HAW HAMBURG

IT-Systeme Dokumentation

Interaktive Videoinstallation mit granularem Synthesizer

22. Februar 2024

Gruppe:	Ariane Bachmann	2552756
	Benjamin Ghodsi-Moghaddam	2582359
	Bruno Bühler	2625322
	Dennis Jonca	2175314
	Evan Tanggo Peter Simamora	2332397
	Fabian Brunner	2600389
	Rafael Weber	2623881

Studiengang: Meidentechnik B.Sc. WS 23/24

eingereicht bei: Malte Sanders

Inhaltsverzeichnis

1 Einleitung

Das Projekt „VisuSynth“ des Kurses IT-Systeme ermöglicht eine Interaktion zwischen Klängen, Bildern und der eigenen Kreativität zu erschaffen.
Mit einer Kombination aus Software und Hardware sollte eine Möglichkeit geschaffen werden, Töne live aufzunehmen, zu bearbeiten und in visuelle Bewegungen umzuwandeln.

Das Ziel war es, dass auch Unerfahrene im Bereich Audio-Synthese die Möglichkeit haben, durch eine verständliche Bedienung, das Projekt zu verstehen, zu bedienen und eigene kreative Ideen entwickeln zu können.

2 Grundlagen

Der Bereich Audio-Synthese sowie die Schnittstelle MIDI waren nicht dem ganzen Projektteam bekannt. Deshalb war es wichtig, Informationen über diese Bereiche zu sammeln, damit auf einem ähnlichen Wissensstand gearbeitet werden konnte.

2.1 Granular-Synthese

Die Granular-Synthese ist eine Synthese-Form wie z.B. auch die Subtraktive Synthese, FM-Synthese (Frequenzmodulation) oder auch die Wellenform Synthese, die zur musikalischen Klangerzeugung dienen. Einige diese Formen haben sich dabei stark oder weniger stark durchgesetzt. Es dabei wichtig zu unterscheiden, ob analoge oder digitale Synthese verwendet wird.

In den 1960er Jahren wurden die ersten analogen Synthesizer vorgestellt. Am einflussreichsten war dabei der Ingenieur Robert Moog, der bis heute als Pionier in der Welt der Synthesizer und elektronischen Musik gilt. Moog Synthesizer sind bis heute beliebt. Die Klangerzeugung basiert dabei auf die Verschaltung von elektronischen Bauteilen. Mit einer Weiterentwicklung von Computern und der Erfindung erster serienmäßig produzierbaren Mikroprozessoren, kam es auch zu Erfindung erster Synthesizer, die auf digitaler Synthese beruhten. Einen Meilenstein legte dabei der Yamaha DX7. Ein Algorithmus auf einem Mikroprozessor stellt hier per FM-Synthese den Klang her.

Die Granular-Synthese beruht darauf, ein Sample(„Audioschnipsel“) in kleine „Grains“ zu unterteilen, die zwischen 1ms – 50ms lang sind. Diese können dann in verschiedenen Tonhöhen, Geschwindigkeiten oder an verschiedenen Positionen abgespielt werden. Dies sorgt häufig für atmosphärische oder auch experimentelle Klänge. Es benötigt zur granularen Synthese ein vorher aufgenommenes oder gespeichertes Sample.

2.2 Umsetzung

Um diese Synthese-Form zu benutzen war es auch wichtig zu unterscheiden, wie man digitale Synthese umsetzen kann. Eine Möglichkeit wäre gewesen, einen DSP zu verwenden, auf dem der Algorithmus dann läuft, ähnlich wie beim Yamaha DX7. Dies setzte tiefgehende Programmierkenntnisse, sowie die Fähigkeit einen solchen Algorithmus erstellen zu können, Voraus. Eine andere Möglichkeit war es, eine schon vorhandene Umgebung/ Softwareanwendung zu verwenden, die die Synthese dann auf der CPU des PC's/Laptops durchführt. Zusätzlich haben viele dieser Umgebungen auch vorgefertigte Module und Anleitungen, um eigene Synthesizer zu bauen.

Mögliche Tools sind dabei Max for Live (Ableton, Cycling'74), Reaktor (Native Instruments) oder SunVox (Alexander Zolotov).

2.3 MIDI

MIDI („Musical Instrument Digital Interface“) ist eine Schnittstelle, die zur Übertragung von musikalischen Informationen dient. Dabei gibt es verschiedene Datentypen, die über MIDI übertragen werden. Für das Projekt waren vor allem die Werte „Control Change“ sowie „Note On“ und „Note Off“ wichtig.

3 Konzeption

Die Besucher betreten einen abgedunkelten Raum, in dessen Mitte sich, beleuchtet von zwei Scheinwerfern, ein Stehtisch befindet, der die notwendigen Bedienelemente beherbergt. Über einen Projektor werden zunächst Standby-Versionen der Visuals an eine Leinwand im vorderen Teil des Raumes projiziert, die später im Einklang mit dem Audio dynamisch verändert werden.

Zur Verfügung stehen den Nutzern verschiedene Bedienelemente, darunter ein Mikrofon zur Aufnahme einer Audiospur als Basis für den Granular-Synthesizer, sowie eine kleine Auswahl an Instrumenten und klangbildenden Objekten wie z.B. eine Steel Drum, ein Mini-Cajon, eine Ukulele, ein Kalimba usw. Ein MIDI-Keyboard ermöglicht die Kontrolle der Tonhöhe des Synthesizers. Zusätzlich gibt es einen Mikrocontroller mit sechs Potentiometern zur Steuerung der Synthesizer-Parameter (Frequenz, Spray, Mode, Reverb, Delay, Bitcrush), einem Schieberegler zur Positionierung innerhalb der Audiospur und einem Button zur Steuerung des Visualwechsels.

