

nogo0.1

Live-Komposition für Computer und ein Konzert

Georg Holzmann

Feb. 2005, Graz

nogo0.1 ist eine Komposition für das Ende eines Konzerts. Während dem Konzert werden in regelmäßigen Abständen 50-80 zufällige Samples der anderen Kompositionen und Pausen aufgenommen. In nogo0.1 werden diese Klänge nun recycled, verdichtet und aus ihrem linearen zeitlichen Kontext herausgerissen.

1 Konzept

1.1 Gleichzeitigkeit

In der gesamten Komposition werden viele der vorherigen Kompositionen gleichzeitig verarbeitet. Sie werden verdichtet.

Damit wird versucht den linearen Hör-, Erzählmodus aufzulösen, indem nicht mehr das Nacheinander, sondern das Übereinander in den Mittelpunkt gelangt.

Außerdem entsteht dadurch eine gewisse Unsicherheit des Fokus: es existiert eine Vielzahl von möglichen Brennpunkten des Hörers. Man kann den eigenen Hörfokus auf verschiedene Teile der Klangmasse richten, seine "eigene" Komposition hören und immer wieder Elemente der vorherigen Kompositionen wiedererkennen.

1.2 Wiederholung

Durch das gleichzeitige Wiederholen bereits bekannter Klänge wird ebenfalls versucht die erzählende, narrative Form zu umgehen, bzw. den linearen Fortschritt der Zeit in der Musik in Frage zu stellen.

Diese Prinzip wird nun auf zwei verschiedenen Ebenen angewendet:

- durch die Wiederholung der Kompositionen des Konzerts wird die Idee des Fortschritts im Konzert (in der Musik überhaupt ?) in Frage gestellt
- durch die Wiederholung und Schichtung einzelner Teile der aufgenommenen Klänge wird der Fortschritt innerhalb von nogo0.1 in Frage gestellt

1.3 Zustand

nogo0.1 besteht aus 6 voneinander unabhängigen Zuständen.

In jedem Zustand spielen Teile aus verschiedenen zeitlichen Abschnitten des Konzerts mit verschiedenem Verdichtungsgrad miteinander.

Also Nicht-Erzählen, indem zuviel oder zuwenig passiert, indem zeitlich getrennte Teile gleichzeitig passieren.

2 Aufführung, technische Realisation

2.1 Mikrofonierung

Die Mikrofone werden in Mitten der Musiker (oder Lautsprecher) positioniert (siehe Abbildung 1).

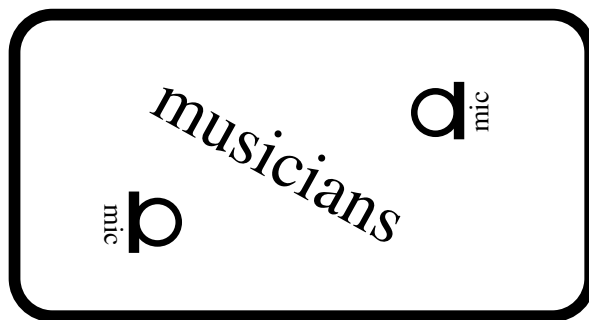


Abbildung 1: Beispiel einer Mikrofonierung mit 2 Mikrofonen

Dadurch entsteht eine fokussierte Aufnahme, d.h. manche Instrumente sind lauter, leiser, andere Klangfarbe, etc.

Falls vorhanden ist die Aufnahme auch mit mehreren Mikrofonen möglich.

2.2 Aufnahme der Klänge

Während des Konzerts werden nun in regelmäßigen Abständen an ca. 25-40 Zeitpunkten Samples zu je 20 Sekunden aufgenommen (d.h. mit 2 Mikrofonen erhält man 50-80 Samples) ¹.

Falls es im Konzert eine Pause gibt, wird die Aufnahme in dieser Zeit unterbrochen (zwischen den einzelnen Kompositionen entstehen schon genug Pausen).

2.3 Implementation

Die Komposition wurde unter Linux in *PD* ², eine grafische open-source Programmiersprache für Audio/Video Processing in der Tradition von MAX/MSP, implementiert (Interface siehe Abbildung 2).

