

Masterarbeit

soundtubes

Aspekte des Designs und der Implementierung eines Musikprogramms für die spezielle Zielgruppe Kinder

Ausgeführt am Institut für Gestaltungs- und Wirkungsforschung
der Technischen Universität Wien

unter Anleitung von Dr. Martin Pichlmair und Dr. Peter Purgathofer

durch

Stefan Schauer

Neulerchenfelderstraße 38/2, 1160 Wien

Februar 2006

„...zu componiren so viel man will ohne musikalisch zu seyn noch etwas von der Composition zu verstehen.“

Aus der Beschreibung eines Würfelwalzerspiels von Wolfgang Amadeus Mozart (KV Anhang 294d) [Scho05]

„Music is deadly important“

Ian McKaye, Fugazi [Kris95]

“Everybody knows

These are hard times

I feel the pitch

But I am planning something new”

DMX Krew “Hard Times” [Dmxk99]

Abstract

In dieser Diplomarbeit wird das Programm “soundtubes” vorgestellt. Dieses Programm ist ein Musikprogramm, das für Kinder konzipiert wurde. Die Grundidee war es, einen loop-basierten Sequenzer zu programmieren, der es dem Benutzer ermöglicht, mit Samples musikalisch zu arbeiten. Trotz dieser sehr konventionellen Zielsetzung wurde versucht, die in Musiksoftware-Programmen häufig verwendeten Interface-Metaphern zu vermeiden, gängige Musikvorstellungen zu brechen und Kindern abseits vom Instrumentalunterricht einen anderen Zugang zu Musik zu ermöglichen. Das Programm wird unter den drei Aspekten Technik, Interface und Sound ausführlich beschrieben.

Inhaltsverzeichnis

Abstract	3
Inhaltsverzeichnis	4
Einleitung	6
Motivation.....	7
Vorbemerkung zur Schreibweise	9
Danksagung.....	9
Sound	10
Onomatopoesie.....	10
Sampling.....	11
Geschichte des Sampling.....	11
Sampling heute	14
Sampling und Coypright.....	15
Loops.....	17
Die Geschichte des Loops	17
Bedeutung des Loops heute.....	19
Soundmodulation.....	20
Tonhöhenänderung.....	20
Lautstärkenänderung	21
Zusammenfassung.....	22
Interface	24
Eingabegeräte.....	24
Direkte Manipulation.....	25
Metapher.....	27
Metaphern in Musiksoftware.....	29
Klaviatur	29
Piano-Roll.....	31
Sequencer.....	34
MIDI.....	35
Zusammenfassung.....	38
Technologie	39
Visuelle Programmiersprachen	39
Max	41

Pure Data.....	44
Vibrez.....	45
Zusammenfassung.....	47
Soundtubes.....	48
Zielsetzung.....	48
Das Programm „soundtubes“.....	49
Inspirationen.....	50
Nanoloop.....	50
Gbloink!.....	52
Drumsteps.....	54
SimTunes.....	56
Elektroplankton.....	57
mu-framework.....	59
Stunts.....	59
Pipemania.....	60
Sound.....	62
Verworfenen Ansätze.....	62
Interface.....	64
Verworfenen Ansätze.....	68
Mögliche Erweiterungen.....	73
Technologie.....	76
Umsetzung.....	76
Verworfenen Vorgangsweisen.....	79
Verworfenen Technologien.....	79
Mögliche Weiterentwicklung.....	80
Zusammenfassung.....	81
Zusammenfassung.....	82
Kritik am Programm.....	82
Persönliche Kritik.....	84
Quellen.....	85
Bildquellen.....	88

Einleitung

Heutzutage ist die Verwendung von Computer zur Musikproduktion etwas völlig Selbstverständliches. Sowohl als Instrument für den Live-Einsatz als auch als wichtiger Teil jedes Studios ist der Computer nicht mehr wegzudenken. Auf den ersten Blick haben aber beide Bereiche wenig miteinander zu tun. Computer sind Maschinen, die auf logischen Prozessen aufbauen und für ein rein rationales Weltbild stehen. Vorgänge, die vom Computer ausgeführt werden können, müssen in einfache Basisoperationen zerlegbar sein. Musik hingegen ist ein künstlerischer Prozess, der wenig mit Logik oder Rationalität zu tun haben muss. Musik stellt oftmals den subjektiven Ausdruck einer Person dar.

Trotz dieser scheinbar unüberbrückbaren Unterschiede und obwohl in anderen Bereichen der Kunst wie der Malerei der Einsatz von Computer völlig unüblich ist, werden Computer sehr häufig für musikalische Zwecke eingesetzt. Dabei kann man vier große Einsatzbereiche definieren:

- Kompositionshilfe

Da jede sprachlich ausformulierte Kompositionsregel programmiert werden kann, ist es möglich, Programme zu schreiben, die selbst Stücke komponieren oder Komponisten bei ihrer Arbeit unterstützen. Diese Art des Einsatzes von Computern stellt die historisch älteste Methode dar. Besonders bei streng mathematischen Kompositionsmethoden wie der Zwölftonmusik oder Musikstücken, die sich stark an dem Zufallsprinzip orientieren, lassen sich Computer gut einsetzen. Namhafte Komponisten wie John Cage oder Iannis Xenakis nutzten diese Möglichkeit.

- Klangsynthese

Computer können eingesetzt werden, um Klänge¹ zu errechnen, die über einen Digital-Analog-Wandler² hörbar gemacht werden. Diese können auf anderen Weg nicht produzierbar werden. Ähnlich wie in der Computergrafik, bei der viele Anstrengungen unternommen werden, fotorealistische Bilder zu

¹ Ein Klang besteht aus mehreren Tönen verschiedener Frequenzen. Die Mischung von Tönen mit einem ganzzahligen Verhältnis der Frequenzen zueinander macht die Klangfarbe eines Instruments aus. Ein Ton ist eine einzelne periodische Schallschwingung.

² Ein Digital-Analog-Wandler übersetzt digitale Audio-Daten in analoge Spannungsschwankungen, die über Lautsprecher in Schallwellen umgewandelt werden.

erzeugen, wird in diesem Bereich oft versucht, „echte“ Instrumente zu imitieren.

- Steuerung von Klangerzeugern

Ein weiterer Einsatzbereich ist die Steuerung von Klangerzeugern, die sowohl externe Synthesizer, die etwa über das MIDI-Protokoll³ angesteuert werden, als auch interne Klangerzeuger sein können. Zu diesem Bereich gehören auch die Aufnahme, Speicherung und Wiedergabe dieser Steuerungsinformationen.

- Klangspeicherung

Ein wichtiger Anwendungsbereich ist die Speicherung von Klängen, die über einen Analog-Digital-Wandler⁴ in für den Computer speicherbare Form gebracht werden. Da dieser Vorgang viel Rechenleistung und große Speicherkapazitäten⁵ benötigt, stellt dieser Bereich einen der jüngsten in diesem Gebiet dar.

[Rusc98]

Ein Grund für den Einsatz von Computern bei der Musikproduktion liegt vielfach in der Arbeitserleichterung. Neben dieser Funktion ist viel bedeutender, dass sich auf diese Weise viele neue Möglichkeiten des Musizierens eröffnen. Der Großteil der Musik-Software ist dabei dezidiert für spezialisierte und geübte Benutzer ausgelegt. Laien und besonders Kindern ist daher der Zugang zur Musikproduktion am Computer nur schwer möglich. Dies wäre aber wünschenswert, da Kindern so ein umfassenderes Verständnis von Musik abseits von Instrumentalunterricht vermittelt werden könnte. Das Programm „soundtubes“ sollte in dieser Lücke Kindern und musikalischen Laien eine Möglichkeit bieten, selbstständig am Computer musikalisch tätig zu werden.

Motivation

Die Gründe, warum ein Musikprogramm für Kinder als Diplomarbeitsthema gewählt wurde, sind vielfältig.

³ MIDI-Protokoll: Musical Instrument Digital Interface: digitales Kommunikationsprotokoll, wird im Kapitel MIDI genauer erklärt.

⁴ Ein Analog-Digital-Wandler wandelt Schallwellen in digitale Daten um. Dabei wird das Audio-Signal in regelmäßigen Abständen abgetastet und die so ermittelten Werte digital codiert und gespeichert.

⁵ Umkomprimiert benötigt ein wav-Audiofile von der Länge von fünf Minuten in CD-Qualität 50,5 Mega-Byte Speicherplatz.

Die Hauptmotivation war der Wunsch, für die Diplomarbeit ein Thema zu wählen, das einen musikalischen Bezug hat, da ich mich seit mehreren Jahren intensiv mit Musik auseinandersetze. Nach Blockflötenunterricht an der Volksschule und einer sechs Jahre dauernden Ausbildung an der Zugposaune bekam ich im Mai 1998 das Programm „Impuls Tracker“. Dabei handelte es sich um ein Trackemusikprogramm⁶ für den PC. Fasziniert von der Möglichkeit allein musikalisch tätig zu sein, verbringe ich seit diesem Zeitpunkt sehr viel Zeit mit der Musikproduktion am Computer.

Wie andere von einem Thema Begeisterte auch, habe ich oftmals erfolglos versucht, andere Personen für mein Hobby zu gewinnen. Da der Großteil der Musikprogramme sehr kompliziert und für einen schnellen und intuitiven Einstieg nicht geeignet ist, war dies sehr oft ein erfolgloses Unterfangen.

Im Jänner 2003 wurde ich Mitglied des im August 2002 gegründeten Gameboymusicclubs (www.gameboymusicclub.org), einer Vereinigung von Musikern, die auf Nintendo Gameboys musizieren. Eines der Programme, das die Musikproduktion am Gameboy ermöglicht, heißt „nanoloop“ und wurde vom Hamburger Oliver Wittchow entwickelt. Das Userinterface des Programms ist sehr reduziert und es stellte sich heraus, dass auch musikalische Laien und Kinder in kürzester Zeit mit diesem Programm Musik produzieren können.

So wurde es ein Ziel des Gameboymusicclubs, Kindern dieses Musikprogramm näher zu bringen und sie abseits der klassischen Musikerziehung und Instrumentenausbildung zum Musizieren zu befähigen. Da der Verein nur wenig finanzielle und ideelle Förderungen für die Schulungen von Kindern erhielt, wurden diese eingeschränkt.

Auf diese Weise wurde mein Interesse an der Konzeptionierung eines einfach bedienbaren Musikprogramms geweckt. Bei Recherchen im Internet wurde bald klar, dass diesem Gebiet kaum Beachtung geschenkt wird. Die wenigen Musikprogramme für Kinder, die zu finden waren, unterschieden sich meist nur in einer farbenfroheren Gestaltung und einer starken Reduzierung der Funktionalitäten von herkömmlicher Musiksoftware.

⁶ Ein Tracker ist eine spezielle Art von Sequenzer am Computer. Der Benutzer kann Samples über mehrere Spuren (Tracks) anordnen. Die Eingabe der musikalischen Parameter (Tonhöhe, Lautstärke, etc.) geschieht nicht graphisch sondern durch alphanumerische Zeichen und dabei teilweise durch hexadezimale Zahlen. Ein Musikstück besteht aus verschiedenen so genannten „Pattern“.

2005 versuchte Wolfgang Kopper, einer der Gründer des Gameboymusicclubs, ein Projekt beim Mozartjahr 2006 einzureichen, das zum Ziel hatte, Kinder zum Komponieren anzuregen. Dies sollte durch eigens programmierte Software auf verschiedenen Plattformen (PC, Sony Playstation, Nintendo Gameboy, etc.) erreicht werden. Für die Plattform PC wurde mein Programm, das sich damals noch in der Konzeptionsphase befand, eingereicht. Nach langem Hin und Her wurde das Projekt aus Kostengründen abgelehnt.

Vorbemerkung zur Schreibweise

In dieser Diplomarbeit wird bei der Erwähnung von Personen nur die männliche Form verwendet, obwohl beide Geschlechter gemeint sind. Wenn der Begriff Musiker im Text auftritt, sind sowohl Musiker als auch Musikerinnen gemeint.

Danksagung

Dank an Wolfgang Kopper, Thomas Grill, Hannes Köcher, Oliver Wittchow, Martin Pichlmair, Peter Purgathofer, jeden der an meiner Begeisterung für Musik leiden musste (besonders meine Wohnungskollegen Florian Resch und Rainer Mandl) und vor allem meinen Eltern, die mir das Studium ermöglicht haben.

Sound

In diesem Kapitel werden die musikalischen und klanglichen Grundlagen des Programms „soundtubes“ vorgestellt. Diese sind die Onomatopoesie, der Loop, das Sampling und die Soundmodulation.

Onomatopoesie

Der Begriff „Onomatopoesie“ ist die Zusammensetzung der beiden griechischen Wörter „onoma“ und „poein“ und heißt wortwörtlich „Namen schaffen“.

Onomatopoesie kann als Lautmalen ins Deutsche übersetzt werden. Ein onomatopoetisches Wort ist also ein Wort, dessen Lautbild die Imitation des Geräusches ist, auf das verwiesen wird. Beispiele dafür sind etwa die Begriffe „Muhen“ oder „Quaken“. Zur Onomatopoesie gehören auch Wörter, die durch Sprachlaute versuchen, bestimmte Gefühle wiederzugeben. Ein Beispiel aus der englischen Sprache ist etwa „She left us looooong time ago.“ Damit soll die lange Dauer verdeutlicht werden. Jede Sprache kennt solche Wörter⁷. Onomatopoetische Wörter versuchen nicht eine exakte Nachbildung des Geräusches zu sein, sondern sie stellen eine Annäherung dar, die im jeweiligen Sprachraum aussprechbar ist. Während in deutschsprachigen Raum Hähne „Kikeri“ von sich geben, wird im englischen Sprachraum ihre Lautäußerung als „cock-a-doodle-doo“ interpretiert. Lautmalerei ist meist ein Phänomen der gesprochenen Sprache und findet nur selten Eingang in die Schriftsprache. [Erns90] [Nuck96]

Ernst Havlik unterteilt die tierlaut-onomatopoetischen Wörter in zwei Gruppen: die umschreibenden und die eigentlichen. Die umschreibenden onomatopoetischen Wörter stellen solche dar, die weniger der Nachahmung eines Lautes dienen, sondern bei denen ein Gefühlsausdruck oder ein Tier charakterisiert werden soll. Die eigentlichen onomatopoetischen Wörter sind solche, die den Laut sprachlich so genau wie möglich abbilden möchten. [Erns90]

Bedeutend ist Onomatopoesie auch in Comics, wo sie zur Verdeutlichung dramatischer Ereignisse eingesetzt wird. [Erns90]

⁷ Während der Großteil der Sprachen nur solche Wörter kennt, baut die in Ecuador gesprochene Sprache „Quechua“ auf diesem Prinzip auf. [Nuck96]

Sampling

Das Sampling bezeichnet die musikalische Arbeit mit Samples. Ein Sample ist ein kurzes aufgenommenes Stück Musik oder Geräusch, das musikalisch in einem anderen Kontext verwendet wird. Die Samples können sowohl in digitaler Form (z.B. als Audiodatei) als auch analog (z.B. Audio-Tonband) vorliegen. Durch die Verwendung von Samples wird die lineare Zeitstruktur gebrochen. Samples aus der Vergangenheit werden in einem Musikstück der Gegenwart wieder verwendet. Es findet also eine De-Kontextualisierung statt.

Geschichte des Sampling

Musikalisch bedeutend wurde das Sampling 1948 durch den Franzosen Pierre Schaeffer, der die „music concrete“ begründete. Diese Musikrichtung versuchte, das Geräusch⁸ musikalisch zu emanzipieren und berief sich dabei auf den Futurismus, der in seinem Gründungsmanifest 1909 mit gängigen Begriffen von Schönheit und Kunst brach. Stücke der „music concrete“ zeichnen sich dadurch aus, dass sie als Arbeitsmaterial Alltagsgeräusche, sozusagen „konkrete Geräusche“, benutzte. Diese wurden aufgenommen, durch verschiedene Techniken verfremdet und zu Musikstücken arrangiert. Hier wurde das Sampling als sehr freie Technik verwendet. Neben der „music concrete“ gab es schon zuvor Versuche musikalisch mit Samples zu arbeiten. 1930 schuf Walter Ruttmann ein Hörstück mit dem Titel „Week-End“, in dem er die Geräusche einer Stadt am Wochenende arrangierte.

Karlheinz Stockhausen lernte zuerst bei Pierre Schaeffer die „music concrete“ kennen. In Köln beschäftigte er sich dann stark mit der Synthese von Klängen. Dabei erkannte er, dass es so produzierten Stücken an Atmosphäre fehlte und verband in seinem Stück „Gesang der Jünglinge“ die beiden gegensätzlichen Pole der „music concrete“ und der auf additive Klangsynthese konzentrierten „Kölner Schule“.

[Rusc98]

In Amerika entwickelte sich um 1952 die „Tape Music“. Dabei wurden Samples von Musikinstrumenten durch Rückkopplungen und Geschwindigkeitsänderungen musikalisch verwertet. Die maßgeblichen Akteure, Otto Luenig und Vladimir

⁸ Ein Rauschen bezeichnet eine nicht periodische Schallschwingung, der keine Frequenz zugeordnet werden kann. Ein Geräusch zeichnet sich auch dadurch aus, verändert sich aber im Zeitlauf stark. In der englischen Sprache wird „Noise“ oft synonym für Geräusch und Rauschen verwendet. In dieser Arbeit wird der Begriff „Noise“ nur als Synonym für Rauschen verwendet.

Ussachevsky, lernten sich 1952 bei einem Konzert von Ussachevsky kennen, das Luenig stark beeindruckte. Im Gegensatz zur „music concrete“ wurde der Bandschnitt sehr selten verwendet. Auch die Entscheidung, Samples von akustischen Instrumenten⁹ als Ausgangsmaterial zu verwenden, stellte einen weiteren großen Unterschied dar. Den Vertretern der „Tape Music“ ging es um die Ausweitung der Ausdrucksmöglichkeit von Instrumenten.

Hugh Le Caine, ein kanadischer Musikinstrumentenerfinder, baute, inspiriert von der „music concrete“, 1955 ein Mehrspurtonbandgerät, bei dem jedes Tonband mit unterschiedlicher Geschwindigkeit abgespielt werden konnte. Die Geschwindigkeit der Bänder wurde über eine Klaviatur gesteuert. Eines der ersten Stücke, die die Möglichkeiten dieses neuen Instruments aufzeigten, war das vom Erfinder komponierte Stück „Dripsody“, das ausschließlich aus dem Klangmaterial einer Aufnahme eines Wassertropfens gemacht wurde. Dieses Mehrspurbandtongerät kann als Vorläufer des modernen Samplers¹⁰ gesehen werden. [Rusc98]

Mit der Erfindung des Mellotrons 1963 wurde ein großer Schritt in die Richtung gemacht, Samples als Ersatz für akustische Instrumente zu verwenden. Ein Mellotron¹¹ ist ein Tasteninstrument, das als Klangerzeuger Tonbänder und entsprechende Abspielgeräte besitzt. Die Tonbänder werden auf Tastenanschlag abgespielt. Auf den Tonbändern wurden sowohl Instrumentenklänge in verschiedenen Tonhöhen als auch komplette musikalische Passagen gespeichert. Die Firma „Streetly Electronics“, die die Mellotrons baute, brachte später die ersten Sampler auf den Markt. [Zand98]

Digitales Sampling wurde erstmals 1985 durch teure Systeme wie dem „Synclavier“ oder dem „Fairlight Computer Musical Instrument“ möglich. Beim digitalen Sampling sind die Samples in digitaler Form als Datei gespeichert, während beim analogen Sampling diese als Audio-Aufnahmen vorliegen. Sampling wurde dabei nur zum einmaligen Abspielen von anderen Audioquellen benutzt. Mit der Verbesserung der Aufnahme- und Wiedergabekapazitäten konnten immer längere Audiosignale

⁹ Der Begriff akustische Musik oder akustische Instrumente wird hier für Musik oder Instrumente verwendet, die ohne Verwendung von elektronischen Schaltkreisen erzeugt wird. Es sind damit auch Musik und Instrumente eingeschlossen, die auf elektronischem Weg verstärkt oder verändert werden. Diese Definition ist an die des Bonner Phonetiker Werner Meyer-Eppler angelehnt. [RuAn98]

¹⁰ Gerät, zur Aufnahme und Wiedergabe von Audio-Aufnahmen.

¹¹ Der Name des Instruments ist eine Mischung der Worte „Melody“ und „Electronic“.

aufgenommen werden. Dadurch konnten Samples wiederholt und so für rhythmische Zwecke eingesetzt werden. [Scha02] [Rose94]

Große Bedeutung besitzt das Sampling auch in der Musikrichtung „Hiphop“, dessen Wurzeln in der musikalischen Arbeit mit Samples von Platten zu finden sind. Die Instrumentals¹² der ersten Hiphop-Stücke bestanden ausschließlich aus Plattensamples. Auch wurde im Hiphop der Sampler erstmals dazu benutzt, um Samples zu wiederholen und so für rhythmische Zwecke zu verwenden. Sampling im Hiphop soll nicht als arbeitserleichternden Weg in der Musikproduktion gesehen werden. Der Produzent Attrell Cordes¹³ meint dazu, dass es oft länger dauert, ein passendes Sample zu finden, als es dauern würde, ein Riff¹⁴ zu finden. Sampling sollte daher nicht als Diebstahl missverstanden werden. Darüber hinaus ist zu bedenken, dass viele Hiphop-Stücke nicht nur aus einem geloopten Sample bestehen. Das Lied „Night of the Living Baseheads“¹⁵ der Gruppe „Public Enemy“ besteht etwa aus 45 verschiedenen Samples. Manche Musiker unterstützen das Sampling ihrer Musikstücke aktiv wie etwa George Clinton, der 1993 das dezidiert für das Sampling zusammengestellte Album „Sample Some of Disc – Sample Some of DAT“ veröffentlichte. Andere, wie die Band „ESG“, die auch vielfach gesampelt wurden, sehen Sampling als Diebstahl. Der Titel der 1992 veröffentlichten EP „Sample Credits don’t pay our bills“ lässt an dieser Einstellung keinen Zweifel. [Debu06] Viele Produzenten sehen das Sampling auch als eine Art von Geschichtsarbeit, durch die alte Musikstücke dem Hörer wieder näher gebracht werden. Nach Tricia Rose ist der Ursprung vieler Techniken des „Hiphop“, unter anderem die des Sampling, in der afrikanischen Musiktradition zu suchen, die sich von europäischen Musikvorstellungen unterscheidet. Die neuen technologischen Möglichkeiten des Samplings wurden dabei in der Tradition der oralen afrikanischen Kultur adaptiert, in der unter anderem auch die Kollage und Breaks¹⁶ eine bedeutende Rolle spielen. [Rose94] Durch das Sampling konnte die Musikrichtung „Hiphop“ ohne Probleme andere Musikstile integrieren und drang so in den

¹² Aufnahme eines Musikstücks ohne Stimme oder Gesang.

¹³ Mitglied der Hiphop-Gruppe P. M. Dawn, die 1991 mit dem Lied „Set Adrift on Memory Bliss“, das auf zwei bekannten Samples aufbaut, international sehr erfolgreich war.

¹⁴ Riff: Akkordfolge, die charakteristisch für ein Lied ist.

¹⁵ Dieses Lied ist auf dem 1988 veröffentlichten Album „It takes a nation of millions to hold us back“ zu finden.

¹⁶ Break: hier Bruch in der rhythmischen Struktur

musikalischen Mainstream vor. Der Erfolg des Lieds „Walk this way“ der Gruppe „Run DMC“ in Zusammenarbeit mit der Rockband „Aerosmith“ zeigte die Massenkompabilität dieses Genre. [Oluf06] [Brun02]

Sampling heute



Abbildung 1: Der Hardware-Sampler AKAI MPC 4000

Die musikalische Arbeit mit Samples wurde ab dem Zeitpunkt, an dem die ersten Sampler am Markt erhältlich waren, zu einem bestimmenden Stilmittel elektronischer Musikproduktion. Samples wie das „Amen“¹⁷-Drumsample wurden millionenfach verwendet. Um 1986 kamen die ersten Tracker-Musikprogramme auf, die dem Benutzer erlaubten, am Computer musikalisch mit Samples zu arbeiten. Da die

¹⁷ Schlagzeug-Sample aus dem Stück „Amen Brothers“ der Soulband „The Winstons“. Dieses Sample stellt eine Basis der Musikrichtungen Jungle, Drum ’n’ Bass und Breakcore dar und wurde auch vielfach im Hip-hop verwendet. [Yout07]

Aufnahme, Umwandlung und Speicherung von Samples am Computer aber ein rechen- und speicherplatzintensiver Prozess sind, stellten Computer lange keine Konkurrenz für Hardware-Sampler dar. Mit der Verbesserung der Rechen- und Speicherkapazitäten wurde der Computer ein immer wichtigeres Werkzeug für die musikalische Arbeit mit Samples. Aber auch heute werden Hardware-Sampler noch sehr gerne verwendet. Die heute am meisten verwendeten Hardware-Sampler kommen aus der „Akai MPC“-Serie, wie die abgebildete „Akai MPC 4000“.

Die letzte musikalische Innovation im Bereich des Sampling stellte das Genre „Clicks’n’Cuts“ dar. Musiker, die dieser Richtung zugeordnet werden, zerkleinern Samples bis zur fast vollständigen Unkenntlichkeit und arbeiten ausschließlich mit diesen Micro-Samples. Bei dem Genre „Clicks’n’Cuts“ handelt es nicht um eine Musikrichtung, sondern um eine Vielzahl von Stilen, wobei der Umgang mit dem Sample-Material die einzige Gemeinsamkeit darstellt.

Sampling und Copyright

Da beim Sampling oftmals copyright-geschütztes Ausgangsmaterial verwendet wird, arbeiten Sampling-Künstler in einem rechtlichen Graubereich. Um das heutige Copyright-Recht zu verstehen, hilft ein Blick auf die geschichtlichen Ursprünge. Mit der Erfindung der beweglichen Drucklettern durch Johannes Gutenberg im 15. Jahrhundert, konnten Schriften erstmals in großer Anzahl kopiert werden. Das erste Copyright wurde 1491 dem Juristen Peter von Ravenna zugesprochen. Dieses wurde als Nachdruckprivileg verstanden, das einem Verlag die exklusiven Druckrechte eines Werkes zusicherten. Es ging also um die Absatzsicherung für einen Verlag und nicht um den Schutz des Urhebers. Im Zeitalter der Renaissance rückte das Bild des Schöpfers und Künstlers als einzelner unabhängiger Genius in den Mittelpunkt. Das Copyright wandelte sich dabei zu einem Instrument des Schutzes des autonomen Urhebers, der unbeeinflusst von äußeren Einflüssen sein Werk erschafft. Plagiate des genialen Werks sollten so verhindert werden. Dieses romantische Verständnis herrscht bis heute vor. Die modernen Copyrightgesetze fußen im Gesetz „Statute of Anne“, das 1710 vom englischen Parlament beschlossen wurde. Der Autor wurde als Inhaber des Copyrights festgelegt. Alle Werke des Verfassers mussten wenn sie vervielfältigt wurden, mit einem Copyright-Vermerk versehen sein. 1787 schlug James Madison bei einer amerikanischen „Constitutional Convention“ vor, das Copyright in die Verfassung aufzunehmen, was ohne Gegenstimmen beschlossen wurde. Dieser Passus wird auch „Progress Clause“

genannt, da die Wichtigkeit des Copyrights damit begründet wurde, dass auf diese Weise der Fortschritt der Wissenschaft gewährleistet sei. Auf der Website der RIAA (Recording Industry Association of America), der Interessensgemeinschaft der Unternehmen der Musikindustrie in den USA, wird dazu bemerkt, dass das Copyright vor dem Recht auf freie Meinungsäußerung und dem Recht der Versammlungsfreiheit Aufnahme in die Verfassung fand. Diese Bemerkung spielt darauf an, dass Copyright-Kritiker sich oftmals auf das Recht der freien Meinungsäußerung beziehen. [Riaa07] [Less04]

Mit dem Aufkommen von Technologien, die das Kopieren von anderen Medien wie Film oder Musik ermöglichten, wurden diese Copyright-Gesetze, die zuvor nur für Druckwerke galten, auf diese ausgeweitet.

