

Interaktive Musik in Computerspielen am Beispiel *DirectMusic*



Einleitung

Die rasante Entwicklung und Verbreitung der Computertechnik im privaten Bereich ist nicht zuletzt den Computerspielen zu verdanken. Was vor etwa 20 Jahren mit kruder Klötzchengrafik und einstimmigen Piepstönen begann, hat sich schnell zu einer gewaltigen Unterhaltungsindustrie entwickelt. Parallelen zur Filmbranche lassen sich heute nicht nur im Bezug auf die Höhe der verfügbaren Budgets für einzelne Projekte ziehen: Auch die verwendete Technik, sowie der Einsatz von dramaturgischen Mitteln zur Erzeugung einer dichten Atmosphäre, ist in beiden Medien oft sehr ähnlich.

Hierbei spielt die Musikuntermalung natürlich eine wesentliche Rolle. Durch den nichtlinearen Spielverlauf und die nicht vorhersagbaren Aktionen des Spielers ergibt sich jedoch - im Gegensatz zur fest definierten Filmhandlung - eine neue Problematik:

Die Musik muss permanent reagieren und sich dem Spielverlauf anpassen, also interaktiv sein. Mehrere Softwarefirmen haben daher in den letzten Jahren verschiedene, teils ähnliche Techniken entwickelt, Musik so nahtlos wie möglich in ihre Spiele einzubinden.

DirectMusic von Microsoft repräsentiert die zur Zeit ausgereifteste dieser ‚Music-Engines‘¹. Als fester Bestandteil des DirectX –Treiberpaketes (siehe Seite 11) ist DirectMusic auf jedem Spiele-tauglichen Windows-PC installiert, und wird ständig weiterentwickelt.

Ich möchte auf den nächsten Seiten eine gute Vorstellung davon vermitteln, was sich mit DirectMusic alles realisieren lässt, und welche Möglichkeiten sich im Einzelnen bieten, die Musik mit dem Spiel interagieren zu lassen. Die programmiertechnischen Details auf Quellcode-Ebene möchte ich dem Leser (und mir) hingegen ersparen, zumal sie einerseits den Rahmen dieser Arbeit sprengen würden, und andererseits auch aus tontechnischer Sicht nicht relevant sind.

Hilfreich zum Verständnis des Textes sind Grundlagen in Sachen MIDI-Sequencing, sowie eine ungefähre Vorstellung von den verschiedenen Spielegenres.

¹ Music-Engine: separater Teil einer Software, der die Musikausgabe steuert

Inhaltsverzeichnis

| | |
|---|----|
| Einleitung..... | 01 |
| Inhaltsverzeichnis..... | 02 |
| | |
| 1. Allgemeine Problematiken beim Vertonen von Computer- / Videospielen..... | 04 |
| 1.1. Interaktivität..... | 04 |
| 1.2. Variabilität..... | 05 |
| 1.3. Datenmenge..... | 05 |
| | |
| 2. Musik in PC-Spielen – Übersicht über die verfügbaren Techniken..... | 06 |
| 2.1. DirectMusic..... | 06 |
| 2.2. Tracker..... | 06 |
| 2.3. Audio Streaming..... | 07 |
| 2.4. General MIDI..... | 08 |
| 2.5. Redbook Audio..... | 08 |
| | |
| 3. Die Entwicklung der interaktiven Musik..... | 09 |
| 3.1. LucasArts iMuse..... | 09 |
| 3.2. Microsofts DirectMusic..... | 11 |
| 3.3. DirectMusic – Konzept und Technik..... | 11 |
| | |
| 4. Überblick über die DirectMusic Komponenten..... | 13 |
| | |
| 5. Die DirectMusic Komponenten im Detail..... | 14 |
| 5.01. Segment..... | 14 |
| 5.01.1. Sequence-basierte Segmente..... | 14 |
| Segment Tracks..... | 15 |
| 5.01.2. Wave-basierte Segmente..... | 16 |
| 5.01.3. Style-basierte Segmente..... | 16 |
| 5.02. Styles..... | 16 |
| 5.03. Patterns..... | 17 |
| 5.04. Groove Levels..... | 18 |
| 5.05. Motifs..... | 18 |
| 5.06. Embellishments..... | 19 |
| 5.07. Transitions..... | 20 |
| 5.08. Secondary Segments..... | 20 |

| | |
|---|----|
| 5.09. Chordmaps..... | 21 |
| 5.10. Signposts..... | 21 |
| 5.11. DLS Collections..... | 22 |
| 5.12. Toolgraphs..... | 23 |
| 5.13. Audiopaths..... | 24 |
| 6. Die Bedienung des DirectMusic Producers..... | 25 |
| 6.1. Project Tree..... | 25 |
| 6.2. Editor Fenster..... | 25 |
| 6.3. Piano Roll Editor..... | 26 |
| 6.4. Band Editor..... | 26 |
| 6.5. DLS Designer..... | 27 |
| 7. Interaktion zwischen Spiel und DirectMusic – ein praktisches Beispiel..... | 28 |
| 8. DirectMusic in der Praxis..... | 29 |
| 9. Interview mit Guy Whitmore..... | 32 |
| 10. Resümee..... | 35 |
| Anhang: Beispiele auf der Audio-CD..... | 36 |
| Quellenverzeichnis..... | 38 |

1. Allgemeine Problematiken beim Vertonen von Computer- / Videospielen

1.1. Interaktivität

„Die Musik macht 50% des Kinoerlebnisses aus“, so ein bekanntes Zitat des Starregisseurs George Lucas. Diese Aussage könnte auch auf viele Computerspiele zutreffen, die in ihrem Verlauf oft eine filmähnliche Handlung erzählen, und deren Reiz darauf beruht, den Spieler für einige Zeit in eine andere Welt zu entführen. Aus verschiedenen Gründen ist eine gelungene musikalische Vertonung solcher Spiele recht schwierig zu realisieren.

Während sich gute Filmmusik u.a. dadurch auszeichnet, dass sie durch den Aufbau kleiner und großer Spannungsbogen gezielt auf bestimmte Ereignisse hinarbeitet, kann die Musik eines Spiels grundsätzlich immer nur im Nachhinein auf die nicht vorhersagbaren Aktionen des Spielers reagieren. Trotz dieser Einschränkung kann ein gut gemachter interaktiver Soundtrack das Spielerlebnis enorm intensivieren. Er warnt den Spieler durch düstere Klänge vor gefährlichen Passagen, unterstützt ihn in Kampfszenen durch treibende Elemente, und feiert ihn nach Erfolgserlebnissen. Im besten Fall bekommt der Spieler das Gefühl, die Musik *weiß*, was gerade passiert. Sie kann einen essentiellen Kommunikationskanal zwischen Spiel und Spieler darstellen. Oft wird die Musik regelrecht als Status-Anzeige für spielerische Elemente benutzt, wo grafische Anzeigen nur unnatürlich und deplaziert wirken würden. So ändert sich in Spielen beispielsweise häufig die Musikuntermalung, sobald der Spieler von einem Gegner entdeckt wurde. Auf diese Weise lässt sich nicht nur die Dichte der Spielatmosphäre steigern, es werden auch unfaire Situationen, frustrierende Erlebnisse und das permanente Wiederholen von schlecht gemeisterten Passagen vermieden. Je vielschichtiger die Interaktion mit dem Spiel ist, desto tiefer muss die Musik-Engine in die eigentliche Klangerzeugung eingreifen. In letzter Konsequenz hieße das, die komplette Musik inkl. der Klangerzeugung einzelner Instrumente über Physical Modelling¹ während des Spiels ‚on-the-fly‘ generieren zu lassen. Bis die dafür notwendige Rechenleistung in durchschnittlichen PCs zur Verfügung steht, müssen wir uns wohl noch einige Jahre gedulden, zumal die Musik-Engine bei der Vergabe von Prozessor-Ressourcen eher stiefmütterlich behandelt wird. Die wenigsten Spieler machen sich Gedanken über die Notwendigkeit und den Nutzen interaktiver Musikelemente, und beurteilen die Musik zunächst nur nach deren Klangqualität - und hier liegt die Messlatte seit der Einführung der CD-ROM mit Redbook-Audiotracks (siehe Kapitel 2.5) bereits auf höchstem Niveau. Diese Technik der Musikwiedergabe bietet dabei aber auch den

¹ Syntheseverfahren, das auf mathematischen Algorithmen basiert, welche die Schwingung der Luftsäule in Resonanzkörpern (bzw. echten Musikinstrumenten) simulieren.

geringsten Grad der Interaktivität, und eignet sich somit fast ausschließlich für Spiele mit rundenbasiertem Ablauf, wie z.B. Beat'em Ups¹ oder Rennspiele. Dass sich auf diese Weise trotzdem auch in handlungsbasierten Actionspielen eindrucksvolle Ergebnisse erzielen lassen, bewies Valve Software mit dem im Jahr 1998 veröffentlichten Ego-Shooter „Half-Life“. Hier wurden einzelne Audio-Tracks auf der CD nur gezielt in Schlüsselstellen des Spielverlaufs getriggert, und dadurch perfekt ins Spielgeschehen eingebettet. Den Rest des Spiels über herrschte quasi angespannte Stille.

In den meisten Fällen lässt sich über diese Einschränkungen allerdings nicht so geschickt hinwegtäuschen, und Spiele-Designer müssen auf andere Techniken zurückgreifen, die fast zwangsläufig Einbußen bei der Klangqualität mit sich bringen. Welches Musiksystem für ein bestimmtes Spiel geeignet ist, muss bereits in dessen frühester Entwicklungsphase geklärt werden. Einen Überblick über die verfügbaren Techniken finden Sie in Kapitel 2.

1.2. Variabilität

Das zweite große Problem bei der musikalischen Untermalung von Computerspielen sind die permanenten Wiederholungen einzelner Passagen. Auch hier haben es schnelle, rundenbasierte Spiele wesentlich leichter als Spiele-Genres mit eher langwierigem Spielverlauf und großer Spieldauer. Als Extrembeispiel seien hier die Online-Rollenspiele genannt, die von passionierten Spielern mehrere Stunden am Tag gespielt werden, und das täglich und über viele Monate hinweg. Hier scheiden vorproduzierte Musikdateien von vornherein aus, zumal auch belastbare Spieler bald lieber komplett auf die Musik verzichten, als der hundertsten Wiederholung eines statischen Musikstücks noch länger zuhören zu müssen.

Variabilität und Interaktivität gehen Hand in Hand, zumal in beiden Fällen die Notwendigkeit besteht, auf möglichst tiefer Ebene in die Klangerzeugung einzugreifen. Entsprechend bringt ein hohes Maß an Interaktivität automatisch viel musikalische Abwechslung mit sich.

1.3 Datenmenge

Ein weiteres Problem, das bei großen Vollpreisspielen allerdings eher selten besteht, ist die begrenzte Größe des verfügbaren Speicherbereichs. Kleinere Werbespiele aber, die in erster Linie über das Internet vertrieben werden, müssen heutzutage auch noch für Modembesitzer innerhalb einer zumutbaren Zeitspanne herunterladen sein. Auch hier sind vorproduzierte Audiodateien aufgrund ihres Datenumfangs ungeeignet.

¹ Rundenbasierte „Prügelspiele“ wie z.B. ‚Street Fighter‘, ‚Tekken‘ oder ‚Mortal Kombat‘

2. Musik in PC-Spielen – Übersicht über die verfügbaren Techniken

2.1. DirectMusic

- Kombination aus Sample-basierter Klangerzeugung, MIDI-basierten Abspieldaten und synchronisierter Wiedergabe von Audiodateien

Interaktivität:

- Sprung zu beliebigen musikalischen Abschnitten während der Wiedergabe
- Stummschalten beliebiger Spuren
- Änderung der Instrumentierung
- Änderung des Tempos und der Tonart
- Generieren neuer Akkordverläufe
- Parameteränderungen innerhalb einzelner Instrumente (Öffnen des Filters etc.)

Klangqualität: Durchschnittlich bis sehr gut (abhängig von Auflösung und Qualität der verwendeten Samples / Audiotracks)

Datenumfang: kommt i.d.R. mit wenig Speicherplatz aus, sofern kein starker Gebrauch von Audiospuren (*Wave-Tracks*) gemacht wird

2.2. Tracker

Tracker sind Musik-Editoren, die Sample-basierte Audiodaten auf meist 32 Kanälen wiedergeben. Dabei kann jedes Sample in beliebiger Tonhöhe und Lautstärke abgespielt werden, und auch einfache Effekte wie Vibrato, Tremolo oder Portamento sind über spezielle Kommandos möglich. Das Arrangement wird aus Musikblöcken von je vier oder acht Takten zusammengesetzt, sogenannten Patterns, die jeweils alle Abspieldaten auf den kompletten 32 Spuren beinhalten. Diese Patterns können interaktiv vom Spiel während der Laufzeit angewählt werden, und bieten damit auch schon die einzige Möglichkeit der Interaktion.

