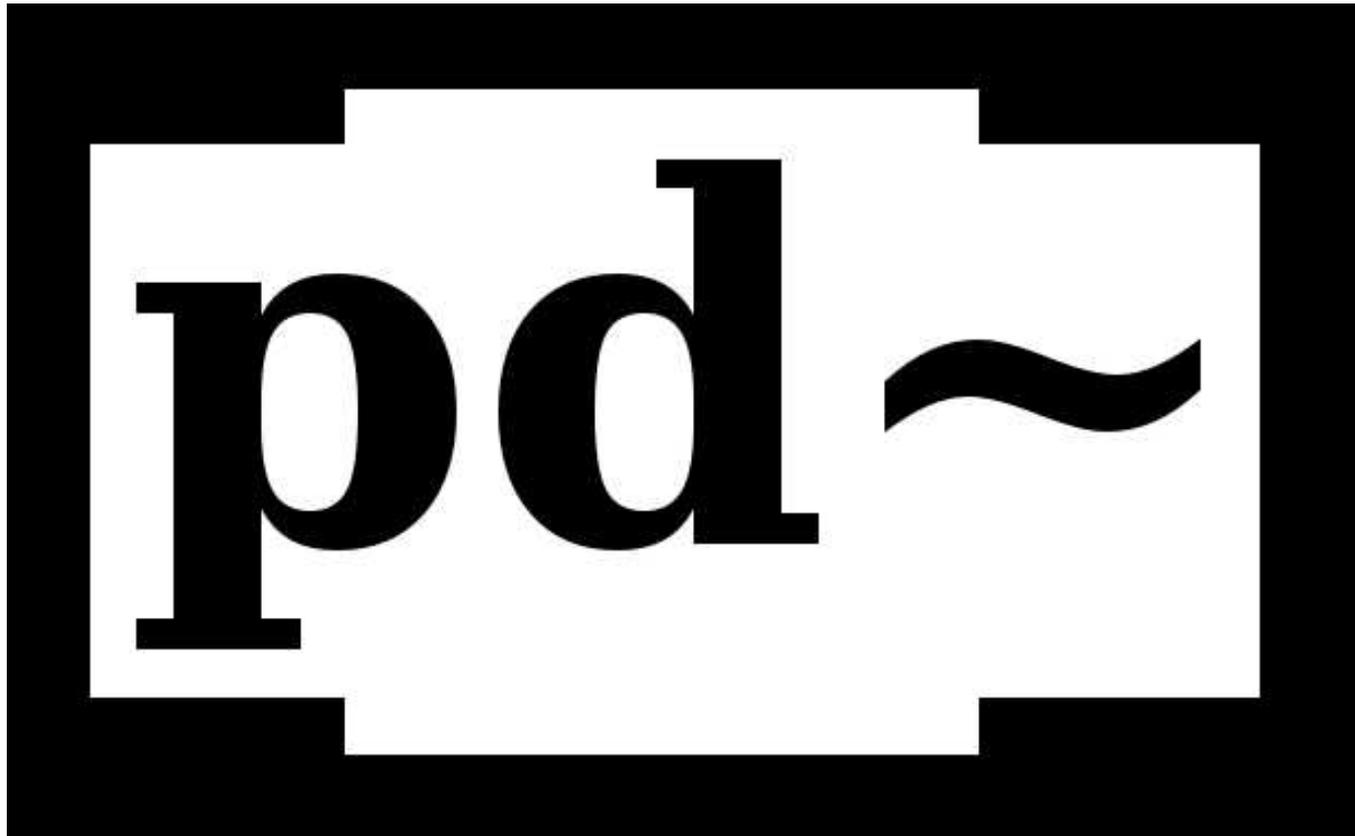


Pure Data

Eine „visuelle“ Programmiersprache von Miller Puckette für
Multimedia



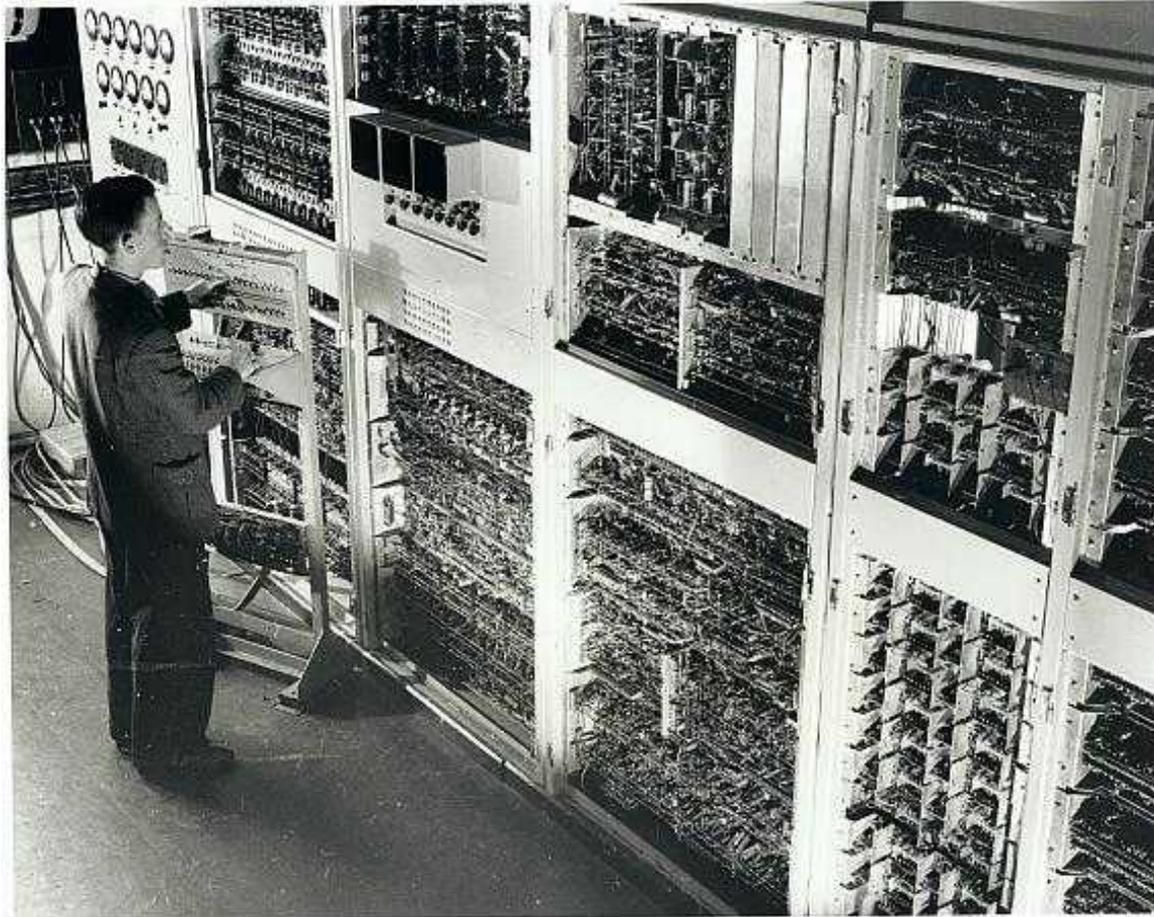
How comes?

Computermusik

-erste dokumentierte digitale Klangerzeugung 1950/51 am CSIRAC
(***C**ouncil for **S**cientific and **I**ndustrial **R**esearch **A**utomatic **C**omputer*)
in Australien.

-Geoff Hill programmierte am CSIR Mk 1 eine klangerzeugende Software.

CSIR Mk 1



Audio?

- Feedback um abgeschlossene Prozesse zu signalisieren.
- Klicken / Bleeps
- Es gibt (leider) KEINE Aufnahmen davon.

Erste Aufnahmen

- England 1951

- Eine BBC Kinderproduktion:

God Save The King, Ba Ba Black Sheep, In The Mood *1

- Computer: Ferranti Mark I

Ferranti Mark II



Magnetic Tape !

Grundlagen der digitalen Klangerzeugung

Idee:

**Computer berechnen Audiodateien und schreiben diese auf Magnetband.
(1957=>LANGSAM!)**

Das Ergebnis wird in der gewünschten Abspielzeit (Audio-Rate) wiedergegeben und danach durch einen Digital-Analog-Wandler (DAC) in Spannung übersetzt. Diese kann durch analoge Wiedergabesysteme hörbar gemacht werden.

