

**Komposition i Max / MSP / Jitter**  
**FoU 09 - 10**  
**Anders P Jensen**

## Historik

Max er et visuelt programmeringssprog for lyd og billede. Udviklingen af Max startede i midt firserne. Programmet som på dette tidspunkt hed "Patcher" blev udviklet af Miller Puckette til IRCAM, Paris (Institut de Recherche et Coordination Acoustique/Musique ). Programmet blev skrevet for Macintosh med henblik på at IRCAM tilknyttede komponister kunne programmere og udvikle interaktiv computer musik. Populært kaldet elektro-akustisk musik med udspring i komponister som Karlheinz Stockhausen, Pierre Schaeffer m.fl.

I slutfirserne blev programmet solgt på licens til Opcode Systems som under ledelse af David Zicarelli udviklede en kommerciel version af Max. Opcode mistede efter nogle år interessen for Max og Zicarelli's eget firma "Cycling '74" overtog agenturet i slut halvfemserne.

På nogenlunde samme tid designede førnævnte Miller Puckette programmet "Pd" (Pure Data). Et gratis open-source program som minder meget om Max. Pd arbejder bl.a. med digital audio processing i real-time, noget som ikke tidligere havde været muligt. De første IRCAM udgaver styrede alene eksterne synthesizere/samplere via midi. Men efterhånden var udviklingen nået så langt at det var muligt at arbejde med real-time digital audio processing.

Zicarelli indbyggede dele af Pd's software i Max - særligt audio delen - og kaldte nu programmet for Max/MSP.

I slutningen af halvfemserne udviklede flere firmaer real-time videoredigeringssoftware som kunne anvendes sammen med Max. I 2003 udviklede Cycling '74 deres egen videoredigerings software til Max kaldet "Jitter".

Derfor hedder programmet i dag Max/MSP/Jitter.

## Anvendelse

Lidt forenklet sagt så kan man sige at programmet varetager tre opgaver:

Max arbejder med midi

MSP arbejder med audio

Jitter arbejder med billeder

Man bygger et patch - en konstruktion - ved at sammensætte objekter med "kabler".

Et objekt er i sig selv et lille program. Disse objekters funktion og anvendelsesmuligheder er mangfoldige og ofte ret komplicerede. Der findes ca. 600 forskellige objekter i programmet.



Ovenstående figur viser en af de simpleste konstruktioner i Max.

Et multiplikationsobjekt er anvendt. Indtastet tal i øverste kasse ganges med 2 og resultatet vises i nederste kasse.

Et patch kunne eks. være en effekt der behandler audio - eks en delayeffekt.

Det kunne også være en avanceret sequencer med helt specielle controllers for midi-beskeder. Mulighederne er utallige.

Styrken ved dette program ligger i at man kan konstruere egne effekter, sequencere, eller ligefrem bygge helt nye instrumenter.