Der derzeit bekannteste Tracker ist ‚FastTracker2‘, mit dem beispielsweise die Musik im 3D-Shooter ‚Unreal‘ (Epic, 1998) erstellt wurde.

Interaktivität: nur die Abspielreihenfolge unveränderlicher Musikblöcke wird passend zum Spielverlauf variiert.

Klangqualität: Durchschnittlich bis gut

Nachteile: recht spartanische Eingabe der Noten; oft sind zeitaufwendige Parameteränderungen und viele Tricks erforderlich, um einen lebendigen Klang zu erzeugen. Außerdem erfordert die Implementierung des Musikformats recht hohen Programmieraufwand.

Datenumfang: bei geschickter Sample-Organisation meist sehr gering

2.3. Audio Streaming

Als Audio Streaming bezeichnet man die Wiedergabe vorproduzierter Audiodateien, unter denen während des Spielverlaufs hin- und hergeblendet werden kann. Die einzelnen Blöcke können also in einem professionellen Studio unter Verwendung von live gespielten Instrumenten, Effekten und Mastering aufgenommen werden, und werden dann normalerweise in 16 Bit-Stereodateien mit einer Abtastrate von 22.050 kHz umkonvertiert. Diese Dateien werden während des Spiels entweder von der Spiele-CD-ROM oder der Festplatte eingelesen, und über einen Speicherpuffer abgespielt.

Interaktivität: gering; nur die Abspielreihenfolge und die Länge der Crossfades ist variabel

Klangqualität: gut bis sehr gut

Vorteil: dank komplett digitaler Musikblöcke kann jede musikalische Stilrichtung authentisch wiedergegeben werden

Nachteil: hoher Speicherbedarf, relativ unflexibel

Datenumfang: hoch

2.4. General MIDI

- Reine MIDI-Daten werden über die GM-Klangerzeugung der Soundkarte abgespielt.

Hierbei handelt es sich nicht um ein Audio-System im eigentlichen Sinne, sondern um die Art der Klangerzeugung. Diese Technik ist weitestgehend veraltet, und wird in der Regel nur noch in einfachen Shareware- und Freeware-Spielen eingesetzt.

Interaktivität: grundsätzlich sehr hoch, interaktive Elemente müssen jedoch erst in die eigentliche Wiedergabe-Routine / Musik-Engine implementiert werden.

Klangqualität: sehr schlecht bis gut, je nach verwendetem Klangerzeuger. Von einfachster FM-Synthese auf alten Soundkarten, bis hin zu professionellen GM-kompatiblen Expandern wie z.B. der Roland JV-Serie, kann alles zur Wiedergabe genutzt werden

Nachteile: die Musik klingt auf jedem System verschieden; Beschränkung auf wenige Standardsounds

Datenumfang: extrem gering

2.5. Redbook Audio

- Die Wiedergabe von Audiotracks auf der Spiele-CD.

Die exzellente Soundqualität wird mit der extremen Einschränkung in Sachen Interaktivität und Variabilität erkaufte.

Interaktivität: praktisch nicht vorhanden

Klangqualität: sehr gut

Vorteile: Es wird weder Platz auf der Festplatte beansprucht, noch Rechenleistung bei der Wiedergabe

Nachteile: Bei der Anwahl neuer Audiotracks legt der Windows-PC aus technischen Gründen eine ‚Gedenk-Sekunde‘ ein, in der das Bild kurz einfriert; extrem unflexibel

Datenumfang: sehr hoch

3. Die Entwicklung der interaktiven Musik

Erste Formen der interaktiven Musik lassen sich schon in den Spielautomaten der frühen 70er Jahre erkennen. So verbreitete der legendäre Automat „Space Invaders“ gepflegte Hektik durch vier Basstöne, die immer schneller gespielt wurden, je näher die feindliche Raumflotte dem Geschütz des Spielers kam. „Asteroids“ setzte auf den gleichen Effekt, kam aber mit lediglich zwei Basstönen aus. Schon hier wurde die ‚Musik‘ als Anzeige für ein spielerisches Element benutzt, denn man konnte anhand der Basstöne gut einschätzen, wie viel Zeit noch bis zum Erscheinen des nächsten feindlichen Raumschiffs blieb.

Als es in den 80er Jahren technisch möglich wurde, digitale Klangdateien in Videospielen wiederzugeben, wurde die Musik als interaktives Element stark vernachlässigt. Die Hersteller waren darauf bedacht, realistische Musik, wie man sie aus dem Radio kannte, in ihre Spiele zu integrieren, und nahmen wenig Rücksicht auf deren dramaturgische Wirkung. Lange Zeit fehlten außerdem die softwaretechnischen Voraussetzungen, interaktive Musik zu realisieren.

3.1. LucasArts' iMuse

Eines der ersten Softwarehäuser, das ein umfassendes Konzept zur Realisierung interaktiver Musik entwickelte, war LucasArts mit der im Jahr 1991 patentierten „iMuse“-Engine („interactive Music Streaming Engine“). Sie kam erstmalig bei dem 2D-Adventure „The Secret of Monkey Island 2“ zum Einsatz, wurde seitdem ständig verbessert und in fast sämtlichen Veröffentlichungen des Softwarehauses verwendet. Das Grundprinzip besteht aus einer Datenbank mit vorgefertigten Musik-Sequenzen in einem abgewandelten MIDI-Format, das um mehrere sog. *Decision Points* erweitert wurde: Dabei handelte es sich um Markierungen innerhalb der Sequenzen, an denen ein Abzweigen zu einer anderen Sequenz, bzw. einem komplett anderen musikalischen Abschnitt möglich wird. Zwei Programmteile regeln dabei die Wiedergabe während des Spiels: Der *Sound Driver*, der für das eigentliche Abspielen der Musikdaten und Interpretieren der verschiedenen Decision Points zuständig ist, und das „*Directing System*“, das als zentraler Programmteil Aktionen des Spielers und Veränderungen des Spielverlaufs an den Sound Driver weitergibt. Umgekehrt ist es auch dem Directing System jederzeit möglich, den Status des Sound Drivers abzufragen, um seine eigenen Aktivitäten entsprechend angleichen zu können.

Mit der ersten iMuse-Version ergaben sich folgende Möglichkeiten:

- Das Abzweigen während der Musikwiedergabe zu vordefinierten Punkten

- Das Transponieren einzelner Stimmen
- Das An- und Ausschalten einzelner Stimmen
- Das Ändern der Lautstärke einzelner Stimmen
- Das Wechseln des Wiedergabe-Instruments einzelner Stimmen
- Das Ändern des Tempos während der Wiedergabe

Durch Kombination dieser Techniken konnte zwischen verschiedenen Spielszenen nahtlos übergeblendet werden, ohne Dissonanzen oder abrupte Tempowechsel zu verursachen. Auch die Intensität der Musik ließ sich so gut dem Spielverlauf angleichen. Michael Land, bis heute verantwortlich für die Musik in fast sämtlichen LucasArts-Projekten, nutzte die Möglichkeiten von iMuse mit viel Liebe zum Detail. Im Stil aufwendiger Filmscores komponierte er eigene Themen für einzelne Charaktere im Spiel, die sich subtil in die Hintergrundmusik der jeweiligen Szene einflochten. Zu einigen Charakteren existieren innerhalb eines Spiels teilweise über zehn verschiedene Variationen ihres Themas.

Da die Klangerzeugung jedoch ausschließlich über die General MIDI-Klänge der jeweiligen Soundkarte erfolgte, klang das Ergebnis für heutige Verhältnisse wenig beeindruckend. Zu dieser Zeit arbeiteten die handelsüblichen Soundkarten mit einer sehr einfachen FM-Synthese, die später weitestgehend durch digitale GM-Sounds ersetzt wurden, die im sog. ‚Wavetable‘-ROM der Soundkarten gespeichert sind. Dadurch verbesserte sich zwar die Klangqualität, die Musik klang jedoch von Soundkarte zu Soundkarte teils stark verschieden. Das größte Manko blieb aber zunächst die Beschränkung auf die GM-Klänge an sich - Das Einbinden von digitalen Klängen war mit iMuse anfänglich nicht möglich. Mittlerweile wurde iMuse jedoch stark erweitert, und basiert in aktuelleren Spielen auf dem Übereinanderschichten von digitalen Audio-Loops.

3.2. Microsoft's DirectMusic

Mitte der 90er Jahre begann Microsoft mit der Entwicklung einer ähnlichen Engine, der „*Microsoft Interactive Music Architecture*“ (IMA), die ursprünglich nur intern für eigene Projekte genutzt wurde. IMA wiederum basierte auf einer eingekauften Technologie namens „AudioActive“, die von „Blue Ribbon Software“ in Atlanta entwickelt wurde. Das Herzstück bestand aus einer Routine, die ein bestehendes Arrangement in seine Bestandteile zerlegen, und nach bestimmten Kriterien wieder zusammensetzen konnte. Die Entwickler von Blue Ribbon Software (übrigens bekannt durch ihren legendären Amiga-Sequencer „Bars ‚n‘ Pipes“), hatten dazu eingehend die Improvisationen echter Live-Bands studiert.

Bald entschied sich Microsoft, diese Technik öffentlich zugänglich zu machen, und nahm IMA in weiterentwickelter Form unter dem Namen *DirectMusic* als festen Bestandteil in das DirectX - Treiberpaket auf. [siehe unten]

Die hochgesteckten Entwicklungsziele formuliert der ehemalige DirectMusic Program Manager bei Microsoft, Bryan Schmidt, so: „Unser heiliger Gral: Den Entwicklern zu ermöglichen, Inhalte mit so hoher Produktionsqualität zu kreieren, die so wirken, als sei die Musik [...] post-production komponiert worden, obwohl sie gerade in Echtzeit und in Abhängigkeit zu den Aktionen des Spielers generiert wird. Es ist ein Ziel auf lange Sicht, das der Kombination von cleverer Software, Änderungen in Entwicklungsprozeduren, der [...] Kreativität von Drittanbietern, und in einigen Fällen auch roher Prozessorleistung bedarf. Wir haben ungefähr 150 Meilen unserer 1000 Meilen Reise hinter uns.“ [1]

Direct X

DirectX stellt eine Treiber-Schnittstelle zwischen modernen Grafik- und Soundkarten und dem eigentlichen Windows-Betriebssystem dar. Neben der DirectMusic-Komponente existierte bis zur DirectX-Version 8.0 auch das separate Modul „DirectSound“, das für die Ausgabe und das interne Mischen digitaler Audiodaten zuständig war. Ab Version 8.0 wurden die beiden Komponenten zu einem einzigen Modul namens „DirectAudio“ zusammengefasst.

3.3. DirectMusic – Konzept und Technik

Neben umfassender Interaktivität war das zweite Hauptziel die Realisierung von Variabilität: Auch ohne aktive Parameter-Änderungen sollten bei der Wiedergabe exakte Wiederholungen eines musikalischen Abschnitts vermieden werden. Außerdem wollte man die Beschränkung auf den GM-Standard aufheben, und es sollte möglich werden, eigene digitale Klänge zu implementieren. Damit würde auch für einen homogenen Klang auf allen Systemen gesorgt, unabhängig von der Soundkarte und deren Klangerzeugung. Dieses Vorhaben realisierte Microsoft mit der Einführung der *Downloadable Sounds* (DLS), die in Kapitel 5.11. ausführlich besprochen werden.

Als weiteres Element wurde ein Prinzip auf die Musik übertragen, das im Bereich der Software-Entwicklung als „Objektorientiertes Programmieren“ schon lange unverzichtbar ist: Bereits fertiggestellte musikalische Komponenten sollten ohne großen Aufwand in anderen Projekten wiederverwendet werden können. In sofern war es unter anderem notwendig, die Noteninformationen nicht als absolute Werte in einer bestimmten Tonart, sondern relativ zu Tonart und Akkordfolgen zu definieren. Dieses Vorhaben wurde mit

dem Konzept der „Styles“ (siehe Kapitel 5.2.) verwirklicht. Durch dieses Prinzip ergaben sich auch weitere umfassende Möglichkeiten der Variabilität und Interaktivität. So kann bei Wiederholungen ein alternativer Harmonieverlauf per Zufall ausgewählt, oder sogar nach bestimmten Vorgaben neu generiert werden. Diese Funktionen werden in Kapitel 5.8 und 5.9 (Chordmaps, Signposts) näher erläutert.

