JUHA ARRASVUORI

Playing and Making Music

Exploring the Similarities between Video Games and Music-Making Software

■

ACADEMIC DISSERTATION
To be presented, with the permission of
the Faculty of Humanities of the University of Tampere,
for public discussion in the Auditorium Pinni B 1096,
Kanslerinrinne 1, Tampere,
on September 6th, 2006, at 13 o'clock.

UNIVERSITY OF TAMPERE

ACADEMIC DISSERTATION
University of Tampere
Department of Music Anthropology
Finland

Distribution
Bookshop TAJU
P.O. Box 617
33014 University of Tampere
Finland

Tel. +358 3 3551 6055
Fax +358 3 3551 7685
taju@uta.fi
www.uta.fi/taju
http://granum.uta.fi

Cover design by
Juha Siro

Printed dissertation
Acta Universitatis Tamperensis 1165
ISBN 951-44-6688-8
ISSN 1455-1616

Electronic dissertation
Acta Electronica Universitatis Tamperensis 543
ISBN 951-44-6689-6
ISSN 1456-954X
http://acta.uta.fi

Tampereen Yliopistopaino Oy – Juvenes Print
Tampere 2006

Abstract

Music-making software titles are special applications for the Sony PlayStation 2 video game console. Music is made with music-making software by selecting, modifying, and arranging digital sound files through their visual representations. This music-making activity is defined as nonlinear multitrack composing. The theory of video games as semiotic domains by the educational linguist James Paul Gee is adapted to define the semiotic domain of multitrack composing, which includes, among other things, the techniques of using multitrack studio equipment in music-making and the meanings given to the resulting music.

The purpose of the study is to determine how nonlinear multitrack composing can be combined with play activities, and to define the design foundations of future video games that involve nonlinear multitrack composing.

The research material includes three music-making software titles (*Magix Music Maker*, *MTV Music Generator 2*, and *eJay Clubworld*) and seven music-themed video games. The research methods include comparing the three music-making software titles with each other and with various definitions of video games. Nonlinear multitrack composing is compared to play activities to demonstrate the similarities and differences. The three music-making software titles are compared in order to identify their common functionality, and through defining a metaterminology of the functions of multitrack studio equipment, it is shown how the features of music-making software relate to the semiotic domain of multitrack composing. A model of the nonlinear multitrack composing process is defined to contain the phases of selecting, sampling, modifying, arranging, and mixdown. The model is used to structure the analyses.

The similarity of music-making software to multitrack studio equipment is demonstrated. The following nonlinear multitrack composing techniques can be realized with music-making software as well as multitrack studio equipment: Sampling, sound design, mixing, programming, sound spatialization, and remixing. The backgrounds of the techniques are defined on the basis of earlier literature. Examples are presented of how the techniques have been used in music-making.

Music-making software titles are compared to the following aspects of video games: Formal elements, dramatic elements, video games as systems and simulations, and mimicry in play. These definitions are based on earlier studies and literature. Making music with video games and music-making software is compared to ludus and paidia types of play activities as defined by the game researcher Gonzalo Frasca. Ludus is defined as playing a game with the objective of producing a winner or a record result, whereas paidia is defined as playing without such objectives.

The following essential similarities between music-making software and video games are identified. Music-making software titles resemble video games that combine playing with creating and experimenting. As in video games with the objective of construction, so also in music-making software immaterial objects are made by selecting, modifying, and arranging prefabricated components. Music-making software titles are interpreted as open

simulations, that is, a kind of “playgrounds” that offer the tools and materials for making music with the approach of paidia play.

The following key differences are shown between music-making software and video games. Music-making software titles do not specify explicit goals as games do. There are no predefined end conditions for the music-making. Music-making software titles do not give feedback and motivate music-makers as games motivate players, for instance by giving rewards and adapting the challenge to match the ability. Managing resources is essential in many games. The tools and materials of music-making software are not resources because they can be used unlimitedly. Music-making software titles do not influence the behavior of the music-maker as games influence the behavior of the player by valorizing certain outcomes of the play activity over others. Because there is no competition and the outcomes are not measured through scores or in terms of winning and losing, music-making software titles do not set up a conflict as ludus type of games do. Thus, making music with music-making software is not ludus type of play.

The results of the study are design foundations of future video games that involve nonlinear multitrack composing. The design foundations are based on the established similarities and differences between music-making software and video games. Following the definition by the game researcher Jesper Juul, the rules of ludus type of video games that involve nonlinear multitrack composing should define outcomes that are variable, quantifiable, and valued. Such outcomes can be achieved, for example, if the material and tools for music-making are defined as resources and their use is measured. The methods that video games use to facilitate learning should be utilized to assist learning nonlinear multitrack composing.

Acknowledgements

First, I wish to thank the supervisor of my postgraduate studies, Professor Timo Leisiö for his tolerance towards my diverse ideas and research interests.

Thanks to Professor Paul Théberge and Professor Frans Mäyrä, the pre-examiners of this thesis for their constructive comments.

In Nokia Research Center, thanks to Jussi Holopainen, Juha Kaario, Kari Lahdensuo, Matti Karlsson, and Professor Jukka Saarinen for giving me opportunities over the years to carry out this research in parallel with my regular work. I am grateful to Game House at the NRC Multimedia Technologies laboratory for funding the language review of the dissertation manuscript.

I also thank Jukka Holm, Jussi Holopainen, and Professor Mikko Lehtonen for reading early versions of this thesis and suggesting improvements. Virginia Mattila reviewed the English language of the manuscript.

Finally, I warmly thank my mother Leena, my close relatives, and all the others who have given me support and encouragement.

Tampere, July 2006

Juha Henrik Arrasvuori

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

Video games are artifacts of the computer era for satisfying the human desire for play and games. In just a couple of decades, video games have become a part of the way of life of possibly hundreds of millions of people. In recent years, video games have established themselves as an influential element in popular culture and as a major industry.

Following the game researcher Jesper Juul (2003: 1, 44), I use the term 'video game' as an umbrella term for games played using computer processing power and memory, and a video display. Video games are played on personal computers and video game consoles including Sony PlayStation 2. Video games are audiovisual media with a lot for players to appreciate, and scholars to explore.

Making music is another universal human activity. Before the era of sound recording, people had to hire musicians or perform themselves in order to hear music. Today, people listen more to music from sound recordings than make music themselves. During the twentieth century, the production, distribution, and consumption of music have become activities rooted in electronic and digital technology.

Multitrack composing emerged in the 1960s from multitrack recording studios as an approach to making music. Recently it has become possible to do this on personal computers. With computer software, tape-based linear multitrack composing was adapted into nonlinear multitrack composing, which involves making music in non-real-time by arranging digital sound recordings (sample files) into pieces of music.[1] The means of nonlinear multitrack composing include, for example, sampling, sound design, programming, and mixing. Making music with the means of nonlinear multitrack composing is also possible with certain software titles for PlayStation 2 video game console. I call these PlayStation 2 titles collectively 'music-making software'.

There are some games for PlayStation 2 that incorporate music-making into playing the game. To name an example, *Rez* is a third-person shoot'em up game with electronic drum sounds instead of conventional weapon sounds. When the player shoots, the drum sounds are replayed so that they match the rhythm of the background music. The video game *Amplitude* includes adaptations of pieces of music by popular artists. In the game, the player must unmute tracks such as bass and drums by pressing the correct buttons at the correct times to hear the songs completely. The player can choose the order in which to open the tracks, and thus in a sense remixes pieces of music while playing. In

[1] Throughout this study, I use the verb 'making' rather than 'producing' or 'creating', even if it is inelegant at times. The former term has economic connotations and the latter term does not aptly describe what nonlinear multitrack composing is about. Making music in non-real-time means that there is no temporal equivalency between a music-making action and the audible outcome. For example, a two minute long sound recording can be duplicated with computer software. Performing the copying and pasting actions takes a couple of seconds of time, but the audible outcome is doubled in length to four minutes.

Amplitude, the mix that results from playing through a game level can be saved for later listening. When a player plays *Rez* or *Ampitude*, he triggers and modifies sounds that become a sort of live performance on top of the background music. In these games, sounds are triggered in real-time. Real-time musical performance is merely one music-making activity.

Few video games incorporate music-making with the means of nonlinear multitrack composing, which is much more complex way to make music than triggering sounds in real-time. The key research question of this study is the following: Is it possible to combine nonlinear multitrack composing with playing video games? I approach this question by describing and comparing the characteristics of music-making software and video games, and defining what kind of music-making activity nonlinear multitrack composing is.

Next, I briefly describe music-making software. After that, I present justifications for my approach and the research question from the theories of semiotic domains and remediation. At the end of Chapter 1, I describe how I will carry out the study.

1.2 Music-Making Software

A number of software titles for making music with the means of nonlinear multitrack composing have been released for video game consoles among them PlayStation, PlayStation 2, and XBox.[2] I explicitly define the following three titles for PlayStation 2 as 'music-making software': *Magix Music Maker*, *MTV Music Generator 2*, and *eJay Clubworld*.[3]

With these software titles, pieces of music are made from brief digital sound recordings (sample files) provided with the software or recorded by the music-maker. The music-making may progress, for instance, so that sample files[4] are browsed and selected from libraries, placed on tracks, modified with effects such as chorus and distortion, and finally combined into a stereo mix.

Some similar software titles for making music with video game consoles have been released over the years. These include *Music* (1998, for PlayStation and PlayStation 2), *Music 2000* (1999, for PlayStation and PlayStation 2), *Music 3000* (2003, for PlayStation 2

---

[2] In this study, I will examine versions of music-making software released in the European market. The reason for focusing on European versions is technical: North American and Japanese versions cannot be used with regular European video game consoles.

[3] "MTV" and "eJay" are brand names rather than the names of the developers of these software titles. There are other "MTV" branded video games as well, such as *MTV Celebrity Deathmatch* based on the television program of the same name.

[4] The term 'sample' is commonly used to refer to the digitized sound in its entirety (a computer file) as well as to each individual binary number making up the sample file (Dobson 1992: 137). In this study, I use the term 'sample file' to refer to the entire digitized sound. A sample file is a digital representation of a sound or several overlapping sounds. Thus, it can be said that a music-maker triggers sounds when, for example, he triggers sample files in real-time in *eJay Clubworld* by pressing buttons.

and XBox), and *MTV Music Generator 3: This is the Remix* (2004, for PlayStation 2 and XBox). I will not examine these other software titles, but focus in-depth on the three titles.

Magix Music Maker, *MTV Music Generator 2*, and *eJay Clubworld* feature slightly different approaches to making music. For example, in *Magix Music Maker* and *eJay Clubworld* sample files are placed directly to the multitrack window, while in *MTV Music Generator 2* sample files are first arranged into patterns that are then placed to the multitrack window. Table 1.1 presents an overview of the key characteristics of the three titles.

Title	*Magix Music Maker*	*MTV Music Generator 2*	*eJay Clubworld*
Developer	Magix Development	Jester Interactive	Unique Development Studios
Publisher	Magix	Codemasters	eJay Entertainment
Year of release in Europe	2001	2001	2002
Max. number of tracks	16	48	20 + in certain modes one additional track for polyphonic drum patterns or monophonic sequencer patterns
Number of pre-made samples files (as stated in product package or user's guide)	c. 3,000	c. 11,000 (or c. 3,660)[5]	c. 10,000
Number of pre-made patterns	--	c. 1,200	--
Sampling option	--	yes	--
Musical styles of sample files or patterns	"Hiphop", "Pop/Rock", "80's", "House", "2step", "Techno/Trance"	"Breakbeat", "Garage", "House", "Indie", "Pop", "Rock", "R&B", "Trance"	"Ambient", "Electro", "Reggae", "Hip-Hop", "Drum & Bass", "House", "Tech House", "Techno"
Sample file or pattern categories	"bass", "drums", "keys", "vocals", "fx", "pad", "guitar", "synth"	"rhythm", "bass", "melody", "vocals"	"bass", "keys", "males" (vocals), "females" (vocals), "drums", "guitar", "loop", "fx", "extra", "spheres", "vocal"

Table 1.1. Key characteristics of the three music-making software titles.

The three music-making software titles feature sixteen or more discrete tracks on which to make pieces of music. They have 3,000 or more prefabricated sample files included in the software. *MTV Music Generator 2* also includes 1,200 prefabricated sequencer patterns. Sample files and patterns cover several musical styles, although the emphasis is on

[5] According to the product package, about 11,000 sample files are included in *MTV Music Generator 2*. The software offers three versions of every sample file at sample rates "11", "22", and "44" KHz (these may or may not be the actual sample rates). It is not explained in the user's guide if these different versions are included in the total number of 11,000 sample files delivered with the software. If this is the case, the actual number of "unique" sample files is c. 3,660.

electronic dance music. Only *MTV Music Generator 2* supports sampling as an optional feature, which is enabled by an external device.

The primary reason I have chosen to call these 'music-making software' is to distinguish them from various music software for PCs. Personal computers have been used for composing music and recording sound since the 1980s. Today there are numerous applications available, including MIDI sequencers, digital audio multitrack recorders, sample editors, software-based synthesizers and drum machines, and complete studio systems equipped with special hardware. In this study, I will refer to music software products for video game consoles as 'music-making software' and music software for personal computers as 'music utilities'. Of course, the former are also applications and the latter are also software. Music-making software titles have similarities to various music utilities for PCs. Comparable features include similar user interfaces and features for recording and playback, programming patterns, mixing, and modifying sounds.

The key differences between music-making software and music utilities include the following:

- Music-making software titles do not support MIDI data formats for file storage or communication between pieces of multitrack studio equipment.
- Sample files from external sound libraries cannot be loaded into music-making software titles.
- No software extensions (like sound synthesis and sound processing plug-ins) or external hardware can be used with music-making software (with the exception of a sampler accessory for *MTV Music Generator 2*)
- Digital sound recording is possible only in *MTV Music Generator 2* and the recordings are short (the maximum length of one recording is some 23 seconds).
- Music-making software cannot display the music in standard notation nor print a score on paper.

For anyone familiar with making music with personal computers, the three music-making software titles probably appear as simplified versions of music utilities. The technical quality of the sample files in music-making software is not as good as in the sample libraries of music utilities, and the variety of sample files is limited. Some of the sample files sound rather silly or "tacky" (for lack of a more descriptive word). The technical quality of the sample files and prefabricated songs in music-making software is closer to that of early 1990s tracker[6] music than the "DVD quality" promised on the product package of

[6] 'Trackers' are music software developed by hobbyists (or as hobby undertakings by professional software developers) primarily for non-professional use. The first tracker software titles were developed around 1987 for Commodore Amiga computers. A number of tracker software have been developed over the years for several personal computer platforms. Tens of thousands pieces of music have been made with trackers by possibly hundreds of people. A lot of this music is available on various sites on the Internet. For an account of the demoscene, see Tasajärvi (2004). For a study of communities of tracker musicians, see Lysloff (2003).

eJay Clubworld. For the aforementioned reasons, the three music-making software titles can be initially interpreted as software toys rather than as comprehensive tools for music-making.[7]

1.3 Remediation and Semiotic Domains

One starting point in this study is to define video games and music-making software as media that have been partly formed through the process of remediation. According to the semioticians Gunther Kress and Theo van Leeuwen (2001: 22), "[m]edia are the material resources used in the production of semiotic products and events, including both the tools and the materials used […]". The game researcher Gonzalo Frasca (2003: 223-225) considers video games the first complex simulational media for mass audiences. Frasca notes that video games appear on the screen similar to and behave to some extent like the machines they simulate. According to the game and media researchers Mark J. P. Wolf and Bernard Perron (2003: 11), video games are different from earlier media because video games are the first medium to combine real-time game play with a navigable onscreen space through substitutes controlled by the player.

Juul notes that games are transmedial because many different media or tools can be used for playing certain games. In other words, games move between media. For instance, card games are played on computers, sports games are played on computers, and so forth. Games are not bound to particular sets of material tools, but to the processing of rules. Computers, for example, are a game medium. (Juul 2003: 4, 42, 50, 167.) Video games played on computers and video game consoles are media that make it possible for people to engage in the activity of play.

According to the media scholars Jay David Bolter and Richard Grusin, remediation is the representation of one medium or its contents in another medium. Remediation involves reforming or improving upon an earlier medium. (Bolter & Grusin 1999: 45, 59.) Remediation makes possible the combination of characteristics from different media, potentially resulting in the development of media hybrids. This is one starting point for this study as it considers how to bring characteristics of the multitrack studio medium to game console and video games media.

Bolter and Grusin (ibid.: 47, 89-103) suggest that computer games (and, in effect, video games) can be interpreted as a remediation of cinema because certain aspects of the medium of cinema have been adapted into the conventions of video games. King and

---

[7] The three music-making software titles also have other features in addition to those for making music. *Magix Music Maker* and *MTV Music Generator 2* allow the user to make music videos from a set of prefabricated video or computer animation clips and visual transformation effects. *eJay Clubworld* features a mode for displaying animations that respond in real-time to the music and the user's actions. *MTV Music Generator 2 and eJay Clubworld* feature multi-user "jamming" modes. In these modes sample files are triggered in real-time with the controller device's buttons. This way, a number of people can "jam" together by triggering sample files. I exclude video-making features and multi-user real-time modes of making music from the scope of this study.

Krzywinska (2002: 149), and Eskelinen and Tronstad (2003: 196) point out that this argument applies only to certain video games. Eskelinen and Tronstad (ibid.) emphasize that some video games cannot be interpreted as remediations of cinema, but can be considered as remediations of existing games such as pinball, card games, and sports. In some pinball video games even features like bumping and “tilt” (malfunction), which originate from pinball machines as physical and mechanical entities, are simulated with software.

Following the aforementioned definition of media by Kress and van Leeuwen (2001: 22), multitrack studio can be interpreted as a production medium of master sound recordings of music. The materials on master sound recordings are disseminated through recording and distribution media like audio CDs and MP3 files. Multitrack studio equipment has been remediated into music utilities for personal computers, which in turn have been remediated into music-making software for PlayStation 2. I will elaborate this argument in Sections 2.7 and 2.8.

Semiotic domains is another key concept in this study. The educational linguist James Paul Gee (2003: 18) defines a semiotic domain as ”[…] any set of practices that recruits one or more modalities (e.g., oral or written language, images, equations, symbols, sounds, gestures, graphs, artifacts, etc.) to communicate distinctive types of meanings.” According to Gee, being literate in a semiotic domain means that a person is able to recognize signs and what those signs stand for in a specific situation in the domain. Being a producer in a semiotic domain means that a person is able to produce signs that are recognizable to people who are literate in that domain. (Ibid.: 18-19, 32.) Gee considers video games to be a family of related semiotic domains (ibid.: 19). Being able to successfully play a video game requires that the player becomes literate in the semiotic domain of that game. A player can also learn how to innovate in the semiotic domain of the game and become a producer, for example, by making new game levels.

In this study, I interpret music as a family of semiotic domains, that is, sets of practices that recruit modalities including sound, oral and written language, and images to communicate various meanings. The meanings that can be communicated through music on sound recordings may vary from explicit to vague. One of the semiotic domains of music is the semiotic domain of multitrack composing, which includes, for instance, multitrack studio equipment, the knowledge people have of using that equipment (nonlinear multitrack composing techniques), and potentially all music made with that equipment. I will elaborate on the semiotic domains of multitrack composing and video games in Chapters 2 and 4 respectively.

1.4 Justifying the Approach

As I stated in Section 1.1, the research question of this study is the following: Is it possible to combine the activities of playing video games and music-making with the means of nonlinear multitrack composing? In other words, the goal of this study is to find out how the activity of nonlinear multitrack composing can be presented in the form of a video game. Why choose this approach?

The music technology researcher Tina Blaine (2005: 28) notes that as people have played video games since their youth, they may have had "[…] their first introduction and subsequent expectations about music-making based on early play experiences with interactive toys, video games, and other media forms rather than traditional musical instruments". It is worthwhile to consider combing music-making and video game playing because a lot of people are familiar with the user interfaces and other characteristics of video games. The same can probably not be said of the user interfaces and other characteristics of music utilities for personal computers. Video games are an established framework for playing and experiencing everything that is associated with the activity of play (e.g., fun, challenge, and excitement). Through proper design, music-making could be introduced as one such activity to video game players. The intent of this study is to define the design foundations for combining playing video games and music-making with the means of nonlinear multitrack composing.

Why choose music-making software for PlayStation 2 as research material? There are several platforms for playing video games, including personal computers, video game consoles, various portable devices, and arcade consoles. Even if personal computers and video game consoles are technically very similar, video game consoles have special uses that differentiate them from the more multipurpose PCs. Each PlayStation 2 game or software has a different graphical user interface (GUI) compared to the relatively uniform GUI in different software titles for PCs, for instance, with Windows operating system. PlayStation 2 has a generic controller device called DualShock 2 that is delivered with the console[8]. PlayStation 2 has a limited amount of system memory, and only a few external devices are available. Game designers have worked around these limitations and many excellent and versatile games have been developed for PlayStation 2. Sony PlayStation 2 was the best selling game console platform during the period I conducted this study. By the end of 2005, Sony had shipped more than 100 million units of PlayStation 2. This commercial success suggests that Sony has developed an excellent platform for games and other entertainment. Future video game console platforms are likely to attempt to replicate certain aspects of PlayStation 2.

The target hardware for the games which design foundations are explored in this study is a device without an alphanumeric keyboard and an interface to special controllers with organ keyboards and sliders. Thus, the future video games that involve nonlinear multitrack composing should be such that they can be used with a generic controller

[8] The DualShock 2 controller features two analog joysticks ("thumb-sticks" that also function as buttons), a directional pad with four buttons, and eleven additional buttons. These controls are described in Fullerton et al. (2004: 297).

device that is also used to play other video games. The three music-making software titles fit into these constraints, which are appropriate because the aim is to explore games and software for casual users rather than for music enthusiasts.

Music-making software titles are a special type of product for PlayStation 2. On the one hand, music-making software can be interpreted to be similar to music utilities for personal computers. However, because of the limited set of features and their poor technical quality, it seems that the target audience for the three music-making software titles is video game players rather than amateur or professional musicians[9]. Because they are intended for the PlayStation 2 console, music-making software titles probably have stricter obligations than music utilities for PCs to be easy and entertaining to use. For these reasons, a music-making software title could be regarded by its users as some sort of video game or software toy for making music.

When the three music-making software titles are interpreted as remediations of multitrack studio equipment, music utilities for PCs, and potentially certain other media (like video games and video editing software), it can be examined what conventions from the preceding media have been included in music-making software. These include, for example, some features from multitrack studio equipment and elements of graphical user interfaces from music utilities. In addition, there are some video editing software features in *Magix Music Maker* and *MTV Music Generator 2*. Thus, it is appropriate to consider in this study the possibility that also certain characteristics from video games could have been included to the three music-making software titles.

I have chosen the three music-making software titles as research material because music-making software is an exceptional type of software product for video game consoles, for which almost solely games have been developed. The target audience for music-making software appears to be game players, rather than professional or amateur musicians who have PlayStation 2 consoles. It seems reasonable to assume that in order for music-making software to be attractive to people playing video games, music-making software should present the activities of making music in ways that can be learned in similar ways as learning occurs in video games. In addition, the interest in using music-making software should be sustained over time in similar ways as video games sustain interest. One more reason for choosing them is that the three music-making software titles are not extendable: They have a fixed set of features, which can be shown through analysis and compared.

---

[9] In her study of informal musical activities in local settings in a town in central England, the music sociologist Ruth Finnegan argues that there is no simple way to define ‘amateur’ and ‘professional’ music activities. An uninformed definition is that an ‘amateur’ does music “for love” and ‘professionals’ full-time to earn a living. Finnegan argues that ‘professionals’ cannot be defined as people having music as their main source of livelihood. Finnegan discovered that in the local settings, the term ‘professional’ was often used to describe a high standard in musicianship rather than an economic position. Finnegan interprets the amateur/professional distinction as a complex continuum within which all practitioners of music can be called ‘musicians’. (Finnegan 1989: 12-15.) In this study, I use the term ‘music-makers’ to describe all people involved in music-making activities.

The three music-making software titles present different approaches to making music with PlayStation 2. Analyzing these three software titles is the best starting point to explore the design foundations of future video games that involve nonlinear multitrack composing. A software title cannot be used for everything, but only for what its features allow to do. The main purpose of use for the features of the three music-making software titles is obviously to make music, but it is far from obvious in what ways music can be made with these software titles. It is necessary to compare the three titles to each other in order to find what they have in common and to explain how their particular features can be used to make music.

To summarize, one starting point for this study is that music-making software can be tools for making music, software toys, or some type of video games. I will examine music-making software from these three viewpoints.

1.5 The Framework of the Study

This study is in part about making music with multitrack studio technology and playing video games. Multitrack studio equipment and the music made with it have been studied by scholars including Manning (1993), Gracyk (1996), Roads et al. (1996), Théberge (1997), Toynbee (2000), Taylor (2001), Pinch and Trocco (2002), Lysloff and Gay (2003), and Katz (2004). Recent multitrack studio equipment includes new media tools such as computer software. The theory of new media (Manovich 2001) provides a technological context for nonlinear multitrack composing. I use the theory of remediation (Bolter & Grusin 1999) to explain how music-making software titles are compatible with other multitrack studio equipment. Some key studies and publications on play, games, and video games include Huizinga (1980 [1949]), Caillois (1961), Frasca (2003), Juul (2003), Fullerton et al. (2004), and Salen and Zimmerman (2004). For a review of earlier studies of play, games, and in particular video games, see Juul (2003: 9-16).

I use previous studies to define the starting points concerning the semiotic domains of multitrack composing and video games. I also use previous studies and some other sources to define nonlinear multitrack composing techniques and a metaterminology of functions.

The framework of this study is presented in Table 1.2.

<table>
<tr><td colspan="2">Key research questions
• What are the similarities and differences between music-making software and video games?
• What are the similarities and differences between making music with nonlinear multitrack composing techniques and playing video games?
• What are the design foundations of future video games that involve nonlinear multitrack composing?</td></tr>
<tr><td>Studies and literature of music-making with multitrack studio technology
• Defining multitrack composing as a music-making activity
• Defining a five-phase model of the nonlinear multitrack composing process
• Defining a metaterminology of functions
• Defining nonlinear multitrack composing techniques
• Base definitions on Goodwin (1990 [1988]), Cary (1992), Frith (1996), Gracyk (1996), Roads et al. (1996), Wishart (1996), Huber and Runstein (1997), Théberge (1997; 2001), Bartlett and Bartlett (1998), Grimshaw (1998), Cutler (2000), Taylor (2001), Lysloff (2003), and others</td><td>Studies of play and video games
• Defining semiotic domains: Gee (2003)
• Defining play and games: Huizinga (1980 [1949]), Caillois (1961), Juul (2003), Frasca (2003), Perron (2003), Salen and Zimmerman (2004), and others
• Defining the formal elements of video games: Costikyan (2002), Juul (2003), Fullerton et al. (2004), Salen and Zimmerman (2004), and others
• Defining the dramatic elements of video games: Costikyan (2002) and Fullerton et al. (2004)
• Defining video games as systems and simulations, and mimicry in play: Caillois (1961), Costikyan (2002), Frasca (2003), Juul (2003), and others</td></tr>
<tr><td colspan="2">Analyses</td></tr>
<tr><td>Part 1
• Describing three music-making software titles and outlining the need for a metaterminology of functions
• Comparing features to functions
• Defining nonlinear multitrack composing techniques and showing how they can be realized with music-making software</td><td>Part 2
• Identifying the similarities and differences between music-making software and video games:
o in the formal elements of video games
o in the dramatic elements of video games
o in definitions of video games as systems and simulations, and mimicry in play</td></tr>
<tr><td colspan="2">Results
• Definitions of the semiotic domain of multitrack composing and nonlinear multitrack composing techniques
• Awareness of the similarities and differences between making music with music-making software and playing video games
• Design foundations of future video games that involve nonlinear multitrack composing</td></tr>
<tr><td colspan="2">Interest of knowledge
• Scholarly
• Research for design</td></tr>
</table>

Table 1.2. The framework of the study.

The interest of knowledge in the present study is both scholarly and R&D oriented. It is scholarly in the sense that the study attempts to provide an awareness of the semiotic domain of multitrack composing and of music-making in video games.

The game researchers Marinka Copier and Joost Raessens (2003: 9) identify six approaches to studying computer games (and, in effect, video games): Games and their properties, design, reception, games as a cultural phenomenon, games as a social phenomenon, and research methods. Of these approaches, the focus of this study is on design. More precisely, the interest of this study could be called ‘research for design’ (see Lunenfeld 2003: 11). The game researchers Jussi Holopainen and Aki Järvinen (2005: 55) note that the objective of research for design “[…] is to come up with systems and models that showcase and validate the results of the research.” Such systems and models are potential future outcomes of this study, but not an integral part of it.

The purpose of this study is to be a prestudy that maps out the design foundations of future video games that involve nonlinear multitrack composing. This study also defines a method for comparing music software that can be used in later studies or design projects. Thus, this study also contributes to software and game design methods. Questions of usability and playability, market feasibility, and the legal issues of licensing sample files and existing music are beyond the scope of the present study.

1.6 How the Study is Conducted

This study is divided into two principal parts. The first part (Chapters 2 and 3) examines the characteristics of the three music-making software titles, how they relate to the semiotic domain of multitrack composing, and what the characteristics of that semiotic domain are. The second part (Chapters 4 and 5) examines the similarities and differences between video games and music-making software, and between making music with the means of nonlinear multitrack composing and the different approaches to playing.

To obtain a cross-section of existing software and video games, I have selected as research material three music-making software titles (*Magix Music Maker*, *MTV Music Generator 2*, and *eJay Clubworld*) and seven music-themed video games (*Amplitude*, *Beatmania*, *Fluid, Get on da Mic, Gitaroo Man*, *SingStar*, and *100% Star*) for PlayStation 2. I will describe, analyze, and compare them systematically in order to identify the key similarities and dissimilarities between music-making software and video games.

This study progresses as follows:

In Chapter 1, I have declared the purpose and interest of knowledge of this study. The results of this study are background data and design foundations for future video games.

In Chapter 2, I define the starting points concerning the semiotic domain of multitrack composing. I describe the background of nonlinear multitrack composing and suggest how it relates to the technological context of new media.

In Chapter 3, I report the first part of the analysis. The purpose of the analysis is to demonstrate the common functionality of the three music-making software titles, to show

how music-making software can be used to realize nonlinear multitrack composing techniques, and to exemplify how music can be made with these techniques. I present the methods of analysis at the beginning of the chapter.

In Chapter 4, I define the starting points concerning the semiotic domains of video games, including the definition of play and games. In Section 4.5, I describe the main characteristics of the seven music-themed games that I use as comparison material.

In Chapter 5, I report the second part of analysis. The method is to examine the three music-making software titles through different aspects of video games in order to demonstrate the similarities and differences between video games and music-making software, as well as between playing video games and making music.

In Chapter 6, I summarize the similarities and differences between video games and music-making software, and as conclusion, present the design foundations of future video games that involve nonlinear multitrack composing. The similarities and differences observed between video games and music-making software are addressed in the design foundations.

Some remarks on the conventions I have used in this study: I have placed the names of video game features and music-making software features in double quotation marks (e.g., "hyper generator" in *eJay Clubworld*), while I have placed terms in single quotation marks (e.g., the metaterm 'pattern sequencer'). Music-making software titles, video game titles, and names of publications and musical works are in italics. In verbatim quotes, italicized emphases are from the original texts. I have used the term 'he' as a neutral pronoun, as suggested by the lexicographer Sidney I. Landau (1989: 3), who recommends that "[…] men use masculine pronouns for neutral use, because they naturally identify with the masculine gender, and that women use feminine pronouns for the analogous reason."

2. The Semiotic Domain of Multitrack Composing

2.1 Music and Semiotic Domains

As I suggested in Chapter 1, following Gee (2003), music can be interpreted as a family of related semiotic domains. Semiotic domains are cultural and historical creations that attempt through their content and social practices to persuade people to think, interact, value, and feel in certain ways (ibid.: 43-44). According to Gee (ibid.: 23), semiotic domains are typically "[…] shared by groups of people who carry them on as distinctive social practices." For example, multitrack studio equipment is developed by people who have certain concepts (distinctive social practices) of how this equipment may or should be used in music-making. In addition, certain musicians may share with the developers their concepts concerning the uses of multitrack studio equipment. The concepts of music-making that these people share are different from the concepts shared, for instance, by people involved in historically informed performances of Baroque music. The semiotic domain of multitrack composing includes distinctive concepts of music-making, of what music is, what can be conveyed through music, and of what music can be.

In Chapter 2, I define the semiotic domain of multitrack composing within which, for instance, people making music with multitrack studio technology are literate and producers (see ibid.: 18-19, 32). In order to map out the characteristics of this domain, I describe the development of sound recording and multitrack studio technology, and evince an explanation of how and why old inventions of studio technology have been recently adapted to computer software.

2.2 Music as Recorded, Modified, and Reproduced Sound

During the twentieth century, major changes occurred in the options for composing and performing music. The music technology scholar Paul Théberge (2001: 3) argues that technology has become a precondition for music-making in popular music. The musicologist Timothy D. Taylor (2001: 139) goes as far as suggesting that it has become "[…] safe to say that there is virtually no contemporary music that does not make use of some kind of electronic technology, whether or not listeners can discern it." The various styles of electronic dance music (e.g., house, techno, and breakbeat) are examples of music that would not exist without electronic music technology.

Before the introduction of sound recording, composing a piece of music was often an activity separate from its performance. In particular, Western composers in the domain of art music used written notation when composing music. This music-making activity is composing on the pitch lattice, which involves defining a piece of music separately from the sounds through which it will eventually be heard. I base the term 'pitch lattice' on the terminology of the composer Trevor Wishart (1996: 8), who has presented the broader concept of 'lattice sonics'. Western musical notation is a two-dimensional lattice of time and fixed pitches (ibid.: 23). As the recording studio became the primary site for making music, especially in rock and pop music, the notated score became superfluous as a means of composing music (Théberge 2004: 143).