## Processen

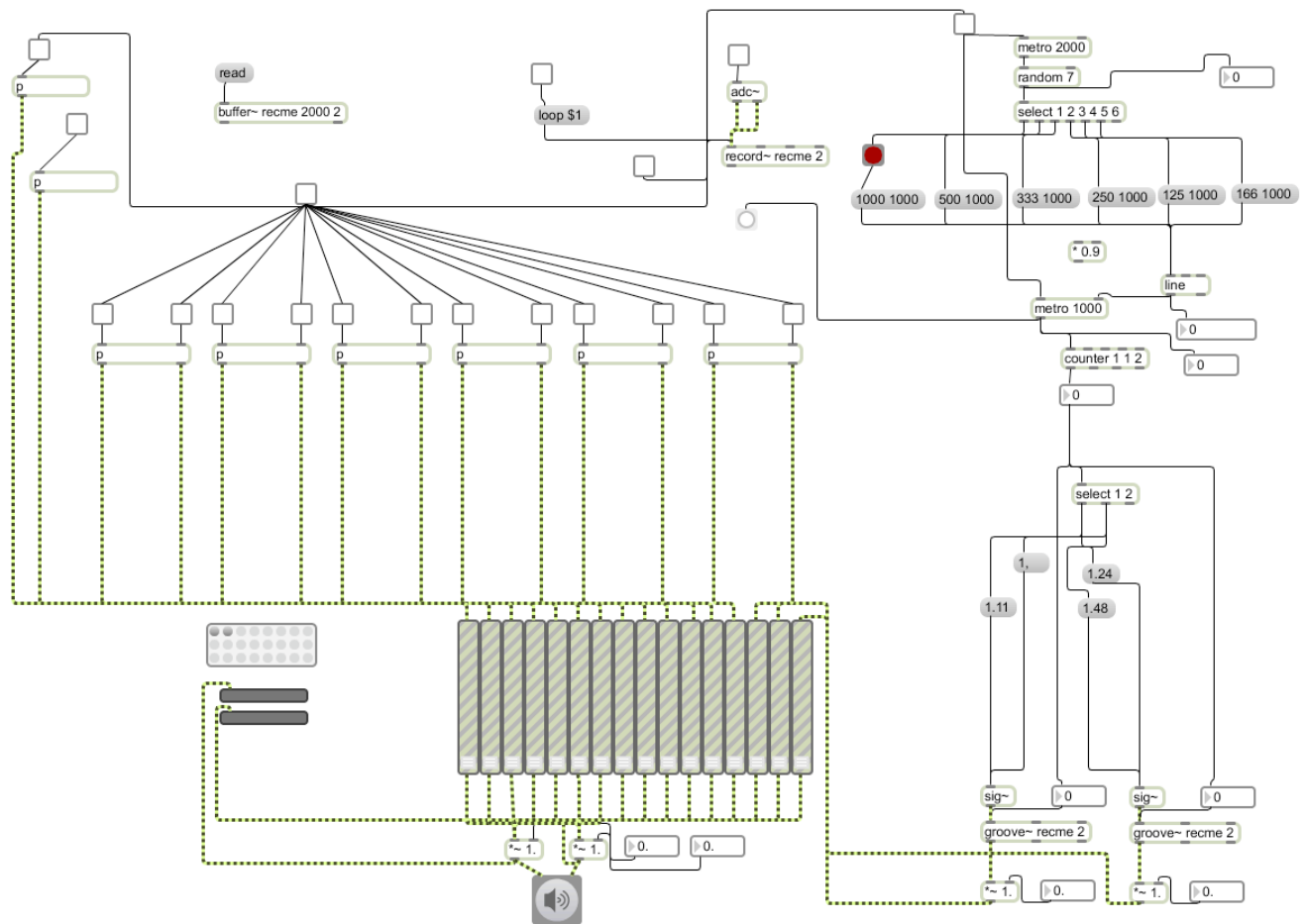
Hvor at man i de fleste andre musikprogrammer ofte kan "føle sig frem" uden at læse for meget i manualen, så er Max væsentligt mere kompliceret og derfor er det helt nødvendigt at tilegne sig en stor portion grundviden om objekterne og deres indbyrdes relation, før at en decideret konstruktion af patches kan forekomme.

Hvis denne del af fasen synes lidt tung - og nærmest minder om at lære et programmeringssprog - så er glæden så meget desto større, når at den egentlige konstruktion begynder.

Jeg har efterhånden bygget en del patches.

Nedenstående eksempel byggede på at real-time audio skulle loopes på alle mulige måder. Her brugte jeg meget tid på at gøre resultatet organisk og menneskeligt v.h.a. objekter som "random" og "drunk".

Som navnene antyder så er det objekter som indeholder en form for tilfældige valg.



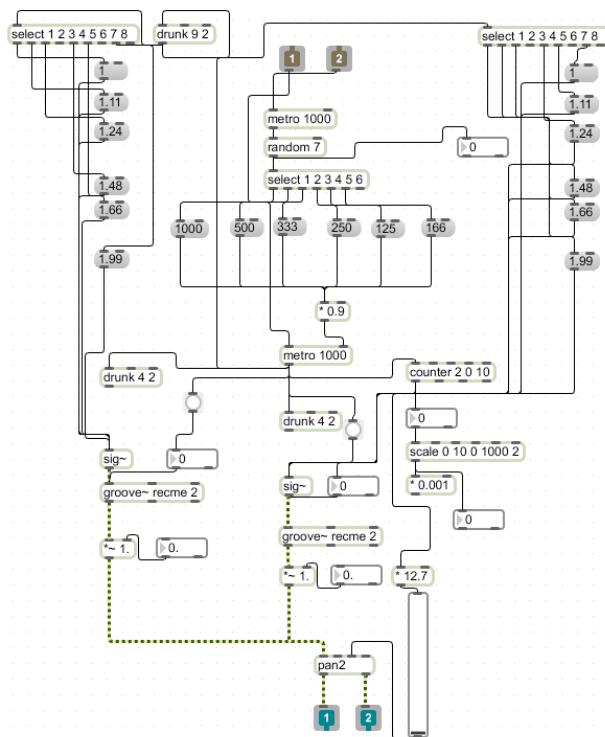
Real-time indspillet audio som bliver loopet/trigget i forskellige skiftende pitch og tempi. Alle de objekter med navnet "P" indeholder nogle subpatches - dette for at gøre konstruktionen overskuelig.

