

das **synthesizer** MAGAZIN

**Circle of Live
im Bassiani**

Neun Stunden
Liveset

**Oto Machines
Bim, Bam, Boum**

Die französische
Effekt-Trilogie

**Pittsburgh
Modular Taiga**

From the East Coast
to the West Coast

**Serge
Precision Oszillator**

Fett, präzise, vintage und
ein Mauerblümchen

**Live Coding
mit Orca**

Esoterik für
kreative Köpfe

**AB JETZT MIT
84 SEITEN!**

Synmag Cribs

Studiobesuch in
der Community



Roland
SH-4d



Elektron
Digitakt 1.5



Mayer
MD900



GS Music
GS-e7



2 https://metasyn.github.io/learn-orca -3 https://youtu.be/Kks8scODgqM -4 https://youtu.be/nKAjx6n971Y -5 https://youtu.be/nKAjx6n971Y?t=222

.D3.....
3u4.....
.:03C..

Der Operator **D** sendet jedes dritte Frame einen Trigger. Dieser Impuls wird durch den Operator **u** entgegengenommen, mittels Euklidischen Rhythmus-Logik verarbeitet und dann durch den Operator **:** als MIDI-Note ausgegeben.

Allgemein gilt für Operatoren in Orca, dass diese, sofern sie als Kleinbuchstaben notiert sind, per Bang getriggert werden müssen. Der Operator **C** ist ein naher Verwandter des Operators **D**, gibt aber Zahlen statt Bangs/Trigger aus. Diese werden auch hier als Modulo-Werte (ganzzahliger Rest) des Teilers, der hinter dem Operator steht, berechnet. Grundsätzlich hat der Clock-Operator die gleiche Modulo-Logik, nur wird hier als Event ein Bang (*), statt des numerischen Wertes ausgegeben. Der Operator **T** stellt einen Track von Werten, typischerweise Noten, zur Verfügung.

.4C4.....
D414THEAD
*:03E84..

Im Beispiel werden bei jedem vierten Frame Clockwerte zwischen **0** und **3** generiert. Das Ergebnis wird nach unten weitergegeben und im vorliegenden Fall in eine Zeichenkette, die einer MIDI-Note entspricht,

eingesetzt. Der Operator **D** gibt dann den Beat für die Ausgabe der MIDI-Events vor. Die MIDI-Note **03E84** gibt dann folgendes an: Kanal (**0**), Oktave (**3**), Note (**E**), Velocity (**8**), Länge (**4**).

Auch hier sind die Zeichen Orca-typisch im Zahlensystem mit Basis 36 notiert. Grundsätzlich kann man in Orca beliebige Folgen von Argumenten für Operatoren dynamisch zusammensetzen, das heißt zur Laufzeit generieren. Somit sind der Kreativität, bezogen auf die erzielbaren Ergebnisse, kaum Grenzen gesetzt.

Interaktive Seite zum Lernen und Ausprobieren von Orca

Unter dem angegebenen Link² kann man das soeben Beschriebene direkt interaktiv ausprobieren und sich viele Tipps und Beispiele ansehen. Hier kann man auch die Logik des zuletzt genannten Beispiels unter ‚misc → chromatic‘ gut nachvollziehen.