Das Autorenprogramm, mit dem die musikalischen Elemente erstellt und aufbereitet werden, ist der *DirectMusic Producer* (DMP). Er steht kostenfrei auf der Microsoft-Website unter folgendem Link zum Download bereit:

<http://msdn.microsoft.com/downloads/default.asp?URL=/code/topic.asp?URL=/msdn-files/028/000/114/topic.xml>

4. Überblick über die einzelnen DirectMusic Komponenten

| | |
|-----------------------|--|
| Segment | stellt den Rahmen dar, in dem alle weiteren Elemente auf einer Zeitachse angeordnet und kombiniert werden. |
| Style | Beinhaltet eine Sammlung rhythmischer und melodischer <i>Patterns</i> , <i>Motifs</i> und <i>Embellishments</i> , sowie eigene <i>Bands</i> . |
| Pattern | Musikabschnitt, dessen Noten sich bei der Wiedergabe automatisch den Akkorden des Songs anpassen. Ein Pattern beinhaltet individuelle <i>Parts</i> , die wiederum jeweils bis zu 32 Variationen enthalten können. |
| Motif | ein spezielles Pattern, das interaktiv über die restliche Musik gelayert werden kann. |
| Embellishment | Spezielles Intro-, End-, Break- oder Fill-Pattern |
| Band | Definition der Instrumentierung inkl. Panorama und Lautstärke-Verhältnisse |
| DLS Collection | Eine Zusammenstellung digitaler Klänge, die auch deren Abspielparameter beinhaltet |
| Sequence | enthält feste, nichtvariable MIDI-Daten , die einem einzelnen Wiedergabe-Kanal (PChannel) zugeordnet sind. |
| Chordmap | ermöglicht variable Akkordfolgen |
| Wave | gesampelte Audiodaten, die als Wave-Track innerhalb eines Segments wiedergegeben, oder in eine DLS Collection importiert werden können. |
| Audiopath | Beinhaltet Instruktionen zur Wiedergabe von Synthesizern mit Audio-Effekten |
| Script | Ermöglicht das Einbinden von DirectMusic-Anweisungen auf Programmier-Ebene |
| Toolgraph | Ermöglicht MIDI-basierte Effekte |
| PChannel | <i>Performance Channel</i> . Entspricht einem Wiedergabekanal, dem eine oder mehrere Sequenzen oder Pattern zugewiesen sind. Das Wiedergabe-Instrument eines PChannels wird über die jeweils aktive <i>Band</i> definiert. |

5. Die DirectMusic Komponenten im Detail

5.01. Segment

Ein Segment ist die Basiseinheit eines DirectMusic - Projekts. Es stellt einen Rahmen für alle Elemente dar, die darin kombiniert abgespielt werden. Der Programmteil des DirectMusic Producers, mit dem sich Segmente erstellen und editieren lassen, ist der *Segment Designer*. Optisch ähnelt er dem Arrangement-Fenster herkömmlicher linearer Software-Sequenzers, und auch dessen Funktion ist ähnlich: Verschiedene Arten von Tracks sind untereinander auf einer horizontalen Zeitachse angeordnet. Statische MIDI-Spuren entsprechen hier den Sequence-Tracks, und das Pendant zu Audio-Spuren wird als *Wave-Tracks* bezeichnet. Entsprechend der DirectMusic Architektur existieren jedoch noch weitere Tracks, die in der Übersicht auf der nächsten Seite aufgelistet sind. Neben der Länge des Segments (Anzahl der Takte), können für jedes seiner Elemente Parameter definiert werden, wie z.B. Tempo, Start- und Stopzeit, Anzahl der Wiederholungen und Austausch der Instrumente.

Die Segmente in DirectMusic sind so aufgebaut, dass sie musikalischen Abschnitten wie z.B. Bridge, Chorus, Refrain etc. entsprechen, in die sich fast jede Musik grob einteilen lässt. Für jeden dieser Abschnitte sollte also ein eigenes Segment erstellt werden. Ein solches ist in der Regel etwa 16 Takte lang (wobei technisch bis zu 999 Takte möglich sind), und kann ständig wiederholt, und dabei mit verschiedenen Wiedergabe-Parametern beeinflusst werden.

Es gibt drei grundlegende Arten von Segmenten: Style-basiert, Wave-basiert und Sequence-basiert. Die drei Varianten können auch miteinander kombiniert werden, indem man beispielsweise in ein bestehendes Style-basiertes Segment eine Sequenz oder einen Wave Track einfügt.

5.01.1. Sequence-basierte Segmente entsprechen in ihrer Funktion weitestgehend einfachen MIDI-Arrangements, und haben nur geringe Möglichkeiten der Interaktivität und Variabilität. Fertige Songs im MIDI-Format lassen sich als Sequence-basierte Segmente direkt in den DirectMusic Producer importieren, um sie danach mit anderen Elementen zu kombinieren, oder sie mit grundlegende Funktionen der Interaktivität auszustatten. Beispielsweise können Änderungen des Tempos und der Instrumentierung, sowie Wiederholungen eingebaut werden.

Segment - Tracks

| | |
|---|--|
| Band Track | Enthält eine beliebige Anzahl von Bands, die nur für das Segment erstellt wurden, oder aus Styles oder MIDI-Files importiert wurden |
| Chord Track | Enthält Akkorde, die an beliebigen Stellen im Song eingefügt werden können. Nach diesen Akkorden richtet sich die Tonhöhe der eingebundenen Styles und Patterns |
| Chordmap Track | Enthält ein Chordmap-File, um variable Akkordfolgen zu generieren. |
| Groove & Embellishment Track | Enthält Anweisungen, den Intensitätsgrad (=„Groove Level“) der Musik zu ändern. Hier werden auch die verschiedenen Embellishments eines Styles angetriggert. |
| Lyrics Notification Track | Enthält Text, der über eine DirectMusic Applikation ausgelesen werden kann, und z.B. Songtexte oder Spielanweisungen für Live-Perfomances beinhalten kann. |
| Marker Track | Enthält Markierungen für geeignete Positionen innerhalb des Segments, an denen die Wiedergabe beendet, ein <i>Secondary Segment</i> gestartet, oder zu einem anderen Segment übergeblendet werden kann. |
| Mute Track | Enthält Mute-Kommandos für jeden beliebigen Ausgabe-Kanal (<i>PChannel</i>). |
| Pattern Track | Enthält individuelle Patterns innerhalb eines Segments. Jedes Pattern ist eine Sequenz, die aus mehreren Variationen besteht, und in Abhängigkeit zu der im Chord Track vorgegeben Tonart transponiert wird. |
| Parameter Control Track | Enthält Daten, die das Timing und die Lautstärkeverhältnisse von Audio- Effekten regeln, die in „ <i>Audiopath</i> “-Dateien enthalten sind |
| Script Track | Enthält ein oder mehrere programmierte Script-Kommandos, welche die Wiedergabe der Performance steuern |
| Segment Trigger Track | Enthält Startbefehle für Segmente oder <i>Motifs</i> zu exakt definierbaren Zeitpunkten. |
| Sequence Track | Enthält eine Sequenz statischer MIDI-Abspieldaten. |
| Signpost Track | Werden in Segmenten nur zusammen mit <i>Chordmaps</i> benutzt |
| Style Track | Enthält einen oder mehrere Styles, die aus musikalischen Mustern, Rhythmiken und Melodien bestehen, die einem bestimmten musikalischen Genre bzw. einer bestimmten Spielumgebung entsprechen |
| Tempo Track | Enthält eine oder mehrere Tempo-Angaben |
| Wave Track | Enthält eine oder mehrere Wave-Dateien |

5.01.2. Wave-basierte Segmente bestehen aus einem oder mehreren Wavetracks, die wie in einem HD-Recording-System synchron abgespielt werden. Mit jeder Audiosequenz können bis zu 32 verschiedene Variationen gleicher Länge eingebunden werden, unter denen dann bei jeder Wiederholung per Zufall ausgewählt wird.

Reine Wave-basierte Segmente sind sehr unflexibel. Tempo und Tonhöhe sind nicht veränderlich, die Musik lässt sich nur durch das Ändern der Abspielreihenfolge und das Ein- und Ausschalten einzelner Tracks variieren.

5.01.3. Style-basierte Segmente

Sie bieten die umfassendsten Eingriffsmöglichkeiten zur Realisierung interaktiver Musik. Das Segment gibt hier nur die allgemeine Struktur des Songs vor, wie z.B. Tempo, Taktart, Tonart, Akkordfolgen und Grad der musikalischen Intensität. Die eigentlichen Noten sind jedoch in den *Styles* gespeichert (siehe Kapitel 5.02). Der gewünschte Style wird innerhalb des *Style Tracks* angewählt, und kann an jeder beliebigen Position geändert werden.

In erster Linie werden Style-basierte Segmente dazu benutzt, diese Styles mit Akkorden zu kombinieren, um eine musikalische Phrase mit einer bestimmten Länge zu erzeugen. Ein solches Segment enthält deshalb einen *Chord Track*, in dem an jeder beliebigen Position neue Akkorde eingefügt werden können. Während der Wiedergabe des Segments gleichen sich die Tonhöhen der Style-Sequenzen automatisch diesen Akkorden entsprechend ihrer musikalischen Funktion an. Beachten Sie hierzu bitte auch Hörbeispiel 2 auf der Audio-CD (siehe Anhang Seite 36).

Das zweite wichtige Element der Style-basierten Segmente ist der *Groove Track*. Mit ihm lässt sich über das Einfügen verschiedener „*Groove Level*“-Werte die Intensität der Musik beeinflussen. Die genaue Funktion der Groove Levels wird in Kapitel 5.04. ausführlich behandelt.

Beim Erstellen eines Style-basierten Segments wird automatisch ein *Band Track* erzeugt. In *Bands* sind die Instrumentierung, die Lautstärkenverhältnisse und die Panorama-Verteilung der einzelnen Wiedergabe-Kanäle definiert. Solange keine anderen *Bands* in den Band Track eingefügt werden, wird die innerhalb des Styles gespeicherte Band ins Segment übernommen.

5.02. Styles

Über Styles können die Klangeigenschaften und die Spielweisen ganzer Musikgenres definiert werden. In ihnen sind relative Noteninformationen, sowie die Rhythmik der einzelnen musikalischen Sequenzen enthalten. So kann beispielsweise ein eigener Style

für Funk, Latin oder Rock erstellt werden. Bei Computerspielen wird auch oft die Musik für eine bestimmte Spiel-Ebene („Level“) zu einem Style zusammengefasst.

Ein Style kann mehrere hundert melodische Phrasen und rhythmische Figuren enthalten, die den traditionellen Spielweisen eines bestimmten Musikgenres entsprechen. Optional können innerhalb eines Styles auch eine oder mehrere *Bands* gespeichert werden, in denen die typische Instrumentierung dieses Musikgenres definiert ist. Die eigentlichen Notendaten innerhalb eines Styles sind wiederum in den sogenannten *Patterns* enthalten, die innerhalb eines Styles nach dem Grad ihrer musikalischen Intensität organisiert und zusammengefasst werden.

5.03. Patterns

Patterns werden im DirectMusic Producer über den „Pattern Editor“ bearbeitet. Ähnlich wie im Segment Designer sind hier mehrere Spuren übereinander auf einer horizontalen Zeitachse angeordnet. Diese Spuren entsprechen festen Wiedergabekanälen, die im DMP als *Performance Channels*, oder kurz „PChannels“ bezeichnet werden. Über welches Instrument die Daten eines solchen PChannels letztendlich ausgegeben werden, richtet sich nach der jeweils aktiven Band.

Die MIDI-Daten, die einem einzelnen PChannel innerhalb eines Patterns zugewiesen werden, sind in sogenannten *Parts* zusammengefasst. Parts entsprechen prinzipiell herkömmlichen MIDI-Sequenzen. Der Hauptunterschied besteht jedoch darin, dass sich die Noten eines Parts immer nach den jeweiligen Akkorden und der Tonart des Segments richten, indem sie während der Wiedergabe automatisch entsprechend ihrer musikalischen Funktion transponiert werden. Zu diesem Zweck enthält ein Pattern immer einen Chord Track, in dem der Akkordverlauf und die Tonart des Patterns definiert sind. Der Chord Track eines Patterns bildet also nur eine Referenz, auf die ein Segment während der Wiedergabe zurückgreift. Wird das Pattern direkt abgespielt, hat er keinerlei Einfluss auf die Tonhöhe.

Der zweite Unterschied zwischen Parts und einfachen MIDI-Sequenzen ist, dass ein Part nicht nur aus jeweils einer einzelnen Sequenz bestehen muss, sondern bis zu 32 Variationen beinhalten kann. Diese „Variations“ sind abgewandelte musikalische Phrasen mit der gleichen Länge und einer ähnlichen musikalischen Funktion. Ihr Zweck ist es, bei Wiederholungen für Abwechslung zu sorgen. Existieren innerhalb eines Parts mehrere Variations, wählt die DirectMusic-Engine in jedem Durchgang per Zufall aus, welche gespielt wird. Um dabei nicht völlige Willkür walten zu lassen, kann jede Variation in einer Matrix für beliebige Akkorde gesperrt werden.

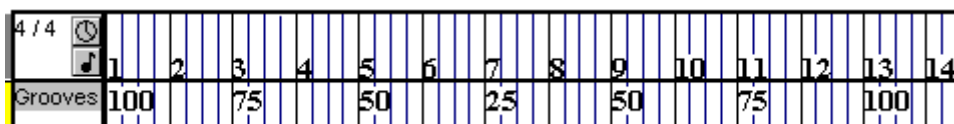
Ein Pattern muss nicht immer zwangsläufig Bestandteil eines Styles sein, sondern kann über Pattern Tracks auch direkt in ein Segment eingefügt werden. Auf diese Weise

beraubt man sich jedoch der umfangreichen Möglichkeiten, die sich mit Styles über die *Groove Levels* eröffnen.