Abbildung 1: Rauminstallation

Der Ablauf beginnt mit der Aufnahme einer Audiodatei über das Mikrofon, indem der Nutzer über den Laptop den Record-Button in der Software SunVox drückt. Nachdem die Aufnahme gestoppt wurde, zeigt der Laptop die Waveform der Audiodatei an. Er ist ab diesem Punkt nur noch Rechenzentrum der Installation, während die Bedienung ausschließlich über das MIDI-Keyboard und den Mikrocontroller erfolgt.

Die MIDI-Daten des Mikrocontrollers und des Keyboards werden an den Laptop gesendet, auf dem die Audiosynthese der zuvor aufgenommenen Audiodatei stattfindet. Das resultierende Audiosignal wird über Lautsprecher in den Raum gegeben und gleichzeitig intern an TouchDesigner weitergeleitet. Dort wird das Audiosignal in Echtzeit hinsichtlich seines Frequenzspektrums und seiner Dynamik analysiert und in sechs verschiedene visuelle Darstellungen umgewandelt, die live gerendert und auf eine etwa 2x2m große Leinwand am vorderen Ende des Raumes projiziert werden.

Die Visuals wechseln nun automatisch alle zwei Minuten, können aber auch direkt vom Nutzer über einen Button auf dem Mikrocontroller gesteuert werden.

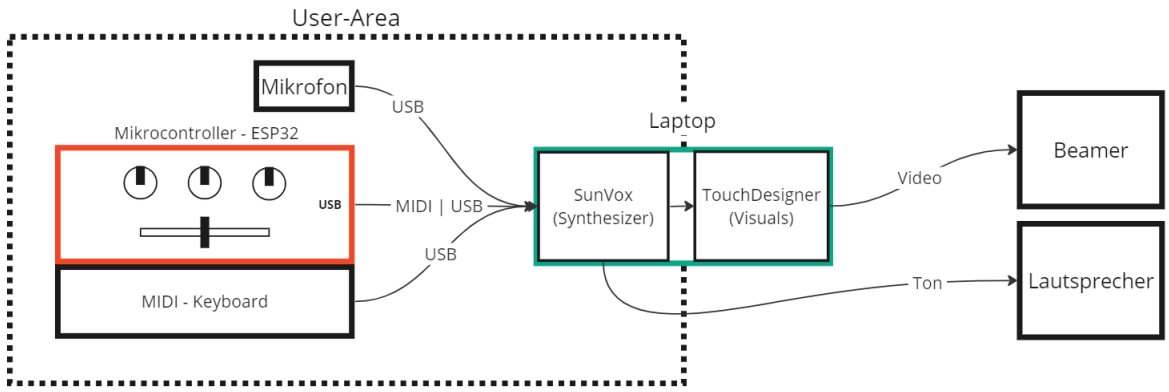

Abbildung 2: Blockschaltbild des kompletten Systems

4 Projektdurchführung

Zum Sammeln und Strukturieren von Ideen wurde das Tool „Miro“ verwendet. Dort konnten erste Ideen skizziert und eine grobe Aufteilung der Aufgabengebiete dargestellt werden. In der ersten Phase des Projektes war es dabei wichtig zu sondieren, welche Möglichkeiten es gibt und welche Werkzeuge dafür benötigt werden. Ein Miroboard war dabei hilfreich, um im Team zu kooperieren und alle Ideen übersichtlich zu halten.

Im Anschluss wurde ergänzend ein Git-Repository angelegt, um alle Materialien, Dateien und Texte abzulegen und für alle Mitglieder des Teams zugänglich zu machen.

Eine Gruppeneinteilung half frühzeitig, um eine klare Arbeitsaufteilung untereinander sicher zu stellen. Dabei wurde sich auf folgende Teams festgelegt:

- **Software**: Recherche nach Software zur Granular-Synthese sowie Erstellung eines Granular-Synthesizers.
- **Hardware**: Planung und Umsetzung eines Controllers mit geeignetem Layout, der Wert von Drehreglern und Fadern per MIDI an einen Computer sendet.
- **Visuals / TouchDesigner**: Erstellung von verschiedenen TouchDesigner-Projekten, die Audio verarbeiten und dazu geeignet reagieren.

4.1 Zeitplanung

Für das Management und die Zeiteinteilung wurden einige Ideen aus Scrum genutzt. So hat das Team sich beispielsweise einmal die Woche zu einem Meeting getroffen und darin den aktuellen Stand, die aktuellen Probleme und eventuelle Lösungen zusammengetragen.

Zudem wurde eine Roadmap erstellt, um den Zeitrahmen abzubilden, den jedes Team zur Verfügung hat. Wichtig war es genug Puffer einzuplanen, da es erfahrungsgemäß immer zu Verzögerungen kommen kann oder Teile des Projektes nicht so wie geplant laufen.

Abbildung 3: Auszug der Roadmap

4.2 Software

Es wurde schnell klar, dass eine Synthese innerhalb des Mikrocontrollers sehr komplex werden könnte. Deshalb wurde festgelegt, die Audio-Synthese auf der CPU des PCs über eine Software zu gestalten. Nachdem verschiedene Umgebungen getestet wurden, hat man sich auf die Software SunVox geeinigt.
SunVox bietet umfassende Möglichkeiten, um individuelle Synthesizer zu bauen. Dabei bleibt es verständlich und im Vergleich zu anderen Programmen leicht zu erlernen. Zudem ist SunVox kostenlos nutzbar. Dies war auch ein wichtiger Faktor, damit alle Personen uneingeschränkt mit dem Programm arbeiten konnten.

4.2.1 SunVox Implementierung

Nach der Entscheidung, SunVox als Synthesizer Software zu nutzen, galt es zuerst, ein Projekt zu bauen, das die Hauptaspekte der granularen Synthese erfüllt und steuern lässt. Also dass ein Sample zu einem gewählten Punkt in einer bestimmten Frequenz abgespielt werden kann. Das Sample sollte dabei entweder selbst aufgenommen oder ausgewählt werden. Dies ließ sich prinzipiell einfach mit den in SunVox eingebauten Modulen umsetzen.
Es gibt in SunVox das sogenannte MultiSynth Modul, welches allgemein genutzt werden kann, um ein Audiosample abzuspielen und dabei bestimmte Eigenschaften verschiedener Module zu transportieren und auf das jeweilige Sample zu übertragen.