Musik wurde 1831 unter den Schutz von US-amerikanischen Copyright-Gesetzen gestellt. Bei gerichtlichen Entscheidungen bezüglich musikalischen Copyrightverletzungen wird hauptsächlich auf die Melodie geachtet. Dem rhythmischen Anteil eines Musikstücks wird weniger Beachtung geschenkt. Dies ist aus der europäischen Musiktradition erklärbar, die stark auf melodische Strukturen fixiert ist. Auch wird die Performance eines Stücks nicht im gleichen Maß beachtet, obwohl diese bei Musik ein wichtigerer Bestandteil als bei Literatur ist. Weiters wird außer Acht gelassen, dass Musikrichtungen wie etwa das Genre „Blues“ von bestimmten harmonischen Grundstrukturen ausgehen. Die Ausweitung der Copyright-Gesetze auf Musik ist daher nicht trivial. [Oluf06] [Rose94]

Wie weiter oben ausgeführt, basiert die Musikrichtung „Hiphop“ auf der Verwendung von Samples. 1991 kam es zu einem Gerichtsurteil, das den Umgang mit Samples im Hiphop stark beeinflusste. Der Rapper¹⁸ Biz Markie wurde von einem Gericht in New York verurteilt, da er, ohne die Erlaubnis der Plattenfirma einzuholen, von einer Platte drei Worte und etwas Musik gesampelt hatte. Durch dieses Urteil änderten sich die Produktionsweise und die musikalische Umsetzung von Hiphopliedern in einem beträchtlichen Maße. Viele Musiker beschränkten die Anzahl der verwendeten Samples stark, um so die Kosten für die Produktion des Musikstückes gering zu halten. Teilweise werden nun Samples neu eingespielt und aufgenommen, um nur mit dem Songschreiber aber nicht mit den Plattenfirmen verhandeln zu müssen. [Ccnm06]

¹⁸ Rapper: Die vortragende Person eines „Raps“. Der Begriff „Rap“ ist aus „Rapid Poetry“ abgeleitet.

Loops

Der Loop ist die oftmalige Wiederholung eines Samples oder einer musikalischen Phrase. Das so genannte „Ostinato“ ist das Gegenstück in der klassischen Musik.

Die Geschichte des Loops

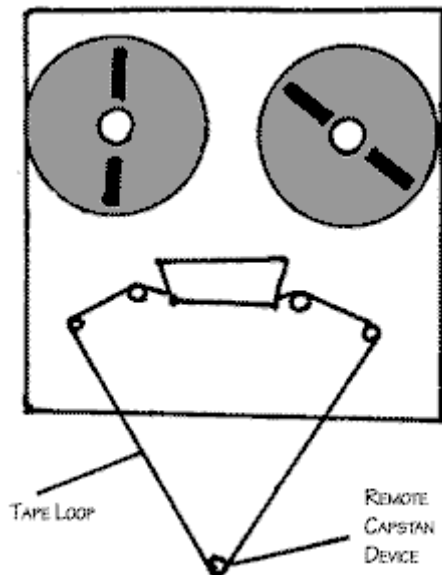


Abbildung 2: Tapeloop

Der Ursprung des Worts „Loop“ kommt von den Tape-Schleifen, mit denen Komponisten wie John Cage, Karlheinz Stockhausen und andere experimentierten. Der Tape-Loop selbst bezeichnet ein Stück bespieltes Audiomagnetband, dessen beiden Enden zusammengeklebt werden und so eine Schleife bilden. Wird diese Schleife durch eine Bandmaschine abgespielt, wird das aufgenommene Audiomaterial immer wieder wiederholt. Stockhausen war einer der Pioniere auf diesem Gebiet und komponierte Stücke für Tape-Loop und Orchester. Durch technische Weiterentwicklungen geriet der Tape-Loop immer stärker in Vergessenheit, die Grundidee des Loops blieb aber. Diese wurde auch von anderen Musikern, die nicht unbedingt der E-Musik zugeordnet werden, aufgegriffen und weiterentwickelt. Besonders die deutsche Band „Kraftwerk“ machte den Loop zum elementaren Bestandteil ihrer Musik. Das 1974 erschienene Album „Autobahn“ war wegweisend für die Entwicklung der elektronischen Musik¹⁹. Die Stücke dieser Platte sind großteils Loops, wobei Variationen in den Klängen zu finden sind. Die

¹⁹ Oftmals ist der Begriff elektrischer Musik in der Literatur zu finden. Der Begriff elektronische Musik ist aber richtig, da bei elektronischer Klangerzeugung die Beeinflussung von mehreren Ladungsträgern entscheidend ist. [RuAn98]

Liedstruktur Strophe-Refrain wurde durch den Loop abgelöst. Die Gruppe „Kraftwerk“ war dabei von der Idee des Maschinellen fasziniert. Sie verstehen sich deshalb auch als „Musikarbeiter“, die zusammen mit ihren Maschinen Musik erzeugen. Der Titel des 1978 veröffentlichten Album „Mensch-Maschine“ kann programmatisch verstanden werden. [BaFI93]

Große Bedeutung erlangte der Loop auch durch das Aufkommen der Musikrichtung „Hiphop“. Eine der bekanntesten Legenden zur Geburtsstunde des Hiphops besagt, dass DJ Kool Herc 1973 bei einer Geburtstagsparty seiner Schwester zum ersten Mal zwei Plattenspieler verwendete, auf denen Pressungen der gleichen Platte lagen. Passagen der Platte verlängerte er, indem er die betreffenden Stellen abwechselnd von einer der beiden Platten abspielte. Auf diese Weise konnte er Loops erzeugen. Musikstücke mit musikalisch reduzierten Loops waren für diese Technik sehr gut geeignet. Besonders beliebt waren Stellen, in denen nur noch Schlagzeug und die Basslinie zu hören waren. Die reduzierte Instrumentierung war deswegen erwünscht, da Hiphopstücke von einem MC²⁰ begleitet wurden, der über die vom DJ produzierten Loops rappte. Auch lassen sich in vielen Hiphopliedern Loops aus Stücken der Gruppe „Kraftwerk“ finden. Das 1983 veröffentlichte Stück „Renegades of Funk“ eines Pioniers des Hiphops, des New Yorker DJs Afrika Bambaataa, läßt an dieser Inspirationsquelle keinen Zweifel. Dies ist eher ungewöhnlich, da die Philosophie hinter der Gruppe „Kraftwerk“ und der Musikrichtung Hiphop auseinanderklaffen. [Hogg02]

1979 veröffentlichte die Gruppe „Sugarhill Gang“²¹ das Lied „Rapper’s Delight“. Dies wurde das erste HipHop-Lied, das es in die US-amerikanischen Top40 schaffte. Durch das Vordringen des Hiphops in den musikalischen Mainstream wurde die Bedeutung des Loops in der zeitgenössischen Musik verstärkt. [Hogg02]

Die fundamentale Bedeutung des Loops in der Musikrichtung „Hiphop“ wird mit dem Ursprung aus der afrikanischen Musiktradition begründet, wo die Repetition ein wichtiges Element darstellt. Melodische Phrasen sind dabei kurz gehalten und werden oftmals wiederholt, während hingegen in der europäischen Musiktradition der

²⁰ MC: Master of Ceremony: anderer Begriff für Rapper

²¹ Sylvia Robinson, die Managerin des Sugarhill Labels, stellte diese Band zusammen. Eine Ironie des Schicksals, dass die Musikrichtung Hiphop, die stark auf „Street Credibility“ bedacht und von männlichen Sexismen durchdrungen ist, durch eine von einer Labelmanagerin zusammengestellte Band in das Bewusstsein des musikalischen Mainstream vordrang.

Einmaligkeit der Melodien große Bedeutung zugemessen wird. Dem Loop wird in der afrikanischen Tradition die Funktion zugesprochen, Vergangenheit und Zukunft auszulöschen und so das Vergehen von Zeit nicht mehr wahrnehmbar zu machen. Eine andere Interpretation sieht die Repetition als Mittel um Kontinuität und Sicherheit zu vermitteln. [Rose94] [Oluf06]

Bedeutung des Loops heute

Heutzutage ist der Loop ein bestimmendes Stilmittel in der zeitgenössischen popkulturellen Musik. Kulturtheoretiker wie der Franzose Jacques Attali sehen den Loop als eine Manifestation einer industriell geprägten Massenkultur. Die Repetition kann als ein Zeichen einer standardisierten Massenproduktion gedeutet werden. [Rose94]



Abbildung 3: Der Hardware-Step-Sequencer Korg Electribe mkII

Betrachtet man die zeitgenössische elektronische Musik, sind Loops nicht mehr wegzudenken. Viele populäre Musikrichtungen wie Techno oder Drum 'n' Bass leben von repetitiven Wiederholungen. Loops sind dabei zu einem starren Korsett geworden. Tape-Loops konnten unterschiedliche Längen haben. Die erzeugte Musik konnte so eine komplexe Struktur haben. Heute ist der Großteil der Loops einem

strengen Schema von sechzehn Schritten, so genannten „Steps“, unterworfen. In der elektronischen Musikproduktion wurde dieses Schema in die Bauweise der Instrumente integriert. Der Großteil der Hardware- und Softwaresequenzer²² ermöglichen daher lediglich die Eingabe von sechzehntaktigen Sequenzen, die geloopt werden. Dieses Konzept ist etwa in den Hardware-Step-Sequenzern aus der „Korg Electribe“-Serie zu finden, wie zum Beispiel in dem abgebildeten Drumcomputer „Korg Electribe mkII“. Die Leiste der beleuchteten Buttons zeigt dabei die mit Klängen belegten Steps an. Wird der Play-Button gedrückt, zeigt ein Aufleuchten des aktuellen Steps an, wie weit der Loop bereits durchlaufen wurde. Neben der Verwendung von aufgezeichneten oder programmierten Loops, gibt es auch die Möglichkeit, Loops live während der Darbietung aufzuzeichnen und zu verwenden. Ein Beispiel dafür sind die Stücke der Reihe „Differenz/Wiederholung“, die von Bernhard Lang am Institut für Elektronische Musik und Akustik in Graz realisiert worden sind. [Seid02]

Soundmodulation

Klänge können nach der Erzeugung auf verschiedene Arten verändert werden. Die größte Unterscheidung zwischen den verschiedenen Möglichkeiten liegt darin, ob die Veränderung in Echtzeit geschieht oder ob der Klang aufgezeichnet und dann bearbeitet wird. Die letzt genannte Möglichkeit wird auch als Offtime-Soundmodulation bezeichnet. Darunter fallen Techniken wie die Fourieranalyse²³ oder die Umkehrung des Audio-Signals. Andere wie etwa Filter- und Verzerrungsfunktionen können sowohl als Echtzeit-Soundmodulationen als auch als Offtime-Techniken realisiert werden. Im folgenden Kapitel werden die Tonhöhen- und Lautstärkenmodulation von aufgezeichneten Audiosignalen besprochen.

Tonhöhenänderung

Die Tonhöhe eines Klangs kann nach der Erzeugung auf verschiedene Arten verändert werden. Die zwei wichtigsten werden hier aufgeführt. Eine Möglichkeit ist

²² Sequenzer: Ein Sequenzer ist ein elektronisches Gerät oder ein Programm, das eine Abfolge von musikalischen Ereignissen speichert. Dieses Thema wird in einem späteren Kapitel genauer behandelt.

²³ Fourieranalyse: Die vom französischen Physiker und Mathematiker Jean Baptiste Joseph Fourier (1768-1830) entwickelte Analyse, bei der das Signal in harmonische Schwingungen zerlegt wird.

die Änderung der Abspielgeschwindigkeit. Die Tonhöhe wird verändert, indem die Abspielgeschwindigkeit erhöht oder verlangsamt wird. Dies stellt eine sehr einfache Methode dar, deren Wurzeln in den analogen Medien Tonband und Schallplatte zu finden sind. Sie wurde von Musikern der „Tape Music“ verwendet und ist auch im Mehrspurtonbandgerät von Le Caine implementiert. [Rusc98]

Der Nachteil an dieser Methode ist, dass sich der Klang des Samples verändert, da die Formanten²⁴ des Klangs falsch wiedergegeben werden. Daher kommt es beim schnelleren Abspielen von Stimmaufnahmen zum „Mickey-Mouse“-Effekt. Ein weiterer unangenehmer Effekt ist, dass sich bei der Änderung der Abspielgeschwindigkeit auch die Länge des Samples verändert. [Bast06]

Eine andere Möglichkeit ist das so genannte „Pitch-Shifting“. Dabei wird versucht, die Tonhöhe eines Samples zu verändern, ohne die Länge des Samples oder den Klang des Samples zu beeinflussen. Dies kann über verschiedene Wege erreicht werden. Pitch-Shifting kann etwa über den Doppler-Effekt²⁵ umgesetzt werden. [Puck03] Eine andere Lösung, die darauf bedacht ist, die Formanten menschlicher Stimme unbeeinflusst zu lassen, heißt PSOLA (Pitch Synchronous Overlap/Add). Dabei wird das Signal in kleinste Teile zerschnitten, diese Teile in der Tonhöhe erhöht und überlappend zusammengesetzt. [Bern05]

Lautstärkenänderung

Die menschliche Empfindung der Lautstärke eines Signals ist keine physikalische Größe. Dieses wird vom Schalldruckpegel, der zeitlichen Struktur und des Frequenzspektrums des Signals beeinflusst.

Das in Gesetzestexten herangezogene Maß der Lautstärke eines Signals ist der Schalldruckpegel. Dabei handelt es sich um das logarithmische Verhältnis des Schalldrucks²⁶ zu einem Referenzschalldruck, dessen Wert bei Messungen, bei denen die Lautstärke festgestellt werden soll, den Wert der Hörschwelle²⁷ des Menschen zugewiesen bekommt. Die Einheit ist Dezibel (dB). Aus der Definition des

²⁴ Formant: Die Maxima von Frequenzen, die durch die Klangerzeugung am stärksten verstärkt werden. Die Formanten bestimmen stark die Klangfarbe eines Instruments. [BeSt05]

²⁵ Doppler-Effekt: Der nach dem österreichischen Physiker Christian Doppler benannte Effekt, dass sich die Frequenz eines Signals verändert, wenn diese sich relativ zum Betrachter bewegt.

²⁶ Schalldruck: Druckschwankungen eines Medium

²⁷ Die Hörschwelle des Menschen liegt bei einer Frequenz von 1000 Hertz bei einem Schalldruck von 20 Mikro-Pascal.

Schalldruckpegels lässt sich ablesen, dass eine Verdopplung des Schalldrucks keine Verdopplung des Schalldruckpegels bedeutet. In der Praxis heißt das, dass erst zehn Geigen doppelt so laut wie eine Geige wahrgenommen werden. Das Lautstärkeempfinden hängt aber auch von der Frequenz eines Tons ab. Ein Ton mit 60 Hertz und einem Schalldruckpegel von 40 dB wird weniger laut wahrgenommen als ein 1000 Hertz-Ton des gleichen Schalldruckpegels. In experimentellen Tests an Menschen wurden die Kurven des Lautstärkeempfindens von Tönen unterschiedlicher Frequenz ermittelt. Diese psychoakustische Größe wird als Lautstärkepegel bezeichnet und besitzt die Einheit „Phon“. [Roed77]

Neben dieser Maßeinheit wurde auch noch die Einheit „sone“ eingeführt, die das Maß für die Lautheit ist. Diese wurde vom Psychologen Stanley Smith Stevens eingeführt und bezeichnet die subjektive Wahrnehmung der Lautstärke. [Schic79]

Der Zusammenhang zwischen Lautheit und Lautstärkepegel ist bei Tönen im Bereich zwischen 40 und 120 Phon annähernd logarithmisch. [Roed77]

Bei Klängen spielt das Frequenzspektrum des Signals eine wichtige Rolle. Je breitbandiger ein Signal ist, desto lauter wird es wahrgenommen. Auch spielt die zeitliche Struktur eine Rolle in der auditiven Wahrnehmung. Die psychoakustische Größe der „Lautheit“ kann also bei Klängen nicht vollständig von physikalisch messbaren Größen determiniert werden. [KaSt98]

Zusammenfassung

In diesem Kapitel wurden die musikalischen Grundlagen des Programms „soundtubes“ vorgestellt. Dabei handelt es sich um den Loop, das Sampling und die Soundmodulation. Sowohl der Loop als auch das Sampling sind Techniken, die erstmals nach 1945 von elektro-akustischen²⁸ Komponisten verwendet wurden. Anfangs wurden beide Techniken noch sehr frei eingesetzt. Mit dem Einzug in den musikalischen Mainstream wurden sie aber mehr und mehr zu einem starren Korsett, das heutzutage in vielen elektronischen Sequenzern auf technologischer Ebene verankert ist. Grosse Bedeutungen erlangten beide Techniken auch in der Musikrichtung „Hiphop“, in der sie die musikalische Basis darstellen.

²⁸ Elektro-akustische Musik bezeichnet heute komponierte Musik, die auf elektronischen Weg erzeugt wird. Mit dem Begriff will man sich dezidiert von der elektronischen Musik abgrenzen, die als Teil der Popmusik verstanden wird.

In diesem Kapitel wurden weiters zwei Techniken der Soundmodulation, die Tonhöhenänderung und die Lautstärkenänderung, diskutiert. Soundmodulation meint hier, die Veränderung eines Klangs nachdem dieser bereits erzeugt worden ist. Die Änderung der Tonhöhe kann auf verschiedene Arten realisiert werden. Hier wurden die beiden Techniken der Änderung der Abspielgeschwindigkeit und des so genannten „Pitch-Shifting“ vorgestellt. Die Lautstärkenänderung ist eine einfach realisierbare Technik, wobei aber bedacht werden muss, dass die Lautstärke eines Klangs nicht physikalisch gemessen werden kann, da sie eine psychoakustische Größe ist.

Interface

In diesem Kapitel werden die Grundlagen des Interfaces des Programms „soundtubes“ erläutert. Dabei werden die Themen der Eingabegeräte, der direkten Manipulation und der Metaphern diskutiert. Um die Verwendung der Metaphern im Interface zu verdeutlichen, werden gängige Metaphern in Musikprogrammen und ihre Entstehungsgeschichte vorgestellt

Eingabegeräte

Die Eingabegeräte Maus und Tastatur sind die wichtigsten Computerperipheriegeräte. Die um 1870 erfundene Tastatur und die um 1960 erfundene Maus behaupten sich trotz der Entwicklung von vielen anderen Eingabegeräten wie Joysticks, Touchpads oder Graphik-Tablets. [Togn95] Schon 1985 gingen Hutchins, Hollan und Norman in einem Paper davon aus, dass die bestehenden Interaktionsmöglichkeiten über Maus, Tastatur und Bildschirm nicht ausreichend sind und bald durch andere Geräte erweitert werden. [HuHo85] Ein Grund dafür, dass Mäuse weit verbreitete Interface-Geräte sind, liegt unter anderem darin, dass die Maus ein kinästhetisches Eingabegerät ist. Die kinästhetische Wahrnehmung ist ein Teil des haptischen Sinnes, der für Bewegungsempfinden zuständig ist. Um die Benutzung der Maus einfach zu gestalten, sollten das kinästhetische und das visuelle Wahrnehmen in Relation stehen. Wenn der Benutzer die Maus bewegt, sollte sich der Cursor eine in Verhältnis stehende Strecke in die gleiche Richtung bewegen. [Togn91] Die Maus fällt unter die Kategorie der Zeigegeräte. Die ersten Zeigegeräte stellten die „Lightguns“ dar. Diese waren die Eingabegeräte für die ersten Radarschirmcomputer, die um 1950 für das amerikanische Militär entwickelt wurden. Dieses Zeigegerät weitete erstmal das Frame-Konzept aus. Der Bildschirm als klar abgegrenzte Fläche, in der eine andere Realität dargestellt wird, ist ein altes Konzept, das sich von gemalten Bildern bis hin zu Fernseh- und Computerbildschirmen durch die Geschichte verfolgen lässt. Der Inhalt des Frames war dabei immer nur eine Fläche zur Darstellung. Mit den „Lightguns“ und später mit der Maus wurde der Bildschirm eine Fläche für die Interaktion. Da der Bildschirm aber nun synchron für die Eingabe und Ausgabe von Informationen benutzt wird, sind Interaktionsdesigner gefordert, Eingabe- und Ausgabebereiche klar erkenntlich

zu machen. Darüber hinaus wurde in einem Frame immer nur ein Bild dargestellt. Das in den wichtigsten Betriebssystemen implementierte Interaktionsprinzip WIMP²⁹ erlaubt die Darstellung von mehreren Fenstern innerhalb eines Frames, was für Interaktionsdesigner eine weitere Herausforderung ist. [MaLe01]

Die Maus selbst wurde von Douglas Engelbart entwickelt. 1970 wurde diese Idee am Xerox PARC³⁰ weiterentwickelt, wo die erste Kugelmaus entstand. [Togn95] Im Laufe der Zeit änderte sich die Anzahl der Maustasten. Einige Mäuse wurden mit bis zu fünf Tasten ausgestattet. Die heutzutage am häufigsten verwendete Standard-Maus besitzt zwei Maustasten und ein Scrollrad.

Obwohl mit diesen Tasten eine Vielzahl von Operationen unterstützt werden könnten, hat sich eine kleine Anzahl von unterstützten Benutzungsarten durchgesetzt. Diese sind der Einfach- und Doppelklick, „Drag-and-Drop“ und das Scrollen. Mit ihnen werden auch kurz die Operationen, die sie auslösen, angeführt. Ein Einfachklick der linken Maustaste wird zur Markierung von Elementen oder zum Setzen eines Bearbeitungscursors in Texteditoren benutzt. In Internet-Browsern werden durch Einfachklicks Links geöffnet. Ein Einfachklick der rechten Maustaste öffnet ein kontextabhängiges Menü. Ein Doppelklick der linken Maustaste öffnet das Element. „Drag-and-Drop“ bezeichnet das Ziehen der Maus bei gedrückter linker Maustaste. Dies wird benutzt, um markierte Elemente zu verschieben oder größere Markierungen vorzunehmen. Auch werden „Drag-and-Drop“ sowie Einfachklicks mit der linken Maustaste oftmals in Verbindung mit dem Gedrückt-Halten einer Tastatur-Taste verwendet, um komplexere Markierungen vorzunehmen. Das Scrollrad ermöglicht bei Programmen, die nur einen Teil der ganzen Ansicht zeigen, das Verschieben des Ausschnitts nach oben oder unten.

Direkte Manipulation

Der Begriff „Direkte Manipulation“ wurde 1983 von Ben Shneiderman eingeführt. Vier Merkmale kennzeichnen Interfaces, die nach diesem Prinzip gestaltet werden.

- Das Objekt des Interesses ist ständig repräsentiert.

²⁹ WIMP: Windows, icon, menu, pointing device. Eine andere Bezeichnung für die ebenfalls am Xerox PARC entwickelte, heute in vielen Betriebssystemen implementierter Form des graphischen Userinterfaces. Dass die Maus als „pointing device“ explizit als Kennzeichen solcher Interfaces aufgeführt wird, zeigt die Bedeutung dieses Eingabegeräts.

³⁰ Xerox Palo Alto Research Center

- Die Interaktion geschieht über physische Eingabegeräte wie Mäuse oder Joysticks.
- Die getätigten Operationen sind sofort am Objekt sichtbar. Dies wird im Interface-Design auch als „Feedback“ bezeichnet.
- Das Programm ist mit minimalem Wissen benutzbar. Laien können schnell mit dem neuen Programm umgehen und erweitern ihr Wissen über die Handhabung bei der Benutzung. Die Lernkurve ist steil, jedoch um zum Experten zu werden, bedarf es längerer Beschäftigung.

Eine Hoffnung hinter dem Paradigma der direkten Manipulation ist, dass der Benutzer sich voll und ganz auf die Aufgabenstellung, die er am Computer lösen möchte, konzentrieren kann. Der Lernprozess wird beschleunigt. Der Computer als „Ding“ zwischen Mensch und der Umsetzung seiner Ziele verschwindet. Der Computer wird transparent, da der Benutzer direkt an seinen Aufgaben arbeitet. Komplexe Befehlsfolgen werden durch direktes Interagieren abgelöst.

Konstruktivistische Lerntheorien nach Piaget³¹ stehen hinter diesen Überlegungen. [Shne83] [Huho85]

Wie bei einem Theaterstück, müssen die Benutzer bei Interfaces, die nach dem Paradigma der direkten Manipulation gestaltet sind, absichtlich daran glauben, dass das Gesehene der Realität entspräche. Als „Belohnung“ für diese Selbsttäuschung können die Benutzer schneller ihre gewünschten Handlungen umsetzen. [HuHo85] Mit der direkten Manipulation in Verbindung steht der Begriff des „direct engagement“. Damit wird das Gefühl des Benutzers beschrieben, wenn dieser direkte Interaktion mit einem Objekt wahrnimmt. Das System, mit dem der Benutzer arbeitet, soll ihm eine sofortige Rückmeldung oder zumindest die Illusion davon geben. Weiters soll auf vorhergehende Ausgaben zurückgegriffen werden können und das Interface soll dezent gestaltet sein. Das Interface wird also aus der Sicht des Benutzers betrachtet. [Laur93] Dabei sollen die Benutzerschnittstellen so gestaltet werden, dass möglichst wenig „Denkarbeit“ in den Umgang mit dem Interface investiert werden muss. Man erhofft sich so ein stärkeres Gefühl der Direktheit in der Interaktion. [HuHo85]

Direkte Manipulation wird weder vom Begründer des Begriffs, Ben Shneiderman, noch von anderen Forscher auch als Allheilmittel für Computer-Interface-Probleme

³¹ Jean Piaget: Schweizer Entwicklungspsychologe, der als Vorläufer des radikalen Konstruktivismus betrachtet wird.

gesehen. Abstrakte Konzepte wie Variablen sind schwer umzusetzen. Oftmals mangelt es an Präzision bei der Eingabe. Durch die Konzentration auf den Benutzer bei der Gestaltung des Interfaces, kann dieser keine neuen Denkweisen im Umgang mit dem System erschließen. Verständnisprobleme über den Bereich, an dem der Benutzer arbeitet, werden vom Benutzer als Probleme des Interfaces aufgefasst. [HuHo85]

Metapher

Eine Metapher ist im eigentlichen Sinne eine rhetorische Figur, bei der ein Wort in einer übertragenen Bedeutung verwendet wird. Metaphern werden meist als poetische Strukturen verstanden, die der Ausschmückung der Sprache dienen. Wie im Buch „Metaphors we live by“ von Lakoff³² und Johnson³³ ausgeführt wird, stellen Metaphern grundlegende Denkstrukturen dar, die unser Denken und unsere Wahrnehmung formen. Dabei werden bestimmte Aspekte der Wirklichkeit hervorgehoben und andere in den Hintergrund gedrängt. Ein Beispiel aus dem Buch macht diesen Ansatz nachvollziehbar. Ein ausländischer Student, der einen Kurs von Johnson belegte, verstand den Begriff der „Solution of problems“ (die Lösung der Probleme) nicht als Auflösung der Probleme, sondern als eine chemische Lösung, in der alle Probleme schwimmen, sich teilweise auflösen, teilweise mit anderen Problemen eine Bindung eingehen und so zu neuen Problemen werden. Probleme sind diesem Verständnis nach nie wirklich (auf)gelöst, sondern treten ein Leben lang in abgewandelter Form wieder auf. Die vollständige (Auf-)Lösung eines Problems ist nicht in der Form möglich, wie diese Metapher meist verstanden wird. [LaJo80]

Metaphern können dabei nicht explizit formuliert werden, da sie durch ihre wirklichkeitsstrukturierende Wirkung erst den zwischenmenschlichen Diskurs ermöglichen. Anhand des Beispiels der „Solution of problems“ kann dies verdeutlicht werden. Wenn Personen von der Lösung eines Problems reden, so reden sie vom gleichen mentalen Bild, das hinter dieser Metapher steht. Der Dialog kann geführt werden, ohne sich vorher über die Bedeutung auszutauschen. [Pflü04]

Es wird daher davon ausgegangen, dass die Verwendung von Metaphern in Benutzerschnittstellen dem Benutzer hilft mit dem Computer zu arbeiten. Dem Benutzer soll ein mentales Modell der Benutzung vermittelt werden. Es geht daher

³² George P. Lakoff: Professor der Linguistik an der Universität von Kalifornien, Berkeley.

³³ Mark L. Johnson: Professor an der Abteilung für Philosophie an der Universität Oregon.

nicht um eine lebensechte Nachbildung. [Laur93] Eine der bekanntesten Metaphern ist die Desktop-Metapher, die in 70er Jahren am Xerox PARC entwickelt wurde. Die Benutzerschnittstelle ist dabei als ein Schreibtisch gestaltet, auf dem Dokumente, Ordner und Anwendungen angelegt werden können. Die verfügbaren Funktionalitäten gehen aber weit über die Nachahmung der Realität hinaus. Trotzdem kann aus dem Verständnis an einem Schreibtisch zu arbeiten, der Computer einfach bedient werden.