5.04. Groove Levels

Patterns lassen sich über Styles nach dem Grad ihrer musikalischen Intensität anordnen, der in DirectMusic als „Groove Level“ bezeichnet wird. Ein Pattern mit einem hohen Intensitätsgrad wird eher viele rhythmische Elemente, komplexe melodische Muster und mehrere Instrumente beinhalten, die gleichzeitig spielen. Dies entspräche einem hohen Groove Level. Ein weniger aktives Pattern mit einem niedrigen Groove Level hat dementsprechend weniger Instrumente und eher einfache rhythmische Sequenzen. Nachdem man ein Pattern kreiert hat, kann man diesem einen Groove Level - Bereich zwischen den Werten 1 und 100 zuweisen. In der Praxis werden die Patterns meistens so organisiert, dass hierbei der Wert 1 der niedrigsten, und 100 der höchsten Intensität entspricht. Ebenso könnte man aber auch den Wert 20 dem Pattern mit der maximalen Intensität zuweisen, oder jeden beliebigen Wert jeder beliebigen Intensität. Der Groove Level ist also nur ein Indikator und ein Steuerelement für den Benutzer, hat aber so gesehen keinen eigentlichen Einfluss auf die Intensität die Musik.

Die Groove Levels werden im Groove Track eines Segments als einfache Zahlenwerte an beliebigen zeitlichen Positionen eingefügt. Bei jeder Änderung schaltet die DirectMusic-Engine auf das Pattern um, in dessen Gültigkeitsbereich („Groove Range“) der aktuelle Groove Level fällt. Kommen mehrere Patterns in Frage, entscheidet der Zufall, welches gespielt wird.



Groove Track eines Segments mit Groove Level Verlauf über 14 Takte

5.05. Motifs

Ein Motif ist ein spezielles Pattern mit einer besonderen Funktion. Es ist eine meist kurze melodische oder rhythmische Figur, oder ein kurzer Sound, der über die restliche Musik gelegt wird. Ein Motif kann zu jeder Zeit während einer DirectMusic-Performance gespielt werden, und verhält sich wie ein instrumentales ‚Solo‘, das spielt, ohne die laufende Musik zu unterbrechen. Über das „Properties“-Fenster eines Motifs lassen sich dessen Wiedergabe-Eigenschaften einstellen, wie z.B. die Anzahl der gewünschten Wiederholungen.

Motifs können entweder von direkten Aktionen des Benutzers getriggert werden, wie z.B. das Drücken einer Taste, das Bewegen der Maus usw., oder von bestimmten Ereignissen innerhalb eines Spiels, wie z.B. dem Erscheinen eines bestimmten Spiel-Charakters. Bereits spielende Patterns werden von Motifs nicht unterbrochen, und auch das simultane Abspielen mehrerer Motifs ist möglich. Das Abspielen kann sofort und ohne Rücksicht auf die aktuelle Songposition erfolgen, oder wahlweise erst auf einen bestimmten Schlag oder ab einem neuen Takt. In der Regel beträgt die Länge eines Motifs nicht mehr als einen Takt. Ein Pattern lässt sich jederzeit in ein Motif umwandeln, und umgekehrt. Auch Motifs werden bei der Wiedergabe innerhalb eines Segments passend zu dessen Akkordverlauf transponiert.

5.06. Embellishments

Eine weitere spezielle Variante der Patterns sind die Embellishments. Auch sie sind normalerweise eher kurz, und übernehmen jeweils eine der folgenden fünf Sonderfunktionen innerhalb eines Styles:

| | |
|---------------|--|
| Intro | Eine kurze Einführungsphrase, mit dem ein Abschnitt beginnen soll |
| Fill | Kurze Übergangs-Sequenz zwischen zwei musikalischen Phrasen |
| Break | Kurzer Einwurf, in dem die Rhythmusbegleitung pausiert, und einzelne Instrumente hörbar werden |
| End | Kurze Abschlussphrase, um ein Stück zu beenden |
| Custom | Frei benennbare Zwischenpassage mit einer beliebigen Funktion |

Die Custom-Embellishments werden als unterstützende Patterns benutzt, um Akzente innerhalb eines Styles zu setzen, und um im dynamischen Arrangement für mehr Abwechslung zu sorgen. Nachdem Embellishments in einem Style kreiert wurden, reagieren diese auf die Intro-, Fill, Break- und End-Kommandos, die im Groove / Embellishment-Track eines Segments platziert wurden. Den Custom-Embellishments wiederum sind spezielle Groove Level Werte zugewiesen, die oberhalb des regulären Bereichs liegen, also ab 100 aufwärts. Sie müssen daher im Groove Track explizit als Embellishments angetriggert werden, und stehen in der regulären Pattern-Auswahl nicht zur Verfügung.

Embellishments werden oft als *Transition-Patterns* genutzt. Dabei handelt es sich um ausgewählte Patterns, die dafür geeignet sind, einen musikalischen Übergang zwischen zwei Segmenten zu herzustellen.

5.07. Transitions

Als „Transitions“ werden in DirectMusic die Übergänge zwischen zwei Komponenten bezeichnet. Bei den Komponenten muss es sich entweder um Segmente, Patterns oder um Motifs handeln, da alle diese Elemente eine eigene Zeitachse haben, ohne die ein solcher Übergang nicht möglich ist. Wie abrupt oder wie fließend ein solcher Übergang erfolgen soll, lässt sich über zahlreiche Parameter definieren. Fließende Übergänge erstrecken sich über einen oder mehrere Takte, und können dabei auf Akkorde, Patterns und / oder Embellishments von Quell- und Zielkomponente zurückgreifen. In diesem Fall erzeugt DirectMusic ein eigenes „Transition Segment“, das zwischen den beiden Komponenten abgespielt wird. Bei abrupten Übergängen geschieht der Wechsel ohne diesen Zwischenschritt. Auf Wunsch kann die Transition aus einer dritten Komponente generiert werden, wie z.B. einem speziellen Pattern oder einem Segment.

5.08. Secondary Segments

Die DirectMusic-Engine kann nicht nur einzelne Segmente nacheinander, sondern bis zu 100 simultan abspielen. Die Musik einer DirectMusic Performance kann also aus vielen separaten Segmenten bestehen, die während der Wiedergabe übereinandergelegt werden. Die Segmente können zu jeder Zeit, über jede Dauer, in jeder beliebigen Reihenfolge und synchron abgespielt werden.

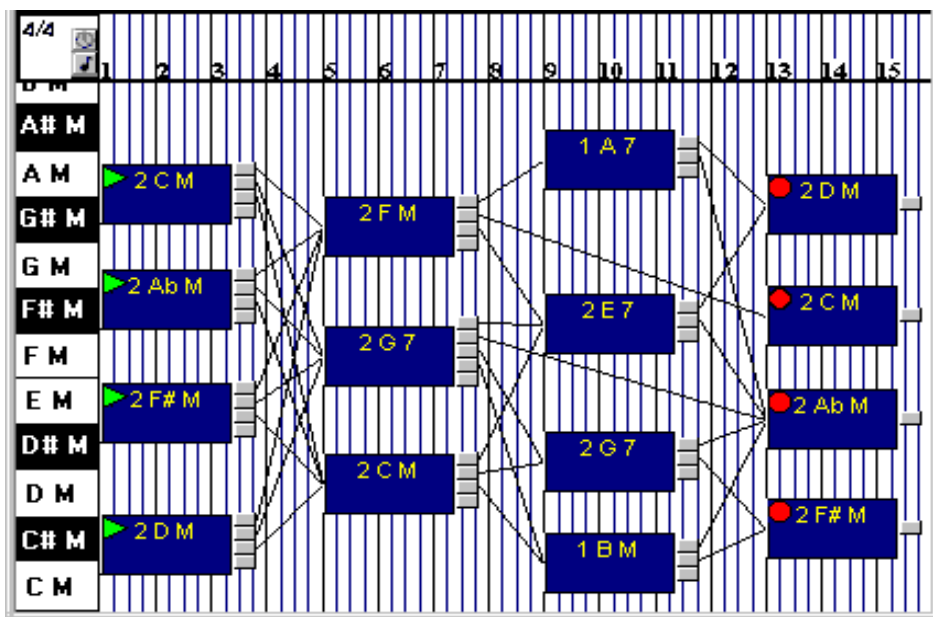
Man kann ein Segment als ‚primary‘ deklarieren, und alle anderen, die über dieses Basis-Segment gelegt werden sollen, als ‚secondary‘. Ein primary Segment stellt meistens das Grundgerüst eines Musikstücks dar, während beliebig viele secondary Segments als Ergänzung dazuaddiert werden. Das primary Segment könnte also beispielsweise Rhythmus- Parts beinhalten, während die secondary Segments für instrumentale Sequenzen sorgen, die interaktiv über die Rhythmus Sektion gelegt werden. Die Länge von secondary Segments ist technisch unbegrenzt, sie sind aber in der Praxis normalerweise nur wenige Takte lang.

Secondary Segments lassen sich in ihrem Eigenschafts-Fenster so einrichten, dass ihr Einsatz rhythmisch und harmonisch auf das primary Segment abgestimmt ist. Um mit verschiedenen DirectMusic Komponenten als secondary Segments zu experimentieren, existieren im DMP mehrere ‚Secondary Transport Controls‘, die beliebigen Segmenten zugewiesen werden können, um diese auf Mausklick abzuspielen. Bei den secondary Segments muss es sich entweder um Segmente oder um Motifs handeln.

5.09. Chordmaps

Die einfachste Art, den Harmonieverlauf eines Segments zu definieren, ist der Chord Track, in dem immer nur eine einzige fest definierte Akkordfolge abgelegt ist. Im Hinblick auf Variabilität und Interaktivität bietet sich dazu jedoch eine interessante Alternative: die *Chordmaps*. Dabei handelt es sich um separate Dateien, in denen eine oder mehrere Akkordfolgen definiert sind, die später über den *Chordmap Track* in ein Segment eingebunden werden können. In einer Chordmap sind typischerweise mehrere Harmonieverläufe eingezeichnet, unter denen DirectMusic bei der Wiedergabe per Zufall auswählt. Im Chordmap-Designer können beliebig viele Akkorde eingefügt, und anschließend per ‚drag-and-drop‘ miteinander verbunden werden. Auf diese Weise lassen sich viele verschiedene alternative ‚Routen‘ für den Harmonieverlauf eines Segments einzeichnen.

Innerhalb des DMPs kann über den „Compose Chords“- Button jederzeit während des laufenden Songs ein neuer Akkordverlauf ausgewählt werden. Diese Funktion kann auch ein Spiel jederzeit auslösen, um beispielsweise nach einem Szenenwechsel für musikalische Abwechslung zu sorgen.



Chordmap Designer

5.10. Signposts

Variable Akkordfolgen lassen sich noch auf eine andere Weise erzeugen. Statt die erstellten Akkorde direkt miteinander zu verknüpfen, lassen sie sich auch in Gruppen mit einer ähnlichen musikalischen Funktion zusammenfassen. Innerhalb eines Segments werden dann auf der Zeitachse nur die Punkte definiert, an denen ein Akkordwechsel stattfinden soll, sowie die gewünschte Gruppe, der ein Akkord zugeordnet sein soll. Dies

geschieht über die sogenannten *Signposts* und den entsprechenden *Signpost Track*. Die „Compose Chords“-Funktion arbeitet hier prinzipiell genauso, und wählt bei jedem Aufruf zu jedem Signpost einen anderen Akkord aus der zugeordneten Gruppe aus. Um für Homogenität zu sorgen, können einmal generierte Akkordfolgen auf Wunsch bei Wiederholungen beibehalten werden.

5.11. DLS - Collections

Ein „Downloadable Sound“ (DLS) ist eine digitale Audiodatei, die über den „DLS Designer“ importiert und mit verschiedenen Abspielparametern versehen wurde. Prinzipiell stellt das DLS-Konzept einen vollwertigen, wenn auch relativ einfach strukturierten Software-Sampler dar. Mehrere DLS-Instrumente können in *DLS-Collections* zusammengefasst, und danach über die *Bands* den Wiedergabekanälen zugewiesen werden.

Da Downloadable Sounds unabhängig von der GM-Klangerzeugung einer Soundkarte wiedergegeben werden, ist ein gleichbleibender Klang auf allen Systemen gewährleistet. Der zweite große Vorteil ist, dass auch individuelle Sounds fernab der General MIDI-Spezifikationen eingebunden werden können. Für die Kompatibilität zu General MIDI wurde gesorgt, indem mit DirectX automatisch eine DLS-Collection namens ‚GM.dls‘ installiert wird, die GM-Klänge der Firma Roland enthält. Diese DLS-Collection kann auf jedem System über den „Microsoft Synthesizer“ angesteuert werden, und steht auch standardmäßig im DirectMusic Producer zur Verfügung.