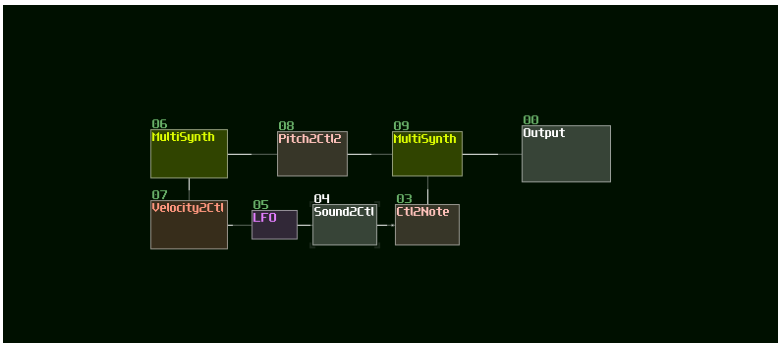

Abbildung 4: Aufbau des Granulator Moduls in SunVox

Es kann außerdem direkt die angespielte Phase eingestellt werden, weshalb sich darüber endlich auch die Position des Samples bestimmt, die hardwareseitig vom Slider kontrolliert werden sollte. Dass das Sample dann aber nicht nur einmal abgespielt wird, sondern in sich wiederholenden Grains, wird durch einen Low Frequency Oscillator sichergestellt, der auch als Modul in SunVox vorhanden ist. Sobald der MultiSynth gespielt wird, wird auch der LFO angesteuert und triggered in einer einstellbaren Frequenz das Sample immer wieder neu an. Wie lang das Sample abgespielt wird, bestimmt sich im Endeffekt aus der gewählten Frequenz des LFO, um aber darüber hinaus noch mehr vom Sample in einen eingestellten Sound hineinzubekommen, lässt sich die gewählte Position, die Phase des MultiSynth, auch mehr oder weniger randomisieren, was zu einer Art Position Spray führt, was noch mehr Abwechslung in die Soundgestaltung bringen sollte.

Ein großer Pluspunkt für SunVox ist die einfache Anbindung von MIDI-Geräten. Ein jeweiliger Controller muss nur am Rechner angeschlossen sein und man kann Parameter für Parameter auf die Bedienelemente legen wie man will. Dabei hätte man sogar experimentell mehrere Parameter auf ein Potentiometer mappen können. Für das MIDI-Keyboard, das zum Spielen verwendet werden sollte, musste nur festgelegt werden, welches Modul die Keyboard-MIDI-Befehle annehmen soll. Das war entsprechend unser selbst gebautes Granulator Modul.

Das Sampler Modul von SunVox stellt exakt die gewünschten Funktionen zum Aufnehmen und Reinladen von Samples bereit. Problem hier war aber, dass diese Funktionen nur per Mausklick im Interface auswählbar sind, was die Vorstellung einer vollends über Bedienelemente am Mikrocontroller gesteuerten Installation störte. Eventuell wäre es möglich gewesen, wenn man, statt direkt mit der SunVox Software zu arbeiten, die ebenso vorhandene Library genutzt hätte, die sich über Code steuern lässt.

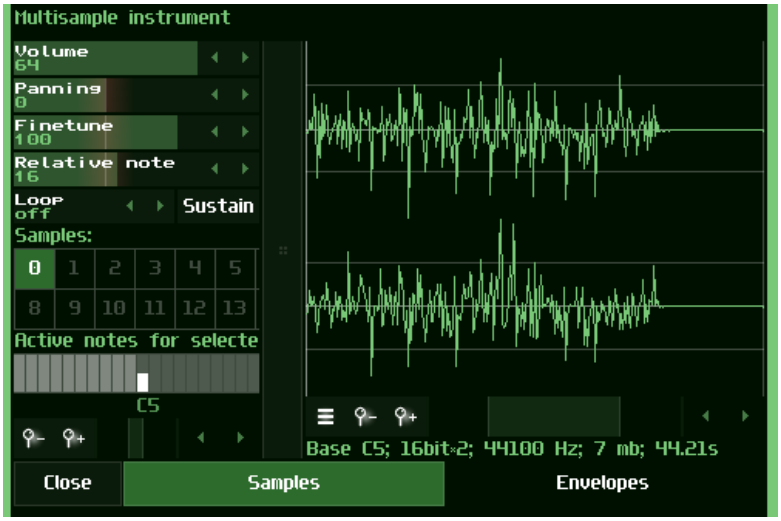

Abbildung 5: Anzeige der Waveform des Samples in SunVox

Die Entscheidung fiel dann aber schnell darauf, mit dem Programm weiterzuarbeiten und die Installation anzupassen und flexibler zu gestalten. Dies war einerseits übersichtlicher und gerade im Zusammenspiel mit der Peripherie um einiges schneller und flexibler in der Konfiguration und Fehleranalyse. Es hätte zudem deutlich mehr Zeit benötigt, sich in den Code der Software hineinzuarbeiten, um am Ende eventuell festzustellen, dass die Installation trotzdem nicht wie geplant umgesetzt werden kann.

SunVox bietet mit dem Sampler die Möglichkeit, die Waveform des Samples anzeigen zu lassen, inklusive Cursor, der anzeigt, wo das Sample aktuell abgespielt wird. Eine solche Visualisierung war von Anfang an für die Installation vorgesehen, nur war nicht eindeutig, wie sie eingebunden werden kann. So fiel dann die Entscheidung, einen extra Bildschirm respektive den Laptop, über den die Software am Ende lief, aufzustellen, um die Steuerung und damit die Funktionalität der Installation sicher zu stellen und gleichzeitig noch die Waveform anzeigen zu können.