Die strukturierende Wirkung der Metapher schränkt die Form der Interaktion stark ein, da diese, um dem Benutzer bei der Bedienung und dem Verständnis zu helfen, nicht gebrochen werden darf. Der Zuwachs an Bedienungsfreundlichkeit durch ihre Verwendung kann bei einer Erweiterung der Funktionalitäten zu einem Hindernis werden. Dies ist besonders dann der Fall, wenn die Ausweitung der Funktionalität nicht entspricht. Das Beispiel der Internet-Suchmaschine „Yahoo!“ macht diesen Punkt anschaulich. Diese besaß 1999 neben der Suchfunktionalität ein Ordnersystem, in dem Internetseiten eingeordnet waren. Dieser Ansatz ist der Desktop- und Ordner-Metapher sehr ähnlich und der Netzstruktur des Internets nicht entsprechend. [Yaho99]

Die Auswahl einer geeigneten Metapher ist keine triviale Aufgabe. Große Hoffnungen wurden in die Verwendungen von Metaphern gelegt, die der „wirklichen“ Welt entlehnt sind. Studien kamen zum Schluss, dass wenn das Benutzerinterface dem Benutzer kein mentales Modell zur Funktionsweise nahe legt, der Benutzer sich eine eigene Erklärung konstruiert. Ein Problem an der Verwendung von Metaphern, die der Realität entlehnt sind, ist, dass die Möglichkeiten des Einsatzes von Computern umfangreich sind und nur annäherungsweise so beschrieben werden können. Eine Studie von Blackwell und Green legt den Schluss nahe, dass der Einsatz von Metaphern aus der Realität dem Benutzer nicht automatisch den Umgang mit dem Computer erleichtert. [BIGr99]

Ein weiteres Problem bei der Verwendung von Metaphern für Benutzerinterfaces ist, dass der Benutzer diese leicht entschlüsseln können muss. Eine Metapher, die für den Programmierer klar verständlich ist, muss nicht unbedingt von der Anwenderseite verstanden werden können. [Shne83]

Metaphern in Musiksoftware

In diesem Kapitel werden Metaphern vorgestellt, die im Großteil der Musiksoftware verwendet werden. Die Metaphern sind nicht nur eine Hilfestellung für den Benutzer, sondern sie stellen dabei implizit eine bestimmte Vorstellung von Musikproduktion dar. Wie auch bei anderen Interfaces sind die hier vorgestellten Metaphern aus der Realität entnommen. Daher besitzen diese eine geschichtliche Entwicklung, deren Kenntnis das Verständnis um die Wirkung verstärkt.

Klaviatur

Die Klaviatur bezeichnet das Bild eines Tastaturausschnitts eines Tasteninstrumentes wie etwa eines Klaviers.

Hinter der Metapher der Klaviatur steht die Vorstellung, dass der Benutzer nur Klänge von diskreten Frequenzen erzeugen möchte, obwohl die elektronische Klangerzeugung keine Notwendigkeit für diskrete Notenwerte kennt. Neben dieser Einschränkung ist es dem Benutzer nicht möglich, den Klang des Instruments über das Interface zu verändern. Darüber hinaus ist die Klaviatur für andere elektronische Klangerzeugungstechniken wie dem Sampling kein geeignetes Interface.



Abbildung 4: Ein Beispiel für eine Klaviatur

Der Grund für die Verwendung der Klaviatur in vielen Musikprogrammen liegt darin, dass der Großteil der elektronischen Musikinstrumente so bedient wird. Schon das erste elektronische Musikinstrument, der „Musical Telegraph“, der 1876 von Elisa Gray erfunden wurde, musste mit einer Klaviatur bedient werden. Zwar legte das zur Klangerzeugung notwendige Schließen eines Stromkreises die Verwendung einer Tastatur nahe, aber zwingende Gründe gab es dafür nicht. Ein wichtiger Grund lag eher darin, dass der Großteil der akustischen klangerzeugenden Instrumente über Handbewegungen gesteuert wurde. [KiHu99]

Sogar elektronische Musikinstrumente, die nicht unbedingt die Verwendung des Interfaces Klaviatur nahe legen, werden so gesteuert. Ein gutes Beispiel dafür ist das Mehrspurtonbandgerät von Le Caine, das im Kapitel „Sampling“ bereits vorgestellt worden ist.

Im Verlauf der Zeit hat es viele verschiedene Versuche gegeben, das Konzept der Klaviatur abzulösen oder zu erweitern. Viele Musiker waren und sind mit den Interaktionsmöglichkeiten der Musikinstrumente unzufrieden. Jörg Piringer beschäftigte sich in seiner Diplomarbeit „Elektronische Musik und Interaktivität“ ausgiebig mit dem Thema und stellt auch viele verschiedene Interfaces und Interfaceversuche vor. [Piri01]



Abbildung 5: Lev Sergejewitsch Termen spielt am Theremin

Das erfolgreichste elektronische Musikinstrument mit einem alternativen Interface ist das Theremin. Dies ist ein berührungslos zu spielendes Instrument, das erstmals 1921 von Lev Sergejewitsch Termen in Moskau präsentiert wurde. Die Tonhöhe und Lautstärke werden durch ein elektrisches Feld eines Kondensators gesteuert, der aus der Hand des Spielers und der Antenne des Theremins gebildet wird. Jede kleinste Veränderung der Handstellung verändert die Kapazität des Kondensators und somit die Tonhöhe. Dabei zeigen sich zwei große Stärken der Klaviatur. Während der Theremin auf jede minimale Handbewegung reagiert, ist die Klaviatur einfach zu bedienen, da nur dezidierte Handbewegungen das Instrument steuern. Auch lassen sich mit der Klaviatur schnelle Tonhöhenwechsel und eine genaue Tonhöhensteuerung einfacher bewerkstelligen. [Rusc98]

Piano-Roll

Eine Metapher, die sich aus der Klaviatur entwickelt hat, ist die Piano-Roll, die sich in vielen Musikprogrammen finden lässt.

Die Piano-Roll ist das Speichermedium für Musikstücke bei Drehorgeln, Spieluhren, Glockenspielen und Klavierspielautomaten und trat daher erstmals zur gleichen Zeit wie diese Musikautomaten auf. Die Piano-Roll steuert bei der Drehorgel, welche Pfeife zu welcher Zeit gespielt werden soll. Daher kann sie als erster technischer Sequenzer angesehen werden. Da Orgeln normalerweise über Klaviaturen gespielt werden und bei der Abtastung der Piano-Roll bei Drehorgeln die Pfeifen über Klaviatur ähnliche, so genannte „Claves“, gesteuert werden, lässt sich hier eine Verwandtschaft herstellen.



Abbildung 6: Drehorgel mit Piano-Roll

Die Piano-Rolls und Drehorgeln traten erstmals im arabischen Raum auf. Die ersten Piano-Rolls waren Metallwalzen mit Stiften, auf denen die musikalischen Informationen gespeichert sind. Der Spieler dreht an einer Kurbel, wobei diese Walze in Drehung versetzt wird und so die Melodie abgespielt wird. Erst um 1700 kam die Drehorgel in Europa auf. Gesicherte Informationen über die Entstehung und die Geschichte gibt es aber wenige, da dieses Instrument eines der unteren Volksschichten war. Trotzdem haben einige bekannte Komponisten wie Mozart, Händel oder Hadyn Stücke für die Drehorgel arrangiert oder komponiert. 1702 wurden im Auftrag der Pariser „Académie Royale des Sciences“ alle in Frankreich

üblichen Instrumente beschrieben. Darunter befand sich auch die Drehorgel, die als „orgue de Barbarie“ bezeichnet wurde. Diese Bezeichnung lässt die Vermutung zu, dass diese von Deutschland aus in Europa Verbreitung fand. Ein Zentrum der Drehorgelproduktion war Waldkirch in Bayern. [Köni02] Der Begriff der „Drehe-Orgel“ wurde erstmals 1742 in Valentin Trichters Gesellschaftslexikon erwähnt. Das Instrument wurde damals dazu benutzt, Vögeln das Singen zu lernen. [MeKr94] Die Drehorgel wird mit dem so genannten „fahrenden Volk“ und Straßenmusikern assoziiert. Mit dem Raritätencharakter dieses Instruments ließ sich gut Geld verdienen. Noch im 19. Jahrhundert waren Musikautomaten wie die Drehorgel die große Attraktion auf Jahrmärkten. Dabei trat diese immer im Verband mit anderen Raritäten auf. Anfangs als Begleitinstrument zu Vorführungen der „Latema magica“³⁴ verwendet, untermalte sie später den Gesang von Moritätensängern³⁵. In Wien verdienten viele Invaliden als Drehorgelspieler ihren Lebensunterhalt. So wurden etwa die Kriegsinvaliden des Siebenjährigen Kriegs (1756 - 1763) vor die Wahl gestellt, eine schlecht bezahlte Anstellung im öffentlichen Dienst zu bekommen oder als Drehorgelspieler Geld zu verdienen. Dieses Instrument wurde deshalb gewählt, da es auch bei Verlust eines Beins oder Arms noch immer bedienbar war. Die Kriegsinvaliden zogen mit der so genannten „Maria-Theresia-Konzession“ vom 3. Mai 1778 von Lokal zu Lokal. Noch 1838 gab es rund 800 solcher Konzessionen. [Köni02]

1871 wurden erstmalig Karton-Lochkarten zur Steuerung von Klavierspielautomaten benutzt. Diese Lochkarten wurden 1805 von dem Franzosen Joseph Marie Jacquard zur Steuerung von Webmaschinen erfunden. 1885 führte die französische Drehorgelbauerfamilie Gavioli die Karton-Notenbänder für Drehorgeln ein. Bis dahin wurden bestiftete Metallwalzen benutzt, deren Herstellung sehr teuer und arbeitsintensiv war. Um acht Melodien von der Länge von einer Minute zu programmieren, mussten ungefähr 20.000 Stifte geschlagen werden. Um 1900 erfanden die Waldkircher Orgelbauer Ruth die pneumatische Abtastung, durch die es möglich wurde, Notenbänder aus Papier zu verwenden.

³⁴ Laterna magica: Ein Bild wurde mittels einem Hohlspiegel auf eine künstlich erzeugte Rauchwolke gelenkt und erzeugte so die Illusion eines plastischen Bilds.

³⁵ Moritaten sind im 19. Jahrhundert weit verbreitete Lieder, die oftmals von Mordtaten (daher der Name) handeln und eine moralische Botschaft haben.

In Wien, wo sich eine eigene Drehorgelkultur herausgebildet hat, werden heute noch Drehscheiben verwendet, die wesentlich robuster als Papiernotenbänder sind. Diese Eigenschaft wird deswegen geschätzt, da der so genannte „Werkmann“³⁶ die Kurbeln nicht gleichmäßig dreht, sondern durch Geschwindigkeitsänderungen³⁷ im Drehen in den Musikstücken noch eigene Akzente setzt. Diese Technik setzt eine genaue Kenntnis des Musikstücks voraus und kann nur mit Drehscheiben umgesetzt werden, da Papiernotenrollen zu sehr beansprucht werden würden. [MeKr94]
[MaSc00]

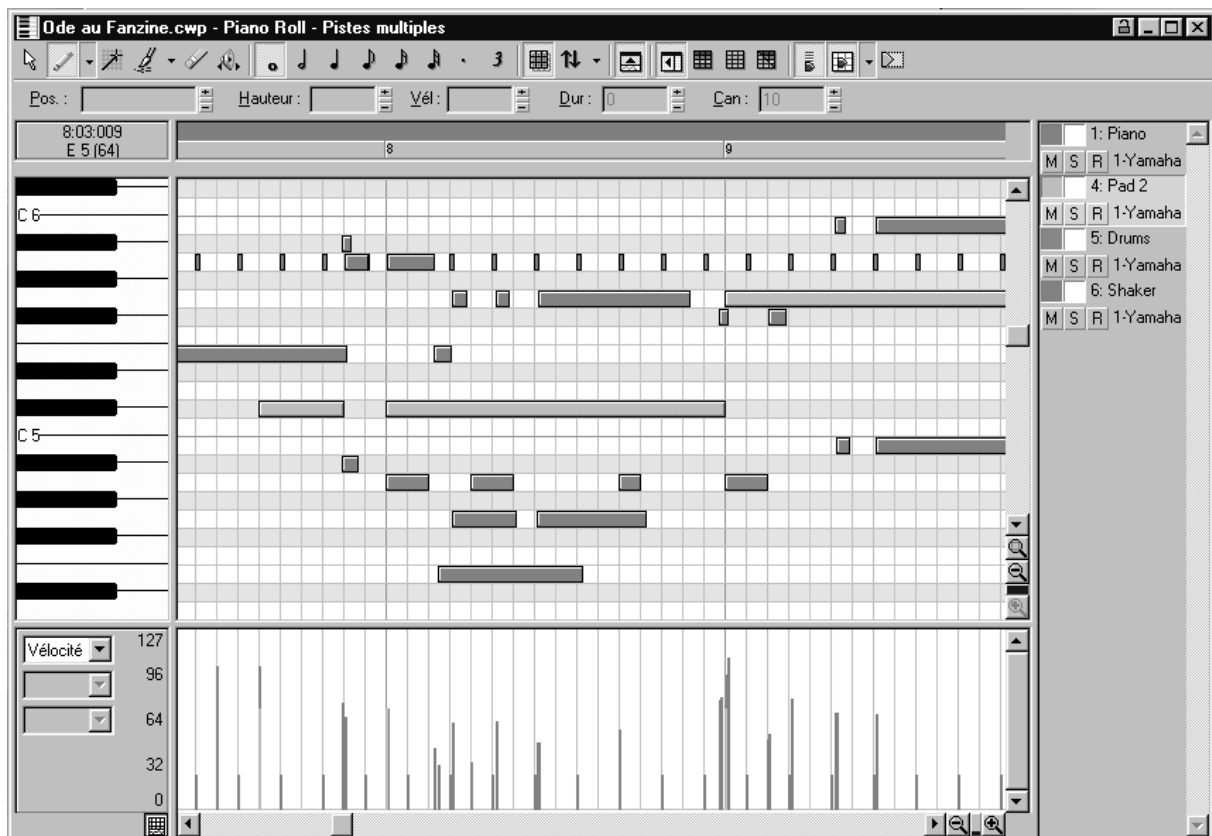


Abbildung 7: Piano-Roll im Programm „Cakewalk Sonar“

Die Piano-Roll stellt wie auch andere Notationsarten eine zweidimensionale Notation dar, wobei immer die Zeitdimension eine der beiden Achsen darstellt. Die Zeit läuft dabei für alle musikalischen Ereignisse gleich schnell und es besteht eine kleinste gemeinsame Zeiteinheit. Der Drehorgelspieler kann aber durch die Geschwindigkeit der Drehung der Kurbel diese kleinste gemeinsame Zeiteinheit variieren und so die auf der Piano-Roll festgehaltene Melodie persönlich interpretieren. [MeKr94]

³⁶ Wiener Bezeichnung für Drehorgelspieler. Die Bezeichnung kommt von dem wienerischen Ausdruck „Werk“ für die Drehorgel.

³⁷ Diese Technik wird auf Hochdeutsch übersetzt als „Viereckig-Herumdrehen“ bezeichnet.

Sequenzier

Ein Sequenzer ist ein elektronisches Gerät oder ein Programm, das eine Abfolge von musikalischen Ereignissen speichert. [Piri01]

Die Piano-Roll mit dazu passendem Abspielgerät stellt einen Sequenzer dar. Ähnlich wie bei einer Piano-Roll werden Sequenzer meist nur dazu verwendet um Notenwerte zu speichern. Andere Parameter werden meist in nur sehr geringem Maße von Sequenzern gesteuert. Das ist eigentlich paradox, da es bei elektronischen Klangerzeugern erstmals möglich war, eine Vielfalt von verschiedenen Klängen zu erzeugen und die Klanggestaltung auch vollständig zu steuern. Besonders diese Möglichkeit faszinierte den Großteil der Komponisten bei ihrem ersten Kontakt mit elektronischer Musik. [Rusc98]

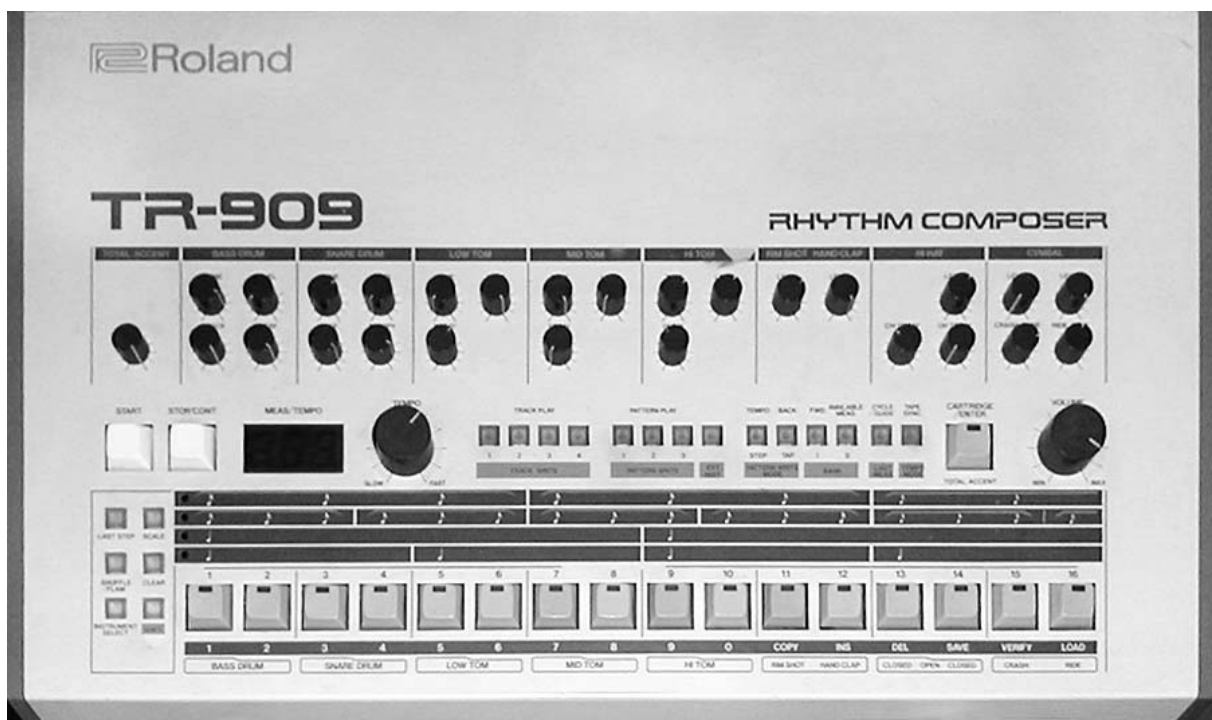


Abbildung 8: Drumcomputer Roland TR 909, Markteinführung: 1983

Die ersten elektronischen Sequenzer wurden nach dem zweiten Weltkrieg konstruiert. 1955 entwickelte die Firma RCA einen elektronischen Klangerzeuger, der erstmals mit dem Begriff „Synthesizer“ bezeichnet wurde. Dieser Synthesizer konnte nicht nur Klänge erzeugen. Auch war ein elektronischer Sequenzer eingebaut, der über Lochkartenstreifen gesteuert wurde. Zur gleichen Zeit konstruierte der Kanadier Hugh Le Caine den „Coded Music Apparatus“. Dieses Gerät wandelte fünf auf einer Papierrolle gezeichnete Kurven in entsprechende Klänge um. Der Erfinder stellte sich vor, dass Komponisten Papierrollen mit den gezeichneten Kurven an ihn

schicken und er ihnen die entsprechenden Musikstücke aufgezeichnet zurückschicken würde. [Zand98] [Rusc98]

Die Produktion von elektronischer Musik war durch das vielfache Hantieren mit Tonbändern ein langwieriger Prozess. Donald Buchla und Morton Subotnick, die zum Umkreis des „San Francisco Tape Music Center“ gehörten, träumten von einer Möglichkeit, alle Produktionsschritte in einer Einheit zu vereinen. 1963 stellte Buchla den ersten Prototypen des „100 Series Modular Electronics Music System“ vor, der zwei achtstufige und einen sechzehnstufigen analogen Sequenzer eingebaut hatte. 1971 entwickelten David Cockerell und Peter Zinovieff für die Firma EMS den ersten digitalen Sequenzer. [Rusc98]

Da ein Sequenzer eine zeitliche Abfolge speichert, muss es eine Repräsentation der Zeitebene geben. Diese Zeitebene ist für alle Spuren gleich und macht eine Definition eines kleinsten gemeinsamen Grundtaktes notwendig. Diese kleinste Zeiteinheit definiert aber im Großteil der Sequenzer nicht nur die Grundgeschwindigkeit, sondern auch die kleinste musikalische Grundeinheit. Das Beispiel eines Hardware-Drumsequenzers wie dem abgebildeten Roland TR 909, macht diesen Punkt anschaulich: Ein „normaler“ Drumsequenzer besitzt sechzehn Steps, die entweder ein- oder ausgeschaltet sein können. Wird die Grundgeschwindigkeit auf 100 BPM (beats per minute) eingestellt, ist die Eingabe von feineren musikalischen Strukturen nicht möglich. Im Gegensatz dazu kann man bei der Notenblattnotation, unabhängig von der Grundgeschwindigkeit des Stückes, kleinere Notenwerte setzen. Bei Hardware-Sequenzern ist aber meist nur die Eingabe von ganzen Noten möglich. Will man nun feinere Strukturen bauen, müsste man die Geschwindigkeit um ein Vielfaches erhöhen. Da aber die eingegeben sechzehn Steps immer geloopt werden und die Eingabe von komplexeren Strukturen erst bei neueren Geräten möglich ist, muss zu Gunsten eines Grundrhythmus auf diese komplexe Strukturen verzichtet werden.

Die Beschränkung der Sequenzer auf sechzehn Steps ist auch der Grund warum eine Vielzahl elektronischer Musikstücke im Viervierteltakt gehalten ist. [Piri01]

MIDI

MIDI ist die Abkürzung für „Musical Instrument Digital Interface“. Dabei handelt es sich um ein digitales Kommunikationsprotokoll für elektronische Klangerzeuger, Sequenzer und andere elektronische Musikinstrumente. Als Vorläufer kann das analoge Kommunikationsprotokoll CV (Control Voltages) angesehen werden, das bei

analogen Modulsynthesizern die Kommunikation zwischen den einzelnen Modulen ermöglichte. Durch CV konnten erstmals Klangerzeugung und Klangsteuerung voneinander getrennt werden. CV war aber kein standardisiertes Protokoll und daher unterschied sich die Implementierung bei den einzelnen Herstellern. Durch die Einführung des MIDI-Standards wurde es möglich, dass beliebige elektronische Steuerungsgeräte unterschiedliche elektronische Klangerzeuger steuern konnten. Der Sequenzer als Einzelgerät konnte so realisiert werden. Auch konnten im Gegensatz zu CV über MIDI mehrstimmige Synthesizer gesteuert werden. Das MIDI-Protokoll kann über sechzehn Kanäle Daten wie Notenwerte, Steuerungsbefehle und ein Clocksignal übertragen. Es werden nur Befehle aber keine Audiosignale übertragen. Über Notenbefehle kann die Tonhöhe und über Steuerungsdaten können Parameter der Klangerzeuger gesteuert werden. Um unterschiedliche Geräte zu synchronisieren, kann ein Clocksignal weitergeleitet werden. Um MIDI übertragen zu können, müssen die Geräte über spezielle MIDI-Kabel verbunden sein. MIDI-Daten können im plattformübergreifenden Standard-MIDI-Format aufgezeichnet und gespeichert werden. Später wurde das MIDI-Protokoll erweitert, um damit etwa Multimedia-Geräte steuern zu können.



Abbildung 9: MIDI-Kabel

Die Einführung des MIDI-Protokolls ging auf die Initiative von Dave Smith, dem Gründer des Synthesizerherstellers „Sequential Circuits“, zurück. Er schlug vor, ein gemeinsames Kommunikationsprotokoll für elektronische Musikinstrumente namens Universal Synthesizer Interface (USI) einzuführen. Dies war eine neue Idee, da bis dahin jeder Synthesizer-Hersteller aus Angst vor Verkaufsverlusten sein eigenes Kommunikationsprotokoll entwickelte. Die Idee fand dennoch großen Zuspruch und so wurde von 1981 bis 1983 die Spezifikation des MIDI-Standards entwickelt. MIDI wurde 1983 erstmals im Synthesizer „Sequential Circuits Prophet 600“ vollständig implementiert. Heute ist MIDI sehr etabliert und stellt einen Standard im Bereich der elektronischen Musikerzeugung dar.

Durch die Einführung dieses Kommunikationsprotokoll wurden einige Vorstellungen von Musik technologisch verankert.

MIDI ist sehr stark auf akustische Instrumente eingestellt. Es wurde davon ausgegangen, dass Musiker bei der Musikproduktion hauptsächlich auf akustische Klänge zurückgreifen. Deshalb wurde jedem MIDI-Kanal ein anderes Instrument zugeteilt, die gemeinsam eine Zusammenstellung der gängigsten akustischen Instrumente darstellen. So ist der MIDI-Kanal 10 für Schlagzeugklänge vorgesehen. Werden MIDI-Files am Computer abgespielt, wird auf diese Zuordnung von Kanälen auf Instrumente zurückgegriffen. Das so entstehende Klangbild ist jedem als typischer MIDI-Klang bekannt. Der 1991 eingeführte General MIDI Standard ging hier noch einen Schritt weiter und fixierte eine Zuteilung von einer größeren Anzahl von MIDI-Kanälen auf genau spezifizierte Instrumente. So ist bei General MIDI etwa der MIDI-Kanal 25 für das Instrument „Acoustic Guitar (nylon)“ vorgesehen.

MIDI war als Kopplung zwischen Klaviatur-Interface und Klangerzeuger gedacht. Es ist also zur Übertragung von diskreten Notenwerten gedacht. Töne, deren Frequenz zwischen der diskreter Notenwerte liegt, können nur über den Umweg der Verwendung des Pitch-Kommandos³⁸ übertragen werden.

Ein weiterer Nachteil von MIDI ist, dass die Datenübertragung sehr langsam ist. Pro Sekunde lassen sich maximal 3125 Bytes senden. Zur Übertragung von Notenwerten ist dieses Limit vollkommen ausreichend, aber bei der kontinuierliche Übertragung von Steuerdaten kann diese Übertragungsrate zu gering sein. Schon der Atari-ST, der 1985 auf den Markt kam und eine MIDI-Schnittstelle eingebaut hatte, hätte aufgrund seiner Taktfrequenz eine 256-mal so schnelle Übertragung ermöglicht. Auch sollten MIDI-Kabel nicht länger als 15 Meter sein, da sonst die Übertragungsgeschwindigkeit weiter sinkt. [KiHu99]

Trotz dieser vielen Kritikpunkte gibt es von Seiten der Musikinstrumentenindustrie keine bemerkbaren Versuche, ein neues Kommunikationsprotokoll zu entwickeln oder bestehende Protokolle wie OSC³⁹ zu einem Nachfolge-Standard zu machen.

³⁸ Pitch-Kommando: Auf einem MIDI-Keyboad gibt es neben der Klaviatur auch ein so genanntes Pitch-Rad, mit dem die Frequenz des erzeugten Klangs stufenlos zum nächsten Halbton erhöht oder erniedrigt werden kann.

³⁹ OSC: OSC: Open Sound Control: Auf dem Interprotokoll UDP basierendes Kommunikationsprotokoll, entwickelt von der Universität Berkley, das in „Pure Data“ oder „MAX“ verwendet werden kann.

Selbst innerhalb von Musikprogrammen lebt der MIDI-Standard, wenn nicht implementiert, als Metapher weiter. Das verwundert umso stärker, als die Verbindung zwischen Computer und Klangerzeugern oftmals schon über „USB“⁴⁰ oder „Firewire“⁴¹ realisiert wird.

Zusammenfassung

In diesem Kapitel wurden die Grundlagen der Benutzerschnittstelle des Programms „soundtubes“ vorgestellt. Da die Maus in diesem Programm als Eingabegerät benutzt wird, wurde die Entwicklung der Maus kurz präsentiert. Mit diesem Peripheriegerät geht eine Ausweitung des „Frame“-Konzepts einher, das sich von Gemälden bis zu Fernsehbildschirmen verfolgen lässt. Durch die Maus wurde der Bildschirm von einer Darstellungsfläche zu einer Fläche der Interaktion. Das Programm „soundtubes“ lässt sich auch als Beispiel für ein Interface sehen, das nach den Prinzipien der direkten Manipulation gestaltet wurde. Dieser Begriff wurde 1983 von Ben Shneiderman eingeführt und kennzeichnet Interfaces, die es dem Benutzer ermöglichen sollten, seine Ziele direkt umzusetzen ohne davon abgelenkt zu werden, am Computer zu arbeiten. Um das Programm „soundtubes“ einfach bedienbar zu machen, wurden Metaphern verwendet. Diese Vorgangsweise ist bei der Gestaltung von Benutzerschnittstellen sehr verbreitet. Eine Metapher soll dabei nicht als rhetorische Sprachfigur verstanden, sondern als grundlegende Denkstruktur gesehen werden, die unser Wahrnehmen und Denken formt. Daher werden Metaphern vielfach in Benutzerschnittstellen verwendet. Es wurde davon ausgegangen, dass Metaphern, die der Realität entlehnt sind, vom Benutzer besonders einfach verstanden werden. Eine Untersuchung von Blackwell und Green stellt diese Annahme in Frage. Metaphern werden auch in Musik-Software verwendet. Die am häufigsten verwendeten wurden in diesem Kapitel vorgestellt, wobei auf ihre Geschichte eingegangen wurde. Diese sind die Klaviatur, die Piano-Roll, der Sequenzer und MIDI.

⁴⁰ USB: Abkürzung für „Universal Serial Bus“. Dies ist ein serielles Bussystem zur Verbindung von Computer mit externen Geräten. In den Großteil der handelsüblichen Computer eingebaut.

