

Deckblatt einer wissenschaftlichen Bachelorarbeit

Familienname: **Cladders**

Vorname: **Jakob**

Matrikelnummer: **0931507**

Studienkennzahl: **213**

Thema der Arbeit:

EarSCAPE – Ein Audio-Only Game

Angefertigt in der Lehrveranstaltung: **Computermusik und Multimedia**

Name der Lehrveranstaltung

Vorgelegt am: **07.10.2020**

Datum

Beurteilt durch: **Johannes Zmoelnig, DI**

Leiter_in der Lehrveranstaltung

*E*ar*SCAPE*

ein Audio-Only Game



Institut für Elektronische Musik
und Akustik

Kunstuniversität Graz



Bachelorarbeit

Im Rahmen des *Computermusik und Multimedia* Seminars

von Jakob Cladders

Betreuer: DI IOhannes M. Zmoelnig

Abstract

You regain consciousness but you can't see anything. It's like you're blind. You know you have to get out of here, but all you can rely on is your sense of hearing...

EarSCAPE ist ein Audio-Game aus dem Genre der sogenannten *Escape-Games*.

Bei diesen muss der Spieler üblicherweise durch das Lösen von verschiedenen Rätseln aus einem oder mehreren Räumen entkommen.

Bei *EarSCAPE* befindet sich der Spieler in *einem* Raum mit *einem* Ausgang und für eine erfolgreiche Flucht muss er innerhalb eines begrenzten Zeitraums verschiedenen auditiven Hinweisen folgen. Dabei gibt es keine visuelle Unterstützung.

Die auditive Orientierung basiert auf binauraler Spatialisierung mittels *Head Related Transfer Functions* (HRTFs), was das Benutzen von Kopfhörern notwendig macht.

Die Implementierung erfolgte in Pure Data (Pd) auf eine Art und Weise, die dem Anwender die Möglichkeit geben soll, eigene Levels zu schaffen indem er *Audio-Events* im Raum platzieren und ihnen eigene Samples zuordnen kann.

Da es keinen visuellen Output gibt, ist das Spiel auch für Personen mit Sehbeeinträchtigungen bis zur Blindheit spielbar.

EarSCAPE is an audio-game in the genre of so called *escape games*.

Usually the player has to succeed in different tasks to finally escape from one or more rooms.

In *EarSCAPE* the player is trapped in a single room with only one exit and has to follow different auditive cues to successfully escape the room within a given time.

There is no visual support. The auditive orientation is based on binaural spatialization using *Head Related Transfer Functions* (HRTFs), so the use of headphones is obligatory. The implementation was done in Pure Data (Pd) in a way, that it is accessible to the user to create own levels by placing *audio events* in a room and assigning own samples to it.

For its non-visual output, the game is also playable for people without the ability to see.

Inhaltsverzeichnis

1	Einleitung	4
2	Motivation	4
2.1	Audio Games	4
2.2	Inspiration	5
3	Konzeption	6
3.1	Spielkonzept	6
3.2	Technische Konzeption	8
3.2.1	HRIR - HRTF	8
3.2.2	KEMAR	9
3.2.3	Pure Data	10
3.2.4	[earplug~]	11
4	Umsetzung	11
4.1	Beschreibung	11
4.2	Steuerung	12
4.3	Einstellungen	13
4.4	Raumklang	14
4.5	Audio-Events	15
4.5.1	Platzierung von Audio Events	16
4.5.2	Zuordnung von Samples zu Audio-Events	16
4.5.3	Bang Management	16
4.6	Erstellen eigener Levels	17
5	Rück- und Ausblick	19
	Literatur	21

1 Einleitung

Da die vorliegende Arbeit im Seminar *Computermusik und Multimedia* entstanden ist, wird im Kapitel Motivation (Kap. 2) auf die Begrifflichkeit *Audio Game*, dem Thema des Seminars, eingegangen und die eigene Inspiration zum Entwurf eines solchen dargestellt. Das anschließende Kapitel Konzeption gliedert sich in zwei Unterkapitel. Im ersten Teil, Spielkonzept, wird geklärt um was für eine Art Spiel es sich bei *EarSCAPE* handelt, im zweiten Teil, Technische Konzeption, wird dargestellt, auf welchen Grundlagen dieses Konzept realisiert werden soll. Der tatsächlichen Realisierung widmet sich dann das Kapitel Umsetzung. Hier werden auch grundlegende Bausteine der Programmierung von *EarSCAPE* dargestellt und darauf eingegangen wie Anwender eigene Level erstellen können. Als Abschluss dieser Arbeit wird im Kapitel Rück- und Ausblick herausgearbeitet, welche Änderungen und Probleme sich bei der Umsetzung ergeben haben und welche zusätzlichen Features und Verbesserungen für die Zukunft wünschenswert wären.

2 Motivation

2.1 Audio Games

Laut Wikipedia ist ein *Audio Game* ein elektronisches Spiel, das sich auf auditives und taktiles Feedback beschränkt, also komplett auf visuelles Feedback verzichtet (vgl.[10]). Diese klare Abspaltung vom Visuellen und somit vom *Video Game* ist sicherlich darin begründet, dass viele *Audio Games* als sogenannte *Blind-Accessible Games* designed wurden, also aus dem Bedürfnis heraus, elektronische Spiele auch für Menschen mit Sehbehinderung zugänglich zu machen.

Betrachtet man *Audio Games* als entsprechendes Gegenstück zu *Video Games* macht diese klare Trennung wenig Sinn. Im Umkehrschluss müsste ein *Video Game* auf Audio als zusätzliche Möglichkeit, *Game Content* zu vermitteln, verzichten. Man stelle sich einen First-Person Shooter vor, bei dem man nicht hören kann, ob auf die Spielfigur geschossen wird oder eine Handgranate o.ä. in der Nähe landet.

Um Verwirrung zu vermeiden sollten deshalb Spiele, die komplett ohne visuelles Feedback auskommen, besser *Audio-Only Games* genannt werden.

Sofern man überhaupt zwischen *Video Game* und *Audio Game* unterscheiden möchte, sollte man dies viel mehr entsprechend der Rolle für das jeweilige Spielkonzept tun.