Eine DLS-Collection besteht zunächst aus einzelnen *Instruments*. Ein Instrument entspricht dem, was bei anderen Synthesizern oft als ‚Map‘ oder ‚Bank‘ bezeichnet wird: Eine komplette Klaviatur, die meist aus mehreren einzelnen Tastaturzonen („Regions“) besteht, denen jeweils eine Wave-Datei zugewiesen ist. Diese Regions können sich überlappen, und in beliebig vielen Ebenen (*Layern*) übereinandergeschichtet werden. Seit der Erweiterung auf *DLS Level 2*, die mit DirectX 8.0 einherging, kann jeder einzelnen der 128 MIDI-Noten eines Kanals ihre eigene Region zugewiesen werden. Die Erweiterung ermöglichte unter anderem auch das Importieren von Stereo-Samples, sowie die Zuweisung eines Velocity-Bereichs zu jeder Region.

Jedes Instrument hat mindestens ein eigenes Set von Abspielparametern, genannt *Articulations*, über die sämtliche Komponenten der DLS-Architektur programmiert werden.

Hierbei stehen folgende Module zur Verfügung:

- je eine AHDSR-Hüllkurve für Lautstärken- und Tonhöhenverlauf
- 1 LFO zur Modulation von Lautstärke, Tonhöhe und Cutoff-Frequenz („ModLFO“)
- 1 LFO zur Modulation der Tonhöhe („VibLFO“)

- 1 resonanzfähiges Tiefpassfilter, dessen Einsatzfrequenz über ModLFO, die Pitch-Hüllkurve, und über MIDI-Controller 74 gesteuert werden kann

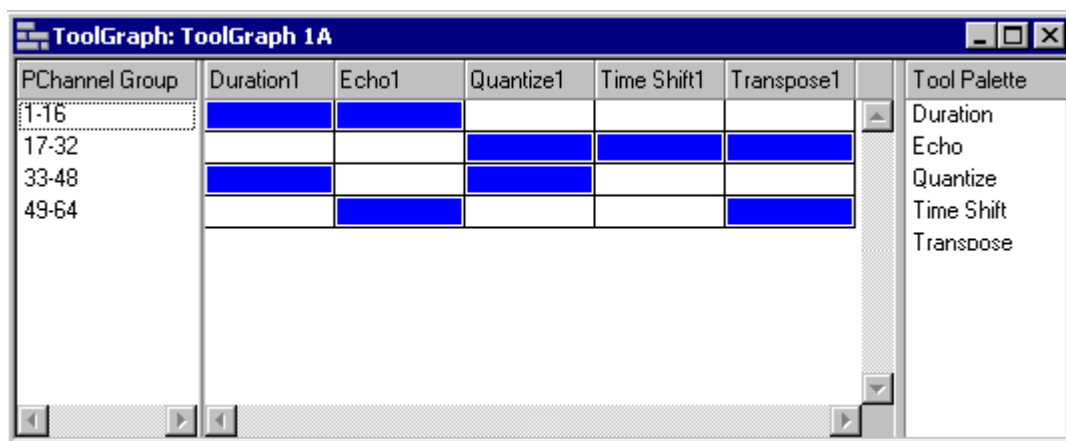
Innerhalb einer Articulation ist außerdem definiert, wie stark Tonhöhe und Lautstärke über Modwheel und ChannelPressure moduliert werden sollen. Es kann immer nur eine Articulation pro Instrument aktiv sein. Ihre Abspielparameter gelten Region-übergreifend für das komplette Instrument.

Screenshots des DLS-Designers finden Sie in Kapitel 6.5 .

5.12. Toolgraphs

Tools sind einfache MIDI-basierte Effekte, mit denen die Abspielparameter ergänzt oder verändert werden können, bevor sie an einen Synthesizer geschickt werden. Ihre Funktionen entsprechen weitestgehend denen, die in herkömmlichen Sequenzern über die ‚Instrument Parameter Box‘ (Logic) oder den ‚Inspector‘ (Cubase) eingestellt werden. Tools können die Sequenzen quantisieren und transponieren, die Notenlängen verändern, oder auch ein MIDI-basiertes Echo generieren.

Über den Toolgraph Designer werden Toolgraphs kreiert, die jeweils ein oder mehrere Tools beinhalten. Die Tools werden den PChannels über beliebig große Gruppen zugeordnet. Der MIDI-Datenstrom fließt seriell durch jedes einzelne Tool. Die Daten werden also von einem Tool manipuliert, und so verändert immer an das folgende weitergegeben.



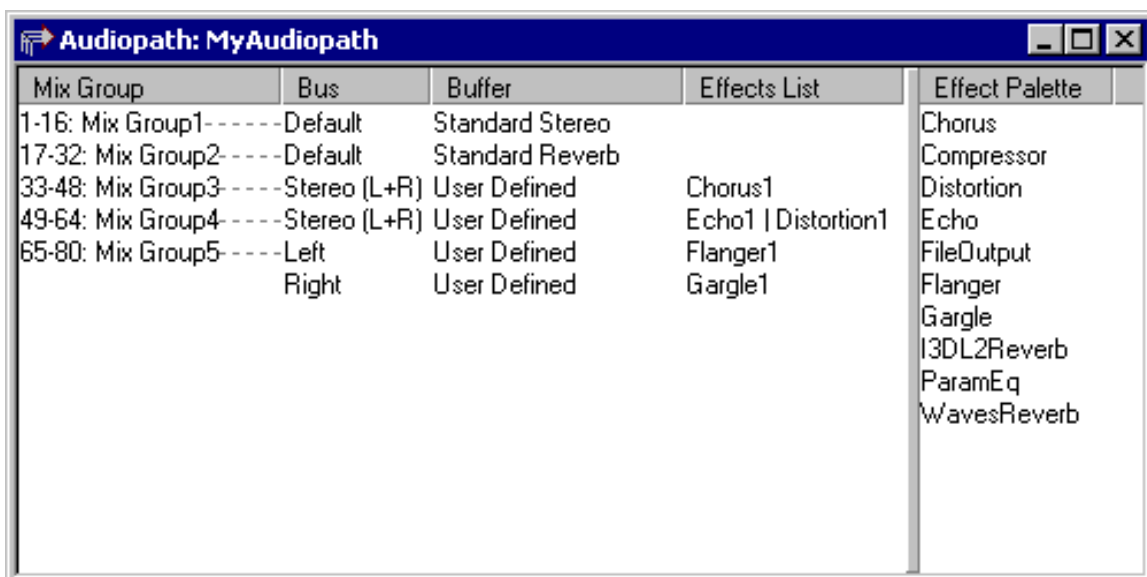
Toolgraph mit Tools für vier verschiedene PChannel-Gruppen. Die blauen Rechtecke zeigen an, dass das jeweilige Tool für die entsprechende Gruppe aktiv ist.

5.13. Audiopaths

Ein Audiopath ist eine Konfiguration, die Einstellungen zur Wiedergabe über die Audiohardware beinhaltet, und die unabhängige Nutzung von Filtern und Effekten wie Hall oder Chorus für jeden einzelnen *PChannel* regelt.

Eigene Audiopaths lassen sich über den *Audiopath Designer* erstellen, und dann Segmenten und anderen Komponenten zuordnen, um zu gewährleisten, dass diese jederzeit mit den richtigen Einstellungen wiedergegeben werden. Eine Liste mit nicht editierbaren Standard-Audiopaths steht im DMP bereits zur Verfügung, die auch als Grundlage für eigene Audiopaths des Benutzers dienen können.

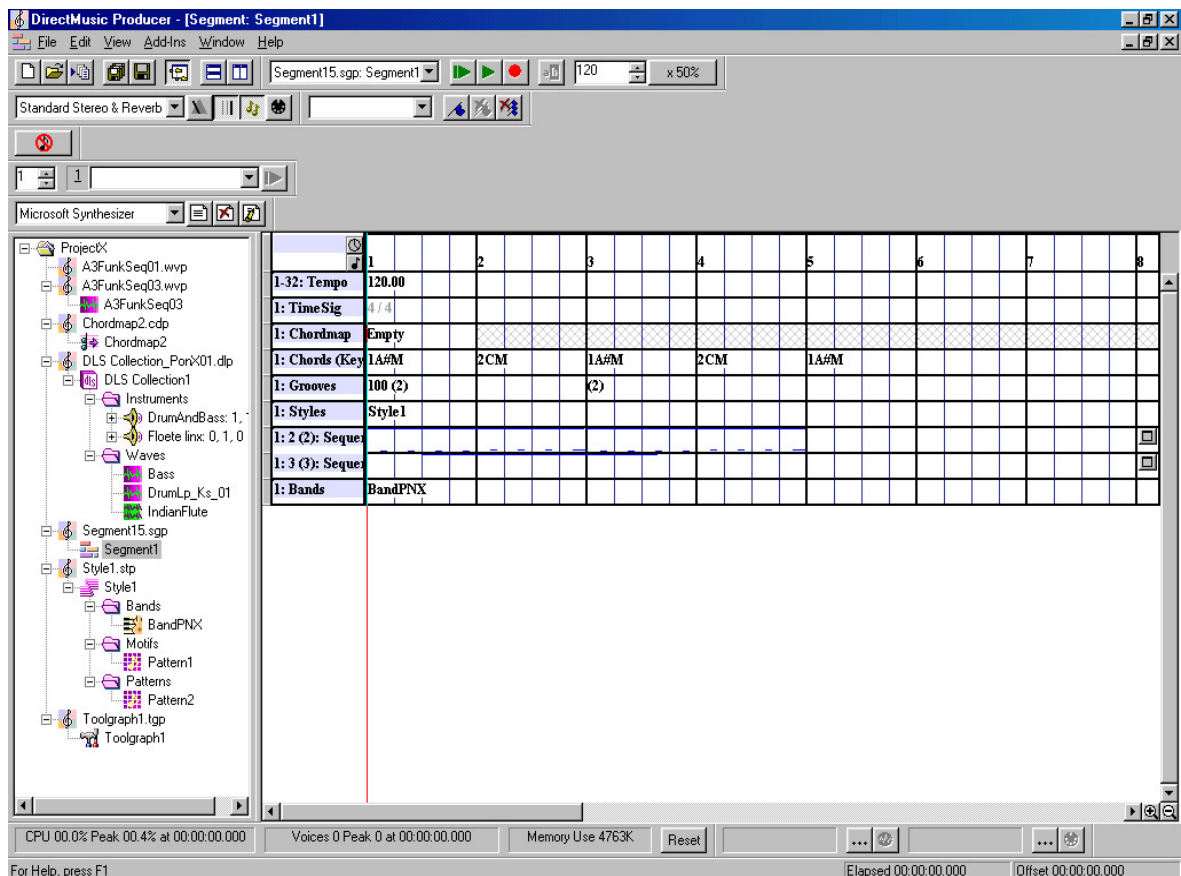
Beim Erstellen eigener Audiopaths werden die einzelnen PChannels in beliebig viele Gruppen („Mix Groups“) zusammengefasst, die sich jeweils die gleichen Effekt-Einstellungen teilen. Neben dem Routen des Signals über Effektbusse zum Ansteuern von Effekten wie Chorus oder Hall, ist auch das Insertieren virtueller Regelverstärker möglich. Hier stehen Kompressor, parametrischer EQ und Distortion zur Verfügung. Jede Mix Group ist einem bestimmten Audioausgang zugewiesen. In den meisten Fällen wird man dabei auf den *Microsoft Software Synthesizer* zurückgreifen, da über ihn die DLS-Sounds abgespielt werden, und somit ein gleichbleibendes klangliches Ergebnis auf allen Soundkarten garantiert ist. Stattdessen lässt sich aber auch die individuelle GM-Klangerzeugung der Soundkarte, sowie externes MIDI-Equipment ansteuern. Eigene Audiopaths werden entweder als Teil eines Segments, oder als separate Datei abgespeichert.



| Mix Group | Bus | Buffer | Effects List | Effect Palette |
|-------------------|------------------|-----------------|---------------------|----------------|
| 1-16: Mix Group1 | -----Default | Standard Stereo | | Chorus |
| 17-32: Mix Group2 | -----Default | Standard Reverb | | Compressor |
| 33-48: Mix Group3 | ----Stereo (L+R) | User Defined | Chorus1 | Distortion |
| 49-64: Mix Group4 | ----Stereo (L+R) | User Defined | Echo1 Distortion1 | Echo |
| 65-80: Mix Group5 | ----Left | User Defined | Flanger1 | FileOutput |
| | Right | User Defined | Gargle1 | Flanger |
| | | | | Gargle |
| | | | | I3DL2Reverb |
| | | | | ParamEq |
| | | | | WavesReverb |

Audiopath Designer

6. Die Bedienung des DirectMusic Producers



Im oberen Drittel des DMP-Fensters sind die Standard-Bedienelemente untergebracht. Neben den Datei-Management-Buttons und den Segment-Transportleisten befindet sich hier auch die Marker-Sektion, über die sich Fenster-Anordnungen schnell abspeichern und aufrufen lassen. Das durchgestrichene Notensymbol repräsentiert den ‚Panic‘-Button für Notenhänger („All-Notes-Off“).

