



Fakultät Informatik, Mathematik und
Naturwissenschaften
Studiengang Informatik Master

Projektarbeit zur Vorlesung Computermusik

BrandtBrauerFrick.hs

Autoren: Nico Mehlhose, Raphael Drechsler
Abgabedatum: 01.02.2019

1 ABSTRACT

Raph:

TODO Wirkung? Klar machen!

BrandtBrauerFrick.hs

Brandt Brauer Frick ist ein Techno-Projekt aus Berlin. Die Basis des Projekts bilden Klänge aus dem Instrumentarium der klassischen Musik, welche anfangs gesampelt, später in einem zehnköpfigen Ensemble auch live vorgeführt wurden.[1]

Ziel des Projektes:

Die Umsetzung des Songs "Pretend" von Brandt Brauer Frick entweder in Tidal oder Euterpea. Dabei Orientierung an der Live-Aufführung (<https://www.youtube.com/watch?v=KCpLXpMB7F8>).

Herausforderungen:

- Evaluation ob Tidal oder Euterpea genutzt werden soll:
- Untersuchung der Frage ob klassische Klänge am ehesten in Euterpea oder Tidal nutzbar sind. (Durch repetitiven Charakter des Liedes würde sich Tidal zur Live-Vorführung eignen)
- Analyse der einzelnen musikalischen Bausteine und deren Implementierung.

2 TIDAL ODER EUTERPEA

Nico

Dieses Thema soll sich um die Evaluation zwischen Tidal und Euterpea handeln. Unsere Entscheidung Tidal zu nehmen beruht gewiss nicht auf einer zufälligen Entscheidung. In diese Entscheidung ist der Programmieraufwand, vorhandenen Informationen und die Möglichkeit den Synthesizer zu erweitern. Bei dem Programmieraufwand wird sehr schnell klar, dass durch das Lied *Pretent* von BrandtBrauerFrick Tidal besser geeignet ist als Euterpea. Der erste Gesichtspunkt der betrachtet wurde ist die Repetitivität des Songs, welcher in Euterpea zwar auch umsetzbar ist aber in Tidal von Anfang an gegeben ist, da Tidal die Sounds immer in einem Loop abspielt. Bei den vorhandenen Informationen stellt sich heraus, dass es keine Offiziellen Notenblätter für das Lied Online gibt, wodurch Euterpea etwas an Bedeutung verliert, da Euterpea für genaue Notenbestimmungen perfekt geeignet wäre. Da dieser Fakt aber nicht vorliegt, kann das selbe Maß an Genauigkeit auch mit Tidal erreicht werden. Der letzte und für uns wichtigste Punkt war die Erweiterbarkeit der Sounds. Die Wichtigkeit darin besteht in der entfremdeten Benutzung der Musikinstrumente in dem Lied. In Euterpea haben wir nach einiger Recherche keinen weg gefunden Sounds hinzuzufügen um diese später zu verwenden. In Tidal allerdings existiert diese Möglichkeit mittels des Befehl . Mit diesem Befehl lässt sich ein Verzeichnis in Tidal integrieren.

3 ANALYSE DES STÜCKS PRETEND – ALLGEMEIN

Raph.

TODO: bpm, Tonart, Score bzw. Logic-Screenshot → wann was?

Tempo: 130bpm.

setcps (130/60/4)

4 ANALYSE DES STÜCKS PRETEND

Im folgenden Abschnitt sollen die zehn Instrumentalisten untersucht werden.

4.1 Instrument 0: Was ist pro Instrument TODO?

TODO: Nach Bearbeitung Hilfskapitel entfernen.

Raph.

Welche Figuren? - Welche Wirkung? - Welche Noten?

Nico

Wie klingt das Instrument?