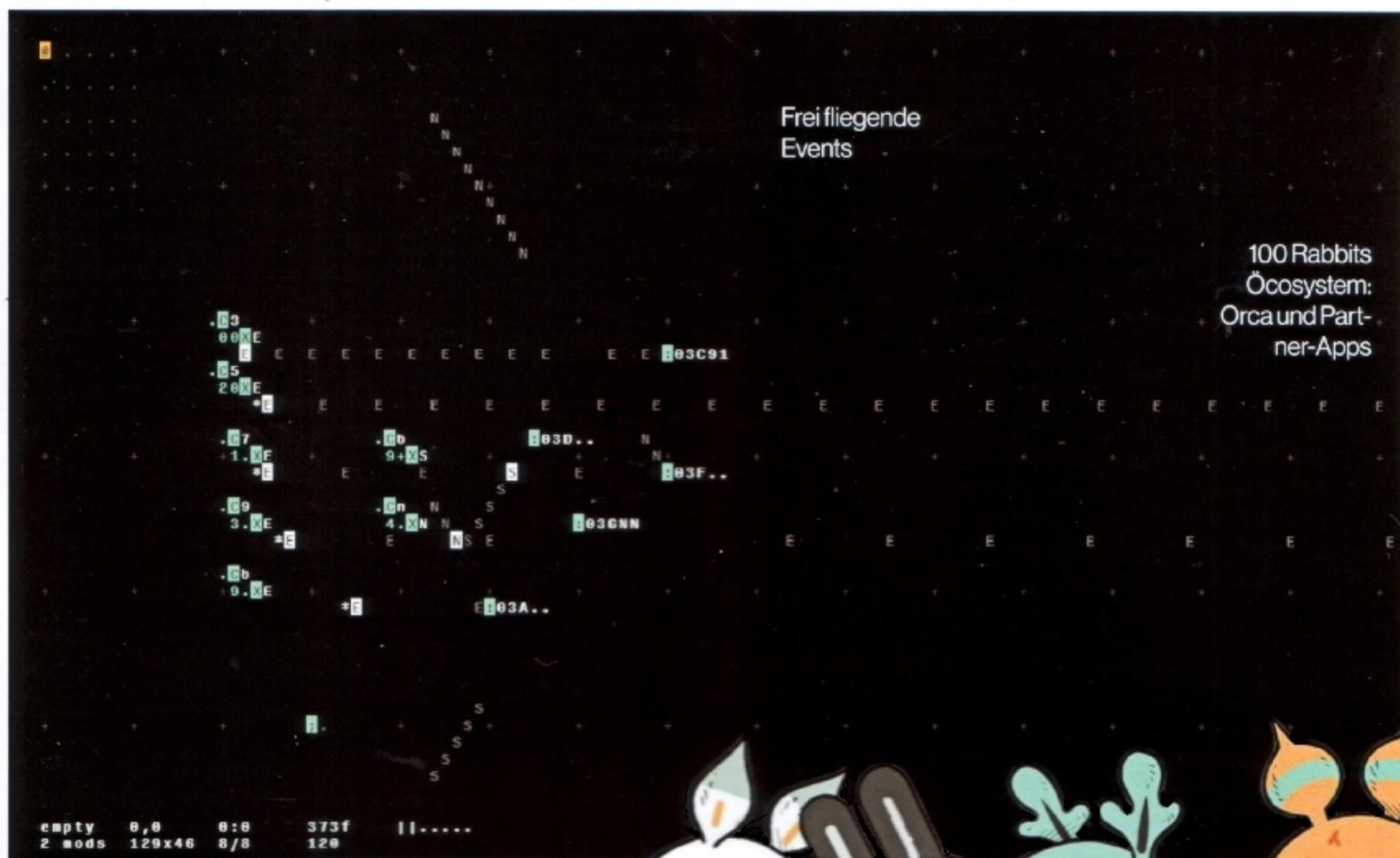
Systemverbindungen und Klangänderungen mit Orca

Orca kennt keine direkten Operatoren zur Klangänderung, aber indirekt ist hier trotzdem fast alles möglich, was man aus gängigen DAWs kennt. Die Auswahl verschiedener Instrumente lässt sich sehr gut über unterschiedliche MIDI-Kanäle abbilden. Für fließende Klangänderungen

kann der Operator **!** genutzt werden. Dieser Operator gibt MIDI-Controller-Events aus. Noch komplexere Änderungen sind mit dem Operator **=** möglich, der einen beliebigen OSC-String ausgeben kann. Dies setzt natürlich einen OSC-fähigen Synthesizer als Empfänger voraus. Auch bei MIDI muss der Ausgangsport von Orca mit dem Eingangsport des Synths oder der DAW verbunden werden. Hierzu kann man entweder zwei MIDI-Ports physikalisch verbinden oder rechnerintern eines der gängigen MIDI-Rerouting Tools für solche Fälle verwenden.