Somit stand schnell ein Projekt, dass grundlegend als Granular Synthesizer funktioniert. Von da an war die Hauptaufgabe, auszuloten, welche ansteuerbaren Parameter sich in SunVox am besten für die Bedienung an unserer Hardware eignen. Viele Optionen standen zur Auswahl, die teilweise stark den Sound beeinflussten, wie zum Beispiel die Randomisierung der Tonhöhe. Um es nicht zu kompliziert zu halten, da es ohnehin mit einem gewissen Anspruch verbunden ist, den Synthesizer zu spielen, wurden am Ende aber nur die wichtigsten Parameter Position, Loop Frequency, Position Spray sowie die, die den Sound nochmal abrunden, Reverb, Delay und Distortion gemappt (letztere waren ebenfalls direkt als Modul in SunVox vorhanden)

Abbildung 6: Parameter des Granulator Moduls

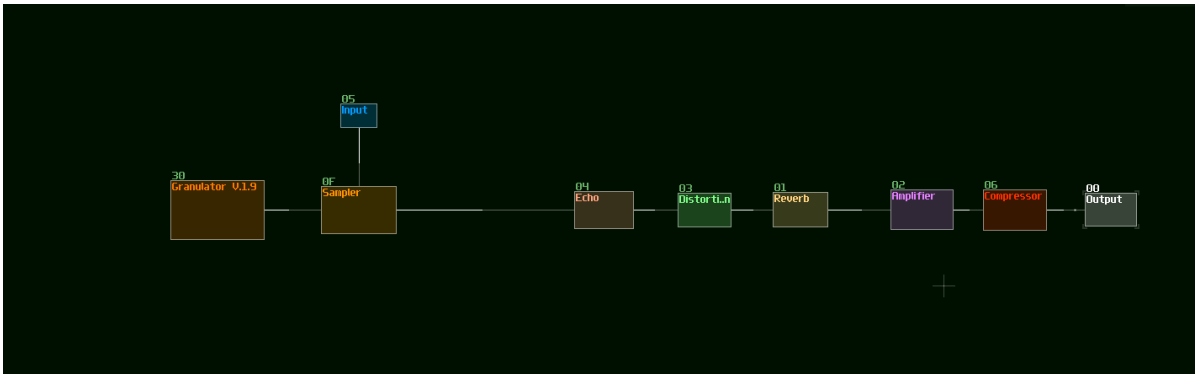

Abbildung 7: Aufbau des Synthesizers in Sunvox

Um zusätzlich das Problem zu beheben, dass es zu Artefakten in der Audio kommt, wenn das Sample durch die gewählte Position bei hoher Amplitude abgehakt wird, wurde noch versucht, mit einem ADSR Attack und Release für die Grains einstellbar zu machen, allerdings hat dieser nicht wie erwartet für die einzelnen Grains gegriffen, sondern nur für einen gespielten Input, der natürlich viele Grains hintereinander triggered.

Als Alternative lässt sich die Hüllkurve des Samples zwar direkt im Sampler selbst einstellen, dies muss aber jedes Mal aufs Neue manuell gemacht werden, sobald ein neues Sample geladen wird. Das wurde dann aber entsprechend in Kauf genommen und es blieb bei den genannten Parametern, die die Hardware steuern sollte.

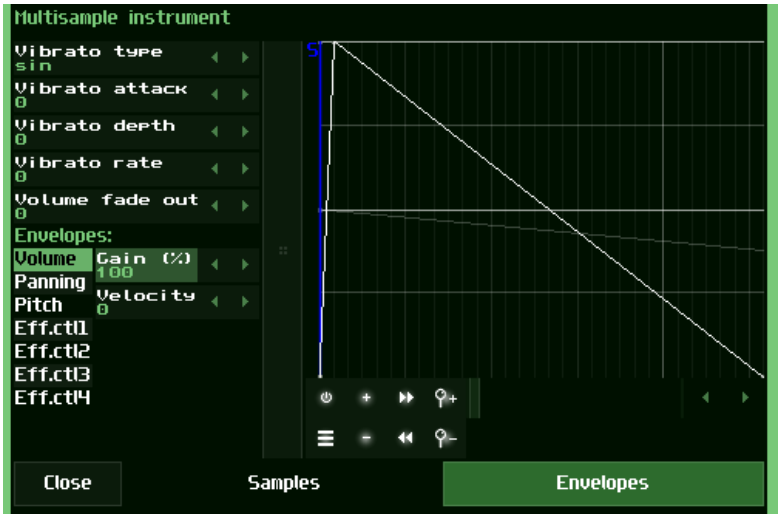

Abbildung 8: Manuelle Einstellung der Hüllkurve eines Samples

4.3 Hardware

Im Bereich der Hardware-Umsetzung waren die Rahmenbedingungen, durch die Ansteuerung des Synthesizers bestimmt. Demnach sollten Drehregler, Schieberegler und Buttons zur Verfügung stehen, die die Effekte und Mechanismen des Gerätes bedienbar machen.

4.3.1 Ausgangssituation und Anforderungen

Im Laufe der ersten Wochen, der Entwicklungsphase auf dem Miroboard, wurde klar, dass der Synthesizer MIDI Daten ausgeben sollte. Dies ist vorallem deshalb sinnvoll, da sowohl SunVox als auch Touch Designer dieses Format gut nutzen können.

Die Wahl des Mikrocontrollers fiel zunächst auf den ESP32, da er günstig ist und man davon ausging, er würde für alle Funktionen ausreichen. Später stellte sich heraus, dass der ESP über eine USB-to-UART Schnittstelle mit dem Computer verbunden wird. Er bot also keine native, vollwertige USB Schnittstelle, die man benötigte um MIDI Daten zu verarbeiten. Man wich dann auf den Teensy 4.1 aus.

