

# My Marvellous Impression Management Impotence: Dokumentation

Helge Fischer

<http://www.hfischer.info>

Matrikelnummer: 15497

Dokumentation der Bachelor-Abschlussarbeit

Studiengang Digitale Medien an der Hochschule für Künste, Bremen

Betreuende Professoren: Prof. Christoph Lischka (Erstgutachter) und Prof. Kilian Schwoon (Zweitgutachter)

Oktober 2007

„My Marvellous Impression Management Impotence“ ist eine interaktive, agentenbasierte Klanginstallation mit vierundzwanzig Lautsprechern und acht Mikrophonen, deren Ein- und Ausgabesignale von einer selbstgeschriebenen Software in einem Computer verarbeitet und gesteuert werden. Ich beschreibe im Folgenden die Merkmale und das zu beobachtende Verhalten der Klanginstallation, beschreibe grob die Struktur des dahinterliegenden Programms, gehe auf die verwendete Technik ein und schliesse mit einigen nicht autoritativen persönlichen Gedanken zur möglichen Deutung der Arbeit.



Abbildung 1: Installationsansicht

## **Inhaltsverzeichnis**

Das Setup und das Verhalten des Systems und der Klangagenten <i>(Nach welchen Regeln funktioniert das System?)</i>	<b>S. 4</b>
Die Programmierung im Computer <i>(Der grobe Aufbau des Programms.)</i>	<b>S. 5</b>
Entwicklung <i>(Ein paar Worte zur Technik und Vorgehensweise.)</i>	<b>S. 6</b>
Interpretationsansätze <i>(Eigene Gedanken zur Arbeit, tatsächlich ohne Autoritätsanspruch.)</i>	<b>S. 8</b>
Dankeschön	<b>S. 9</b>
Ehrenwörtliche Erklärung	<b>S. 10</b>
Anhang	

## Das Setup und das Verhalten des Systems und der Klangagenten

(Nach welchen Regeln funktioniert das System?)

II Vierundzwanzig Lautsprecher sind an der Decke eines Raumes in gleichmässigen Abständen in einem Raster angeordnet. Gleichzeitig hängen acht Mikrophone ebenfalls gleichmässig verteilt von der Decke. Jedes Mikrofon nimmt nun eine zwei Sekunden lange Klangdatei auf, wenn der am Mikrofon gemessene Schall eine bestimmte Amplitude (Lautstärke) übersteigt und wenn die gemessene Amplitude um etwas grösser ist, als der Durchschnitt der an den anderen Mikrofonen gemessenen Werte. Aufgenommen werden können sämtliche auditive Ereignisse im Raum, wie auch Ereignisse, die wir Menschen nicht wahrnehmen, dafür aber die Mikrofone. So fangen manche Mikrofone mitunter auch Radiowellen ein, je nachdem, wo sie im Raum platziert sind.

Wird eine Klangdatei aufgenommen, entsteht im Lautsprecherraster an der Stelle, wo das Mikrofon hängt, ein „Klangagent“. Die Klangdatei bekommt also bestimmte Attribute zugewiesen, die bestimmen, wie und wo die Klangdatei im Folgenden innerhalb des Lautsprecherrasters abgespielt wird. Man könnte auch vom Verhalten des Klangagenten sprechen, der sich nun autonom und festgelegten Regeln folgend innerhalb des Rasters bewegt und verändert. Das Lautsprecherraster könnte metaphorisch als ein Ökosystem betrachtet werden, innerhalb dessen Klangagenten für eine gewisse Zeit leben und interagieren.

Jeder Klangagent bekommt bereits zu seiner Entstehung eine festgelegte Lebenszeit, die zwischen sechzehn und siebzig Sekunden liegen kann. Während dieser Lebenszeit wird der Klang geloopt aus verschiedenen Lautsprechern wiedergegeben. Gestartet wird die Wiedergabe beim Lautsprecher, der in der Nähe des aufnehmenden Mikrofons hängt. Hernach sucht sich der Klangagent eine Richtung innerhalb des Lautsprecherrasters aus (Nord, Nord-Ost, Süd-Ost, Süd, Süd-West oder Nord-West) und wandert zum nächstgelegenen Lautsprecher in dieser Richtung, das heisst die Wiedergabe erfolgt nun in einem benachbarten Lautsprecher. Der Klangagent behält diese Richtung bei, bis er zum Rand des Rasters kommt und also sich nicht mehr weiter in die gleiche Richtung bewegen kann. Nun entscheidet er sich zufällig für eine neue Richtung und behält diese bei, bis er wieder an den Rand kommt und so weiter.

Während der gesamten Lebensdauer der Klangagenten wird die Abspielgeschwindigkeit sukzessive verringert: Je kürzer die zuvor festgelegte Lebenszeit, desto schneller. Man kann also bereits am Anfang hören, wie lange in etwa der Klang noch im System vorhanden sein wird, je nachdem wie schnell der Klang im Laufe der Zeit wiedergegeben wird. Ist der Agent bereits sehr langsam (und klingt somit tiefer), wird er bald gelöscht werden. Ausserdem werden mit der Zeit zufällig kleine Teile der Klangdatei gelöscht, der Klang erhält kurze stille Abschnitte und klingt damit mehr und mehr abgehackt und gebrochen. Die Abspielgeschwindigkeit und das Löschen einzelner Stellen lassen uns also den Alterungsprozess des Klangagenten erfahren. Kurz bevor der Agent „stirbt“ erhöht sich übrigens seine Wiedergabelautstärke und der Klang verabschiedet sich mit grossem Getöse aus dem System.

Jeder Klang bekommt bei der Entstehung zufällig einen anderen Bandfilter zugewiesen, der bestimmte Frequenzen des Klangspektrums verstärkt und andere stark abschwächt. Dies verhindert mitunter unangenehme Feedback-Geräusche und verleiht jedem Klang sein eigenes Klangcharakteristikum.

Innerhalb des Systems können sich höchstens acht Klangagenten gleichzeitig befinden. Ist diese Zahl erreicht, geschehen keine weiteren Aufnahmen und keine neuen Agenten werden ins Leben gerufen. Dieser Zustand wird in der Regel aber nur für sehr kurze Zeit erreicht, wie ich gleich erkläre.

Treffen sich zwei Klangagenten in demselben Lautsprecher, werden beide Klänge gelöscht und ein neuer Klang, gemischt aus den sich treffenden, entsteht. Man könnte also sagen, zwei Klangagenten bekommen ein „Kind“, verausgaben sich dabei aber so sehr, dass sie danach „sterben“. Der daraus entstehende neue Klangagent hat wieder die volle Lebensdauer (sechzehn bis siebzig Sekunden) und kann auch wiederum mit anderen Klangagenten „Kinder“ erzeugen, wenn er diese trifft. Hiermit entsteht ein sich selbst regulierendes System, dass verhindert, dass sich zu viele Klänge im Raster bewegen. Denn: Haben wir zum Beispiel acht Klangagenten im System, ist die Wahrscheinlichkeit bei den nur vierundzwanzig Lautsprechern recht gross, dass sich zufällig zwei Klänge treffen, daraus ein neuer hervorgeht, die „Eltern“ aber beide „sterben“ und somit plötzlich ein Klangagent weniger im System ist. Befinden sich jedoch nur wenige Klänge im System, ist die Wahrscheinlichkeit ungleich niedriger, dass diese sich treffen und aus zwei Klängen einer wird.

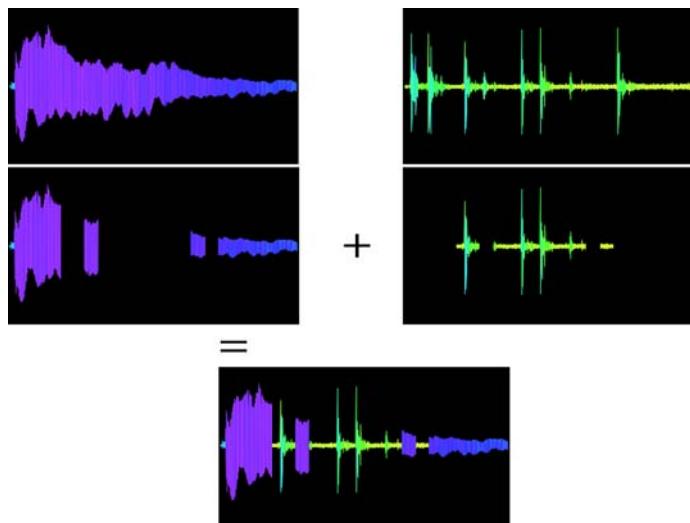


Abbildung 2: Zwei Soundfiles werden gemischt.