Before the era of sound recording, musical performance was required in order to experience music audibly. People either attended concerts or performed the music themselves at homes. Each performance of a musical work varied with the number and type of used instruments, the skill of the performers, and the acoustics of the location of the performance. (Fikentscher 2003: 291.) With the introduction of sound recordings, practical music-making skills became less important. The music sociologist Simon Frith (2001: 30-31) suggests that the gramophone was different from musical instruments because it enabled anyone to “play” music without practicing musical performance skills. Thus, “[m]usic could be made at home as a matter of consumption rather than technique” (ibid.: 31).

The early developments of sound recording have been extensively documented, for instance by Oliver Read and Walter Welch (1976), and Jonathan Sterne (2003). Several innovators developed sound recording methods during the last decades of the nineteenth century. During the twentieth century, sound records became almost synonymous with music in industrial societies (Gronow 1996 [1983]: 56). Most of the music listened to today throughout the world originates from records, either directly or mediated through radio and television (Gronow 1996: 5). Sound recording has become the dominant mode in the production of music (Théberge 1997: 244). In the present day, music is composed more for purposes of reproduction (“replay”) from records than as material for live performances. Sound recordings can be duplicated and distributed through various technologies over vast geographical distances. Through sound recordings, music made in one place can be listened to in another place and at a different time.

Several scholars and writers have discussed in detail the history of making music with sound recording technology (Cunningham 1998; Gronow & Saunio 1998). Since the first decades of the twentieth century, sound recordings have typically been made in recording studios, which are spaces specifically designed for this purpose. Sound recording studios have become spaces for special setups of equipment and work practices.

During the first half of the twentieth century, it was largely a technical necessity to record complete live performances of pieces of music. Before the introduction of sound-on-sound recording, songs were composed, arranged, and rehearsed before the recording session. During the recording session, it was common for the song to be played through in its entirety a couple of times and these ‘takes’ were recorded. The musicologist Jason Toynbee (2000: 82-83) calls this a ‘documentary mode’ of recording. Each take was a document of a complete performance of a piece of music. The take that was considered the best was released to the public. Even before the era of multitrack recording began, there had been experiments with sound recording in ways that cannot be characterized as ‘documentary’.

Editing a sound recording involves removing segments, changing their order, and assembling a complete piece of music from various takes recorded at different times. Editing already became technically possible with optical sound film recording (Chanan

1995: 141). Nevertheless, editing was not accepted as common practice until the 1940s with the introduction of magnetic tape recording (Cunningham 1998: 25-32).

There were experiments with sound-on-sound recording throughout the 1920s and 1930s (Cutler 2000: 95). Two-track (stereo) tape recorders were commercially introduced in the early 1950s. The first serial produced four-track multitrack recorder was introduced in 1958 (Fikentscher 2003: 293). Between the mid-1960s and the late 1970s, the multitrack recording capabilities of one device gradually expanded from eight, to sixteen, to twenty-four tracks, and beyond (Negus 1992: 25). It was known from early on how to synchronize two or more multitrack recorders in order to increase the number of tracks available for recording.

Sound-on-sound multitrack recording made it possible to record separate parts of a song onto different tracks at different times. When these are mixed into a mono or stereo recording, the final version of the piece of music can be constructed by editing.

In addition to new equipment for recording sound, novel devices for modifying sound were also developed. Apparently some methods for modifying sound were discovered by accident, such as the fuzz effect that was supposedly discovered by musicians who played electric guitars through damaged loudspeakers (Gracyk 1996: 122-123). The fuzz effect was perceived as sounding good, so these incidents led to the development of specific devices for producing fuzz effects.

The complexity of the recording studio increased during the 1950s and 1960s when different types of sound processing equipment were introduced. This equipment includes reverb units, filters, compressors, echo devices, distortion units and many more. In the mid-1960s, innovators such as Robert A. Moog and Donald Buchla and their associates combined the essential components of an electronic music studio into their modular synthesizer systems. (Manning 1993: 117-155.)

The uses of multitrack studio equipment in music-making brought new forms of collaboration between musicians, producers, and engineers. Musicians began to have control over the techniques of recording, and thus the sound of their music. Some producers became closely connected to the musicians they worked with and some producers came to be recognized as artists. (Théberge 2004: 141-143.) The role of the sound engineer changed from that of a technician to a creative artist who contributed to how the recorded music sounded (see Kealy 1990 [1979]).

The music technology scholar Mark Grimshaw argues that the power of a recording studio is in the hands of those who can mould sound, not simply record it. In the 1950s and 60s, unions and regulations could prevent anyone except sound engineers from having access to studio equipment. For this reason, some musicians built their own studios in their homes in order to become familiar with the equipment. (Grimshaw 1998: 129.)

Inventions originally made in the fields of computer and digital technology were gradually transferred to multitrack studios. For example, digital multitrack recording was first introduced in 1978 in a commercial product. The first computer-based sound editing system was developed by the Soundstream Corporation in the early 1980s. (Roads & Strawn 1996a: 13.) Computers were adapted for multitrack studio use first as control devices (sequencers, synthesizer and sampler patch editors) and some years later as sound recording, editing, processing, and synthesizing devices.

One way to look at the development of tape editing, synthesizers, multitrack recorders, and other studio equipment is to interpret them as outgrowths of the desire to extend the scope of music-making beyond the performance limitations of human neuromuscular skills (Rhea 1984 [1979]: 3). The pianist and composer Glenn Gould (2004 [1966]: 118) remarked that by taking advantage of the possibility to assemble a piece of music from two or more takes, it is possible to "[…] transcend the limitations that performance imposes upon the imagination." In addition to putting together the best segments of several takes, multitrack studio equipment enabled a precise control of such parameters as amplitude, timbre, modulation, and timing. Indeed, the introduction of various kinds of multitrack studio equipment offered new opportunities for making music. From the late 1950s onwards, producers like Ahmet Ertegun and Phil Spector approached music on records as sculptured sound, and not as real-time performances that were recorded (Gracyk 1996: 16).

In his study of the aesthetics of rock music, the philosopher Theodore Gracyk (1996) argues that with sound recording, musicians working in studios began to compose with sound instead of written notation. Songs were composed during the recording sessions, by exploiting the possibilities of the studio. (Ibid.: 12.) During the 1960s, studios became something different from locations where already written and well rehearsed songs were recorded in a 'documentary mode'. Instead, songs were extensively worked on in the studio using various equipment.

The producer and artist Brian Eno (1998: 17) suggests that a profound change occurred in how the recording studio was perceived:

> Until the late Sixties, the recording studio had been a passive transmitter. A band went into the studio with the aim of getting a well-rehearsed, pre-existing song on to tape as faithfully as possible. "Fidelity" was a big word then. But after witnessing the achievements of a handful of visionaries, people began to see the studio as a labyrinth of possibilities and started experimenting with echo, reverb and tape repeats to make more of their sound, something that didn't originate from the band.

In contrast to the earlier documentary mode of sound recording with complete takes, Toynbee (2000: 55) defines modern popular music as 'processive music' that is constructed in a sequence of multiple takes, overdubs, and editing. The core activity of a musician, performing music, became fragmented as a consequence of sound-on-sound recording, recording multiple takes, and editing. Takes were typically short fragments of the piece of music, not necessarily complete musical phrases. (Chanan 1995: 144.)

According to Frith, with technological means such as editing and multitracking, a recording ceased to refer to a specific performance and came to refer to a constructed performance. The "original" ceased to be the record of an event (a performance) and became an idea. (Frith 1996: 234.) In a similar way, the philosopher Evan Eisenberg notes: "The word 'record' is misleading. Only live recordings record an event; studio recordings, which are in the great majority, record nothing. Pieced together from bits of actual events, they construct an ideal event." (Eisenberg 1988: 89.)

Toynbee suggests that the manipulation of sound in recording studios led to a shift from the documentation of a performed song to the construction of a 'song-sound'. Production values are central in the construction of the 'song-sound'. (Toynbee 2000: 89.)

The ethnomusicologist Kai Fikentcher (2003: 293) defines sound recordings made with multitrack technology as a 'sonic score', which he considers an authoritative text. Indeed, recording made each musical performance as permanent as the written score. Sound recordings capture more of a musical performance than what can be described with written notation. Frith (2001: 31) suggests:

> The most important effect of recording technology […] was that it allowed music to be stored and retrieved as it was performed rather than as it was written […] Recording […] allowed the mass distribution for repetitive listening of the particularities of a specific performance. Musical qualities such as timbre, tone and rhythmic individuality could now be stored as effectively as melodies and harmonies had been by written notes.

Gracyk (1996) argues that on records, the musical work is the sound recording. Music on sound recordings should be considered as a different kind of art from music performed live. From the 1950s onwards, recording technology allowed producers, engineers, and musicians to experiment with various approaches to sound and thus develop their ideas over time. They could listen to and analyze earlier versions of the recorded song. The arrangement is just one aspect of the aesthetic impact of a piece of music. The way a recorded piece of music sounds is a result of decisions made during the recording process. Thus, Gracyk (ibid.: 16) argues that all qualities that can be heard from a piece of music on a sound recording are "[…] expressive elements and relevant features of the musical work".

When a piece of music is made in a studio, all sounds can be relevant elements to that musical work, including sounds that were previously considered not to belong to the domain of music. As an example, Gracyk (ibid.: 56) discusses a song by the hip hop group Run-D.M.C:

> Run-D.M.C. may have been the first to sample records that carried the unmistakable sound of damaged vinyl, with "Peter Piper" (1986); audible clicks and pops were soon in vogue on rap releases. […] While Run-D.M.C. did not place each individual popping sound, they chose to feature the sounds.

Gracyk (ibid.) argues that details such as the clicks of a vinyl record or the squeak of a piano pedal are part of the work of a recorded piece of music if it has been in the artists' power to include or exclude these details from the released versions of this music. Gracyk argues that every sound that can be heard from a sound recording should be treated as deliberate and relevant. For example, "[i]f a performer coughs during a live performance, we dismiss it as an irrelevancy, but if someone coughs on a record, our knowledge that the cough could have been deleted (or another “take” released) invites us to regard it as part of what is communicated.” (Ibid.: 79.)

In other words, what details of a piece of music on a sound recording are aesthetically relevant is a matter of knowing which details were under the artists’ control during the making of the recording. (Ibid.: 56-57.)

Théberge’s (1997: 244) observation that sounds, rather than instruments, are coming to be understood as the basic components of music is an important one in defining the semiotic domain of multitrack composing. A synthesizer may have thousands of preset sounds, so the music-maker *must* select the sounds he considers suitable for each piece of music. In a similar way, there is no fixed set of sounds on a sampler, but the desired sounds have to be selected and loaded or recorded onto the device. The sounds must also be legally cleared if they are to be used in recordings that are released to the public.

2.3 Composing in the Multitrack Studio

Théberge (2004: 139) argues that during the latter half of the twentieth century, what it means to be a musician has changed in many ways. Théberge argues that composers, performers, producers, engineers, synthesizer programmers, turntablists, sample artists, mixers, remixers have become ‘technologists’, that is, “[…] individuals whose knowledge and skill in the uses of the recording studio, and the various technologies associated with it, have become as important as their knowledge of music, style, or vocal and instrumental performance”. (Ibid.) For these people making music in multitrack studios, a sound is a modifiable building block of music.

Théberge (2001: 10) describes the environment of the multitrack studio as follows:

> The multitrack tape recorder is one of the principal technical devices within an overall technical environment - the multitrack studio - that needs to be understood as a ‘technology’ in larger terms. The studio is an environment made up of a specifically designed set of acoustic spaces within which one finds a wide range of technical devices: microphones, tape recorders, mixing console, signal processors, monitors, headphones and, more recently, digital samplers, synthesisers and computers.

A typical multitrack studio has a great number of different types of equipment varying from tape recorders to patch bays. To name one systematic approach to classify this equipment, the sound engineer David Gibson (1997: 28) has suggested a practical categorization of studio equipment into five groups, namely sound creators, sound routers, sound storers, transducers, and sound manipulators.

Various recording studio equipment such as multitrack recorders, effects devices, sequencers, and samplers - many of which are also used separately outside the studio - are integrated into, and are used as, a single entity. That entity is the multitrack recording studio as a compositional tool (see Eno 2004). The multitrack recorder, its controller, and a mixer are at the center of this compositional tool (Théberge 1997: 215-231). Since the 1980s, most of this equipment has been subjected to computer control and from the beginning of the 1990s, most components of the multitrack studio have been realized as computer software. Théberge (ibid.: 227, 244) notes that the recording controller and mixing desk are the centerpiece of the software much as they are in studios with hardware equipment.

2.4 Functions and Techniques

In my definition, pieces of multitrack studio equipment have certain 'functions'. I use the term 'function' as Richard Dobson (1992) does in his publication *A Dictionary of Electronic and Computer Music Technology*. For Dobson, a drum machine, a sampler, and a sequencer have different functions. There are for instance compression, expansion, filtering, reverb, and signal-processing functions. To name an example, the basic functions of an analog or digital tape deck are sound recording and reproduction. Samplers also have these two functions, even if the maximum duration for the sound recordings is more limited.

A piece of multitrack studio equipment can have characteristics that distinguish it from other similar pieces of equipment. These characteristics can be defined in technical terms such as frequency response, distortion, sampling rate, and sampling frequency. However, the important consequence of these special characteristics is that certain pieces of equipment sound different from others in subtle ways. This special sound can be a property of the sound sources (e.g., synthesizer and drum machine) or caused by the transducer characteristics of these devices (e.g., effects device, microphone, amplifier, and sampler). Few pieces of multitrack studio equipment are truly transparent[10] as most of them "color" in certain ways (e.g., by causing a desired type of distortion) the audio signal that passes through them. This is especially true of older equipment, and one reason why "vintage" equipment is still used today.

The introduction of new equipment does not necessarily render earlier equipment obsolete. To name one example, technically a 16-bit sampler with 48 KHz sampling rate is better than an 8-bit sampler with 20 KHz sampling rate because it offers a better dynamic range and accuracy. However, the "coarse" sound coming from an 8-bit sampler can still be useful, for example in those types of music that strive for a low-fidelity aesthetic. The same can be said, for instance, of drum machines with 8-bit vis-à-vis 16-bit samples. The composer and electronic music scholar Simon Waters (2000: 67-68) suggests that there is in sound reproduction a continuum of resolution of high fidelity/low fidelity, which he considers analogous to the spatial continua of left/right and distant/present.

Because of sound recording, contemporary audiences can hear for example what types of sounds were produced with analog synthesizers in the late 1960s or how samplers were used in the early 1980s. Théberge (1997: 204) suggests that sounds have the "[…] ability to act as a kind of referent for the object which is the source of the sound […]." Recognizing the characteristic sound of a piece of multitrack studio equipment is a

[10] For discussions on the "myth of transparency" and "perfect fidelity" in audio technology, see respectively Truax (2001: 121-123) and Sterne (2003: 215-286).

part of the knowledge about that particular device. This knowledge is put to use when the device or similar devices are used to make music.

Several authors have discussed ‘recording production techniques’ (Nazarian 1988) or ‘recording techniques’ (Huber & Runstein 1997; Bartlett & Bartlett 1998). For instance, Bruce Nazarian (1988: 9) defines techniques like tracking, overdubbing, mixing, editing, and mastering. The composer Barry Schrader (1982) discusses what he calls ‘tape composition techniques’ that include loops, editing, speed change, direction change, and delay. Techniques like these are established and socially shared ways of using certain pieces of multitrack studio equipment in specific ways, at various stages of music-making in a multitrack studio.

My definition of ‘multitrack composing techniques’ is as follows: A technique is realized when a person makes music by using in a certain way a single function or several functions of one or several pieces of multitrack studio equipment. Realizing a multitrack composing technique often produces a distinct audible outcome. Techniques emerge as music-makers work with different pieces of studio equipment. The characteristics of a piece of equipment influence what can be done with it. In some cases, the invention of a new technique has been a response to overcome the limitations of a piece of studio equipment. Artists have turned the limitations into new possibilities with regard to sound, form structure, using sounds to refer to things outside the piece of music, and so forth.[11]

Techniques invented more than 50 years ago, such as slap back echo (see Toynbee 2000: 85-86), can still be useful today. Techniques are taught as part of a recording engineer’s education and of studio work training for musicians. Some techniques may be learned simply while making music with the equipment.

Frith (2001: 244-245) argues that people pay attention in new ways to the music made in multitrack studio:

> Multi-track recording […] has had a long-term aesthetic effect: we now hear (and value) music as layered; we are aware of the contingency of the decision that put this sound here, brought this instrument forward here. It’s as if we always listen to the music *being assembled* […].

Following Frith, it could be suggested that appreciating music made with multitrack studio technology involves valuing the artists’ realizations of all multitrack composing techniques, not just mixing. In the final mix of a sound recording, sounds may have been used the way they were originally recorded, but they may also have been heavily modified with equipment like equalizers, compressors, and effect processors.

Artists can use multitrack studio equipment in ways the designers of the devices may not have thought of. New techniques can be invented when pieces of multitrack studio

[11] The musicologist Helga de la Motte-Haber (2002 [2000]: 200) notes that often in the history of music, “[...] the artist’s intentions [have] preceded the possibilities of technical realization.” These artists may then have worked with instrument builders, and more recently with engineers, to develop new musical instruments and pieces of equipment that enabled to realize the artist’s intentions.

equipment are experimented with. When artists invent new uses to an existing piece of equipment, it can be argued that they invent new techniques for that device. Théberge (2001: 15) describes one such case: “Using innovative techniques such as mixing and ‘scratching’, dance club deejays transformed the turntable, a quintessentially reproductive device, into a *productive* one, a musical instrument of the first order.” Selecting and mixing musical material (choosing what sounds are heard together), and scratching (manipulating the rhythm and pitch of the selected material) are some of the means of using turntables as musical instruments.

The cultural scholar Tricia Rose (1994: 63) argues that technology has rarely been adopted by black artists in a straightforward fashion, but has been picked up and revised in keeping with long-standing black cultural priorities. For instance, the uses of music technology in making of rap music are linked to the poetic traditions, oral forms, practices, stylistic and compositional priorities found in black cultures. According to Rose, black artists' approaches are directed in particular towards sound, rhythm, timbre, motion, and community in music. (Ibid.: 64, 84, 95-96.)

Rose (ibid.: 74-75) argues that creative rap musicians and producers approach sound manipulation in ways that are different from the originally intended uses of studio equipment such as samplers:

> Using the machines in ways that have not been intended, by pushing on established boundaries of music engineering, rap producers have developed an art out of recording with the sound meters well into the distortion zone. When necessary, they deliberately *work in the red*. If recording in the red will produce the heavy dark growling sound desired, rap producers record in the red. If a sampler must be detuned in order to produce a sought-after low-frequency hum, then the sampler is detuned. Rap musicians are not the only musicians to push on the limits of high-tech inventions.

All musicians, record producers, and sound engineers can hear from sound recordings the audible results of techniques and adapt these techniques in their own music. It is safe to assume that black rap artists are not the only musicians who adapt musical technologies on the basis of their cultural priorities. According to the musicologist Alan Durant (1990: 180):

> Music is never simply led by technological invention, as is suggested in crude forms of technological determinism. Rather, technological developments take on their particular character only in specific instantiations within prevailing, but also changing, social relations and contexts. They lead to networks of reactions, responses and effects that cannot be predicted merely from the resources or design of the technology itself.

In a similar way, the musicologist Richard Middleton (1990: 90) argues that technology and musical technique, content and meaning in music, generally develop together in a dialectical fashion. Théberge (1997: 159) suggests that technologies become musical instruments through their *use*, not through their form. Musicians refine music technologies through unexpected and alternative uses (Théberge 2001: 3). Théberge (1997: 160) proposes that musical instruments are never completed at the stage of design and manufacture, but that they are "made-over" by musicians when used to make music. A musical instrument, such as the electric guitar, has been redefined several times, with innovations to new variations to sound modifying and performance techniques (see Walser 1993). The same can be said of pieces of multitrack studio equipment that are "made-over" when musicians invent new multitrack composing techniques that can be realized with the equipment. Following Middleton, Théberge, and Walser, it is appropriate to argue that pieces of equipment become musical instruments through the ways in which they are used to make music. For this to happen and become a broadly accepted practice, changes in perceptions and concepts of music are necessary.

In my definition, techniques are separate from specific pieces (for example, models) of multitrack studio equipment. Certain techniques can be realized with certain types of multitrack studio equipment. To name one example, the technique of replaying a recorded sound in reverse can be realized with analog magnetic tape, a sampler, or computer software. Another example is echo, that is, delaying a sound for some 50 ms or longer and blending this with the original sound. The technique of adding echo to a sound may be realized with a tape delay machine, a digital delay line, or computer software. The effect of echo that can be heard on sound recordings is the audible outcome of this technique.

Functions are not the same thing as techniques. The functions of a piece of multitrack studio equipment form a potential that may or may not be taken up by the music-makers by using them to realize techniques. Certain techniques may be realized in a straightforward fashion by using a single function. Some techniques may be realized by using a couple of functions simultaneously or successively. Some functions and techniques may be described with the same term. This is the case especially when a technique can be realized in a straightforward fashion by using a single function.

When a new technique has been invented for a piece of equipment, the realization of that technique may be facilitated in future versions of that device, for instance, as a predefined feature.

The functions of a piece of multitrack studio equipment may also be used for other purposes than music-making, for instance in the production of movie soundtracks.[12]

2.5 Multitrack Composing as a Music-Making Activity

Scholars including Jones (1992), Negus (1992), Théberge (1997; 2001), and Toynbee (2000) suggest that the multitrack recording studio has become a compositional tool, an entity rather than a collection of equipment for different purposes. According to Théberge (2001: 9), "[m]ultitrack recording is not simply a technical process of sound production and reproduction; it is also a *compositional* process and is thus central to the creation of popular music at the most fundamental level […]." This may be the case with the making of

[12] Multitrack composing techniques and associated pieces of equipment may be invented, refined, and dispersed, for instance, the way the musicologist Olivier Julien (1999) discusses the case of automatic double-tracking.

many forms of popular music, but not all of them. Pieces of music are not simply recorded in the multitrack studio; editing, mixing, and sound processing are also essential in making music in the multitrack studio. As discussed earlier, it is a common practice that songs are not composed and arranged beforehand, but during the working process in the studio. The music journalist Simon Reynolds describes the process of making electronic dance music as follows: “Once, it was possible to distinguish between music and its production, between the songs and the recording tricks with which it’s embellished. But with dance tracks, the music *is* the production.” (Reynolds 1998: 376.) It could be argued more broadly that in the multitrack studio, composing, recording and production have blended into one activity, namely multitrack composing.

The soundscape scholar Barry Truax (2001: 231) argues:

> With the exception of composers who become instrument builders […], composers of instrumental music generally use the sounds available to them, rather than design the sound directly. […] In the practice of electroacoustic music, the roles of instrument builder, performer, and composer become much more closely linked.

The means of what Truax calls electroacoustic music are used in many different styles of music. When making music with synthesizers and samplers, certain sounds must be selected or designed by the music-maker, because these musical instruments do not have a single fixed timbre like most acoustic instruments.

The popular music scholar Keith Negus (1992: 31-32) argues that sound recording equipment is “an instrument of composition” and not merely a means for reproducing musical works. Few studios have identical equipment setups and the setup of a particular studio may not remain the same for long as new devices are brought in. It is how these various pieces of equipment are used as an entity that renders a multitrack studio into an instrument for composing music. As I suggested earlier, the multitrack studio can be interpreted as one complex instrument because it is an amalgamation of various types of equipment integrated for the purpose of making music (of course, radio plays, for instance, can also be made in a multitrack studio).

The traditional occidental concept of ‘composing’ music or the view of the activities of a ‘composer’ are no longer sufficient to describe music-making in the multitrack studio. My suggestion is that practices that were previously called ‘production techniques’ should now be considered to be multitrack composing techniques because they are essential in making music with multitrack studio technology. The outcomes of multitrack composition are ‘pieces of music’, ‘songs’, or simply ‘music’. In this study, I refrain from using the term ‘composition’ in the sense of a ‘musical composition’. I also propose a distinction between music-making activities of real-time ‘performance’ and non-real-time ‘composing’.

I define ‘multitrack composing’ as a music-making activity, which is a combination of three types of practices:

1. Practices of songwriting (including techniques of composing through written notation, arranging, and writing lyrics)

2. Practices of performance (including techniques of performing music on various instruments from written notation, memory, or by improvising)
3. Practices of sound recording production (including, for example, techniques associated with recording, overdubbing, mixing, sound processing, and editing).

All these practices are necessary in order to make music in the multitrack studio. Obviously, not all of these do have to be mastered by every person working in the studio. However, for example a musician working alone in a home studio must have some knowledge of most of these practices.

In addition to these practices, there are other practices associated with professional sound recording production, such as planning recording sessions, hiring musicians for the sessions, the division of labor during recording sessions, people management skills, documenting the recording session, and economic considerations.

To summarize the discussion so far, I have described the multitrack studio as a complex entity for making music. I have proposed the term 'multitrack composing' as an overall description covering the various tasks that are undertaken when making music in a multitrack studio.

2.6 Multitrack Studio Technology in the Context of New Media

I noted earlier that the components of a multitrack studio have been realized as software for personal computers. My starting point is that certain aspects of music utilities for personal computers and music-making software for video game consoles are best understood through the technological context of new media[13].

According to Manovich (2001: 6), the essence of new media is computer-based representation and production processes. With appropriate software, personal computers and video game consoles like PlayStation 2 are multi-purpose tools that can be used to store, duplicate, generate, manipulate, and distribute material from every medium that can be represented digitally (ibid.: 25-26).

Manovich defines digital representation as an umbrella term for three concepts, namely 'digitization' (analog-to-digital conversion), 'a common representational code', and 'numerical representation'. Digitization is analog-to-digital conversion. Combining material from different media into a single file is possible because of the use of a common representational code (binary code). Numerical representation turns media material into computer data, and makes it programmable. Numerical representation makes it possible to use computers as media synthesizers (with which any material can be generated from

[13] My definition of new media is based on Manovich (2001). Concerning this term, a valid question is how "new" such media is within a couple of decades. New media in the definition of Manovich should be interpreted as an art historical term for computer media of the very late twentieth and early twenty-first centuries, much as certain media of the 1920s that were also described as "new", like new vision and new architecture.

scratch), as media manipulators (for modifying and duplicating existing material), and as media distributors (files can be sent from one computer to another when they are interconnected).[14] (Ibid.: 25-26, 52.)

Users treat image, sound, video, and other media files as 'new media objects'. These files are numerical representations of media generated from scratch (or, rather, through preprogrammed algorithms) or digitized from analog media sources (print media, photographs, vinyl disc, magnetic tape, etc.). (Ibid.: 27.) There are a couple of key operations (like selecting, compositing, copying, pasting, deleting) of computer-based authorship that are similar regardless of whether the material is images, audio, or text (ibid.: 142). If a person learns these operations when authoring material in one medium, the threshold is lower for him to make material in some other medium.

Manovich interprets the background of new media as the convergence of computer and media technologies. Manovich suggests that the synthesis of these two histories is a translation of all media into numerical data accessible through computers: "The result is new media — graphics, moving images, sounds, shapes, spaces, and texts that have become computable; that is, they comprise simply another set of computer data." (Ibid.: 20.)

According to Manovich (ibid.: 69), the uses for computers changed during the 1990s, leading to the emergence of new media:

> In the beginning of the decade, the computer was still largely thought of as a simulation of a typewriter, paintbrush or drafting ruler—in other words, as a tool used to produce cultural content that, once created, would be stored and distributed in the appropriate media—printed page, film, photographic print, electronic recording. By the end of the decade, as Internet use became commonplace, the computer's public image was no longer solely that of a tool but also a universal media machine, which could be used not only to author, but also to store, distribute, and access all media.

Probably everyone who has used computers since the early 1980s can agree with Manovich's claim about the change in how computers are used and for what purposes. The introduction of graphical user interfaces in Mac, Windows, and other operating systems made possible a "what you see is what you get" approach to word processing, publishing, graphics creation, and many other applications. Internet has become important as a source for material, a place for publishing material, and a channel for communication between people. I concur with Manovich's claim of the computer's "public image" as a universal media machine. Nevertheless, this argument should be studied further through studies of the social construction[15] of computers in various parts of the world.

---

[14] For a detailed discussion on digitization and numerical representation of sound, see Roads and Strawn (1996a).

[15] By the 'social construction' of a technology, I refer to studies on how "[…] a technological artifact is negotiated between the many 'relevant social groups' who participate in its development, and the relevant social groups who share a particular meaning of the technology." (Pinch & Trocco 2002 [2000]: 67.)

2.7 Remediation in the Multitrack Studio Medium

In this section, I propose an explanation based on the theory of remediation (Bolter & Grusin 1999) of how music-making software for PlayStation 2 can be compatible with multitrack studio equipment.

According to Bolter and Grusin (ibid.: 45), remediation is a defining characteristic of new media.[16] Their starting point is that a medium cannot operate in isolation, but every medium is in different types of relationships with other media (ibid.: 65). Bolter and Grusin (ibid.) argue:

> There may be or may have been cultures in which a single form of representation (perhaps painting or song) exists with little or no reference to other media. Such isolation does not seem possible for us today, when we cannot even recognize the representational power of a medium except with reference to other media.

Not all characteristics of an earlier medium may be taken as such into a new medium. Remediation involves refashioning materials and practices of an earlier medium in a new medium: “When artists and technicians create the apparatus for a new medium, they do so with reference to previous media, borrowing and adapting materials and techniques whenever possible.” (Ibid.: 68.) One example of remediation can be found in computer graphics software, where “[…] paint programs [have] borrowed techniques and names from manual painting or graphic design practices: paintbrush, airbrush, color palette, filters, and so on.” (Ibid.: 69.)

Remediation may be a conscious attempt to improve an earlier medium (ibid.: 59). Remediation enables the combination of characteristics from different media, potentially resulting in the development of media hybrids.

As I noted earlier, music utilities for personal computers continue the multitrack studio model of music-making. Since the 1990s, most components of a multitrack studio have been implemented as computer software, including multitrack recorders, sequencers, mixers, effects, synthesizers, samplers, and drum machines. Technologically, this trend can be explained with the increased computational power of personal computers. With proper hardware and software, computers can be used to record, manipulate, and synthesize sound. Alternatively, this trend could be explained with the theory of remediation as proposed by Bolter and Grusin. In order to do this, I have first to define what kind of medium multitrack studio is.

Théberge (2001: 11) suggests that artists are “composing” with the medium of multitrack studio. It should be noted that Théberge places the word composing in quotation marks. Obviously, this is a different type of composing when comparing what the term referred to for instance in the nineteenth century in Western high culture.

Kress and van Leeuwen (2001: 22) define ‘media’ as follows:

---

[16] Remediation is not unique to new media. Bolter and Grusin (1999: 11) argue that the process of remediation can be found, for instance, in centuries of Western visual representation.

> Media are the material resources used in the production of semiotic products and events, including both the tools and the materials used […] e.g. the musical instrument and air […]. Recording and distribution media have been developed specifically for the recording and/or distribution of semiotic products and events which have already been materially realised by production media […].

Following this definition, the multitrack studio can be interpreted as a production medium of master recordings of music that are disseminated through recording and distribution media such as audio CDs and MP3 files. Multitrack studio is also an artistic medium because music-making is an artistic activity.

Following the art critic Timothy Binkley, Gracyk suggests that an artist's medium is a network of conventions that delimits a realm over which materials and aesthetic qualities are mediated. Binkley argues that a medium includes the parameters within which to search for and experience the aesthetic qualities of that particular medium. For the audience, a medium is a framework within which certain things can be experienced. (Gracyk 1996: 69-70.)

Material objects (like pieces of studio equipment) alone do not constitute an artistic medium. Gracyk writes: "[a]n artistic medium is a mode of organizing perception and of unpacking meanings. […] A medium arises from a set of human *practices* with some range of materials." (Ibid.: 69-70.) Thus, "[t]o understand a medium is to know which qualities are relevant to something's counting as a specific work." (Ibid.: 70.)

A medium focuses the perception of the artists and audiences on certain qualities of the music made within the medium (ibid.: 69-70). The audible outcomes of the multitrack composing techniques are examples of such qualities. In the multitrack studio medium, the set of practices mentioned by Gracyk (ibid.: 70) include multitrack composing techniques. Music is made in the multitrack studio medium in part through these techniques. The materials of the multitrack studio medium are recorded sounds.

According to Gracyk (ibid.: 71), "[f]or most music, the medium is a range of allowable sounds together with principles for structuring those sounds." With the introduction of samplers, for example, the range of potential sounds in music has been extended. However, for reasons rooted in musical style conventions, it cannot be assumed that every sound could be incorporated into a piece of music, even if technology allowed it. Certain sounds may not be considered appropriate (or "allowable" in the sense suggested by Gracyk) in certain styles of music.

Making music within the multitrack studio medium involves certain ways of structuring sounds into pieces of music. These ways include, for example, overlapping and editing sounds. Modifying sounds after they have been performed and recorded is a part of structuring them into a piece of music.