Bereitgestellt wurden dann:

- mehrere (Dreh-) Potentiometer
- ein (Schiebe-) Potentiometer
- ein einfacher Button
- eine 18x24 Lochrasterplatine
- Buchsenleisten und Steckverbinder (Jumper)
- diverse Kabel

4.3.2 Proof of Concept (mit Code)

Zunächst wurde ein einfacher C++ Programmcode in der Browseranwendung Wokwi geschrieben und dort mit einigen Reglern simuliert.

Dann wurde mithilfe eines Breadboards eine erste Schaltung mit einem Drehregler und einem Schieberegler aufgebaut und der Code in die Arduino IDE übertragen. Dort wurde dieser dann um die MIDI Funktionalität erweitert. Dazu wurden die Bibliotheken USB-MIDI und MIDI Library von dem Autor lathoub und die Bibliothek MIDIUSB von Gary Grewal importiert. Diese sind direkt über die IDE enthalten und nicht extern.

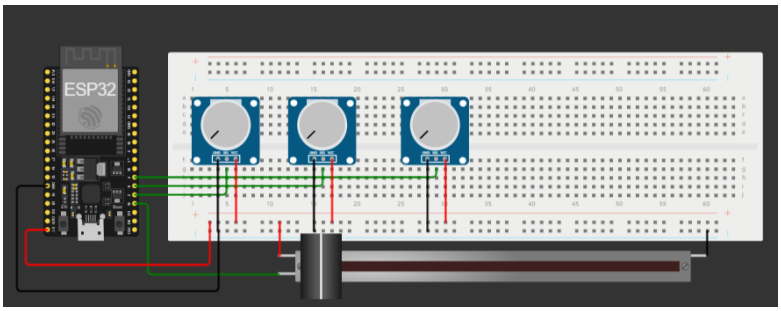

Abbildung 9: Simulation in Wokwi

Dann wurde der Code getestet und mit serieller Ausgabe überprüft. Der Teensy wurde zu diesem Zeitpunkt vom Rechner als MIDI Controller erkannt.

Die Potentiometer wurden also als analoge Signale gelesen und ihre Werte als MIDI Daten weitergegeben. Wobei der Mikrocontroller dann als externer MIDI Controller verwendet wurde.

Hier ein Codeausschnitt, der für genau ein Potentiometer funktioniert. Er kann nach belieben erweitert werden.

```
#include <Bounce.h>
// MIDI Channel Nummer, an den gesendet wird
const int channel = 1;
// MIDI Controller Nummer für alle analogen Eingänge
const int controllerA0 = 23; // 23 geht als 24 = frequency raus
// <-
//An diser Stelle können noch mehr Controller Nummern vergeben werden.


void setup() {
  Serial.begin(115200);
}
// Initialwerte für Eingänge
int previousA0 = -1;
// <-
//An diser Stelle können noch mehr Eingänge initiiert werden.

elapsedMillis msec = 0;

void loop() {
  // msec um Geschwindigkeit des Lesens und Sendens zu beschränken und Flut zu vermeiden
  if (msec >= 20) {
    msec = 0;
    // analoge Werte auslesen, darauf achten welchen A-Pin der Mikrocontroller hat
    // Werte durch acht teilen, um Eingang zu glätten und auf 1024 / 8 = 128 (0-127 MIDI Werte) aufzulösen
    int n0 = analogRead(A9) / 8;
    // <-
    //An diser Stelle können noch mehr Eingänge gelesen werden.
    // prüfen, ob sich Wert geändert hat. Wenn ja, dann senden.
    if (n0 != previousA0) {
      Serial.println("ControlChangeOn1");
      Serial.println(n0);
      // usbMIDI Objekt mit sendControlChange(controller, value, channel)
      usbMIDI.sendControlChange(controllerA0, n0, channel);
      // aktuellen Wert speichern, um mit nächten zu vergleichen
      previousA0 = n0;
    }
    // <-
    //diese if-Abfrage kann für alle bestehenden Eingänge wiederholt werden.
  }
  while (usbMIDI.read()) {
    // mache nichts an dieser Stelle, um eintreffende MIDI Nachrichten zu ignorieren
  }
}
```

Abbildung 10: Kompletter Code für einen analogen Eingang

Für den Synthesizer wurden final insgesamt acht Inputs genutzt.

```
// MIDI Controller Nummer für alle analogen Eingänge
const int controllerA0 = 23; // 24 = frequency
const int controllerA1 = 22; // 23 = spray
const int controllerA2 = 21; // 22 = mode
const int controllerA3 = 20; // 21 = reverb
const int controllerA4 = 19; // 20 = delay
const int controllerA5 = 18; // 19 = bitcrush
const int controllerA6 = 17; // 18 = position (slider)
const int controllerA7 = 16; // 17 = visuals
```

Abbildung 11: Alle initiierten Eingänge

4.3.3 Aufbau

ddd

4.3.4 Gehäuse

Nachdem nun alles Verkabelt ist, wurde es Zeit für ein passendes Gehäuse. Im Gespräch waren eine Holzkonstruktion oder ein 3D-Druck. Es wurde sich jedoch entschieden, mit dem 3D-Drucker zu arbeiten, weil es praktischer ist. Die erste Skizze ähnelte der Vision unseres Projekts hinsichtlich des Aussehens.
Um den Synthesizer-Look zu erreichen wurden die sechs Drehregler nebeneinander auf dem oberen Teil des Gehäuses angeordnet. Darunter der Schieberegler und der Button. Das erste Gehäuse sollte dann eine Länge, Breite und Höhe von (19 x 9 x 5) cm haben.