Das Mischen der zwei sich treffenden Klänge geschieht in der Weise, dass zufällig bestimmte Teile der Klänge zu einem neuen Klang zusammengefügt werden. Der neue Klang könnte zum Beispiel am Anfang aus dem ersten Zehntel des ersten Klanges bestehen, dann folgen vielleicht das zweite und dritte Zehntel des zweiten Klanges und wieder das vierte des ersten Klanges und so weiter.

## Die Programmierung im Computer

(Der grobe Aufbau des Max/MSP-Programms.)

Max/MSP-Patch "20"

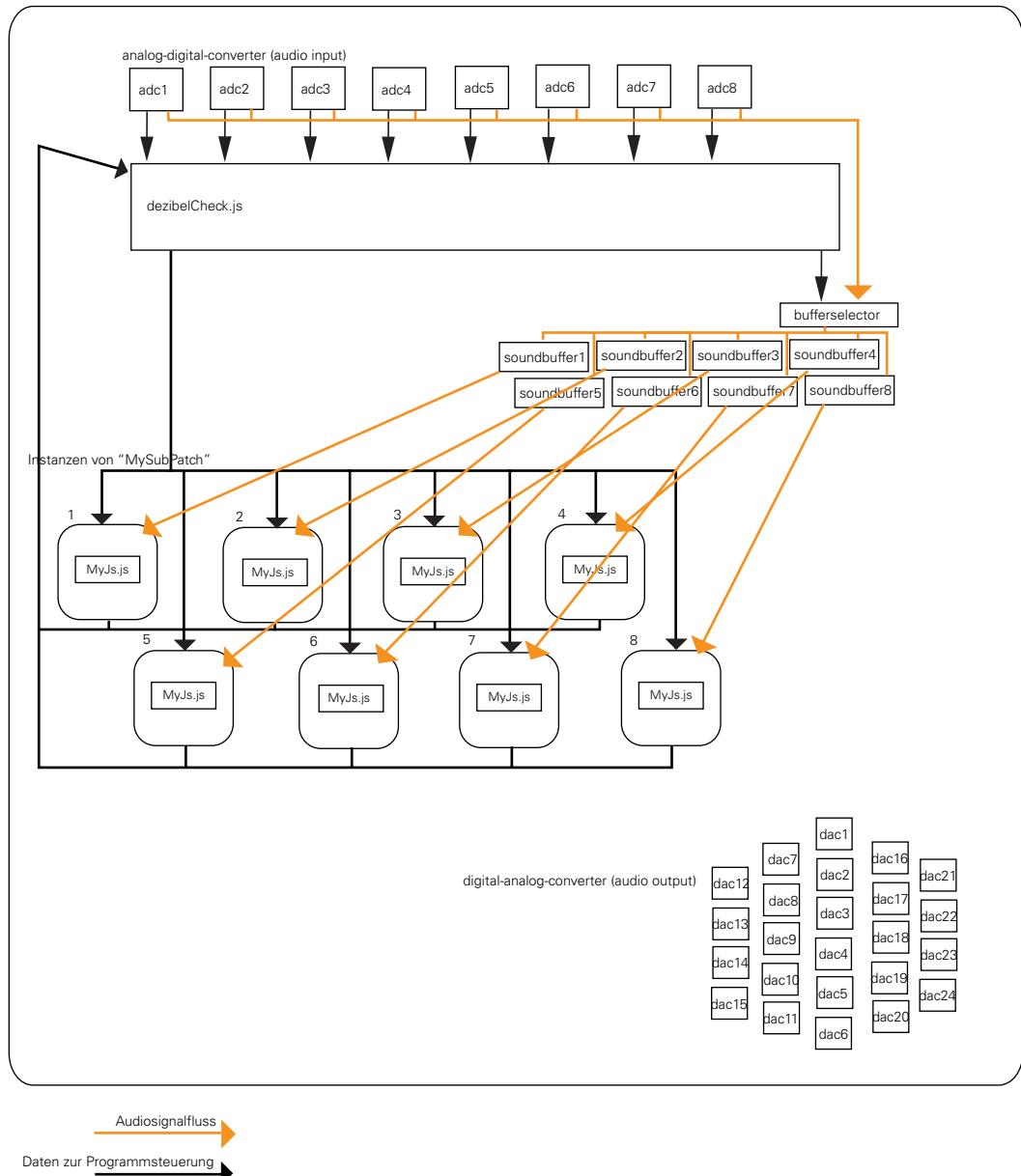


Abbildung 3: Schematische Darstellung des Programmaufbaus (vereinfacht).

|| Die im Computer erfolgende Bearbeitung der Klangdaten wurde mit der speziell für Audioanwendungen konzipierten Programmiersprache Max/MSP und in Teilen mit JavaScript realisiert. Im Anhang befinden sich Screenshots der Max/MSP-Patcher und Ausdrücke des JavaScript Codes.

Der Patch „22“:

Die Max/MSP-Patcher wurden während der Entwicklung fortlaufend nummeriert, die zur Zeit finale Version ist Nummer 22. Dieser Patch beinhaltet das gesamte Programm. Er besteht im Groben aus dem Javascriptobjekt „dezibelCheck.js“, das mit den acht Instanzen des Patcher „mySubPatch“ kommuniziert. Es gibt eine Instanz des „mySubPatch“ für jeden Klangagenten, dieser bestimmt das Verhalten und die Charakteristika des jeweiligen Klangagenten. Ausserdem befinden sich in diesem Patch Max/MSP-Objekte, die das Mischen von zwei Klangdateien ausführen, wenn sie dafür von dem „dezibelCheck.js“-Objekt den Auftrag bekommen. Die höchstens acht Klangdateien werden in Buffern gespeichert. Sowohl das Objekt „dezibelCheck.js“ als auch „mySubPatch“ können auf diese Buffer zugreifen.

Das Javascriptobjekt „dezibelCheck.js“:

ist das Objekt, das auf globaler Ebene am meisten weiß und bestimmte Daten der einzelnen Klangagenten

bekommt, wie auch die Eingangspegel der Mikrophone. Das Objekt:

- weiss um die Eingangspegel der Mikrophone,
- weiss, wo die bereits entstandenen Klangagenten im Raster sind,
- weiss, wie lange die jeweiligen Klänge noch leben und
- weiss, welche Instanzen von „mySubPatch“ keinen Klang/Soundbuffer zugewiesen bekommen haben und also einen neuen Klang aufnehmen können.

Mit Hilfe dieser Daten bestimmt das Objekt, wann welches Mikrophon aufnimmt, wann also ein neuer Klangagent entsteht: Dies kann nur passieren, wenn weniger als acht Klangagenten im System sind. Der neu aufgenommene Klang wird dann einer Instanz von „mySubPatch“ zugewiesen, die derzeit keinen Klang unterstützt. Das Objekt gibt hierfür die Nummer des Mikrofons aus, von welchem aus aufgenommen werden soll, bestimmt, in welchem Soundbuffer der Klang gespeichert werden soll und welche Startposition der neue Klangagent innerhalb des Rasters bekommt. Die Startposition ergibt sich aus der Position des aufnehmenden Mikrofons.

Das Objekt weiss ebenfalls, wann sich zwei Klangagenten im selben Rasterfeld befinden und also zusammengemischt werden: Dabei wird ein Klangagent gelöscht, der andere bekommt Datenanteile des zu löschen Klang. (Dies ist nicht in der Abbildung illustriert.)

Der Patch „MySubPatch“:

Ihn gibt es achtmal, für jeden möglichen Klangagenten einen. Hier werden die Klänge abgespielt und die Signale werden an die Ausgabekanäle gesandt (dac1 bis dac24). Das Herzstück ist das Javascriptobjekt „myJs.js“. In diesem Javascriptobjekt werden die Eigenschaften des Klangagenten festgelegt und über die Zeit geändert; die Umsetzung geschieht in den umliegenden Max/MSP-Objekten. Das Javascriptobjekt berechnet also alle aktuellen Werte und gibt diese an die Max/MSP-Objekte weiter. Jeder „MySubPatch“ über gibt die Lebensdauer und den Aufenthaltsort des Agenten stets an das globale Javascriptobjekt „dezelbelCheck.js“.

Das Javascriptobjekt „myJs.js“:

bekommt bei der Entstehung eines neuen Klangagenten dessen Startposition im Lautsprecherraster, bestimmt aber von nun an selbst, wo der Klang hinwandert. Das Objekt:

- bestimmt, in welche Richtung sich der Klang bewegt,
- bestimmt die Abspiellautstärke des Klanges,
- bestimmt, wann Teile des Klanges gelöscht werden (Alterungsprozess),
- bestimmt die Lebenserwartung des Klanges und
- bestimmt die Center-Frequenz des anzuwendenden Bandfilters.