⁴¹ Firewire: Eine von Apple entwickelte Schnittstelle.

Technologie

In diesem Kapitel werden die technologischen Grundlagen des Programms „soundtubes“ vorgestellt. Das Programm wurde in der Programmiersprache „vibrez“ programmiert, die auf „Pure Data“ basiert, die wiederum ihre Ursprünge in „MAX“ hat. Diese Programmiersprachen werden in diesem Kapitel vorgestellt. Da es sich bei diesen drei um visuelle Programmiersprachen handelt, wird diese Gruppe am Beginn des Kapitels behandelt.

Visuelle Programmiersprachen

Visuelle Programmiersprachen (visual programming languages – VPL) sind Programmiersprachen, bei denen mittels graphischen Elementen programmiert wird. Allgemeiner definiert benutzen diese mehr als eine Dimension für die Programmierung. Daher kann auch Text ein Element einer visuellen Programmiersprache sein. [Burn99]

Visuelle Programmiersprachen können in fünf Kategorien unterteilt werden, die unabhängig vom jeweiligen Einsatzgebiet sind.

- Tabellenbasierte VPL: Das Programmieren geschieht über das Anlegen einer Tabelle und die Spezifizierung des Inhalts. „Form/3“ ist ein Beispiel dafür.
- Programmierung durch Demonstration: Der Benutzer „zeigt“ dem Computer das gewünschte Verhalten. Das Erstellen von Makros⁴² fällt in diese Kategorie. Die beiden ersten stellen die historisch ältesten visuellen Programmiersprachen dar.
- Datenstrombasierte VPL: Dazu zählen „MAX“, „Pure Data“ und „vibrez“. Programmiert wird über die Verknüpfung von funktionstragenden Objekten mittels Kanten. Probleme treten bei komplexeren Programmen auf, da bei einer Vielzahl von Verknüpfungen der Überblick über die verbundenen Objekte verloren geht.
- Regelbasierte VPL: Diese sind dadurch gekennzeichnet, dass der Benutzer Regeln visuell definiert. Das Programm „Cocoa“ fällt in diese Kategorie. Dieses ist eine Programmiersprache für Kinder, mit dem Simulationen und

⁴² Makro: Von einem Benutzer gespeicherte Abfolge von Programmbefehlen, wobei die Abfolge der einzelnen Schritte aufgezeichnet wird.

Animationen erstellt werden können. Eine Figur folgt dabei den spezifizierten Regeln.

- Hybride VPL: Neben VPLs, die klar in eine der genannten Kategorien fallen, gibt es auch Programmiersprachen, die Mischformen darstellen. Die Programmiersprache „Rehearsal World“ ist ein Beispiel dafür. Mit ihr können Lernprogramme erstellt werden. [FiGo84]

Ein Grund für das Interesse an visuellen Programmiersprachen ist, dass textbasierte Programmiersprachen einige Nachteile besitzen. Ihr Erlernen ist schwierig, da sich der Benutzer eine völlig neue Grammatik aneignen muss, die sich von gesprochenen Sprachen stark unterscheidet und darüber hinaus auch wenig flexibel ist. Die Lernkurve dabei ist flach. Das intuitive Umsetzen von Ideen ist nur schwer möglich. Das Erfassen eines Programmcodes setzt nicht nur eine gute Kenntnis der Programmiersprache voraus, sondern erfordert auch hohe Konzentration. Der gravierendste Nachteil ist, dass beim Programmieren sehr schnell Fehler im Programmcode gemacht werden und diese oftmals nur schwer zu finden und zu korrigieren sind.

Daher werden große Hoffnungen in visuelle Programmiersprachen gesteckt. Ein Grund dafür ist, dass die menschliche Wahrnehmung über visuelle Darstellungen sehr schnell große Mengen von Daten erfassen und verarbeiten kann. [MuDa03] Visuelle Programmiersprachen sind Beispiele für direkte Manipulation. Daher werden die im Kapitel „Direkte Manipulation“ ausgeführten Hoffnungen auch in VPLs gesteckt.

Die Gründe, warum visuelle Programmiersprachen nicht häufiger verwendet werden, sind vielfältig.

Zum einen gehören sie zu den so genannten „High-Level“-Programmiersprachen. Diese versuchen es dem Benutzer so einfach wie möglich zu machen, seine Ziele umzusetzen. Der Benutzer soll sich voll auf seine Aufgaben konzentrieren können und nicht durch den Vorgang der Programmierung und durch die Bedienung des Computers abgelenkt werden. Der Computer und die Programmiersprache sollen unsichtbar, transparent sein. Dieser Ansatz wird in der Informatik daher als „Transparenz“ bezeichnet. Ein Problem an „High-Level“-Programmiersprachen ist, dass durch die Konzentration auf die Benutzung und die Ziele, die der Benutzer erreichen möchte, oft die allgemeine Anwendbarkeit verloren geht. Versucht man diesen Umstand dadurch zu umgehen, indem man die Programmiersprache um eine

Vielzahl von Lösungen für spezialisierte Anwendungsbereiche erweitert, wird diese Sprache immer schlechter benutzbar. Das Beispiel der Programmiersprache „LISP“ zeigt dies. [HuHo85]

Ein anderes Problem stellt sich durch das so genannte Deutsch-Limit. Peter Deutsch, Informatik-PhD der Universität Berkley und Entwickler des PDF-Erzeugungsprogramm GhostScript konstatierte, dass das Problem der visuellen Programmiersprachen sei, dass man nicht mehr als 50 visuelle Primitive⁴³ zur gleichen Zeit am Bildschirm haben könne. Umfangreichere Programme stoßen sehr bald an diese Grenze. [Tomi06]

Ein weiteres Problem ist, dass es visuellen Elementen und Icons oftmals an Ausdruckskraft mangelt. Wird stattdessen Text zur Bezeichnung eines Elements benutzt, verbrauchen dieses mehr Platz, was das Programm schlechter erfassbar macht.

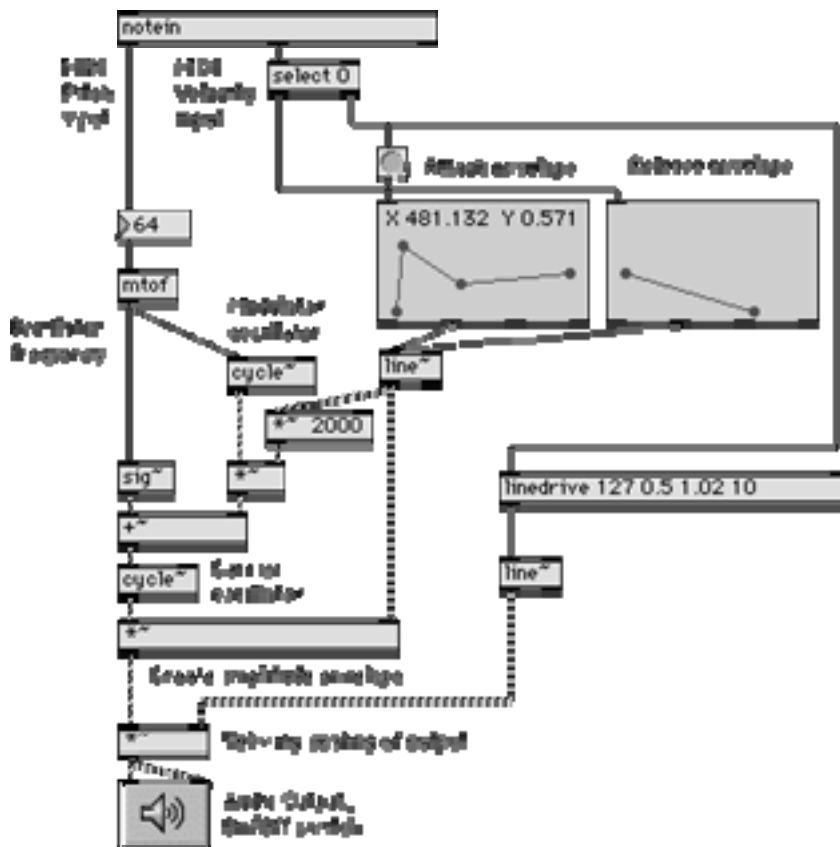
Aus diesen Gründen ist der Großteil der visuellen Programmiersprachen entweder sehr generell gehalten oder sie werden in sehr spezifischen Problembereichen eingesetzt. [MuSa03] Sie werden allgemein eher als „Spielzeug“-Anwendungen angesehen. Währenddessen sind visuelle Programmierumgebungen (visual programming environment - VPE) weit verbreitet und akzeptiert. Unter diese Kategorie fallen Programmiersprachen wie Microsoft Visual Basic. Diese stellen dem Benutzer Editoren zur Verfügung, mit denen graphische Userinterfaces erstellt werden können. Während bei VPL die Programmierung nur über visuelle Elemente geschieht, werden bei VPE die Elemente in (Text)-Code umgesetzt. Visuelle Programmierung ist dabei nur ein Hilfsmittel. [Burn99]

Max

Die Programmiersprache „MAX“ wurde 1987 von Miller Puckette am IRCAM (Institute de Recherche et Coordination Acoustique/Musique) in Paris entwickelt. Das IRCAM widmet sich der Erforschung elektro-akustischer Musik und ist dabei eine der weltweit führenden Einrichtungen. Eine federführende Person hinter der Gründung 1977 war der französische Komponist Pierre Boulez. 1981 wurde dort das digitale Klangerzeugungssystem 4X, ein Multiprozessorensystem, installiert. Miller Puckette,

⁴³ Primitiv: Einfache zwei- oder dreidimensionale graphische Grundform, die von Graphiksoftware verwendet wird. Meist sind dies Punkte, Strecken, Polygone oder Ellipsen.

Er benannte Software „MAX“ nach Max Matthew, dem Programmierer der ersten digitalen Klangerzeugungssoftware „MUSIC I“⁴⁵.



Das Programm ist ein graphisches, objektorientiertes Realtime-Modularsystem. Die einzelnen Module können durch Kanten verbunden werden. Diese Herangehensweise wurde von der Bauweise der Analogmodular-Synthesizer⁴⁶

⁴⁵ 1957 begann Max Matthew, der damals bei den Bell Labs angestellt war, ein Musikprogramm für IBM Computer zu entwickeln. Eigentlich arbeitete er daran, eine Möglichkeit zu finden, Telefongespräche zu digitalisieren, wobei er zu Versuchszwecken mit der Übertragung von Musik experimentierte. Aus diesen Forschungen entwickelte sich dann „MUSIC I“. Dieses Programm war das erste Programm das Klänge am Computer synthetisierte.

42

inspiriert, bei denen die einzelnen Module durch so genannte „Patch-Kabel“ verbunden werden. Eine Zielsetzung des Programms war, nicht nur für Informatiker bedienbar zu sein. Da das Programm als Steuerungssoftware für das 4X-System konzipiert wurde, verarbeitete „MAX“ anfangs nur Steuer- und MIDI-Befehle und erzeugte keine Audio-Signale. Größere Bekanntheit erreichte das Programm erstmals durch die Komposition „Pluton“ von Philippe Manoury. [Mill97]

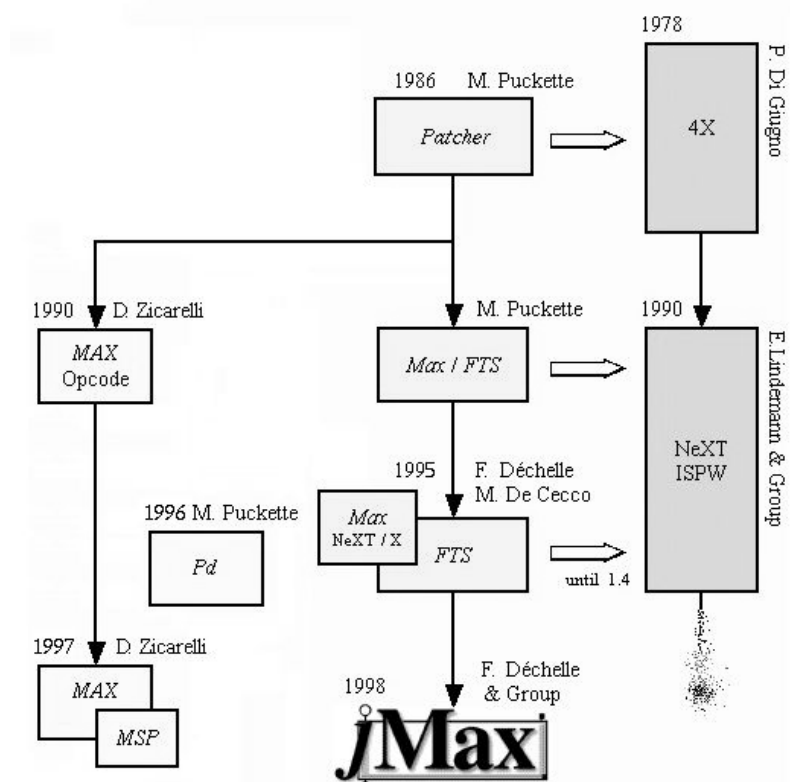


Abbildung 11: Die Entwicklungsschritte von MAX

1990 wurde „MAX“ an die Firma Opcode lizenziert und kommerziell vertrieben. Die allgemeine Tendenz der Entwicklung der Computertechnologie ging währenddessen von Multiprozessorsystemen wie dem 4X zu Uniprozessorsystemen wie PCs, die immer leistungsfähiger wurden. „MAX“, das bis dahin nur Steuerdaten verarbeiten konnte, mit Klangsynthese-Funktionalitäten zu versehen, schien ein logischer Schritt. Daher entwickelte Miller Puckette die FTS (Faster than sound)-Soundengine, die eine Echtzeit-Audioprocessing möglich machte. 1995 verließ Miller Puckette das IRCAM. Unter François Déchelle wurde das FTS-System weiterentwickelt. [Free06]

die die Verbindungen zwischen den Modulen herstellen. Das englische Wort „patch“ heißt übrigens übersetzt „flicken“.

Grossen Einfluss auf „Pure Data“ hatte die Entwicklung des „Animal“-Programm am IRCAM, bei der Miller Puckette beteiligt war. Dieses Projekt begann im August 1988 mit dem Ziel, ein System zu programmieren, das auf Echtzeitsignal- und Echtzeit-Event-Verarbeitung ausgerichtet war. Das System lief auf einem NeXT-Computer und war wie „MAX“ eine graphische, objektorientierte Programmierumgebung. [PuLi92] Ein wichtiger Unterschied ist, dass der Quellcode von „Pure Data“, im Gegensatz zu „MAX“, frei zugänglich ist. Daher bildete sich bald eine weltweite Community, die zahlreiche Erweiterungen entwickelte, wie etwa die Videoverarbeitungserweiterung GEM (Graphical Environment for Multimedia), die von Mark Danks programmiert wurde. [Mill97] Eines der Zentren der „Pure Data“-Community ist Österreich und dabei besonders das Institut für Elektronische Musik und Akustik in Graz. Auch in Wien sind einige aktive Mitglieder der Pure-Data-Szene beheimatet. Zum Beispiel entwickeln hier seit 2004 basierend auf „Pure Data“ Thomas Grill und Hannes Köcher die Multimedia-Programmierungsumgebung „vibrez“.

Vibrez

Diese Programmiersprache ist wie „MAX“ und „Pure Data“ eine graphische datenstrombasierte objektorientierte Programmiersprache, die auf „Pure Data“ basiert.

Der größte Vorteil von „vibrez“, mit der das Programm „soundtubes“ programmiert wurde, ist die Möglichkeit, das Userinterface in einem OpenGL⁴⁷-Fenster anzeigen zu lassen. In diesem Fenster lassen sich frei konfigurierbare, flexible Interfaces zusammenstellen. Mausbewegungen und Klicks können einfach von einer Applikation in „vibrez“ aufgenommen und weiterverarbeitet werden. Da „vibrez“ auf „Pure Data“ aufbaut, können alle Externals⁴⁸ und Patches⁴⁹, die für „Pure Data“ programmiert wurden, auch in „vibrez“ verwendet werden.

Die Idee zu „vibrez“ stammt von dem Programmierer Hannes Köcher, der schon einige Erfahrungen mit „MAX/MSP“ gemacht hatte. Er programmierte um das Jahr 1999 in „MAX“ einen Prototyp für ein Modularsystem mit einer graphischen

⁴⁷ OpenGL: Open Graphics Language: plattformunabhängige API (Application Programming Interface) für Computergraphikprogrammierung.

⁴⁸ externe Programmteile, die in das Programm geladen werden können

⁴⁹ Programmteile in „Pure Data“, die wie Objekten in objektorientierten Programmiersprachen eine Kapselung aufweisen.

Oberfläche, das er „Instrumentbuilder“ nannte. Dieses System war ein einfaches modulares Audio-System, das frei konfigurierbar war. Es stellte sich bald heraus, dass eine Lösung rein auf „MAX/MSP“ nicht zielführend ist. Bei einem Universitätslehrgang für elektronische Komposition an der Musikuniversität Wien lernten sich 1999 die beiden Entwickler Thomas Grill und Hannes Köcher kennen. Thomas Grill, der damals schon zehn Jahre Programmiererfahrung hatte, erfuhr von Hannes Köchers Versuchen, ein Modularsystem zu programmieren. Während diesem Lehrgang fing Thomas Grill an, Externals für „Pure Data“ zu programmieren und lernte so das Programm besser kennen. Um 2001 begannen Thomas Grill und Hannes Köcher sich mit den Spezifikationen des geplanten Modularsystems auseinanderzusetzen. Erste Prototypen wurden dann auf verfügbarer Software programmiert. „Pure Data“ wurde als Ausgangspunkt gewählt, da Thomas Grill das Programm schon gekannt hatte und für ihr geplantes Modularsystem eine Audio-Engine⁵⁰ notwendig war. Andere verwendbare Audio-Engines wären „SuperCollider“⁵¹ oder „MAX/MSP“ gewesen. „SuperCollider“ wurde sehr bald ausgeschlossen, da beide Programmierer kaum Erfahrungen mit dem Programm hatten und die Bedienung des Programms nicht ihrer Arbeitsweise entsprach. Gegen die Verwendung von „MAX“ entschied man sich, da es ein kommerzielles Produkt ist, für das Lizenzgebühren zu zahlen wären. Ein anderer Grund für die Wahl von „Pure Data“ war, dass für dieses Programm bereits viele Externals programmiert worden sind und wie oben ausgeführt auch in „vibrez“ genutzt werden konnten. Da darüber hinaus „Pure Data“ Open-Source ist, können Fehler in der Software selbst behoben werden, was bei „MAX“ nicht möglich gewesen wäre.

Die Motivation „vibrez“ zu entwickeln war, dass beide Programmierer bemerkt hatten, dass es dem Spielen mit Softwareinstrumenten an Feedback mangelte. Viele Programme sind unintuitiv und unergonomisch gestaltet. Man kann sie nicht sofort handhaben. Entwicklersysteme wie „MAX“ und „Pure Data“ sind Beispiele dafür. Andere Programme, die intuitiver gestaltet sind, wie zum Beispiel „Ableton live“ oder „Reason“, schränken den Benutzer in der musikalischen Gestaltungsmöglichkeit ein. Ihre größte Motivation war ein System für den Eigengebrauch zu entwickeln. Zum Zeitpunkt des Interviews waren Thomas Grill, Hannes Köcher und Tim Blechmann am Projekt „vibrez“ beteiligt. Tim Blechmann arbeitete dabei an kernel-

⁵⁰ Audio-Engine: Klangsynthese-Programm

⁵¹ SuperCollider: Echtzeitklangsynthese-Programmiersprache

spezifischen Teilen. Thomas Grill programmierte hauptsächlich am Kern des Programms, während Hannes Köcher sich mit der Usability beschäftigte. Das Projekt wurde zwei Jahre lang durch die Förderung „creative industries“ unterstützt, die von der Stadt Wien initiiert wurde.

Das Ziel der Entwicklung ist eine Erstversion eines Consumerprodukts eines Modularsystems. Das soll aber nicht das einzige Produkt bleiben. Ihr Ziel ist es, eine frei konfigurierbare Entwicklungsumgebung für Multimedia zu programmieren mit der der Benutzer einfach kombinierbare Oberflächen gestalten kann. Das Programm soll als Tool für Performances oder Multimedia-Installationen genutzt werden können. Es soll auch die Vernetzungen mit anderen Rechnern unterstützt werden. Wichtig ist ihnen weiters, dass Video und Audio ohne Probleme zusammenarbeiten können. Zum Zeitpunkt des Interviews war geplant, im Frühjahr 2006 das Erstprodukt zu veröffentlichen. [Gril05]

Zusammenfassung

In diesem Kapitel wurden die technologischen Hintergründe zum Programm „soundtubes“ beleuchtet. Dieses Programm wurde mit „vibrez“, einer datenstrombasierten objektorientierten visuellen Programmiersprache realisiert. Diese baut auf die visuelle Programmiersprache „Pure Data“ auf, die vom Miller Puckette entwickelt wurde. Dieser entwickelte zuvor am IRCAM in Paris die visuelle Programmiersprache „MAX“, die als Vorgänger gesehen werden kann. „vibrez“ wurde von beiden Wienern Programmierer Thomas Grill und Hannes Köcher entwickelt. „soundtubes“ wurde auf einer Vorversion implementiert. Das Ziel der Endversion ist es, dem Benutzer ein einfach konfigurierbares modulares System für Multimedia-Anwendungen und Installationen zur Verfügung zu stellen. Da „soundtubes“ mit einer visuellen Programmiersprache umgesetzt wurde, wurden in diesem Kapitel diese Kategorie der Programmiersprachen vorgestellt. Diese werden selten benutzt und im Allgemeinen eher als „Spielzeugprogramme“ angesehen. Ein Grund dafür ist, dass visuelle Programmiersprachen High-Level-Programmiersprachen sind, die versuchen, die Zielsetzungen des Benutzers so einfach wie möglich umsetzbar zu machen. Dabei geht aber die generelle Anwendbarkeit verloren. Auch das Deutsch-Limit, das besagt, dass nicht mehr als 50 Primitive am Bildschirm sein könnten, stellt ein Problem visueller Programmiersprachen dar.

Soundtubes

„soundtubes“ ist der Titel des Musikprogramms für Kinder, das im Zuge der Diplomarbeit entwickelt wurde. In diesem Kapitel wird das Programm ausführlich vorgestellt. Das Programm wurde mit Unterbrechungen im Zeitraum zwischen Februar 2005 und Dezember 2006 entwickelt. Durch diese lange Entwicklungszeit konnten verschiedene Variationen der Grundidee ausgetestet werden. Dabei gab es keine getrennte Konzeptions- und Implementierungsphase, was zur Folge hatte, dass technische Einschränkungen das Konzept beeinflussten. Es konnten aber auch konzeptionelle Überlegungen sofort ausprobiert werden, was verhinderte, dass unumsetzbare Ideen verfolgt wurden.

Zielsetzung

Bei der Konzeption des Programms „soundtubes“ wurde versucht, folgende Zielsetzungen zu erreichen.

Das Programm sollte Laien und Kinder schnell und einfach dazu befähigen, musikalisch allein am Computer tätig zu werden. Es war kein Ziel ein Instrument oder ein audiovisuelles Musikprogramm zu programmieren. Das Programm sollte ein loop-basierter Sequenzer werden, der aber von den Konventionen des sechzehnstufigen Loops abweicht.

Die Verwendung von Metaphern sollte dem Benutzer den Einstieg und Umgang mit dem Programm so einfach wie möglich gestalten. Diese Metaphern sollten sich aber von den in anderen Musikprogrammen verwendeten wie der Klaviatur oder der Piano-Roll unterscheiden. In der Gestaltung des Programms wurden absichtlich Einschränkungen in Kauf genommen, die sich auf den Funktionsumfang auswirken. Ein weiterer wichtiger Punkt war es zu versuchen, in der Klanggestaltung einen anderen Zugang zur Musikproduktion aufzuzeigen.

Das Programm „soundtubes“

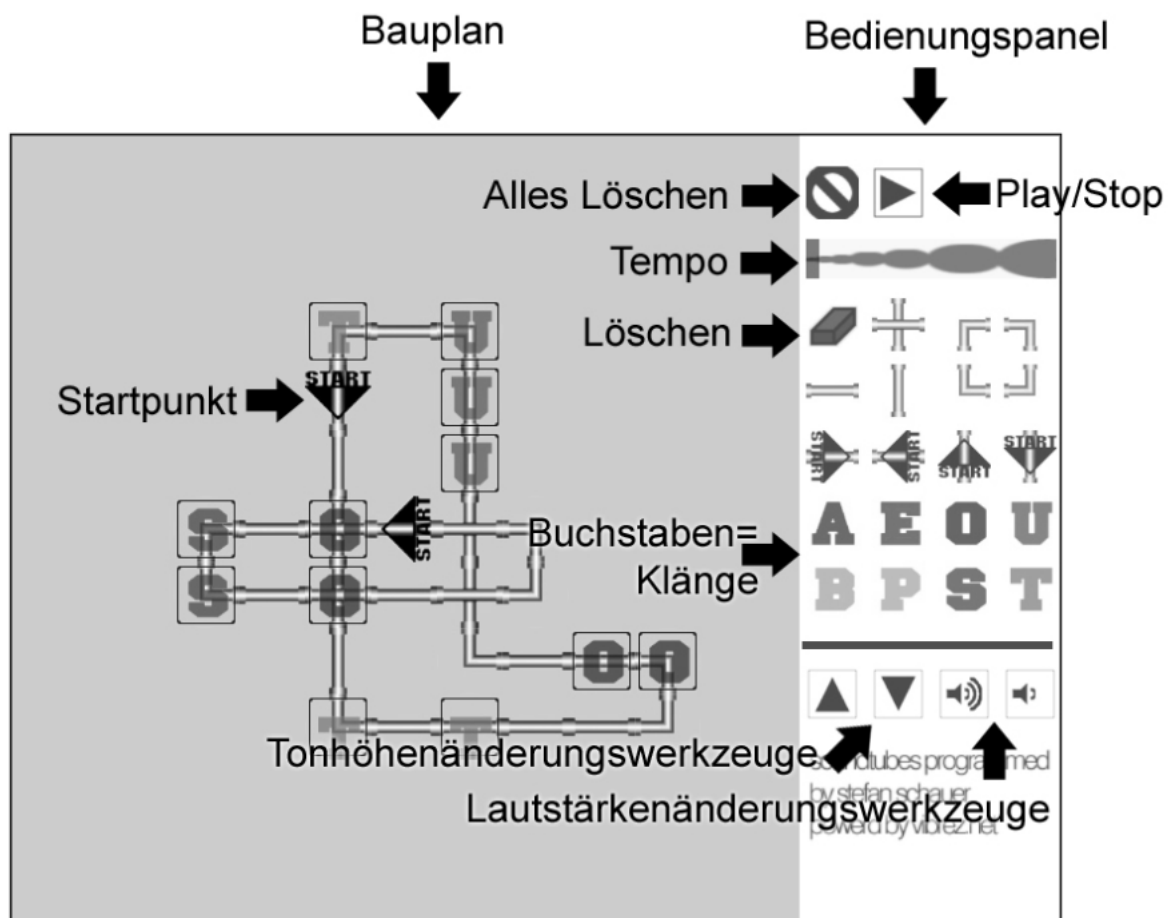


Abbildung 13: Die Bedienelemente des Programms "soundtubes"

Das Programm „soundtubes“ wurde mit der Zielsetzung konzipiert, möglichst einfach bedienbar zu sein. Daher wird „soundtubes“ nur mit der Maus und mit Einfachklicks der linken Maustaste bedient. Das Interface ist in einen Bauplan und ein Bedienungspanel aufgeteilt. In letzterem finden sich alle Kontrollwerkzeuge und Bauelemente wie die Rohrleitungen oder Buchstaben. Die Buchstaben sind der Träger des Klangs des jeweiligen gesprochenen Buchstaben. Mit einem Klick auf eine Rohrleitung oder auf einen Buchstaben wird dieses Element ausgewählt und wird mit einem Klick auf eine Stelle am Bauplan abgelegt. Bei einem Klick auf eine Rohrleitung im Bedienungspanel wird ein Signalton, bei einem Klick auf einen Buchstaben der Klang des Buchstabens abgespielt. Der Benutzer weiß, welches der Elemente gewählt wurde, da im Bedienungspanel angewählte Buchstaben größer und angewählte Rohrleitungen dazu noch dunkler dargestellt werden. Auch wird das Element leicht durchscheinend an der Cursorposition angezeigt, sobald sich der Cursor am Bauplan befindet. Um eine vollständige Rohrleitung zu bauen, muss ein

Element dieser Leitung ein Startpunkt sein. Es können auch mehrere Startpunkte Elemente einer zusammenhängenden Leitung sein. Die Buchstaben können wie die Rohrleitungen auf dem Bauplan abgelegt werden. Über die Tonhöhen- und Lautstärkenwerkzeuge kann der Benutzer die Tonhöhe und die Lautstärke des Buchstabens verändern. Bei einer Tonhöhenänderung wird dieser nach oben oder nach unten verschoben. Bei einer Lautstärkenänderung wird der Buchstabe größer oder kleiner dargestellt. Drückt der Benutzer auf den „Play“-Button, beginnen von den Startpunkten aus Verdickungen durch die Rohrleitungen zu laufen. Treffen diese auf einen Buchstaben, wird er, abhängig von der eingestellten Tonhöhe und Lautstärke, abgespielt. Um dem Benutzer den Zusammenhang zwischen hörbaren Output und dem Bauplan zu verdeutlichen, wird der Buchstabe mit roten konzentrischen Kreisen herausgehoben. Es können gleichzeitig 18 solcher Verdickungen durch gebaute Rohrleitungen laufen. Trifft eine solche auf einen Startpunkt, der der Laufrichtung entgegensteht, so ändert sie diese dort. Der Name des Programms kommt daher, dass sich die Rohrleitungen als klangliche Röhren interpretieren lassen. Ins Englische übersetzt und abgewandelt kommt man zur Wortneuschöpfung „soundtubes“.