### Musikeksempel

På medfølgende CD (track 02) høres en demonstration af denne patch.

Der synges een konstant tone i real-time og programmet sørger for resten.

Nedenstående billede viser ét af disse subpatches. De er i øvrigt alle forskellige.



Min oprindelige tanke var at bygge et patch som nærmest kunne lave selv det mest ordinære audio-input om til noget interessant. Og det lykkedes faktisk også, men hurtigt stod det klart at det er interaktionen mellem computer og menneske der er inspirerende at høre på. Særligt med henblik på progressionen i det enkelte stykke musik. Elektronisk musik kan nogle gange have tendens til at blive stillestående. Derfor at interaktionen over tid efter min mening essentiel.

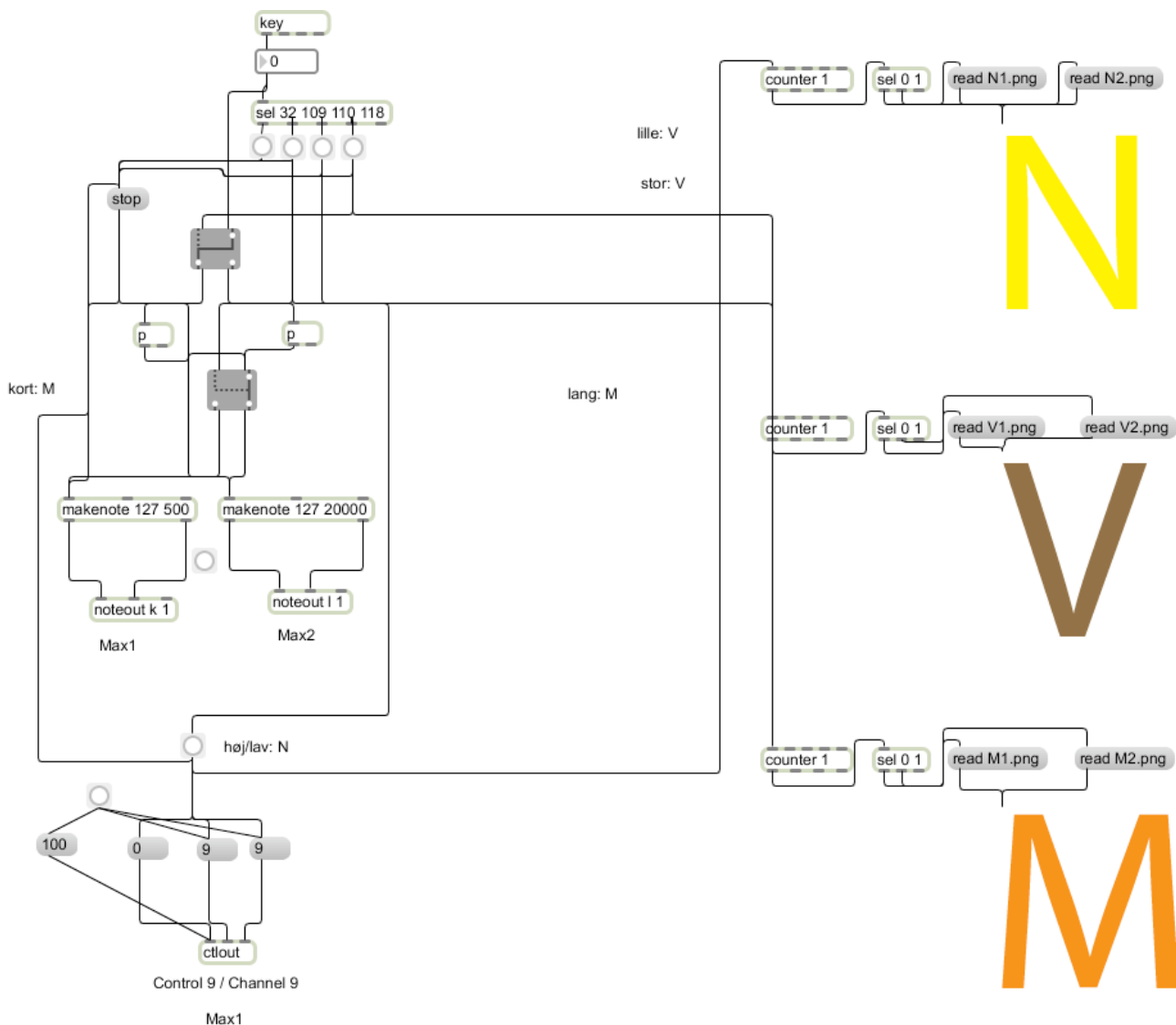
I stedet for at bygge løs - den ene mere komplicerede konstruktion efter den anden - prøvede jeg at forestille mig hvad der ville være interessant at bygge som *ikke* fandtes i forvejen. Hvis man tænker nærmere over det, så er mange af de effekter vi bruger og genanvender i nye maskiner, baseret på eksperimenter der ligger helt tilbage i 60'erne. For mig virker det irrelevant at rekonstruere eks. en delaymaskine eller en subtraktiv synthesizer. Selvfølgelig kan man udvikle noget helt nyt ud af disse to velkendte områder, men det virkede bare mere inspirerende at viske tavlen helt ren.

