/MSP.



Η γλώσσα MAX/MSP



Τα εικονίδια της Max/MSP είναι γραφικές αναπαραστάσεις προγραμματιστικών ενεργειών. Κάθε αντικείμενο εκτελεί μια ή περισσότερες συγκεκριμένες εργασίες. Η δημιουργία ενός προγράμματος από το χρήστη λειτουργεί περίπου όπως μια σχεδιαστική εφαρμογή, όπου ο χρήστης επιλέγει ένα αντικείμενο από το Icon-bar και το τοποθετεί στο σημείο του Patcher Window που επιθυμεί.



Η γλώσσα MAX/MSP

Υπάρχουν δύο βασικοί τύποι αντικειμένων :

- **Γραφικά ή διεπαφής χρήστη** : όπως slider, dial, number box κ.λ.π.,
- **Object Box** : τα οποία είναι τα πιο συνηθισμένα στην Max/MSP. Η εργασία την οποία εκτελεί ένα object box εξαρτάται από το όνομά του. Ο χρήστης μπορεί είτε να πληκτρολογήσει κατευθείαν το όνομα του αντικειμένου, είτε να επιλέξει από μια κατηγοριοποιημένη λίστα αντικειμένων (Object List).

ο χρήστης επιλέγει ένα αντικείμενο από το Icon-bar και το τοποθετεί στο σημείο του Patcher Window που επιθυμεί.



Από αριστερά: Dial Object Box Object List



Η γλώσσα MAX/MSP

Υπάρχουν δύο βασικοί τύποι αντικειμένων :

- **Γραφικά ή διεπαφής χρήστη** : όπως slider, dial, number box κ.λ.π.,
- **Object Box** : τα οποία είναι τα πιο συνηθισμένα στην Max/MSP. Η εργασία την οποία εκτελεί ένα object box εξαρτάται από το όνομά του. Ο χρήστης μπορεί είτε να πληκτρολογήσει κατευθείαν το όνομα του αντικειμένου, είτε να επιλέξει από μια κατηγοριοποιημένη λίστα αντικειμένων (Object List).

ο χρήστης επιλέγει ένα αντικείμενο από το Icon-bar και το τοποθετεί στο σημείο του Patcher Window που επιθυμεί.



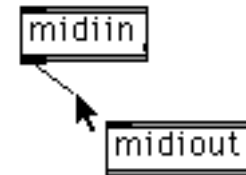
Από αριστερά: Dial Object Box Object List



Η γλώσσα MAX/MSP

Υπάρχουν **δύο** βασικοί **τύποι αντικειμένων** :

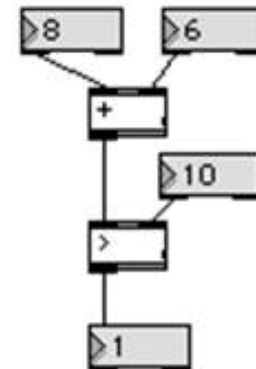
- Η σύνδεση των αντικειμένων γίνεται με τα **εικονικά καλώδια (patch-cords)** παρόμοια με τον τρόπο που συνδέονταν τα παλιά αναλογικά synthesizers. Για τη δημιουργία ενός patch-cord ο χρήστης κάνει click & drag από το outlet ενός αντικειμένου προς το inlet ενός άλλου αντικειμένου. Μετά τη δημιουργία του patch-cord, οποτεδήποτε το πρώτο αντικείμενο στέλνει δεδομένα στην έξοδό του το δεύτερο αντικείμενο τα δέχεται στην είσοδό του.



- Τα προγράμματα κατασκευάζονται δημιουργώντας ομάδες από συνδεδεμένα αντικείμενα, τα οποία παίρνουν πληροφορία, την επεξεργάζονται και στέλνουν τα αποτελέσματα παρακάτω. Το μοντέλο αυτό λέγεται **dataflow**.