Inspirationen

In diesem Kapitel werden Programme vorgestellt, die ähnliche Ansätze verfolgen. Manche der vorgestellten Programme waren Inspirationsquellen, andere zeigen Parallelen zum Programm „soundtubes“. Neben Musikprogrammen werden auch zwei Computerspiele vorgestellt.

Nanoloop

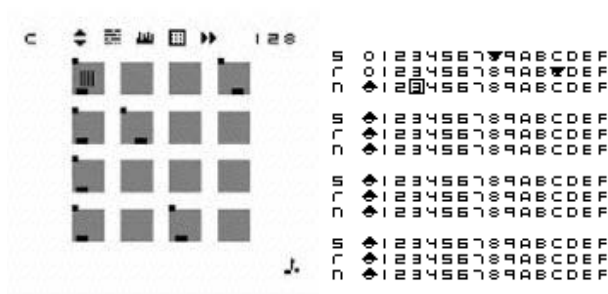


Abbildung 14: Screenshots des Programms „nanoloop“

Das Programm „nanoloop“ ist ein Musikprogramm für den Nintendo Gameboy. Es wurde vom Hamburger Oliver Wittchow entwickelt. „nanoloop“ war die Abschlussarbeit für seine Post-Graduate-Ausbildung an der freien Akademie der

Künste in Hamburg. Im Mai 2005 war Oliver Wittchow drei Wochen lang in Wien um von Kindern eines Hortes seine Neuentwicklung „fee“⁵² testen zu lassen.

„nanoloop“ ist ein Sequenzer, dessen Aussehen und Bedienung sich stark vom üblichen Software-Sequenzer unterscheidet. Das Interface ist äußerst reduziert und konzentriert sich auf das Wesentliche. Die sonst in Nintendo-Spielen üblichen Behübschungen fehlen völlig. Die Entscheidung, sich von gewöhnlichen Metaphern zu verabschieden, hing auch mit der Unzufriedenheit mit vorhandenen Hardware- und Software-Sequenzern zusammen. Deren Benutzung schien Oliver Wittchow zu kompliziert. Viele implementierte Funktionen waren ihm zu stark an musikalische Vorstellungen gebunden, die ihm uninteressant schienen.

Im Programm „nanoloop“ ist der Loop auf sechzehn Stufen, so genannte „Steps“, reduziert, die in einer vier mal vier großen Matrix angeordnet sind. Die Parameter der gesetzten Töne sind durch Striche visualisiert, die durch Bewegung mit dem Cursor verändert werden. Auch die Tonhöhe ist auf diese Weise dargestellt. Dieser wird so kein größerer Stellenwert als anderen musikalischen Parametern zugewiesen. Diese eher ungewöhnliche Entscheidung hat ihren Ursprung aus dem Wunsch Oliver Wittchows, das Programm radikal modern zu gestalten und auf traditionelle Elemente so weit wie möglich zu verzichten. Ein weiterer Faktor, der diese Entscheidung beeinflusste, war, dass „nanoloop“ für die Produktion von „Minimal Techno“-Stücken gedacht war. In diesem musikalischen Genre sind Melodien von geringer Bedeutung. Ursprünglich war auch keine Möglichkeit zur Speicherung der Stücke vorgesehen. Die Idee war, ein reines „Plug-and-Play“-Instrument zu schaffen.

Ein weiterer Grund für das neuartige Interface liegt darin, dass Oliver Wittchow nicht aus der Demo⁵³- oder Chipmusic-Szene⁵⁴ stammt, die sich schon länger mit der Programmierung von Musikprogrammen beschäftigt, die auf alten Computern und

⁵² „fee“: Von Oliver Wittchow entwickeltes Musikprogramm für vier Mitspieler, das eigentlich für die Sony Playstation gedacht war. Die vier Benutzer arbeiten mit Gamepads gleichberechtigt an einem sechzehnstufigen Sequenzer.

⁵³ Demo: Eine Demo bezeichnet hier musikunterlegte Echtzeitanimationen, die für verschiedene Spielkonsolen und Computer programmiert werden. Dabei wird versucht, die Ressourcen der Geräte voll auszunutzen und möglichst beeindruckte „Demos“ zu programmieren. Auf so genannten „Demo-Partys“ treten die verschiedenen Demo-Gruppen mit ihrem „Demos“ gegeneinander an.

⁵⁴ Chiptunes: Auch 8-Bit-Musik genannt. Musik, die mit Hilfe alter Spielkonsolen produziert wird. Beliebte sind dabei besonders der Nintendo Gameboy und der Commodore 64. Rund um diese Musik hat sich eine kleine internationale Szene gebildet.

Spielkonsolen laufen. Er erlernte das Programmieren erst im Laufe der Entwicklung von „nanoloop“. Die Idee zum Programm selbst kam von Sebastian Burdach, einem Freund von Oliver Wittchow. Dieser programmierte für den Nintendo Gameboy einen simplen Sequenzer mit einem anderen Userinterface, verlor dann aber das Interesse an einer Weiterentwicklung. Oliver Wittchow begann sich dafür zu interessieren und programmierte für sich selbst eine erste Version. 1998 präsentierte er eine erste „nanoloop“-Version bei einem Wettbewerb für „lo-fi“-Instrumente⁵⁵ in Köln. Die positive Resonanz des Publikums machte ihm klar, dass er das Programm „nanoloop“ weiterentwickeln sollte. „nanoloop“ 1.0 für den Nintendo gameboy classic wurde 2000 veröffentlicht. Diese Version erlaubte dem User mit einer Noise-, einer Vier-Bit-Wavetable- und einer Pulse-Spur Musik zu komponieren. 2004 kam dann „nanoloop“ 2.0 für den Nintendo gameboy advance auf den Markt, das acht Spuren zur Verfügung stellt. Diese Spuren waren in ihrem Klang weitgehend frei zuweisbar. 2005 folgte dann „nanoloop“ 1.3 für den Nintendo gameboy classic mit zwei Pulse-, einer Vier-Bit-Wavetable- und einer Noise-Spur. [Witt05] [Witt06]

Gbloink!

Gbloink! ist ein experimentelles Musikprogramm von Phil Jones, das er selbst als „sound toy“ bezeichnet. Der Entwickler ist ein britischer Programmierer, der zurzeit in Brasilien arbeitet. Er besitzt den Bachelor-Titel des Londoner Queen Mary Colleges in Computersystemen und Mikroelektronik und den Titel Master of Philosophy in künstlicher Intelligenz von der Sussex Universität in England.

Das Programm wurde von dem Spiel „Mousetrap“⁵⁶ inspiriert, bei dem der Spieler einen den Reflexionsgesetzen gehorchenden herumspringenden Ball durch das Ziehen von Linien in eine Falle leiten musste. Jahre später integrierte die Mutter von Phil Jones dieses Spielprinzip in eine Lernsoftware, die auf dem Buch „The phantom tollbooth“⁵⁷ basierte. Dabei musste der Spieler die entflohenen Klänge des Doktor Dischords einfangen. Nach der Beschäftigung mit elektronischer und experimenteller

⁵⁵ Lo-fi: Low Fidelity. Gegenteil von High Fidelity. Instrumente, die nicht auf eine lupenreine Klangwiedergabe konzentriert sind.

⁵⁶ Mousetrap ist auch der Name eines dem Spielprinzip „Pac Man“ sehr ähnlichen Arcade-Spiels, das 1981 von der Spielefirma „Exidy“ veröffentlicht wurde.

⁵⁷ Das Kinderbuch „the phantom tollbooth“ wurde 1961 von Norton Juster geschrieben und handelt von einem Jungen, der durch eine magische Zollstation in das Königreich der Weisheit gelangt. Auf der Reise durch dieses Land gelangt er auch in das Tal der Klänge, in dem Doktor Dischord wohnt.

klassischer Musik entstand um 1987 die Idee eines Musikprogramms, bei dem die Musik dadurch entstand, dass ein Ball durch ein Labyrinth hüpfte. Dieses nicht realisierte Programm wurde mit dem Arbeitstitel „musical maze“ versehen. Erst 1997 verwirklichte er diese Idee in der vorliegenden Form unter dem Titel „Gbloink!“.

[Jone07]

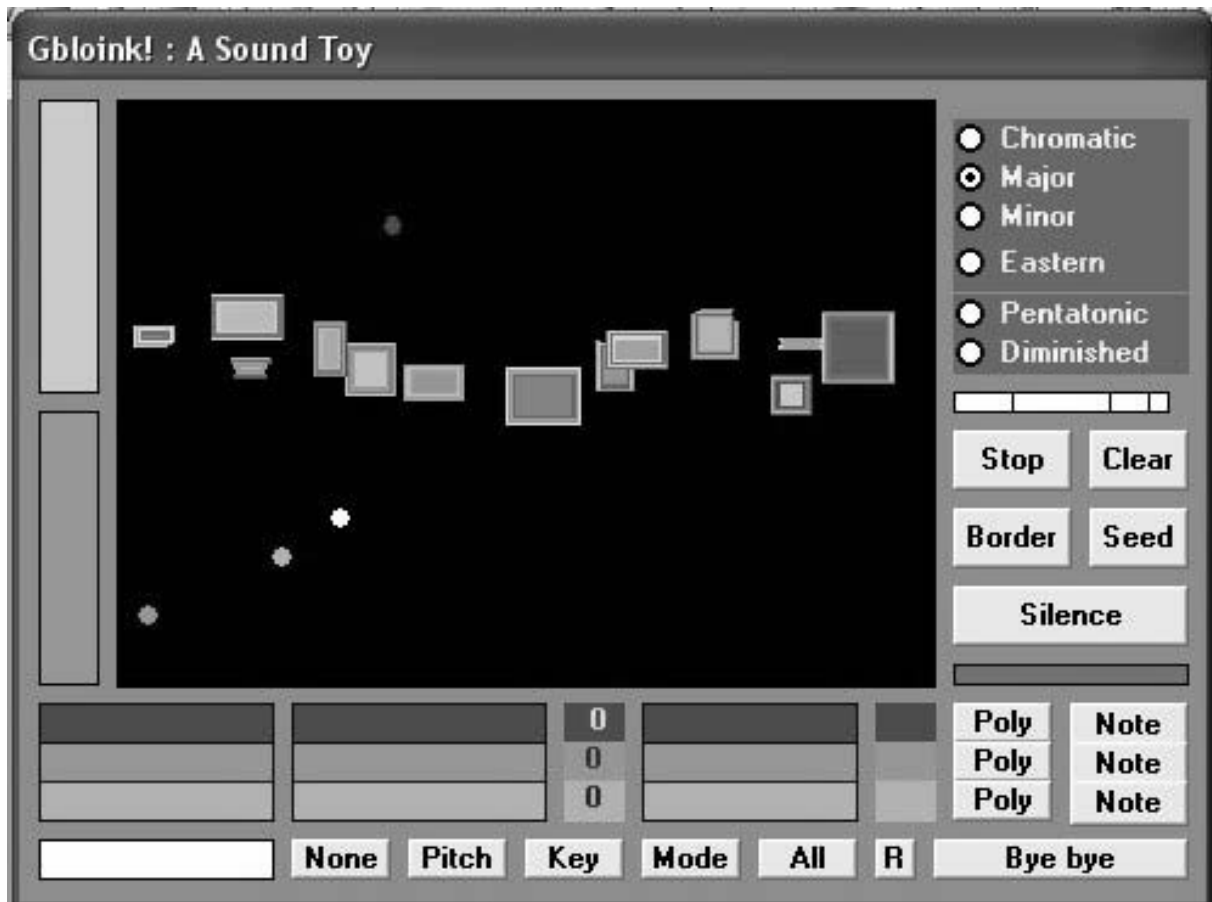


Abbildung 15: Screenshot des Programms „Gbloink!“

Das Programm ist sehr einfach. Per Mausklick platziert der Benutzer rechteckige Hindernisse in ein Feld. Durch das Drücken des Knopfes „Start“ setzt man vier Bälle in diesem Feld frei, die nach den Reflexionsgesetzen von den Hindernissen und Begrenzungen des Felds abprallen. Wenn dies passiert, wird ein Ton, dessen Tonhöhe von der vertikalen Position des Balles abhängt, ausgelöst. Der vierte Ball verändert verschiedene Parameter der Bälle, auf die er trifft.

Das Interessante an diesem Programm ist, dass eine Art steuerbarer Zufall die Musik erzeugt. Anders als bei den elektro-akustischen Komponisten der Anfangszeit, die entweder den Zufall oder eine mathematisch deterministische Funktion als grundlegendes Prinzip ihrer Komposition festlegten, wird hier eine attraktive Mischung der beiden Möglichkeiten benutzt. [Rodw99] [Gblo05]

In diesem Programm wird das physikalische Prinzip der Reflexionsgesetze verwendet, ebenso wie im Programm „Sounder“ von Jack Freudenheim. [Gola94] Der Benutzer hat nur indirekt Einfluss auf den musikalischen Output. Anders als bei anderen Musikprogrammen ist das Ziel des Programms nicht, dem Benutzer zielführendes Komponieren zu ermöglichen. Das Spielerische wird stärker in den Vordergrund gestellt.

Phil Jones sieht in „Gbloink!“ eine andere Art der Interaktion mit dem Computer. Der Großteil der Programme funktioniert nach dem Prinzip, dass der Benutzer dem Computer Anweisungen gibt. Bei „Gloink!“ besitzt der Benutzer nicht die volle Kontrolle über das Programm, aber er kann versuchen, den Output in die gewünschte Richtung zu steuern. Anwender, die keine genaue Vorstellung von dem erwünschten Output haben, kommt diese Art der Interaktion sehr entgegen. [Jone07] Die Klangerzeugung geschieht über MIDI-Instrumente, die keine hohe klangliche Qualität aufweisen.

Drumsteps

Das Programm „Drumsteps“ wurde im Rahmen der PhD-Arbeit von Kevin Jennings programmiert. Der Programmierer ist sowohl ausgebildeter Musiker als auch Bachelor der Physik. Das Studium „Music Performance“ schloss er mit dem Bachelor-Titel am Trinity College in Dublin und mit dem Master-Titel an der National University of Ireland ab. Als Gitarrist trat er in ganz Europa auf. Weiters hat er schon mehr als 15 Jahre Erfahrung in Musikerziehung an verschiedensten Schulen in Irland und den USA. Er war auch beteiligt am „ToySymphony“-Projekt der Hyper-Instrument-Group des MIT. Im Moment ist er Research Fellow am CRITE (Centre for Research in I.T. in Education am Trinity College Dublin und MediaLab Europe). [Crit06]

„Drumsteps“ ist als Programm für den Musikunterricht gedacht. Kindern soll dabei Rhythmik näher gebracht werden. Jennings hält sich dabei an konstruktivistische Lerntheorien nach Seymour Papert. Dieser sieht als Problem, dass Kinder bei Lernsoftware oft mit Konzepten außerhalb ihrer eigenen Lebenswelt konfrontiert werden. Der gewünschte Lerneffekt tritt daher nicht ein. Als Lösung für dieses Problem schlägt er die Konstruktion von so genannten „Microworlds“ vor. Das sind

reale oder virtuelle immersive⁵⁸ Umgebungen, in denen Kinder die Regeln dieser Mikrowelt über das Ausprobieren erleben. [Jenn02]

„Drumsteps“ ist ein Schlagzeug-Sequenzler, der nach dem Prinzip der Gravitation funktioniert. Bälle sind die Träger der Schlagzeug-Klänge. Die Klänge können den Bällen frei zu gewiesen werden. Treffen sie auf Stufen oder Treppen, wird der Klang ausgelöst. Durch Wumlöcher können Bälle zu einem anderen Punkt transportiert werden. So können einfache Loops programmiert werden. Komplexere Strukturen können über Falltüren programmiert werden, die durch so genannte Trigger-Steps geöffnet werden. [Drum05] [JeTa02]

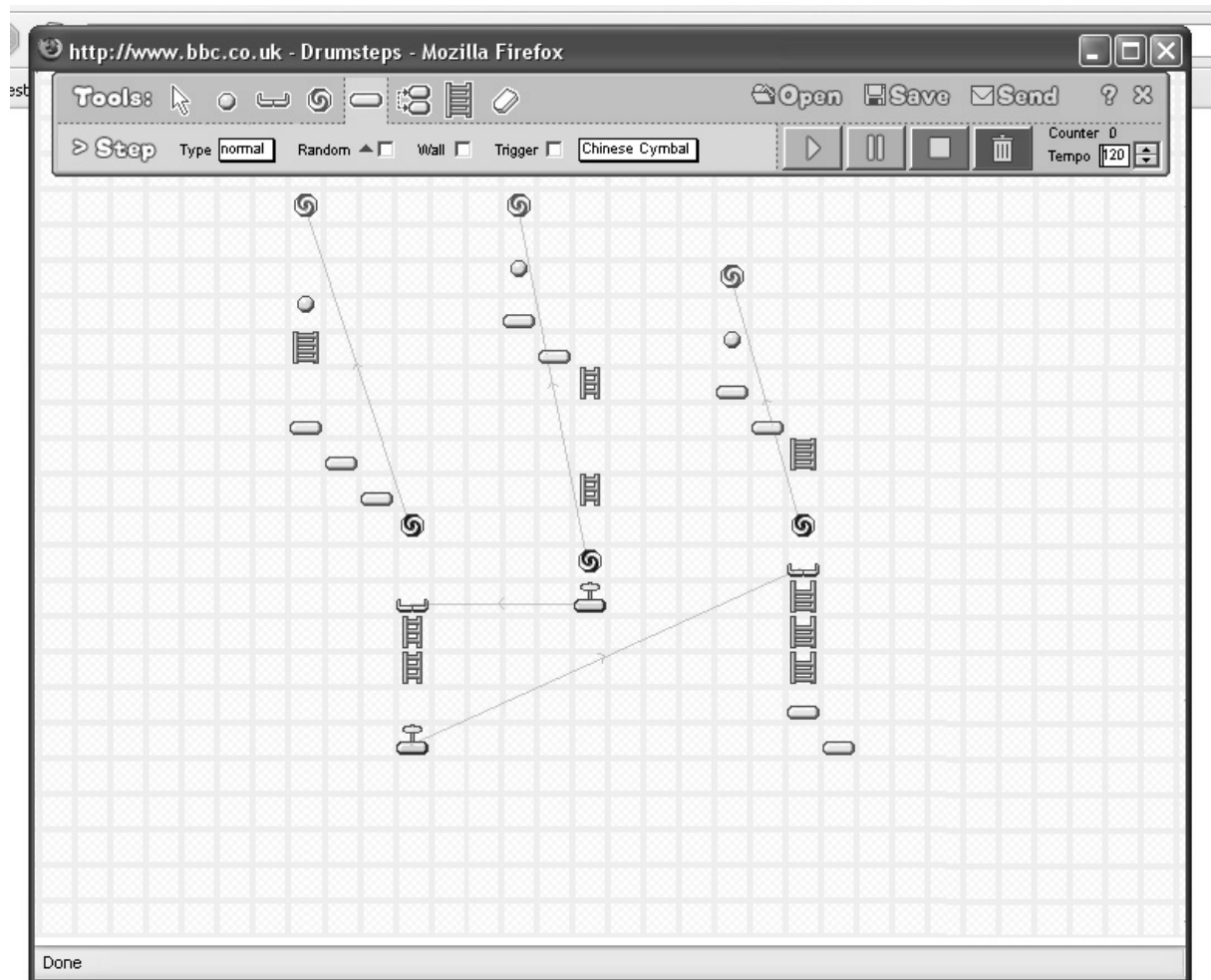


Abbildung 16: Screenshot des Programms „Drumsteps“

Als Inspirationsquelle dienten Kevin Jennings verschiedene Spiele, die Navigation im Raum notwendig machen. Unter anderem nannte er „the lemmings“, „TIM (the incredible machine)“, „Kong“ und „Supermario“. Besonders das Spiel „TIM“ ähnelt

⁵⁸ Immersiv: Realität, die den Akteur dazu bewegt, sie als echt zu sehen und dementsprechend zu handeln.

„Drumsteps“ sehr stark. Bei diesem Spiel geht es darum, in einer Mikrowelt gewisse Aufgaben (z.B. „Verjage die Maus“) mit der Verwendung verschiedener technischer Hilfsmittel (z.B. Bälle, Förderbänder, etc.) zu lösen.

„Drumsteps“ basiert wie auch „Gbloink!“ auf einem physikalischen Prinzip, das jedem bekannt ist. Neben der einfachen Programmierung von Schlagzeugspuren ist es auch ein leicht verständliches Notationssystem. [Jenn05]

Bei der Klangerzeugung greift das Programm auf MIDI-Schlagzeugklänge zurück. [JeTa02]

SimTunes

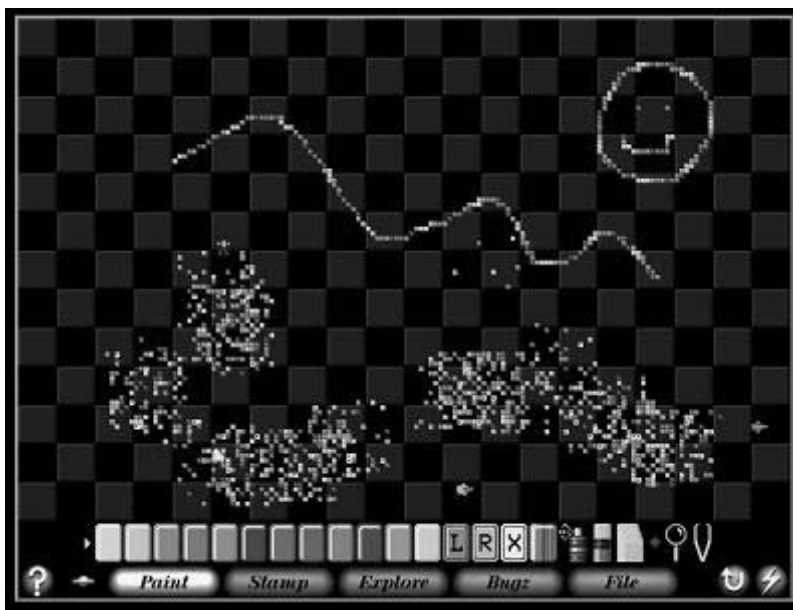


Abbildung 17: Screenshot des Programms „SimTunes“

Maxis⁵⁹ veröffentlichte 1996 das Spiel „SimTunes“. Dieses Spiel wurde vom Medienkünstler Toshio Iwai entwickelt, der im Bereich von computergenerierter Musik und Interfaces tätig ist. 2005 präsentierte er bei der „Ars Electronica“ einen neuartigen Sequenzer namens „Tenori-on“, den er für die japanische Firma Yamaha entwickelt hatte. 2006 wurde er bei der „Ars Electronica“ in Linz als „Featured Artist“ eingeladen. Im Moment arbeitet er als „Artist-in-Residence“ am IAMAS (International Academy of Media Arts and Sciences) in Ogaki in Japan.

⁵⁹ Maxis: Amerikanische Spielesoftware-Firma, die 1987 gegründet wurde. Neben SimTunes veröffentlichten sie unter anderem SimCity, SimEarth und Die Sims.

Das Programm „Music Insects“, das als Vorläufer von „SimTunes“ gesehen werden kann, wurde von ihm während eines „Artist-in-Residence“-Aufenthalts 1992 am „Exploratorium“ in San Francisco entwickelt. [Iwai06]

Das Spielprinzip von „SimTunes“ ist einfach. Vier Insekten, so genannte Bugs, laufen selbsttätig in einer Richtung über ein Spielfeld. Das Spielfeld kann mit Farbquadraten belegt werden. Läuft ein Bug über ein solches Feld, wird ein Klang abgespielt, dessen Tonhöhe von der Farbe abhängt. Die Klangfarbe hängt von dem Instrument ab, das man dem Bug zugewiesen hat. Die Richtung, in die die Bugs laufen, kann durch besondere Feldelemente verändert werden, die die Bugs zu einer Drehung um 90 oder 180 Grad zwingen. Mit einem Pinzettenwerkzeug kann der Benutzer die Bugs nehmen und auf einen anderen Platz versetzen.

Um dem Benutzer einen schnellen Einstieg zu ermöglichen, müssen bei der Erstbenutzung die wichtigsten Funktionen unter Anleitung des Programms angewandt werden.

„SimTunes“ zeigt einen sehr spielerischen Zugang zum Musikproduzieren. Planvolles Musizieren ist zwar prinzipiell möglich, wird aber nicht sehr gut unterstützt, da zum Beispiel die Tonhöhe als Farbwert kodiert ist, was ein Komponieren von Melodien erschwert. Wie auch bei „Gbloink!“ wurde kein großer Wert auf die Klangfarbe gelegt und es werden ebenfalls MIDI-Instrumente verwendet.

Golan Levin sieht im Programm „SimTunes“ ein audiovisuelles System, welches das gleichzeitige Bearbeiten von Bild und Ton erlaubt. Das Ergebnis der Beschäftigung mit SimTunes kann sowohl ein Bild als auch eine Melodie sein⁶⁰. [Gola94]

Elektroplankton

Das Spiel „Elektroplankton“ für den Nintendo Gameboy Double Screen stellt eine Zusammenfassung von zehn Musikspielen dar. Das Besondere an dieser tragbaren Spielkonsole ist, dass es neben einem „normalen“ Bildschirm einen Touchscreen besitzt. Das Programm wurde ebenfalls von Toshio Iwai entwickelt. Das Spiel wurde

⁶⁰ Wolfgang Kopper, ein Gründer des Gameboymusicclubs, sammelte während seiner Arbeit in einem Kinderhort in Wien Ottaking, Erfahrungen mit SimTunes. Er versuchte dort Kinder zur musikalischen Arbeit am Computer zu motivieren. SimTunes wurde von den Kindern sehr gut angenommen. Ein Kind arbeitete mit SimTunes in der Form, dass es Noten auf das Spielfeld malte. Diese gemalten Notenzeichen wurden durch das Programm wiederum in Klänge umgewandelt. Das Endergebnis war somit eine Mischung zwischen Bild und Ton.

2005 in Japan veröffentlicht, wo das Genre der Musikspiele mehr Tradition als in Europa hat. Im Juli 2006 wurde „Elektroplankton“ auch in Europa veröffentlicht. Eines der Spiele namens „Luminaria“ zeigt starke Parallelen zu „soundtubes“. Dabei sind vier Bälle, genannt Planktons, in einem Vektorfeld an den Ecken des Spielfelds platziert. Über den Touchscreen kann das Vektorfeld so ausgerichtet werden, dass sich kleine Loops bilden, in denen die Planktons zirkulieren. Bei jeder Bewegung des Planktons wird ein Klang ausgelöst. Ein ähnliches Prinzip ist auch im Programm „Vector Field“ von Lukas Girling zu finden, das am Interval Research Center 1997 entwickelt wurde. Kennzeichnend ist, dass der Benutzer nicht direkt den musikalischen Output steuert, sondern die Umgebung verändert. [Gola94] Neben diesem Spiel gibt es noch neun andere zur Auswahl. Jedes einzelne ermöglicht dem Benutzer schnell musikalisch tätig zu werden. Da es sich nur um Musikspiele handelt, sind alle Ergebnisse flüchtig und nicht speicherbar.

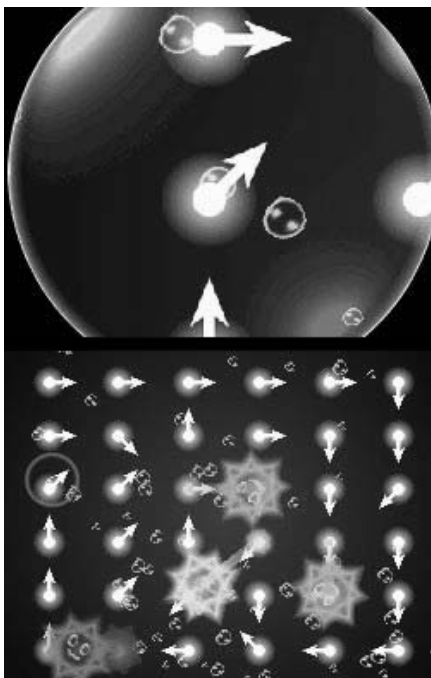


Abbildung 18: Screenshot des Programms „elektroplankton“

Elektroplankton ist somit näher beim Instrument als beim Sequenzer. Vom Benutzer wird ständige Aktivität gefordert. Minimale Interaktion mit dem Spiel führt schon zu hörbaren Ergebnissen. Neben dem exzellenten Sounddesign ist auch die graphische Umsetzung sehr gut gelungen. Die Möglichkeiten, die der Touchscreen bietet, wurden dabei sehr gut ausgenutzt. Wie im letzten Kapitel ausgeführt, meinte Oliver Wittchov, der Programmierer von „nanoloop“, bei einer Unterhaltung, dass

„Elektroplankton“ seinen Vorstellungen von einem idealen Musikprogramm für Kinder sehr nahe kommt.

mu-framework

Leonard Ritter, in der Demo-Szene unter dem Namen „Paniq“ als Mitglied der deutschen Gruppe „Farbrausch“ bekannt, plante unter dem Titel „mu-framework“ ein leicht bedienbares Musikprogramm, das folgende Kriterien erfüllen sollte.