Jeg valgte derfor at bygge et instrument som i sin fysiske anvendelse er baseret på computertastaturet. Tasterne styrer diverse midinotes som er designet i forskellige intervalpar som sendes videre til en softsampler - i dette tilfælde Logic's EXS24. Nogle taster styrer tonelængde og dynamik samt ændring af intervalstørrelser.

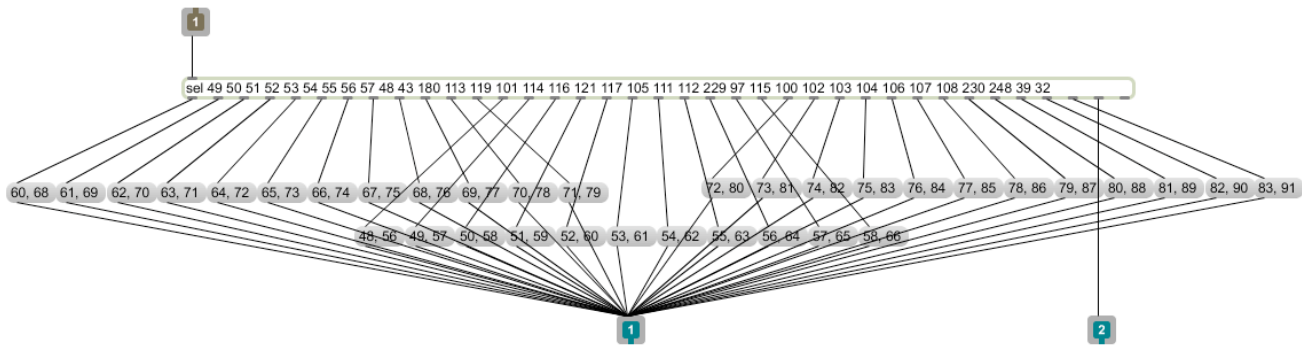
Jeg sørgede bevidst for at designe fordelingen af intervaller/taster således at strukturen ikke mindede for meget om eksisterende musiktasteinstrumenter - eks. klaver og harmonika. Dette for at bibeholde en form for friskhed og spontanitet i tilgangen til instrumentet.

Da instrumentet genererer ret komplicerede clusterlignende klange, valgte jeg at anvende en "ren" lyd. Jeg samlede derfor en triangel waveform fra min gamle analoge Korg 700S, som efter min mening har en særlig "dyb" grundlyd. D.v.s. at jeg styrede en softsampler i Logic via Max fra mit computertastatur.