Um beim Beispiel des klassischen First-Person Shooters zu bleiben: üblicherweise werden zwar über den Ton zusätzliche Hinweise gegeben, die grundsätzliche Orientierung, das Zielen etc. basiert aber fast ausschließlich auf visuellen Informationen¹. Einige „neue“ Spiele, vor allem für Smartphones und Tablets, setzen auf alte Spielkonzepte vergleichbar mit Super Mario (2D Jump'n'Run Spiele) und verbinden diese mit Steuerungsmöglichkeiten über die eigene Stimme, ermöglicht durch die integrierten Mikrofone². Ausgewertet wer-

¹An dieser Stelle sei angemerkt, dass es durchaus auch Audio-Only First-Person Shooter gibt, bspw. *Shades of Doom* (in Anlehnung an die Ego-Shooter Reihe *Doom*)

²bspw. *Yasuhati* oder *Scream Go*

den dazu meist zeitlich gefenstertere Schallintensitäten oder Tonhöhen. Hierbei fällt eine Einteilung in *Video-* oder *Audio Game* schwer und ist wahrscheinlich auch wenig sinnvoll. Im Rahmen des Seminars wurde sich darauf geeinigt, dass unter dem Oberbegriff *Audio Game* alle Spiele einzuordnen sind, bei denen Audio zentraler Bestandteil des Spielkonzepts oder der Spielmechanik ist und somit der Begriff *Audio Game* nicht in klarer Abgrenzung zum *Video Game* zu sehen ist.

2.2 Inspiration

Als Zielsetzung des Seminars sollte jeder Teilnehmer ein eigenes *Audio Game* entwickeln. Bei der entsprechenden Hintergrundrecherche wurde bald deutlich, dass viele der Spiele durch ihre plattformspezifischen Anforderungen einen großen Nutzerkreis von vornherein ausschließen. Daher wurde der Entschluss gefasst *EarSCAPE* in *Pure Data* (kurz: *Pd*; siehe Kap. 3.2.3) zu programmieren, um so weitestgehend plattformunabhängig zu bleiben. Da relativ wenige der verfügbaren *Audio Games* auf linux-basierten Plattformen funktionieren, wurden für die Recherche zu weiten Teilen Browserspiele und Hintergrundinformationen genutzt.

Von den getesteten *Audio Games* waren vor allem diejenigen interessant, bei denen der primäre Informationsgehalt durch Audio vermittelt wurde. Beispielsweise gibt es verschiedene Neuauflagen des Sega Klassikers *Frogger* von 1981, bei denen versucht wurde, den ursprünglich rein visuell gestalteten Informationsfluss durch einen auditiven zu ersetzen. *Frogger* ist ein *Arcade-Spiel*, bei dem die Spielfigur (ein Frosch) eine vielbefahrene, mehrspurige Straße sowie einen Fluss mit verschiedenen Hindernissen überqueren muss (Abb. 1). Das ist vor allem deswegen mit einer gewissen Geschicklichkeit verbunden, weil sich die Objekte auf den verschiedenen Spuren mit unterschiedlichen Geschwindigkeiten bewegen.



Abbildung 1: Arcade Game *Frogger*

Bei den *Audio Game* Varianten des Spiels wurde das Straßen-Szenario realisiert, wobei die Standortpositionen der Fahrzeuge durch ein dynamisches Panning von passenden

Audiosamples und einer Lautstärkeregelung entsprechend der Entfernung der Fahrzeuge vermittelt werden.

Diese Form der Stereolokalisation, die sich rein auf interaurale Pegeldifferenzen (*ILD - Interaural Level Differences*) stützt, funktioniert für die Links-Rechts-Ortung mit den benutzten Samples gut. Für *Frogger*, mit seiner zweidimensionalen Spielebene, wäre auch eine Unterscheidung zwischen Vorne und Hinten wichtig, die aber wegen der Uneindeutigkeit der gegebenen auditiven Informationen nicht möglich ist.

Die Tatsache, dass sich, unter der Prämisse eines mit Kopfhörern am Computer sitzenden Spielers, wesentlich besser funktionierende Möglichkeiten zur Lokalisation realisieren lassen, haben den Autor dieser Arbeit dazu bewogen, ein auf binauraler Spatialisierung beruhendes *Audio Game* zu designen.

Um auszuloten wie gut die Orientierung in einem virtuellen Raum auf diese Weise funktioniert, sollte es außerdem ein *Audio-Only* Spiel sein.

Mit diesen Rahmenbedingungen und einigem Ausprobieren mit dem *Pd-External [earplug~]* und verschiedenem Audiomaterial entstand so das Spielkonzept von *EarSCAPE*.

3 Konzeption

Im Unterkapitel *Spielkonzept* (Kap. 3.1) wird kurz die Frage behandelt, was ein *Spiel* eigentlich ausmacht, um darauf aufbauend ein Spielkonzept zu entwickeln. Im Unterkapitel *Technische Konzeption* (Kap. 3.2) geht es darum wie dieses Spielkonzept realisiert werden soll, also um die theoretische Grundlage zur Spatialisierung, die benutzte Programmiersprache und zusätzliche Librarys.

3.1 Spielkonzept

A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome.[8, p.96]

Ein *Spiel* ist also ein System, das für den Spieler

- Regeln
- einen (künstlichen) Konflikt
- und ein zählbares Ergebnis

bereitstellt.

Wie der Name *EarSCAPE* andeutet (Wortschöpfung aus **Ear** und **Escape**) handelt es sich bei dem entworfenen *Audio Game* um ein Spiel aus dem Genre der *Escape Games*, einem Sub-Genre der *Adventures*. Bei einem solchen muss sich der Spieler üblicherweise aus einem geschlossenen Raum befreien, typischerweise durch das Lösen verschiedener

Rätsel, um so einen Ausgang zu finden oder eine Tür zu öffnen. Als erstes *Escape Game* gilt das *Text Adventure* ‚*Behind closed doors*‘ von John Wilson aus dem Jahr 1988. In den 90er Jahren entstanden weitere *Escape Games* als *Point-and-Click* Varianten³, bevor das Genre 2003 durch ‚*Crimson Room*‘ von Toshimitsu Takagi größere Popularität erlangte (vgl.[11]). Inzwischen gibt es in vielen Städten sogenannte *Live Escape Games* bzw. *Escape Rooms*, also Umsetzungen des Spielkonzepts als reales Gruppenspiel.