## Entwicklung

(Ein paar Worte zur Technik und Vorgehensweise.)

|| Das Konzept der Arbeit ist während meines Auslandstudiums in Malmö entstanden, nachdem ich ein Research-Paper vom Viktoria Institute in Göteborg ([www.viktoria.se](http://www.viktoria.se)) gelesen habe. Das Paper hat den Titel „Designing a Mobile Music Sharing System Based on Emerging Properties“ und beschreibt das Konzept eines MP3-Players, bei dem die MP3-Dateien sich von einem Player zum anderen bewegen können, wenn sie zum Beispiel in ihrem angestammten Player nicht mehr gespielt werden. (Das will vermutlich Niemand, aber es ist auch nur eine Idee.)

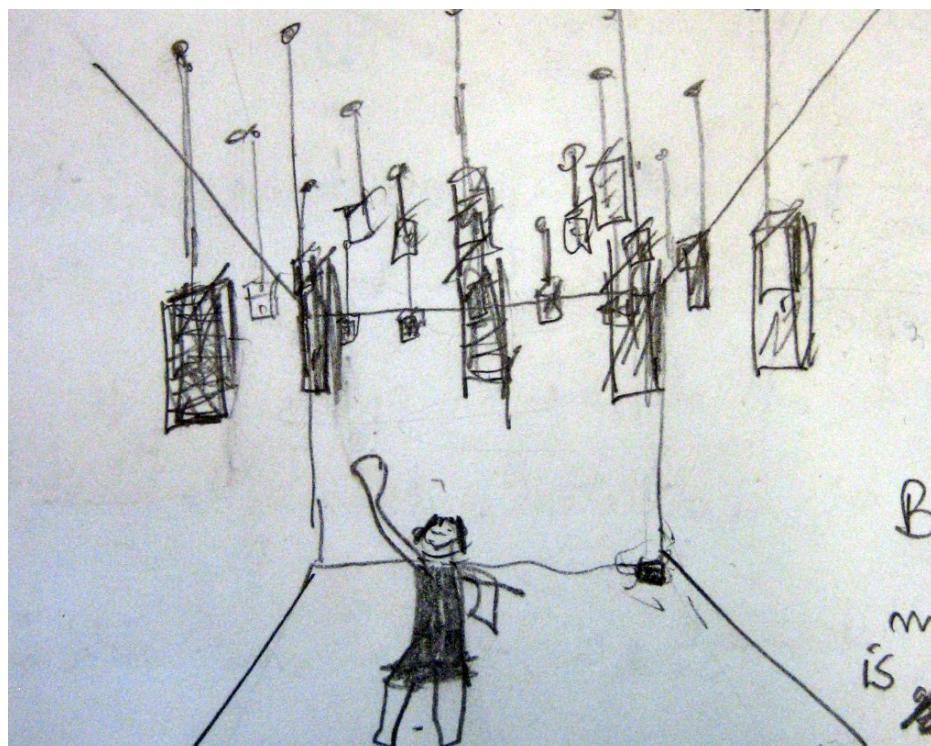


Abbildung 4: Eine erste Skizze.

Nach der Konzeption wurde zunächst ein kleines Programm in Processing realisiert, um damit mögliche Bewegungsverhalten der Klänge visuell auszuloten. Insbesondere war dies nötig, um herauszufinden, wie gross das Lautsprecherraster mindestens sein müsste, um Bewegungsmuster überhaupt erkennen zu können und Interaktionen zwischen verschiedenen Klängen zu ermöglichen. So wurden verschiedenen Rastergrößen und -anordnungen am Bildschirm ausprobiert: Klänge repräsentierende farbige Kreise bewegen sich innerhalb eines Rasters hin und her. Schliesslich fiel die Wahl auf vierundzwanzig Lautsprecher; diese erschienen ein genügend grosses Raster zu bilden, in denen sich mehrere Klangagenten würden bewegen können, gleichzeitig ist es klein genug, so dass es mit den zur Verfügung stehenden (insbesondere auch limitierten finanziellen) Mitteln noch realisierbar war. Die vierundzwanzig Felder (Lautsprecher) sollten in einem nicht gleichseitigen Sechseck angeordnet sein.

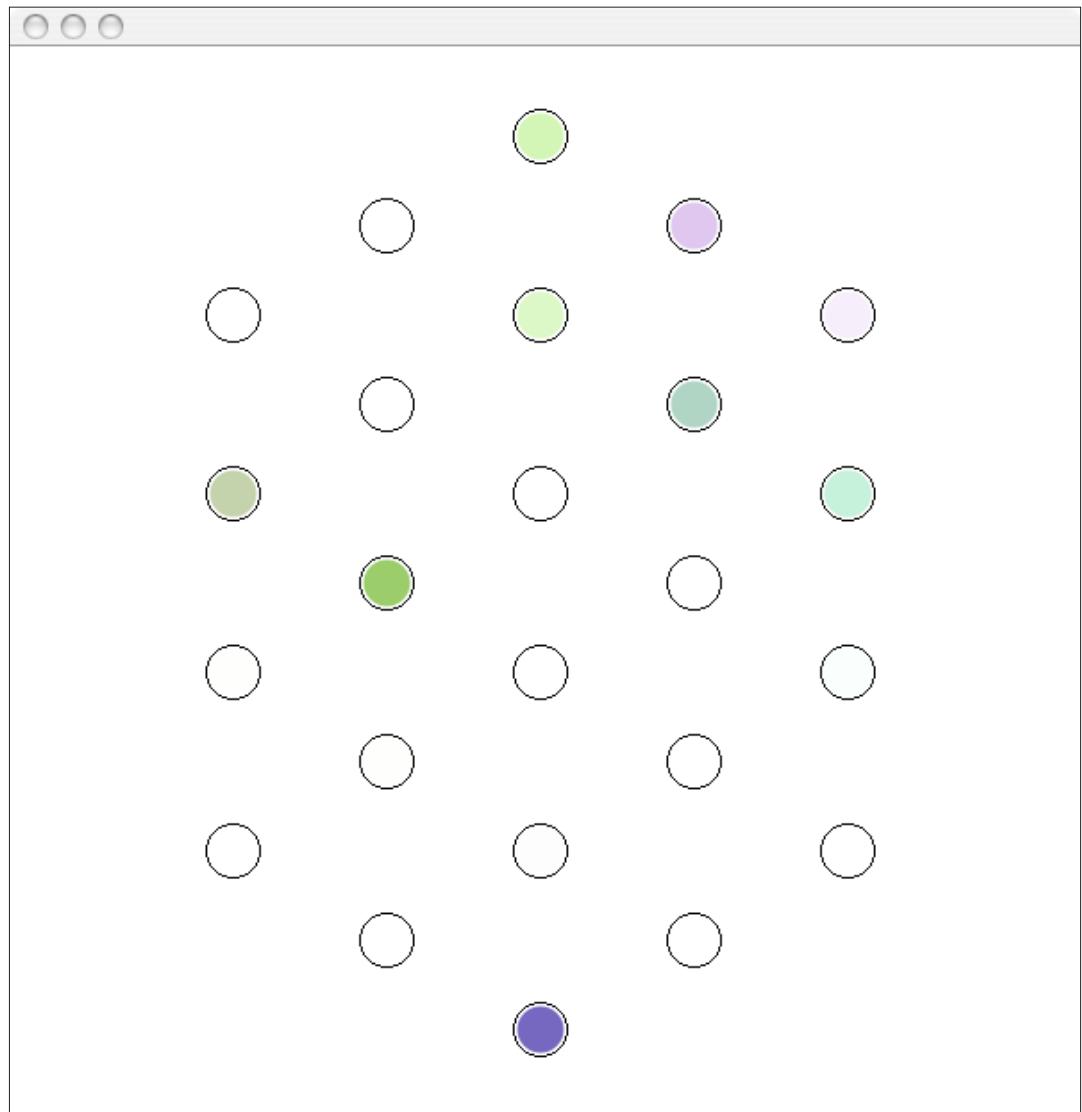


Abbildung 5: Screenshot der interaktiven Processingskizze. Durch Mausklicks auf einen der (leeren) Kreise wird ein neuer farbiger Kreis ins Leben gerufen, der sich dann im Raster bewegt.

Die Lautsprecher wurden dann also in gleichmässigen Abständen an der Decke befestigt und jeder erhielt eine Art Lautsprecherbox aus gekürzten Ofenrohren, die den Schall möglichst nur in eine Richtung (nämlich nach unten) lenken sollen, damit man klarer zu erkennen vermöge, wo sich ein Klang in etwa gerade aufhält. Die acht Mikrophone wurden ebenfalls gleichmässig im Raum verteilt, hängen von der Decke und wurden jeweils einem bestimmten Lautsprecher zugeordnet. Jedes Mikrophon hat einen eigenen Vorverstärker, wie auch jeder Lautsprecher einen eigenen kleinen Verstärker hat.