Nedenstående viser hovedkonstruktionen. Jittergrafik i form af store bogstaver der skifter farve når disse taster aktiveres. N styrer intervalstørrelser, V styrer skift mellem to EXS24 samplere med h.h.v. kort og lang lyd m.v. Og endelig M styrer via midi control automation i Logic et filter bypass.



Her ses et af de tilhørende subpatches, med diverse intervalfordelinger. Musikstykket på cd'en er klippet sammen af flere forskellige patches - med forskellige intervalstørrelser og kombinationer.



Trods at disse grafiske fremstillinger for ikke Max brugere kan virke komplicerede, så er det et af de mindre komplicerede patches jeg hidtil har bygget.

At få et computertastatur til at sende en række intervaller via midi til at et sequencer program med en indbygget sampler som har sin lyd fra en analog synthesizer.

Ikke desto mindre er dette patch efter min mening pt det mest musikalske.

Trods at der har været mange forhindringer undervejs i konstruktionen af dette patch, så var det mest tidskrævende at få selve idéen til konstruktionen.

Kompositorisk er det noget af en omvæltning at arbejde med musik på denne vis. Ofte har man konstant "lyd i øret" når musik skabes. Enten via et musikinstrument eller via forprogrammeret musiksoftware. Her var der lange perioder af stilhed, når noget skulle designes eller programmeres. I starten kunne man føle sig mere som programmør end som komponist, men i takt med at kendskabet til objekterne voksede, begyndte musikken også at klinge for det indre øre. Lidt i stil med partiturskrivning. Så på en måde kan man sige at kompleksiteten i programmet tvinger det indre øre til at arbejde på højtryk. Og netop at forestillingsevnen og opfindsomheden kommer på arbejde, ser jeg som en klar fordel rent kompositorisk. I takt med at der udvikles eks. notationsprogrammer med indbyggede orkestersamples og vi derved med det samme får en klang præsenteret, daler den indre forestillingsevne. Jeg taler ikke for at blyant og papir skal tilbage, men blot at man skal være bevidst om at de fleste værktøjer ofte tilskynder en særlig arbejdsform som kan blokere den enkeltes frie forestillingsevne.

## Resultatet

Kan høres på den medfølgende CD track 01

Jeg får en oplevelse af stoflighed på samme måde som når man hører et akustisk instrument. Dette var et af mine mål. Ligeledes at værket udvikler sig fra A til B uden nogen gentagelse. Som lytter bliver jeg udfordret til selv at danne en sammenhæng både i følelse og forløb. Balancen mellem at opleve en på forhånd dikteret oplevelse og samtidig bevare en åbenhed for fortolkning synes jeg fungerer.

Instrumentet klinger nærmest som en Theremin eller måske snarere en Ondes Martenot. Et elektronisk instrument fra 20'erne som bl.a. Messiaen arrangerede for. Den kompositoriske proces peger ligeledes i retning af det 20. århundredes moderne klassiske komponister som ofte arbejder ud fra nogle selvdefinerede kompositionsprincipper med henblik på at finde nye ubetradte musikterritorier.

Netop kombinationen af den dogmatiske grundidé - i dette tilfælde konstruktionen af et nyt instrument - og den efterfølgende menneskelige interaktion finder jeg meget inspirerende. At man har en slags forskningsfase efterfulgt af forsøg der beror på intuition virker fyldestgørende for mig.

Max er et glimrende værktøj til at netop denne arbejdsproces grundet programmets førromtalte frie udviklingsplatform.

## Undervisning

Jeg fornemmer en stigende interesse for denne software på RMC, og jeg har allerede givet timer i denne software på h.h.v. M og S linien. Et decideret undervisningsforløb er oplagt. Med den forventede indførsel af tværfagelige kurser vil jeg tilbyde kurser indenfor denne software. Komplexiteten i programmet gør at at kurser formodentlig skal være casebaseret med en forholdsvis stram progression og målsætning. At studerende lærer få objekter som til gengæld har vidtgående designmuligheder.

AJ/14.02.11

## Bilag

MusikCD

Track 01 - Den færdige komposition.

Track 02 - Demo af real-time audioprocessing, hvor at der kun synges een konstant tone.

© NCB. CD'en er kun til intern dokumentation og må ikke kopieres eller uploades til web m.v.