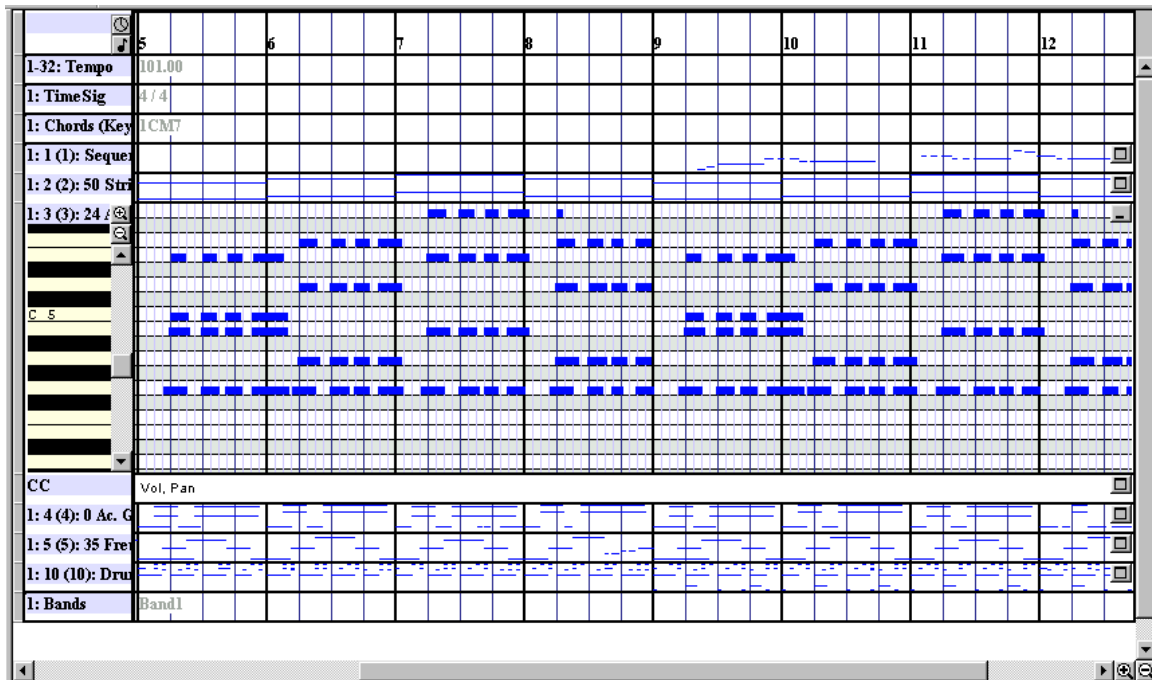
6.1. Project Tree

Der Rahmen auf der linken Seite beinhaltet den „Project Tree“. Ähnlich dem Aufbau des Windows-Explorers und des Registry-Editors sind hier sämtliche Komponenten des Projekts aufgelistet. Subkomponenten lassen sich zur besseren Übersicht durch Anklicken des ‚+‘- Symbols ein- bzw. ausblenden.

6.2. Editor-Fenster

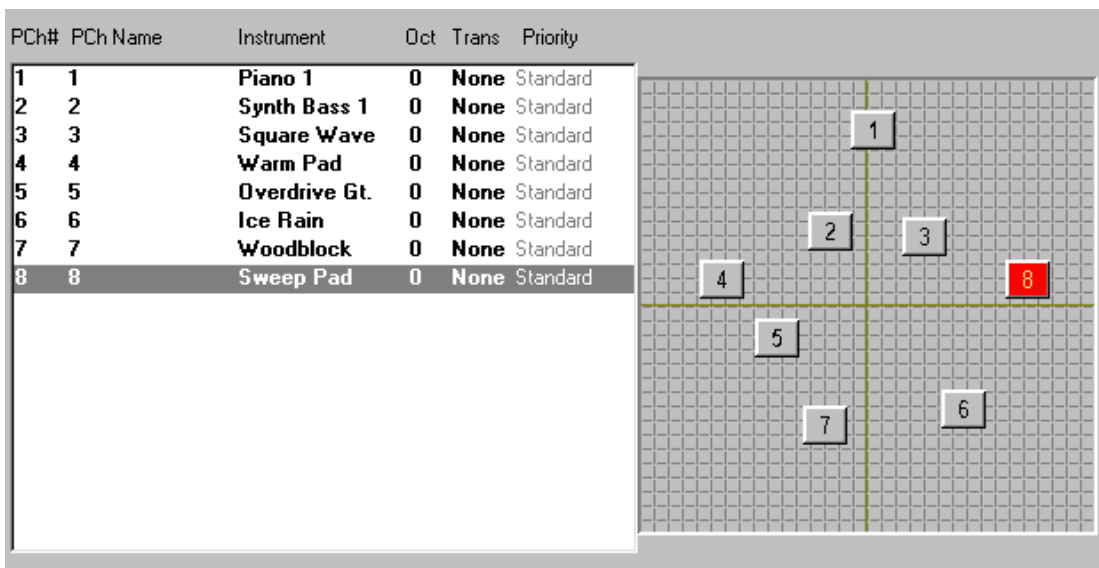
Das große Hauptfenster auf der rechten Seite beinhaltet den jeweils zuständigen Editor für die im Project Tree angewählte Komponente. In obigen Beispiel handelt es sich um den *Segment Designer*.

6.3. Piano Roll Editor



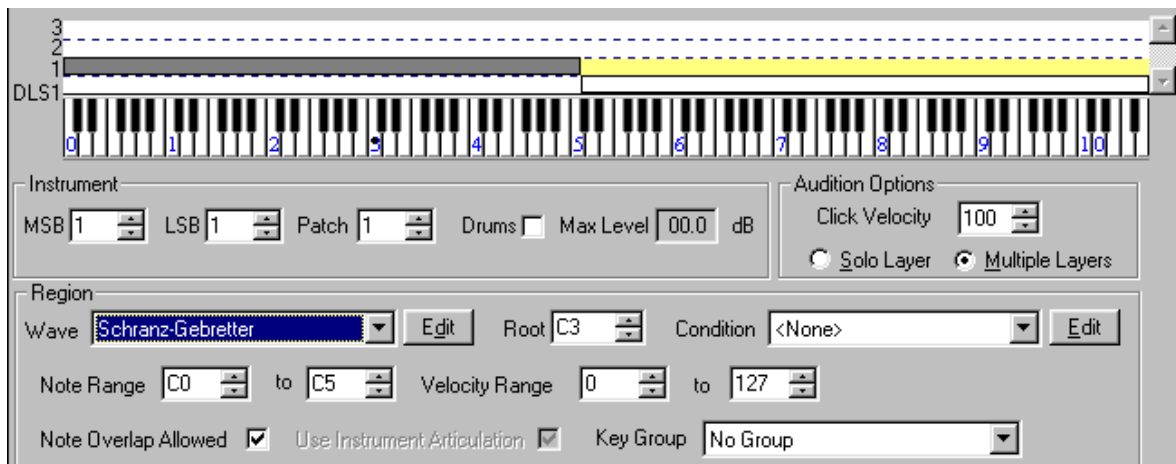
Der Piano-Roll-Editor ist der Standard-Editor für Sequenzen und Pattern-Parts. Er entspricht optisch weitestgehend den vergleichbaren Editoren bekannter Sequenzer (Matrix-Editor / Logic ; Key-Editor / Cubase), ist allerdings in Bedienung und Funktionsumfang wesentlich spartanischer. In der Regel ist man besser beraten, die MIDI-Sequenzen in herkömmlichen Sequenzern zu erstellen, und danach in den DMP zu importieren.

6.4. Band Editor

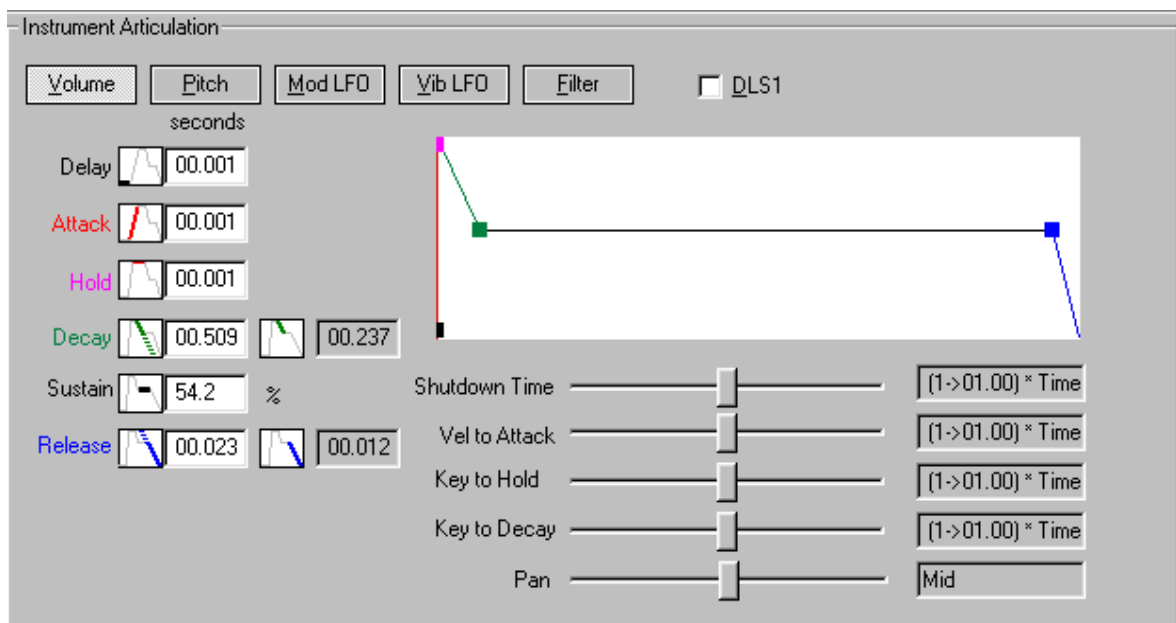


Im Band-Editor lassen sich die einzelnen PChannels den Instrumenten aus DLS-Collections oder der General MIDI-Klangerzeugung der Soundkarte zuweisen. Per drag-and-drop-Funktion werden die Klänge im Stereobild verteilt. Das Verschieben auf der vertikalen Achse regelt die Lautstärke des angewählten Instruments.

6.5. DLS Designer



In der oberen Hälfte des DLS-Designer-Screens werden die verschiedenen Samples ihren Tastaturzonen (*Regions*), Velocitybereichen und Layern zugewiesen.



In der unteren Hälfte werden die klangbearbeitenden Einstellungen der Tonhöhe- und Lautstärke-Hüllkurven, Filter und LFOs vorgenommen. Ein solches Parameter-Set wird in DMP unter der Bezeichnung „Articulation“ verwaltet.

7. Interaktion zwischen Spiel und DirectMusic – ein praktisches Beispiel

Eine typische Spielsequenz in einem Action-Adventure könnte also beispielsweise folgendermaßen ablaufen:

Wir steuern unseren Spielcharakter in einer Fantasy-Welt über offenes Feld. Noch ist alles friedlich, und im Hintergrund spielt DirectMusic ein Style-basiertes Standard-Segment ab, das dieser Landschaft zugeordnet ist. Es besteht aus Patterns mit mehreren Variations, sodass beim Umherwandern für genug musikalische Abwechslung gesorgt ist. Unser(e) Held(in) erklimmt eine Bergkuppe, und wir entdecken dahinter eine verfallene Burg. Über ein Status-Bit, das gesetzt wird, sobald ein bestimmter Punkt oder eine bestimmte Gesamtfläche der Burg-Polygone im Blickfeld des Spielers liegt, wird ein kurzes Motif getriggert, das einen düsteren Flächensound über bzw. unter die restliche Musik legt.

Während wir die Burg betreten, wird eine Transition zu einem zweiten Segment eingeleitet, das auf weniger harmonischen Akkorden basiert. Das drohende Unheil kündigt sich während des Übergangs musikalisch durch das Angleichen unseres Standard-Segments an die unheimlichen Akkorde des Ziel-Segments an, auch dessen tiefe Basstöne sind bereits zu hören. Nach der zweitaktigen Transition wird schließlich nur noch das zweite Segment abgespielt. Dieses greift auf einen Style mit mehreren Patterns zu, die nach steigender Intensität verschiedenen Groove-Level Bereichen zugeordnet sind. Im Moment bewegen wir uns auf einem niedrigen Groove-Level, wodurch nur der Bass und gelegentliche Flächensounds zu hören sind. Das ändert sich, als unser Held durch eine unkluge oder ungeschickte Aktion Lärm verursacht, und die Aufmerksamkeit der (selbstverständlich feindseligen) Burgbewohner auf sich zieht, was unmissverständlich durch einen gesteigerten Groove Level deutlich gemacht wird: Im nun aktuellen Pattern gesellen sich hektische Pizzikato-Läufe zum düsteren Grundklang. Um für noch mehr Unruhe zu sorgen, wird auch das Tempo leicht angehoben, und das gesamte Segment um einen Halbtonschritt nach oben transponiert, wobei die Rhythmus-Sequenzen unbeeinflusst bleiben.