Regeln im Sinne eines bestehenden Regelwerks, in dem sie aufgelistet und evtl. erklärt werden, gibt es bei einem *Escape Game* nicht. Ausnahmen bilden die Live-Varianten, bei denen es häufig Regeln gibt, die den Raum, die Gegenstände und die Spieler schützen sollen, aber nichts mit dem eigentlichen Spielkonzept zu tun haben. Die Regeln eines *Escape Games* sind vielmehr „*physikalischer*“ Natur, bzw. Nachbildungen davon. Manche sind dabei offensichtlich:

- durch eine Wand kann man nicht hindurchgehen
- im Dunkeln kann man nicht sehen
- ...

Andere Regeln liegen weniger auf der Hand und müssen durch Ausprobieren erfahren werden:

- kann die verrostete Tür geöffnet werden?
- lässt sich der Tisch verschieben?
- ...

Diese Regeln auszuprobieren und kennenzulernen ist immanenter Bestandteil des Spiels und des Reizes dieses Genres.

Der **Konflikt** bei einem *Escape Game* ist ein sehr offensichtlicher. Der Spieler ist in einem Raum gefangen, eine Situation, die man sich auch ohne viel Phantasie vorstellen kann und die sehr motivierend wirkt etwas daran zu ändern. Häufig wird der Konflikt durch eine zusätzliche zeitliche Komponente verschärft, bspw. durch knapper werdende Luft, einen steigenden Wasserpegel oder Ähnliches.

Das **Ergebnis** wird in erster Linie durch Erfolg, also der gelungenen Flucht, oder Misserfolg bestimmt. Darüberhinaus könnten sich Spieler anhand der benötigten Zeit untereinander messen.

³bspw. ‚*Mystery Of Time And Space*‘ (*MOTAS*) oder ‚*Noctropolis*‘

3.2 Technische Konzeption

Die Schwierigkeit bei einer *Audio-Only* Variante eines *Escape Games* liegt vor allem darin, dem Spieler zu ermöglichen sich anhand von auditiven Informationen in einem virtuellen Raum zu orientieren. Dazu muss ein „auraler Raum“⁴ mit verschiedenen auditiven Ereignissen (im Folgenden *Audio Events* genannt) geschaffen werden, in dem sich der Spieler bewegen kann. Damit die Orientierung in einem solchen Raum funktionieren kann, müssen die Richtungsinformationen der Audio Events immer in Relation zur Position, Blickrichtung und Bewegung des Spielers stehen. Setzt man ein lineares, zeitinvariantes System (LTI-System) voraus⁵, sind diese Richtungsinformationen in Form von frequenzabhängigen *Interaural Time Differences* (ITDs) und *Interaural Level Differences* (ILDs) gegeben[3, p.2]. Diese interauralen Unterschiede macht man sich auch bei der Laufzeitstereofonie (ITD) und bei der Intensitätsstereofonie (ILD) zu Nutzen, also den grundsätzlichen Stereofonieverfahren, bei welchen allerdings der frequenzabhängige Einfluss des Oberkörpers, des Kopfes und der Ohrmuschel unbeachtet bleibt. Um die Problematik des sogenannten „*cone of confusion*“ zu beheben (oder in der Praxis zumindest zu verringern) sind gerade diese Einflüsse relevant.

3.2.1 HRIR - HRTF

„*The effect of the acoustic filtering of torso, head and pinna can be described in terms of a linear time-invariant system by the so-called head-related transfer functions, HRTFs*“[1, S.94]

Nutzt man diese akustische Filterung, kann man einem trockenen Monosignal jede mögliche Richtungsinformation „hinzufügen“⁶. Dazu muss das Signal mit der *Head Related Impulse Response* (HRIR) gefaltet werden, da das Ausgangssignal eines LTI-Systems bei beliebiger Anregung durch die Faltung des Eingangssignals mit der Stoßantwort des Systems entsteht[5, S.76]. Die *Head Related Transfer Function* (HRTF) entspricht der Laplace-Transformierten der HRIR⁷. Im Frequenzbereich vereinfacht sich die Faltung zu einer Multiplikation.

Da die Geometrie des Oberkörpers, des Kopfes und der Ohrmuschel sehr individuell ausfallen, sind auch die HRTFs individuell. Eigentlich müsste also für jeden Menschen eine eigene Messung der HRTF vorgenommen werden, wobei eine solche Messung arbeits- und zeitintensiv ist. Aus diesem Grund werden häufig HRTFs verwendet, die mithilfe eines genormten Kunstkopfes gemessen wurden, wodurch die Lokalisierungsgenauigkeit stark sinkt⁸. Allerdings gibt es, beispielsweise vom IRCAM, Datenbanken mit bestehenden

⁴gemeint ist ein Raum, den man nur über den Hörsinn erfahren kann

⁵Eine zutreffende Annahme, wenn sich weder Schallquelle, noch Hörer oder Reflexionsflächen schneller als Schallgeschwindigkeit bewegen

⁶dabei ist es, wie in der Realität, frequenzabhängig wie gut die Ortung funktioniert

⁷man beachte, dass die Begriffe HRTF und HRIR häufig synonym verwendet werden

⁸abhängig davon wie ähnlich die eigene Kopf- und Oberkörpergeometrie der des benutzten Kunstkopfes ist

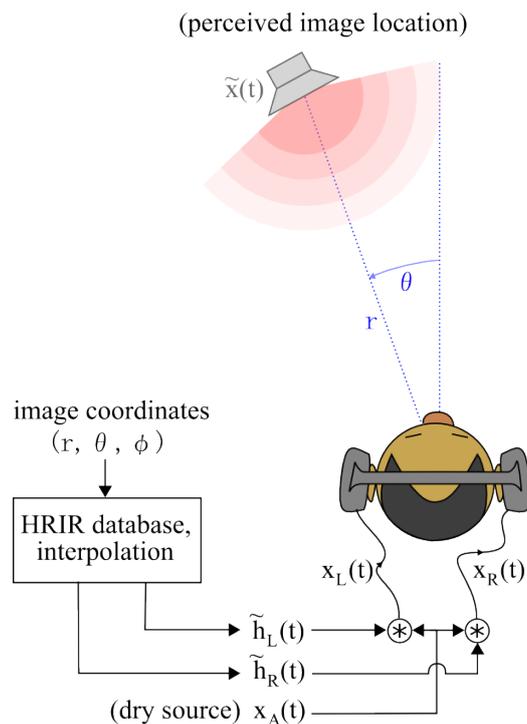


Abbildung 2: Binaural Spatialization by HRIR Convolution

HRIRs, aus denen man sich eine möglichst passende „aussuchen“ kann⁹. Der Nachteil von personalisierten HRTFs ist neben der aufwendigen Messung auch der zusätzliche Aufwand für den Anwender und den Programmierer eine solche in das System einzubinden. Aus diesem Grund wurde für diese Arbeit auf eine mithilfe eines Kunstkopfes gemessene HRTF zurückgegriffen. Die zugrunde liegenden Messungen wurden 1994 am MIT von Bill Gardner und Keith Martin durchgeführt.