- Wie klingt das live? Einzelne Bestandteile? (Marimba gespielt mit Holzsticks und verschiedene Kuhglocken) - Wie klingt das in welcher Figur? (zB. BD laut, leise) - Welchen Klang wählen (evaluation - SD-Instrument nutzbar?, WAV suchen/selber aufnehmen, Instrument coden)

4.2 Instrument 1: Schlagzeug

4.2.1 Figuren

WIP

Figur 1

treibender Grundrhythmus, steigende Lautstärke BD



Abbildung 1: Schlagzeug Figur 1

```
d1 $ sound "[bd hh]*4"
```

```
Nicos Code: d1 $ whenmod 200 100 (|*|gain "0.8 0.7 0.8 0.7 0.8 0.7 0.8 0.7" ) $
             sound "[bd hh]*4" # gain "0.5 0.7 0.5 0.7 0.5 0.7 0.5 0.7" # midinote 58
```

Figur 2

Wie 1 mit Fills

TODO Logic, Noten, Code oder Absicht und Programmlogik

Figur 3

Wie 1 und 2, noch mehr Fills

TODO Logic, Noten, Code oder Absicht und Programmlogik

Figur 4

Wie 1, keine Fills auf hh, zyklische Bewegung auf rim: Angenommen 5Schläge Pro Takt.

```
d1 $ sound "[[bd hh]*4, rm*5]"
```

```
d1 $ sound "[[bd hh]*4, rm rm rm rm ~]"
```

um einen Cycle draus zu machen rand nutzen:

```
d1 $ sound "[bd hh]*4, rm rm rm rm rm?]"
```

oder

```
d1 $ slow 2 $ stack [
sound "[bd hh]*8",
fastcat[sound "[rm rm rm rm rm]", sound "[rm rm rm rm ~]"]
]
```

Figur 5

Nur Triolen auf Rim.

```
d1 $ slow 2 $ fastcat[sound "[rm rm rm rm rm]", sound "[rm rm rm rm ~]"]
```

Figur 6

Wie Figur 4 aber kräftig gespielt.

Figur 7

Treibender Rhythmus, viele Fills -> Was analoges zu Figur 3

TODO Logic, Noten, Code oder Absicht und Programmlogik

Im Ablauf: **Block mit Figur 6**

Figur 8

Wie 4 aber ohne BD.

```
d1 $ slow 2 $ stack [
sound "[~ hh]*8",
fastcat[sound "[rm rm rm rm rm]", sound "[rm rm rm rm ~]"]
]
```

Im Ablauf: **Block mit Figur 6**

4.2.2 Klangbild

Töne: hh, bd (dumpf, wenig knackig)

bd wird lauter

rim

4.3 Instrument 2: Pauken

4.3.1 Figuren

OFFEN

TODO (Hierzu Studio-Version hören)

4.3.2 Klangbild

3 Kesselpauken

4.4 Instrument 3: Marimba

4.4.1 Figuren

OFFEN**TODO**

Aufbau beschreiben mit Glocken.

Random-Funktion benötigt

4.4.2 Klangbild

Holzsticks auf Marimba in verschiedenen Tonhöhen wobei eher rhythmisch als melodisch eingesetzt, dazu Kuh-Glocken bereitstellen für Random-Funktion

4.5 Instrument 4: Tuba**4.5.1 Figuren****WIP****Figur 1**

Figur über einen Takt. Schlag auf Tuba-Mundstück als rhythmisches Element auf zweite Zählzeit im Takt.

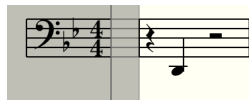


Abbildung 2: Tuba Figur 1

```
d1 $ sound "[~ sn ~ ~]"
```

```
Code Nico: d1 $ sound "~ bd ~ ~" # midinote 15
```

4.5.2 Figuren**Figur 2**

Figur über 2 Takte. Instrumentalist bläst in die Tuba ohne dass die Lippen vibrieren, um ein Rauschen zu erzeugen. Pause am Ende der Figur als Atempause angenommen.



Abbildung 3: Tuba Figur 2

```
--Idee: sound, der 2 Takte dauert alle 2 Takte 1x anspielen
```

```
d1 $ sound "blasesoundTuba"
```

Nico: geht nicht da kein sound mit gleichbleibender schwingung

Figur 3

Wie Figur 1, hier allerdings kurzes tonloses Pusten stoßweise gespielt anstelle von Schlag auf Mundstück.

