

Notizen zu *Single Finger Mode* für Klavier und Computer

Simon Bahr

Single Finger Mode ist ein Stück für Klavier und Computer. Der*die Pianist*in spielt zusammen mit dem Computer auf demselben Instrument. Das Klavier muss daher mit einer MIDI-Schnittstelle (Ein- und Ausgang) ausgestattet sein. Das Stück ist algorithmisch komponiert, was es dem Computer ermöglicht die Genauigkeit der Live-Performance (Tonhöhen, Dauer, Dynamik, usw.) im Vergleich zur Partitur zu bewerten. Je besser der Computer die Leistung bewertet, desto mehr wird er gezähmt und passt sich der gespielten Musik an. Je weniger gut der Computer die Leistung bewertet, desto mehr erwacht er zu eigenem Leben: Er mag tonale Akkorde und Konsonanzen sehr und ist immer für etwas Spaß zu haben.

Technische Realisation

Das Stück wurde auf einem Linux-Computer mit *Steal Bank Common Lisp* und *Pure Data 0.50.2* realisiert. Die Aufführung erfordert das Lisp-Package *Slippery Chicken* von Michael Edwards¹ und die Pure Data Library *mrpeach* von IOhannes m zmölnig². Lisp und PD kommunizieren via *Open Sound Control*. Zur Aufführung ist die Datei *singlefingermode-*

perform.lsp zu laden und das PD-Patch zu starten. Midi-Ein- und Ausgang von PD sind mit dem Klavier zu verbinden. Das Stück wird im PD-Patch gestartet und gestoppt. Weitere Interaktionen mit dem Computer während der Aufführung sind nicht erforderlich.

Zur Komposition

In der Notation des Stücks wird nur eine vorab definierte Gruppe der Tasten der Klaviatur verwendet.³ Der Computer kann sämtliche Tasten der Klaviatur bedienen. Für jede seitens des*der Performer*in spielbarer Taste wird ein Regelsatz definiert. Die Regeln beinhalten (1.) welche Tasten dieser Taste folgen dürfen, (2.) welche Dauer die nächste Note haben wird, (3.) relativ zur n-ten letzten Note, (4.) welche Dauer die vorhergehende Pause haben wird, (5.) relativ zur n-ten vorhergehende Note und (6.) welche Dynamik die Note haben wird (7.) relativ zur n-ten letzten Note. Auf Basis dieser Regeln ist es möglich, sowohl in der Komposition als auch während der Performance die Eigenschaften der jeweiligen nächsten Note zu berechnen.⁴

³Alle übrigen Tasten werden in der Performance als ‚schwerwiegende Fehler‘ bewertet.

⁴Diese Berechnung ist immer relativ, kann in der Performance also auf absolute Zeit- und Velocitywerte,

¹<https://michael-edwards.org/sc/>

²<https://puredata.info/downloads/mrpeach>

Es werden bestimmte Grenzwerte für Dauern und Dynamiken festgelegt, die während der Komposition nicht über- bzw. unterschritten werden sollen.⁵ Da jede Tonhöhe einen bestimmten Möglichkeitsraum an Folgetönen – und damit auch Dauern, Dynamiken, Pausen – mit sich bringt, können verschiedene Kandidaten ermittelt werden, die auf das jeweils vorherige Ereignis folgen können. Sofern sich sämtliche Parameter eines Folgekandidaten innerhalb der Grenzwerte bewegen, wird er in Betracht gezogen. Kommt kein Kandidat zustande, dann endet das Stück. Es endet außerdem sobald 1000 Töne erzeugt wurden.

Während der Kompositionsalgorithmus in einer Schleife die Töne des Stücks ermittelt, wird für jede Tonhöhe die Häufigkeit ihres Auftretens mitgezählt. Bisher weniger häufig vorgekommene Tonhöhen werden präferiert. Dies ist ein entscheidender Aspekt der Komposition, der verhindert, dass sich eine seriell anmutende, immer gleiche ‚Reihe‘ bildet, die sich stetig wiederholt. So entstehen bspw. die zwei Stellen im Stück, in denen wenige Töne immer wieder direkt aufeinander folgen: Sie sind durch Regeln im Verlauf des Stücks weniger wahrscheinlich zu erreichen. Werden sie dann erreicht, so ‚verfängt‘ sich der nach Gleichberechtigung strebende Algorithmus bei ihnen und wiederholt sie so lange, bis Ausgleich geschaffen ist und das Gefüge in eine wieder neue Richtung kippt.

Der kompositorische Eingriffspunkt nach Definition des beschriebenen Prozesses ist der Regelsatz selbst: Er ist zwar ebenfalls algorithmisch erzeugt, doch sind die konkreten Wer-

te, die die Basis dieses Algorithmus bilden, in einer Experimentierphase ermittelt worden, in der immer neue mögliche Versionen des Stücks erzeugt und bewertet wurden. Diese Arbeitsweise bringt mit sich, dass die finale Version zwar einerseits eine bewusste kompositorische Entscheidung ist, andererseits die Einzelteile des Stücks nicht zu isolieren sondern nur im Paket zu haben sind. Es ist in gewissem Sinne ein produktiver Kompromiss zwischen persönlichem Geschmack und algorithmischem Prozess.