3.2.2 KEMAR

KEMAR steht für **K**nowles **E**lectronics **M**anikin for **A**coustic **R**esearch. „*An extensive set of head-related transfer function (HRTF) measurements of a KEMAR dummy head microphone has recently been completed. The measurements consist of the left and right ear impulse responses from a Realistic Optimus Pro 7 loudspeaker mounted 1.4 meters from the KEMAR. Maximum length (ML) pseudo-random binary sequences were used to obtain the impulse responses at a sampling rate of 44.1 kHz. A total of 710 different positions were sampled at elevations from -40 degrees to +90 degrees. Also measured were the impulse response of the speaker in free field and several headphones placed on the KEMAR. This data is being made available to the research community on the Internet via anonymous FTP and the World Wide Web.*“ [2]¹⁰

⁹<http://recherche.ircam.fr/equipements/salles/listen/index.html>

¹⁰genauere Informationen zur Messung finden sich unter <http://sound.media.mit.edu/resources/KEMAR.html> bzw. [2]



Abbildung 3: HRTF Measurement with KEMAR Dummy-Head at TU Berlin

Aufgrund der Wahl einer nicht-personalisierten HRTF ist natürlich mit Einschränkungen bei der Lokalisationsschärfe zu rechnen. Andererseits ist belegt, dass der Lerneffekt bei der Nutzung von nicht-personalisierten HRTFs hoch ist[13, S.61], da sich Körper, Kopf und Ohrmuscheln im Laufe eines Lebens durchaus verändern, die eigene HRTF also keine unveränderliche Größe ist. Es ist also mit einem Trainings- bzw. Lerneffekt zu rechnen.

3.2.3 Pure Data

Bei Pure Data (Pd) handelt es sich um ein „*real-time graphical programming environment for audio processing*“[4, S.1].

Als datenstrom-orientierte, visuelle Patcher-Programmiersprache könnte man Pd als Open-Source Variante von Max/MSP bezeichnen. Pd funktioniert plattform-unabhängig, auch auf sogenannten *embedded devices* und Smartphones (via libpd, DroidParty, PdParty)[7]. Die Unabhängigkeit vom Betriebssystem und die Tatsache, dass es sich um *Open Source* Software handelt, haben den Autor dieser Arbeit dazu bewogen, *Pure Data* als Programmiersprache für *EarSCAPE* zu benutzen. Wer sich weitergehend in *Pure Data* einarbeiten möchte, dem sei die Website von Johannes Kreidler <http://pd-tutorial.com/german/index.html> ans Herz gelegt.

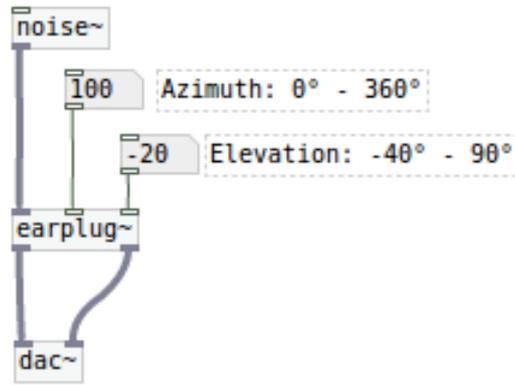


Abbildung 4: Basic Pd-Patch using earplug~

3.2.4 [earplug~]

[earplug~] ist ein Pd-External, das von Pei Xiang, David Camargo und Miller Puckette entwickelt wurde um Experimente zur auditiven Darstellung räumlicher Bewegungen durchzuführen[12]. Anders als bei Systemen, die, angelehnt an das Ambisonics Konzept, verschiedene statische HRTFs benutzen um eine dreidimensionale Spatialisierung zu ermöglichen[6], verwendet [earplug~] Lineare Interpolation um zwischen 368 Positionen auf einer Kugeloberfläche zu interpolieren (eigentlich nur ein Teilstück, der untere Bereich der Kugeloberfläche wird ausgelassen: 360° Azimuth, aber nur -40° bis 90° Elevation). Dazu wird auf die am MIT gewonnenen Daten mit KEMAR-Kunstkopf zurückgegriffen. Um Rechenleistung zu sparen werden nur 128 Samples lange Impulsantworten (ursprünglich 512) verwendet, wodurch eine ausreichende Lokalisierungsgenauigkeit erreicht wird[9].

Das [earplug~] -Objekt interpoliert also aufgrund der Eingaben für Azimuth und Elevation die entsprechenden HRTFs bzw. HRIRs für das rechte und linke Ohr und vollzieht die Faltung mit einem Mono-Audiosignal, siehe Abbildung 4.

4 Umsetzung

In diesem Kapitel wird ein kurzer Überblick über den Aufbau von *EarSCAPE* gegeben, um so aufzuzeigen wie der *aurale Raum* im Spiel realisiert wird. Dazu wird der Aufbau von „Außen“ nach „Innen“ dargestellt und auf die wichtigsten Funktionen und Subpatches eingegangen.