Synchronisation mit der DAW

Via MIDI-Clock lässt sich Orca zur DAW oder einem Synth synchronisieren oder schickt alternativ MIDI-Clock-Events an den gesetzten Ausgang. Orca ist für viele verschiedene Plattformen verfügbar. Als Desktop-Version für macOS, Windows oder Linux. Die UXN-Version ermöglicht es Orca, auf mehreren „Kleincomputern“ oder nicht mehr aktuellen Umgebungen wie dem Nintendo DS, Amiga oder DOS zu laufen. Neben der bereits erwähnten Browser-Version gibt es auch eine reine Terminal-Version. Hier arbeitet man ausschließlich mit Tastatur und Text. Da Orca selbst textbasiert ist, stellt das optisch nur eine geringe Einschränkung dar. Die Terminal-Version ist in der Programmiersprache C geschrieben und lässt sich auf viele unterschiedliche Zielsysteme portieren. Zusätzlich gibt es eine Version, die bereits speziell an das Monome



Und wie funktioniert das jetzt alles zusammen?

Orca hat eine ähnliche visuelle Logik wie Max oder PD, d. h. es gibt virtuelle Module, die Trigger oder Werte austauschen. In dieser Logik gedacht, bestehen Module in Orca aus einem Operator und dessen (zum Teil optionalen) Argumenten. Diese „Module“ erzeugen dann Bangs (Trigger-Events) oder numerische Werte (base 36), die wiederum von anderen Operatoren verarbeitet werden können. Um das Ganze hörbar oder auch sichtbar zu machen, muss am Ende der Signalkette immer mindestens ein „Modul“ zur Ausgabe bereitstehen wie der MIDI-Operator. Dieser wird als **D** notiert und ihm folgen die Werte („Argumente“), die benötigt werden, um ein Klangevent zu erzeugen: *channel, octave, note, velocity und length*. Die Argumente werden ohne Zwischenräume und als base36-Werte notiert. Ein Beispiel, das im Folgenden noch näher erläutert wird, lautet **:03E84**. Fehlen Argumente, werden von Orca entsprechende Default-Werte benutzt. Der Ausdruck mit Standard *velocity* und *length* würde also **:03E** lauten. Besonders interessant wird es, wenn man keine festen Werte verwendet, sondern solche Anweisungen zur Laufzeit dynamisch generiert. Wie das genau geht, wird im Folgenden erläutert.

Zusätzlich zu MIDI unterstützt Orca die Netzwerkprotokolle UDP oder OSC zur Ausgabe. UDP (User Datagram Protocol) wird mit dem Operator **U** notiert, OSC

(Open Sound Control) mittels **O**. Welches Protokoll am günstigsten ist, hängt im Wesentlichen vom Empfänger ab. So kann Orca die Klangerzeuger der Life Coding-Umgebung Sonic PI gut über OSC ansteuern.

Ähnlich wie bei den erwähnten visuellen Programmiersprachen, wird bei Orca der Code nicht direkt von oben nach unten ausgewertet, sondern man kann den Signalfluss variieren. Die Operatoren N, E, S und W generieren Events nach Norden, Osten, Süden oder Westen. Dies ist in gewissem Maße vergleichbar mit Patchkabeln bei modularen Synthesizern oder bei Programmierumgebungen wie Max oder PD (Pure Data), die solche Beziehungen graphisch visualisieren. Allerdings haben diese Patchkabel/Signalflüsse bei Orca noch eine zusätzliche Dynamik, da die erzeugten Events sich im 2D-Raum bewegen, bis sie auf einen anderen Operator oder ein anderes Event treffen. Eine Eigenschaft, die wiederum kreativ eingesetzt werden kann. Eine gewisse Ausnahme von dieser Kontrollfluss-Logik stellen Variablen in Orca dar. Diese können an jeder Stelle im Raum gespeichert oder ausgelesen werden. Variablen sind somit als räumliche Abkürzungen oder sozusagen als kabellose Verbindungen benutzbar, ähnlich wie dies in Max mittels *send/receive* möglich ist. Allerdings muss man beim mehrfachen Auslesen iden-

tischer Variablen beachten, dass der aktuelle Wert immer dem zuletzt gesetzten entspricht – bezogen auf eine Leserichtung von links oben nach rechts unten.

Orcas Operatoren/Trigger pro Frame oder Bang?