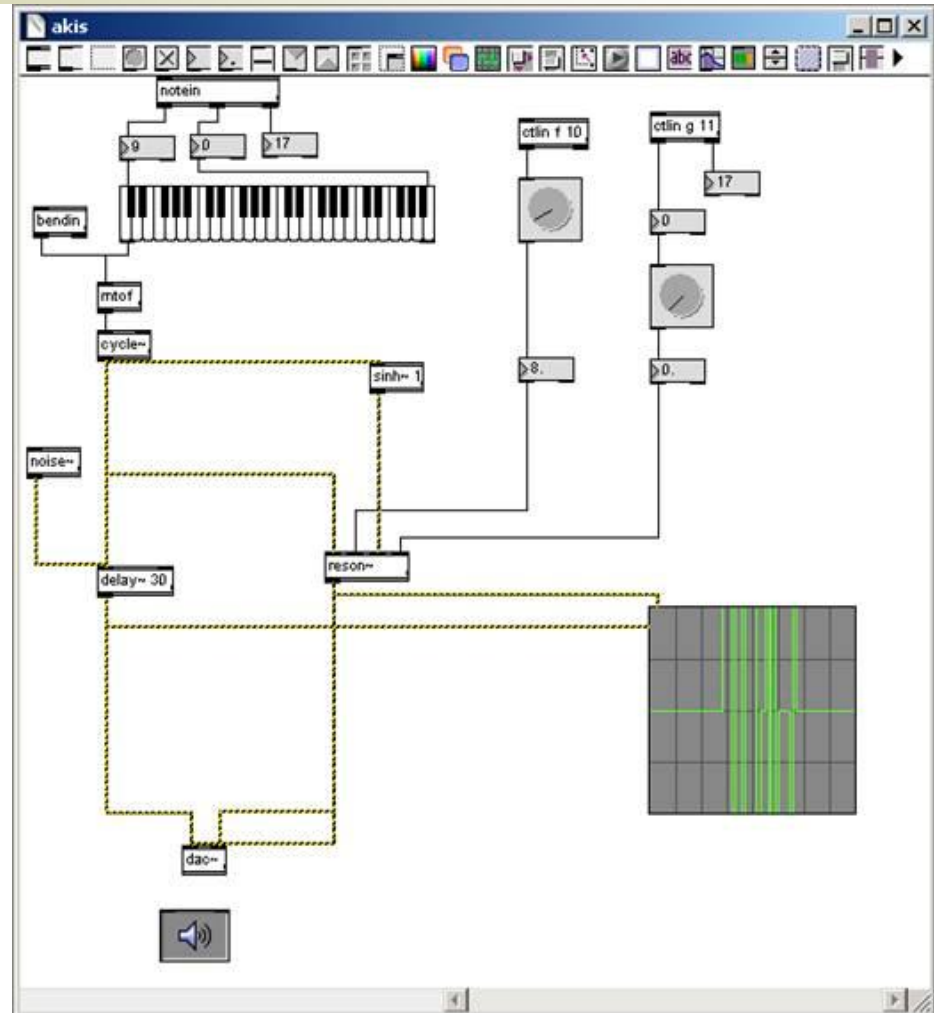
Στο παρακάτω patch η Max προσθέει 2 αριθμούς (8 και 6) και ακολούθως τους συγκρίνει με έναν άλλο αριθμό (10).

Η έξοδος του προγράμματος είναι 1 (true).