Über Kabel sind alle Lautsprecher und Mikrophone mit drei acht-kanälichen A/D & D/A Wandlern verbunden, also mit drei Interfaces, welche die analogen Audiosignale in digitale Signale umwandeln und umgekehrt. Diese A/D & D/A Wandler sind nun mit optischen Kabeln (Lichtwellenleiter, der Markenname ist TOSLINK) mit der Soundkarte im Computer verbunden. Die A/D & D/A Wandler kommen von der Firma Behringer und heissen „Ultragain Pro-8 Digital ADA8000“, die Soundkarte im Computer kommt von der Firma RME und heisst „Hammerfall DSP System HDSP 9652“.



Abbildung 6: Zu sehen sind zwei von eigentlich drei Soundinterfaces, das Netzteil für die Verstärker und Kabel, die zu den Lautsprechern und Mikrofonen führen.

## Interpretationsansätze

(Persönliche Gedanken zur Arbeit, tatsächlich ohne Autoritätsanspruch.)

Die entstehenden Klangräume sind meist von recht eigenwilliger Natur und man hört deutlich die computergestützte Bearbeitung der aufgenommenen Klänge. Die Klänge werden abgehackt oder neu zusammen gemischt, schneller oder langsamer abgespielt, ändern ihre Ausgabelautsprecher, es gibt starke Brüche und so weiter. Zumeist erkennt man am Anfang noch den Klang wieder, wie er ursprünglich in das System aufgenommen wurde (man erkennt also zum Beispiel die eigene Stimme wieder); doch ist der Klang erst einmal im System, hat der Betrachter/Interakteur keine Kontrolle mehr über die zuvor veräusserten Laute und Geräusche. Es ist ganz klar, dass das Veräusserte nun medial vermittelt wieder in den Raum zurückgegeben wird. Und diese mediale Vermittlung geschieht mit sehr deutlichen Verfremdungen und Bedeutungsverschiebungen, wenn wir den Klängen eine Bedeutung überhaupt zugestehen wollten. Der Klang steht auch nicht für sich alleine, sondern tritt in Beziehungen zu anderen Klängen, die sich ebenfalls im System aufhalten. Zumeist haben wir als Betrachter nun auch Probleme, der Entwicklung eines bestimmten Klanges zu folgen; vielmehr entsteht ein (mitunter kakophonisches) Ganzes.

Wird ein neuer Klang aufgenommen, so erfassen die Mikrophone natürlich nicht nur auditive Live-Ereignisse, sondern auch die Klänge, die bereits vom System über die Lautsprecher ausgegeben werden. Das System arbeitet also auch mit Rekursion und aktuelle und zukünftige Zustände hängen immer auch von der Geschichte, also von den vorherigen Zuständen ab. Diese Beziehung des aktuellen Zustandes zu vorherigen Zuständen lässt also den Installationsraum auch zu einer Art Gedächtnisraum (Memory Room) werden. In verschiedenen Räumen (mit unterschiedlicher Geschichte) tönte die Installation jeweils anders.

Gleichzeitig nimmt das System mitunter auch elektromagnetische Wellen auf, die wir Menschen gar nicht wahrnehmen. Hierzu gehören, je nach Standort der Installation, Radiowellen oder auch Signale von Mobilfunktelefonen. Doch auch diese Wellen gehören natürlich zu dem Raum und gehören zu seiner Geschichte.

Das System verhält sich durchaus widerborstig, könnte man sagen, wenn man es mit einem menschlichen Attribut charakterisieren wollte. Das System ist in keiner Weise über seine mögliche Wirkung auf uns informiert. Es weiss nichts über die Tonhöhen der Klänge, über Harmonie, über einen etwaigen Rhythmus und so weiter. Es ist weitestgehend „blind“ uns gegenüber. Das System versteht also seine Umgebung nur bedingt und die Menschen in der Installation können die gehörten Klänge auch nicht immer zu einem für sie ästhetischen Klangraum zusammenfügen und also das System auch nicht immer verstehen. Wir haben Probleme, es zu kontrollieren und wir können es nicht in der Art bespielen, wie wir ein Musikinstrument bespielen können.

Trotzdem gibt es Momente, in denen uns die entstehenden Klangästhetiken verwundern, ob ihrer schroffen und schönen Art. In solchen Momenten hat die Verständigung zwischen uns und dem System funktioniert, obwohl niemand viel über den anderen weiß.

Die Wahl des Titels schliesst sich den vorangegangenen Bemerkungen an. Ich bin mir nicht sicher, ob ich mit dieser Wahl in zum Beispiel einem halben Jahr noch zufrieden bin, aber momentan arbeite ich mit diesem: My Marvellous Impression Management Impotence. Ein wenig holprig kann man dies vielleicht mit „Meine wundersame

„Inkompetenz meinen Eindruck zu steuern“ übersetzen. Den Eindruck also, den ich auf andere Menschen machen möchte, aber nicht kontrollieren kann oder meine Unfähigkeit, das Bild, in dem ich gesehen werden möchte, zu bestimmen. Und tatsächlich glaube ich, dass man diesen Eindruck oder dieses Bild nur in einem sehr begrenzten Rahmen kontrollieren kann, egal ob dieses Bild medial vermittelt oder live dargestellt und von anderen aufgenommen wird. Das Bild, welches man von seiner Selbstdarstellung hat, kann von anderen nur anders gesehen werden, da diese anderen Menschen sich eine von der eigenen verschiedene Wirklichkeit konstruieren und Dinge aufgrund ihrer Geschichte und physischen und psychischen Konstitution anders wahrnehmen und erfinden. Und genau dies tut die Installation: Das von den Mikrofonen aufgenommene Klangmaterial zu bearbeiten, zu mischen, etwas Neues zu konstruieren; und stets spielen bei dieser Konstruktion frühere Zustände des Systems mit hinein. Das System kreiert sich aus dem Wenigen, was es aus der Umgebung bekommt, seine eigene Realität. Und diese können wir manchmal verstehen (es gibt ästhetische Klangräume) oder auch nicht (es gibt ein kakophonisches Ganzes).

Das Wort „marvellous“ im Titel deutet jedoch an, dass diese radikal konstruktivistische Sichtweise nicht notwendigerweise schlecht sein muss, sondern dass man sich vielleicht um sein „Impression Management“ gar nicht so viel scheren muss und etwas mehr Gelassenheit und Freiheit gewinnt, indem man die eigene Hilflosigkeit ein Stück weit anerkennt.

## Dankeschön

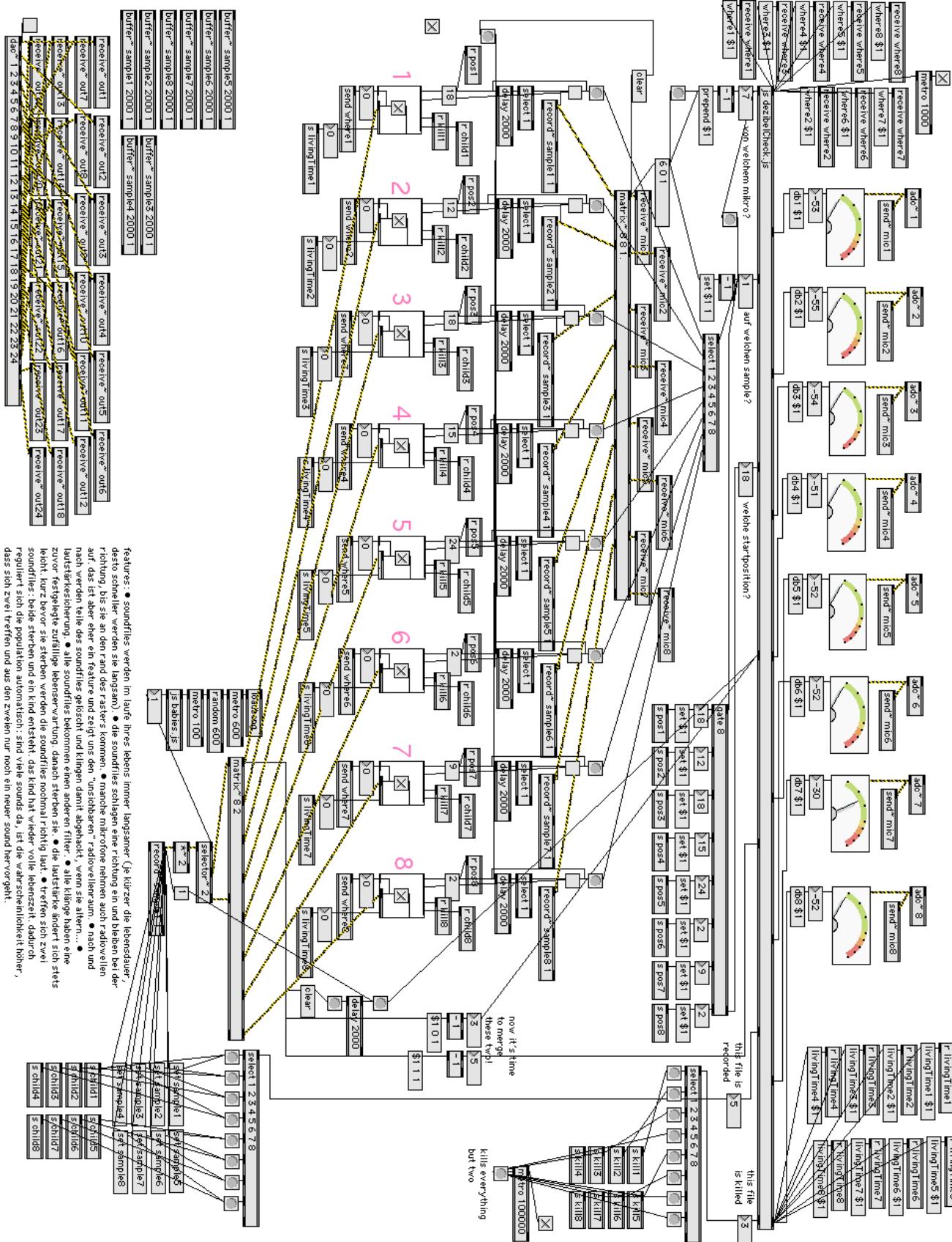
|| Diese Abschlussarbeit ist nicht im luftleeren Raum entstanden und ohne die Infrastruktur, die eine Kunsthochschule bieten kann, wäre es nicht gegangen. Speziell die folgenden Menschen haben mir geholfen und ich bin ihnen zu Dank verpflichtet: Christoph Lischka war mein Betreuer und Kilian Schwoon mein Zweitbetreuer für diese Arbeit. Jukka Böhm, unser Werkstattleiter an der Hochschule, hat mir oft bei Problemen geholfen und Katrin Caspar hat zeitgleich mit mir an ihrer Abschlussarbeit gesessen und mich immer wieder motiviert. Vielen Dank an Euch!