1. user interface paradigm: "a powerful toy"
2. should be as comprehensible as possible
3. should be intuitive (and i really mean: intuitive)
4. target audience: kids, professional musicians
5. all objects seen in the application should represent something musical
6. no text, use animations and little movies to explain certain tasks
7. synaesthetical user interface approach
8. no classic music program approaches: no sequencer, no piano roll, no tracker
9. should be useful for live performances from scratch
10. should be useable on drugs (should be fun working with even if your perception is impaired or altered)
11. supports heavy multiuser interaction

Das Projekt wurde nicht weiterverfolgt. Die angeführten Kriterien bildeten jedoch für die Konzeptionierung des Programms „soundtubes“ eine geeignete Grundlage.

[Mufr05]

Stunts

Bei „Stunts“ handelt es sich um ein 1990 von der amerikanischen Softwarefirma „Brøderbund“⁶¹ für den PC veröffentlichtes Autorennspiel. Neben der Möglichkeit, Rennen zu fahren, war das Spiel mit einem Streckeneditor ausgestattet, mit dem eigene Strecken gebaut werden konnten. Im Kasten rechts wählt der Benutzer per Mausklick die Straßenelemente aus, die er am Bauplan links per Mausklick absetzt. Hat der Benutzer ein Element angewählt, so wird dieses an der Cursorposition blinkend angezeigt. Der Bauplan selbst ist in quadratische Flächen aufgeteilt. Der Cursor wird dabei immer am nächstliegenden Quadrat angezeigt. Nicht zufällig

⁶¹ Die Firma hat neben „Stunts“ auch die Spiele „Prince of Persia“ und „Myst“ veröffentlicht. Das letztere stellte bis zur Veröffentlichung des Spiels „Die Sims“ das meist verkaufte Computerspiel dar.

sehen erste Versionen des Programms „soundtubes“ dem Streckeneditor des Spiels „Stunts“ sehr ähnlich. Auch die Endversion lässt noch diese Inspirationsquelle erahnen.



Abbildung 19: Screenshot des Streckeneditors des Spiel "Stunts"

Pipemania

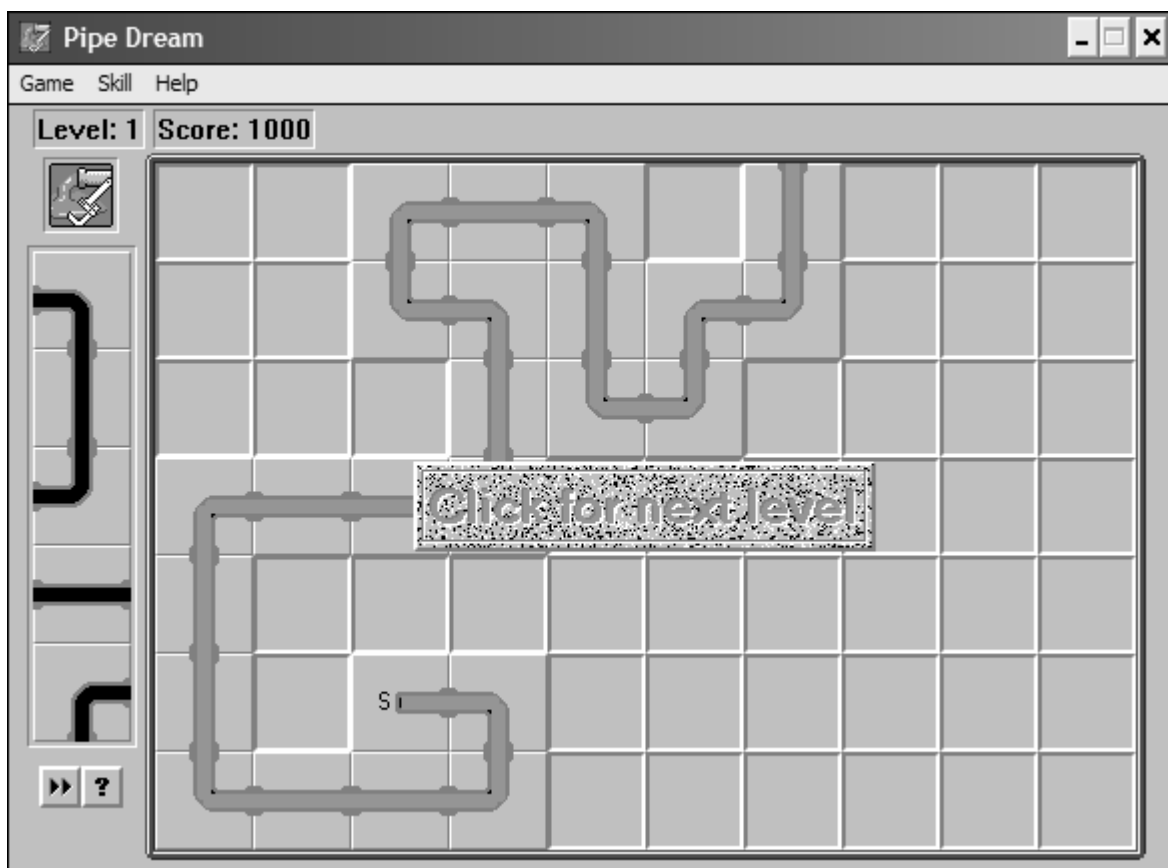


Abbildung 20: Screenshot des Spiels "Pipemania"

Das 1989 von LucasFilm Games veröffentlichte Spiel „Pipemania“⁶² zeigt ebenfalls Ähnlichkeiten zum Programm „soundtubes“. Das Spielprinzip ist einfach. Der

⁶² „Pipe Mania“ war der europäische Titel des Spiels. Der Originaltitel lautete „Pipe Dreams“.

Benutzer bekommt in zufälliger Reihenfolge verschiedene Rohrteile präsentiert. Aus diesen soll er einen Pfad von einem Start- zu einem Endpunkt bauen. Nach Ablauf eines Countdown beginnt vom Startpunkt weg eine Wasserflut zu laufen. Wie auch beim Streckeneditor von „Stunts“ ist auch bei diesem Spiel der Bauplan in klar abgegrenzte Quadrate unterteilt.

Sound

„soundtubes“ ist ein loop-basiertes Musikprogramm, das die Sequenzierung von Samples erlaubt. Diese Samples können noch in Lautstärke und Tonhöhe moduliert werden. Die Samples selbst sind gesprochene Buchstaben, die von einem Sprachsynthese-Programm erzeugt wurden. Die Samples können sechs Stufen nach unten und sechs Stufen nach oben in der Tonhöhe verändert werden, wobei diese Stufen nicht an Notenwerten orientiert sind. Der Benutzer ist somit nicht dem Zwang der Tonalität unterworfen. Die Tonhöhe der Samples wird durch die Veränderung der Abspielgeschwindigkeit moduliert. Dabei wandeln sich die klanglichen Qualitäten der Samples und entfernen sich akustisch stark von ihrem Ursprung. Sie werden so zu einem musikalisch interessanten Material. Die Lautstärke kann in drei Stufen verringert oder erhöht werden. Dabei wird keine Rücksicht auf die ursprüngliche Lautstärke des Samples genommen.

Die Verwendung von sprachlichem Lautmaterial in „soundtubes“ kann unter dem Gesichtspunkt der Onomatopoesie gesehen werden. Der Benutzer kann mit diesem Programm versuchen, Gefühl, Stimmungen oder Geräusche in lautmalerische Wortneuschöpfungen zu übertragen.

Die Länge der Loops ist nicht festgelegt. Der Benutzer ist auf kein sechzehnstufiges Schema beschränkt, das sich in vielen Sequenzern finden lässt.

Verworfenе Ansätze

Hier werden Ansätze der Soundgestaltung vorgestellt, deren Implementierung überlegt wurde und schlussendlich verworfen wurde.

- Klangsynthese mit VSTI⁶³

Es wurde angedacht, dem Benutzer zu ermöglichen, Klänge zu synthetisieren. Um den Klang möglichst umfangreich gestaltbar zu halten, wurde daran gedacht, VST-Instrumente zu verwenden, die über OSC (Open Sound Control) angesteuert werden. Diese Möglichkeit wurde aus Gründen des Interface-Designs verworfen.

- Aufnahme-Möglichkeit

⁶³ VSTI: Virtual Studio Technology Instrument: Von der deutschen Musiksoftwarefirma Steinberg entwickeltes System zur Kommunikation zwischen VST-Host und virtuellen Instrumenten (VSTI).

Um den Benutzer und besonders Kinder stärker in das Programm einzubinden, wurde überlegt, dem Anwender die Möglichkeit zu geben, sich selbst aufzunehmen und diese Samples im Programm zu verwenden. Da Kinder von Mikrofonen und der sich bietenden Möglichkeit, die eigene Stimme, also sich selber, anders zu hören, sehr begeistert sind, hätte diese Funktion das Programm für Kinder interessanter gemacht. Die Implementierung dieser Funktion wurde am Beginn der Entwicklung diskutiert, wurde dann aber nicht weiterverfolgt und daher nicht ins Programm aufgenommen.

Interface

Die Zielsetzung des Interface von „soundtubes“ war es, eine möglichst intuitive Benutzerschnittstelle zu schaffen, die es auch ungeübten Benutzern ohne lange Lernphase erlaubt, das Programm zu benutzen.

Das Interface des Programms ist daher nach den Prinzipien der direkten Manipulation gestaltet. Das Objekt, der Bauplan, ist ständig im Blickfeld des Benutzers. Veränderungen am Bauplan werden sofort angezeigt. Der Benutzer interagiert über das physikalische Eingabegerät der Maus. Das Programm ist einfach verständlich und kann von Laien ohne großes Vorwissen benutzt werden. Eine Besonderheit an „soundtubes“ ist, dass die Rohrleitungen - auch noch während die Verdickungen durch die Röhren laufen - verändert werden können.

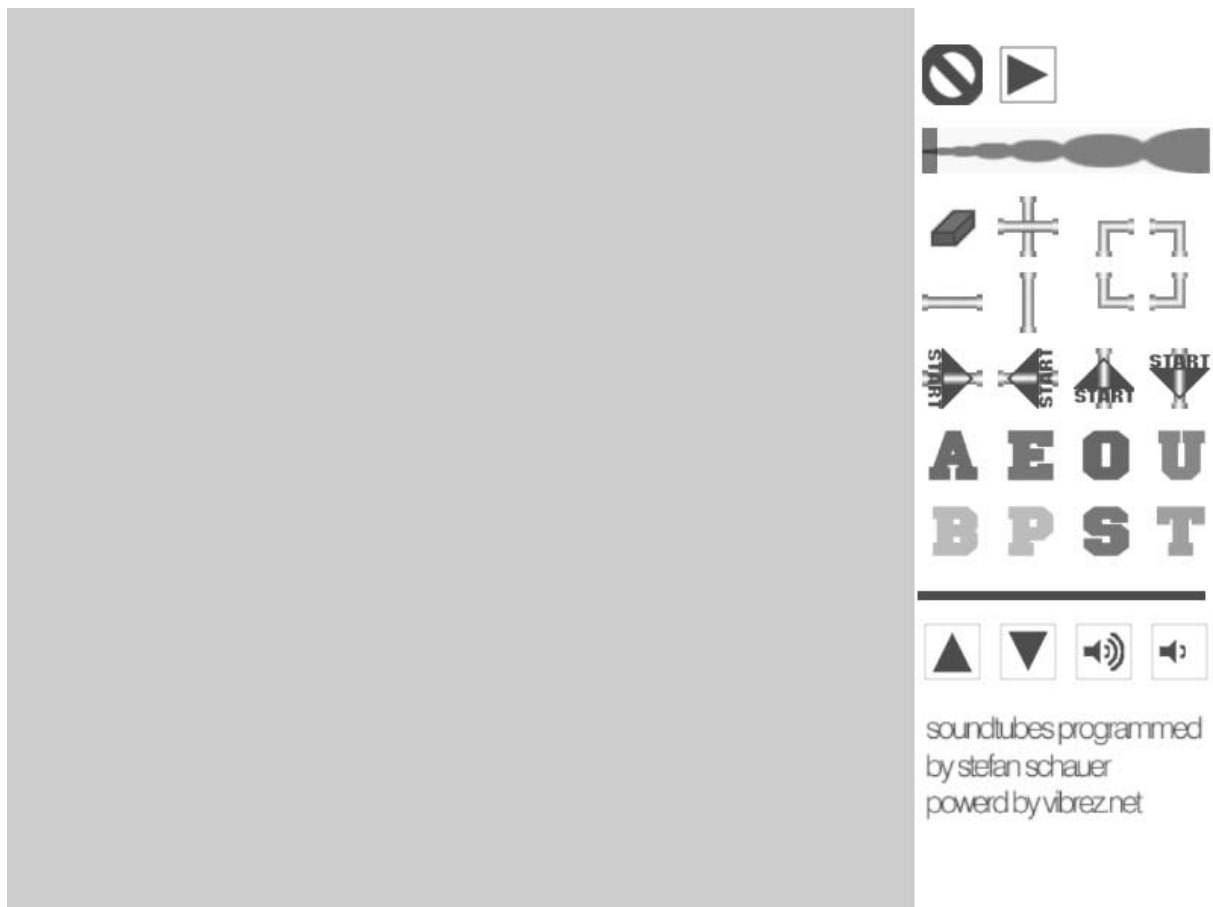


Abbildung 21: Screenshot der Endversion des Programms „soundtubes“ nach dem Start

Durch den Einsatz folgender Techniken wurde versucht, das Programm einfach bedienbar zu machen.

- Eingabegerät Maus

Als Eingabegerät werden lediglich die Maus und dabei nur der Einfachklick der linken Maustaste benutzt. Bei einer deutschen Untersuchung unterschiedlicher Kinder-Musiksoftware wurde beobachtet, dass Kinder Schwierigkeiten haben, die Maustasten zu unterscheiden. Funktionen in diesen Programmen, die über einen rechten Maustastenklick erreichbar waren, wurden daher nicht benutzt. Auch wurde bei den getesteten Programmen die Verwendung von Shortcuts (Tastenkombinationen) zur Bedienung bemängelt. [GöRu05]

Darüber hinaus wurde bei „soundtubes“ auf komplizierte Bedienungsmodi der Maus wie „Drag-and-Drop“ verzichtet. Stattdessen wird das Programm über Einfach-Klick-Operationen bedient. Diese sehr reduzierte Bedienung sollte die Benutzung des Programms „soundtubes“ so einfach wie möglich gestalten.

- Feedback-Mechanismen

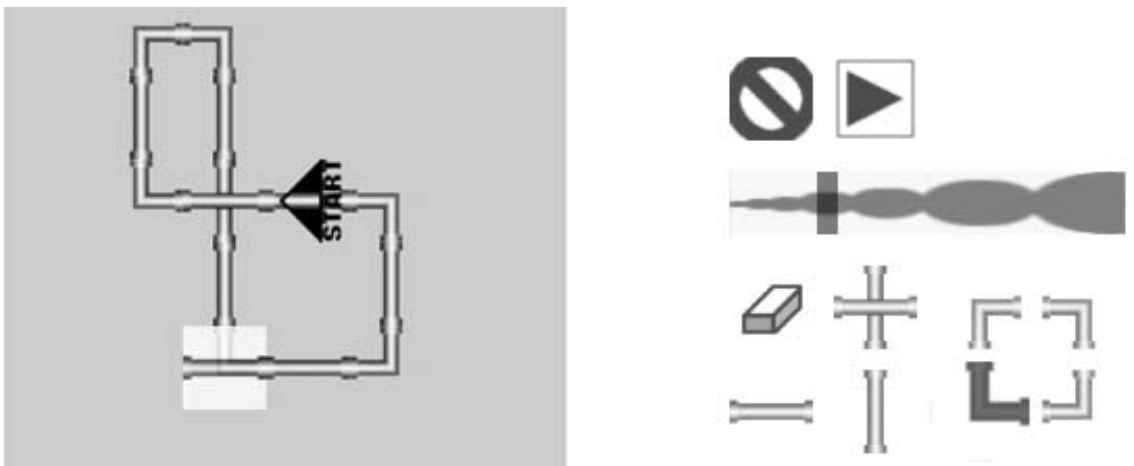


Abbildung 22: Zwei Detailausschnitte aus dem Programm "soundtubes"

Damit der Anwender zu jedem Zeitpunkt der Benutzung des Programms weiß, welche Funktion er gerade verwendet, sind Rückmeldungsmechanismen implementiert. Zum einen nimmt der Cursor das Aussehen der angewählten Funktionalität an sobald er sich auf dem Bauplan befindet. Zum anderen wird die angewählte Funktion im Bedienungspanel gekennzeichnet. Wird ein Rohrleitungselement angewählt, so wird dieses größer und dunkler dargestellt. Wählt der Benutzer ein Element oder eine Funktion aus, so wird ihm nicht nur visuell sondern auch auditiv Rückmeldung gegeben. Wird ein Buchstabe ausgewählt, so wird dieses Sample abgespielt. Durch diese Rückmeldungsmechanismen ist der Anwender zu jedem Zeitpunkt der Benutzung informiert, welche Funktion oder welches Element er gerade gewählt hat, ohne abgelenkt zu sein.

- Rohrleitungsmetapher

Ein wichtiger Punkt bei der Entwicklung des Programms „soundtubes“ war es, einen loop-basierten Sequenzer zu entwickeln, der aber den Benutzer nicht in ein sechzehnstufiges Schema zwingt, wie es bei vielen anderen Sequenzern der Fall ist. Daher wurde nach einer Metapher gesucht, die das Erstellen von unterschiedlich langen Loops erlaubt. Das Programm „Drumsteps“ war ausschlaggebend, eine Metapher zu wählen, die der Realität entlehnt ist. Schließlich fiel die Entscheidung, die Loops als Straßen zu visualisieren. Im Laufe der Entwicklung wurden, wie weiter unten genauer ausgeführt, die Straßen durch Rohrleitungen ausgetauscht.

- Darstellung des Klangmaterials

Ein großes Problem von sample-basierten Musikprogrammen ist die Darstellung der Samples in einer Form, die Rückschlüsse auf das Klangmaterial zulässt.

Eine einfache Lösung ist, die Samples durchzunummerieren. Will der Benutzer ein Sample an einer bestimmten Stelle der Sequenz abspielen, so muss er die Nummer des Samples dort eintragen. Er muss also die Zuordnung zwischen Nummer und Sample wissen. Diese Herangehensweise ist in vielen Tracker-Programmen zu finden, wie etwa dem Programm „Fast Tracker II“.



Abbildung 23: Screenshot des Programms "FasttrackerII"

Eine andere Lösung ist die Verwendung der Darstellung des Oszillogramms⁶⁴ des Samples. Diese Variante ist aber wenig aussagekräftig, da sie nur den Verlauf der Intensität eines Samples gegen die Zeit zeigt, was aber keine genaueren Informationen über die klanglichen Eigenschaften übermittelt. Ein

⁶⁴ Oszillogramm: Diagramm, das in der y-Achse die Intensität eines Signals und in der x-Achse den Zeitverlauf anzeigt.

Vorteil dieser Herangehensweise ist, dass die Länge des Samples angezeigt wird. Im aufnahmeorientierten Programm „Steinberg Cubase“⁶⁵ oder im Apple-Musikprogramm „GarageBand“ werden die Samples auf diese Weise dargestellt.

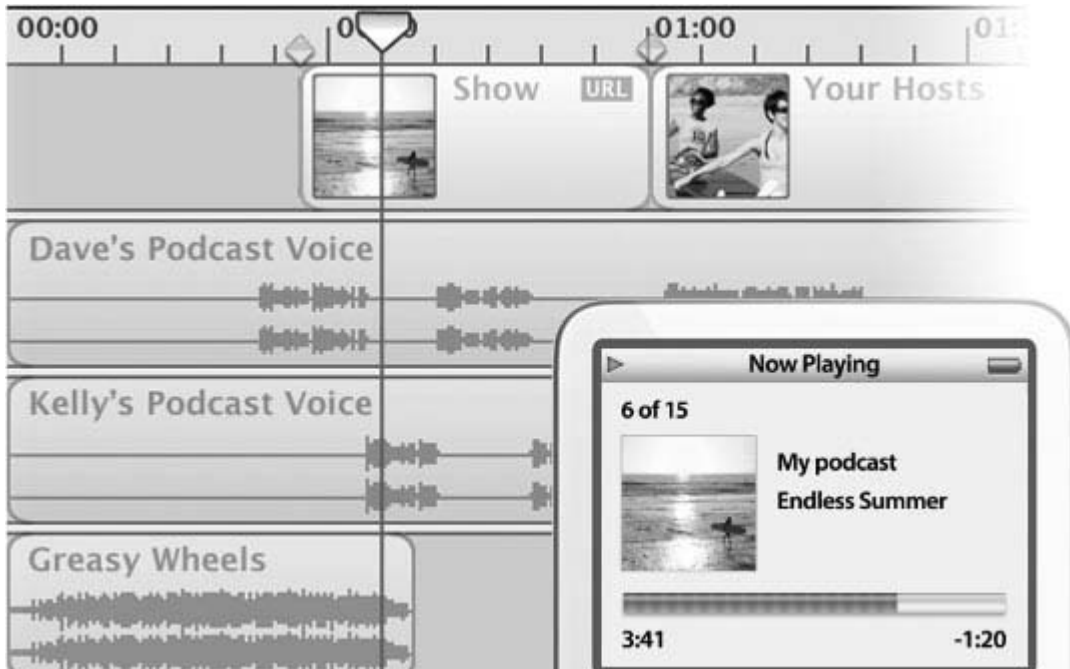


Abbildung 24: Screenshot des Programms "GarageBand"

Eine weitere Lösung ist, Samples von akustischen Instrumenten als Soundmaterial zu verwenden, die über eine vereinfachte Darstellung des Instruments visualisiert wird. Ein Nachteil dieser Herangehensweise ist, dass die Anzahl der verwendbaren Samples beschränkt ist. So ist es zum Beispiel nur schwer möglich zwei verschiedene Basstrommeln unterscheidbar zu visualisieren. Auch ist das Klangmaterial auf Aufnahmen von akustischen Instrumenten eingeschränkt. Oftmals werden bei dieser Herangehensweise MIDI-Klänge als Klangmaterial benutzt.

⁶⁵ Cubase ist ein Produkt des Unternehmens Steinberg. 1984 von den beiden Deutschen Karl Steinberg und Manfred Rürup gegründet, veröffentlichten sie als erstes Produkt Pro-16, einen 16stimmigen MIDI-Sequencer für den Commodore C-64. 1996 wurde Cubase veröffentlicht. Dieses Programm war zum Veröffentlichungszeitpunkt eines der innovativsten Produkte zur Musikproduktion am Computer. Aus dem Aufnahme- und MIDI-Sequencer-Programm Cubase wurde im Laufe der Zeit ein Programm zur umfassenden Musikproduktion. Technische Innovationen wie VST (Virtual Studio Technology) und ASIO (Advanced Sound Input Output) von Steinberg sind heutzutage Standard und wurden von vielen anderen Programmen übernommen. [Stein05]

Das Problem der Sample-Darstellung wurde bei „soundtubes“ so gelöst, dass nur Samples von Buchstaben verwendet werden und diese durch die entsprechenden Buchstaben visualisiert werden. Um die Samples noch besser unterscheidbar zu machen, wurden die Buchstaben mit unterschiedlichen Farben versehen.

- **Sample-Parameter**

Das Programm „soundtubes“ erlaubt dem Benutzer die Samples in Tonhöhe und Lautstärke zu verändern.

Tracker-Musikprogramme besitzen auch diese Funktion. Musikprogramme dieser Kategorie sind dadurch gekennzeichnet, dass die Tonhöhe meist über Notenwerte eingegeben wird, wobei die Originaltonhöhe des Samples als der Notenwert C-4 codiert ist. Die Lautstärke wird dabei als hexadezimaler Wert eingegeben.

Im Programm „soundtubes“ wird die Tonhöhe über den vertikalen Versatz, die Lautstärke über die Größe des Buchstaben dargestellt. Diese Darstellung ist nicht sehr genau, wurde aber gewählt, weil auf die exakte Repräsentation kein großer Wert gelegt wurde. Ein ähnlicher Ansatz ist im Gameboymusikprogramm „nanoloop“ zu finden.

Verworfenе Ansätze

Hier werden Interface-Ansätze vorgestellt, deren Implementierung überlegt wurde. Diese machen die endgültige Realisierung des Programms „soundtubes“ verständlicher, da sie Stufen der Entwicklung des Programms zeigen.

- **Klangsynthese**

Es wurde überlegt, dem Benutzer die Möglichkeit zur Verfügung zu stellen, Klänge zu synthetisieren. Um eine interessante Klanggestaltung zu ermöglichen, hätte der Benutzer eine Vielzahl von Klangparametern verändern können müssen. Im bestehenden Interface hätte der Benutzer aber nur wenige Parameter verändern können. Andere Interface-Lösungen hätten gewählt werden müssen. Diese hätten die Darstellung der weiteren Parameter in einem anderen Bereich der Benutzerschnittstelle oder eine völlige Änderung des Designs notwendig gemacht.

- **Icon „Löschen“**

Da die Zielgruppe des Programms „soundtubes“ Kinder sind und daher nicht unbedingt davon ausgegangen werden kann, dass beschriftete Buttons gelesen werden können, wurden alle Bedienungselemente als graphische Icons realisiert. Dabei bereitete die graphische Umsetzung des Icons für das Löschen einzelner Elemente und für das Löschen des gesamten Bauplans besondere Probleme.



Abbildung 25: Icon „Löschen eines einzelnen Elements“ Mai 2005 bis Mai 2006 und Juni 2006 bis Endversion Dezember 2006



Abbildung 26: Icon „Löschen des gesamten Bauplans“ Juni 2005 bis Dezember 2005 und Jänner 2006 bis Endversion Dezember 2006

- Abstrakte Straßen-Metapher

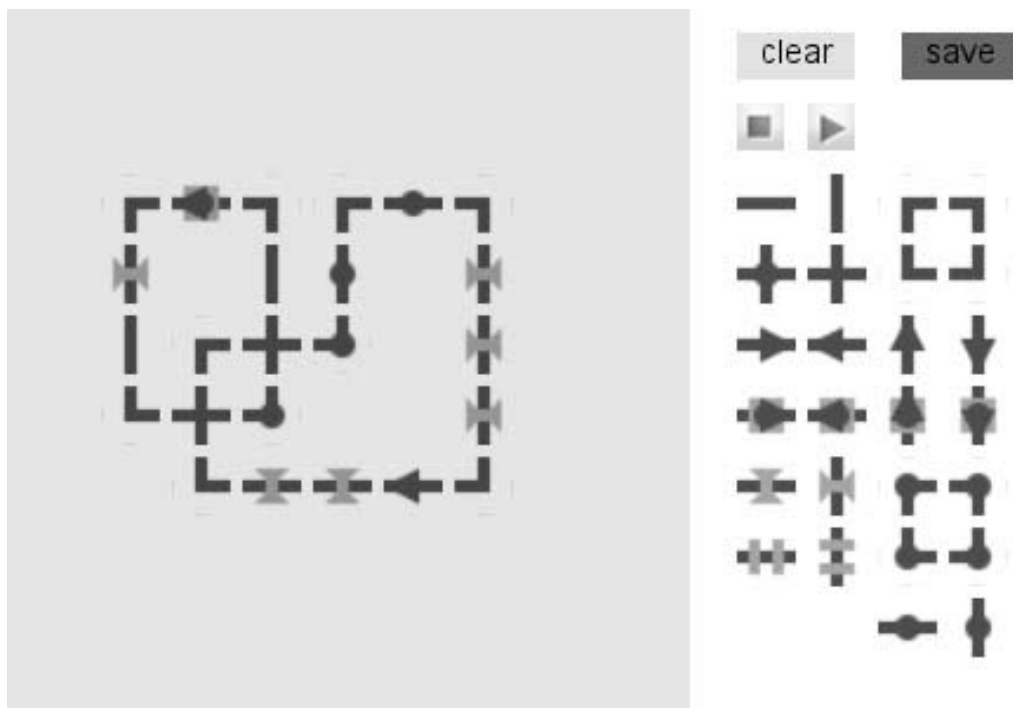


Abbildung 27: Screenshot des Programms „soundtubes“ Juni 2005

Am Anfang der Entwicklung des Programms wurde das Interface sehr minimalistisch und einfach gehalten. Die Straßen-Metapher wurde dabei eher vernachlässigt. Von diesem sehr reduzierten Design wurde bald abgegangen,

da Kinder als Zielgruppe gewählt wurden. Das Aussehen wurde wieder stärker an die Straßen-Metapher angelehnt. In der Endversion wurde wieder auf ein minimalistischeres Design zurückgegriffen.