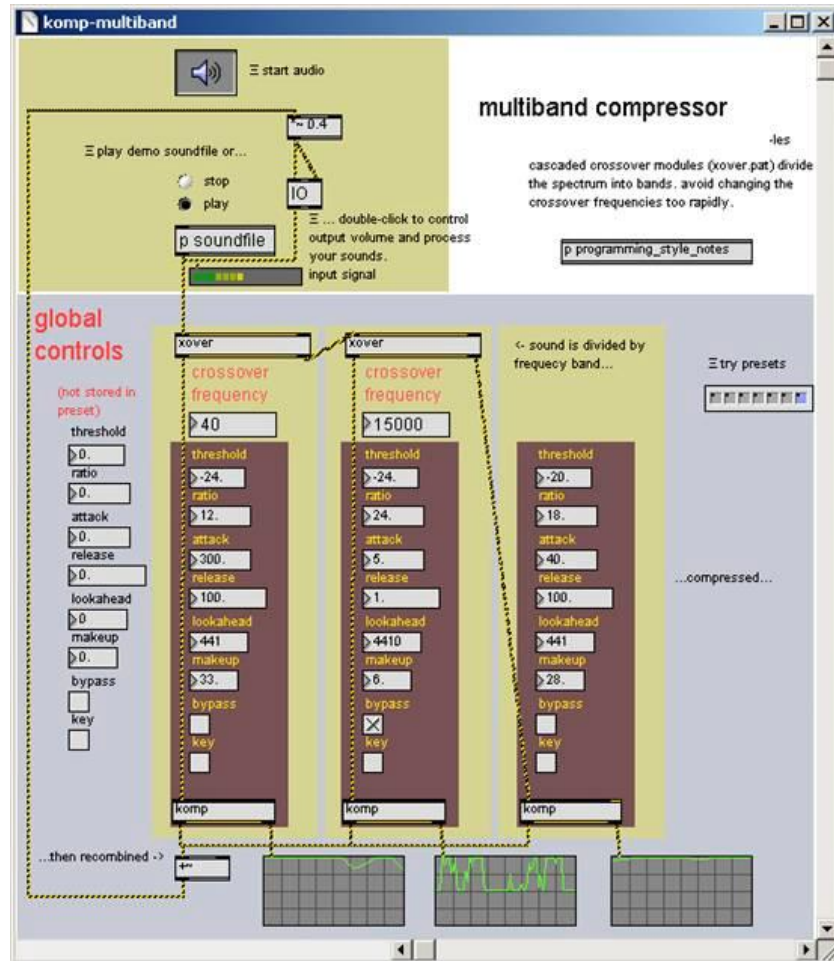
Η γλώσσα MAX/MSP

Η εικόνα μας παρουσιάζει ένα απλό **software synthesizer**. Στο επάνω τμήμα του βρίσκονται τα Max αντικείμενα τα οποία παίρνουν δεδομένα από MIDI controller (pitch κ.α.). Στη συνέχεια αυτά τα μηνύματα περνούν στα MSP αντικείμενα τα οποία συνθέτουν και αναπαράγουν τον ήχο. Έχει ενδιαφέρον να παρακολουθήσουμε το message path που σχηματίζουν τα Max αντικείμενα (λεπτή μαύρη γραμμή) και το signal network που σχηματίζουν τα αντίστοιχα MSP (κιτρινόμαυρη παχιά γραμμή).