Als wir uns einer Tür nähern setzt der Bass aus, und ein immer lauter werdendes Tremolo erklingt. Dieser Effekt wird durch ein Embellishment erzielt, auf das umgeschaltet wird, sobald ein gewisser Abstand zur Tür unterschritten wurde. Mit einem markerschütternden Streicher-Akkord endet das Embellishment, die Tür springt auf, mehrere Monster stürmen auf den Spieler zu, und der Groove Level wird auf den Maximalwert von 100 gesetzt. Wahlweise könnte an dieser Stelle auch auf ein anderes Segment umgeschaltet werden. Ein Pattern mit aufpeitschenden Percussion- und Streicher-Parts untermalt das obligatorische Kampfgeschehen. Nachdem auch der letzte

Gegner von unserem Helden niedergestreckt wurde, ertönt eine heroische Melodie in Form eines weiteren Custom-Embellishments. Danach wird das Segment mit gemäßigttem Groove Level und geruhsamerem Tempo fortgesetzt.

8. DirectMusic in der Praxis

“Mit der Nutzung von Technologien wie DirectMusic ist es möglich, Spiele eher (oder unter Umständen ganz genauso) wie [...] Filme klingen zu lassen. Das mag nach Reklame klingen, aber mit „DirectMusic“ ist nicht das momentane DirectMusic gemeint, sondern eher das, in was es sich nach unseren Vorstellungen entwickeln wird.“ [2]

Dieses Zitat des DirectMusic Entwicklers Bryan Schmidt lässt schon Rückschlüsse auf die Klangqualität zu, die DirectMusic momentan bietet. In der Tat kann eine mit DirectMusic generierte Musik-Performance klanglich kaum mit teuren Studio-Aufnahmen mithalten. Das liegt zum einen daran, dass die Musik-Engine bei der Verteilung der Prozessorressourcen gegenüber dem restlichen Spiel relativ wenig Priorität hat, weshalb auch heute noch häufig mit Mono-Klängen gearbeitet wird, die auf eine Samplingfrequenz von 22,050 kHz herunterkonvertiert wurden. Mit der steigenden Leistungsfähigkeit neuer Prozessor- und Soundkarten-Generationen dürfte dieses Manko hoffentlich bald der Vergangenheit angehören. Dennoch wird es immer qualitative Diskrepanzen zwischen interaktiv generierter Musik und professionellen Studio-Aufnahmen geben. Man sollte den Stellenwert der Klangqualität jedoch auch nicht überbewerten. Zum einen muss man bedenken, dass der Spieler nicht mit gespitzten Ohren auf die Hintergrundmusik achtet, sondern in der Regel voll mit dem eigentlichen Spielverlauf beschäftigt ist, wobei die Musik auch von den meist wesentlich lautereren Soundeffekten überdeckt wird. Zum anderen tröstet der immense Gewinn an Atmosphäre, Dramatik und Variabilität, den der Einsatz von DirectMusic ermöglicht, über leichte Abstriche in Sachen Klangqualität hinweg.

Die effektivsten Möglichkeiten, die Musik auf das Spielgeschehen abzustimmen, bieten sich sicherlich mit dem Einsatz verschiedener Groove Levels. Eine recht naheliegende Anwendung ist beispielsweise, das Sinken der Lebensenergie des Spielers mit einem Ansteigen des Groove Levels zu verbinden, der sich wiederum auf die Intensität der Musik auswirkt. Eine weitere Möglichkeit wäre der Einsatz eines secondary Segments, das ab einem kritischen Energiestand getriggert wird, und dramatische Akkorde oder düsteres Grollen über die Musik legt.

Die Implementierung solcher Features erfordert allerdings nicht nur die Sachkenntnis des Musikers, sondern auch die des Programmierers. Dieser muss genau wie der Musiker eine gute auditive Vorstellungskraft haben, und von Produktionsbeginn an quasi die Rolle eines Co-Komponisten übernehmen. Eine solch enge Zusammenarbeit während der Entwicklung ist heutzutage jedoch eher die Ausnahme. In der Regel wird die Musik in einer komplett getrennten Abteilung oder von Einzelpersonen produziert, und erst am Schluss in das schon fast fertige Produkt eingebunden. Oft werden auch namhafte Komponisten aus der Filmbranche mit der Vertonung von Computerspielen beauftragt. Den ersten Großauftrag in dieser Richtung erhielt der englische Filmkomponist Barrington Pheloung, der über drei Stunden Musik für das 1996 erschienene Adventure „Baphomets Fluch“ beisteuerte. Die digitalisierten Musiksequenzen wurden in einzelne Abschnitte zerlegt, als Wav-Dateien in das Spielverzeichnis installiert, und schließlich über ein eigenes Streaming-System wiedergegeben.

Dieses Prinzip funktioniert im Rahmen eines Szenen-basierten Spielprinzips sehr gut, die Implementierung weitreichender interaktiver Elemente ist bei einer solchen Herangehensweise jedoch kaum möglich. Vielen Komponisten aus dem Filmbereich fällt es außerdem sehr schwer, sich bei der Arbeit eine ‚interaktive‘ Denkweise anzueignen. Sie sind es gewohnt, fest vorgegebene Szenen zu vertonen, und kennen sich in der Regel nicht mit den technischen Möglichkeiten und Komponenten einer Musik-Engine aus. Möglicherweise besteht also bald ein erhöhter Bedarf an DirectMusic-Experten, die solche vorproduzierten Musikfragmente zu interaktiver Musik aufbereiten. Hierbei könnte auch die interaktive Vertonung von Webseiten in naher Zukunft eine Rolle spielen.

Ein weiterer Grund für die momentan eher seltene Nutzung von DirectMusic ist der recht steinige Einstieg in die Materie. Neben der Flut von neuen Begriffen ist vor allem der Bereich der Chord-Funktionen für Einsteiger schwer zu durchschauen. Auch erfahrene Benutzer ignorieren diese DirectMusic-Komponente teilweise völlig, zumal sie befürchten, damit die Kontrolle über ihre Musik zu verlieren. Bei all der Interaktivität und Variabilität ist die Angst groß, dass die liebevoll komponierten Sequenzen zu einem dissonanten Gemisch verbogen werden, dessen Ursachen sich irgendwo in dem scheinbar undurchdringlichen Parameterwust verbergen. Diese Befürchtungen sind nur zum Teil begründet. So bietet das gezielte Zeichnen von Chordmaps genug Kontrolle, um diese Funktion auch musikalisch sinnvoll einsetzen zu können. Das zufällige Auswählen der Signpost-Akkorde aus Gruppen hingegen stellt für die meisten Anwender allerdings eher ein musikalisches Experiment bzw. Russisches Roulette dar, das mit dem Wohlklang gespielt wird. Hier bewegt man sich schnell auf einem schmalen Grat zwischen größtmöglicher Variabilität und reiner Willkür, die mit der gezielten Erzeugung von Atmosphäre und Emotionen nichts mehr zu tun hat.

Grundlegende Chord-Funktionen stellen jedoch ein sehr flexibles Werkzeug dar, das richtig eingesetzt einen großen Teil zu einer dichten Spielatmosphäre beitragen kann. Ein gängiges Stilmittel in Filmen ist zum Beispiel das leise Wiederholen eines ursprünglich dominierenden Themas, das später erneut als leises Echo in einer komplett anderen Tonart und Instrumentierung zu hören ist. Dies lässt sich sehr gut mit *Motifs* realisieren, die sich dynamisch an den aktuellen Harmonieverlauf angleichen. Aber auch die DirectMusic-Entwickler selbst stufen die Chord-Funktionen als eher sekundär ein: „Wir haben uns vielleicht selber keinen Gefallen getan, indem wir diesen Teil der Architektur so betont haben. Er ist interessant, und einige Musikrichtungen funktionieren besser als andere, aber ich persönlich ordne ihn als weitaus weniger wichtig ein, als die Synchronisation / Kontinuität, die Interaktivität und Variabilität im allgemeinen.“ ([3] David Yackley)

Auch die Zielsetzung der DirectMusic-Entwickler, orchestrale Musik zu erzeugen, die möglichst realistisch klingt, wird von vielen modernen Musikern als nicht mehr aktuell angesehen. Weder der Zuschauer im Kino, noch der durchschnittliche Radiohörer oder Computerspieler kümmern sich darum, ob die Musik „realistisch“ klinge oder nicht. Sie kritisieren die Klangarchitektur in DirectMusic, das biedere Konzept der ‚Bands‘ und ‚Styles‘, und wünschen sich stattdessen ausgefallenerere und flexiblere Eingriffsmöglichkeiten ins Klanggeschehen. Hierzu ist anzumerken, dass sich mit DLS2 schon recht gute Möglichkeiten bieten, auch synthetische Klänge zu erzeugen, auch wenn die Qualität des resonanzfähigen Filters sicherlich keine neuen Maßstäbe setzt. Auf weitere Extensionen des DLS-Standards darf man hoffen und gespannt sein. Aber schon jetzt kann man mit gesampelten Filterfahrten, Sync-Sounds oder Pulsweiten-Modulationen schon recht gute Effekte erzielen, ohne auf ausgefeilte Klangsyntheseverfahren innerhalb der Musik-Engine angewiesen zu sein.

In einer konsequenten Erweiterung des DLS-Konzepts sehen manche Anwender auch die Zukunft DirectMusics, bzw. die der interaktiven Musik allgemein:

„Im Moment haben wir die Downloadable Sounds (DLS). Für die Zukunft habe ich eine ‚Downloadable Studio‘ Theorie. Ein Stück interaktiver Musikdaten wird ein virtuelles Studio enthalten, das aus DLS Collections, Software Synthesizern [...], Physical Modeling Synths, DSP Effekt Plug-Ins [...] besteht, die alle mit einer virtuellen Mix-Einheit verbunden sind. Das ‚Studio‘ wird auf den Rechner des Spielers / Endverbrauchers heruntergeladen, aber unsichtbar für ihn sein. Er wird nur komplexe interaktive Musik hören, die vom Spiel [...] kontrolliert wird.“ [4] (Guy Whitmore)

Zur Realisierung eines solchen Systems bedürfte es heute einer neuen Soundkarten-Generation mit eigener DSP-Sektion, die bei ausreichender Nachfrage bald für die Masse

erschwinglich sein könnte. Allerdings macht sich kaum ein Spieler Gedanken über die verschiedenen Musik-Engines, oder den Nutzen von interaktiver Musik. Mehrinvestitionen bei Soundkarten sind für die Mehrheit nur vertretbar, solange dadurch der Hauptprozessor entlastet, und somit das restliche Spiel beschleunigt wird. Hauptkriterium bei der Beurteilung der Musik ist zunächst ausschließlich deren klangliche Qualität - Die Vorteile eines guten interaktiven Soundtracks entpuppen sich erst während des Spiels auf subtile Weise. Tatsächlich realisieren viele Spieler oft gar nicht bewusst, dass die Musik auf das Spiel reagiert, empfinden das Spielerlebnis aber als intensiver und spannender.

9. Interview mit Guy Whitmore, "Whitmoreland Productions"

Guy Whitmore hielt bereits im Jahr 1998 Referate über die Nutzung des DirectMusic Vorgängers IMA, und ist heute einer der meistbeschäftigten DirectMusic - Komponisten weltweit. Seit 1994 komponierte er zahlreiche interaktive Soundtracks zu bekannten Spielen, wie z.B. den sehr erfolgreichen 3D-Shootern „Shogo“ und „No One Lives Forever“.