Like many earlier media, new media oscillate between the logics of transparency ('immediacy') and opacity ('hypermediacy') (Bolter & Grusin 1999: 19). The logic of immediacy renders automatic the act of representation while the logic of hypermediacy multiplies the signs of mediation in an attempt "[…] to reproduce the rich sensorium of human experience." (Ibid.: 33-34.) In the multitrack studio medium, the documentary mode of recording is concerned with transparency, whether or not that can be achieved (see

Toynbee 2000: 86-89). In the multitrack studio medium, the logic of hypermediacy is about reproducing real-time musical performances as well as sound collages made in non-real-time, programmed sequencer patterns, and modified sounds. The multitrack studio medium is a hybrid of the apparently transparent reproduction of live performances and the opacity of sound processing and manipulation.[17]

To summarize, the multitrack studio medium introduces new concepts to music-making activities: Music (as recorded sound) does not have to be a transparent document of a real-time performance. 'Multitrack composing' is the activity of using the multitrack studio medium for making music. The materials of the multitrack studio medium are recorded sounds. The content of the sound recordings can be sound caused or organized (composed and performed) in many different ways by humans (e.g., through models of music such as the pitch lattice, functional harmony, and Indian *raga*) or by nature (e.g., the sound of thunder and whale song). Regardless of the content of a sound recording, it can be modified with the tools of the multitrack studio medium in several ways including editing, pitch shifting, changing loudness, altering spectral characteristics, reversing, and applying spatial effects. Working within the multitrack studio medium also involves overlapping and sequencing these materials.

2.8 Refashioning the Multitrack Studio Medium

In this section, I propose an interpretation of the relationship between music-making software, music utilities for personal computers, and all other multitrack studio equipment.

According to Bolter and Grusin, remediation involves the refashioning of materials and practices: "When artists and technicians create the apparatus for a new medium, they do so with reference to previous media, borrowing and adapting materials and techniques whenever possible." (Bolter & Grusin 1999: 68.) During remediation, the characteristics of an earlier medium may not be adapted as such into the new medium. Remediation can be viewed as reforming or improving upon an earlier medium by, for instance, filling a lack or repairing a fault in the preceding medium. (Ibid.: 59-60.)

During remediation some characteristics of the older medium may be adopted as such, refashioned, or completely left out. In multitrack studio equipment the characteristics that can be remediated include functions, layout of user interface, and the characteristic sound.

Many music utilities for PCs utilize the visual metaphors of tape recorder and mixing desk controls, which are used with alphanumeric keyboard and mouse. Durant's (1990: 189) observations of MIDI sequencer software are of importance when considering remediation in the multitrack studio medium:

[17] A medium is socially constructed and this affects the experiences of that medium and its contents. What seems as immediate to one group of people might appear hypermediated to another group. (Bolter & Grusin 1999: 70.)

> Instructions in composing software are often modelled on tape-machine commands (rewind, record, playback, fast forward, etc), on the assumption that multi-track recorders form a known reference-point for the generation of 'analog' musicians who are currently re-educating themselves for digital technologies. A relatively known operating format is used as a model for conceptualizing compositional and editing processes which actually work on a very different technical basis.

Direct appropriation in remediation (adopting with a very few changes the layout of the user interface of a device) can be explained by reasons of familiarity along the lines argued by Durant. Music-makers can realize the same techniques in more or less the same ways regardless of whether they use studio hardware or computer software. Another important observation made by Durant is that the technical base (the actual implementation) is separate from the visual appearance of the interface. In many cases, the user interface has been remediated in a familiar way from studio hardware to computer software, although the technology beneath the software's user interface is very different.

The remediation of a medium can change the working methods. For example, when editing an analog sound recording on magnetic tape the choice where to splice the tape was done by listening, and by moving the reels manually. Early samplers such as the Emulator did not have the facilities to represent sample files visually. Truncating a sample file on the Emulator was done by listening and manually adjusting the truncation points with two sliders. Computer displays changed decisions made during sound editing so that they were not done exclusively through listening but also through looking at visual representations of sounds and their modifications. These visual representations could be very detailed, so sound could be manipulated with great accuracy.[18] The increased control over detail is one important aspect of refashioning in the multitrack studio medium remediation process.

The distinctive sound of a piece of multitrack studio equipment may or may not be adapted in the remediated form. In some cases (like valve amplifier software filter plug-ins[19]), only the sound or signal-altering characteristics (and not the original functionality, which is amplification and transduction of audio signals) has been included in the remediated form.

The remediation of multitrack studio equipment from dedicated devices to computer software has caused transformations in music-making when conventions from other computer media have been introduced into music-making activities. Music-making software, music utilities for PCs, digital audio workstations, and hard disk drive -based "portastudios" are nonlinear. I use the term 'nonlinear' to refer to a sound storage medium

---

[18] The concepts of 'functions' and 'techniques' provide one explanation for understanding the rapid and extensive acceptance of computer-based multitrack composing applications since the mid-1980s. The techniques artists, producers, and sound engineers have learned with hardware studio equipment could be realized with the functions of computer software that allowed working in non-real-time. Thus, the techniques could be realized with software more precisely than before.

[19] Guitar amplifier and effects modeling software plug-ins include, for example, *Amplitube* by IK Multimedia and *Guitar Rig* by Native Instruments.

in which any data point can be accessed instantly (Bartlett & Bartlett 1998: 421). Such storage media are, for instance, semiconductor memory and hard disk drives. Access to this type of storage media is called random access. In contrast, serial access recording media (like magnetic tape) require locating the specific point in the recording before it can be accessed because the sound recording is linear.[20]

By the term 'nonlinear', I also refer to the dimension of time. Random access makes it possible to manipulate a sample file or to combine sample files in non-real-time compared to the time of the musical material. For instance, the calculations required to amplify a five-minute long digital sound recording may take only seconds for the computer software. The music-maker may define with one mouse click a fade that gradually changes the volume of a track over the course of several minutes. There is no temporal proportion between the input (for instance defining a modification) and its affect on the audible output. For example, with one selection it is possible to set a sample file to be replayed repeatedly in a loop.

Another notable change that has occurred with the shift from linear to nonlinear multitrack recording is that discrete tracks physically located on a multitrack tape have been replaced by digital sound recordings (sample files). The temporal relationship between materials stored linearly on the tracks of a multitrack tape is more or less locked, even if this could be altered to some degree by delaying the signal on some tracks. With nonlinear multitrack recording, the temporal relationship between each discrete sound recording can easily be changed. It is easy to move one sample file earlier or later in relation to the other sample files.

Nonlinear multitrack composing techniques are examples of remediation as the reforming or improving of an earlier medium. Some nonlinear multitrack composing techniques are remediations of earlier multitrack composing techniques with the key difference that music-makers can realize them in non-real-time. Nonlinear multitrack composing techniques have extended the options of music-making. Next, I discuss in detail the characteristics of nonlinear multitrack composing.

A key feature of software for multitrack recording is the graphical user interface. In what I call the 'multitrack window', the dimension of time is typically represented on the horizontal axis from left to right. The vertical axis contains the visual representations of the 'virtual tracks', each of which contains one sample file or several consecutive sample files. Sample files are represented as icons one on top of another on the screen, and a common convention is to display in the icon an approximation of the waveform of the digital sound recording.

Software for nonlinear multitrack composing have in common several generic operations with other new media tools. The generic operations supported by all software with graphical user interfaces are available for making music. These operations include cut, copy, paste, drag-and-drop, and selecting items from menus (Manovich 2001: 120-121).

[20] One definition used in audio technology terminology to the term 'nonlinear' is to describe an audio device that distorts the signal passing through it. This is not covered by my definition of the term 'nonlinear'.

It seems that the shift from tape-based linear multitrack composing to nonlinear multitrack composing has led to the change from longer complete performances recorded as separate layers (tracks) to shorter fragments (sample files). Depending on the style of the music being made, takes may be much shorter in nonlinear multitrack composing than when using linear tape-based multitrack equipment. In nonlinear multitrack composing it is common to loop short recordings (occasionally as short as a few seconds) and use the same sample files in different parts of a piece of music, and sometimes in other pieces of music as well. [21]

In nonlinear multitrack composing, the term ‘track’ has a different meaning than in linear tape-based multitrack recording, where a limited number of tracks may physically exist on a magnetic tape. In nonlinear multitrack composing, the term ‘track’ is a metaphor originating from tape-based multitrack recording. A track may contain several consecutive sample files, the order of which can be rearranged.[22]

In nonlinear multitrack composing, various sound processing effects can be applied to the sample files in real-time or non-real-time. For example, it is possible to change the pitch of one sample file without affecting the pitch of the other sample files. This has been possible to some degree in tape-based multitrack recording by routing the material on one track to a pitch shifting effects processor. However, with nonlinear multitrack composing, pitch shifting has become easy to implement.

One characteristic of nonlinear multitrack composing is to use material from sample libraries. This material may be used as temporary material (for example, drum track) to guide the composing process. This material may also end up in an unmodified form on the final sound recording. Sample libraries are available on CDs and DVDs, as well as on the Internet.

A sample file is the basic unit in nonlinear multitrack composing. A sample file is a set of numerical data that represents a sound in the digital domain. Eventually, the digital representation is converted through transduction via loudspeakers into an audible form (Truax 2001: 9-10). In nonlinear multitrack composing, sounds are modified and combined through their digital representations (sample files) into pieces of music. The combinations and modifications are defined on the basis of what the sounds originally sound like and what they sound like after the combination or modification. These decisions are made on the basis of *listening* rather than, for instance, spectral analysis of the sample files.

[21] I am not arguing that the introduction of nonlinear multitrack composing caused a major break from the past by replacing long linear performances with short fragments. Because of punch-in recording in linear tape-based recording, the material on a single track has not necessarily been from one take. In addition, during the process of editing, a piece of music could have been compiled from various downmixed parts. Hence, nonlinear multitrack composing cannot be considered a radical change from previous multitrack composing techniques. Nevertheless, it offers more control over the temporal relationships of separately recorded material.

[22] In the context of deejaying, the term ‘track’ can also mean a piece of music. According to Cox and Warner (2004: 416), ”[i]n DJ culture, the term ‘track’ is preferred to the term ‘song’ because tracks are often not seen as completed entities but as sets of elements to be combined with other such elements by the DJ in the creation of a mix.”

Because sample files can be replayed at variable pitches (often in steps of semitones or tones) or for different durations than the original sample files, aspects of pitch lattice composing are part of nonlinear multitrack composing.

An aspect of new media in nonlinear multitrack composing is that sample files are distinct from the definitions of how they are assembled into pieces of music. The same sample file can be arranged into several different pieces of music. This is possible with component audio file formats, which keep each sample file discrete, thereby allowing maximum control over them. In contrast, consumer audio formats such as CD audio and MP3 files are compound. The earliest component audio file formats were probably the various file formats used by tracker software. Current component formats include the proprietary file formats used by music utilities like *GarageBand* and *ACID* [23].Music-making software titles store music in component file formats.

Mixing and effects settings, and all parameters of a performance with a MIDI device, for example, are defined with control data. Control data may be a single value or it can be a chain of values that change over time. Control data can be generated by algorithms or originate from a performer’s gestures that have been converted into digital data with some controller device.[24] Control data can be recorded and stored separately from the parameters it affects. It is possible to manipulate control data with the alphanumeric keyboard or mouse after it has been recorded. Control data can be edited, shortened or extended in time, compressed, or expanded in range. Control data is stored alongside sample files in component audio file formats.

Nonlinear multitrack composing is not special because it is done in non-real-time because composing music through written notation also takes place in non-real-time. The significant difference from the earlier forms of composing is that in nonlinear multitrack composing, the audible outcome is defined through choosing, placing, and modifying sample files. Compositions in the form of written notation had to be performed by musicians to get them into audible form. In nonlinear multitrack composing, the relationship between tactile actions and the musical outcome is problematic. In performance with acoustic instruments, the audible outcome is a direct result of musicians’ physical actions. The ethnomusicologist Marko Aho (2005: 227) argues that in music made with digital tools there is less musicianship in the old sense of the term, as not as much of the tactile resources of the body are used and the making of music is more based on cognition. Physical movement is the common factor in singing and playing musical instruments. Aho argues that when people perform music, they want to produce sound physically. “Playing” air guitar and tapping rhythms with hands and feet are examples of making music and reacting to music kinetically. (Ibid.: 222-223, 227.) Following Aho, it

---

[23] ‘Breakout tracks’ are sets of sample files lifted from multitrack master recordings of songs that are made available to remixing contest participants (Hawkins 2004: 6). Such contests (with music varying from artists like Madonna to Karlheinz Stockhausen) have been organized regularly in the web-site ACIDplanet (http://www.acidplanet.com), which promotes multitrack software package *ACID* and its sample libraries.

[24] For discussions on musical control devices and gestural control of music, see Roads et al. (1996); Wanderley and Battier (2000).

could be argued that nonlinear multitrack composing is based more on music-makers' cognition than their physical actions.

The ethnomusicologist René T. A. Lysloff argues that music made with computers compels reconsideration of the Carthesian mind-body divide in music. Musicians are evaluated according to their skills in real-time performance. Performers have a relationship through their bodies with the sounds they produce. Performers are considered as interpreters, not as creators. Western art music composers are considered as musical thinkers, and compositions as the results of their creative minds. In music produced with computer technology, the division between mind and body is blurred. With computer software, the composer creates and interprets music by determining all aspects of a piece of music. (Lysloff 2003: 45-46.)

I exclude the physical aspects of music-making from the scope of this study because they are mostly not part of nonlinear multitrack composing. Of course, nonlinear multitrack composing involves certain physical actions on the part of the music-makers, ranging potentially from making selections with a controller device to tapping one's feet to the beat of the music. In addition, during nonlinear multitrack composing the music-maker may associate gestures with the material he defines in non-real-time. Thus, I do acknowledge that there are complex relationships between making music with the nonlinear multitrack composing techniques and experiencing music physically and mentally.

One further aspect of remediation is that the remediated form of a medium may reiterate social problems and inequalities associated with the earlier medium. People who have been discouraged from using the earlier version of a medium may also be discouraged to use the remediated form. Male musicians have dominated the use of recent music technology, and women have rarely been encouraged to learn and use music technology on the studio or the stage (Bayton 1990 [1988]).[25] Théberge (2001: 13) writes:

> Recent studies conducted by the music instrument industry have suggested that even among women with computer and music instrument skills, the use of music software is extremely limited. […] In reproducing the multitrack studio in software form, programmers implicitly assume that users *already* have the knowledge required for its use, thus reproducing, perhaps, the social inequalities associated with access to the earlier technology as well.

In Section 6.3.1, I will return to this issue of the remediation process reiterating the inequalities of earlier media.

[25] For a discussion on the scarcity of women in deejay battles, see Katz (2004: 131-136). See also Rose (1994: 57).

2.9 Summary

Following Gee (2003), I have defined multitrack composing as a semiotic domain. There are several semiotic domains of music, and theoretically, the semiotic domain of multitrack composing may overlap with several of them. The three music-making software titles *Magix Music Maker*, *MTV Music Maker 2*, and *eJay Clubworld* belong to a sub-domain of that domain. I call this sub-domain simply 'music-making software'. There is no reason to call it anything other. Alternatively, it could be considered that each music-making software title forms a sub-domain of the semiotic domain of multitrack composing. In Chapter 3, I examine the similarities and differences between the three music-making software titles.

Multitrack composing and nonlinear multitrack composing are special music-making activities and their outcomes contribute to defining the semiotic domain of multitrack composing. Multitrack studio equipment has been developed (among other purposes) to enable this activity. The signifiers of multitrack studio equipment (including, e.g., user interface layouts and graphics, and the audible outcomes of realizing certain techniques) and the knowledge how to use the equipment are a part of the semiotic domain of multitrack composing.

Several other elements also contribute to defining this semiotic domain, including:

- all sound recordings realized with multitrack studio equipment;
- the concepts and values concerning this music;
- multitrack studio equipment (hardware and software);
- the knowledge people possess about multitrack studio equipment and the ways of using it;
- written texts;
- audiovisual material;
- live music performances.

Pieces of multitrack studio equipment, such as tape recorders, mixing desks, effect devices, samplers, and computers equipped with music software are a part of the semiotic domain of multitrack composing in the sense that these pieces of equipment have certain meanings and expectations attached to them. People have ideas and preconceptions of what can and cannot be done with a certain piece of equipment, such as an echo effect device or a sampler. These ideas can be extended and preconceptions may be disproved while making music with this equipment. Knowledge about how to achieve specific audible outcomes with certain pieces of equipment is also a part of this semiotic domain.

The knowledge that musicians, sound engineers, record producers, scholars, and music enthusiasts possess about multitrack studio equipment is a part of the semiotic domain of multitrack composing. Everything from the overall concepts to the most detailed tricks is a part of this knowledge. Some of this knowledge is documented in written texts. However, not necessarily everything has been documented. A lot of this knowledge is transferred directly between people through social interaction and indirectly through music. Some things can be learned through trial and error.

The musicologist Will Straw (2001: 58-59) argues that Internet has increased the availability of music and related material:

> […] the Internet has led to the wide-spread circulation of old album covers, public domain recordings from decades earlier, and fan sites devoted to the most obscure and faded of musical moments. The musical past now seems more minutely differentiated, richly documented and abundantly accessible than at any other historical moment.

In addition to the public domain recordings Straw mentions, a great number of commercial recordings are also available on the Internet. The Internet is also a channel for communicating with other people in order to gain access to music that is not available in digital format. The Internet is a portal for "interfacing" with the semiotic domain of multitrack composing, for instance by uploading and downloading music and sample files, and accessing written texts concerning the domain.

Sound recordings – potentially all the music made with multitrack studio equipment – form a large part of the semiotic domain of multitrack composing. My justification for this argument is that it is possible to hear from sound recordings the audible outcomes of the multitrack composing techniques that have been realized to make the music. It is possible to hear the audible outcomes of multitrack composing techniques from sound recordings, and on the basis of this, to attempt to realize them oneself with the available hardware and software.[26]

Live performances also contribute to defining multitrack composing. In addition to studio use, pieces of multitrack studio equipment are used in live performances. Some elements of a live performance may originate from a multitrack recorder or a MIDI sequencer and that material may be modified in real-time during the performance. Certain multitrack composing techniques may be adapted as live performance techniques. To name one example, deejays blur the distinction between mechanistic reproduction of sound recordings, studio work, and live performance (Théberge 2001: 15). Live performances reach most people through recordings.

Multitrack composing techniques are a part of the semiotic domain of multitrack composing. Nonlinear multitrack composing techniques are remediations of earlier multitrack composing techniques. The defining characteristic of nonlinear multitrack composing techniques is that they are realized in non-real-time, whereas earlier multitrack composing techniques are such that they are realized in real-time.

The semiotic domain of multitrack composing also includes written texts, and audiovisual materials. By 'written text', I refer to books and articles on the topic written, for instance, from a theoretical or practical point of view. In addition, music reviews contribute to defining multitrack composing. Audiovisual materials showing artists in performances or

---

[26] It seems that the further back in history the original sound recordings have been made, the more perceptible the audible outcomes of the multitrack composing techniques are. For example, the use of echo and reverberation in sound recordings from the 1960s is really obvious in many cases, as are some early uses of sampling like the "stuttering effect" (one example of this is in Paul Hardcastle's song *19*) and replaying a sample file rapidly up or down a scale.

working in studios, and video tutorials also take part in defining what multitrack composing is and what multitrack studio technology can be used for.

The relationships between the semiotic domain of multitrack composing and other semiotic domains of music are complex and I will consider them only briefly in the following. Using multitrack studio technology in music-making is the defining characteristic of the semiotic domain of multitrack composing. Through sound recording and sampling, material from other musical domains can be incorporated into the semiotic domain of multitrack composing. Sample files converted through transduction processes (Truax 2001: 121) into an audible form can be signifiers through which people form meanings. As scholars including Taylor (2001), the ethnomusicologist Erkki Pekkilä (1996: 230-231), and the soundscape scholars R. Murray Schafer (1994: 90-91) and Barry Truax (2001: 134-136) have pointed out, through recording or sampling a musical phrase loses the link to the context in which it was originally produced. According to Taylor (2001: 135), this means that sonic signifiers are separated from their signifieds and from their makers. The link to the original context is necessary for understanding the original intentions and meanings of the recorded or sampled sound. If there is no such link, the recorded or sampled sound may become attached to a variety of signifieds. It seems to be a tendency in the semiotic domain of multitrack composing that original intentions and meanings are discarded.

3. Analysis Part 1: Music-Making Software and the Semiotic Domain of Multitrack Composing

3.1 Starting Points

The first part of the analysis includes comparing the three music-making software titles between each other and other multitrack studio equipment. The reason for so doing is to identify the common functionality of the three music-making software titles, and explain how these are related to the semiotic domain of multitrack composing.

In Chapter 2, I defined three aspects common to all multitrack studio equipment, including music-making software:

- Functions: The potential that a piece of equipment or software has for making music.
- Features: The names given to the functions by the equipment manufacturers or software developers. The same function may be labeled differently by different manufacturers or developers.
- Techniques: The knowledge and skills that are separate from specific pieces of equipment. Techniques can be realized with various types of multitrack studio equipment and software.

The functionality is described by developers as the ‘features’ of a specific piece of equipment or software. In other words, ‘functions’ appear to music-makers as ‘features’. The terminology used to describe features may be established and widely accepted, or it may have been coined by the developer. To mention one example of a potentially confusing term, in *MTV Music Generator 2*, the term “riff” does not describe what this feature is: A user-programmable pattern (a sequence of sample file triggering definitions). In *MTV Music Generator 2*, even single sample files may contain riffs in the sense this term is commonly understood. The same function may be supported in all three music-making software titles, but it may be called a different feature in each software.

The features of different pieces of equipment or software cannot be directly compared. Defining a metaterminology of functions and translating features into functions make it possible to compare the features of the three music-making software titles. In this study, ‘functions’ are metaterms that are defined so as to describe the common ground of the features of the three music-making software titles (and potentially all other multitrack studio equipment).

Numerous multitrack composing techniques have been developed over the years, including overdubbing, punch-in recording, sequencer programming, speed shifting, editing, and others (see Nazarian 1988; Huber & Runstein 1997; Bartlett & Bartlett 1998). Even though these writers call these ‘recording techniques’ or ‘recording production techniques’, I will include them in ‘multitrack composing techniques’.

To realize a multitrack composing technique involves using the features of a piece of studio equipment or software in a specific way to achieve the desired audible outcome. Techniques are knowledge possessed, for instance, by musicians, record producers, and sound engineers. Music-makers know how to realize multitrack composing techniques with certain pieces of equipment or software, and they can anticipate what the result will eventually sound like.[27]

Multitrack composing techniques are not bound to specific pieces of equipment. Music-makers realize techniques by using certain features in certain ways: More or less the same technique, and more or less the same audible outcome, can be realized and achieved with different hardware and software. Some techniques correspond directly to the features of studio equipment or software. In other cases, several features must be used in order to realize a multitrack composing technique.

Following the theory of remediation as defined by Bolter and Grusin (1999), my starting point is that the developers of music-making software have adapted certain functions and user interface conventions of multitrack studio equipment. This way, music-making software titles are to some degree “compatible” with all multitrack studio technology.

As mentioned in Section 1.2, the functionality of music-making software is in certain ways different from the functionality of many music utilities for personal computers. For example, with music-making software, no long sound recordings can be made, there is no support for MIDI, and there are no effects plug-ins or versatile mixing automation. Nevertheless, several multitrack composing techniques can be realized with the features of music-making software.

Music-making software titles are a part of the semiotic domain of multitrack composing because of the techniques that can be realized with them. The features available for realizing techniques are mostly of the nonlinear (non-real-time) type in music-making software. Thus, they are called ‘nonlinear multitrack composing techniques’.

In order to make comparisons between the three music-making software titles, a metaterminology is needed to “translate” features into functions (a common terminology), and at a later stage in this study, to show what features and functions can be used to realize what multitrack composing techniques.

3.1.1 Method for Part 1 of the Analysis

The first part of the analysis has three stages. Different research techniques and procedures are used in each stage. These, and the key questions for each stage are shown in Table 3.1.

[27] Experimental music is an exception to this, because the making of such music is based on the idea that the outcome cannot be anticipated.

Stage	Focal questions	Research techniques and procedures	Outcomes
1. Systematic description	How does music-making progress with each music-making software title? What features does each music-making software title have?	1) Define a five-phase model of the nonlinear multitrack composing process 2) Describe the GUI of each music-making software title 3) Describe the features and music-making actions associated with each phase of the nonlinear multitrack composing process	A description of the music-making actions and procedures for using each music-making software title An account of features for music-making An outline of the required metaterminology of functions
2. Comparing features to functions	How do the features of the three music-making software titles relate to each other?	1) Define a metaterminology of functions 2) Compare features through the metaterminology and present the comparison in a table	An account of functions supported by the three music-making software titles
3. Defining nonlinear multitrack composing techniques, and comparing features and functions to techniques	How do the features of the three music-making software titles relate to the semiotic domain of multitrack composing? What nonlinear multitrack composing techniques can be realized with the features? What can be done with these techniques?	1) Define nonlinear multitrack composing techniques, their background, and essential uses applications in music-making 2) Define in a table what features can be used to realize what nonlinear multitrack composing techniques	Definitions and descriptions of nonlinear multitrack composing techniques An account of nonlinear multitrack composing techniques that can be realized with music-making software

Table 3.1. Summary of main questions, research techniques and procedures, and outcomes of Part 1 of the analysis.

Stage 1 is reported in Sections 3.2 to 3.5.6, stage 2 in Sections 3.6 to 3.7, and stage 3 in Section 3.8. Conclusions to the first part of the analysis are presented in Section 3.9. The research techniques and procedures used in each stage are explained in the respective opening sections.

3.2 Analysis Part 1 Stage 1

3.2.1 Defining a Model of the Nonlinear Multitrack Composing Process

In this study, I will use the hypothetical five-phase model of the nonlinear multitrack composing process illustrated in Figure 3.1. I have constructed this model on the basis of personal experiences of making music with personal computers, multitrack studio equipment, and music-making software. The purpose of this model is practical: I use the phases to structure the analysis of the three music-making software titles. I do not claim that this is a generic model suitable for analyzing all music-making processes in multitrack studios. I will not address the issue of how this model applies in practice to music-making activities. It certainly seems that a similar model of music-making does exist in practice.

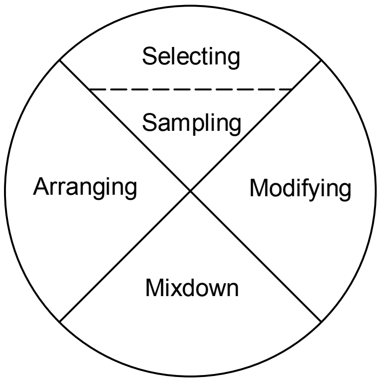

Figure 3.1. Five phases of the nonlinear multitrack composing process.

The phases in the model are discrete in the sense that it is possible to be involved in only one of them at a time. However, the phases are not as independent or isolated as the figure perhaps would suggest. There is a constant oscillation between these phases during music-making.

The selecting phase involves choosing suitable sample files from a sound library.[28] The sample files in the library may be prefabricated material or earlier recordings made by the music-maker. The selecting phase requires listening to sounds and considering how these could be used in a piece of music. The sampling phase involves the music-maker selecting the material to sample and recording the samples. Thus, these phases of sampling and selecting are closely related.

In the modifying phase, sample files are altered in various ways (see Section 3.6.3). In the nonlinear multitrack composing process, a sample file is modified by applying modification effects to it through the visual representation of the sample file or a part of it. Some aspects of the modifying phase are similar to the technique of 'modulation' in

[28] Théberge (1997: 200) argues that music-making with synthesizers and samplers has since the 1980s become allied to a form of consumer practice, the exercise of taste and choice, "[…] where the process of selecting the "right" pre-fabricated sounds and effects for a given musical context has become as important as "making" music in the first place."

electronic music. Moog (1984 [1976]: 31) defines ‘modulation’ in electronic music as the use of one waveform to shape another waveform (certain waveforms are eventually converted into an audible form). When discussing modulation as a potential carrier of musical expression, Moog (ibid.) emphasizes that sound in itself is not music:

> Sound is merely a *carrier* of information. Our ears and brains are very good at detecting changes in sound. When these changes carry a message with artistic content and are in forms that do not abuse our built-in-sound-receiving and processing equipment, then we have music. […] synthesizers have expanded the concept of musician control from direct physical production of sound (as on acoustic instruments) to control over multiple layers of sound processing. The modulations then become *carriers* of information, along with the tone itself; the musician exercises control by determining how the modulation changes.

Even though Moog presented this argument during the era of analog sound synthesis and processing equipment when modulation was typically done manually in real-time as part of musical performance, this idea can nevertheless be applied to nonlinear multitrack composing. Modifying sample files in real-time or non-real-time contributes to the information a piece of music may convey as a sound recording. The modifying phase is not obligatory in the nonlinear multitrack composition process because a music-maker can use sample files in a piece of music without modifications.

In the arranging phase, sample files are placed so that they are alongside each other or overlap, thus forming a piece of music. By ‘arranging’ I refer to the placing of sample files *through their visual representations* (icons) in the multitrack window, and making a piece of music through these representations. However, the decisions made while arranging sample files into a piece of music are based on *listening* to their characteristics. Arranging also includes aspects of the traditional definition of arranging a piece of music.[29]

In the mixdown phase, sample files are mixed into an audio signal that is in mono, stereo, or a surround format (Dolby Digital is supported in *MTV Music Generator 2*). Following Cary (1992: 301), I define ‘mixdown’ as the process (rather than the outcome) of reducing several tracks to one or two by combining them. In nonlinear multitrack composing, the tracks contain sample files. In music-making software, mixdown settings are defined through visual representations of the volume levels and pan positions of sample files or tracks containing several sample files. Processing the sound through effects like compression and equalization is also part of mixdown.

This five-phase model should not be viewed as a rigid mechanistic music-making process. The order of working on each sample file or track does not necessarily progress from, for example, selecting or sampling to modifying to arranging to mixdown. During the nonlinear multitrack composing process, the music-maker can constantly shift from one phase to another. As Lysloff (2003: 47) suggests, with music software “[a] composition (or any part of it) can be immediately “played back” by the computer (using a tracker software

[29] Technically, it is conceivable that a piece of music could be arranged entirely through the visual representation of the sample files, without listening to them. In practice, this would require extremely accurate visual descriptions of the contents of the sample files.

program) so that the composer can listen to any passage throughout the compositional process." Thus, changes made to one track or sample file may after listening to the altered passage cause the music-maker to go back to make changes to some earlier tracks or sample files. When, for instance, considering what sample files to select, the music-maker is at the same time also considering the arrangement. Mixdown is not necessarily a concluding phase, because in nonlinear multitrack composing mixdown takes place all the time when a piece of music is listened to as it is being made.

When a music-maker considers a piece of music complete, he can record it on a personal computer with which he may edit and process it further, and finally store it to an audio CD. All three music-making software titles allow the piece of music to be stored in the memory card attached to PlayStation 2. What is stored is the data defining the position of each sample file in a piece of music, the modifications applied to them, the mixing settings, and in the case of *MTV Music Generator 2*, also the sample files recorded by the music-maker.

3.2.2 Stage of Systematic Description

In Sections 3.3 to 3.5.6, I present a systematic description of music-making software. The method at this stage is to describe the characteristics of each music-making software title, list their features, and so to outline the need for a metaterminology of functions.

I will describe the graphical user interface (GUI) of each music-making software title. This includes descriptions of the virtual space of the software as well as the visual representations of sample files and their modifications. Such details of GUIs are of importance, because, according to Winkler (1998: 109), they influence the way people think and act while making music with computers.

Sample files, and the modifications and effects that can be applied to them are represented visually in the GUI. I use the term 'represent' in the sense that the researcher and composer Jean-Claude Risset (1991: 8) proposes: "*Representation* of something makes it present, visible — to the eye or to the mind. It offers a substitute similar in some ways to the real thing." In music-making software, sample files are accessed and manipulated through their visual representations (such as icons or waveform graphics). I describe these representations in some detail because their comprehensibility is a fundamental enabling factor in the use of music-making software.

'Virtual space' is the space formed by the various windows and menus in a music-making software title. In music-making software, these spaces represent musical data in certain ways and the music-maker can do certain things in particular parts of the virtual spaces. Video games also have virtual spaces. Wolf (2001c) defines eleven types of virtual space in video games. I use Wolf's definitions to describe and compare the virtual spaces of music-making software. In Chapter 5, I compare the virtual spaces of music-making software to the virtual spaces of video games.

A crucial part of the virtual space of a music-making software title is the 'multitrack window', where visual representations of sample files or patterns are displayed on overlapping tracks. The dimension of time is represented horizontally or vertically in the multitrack window.

As a part of the systematic description, I illustrate the music-maker's actions during each of the five phases of the nonlinear multitrack composing process. Describing these actions involves explaining, for instance, how the features of the music-making software titles are used to make music. Fullerton et al. (2004: 60-64) define procedures in video games as the methods of play and the actions players take when trying to achieve the game's objectives: "Who does what, where, when, and how?" I have adapted this definition of procedures to describe the actions of using music-making software. I will describe the procedures associated with each phase of the nonlinear multitrack composing process. The reason for describing the procedures of music-making software is that in Chapter 5 I will compare them to the procedures of video games.

3.3 Magix Music Maker

3.3.1 Generic Description

Magix has developed a number of music software for personal computers with Microsoft Windows operating systems. *Magix Music Maker* for PlayStation 2 was released onto the European market in 2001.[30]

There are at least two product package versions of the European release. One of these describes the product in the following ways: "MAGIX music maker is pure creativity", "Play with music", "TOP! Tested by chart-topping producers!", "More than a game!", "100% fun", and "Live - at home or at the party - Guaranteed!". As these examples illustrate, the product package presents *Magix Music Maker* as a tool for making music with a playful approach. In addition, there is an emphasis on the aspect of having fun while using it to making music.

There are in the software two short tutorial videos called "First Steps" and "Tips & Tricks" in English and German. There is also a help screen with explanations of the controls.

According to the product package, about 3,000 sample files are included in *Magix Music Maker*. These are referred to as "sounds" and "loops" in the user's guide. Sample files are also referred to as "objects" in the GUI. Confusingly, the term "loop" is also defined in the user's guide as the playback region that can be defined by the music-maker.