Max Mathews

- Programmierer an den Bell Telephone Labs
- Sprachsynthese
- Entwickelt 1957 die Programmiersprache „Music 1“.
- 1 Timbre (Klangfarbe, Triangle), 1 Stimme.
- Newman Guttman schreibt im selben Jahr das erste Stück für Music 1.
„The Silver Scale“ (ca.20 sek.) *2

Music 2

- 1958 +/-
- 4 Stimmen
- Wavetable Synthese!

UNIT GENERATORS

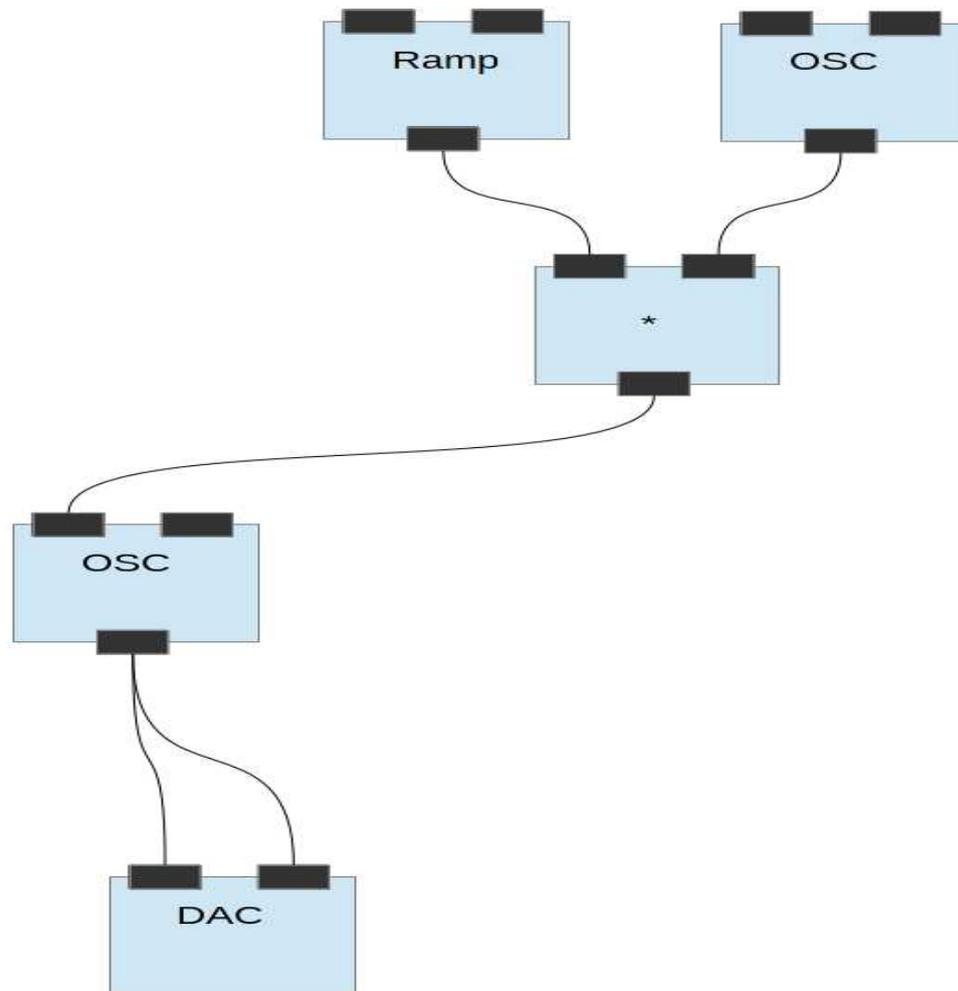
Music 3

Unit Generator Konzept

Virtuelle Klangerzeuger oder Steuerobjekte können modular in Reihe geschaltet werden.

Dadurch können komplexe Systeme erzeugt werden.

Unit Generator Schaltung



Miller Puckette



Miller Puckette

- Autor Pure Data
- Mathematiker, Musiker.
- Schreibt etwa 1985 Max am IRCAM (*Institut de Recherche et Coordination Acoustique/Musique*). Der Name lehnt an Max Mathews an.
- Sinngemäß wird das Prinzip der „Audio-Units“ übernommen und visuell dargestellt.
- Max war eine Interaktionsfläche für Komponisten um Computersysteme für Komponisten zugänglich zu machen.
- Erste Komposition „Pluton“ von Phillipe Manourie ca. 1988. *₃

Pure Data

- **Miller Puckette schreibt Pure Data etwa 1994/96.**
- „PD“ orientiert sich stark am Vorbild Max.
- Visuelle Programmierung
- Audio-Units (hier „Objekte“)
- Datenstrom-Orientiert
- „Echtzeit“
- PD ist „Open Source“ und wird kontinuierlich weiterentwickelt.
- Aktuelle Version ist 0.43.4
- Plattformen Linux, Mac, Windows.

u.v.a

Pure Data

Konfigurierung

- Soundkarten
- MIDI
- „Path“
- „Startup“

Oberfläche

- Öffnen eines neuen Projekts.
- Help Browser.