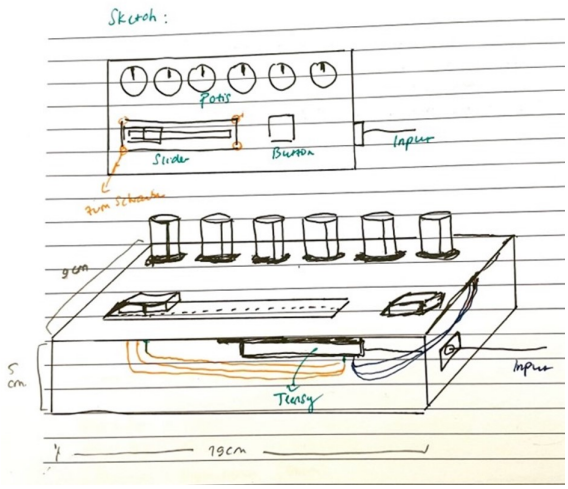

Abbildung 12: Erster Entwurf des Gehäuses

Zum Modellieren haben wir die Software Fusion360 von AutoDesk genutzt und zum erzeugen des G-Code die Drucker-eigene Software CraftWare Pro.

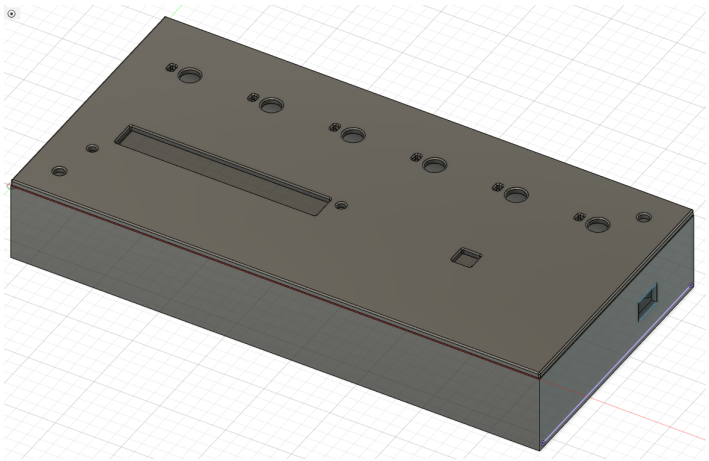

Abbildung 13: 3D Modell des Gehäuses in Fusion360

Der Teensy liegt in einer Fassung im Gehäuse ungefähr an der Stelle wo der Button ist. Die Hälse der Drehregler konnte man durch die Blende schieben und dann mit einer Mutter auf der anderen Seite festschrauben. Ähnlich konnte auch der Schieberegler befestigt werden.
Einzig die Blende mit dem unteren Teil des Gehäuses zu verbinden, war etwas aufwendiger. An dieser Stelle wurden dann zwei Gewindeeinsätze heiß gemacht und in das Gehäuse bestehend aus PLA Filament geschmolzen.

Abbildung 14: Finalisierter Controller für den Synthesizer

4.4 Touch Designer

In Touch-Designer war das Ziel interessante, audioreaktive Visuals erzeugen, die den granularen Synthesizer visuell untermalen. Man hatte außerdem den Anspruch, dass zwischen den Visuals automatisch oder manuell umgeschaltet werden kann.

4.4.1 Gesamtstruktur

In der Gesamtstruktur des Touch-Designer Files sind die einzelnen Visuals als .tox-Dateien eingefügt. Auf diese Weise können alle Visuals parallel laufen. Am Anfang gibt es einen Audioinput, auf den alle Visuals zugreifen. So wird nicht für jedes Visual ein eigener Input benötigt, sondern alle haben nur einen „In“ Chop, über den sie auf den gleichen Audioinput zugreifen können. Am Ende jeder .tox-Datei ist ein „Out“ Chop, worüber die Ausgänge der Visuals alle in einen „Switch“ laufen. Dieser Switch geht in ein „Cross“.

In den anderen Eingang des „Cross“ geht ein schwarzes Bild, welches für den Übergang zwischen den Visuals benötigt wird. Am Ausgang des „Cross“ hängt ein „Window“, welches der finale Ausgang für Touch-Designer ist. Für den Übergang gibt es einen automatischen und einen manuellen Modus.

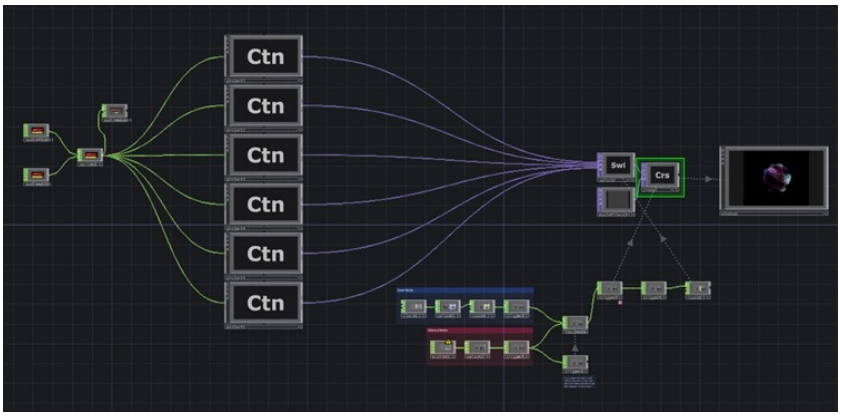

Abbildung 15: Struktur des Touch Designer Programms

Im Automatischen Modus zählt ein Timer hoch, der nach einer bestimmten Zeit den Übergang auslöst. Beim Übergang wechselt der „Cross“ für einen kurzen Moment auf das schwarze Bild. Hier gibt es einen Fade und keinen harten Cut. In dem Moment, indem das Bild Vollständig schwarz wird, springt der „Switch“ auf das nächste Visual und der „Cross“ fadet aus dem schwarzen Bild zurück auf das neue Visual. Für den manuellen Modus gibt es einen extra Knopf auf dem Controller, der ein Midi Signal an Touch-Designer schickt und den Übergang einleitet. Nach dem manuellen Wechsel bleibt das ausgewählte Visual für eine bestimmte Zeit stehen. Läuft diese Zeit ab, ohne dass der Knopf erneut gedrückt wurde, springt Touch-Designer wieder in den automatischen Modus, damit nicht dauerhaft ein Visual stehen bleibt, sobald der Knopf einmal gedrückt wurde.

