

„8-fach polyphoner Trigger-Response“: verschiedene Algorithmen, 8 rhythmische Patterns, 8 Klangfarben!

Hardware/Software:

Master-Atari: Programm TRIGGER.BAS

- sendet ein Pattern von 8 mal 16 Triggerzahlen,
- Tempor in 1/16 ist einstellbar,
- Pattern ist programmierbar,
- Anzahl der Kanäle 1-8 ist wählbar.

Die Trigger lauten Midi=1, Midi=2 usw. bis Midi=8

Slave-Atari 1 bis 8: Programm Triggerresponse mit eigenem Namen, z.B. RANDOM1.BAS

- geben ein Midi-Event aus, wenn der entsprechende Trigger eintrifft,
- Midi-Kanal = Triggerzahl,
- Im Unterprogramm „Algorithmus“ kann eingestellt werden, nach welchem Algorithmus die Tonhöhe und Velocity errechnet werden,
- Am Bildschirm wird eine Punktegrafik der gesendeten Töne angezeigt.

Ad lib. können im Unterprogramm „Algorithmus“ weitere features untergebracht werden, zum Beispiel

- Klangfarbenwechsel am angesprochenen Soundmodul,
- Lichtimpulse für midifiziertes Licht,
- Möglichkeiten, Parameter des Algorithmus durch Eingaben zu verändern.

Ad lib. können im Hauptprogramm weitere interaktive Aktionen eingebaut werden, zum Beispiel

- die Möglichkeit, sich vom Trigger abzukoppeln und metrisch frei zu senden oder
- ein Kanalwechsel .

Kompositionsparameter:

- Wahl und Programmierung des Algorithmus,
- Komposition der rhythmischen Patterns pro Kanal,
- Wahl der Klangfarbe am jeweiligen Soundmodul,
- Konzept der Interaktion der SpielerInnen,
- gegebenenfalls Interaktionen an einem Slave-Atari durch SpielerIn.

Spiel- und Improvisationsparameter:

- Bedienung des Algorithmus,
- Interaktion mit den „freien“ Algorithmusparametern,
- Interaktion mit den anderen SpielerInnen nach Plan,
- Klangfarbenveränderungen am Soundmodul.

Alle SpielerInnen sind auch KomponistInnen!

Fraktale Musik: Elementarlehre der Musikprogrammierung

Zwei Midi-Konventionen

- Note-ON-Befehl besteht aus 3 Midizahlen:
143+Ch für Midikanal Ch=1 bis Ch=16 bedeutet „Note ON auf Kanal Ch“
Taste für die anzuschaltende „Taste“ (reicht von 0 bis 127, c' = 60, c'' = 72 usw.)
Velocity für die Anschlagsstärke (reicht von 0 bis 127)
- Note-OFF-Befehl ist ein ON-Befehl mit Velocity = 0.
- Programm-Wechsel besteht aus 2 Midizahlen:
191+Ch für Programmwechsel auf Kanal Ch.
Programmnummer (von 0 bis 127).

Omikron Basic

Disketten: DD statt HD verwenden. Am Atari formatieren oder am PC mit 720 KB formatieren.

Starten des Programms Omikron Basic: Doppelklick auf Ikon **OM-BASIC.PRG**

Nach Start **HELP**-Taste drücken: Der „Editor“ steht bereit. Dort Programmtext eintippen oder ein Programm laden: Menu FILE, Eintrag **LOAD**.

Regel: Jede getippte Programmzeile steht erst nur auf dem Bildschirm. Sie gelangt in den Speicher, wenn **RETURN** gedrückt wird. Sie wird fest gespeichert, wenn das gesamte Programm unter einem Namen mit Endung BAS unter **SAVE** im Menu FILE auf Diskette gespeichert wird.

Programm starten durch **RUN** im Menu RUN.

Zeilennummern ein-/ausschalten mit Tastenkombination [Control] + [Clr Home].

Midi-Programmierung

BIOS(,3,3,M) befördert die Zahl M an den Midi-Ausgang des Computers.

Um mehrere Midizahlen an den Midi-Ausgang zu befördern, schreibt man mehrere BIOS-Befehle untereinander oder durch Doppelpunkte getrennt nebeneinander.

BIOS(M,1,3) stellt fest, ob am Midi-Eingang eine Zeichen vorliegt. Falls dies der Fall ist, ist M = -1.

BIOS(X,2,3) liest das Zeichen aus dem Zwischenspeicher (Buffer) des Midi-Eingangs und speichert es unter der Bezeichnung X ab.