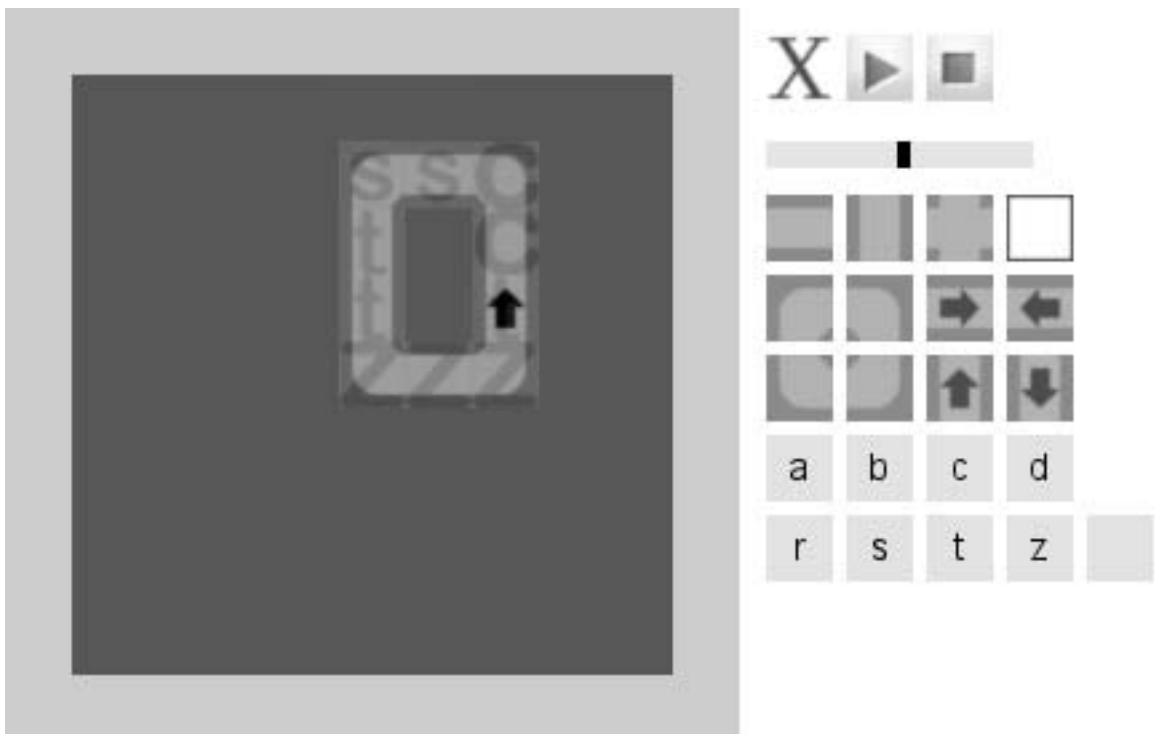


Abbildung 28: Screenshot des Programms „soundtubes“ Juli 2005

- Straßenelemente als Audio-Effekt

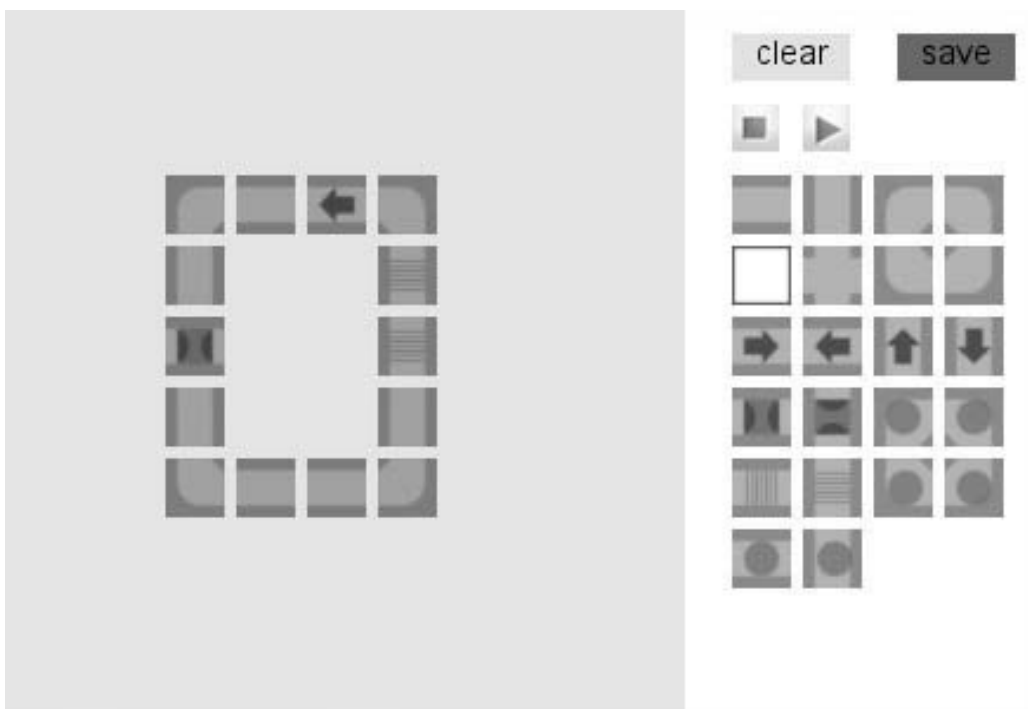


Abbildung 29: Screenshot des Programm „soundtubes“ Mai 2005

Die Straßen-Metapher ließ die Integration von Audio-Effekten in das Interface zu. Geplant waren Vibrato- und Echo-Effekte. Speziell gekennzeichnete Straßenelemente sollten Träger dieser Effekte sein. Die dazu verwendeten Metaphern wurden aus der Realität entlehnt. Der Vibrato-Effekt wurde als Straße mit Bodenschwellen umgesetzt. Der Echo-Effekt wurde als untertunnelte Straße visualisiert. Diese Idee wurde aus drei Gründen wieder aus dem Interface entfernt. Zum einen wurde im Laufe der Entwicklung von der Straßen-Metapher abgegangen, was auch die Umsetzung dieser Idee unmöglich machte. Zum anderen wurde die Entscheidung getroffen, Samples als Buchstaben auf die Straßen zu legen. Die Sichtbarkeit der darunter liegenden Straßen wäre somit nicht einwandfrei gewährleistet gewesen. Weiters war eine klar erkennbare graphische Umsetzung der beiden speziellen Straßenelemente nicht einfach zu bewerkstelligen.

- Straßen-Metapher und Umsetzung der Funktionen Lautstärken- und Tonhöhenänderung

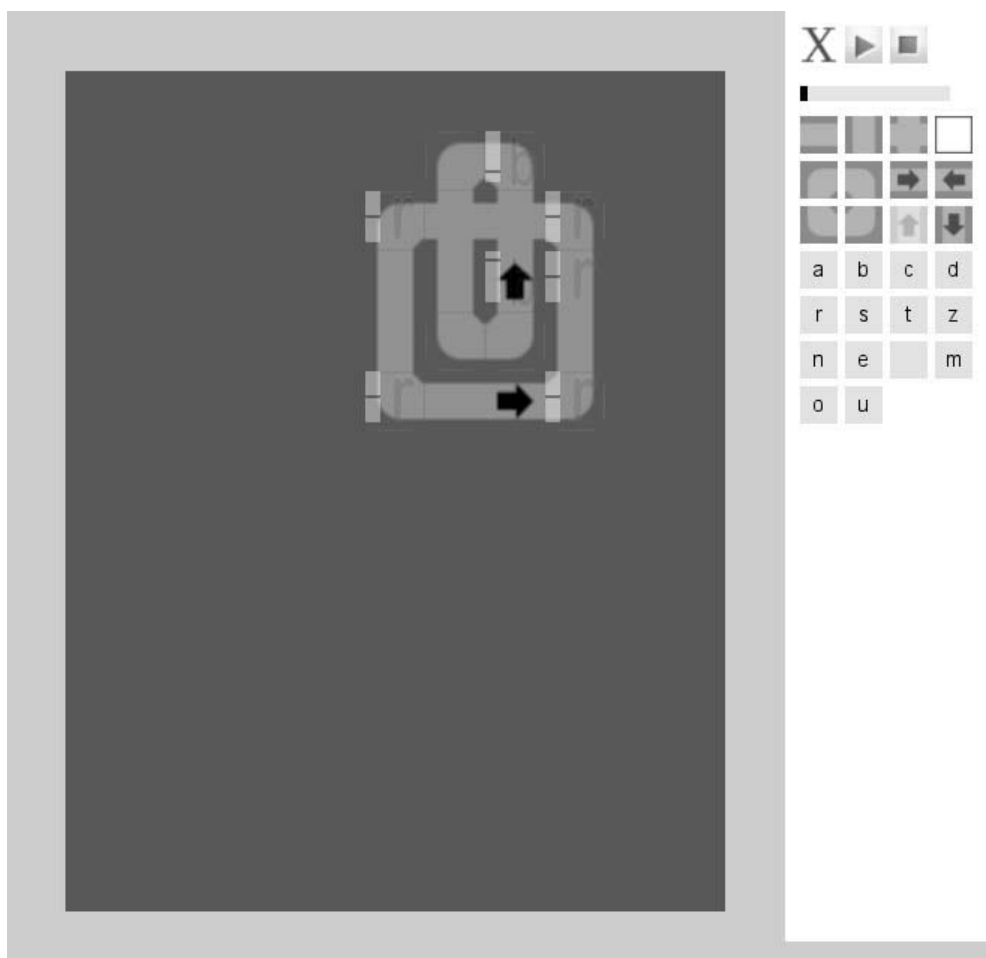


Abbildung 30: Screenshot des Programms „soundtubes“ November 2005

Die Grundidee bei der Entwicklung des Programms „soundtubes“ war Loops als Straßen umzusetzen. Diese Metapher hat den Vorteil, dass sie die Umsetzung von unterschiedlich langen Loops ermöglicht. Darüber hinaus ist sie Kindern vertraut, die somit ohne Probleme mit dem Programm umgehen können sollten. Es wurde daher versucht, diese Metapher möglichst lebensecht umzusetzen. So waren als Visualisierung des Laufpunkts Fahrzeuge angedacht.

Im Zuge der Versuche Tonhöhen- und Lautstärkenveränderung in das Interface aufzunehmen, stellten sich Probleme der Umsetzung dieser Funktionen. Daher wurde von der Straßen-Metapher abgegangen.

Verschiedene Gründe gaben den Ausschlag für diese Entscheidung. Der grün gefärbte Hintergrund des Bauplans erschwerte die Sichtbarkeit anderer Elemente. Die Darstellung einer Straße verbrauchte viel Platz, um noch als Straße erkannt zu werden. Daher blieb wenig Platz für die Anzeige anderer Elemente.

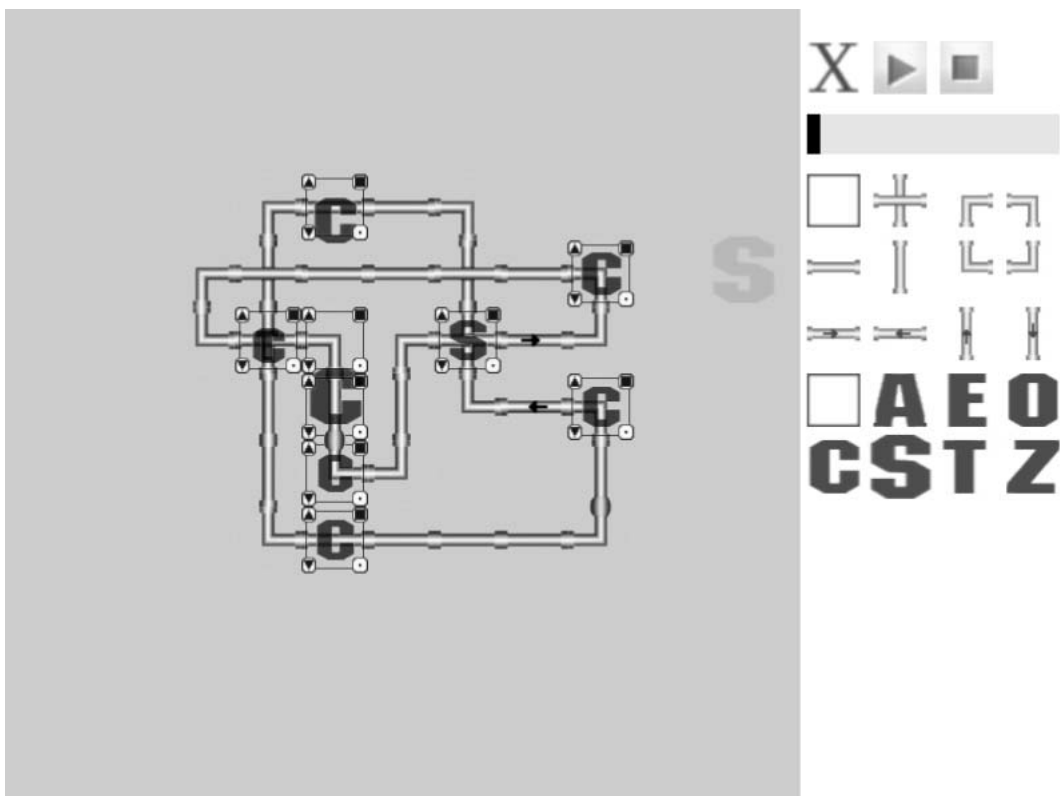


Abbildung 31: Screenshot des Programms „soundtubes“ Dezember 2005

Aus diesen Gründen wurde von der Straßen-Metapher abgegangen und die Straßen wurden durch Rohrleitungen ersetzt. Diese wurden auf einem grauen Hintergrund dargestellt, was die einzelnen Elemente sehr gut sichtbar machte. Anstelle von Laufpunkten, die auf den gebauten Straßen fahren, zwängen sich

nun Verdickungen durch die Rohrleitungen. Die Bedienelemente zur Lautstärken- und Tonhöhenänderung wurden anfangs direkt am Sample angebracht. Da diese Elemente nur sehr klein dimensioniert werden konnten, wurde diese Lösung verworfen. Stattdessen wurden diese Funktionen als anwählbare Werkzeuge ins Bedienungspanel verlegt.

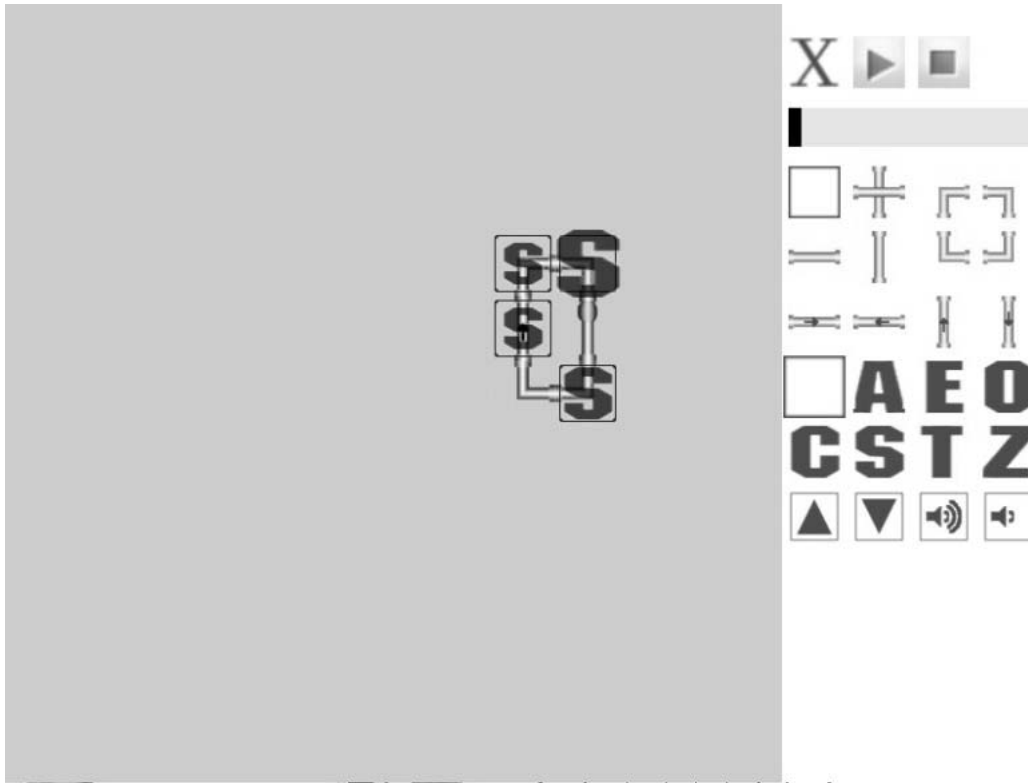


Abbildung 32: Screenshot des Programms „soundtubes“ Dezember 2005

Mögliche Erweiterungen

Im Folgenden werden mögliche Weiterentwicklungen des Programms „soundtubes“ vorgestellt, die angedacht wurden.

- **Speicherfunktionalitäten**

Die Speicher- und Ladefunktionen wurden in „soundtubes“ integriert, wegen technischen Problemen aber wieder aus dem Programm genommen. Diese Funktion wurde ohne den sonst üblichen Speicher- und Ladedialog realisiert. Dieser hätte die konsistente Gestaltung des Programms gebrochen. Zuerst wurde eine eher gewöhnliche Lade- und Speichemetapher ausprobiert.

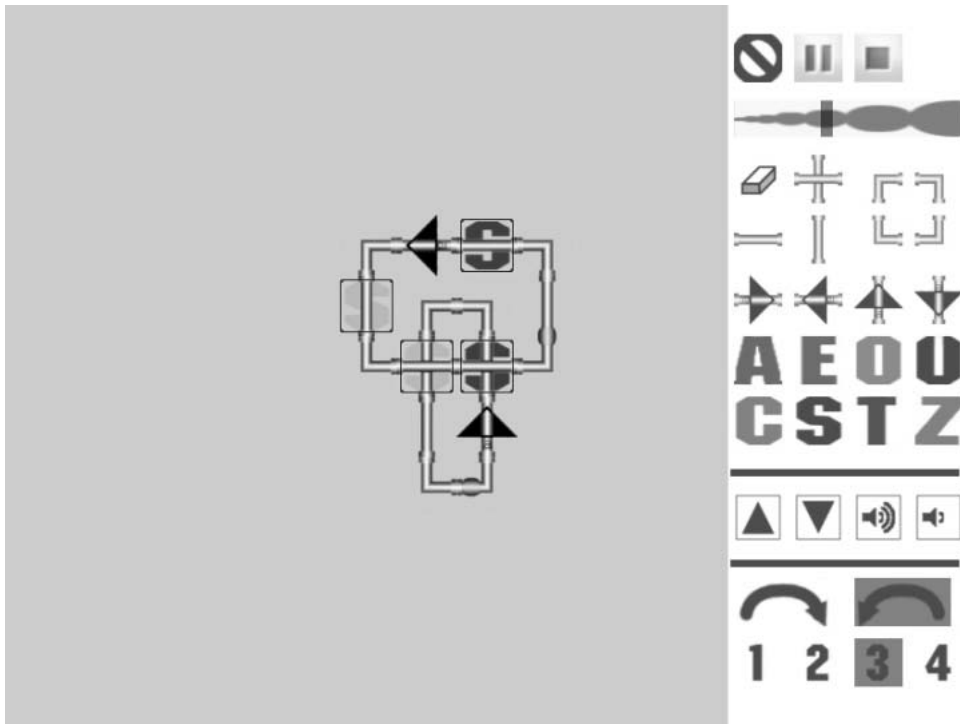


Abbildung 33: Screenshot des Programm „soundtubes“ Juli 2006

Diese wurde wegen schlechter Lesbarkeit verworfen und durch eine Karteikartenreiter-Metapher ersetzt. Durch das Anwählen eines anderen Karteikartenreiters wurde der Bauplan gespeichert. Diese Metapher ist leicht verständlich und schon aus anderen Programmen bekannt. Darüber hinaus verliert das Speichern den Charakter des Archivierens, da die gespeicherten Baupläne nicht wirklich „weg“ sondern nur gerade nicht sichtbar sind.

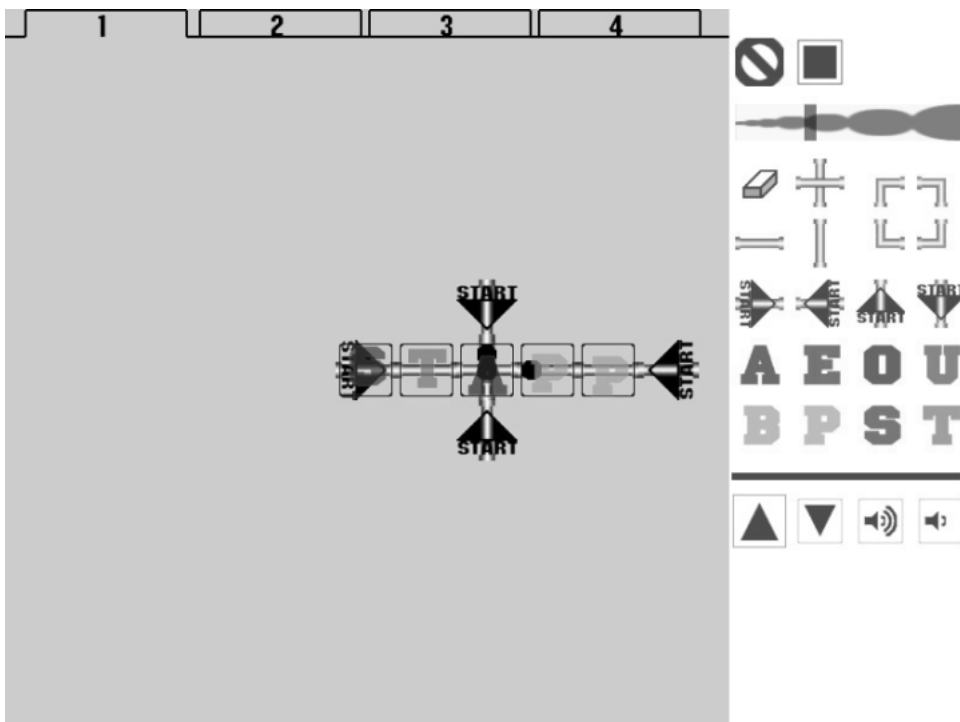


Abbildung 34: Screenshot des Programm „soundtubes“ September 2006

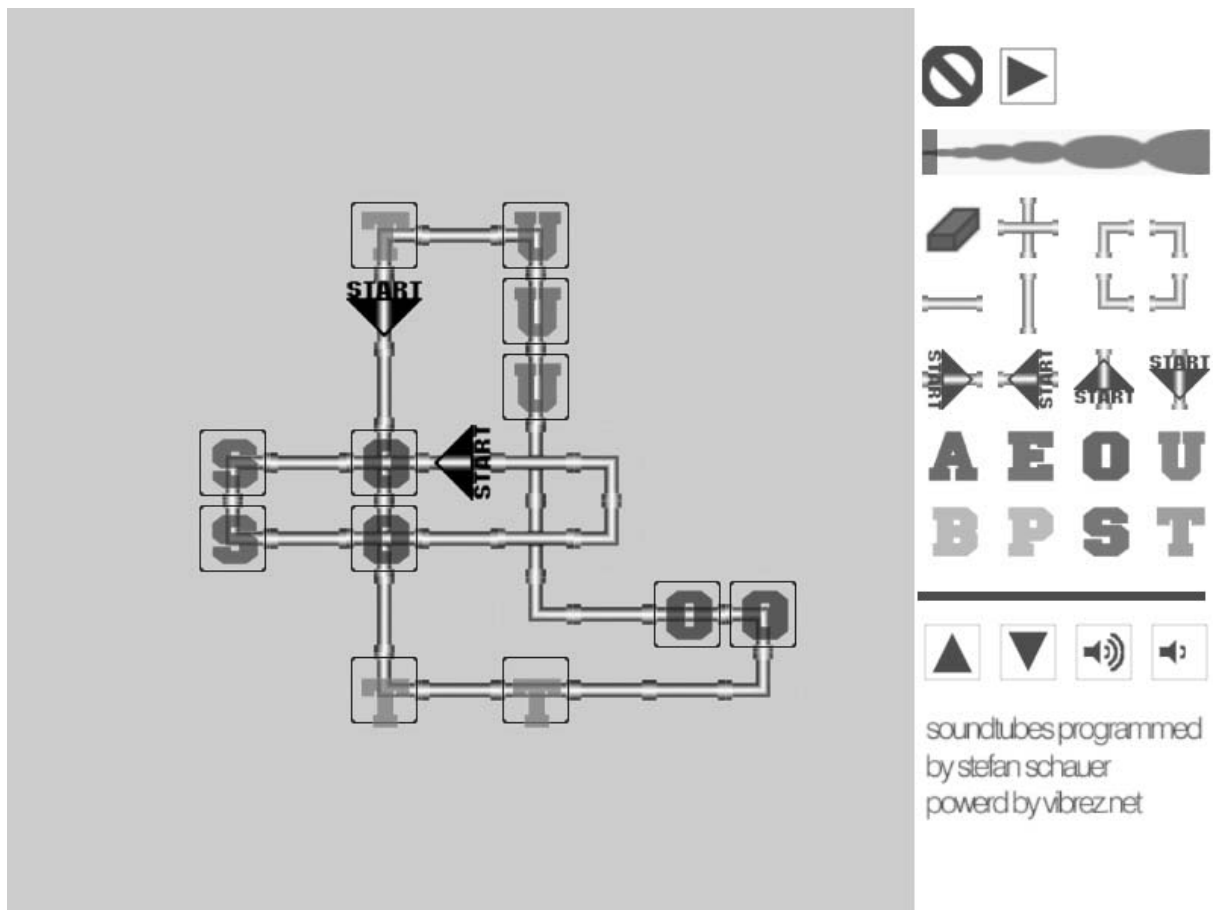


Abbildung 35: Endversion des Programm "soundtubes" Dezember 2006

Technologie

Das Programm „soundtubes“ wurde mit der Programmiersprache „vibreZ“ in der Version 0.0.3. programmiert, die im April 2005 den Testern zur Verfügung gestellt wurde. Die Nachfolge-Versionen von „vibreZ“ unterscheiden sich stark von den Vorläufern.

Umsetzung

Die Benutzeroberfläche des Programms „soundtubes“ ist 800 mal 600 Pixel groß. Diese Fläche ist in zwei Teile geteilt. Auf der linken Seite befindet sich ein 600 mal 600 Pixel großes Feld, das mit Rohrleitungen bebaut werden kann. Dieses Feld wird im Folgenden als Bauplan bezeichnet. Auf der rechten Seite befindet sich auf einem 200 mal 600 Pixel großen Feld die Steuerung des Programms. Dieser Teil wird im Folgenden als Bedienungspanel bezeichnet.

Der Bauplan ist in elf mal elf 50 Pixel große Quadrate aufgeteilt. Rund um dieses Feld befindet sich ein 25 Pixel breiter Rand. Jedes Quadrat ist mit drei übereinander liegenden Panels⁶⁶ versehen, die ohne Belegung durch eine Rohrleitung oder einen Buchstaben nicht sichtbar sind.

Wird im Bedienungspanel ein Rohrelement, ein Buchstabe oder ein Steuerungselement ausgewählt, wird ein Panel erzeugt, das nur am Bauplan sichtbar und mit der Maus mitbewegt wird. Klickt man mit einer ausgewählten Rohrleitung auf den Bauplan, wird diese Textur vom untersten Panel übernommen. Wurde ein Buchstabe ausgewählt, werden gleichzeitig zwei Panels sichtbar. Das erste Panel zeigt nur die Umrandung des 50 mal 50 Pixel großen Quadrats an. Das zweite stellt den gewählten Buchstaben dar. Der Buchstabe kann unter Verwendung der Tonhöhen- und Lautstärkenwerkzeuge nach oben oder nach unten verschoben bzw. vergrößert oder verkleinert werden.

Jede Veränderung am Bauplan wird über ein selbst programmiertes Python-External in ein externes Textfile namens matrix.mtx geschrieben. Wird ein Startpunkt am Bauplan abgelegt, werden die Koordinaten dieses Punkt auch in die Datei start.mtx geschrieben. Für jeden Startpunkt werden zwei Einträge angelegt, die durch einen Tabulator voneinander getrennt sind. Wird ein neuer Startpunkt gesetzt, werden die

⁶⁶ Ein Panel ist in „vibreZ“ ein rechteckiges Element, das mit Textur belegt werden kann.

neuen Einträge in eine neue Zeile geschrieben. Der erste Eintrag der Zeile ist die X-Koordinate, der zweite die Y-Koordinate des Panels auf dem der Startpunkt liegt.