Figur 4

Tiefe Töne durch Tuba, Tonhöhe nicht entscheidend und fast nicht mehr wahrnehmbar. Gefühl von Bedrohung. Rollt langsam an

TODO Logic, Noten, Code

Figur 5

Wie Figur 4, kräftig ausgespielt.

Figur 6

Wie Figur 5, maximal kraftvoll ausgespielt. Eine Oktave höher gespielt daher Tonhöhe der einzelnen Töne gut erkennbar.

TODO Code

4.5.3 Klangbild

schlagen, blasen(impulsartig,durchgehend), spielen(tief, hoch)

4.6 Instrument 5: Posaune

4.6.1 Figuren

FERTIG

Figur 1

Figur über einen Takt. Kurzes, tonloses Pusten in die Posaune. Stoßweise gespielt als rhythmisches Element auf letzte Achtelnote im Takt.



Abbildung 4: Posaune Figur 1

```
d1 $ sound "[[][[[~ sn]]]"
```

Figur 2

Erzeugen von Rauschen analog zu Figur 2 - Tuba. Dabei Lautstärke zum Ende des Stückes hin zunehmend.

```
--Idee: sound, der 2 Takte dauert alle 2 Takte 1x anspielen
--Frage: Lautstaerke?
d1 $ sound "blasesoundPosaune"
```

4.6.2 Klangbild

blasen(impulsartig,durchgehend)

4.7 Instrument 6: Violine

4.7.1 Figuren

OFFEN

...

4.7.2 Klangbild

Ruhige Parts, Hektik (schnell gespielte Töne)

4.8 Instrument 7: Chello

4.8.1 Figuren

OFFEN

(Hierzu Studio-Version hören)

4.8.2 Klangbild

Ruhige Parts, Hektik (schnell gespielte Töne)

4.9 Instrument 8: Harfe

4.9.1 Figuren

vorerst FERTIG

Figur 1

TODO Starke Annahme - hier vllt was mit Random aus Skala nehmen machen.



Abbildung 5: Harfe Figur 1

```
d1 $ slow 2 $ stack [
  midinote "[74 72]*4" # s "gtr",
  midinote "[62 60]*4" # s "gtr",
  slow 2 $ fastcat [
    midinote "70 69" # s "gtr",
    midinote "67" # s "gtr",
    midinote "70 69" # s "gtr",
    midinote "66" # s "gtr"
  ]
]
```

4.9.2 Klangbild

...

4.10 Instrument 9: Flügel

4.10.1 Figuren

WIP

Figur 1

Nur ein Tiefer Ton.