Sonstige Programmierung

REPEAT ... UNTIL bedeutet: Das Programm, das zwischen REPEAT und UNTIL steht, wird so lange durchlaufen, bis die Bedingungen, die hinter UNTIL steht, erfüllt ist. („Schleife“)

FOR N = 1 TO 12 ... NEXT ist eine Schleife, die genau 12 mal durchlaufen wird, wobei die Zahl N bei jedem Durchlauf einen der Werte 1, 2, usw. bis 12 erhält.

IF ...[Bedingung]... THEN ...[Befehl]... ENDIF bedeutet: Wenn eine Bedingung erfüllt ist, dann führe den Befehl aus, ansonsten nicht.

Kommt ein Wort (oder ein Buchstabe) mit Gleichheitszeichen vor, so ist dies die Bezeichnung eines Speicherplatzes, auf dem Zahlen abgelegt werden können.

Kommt ein Wort ohne Gleichheitszeichen vor, so sucht das Programm nach einem Unterprogramm, das durch **DEF PROC Wort** gekennzeichnet ist und mit **RETURN** endet.

Taste = RND(24) bedeutet: Es wird ein Speicherplatz mit der Bezeichnung „Taste“ eingerichtet. ein Zufallswert zwischen 0 und 23 wird auf diesem Platz abgelegt.

Vel = 23+Taste/2 bedeutet: Es wird ein Speicherplatz mit der Bezeichnung „Vel“ eingerichtet. Dort wird eine Zahl abgelegt, die sich aus dem Inhalt des Speicherplatzes „Taste“ nach der angegebenen Formel errechnet.

PRINT @(Z, S); „Zeichenfolge“ gibt die Zeichenfolge „Zeichenfolge“ beginnend mit der S-ten Spalte in der Z-ten Zeile des Bildschirms aus.

INPUT @(Z,S); „Gib ein:“; N schreibt zunächst „Gib ein:“ auf den Bildschirm. Das Programm wartet, bis eine Zahl eingetippt und mittels RETURN in den Computerspeicher gesendet wird. Dann wird die eingetippte Zahl auf dem mit „N“ bezeichneten Speicherplatz abgelegt und das Programm läuft weiter.

WAIT Z hält das Programm Z Sekunden lang an.

Zufallsprogramm (elementar):

<pre> 1. REPEAT 2. Zufallszahl=RND(25) 3. Taste = 60+Zufallszahl 4. BIOS(,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3,1 27) 5. WAIT 0.1 6. BIOSD(,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3, 0) 7. UNTIL LEN(INKEY\$) 8. END </pre>	<p>Ziffer 1.-7 = Schleife, die solange durchlaufen wird, bis „LEN(INKEY\$)<>0 erfüllt ist, was dann der Fall ist, wenn irgendeine Kbd-Taste gedrückt wird...</p> <p>Zeile 2: Zufallszahl zwischen 0 und 24 wird ermittelt.</p> <p>Zeile 3: Eine Zufallstaste zwischen 60 und 84 wird errechnet.</p> <p>Zeile 4: Note-ON, Zeile 5: Note-OFF</p> <p>Zeile 5: Länge des Tons 0,1 Sekunden</p>
---	--

Struktur des Programms: Zufallsfeld durch Tastendruck verändern...