4.1 Beschreibung

Abbildung 5 zeigt den Startbildschirm von *EarSCAPE*. Beim Start sind die Samples und Einstellungen für den Testlevel geladen, das Spiel wird durch Drücken der Leertaste

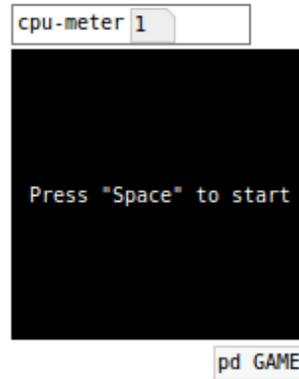


Abbildung 5: grafische Oberfläche von *EarSCAPE*

gestartet. Im Subpatch `[pd GAME]` findet man die eigentliche, organisatorische Oberfläche (siehe Abbildung 6), die vor dem Spieler verborgen bleiben soll. Im oberen Bereich, der mit *LEVEL DESIGN* überschrieben ist, finden sich alle Subpatches, die zum Erstellen eigener Levels benötigt werden. In den `[pd AudioEvent]`-Subpatches kann einem *Audio Event* ein Sample und eine Position zugewiesen werden. Im `[pd event-management]` wird für alle *Audio Events* festgelegt, wann die Samples abgespielt werden sollen (bspw. im Loop oder als Reaktion auf Berührung etc.). Wie diese Zuordnung realisiert wird, wird im Kapitel *Bang Management 4.5.3* ausführlicher dargestellt. Diese Samples werden je nach Relation zum Spieler (Entfernung, Winkel) durch die entsprechenden HRTFs gefiltert. Im `[pd person]`-Subpatch können bis zu 5 „personenbezogene“ Samples geladen werden, die immer in fester Relation zum Spieler abgespielt werden (Azimuth: 0°; Elevation: -40°), beispielsweise für Atemgeräusche, Herzschlag, Schritte und vergleichbares. Ähnlich wie bei den *Audio Events* wird im `[pd person-management]` festgelegt, wann welche Samples abgespielt werden sollen. In `[pd room]` können Einstellungen am Hall vorgenommen werden (als Reverb benutzt wird `[rev3~]` von Miller Puckette). In `[pd settings]` werden Angaben zur Raumgröße, der Dauer eines Schrittes, der Dämpfung in dB pro Schritt etc. gemacht. Im Kapitel *Erstellen eigener Levels 4.6* wird ausführlicher auf die unter *LEVEL DESIGN* zu findenden Subpatches eingegangen.

Darunter finden sich zum einen Informationen zur aktuellen Position des Spielers und Relationen zu den verteilten *Audio Events*, die in erster Linie zum Debuggen benötigt wurden, sowie Subpatches, die nicht für das Erstellen eigener Levels bearbeitet werden müssen.

4.2 Steuerung

Die Spielfigur wird in *EarSCAPE* über die Tastaturtasten **W**, **A**, **S** und **D** gesteuert. Die Taste **W** bewirkt einen Schritt nach vorn und **S** einen Schritt nach hinten. **A** und **D** drehen die Blickrichtung der Spielfigur nach links bzw. rechts ohne dass sich der Spieler von der aktuellen Position fortbewegt, ändern so aber natürlich die Richtung in die sich



Abbildung 6: [pd GAME]

die Spielfigur mit dem nächsten Schritt bewegen wird. Die Organisation der Steuerung verbirgt sich im Subpatch [pd controls].

4.3 Einstellungen

Im Subpatch [pd settings] werden allgemeine Einstellungen für *EarSCAPE* vorgenommen. Von links nach rechts (siehe Abb. 7) handelt es sich um Einstellungen bezüglich

- Raumgröße
- Dauer eines Schrittes [ms]
- Dämpfung
- zur Verfügung stehender Zeit [s]
- Startposition der Spielfigur
- Position der Tür

Als willkürliches Längenmaß ist dabei ein „Schritt“ festgelegt. Darauf bezieht sich auch die Dämpfung, die also angibt um wieviel dB (RMS) pro Schritt die Signalleistung abfällt.

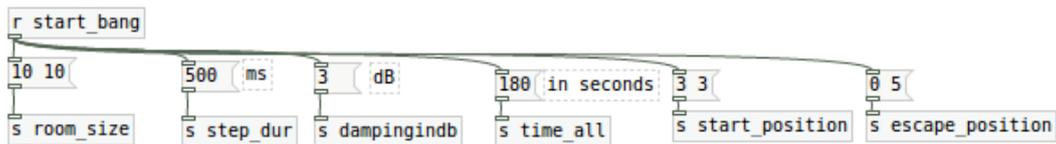


Abbildung 7: [pd settings]

4.4 Raumklang

Um den *auralen Raum* etwas realistischer zu gestalten und dem Spieler ein Feedback über das Verhältnis seiner Entfernung zu einem der *Audio Events* und der Größe des Raumes zu ermöglichen, wird in *EarSCAPE* das [rev3~]-Objekt von Miller Puckette genutzt. Dazu wird von jedem *Audio Event* und den personenbezogenen Geräuschen ein ungefilterter Anteil (also ohne Richtungsinformation) in den Hall-Effekt geschickt (siehe Abb. 8). Dieser Hall-Anteil ist unabhängig von der Entfernung zwischen Spielfigur und *Audio Event*, ändert sich also - im Gegensatz zum Direkt-Anteil - nicht durch die Position an der sich die Spielfigur befindet. Der durch das HF-Damping frequenzabhängige Hallradius ist durch die Leveleinstellung des Effekts und die unter [pd settings] gemachte Einstellung bezüglich Dämpfung vom Level-Designer zu beeinflussen.

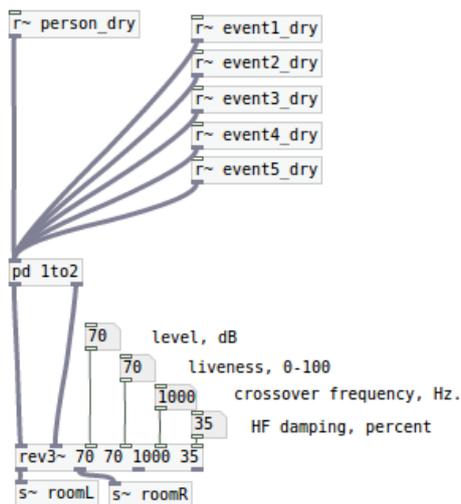


Abbildung 8: [pd room]

Der Subpatch [pd 1to2] hat nur die Funktion den Patch übersichtlich zu halten und erzeugt aus einem Monoeingang zwei gleiche Monoausgänge, da es sich bei [rev3~] um einen Stereoeffekt handelt.

Diese einfache Implementierung wird natürlich den akustischen Eigenschaften eines Raumes nicht gerecht, hilft aber auf sehr einfache Art und Weise die Größe des Raumes in Relation zur Entfernung der Schallereignisse zu setzen.