„Wann hast du zum ersten Mal mit der DirectMusic Technologie gearbeitet ?“

Guy: „Ich glaube es war Ende '97, als mein damaliger Boss Dan Bernstein eine Beta-Version der *Microsoft Interactive Music Architecture*, dem DirectMusic-Vorgänger, auf meinen Tisch warf und sagte: „Hier. Guck mal, was du damit anfangen kannst.“ Ein paar Wochen später hatte ich ein kleines Demo fertig, dass ich dem „Shogo“-Team zeigte. Es war gut genug, dass wir uns entschieden, IMA in unsere LithTech-Engine zu integrieren, zumal DirectMusic noch nicht veröffentlicht war.“

„Was genau hat DirectMusic für dich interessant gemacht ?“

Guy: „Zwei Dinge haben mich sofort an DirectMusic fasziniert: Die Möglichkeit, nahtlose Übergänge zwischen zwei musikalischen Abschnitten oder Musikstücken zu erzeugen, und der DLS Standard, der uns das Erstellen eigener Soundsets ermöglichte, und für den konstanten Klang auf allen Plattformen sorgte. Sobald ich erkannt hatte, dass DirectMusic / IMA diese Dinge konnte, entschied ich mich, es zu nutzen.“

„Was sind deine allgemeinen Erfahrungen bei der Arbeit mit DirectMusic ? Was gefällt dir daran ?“

Guy: „Mit DirectMusic zu arbeiten ist ein ständiger Entdeckungs-Prozess. Die interaktiven Konzepte selbst entwickeln sich noch, und es gibt noch keine Regeln. Man muss seine eigenen Regeln machen, oder man geht in all den Möglichkeiten unter. Mit jedem Musikstück, das ich in DirectMusic kreierte, versuche ich, eine oder zwei neue Techniken zu meinem Repertoire hinzuzufügen.“

„Wie gehst du generell beim Komponieren eines Spiele-Soundtracks vor?“

Guy: „Seit Shogo und Blood 2 komponiere ich zunächst einen kompletten linearen Redbook – Score, wobei ich meinen Hauptsequenzer (Digital Performer) und sämtliche Instrumente in meinem Studio benutze. Dann zerlege ich die Musik in die DirectMusic entsprechenden Bestandteile. Danach versuche ich, die Musik wieder von grund auf ‚nicht-linear‘ zu konzeptionalisieren. Ich fange wiederum in im Performer an, benutze ihn jedoch eher als musikalisches Notizbuch, um Themen, Klänge und Ideen in den Computer zu bekommen. Basierend auf diesen Themen erstelle ich dann komplett neue Musikblöcke im DMP. Auf diese Weise kann ich mit der Interaktivität und den Variationen arbeiten, während ich die Musik arrangiere. Eine interessante Technik, die ich benutzt habe, ist sehr improvisatorisch und nicht-linearer: Ich nehme improvisierte Gitarren-Parts auf, zerschneide dann die Datei in kurze Samples, und erstelle daraus eine eigene DLS-Bank. Danach kann in die gespielten Sequenzen innerhalb des [DirectMusic] Producers wieder zusammenbauen, oder komplett neue Gitarren-Parts kreieren.“

„Welche Dinge haben dich an DirectMusic am meisten frustriert?“

Guy: „Ich benutze Sequenzer-Software [...], die seit über einem Jahrzehnt weiterentwickelt wird. [...] DirectMusic Producer wirkt manchmal etwas unbeholfen, vor allem im Bezug auf seine Sequencing-Funktionalität, weil es einfach noch nicht all die Jahre der Entwicklungszeit hinter sich hat. Vor allen Dingen wurde es noch nicht ausgiebig in echten Produkten und Spielen getestet. Microsoft wird auf diesem Wege entscheidendes Feedback bekommen, sodass das Programm reifen, und mit der Zeit eine elegantere Schnittstelle erhalten kann. Und ganz davon abgesehen entwickeln sich auch die inhaltlichen Konzepte noch, auf denen DirectMusic basiert. Microsoft wird während dieser Entwicklung zügig auf das Feedback der Benutzer eingehen müssen. Bis jetzt kann ich bestätigen, dass Microsoft auch tatsächlich zuhört.“

„Von welcher einen Sache wünschst du dir, dass man sie dir gesagt hätte, als du angefangen hast, DirectMusic zu benutzen ?“

Guy: „Sei geduldig, und lerne immer eine Sache nach der anderen.“

„Was an DirectMusic hat dich am meisten verwirrt ?“

Guy: „Zunächst war die Terminologie teilweise verwirrend, „Style“ zum Beispiel. Aber sobald man erkannt hat, dass sie einfach nur Bezeichnungen sind, und nicht wörtlich genommen werden sollen, wird einiges klarer.“

„Was hat dich an DirectMusic am meisten überrascht ?“

Guy: „Die Tiefe des Programms. Man braucht vielleicht nur Wochen, um die Techniken zu erlernen, aber um sie zu meistern, braucht man Jahre.“

„Welche Funktionen von DirectMusic hoffst du, dir in deinem nächsten Projekt erschließen zu können?“

Guy: „Ich experimentiere schon mit allen Aspekten von DirectMusic. Mein erstes Ziel war, die Musik so edel klingen zu lassen wie meine CD-Audio Musik, weil meiner Meinung nach interaktive Musik witzlos ist, die immer noch nach General MIDI klingt. Deshalb habe ich mich so intensiv mit den DLS-Techniken beschäftigt. Ich habe in letzter Zeit auch ausgiebig Motifs und Secondary Segments eingesetzt, um mit verschiedenen musikalischen Ebenen arbeiten zu können. Und ich fange an, die harmonische Funktionalität auszuprobieren, wie z.B. die Chordmaps. Die Art der Musik, die ich mache, bestimmt die DirectMusic-Techniken, die ich benutze.“

„Hast du noch irgendwelche Ratschläge für Komponisten, die den Einsatz von DirectMusic in Betracht ziehen ?“

Guy: „Mein Ratschlag für DirectMusic-Neulinge ist, es am Anfang einfach zu halten. Mit wenigen effektiven interaktiven Techniken kommt man schon recht weit. Behaltet außerdem den Musikstil im Hinterkopf, den ihr schreibt, und welche Art von Interaktivität für euer Programm geeignet ist. Mein erster Schritt beim Komponieren eines Spiele-Soundtracks ist es, sich mit dem Designer und dem Programmierer hinzusetzen und zu

entscheiden, welche Art musikalischer Interaktivität wir gerne in dem Spiel hören würden. Alles andere sollte sich nach diesen Zielen richten.“

10. Resümee:

Auch ich habe vor der Recherche zu dieser Arbeit nie wirklich bewusst auf die interaktiven Elemente in der Hintergrundmusik eines Spiels geachtet, und war überrascht, wie viel solche Funktionen zur Spielatmosphäre und zu einem Kino-ähnlichen Spielerlebnis beitragen können. Hierbei machen variable Intensitätsgrade den mit Abstand größten Unterschied.

Bei all dem Streben nach möglichst vielen variablen Elementen stellt sich aber auch schnell die Frage nach dem Verhältnis zwischen Aufwand und tatsächlichem Nutzen. Die Pulsweite eines Bass-Sounds passend zum Spielgeschehen zu modulieren, macht sicher nur in Ausnahmefällen Sinn. Mit den in DirectMusic enthaltenen Funktionen ist glaube ich ein Punkt erreicht, an dem für ein ausreichend großes Potential zur Verwirklichung guter interaktiver Soundtracks gesorgt ist, ohne zwangsläufig die Programmierer (und nicht zuletzt den Komponisten selbst) zu überfordern. Was aber hoffentlich in naher Zukunft folgen wird, sind technische Erweiterungen wie z.B. Echtzeit-Timestretching, um die statischen Wavetracks flexibler zu machen. Außerdem kann ich mich Guy Whitmore in dem Punkt anschließen, dass die Benutzeroberfläche des DMPs in naher Zukunft überarbeitet werden sollte. Vor allem mehr sinnvolle Keyboard-Shortcuts wären wünschenswert, um die teils recht träge Navigation zu beschleunigen.

Bei diesen Kritikpunkten muss man jedoch bedenken, dass es nicht nur viele Jahre der Entwicklung sind, die DMP von professioneller Sequenzer-Software trennen, sondern auch z.T. weit über 1000 DM Anschaffungskosten.

Anhang: Beispiele auf der Audio-CD

Track 1: 'The Dig' [iMuse]

Das 1995 erschienene 2D-Adventure „The Dig“ war eines der ersten LucasArts-Spiele, bei der die um Audio-Streaming erweiterte iMuse-Engine zum Einsatz kam. Das Klangbeispiel soll in erster Linie die damalige Klangqualität demonstrieren, und enthält mehrere Überblendungen zwischen einzelnen Musikblöcken. Die erste erfolgt an Position [00:03], als unser Spielcharakter einen neuen Screen betritt. An [00:17] wird eine Cutscene¹ eingeleitet, in der die Kamera selbstständig über die soeben betretene Halle gleitet. Es folgt ein kurzer Monolog unseres Helden, und schließlich wird wieder zur eigentlichen Hintergrundmusik des entsprechenden Spielbereiches überblendet [01:10].

Track 2: Chord-Funktionen [DirectMusic]

Eine Sonate wurde zunächst in ein Pattern importiert, wobei im Chord-Track des Patterns auch die Tonart der Sonate und ihr Akkord-Verlauf (mit bis zu vier Akkordwechseln pro Takt) angegeben wurde. Dieses Pattern wird nun über ein Segment abgespielt, das anfänglich die gleichen Original-Akkorde benutzt. Nach sieben Takten [00:14] wiederholt sich das Pattern, wobei der Chord Track des Segments diesmal statisch auf A-Moll gesetzt wurde. Durch das Ausblenden der harmonischen Komponente wird das Stück auf seine melodischen Konturen reduziert. Ab Takt 11 [00:25] wird die ursprüngliche Akkordfolge wieder fortgesetzt.

Track 3-5: 'No One Lives Forever' [DirectMusic]

Eines der wenigen aktuellen Spiele, das die Möglichkeiten von DirectMusic gut ausschöpft:

Als Persiflage auf die frühen James Bond - Filme der 60er-Jahre schleicht und schießt sich der Spieler in Gestalt einer smarten Agentin durch zahlreiche Spiel-Szenarien.

¹ automatisch ablaufende Szene innerhalb eines Spiels. Cutscenes sind entweder vorproduzierte Videosequenzen, oder werden über die Spiele-Engine dargestellt.

Track 3:

| | |
|----------------------|--|
| [00:00] – [00:07] | Noch schleichen wir unerkant durch feindliches Terrain. Nur die eigenen Schritte sind zu hören. |
| [00:07] | Spannungsgeladene Musik setzt ein und deutet an, dass Gegner in der Nähe sind |
| [00:15] | Wir haben einen Gegner entdeckt und nehmen ihn aus der Ferne ins Visier. |
| [00:27] | Nachdem die Schüsse ihr Ziel verfehlt haben, ist die Aufmerksamkeit des Gegners geweckt -> der Groove Level wird erhöht, und eine Percussion-Sequenz setzt ein. |
| [00:39] – [00:45] | Wir verlassen unsere Deckung und strecken wenig damenhaft mehrere Gegner mit einem Maschinengewehr nieder. Die Musik steigert kurz ihre Intensität, und endet mit einem End-Embellishment, nachdem die Gefahr beseitigt ist. |

Track 4:

Wir treiben den Groove Level auf Maximum, indem wir hektisch umherlaufen und großzügig von der Schusswaffe Gebrauch machen. Die nun gespielten Patterns enthalten viele musikalische Elemente, außerdem wird die Gesamtlautstärke der Musik leicht angehoben. Schließlich entkommen wir aus der Gefahrenzone, und die Musik verstummt nach einem weiteren End-Embellishment. Bemerkenswert ist die gute Qualität der Saxophon-Parts, die über Wavetracks in das Arrangement eingebunden wurden.

Track 5:

In dieser Spielsequenz laufen wir zunächst durch eine nächtliche Landschaft auf einen feindlichen Außenposten zu. Je näher wir dem Lager kommen, desto mehr Spuren / Parts werden zum Arrangement addiert. Nach einem Schuss aus dem Scharfschützengewehr fährt ein Geländewagen auf unsere Heldin zu [01:00], den wir mit gezielten Schüssen zur Explosion bringen [01:11]. Die Explosion triggert eine Transition auf ein zweites primary Segment, das den nun folgenden Schusswechsel untermalt.

Quellenverzeichnis

Zitate:

- [1] (Seite 11) - Brian Schmidt, IA-SIG Newsletter 01/1999
- [2] (Seite 29) - Brian Schmidt, IA-SIG Newsletter 01/1999
- [3] (Seite 31) - David Yackley, DirectMusic Group, 03.06.2001
- [4] (Seite 32) - Guy Whitmore, IA-SIG Newsletter 02/1999

Benutzte Quellen:

- **DirectMusic Producer Online-Hilfe**
- **Interactive Audio Special Interest Group** (<http://www.iasig.org>)
Interactive Audio Journal Ausgaben 1999 - 2000
- **Yahoo-Groups:** DirectMusic Group, GameAudioPro Group

Emails von: Guy Whitmore (Whitmoreland Productions), David Jackley (Microsoft), Brian Schmidt (Microsoft), James Dooley (Microsoft), Bjorn Lynne (Lynnemusic), Tommy Tallarico (Tallarico Studios), Colin Anderson (DMA), Alistair Hirst (Electronic Arts), Jason Booth, Ciaran Walsh

- **US Patent 5315057:** Method and apparatus for dynamically composing music and sound effects using a computer entertainment system (www.delphion.com) *iMuse*
- **Game Developer Magazine**
- **weitere Websites:**

www.music4games.net ; www.lynnemusic.com ; www.lool.net/Game_Music/ ;
www.gamasutra.com ; www.microsoft.com/directx ; www.lucasarts.com ;
<http://imuse.mixnmojo.com> ; www.clefmusic.freemove.co.uk ; www.huelsbeck.com

Abbildung auf der Titelseite: „Asteroids“-Arcade-Automat (www.classicgaming.com)