Quellcode	Erläuterung
<pre> 'Zufallsfeld durch Tastendruck (RANDOM0.BAS) BIOS (,3,3,192): BIOS (,3,3,36) Scan=75 PRINT @(2,2);"Zufallsambitus durch Tasten gesteuert" PRINT @(3,2);"END durch SPACE-Taste." PRINT @(4,2);"Zufallsumfang = ";Scan REPEAT 'Abfrage der Tastatur (unverst,ndlich) A\$= INKEY\$ IF A\$<>"" THEN A= CVI(A\$) Scan=A SHR 16 PRINT @(4,2);"Zufallsumfang = " PRINT @(4,2);"Zufallsumfang = ";Scan ENDIF 'Algorithmus Taste=60+ RND(Scan) IF Taste<128 THEN BIOS (,3,3,144): BIOS (,3,3,Taste): BIOS(,3,3,127) WAIT .1 BIOS (,3,3,144): BIOS (,3,3,Taste): BIOS (,3,3,0) ENDIF UNTIL Scan=57 PRINT @(4,2);"ENDE " END </pre>	<p>Alle Zeilen mit einem Apostroph am Anfang werden ignoriert (dienen der Orientierung) 192 der Program-Change-Befehl für Midikanal 1, das Programm selbst hat die Nummer 36. Scan= 75 ist ein Anfangswert (siehe unten).</p> <p>Der Computer durchläuft die Schleife REPEAT - UNTIL so lange, bis Scan = 57 ist (gleichbedeutend: SPACE-Taste wurde gedrückt).</p> <p>Mit dem Unterprogramm „Abfrage der Tastatur“ fragt der Computer laufend ab, ob eine Taste gedrückt wurde. Wenn ja, wird die Tastennummer unter der Bezeichnung Scan gespeichert. Hier bedeutet A\$<>,,“ dasselbe wie „eine Taste wurde gedrückt“.</p> <p>Im Unterprogramm „Algorithmus“ sucht der Computer zufallsbedingt eine Zahl zwischen 0 und Scan-1 aus. Das Ergebnis wird zu 60 dazugezählt. Dauer des Zufallstones beträgt 0,1 Sekunden.</p> <p>Hier endet die Schleife! Sie wird durchlaufen bis Scan = 57 ist (Nummer der Spacetaste ist 57).</p>

Das Timing dieses Programms ist durch die „WAIT-Zeit“ und die Schnelligkeit des Computers bestimmt. Letztere spielt keine hörbare Rolle, ist aber eine Ursache dafür, dass derartige Programmabläufe nicht auf mehreren Computern synchron laufen können (Phasenverschiebung der Minimal Music!).

Dasselbe Progreamm mit einem extrenen Zeitgeber (TRIGGER-Programm)

```

0 'Zufallsfeld getriggert: Tastendruck (RANDOM4.BAS)
1
2 PRINT @(4,2);"Zufallsambitus durch Kbd-Tasten."
3 PRINT @(5,2);"Zufallsfolgen getriggert."
4 PRINT @(7,2);"Ende mit SPACE!"
5 INPUT @(8,2);"Midi-Empfangskanal = ";Kanal
6
7 BIOS (,3,3,192): BIOS (,3,3,36)
8 Scan=75
9 Clear_Buffer
10
11 REPEAT
13 'Tastaturabfrage
14 A$= INKEY$
15 IF A$<>""
16 THEN
17 A= CVI(A$)
18 Scan=A SHR 16
19 PRINT @(10,2);"Gradient: = ";Scan;" "
20 ENDIF
22 'Midiingang Abfrage
23 BIOS (M,1,3)
24 IF M=-1
25 THEN
26 BIOS (X,2,3)
27 IF X=Kanal-256 THEN Algorithmus
28 ENDIF
30 UNTIL Scan=57
31
32 'Abschalten des letzten Tons
32 BIOS (,3,3,143+Kanal): BIOS (,3,3,Taste_Alt): BIOS
(,3,3,0)
34 PRINT @(3,3);"Ende!"
35 END
36
38 DEF PROC Algorithmus
39 'Errechnen und Play
40 Taste=60+ RND(Scan)
41 IF Taste<128
42 THEN
43 BIOS (,3,3,143+Kanal): BIOS (,3,3,Taste_Alt): BIOS
(,3,3,0)
44 BIOS (,3,3,143+Kanal): BIOS (,3,3,Taste): BIOS
(,3,3,127)
45 Taste_Alt=Taste
46 ENDIF
47 RETURN
48
49 DEF PROC Clear_Buffer
50 REPEAT
51 BIOS (M,1,3)
52 IF M = -1 THEN BIOS (X,2,3)
53 UNTIL M=0
54 RETURN

```

Zeile 5 bedeutet: erst wenn eine Zahl eingetippt und mit RETURN bestätigt wurde, läuft das Programm weiter und speichert die eingetippte Zahl unter „Kanal“ ab.

Das Wort „Clear_Buffer“ lässt den Computer in das Unterprogramm Zeile 49 bis 54 springen.

Die Schleife fragt nicht nur die Tastatur ab (wie bei RANDOM0.BAS), sondern in Zeile 23 bis 28 auch den Midi-Eingang. Liegt dort eine Midi-Zahl, so ist M = -1. In diesem Falle wird mit BIOS (X,2,3) die entsprechende Zahl als X ermittelt. Sie ist Midi-Zahl - 256. (Warum das so ist, weiß niemand...) Trifft bei Kanal=1 die Midizahl 1 ein, so ist in Zeile 27 die IF-Bedingung erfüllt und das Programm springt in das Unterprogramm „Algorithmus“.