4.5 Audio-Events

Die *Audio Event*-Subpatches sind die zentralen Bausteine von *EarSCAPE*, in welchen der Anwender Samples laden, einen entsprechenden Gain einstellen und eine Position im Raum zuweisen kann (siehe Abb. 9).

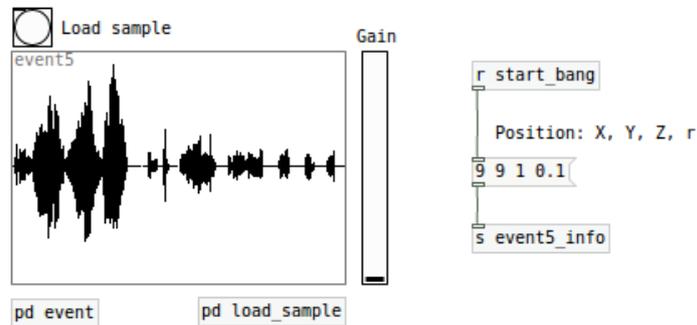


Abbildung 9: [pd AudioEvent]

Die eigentliche Signalverarbeitung findet im Subpatch [pd event] statt (siehe Abb. 10). Hier wird das Signal, das mithilfe von [tabplay~] aus dem Sample erzeugt wird, durch [earplug~] mit den entsprechenden HRTFs gefaltet. Die notwendigen Richtungsinformationen werden in [azi_ele_damp] aus der Relation zwischen der Position des Spielers und der des *Audio Events* berechnet. Änderungen werden dabei nicht als Sprung sondern als kontinuierlicher Datenfluss ausgegeben, angepasst an die Dauer eines Schrittes. Die frequenzunabhängige Dämpfung wird dabei einfach als Funktion der Entfernung zwischen Spieler und Event berechnet.

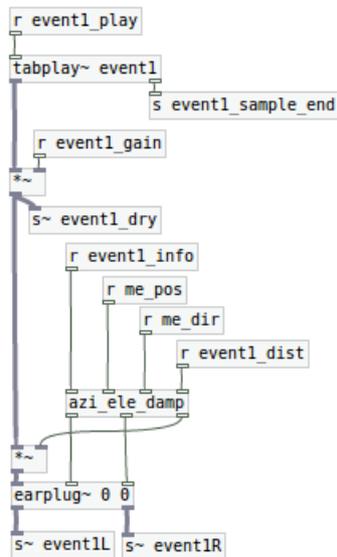


Abbildung 10: [pd event]

4.5.1 Platzierung von Audio Events

Wie in Abbildung 9 auf der rechten Seite zu sehen ist, wird die Position des *Audio Events* durch eine vierteilige Message übergeben. Die ersten drei Bestandteile stehen für die x-, y- und z-Komponente eines rechtsdrehenden Koordinatensystems. Die vierte Komponente gibt den Radius des *Audio Events* an. Der Einfachheit halber werden alle *Audio Events* als kreisförmig angenommen. Sobald die Spielfigur an den Rand dieses Kreises stößt, wird eine *Bang Message* ausgelöst. Wie auf diese reagiert wird, wird im *Bang Management* (siehe Kapitel 4.5.3) festgelegt.

4.5.2 Zuordnung von Samples zu Audio-Events

Durch Klicken des Buttons neben „*Load sample*“ (siehe Abb. 9) öffnet sich ein Browser mit dem man auf einfache Weise zum gewünschten Sample navigieren kann. Bei einem geeigneten Sample sollte es sich um ein Monosignal mit einer Abtastrate von 44,1kHz handeln, ansonsten entstehen unerwünschte Artefakte. Das ausgesuchte Sample wird in das jeweilige Array geladen und kann so mithilfe von `[tabplay~]` wiedergegeben werden.

4.5.3 Bang Management

Das *Bang Management* ist die Schaltzentrale von *EarSCAPE*. Hier wird festgelegt, wann welches Sample gespielt wird. Es gibt ein eigenes *Bang Management* für die *Audio Events* (`[pd event-management]`) und eines für die personenbezogenen Samples (`[pd person-management]`). Als „Auslöser“ für das Abspielen eines Samples stehen verschiedene *Bang Messages* zur Verfügung (siehe Abb. 11)

- Start des Spiels (`start_bang`)
- Schritt (`step_bang`)
- Wandberührung (`room_bang`)
- Ende eines Samples (`event#_sample_end`)
- Berührung eines *Audio Events* (`event#_hit_bang`)

Soll ein Sample in einer Schleife abgespielt werden, muss einfach die entsprechende *Bang Message* für das Ende des Samples mit dem Auslöser für das Abspielen des Samples (`event#_play`) verbunden werden. In den Subpatches `[pd bang_actions_#]` wird im einfachsten Fall die ankommende *Bang Message* direkt durchgeleitet. Je nach Level- und Sounddesign ist es aber möglicherweise gewünscht abwechselnd verschiedene Bereiche des Samples abzuspielen. Dies ist möglich indem man `[tabplay~]` zwei-Elemente Listen schickt, wobei das erste Element der Startpunkt innerhalb des Samplearrays ist und das zweite die Anzahl an Elementen angibt, die aus dem Array ausgelesen werden sollen. Je nach Programmierung ist es so möglich, das *Level-* und *Sounddesign* interessanter zu gestalten.