4.4.2 Visuals

Color Blobs

In diesem Audio-Visualizer wird das Spektrum des Audio-Inputs genutzt, um Kreise zu erstellen und deren Größe zu verändern. Diese werden durch eine Aneinanderreihung von Feedback, Blur, Displace, Mirror und anderen TOP-Nodes zu einem Interessanten, sich bewegenden „rohrschachtest-artigen“ Muster. Die Farbgebung wird über eine Ramp- und einen Lookup-TOP gesteuert.

Abbildung 16: „Color Blobs”

Point Cloud

Eine Röhre wird verformt und durch ein Noise-TOP geschickt. Das Abbild der Röhre wird dann durch eine Point Cloud aus kleinen Quadraten abgebildet. Das Noise wird durch den Audio-Input verändert, wodurch die Audio-Reaktivität erzeugt wird.

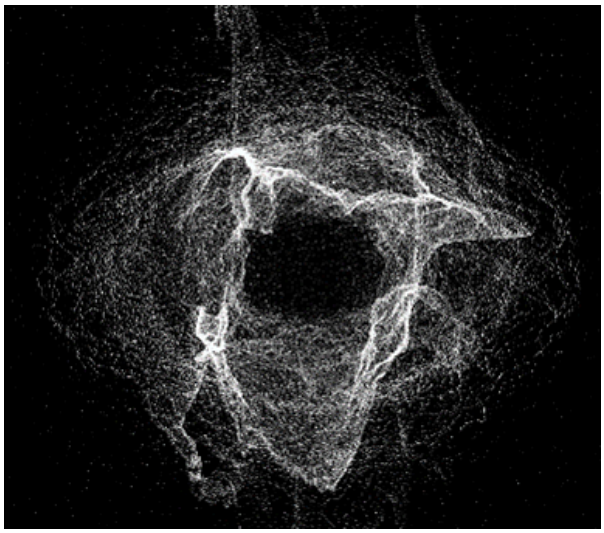

Abbildung 17: „Point Cloud“

Fuzzy Ball

Hier wird eine verformte Kugel mit einem PHONG-Material versehen, dessen Farbe, Height-Map und Normal-Map, durch ein sich bewegendes Noise erstellt werden. Das Material wird in einer Gitter-Optik angezeigt. Farbe wird in Form von Beleuchtung hinzugefügt. Die Verformung und die Beleuchtungsintensität werden über den Audio-Input gesteuert.

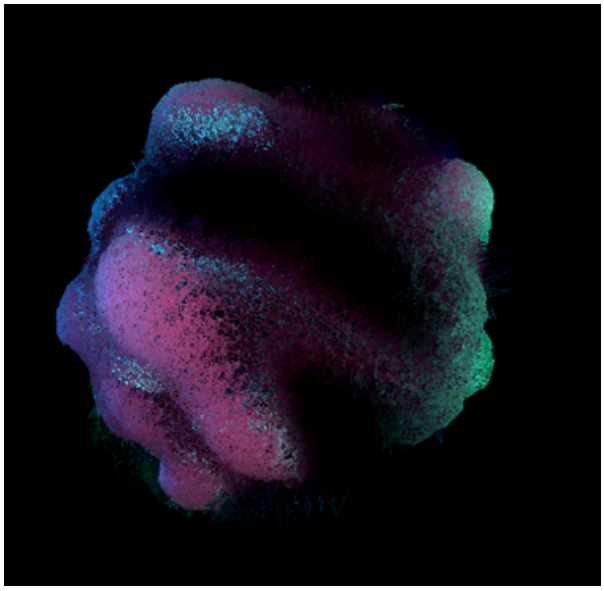

Abbildung 18: „Fuzzy Ball“

Purple Fog

Dieses Visual basiert auf einem Circle, dessen Größe auf das Audiosignal reagiert. Zudem wird er durch verschiedene Filter, wie zum Beispiel „Displace“ und „Edge“ in seiner Form verändert. Das Top „Ramp“ sorgt hier für den Farbverlauf. Durch ein „Noise“ wird der Circle noch stärker verzerrt und es entsteht dann die Finale Struktur. Um den Effekt zu verstärken, wird anschließend ein Feedback Loop eingebaut.