A "sound" or a "loop" is represented in the multitrack window as an icon showing a waveform graphic. The waveform graphic bears no resemblance to the actual waveform of the sample file and this is misleading. It is not possible to align the strong parts of a sample file (peaks in waveform) through their visual representations as can be done in

[30] The North American version of *Magix Music Maker* was released in 2003 and it is apparently different from the European version. For technical reasons - North American versions of PlayStation 2 titles cannot be used with regular European consoles - I will not examine the North American version in this study.

many music utilities for PCs. It must be determined through listening when sample files are aligned properly. The waveform graphics in the icons representing the sample files suggest that the sample files are single-channel. This is not the case because all sample files are apparently in stereo. Some sample files have some type of “stereo effect” such as a noticeable difference between left and right channel.

Sample files are grouped into six styles, namely “Hiphop”, “Pop/Rock”, “80’s”, “House”, “2step”, and “Techno/Trance”. Each style contains seven of the following eight categories of sample files (called “instruments”): “bass”, “drums”, “keys”, “vocals”, “fx”, “pad”, “guitar”, and “synth”. Sample files from all these styles can be used in a single piece of music. There is a demo song in each of the six styles. The demo songs can be used as a starting point for a new piece of music or the music-maker may choose to start from scratch.

Sample files are referred to as “objects” in the “arranger”, that is, the multitrack window. Modifications are applied to the sample files through the icons representing them in the “arranger” window. The pointer is placed on top of the icon and a menu for modifying sample files is opened with the circle button. The following features become available:

- “reset object effects”;
- “copy object”;
- “delete object”;
- “echo slow”, “echo medium”, “echo fast”;
- “distortion: soft”, “distortion: medium”, “distortion: hard”;
- “filter low”, “filter medium”, “filter high”;
- “reverb short”, “reverb medium”, “reverb long”;
- “volume x2”, “volume x4”;
- “reverse”;
- “pitchshift” -12 to +12 semitones, in steps of a semitone;
- “resample” -12 to +12 semitones, in steps of a semitone;
- “timestretch” 50% to 200% in steps of 1%;
- “mute track”.

I will explain these features in the following sections.

3.3.2 GUI

The virtual space of *Magix Music Maker* consists of the main window (or the “screen”, as it is called in the user’s guide) which is divided into “mixer”, “toolbar”, “arranger”, and “explorer”. The latter three have windows or menus beneath them.

L2 button activates the “toolbar”, which is located at the top of the screen. The “toolbar” has the following options: “new arrangement”, “load arrangement”, “save arrangement”, “mixer”, “options”, “play/stop”, “help”, and “about”. The CPU consumption, memory usage, and an animated sound pressure level meter are also displayed on the “toolbar”. When the “mixer” is selected from the “toolbar”, the mixer window appears on top of the screen. The features on the main screen cannot be accessed when the mixer is active.

R1 button activates the “arranger”. The maximum number of tracks is sixteen. The tracks can be of the following types: “bass”, “drums”, “keys”, “vocals”, “fx”, “pad”, “guitar”, and “synth”. Depending on what “style” is chosen, all eight or only seven of these types are available. Despite defined track types, sample files of different types and from different styles can be placed on a single track. The R2 button activates the “explorer”, which provides access, for instance, to the sound library.

3.3.3 Selecting Phase

Sample files are browsed and selected from the “instrument directory”. When *Magix Music Maker* is launched, the pointer is located in the “instrument directory” and starts automatically to play in loop the first sample file in the list in the “drum” instrument category. This is a good way to invite the music-maker to do something. In the “instrument directory” the “sounds” and “loops” have been organized first into six styles and then into instrument types. It is in no way indicated if the music-maker has selected a one-shot “sound” or a “loop”. This can only be heard when browsing through the “sounds” and “loops”.

The hierarchy in “instrument directory” is as follows: The main category is “style”, beneath which is “instrument”, beneath which are located “sound” and “loop”. One “instrument category” is changed to another by pressing the right thumb-stick on the DualShock 2 controller to left or right. “Style” is changed by first pressing the right thumb-stick “in” and then pressing it left or right. The name of the style that is currently selected is displayed in the left bottom corner of the screen. In this window it is also possible to load a demo song by pressing the circular button.

The number in the sample file name seems to suggest what sample files sound good together. For instance, in the style “Techno/Trance”, sample files “elec1” (a bass pattern) and “strad1” (a string chord) have notes from the same chord. It appears also that color-coding in the icons (green, yellow, gray, and blue) is used to suggest what sample files sound well together.

Selecting a sample file into an empty slot of a track can be accomplished by placing the pointer in that slot and starting the playback of the song. When the playback progresses to the slot where the pointer is, the sample file that is currently selected in the “instrument directory” is played as if it was already a part of the song.

3.3.4 Modifying Phase

After a sample file has been selected, it is pasted to a track on the “arranger”. After that, the sample file can be modified. Table 3.2 shows the features that are available one at a time for modifying a sample file.

Features	Description
“echo slow”, “echo medium”, “echo fast”	An echo effect applied with long delay (“echo slow”) to short delay (“echo fast”). “Echo slow” has fewer repetitions than “echo fast”.
“distortion: soft”, “distortion: medium”, “distortion: hard”	Adds three types of distortion to the sample file.
”filter low”, “filter medium”, “filter high”	Low-pass filtering, band-pass filtering, or high-pass filtering of the sample file. The filter parameters cannot be adjusted.
“reverb short”, “reverb medium”, ”reverb long”	The “reverb” effect sounds like a fading echo of the high frequency components of the sample file, rather than like a reverberation effect.
“volume x2”, “volume x4”	Roughly doubles or quadruples the volume of the sample file. For some reason, “volume x4” causes distortion in most sample files it is applied to.
“reverse”	Turns the sample file backwards.
“pitchshift” -12 to +12 semitones	Raises or lowers the pitch of the sample file. The duration of the sample file is not altered.
“resample” -12 to +12 semitones	The duration and the pitch of the sample file are changed. Positive value lowers pitch and negative value raises pitch, not the other way, as would appear logical. The timbre of the sample file is somewhat altered.
“timestretch” 50% to 200%	The sample file is played faster or slower, without changing the pitch of the sound. However, at extreme settings the original pitch barely remains and the timbre becomes somewhat altered.

Table 3.2. Features for modifying sample files in *Magix Music Maker*.

Any of these sound modifying effects can be applied several times to a sample file. However, in practice, the sample file becomes distorted after a couple of effects have been applied to it. It is possible to simultaneously remove all effects applied to a sample file.

A sample file can be modified during the playback of a piece of music. However, the calculations required to carry out the modifications can take a couple of seconds and nothing is heard from the sample file until the calculations have been completed.

The GUI does not show visually when a modification has been applied to a sample file. For instance, if a sample file is reversed, the waveform graphic on the icon is not reversed. The modifications made to a sample file can only be heard during playback.

3.3.5 Arranging Phase

After a sample file has been selected or modified, it can be duplicated or relocated on the multitrack window. The start and end points of a sample file may have to be adjusted to fit the piece of music that is being made. The start point of a sample file is adjusted by selecting the icon representing the sample file, holding L1 button, and moving the left thumb-stick to the right. The end point is changed by holding L1 button, and moving the right thumb-stick to the left. A sample file is duplicated by holding L1 button and pressing the right thumb-stick to the right. All modifications applied to this sample file are duplicated at the same time. A sample file duplicated this way appears as a single object in the GUI. This is similar to setting the sample file to loop a certain number of times. When a sample file is copied, it is duplicated and automatically pasted to the first vacant slot on the track. A sample file copied this way appears as two or more separate objects in the GUI. There is no “cut” operation as such, but the original sample file can be deleted after it has been copied, so in practice it is possible to cut a sample file from the “arranger”.

When adjusting the start and end points of a sample file, the changes are not shown visually in the icon representing the sample file. The waveform in the icon remains the same even if the start point of the sample file is adjusted. This can be confusing to the music-maker.

A sample file is repositioned on a track by holding L1 button and moving the left and right thumb-sticks to left and right. Alternatively, the repositioning can be done in quantified steps on a track by holding L1 button and pressing the left and right directional buttons. A sample file is moved to another track by holding L1 button and pressing the up and down directional buttons. By moving sample files from one track to another, sample files from different styles can be placed on the same track.

The volume of a single sample file is altered by holding L1 button and moving the right thumb-stick up or down. A change in the volume is represented visually as a change in the height of the icon representing the sample file.

A volume fade-in is applied to a sample file by holding L1 button, pressing the left thumb-stick “in” and moving it to the left and to the right. A fade-out is applied to a sample file by holding L1 button, pressing the right thumb-stick “in” and moving it to the left and to the right. Fade-ins and fade-outs are represented visually through graphics that appear on top of the icon of the sample file.

A track can be muted or unmuted. The command for unmuting a muted track is confusingly called “mute track”.

The time signature is fixed at 4/4. The tempo cannot be adjusted.

3.3.6 Mixdown Phase

When a number of sample files have been placed in the “arranger”, the volume levels and pan positions of each track have to be adjusted. The “mixer” window is selected from the “toolbar” and appears on the top of the main window. The “mixer” window displays volume sliders and pan position knobs for six tracks, an effects selecting box, effect wet/dry level slider, and a master volume slider. Animated sound pressure level meters are displayed for each of the six tracks, as well as for the master volume. The music-maker moves between these with the directional buttons of the controller and makes adjustments with the right thumb-stick. The playback of a song is started and stopped with the start button on the controller.

The volume faders and pan position knobs of six tracks can be controlled at one time in the “mixer”. Track volume and pan position can be changed manually to one track at a time before or during playback. The six tracks are those that are visible at the time in the “arranger”. If the music-maker wants to control the volume of a track that is not currently visible, he must go back to the “arranger”, scroll the window so that it displays the desired track, and then return to the “mixer”. The category of the track (“bass”, “drums”, “pad”, etc.) is shown as the track’s label in the “mixer”. However, as described earlier, sample files from different categories can be placed to the same track. The name of a track is defined by the category of the sample file that is first placed to the track. The track name can be reset by deleting all sample files on the track. *Magix Music Maker* enables live mixing of one track at a time as the user interface makes it possible to choose tracks and modify their properties fast enough with the controller.

In addition to reverb and echo effects that are applied individually to each sample file, one master effect can be defined from the “mixer”. This effect affects the sample files on all tracks with equal intensity. The master effects are labeled “room”, “studio A”, “studio B”, “studio C”, “hall”, “pipe”, “space”, “echo” (a single delay), and “delay” (multiple delays, rapid fade). All of these are different types of reverb, because even the “echo” and “delay” effects appear to produce some reverberation. The type of the effect and the wet/dry relationship of the master effect can be adjusted manually before and during playback.

It is possible to do scratching[31] during playback in the “arranger” by pressing the right thumb-stick to left and right. This affects the material on all tracks. Fast forwarding at double speed (and at double pitch) is done by holding the right thumb-stick to the right during playback. It is possible to control the master volume in the “arranger” in real-time by pushing the right thumb-stick up or down.

3.4 MTV Music Generator 2

3.4.1 Generic Description

MTV Music Generator 2 was developed by Jester Interactive and released in 2001. The product package claims the software is from the makers of *Music* (1998) and *Music 2000* (1999) (both titles are for PlayStation). Apparently, *MTV Music Generator* (NTSC version released in North America) for PlayStation is very similar to *Music 2000* (PAL version

[31] Rose (1994: 53) defines ‘scratching’ as follows: “Scratching is a turntable technique that involves playing the record back and forth with your hand by scratching the needle against and then with the groove.” Deejay Grandmaster Flash is recognized for perfecting scratching and making it known.

released in Europe). *MTV Music Generator 3* for PlayStation 2 and XBox was released in 2004.[32]

The term 'generator' refers to an automated or at least computer-assisted process of making music. Such type of authorship is along the lines with which Manovich (2001; 2006 [2002]) describes the making of art with new media tools. It is interesting that the developers of *MTV Music Generator 2* suggest that making of music is generating, that is, something automated and impersonal. Perhaps the term 'generator' is appropriate to describe, for example, the uses of the chord and arpeggio generating features.

MTV Music Generator 2 is advertised with the persona of deejay David Morales, who appears on the front cover of the product package and in a brief video clip in the software. Morales' introduction video appears to have been recorded in MTV studios in New York City. This introduction is only a couple of minutes long and is more like a commercial, where Morales uses *MTV Music Generator 2* together with his deejay equipment. The texts in the product package claim the software is a tool for making music: "Whether you're a superstar DJ or just starting out, MTV Music Generator 2 is all you need to create killer tunes in minutes." The advertising of *MTV Music Generator 2* seems to present the software as a more serious tool than, for instance, the advertising copy for *Magix Music Maker*.

When the software is started, the first thing shown is the introduction video with Morales. Next follows a tutorial screen, where the controls are explained. The tutorial in the software consists of a single screen with an image and some written text. There is a more hands-on tutorial to music-making in the printed user's guide explaining the stages of making a piece of music. The tutorial and software commands are available in five languages: English, French, German, Italian, and Spanish.

If the music-maker does not do anything for some time, a demo song is loaded automatically and the playback begins. However, this song is not loaded to the multitrack window for the music-maker to start modifying immediately (the way demo songs are loaded in *eJay Clubworld*).

MTV Music Generator 2 includes adaptations of songs originally by artists and bands including Apollo 440, Bentley Rhythm Ace, Cassandra Fox, Dano, Funkmaster Flex, Gorillaz, Halo Varga & Dano, Photek, Roni Size, Tony Hewitt, and Zombie Nation. These songs can be used as bases of remixes.

According to the product package, about 11,000 sample files are included in *MTV Music Generator 2*. These are referred to as "samples" in the software. There are three versions of every sample file at sample rates "11", "22", and "44" KHz. It is not explained in the user's guide if these different versions are included in the total number of 11,000 sample files. If this is the case, the actual number of "unique" sample files is about 3,660. Nevertheless, many of the "same" sample files do sound different at different sample rates.

[32] The apparent similarity with *MTV Music Generator 2* and MTV television networks is that in the videos made with the software, the famous MTV logo appears superimposed on top of the video. However, the logo is displayed close to the bottom right corner, rather than the top right corner, as has been the convention for MTV television channels.

3.4.2 GUI

The pointer is moved with the left thumb-stick of the DualShock 2 controller. Selections are made with the cross button. Lists are scrolled up and down with the up and down directional buttons. The contents of folders for instance in the “sample library” are scrolled with left and right directional buttons. Menus are opened by placing the pointer on top of some GUI component and pressing the circle button. A GUI component such as a pattern or a note icon is deleted by placing the pointer on top of it and pressing the square button. Just like in *Magix Music Maker*, the playback of a piece of music is started and stopped with the start button on the controller.

The multitrack window is called “track window”. There time is represented vertically, from top to bottom. Every measure is numbered. The maximum length of a song is 999 measures. The measure number can be set as a bookmark by clicking on it with the cross button. “Labels” with some written text can be added next to the measure numbers. These are for marking the beginning of a new part in a piece of music. The “track window” is scrolled in four directions with the directional buttons.

There are four types of tracks, namely “rhythm”, “bass”, “melody”, and “vocals”. The music-maker can insert “riffs” of these four types into the track window. The maximum number of tracks is 48. Although several sample files can be placed simultaneously on a track, the maximum number of overlapping sample files during playback seems also to be 48. If this number is exceeded, some sample files can no longer be heard.[33]

Each track contains a number of “riffs”. The same “riff” can be duplicated on the same track or some other track of the same track type. One track can contain different “riffs”. Each “riff” can have a different volume level. The icon of each “riff” is a colored block that has a specific color depending on the type of the track. There are a couple of variations of each color, so the music-maker can distinguish the “riffs” by their color.

The length of a “riff” may be one or several bars. A “riff” is selected from the “riff palette” located in the left border of the screen. A “riff” is made in the “riff editor”, which is a separate window from the “track window”. A sample file appears as a “note” in the “riff”. Each “note” can be routed to the master effect. A “note” can also be routed to rear loudspeakers. Thus, *MTV Music Generator 2* supports surround sound, but this feature was not examined because there was no suitable decoder available.

The time of the playback position and pointer position are displayed in the upper right corner of the multitrack window. At the lower right corner of the screen, four animated blocks display each beat of a measure during playback. On the left side of the main screen are buttons for the following features: “undo”, “redo”, ”options”, “load/save”, “skin selection”, and “jam session”. On the bottom side of the main screen are buttons for the following features: “goto song”, “goto video”, “goto start”, “rewind”, “play”, “fast forward”, and “goto end”.

Envelopes are used to define the changes in volume, pitch bend, stereo panning, and filter cutoff. Depending on the feature they control, some envelopes have only a start point

[33] 48 audio channels is a limit set by PlayStation 2 hardware. However, this limitation has been worked around in some titles, such as *Music 3000* supporting 64 audio channels.

and an end point, while other envelopes have several folding points that can be defined by the music-maker.

3.4.3 Selecting Phase

“Riffs” are browsed and selected from the “riff library”. When a “riff” has been selected from the library or defined by the music-maker, it is automatically added to the “riff palette” in the “track window”.

Sample files are browsed and selected from the “sample file library”. “Notes” (sample files played at certain pitches) include single notes of certain instruments played at a certain pitch, and patterns with several notes or drum beats. Some sample files contain music from multiple instruments such as drums and bass (e.g., sample file called “bass groove 1” that is stored in the “bass” category). Hence, it can be confusing to the music-maker what actually is a “riff” and what is a “note” (sample file). When a sample file has been selected from the library, it appears in the “sample palette” in the “riff editor” window.

“Riff library” and “sample library” are organized so that there is a different selection available depending on the type of the track that is being made (“rhythm”, “bass”, “melody”, or “vocals”). “Samples” and “riffs” in each selection are divided into folders. Folders are changed with left and right directional buttons. The lists are scrolled with up and down directional buttons on the controller.

The ready-made “riffs” are grouped into eight categories (that appear in the GUI as folders) called “Breakbeat”, “Garage”, “House”, “Indie”, “Pop”, “Rock”, “R&B”, and “Trance”. About 1,200 ready-made riffs are included. These can be modified in the “riff editor”. “Riffs” can be listened to from the lists before they are selected.

There is a version of every sample file available in three sample rates: “11”, “22”, and “44” KHz. Before they are chosen, sample files can be listened to at all these sample rates. Obviously, the sample rate of the selected sample files has an effect on memory consumption. The higher the sample rate, the more memory a sample file requires.

The “riff” and sample file names are not entirely descriptive in every case because they are typically shorter than ten characters. Multiple “riffs” and sample files can be selected while browsing through them, and then be loaded simultaneously.

3.4.4 Sampling Phase

Recording samples is possible with an USB sampling device that is sold as a separate accessory. At least two models of an USB sampler has been released for PlayStation 2; one is distributed by Codemasters (for *MTV Music Generator 2*) and the other by Jester (for *Music 3000*). In this study, the sampling feature was used with the Jester version (the device was manufactured by Joytech). There is a microphone connection and a stereo line level input in the USB sampling device. A small microphone was distributed with the USB sampling device by Jester.[34]

It is possible to sample a maximum of about 23 seconds of sound in one take at the sampling rate of "44 KHz" (that is apparently the CD audio standard sampling frequency of 44.1 KHz). Only single-channel (mono) sampling is possible. There is no volume level control in the software. To avoid distortion during sampling, the appropriate volume level must be set from the sound source.

A waveform representation of the sampled material is drawn to the screen during recording and when monitoring the input. There is a delay of about one second between a sound and its visual representation appearing on the screen.

"Riffs" and "samples" loaded from the software CD fill the memory of the PlayStation 2. In practice, there is memory space for a total of about 60 seconds of material sampled by the music-maker. Sample files recorded by the user cannot be resampled to a lower sampling rate. The user-made sample files are saved as part of the song on the memory card.

3.4.5 Modifying Phase

After a sample file has been selected from the sample library or recorded by the music-maker, it can be modified. In *MTV Music Generator 2*, this is done in the "sample editor". The features of "sample editor" are explained in Table 3.3.

[34] The microphone included with the video game *SingStar* can be used in *MTV Music Generator 2* to sample sound.

Features	Description
“cut”, “copy”, “paste”	These features are used to cut, copy, and paste the selected section of a sample file.
“zoom in”, “zoom out”	Brings the selected section of the waveform graphic of a sample file closer or further away in the “sample editor” window. This means that more or less detail of a sample file is shown.
“trim”	Used for deleting everything outside the selected section of a sample file.
“normalise”	The volume of a sample file is increased to the highest possible value before clipping.
“reverse”	The selected section of a sample file is turned backwards.
“mix”	Whatever has been copied (a user-made sample or a sample file from the library) can be mixed with any sample file. The mixing ratio is 50/50.
“select all”	Used to select the entire sample file.
“remove DC”	Corrects the DC offset in a sample file.
“silence”	Silences the selected section of a sample file.
“set loop point”, “delete loop point”	Used to define and remove the loop point in a sample file.
“low pass filter”, “band pass filter”, “high pass filter”	Low-pass filtering, band-pass filtering, or high-pass filtering of the selected section of a sample file. The filter settings are set with a user-definable envelope with several folding points. The envelope graphic represents the filter cutoff frequency in relation to time, from the beginning to the end of the selected portion of a sample file. The cutoff frequency, or frequencies in the case of the band-pass filter, and the number of the sample within a sample file are also shown.
“time stretch”	Extends or abridges the duration of a sample file (supposedly) without affecting its pitch. This effect applies to the entire sample file. The amount of “time stretch” is defined in steps of 5%. For instance, at 50% setting the sample file is replayed faster compared to the original, and at 150% setting it is replayed slower.
“volume -5%”, “volume +5%”	Decreases or increases the volume of the selected region of a sample file by 5%.
“USB Sampler”	Starts the sampling feature.

Table 3.3. Features for modifying sample files in *MTV Music Generator 2*.

In the sample editing window, the waveform of a sample file is displayed so that time is represented horizontally and amplitude horizontally.

With “downgrade sample” and ”upgrade sample” features, in addition to a change in the sampling frequency, the duration and pitch of sample file are also affected. Sample files from the library can be modified in the same ways as sample files recorded by the user. However, user-made sample files cannot be downgraded or upgraded.

In “sample editor”, the region selected from a sample file is called “edit area”. If the sample file that has been copied or cut is pasted into or mixed with a sample file with a lower or higher sampling frequency, the pasted or mixed sample file will be replayed faster or slower and an octave lower or higher in pitch.

The option for undoing the previous command is not effective in “sample editor”. This is a major constraint, because all changes made to a sample file are either kept or lost if a single mistake is made. Because of this, it is more laborious to experiment with the various sample file manipulations.

3.4.6 Arranging Phase

With *MTV Music Generator 2*, the phase of arranging consists of building “riffs” from sample files and then making a piece of music from these “riffs” and prefabricated “riffs” selected from the library.

The multitrack window is labeled “track window”. It displays a number of tracks side by side. Time is represented vertically, from top to bottom. The following tracks are shown from left to right in this order: “song volume”, “song BPM”, “song effects”, “rhythm” track, “bass” track, “melody” track, and “vocals” track. Four additional tracks are for the video-making feature.

When the music-maker inserts or defines a “riff” to a “rhythm”, “bass”, “melody”, or “vocals” track, a new empty track of the same type is automatically added to the right side of the track where the “riff” was inserted or defined.

By placing the pointer on top of a track and measure, and pressing the circle button, a menu with the following features appears: “riff editor”, “riff library”, “full screen video”, and “mute track” (or alternatively “unmute track”, if the track is already muted).

The “riff editor” is visually and functionally similar to the piano-roll editing feature in many MIDI sequencer software, except that in “riff editor” pitch is represented horizontally (pitch increases towards the right side of the window) and time is represented vertically progressing from top to bottom. In most MIDI sequencer software, time is represented horizontally and pitch is represented vertically. Pitches from notes C1 to C7 are available in the “riff editor”. The pitches are not absolute, but depend on the pitch defined for the sample file and its sample rate. An area of a little less than three octaves and slightly more than one measure is displayed at one time in the “riff editor”. The “riff editor” window is scrolled vertically (time) and horizontally (pitch) with the directional buttons on the controller.

The first thing to do in “riff editor” when creating a new pattern is to select a sample file from the “sample library” or to start the “USB sampler” feature. After sample files have been selected, icons representing them appear on the left side of the “riff editor” screen. A sample file is selected to use in “riff editor” through the icon representing it.

The pitch and duration of a new “note” are defined on the piano-roll grid by selecting the appropriate pitch and start point with the pointer and pressing the cross button. While the cross button is being held down, the length of the “note” is defined by pressing the left thumb-stick down until the “note” is of the desired length. The parameters of the previous note (or those of a note selected by the music-maker through the feature “use this note’s settings”) are by default applied to subsequent notes.

Each “note” has the following settings: pitch (chosen from an equally tempered chromatic scale), “pitch envelope”, “pan envelope”, “volume envelope”, and “note parameters”. In the “note parameters” window, it is possible to define the start point of a sample file through its waveform graphic, the pitch bending of a “note” (defined through a keyboard graphic on the screen), muting the “note”, routing the sample file to rear loudspeakers, and the volume between 0% and 100%. There are sixteen “trigger” points for triggering the sample file and routing (or not routing) it to the “song effect”. Some notes in a “riff” may be routed and some other “notes” may not be routed to the “song effect”. Confusingly, this feature is called “on with reverb”, even if this applies to all the other master effects in addition to reverb.

When a group of “notes” have been selected, the following features become available: “copy selection”, “cut selection”, “delete selection”, “note volume envelope”, “note pitch envelope”, “note pan envelope”, “note parameters”, “note volume over time”, “note offset over time”, “note pan over time”, “humanise”, and “real time record”.

The “note auto chord” feature generates a chord based on the selected “note”, which becomes the root note of the chord. The following chord types can be generated: “major”, “minor”, “seventh”, “minor seventh”, “major seventh”, “sixth”, “minor sixth”, “augmented”, “diminished”, “suspended”, “suspended second”, “minor seventh flattened fifth”, “minor ninth”, and “eleventh”.

A “riff” may contain different overlapping sample files (“notes”) from the same instrument category, which is defined by the type of the track. However, these sample files cannot overlap on the same “note” pitch in the “riff editor”. User-made sample files can be inserted into “riffs” of all types of tracks.

Sample files may be looped. If there is no loop defined in a sample file, the length of a “note” in “riff editor” does not correspond to the length of the sample file after the end of the sample file has been reached. The “note” can be much longer in the “riff” than the length of the sample file. The feature “replace sample” makes it possible to set a different sample file to the existing “notes” in a “riff”.

When the music-maker has made a “riff”, it appears in the “song palette” in the left corner of the multitrack window. The name of the “riff” appears when the pointer is placed on top of the icon representing the “riff”. By pressing the circle button on the controller, a menu with the following features appears: “use riff”, “delete riff”, “clone riff”, “demo riff”, “riff volume”, “rename riff”, “random colour”, and “full screen video”. The “riff” can be listened to (“demo riff”) by pressing the start button on the controller. “Riffs” from the “song palette” can be selected and inserted into tracks during the playback of a piece of music.

The tempo of a piece of music is defined on a special track called “song BPM” in the multitrack window. Tempo changes are inserted into this track. The tempo change may take place immediately or as a transition and it is up to the music-maker to define how many measures this transition lasts. Three different “ramps” are available for defining the tempo transition. The tempo can range from 40 BPM to 500 BPM.

The following time signatures are supported: 2/4, 3/4, 4/4, 5/4, 6/4, and 7/4. It is possible to use “riffs” from the “riff library” only when the time signature of the song is 4/4.

3.4.7 Mixdown Phase

The loudness and pan position of each sample file are not defined through the properties of a track, but on the level of each “note” in a “pattern”. Hence, there is no real-time control over the volume level or pan position of each track or “riff”. Individual tracks can be muted and unmuted, but not during playback.

One track called “song effects” is for defining the master effect that affects the “notes” defined by the music-maker. A “song effect” is represented as a block in the “track window”. The settings of the “song effect” are brought to the screen by placing the pointer on top of the block icon on the track and pressing the cross button. Nine effects are available as “song effect”, including “room”, “studio A”, “studio B”, “studio C”, “hall”, “space”, “pipe”, “echo”, and “delay”. All effects have a “start depth” and an “end depth” setting. “Depth” affects the wet/dry ratio. “Echo” and “delay” effects also have “start delay”, “end delay”, “start feedback”, and “end feedback” settings. All these settings are adjustable within the range from 1 to 100.

Only a single effect can be in use at a time. Every effect has a start value and an end value. This is used to generate transitions such as an increasing the amount of reverberation over time. Transitions are defined with the accuracy of one measure.

Although no echo feature is available, it is possible to achieve an echo-like effect by defining in the “riff editor” that a sample file is to be replayed twice but the second time slightly later, at a lower volume level, and possibly at a different pan position. This method has been used, for instance, in a lot of tracker music.

The playback of a piece of music can be “scratched” by pressing the up button (for fast backward playback) and the down button (for fast forward playback).

3.5 eJay Clubworld

3.5.1 Generic Description

eJay Clubworld was developed by Unique Development studios and released in Europe in 2002. The subtitle in the product package is “the music making experience”. The product package carries blurbs like: “Make your own hit without any musical training in a matter of minutes. Eight real-world clubs and locations are waiting for you: each with a different and current music style for your creativity to explore.” In other words, it is suggested that *eJay Clubworld* is a kind of a simulator that takes music-makers to “real-world clubs” with “user-friendly tools” such as “synthesizer”, “drum machine”, and “virtual turntable”. It should be noted that the texts of *eJay Clubworld* promise “exploring creativity”, as did the advertising copy of *Magix Music Maker*. Like *MTV Music Generator 2*, the advertising texts of *eJay Clubworld* suggest that it is easy and quick to make music with it.

For some reason, *eJay Clubworld* was given a 15+ user age recommendation by ELSPA (European Leisure Software Publishers Association), whereas *MTV Music Generator 2* and *Magix Music Maker* were given a 3+ age recommendation.

Eight musical styles are represented as different clubs around the world: “Ambient” (“Athena”, in Bombay), “Electro” (“Zouk”, in Singapore), “Reggae” (“Old Mill”, in Virgin Islands), “Hip-Hop” (“Brooklyn Bridge”, in New York), “Drum & Bass” (“The End”, in London), “House” (“Queen Club”, in Paris), “Tech House” (“Amnesia”, in Ibiza) and “Techno” (“U 60311”, in Frankfurt). Different styles (clubs) have slightly different features available.

When the software is started, the first thing shown is a brief introduction video with deejay Carl Cox. This functions similarly to the attraction mode in, for instance, arcade games. After the video follows the main menu. Here the music-maker can select between one of the eight clubs, the “jukebox”, and the “tutorial”.

In the “jukebox”, songs made in different clubs and stored on the memory card are placed on a playlist and replayed from there.

The “tutorial” covers the following topics: “the main controls”, “the basics of arranging”, “mixer”, “groove generator”, “hyper generator”, ”scratch mode”, “live jam”, “video mode”, and “jukebox”. The tutorial is narrated in English, subtitled in various languages, and shows animations of the topics explained by the narrator. Five languages are available for subtitles: English, French, Italian, Spanish and German. Most of the features, sample file names, and categories are in English in spite whatever language has been selected.

When a club has been selected and before it is being loaded from the disc, some trivia and a video about the club are shown on the screen, possibly to help convey the sense of the music-maker being transferred there.

A demo song, called “resident mix” in the user’s guide, is loaded every time when the music-maker enters a club. There is a different demo song in each club. This is a good way in the beginning to “push” the music-maker to do something. The “resident mix” can be remixed or used as a starting point for a new piece of music.

About 10,000 sample files are included in *eJay Clubworld*. These are referred to as “loops”, “one shots”, and “sounds” in the software and as “sound samples” in the user’s guide. Some sample files have a noticeable stereo effect such as panning between the channels. Most sample files have some amount of reverberation in them, notably more than the sample files in *MTV Music Generator 2*.

“Brooklyn Bridge” club has a special “scratch mode”, where the playback of one sample file is looped continuously at different speeds (and at different pitches) forwards or backwards. The “scratched” sample file is mixed with a piece of music. It is possible to change the tempo (and pitch) of the song or to mute it.

3.5.2 GUI

The main menu has these options: “tutorial”, “jukebox”, and the eight clubs. I have discussed the “tutorial” and “jukebox” earlier. The music-maker moves from one club to another via the main menu.

After entering a club, the music-maker moves through a transitional screen between the following screens: "arranger", "mixer", "live jam", "video mode", "hyper generator" (only in two clubs), "groove generator" (in only two clubs), "scratch mode" (in only one club), and "exit club" that takes the user back to the main menu. The user enters the intermediate screen by pressing R2 button.

Each screen is divided into "operation fields". The music-maker switches between these by holding down R1 button and pressing the directional buttons. There is no pointer available in most windows of the GUI. Various options and features are moved through with the four directional buttons, and on most screens selections are made with the cross button. Pressing the triangle button starts or stops the playback of the song and L2 button rewinds to the beginning of the song.

The multitrack window is called "arranger". There time is represented horizontally, from left to right. The numbers of the measures appear on the top of the screen. The "arranger" shows twelve measures from nine tracks at the same time.

The "arranger" is divided into five "operation fields". The "controls" field is for controlling the master volume and the song tempo. There is also a reset button for restoring the original settings of these two. The sample file library is divided into two "operation fields". One of them is for selecting sample file category and the other for browsing and selecting sample files. One "operational field" includes the tracks. Here sample files are added, moved or deleted through icons representing them. The music-maker controls a pointer in the form of a crosshair viewfinder. One "operational field" is for muting and unmuting tracks. There is also a solo mode for selecting one track and muting all the other tracks, or muting selected tracks. One feature not mentioned in the user's guide is that the effect of 'chopping' can be achieved during playback by holding down the button that browses through the track-muting modes.