Einer der wichtigsten „special operators“ ist **D**. Dieses Zeichen wird zur Ausgabe von MIDI genutzt. Im folgenden Beispiel gibt das Konstrukt mittels ‚Delay‘ jedes Achtel die Note C in der dritten Oktave aus.

```
D8 . . .
. :03C
```

Der Operator **D** sendet hierfür bei jedem achten Frame ein Bang (*) nach Süden, sprich nach unten. Die Framerate (sozusagen der Taktimpuls) und somit das Tempo hängen von dem global einstellbaren BPM-Wert ab. Ein Beat besteht aus 8 Frames. Das ist auch mittels Anzeige, die in der Statuszeile typischerweise auf 8/8 steht, ersichtlich. Dies ist aber nicht als eine Taktart im engeren Sinne anzusehen. Die generierten Noten werden rhythmisch ausschließlich durch die Triggerfrequenzen der Operatoren strukturiert. Den Begriff Bang kennen vielleicht einige von Max/PD. Auch hier steht er für den für Trigger-Impuls wie bei modularen Synthesizern.

Norns angepasst ist und somit auf dem Raspberry Pi läuft. Wie das Norns zusammen mit Orca funktioniert, kann man sich in mehreren Videos auf Youtube anschauen, z. B. hier³:



UXN ist genau wie Orca ein Projekt von Devine Lu Vega: „Das Uxn-Ökosystem ist ein kleiner Personal-Computing-Stack, der als Host für Tools und Spiele dient und in seiner eigenen Assembler-sprache programmiert werden kann.“

Anwendungsmöglichkeiten von Orca

Beim „klassischen Live Coding“ wird die Musik komplett durch improvisierten Code erzeugt, es sind aber natürlich auch andere Anwendungen möglich. So habe ich Orca als Inspirationsquelle und quasi „Session-partner“ schätzen gelernt. Wenn die DAW im Sync mitläuft, kann man einfach den MIDI-Datenstrom aufzeichnen und später geeignete Clips weiterverarbeiten. Oder man geht umgekehrt vor, indem Orca ein melodisches oder rhythmisches Pattern spielt und man in der DAW dazu improvisiert. Hierzu bieten sich gerade die Euclid-Pattern-Generatoren oder Stepsequenzer an. Mit ein paar kleinen

Tricks⁴ kann man Orca auch wie einen „klassischen Tracker“ verwenden.



Wer ist 100 Rabbits?

100 Rabbits ist ein kleines Künstlerkollektiv, bestehend aus Devine Lu Linvega (they) und Rek Bell (they/them, or iel). Nach eigenen Angaben „erforschen sie die geplante Fehlerhaftigkeit moderner Technologie an den Grenzen der hypervernetzten Welt“. Devine und Rek leben und arbeiten auf einem 10-Meter-Segelschiff namens Pino, mit dem sie auf dem Pazifischen Ozean zwischen Kanada und Japan unterwegs sind.

Alles auf Anfang

Die Dynamik einer interaktiven Live Coding-Sprache lässt sich in einem Artikel wie diesem mittels statischem Text nur unzureichend beschreiben. Man sollte wie mit einem Synth oder insbesondere einem Modularsystem damit herumexperimentieren. Dann wird vieles verständlicher und nach einiger Zeit intuitiv benutzbar, selbst wenn man in die Internas oder den Programmierer-Jargon nicht so tief einsteigen will. Wie heißt es auf der Orca-Seite und am Anfang dieses Artikels: „Orca ist eine zweidimensionale

esoterische Programmiersprache, in der jeder Buchstabe des Alphabets ein Operator ist, wobei Kleinbuchstaben pro Bang agieren, Großbuchstaben operieren pro Frame.“ Und „Die Applikation ist kein Synthesizer, sondern eine flexible Live Coding-Umgebung, die in der Lage ist, MIDI, OSC oder UDP zu versenden, um damit Klangerzeuger mittels Anwendungen wie Ableton, Renoise, VCV Rack oder SuperCollider anzusteuern.“ Hoffentlich hilft dieser Artikel, diese Konzepte besser nachvollziehen zu können. Dann steht dem Jammen mit Orca nichts mehr im Wege. Ich kann aus eigener Erfahrung sagen, dass solche Experimente mit neuen musikalischen Ideen belohnt werden, auf die man sonst nicht gekommen wäre. //

