

Undichte Stellen stopfen.

4:00 min 4-kanal (2003)

Das Ausgangsmaterial für dieses Stück sind zwei Samples mit Flugzeuggeräuschen. Ein in Lisp geschriebener Algorithmus definiert linear die Werte der Einsatzabstände, Dauern, Sampletranspositionen und den Mittelfrequenzen 1 und 2 des Filters, die mit die Samples einen Prozess einleiten und ausformulieren.

Im Orc-File werden die Ereignisse mit eine Hüllkurve multipliziert (0 bis 1 in der Mitte, zurück nach 0). Die Mittelfrequenzen haben die Funktion eine Filterbank im Verhältnisse Grundton $\cdot(1+0.1\cdot 2^n)$ zu steuern. Diese bewegen sich innerhalb des Samples von 1000 bis 1100 Hz, zu 1100 bis 900 Hz am Ende. Die Bandbreite ist im Orc-File (samples-con-fil.orc) fest gelegt.

Im einen zweiten Orc-File (rand4.orc) wird die gesamte Form des Stückes, mit den output von samples-con-fil.orc als Material, definiert. Dieses wird in der Amplitude mit randi (c-sound) moduliert. Es entstehen verschiedenen rhythmische Bewegungen für die verschiedenen Lautsprecher. Am Anfang kommt der output aus den Kanälen 1 und 2. Die Bewegungen fangen mit den Wert 1 (pro Sekunde) an. Dieser steigert sich bis Sekunde 70 auf 1000, wobei eine Art von Rauschen entsteht. In der Sekunde 100 treten die hinteren Kanäle 3 und 4 mit eine langsame Modulation auf. In der Sekunde 110 verschwindet die Modulation in die Kanäle 1 und 2. In der Sekunde 130 erreicht die Modulation 1000 auf die Kanäle 3 und 4. In 30 Sekunden sinkt er auf 1.

In der Sekunde 90, als man in die Kanäle 1 und 2 Rauschen hört, treten langsame Modulationen ein die diesen neuen Klang auch die Bewegungen des originalen Materials geben.

Der letzte Schritt der Komposition von diesen Stück entsteht bei randi4cmix.orc, wo das Ergebnis von rand4.orc am Ende durch transponierte Wiederholungen der letzten 25 Sekunden für ein Fade-out des Stückes ergänzt wird.

Lisp File:

```
(defun my-lin (idx start end)
  (+ (* (- end start) idx) start)) ;lineare Funktion
;=====
=

(defobject score-file (i) ;Definiert eine neue Objektklasse
  (time duration ifreq1
    ifreq2
    ifreqsample)
  (:parameters time duration ifreq1 ifreq2 ifreqsample))

;=====

(progn ;Returns value of the last expression.
  (defprocess tepich ()
    (let ((e (new score-file))) ;New: allocates and initializes a new
      ;object
      (process for count from 1 to 80 do ;80 Ereignisse
        (let ((normalized-idx (/ count 80)))
          (sv e
            ins      4
            time      (now)
            duration  (my-lin normalized-idx 6.0 15.0) ;Dauer von 6 bis 15 Sek. in
                                                         ;Intervallen von (15-6)*1/80.
            ifreq1    (my-lin normalized-idx 1000.0 1100.0) ;Mittelfrequenz am Anfang.
            ifreq2    (my-lin normalized-idx 1100.0 900.0) ;Mittelfrequenz am Ende.
            ifreqsample (my-lin normalized-idx 0.5 0.01) ;Sample Transposition.
          )
          (output e) ;Object to the open output stream.
          (wait      (my-lin normalized-idx 3.8 1.1)))) ;Wait sets process to run again.
                                                         ;Einsatzabstand
        (events (tepich) "/users/doerries/la_movie/dife_con_fil3.sco") ;Erzeugung von Sco-file
      )
    )
  )
```

samples_con_fil.orc :

sr = 48000
kr = 48000
ksmps = 1
nchnls = 2

instr 1

ifreq1 = p4 ;Mittelfrequenz am Anfang des Samples
ifreq2 = p5 ;Mittelfrequenz am Ende des Samples
ifreqsam = p6 ;Transposition des Samples

asig1, asig2 diskinn "/snd/icemserv/doerries/exmovie/pegados1.aiff", ifreqsam

kbw linseg 1, p3/4, 10, p3/4, 100, p3/4, 10, p3/4, 1
kamp linseg 0, p3/2, 1, p3/2, 0 ;Hüllkurve
kfreq linseg p4, p3, p5 ;Mittelfre. von 1000 zu 1100 bis 1100 zu 900

afilt1 reson asig1, 1*kfreq, kbw ;Filterbank,(links) Grundton*(1+0.1*2^n).
afilt2 reson asig1, 1.2*kfreq, kbw ;Rekursives Bandpassfilter 2. Ordnung
afilt3 reson asig1, 1.4*kfreq, kbw ;12 dB/Oktave
afilt4 reson asig1, 1.8*kfreq, kbw
afilt5 reson asig1, 2.6*kfreq, kbw
afilt6 reson asig1, 4.2*kfreq, kbw
afilt7 reson asig1, 7.4*kfreq, kbw
afilt8 reson asig1, 13.8*kfreq, kbw