## **Ehrenwörtliche Erklärung**

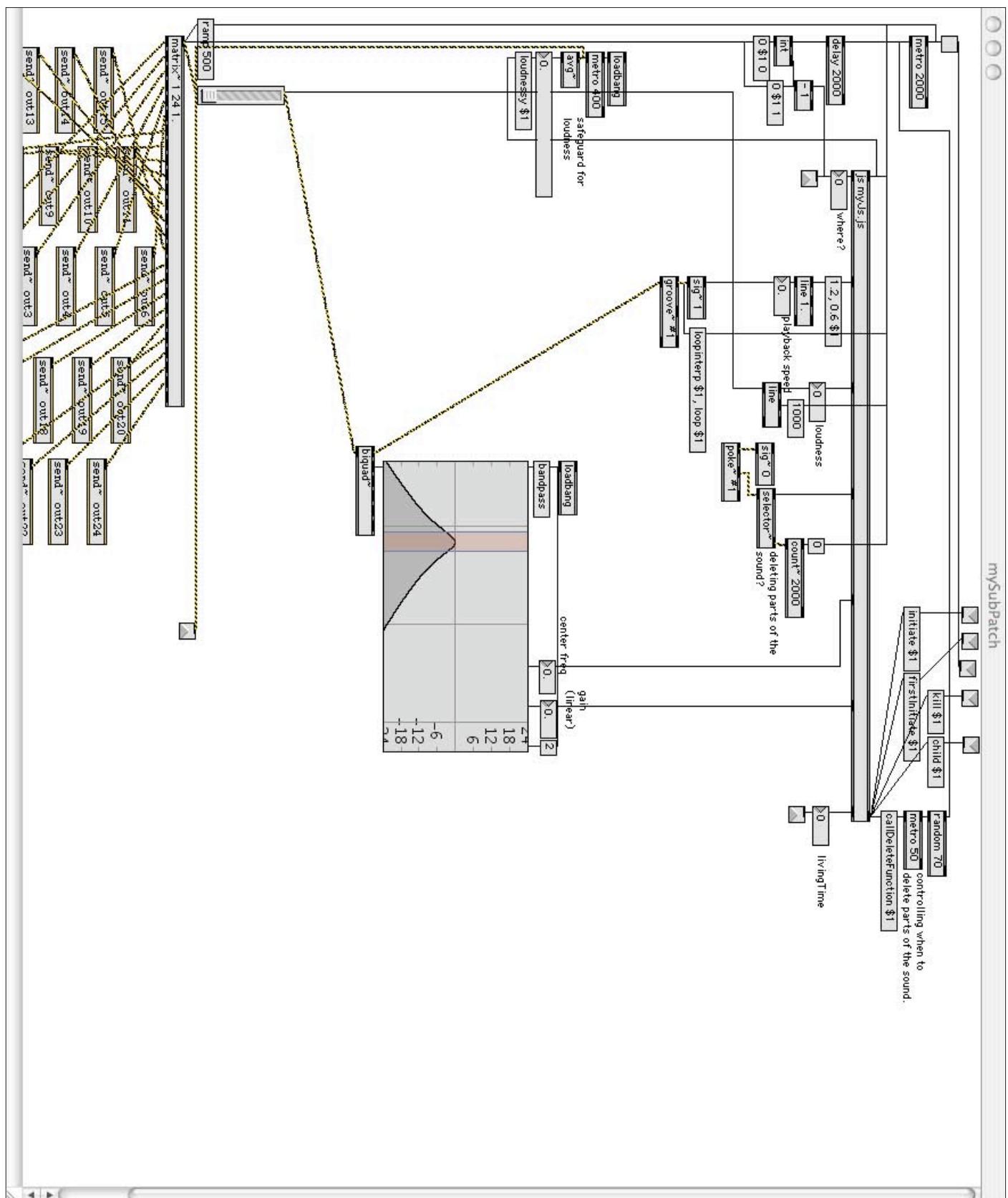
Hiermit erkläre ich, dass ich die vorliegende Arbeit und die eigentliche Installation selbstständig und ohne andere als mit den angegebenen Hilfestellungen angefertigt habe. Soweit ich es weiss, wurde die Arbeit bisher nicht in gleicher oder ähnlicher Form einer Prüfungsbehörde vorgelegt und ist auch bislang nicht veröffentlicht worden.

# Anhang

|| Screenshot des Max/MSP-Patches „22“:



## || Screenshot des Max/MSP-Patches „mySubPatch“:



## || Code der Datei „myJs.js“:

```
/////////////////////////////ATTRIBUTE:  
//inlets and outlets and global variables  
inlets = 2;  
outlets = 7;  
  
var x = 0; //where is the soundfile?  
var direction; //where to is the soundfile going? 1=north,2=north-east...  
var directionChangeProb; //how likely is it that the sounds change their direction?  
var loudness = 0; //how loud is the playback?  
var deleting = 0; // is the program deleting parts of the sound?  
var deleteChance = 0.0; // how likely is it that parts of the sound are becoming deleted?  
var filterCenterFrequency;  
var filterGain;  
var livingTime = 0; //what ist the life expectancy of the sound?  
  
var randomNumber;  
var old_x;  
var new_x;  
var verkehrt = 0;  
var minLifeEx = 8; //minimum life expectancy (in 2second-steps!)  
var maxLifeEx = 35;  
  
/////////////////////////////METHODEN:  
function callDeleteFunction(myBang){  
    var randomNumber2 = Math.random();  
    if(deleting == 1){  
        outlet(3,0);  
        deleting = 0;  
    }  
    if(randomNumber2 < deleteChance && deleting == 0){  
        outlet(3,1);  
        deleting = 1;  
    }  
}  
/////////////////////////////  
function loudness(myLoudness){ //safeguard for amplitude...  
    if(myLoudness > 0.0007){  
        loudness = loudness - 2;  
        outlet(2,loudness);  
    }  
}  
/////////////////////////////  
function kill(mymybang){ //the soundfile becomes killed.  
    livingTime = 0;  
}  
/////////////////////////////  
function child(mymybang){ //the soundfile gets a new (probably higher) life expectancy.  
    livingTime = getRandomInt(minLifeEx,maxLifeEx);  
    outlet(6,livingTime);  
    loudness = loudness + 3;  
    deleteChance = 0.0;  
}  
/////////////////////////////  
function firstInitiate(myBang){  
    loudness = 0;  
    deleteChance = 0.0;  
    outlet(2,loudness);  
  
    //deciding about the life expectancy...  
    livingTime = getRandomInt(minLifeEx,maxLifeEx);  
    outlet(6,livingTime);  
  
    //deciding about the filter properties...  
    var decision = Math.random();  
    if(decision < 0.5){  
        filterCenterFrequency = getRandomInt(200,400);  
        filterGain = 2;  
    }  
    else if(decision >= 0.5 && decision < 0.75){  
        filterCenterFrequency = getRandomInt(400,1000);  
        filterGain = 1;  
    }  
    else if(decision >= 0.7){  
        filterCenterFrequency = getRandomInt(1000,4000);  
        filterGain = 1;  
    }  
    outlet(4,filterCenterFrequency);  
    outlet(5,filterGain);  
}  
/////////////////////////////  
function initiate(a){ //now the recording is ended and this function gets the number, where the sound should start from...  
    x = a;  
    new_x = x;  
  
    deleteChance = 0.0;
```

```

loudness = 130;
directionChangeProb = 0;

outlet(1,livingTime*2*1000); //telling the time in which the soundfile has to slow down.

//setting the initial direction...
var d = Math.random();
if(d <= 1/6){direction = 1;}
else if(d >= 1/6 && d <= 2/6){direction = 2;}
else if(d >= 2/6 && d <= 3/6){direction = 3;}
else if(d >= 3/6 && d <= 4/6){direction = 4;}
else if(d >= 4/6 && d <= 5/6){direction = 5;}
else if(d >= 5/6){direction = 6;}

}

///////////////////////////////
function setDirection(b){
    direction = b;
}

/////////////////////////////
function getRandomInt(min, max){
    return Math.floor(Math.random() * (max - min + 1)) + min;
}

/////////////////////////////
//what to do when receiving a BANG:
function bang(){
    deleteChance = deleteChance + 0.004;
    if(livingTime > 0){           //the soundfile becomes older...
        livingTime--;
    }
    if(livingTime == 3){loudness = loudness+8;} //shortly before the sound dies it becomes very loud!
    if(livingTime == 2){loudness = loudness+10;}
    if(livingTime == 1){loudness = loudness+12;}

    if(livingTime == 0){      //now the soundfile is dead.
        loudness = 0;
        x = 0;
        new_x = 0;
    }
    outlet(6,livingTime);

    var anotherRandomNumber = getRandomInt(-2,2); //changing the amplitude slightly.
    loudness = loudness + anotherRandomNumber ;

    old_x = x;
    x = new_x;

}

/////////////////////////////
//decpcion about where the soundfile is going...
randomNumber = Math.random();
if (x == 15){
    //direction has to change...
    var myRandomNumber = Math.random();
    if(myRandomNumber <= 1/3){new_x=11; direction = 2;}
    if(myRandomNumber >= 1/3 && myRandomNumber <= 2/3){new_x=10; direction = 3;}
    if(myRandomNumber >= 2/3){new_x=14; direction = 4;}
}
else if (x == 14){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x = 15; direction = 1;}
        if(direction == 4){new_x = 13; direction = 4;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb || direction == 5 || direction == 6 || direction == 2 || direction == 3){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/4){new_x = 15; direction = 1;}
        if(myRandomNumber >= 1/4 && myRandomNumber <= 2/4){new_x = 10; direction = 2;}
        if(myRandomNumber >= 2/4 && myRandomNumber <= 3/4){new_x = 9; direction = 3;}
        if(myRandomNumber >= 3/4){new_x = 13; direction = 4;}
    }
}
else if (x == 13){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x=14; direction = 1;}
        if(direction == 4){new_x=12; direction = 4;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb || direction == 5 || direction == 6 || direction == 2 || direction == 3){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/4){new_x=14; direction = 1;}
        if(myRandomNumber >= 1/4 && myRandomNumber <= 2/4){new_x=9; direction = 2;}
        if(myRandomNumber >= 2/4 && myRandomNumber <= 3/4){new_x=8; direction = 3;}
        if(myRandomNumber >= 3/4){new_x=12; direction = 4;}
    }
}
else if (x == 12){
    //direction has to change...
    var myRandomNumber = Math.random();
    if(myRandomNumber <= 1/3){new_x=13; direction = 1;}
}

```

```

        if(myRandomNumber >= 1/3 && myRandomNumber <= 2/3){new_x=8;direction = 2;}
        if(myRandomNumber >= 2/3){new_x=7;direction = 3;}
    }
    else if (x == 11){
        //no change in direction...
        if(randomNumber > directionChangeProb){
            if(direction == 5){new_x=15; direction = 5;}
            if(direction == 2){new_x=6; direction = 2;}
        }
        //direction has to change...
        if(randomNumber <= directionChangeProb || direction == 1 || direction == 6 || direction == 3 || direction == 4){
            var myRandomNumber = Math.random();
            if(myRandomNumber <= 1/4){new_x=15;direction = 5;}
            if(myRandomNumber >= 1/4 && myRandomNumber <= 2/4){new_x=10;direction = 4;}
            if(myRandomNumber >= 2/4 && myRandomNumber <= 3/4){new_x=5;direction = 3;}
            if(myRandomNumber >= 3/4){new_x=6;direction = 2;}
        }
    }
    else if (x == 10){
        //no change in direction...
        if(randomNumber > directionChangeProb){
            if(direction == 1){new_x=11; direction = 1;}
            if(direction == 2){new_x=5; direction = 2;}
            if(direction == 3){new_x=4;direction =3;}
            if(direction == 4){new_x=9;direction = 4;}
            if(direction == 5){new_x=14;direction =5;}
            if(direction == 6){new_x=15;direction = 6;}
        }
        //direction has to change...
        if(randomNumber <= directionChangeProb){
            var myRandomNumber = Math.random();
            if(myRandomNumber <= 1/6){new_x=11;direction = 1;}
            if(myRandomNumber >= 1/6 && myRandomNumber <= 2/6){new_x=15;direction = 6;}
            if(myRandomNumber >= 2/6 && myRandomNumber <= 3/6){new_x=14;direction =5;}
            if(myRandomNumber >= 3/6 && myRandomNumber <= 4/6){new_x=9;direction =4;}
            if(myRandomNumber >= 4/6 && myRandomNumber <= 5/6){new_x=4;direction =3;}
            if(myRandomNumber >= 5/6){new_x=5;direction =2;}
        }
    }
    else if (x == 9){
        //no change in direction...
        if(randomNumber > directionChangeProb){
            if(direction == 1){new_x=10; direction = 1;}
            if(direction == 2){new_x=4; direction = 2;}
            if(direction == 3){new_x=3;direction =3;}
            if(direction == 4){new_x=8;direction = 4;}
            if(direction == 5){new_x=13;direction =5;}
            if(direction == 6){new_x=14;direction = 6;}
        }
        //direction has to change...
        if(randomNumber <= directionChangeProb){
            var myRandomNumber = Math.random();
            if(myRandomNumber <= 1/6){new_x=10;direction =1;}
            if(myRandomNumber >= 1/6 && myRandomNumber <= 2/6){new_x=14;direction =6;}
            if(myRandomNumber >= 2/6 && myRandomNumber <= 3/6){new_x=13;direction =5;}
            if(myRandomNumber >= 3/6 && myRandomNumber <= 4/6){new_x=8;direction =4;}
            if(myRandomNumber >= 4/6 && myRandomNumber <= 5/6){new_x=3;direction =3;}
            if(myRandomNumber >= 5/6){new_x=4;direction =2;}
        }
    }
    else if (x == 8){
        //no change in direction...
        if(randomNumber > directionChangeProb){
            if(direction == 1){new_x=9; direction = 1;}
            if(direction == 2){new_x=3; direction = 2;}
            if(direction == 3){new_x=2;direction =3;}
            if(direction == 4){new_x=7;direction = 4;}
            if(direction == 5){new_x=12;direction =5;}
            if(direction == 6){new_x=13;direction = 6;}
        }
        //direction has to change...
        if(randomNumber <= directionChangeProb){
            var myRandomNumber = Math.random();
            if(myRandomNumber <= 1/6){new_x=9;direction =1;}
            if(myRandomNumber >= 1/6 && myRandomNumber <= 2/6){new_x=13;direction =6;}
            if(myRandomNumber >= 2/6 && myRandomNumber <= 3/6){new_x=12;direction =5;}
            if(myRandomNumber >= 3/6 && myRandomNumber <= 4/6){new_x=7;direction =4;}
            if(myRandomNumber >= 4/6 && myRandomNumber <= 5/6){new_x=2;direction =3;}
            if(myRandomNumber >= 5/6){new_x=3;direction =2;}
        }
    }
    else if (x == 7){
        //no change in direction...
        if(randomNumber > directionChangeProb){
            if(direction == 3){new_x=1;direction =3;}
            if(direction == 6){new_x=12;direction = 6;}
        }
        //direction has to change...
        if(randomNumber <= directionChangeProb || direction == 5 || direction == 4 || direction == 1 || direction == 2){
            var myRandomNumber = Math.random();
            if(myRandomNumber <= 1/4){new_x=12;direction =6;}
            if(myRandomNumber >= 1/4 && myRandomNumber <= 2/4){new_x=8;direction =1;}
        }
    }
}