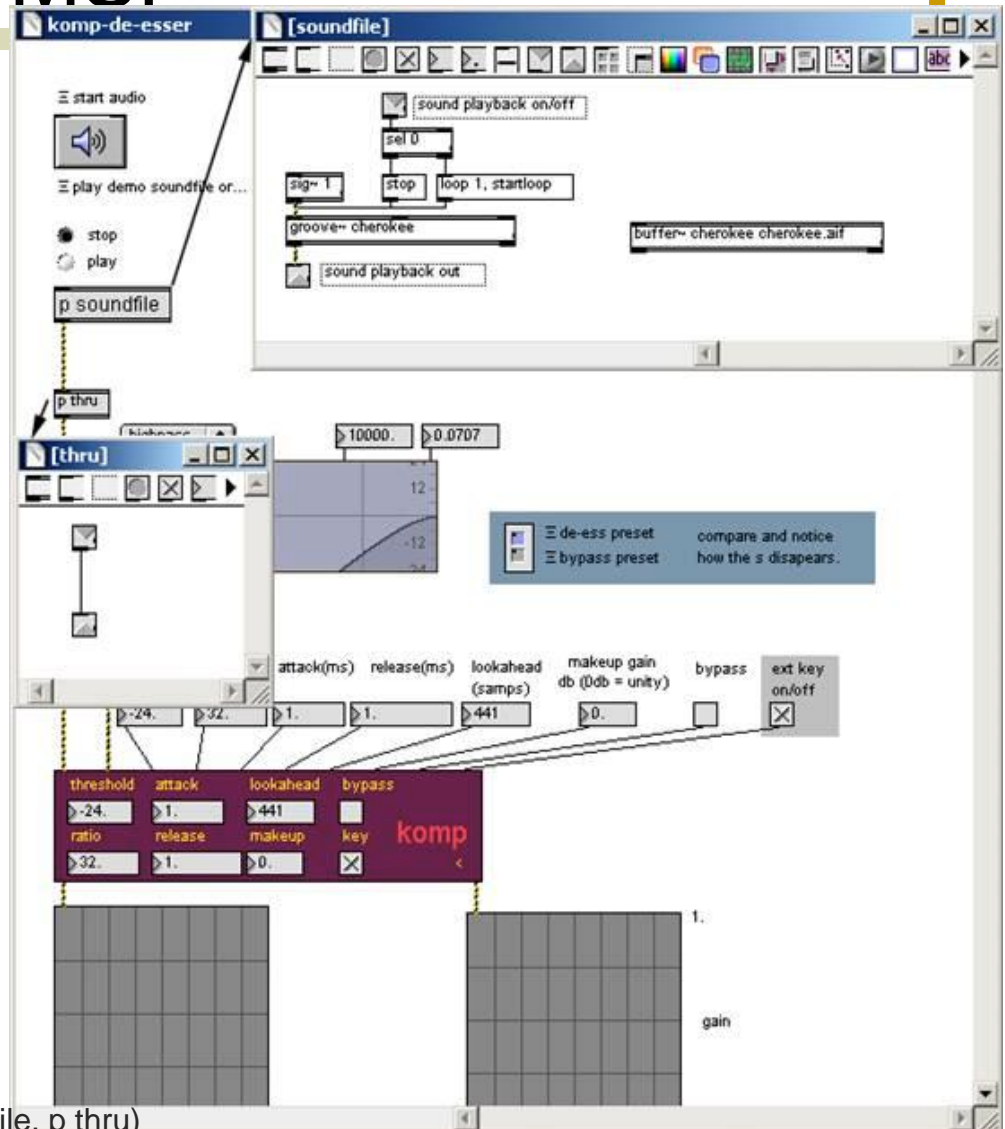
Η γλώσσα MAX/MSP

Ημπορούν να κατασκευαστούν βέβαια και πιο πολύπλοκα αντικείμενα: φαίνονται οι δυνατότητες της Max/MSP για το χτίσιμο μιας πιο φιλικής προς το χρήστη διεπαφής με ομαδοποίηση των αντικειμένων σε πλαίσια και σχόλια για να γίνει κατανοητή η λειτουργία του. Έτσι δημιουργούνται προγράμματα Max/MSP με ανεπτυγμένο Interface



Η γλώσσα MAX/MSP

Ένα άλλο ευέλικτο χαρακτηριστικό είναι η **οργάνωση ενός μεγάλου patch σε μικρότερα sub-patches**. Έτσι τμήματα του κώδικα που είναι μεγάλα σε μέγεθος ή ξεχωριστά από τα υπόλοιπα τμήματα μπορούν να ομαδοποιηθούν. Το sub-patch εμφανίζεται ως ένα απλό αντικείμενο με το γράμμα "p" και το όνομά του. Το μόνο που χρειάζεται να ξέρουμε είναι ο αριθμός κι ο τύπος των ορισμάτων καθώς και ο αριθμός και ο τύπος των αποτελεσμάτων, όμοια με μια συνάρτηση δομημένου προγραμματισμού. Μπορούμε να έχουμε εμφωλευμένα sub-patches σε οποιοδήποτε βάθος. Για να δούμε τα περιεχόμενα του sub-patch απλώς κάνουμε διπλό κλικ στο αντικείμενο.



Τέλος 8ης Διάλεξης



Πηγές και βοηθήματα:

- Ηλεκτρονικές σημειώσεις Δ. Πολίτη 2015 @ <http://elearning.auth.gr>
- Δ. Πολίτη, Γλώσσες και Διεπαφές στη Μουσική Πληροφορική Εκδόσεις Κλειδάριθμος, Αθήνα 2007