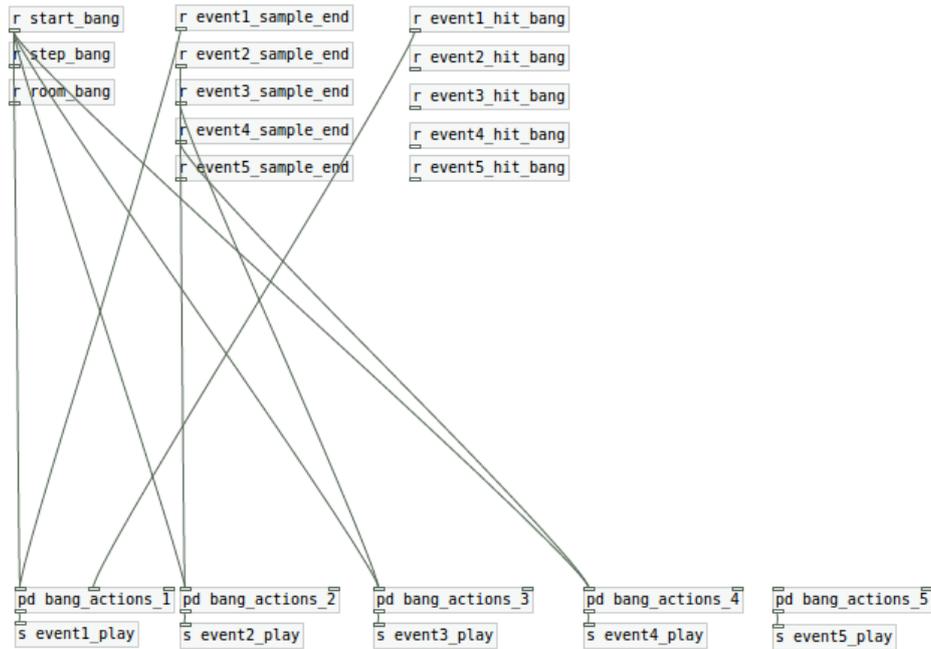


Abbildung 11: [pd event-management]

4.6 Erstellen eigener Levels

Wie man ein eigenes Level erstellt und was dabei zu beachten ist, wird in diesem Kapitel anhand des Probe-Levels beispielhaft dargestellt. Beim Erstellen eigener Levels ist *EarSCAPE* in erster Linie als Werkzeug zu sehen, mit dessen Hilfe ein *auraler Raum* geschaffen werden kann, in dem sich die Spielfigur bewegen kann. Dass aus dieser Situation darüberhinaus ein *Escape Game* entsteht, ist Aufgabe des Programmierers. Für die Erstellung eigener Levels müssen folgende Bereiche beachtet werden:

- Allgemeine Einstellungen (Zeitfaktor, Raumgröße etc.)
- Auswahl der Samples
- Platzierung der *Audio Events*
- Bang Management für die *Audio Events* und die personenbezogenen Samples

Die Auswahl der Samples als wesentlicher Teil des Sounddesigns bestimmt maßgeblich inwiefern das designte Level als *Escape Game* funktioniert. Das Sounddesign muss dem Spieler unmissverständlich das Ziel des Spiels aufzeigen: die Flucht aus dem Raum. Zusätzlich sollte durch das Sounddesign die zeitliche Komponente deutlich werden, wobei diese Aufgabe eher mit einem geschickten *Bang Management* erfüllt werden kann. Am Beispiel des Probe-Levels werden diese Forderungen durch ein Bedrohungsszenario erfüllt: die Spielfigur erwacht in einem Raum, in dem deutlich ein großer Brandherd zu

hören ist. Das Herzklopfen der Spielfigur wird mit zunehmender Spieldauer immer schneller, irgendwann wird es unregelmäßig und schlägt immer sporadischer bis es gänzlich aufhört. Auf diese Weise sollte dem Spieler spätestens nach dem ersten Spieldurchlauf klar sein, dass die Zeit drängt und dass etwas an der bestehenden Situation verändert werden muss.

Das Herzklopfen zählt zu den personenbezogenen Samples (wie auch Schritte, Atmen etc.), die in Relation zur Spielfigur immer an der gleichen Stelle bleiben. Der veränderte Herzrhythmus ist dabei also eine Frage des entsprechenden *Bang Managements*, das unter [pd person-management] (Abb. 6) vorgenommen werden kann.

Der Brandherd ist ein *Audio Event*, besitzt also eine feste Position im Raum und verändert seine Relation zur Spielfigur durch die Bewegung ebendieser. Im Probe-Level handelt es sich einfach um ein Sample, das in Endlosschleife läuft, also eine sehr einfache Form des *Bang Management*.

Die wesentlichen Anforderungen, die ein *Escape Game* an den Programmierer stellt, sind damit erfüllt. Die **Regeln** und das **Ergebnis** sind durch die Programmierung von *EarSCAPE* gegeben, der **künstliche Konflikt** entsteht durch das entsprechende Sounddesign.

Zusätzliche *Audio Events* sollten platziert werden, um den *auralen Raum* interessanter zu gestalten, außerdem sollte sich der *Level Designer* Gedanken darüber machen, wie er den Spieler zum Ausgang führt. Im Probe-Level liegt der „Schlüssel“ darin, dass man einer Biene durch den Raum folgt, die so den Weg zum Ausgang markiert.

Eine sinnvolle Platzierung der *Audio Events* ist dem Anwender überlassen. Logischerweise sollten sie innerhalb der angegebenen Raumdimensionen liegen und den Weg zum Ausgang nicht vollständig versperren. Auch Start- und Endpunkt sollten nicht im Sperrbereich von *Audio Events* liegen, eine „Sicherung“ gegen solche Unmöglichkeiten ist nicht implementiert.

5 Rück- und Ausblick

Das größte Fragezeichen für den Autor dieser Arbeit war es, inwiefern eine zufriedenstellende Orientierung in einem *auralen Raum* mittels standardisierter HRTFs ermöglicht werden kann. Im Zuge dieser Arbeit ist natürlich keine wissenschaftlich fundierte Antwort auf diese Frage zu geben, weil der Schwerpunkt ein völlig anderer war. Trotzdem sollen an dieser Stelle einige Beobachtungen und Feststellungen genannt werden, die während des Programmierens und Testens gemacht wurden.

Die „klassische“ Problematik des *Cone of Confusion* ist auch mit viel „Übung“ mit den standardisierten HRTFs sehr ausgeprägt. Die Mehrdeutigkeit der Informationen durch interaurale Zeit- und Pegeldifferenzen kann also durch die (standardisierte) frequenz- und richtungsabhängige spektrale Gewichtung nicht zufriedenstellend gelöst werden. Durch die Möglichkeit die Spielfigur auf der Stelle zu drehen, ist dieses Problem für die Orientierung allerdings von untergeordneter Bedeutung, da die Vorne-Hinten-Unterscheidung durch die laterale Bewegungsrichtung der Schallquelle in Relation zur Drehrichtung des Kopfes gegeben ist.