Abbildung 6: Flügel Figur 1

TODOCode

Figur 2
TODOLogic, Noten, Code

Figur 3
TODOBeschreibung
TODOCode



Abbildung 7: Flügel Figur 3

Figur 4
TODO Beschreibung

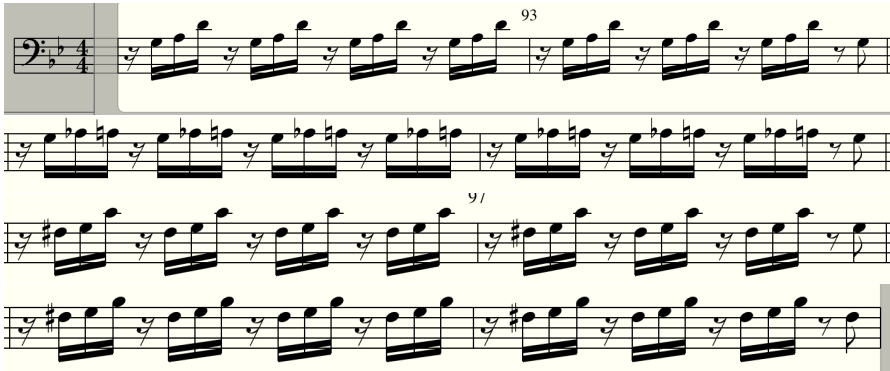


Abbildung 8: Flügel Figur 4

```
d1 $ slow 8 $ fastcat [  
midinote "~ 67 69 74 ~ 67 69 74 ~ 67 69 74 ~ 67 69 74 ~ 67 69 74 ~  
67 69 74 ~ ~ 67 ~ " # s "superpiano",  
midinote "~ 67 68 69 ~ 67 68 69 ~ 67 68 69 ~ 67 68 69 ~ 67 68 69 ~ 67 68 69 ~  
67 68 69 ~ ~ 67 ~ " # s "superpiano",  
midinote "~ 66 67 72 ~ 66 67 72 ~ 66 67 72 ~ 66 67 72 ~ 66 67 72 ~ 66 67 72 ~  
66 67 72 ~ ~ 67 ~ " # s "superpiano",  
midinote "~ 66 67 70 ~ 66 67 70 ~ 66 67 70 ~ 66 67 70 ~ 66 67 70 ~ 66 67 70 ~  
66 67 70 ~ ~ 66 ~ " # s "superpiano"  
]
```

OK

Figur 5



Abbildung 9: Flügel Figur 5

```
d1 $ slow 2 $ midinote "~ 67 69 74 ~ 67 69 74 ~ 67 69 74 ~ 67 69 74 ~ 67 69 74  
~ 67 69 74 ~ 67 69 74 ~ ~ 67 ~ " # s "superpiano"
```

OK

4.10.2 Klangbild

...

4.11 Instrument 10: Moog Synthesizer

4.11.1 Figuren

vorerst FERTIG

Figur 1

Basslauf über 8 Takte.



Abbildung 10: Moog Figur 1

```

--Arbeitsstand
--2Takte
d1 $ midinote "[[55 55][54 55 ~ ~]]" # s "moog" # cut 1
--2Takte
d1 $ midinote "[[50 50][49 50 ~ ~]]" # s "moog" # cut 1
--2Takte
d1 $ midinote "[[48 48][47 48 ~ ~]]" # s "moog" # cut 1
--1 Takte
d1 $ midinote "[[58 57][56 57 ~ ~]]" # s "moog" # cut 1
--1 Takte
d1 $ midinote "[[57 ~][56 57 ~ ~]]" # s "moog" # cut 1

---Verbunden:
d1 $ slow 8 $ fastcat [midinote "[[55 55][54 55 ~ ~]]*2" # s "moog" # cut 1,
midinote "[[50 50][49 50 ~ ~]]*2" # s "moog" # cut 1,
midinote "[[48 48][47 48 ~ ~]]*2" # s "moog" # cut 1,
midinote "[[58 57][56 57 ~ ~]] [[57 ~][56 57 ~ ~]]" # s "moog" # cut 1
]

```

Figur 2

Basslauf über einen Takt.



Abbildung 11: Moog Figur 2

```

--Arbeitsstand
d1 $ midinote "[[[[50 ~ ~ 50]][~ 50]][55 57 60 58]]" # s "moog" # cut 1

```

4.11.2 Klangbild

...

5 OPTIONAL: GESANG

TODO: Audacity-Filter gegen Stimme

Kür: Ggf. selbst Sample erzeugen

6 PERFORMANCE

TODO:

Spuren per Stack verbinden?

--10 Instruments:

```
p "i1" $ sound "bd sd:1"
p "i2" $ sound "hh hh hh hh"
p "i3" $ sound "hh hh hh hh"
p "i4" $ sound "hh hh hh hh"
p "i5" $ sound "hh hh hh hh"
p "i6" $ sound "hh hh hh hh"
p "i7" $ sound "hh hh hh hh"
p "i8" $ sound "hh hh hh hh"
p "i9" $ sound "hh hh hh hh"
p "i10" $ sound "hh hh hh hh"
```

Konzept für Ablauf Performance?

Wie mehrtaktige Figuren mit every zusammenfassen?

LITERATUR/QUELLEN

- [1] Wikipedia - brandt brauer frick. https://de.wikipedia.org/wiki/Brandt_Brauer_Frick. Zugriff: 15.12.2018.