The "arranger" contains 20 tracks and in certain clubs there is also one "hyper generator" or "groove generator" track. All tracks appear to contain stereo sample files, even if the contents of a track are represented visually as single-track files. The maximum length of a song is 999 measures. The time signature is fixed to 4/4. The current position in the song during playback is displayed with an animated line.

The visual representation of a sample file is a block with a colored line and a number from 000 upwards. The category of the sample file can be recognized through this color. The name of a sample file is not displayed in the "arranger". The waveform of the block is animated in the "arranger" during playback. The animation looks like an oscilloscope showing a waveform. This visual representation is rather useless as a visual representation of the contents of the sample file. There are two controls in the "arranger", namely "VOL" (volume) and "BPM" (tempo). There is also a reset button on the screen for restoring the original values to these two controls.

The "mixer" screen is divided into six "operation fields". The "controls" and track-muting fields are the same as in the "arranger". The "mixer" screen displays the settings of each track: "LEVEL" (a horizontal slider for defining volume level), "PAN" (a knob for setting pan position), "DLY" (a knob for adjusting delay effect wet/dry level), "CHR" (a knob for adjusting chorus effect wet/dry level), and "RVB" (a knob for reverb effect wet/dry level). There is also a horizontal volume level “meter” for each track. “LEVEL” is in a separate “operation field” from the other track controls. The track settings are adjusted so that the music-maker first moves the selection (represented as an illuminated knob) to the desired control and selects it by pressing the cross button (so the knob starts to blink). Nine tracks are shown simultaneously in the “mixer”. The “operational field” labeled “racks” is for selecting “send” and “master” racks that appear alternatively to the right of this field. The “operational field” with “send” and “master” rack contains the controls for adjusting “delay”, “chorus”, “reverb”, “EQ”, and “compressor” effects.

“Hyper generator” in clubs “U60311” and “Amnesia” is divided into six “operation fields”. One of these is a pattern sequencer, where pitch is represented vertically and time horizontally. Time is quantified in 1/16th notes, so there are sixteen “notes” in each pattern (measure). “Notes” appear as dots in a “matrix” next to an organ keyboard. The music-maker moves a crosshair pointer in the “matrix” with the directional buttons and left thumb-stick on the DualShock 2 controller. The current location within each pattern is displayed with an animated line during playback. The sample file category is selected in one “operational field” and the sample file is selected in another field. One operational field is for moving, copying, and deleting patterns. The “matrix” can be reset in one of the “operation modes”. The “controls” and track-muting fields are the same as in the “arranger”.

The “groove generator” in clubs “The End” and “Queen Club” is divided into seven “operation fields” in some ways similar to the “hyper generator”. Thus, I will describe only the differences in the following. There are ten tracks in the pattern sequencer window. Each track can have a different drum sample file assigned to it. Drum sample files are triggered by “kicks” defined by the music-maker. As in “hyper generator”, time is quantified in 1/16th notes. “Kicks” appear as round dots in the “matrix”. One “operational field” is for selecting one of the four modes for making patterns in “groove generator”. These modes are called “matrix”, “jam”, “record”, and “configure”.

“Scratch mode” in club “Brooklyn Bridge” has four “operation fields”. One of these contains two turntables. One turntable controls the piece of music defined in the “arranger”, and the other turntable controls one sample file that is replayed in a loop. The left turntable is controlled with the left thumb-stick, and the right turntable is controlled with the right thumb-stick. Moving the thumb-stick to the right increases the playback speed of the material in both turntables. On the left turntable, moving the thumb-stick to the left decreases the playback speed. On the right turntable, moving the thumb-stick to the left plays the sample file in reverse at the speed defined by the amount by which the thumb-stick is moved. Hence, scratching is possible in a way sounding somewhat like scratching on a vinyl turntable. The left turntable can be muted by pressing L1 button and the right turntable can be muted by pressing R1 button. There is a horizontal slider for defining the mix ratio between the material from these two turntables. This slider is controlled with left and right directional buttons. The song is on the left hand turntable. The sample file on the right hand turntable is selected from two “operation fields”. In one field the sample file category is selected and in the other field the sample files are browsed and selected in a similar way as in the “arranger”. The controls for master volume and tempo are in a similar “operation field” as on the other screens.

“Live jam” is a mode in which the music-maker triggers sample files manually. Several controller devices can be used for this purpose. Hence, a number of people can “jam” together in real-time. In “video mode” an animation is shown and the music-maker has some real-time control over the abstract animated graphics.

3.5.3 Selecting Phase

Sample files selected from the “sample archive” are the basis for making a piece of music with *eJay Clubworld*. Sample files are divided into the following categories (called “sound groups” in the user’s guide): “bass”, “keys”, “males” (vocals), “females” (vocals), “drums” (typically one-shots), “guitar”, “loop”, “fx” , ”extra”, “spheres”, and “vocal”. Sample files from a maximum of eight of these categories are available in a club. The number of sample files in each category varies between the categories from about a dozen to almost two hundred.

Samples are browsed so that the category is selected first and then the “operation field” is set to show the sample files. After that the sample files can be browsed with the four directional buttons. Sample files can be previewed by pressing the square button.

The length of the sample files varies from one measure to nine measures. Sample files can be browsed but they cannot be previewed from the “sample archive” during the playback of a piece of music.

3.5.4 Modifying Phase

Sample files are modified through the “mixer” screen. All sample modifying effects affect the contents of an entire track, that is, a single sample file or several sample files. Thus, modifying sample files could be interpreted as being part of the mixdown phase. Nevertheless, I examine the phase of modifying sample files with *eJay Clubworld* as a distinct phase of the nonlinear multitrack composing process as I have done with the two other music-making software titles. Features for modifying sample files are shown in Table 3.4.

Feature	Description
“delay”	An echo-type effect applied to all sample files in a track. The wet/dry level is defined separately for each track. The following parameters of the “delay” effect can be adjusted: “feedback”, “predelay”, and “volume”.
“chorus”	Bartlett and Bartlett (1998: 405) describe chorus as a “wavy, shimmering effect”. This effect is applied to all contents of a track. The wet/dry level is defined separately for each track. Two parameters of the “chorus” effect can be adjusted: “speed” and “drive”.
”reverb”	A reverberation effect applied to all sample files in a track. The wet/dry level is defined separately for each track. The following parameters of the “reverb” effect can be adjusted: “time”, “predelay”, and “volume”.
“EQ”	Ten-band equalization affecting every track or is not used at all.
“compressor”	Compression effect that affects all or no tracks. The following parameters of the “compressor” effect can be adjusted: “drive”, “speed”, and “gain”.

Table 3.4. Features for modifying sample files in *eJay Clubworld*.

An interesting detail is that at the shortest “predelay” levels, the “delay” effect sounds much like a flanging effect. Bartlett and Bartlett (1998: 412) define ‘flanging’ as “[a] special effect in which a signal is combined with its delayed replica, and the delay is varied between 0 and 20 milliseconds. A hollow, swishing, ethereal effect like a variable-length pipe, or like a jet plane passing overhead.” “EQ” and “compressor” are typically used during mixdown for shaping the overall tonal balance of a piece of music. In “Scratch mode” in “Brooklyn Bridge” club, one selected sample file can be played in reverse by controlling the playback in real-time. However, as it requires manual real-time input, this cannot be interpreted as a similar function to reversing a sample file in *Magix Music Maker* and *MTV Music Generator 2*.

3.5.5 Arranging Phase

In *eJay Clubworld*, each track can contain sample files from different categories. This may appear confusing to the music-maker because there is a lot to remember, including what sample files are located in what tracks. The tracks cannot be labeled, so they can be identified only through their numbers in the “mixer”.

Sample files are aligned in the “arranger” with the precision of a measure. The start points of the sample files are always automatically perfectly aligned. It is possible to adjust the end point of a sample file with the precision of one measure by placing another sample file on top of the latter part of a sample file on a track. The added sample file may then be removed but the earlier sample file is now abridged.

Changing tempo affects the overall pitch of the song. The difference between the maximum and minimum tempo varies between about 25 BPM to 50 BPM depending on the club.

The music-maker can define his own patterns with “hyper generator” and “groove generator”. “Hyper generator” is available in clubs “U60311” (with “Techno” style) and “Amnesia” (with “Tech House” style). The tutorial suggests that “hyper generator” is for creating “wicked sounds and melodies”. “Hyper generator” patterns are called “sequences” in the tutorial. “Hyper generator” has a two-octave range (a to a, chromatic scale). Each measure is a grid, or “matrix”, as it is called in the software, of 25*16 notes. The “matrix” appears so that pitch is represented vertically (increasing from bottom to top) next to an organ keyboard graphic. Time is represented horizontally from left to right. The resolution is 16 notes per measure. Notes can be inserted into the “matrix” while the one-bar long pattern is being replayed in a loop and also when it is not. The pattern is monophonic, that is, only one sample file can be triggered at a time. Each “note” triggers a sample file at the specified pitch. The sample file is replayed in its entirety unless there is a new note in the pattern replacing or retriggering it from the beginning. One pattern can contain notes from a single sample file. The sample file may be different in every pattern.

It is not possible to transform a pattern made in “hyper generator”, for instance, by transposing all notes simultaneously. All modifications to a pattern have to be made manually note by note.

In “hyper generator”, the sample file can be selected from four categories called “acid”, “bass”, “extra”, and “mellow”. Some of the sounds in club “Amnesia” (in groups “acid” and “extra”) are percussive, so certain types of drum and percussion patterns can be made with “hyper generator”.

“Groove generator” is available in clubs “The End” (“Drum & Bass” style) and “Queen Club” (“House” style). “Groove generator” is a “matrix” of 10*16 drum beats. Drum sample files are displayed on top of each other. Time is represented horizontally in a similar way as in “hyper generator”. The sample files in the grid can be changed. Sample files are chosen from categories called “kick”, “snare”, “hihat”, “percussion”, and “fx”. The setup of drum sample files is the same for all patterns. If a sample file is changed in one pattern, the change affects all patterns in the song. Up to ten drum sample files can be triggered simultaneously in a pattern.

“Groove generator” has four modes called “matrix”, “jam”, “record”, and “configure”. Notes can be inserted while the one-bar long pattern is being replayed in a loop and also when it is not. Each “note” triggers the sample file, which is played in its entirety unless there is a new note re-triggering it on the same track. The volume level of each drum track cannot be adjusted separately. However, accents can be realized by placing the same or similar sample files on two or more tracks on the accented beats.

The one-bar long patterns made in “hyper generator” or “groove generator” appear in the “arranger”. The patterns can be deleted, moved, and copied in the “arranger”, “hyper generator” or “groove generator”. Several sample files from several tracks can be copied and moved simultaneously. A pattern made with “hyper generator” or “groove generator” cannot be duplicated so that if a copied pattern is modified, the original pattern would remain unaltered.

3.5.6 Mixdown Phase

Track volume and pan position settings are adjusted on the "mixer" screen. Nine tracks are visible at one time in the "mixer". Because the tracks are not labeled in the mixer, the music-maker cannot see what sample files each track contains. However, when a sample file is being replayed on a track, this is displayed in real-time as an animation in the volume meter of each track.

The "mixer" screen has two "racks" labeled "master" and "send". These overlap, so they are accessed one at a time. The "master rack" contains a compressor and a ten-band equalizer. The bands are labeled "40 Hz", "80 Hz", "160 Hz", "320 Hz", "640 Hz", "1.3 KHz", "2.5 KHz", "5 KHz", "10 KHz", and "20 KHz". This is an approximation of an octave band equalizer. The levels of how much the equalization emphasizes or de-emphasizes each band are not explained. This means that the music-maker must decide on the appropriate equalization level by experimenting and listening. "Compressor" has three adjustable parameters called "drive", "speed", and "gain". Equalizer and compressor effects apply to all or none of the tracks. "Send" rack contains effects called "delay", "chorus", and "reverb". These effects apply to one or more tracks with individually adjustable levels for each track.

It is not possible to fast-forward during playback, except manually in the "scratch mode" in "Brooklyn Bridge" club, where fast-forwarding raises the overall pitch.

3.6 Analysis Part 1 Stage 2: Defining a Metaterminology of Functions

In this stage of the analysis, I define a metaterminology of functions. From each music-making software title I collect the features as they have been labeled in the software or the user's guide. I present the features in tables grouped on the basis of the metaterms. The metaterms are grouped on the basis of the five phases of the nonlinear multitrack composing process. The terminology is adapted from music technology dictionaries such as Cary (1992) and Dobson (1992), practical guides to studio technology like Huber and Runstein (1997), and Bartlett and Bartlett (1998), and tutorials including Roads et al. (1996).

3.6.1 Selecting

In all three music-making software titles, sample files are stored in folders that appear to the music-maker as sample file libraries. Metaterms are needed to describe three functions related to the phase of selecting sample files and patterns. These are 'sample file', 'pattern', and 'sample file library' as defined in Tables 3.5 to 3.7. Patterns are not sample files but selecting predefined patterns for use is an alternative to selecting sample files, and takes place during the same phase of the nonlinear multitrack composing process.

Function metaterm	'sample file'
Definition and background	A 'sample file' is the entire digital sound recording, stored as a discrete file.
Music-making software features	*Magix Music Maker*: "sound", "loop", "object" (term in user's guide), "building block" (term in user's guide) *MTV Music Generator 2*: "sample", "note" (term in riff editor) *eJay Clubworld*: "loops" and "one shots", "sounds" and "sound samples" (term in user's guide)
Note	A 'sample file' may be a recording of any sound, including single notes (pitched or non-pitched), composite sounds (e.g., orchestra hit and organ chord), to sequences (e.g., drum and ostinato patterns), and sound effects (without a distinct pitch).

Table 3.5. Definition of the metaterm 'sample file'.

Function metaterm	'pattern'
Definition and background	A 'pattern' is a sequence of a number of sample files replayed at various pitches. A 'pattern' consists of triggering commands, e.g., when to trigger what sample file at what pitch and at what volume level. The samples that are triggered are stored separately from the pattern. Thus, new sample files can be assigned to existing triggering commands.
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: "riff" *eJay Clubworld*: "sequences", "grooves"
Note	In my definition, 'patterns' are made with the 'pattern sequencer' function (see Section 3.6.4).

Table 3.6. Definition of the metaterm 'pattern'.

Function metaterm	'sample file library'
Definition and background	This term has been established in everyday use.
Music-making software features	*Magix Music Maker*: "instrument directory" *MTV Music Generator 2*: "sample library", "riff library" *eJay Clubworld*: "sound groups"
Note	I am hesitant to call these libraries 'databases' (see Manovich 2001: 37, 224) because search-term queries to their contents cannot be made. The contents of the sample file libraries of music-making software are retrieved only through browsing.

Table 3.7. Definition of the metaterm 'sample file library'.

3.6.2 Sampling

Only one metaterm, namely 'user sampling' as defined in Table 3.8, is required to describe the function of recording samples.

Function metaterm	'user sampling'
Definition and background	Sampling is the digital recording of relatively short segments of sound. Sampling also involves selecting the material for sampling and building up the recording setup.
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: "USB sampler" *eJay Clubworld*: not supported
Note	By using the term 'user sampling', I emphasize that it is the music-maker who performs the sampling instead of merely using existing material from a sample file library. Of course, the software developers have sampled the material in these libraries.

Table 3.8. Definition of the metaterm 'user sampling'.

3.6.3 Modifying

Metaterms are needed for nine functions in the phase of modifying sample files. These are 'truncation', 'normalizing', 'adjusting sample file volume', 'reverse', 'chorus', 'fattening', 'distortion', 'fixed filter', and 'dynamic filter'. These metaterms are defined in Tables 3.9 to 3.17.

Function metaterm	‘truncation’
Definition and background	Defining a new start point, end point, and optionally loop segment for a sample file, and deleting the material outside this section.
Music-making software features	*Magix Music Maker*: supported (not labeled in software; material outside the section is not deleted) *MTV Music Generator 2*: “trim” *eJay Clubworld*: not supported
Note	Adjusting sample file duration (how much of a sample file is replayed) is not included in the ‘truncation’ function.

Table 3.9. Definition of the metaterm ‘truncation’.

Function metaterm	‘normalizing’
Definition and background	Huber and Runstein (1997: 476) define ‘normalizing’ as “[a] specialized gain-related process that makes the best use of a digital system’s dynamic range by automatically determining the amount of gain required to increase the level of the highest amplitude signal to its full-scale amplitude value, and then increasing the level of the selected region or entire file by this gain ratio.” By normalizing, the volume level of a sample file is set at the highest possible value before clipping.
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: “normalise” *eJay Clubworld*: not supported

Table 3.10. Definition of the metaterm ‘normalizing’.

Function metaterm	‘adjusting sample file volume’
Definition and background	Increasing or decreasing the volume of a sample file. The increase or decrease may be a constant setting or follow an envelope with several folding points.
Music-making software features	*Magix Music Maker*: “volume x2”, “volume x4” *MTV Music Generator 2*: “volume -5%”, “volume +5%” *eJay Clubworld*: not supported

Table 3.11. Definition of the metaterm ‘adjusting sample file volume’.

Function metaterm	'reverse'
Definition and background	Manipulating a sample file so that it is replayed backwards.
Music-making software features	*Magix Music Maker*: "reverse" *MTV Music Generator 2*: "reverse" *eJay Clubworld*: not supported
Note	In *eJay Clubworld*, one sample file at a time can be replayed in reverse in "scratch mode". Thus, this feature cannot be considered an instance of the 'reverse' function.

Table 3.12. Definition of the metaterm 'reverse'.

Function metaterm	'chorus'
Definition and background	Bartlett and Bartlett (1998: 405) define 'chorus' as "[a] special effect in which a signal is delayed by 15 to 35 milliseconds, the delayed signal is combined with the original signal, and the delay is varied randomly or periodically. This creates a wavy, shimmering effect."
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: not supported *eJay Clubworld*: "chorus"

Table 3.13. Definition of the metaterm 'chorus'.

Function metaterm	'fattening'
Definition and background	Gibson (1997: 66) defines 'fattening' as delaying a sound less than c. 30 ms and mixing it with the original sound. "Fattening is the most effective of all delay times in making a thin or irritating sound fatter and fuller." (Ibid.)
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: supported (can be realized by duplicating a sample file to another pattern, but placing it slightly later than the original sound) *eJay Clubworld*: "chorus" (at certain settings, but applies to all sample files on a certain track)
Note	In *Magix Music Maker*, the "delay" feature cannot be used for 'fattening' because the available delay times are too long.

Table 3.14. Definition of the metaterm 'fattening'.

	‘distortion’
Definition and background	Bartlett and Bartlett (1998: 409) define ‘distortion’ as “[a]n unwanted change in the audio waveform, causing a raspy or gritty sound quality. The appearance of frequencies in a device’s output signal that were not in the input signal. (Distortion can be desirable - for an electric guitar, for example.)”
Music-making software features	*Magix Music Maker*: “distortion: soft”, “distortion: medium”, “distortion: hard” *MTV Music Generator 2*: supported (can be realized by increasing a sample file’s amplitude level excessively or by duplicating the same sample file to several patterns and tracks, and overlapping them) *eJay Clubworld*: supported (can be realized by increasing a sample file’s track level excessively or by duplicating the same sample file to several tracks and overlapping them)

Table 3.15. Definition of the metaterm ‘distortion’.

Function metaterm	‘fixed filter’
Definition and background	A filter, which settings does not change over time. According to Bartlett and Bartlett (1998: 412), a filter “[…] sharply attenuates frequencies above or below a certain frequency.”
Music-making software features	*Magix Music Maker*: “filter low”, “filter medium”, “filter high” *MTV Music Generator 2*: “low pass filter”, “band pass filter”, “high pass filter” *eJay Clubworld*: not supported
Note	In *eJay Clubworld,* the “EQ” feature cannot be considered similar to the ‘fixed filter’ function because equalization applies to all tracks.

Table 3.16. Definition of the metaterm ‘fixed filter’.

Function metaterm	‘dynamic filter’
Definition and background	A filter, which settings change dynamically (either manually in real-time or in non-real-time based on a preprogrammed configuration). A filter “[…] sharply attenuates frequencies above or below a certain frequency” (Bartlett & Bartlett 1998: 412). A ‘dynamic filter’ can also be used as a ‘fixed filter’.
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: “low pass filter”, “band pass filter”, “high pass filter” *eJay Clubworld*: not supported
Note	A common audible outcome of using a ‘dynamic filter’ is the filter sweep effect that originates from analog synthesizers.

Table 3.17. Definition of the metaterm ‘dynamic filter’.

3.6.4 Arranging

Metaterms are needed to describe seven functions used during the arranging phase. These are ‘pattern sequencer’, ‘cut/copy/paste/delete’, ‘tempo control’, ‘adjust sample file start/end point’, ‘pitch shift’, ‘time compression/expansion’, and ‘sample rate conversion’. These metaterms are defined in Tables 3.18 to 3.24.

Function metaterm	‘pattern sequencer’
Definition and background	‘Pattern sequencer’ is for making ‘patterns’, that is, sequences of a number of sample files played at various pitches. A ‘pattern’ consists of triggering commands: When to trigger what sample, at what pitch, and at what volume level. Commands such as these are defined with the ‘pattern sequencer’ through, e.g., graphical interfaces.
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: “riff editor” *eJay Clubworld*: “hyper generator”, “groove generator”

Table 3.18. Definition of the metaterm ‘pattern sequencer’.

Function metaterm	'cut/copy/paste/delete'
Definition and background	Duplicating, relocating, and deleting sample files and patterns in 'multitrack window'.
Music-making software features	*Magix Music Maker*: "copy", "delete" *MTV Music Generator 2*: "use riff" (copy), "cut riff", "delete riff", "clone riff" (copy-paste the selected riff), "replace" (paste), "replace all" (paste several times) *eJay Clubworld*: "select" (equivalent to cut and copy), "place" (paste), "drop" (delete)
Note	'Cut', 'copy', 'paste' and 'delete' are separate operations, but I have grouped them into one function for practical reasons.

Table 3.19. Definition of the metaterm 'cut/copy/paste/delete'.

Function metaterm	'tempo control'
Definition and background	Defining the tempo at which a piece of music is replayed. 'Tempo control' may be a programmable dynamic value or a manually defined fixed value.
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: "BPM", programmable *eJay Clubworld*: "BPM", manual (labeled "tempo" in user's guide)
Note	In *MTV Music Generator 2*, the music-maker defines the start tempo, end tempo, type of transition, and transition length. Based on these definitions, the tempo increases or decreases during playback.

Table 3.20. Definition of the metaterm 'tempo control'.

Function metaterm	'adjust sample file start/end point'
Definition and background	Defining the points from which the playback of a sample file begins and ends. It is one characteristic of nonlinear multitrack composing to use the same sample file, but different segments of it, in different parts of a piece of music.
Music-making software features	*Magix Music Maker*: supported *MTV Music Generator 2*: supported *eJay Clubworld*: supported (however, only the end point can be adjusted)

Table 3.21. Definition of the metaterm 'adjust sample file start/end point'.

Function metaterm	'pitch shift' (semitone steps, bend)
Definition and background	Transposing the pitch of a sample file in semitone steps or with a bend (resulting in portamento). The original duration of the sample file is either retained or not retained.
Music-making software features	*Magix Music Maker*: "pitchshift" (semitone steps) (original duration is retained) *MTV Music Generator 2*: supported (in "riff editor" in steps of one semitone or through "note pitch envelope" [in semitone steps; bend; original duration is not retained]). *eJay Clubworld*: supported (in semitone steps). In "hyper generator", certain sample files can be replayed at different pitches. The duration is the same for all the pitches of the same sample file.
Note	'Pitch shift' is the basic method of defining a melody with a single sample file.

Table 3.22. Definition of the metaterm 'pitch shift'.

Function metaterm	'time compression/expansion'
Definition and background	Changing the duration of a sample file without affecting its pitch (Roads & Strawn 1996a: 47).
Music-making software features	*Magix Music Maker*: "timestretch" *MTV Music Generator 2*: "time stretch" *eJay Clubworld*: not supported
Note	By 'time compression/expression', Roads and Strawn (ibid.) also refer to changing the pitch of a sample file without affecting its duration. I use the metaterm 'pitch shift' to refer to such a function. In practice, depending on the compression/expansion software algorithm, the timbre begins to change when the compression/expansion is applied beyond a certain range.

Table 3.23. Definition of the metaterm 'time compression/expansion'.

Function metaterm	'sample rate conversion'
Definition and background	Changing the sample rate of a sample file, including or excluding pitch shift (Roads & Strawn: 46-47). With pitch shift, 'sample rate conversion' also alters duration and timbre.
Music-making software features	*Magix Music Maker*: "resample" (in steps of one semitone). Changes pitch, duration, and alters the timbre. *MTV Music Generator 2*: "downgrade sample", "upgrade sample" between three sample rates (11, 22, and 44 KHz). Does not change pitch or duration or alter timbre. *eJay Clubworld*: not supported

Table 3.24. Definition of the metaterm 'sample rate conversion'.

3.6.5 Mixdown

Metaterms are needed to describe seven functions used in the mixdown phase. These are ‘track volume’, ‘pan’ (constant)/’panning’ (variable), ‘reverb’, ‘echo’, ‘master EQ’, ‘master compression’, and ‘master tune control’. These metaterms are defined in Tables 3.25 to 3.31.

Function metaterm	‘track volume’
Definition and background	Cary (1992: 521) defines ‘volume’ as “[t]he power level of an audio frequency complex wave in electrical form […]”. Cary (ibid.: 213) defines ‘gain’ as “[t]he amount by which a signal is magnified by an amplifier. […] When it affects loudness, a gain control is usually called a volume control.” Hence, ‘track volume’ involves controlling the power level of discrete sample files located on a track.
Music-making software features	*Magix Music Maker*: supported (not labeled in software, used in “mixer”; real-time manual) *MTV Music Generator 2*: “riff volume” (automatic variable) *eJay Clubworld*: supported (not labeled in software, used in “mixer”; real-time manual)
Note	‘Track volume’ can be of three types: Constant, manual variable, and automatic variable. Constant ‘track volume’ is defined with the volume slider of the mixer. Manual variable ‘track volume’ is adjusted in real-time with the volume sliders of the mixer. Automatic variable ‘track volume’ is defined in non-real-time with an envelope. In *MTV Music Generator 2*, the volume of each “riff” in a track can have its own volume settings. The volume is adjusted separately for each riff. There is no volume control for each track individually.

Table 3.25. Definition of the metaterm ‘track volume’.

Function metaterm	‘pan’ (constant)/’panning’ (variable)
Definition and background	Cary (1992: 352) defines ‘pan’ and ‘panning’ as follows: “From ‘panorama’, and a camera term adopted for audio use. The apparent horizontal movement of a sound source with respect to a stationary listener. The illusion cannot be maintained beyond about 90° with only two speakers, but complete 360° rotation is possible with three or more.”
Music-making software features	*Magix Music Maker*: “panorama” (term in user’s guide; the feature is not labeled in the software), static or dynamic manual *MTV Music Generator 2*: “note pan envelope” (static or dynamic automatic) *eJay Clubworld*: “pan” (static or dynamic manual)
Note	Can be applied to all sample files on a track or to a single sample file. ‘Pan’ (constant) is a fixed predefined setting that cannot be adjusted in real-time. ‘Panning’ (variable) involves moving a sample file from one loudspeaker to another during playback. ‘Panning’ (variable) can be defined with an envelope in non-real-time or be controlled manually in real-time.

Table 3.26. Definition of the metaterm ‘pan’ (constant)/’panning’ (variable).

Function metaterm	‘reverb’
Definition and background	‘Reverb’ is the electronically generated reverberant effect to a sound (Dobson 1992: 130).
Music-making software features	*Magix Music Maker*: “room”, “studio A”, “studio B”, “studio C”, “hall”, “pipe”, “space” *MTV Music Generator 2*: “room”, “studio A”, “studio B”, “studio C”, “hall”, “pipe”, “space” *eJay Clubworld*: “reverb”
Note	‘Reverberation’ is the acoustic phenomenon (ibid.: 130). Interestingly, the seven types of reverb have been labeled exactly the same in *Magix Music Maker* and *MTV Music Generator 2*.

Table 3.27. Definition of the metaterm ‘reverb’.

Function metaterm	‘echo’
Definition and background	Bartlett and Bartlett (1998: 410) define ‘echo’ as “[a] delayed repetition of a signal or sound. A sound delayed 50 milliseconds or more, combined with the original sound.”
Music-making software features	*Magix Music Maker*: “delay” and “echo” (in “effects”). As modifications to sample files: “echo slow”, “echo medium”, and “echo fast” *MTV Music Generator 2*: “delay” and “echo” (in “song effects”) *eJay Clubworld*: “delay” (with most “predelay” settings)
Note	Echo can be used to realize a rhythmical effect or a spatial effect by placing the repeated sound in a different pan position from the original. Certain types of echo are distinctive because of their use in popular music. One such type of echo is called ‘slapback’, that is “[a]n echo following the original sound by about 50 to 200 milliseconds, sometimes with multiple repetitions.” (Ibid.: 431.)

Table 3.28. Definition of the metaterm ‘echo’.

Function metaterm	‘master EQ’
Definition and background	‘EQ’ or ‘equalization’ is “[T]he adjustment of frequency response to alter the tonal balance or to attenuate unwanted frequencies.” (Bartlett & Bartlett 1998: 411.) The term ‘master EQ’ indicates that the same equalization is applied to all tracks.
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: not supported *eJay Clubworld*: “EQ”
Note	Equalization is applied during mixdown for adjusting the tonal balance of a piece of music. Bartlett and Bartlett (ibid.: 435) define ‘tonal balance’ as “[t]he balance or volume relationships among different regions of the frequency spectrum, such as bass, midbass, midrange, upper midrange, and highs.”

Table 3.29. Definition of the metaterm ‘master EQ’.

Function metaterm	‘master compression’
Definition and background	Compression is the reduction in dynamic range caused by a compressor device or software (Bartlett & Bartlett 1998: 406). Huber and Runstein define ‘compression’ as follows: “When the input signal exceeds a predetermined level […] called the *threshold*, the gain is reduced by the compressor and the signal is attenuated. By attenuating the louder signal levels, you are, in fact, reducing the program’s overall dynamic range. Because the range between the loudest and the softest signals is “compressed” by increasing the signal’s overall gain, the average (rms) level will be greater. Thus the signal will be perceived as being louder than it otherwise would be.” (Huber & Runstein 1997: 362.) By ‘master compression’ I mean that the same amount of compression is applied to all tracks.
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: not supported *eJay Clubworld*: “compressor”

Table 3.30. Definition of the metaterm ‘master compression’.

Function metaterm	‘master tune control’
Definition and background	Adjusting the overall tune with one control, that is, raising or lowering the pitch of the sample files on all tracks.
Music-making software features	*Magix Music Maker*: not supported *MTV Music Generator 2*: not supported *eJay Clubworld*: supported (combined with the “BPM” feature, that is, ‘tempo control’ function)

Table 3.31. Definition of the metaterm ‘master tune control’.

3.7 Comparing Features to Functions

The comparison addresses the following questions: How do the features of the three music-making software titles relate to each other, and through functions supported by other multitrack studio equipment, to the semiotic domain of multitrack composing?

I use the function metaterminology to compare the features of the three music-making software titles. I present the comparison in Table 3.32, which is organized on the basis of the five-phase nonlinear multitrack composing process. In the table, ‘x’ indicates that a feature is supported or can be realized, but has not been given a name in the software or in the user’s guide.

This comparison shows which functions are supported in one, two, or all three music-making software titles.

Phase	Function	*Magix Music Maker* features	*MTV Music Generator 2* features	*eJay Clubworld* features
Selecting	'sample file'	"sound", "loop", "object", "building block"	"sample", "note"	"loops" and "one shots", "sounds", "sound samples"
	'pattern'	--	"riff"	"sequences", "grooves"
	'sample file library'	"instrument directory"	"sample library", "riff library"	"sample archive"
Sampling	'user sampling'	--	"USB sampler"	--
Modifying	'truncation'	x	"trim"	--
	'normalizing'	--	"normalise"	--
	'adjusting sample file volume'	"volume x2", "volume x4"	"volume -5%", "volume +5%"	--
	'reverse'	"reverse"	"reverse"	--
	'chorus'	--	--	"chorus"
	'fattening'	--	x	"chorus"
	'distortion'	"distortion: soft", "distortion: medium", "distortion: hard"	x	x
	'fixed filter'	"filter low", "filter medium", "filter high"	"low pass filter", "band pass filter", "high pass filter"	--
	'dynamic filter'	--	"low pass filter", "band pass filter", "high pass filter"	--
Arranging	'pattern sequencer'	--	"riff editor"	"hyper generator", "groove generator"
	'cut/copy/paste/delete'	"copy", "delete"	"use riff", "cut riff", "delete riff", "clone riff", "replace", "replace all"	"select", "place", "drop"
	'tempo control'	--	"BPM"	"BPM"
	'adjust sample file start/end point'	x (start and end points)	x (start and end points)	x (only end point can be adjusted)
	'pitch shift' (semitone steps, bend)	"pitchshift" (semitone steps; original duration is retained)	x (semitone steps, bend; original duration is not retained)	x (semitone steps)
	'time compression/expansion'	"timestretch"	"time stretch"	--
	'sample rate conversion'	"resample" (pitch and duration changes)	"downgrade sample", "upgrade sample"	--
Mixdown	'track volume'	x (real-time, manual)	"riff volume" (automatic variable)	x (real-time, manual)
	'pan' (constant)/ 'panning' (variable)	"panorama"	"note pan envelope"	"pan"
	'reverb'	"room", "studio A", "studio B", "studio C", "hall", "pipe", "space"	"room", "studio A", "studio B", "studio C", "hall", "pipe", "space"	"reverb"
	'echo'	"delay", "echo" (in "mixer"), "echo slow", "echo medium", "echo fast" (modifications to sample file)	"delay", "echo"	"delay"
	'master EQ'	--	--	"EQ"
	'master compression'	--	--	"compressor"
	'master tune control'	--	--	x (combined with tempo control)

Table 3.32. Overview of functions supported in music-making software and the corresponding features.