Diese Feststellung wirft natürlich die Frage auf, ob für eine ausreichende Orientierung im *auralen Raum* ein Spatialisierungskonzept ausreichen würde, das rein auf ILDs und ITDs basiert. Da allerdings auch ein möglichst realistisches Klangbild für das Spielkonzept von *EarSCAPE* von Bedeutung ist, ist die zusätzliche spektrale Gewichtung, die die Nutzung von HRTFs mit sich bringt, ein definitiver Gewinn.

Für den Autor dieser Arbeit funktioniert die Orientierung auf diese Weise sehr gut, erstmalige Benutzer benötigen allerdings eine gewisse Eingewöhnungszeit. Allerdings liegt die Vermutung nahe, dass diese Eingewöhnungszeit eher der ungewohnten Spielsituation, insbesondere der ungewohnten Verbindung von auditivem Feedback und Tastensteuerung, geschuldet ist.

Ein großer Schwachpunkt, dessen Verbesserung sowohl dem realistischen Klangbild als auch einer verbesserten Orientierung zugute käme, ist der Raumklang. Dieser besteht letztlich nur aus einer diffusen Nachhallfahne, die einen gewissen Eindruck über die Größe eines Raumes vermitteln kann, allerdings darüberhinaus keinen Beitrag zur Positionsbestimmung der Spielfigur beisteuert. Hierzu wären steuerbare Erste Reflexionen nötig, die dem Spieler einen Eindruck über die Entfernungen zu Wänden vermitteln könnten oder beispielsweise eine spektrale Filterung je nach Entfernung zur Wand (an der Wand und in Raumecken erhöhte Leistung im unteren Frequenzspektrum) oder ähnliche Maßnahmen. In diesem Bereich gäbe es einige Möglichkeiten zur Verbesserung.

Verbesserungswürdig ist auch die Repräsentation der Entfernung zwischen Spielfigur und *Audio Event*. Hierbei wirkt nur eine frequenzunabhängige Dämpfung, eine spektrale Veränderung durch die Luftabsorption wurde nicht implementiert. Ein weiterer Punkt, der im Nachhinein als Implementierungsfehler betrachtet werden kann, ist die Tatsache, dass für das *Bang Management* nur eine einzelne *Bang Message* zur Verfügung steht, die das Berühren einer der Raumwände anzeigt. Es ist also nicht ohne Weiteres möglich,

einem Sample, das den Zusammenstoß mit einer Wand repräsentiert, eine Richtungsinformation als zusätzliches auditives Feedback aufzuprägen. Ein frontaler Zusammenstoß klingt also gleich wie ein seitliches „Vorbeischrammen“.

Insgesamt gibt es also einige Verbesserungsmöglichkeiten, auf die im Rahmen dieser Arbeit und in Anbetracht der benötigten Rechenleistung allerdings verzichtet wurde.

Trotz der genannten Verbesserungsmöglichkeiten ist es mit *EarSCAPE* mit wenig Aufwand möglich, *aurale Räume* zu schaffen, in denen sich ein Spieler zuverlässig orientieren kann.

Literatur

- [1] BAUMGARTNER, R., P. MAJDAK und B. LABACK: *Assessment of Sagittal-Plane Sound-Localization Performance in Spatial-Audio Applications*. In: BLAUERT, JENS (Herausgeber): *The Technology of Binaural Listening*, Seiten 93–120. Springer, 2013.
- [2] GARDNER, BILL und KEITH MARTIN: *HRTF Measurements of a KEMAR Dummy-Head Microphone*. Technischer Bericht, MIT Media Lab Perceptual Computing, 1994.
- [3] KOHLRAUSCH, A., J. BRAASCH, D. KOLOSSA und J. BLAUERT: *An Introduction to Binaural Processing*. In: BLAUERT, JENS (Herausgeber): *The Technology of Binaural Listening*, Seiten 1–32. Springer, 2013.
- [4] KREIDLER, JOHANNES: *Loadbang: Programming electronic music in Pure Data*. Wolke, 2 Auflage, 2013.
- [5] MEYER, MARTIN: *Signalverarbeitung: Analoge und digitale Signale, Systeme und Filter*. Springer Vieweg, 7 Auflage, 2014.
- [6] NOISTERNIG, MARKUS, ALOIS SONTACCHI, THOMAS MUSIL und ROBERT HÖLDRIK: *A 3D Ambisonic based Binaural Sound Reproduction System*. In: *AES 24th International Conference on Multichannel Audio*, 2003.
- [7] PUREDATA.INFO: *More about Pure data*. [Zugriff 14.07.2017: <https://puredata.info/>].
- [8] SALEN, K. und E. ZIMMERMAN: *Rules of Play: Game Design Fundamentals*. The MIT Press, 2003.
- [9] SONTACCHI, ALOIS, MARKUS NOISTERNIG, PIOTR MAJDAK und ROBERT HOLDRIK: *Subjective Validation of Perception Properties in Binaural Sound Reproduction Systems*. In: *Audio Engineering Society Conference: 21st International Conference: Architectural Acoustics and Sound Reinforcement*, Jun 2002.
- [10] WIKIPEDIA: *Audio game*. [Zugriff 14.07.2017: https://en.wikipedia.org/wiki/Audio_game].
- [11] WIKIPEDIA: *Escape the room*. [Zugriff 14.07.2017: https://en.wikipedia.org/wiki/Escape_the_room].
- [12] XIANG, PEI, DAVID CAMARGO und MILLER PUCKETTE: *Experiments on Spatial Gestures in Binaural Sound Display*. In: *Proceedings of ICAD 05*, 2005.
- [13] YAMAGISHI, D. und K. OZAWA: *Effects of Timbre on Learning to Remediate Sound Localization in the Horizontal Plane*. In: SUZUKI, Y., D. BRUNGART, Y. IWAYA, K. IIDA, D. CABRERA und H. KATO (Herausgeber): *Principles and Applications of Spatial Hearing*, Seiten 61–70. World Scientific, 2011.



Cladders, Jakob

Familienname, Vorname

0931507

Matrikelnummer

Erklärung

Hiermit bestätige ich, dass mir der *Leitfaden für schriftliche Arbeiten an der KUG* bekannt ist und ich die darin enthaltenen Bestimmungen eingehalten habe. Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst habe, andere als die angegebenen Quellen nicht verwendet habe und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, den 07.10.2020

.....
Unterschrift der Verfasserin/des Verfassers