Ist die Midizahl „Kanal“ eingetroffen, so wird dies Unterprogramm durchlaufen: erst wird die Zufallszahl bestimmt, dann der frühere Ton abgeschaltet und sodann der neue Ton gespielt. Das Abschalten des alten Tons erfordert, dass die entsprechende Tastennummer in Zeile 45 unter einem neuen Namen abgespeichert wird.

Das Clear Buffer-Programm leert den Midi-Eingangs-Buffer. Es werden solange Zahlen abgerufen (und nicht verarbeitet), bis der Buffer leer, d.h. M=0 ist.

Veröffentlichungen von Wolfgang Martin Stroh zur Thematik: Elektronische Musik mal anders herum, algorithmisches Komponieren und MIDI-Experimente

- Aufsatz im Ordner vorhanden (gegebenenfalls mit Diskette)

(B) 1994. - Tonsysteme und Stimmungen hören. Microtuning auf Midi-Instrumenten. Ein Handbuch zum praktischen Experimentieren mit Stimmungen aus aller Welt (= midi-pädagogische Schriftenreihe, Heft 10). 140 Seiten + Diskette. Verlag: musiklabor Berlin, Berlin.

(B) 1991. - Gibt es eine andere Musikwissenschaft? (= Oldenburger Universitätsreden, Nr. 41). 42 Seiten. Verlag: BIS-Verlag, Oldenburg. ISBN 3-8142-1041-7.

(B) 1991. - Midi-Experimente und Algorithmisches Komponieren, Band 2. Programme und Projekte für den Musikunterricht und die Musikpraxis (= midi-pädagogische Schriftenreihe Band 6). 154 Seiten + Diskette. Verlag: musiklabor Berlin, Berlin.

(B) 1991. - Microtones. Synthesizerstimmungen für Musikpraxis, Musikpädagogik und Musikwissenschaft. Software-Paket. 16 Seiten + Diskette. Verlag: Geedes Midisystems, Berlin.

(B) 1990. - Midi-Experimente und Algorithmisches Komponieren, Band 1. Eine Anleitung zum kreativen Programmieren und Komponieren am Computer (= midi-pädagogische Schriftenreihe Band 3). 146 Seiten + Diskette. Verlag: musiklabor Berlin, Berlin.

(B) 1975. - Zur Soziologie der elektronischen Musik. 200 Seiten. Verlag: Amadeus-Verlag, Zürich.

2001 - Tonsystem als Tonvorrat. Zur Aufhebung einer Differenz beim MIDI-Planetarium. In: Positionen. Beiträge zur Neuen Musik 48, 8/2001, S. 38-41.

2001 - Das Oldenburger TechnoMuseum - Ein subjektiv-kollektiver Erfahrungsbericht. In: Musikunterricht heute, Band 4, hg. von Jürgen Terhag. Lugert-Verlag Oldershausen, S. 187 und interaktiver Beitrag auf CD-ROM.

2000 - Musikalische Grenzerfahrungen durch Midifiles? Hörpsychologische Experimente im Musikunterricht. In: Computer und Musikunterricht. Sonderheft von "Praxis Musikunterricht", Lugert-Verlag Oldershausen, S. 36-43. (Mit CD-ROM: Midifiles, Audio- und WAV-Material.)

- 2000 - Virtuell-analoge Synthesizer. Ein workshop zu ihrem Einsatz in der Schule. In: Praxis des Musikunterrichts, Heft 61, 2/2000, S. 40-48. Audiobeispiele und Midifiles auf Hybrid-CD.

1997 - Das MIDI-Planetarium. Zur Revision harmonikalere Vorstellungen von Sphärenmusik. In: Neue Musiktechnologie II, hg. von Bernd Enders. Schott-Verlag, Mainz, S. 74-85. Mit CD-Einspielung der Musikbeispiele.

- 1995. - Algorithmisches Komponieren. Ein Weg zu musikalischer Kreativität und Selbsterfahrung?. In: LOG IN 15, 4/1995, S. 16-21. Verlag: LOG IN GmbH, Berlin

1995 - Musikpädagogische Maßnahmen gegen den Fetischcharakter des Computers. Zum Konzept des algorithmischen Komponierens. In: Musikpädagogische Forschung, Band 16, hg. von Georg Maas, S. 60-68. Verlag: Die Blaue Eule, Essen.