Die Aufgabe des PD-Patches ist es, die einzelnen Samples der anderen Kompositionen aufzunehmen und leicht elektronisch zu verfremden. Während der eigentlichen Performance werden

¹Meine erste Realisierung: 2 Samples alle 4 Minuten, insgesamt: 2x30 Samples

²Pure Data: <http://www.puredata.org>

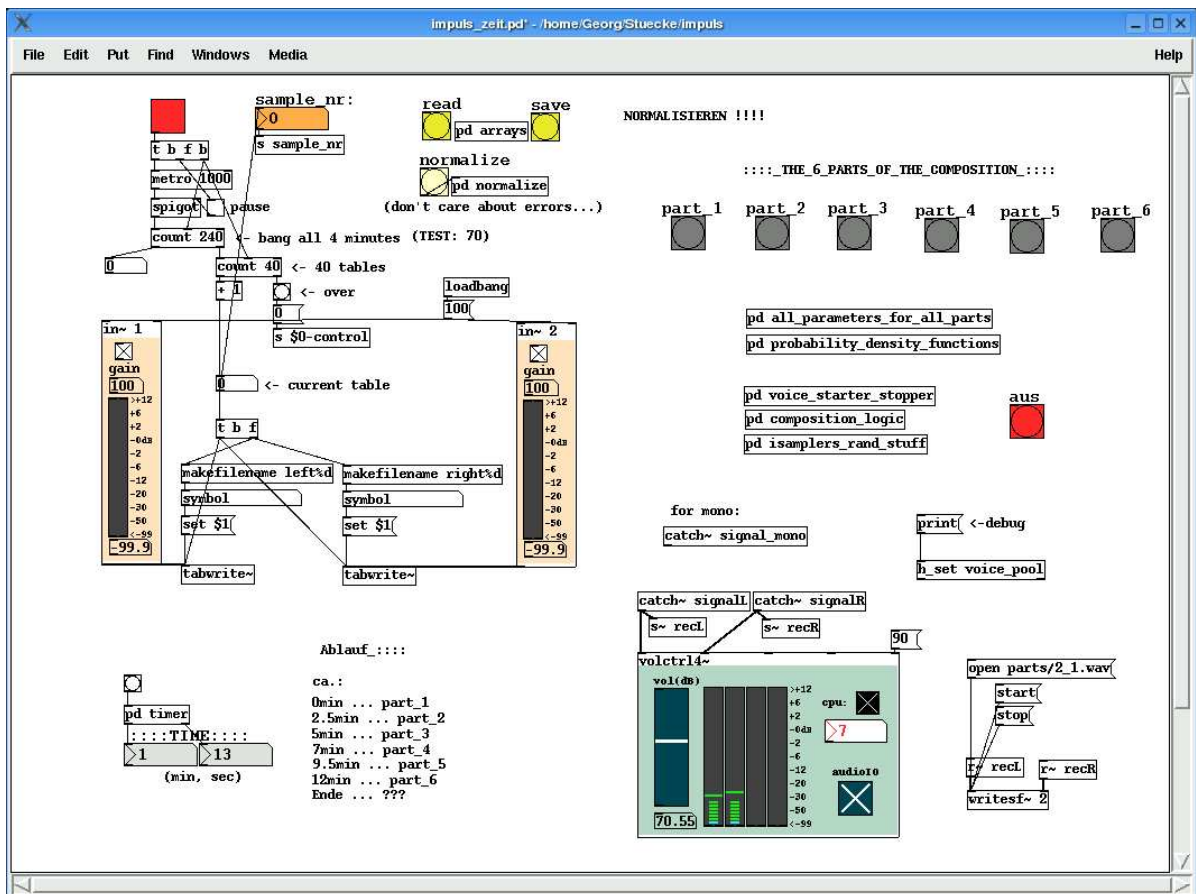


Abbildung 2: Screenshot des PD-patch

nun mit Hilfe von strukturgenerierenden Algorithmen und bestimmten Wahrscheinlichkeitsverteilungen verschiedene Teile des Konzerts miteinander kombiniert.

In Patcher-Programmiersprachen wie PD oder MAX/MSP ist es oft schwierig komplexere musikalische Strukturen zu generieren. Deshalb entwickelte ich die External-Library *PDContainer*³ in C++, welche die Verwendung verschiedenster Datenstrukturen und deren Kommunikation untereinander ermöglicht und somit die Implementation von algorithmischen Kompositionen erleichtern soll.

Der PD-patch (sozusagen die "Partitur") zu nogo0.1 ist frei downloadbar⁴ und kann unter Windows, IRIX, Linux oder MacOS X, da PD open-source und plattformunabhängig ist, reproduziert werden.

2.4 Sonstiges

Dauer der Performance: ca. 15 (+/- 2.5) Minuten

nähere Infos, Aufnahme und PD-patch: <http://grh.mur.at/projects/nogo01.html>

³PDContainer: <http://grh.mur.at/software/pdcontainer.html>

⁴PD-patch nogo0.1: <http://grh.mur.at/projects/nogo01.html>