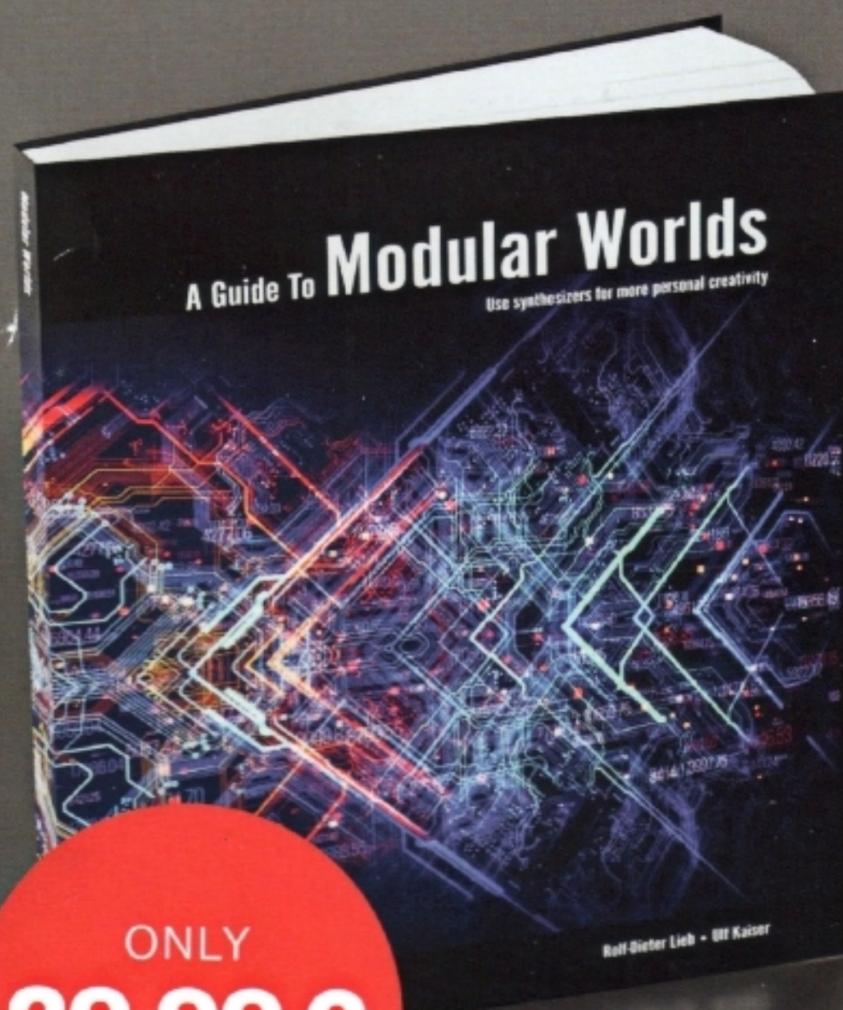
Live Coding mit Orca

- 100 Rabbits: <https://100r.co>
Orca: 100r.co/site/orca.html
- Interaktive Web-Version von Orca mit Tutorial: metasyn.github.io/learn-orca
- Desktop-Versionen zum Download (name your price): hundredrabbits.itch.io/orca

ANZEIGE

A GUIDE TO MODULAR WORLDS

USE SYNTHESIZER FOR MORE PERSONAL CREATIVITY



Sound Synthesis · Oscillation and Waveforms · Wavetables · Samples · Filters · Distortion · Resonators · Amplification · Effects · Modulations · Envelopes · LFOs · AM · RM · FM · PD · Arpeggiators · Sequencers · Modularity · Jacks · Racks · Voltages · Gates · Clocks · Triggers · CVs · Interfaces · MIDI · Analog · Digital · Hybrid · & Software-Designs · Polyphony · Literature · Rhythm Sources · and much more. + Interviews with manufacturers and artists

303 pages · language: english · paperback · countless images and illustrations in colour · ISBN 978-3-00-064145-9

ONLY
29,99 €



ORDER NOW!

www.synthesizermagazin.de