```

```

        if(myRandomNumber >= 2/4 && myRandomNumber <= 3/4){new_x=2;direction =2;}
        if(myRandomNumber >= 3/4){new_x=1;direction =3;}
    }
}

else if (x == 6){
    //direction has to change...
    var myRandomNumber = Math.random();
    if(myRandomNumber <= 1/3){new_x=11;direction =5;}
    if(myRandomNumber >= 1/3 && myRandomNumber <= 2/3){new_x=5;direction =4;}
    if(myRandomNumber >= 2/3){new_x=20;direction =3;}
}

else if (x == 5){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x=6; direction = 1;}
        if(direction == 2){new_x=20; direction = 2;}
        if(direction == 3){new_x=19;direction =3;}
        if(direction == 4){new_x=4;direction = 4;}
        if(direction == 5){new_x=10;direction =5;}
        if(direction == 6){new_x=11;direction = 6;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/6){new_x=6;direction =1;}
        if(myRandomNumber >= 1/6 && myRandomNumber <= 2/6){new_x=11;direction =6;}
        if(myRandomNumber >= 2/6 && myRandomNumber <= 3/6){new_x=10;direction =5;}
        if(myRandomNumber >= 3/6 && myRandomNumber <= 4/6){new_x=4;direction =4;}
        if(myRandomNumber >= 4/6 && myRandomNumber <= 5/6){new_x=19;direction =3;}
        if(myRandomNumber >= 5/6){new_x=20;direction =2;}
    }
}

else if (x == 4){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x=5; direction = 1;}
        if(direction == 2){new_x=19; direction = 2;}
        if(direction == 3){new_x=18;direction =3;}
        if(direction == 4){new_x=3;direction = 4;}
        if(direction == 5){new_x=9;direction =5;}
        if(direction == 6){new_x=10;direction = 6;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/6){new_x=5;direction =1;}
        if(myRandomNumber >= 1/6 && myRandomNumber <= 2/6){new_x=10;direction =6;}
        if(myRandomNumber >= 2/6 && myRandomNumber <= 3/6){new_x=9;direction =5;}
        if(myRandomNumber >= 3/6 && myRandomNumber <= 4/6){new_x=3;direction =4;}
        if(myRandomNumber >= 4/6 && myRandomNumber <= 5/6){new_x=18;direction =3;}
        if(myRandomNumber >= 5/6){new_x=19;direction =2;}
    }
}

else if (x == 3){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x=4; direction = 1;}
        if(direction == 2){new_x=18; direction = 2;}
        if(direction == 3){new_x=17;direction =3;}
        if(direction == 4){new_x=2;direction = 4;}
        if(direction == 5){new_x=8;direction =5;}
        if(direction == 6){new_x=9;direction = 6;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/6){new_x=4;direction =1;}
        if(myRandomNumber >= 1/6 && myRandomNumber <= 2/6){new_x=9;direction =6;}
        if(myRandomNumber >= 2/6 && myRandomNumber <= 3/6){new_x=8;direction =5;}
        if(myRandomNumber >= 3/6 && myRandomNumber <= 4/6){new_x=2;direction =4;}
        if(myRandomNumber >= 4/6 && myRandomNumber <= 5/6){new_x=17;direction =3;}
        if(myRandomNumber >= 5/6){new_x=18;direction =2;}
    }
}

else if (x == 2){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x=3; direction = 1;}
        if(direction == 2){new_x=17; direction = 2;}
        if(direction == 3){new_x=16;direction =3;}
        if(direction == 4){new_x=1;direction = 4;}
        if(direction == 5){new_x=7;direction =5;}
        if(direction == 6){new_x=8;direction = 6;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/6){new_x=3;direction =1;}
        if(myRandomNumber >= 1/6 && myRandomNumber <= 2/6){new_x=8;direction =6;}
        if(myRandomNumber >= 2/6 && myRandomNumber <= 3/6){new_x=7;direction =5;}
        if(myRandomNumber >= 3/6 && myRandomNumber <= 4/6){new_x=1;direction =4;}
        if(myRandomNumber >= 4/6 && myRandomNumber <= 5/6){new_x=16;direction =3;}
    }
}

```

```

        if(myRandomNumber >= 5/6){new_x=17;direction =2;}
    }
}
else if (x == 1){
    //direction has to change...
    var myRandomNumber = Math.random();
    if(myRandomNumber <= 1/3){new_x=7;direction =6;}
    if(myRandomNumber >= 1/3 && myRandomNumber <= 2/3){new_x=2;direction =1;}
    if(myRandomNumber >= 2/3){new_x=16;direction =2;}
}
else if (x == 20){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 3){new_x=24;direction =3;}
        if(direction == 6){new_x=6;direction = 6;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb || direction == 1 || direction == 2 || direction == 4 || direction == 5){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/4){new_x=6;direction =6;}
        if(myRandomNumber >= 1/4 && myRandomNumber <= 2/4){new_x=5;direction =5;}
        if(myRandomNumber >= 2/4 && myRandomNumber <= 3/4){new_x=19;direction =4;}
        if(myRandomNumber >= 3/4){new_x=24;direction =3;}
    }
}
else if (x == 19){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x=20; direction = 1;}
        if(direction == 2){new_x=24; direction = 2;}
        if(direction == 3){new_x=23;direction =3;}
        if(direction == 4){new_x=18;direction = 4;}
        if(direction == 5){new_x=4;direction =5;}
        if(direction == 6){new_x=5;direction = 6;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/6){new_x=20;direction =1;}
        if(myRandomNumber >= 1/6 && myRandomNumber <= 2/6){new_x=5;direction =6;}
        if(myRandomNumber >= 2/6 && myRandomNumber <= 3/6){new_x=4;direction =5;}
        if(myRandomNumber >= 3/6 && myRandomNumber <= 4/6){new_x=18;direction =4;}
        if(myRandomNumber >= 4/6 && myRandomNumber <= 5/6){new_x=23;direction =3;}
        if(myRandomNumber >= 5/6){new_x=24;direction =2;}
    }
}
else if (x == 18){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x=19; direction = 1;}
        if(direction == 2){new_x=23; direction = 2;}
        if(direction == 3){new_x=22;direction =3;}
        if(direction == 4){new_x=17;direction = 4;}
        if(direction == 5){new_x=3;direction =5;}
        if(direction == 6){new_x=4;direction = 6;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/6){new_x=19;direction =1;}
        if(myRandomNumber >= 1/6 && myRandomNumber <= 2/6){new_x=4;direction =6;}
        if(myRandomNumber >= 2/6 && myRandomNumber <= 3/6){new_x=3;direction =5;}
        if(myRandomNumber >= 3/6 && myRandomNumber <= 4/6){new_x=17;direction =4;}
        if(myRandomNumber >= 4/6 && myRandomNumber <= 5/6){new_x=22;direction =3;}
        if(myRandomNumber >= 5/6){new_x=23;direction =2;}
    }
}
else if (x == 17){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x=18; direction = 1;}
        if(direction == 2){new_x=22; direction = 2;}
        if(direction == 3){new_x=21;direction =3;}
        if(direction == 4){new_x=16;direction = 4;}
        if(direction == 5){new_x=2;direction =5;}
        if(direction == 6){new_x=3;direction = 6;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/6){new_x=18;direction =1;}
        if(myRandomNumber >= 1/6 && myRandomNumber <= 2/6){new_x=3;direction =6;}
        if(myRandomNumber >= 2/6 && myRandomNumber <= 3/6){new_x=2;direction =5;}
        if(myRandomNumber >= 3/6 && myRandomNumber <= 4/6){new_x=16;direction =4;}
        if(myRandomNumber >= 4/6 && myRandomNumber <= 5/6){new_x=21;direction =3;}
        if(myRandomNumber >= 5/6){new_x=22;direction =2;}
    }
}
else if (x == 16){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 2){new_x=21; direction = 2;}
    }
}