afilt10 reson asig2, 1*kfreq, kbw ;Filterbank rechts.
afilt20 reson asig2, 1.2*kfreq, kbw
afilt30 reson asig2, 1.4*kfreq, kbw
afilt40 reson asig2, 1.8*kfreq, kbw
afilt50 reson asig2, 2.6*kfreq, kbw
afilt60 reson asig2, 4.2*kfreq, kbw
afilt70 reson asig2, 7.4*kfreq, kbw
afilt80 reson asig2, 13.8*kfreq, kbw

asigba1 balance afilt1, asig1 ;Balancierung gefiltertes mit original Signal
asigba2 balance afilt2, asig1
asigba3 balance afilt3, asig1
asigba4 balance afilt4, asig1
asigba5 balance afilt5, asig1
asigba6 balance afilt6, asig1
asigba7 balance afilt7, asig1
asigba8 balance afilt8, asig1

asigba10 balance afilt10, asig2
asigba20 balance afilt20, asig2
asigba30 balance afilt30, asig2
asigba40 balance afilt40, asig2
asigba50 balance afilt50, asig2
asigba60 balance afilt60, asig2
asigba70 balance afilt70, asig2
asigba80 balance afilt80, asig2

outs (asigba1+asigba2/2+asigba3/3+asigba4/4+asigba5/5+asigba6/6+asigba7/7+asigba8/8)/3.2*kamp,
(asigba10+asigba20/2+asigba30/3+asigba40/4+asigba50/5+asigba60/6+asigba70/7+asigba80/8)/2*kamp ;Unharmonisches Spektrum

endin

instr 2 ;Gleich wie instr1, anderen Sample.'

randi4c.orc

sr = 48000
kr = 48000
ksmps = 1
nchnls = 4 ;4-Kanal

instr 1

asig1, asig2 disk in "/snd/icemserv/doerries/exmovie/samples_con_fil.aiff", 1 ;Klangteppich
;als Material

kamp linseg 0.001, 30, 1, 160, 1, 10, 0.001 ;Fade-in und -out von ganzem Stück

kpausen1 linseg 5, 30, 10, 10, 0, 10, 9, 10, 100, 10, 1000, 30, 1000, 10, 0, 90, 0.1
kpausen2 linseg 10, 30, 5, 10, 0, 10, 10, 10, 100, 10, 1000, 30, 1000, 10, 0, 90, 0.11
kpausen3 linseg 0, 100, 0, 20, 100, 10, 1000, 30, 1, 40, 0.1
kpausen4 linseg 0, 100, 0, 20, 100, 10, 1000, 30, 1, 40, 0.11 ;Die verschiedenen Dichten
;von Bewegung erzeugen formales Charakter des Stücks

asigrandi1 randi 2, kpausen1
asigrandi2 randi 2, kpausen2 ;Straight-line interpolations between each number
asigrandi3 randi 2, kpausen3 ;and the next (Band-limited noise. Kpausen specifys
asigrandi4 randi 2, kpausen4 ;rate of new random numbers).

asigmix1 = asig1*asigrandi1*kamp ;LV Ampmod. Von Klangteppich mit Randi Werte
asigmix2 = asig2*asigrandi2*kamp ;RV
asigmix3 = asig1*asigrandi3*kamp ;LH
asigmix4 = asig2*asigrandi4*kamp ;RH

;Rekursive Bearbeitung der Bewegungsform. AmpMod. wird so stark das es Rauschen wird, und wird ;wieder Ampmoduliert.

kpausenb1 linseg 0, 80, 0, 10, 8, 5, 100, 5, 1000, 10, 1000, 10, 0, 80, 0
kpausenb2 linseg 0, 80, 0, 10, 10, 5, 120, 5, 1000, 10, 1000, 11, 0, 79, 0

asigrandib1 randi 2, kpausenb1
asigrandib2 randi 2, kpausenb2

asigmixb1 = asigmix1*asigrandib1
asigmixb2 = asigmix2*asigrandib2

kamphinten linseg 0.001, 70, 0.001, 20, 1, 110, 1

outq asigmixb1*.6, asigmixb2*.6, asigmix3*kamphinten*.6 ,asigmix4*kamphinten*.6

endin

randi4cmix.orc

;Endmischung von randi4c mit Transpositionen am Ende.

sr = 48000
kr = 48000
ksmps = 1
nchnls = 4

instr 1
asig1, asig2, asig3, asig4 diskid "/snd/icemserv/doerries/exmovie/randi4c.aiff", 1
outq asig1*.75, asig2*.75, asig3*.75, asig4*.75
endin

instr 2
asig1, asig2, asig3, asig4 diskid "/snd/icemserv/doerries/exmovie/randi4c.aiff", p4, 175
kamp linseg 0.001, p3/2, 1, p3/2, 0.001
outq asig1*kamp, asig2*kamp, asig3*kamp*1.5, asig4*kamp*1.5
endin

randi4cmix.sco

i 1 0 200
;p1 p2 p3 pitch
i 2 185 22 1
i 2 190 22 0.9
i 2 195 22 0.8
i 2 200 22 0.7
i 2 208 22 0.65
i 2 218 22 0.6