As Table 3.32 indicates, the developers of the three music-making software titles have used varying terminology to label the features. Thus, a function metaterminology was necessary to enable making comparisons between the three music-making software titles.

MTV Music Generator 2 supports most functions, 23 out of 27. The unsupported functions are ‘chorus’, ‘master EQ’, ‘master compression’, and ‘master tune control’. The absence of these functions limits the music-makers’ options to adjust the overall sound simultaneously, meaning that that the overall sound must be adjusted by modifying each sample file individually. The absence of the ‘chorus’ effect can be worked around by duplicating a sample file to an available track and slightly offsetting it in pitch and time to thicken the resulting sound.

eJay Clubworld supports 18 out of 27 functions. The unsupported functions include ‘user sampling’, ‘truncation’, ‘normalizing’, ‘adjusting sample file volume’, ‘reverse’, ‘fixed filter’, ‘dynamic filter’, ‘time compression/expansion’, and ‘sample rate conversion’. The lack of user sampling means that the music-maker is confined to making music with the sample files available in the sample file library. The sample file modifying functions are more limited than in the two other music-making software titles because the functions ‘truncation’, ‘reverse’, ‘fixed filter’, and ‘dynamic filter’ are not supported. “Hyper generator” and “groove generator” are limited as pattern sequencers compared to the “riff editor” in *MTV Music Generator 2*. It is not possible to alter the pitch of the sample files except for the set of sample files that is available in “hyper generator”.

Magix Music Maker supports 16 out of 27 functions. The unsupported functions are ‘pattern’, ‘user sampling’, ‘normalizing’, ‘chorus’, ‘fattening’, ‘dynamic filter’, ‘pattern sequencer’, ‘tempo control’, ‘master EQ’, ‘master compression’, and ‘master tune control’. The lack of user sampling means that the music-maker has to settle for the 3,000 samples in the sample file library. Because there is no support to ‘pattern sequencer’ function, the programming options are limited. As with *MTV Music Generator 2*, the lack of an equalizer and a compressor limits the options to adjust the overall sound with one control setting. This means that the overall sound has to be adjusted by modifying each sample file individually. The songs are played at a fixed tempo because there is no way to control the tempo.

Most of the features of the three music-making software titles operate in non-real-time. Some features, in particular in *eJay Clubworld*, also respond in real-time.

The principal features of the three music-making software titles’ graphical user interfaces are summarized in Table 3.33.

	Magix Music Maker	*MTV Music Generator 2*	*eJay Clubworld*
Type of virtual space (Wolf 2001c)	Adjacent spaces displayed one at a time (with the exception of the "mixer" window). Horizontal and vertical scroll within multitrack window.	Adjacent spaces displayed one at a time. Horizontal and vertical scroll within multitrack window.	Adjacent spaces displayed one at a time. Horizontal and vertical scroll within multitrack window.
Multitrack window	"arranger"	"track window"	"arranger"
Representation of time	Horizontal, left to right	Vertical, top to bottom	Horizontal, left to right
Representation of sample file or pattern in multitrack window	Block with one of four colors (possibly suggesting which sample files sound well together) and a waveform (not representing the actual waveform of the sample file).	Block with user-definable color (about four color variations of the four track type colors). No approximation of the contents of a pattern is shown.	Block with color representing sample file category. The waveform of a sample file is animated during playback.

Table 3.33. Comparison of the main characteristics of the GUIs of the three music-making software titles.

The graphical user interfaces and virtual spaces of the three music-making software titles differ from each other. In other words, the means for nonlinear multitrack composing are represented in different ways. A minor similarity is that the virtual spaces of all three software titles are of a similar type: They are divided into adjacent spaces accessed one at a time. One interesting peculiarity is that the progression of time is represented in the multitrack window and pattern sequencer of *MTV Music Generator 2* vertically instead of the conventional horizontal way used, for instance, in musical notation and in the other two music-making software titles. *Magix Music Maker* appears to use numbers in sample file names and the color of the icons representing the sample files to suggest which sample files sound good together. Such coding is not used in the other two music-making software titles.

In the next section, I define the multitrack composing techniques that can be realized with the features of the three music-making software titles and present some examples of what can be done with them.

3.8 Analysis Part 1 Stage 3: Defining Nonlinear Multitrack Composing Techniques

By linking features to functions, I demonstrated in the previous section how music-making software titles are compatible with multitrack studio equipment. The features of music-making software are only potential means for making music. The purpose of the current section is to show what nonlinear multitrack composing techniques can be realized by using what music-making software features. This will conclude the discussion on the semiotic domain of multitrack composing.

In Section 3.8, I define each nonlinear multitrack composing technique. I adapt the definitions from tutorials including Roads et al. (1996), and academic work like Goodwin (1990 [1988]; 1991; 1997), Rose (1994), Frith (1996), Taylor (2001), Théberge (1997; 2001; 2003; 2004), Lysloff (2003), and others. I define six nonlinear multitrack composing techniques that can be realized with music-making software. These are 'sampling', 'sound design', 'programming', 'mixing', 'sound spatialization', and 'remixing'. Each technique has a number of different approaches and applications. For instance, there are several applications of sampling. Throughout Section 3.8, I will describe some applications of each technique.

Nonlinear multitrack composing techniques enable artists to suggest certain concepts, actions, and meanings (like 'theft') when making music. The composer Chris Cutler (2000: 97) argues that a sound 'plundered' from a sound recording allures people to use it because of all the associations to the sound. Some artists may deliberately include meanings like these in their music. It is an entirely different matter, which I will not discuss in this study, if and how other people perceive these meanings from the music.

Although I discuss the six techniques separately, they are closely related. For instance, during the phase of mixdown, aspects of the techniques of 'sound spatialization' and 'sound design' must also be considered by the music-maker.

Although most of the features of music-making software are of the nonlinear type (they operate in non-real-time), the techniques that can be realized with them are nevertheless comparable to the techniques that can be realized with multitrack studio equipment with features of the real-time type (e.g., sound design, mixing, and sound spatialization). The audible outcomes may be similar, but not necessarily identical. There is no real-time technique equivalent to programming.

The definition of each technique includes a description of its background (including the equipment with which the technique was originally realized and the motivations for developing the equipment or technique), examples of applications of the technique, and examples of what the technique can be used for in the three music-making software titles. The examples are from different types of music, as I believe that concepts from one style of music can be adapted into other types of music. I also consider how the tools for realizing the technique have been remediated. I examine the similarities between music-making with nonlinear multitrack composing techniques and the use of other new media tools. I consider the following questions proposed by Manovich (2001: 8): "What are the ways in which new media relies on older cultural forms and languages, and what are the ways in which it breaks with them? […] How do conventions and techniques of old media […] operate in new media?”

Several books have been written about the contexts of multitrack studio technology and the music made with this technology, including Manning (1993), Gracyk (1996), Roads et al. (1996), Théberge (1997), Emmerson (2000), Taylor (2001), Pinch and Trocco (2002), and Lysloff and Gay (2003). I will not go to such a level of detail in defining the nonlinear multitrack composing techniques. Rather, I will confine myself to a level that serves the purposes of this study. The purpose of defining the nonlinear multitrack composing techniques is to understand what they can be used for, and why they are important in contemporary music-making.

In defining the nonlinear multitrack composing techniques, the angle of approach is how multitrack studio technology and music-making software gives individuals means to make music. I do not consider how multitrack studio technology has changed the role of musicians in bands or the social relationships in studio work.

In association with defining each nonlinear multitrack composing technique, I present in a table which music-making software features can be used to realize the technique. The technique of remixing is an exception, because it can be realized potentially with all features. I have included the associated functions in the tables in order to enable future comparisons of how other multitrack equipment relates to features and techniques, even if such comparisons are beyond the scope of this study.

3.8.1 Sampling

Sampling is the digital recording and playback of sound. As a nonlinear multitrack composing technique, sampling is different from digital multitrack recording. In my definition, the latter involves recording longer takes, whose pitch and duration are not changed. In contrast, sample files are short digital sound recordings that may be looped or replayed at different pitches. Hence, the nonlinear multitrack composing technique of sampling can be realized with a sampler, but not with a digital multitrack recorder. In music utilities for personal computers, digital multitrack recording and sampling overlap. Using prefabricated material from sample file libraries has always been a part of using samplers. However, in my definition, ‘sampling’ involves recording the samples. Thus, sampling includes selecting the material to sample.

The idea of recording a sound and replaying it at different pitches has been implemented long before the introduction of digital sampling. The first attempts at this include experiments made with optical film sound tracks in the 1930s and *musique concrète* developed by the composer Pierre Schaeffer and his associates in the 1940s. One early device that is in some ways similar to present-day samplers is one the two Phonogène machines constructed for Schaeffer in the early 1950s. The device made it possible to replay pre-recorded tape loops at different speeds controlled by a twelve-note keyboard with a switch that provided twenty-four pitch transpositions (Manning 1993: 27.) The idea of making music by replaying recorded sounds was implemented in all versions of the Chamberlin and Mellotron instruments developed in the 1950-70s. However, these tape-based instruments in themselves did not enable to record sounds.[35]

Several corporations and researchers worked in the 1970s to develop digital sound recording and playback. According to Kim Ryrie, a designer of the Fairlight CMI, one of the motivations for developing digital sampling for the Fairlight system was analog synthesizers' inability to produce natural sounds (Vail 1993b: 190, 194). Inventions that made digital sampling possible include the developments in the computer industry, like semiconductor memory chips (increased capacities at lower prices) and rapid-access mass storage media (disk drives and hard disk drives). For instance, the introduction of 16K RAM memory chips in the 1970s made digital sound recording possible in practice. An early version of the Fairlight system featured 4K RAM memory chips that did not have enough capacity for storing digitized sounds, but could store only simple synthesized waveforms. (Vail 1993b: 192-194.) Qasar M8, a prototype of what became the Fairlight system, took about two hours to boot from a paper teletype machine (ibid.: 192). Obviously, rapidly accessible mass storage media for saving and retrieving samples was necessary to make sampling viable. The Fairlight and the other samplers introduced in the early 1980s were equipped with same types of disk drives as the computers of that era. Hard disk drives were brought to samplers as of the mid-1980s. Integrated circuits are used in samplers as well as computers. In other words, advances in the computer industry were exploited in sampler development (and also in the development of other music technology, see Théberge 1997: 58-71).

Sampling was introduced to musicians in the early 1980s with the release of instruments such as Fairlight, Synclavier II, and Emulator. The first samplers were expensive pieces of equipment. Sampling became more widely available in the mid-80s with the introduction of affordable samplers such as Ensoniq Mirage and Akai S612. As early as in 1986, Casio introduced a sampling feature to certain models of their low-cost home keyboard range (Colbeck 1988: 93). Thus, within a couple of years, sampling had become accessible to almost anyone interested.

The remediation of sampling from dedicated hardware to computer software began in the late 1980s, when affordable sampler units were introduced for the Commodore Amiga computer system. Sampling was possible on computers with an inexpensive accessory because computers had the required memory and storage capabilities by default. Tracker software was developed to make music from sampled material (see Lysloff 2003). With the exception of the expensive Fairlight and Synclavier II systems, hardware samplers had very small displays for viewing and editing the sampled material. Computer software introduced the visual editing and manipulation of sample files. This was partly possible because computers already featured operating systems with a GUI. Dedicated hardware samplers could record for short times only because they did not feature direct-to-disk

[35] Roads (1996a: 119-120) discusses photoelectric and tape-loop devices that were based on the playback of recorded sounds. The musician Frank Samagaio (2002) has written a history of Chamberlin and Mellotron instruments. According to Samagaio (ibid.: 6), there existed in the mid-60s a prototype of a Mellotron Mark II that enabled recording to some of the magnetic tapes inside the instrument.

recording, whereas computers with hard disk drives can record up to hours of sound in one take. On computers, the distinction between sampling and recording longer takes is blurred.

There are several applications of the nonlinear multitrack composing technique of sampling. Next, I reiterate the definitions of sampling by the media scholar Andrew Goodwin (1990 [1988]) and Cutler (2000). I expand these definitions with arguments from other scholars.

Goodwin (1990 [1988]: 270) defines three strands of sampling as 'realist', 'modernist', and 'postmodernist'. Goodwin notes that these correspond roughly to the categories of realism, modernism and postmodernism. These categories are appropriate to describe music-makers' attitudes towards sampling. Obviously, all three approaches to sampling can be used while making a piece of music.

According to Goodwin (ibid.), 'realist sampling' is

> […] the "hidden" sampling involved in using a machine such as a Linn drum to reproduce "real" drum sounds, or in the process of using a Fairlight or Synclavier to steal a sound. This use of sampled sounds is motivated largely by economics rather than aesthetics — getting "good" sounds and the "right" performance from a machine is cheaper and easier than hiring musicians. In this kind of sampling the object is transparency since the producer is using the technology to achieve a realist effect (the imitation of a "real" performance) without calling attention to the mediating role of production technology. And this use of sampling is indeed so pervasive that we no longer notice it.

Théberge (2001: 15) argues in a similar vein when noting that samplers were "[…] designed to reproduce the sounds of conventional musical instruments, thereby making studio production more economical by eliminating the need for backing musicians." This type of imitative use of samplers, essentially 'realist sampling', appears common in contemporary music-making. This type of sampling enables composers to use in their music sounds that previously required the contribution of studio musicians. However, it is doubtful if the imitative use was the primary motivation for developing samplers, or if the motivation was something else, for example obtaining more complex waveforms for sound synthesis material in order to design new kinds of sounds.

Goodwin (1990 [1988]: 270) argues that realist sampling is so pervasive that people are likely not to pay any attention to it. Théberge (2001: 15) concurs that many applications of sampling are imitative. As a nonlinear multitrack composing technique, realist or imitative sampling is transparent and thus its use cannot be heard on sound recordings if the sampling is of high fidelity.

Many recent sound synthesis methods are based on sampling, even if the recording of samples is not necessarily done by the music-makers themselves. Possibly several hundred CDs and DVDs with sample files have been released over the years. These contain vast amounts of material for immediate use in multitrack composing. Recently, sound libraries have been made available on the Internet.

Goodwin (1990 [1988]: 270) defines 'modernist sampling' as self-conscious exposure of the craft of sampling, that can be a celebration of playfulness. According to Goodwin, modernist sampling is more explicit than realist sampling. Through modernist sampling, “[s]ome producers have created records and remixes that celebrate playfulness, sometimes through a kind of baroque overindulgence.” (Ibid.) For these producers, modernist sampling is sometimes the “[…] self-conscious exposure of their own craft.” (Ibid.)

As a nonlinear multitrack composing technique, the approach of modernist sampling requires that the source of the sampled material is recognized by the audience either as exactly where it is from or that the sampled fragments originate from other music. This can be achieved, for instance, by adding crackle from a vinyl record to an otherwise hi-fi sample, recording the sample in low fidelity, or stuttering the sample.

According to Goodwin (ibid.: 271), ‘postmodernist sampling’ applies to artists who make a politics out of stealing. In this strand of sampling, taking fragments from existing sound recordings is a part of the meanings of the new piece of music. Taylor (2001: 157) considers Goodwin’s postmodernist sampling as a practice where “[…] samples are juxtaposed sounds in a web of texts that deny any evaluation of them, leaving the listener only a patchwork of fragmentary references that render impossible any definitive hearing or interpretation.”

With ‘stealing’, Goodwin refers in part to the unlicensed use of samples from copyrighted sound recordings. This was a much publicized issue during the latter half of the 1980s and the early 1990s. Unlicensed sampling was defined as ‘theft’ by a judge in a ruling to a lawsuit in 1991 (Théberge 2004: 146). Such stealing is discouraged because releasing sound recordings with unlicensed samples can lead to costly lawsuits and possible demands to withdraw the sound recording from the market.[36]

In practice, it can be difficult for the audience (and music scholars) to distinguish between modernist and postmodernist sampling if the intentions of the artist are unknown.

Cutler’s (2000) concept ‘plunderphonics’ is an alternative definition of different applications of sampling. The term ‘plunderphonic’ originates from the title of a 12” EP by John Oswald published in 1988 (ibid.: 88; see also Oswald 2004). Plunderphonics is a musical practice concerned with appropriating sound recordings of other people’s music and using these as raw material for making one’s own music. Cutler (ibid.: 89-90) argues that this musical practice undermines originality because it deals with copies, it speaks with the voices of others, and breaching copyright is a condition for its existence.

Cutler defines five applications of plunderphonics. In the application that Cutler calls ‘there it is’, materials are derived directly from sound recordings and are manipulated. In the plunderphonics application of ‘partial importations’, samples from, e.g., ethnic music, sound effects, and whale song are used as important material and the rest of the music is constructed around them. The application of ‘total importation’ involves re-hearing existing records that become simultaneously the subject and object of a new work. In the application of ‘sources irrelevant’, recognizing the plundered material is not necessary because samples are recorded for the purpose of obtaining raw material. This application is similar to Goodwin’s realist sampling, in other words, it can be a way to save time and

[36] For discussions on music and copyright form different viewpoints, see Frith and Marshall (2004).

money. In the application of ‘sources untraceable’, the sampled material is transformed so profoundly that their sources can no longer be recognized. Alternatively, this application of plunderphonics suggests that there is no need to originate. (Ibid.: 107-109.)

Theft must be recognized by the audience, because otherwise it is not theft from the viewpoint of the audience but unacknowledged appropriation. Theoretically, sampling is transparent so there is no audible difference between the source material and the sample recorded from it. If an artist wishes to make plundering obvious, he may record the sample in low-fidelity, add some crackle or noise to it, filter it (perhaps to suggest that the sample is from a low-fidelity radio or television broadcast), or to stutter the sample in the piece of music.

Waters (2000: 61) suggests that sampling is not necessarily restricted to music, because there are parallels to sampling in other media such as video art and cinema. Cutler (ibid.: 109) points out that the re-seeing and re-hearing of familiar material is a well-established artistic practice. For instance, montage, collage, and bricolage have been employed in the visual arts since the beginning of the twentieth century. According to Cutler (ibid.: 101-102), digital sampling was introduced into music in the early 1980s, into a climate where plundering had already been established in the form of scratching.

As a nonlinear multitrack composing technique, sampling (and especially the strand of modernist sampling) involves considerations of the associations that the sampled fragment can evoke. Gracyk (1996: 97) writes: “Framed as questions about ownership, legal fights about money can mask underlying battles about whose meanings are going to be left in the audience’s mind.” In other words, Gracyk argues that the issues around sampling are in part about the possibilities of recontextualizing the meanings of sounds and the problem of who has the power to control the meanings of a sound or a fragment of a piece of music.

Théberge argues that the aesthetics of sampling involve original uses of the sampled material. These uses include manipulation, transformation, and combination with other sounds, which may be sampled, generated on synthesizers, or performed and recorded by the sampling artist. In other words, making music through sampling is not simply about copying. (Théberge 2004: 148-149.)

The issue of who owns a sound may restrict music-making. As Cutler (2000: 110) points out, the fear of legal action may discourage artists from realizing certain ideas. Thus, copyright laws may act as a form of self-censorship restricting musical expression. Théberge (2004: 154) proposes that a new balance must be found between economic interests and the right to make creative use of sampling. Théberge (ibid.) argues that the record industry and legislators must re-examine copyright law for its outmoded notions of originality, creativity, and ownership.

Music made through sampling can reflect the musical taste and record collection of the artist. To name an example, the rap music producer Eric Sadler explains that he makes music using a collection of 20,000 records as the main source (Rose 1994: 89). The references that can be perceived in a piece of music may be a part of the artist’s expression. I will elaborate this issue in association with the ‘remixing’ technique.

Rose (ibid.: 79) points out that it can take more time to look for material to sample than to program similar musical material with a drum machine or sequencer:

> Sampling, as it is used by many of hip hop's premiere producers, is not merely a shortcut used to "copy" musical passages. If this were so, then producers would spare the legal costs of clearing usage of other artists' recorded material by programming or replaying very similar musical sequences.

Looking for material to sample involves, for instance, considerations of special sound qualities[37], which may not be possible to achieve with a drum machine or synthesizer. Rose (ibid.: 89) argues:

> Sampling in rap is a process of cultural literacy and intertextual reference. Sampled guitar and bass lines from soul and funk precursors are often recognizable or have familiar resonances. Some samples are taken from recent charted songs, making them eminently recognizable. […] In addition to the musical layering and engineering strategies involved in these soul resurrections, these samples are highlighted, functioning as a challenge to know these sounds, to make connections between the lyrical and musical texts. It affirms black musical history and locates these "past" sounds in the "present".

Cutler (2000: 101, 106) suggests that making music by sampling can be interpreted as critical consumption, and involve originality in hearing. Some segments of a piece of music may be so fabulous that they deserve to be sampled, while other fragments may be so silly they have to be sampled. Some fragments are sampled because they "fit" the piece of music that is being made. Most samples are critically evaluated by sampling artists before being placed into new pieces of music. One aspect of sampling as a nonlinear multitrack composing technique is to look for material to sample. Listening to old records or obscure television programs may lead to finding interesting material to sample. Hip hop artists and deejays call hunting for LPs with rare and unusual material as "digging in the graves" (Katz 2004: 11). Seeking for vinyl records with special or even "strange"[38] music from, for example, flea markets, can be considered a part of music-making.

Sampling enabled artists to add every kind of sound to their music, including sound effects. Sometimes these special sounds are used in a rhythmic way so that they are more than mere sound effects. Sound effects were already used in pop music recordings in the 1950s. Perhaps their purpose in some cases was to be "establishing shots" (as in movies) or "sound coulisses" suggesting the location in which the story told in the lyrics of the song

---

[37] The musicologist Mark Katz (2004: 140) notes that sampling captures the 'sonic aura' surrounding a sound, including "[t]he reverberation that imparts a sense of space, and the slight but constant ambient noise – a patina, perhaps – that is a by-product of imperfect recording fidelity."

[38] A series of books and audio CDs with the generic title *Incredibly Strange Music* have been released. The books were edited by V. Vale and Andrea Juno. A similarly themed book and series of audio CDs is *Songs in the Key of Z: the Curious Universe of Outsider Music*, with the book authored by Irwin Chusid.

is set, or to suggest certain moods. This application of sampling can be considered another aspect of 'realist sampling'. However, sound effects used this way are often highly stylized rather than documentary recordings (this is also the case with the sound effects in movies and television). This is an aspect of sound design that I will discuss later.

As shown in Table 3.34, of the three music-making software titles, sampling is possible only in *MTV Music Generator 2* with the optional USB sampler accessory. It is possible to sample a maximum of about 23 seconds of sound in one take. All applications of sampling discussed earlier can be realized with *MTV Music Generator 2*.

Music-making software feature	**Function**	**Realizable technique**
Magix Music Maker: none	'user sampling'	'sampling'
MTV Music Generator 2: "USB sampler"		
eJay Clubworld: none		

Table 3.34. The feature and function for realizing the nonlinear multitrack composing technique of sampling.

3.8.2 Sound Design

The term 'sound design' can be understood as a verb and as a noun. The definition of 'sound design' as a nonlinear multitrack composing technique should include both these aspects.

As a noun, 'sound design' is what characterizes a piece of music as an entity, that is, the audible outcome of the sound design activity (Winkler 1998: 221). The characteristic sound design of a piece of music may be used by a musician as a model for his own sound design.

As a verb, 'sound design' refers to selecting suitable sounds when browsing a sample file library or searching for material to sample. Sound design also involves modifying the selected material in various ways in order to achieve the desired audible outcomes. With music-making software, sample files must be used as starting points, because there is no support sound synthesis.[39]

In a sense, sound design is a practice as old as the building of musical instruments. Generating sounds and controlling their characteristics with electricity became possible in the late nineteenth century. What could be labeled as an additive sound synthesis method was first used in the Telharmonium constructed at the end of the nineteenth century, and since the 1930s in Hammond organs, for instance. Synthesizers using subtractive sound

[39] I avoid using the term 'timbre', because it is conventionally used to describe the tone color of an acoustic instrument. A sampler and most synthesizers do not have a timbre in the sense of a distinguishable range of tone color that can be produced on a certain acoustic instrument. Nevertheless, a sample file can reproduce the timbre of an acoustic instrument. When that sample file is modified, the timbre becomes altered.

synthesis methods became common during the 1960s and early 1970s with the introduction of synthesizer systems by Moog, Buchla, EMS, ARP, and other manufacturers. Computers have been used to control sound synthesis since the 1960s. Computer software enables more precise control over the parameters of sound compared than what a performer can control in real-time with a couple of controllers. (Manning 1993.) In the sense of controlling the parameters of sound with computers in non-real-time, sound design is connected to the nonlinear multitrack composing technique of programming.

Many new types of synthesizers have introduced new methods of sound design. Modular analog systems of the 1960s afforded musicians the possibilities of sound design through subtractive synthesis. In the 1980s, digital FM synthesis on synthesizers such as Yamaha DX7 introduced a different approach to sound design.[40] Sound design with music-making software is similar to the type of synthesizers that allow samples to be recorded and modified. However, sound design with a synthesizer featuring knobs, sliders, and buttons as a physical interface to the parameters of a sound is very different from sound design with music-making software. Sound design with music-making software is more conceptual than tactile.

The computer music scholar Julius Smith (quoted in Chadabe 1997: 252) suggests that with software, there are no limits to how a musician can change an instrument:

> You are not restricted by the real world any more, by the physical construction, not by physics. You can do anything that you conceive of. It's a little like movie cartoons in the 1950s, where you can stretch a character's neck to fourteen feet, or you can hit a guitar over someone's head and have it wrap around their head fourteen times and still be ready to play. It's whatever your imagination wants.

In other words, Smith suggests that sound design is above all conceptual. New sounds are imagined by the music-maker and are then realized with a more or less physical approach depending on the interface of the instrument used. Alternatively, sound design can be approached by using existing sounds as a starting point, and modifying these with the interface of the instrument. The physical interface for using the three music-making software titles, the DualShock 2 controller, does not enable a versatile tactile approach to sound design.

Grimshaw suggests that the interest in the manipulation and creation of sounds is in part symptomatic of the compositional desperation of the serious music of the twentieth century. In popular music, the repetitive musical conservatism and formulaic composing methods provided the space and necessity for exploring sounds. (Grimshaw 1998: 129.) People making music for radio and television advertisements were also interested in new sounds. Since the 1950s (if not even earlier), science fiction movies and television shows required new kinds of sound effects. Sound design with synthesizers was a response to these needs.

---

[40] For discussions on various sound synthesis methods, see Roads et al. (1996); Miranda (2000 [1998]).

Gracyk argues that rock musicians became “painterly” when they took on sound recording technology as their primary means of making music. He suggests that these musicians are like painters, for whom color is the essence of art, not subordinate to line and drawing. For these musicians sounds and their colors are essential properties of a musical work. (Gracyk 1996: 66.) The same attitude can be identified in other music in addition to rock, for instance electronic music.

When synthesizers, samplers, or music-making software are used to make music, certain sounds *must* be selected from the set of prefabricated sounds, or they must be designed by the music-maker. In other words, sound design is a necessity in nonlinear multitrack composing.

Moog suggests that “[m]usic is the art of shaping and controlling sound. […] musical sound is merely the carrier of an artistic message, not the message itself. The artist determines the musical message by the way he or she shapes and articulates the sounds […].” (Moog 1984 [1978]: 41.) Shaping sounds during real-time performance is a part of the musical expression (if not the literate or suggestive “artistic message” that Moog proposes) of many artists. Much of the same can be done in non-real-time with music-making software. The nonlinear multitrack composing technique of sound design involves music-makers in defining and controlling all aspects of sound in very precise detail.

The special characteristics of pieces of studio equipment can also be used in sound design. In Section 2.4, I noted that few pieces of multitrack studio equipment are truly transparent, bur “color” in more or less subtle way the audio signals that pass through them. Old and even malfunctioning pieces of equipment can transform sounds in interesting ways and thus be used in sound design. For example, the techno artist Derrick May (quoted in Trask & Barr 1996: 52) has explained his working methods with seemingly transparent devices like samplers and cassette recorders as follows:

> At the moment I’m using cheap samplers like the Mirage and Akai S612 […] Sampling things, then recording them onto cassette, sampling them again, recording them onto cassette again, sampling them again and so on. The idea is to pick up noise, to create some sort of different feel to the music.

In other words, May uses the low-fidelity characteristics of these devices to modify sounds. According to Reynolds (1998: 100), the members of the techno music project LFO worked in much the same way: “[they] would create a bass sound, record it into cassette with the recording levels right up in the red zone, sample that deliberately distorted sound, and repeat the process: all in the search for the heaviest, hurtful-est bass sound.” Engineers have tried to keep distortion to a minimum in most studio equipment, but musicians have found distortion useful, for example, in sound design. Sample files can be distorted in all three music-making software titles.

Sound design has parallels to tendencies in other media. Manovich argues that all electronic media technologies are based on modifying a signal by passing it through various filters. The shift from material object to signal accomplished by electronic technologies was a fundamental conceptual step towards new media. When media material is in the form of a signal, it can be modified in real time through filters and other components. Most equipment for electronic media production, transmission and reception includes controls for modifying the signal. (Manovich 2001: 132-133.) In other words, a common characteristic of nonlinear multitrack composing and new media is that both involve modifying existing material.

Manovich (ibid.: 126) suggests that the music synthesizers of the 1960s were the first instruments that embodied the logic of new media: Selection from a menu of choices. Building sounds with an analog synthesizer is close to this because musicians begin with a simple waveform, and then shape this signal by selecting filters, amplifiers, and other components modifying the signal. Much of the same applies to the digital signal processing technologies of today: The audio signal is first digitized and then it is transformed numerically through algorithms. The graphical user interface to these software algorithms may look like the user interface to analog synthesizer modules.

Sound design is also linked to present-day trends of music consumption. Frith (1996: 243) notes that popular music directs the listener's attention to sound, which is perceived instantly. Within a couple of seconds, the audience can perceive if a piece of music is, for example, punk, techno, or country (unless the piece of music blends distinct styles). In popular music, the perception of sound may be more important than considering a composition as an entity, that is, as a collection of themes and variations that make sense in the framework of the entire piece of music (ibid.).

When replayed from sound recordings, a piece of music may not be heard from the beginning. Thus, it could be argued that in popular music any part of a song should somehow catch the attention of the audience. One way to attempt to grab attention is through a catchy sound design. In contemporary music production, a lot of attention is paid to the shaping of the sounds, to make them somehow special or novel.

The musician Robert Ashley suggests that on pop records, people can hear expensive tailored sounds, which several artists may have worked on: "[w]hen you listen to pop music, the best pop music, you hear very expensive sounds, like designer sounds, Gucci sounds." (Ashley quoted in Chadabe 1997: 259.) This also means that sounds can become objects of economic exchange (see Théberge 1997: 75-83).

Théberge (1997: 203) notes that "[c]ontemporary music-making demands that any musical sound be as available as any other; technological reproduction guarantees availability […] musicians (and audiences) in the West have attained an unprecedented, technically mediated access to the musical sounds and traditions of the world […]." Théberge considers sound libraries one aspect of the commodification of musical sounds. In the 1980s, a small industry emerged to provide sounds for digital synthesizers and in the following decade for samplers. Developing extensive sample file libraries requires great effort and financial investments. Thus, sound library and synthesizer manufacturers have become dependent on outside developers specialized in certain musical styles. Many musicians, producers, and sound engineers have contributed to sample libraries. (Théberge 1997: 83; 2003: 95-96.)

Sounds can function as hooks in a piece of music (Théberge 1997: 195). Popular music scholar Roy Shuker (2002: 168) defines 'hook' as follows: "The melodic or rhythmic pattern that is catchy and 'hooks' or attracts the listener to want to listen to the rest of the song, and, more importantly, want to hear repeated playings." Obviously, the perception of what is a 'hook' is highly subjective, as becomes apparent by Shuker's use of the vague term 'catchy' in his definition. Nevertheless, the way a sound has been designed can make it a hook in a piece of music.

The composer Aaron Marks (2001: 27) considers the sound design of sound effects similar to making a piece of music, albeit on a smaller scale:

> Consider the creation of a sound effect to be like a mini musical composition. You first choose the sounds, which will work best to reach your end goal. You cut, paste, layer, EQ, pan, fade, add some effects processing, do a final mix, and voilà, instant sound effect. You build a musical piece the same way. Well, it's practically the same—except the choice of instrumentation is slightly different and instead of the final mix being minutes long, it's seconds long.

Marks' suggestion can be interpreted as applying to sound design in general. In pieces of music where there are few overlapping sounds, the audience can better perceive them and thus pay attention to the sounds if they are interesting in themselves.

Certain sounds are associated with specific historical eras of recorded music. Lysloff (2003: 47) notes that "[…] certain musical instruments may remain in the collective imagination only as sounds, not as physical objects. This is already beginning to happen with particular instruments, such as the Hammond B3 organ or the Moog synthesizer." Of course, there is no single sound in a modular Moog synthesizer or even an electric organ like Hammond B3. It is more accurate to say that these instruments remain in the collective imagination as certain *ranges* of sounds. For example, an analogue Moog synthesizer has a different range of sounds from a digital Yamaha DX100 synthesizer.

Théberge (1997: 120) argues that sounds of vintage electronic instruments can give musicians a form of direct sonic and in some cases iconic access to music of the past. Something similar applies to the so-called world music samples (Théberge 2003). According to Théberge, one discourse of world music sampling suggests that the sounds of a musical instrument embody the musical culture the instrument is associated with. One discourse of the selling arguments for these sound libraries suggests that samples carry with them primal powers that can be transferred to the music-maker who uses the samples in their music. In turn, these primal powers can be used to suggest certain effects to the audiences. (Ibid.: 99-100.)