Die Rolle des Computers in der Aufführung

Die Regeln für eine bestimmte Taste bleiben während des gesamten Zeitraums statisch. Infolgedessen benötigt der Computer keine Informationen über die aktuelle Position in der Partitur. In der Probe kann jeder Abschnitt des Stücks in beliebiger Reihenfolge wiederholt werden, ohne dass Änderungen im Patch erforderlich sind.⁶

Während der Aufführung sind Midi-In- und Output mit dem Computer verbunden. Der Computer liest das eingehende Midi-Signal (via Pure Data), analysiert es entsprechend der auch dem Kompositionsalgorithmus zugrunde liegenden Regeln (via Common Lisp/Slippery Chicken) und errechnet aus dem Durchschnitt der Trefferquoten für die Parameter (1.) grundsätzlich mögliche Taste, (2.) an dieser Stelle korrekte Taste, (3.) Tondauer, (4.) Dauer der vorhergehenden Pause, (5.) Dynamik

in der Komposition auf Notenwerte und dynamische Symbole angewandt werden.

⁵Extrem lange bzw. kurze Töne unterhalb einer 64-tel bzw. überhalb einer ganzen Note, sowie extrem laute bzw. leise Töne unterhalb von ppp bzw. überhalb von fff dürfen nicht vorkommen.

⁶Der Computer muss sich allerdings kurz ‚gewöhnen‘ und der aktuellen Position anpassen, bis die ersten Töne im Buffer wieder zu korrekten Ergebnissen führen können. Die ersten Noten können als falsch angesehen werden und der Punktstand vorübergehend etwas abnehmen.

und (6.) Zustand des Pedals einen Punktestand (*Score*). Der vorausgegangene Punktestand fließt ebenfalls in die Berechnung mit ein.

Auf Basis eines Buffers, der die zuletzt gespielten Töne enthält, wird parallel dazu versucht, eine aktuelle Tonart zu ermitteln.⁷ Auf dem Grundton dieser Skala werden einige fest eingespeicherte Akkorde (*Best-of Tonal Chords*) wiederum mit dem Tonbuffer abgeglichen. Hierbei spielen die Töne nun eine absteigende Rolle je weiter hinten sie sich im Buffer befinden. Die Akkorde wechseln also schneller und enthalten mit hoher Wahrscheinlichkeit den aktuellen Melodieton. Die Töne der so generierten Akkorde werden nun auf verschiedene Oktaven verteilt, je nach (1.) Oktave des Melodietons, (2.) Nähe zum entsprechenden Ton des vorausgegangenen Akkords und (3.) eines festen Bereichs für den jeweiligen Ton (z.B. Grundton unten, etc.). So wird mit jedem Anschlag ein (meistens sechstöniger) Akkord ermittelt.

Die Dauer, die Anfangszeit und die Dynamik der auszulösenden Klänge werden auf wesentlich einfachere Weise berechnet, da sie sich sehr stark am Eingangssignal orientieren sollen. So wird ein enger Zusammenhang zwischen den von dem*der Performer*in und dem Computer gespielten Tönen angestrebt. Dennoch hängen auch diese Parameter vom aktuellen Punktestand ab und verhalten sich eigenständiger je niedriger dieser ist.

Wenn die Punktzahl stark sinkt (im Verhältnis zum Durchschnittszustand), wird ein sog. *Special Program* ausgelöst.⁸ Der Computer ver-

hält sich dann kurzfristig so, wie es ihm gefällt, ohne oder mit nur geringem Einfluss durch die eingehenden Daten. Das zu wählende *Special Program* wird jedes Mal geändert, wenn das Pedal gedrückt wird. Es gibt auch ein spezifisches *Special Program*, das bei gedrücktem Pedal verwendet wird.

Zu Notation und Umsetzung

Single Finger Mode ist vor allem nach zwei Gesichtspunkten komponiert: Zum einen soll die radikal algorithmische Arbeitsweise die Umsetzung des Konzepts, d.h. die Bewertbarkeit des Gespielten durch den Computer, ermöglichen. Zum anderen soll die Komplexität des Stücks den*die Performer*in dazu bringen, dem Computer mittels notwendiger Schwankungen im Punktestand verschiedene musikalische Welten zwischen ‚eher tonal‘ – ‚entfernt tonal‘ – ‚atonal‘ zu entlocken. Aus diesem Grund ist die Partitur an einigen Stellen nur mit Einschränkungen bzw. Schwerpunktsetzungen auf bestimmte musikalische Parameter spielbar. Es geht nicht darum, die notierten Rhythmen, etc. immer exakt abzubilden, sondern eher um die Entscheidungen, die seitens dem*der Performer*in getroffen werden, um dieser ‚Anleitung zum Bezwingen des Computers‘ zu folgen. Es ist außerdem ausdrücklich gewährte Freiheit, die Notation ganz, nur in Teilen oder gar nicht zu berücksichtigen. Gerahmt wird das Stück vordergründig durch den Computeralgorithmus, weniger durch die Notation.

⁷Der Algorithmus geht alle Durtonarten durch, berechnet, wieviele Töne jeweils auf der Skala liegen, und wählt das beste Ergebnis aus. Bei gleicher Qualifikation wird die Tonart mit den wenigsten Vorzeichen bevorzugt.

⁸z.B.: *High Flicker*, *Major Madness*, *Groovin' 'till diz-*

zy und *The Demo Song*.