```

```

        if(direction == 5){new_x=1;direction =5;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb || direction == 3 || direction == 4 || direction == 1 || direction == 6){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/4){new_x=17;direction =1;}
        if(myRandomNumber >= 1/4 && myRandomNumber <= 2/4){new_x=2;direction =6;}
        if(myRandomNumber >= 2/4 && myRandomNumber <= 3/4){new_x=1;direction =5;}
        if(myRandomNumber >= 3/4){new_x=21;direction =2;}
    }
}
else if (x == 24){
    //direction has to change...
    var myRandomNumber = Math.random();
    if(myRandomNumber <= 1/3){new_x=20;direction =6;}
    if(myRandomNumber >= 1/3 && myRandomNumber <= 2/3){new_x=19;direction =5;}
    if(myRandomNumber >= 2/3){new_x=23;direction =4;}
}
else if (x == 23){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x=24; direction = 1;}
        if(direction == 4){new_x=22;direction = 4;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb || direction == 2 || direction == 3 || direction == 5 || direction == 6){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/4){new_x=24;direction =1;}
        if(myRandomNumber >= 1/4 && myRandomNumber <= 2/4){new_x=19;direction =6;}
        if(myRandomNumber >= 2/4 && myRandomNumber <= 3/4){new_x=18;direction =5;}
        if(myRandomNumber >= 3/4){new_x=22;direction =4;}
    }
}
else if (x == 22){
    //no change in direction...
    if(randomNumber > directionChangeProb){
        if(direction == 1){new_x=23; direction = 1;}
        if(direction == 4){new_x=21;direction = 4;}
    }
    //direction has to change...
    if(randomNumber <= directionChangeProb || direction == 2 || direction == 3 || direction == 5 || direction == 6){
        var myRandomNumber = Math.random();
        if(myRandomNumber <= 1/4){new_x=23;direction =1;}
        if(myRandomNumber >= 1/4 && myRandomNumber <= 2/4){new_x=18;direction =6;}
        if(myRandomNumber >= 2/4 && myRandomNumber <= 3/4){new_x=17;direction =5;}
        if(myRandomNumber >= 3/4){new_x=21;direction =4;}
    }
}
else if (x == 21){
    //direction has to change...
    var myRandomNumber = Math.random();
    if(myRandomNumber <= 1/3){new_x=22;direction =1;}
    if(myRandomNumber >= 1/3 && myRandomNumber <= 2/3){new_x=17;direction =6;}
    if(myRandomNumber >= 2/3){new_x=16;direction =5;}
}
////////////////////////////////////////////////////////////////

outlet(0, x); //outlet number 0, send the place x
outlet(2, loudness);
////////////////////////////////////////////////////////////////

//wenn der soundfile von einem speaker zurück zum vorherigen
//speaker wandert, wird er rückwärts abgespielt.
/*if(old_x == new_x){
    generalPlaybackSpeed = -generalPlaybackSpeed;
}
*/
}

```

## || Code der Datei „dezibelChecki.js“:

```
//This part decides about where and when to record...

inlets = 9;
outlets = 7;

var input1, input2, input3, input4, input5, input6, input7, input8;
var average; //the average input
var where11 = 31; //where are all the soundfiles?
var where22 = 32;
var where33 = 33;
var where44 = 34;
var where55 = 35;
var where66 = 36;
var where77 = 37;
var where88 = 38;
var minimumInput = -45;//how strong does the input at least have to be before recording starts?
var mergingProb = 1;
var mergingPossibleNow = 0;
var recordingPossible = 2; //is recording allowed? it's allowed every two seconds!!
var livingTime11, livingTime22, livingTime33, livingTime44, livingTime55, livingTime66, livingTime77, livingTime88;
var averagePlus = 15; //how much stronger has the input to be before becoming recorded than the average?

/////////////////////////////
function getRandomInt(min, max){
    return Math.floor(Math.random() * (max - min + 1)) + min;
}

/////////////////////////////
//deciding whether or not two soundfiles should merge...
function babies(soundfile1, soundfile2){
    if(mergingProb > Math.random()){
        outlet(3,soundfile1);
        outlet(4,soundfile2);
        if(Math.random() > 0.5){ //soundfile1 becomes recorded
            outlet(5, soundfile1);
            outlet(6, soundfile2); //this file becomes killed.
        }
        else { //soundfile2 becomes recorded
            outlet(5, soundfile2);
            outlet(6, soundfile1); //this file becomes killed.
        }
    }
}

/////////////////////////////
//getting all the decibel values from the different microphones:
//in the first (now following) function it is decided if recording happens or not...
function db1(a){
    input1 = a;

    average = (input1+input2+input3+input4+input5+input6+input7+input8) / 8;

    if(livingTime11 == 0 || livingTime22 == 0 || livingTime33 == 0 || livingTime44 == 0 || livingTime55 == 0 || livingTime66 == 0 || livingTime77 == 0 || livingTime88 == 0){
        if(recordingPossible >= 2){
            if(input1 > minimumInput && input1 > average +averagePlus /*&& where11 != 21 && where2 != 21 && where33 != 21 && where44 != 21 && where55 != 21 && where66 != 21 && where77 != 21 && where88 != 21*/){
                if(livingTime11==0){
                    outlet(0,1); //record from micro1
                    outlet(1,1); //record to sample1
                    outlet(2,21); //start soundfile at speaker21
                    recordingPossible = 0;
                }
                else if(livingTime22==0){
                    outlet(0,1);
                    outlet(1,2);
                    outlet(2,21);
                    recordingPossible = 0;
                }
                else if(livingTime33==0){
                    outlet(0,1);
                    outlet(1,3);
                    outlet(2,21);
                    recordingPossible = 0;
                }
                else if(livingTime44==0){
                    outlet(0,1);
                    outlet(1,4);
                    outlet(2,21);
                    recordingPossible = 0;
                }
                else if(livingTime55==0){
                    outlet(0,1);
                    outlet(1,5);
                    outlet(2,21);
                    recordingPossible = 0;
                }
                else if(livingTime66==0){
                    outlet(0,1);
                }
            }
        }
    }
}
```

```

        outlet(1,6);
        outlet(2,21);
        recordingPossible = 0;
    }
    else if(livingTime77==0){
        outlet(0,1);
        outlet(1,7);
        outlet(2,21);
        recordingPossible = 0;
    }
    else if(livingTime88==0){
        outlet(0,1);
        outlet(1,8);
        outlet(2,21);
        recordingPossible = 0;
    }
}
else if(input2 > minimumInput && input2 > average +averagePlus /*&& where11 != 15 && where22 != 15 && where33 != 15 && where44 != 15 &&
where55 != 15 && where66 != 15 && where77 != 15 && where88 != 15*/){
    if(livingTime11==0){
        outlet(0,2); //record from micro2
        outlet(1,1); //record to sample1
        outlet(2,15); //start soundfile at speaker15
        recordingPossible = 0;
    }
    else if(livingTime22==0){
        outlet(0,2);
        outlet(1,2);
        outlet(2,15);
        recordingPossible = 0;
    }
    else if(livingTime33==0){
        outlet(0,2);
        outlet(1,3);
        outlet(2,15);
        recordingPossible = 0;
    }
    else if(livingTime44==0){
        outlet(0,2);
        outlet(1,4);
        outlet(2,15);
        recordingPossible = 0;
    }
    else if(livingTime55==0){
        outlet(0,2);
        outlet(1,5);
        outlet(2,15);
        recordingPossible = 0;
    }
    else if(livingTime66==0){
        outlet(0,2);
        outlet(1,6);
        outlet(2,15);
        recordingPossible = 0;
    }
    else if(livingTime77==0){
        outlet(0,2);
        outlet(1,7);
        outlet(2,15);
        recordingPossible = 0;
    }
    else if(livingTime88==0){
        outlet(0,2);
        outlet(1,8);
        outlet(2,15);
        recordingPossible = 0;
    }
}
else if(input3 > minimumInput && input3 > average +averagePlus /*&& where11 != 24 && where22 != 24 && where33 != 24 && where44 != 24 &&
where55 != 24 && where66 != 24 && where77 != 24 && where88 != 24*/){
    if(livingTime11==0){
        outlet(0,3); //record from micro3
        outlet(1,1); //record to sample1
        outlet(2,21); //start soundfile at speaker24
        recordingPossible = 0;
    }
    else if(livingTime22==0){
        outlet(0,3);
        outlet(1,2);
        outlet(2,24);
        recordingPossible = 0;
    }
    else if(livingTime33==0){
        outlet(0,3);
        outlet(1,3);
        outlet(2,24);
        recordingPossible = 0;
    }
    else if(livingTime44==0){
        outlet(0,3);
        outlet(1,4);
        outlet(2,24);
    }
}

```

```

        recordingPossible = 0;
    }
    else if(livingTime55==0){
        outlet(0,3);
        outlet(1,5);
        outlet(2,24);
        recordingPossible = 0;
    }
    else if(livingTime66==0){
        outlet(0,3);
        outlet(1,6);
        outlet(2,24);
        recordingPossible = 0;
    }
    else if(livingTime77==0){
        outlet(0,3);
        outlet(1,7);
        outlet(2,24);
        recordingPossible = 0;
    }
    else if(livingTime88==0){

```

|| Code der Datei „babies.js“:

```

inlets = 1;
outlets = 1;

function bang(){
    var randomNumber = Math.random();
    if(randomNumber > 0.5){
        outlet(0,1);
    }
    if(randomNumber <= 0.5){
        outlet(0,2);
    }
}

```