- 1993. - Harmonie, Chaos und Computer - Neue Technologien im New Age. In: Neue Musiktechnologie, hg. von Bernd Enders und Stefan Hanheide, S. 94-108. Verlag: Schott, Mainz

1993. - Zus. mit Pape, Jörg: Standard Midifiles - ein didaktisches Plädoyer für eine neue Industriennorm. In: Musik und Unterricht, Heft 23, 11/1993, S. 36-39. Verlag: Friedrich, Seelze

1991. - Die Grenzen der Echtzeit. Soll, muß und kann auf Computern improvisiert werden?. In: KEYS. Magazin für Musik und Computer Heft 1/1991, S. 116-121. Verlag: Presse Project Verlag, Bergkirchen

1990. - Musikalische Rebellion gegen Musikcomputer? Zur Begründung des Musikprojekts "Brain & Body". In: Neue Zeitschrift für Musik 5/1990, S. 3-9. Verlag: Schott, Mainz

1990. - Programmierte Verstimmung. BASIC-Programmierung von Microtuning. In: KEYS. Magazin für Keyboard, Computer und Recording, Heft 5, S. 54-57. Verlag: Presse Project Verlag, Bergkirchen

- 1987. - Anregungen zum Musizieren im MIDI-Verbund, Teil I. In: Populäre Musik im Unterricht Nr. 18, Mai 1987, S. 23-28. Verlag: Institut für Didaktik Populärer Musik, Oldershausen

- 1987. - Anregungen zum Musizieren im MIDI-Verbund, Teil II. In: Populäre Musik im Unterricht Nr. 19, September 1987, S. 24-27. Verlag: Institut für Didaktik Populärer Musik, Oldershausen

- 1986. - Der "Commodore 64" als preiswerter Klangsammler und Rhythmusautomat. In: Populäre Musik im Unterricht Nr. 16, August 1986, S. 20-24. Verlag: Institut für Didaktik Populärer Musik, Oldershausen

- 1986. - Erfahrungen vor dem Bildschirm bei schlechtem Wetter. In: Populäre Musik im Unterricht Nr. 15, März 1986, S. 5-10. Verlag: Institut für Didaktik Populärer Musik, Oldershausen

1984. - Möglichkeiten und Grundlagen der Klangproduktion. In: Musik in der Kollegschule, hg. von Ulrich Günther (= Curriculum Heft 33), S. 105-126. Verlag: Landesinst. f. Schule und Weiterbildung, Soest

1983. - Zur Soziologie der elektronischen Musik in den 80er Jahren. In: Schweizerische Musikzeitung 1/1983 (123. Jahrgang), S. 14-25. Verlag: Hugh & Co, Zürich.

1979. - Musikbezogene Physik. In: Musikunterricht in der Sekundarstufe II, hg. von A. Bojanowski und U. Günther, S. 106-117 (= Athenäum Taschenbücher 3154). Verlag: Athenäum, Königstien/Taunus

1974. - Hochschuldidaktische Probleme bei der Integration elektronischer Studioarbeit in ein musikwissenschaftliches Studium. In: Musik. Fachdidaktisches Studium in der Lehrerbildung, hg. von Wolfgang Schmidt, S. 257-274. Verlag: Oldenbourg, München

1974. - Mathematik und Musikterminologie. In: Zur Terminologie der Musik des 20. Jahrhunderts, hg. von Hans Heinrich Eggebrecht, S. 33-59 (= Veröffentlichung der Walcker-Stiftung, Heft 5). Verlag: Walcker, Stuttgart

1973. - Die elektronische Szene 1972. In: Melos. Zeitschrift für Neue Musik, Heft V, September/Oktober 1973, S. 279-287. Verlag: Schott, Mainz

1972. - Über die Schwierigkeiten bei der mathematischen Darstellung musikalischer Sachverhalte. In: Neue Zeitschrift für Musik X, S. 567-570. Verlag: Schott, Mainz

Weitere Texte im Ordner:

- Vortrag 1998: *Fraktale Musik* (mit Diskette)
- Vortrag 1995: *Vom Algorithmus zum Gamelan* (mit Diskette)
- Unterrichtseinheit „Musik und Grafik selber programmieren!“ 1995 (mit Diskette)
- „Fractal Workstation“ 1993 (mit Verkaufs-Diskette)
- Arbeitsblätter aus Kursen der Semester 1994, 1991/92, 1990/91
- Vortrag 1990: *Harmonie und Chaos - Einblick in ein neues Weltbild durch Midi-Experimente?* (mit Diskette)