Die Datei „start.mtx“ könnte so aussehen:

7 2

9 7

Die Datei matrix.mtx ist ein Textfile, wobei jede Zeile vier Einträge hat, die jeweils durch einen Tabulator voneinander getrennt sind. Nach den vier Einträgen folgt ein Zeilenumbruch. Pro 50 mal 50 Pixel Quadrat werden vier Zeilen mit vier Einträgen geschrieben. Da es 11 mal 11 also 121 Quadrate gibt, ist die Länge der Textdatei 484 Zeilen.

Der erste Eintrag der Zeile ist die X-Koordinate, der zweite die Y-Koordinate des Panels. Der dritte Eintrag ist eine Zahl zwischen 1 und 4, die zeilenweise um eins erhöht wird, um nach 4 wieder bei 1 anzufangen. Der Eintrag, der der Zahl 1 folgt, ist der Code des Rohres, der auf diesem Panel liegt. Die Rohre sind wie folgt codiert:

- 11 Waagrechtes gerades Rohr
- 12 Senkrechtes gerades Rohr
- 21 Von unten nach rechts gekrümmtes Rohr
- 22 Von unten nach links gekrümmtes Rohr
- 23 Von oben nach rechts gekrümmtes Rohr
- 24 Von oben nach links gekrümmtes Rohr
- 31 Rohrkreuzung
- 41 Waagrechter Startpunkt, der nach rechts weist
- 42 Waagrechter Startpunkt, der nach links weist
- 43 Senkrechter Startpunkt, der nach oben weist
- 44 Senkrechter Startpunkt, der nach unten weist
- 50 Kein Rohr

Der Eintrag, der der Zahl 2 folgt, ist der Eintrag des gewählten Buchstabens. Die Buchstaben werden nach ihrer Position im Alphabet codiert. Ist kein Buchstabe auf diesem Panel, wird der Code 50 eingetragen.

Der Eintrag, der der Zahl 3 folgt, stellt die Tonhöhe des Buchstabens dar. Dieser kann einen Wert zwischen 0 und 12 annehmen. Der Defaultwert ist 6.

Der Eintrag, der der Zahl 4 folgt, ist die Lautstärke. Dieser kann einen Wert zwischen 0 und 6 annehmen. Der Defaultwert ist 3.

Ein Ausschnitt der Datei matrix.mtx könnte etwa so aussehen:

1	1	1	50
1	1	2	50
1	1	3	6
1	1	4	3
1	2	1	11
1	2	2	1
1	2	3	4
1	2	4	5

Wird der „Play“-Button gedrückt, wird zuerst die Datei start.mtx ausgelesen und die Koordinaten der einzelnen Startpunkte werden ausgegeben. Jedes Koordinatenpaar bekommt eine einzelne Abarbeitungspipeline zugewiesen. Es gibt 18 solcher. An diese werden die beiden Zahlenwerte weitergeleitet. Die Abarbeitungspipeline wurde als Python-External realisiert. Dort wird die Datei matrix.mtx geladen. Beginnend mit den Startkoordinaten werden alle Informationen zu diesem Panel ausgelesen. Das Panel, das gerade bearbeitet wird, ist am Bauplan durch ein Panel gekennzeichnet, das mit einer Rohrwölbung texturiert ist. Liegt ein Buchstabe auf dem Panel, wird die Information über den Buchstaben gemeinsam mit der Tonhöhen- und Lautstärkeninformation an ein Modul geschickt, das diese Informationen in Klang umwandelt. Gleichzeitig wird ein eigenes Panel kurz sichtbar, das durch rote konzentrische Kreise anzeigt, dass ein Sample abgespielt wird. Die Information, auf welcher Rohrleitung man sich befindet, wird dazu benutzt, um das nachfolgende Panel zu ermitteln. Befindet sich die Wölbung auf einer waagrechten Rohrleitung, so ist die nächste X-Koordinate um einen Zähler größer oder kleiner. Ist die nachfolgende Koordinate ohne Rohrleitung, so wird der Durchlauf für diese Abarbeitungspipeline gestoppt und das Wölbungspanel wird unsichtbar geschaltet. Ändert sich die Laufrichtung um 90 Grad, so wird auch die Wölbung-Textur mitgedreht, damit die Darstellung korrekt bleibt.

Die Klangerzeugung geschieht über Samples. Diese werden beim Programmstart geladen. Um einen bestimmten Klang abzuspielen, muss die entsprechende Nachricht an das Modul geschickt werden, das dieses bestimmte Sample in der gewünschten Tonhöhe und Lautstärke abspielt. Um das Abspielen von mehreren Samples zur gleichen Zeit zu ermöglichen, ist jeder Abarbeitungspipeline ein solches Modul zugeordnet. Alle ausgegebenen Klänge werden an einen Audio-Ausgang geschickt. Um ein Übersteuern zu vermeiden, ist vor dem Ausgang noch ein Modul

vorgeschaltet, das beim Übersteigen eines gewissen Grenzwerts, die Lautstärke der Audio-Ausgabe verringert.

Verworfenе Vorgangsweisen

Hier werden verworfene Programmiervorgangsweisen vorgestellt.

- **Dynamische Abarbeitungspipelines**
Die Abarbeitungspipelines wurden anfangs dynamisch im Programm erzeugt. Sobald der Benutzer den „Play“-Button gedrückt hatte, wurden entsprechend der Anzahl der Startpunkte Abarbeitungspipelines erzeugt. Der Vorteil an dieser Vorgangsweise war, dass theoretisch eine nach oben offene Zahl von Startpunkten gesetzt werden konnte. Der Nachteil war aber, dass bei ansteigender Zahl von Abarbeitungspipelines die Reaktionszeit des Programms nachließ und im musikalischen Output Störgeräusche wie Knackser auftraten.
- **Matrix-External**
Anfangs wurde der Bauplan durch ein bereits fertig gestelltes Matrix-External in ein Textfile gespeichert. Da die Zugriffszeiten nicht zufrieden stellend waren, wurde dieses Modul als Python-External neu programmiert.

Verworfenе Technologien

Nachdem das Thema der Diplomarbeit feststand, wurde mit der Suche nach einer Programmierumgebung begonnen, auf der es möglich war, das Programm zu verwirklichen. Es wurde nach einer Programmierumgebung gesucht, die deren Konzepte bekannt waren und darüber hinaus eine gute Audio-Unterstützung zur Verfügung stellte.

- **„Pure Data“ und „Macromedia Director“**
Da „Pure Data“ über eine gute Audio-Unterstützung verfügt, eine große weltweite Community vorhanden ist und darüber hinaus umfangreiche Erweiterungen und Hilfestellungen im Internet zu finden sind, wurde anfangs diese Programmiersprache in Betracht gezogen. Die fehlende graphische Benutzerschnittstelle machte eine Kombination mit einer anderen Programmierumgebung notwendig. Fares Kayali, der seine Diplomarbeit ebenfalls am Institut für Gestaltungs- und Wirkungsforschung geschrieben hatte, löste dieses Problem, indem er „Pure Data“ mit einer graphischen Benutzeroberfläche kombinierte, die auf „Macromedia Director“ lief. Die

Kommunikation zwischen diesen beiden Programmen wurde über OSC (Open-Sound-Control) gelöst. Da aber „Macromedia Director“ nur sehr starre Interfaces zulässt, wurde diese Möglichkeit nicht mehr in Betracht gezogen.

- **mu-Framework**

Durch Zufall kam es zur Bekanntschaft mit dem „mu-Framework“ des deutschen Demoszene-Aktivisten Leonard Ritter aka „paniq“. Dieser war begeistert, dass er Unterstützung bei der Entwicklung des Frame-Works bekommen sollte. Dieses sollte sowohl eine graphische Benutzerschnittstelle als auch Audiounterstützung bereitstellen. In kürzester Zeit stellte sich aber heraus, dass die erwarteten Programmierfähigkeiten zu hoch gesteckt waren. Daher wurde von dieser Lösung wieder abgegangen. Das Framework wurde nicht weiterentwickelt.

Mögliche Weiterentwicklung

Hier werden mögliche Weiterentwicklungen des Programms „soundtubes“ aus technologischer Sicht vorgestellt.

- **Speicherungsfunctionalität**

Eine sinnvolle Erweiterung von „soundtubes“ wäre eine Integration einer Speicher- und Ladefunktion. Es wurde bereits versucht, diese zu integrieren, doch traten Probleme auf, die nicht ohne größeren Aufwand zu bereinigen gewesen wären. Deshalb wurde dieser Teil aus dem Programm entfernt.

- **Portierung auf andere Programmiersprache**

„soundtubes“ wurde mit der Programmiersprache „vibrez“ in der Version 0.0.3. programmiert. Diese Version unterscheidet sich stark von den folgenden Versionen, da die verwendete Syntax von diesen nicht mehr unterstützt wird. Darüber hinaus läuft vibrez 0.0.3. und somit „soundtubes“ nur auf Rechnern, die das Betriebssystem „Microsoft Windows XP“ installiert haben. Daher wäre eine Portierung des Programms auf eine andere plattformunabhängigere Programmiersprache wie etwa C++ sinnvoll.

Für das Prototyping stellt vibrez 0.0.3. eine sehr gute Plattform dar, aber viele Kleinigkeiten stellen in „Pure Data“-Programmierung nicht so erprobte Personen vor größere Probleme. Der große Vorteil am „vibrez“ ist, dass der Benutzer schnell Interfaces zusammenstellen kann. Auch ist die

Klangerzeugung und -steuerung sehr einfach zu realisieren, da sich der Programmier nicht um Soundtreiberprobleme kümmern muss.

Zusammenfassung

In diesem Kapitel wurde das Programm „soundtubes“ vorgestellt. Da das Programm mit Unterbrechungen über einen Zeitraum von fast zwei Jahren entwickelt wurde, gab es eine Vielzahl von Veränderungen und Entscheidungen, die in diesem Kapitel dargelegt wurden. Die im Programm „soundtubes“ verwirklichten Ideen sind von verschiedenen Programmen inspiriert. Diese sind die Musikprogramme Gbloink!, Drumsteps, Elektroplankton, SimTunes und das Gameboymusikprogramm nanoloop wie auch die beiden Spiele Stunts und Pipemania. Das Wissen um diese Einflüsse macht die Endversion verständlicher.

Das Programm wurde in den drei Aspekten Sound, Interface und Technologie erläutert. „soundtubes“ stellt ein loop-basiertes Musikprogramm dar, das die Sequenzierung von Samples erlaubt, die in Tonhöhe und Lautstärke moduliert werden können. Betrachtet man das Interface, so wird die Metapher der Rohrleitung benutzt, um die Idee des Loops anschaulich zu machen. Der Benutzer kann über das Eingabegerät der Maus diese Rohrleitungen direkt manipulieren. Technologisch gesehen wurde „soundtubes“ mit der objektorientierten datenstrombasierten Programmiersprache „vibrez“ umgesetzt.

Zusammenfassung

In dieser Diplomarbeit wurde das Programm „soundtubes“ beschrieben. In den Kapiteln Sound, Technologie und Interface wurden die Hintergründe des Programms behandelt. Im Kapitel „soundtubes“ fanden sich dann diese drei Aspekte des Programms wieder und es wurden die gewählten Lösungen präsentiert.

Das Programm „soundtubes“ ist ein loop-basierter Sequenzer, der es dem Benutzer erlaubt, mit Samples musikalisch zu arbeiten. Die Samples können darüber hinaus in Lautstärke und Tonhöhe verändert werden. Daher wurden die Grundlagen des Samplings und die Ursprünge des Loops und der Soundmodulation behandelt. Da die zur Verfügung gestellten Klänge Buchstabenlaute sind, wurde auch auf das Thema der Onomatopoesie, der lautmalerischen Wörter eingegangen.

„soundtubes“ wurde in der visuellen Programmiersprache „vibrez“ programmiert, die auf der ebenfalls visuellen Programmiersprache „Pure Data“ basiert, deren Programmierer Puckette Miller auch die visuelle Programmiersprache „MAX“ entwickelt hat. Daher wurden diese drei Programmiersprachen vorgestellt und allgemein auf die Gruppe der visuellen Programmiersprachen eingegangen.

Das Interface des Programms „soundtubes“ wurde nach den Prinzipien der direkten Manipulation gestaltet. Um es für den Benutzer einfach benutzbar zu machen, wurden in der Gestaltung der Benutzerschnittstelle Metaphern verwendet. Als Eingabegerät fungiert ausschließlich die Maus. Der Benutzer bedient das Programm mit Einfachklicks der linken Maustaste. Auf komplizierte Mausoperationen wie „Drag-And-Drop“ oder Doppelklick wurde verzichtet.

Das Programm wurde mit Unterbrechungen in einem Zeitraum von zwei Jahren entwickelt. Das ermöglichte das Ausprobieren von verschiedenen Variationen der Grundidee.

Kritik am Programm

Im Mai 2006 war Oliver Wittchow, der Programmierer von „nanoloop“ ein Monat lang in Wien, um sein neuestes Programm „fee“ von Kindern und Erwachsenen ausprobieren zu lassen. Im Zuge des Aufenthalts trafen wir uns zum Gedankenaustausch. Wir beide haben unsere Programme von Kindern im Alter zwischen sieben und zehn Jahren testen lassen. Mein Programm war eigentlich auf diese Altersgruppe ausgelegt. Tests im Jänner 2006 zeigten mir aber, dass diese

Zielgruppe nicht sehr gut erreicht wurde. Die folgenden, von uns beiden festgestellten Eindrücke sind völlig subjektiv und würden einer wissenschaftlichen Überprüfung bedürfen, die den Umfang dieser Diplomarbeit sprengen würde. Wir beide stellten fest, dass Kinder sehr stark von der visuellen Ebene angesprochen wurden. Die Musik, die dabei entstand, wurde von den Kindern als Beiwerk akzeptiert aber nicht als Hauptelement betrachtet. Das Programm „SimTunes“ funktioniert genau auf dieser Ebene. Dieses Programm ist eher eine Sonifikation, also eine akustische Umwandlung eines Bildes, als ein Musikprogramm und wurde daher gern von Kindern verwendet. Auch wird die Relation zwischen Bild und Ton von dieser Benutzergruppe schlecht verstanden. Deshalb werden visuelle Elemente kaum als musikalisch bedeutsam interpretiert. Ist dieser Zusammenhang visuell nicht ausdrücklich bemerkbar, wird er von Kindern nicht nachvollzogen. Oliver Wittchow merkte an, dass Kinder mit dem Notensetzen wie es bei dem Programm „fee“ implizit verlangt wird, überfordert sind und viel lieber Vorschläge vom Computer austesten und verändern. Kinder haben auch kein großes Verlangen nach Musizieren. Der Spielaspekt ist ihnen oftmals wichtiger. So wurde die „fee“ häufig so benutzt, dass Kinder sich gegenseitig mit ihrem Cursorpunkt verfolgten und die von den anderen Spielern gesetzten Noten zu löschen versuchten. Dass dabei keine Musik entstand, war ihnen egal. Oft wurde die „fee“ oder auch „nanoloop“ als Spiel verstanden, bei dem es darum geht, alle Felder des Sequenzer so schnell wie möglich zu füllen.

[Witt06]

Für mich kristallisierte sich daher heraus, dass Kinder im Volksschulalter kaum an Musikprogrammen mit Sequenzerfunktionalitäten interessiert sind. Das „Tun“ ist für Kinder in dieser Altersgruppe viel interessanter als das „Erbauen“ eines Musikstücks. Um Kinder bei einer Sache zu halten, muss von ihnen explizit Aktivität gefordert sein. Andernfalls verlieren sie bald das Interesse daran. Ich denke daher, dass ein Musikprogramm für Kinder im Volksschulalter einen stark spielerischen Charakter haben sollte. Anstatt sich auf die Sequenzerfunktionalitäten zu konzentrieren, sollte das Programm eher in Richtung Instrument gehen.

Oliver Wittchow zog für sich den Schluss daraus, dass ein ideales Musikprogramm für Kinder im Volksschulalter ein Programm am Nintendo Gameboy wäre, das aus verschiedenen Mini-Programmen besteht. Diese sollten unabhängig voneinander „spielbar“ sein, aber gemeinsam eine Art Orchester bilden. [Witt06]

Persönliche Kritik

Diese Diplomarbeit war der Versuch, ein leicht bedienbares Musikprogramm zu programmieren, das Kindern und Laien einen schnellen Zugang zur Musikproduktion am Computer bietet. Darüber hinaus sollten gängige Interface-Metaphern vermieden und klanglich ein neuer Zugang zu Musik eröffnet werden. Zwei Annahmen, die zu Beginn der Entwicklung gemacht wurden, stellten sich als falsch heraus.

Wie oben ausgeführt, sind Kinder vielfach eher an einem Instrument als an einem Sequenzer interessiert. Die visuelle Ebene von Computerprogrammen spricht sie viel mehr an als die akustische. Ein Musikspiel mit einem Interface, das dem Benutzer erlaubt mit Vorschlägen des Computers zu experimentieren, wäre für Kinder viel ansprechender.

Eine weitere Annahme, die gemacht wurde, ist, dass die Probleme der Benutzer mit Musikprogrammen aus schlecht gestalteten Interfaces resultieren. Dies stimmt zwar zu großen Teilen, es sollte aber nicht vergessen werden, dass die Musikproduktion am Computer wie auch das Spielen eines Instruments jahrelanger Übung bedarf. Die Umsetzung musikalischer Ideen in entsprechende Musikstücke bedarf einer intensiven Auseinandersetzung des Benutzers mit dem Programm. Daher sind viele Musikprogramme entweder schwierig zu erlernen oder sie legen dem Benutzer gewisse Vorstellungen von Musik nahe.

Dennoch scheint die ursprüngliche Zielsetzung bei der Entwicklung des Programms, ein einfach bedienbares Musikprogramm zu programmieren, nicht verfehlt worden zu sein. Es bleibt nur zu hoffen, dass durch „soundtubes“ einem größeren Personenkreis die Musikproduktion am Computer ermöglicht wird.

Quellen

- [BaFl93] Bartz Edek, Flos aus „Kraftwerk Unplugged?“ aus Genetische Kunst – Künstliches Leben“ Festival-Katalog Ars Electronica 1993
- [Bast06] Bastien, Patrick: Pitch shifting and voice transformation techniques, TC Helicon, <http://www.tc-helicon.com/Whitepapers>, letzter Zugriff: 18.12.2006
- [Bern05] Bernsee M. Stephan: Time Stretching And Pitch Shifting of Audio Signals, <http://www.dspdimension.com>, 2005
- [BlGr99] Blackwell A.F., Green T.R.G.: Does Metaphor Increase Visual Language Usability, Proceedings 1999 IEEE Symposium on Visual Languages VL'99, pp. 246-253.
- [Brun02] Brunner Natalie: „Hiphop und Foucault“ Diplomarbeit, Fakultät für Human- und Sozialwissenschaften der Universität Wien 2002
- [Burn99] Burnett, Margaret M.: Visual Programming. In: Encyclopedia of Electrical and Electronics Engineering (John G. Webster, ed.), John Wiley & Sons Inc., New York.1999
- [Ccnm06] http://ccnmtl.columbia.edu/projects/law/library/cases/case_grandwarner.html, letzter Zugriff: 15.12.2006
- [Crit06] <https://www.cs.tcd.ie/crite/people/KevinJennings.php>, letzter Zugriff: 14.12.2006
- [Debu06] Belzer Nikolaj „ESG, die Mütter des Beats“ in De:bug, Magazin für elektronische Lebensaspekte, Ausgabe 104, 2006
- [Dmxk99] Dmx Krew: „Hard Times“ von „We are DMX“, Rephlex 1999
- [Drum05] <http://www.bbc.co.uk/schools/communities/onionstreet/music/>, letzter Zugriff 23.10.2005
- [Erns90] Ernst Eva-Maria: „Die Onomatopoetik der Tierlaute und der entsprechenden Verben im Deutschen, Fränkischen und Italienischen“, Diplomarbeit Universität Wien 1990
- [FiGo84] Rehearsal World: Programming by Rehearsal
<http://www.acypher.com/wwid/Chapters/04Rehearsal.html>, letzter Zugriff: 11.1.2007
- [Free06] http://freesoftware.ircam.fr/article.php3?id_article=5, letzter Zugriff: 13.12.2006

- [Glbo05] <http://www.hitsquad.com/smm/programs/Gbloink!>, letzter Zugriff 20.10.2005
- [Gril05] Interview mit Thomas Grill, Wien 03.11.2005
- [Gola94] Golan Levin, Painterly Interfaces for audiovisual performances, Thesis 1994, MIT
- [GöRu05] Götze Marcel, Ruf Henning, Strothotte Thomas: "Interaktive Musikvisualisierung für Kinder" in C. Stary (Hrsg.): „Mensch & Computer 2005: Kunst und Wissenschaft – Grenzüberschreitungen der interaktiven ART“. München: Oldenbourg Verlag. 2005, S. 221-230
- [Hogg02] Hoggarth Janet: How to be a DJ, Penguin Books 2002
- [HuHo85] Hutchins Edwin, Hollan James, Norman Donald: "Direct Manipulation Interfaces" Human-Computer-Interaction Volume 1, Seite 311-338, 1985
- [Iwai06] http://ns05.iamas.ac.jp/~iwai/iwai_cv.html, letzter Zugriff: 31.1.2007
- [Jenn02] DrumSteps - A Constructionist Approach to Music Learning, 2002
- [Jenn05] Emailinterview mit Kevin Jennings, 07.11.2005
- [JeTa02] Jennings, K., Tangney, B. Drumsteps - A Percussion MicroEnvironment. Proceedings of the Ninth International Technological Directions in Music Learning Conference (TDML), San Antonio, Texas. (p. 55-64) Jänner 2002
- [Jones07] Emailinterview mit Phil Jones, 02.02.2007
- [KaSt98] Kalivorda Manfred, Steiner Johannes: Taschenbuch der Angewandten Psychoakustik, Springer Verlag 1998
- [KiHu99] Ross Kirk, Andy Hunt: Digital Sound Processing for Music and Multimedia, Focal Press 1999
- [Köni02] König Gerda: „Der letzte Werkelmann. Eine Wiener Type, die nie ganz verschwindet?“ Dissertation Universität Wien 2002
- [Kris95] Rainer Krispel, www.servus.at/hillinger/FM95.html
- [Laur93] Laurel Brenda „Computers as Theatre“ Addison-Wesley 1993
- [LaJo80] Lakoff George, Johnson Mark: Metaphors we live by, University of Chicago Press 1980
- [Less04] Lessig Lawrence: "free culture" penguin press 2004
- [MaLe01] Manovich Lev: "the language of new media" MIT Press 2001
- [MaSc00] Maar Oliver, Schaller-Pressler Gertraud: „Wie ein mechanisches Geburtstagsständchen entsteht“ in Haid Gerlinde, Hemetek Ursula;

- Pietsch, Rudolf (Hg.): Volksmusik - Wandel und Deutung. Festschrift Walter Deutsch zum 75. Geburtstag (Schriften zur Volksmusik 19). Wien: Böhlau, 2000
- [MeKr94] Metzger Wolfram, Kreiss Jakob: "Drehorgeln. Schaurig-schön" INFO-Verlagsgesellschaft Karlsruhe 1994
- [Mufr05] <http://wiki.the-mu.com>, letzter Zugriff 21.10.2005
- [MuSa03] Munindra Nath Das, Sanghmitra Gite: A Study of Visual Programming Languages, Final Project 2003
- [Nuck96] Nuckolls, Janis: „Sounds like life“, Oxford University Press 1996
- [Oluf06] Olufunmilayo B. Arewa: "From J.C. Bach to Hip Hop: Musical Borrowing, Copyright and Cultural Context" 2006
- [Piri01] Piringer, Jörg: Elektronische Musik und Interaktivität, Diplomarbeit Technische Universität, Institut für Gestaltungs- und Wirkungsforschung 2001
- [Pflü04] Pflüger Jörg: Writing Building Growing: Leitvorstellungen der Programmiergeschichte 2004 in: Hans Dieter Hellige (Hg.): "Geschichten der Informatik. Visionen, Paradigmen, Leitmotive", Springer Berlin 2004, S. 275-319
- [Puck97] Puckette, Miller: Pure Data: another integrated computer music environment, Proceedings, Second Intercollege Computer Music Concerts 07.05.1997
- [Puck03] Puckette, Miller: Theory and Techniques of Electronic Music, University of California, San Diego 2003
- [PuLi92] Miller Puckette, Eric Lindemann, and Cort Lippe: "The Ircam Signal Processing Workstation Prototyping Environment", in Proceedings of the ACM CHI 92 Human Factors in Computing Systems Conference, 1992
- [Oza00] Ozab, David: Beyond the Barline 06.05.2000
<http://www.atpm.com/6.05/barline.shtml>, letzter Zugriff 13.12.2006
- [Riaa07] www.riaa.com, letzter Zugriff: 18.1.2007
- [Rodw99] Rodwell, Tom: Desktop Absurdities - Revelatory absurdism and finesse in new music software 1999
<http://trace.ntu.ac.uk/frame3/articles/desktop.htm>, letzter Zugriff 20.10.2005

- [Roed77] Roederer Juan: Physikalische und psychoakustische Grundlagen der Musik, Springer Verlag 1977
- [Rose94] Rose Tricia: "Black Noise" 1994
- [Rusc98] Ruschkowski, André: Elektronische Klänge und musikalische Entdeckungen, Redam 1998
- [Scha02] Schanze, Helmut: Metzler Lexikon Medientheorie / Medienwissenschaften J.B. Verlag 2002
- [Schic79] Schick August: Schallwirkung aus psychologischer Sicht, Klett-Cotta 1979
- [Scho05] www.schott-cms.com/nocache/smi/produkte/multimedia/Wissen/musikalische_wuerfelspiele, letzter Zugriff 05.03.2005
- [Seid02] Seid, Hannes: "Verstärkung, Aufzeichnung und Synthese - Einfluss elektronischer Mittler und Klangerzeuger auf Live-Musik" Diplomarbeit Folkwang Hochschule Institut für Computermusik und elektronische Medien 2002
- [Shne83] Shneideman, Ben: Direct manipulation: a step beyond programming languages. Computer 16(8): 57-69, August 1983.
- [Steinb05] <http://www.steinberg.de>, letzter Zugriff 02.11.2005
- [Togn91] Tognazzini Bruce: Tog on Interface, Addison-Wesley 1991
- [Togn95] Tognazzini Bruce: Tog on Software Design, Addison-Wesley 1995
- [Tomi06] Tominski, Christian: Event-Based Visualization for User-Centered Visual Analysis. PhD thesis, 2006, Universität Rostock.
- [Witt05] Email-Interview mit Oliver Wittchov 24.10.2005
- [Witt06] Email-Interview mit Oliver Wittchov 06.06.2006
- [Zand98] Zander, Horst: Das PC-Tonstudio, Franzis-Verlag 1998
- [Yaho99] <http://web.archive.org/web/19990117061135/http://www.yahoo.com/index.html>, letzter Zugriff: 28.1.2007
- [Yout07] <http://www.youtube.com/watch?v=5SaFTm2bcac>, letzter Zugriff: 26.1.2007

Bildquellen

- Abbildung 1: <http://media.zzounds.com/media/brand,zzounds/mpc4000blue-3b88f32cef1631c07bbbfa89d434014.jpg>

- Abbildung 2: http://www.prosoundweb.com/recording/tapeop/edit/edit5_tapeloop.gif
- Abbildung 3: http://media.zzounds.com/media/brand,zzounds/ES-1MKII_fa3bf25f97d065722fbb593e4960a4a07.jpg
- Abbildung 4: <http://de.wikipedia.org/wiki/Bild:Piano-keyboard.jpg>
- Abbildung 5: http://de.wikipedia.org/wiki/Bild:Leon_Theremin_Playing_Theremin.jpg
- Abbildung 6: <http://skeptictank.net/toys/grinder.jpg>
- Abbildung 7: http://fr.audiofanzine.com/produits/tests/cakewalk_sonar2/PianoRoll_large.gif
- Abbildung 8: http://de.wikipedia.org/wiki/Bild:Roland_TR-909.jpg
- Abbildung 10: <http://www.cycling74.com/images/msp1.gif>, Zugriff 13.12.2006
- Abbildung 11: <http://freesoftware.ircam.fr/IMG/jpg/history-2.jpg>, Zugriff 13.12.2006
- Abbildung 12: Paul, Leonard J.: Audio Prototyping with Pure Data, 28.05.2003, <http://www.gamesutra.com>
- Abbildung 14: http://www.nanoloop.de/13seq_1.gif
- Abbildung 17: http://sparkletrain.blogspot.com/2006_01_01_sparkletrain_archive.html
- Abbildung 18: http://www.gamezone.de/reviews_detail.asp?gameid=23164&review=Electroplankton
- Abbildung 20: <http://en.wikipedia.org/wiki/Image:Pipedream.png>
- Abbildung 23: <http://de.wikipedia.org/wiki/Bild:FasttrackerII.png>
- Abbildung 24: <http://images.apple.com/ilife/garageband/features/images/podcastartworktrack20060118.jpg>