Some music-makers may attempt to eliminate the referential properties of sounds. Others may exploit the senses of identity certain sounds can impart. For instance, some pop music "[…] has retained the sense of identity that sounds carry with them – their ability to act as a kind of referent for the object that is the source of the sound – thus leading to an aesthetic fundamentally based in collage and ironic juxtaposition." (Théberge 1997: 204.)

Taylor (2001: 152) notes that in hip hop music, sampling has been a way to do homage to musical forebears, or to establish a kind of musical community when artists sample material from music they like. Taylor (ibid.: 153) suggests that "[…] almost all hip hop sampling practices are involved with making meanings, meanings that make sense to the musicians and to their fans."

Taylor (ibid.: 139-140) hypothesizes that music as social activity may become a thing of the past for many musicians:

> It may well be […] that one of the reasons that sampling has itself become a kind of art form is that it provides aural glimpses of the social, metonymized into fragments of acoustic musicking, but in their new contexts of electronically generated music, these glimpses are historical.

When sounds give iconic access to different times and locations, musicians can suggest specific eras and places with certain sounds. Thus, when they make music with those sounds, musicians in their imagination may relive or "visit" those eras and places. Musicians may use samples because of the meanings they suggest, or they may consider samples as raw material that is incidental to the piece of music (ibid.: 154). Lysloff (2003: 47) points out that sounds may as well be for some musicians just "[…] free-floating sonic symbols without a material referent."

When music is heavily based on samples, some sounds may soon become "worn out" in the ears of the audience. For instance, in the mod scene, "[…] novice composers are often ridiculed for using samples that had become hackneyed as a result of their popularity and, thus, are encouraged to come with unique sounds (or at least less widely heard samples) for their music." (Ibid.: 49.) A widely recognized sound may be rendered fresh by modifying it in some way.

Outside the domain of music, the nonlinear multitrack composing technique of sound design has parallels in the making of sound effects for movie soundtracks, radio plays, and user interface sounds to computer software and other electronic equipment. Sound design has led to the situation that natural sounds may in some situations appear small and in effect "unnatural" when compared to the designed sounds. Aho (2005: 229) argues that as sounds are built by digital means through modifying and mixing several sounds, they render the original sounds "unnatural". People know from movies how, for instance, a gunshot "should" sound, and compared to this, an actual gunshot may sound tiny and "unnatural". Thus, Aho suggests that the unnatural has become natural. (Ibid.) The unnatural sound is amplified and modified, in other words, it is the designed sound. Because of hearing designed sounds in the media, people have become more perceptive of everyday natural sounds (Truax 2001: 152-153). Sound effects have become commodities, as plenty of sound effect libraries are available.

In music-making software, sound design is based on modifying sample files because there is no support for sound synthesis. Sampling is required to obtain the material for sound design. In *Magix Music Maker* and *eJay Clubworld*, which do not enable user sampling, the sample files have been provided by the developers of the software titles. As is shown in Table 3.35, several functions can be used to realize sound design with music-making software. As a nonlinear multitrack composing technique, sound design takes place especially during the phase of modifying sample files.

Music-making software features	Functions	Realizable technique
Magix Music Maker: support of ‘truncation’ function, “volume x2”, “volume x4”, “reverse”, “distortion: soft”, “distortion: medium”, “distortion: hard”, “filter low”, “filter medium”, “filter high”	‘truncation’, ‘normalizing’, ‘adjusting sample file volume’, ‘reverse’, ‘chorus’, ‘fattening’, ‘distortion’, ‘fixed filter’, ‘dynamic filter’	‘sound design’
MTV Music Generator 2: “USB sampler”, “trim”, “normalise”, “volume -5%”, “volume +5%”, “reverse”, support of ‘distortion’ function, support of ‘fattening’ function, “low pass filter”, “band pass filter”, “high pass filter”		
eJay Clubworld: “chorus”, support of ‘distortion’ function, “EQ”		

Table 3.35. Features and functions for realizing the nonlinear multitrack composing technique of sound design.

Sound design in a multitrack studio involves modifying samples, using various sound synthesis methods, and processing the recorded material with effects. Sound design with music-making software is limited to modifying sample files. The nonlinear multitrack composing technique of sound design can be realized with various features of music-making software, including reverse, distortion, filter, chorus, and so forth.

The sound design options with music-making software appear limited when compared to software synthesizers on PCs. *MTV Music Generator 2* is the most sophisticated of the three music-making software titles in sound design options, especially because it supports sampling (choosing the raw material for sound design). In other music-making software titles, the material is limited to what is available in the sample file library. Sound design enables music-makers to use the sounds they want, instead of being content with prefabricated sounds.

Besides shaping sample files, sound design also involves considering the overall sonic characteristics of a piece of music. Thus, sound design is linked to shaping the tonal balance which is a part of the nonlinear multitrack composing technique of ‘mixing’ that I will discuss in Section 3.8.4.

3.8.3 Programming

The nonlinear multitrack composing technique of ‘programming’ involves making a piece of music in non-real-time by defining patterns, assembling patterns into compositions, defining loops, adjusting sample file durations, and so forth. Thus, programming takes place mainly in the phase of arranging in the nonlinear multitrack composing process.

Programming is about making music by working directly with the sounds that make up a piece of music. As Lysloff notes when discussing music-making with trackers: “Instead of writing *for* certain musical instruments, as in past Western musical practice, mod composers create music *with* instruments. In other words, mod composers work directly with sound rather than writing instructions (i.e., musical notation) […].” (Lysloff

2003: 45.) People make music with trackers by defining instructions for the software to carry out, including instructions to replay what sample file at what sample rate and volume level from what channel.

In a sense, realizing all nonlinear nonlinear multitrack composing techniques could be considered programming, because these techniques involve defining and modifying in non-real-time all aspects of a piece of music. Defining the sounds on synthesizers and samplers is sometimes called ‘programming’. However, I interpret such activities as sound design rather than programming. My definition of ‘programming’ also excludes the activity of deejays selecting records for their sets.

If programming is interpreted as the activity of defining in non-real-time the properties of a piece of music, the first manifestations of musical programming were defining the notes of a piece of music with various mechanical musical automata. Early examples of such devices include programmable mechanic carillons constructed by Dutch instrument builders in the thirteenth century. The music these machines played was defined one note at a time. Roads (1996d: 663) suggests that today we would call this type of defining music step mode recording. Player pianos became common in the nineteenth century. Roads (ibid.: 664-666) discusses experiments conducted from the 1930s to the 1950s to control electric oscillators with punch cards and paper tape strips.

Roads (ibid.: 667-668) suggests that the requirements of integral serial composition techniques motivated the development of early electronic sequencers. Integral serial patterns can be realized by using a sequencer featuring multiple parallel “tracks” that are programmed individually. Roads (ibid.: 668) notes that the logic inherent in the technology of analog sequencers also led to repetitive, minimalist styles of music.

Buchla (quoted in Vail 1993a: 99), who in the early 1960s developed one of the first models of analog sequencers with 8 and 16 steps, recollects his motives as follows:

> The sequencer […] had its basis actually in tape music […] as a tape replacement. I thought of the sequencer as a way of rapidly sequencing a series of predetermined pitches and avoiding a number of tape splices. The traditional way at the time of assembling a tone row was to take an inch and a quarter of *A*, followed by three-quarters of an inch of *B*, and so on, and then splice all your pitches together. It was a very tedious way of making a tape.

In other words, Buchla developed the sequencer to make certain stages of music-making (by the means of 1960s tape music) less tedious. Buchla’s motives are similar to those of Roger Linn for developing the programmable LM-1 drum machine in the late 1970s. Linn (quoted in Vail 1993c: 252) contemplated his motives for beginning to develop the drum machine as follows:

> I had a little four-track recorder and some gear in one room of my house where I was making demos. Of course, the hardest thing to record is drums in your home studio. […] I wanted a drum machine that did more than play preset samba patterns and didn’t sound like crickets.

In other words, Linn set out to develop a drum machine that sounded realistic compared to the simple rhythm devices of the era, enabled more complex rhythm patterns than presets, and removed the need for hiring a drummer and for recording drum kits in home studios. Buchla and Linn wanted to make certain aspects of music-making easier with their inventions.

Programming emerged as a nonlinear multitrack composing technique even before personal computers became common as control devices in multitrack studios. Early microprocessor controlled sequencers, such as Roland MC-8 introduced in 1977, were programmed by defining notes one-by-one in non-real-time (i.e. in step mode). On sequencers and drum machines, ‘programming’ refers to defining the notes of a song with a numeric keyboard, typically by recording a couple of measures in step mode or real-time, manipulating this data (e.g., quantizing and transposing musical notes), copying and pasting notes, chaining measures into parts and parts into songs. In addition, tempo, data from controllers like pitch bend wheels, and patch (sound program) changes are programmed into sequencers in step mode or real-time.

When developing his first drum machine, Linn defined a method of programming rhythms through patterns and chains, and this method was later adapted into sequencers. Linn also came up with the concept of defining slight irregularities into a beat that was otherwise very rigid. Years later, defining irregularities in the timing, loudness, and pitch of musical notes was labeled “humanize” in some sequencer software (and as “humanise” *MTV Music Generator 2*).

According to Roads and Strawn, experiments with computer sound synthesis were started in 1957 by Max V. Matthews and other researchers working at the Bell Telephone Labs. From early on, the computer’s capability was recognized to be in generating and controlling any frequency rapidly and precisely. From these experiments evolved the *Music* series programming languages. For instance, in 1958, *Music II* language made it possible to program four independent voices with sixteen selectable waveforms. (Roads & Strawn 1996b: 87-90.) These first synthesizers were programmed, in other words, the music that was to be generated was defined in non-real-time.

Since the 1980s, MIDI has made it possible to connect synthesizers, samplers, drum machines and effect devices to computers and hence place them under computer control. Since 1983 when the MIDI specification was made public, most multitrack studio equipment has been equipped with this interface. MIDI was included as a standard feature in Atari ST series of computers that was first introduced around 1985. MIDI adapters were available for most computer models. Initially computers were used to record, manipulate, and playback MIDI data. Recording and manipulating digital audio alongside MIDI data became common In the 1990s.

The communication scholar Steve Jones (1992: 206) suggested already in the early 1990s that “[…] the image of the sound recordist is becoming like that of the computer hacker.” In other words, Jones suggests how others perceive sound recordists like sound engineers and musicians using sequencers and computers, not necessarily how they see their activities themselves. Nevertheless, programming music in non-real-time with an alphanumeric keyboard does resemble computer programming to some degree (Lysloff

2003: 33). ‘Hacking’ also refers to making something not so elegantly or by-the-book, possibly using equipment in ways not envisioned by the manufacturers (see Section 2.4).

Negus (1992: 85) describes ‘programming’ as follows:

> Much of what is referred to as ‘programming’ involves the use of existing software packages and is done outside of the studio during pre-production work. A programmer may then be on hand for the first few days of recording or called in later to make any necessary modifications. Some of the most respected programmers are keyboard players who use programming as an extension of their existing musical ability and arranging skills.

In other words, programming is another way (in addition to real-time performance) for musicians to practice their abilities.

Théberge (1997: 244) argues that with the introduction of MIDI sequencer software, a rationalized and calculated form of control over sound production emerged. Programming can give a distanced view of music-making, because it moves the emphasis in music-making from the tactile to the conceptual (see Aho 2005). This means that the music-maker is physically less involved than when performing music with an instrument in real-time. When a piece of music is represented visually on an electronic screen, the music-maker can examine it at various levels, ranging from the entire song at one glimpse to the parameters of a single sound filling the entire screen. The parameters of the song can be modified one at a time, and the audible outcome may be listened to repeatedly. Every change made to the song can be analyzed in detail by looking at the data and by listening to it.

For example, a drummer conceptualizes music-making in a certain way, because he has a tactile and direct relationship to his instruments. A music-maker who uses samples of drum patterns to make music conceptualizes music in a different way because his working material includes arranging digital sound recordings, and not utilizing drum sticks and drums as physical objects. Thus, the programmer’s sense of music may be more abstract in nature. (Théberge 1997: 3-4.)

Goodwin (1991: 88) notes that multitrack recording tied musical performances to a strict pulse, for instance the click track recorded on the multitrack master tape. With sequencers and music software, tempo can be programmed to vary. Goodwin (ibid.: 90-91) suggests that machines such as sequencers and programmable drum machines reverse the tendency towards rhythmic one-dimensionality that was a seen as an effect of earlier multitrack recording.

Because programming takes place in non-real-time, it makes possible music-making activities for people who do not have musical performance skills. For instance, programming drum machines enabled musicians who were not drummers to compose drum patterns. Lysloff (2003: 47) notes: “A composition (or any part of it) can be immediately “played back” by the computer (using a tracker software program) so that the composer can listen to any passage throughout the compositional process.” This approach “[…] provides more possibilities for “untrained” composers to experiment with musical sound and structure.” (Ibid.)

Goodwin (1997: 125) suggests that using a drum machine, "[…] may encourage the programmer to avoid the tried and tested conventions that the body unthinkingly repeats." Goodwin (ibid.) presents an example on this:

> The famous drum break that interrupts New Order's "Blue Monday" is a good example of a fill that few drummers would have considered trying to play, not only because it would be technically quite demanding, but because it is just very slightly removed from what one's hands and feet would normally do with a drum set.

In other words, programming helps musicians to overcome their physical limitations in real-time performance. Thus, the result is different drum patterns, which are not necessarily unauthentic or better, but just different from what a drummer would perform in real-time. Goodwin (ibid.: 126) also notes that programming drum machines has established its own routines and norms.

Programming has introduced new elements to appreciate and enjoy in music. For instance, a continuous exact timing in drums, bass, and ostinato patterns can be very appealing, for instance in dance music.

Digital compositing is an aspect of making music with music-making software that has parallels with all other media material making with new media tools. Manovich (2001: 139) writes:

> As a general operation, compositing is a counterpart of selection. Since a typical new media object is put together from elements that come from different sources, these elements need to be coordinated and adjusted to fit together. Although the logic of these two operations—selection and compositing—may suggest that they always follow one another (first selection, then compositing), in practice their relationship is more interactive. Once an object is partially assembled, new elements may need to be added; existing elements may need to be reworked. This interactivity is made possible by the modular organization of a new media object on different scales. Throughout the production process, elements retain their separate identities and, therefore, can be easily modified, substituted, or deleted.

Manovich's notion of the interaction between 'selection' and 'compositing' operations closely resembles the nonlinear multitrack composing process. 'Selection' is replaced by selecting sample files, and arranging and mixing substitute 'compositing'.

When a new media object is completed, it may be distributed as a single file, in which the separate elements are no longer discretely accessible. The completed object may also retain a modular structure even when it is distributed. (Ibid.: 139.) Music-making software titles store pieces of music in formats that retain their modular structure.

Until music software was available to consumers, multitrack recordings were mixed for distribution and consumption to a stereo recording, where the different elements can no longer be separated. Indeed, some elements may be have been left out of the mix or be inaudible because of acoustic masking from other elements. With music software and component file formats, it becomes technically possible for consumers to make their own mixes and, for instance, bring to the fore elements that were originally omitted.

The nonlinear multitrack composing technique of programming can be realized with several music-making software features. These are shown in Table 3.36.

Music-making software features	Functions	Realizable technique
Magix Music Maker: "copy", "delete", support of 'adjust sample file start/end point' function, "pitchshift", "timestretch", "resample"	'pattern sequencer', 'cut/copy/paste/delete', 'tempo control', 'adjust sample file start/end point', 'pitch shift', 'time compression/expansion', 'sample rate conversion', 'master tune control'	'programming'
MTV Music Generator 2: "riff editor", "use riff", "cut riff", "delete riff", "clone riff", "replace", "replace all", "BPM", support of 'adjust sample file start/end point' function, support of 'pitch shift' function, "time stretch", "downgrade sample", "upgrade sample"		
eJay Clubworld: "hyper generator", "groove generator", "select", "place", "drop", "BPM", support of 'adjust sample file start/end point' function, support of 'pitch shift' function, support of 'master tune control' function		

Table 3.36. Features and functions for realizing the nonlinear multitrack composing technique of programming.

In music-making software, the nonlinear multitrack composing technique of programming includes defining the playback of sample files at certain pitches and at certain volume levels. Programming also includes defining loops, and the start and end points of sample files. In addition, defining the tempo changes in piece of music is a part of programming.

In contrast to linear tape-based multitrack recording, in music-making software editing is not applied to the compound downmixed material, but through the multitrack window straight into the structure of the piece of music on the level of the sample files or patterns. Single sample files can be moved later or earlier in the time frame of the song. This would not be possible when editing compound material.

In music-making software programming, replaces such earlier techniques as recording, overdubbing, and punch-in, which were essential in a tape-based multitrack recording studio. Because of programming, no real-time performances need to be recorded, and with the exception of *MTV Music Generator 2*, cannot be recorded.

Note-by-note programming has been a possibility – but not a necessity – with MIDI sequencer software since the 1980s. In music-making software, programming is not an option but the single method to make music, even if there are limited real-time modes for entering notes in *MTV Music Generator 2* and *eJay Clubworld*.

Music-making software titles are programmed through graphical representations of sample files, their pitches and durations, rather than alphanumeric representations of notes as in some MIDI sequencer software. Music-making software titles automatically mix the overlapping sample files, and the mixing balance is defined by the music-maker.

In music-making software, programming and other nonlinear multitrack composing techniques overlap. For example, applying a fade-out envelope to a sample file in *Magix Music Maker* simultaneously involves the techniques of programming, mixing, and sound spatialization.

3.8.4 Mixing

Bartlett and Bartlett (1998: 419) define the verb 'mix' as combining two or more discrete signals into one signal. Mixing is a necessity in multitrack composing because the separate elements recorded on discrete tracks or as separate digital sound files must be combined in some balance to form a composite signal that is transduced into an audible form with the audio system available (mono, stereo, or surround). As a noun, a 'mix' is one possible combination of the potential of the material on the multitrack recording.

As a nonlinear multitrack composing technique, 'mixing' involves adjusting the balance (volume and pan position) between each of the discrete elements (sample files and patterns) that make up the piece of music. This balance may be altered several times during the course of a piece of music. Mixing also involves applying effects such as compression, equalization, reverb, and echo in order to obtain the desired audible outcome.[41]

Mixing also includes defining the tonal balance of a piece of music. Bartlett and Bartlett (ibid.: 435) define 'tonal balance' as the volume relationships between different regions of the frequency spectrum.

Mixing can be done in real-time or in non-real-time. Mixing in non-real-time involves defining the balance (and the changes to the balance) between the elements that make up a piece of music. Mixing in real-time is about manually adjusting the balance as the piece of music progresses. One way to build a piece of music through real-time mixing is to play in loop a set of multitrack material and make evolving music from these by continuously adjusting the balance between the elements, gradually adding effects such as filtering and echo.

Mixing was necessary if several microphones were used in a recording session. Thus, the first mixing desks were built long before multitrack recorders. Mixers and microphones were introduced into recording studios in the 1920s along with electric sound recording methods. According to the musicologist Susan Schmidt Horning, American studios in the 1950s were equipped with mixing consoles with round volume-control knobs. Because the knobs were rather large, a single person could control only two channels at a time. Mixing desks equipped with horizontally positioned faders enabled one person to simultaneously control the volume levels of several channels. This way, the person controlling the sound mixer could assume a more active role in the recording process. (Schmidt Horning 2002 [2000]: 139.)

---

[41] A 'mix' can also refer to the real-time selection and blending of records by deejays (Fikentscher 2003: 305). This is outside my definition of 'mix'.

The development of mixing desks is connected to the development of multitrack recorders. The more tracks there were on the multitrack recorder, the more channels were required on mixing desks.

The opportunities to shape the material on each track increased with the introduction of mixing desks with equalizers and limiters for each individual channel. Previously there could have been a single equalizer and limiter in use for the entire mixing desk.

Before the development of mix automation, a complicated mixing session could involve several people working at the mixing console. With mix automation, a single person can perform a complicated mix in a number of steps by defining the settings of a few tracks at a time. Mixers with fader automation (for recalling switch settings and fader positions) were introduced in the early 1970s. (Roads 1996b: 367, 378-379.) Automation meant that mixing desks were subjected to microprocessor or computer control. With automation, the configuration of a particular mix can be stored and recalled later. The mixing settings can be defined in great detail, in a way that is not so far from step mode recording and editing with sequencers.

Remediation of the tools for mixing has occurred so that in several music utilities for PCs, the sliders and potentiometers of a mixing desk are visually represented on the screen. Each slider and potentiometer is accessed one at a time with the mouse or some other controller. Some music utilities support special external control devices with sliders, knobs, and buttons.

When listening to sound recordings made with the means of multitrack composing, people have assumed an understanding of what counts as a “proper” mixing balance. Let us call this a ‘conventionalized’ mixing balance. This implies in many cases that the main element (like vocals or lead instrument) is placed at the front, and everything else (“the accompaniment”) in the background is spread broadly in the field between the left and right channel (Tiits 1997: 38). If the mixing balance is substantially different from what is conventional, people may consider the music to be “badly” mixed (Frith 1996: 230).

As a nonlinear multitrack composing technique, mixing includes shaping the tonal balance of a piece of music, for instance through equalization. Mixing is linked to sound design because of the effort to define the tonal balance of a piece of music.

Dynamic compression is common during mixing. A dynamically compressed sound is one characteristic of multitrack recording (Théberge 2001: 10; for a discussion about compression as a phenomenon of acoustic communication see Truax 2001: 145-147, 185-191). Schafer (1994: 95) describes compressed sound as a sound wall, that is, a sound that does not breathe. However, a sound wall may also be something some musicians strive for. Theoretically, there are fewer dynamic nuances in a compressed sound or the dynamic contrasts are more accentuated. However, in a compressed sound recording the nuances of a sound may be more audible. Since the 1970s, a compressed sound has been an audible sign of a professionally produced sound recording. Of the three music-making software titles, the ‘master compression’ function is supported only in *eJay Clubworld*.[42]

In music-making software, a mix is never fixed because the music is kept in a component format. The balance of the elements that make up a piece of music can be readjusted time after time because there is rarely a single appropriate configuration for a mix. There are just a number of different combinations for how to blend the elements. Obviously, the changes that can be made to a song with music-making software go beyond mixing. This suggests that in nonlinear multitrack composing there need not be a definitive version of a piece of music. The nonlinear multitrack composing technique of mixing proposes to music-makers that there can be multiple versions of a piece of music, and each of them may be customized for a specific context of consumption. This brings music closer to the practices of live performance and improvisation instead of the concept of having fixed artworks.

Roads (1996b: 359-362) considers mixing in non-real-time with software flexible and precise, but with drawbacks such as the need for elaborate planning, and the lack of “feel” because there are no faders to manipulate in real-time and thus respond immediately to what one hears.

The activities of arranging and mixdown with music-making software appear to the music-maker similar to what Manovich defines as ‘digital compositing’. Manovich (2001: 136-137) defines ‘digital compositing’ in graphics software as “[…] the process of combining a number of moving image sequences, and possibly stills, into a single sequence with the help of special compositing software […]”. The idea is quite similar in music-making software with overlapping sample files (as, e.g., in the GUI of *Magix Music Maker*) instead of overlapping image sequences (as, e.g., in software like *Premiere* for video editing). Furthermore, “[a]s a general operation, compositing is a counterpart of selection. Since a typical new media object is put together from elements that come from different sources, these elements need to be coordinated and adjusted to fit together.” (Ibid.: 139.) In music-making software, prefabricated sample files are selected from libraries. Mixing in music-making software and compositing in new media creation are parallel activities.

Mixing is more than a technical craft because it requires musical insight and judgment. Roads suggests that the person doing the mixing plays a role similar to the conductor in a concert hall. Both are responsible for the overall balance of the elements of a piece of music. (Roads 1996b: 385.) In addition to defining the balance of the elements, the person doing the mixing decides what elements from the multitrack recording may be completely left out.

The material recorded on a multitrack master recording can be mixed into a number of different combinations resulting in diverse audible outcomes. On the one hand, this is empowering as the material on the multitrack recording can be approached in many ways. On the other hand, this can pose a serious conceptual problem for the music-maker: What

[42] In my definition, ‘compression’ includes both the compression and the expansion of an audio signal. The devices are sometimes called “compressors” although they also expand the audio signal.

criteria should be used to determine which mix is most appropriate with regard to the material on the multitrack master?

As shown in Table 3.37, several music-making software features can be used to realize the nonlinear multitrack composing technique of mixing.

Music-making software features	Functions	Realizable technique
Magix Music Maker: support of ‘track volume’ function, “panorama”, “room”, “studio A”, “studio B”, “studio C”, “hall”, “pipe”, “space”, “delay”, “echo”, “echo slow”, “echo medium”, “echo fast”	‘track volume’, ‘pan’ (constant)/ ‘panning’ (variable), ‘reverb’, ‘echo’, ‘master EQ’, ‘master compression’	‘mixing’
MTV Music Generator 2: “riff volume”, “note pan envelope”, “room”, “studio A”, “studio B”, “studio C”, “hall”, “pipe”, “space”, “delay”, “echo”		
eJay Clubworld: support of ‘track volume’ function, “pan”, “reverb”, “delay”, “EQ”, “compressor”		

Table 3.37. Features and functions for realizing the nonlinear multitrack composing technique of mixing.

The GUI of *Magix Music Maker* enables looping and mixing in real-time. The sample files can also be modified in real-time during playback (but before the modified part is to be played). Real-time mixing is possible also in *eJay Clubworld*. In *MTV Music Generator 2*, mixing cannot be done in real-time.

Although mixing is also necessary in remixing, the latter is a distinct nonlinear multitrack composing technique, which I will discuss in Section 3.8.6. Mixing is also linked to the nonlinear multitrack composing technique of ‘sound spatialization’ because spatial effects such as reverb and echo may be defined during mixing.

3.8.5 Sound Spatialization

As a nonlinear multitrack composing technique, ‘sound spatialization’ is about the conscious use of space as an element in a piece of music. This is not unique to multitrack composing, as Frith (1996: 241) remarks: “[m]aking music has always involved not just the planning of notes on paper, but also the realization of the notes as sounds *in particular settings*, particular aural spaces.” Frith suggests that space was an element in music long before the era of sound recording. Music has always been performed in acoustic spaces and performers are in certain proximity to the audience. (Ibid.: 241-242.) The properties of acoustic spaces and the location of sound sources were largely ignored as parameters of composing until it became possible for composers to control them. Such control became possible for composers with electronic sound reproduction. For a brief discussion of examples of using space in music from the sixteenth century to the 1980s, see Roads (1996c: 452-454).

Sound spatialization is not exclusive to music, because, for instance, radio plays, and movie and television soundtracks also utilize sound spatialization.

I have adapted the term ‘sound spatialization’ from Roads (ibid.: 451). According to Roads, sound spatialization has a virtual and a physical aspect. In recording studios, composers spatialize sounds by imposing delays, filters, panning, and reverberation on them. Composers use these effects to create illusions of sounds emerging from imaginary environments (virtual acoustic spaces). In the physical world, sounds can be projected over multichannel sound systems from different positions. (Ibid.)

Roads (ibid.: 492) notes that

> […] the continued dominance of two-channel media for broadcast and sale make it difficult and impractical to realize more sophisticated spatial processing. A true multiple-channel audio medium distributed on a mass scale would greatly stimulate further advances in musical spatialization.

In my definition, ‘sound spatialization’ is also possible with two-channel (stereo) sound reproduction. This interpretation is partly based on Gibson (1997: 9-14), who defines three parameters in the space between two loudspeakers as left/right (panning), front/back (volume), and up/down (pitch; because low frequencies travel along the floor).

In recent years, multi-channel sound reproduction has become increasingly popular through home cinema systems. Some recordings of popular and classical music have been released in surround formats. Nevertheless, surround audio is currently used mainly in cinema. One practical consideration hindering surround systems from becoming common is the placing and wiring of six or more loudspeakers and the extra cost of the required equipment. There are also a couple of headphone models available that feature surround audio.

On sound recordings, the characteristics of acoustic spaces are perceived through reverberation. The longer the reverberation time, the larger the space appears and the more remote the sound source appears to be. The more distant the sound source is, the fewer high frequency components there are in the sound.

By applying reverberation effects to sound recordings, it is possible to emulate various virtual acoustic spaces. The term ‘virtual’ is justified here because these sounds have not been recorded in real-world acoustic spaces or have been generated with synthesizers.[43]

With electronic instruments, there is no need for acoustic spaces in which to record performances. Completely “dry” signals from the electronic instruments could be used. However, such “dry” sound recordings are quite uncommon. Applying effects like reverb

[43] In some cases, applying reverberation can be a way to modify sounds in the sense of sound design or arranging a piece of music. For instance, in some music by Kraftwerk, reverberation is used to accentuate a drum stroke or sustain a short sound. Examples of this can be heard from the different mixes of *The Robots* from 1991 and the live album *Minimum-Maximum* (2005).

and echo are deliberate choices by the music-makers. The properties of a specific acoustic space can be imposed on any sound recording. (Wishart 1996: 141.) With effects such as reverb and delay, the listener is given a new musical place, so that he experiences as being “in the music” (Frith 1996: 242).

The tools for realizing sound spatialization include loudspeakers, mixing desk, and certain effects devices, most obviously reverb and echo. Before electronic or electromechanic effects devices, reverberation and echo were realized acoustically in recording studios.

In sound reproduction, moving a sound in a space requires a controller with two or more axes, depending on the number of channels and loudspeakers available. On a regular mixing desk, one slider can control the position of a sound on the axis of distance (volume level) and another slider or knob on the horizontal axis. Additional sliders may control the amount and type of reverberation.

Effects devices producing virtual acoustic spaces have been remediated as software. In addition, the characteristics of old reverb devices have been remediated in many software titles. The paths along which sounds move can be defined graphically with software. Computers have removed the need for a physical mixing desk, but loudspeakers or headphones (and perhaps an amplifier) are required to realize sound spatialization with computers and software.

When discussing electronic music and computer music, Roads suggests that sound spatialization has become an aspect of composing: “The art of sound spatialization has assumed a similar position today as the art of orchestration had in the nineteenth century. To deploy space is to choreograph sound: positioning sources and animating movement.” (Roads 1996c: 451.) Of course, this requires that the audience has the equipment necessary to reproduce the sound spatialization as intended by the artist.

Roads (ibid.) argues that the movement of sound through space can create dramatic effects and serve as a structural element in composing: “The virtual and physical *sound stage* around the audience can be treated as a landscape, with its background and foreground, and fixed and moving sources.”

One approach for music-makers to attempt to de-emphasize sound spatialization is to mix the music on the basis of a conventional mixing balance, which I discussed earlier. Another approach is to mix to a balance that sounds like mono or with all elements at the same level.

One significant aspect of sound spatialization is that it extends the scope of music-making with the opportunity to design a piece of music as a scene of sound events occurring in a virtual space. The use of sound spatialization is one way of bringing music closer to the other audio arts (such as radio play), and natural and man-made soundscapes[44].

Movies, radio plays, and video games present soundscapes that have been constructed to convey certain information to the audience.

---

[44] Schafer (1994) has presented the hypothesis that people echo in music the soundscapes that surround them.

According to van Leuween, perspective places the elements of what is represented in a hierarchy by placing some elements in the foreground, some in the middle ground, and some in the background. Hierarchy means that some objects are treated as more important than others. Sound engineers in radio and film, and soundscape researchers divide the soundtrack into three zones. These zones have been called ‘close/middle/far’, ‘foreground/mid-ground/background’, ‘figure/ground/field’, or ‘immediate/support/background’. Sometimes there may be only two zones. What is, for example, figure, ground, or field depends on the listener’s relation to what is represented or on the way that relation has been created in sound spatialization. (van Leuween 1999: 14-18.)

Van Leuween argues that perspective can be used in music as well as in painting to depict landscapes. Musical perspective can also be used to represent aspects of social structures. In Western classical music, melody is the figure and accompaniment is the ground. Thus, the melody can represent an individual that stands out from the background of accompanying voices that represent the community around the individual. (Ibid.: 20-21.)

Perspective in a sound recording is realized by the relative loudness of overlapping sounds. Van Leuween argues that human perception divides overlapping sounds into two or three groups. These groups are placed at different distances from the listener to make the listener relate to them in different ways. The figure is treated as the most important sound, as something the listener is invited to identify with. Sounds placed in the ground are a part of the listener’s social world, but in a less involved way. A sound positioned in the field is treated as existing not in the listener’s social world, but in the listener’s physical world. Sounds with a lot of reverberation appear distant, but also immersive because they are difficult to pinpoint to a particular location. When there is no perspective, there is only figure. (Ibid.: 22-23, 28.)

This perspective is reversed, for instance, in some electronic dance music. The melody (the individual) is in the background and the accompaniment (the group) is in the foreground. (Ibid.: 21-22.) In other words, the perspective in electronic dance music suggests an emphasis on the group instead of the individual (see Tagg 1994).

In addition to placing the elements of a piece of music in a perspective, sound spatialization also involves moving sounds in the virtual acoustic space. Wishart has defined principles for using space as a compositional element. The parameters of the motion of sound (speed, change in speed, regularity or irregularity of pattern of movement) can be exploited in music-making. Wishart discusses six classes of motion: Constant, accelerating, decelerating, accelerating-decelerating, decelerating-accelerating, and irregular. For instance, the steady movement of sound may convey the idea of purposefulness. The movement of sounds acquire their meanings in relationships to the motion (or motionlessness) of other sounds. For instance, the movement of a sound in a piece of music can be interpreted as rapid if there are other sounds that move less rapidly. The movement of sounds may be spatially or temporally coordinated to suggest degrees of “consonance” and “dissonance” in motion. Sounds with a movement that slows down and finally stops can be interpreted to give the feeling of reaching a goal. (Wishart 1996: 203-235.)

In the case of spatialized drum sounds, Théberge (2001: 14) notes:

> The multitrack recording process allows for the sound of the drums and cymbals to be spatially separated in the stereo mix, thus creating an artificially enhanced, spatialisation of the rhythmic structure of the music itself […] The sound of the voice and other instruments and, ultimately, the listener are placed *within* this spatial/rhythmic field.