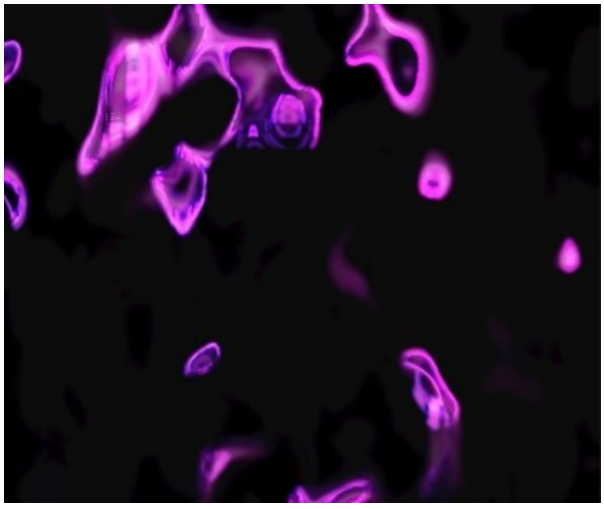

Abbildung 19: „Purple Fog“

Donut

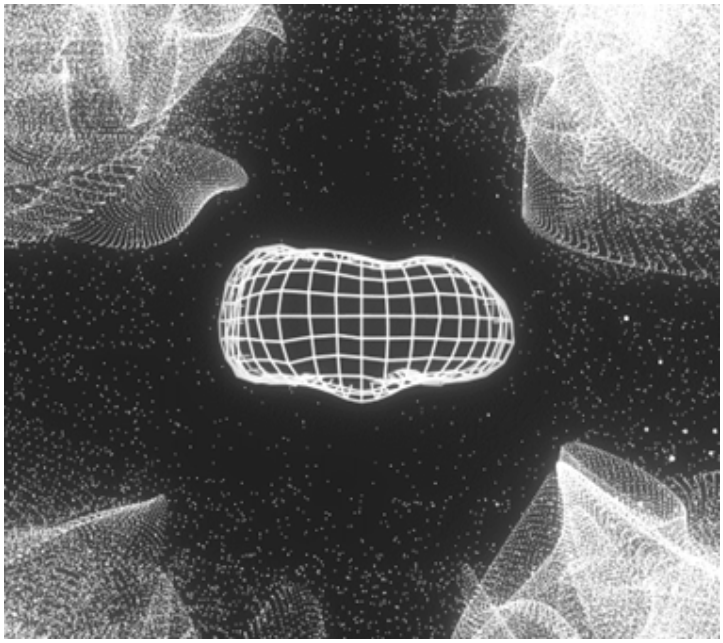

Abbildung 20: „Donut“

Für den Hintergrund wird ein Circle genutzt, aus dem mittels eines „Noise“ Tops eine Wolke aus vielen Kugeln erstellt wird. In diese Wolke wird die Kamera platziert, um den Effekt einer Galaxie zu erzeugen. Für die Wolke wird nun ein Loop erstellt, sodass sich die Kugeln auf die Kamera zu bewegen. Sobald eine bestimmte Position hinter der Kamera erreicht ist, springen die Kugeln wieder auf die Startposition zurück und es entsteht ein Effekt, in dem Kugeln permanent auf die Kamera zu fliegen. Die Geschwindigkeit der Kugeln wird durch den Audioinput beeinflusst. In dieses Feld wird ein Torus eingefügt, welcher durch das „Line Material“ seine Struktur erhält. Damit der Torus durch das Audiosignal verändert werden kann, reagiert zum einen die Größe des Torus auf das Eingangssignal und zum anderen wird ein „Noise“ dazu geschaltet, dessen Amplitude ebenfalls auf das Eingangssignal reagiert. Durch das „Noise“ kommt es zu den Änderungen in der Struktur des Torus.

4.5 Zusätzliche Software zum Verbinden von SunVox und Touch Designer

Damit Touch-Designer das Audio aus Sunvox empfangen kann, muss entweder ein Loopback-fähiges Audiointerface oder eine weitere Software genutzt werden. Man hat sich nach einigem Ausprobieren für die Software VB-Cable entschieden, welche als virtuelles Audiokabel dient, mit dem sich der Output von Sunvox in den Input von Touch-Designer leiten lässt.

Ein weiteres Problem war, dass in Windows nur ein Programm zur Zeit von einem MIDI-Controller angesteuert werden kann. Man wollte allerdings Sunvox und Touch-Designer über den gleichen MIDI-Controller ansteuern. Hierfür war eine Kombination aus zwei zusätzlichen Softwares nötig. Die Software loopMidi wird genutzt, um zwei vir-

tuelle MIDI-Geräte für jeweils Sunvox und Touch-Designer zu erzeugen. Dann wird mit der Software MIDI-OX das angeschlossene MIDI-Gerät in die beiden virtuellen MIDI-Geräte geleitet. So ist es Möglich, dass die Audio-Visualizer in Touch-Designer auf das Audio aus Sunvox reagieren können und mit einem einzelnen MIDI-Controller beide Softwares angesteuert werden können.

5 Fazit

5.1 Zusammenfassung

5.1.1 SunVox

Zu SunVox lässt sich insgesamt sagen, dass es für Audiosynthese und elektronische Musik für ein kostenloses Programm ein mächtiges Werkzeug ist. Leider ist es eher darauf programmiert, Songs zu gestalten und zu produzieren und weniger, selbst als Instrument genutzt zu werden. Das führte dazu, dass in der Ausgestaltung der Installation einige Kompromisse vereinbart werden mussten, da einfache Funktionen wie ein Shortcut zur Aufnahme von Samples in der Software nicht vorhanden sind. Jedoch lässt sich, in Bezug auf den Synthesizer, ein positives Bild ziehen. Trotz der teilweise umständlichen Steuerung von SunVox ist das Ergebnis ein guter Granular Synthesizer, der sich kreativ spielen lässt und mit den Visuals aus TouchDesigner harmoniert.

5.1.2 Touch Designer

Nach ein paar Wochen, die man im Touch Designer Gewerk brauchte, um sich in der Software zurecht zu finden, hat man angefangen selbst Visuals zu bauen. Von diesen wurden sechs ausgesucht, zwischen denen im Finalen Projekt umgeschaltet werden konnte. Zusätzlich hätte man gern in Touch-Designer noch mehr MIDI-Daten des Controllers eingebunden und Lichter im Raum über DMX angesteuert. Die wichtigsten Funktionalitäten, konnten aber gut umsetzt werden.

5.2 Ausblick

Trotz der Tatsache, dass man mit dem Ergebnis zufrieden sein kann, gibt es einige Punkte, die noch umgesetzt werden können bzw. die einem für Austellungen geeignetem VisuSynth (z.B. auf dem Rundgang Finkenau) beitragen würden:

- Die Aufnahme über den Mikrocontroller ansteuern mit dem Ziel das Setup ohne Laptopdisplay, Maus und Tastatur umzusetzen.
- Ein kleines Display auf dem die Waveform der Aufnahme angezeigt wird.
- Eine Möglichkeit zum Spielen von Noten in den MIDI-Controller einbauen . So wäre das zusätzliche MIDI-Keyboard nicht nötig.

- Einen Reset-Knopf einbauen, falls ein unerträglicher Sound erzeugt wurde oder gar nichts passiert.
- Einbindung der MIDI-Parameter in die Touch-Designer-Visuals.
- Licht für die Installation über DMX audioreaktiv ansteuern.
- Mehr Visuals und reaktivere Visuals (häufig haben sie nur bei viel Bass stark reagiert).

6 Quellenverzeichnis