Bedienung

- „Edit-Mode“ Strg+E.
- Mouse, Keyboard

Grundbausteine

- - Bang
- - Objekt-Box
- - Message-Box
- - Number-Box
- - Kommentar

Inlets und Outlets

An allen erzeugten Objekten befinden sich sogenannte Inlets und Outlets.

Diese werden benutzt um die Objekte durch "Patchchords" zu Verbinden.

Hello World!

Um die verschiedenen graphischen Objekte miteinander zu verbinden werden „Patchchords“ verwendet.

Mit Hilfe der Mouse kann das Outlet der „Message Box“ mit dem Inlet von „Print“ verbunden werden

“Hello World!”(Message-Box

|

||Print| Das Objekt „Print“

Hello Noise!

Eine „Number-Box“ wird mit dem Objekt „osc~“ verbunden.
Letzteres mit dem Objekt „dac~“

```
/number/
```

```
|
```

```
|osc~|
```

```
|
```

```
|dac~|
```

Audio-Rate VS Control-Rate

-Um Rechenleistung zu sparen wird zwischen Audio und Kontrol-Rate Unterschieden.

-Audio Events werden in Audio-Rate berechnet, was in der Regel 44100 oder 48000 Samples pro Sekunde beträgt.

-Kontrol-Daten werden etwa einmal pro Millisekunde berechnet.

-Audio-Objekte werden durch eine Tilde ~ gekennzeichnet.
Audio-Patchchords sind etwas „breiter“.

-Kontrol-Objekte haben keine besondere Kennzeichnung.
Die Patchchords sind etwas „feiner“.

Help! Again...

Es ist möglich über ALLEN erzeugten Objekten, Messages o.ä.

Die rechte Maustaste zu drücken.

Es erscheint ein Popup-Window mit dem Vermerk Properties,
Open und Help.

- Über Help geht es zur Dokumentation des jeweiligen Objekts.
- Properties sind die „Grundeinstellungen“ des betreffenden Objekts.

Mathe

Pure Data bietet alle möglichen Objekte an um mathematische Aufgabenstellungen in Echtzeit zu berechnen.

Von Arithmetik (+,-,* usw) über höhere Mathematik (Sinus, Cosinus, Logarithmus....usw) bis zu komplexen Formeln (das |expr| Objekt) enthält PD umfangreiche Möglichkeiten.

Besondere Aufgabenstellungen lassen sich durch „Extensions“ lösen.

Ein „einfacher“ Synthesizer

Ein simpler monophoner, midiststeuerbarer Synthesizer.

Verschiedene Syntheseformen

-

Additive Synthese

Oszillatoren werden geschichtet (addiert...) um ein obertonreiches Signal zu erzeugen.

Substraktive Synthese

Ein obertonreiches Signal (Rechteck/Sägezahn/usw) wird durch „Filter“ im Frequenzspektrum moduliert.

Frequenzmodulation FM

Die Eingangsfrequenz eines Oszillators wird durch einen weiteren moduliert. Dadurch entstehen neue Obertöne.

Wavetable Synthese

- Eine mathematisch vordefinierte Wellenform wird in einem Speicher abgelegt. (Music 2)
- Die „übliche“ Art in der Computermusik.

Sampling

In einem Speicher abgelegte Audiodaten werden ausgelesen.

(Ähnlich der Wavetable-Synthese)

Dies ist relativ Speicheraufwendig!

44100 „Samples“ a 16 Bit Datentiefe (65.536 Möglichkeiten)
pro Sekunde!

Anything goes!

- Granularsynthese
- Spectralsynthese
- Physical-Modelling
- FFT Synthese (Fast-Fourrier)
- Generative/Interaktive Musik
- Interaktion durch HID's
- Arduino (Sensoren...)