As Théberge suggests, sound spatialization can be a way to emphasize the rhythmic elements of a piece of music, for instance, by bouncing sounds between the left and right channel. Alternatively, the rhythmic potential of sound spatialization may be de-emphasized by keeping the elements in fixed positions in the virtual acoustic space.

Music-making software titles have plenty of features for realizing sound spatialization, as shown in Table 3.38.

Music-making software features	**Functions**	**Realizable technique**
Magix Music Maker: "panorama", "room", "studio A", "studio B", "studio C", "hall", "pipe", "space", "delay", "echo", "echo slow", "echo medium", "echo fast", "filter low", "filter medium", "filter high"	'track volume', 'pan' (constant), 'panning' (variable), 'reverb', 'echo', 'pitch shift' (bend), 'fixed filter', 'dynamic filter', 'fattening'	'sound spatialization'
MTV Music Generator 2: "riff volume", "note pan envelope", "room", "studio A", "studio B", "studio C", "hall", "pipe", "space", "delay", "echo", "low pass filter", "band pass filter", "high pass filter", support of 'fattening' function		
eJay Clubworld: support of 'track volume' function, "pan", "reverb", "delay", "chorus"		

Table 3.38. Features and functions for realizing the nonlinear multitrack composing technique of sound spatialization.

In music-making software, sound spatialization involves defining the volume and pan position of the song elements (sample files and patterns) and changing them during the course of a piece of music. The elements appear to move in the virtual acoustic space: Sounds move closer, away, and from left to right or vice versa. At the minimum, sound spatialization involves placing sample files in spatial relationships to each other. This cannot be avoided in stereo mixing.

'Track volume' and 'filtering' (low-pass filtering in particular) are functions that can be used to realize 'sound spatialization', because sounds emerging from sources that are far away are less loud and have fewer high frequency components than sounds emerging from sources that are closer to the listener. The function 'pitch shift' (bend) can be used in music-making software to realize an effect similar to Doppler shift, that is, conveying the illusion of a sound source moving towards and then away from the listener. This illusion can be enhanced by gradually changing the volume and high frequency cutoff of the sample file. The 'fattening' function can be used for sound spatialization because it makes a sound appear to occupy more space between the speakers (Gibson 1997: 66).

3.8.6 Remixing

As a nonlinear multitrack composing technique, 'remixing' is about using an existing piece of music as a starting point for new music-making activities. Remixing is a comprehensive nonlinear multitrack composing technique because it involves using potentially all features of multitrack studio equipment.

Théberge (2004: 148) notes that even before digital sampling, practices such as scratching challenged the idea that recorded songs were finished and fixed works. Since the 1970s, music released on vinyl discs has been used by deejays as raw material for new music. The practice of making and releasing special remixes of pieces of music emerged in the 70s in parallel with the introduction of the 12" maxi single. In the late 1970s and early 1980s, many deejays moved from clubs and radio broadcasting studios into dance music production, bringing with them their workplace concepts from clubs to recording studios (Fikentscher 2003: 305).

Grimshaw (1998) discusses three forms of remixing. Following the communication scholar Jonathan Tankel, Grimshaw (ibid.: 130) defines the first form of remixing as the 'reanimation' of familiar music. This involves, for instance, releasing different mixes of a piece of music that has recently dropped out of the charts and radio playlists. The original piece of music is the reason why these remixes exist. The makers of this type of remix rely on the original version being recognized by the audience.

Grimshaw (ibid.) defines the second form of remixing as 'nostalgia'. The remixed music is less recent than in the case of remixing as reanimation. The examples Grimshaw discusses of remixing as nostalgia are remixed medleys of songs originally released decades ago. These medleys are made by sampling fragments from the original versions or cover versions of these songs. The term 'nostalgia' refers to feelings of nostalgia on the part of the artists doing the remixes. Nostalgia remixes can also be done in an attempt to evoke feelings of bygone eras in the audience.

The third form of remixing provides its own justification. These remixes do not exploit familiarity because there is no original piece of music to reanimate or for which to evoke nostalgia. Each of these remixes is as original as the other. (Ibid.) The third form of remixing emerges from the fact that the material on the multitrack master recording can be mixed in a variety of ways. These remixes are primarily outcomes of the art and craft of remixers. The first and third forms of remixing may overlap because a number of remixes of a song may be produced and released at the same time. One of these remixes may become the "original version" (in the sense of a point of reference) for the audience.

A fourth type of remixing can be identified as ephemeral remixing. During the performance, a deejay may remix a piece of music in real-time. If the deejay's performance is not recorded, the remix can most likely never be repeated identically.

The fifth type of remixing includes making megamixes, that is, medleys of several songs, which may be more or less recent, perhaps suggesting a theme with the selection of certain pieces of music. In megamixes, the juxtapositions of the different pieces of music can be regarded as a special aspect of this type of remixing, which may, for instance, be ironical.

Making ‘mash-up’ recordings can be interpreted as the sixth type of remixing.[45] Making mash-up recordings involves layering two (or more) songs by different artists, and the result is a juxtaposition with often humorous and satirical intentions. One of the pieces of music may be an *a cappella* version. Additional musical material may be made by the music-maker doing the mash-up. These recordings tend to be technically unpolished and the making of them appears to have required little technical skill. Mash-up recordings transform the original recordings by recontextualizing them through juxtaposition. Thus, a reinterpretation becomes necessary for both pieces of music individually and of the mash-up. (Théberge 2004: 153-154.) In other words, making mash-up recordings involves challenging the meanings associated with “fixed” pieces of music. Because of the relatively short sampling times in *MTV Music Generator 2*, it is difficult to do extensive mash-up recordings with it. Making mash-up recordings is not possible in the two other music-making software titles.

As a nonlinear multitrack composing technique, remixing arises from two things. First, the material on a multitrack master recording can be mixed in a great variety of ways. Fikentscher (2003: 304) points out that the term ‘mix’ is synonymous with ‘version’. Second, there are different markets for different versions of a piece of music, for instance, a short version for radio play, an album version for home listening, and an extended dance mix for clubs. Remixing has artistic, innovative, and aesthetic merits, as well as economic reasons (ibid.: 306-307).

Remixing includes redefining the balance of the elements on the multitrack recording and possibly leaving out some material, modifying the material with signal processing methods like filtering, reshaping the form structure of the piece of music, and possibly making new musical material to the remix (ibid.: 305). Sound spatialization can be a part of remixing the elements of a piece of music, for instance through relocating them (on the horizontal axis and in the distance) in the virtual acoustic space.

The possibilities of remixing are more limited if the music-maker has access only to a compound mix of the music. From personal experience I argue that at least the following means of remixing compound material are possible. The music-maker can attempt to isolate certain parts such as vocals from the compound material by filtering it. The form structure of the original material can be rearranged. The pitch of the original compound material can be changed. The original material or a part of it can be reversed. Filtering can be used to make some parts of the original material less prominent. The music-maker can add (overdub) new material on top of the compound material, stutter the sample file in the style of scratching, or loop a fragment of the original material. The original material can be modified with effects like distortion, chorus, and dynamic filtering. A brief pause or “break” can be inserted into the original material. Sound spatialization is also a possibility, for example rhythmically panning the original material in the virtual acoustic space. The echo effect can also be used. All these actions (realizable with *MTV Music Generator 2*) can be taken to suggest that the original compound material has been remixed.

---

[45] Synonyms for mash-up include ‘bastard pop’, ‘blends’, and ‘cut-ups’. ‘Cut-ups’ have come to refer to pieces of music that rely on the humor or pathos of the reconstructed spoken word. A cut-up may remix political speeches to suggest satirical effects. (Wikipedia 2005a.)

All pieces of music that exist in written notation are interpreted when they are performed. There are socially imposed limitations to how much the interpretation can differ before it ceases to be “faithful” to that particular composition. There appear to be no such points of reference in remixing, except if there is an “original” mix that the audience knows well. A remix can be very different in sound design and form structure from the album version of that piece of music. Gracyk (1996: 28) suggests that a remix can be interpreted as a different piece of music when it has been given another name and the intention in remixing has been to produce a new piece of music.

The essential characteristic of a remix is that it sounds different from the original version (Frith 1996: 242). In other words, the audience can perceive by listening that a remix differs from the original version. A remix may also have a different form structure than the original version. Some remixes do not convey a sense of progression, because no parts can be perceived as intro, verse, bridge, chorus, and coda.

Théberge defines remixing as the reworking of pre-existing material to meet the needs of different consumer contexts for music. Thus, remixing fits within the economic interests of the record industry as it allows maximization of the potential profit to be made from a piece of music. (Théberge 2001: 10.) People making their own remixes is a new context of consumption of music. Music made or adapted for this purpose is such that sections of it can be sampled even in compound format (e.g., the mixes are such there are few overlapping instruments, and in those parts with overlapping sounds, the overlapping material can be filtered out). As I noted earlier, there are currently few component music file formats for delivering music to consumers in a remixable form. Remixing can also be done with material in compound formats, but the possibilities for separating overlapping sounds are limited.

Remixing can be a personalization method for a piece of music originally made by someone else. Remixing can be personalization in the sense that it involves defining a new combination (mix) of the elements of the original multitrack material, or adding new elements on top of it.

The dance music producer James Lumb considers remixing as “[…] a way of exploring your own musical history, and presenting it to someone else.” (Lumb quoted in Gerrish 2001: 39.) The dance music artist Derrick May presents a similar argument: ”Most of hip-hop is made by black artists. Most black artists are not imitating, they’re reliving their most gratifying moments, the most interesting times of their life. That’s why they sample and loop these classic old tracks from year and years ago - and also, they’re funky loops.” (May 2000: 127.) According to Rose (1994: 79), in hip hop music, sampling “[…] is about paying homage, an invocation of another’s voice to help you to say what you want to say. It is also a means of archival research, a process of musical and cultural archeology.” Of course, others than hip hop artists may through sampling relive whatever music they consider gratifying. For instance, in the mod scene remixing mod music is common and is generally “[…] considered as a form of homage to a particularly outstanding piece of music or a tribute to a renowned artist” (Lysloff 2003: 48).

Frith (1996: 237) argues that musical taste is “[…] intimately tied into personal identity; we express ourselves through our deployment of other people’s music.” Remixing can be interpreted as an extension of this phenomenon. In the sense suggested by Lumb, May, Rose, Lysloff, and Frith, sampling and remixing may in part be about making concrete the appreciation of an artist and expressing it to other people.

As a nonlinear multitrack composing technique, remixing involves recontextualizing musical material, sampling sonic signifiers from various media and making new combinations of them. It is fair to assume that people have associations and connotations regarding certain pieces of music. Remixing may challenge these associations and connotations.

The users of music-making software do not have access to multitrack master recordings the way professional remixers have. Nevertheless, music-makers can approximate remixing through sampling and programming similar patterns with sample files similar to those in the original versions. In practice, the distinction between remixing and making cover versions may be vague. In a remix, some fragments (often vocals) of the original sound recording are used, whereas a cover version consists entirely of new material.

3.9 Conclusions to Part 1

Nonlinear multitrack composing techniques offer new options in music-making. For instance, sampling is a straightforward way to incorporate any sound into a piece of music. This has not been possible in music before the era of sound recording. The options to place any sound in a piece of music and to use space as an element in music narrow the gap between music and soundscapes. Programming enables detailed defining and control of the properties of every element in a piece of music. Sound design enables music-makers to shape the sounds in a piece of music precisely the way they want them to sound.

Ten functions out of the total of twenty-seven reported in this study are supported by all three music-making software titles, namely ‘sample file’, ‘track volume’, ‘sample file library’, ‘distortion’, ‘cut/copy/paste/delete’, ‘adjust sample file start/end point’, ‘pitch shift’, ‘pan’ (constant)/’panning’ (variable), ‘reverb’, and ‘echo’. These make it possible to realize some aspects of all nonlinear multitrack composing techniques except sampling. The technique of sampling can be realized only in *MTV Music Generator 2*. The possibilities for remixing are limited in *Magix Music Maker* because it does not support sampling or enable programming patterns, meaning that the music-maker is restricted to using the sample files delivered with the software. *eJay Clubworld* is also limited for making remixes because only the sample files included in the software can be used, and the pattern programming options are restricted.

Manovich (2006 [2002]) has examined the transformation of the avant-garde techniques of the 1920s into the conventions of human-computer interface and notes that “[…] software does not simply adopt avant-garde techniques without changing them; on the contrary, these techniques are further developed, formalized in algorithms, codified in software, made more efficient and effective.” In other words, avant-garde techniques of the past are naturalized in software. Some of the avant-garde techniques of 1940s-60s electronic music are available in music-making software like *MTV Music Generator 2*. These techniques include, for example, editing sound recordings, filtering them, changing their speed, and reversing and looping them.

According to Manovich (ibid.), the avant-garde of new media is involved in "[…] accessing and using previously accumulated media in new ways." Manovich (ibid.) notes that the emphasis in new media has shifted from looking for new forms and representations to finding "[…] new ways to deal with the media records obtained by already existing media machines." Nonlinear multitrack composing techniques (especially remixing) are ways to deal with existing media records, in this case sound recordings.

Making music with nonlinear multitrack composing techniques has notable similarities to the making of other media material with new media tools. Manovich suggests that the deejay is a figure who prominently demonstrates the logic of new media: The selection and combination of pre-existing elements. Manovich argues that new art forms emerge from this logic. To name an example, the essence of a deejay's art in live electronic music is in his ability to mix selected elements in real-time in sophisticated ways. (Manovich 2001: 134-135.)

Manovich (ibid.: 142) argues that selection and compositing are the key operations of computer-based authorship. Some of the operations (actions) of making music with music-making software are selecting, and arranging and mixdown (rather than compositing). Selection in graphics software is similar to selecting sample files in music-making software. In the compositing of graphics, the artist determines the position and the order of the overlapping layers of graphical objects. This is similar to arranging sample files and patterns through their graphical representations. Compositing is also similar to mixdown, which involves defining the balance of the discrete sample files or patterns in volume level and spatial balance in a virtual acoustic space.

When making music with music-making software, the assessments on the choices of selecting sample files, modifying, arranging, and mixing them into a piece of music are based on listening. I suggested earlier that some of the knowledge about the semiotic domain of multitrack composing can be acquired by listening to sound recordings. In other words, listening is the main source of knowledge for making decisions when making music with nonlinear multitrack composing techniques.

The approaches to making music with music-making software can vary from the realization of a precise plan to listening to available sample files and making a piece of music based on the properties of these sounds. This continuum of approaches to music-making resembles that between bricolage and a scientist's activities, as defined by Taylor in a discussion of the critique leveled at *musique concrète*. According to Taylor, the bricoleur begins with the materials at hand and makes a structure out of them, whereas the scientist begins with a structure. The bricoleur works in signs, whereas the scientist works with concepts. Concepts are creative, and signs are re-creative. As I noted earlier, sample files can be sonic signifiers when converted into audible form. A scientist begins with an abstract concept and works towards its material realization. (Taylor 2001: 58-59.) This is not to say that there is anything inherently superior in the approach of a scientist.

The approaches are just different and in practice, making music with music-making software may involve aspects of both approaches, for instance, by starting with a plan, but making changes on the basis of the materials at hand.

Earlier I noted that remixing can be one way for an artist to show respect to the subject his admiration. As other people in addition to professional artists have access to multitrack studio equipment, could this argument be applied to all fans showing appreciation by making remixes?

The activities of fans as makers of media texts such as fan fiction, filk songs, and music videos has been observed by researchers including the media scholar Henry Jenkins (1992a; 1992b). Fans lack direct access to the means of mass mediated cultural production and they have limited influence on the decisions made by the industry producing media texts (Jenkins 1992a: 26).[46] According to Jenkins (ibid.: 33), the relationship between the readers and writers (i.e. makers and producers) of media texts is an ongoing struggle for the control over the meanings of the texts. For example, in filking themes from media texts are attached to melodies from folk or popular music with an awareness of the meanings that may be evoked through the juxtaposition of these two (Jenkins 1992b: 216). Filk songs and fan fiction are ways for fans to interpret media texts and comment on them. These activities are ways to open fans' favorite texts to new interests and interpretations, thereby suggesting new meanings. (Ibid.: 231-232.) Writing fan fiction and filking are also ways of extending the contents of media texts or commenting on them (ibid.: 216).

Earlier I suggested that mash-ups could be considered one type of remixing. According to Théberge, making mash-up recordings involves layering songs by different artists. The results are juxtapositions with often humorous and satirical intentions. Mash-up recordings transform the original recordings by recontextualizing them through juxtaposition. (Théberge 2004: 153.)

There is an interesting similarity between 'cross-overs' as one approach for fans to rewrite television shows, and mash-ups that intermix music from two or more artists. In fans' rewritings of television shows, cross-overs blur the boundaries between different media texts and intellectual properties such as two originally non-related television shows (Jenkins 1992a: 170). In official media texts, such crossovers are rare, because they require approval from the owners of both intellectual properties. Fans are free in private to combine these intellectual properties as they see fit. In the domain of music, there has been cross-over type of mash-ups such as DJ Danger Mouse's *Grey Album*, which combines The Beatles' *White Album* and the *Black Album* by Jay-Z (Wikipedia 2005b). This mash-up recording is made from material sampled from the two albums. This approach to making music is quite different from arranging a piece of music for instance "in the style" of The Beatles.

---

[46] Abercrombie and Longhurst (1998) suggest the continuum of 'consumer', 'fan', 'cultist', 'enthusiast', and 'petty producer' to distinguish how active audience members are. Some fan activities Jenkins (1992a; 1992b) describes are in Abercrombie and Longhurst's definition more the activities of an enthusiast or a cultist than a fan. Nevertheless, I have retained Jenkins' use of the term of 'fan'.

Media texts made by fans as extensions to original media texts can challenge the meanings of the original texts. Jenkins (1992a: 279) argues:

> The nature of fan creation challenges the media industry's claims to hold copyrights on popular narratives. Once television characters enter into a broader circulation, intrude into our living rooms, pervade the fabric of our society, they belong to their audience and not simply to the artists who originated them. Media texts, thus, can and must be remade by their viewers so that potentially significant materials can better speak to the audience's cultural interests and more fully address their desires.

The same can be said of music. When a piece of music has been made available either as a sound recording or in written notation, it can be performed (interpreted and thus to some degree remade by its performers) in many different ways, including altering the lyrics to suit the occasion of the performance. With music utilities for PCs and music-making software, especially *MTV Music Generator 2*, pieces of music can be remade in many different ways. For copyright reasons, fans have limited options to make their remakes available to others.

Fans take pleasure in making intertextual connections across a variety of media texts and not just single texts. According to Jenkins, the aesthetic of fan activities centers on the selection, inflection, juxtaposition, and recirculation of existing material. Appropriating and recombination are as much or even more important than original creation. (Ibid.: 36, 223-224.)

Jenkins points out that there may be nothing empowering in the media texts themselves. The empowerment emerges from what fans do with those texts when assimilating them into their lives. Fandom is about exceptional interpretations of media texts. Obviously, the characteristics of any particular media text do to some degree influence the way it can be read and interpreted. (Ibid.: 284.)

Following Jenkins (1992b: 215), remixing music can be interpreted as a form of textual poaching, that is, appropriating media texts and reworking them into something that serves the interests of the music-makers. The meanings suggested by the original texts may, for instance, be contradicted or satirized. The approach of remixing music as textual poaching requires that the sampled and remixed material remains recognizable to the audience.

Remixing can be regarded as one aspect of fan activities, as a way of practicing the analysis and assessment of the sampled material by choosing fragments to sample, potentially challenging the meanings attached to the source material, for example, glorifying or ridiculing the original material.

Fan activities like remixing should be recognized as personal ways of using media texts, most of which are the intellectual property of media corporations. With proper tools, these activities become feasible in the domain of music. In Chapters 4 and 5, I explore the foundations of how to combine using these tools to make music with the activity of playing video games.

4. The Semiotic Domains of Video Games

4.1 Video Games as Semiotic Domains

As discussed earlier, one starting point for this study is the theory of semiotic domains by Gee (2003). Gee (ibid.: 19) defines video games as a family of related semiotic domains. Being involved in the semiotic domains of video games includes, for instance, social practices like the different approaches to playing, talking and writing about video games, and modifying or making new game levels. The semiotic domains of video games include, among other things, the meanings given to game software and the hardware for playing these games, and metatexts about video games, including strategy guides, walkthroughs, and cheat codes (ibid.: 43).

In Chapter 4, I define the background of the semiotic domains of video games. This discussion is an extension to Gee's publication, where the focus was on video games as tools for learning to become literate or a producer in various semiotic domains. The focus in Chapter 4 is on describing the hardware associated with video games, defining play and games, video games as systems, and the theme of music-making videos. I also describe seven music-themed video game titles that I will use later as comparison material. The outcomes of this chapter are the starting points for the second part of analysis in Chapter 5, where I compare video games to music-making software, and playing to music-making.

4.2 Video Games as Hardware and Software

Because several writers have discussed the history of video games in detail (see Herz 1997; Kent 2001; DeMaria & Wilson 2002), I will not reiterate in detail the background and development of video games. Possibly several thousands of video game titles have been developed since the early 1970s. To name one systematic attempt to categorize video games, Wolf (2001e) has defined 42 video game genres.

Video games are rich audiovisual media. In recent video games, the graphics have been photorealistically impressive. Music and sound effects have often been of the same quality as those in television and movie productions, and symphony orchestras have occasionally been hired to record game soundtracks. Video games have become texts of global popular culture with their special characters, actions, conventions, and worlds. Video games can extend, refer to, and comment on media texts like television programs, movies, and graphic novels. Since the 1990s, video games have become influential as source material for motion pictures (Wolf 2001a: 1). Video games are a powerful competing form of entertainment for movies and television.

Even though video games have been available to consumers for only a couple of decades, the cultural landscape of video games has become broad.[47] Wolf and Perron argue that video games have a ubiquity and availability unlike any previous medium. (Wolf & Perron 2003: 20.)

Video games are played with different platforms ranging from personal computers to hand-held devices. Two common platforms for video games are personal computers and video game consoles. Personal computers have been used in homes since the 1980s and one popular use for them has been to play video games. Some observers suggest that video games are driving the development of increasingly powerful PC technology (Mäyrä 2002: 5).

Several generations of video game consoles have existed from the first devices introduced in the mid-1970s, including the more recent models Sony PlayStation 2, Microsoft XBox 360, and Nintendo GameCube. The superficial difference between a PC and a video game console is that the latter is rarely equipped with an alphanumeric keyboard or a mouse. There has been little use for accessories such as printers and scanners with video game consoles.

The term ‘personal computer’ refers to a multipurpose device, in which the functionality can be altered by changing software. In contrast, the term ‘video game console’ has traditionally referred to relatively fixed and closed hardware setups, to which users cannot upgrade the video and sound hardware, or add more memory the way they can on PCs. The term ‘video game console’ also has become to promise simplicity of use. Video game consoles are not multipurpose devices to the same extent as personal computers that can be expanded and configured in various ways.

However, recent video game console models like XBox 360 are no longer so fixed, because they have integrated hard drives and interfaces for external devices. Because PlayStation 2 and XBox 360 are equipped with DVD drives, they can be used to reproduce other forms of entertainment besides games. A recent trend (already established in PCs) in game consoles is network connectivity, which enables participation in multi-player online games and communicating with other players. By loading special software, certain video game console models can run operating systems similar to those for PCs and thus be used for many of the same tasks as personal computers.

All video games for a certain platform are played through the same physical interface consisting of the game controllers, electronic screen, loudspeakers, and possibly other accessories. A video game title appears the same in all game consoles of a specific model (depending obviously on the type of electronic display used). This is typically not the case with PCs, where various combinations of hardware and operating system optimizations may affect the game’s speed, graphics and audio detail levels and such.[48]

---

[47] The report *Global Entertainment and Media Outlook: 2005-2009* issued by PricewaterhouseCoopers presents detailed statistics on the markets of video games. For a discussion on video games as an economic force, see McAllister (2004: 18-24).

[48] For a discussion on the similarities and differences between personal computers and game consoles, see Finn (2002).

A video game console does not do much by itself, but software such as a game, must first be loaded. Specific video games are sometimes referred to as ‘titles’. Video games are also ‘software’ and ‘applications’, but these two terms are rarely used to describe video games. Games for video game consoles are disseminated on distribution media like CD and DVD. Distributing games through computer networks may become a popular mode of distribution in the future.

Various types of game controllers have been developed over the years. Game controllers are input devices that may also give some form of feedback like vibration to the player. A generic controller for the current video game consoles features a couple of joysticks, a directional pad, and about a dozen buttons. There are also special game controllers like adaptations of steering wheels and pedals, handguns, conga drums, guitars, dance mats, and video cameras. Microphones are used in karaoke software, in various games to activate commands given through speech, and in some multiplayer networked games to transmit voice to other players. Blaine (2005) has presented an overview of alternate game controllers as musical interfaces.

4.3 Defining Play, Games, and Video Games

Following Juul (2003: 1, 44), I have chosen the term ‘video game’ as the umbrella term for games played using computer processing power and memory.[49] Salen and Zimmerman (2004: 80) define a ‘game’ as “[…] a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome.” Following this definition, video games can be defined as systems maintained by computer processing power and memory. There is input from players to the system through controllers. Output from the system takes place through an electronic screen, loudspeakers, and controllers equipped with force feedback.

Video games require the activity of play from the people engaged in them. Next, I present a partial definition of the activity of play, and elaborate it further throughout Chapter 5. The cultural historian Johan Huizinga suggested in his pioneering publication *Homo Ludens: a Study of the Play-Element in Culture* that play might have an end of its own (Huizinga 1980 [1949]: 1-45). In other words, there may be no other reason for playing except enjoying the experiences it affords.

In his book *Man, Play and Games*, which is another early study on play and games, the philosopher and anthropologist Roger Caillois (1961: 9) suggests that play has

[49] Some scholars define ‘computer games’ as a subset of video games (Wolf 2001b: 17). Some scholars prefer the term ‘digital games’. I have chosen ‘video game’ as the umbrella term because in this study the focus is on games for PlayStation 2. According to Wolf and Perron (2003: 21 fn. 5), the term ‘video game’ is also the common term used in everyday parlance for these games. The game researcher Ken S. McAllister prefers the term ‘video game’ over ‘computer game’ because “[…] it more accurately describes the central technology that enables the medium. Video is surely important for computer games, but it is not their only defining characteristic.” (McAllister 2004: 205 n. 1.)

attractive and joyous qualities because it is free and a diversion rather than compulsory. Playing involves players having fun (Huizinga 1980 [1949]: 3). Perron (2003: 240) proposes following Huizinga that play is greatly dependent on the attitude of the player: "It is the player's state and presence of mind that determine this free activity and make acceptable the given though arbitrary rules. The fun of play is the fun of the player. This is one of the fundamental characteristic of play and games." (Ibid.: 240.) Obviously, there are many ways to have fun. For instance, intense excitement can be fun, and some humorous aspect of playing can be fun. However, if a game is too easy or too difficult, playing it may not be considered fun.

Play is uncertain in the sense that neither the course nor the result of it can be determined beforehand. Room for innovation is left to the players. (Caillois 1961: 9.) This uncertainty attracts both the players and the audience.

Play can have aspects of mimicry. During the course of play, the player can become someone else, for example a fictitious character, and behave like that character. (Ibid.: 10, 19.) Various activities can be simulated during play.

Play and game are not the same. Some scholars, including Frasca (2003), use the terms 'paidia' and 'ludus' to define the difference between play and game. Frasca has adapted these two terms from Caillois, but uses them differently. According to Frasca (ibid.: 229-230):

> *Paidia* refers to the form of play present in early children (construction kits, games of make-believe, kinetic play) while *ludus* represents games with social rules (chess, soccer, poker). […] It is common to think that *paidia* has no rules, but this is not the case: a child who pretends to be a soldier is following the rule of behaving like a soldier and not as a doctor. […] the difference between *paidia* and *ludus* is that the latter incorporates rules that define a winner and a loser, whereas the former does not.

The game designers Tracy Fullerton, Christopher Swain, and Steven Hoffman define games as formal systems that are goal-based and have explicit rules. In contrast to games, play is informal, not driven by goals, and while it may have rules, play does not depend on those rules for its form. (Fullerton et al. 2004: 89.)

Rules bring order to the spontaneity and tumult of free-form play (Caillois 1961: 28). Playing a particular game involves players in making decisions and performing actions within a structure regulated by the rules of the game. Frasca (2003: 230) points out that both play and games depend on rules. In games, the rules are social and known (at least to a certain extent) to all who participate in the game. Knowing the rules also influences how the audience experiences the game. In paidia play there may be rules, but these are defined by the player and may even be changed during the course of the play activity.

Play is generally considered a free-form activity, and game a rule-based activity (Juul 2003: 20-21). Rules specify affordances (including the actions a player may do) and limitations (including the actions a player may not do) (ibid.: 55). Rules may also define interesting challenges that players cannot easily overcome (ibid.: 5).

Playing a game is tied to the processing of its rules, and in video games computer processing power is used to uphold the rules. Thus, the players do not have to enforce the rules themselves and they can play games in which they do not have to know all details of the rules. Through using cheats officially implemented in a video game, players can modify some of the rules of that game. (Ibid.: 50.)

Quoting the philosopher Andre Lalandé, Frasca defines 'ludus' as an "[…] *activity organized under a system of rules that defines a victory or a defeat, a gain or a loss*." (Lalandé quoted in Frasca 2006 [1999].) Rules bring order to play (Caillois 1961: 7). Games have outcomes because of rules that make it possible to define who wins and who loses, or that one performance in the game is superior or inferior to some other performance. Play may have private or social rules, but it does not produce results. What distinguishes a game from free-form play activities are rules that define a variable, quantifiable, and valued outcome (Juul 2003: 32-34). Conflict is central to games (Salen & Zimmerman 2004: 80). One way to introduce conflict into a game is by having rules specifying conflicting goals between players or teams. Conflict in games is artificial in the sense that it is created for the purpose of struggle to resolve it during the course of the game.

Salen and Zimmerman (ibid.: 95) argue that when a person plays with a toy, it can be difficult to say when the play activity begins and ends. The boundaries of playing and not playing are indistinct, while in games the boundaries are distinct: "As a player steps in and out of a game, he or she is crossing that boundary […] that defines the game in time and space." (Ibid.) Following Huizinga, Salen and Zimmerman (ibid.: 95-96) call this boundary 'magic circle':

> The term magic circle is appropriate because there is in fact something genuinely magical that happens when a game begins. […] Within the magic circle, special meanings accrue and cluster around objects and behaviors. In effect, a new reality is created, defined by the rules of the game and inhabited by its players. Before a game of Chutes and Ladders starts, it's just a board, some plastic pieces, and a die. But once the game begins, everything changes. Suddenly, the materials represent something quite specific. This plastic token is *you*. These rules tell you how to roll the die and move. Suddenly, it matters very much which plastic token reaches the end first.

The magic circle of a game is a space for contemplating and making special decisions, performing special actions, and experiencing certain emotions.

Players are within the magic circle of a game for certain periods of time. In my definition, a 'game session' is the period from starting a video game to reaching an end condition defined by the game (such as reaching the ultimate goal or losing all resources such as lives) (see Björk & Holopainen 2005: 9).

Because of these special qualities that distinguish play from other activities, play can be a special kind of approach to trying new things. Fullerton et al. (2004: 88) argue:

> Play is recognized as a way of achieving innovation and creativity because it helps us see things differently or achieve unexpected results. A playful approach can be applied to even most serious or difficult subjects because playfulness is a state of mind rather than an action.

Play is a potential attitude to doing various things, hypothetically including music-making.

4.4 Video Games as Systems and State Machines

Video games are systems implemented with software, in which complex behavior emerges from the interaction between discrete elements (Fullerton et al. 2004: 107). In general, a system is defined as an assemblage of interrelated and interacting elements or components that comprise a unified whole.

Furthermore, games are rule-based systems (Juul 2003: 5) and rules provide state machines. Juul (ibid.: 57) writes:

> […] a game is a system that can be in different states, it contains input and output functions and definitions of what state and what input will lead to what following state […] When you play a game, you are interacting with the state machine that is the game. […] in computer-based games, the state is stored as variables and then represented on screen.

Games are dynamic systems producing various states while they are being played. For instance, the score changes, the principle of scoring can change, the goals can change, and so forth. Video games are systems with input and output functions. Players feed input to the system through an interface. Rules govern a game's behavior from one state to another. Rules define at what state which input leads to what output and what new state. The output from a game system occurs through the interface. (Ibid.; Järvinen 2003: 69.)

A system such as a video game has a structure. According to Costikyan (2002: 20-21), a game's structure guides the behavior of the player:

> Game structure has to do with the means by which a game shapes player behavior. But a game *shapes* player behavior; it does not determine it. Indeed, a good game provides considerable *freedom* for the player to experiment with alternate strategies and approaches; a game structure is multi-dimensional, because it allows players to take many possible paths through the "game space."

Playing a game involves making choices. Costikyan argues that games are interactive in the sense that a game interacts with the player by changing state in response to the input from the player. For Costikyan, playing a game involves interaction with the purpose of reaching the goals defined by the game. (Ibid.: 11.) System theorists would probably argue against Costikyan and point out that a video game cannot be interactive because it cannot anticipate the actions of its players. In this sense, video games are active, not interactive. Only humans can be interactive because they can anticipate the actions of others. Artificial Intelligence (AI) has been developed for video games partly to better anticipate the players' actions. An alternative interpretation is that games are interactive in the sense that the game state changes in response to the choices made by the player, and the future actions of the player change in response to the altered game state.

The interactivity of a video game is in part confined by its game mechanics. For example, in *Pong*, the player can move the racket in only two directions, even if he plays the game with a joystick that enables control in four or more directions. In *Pac-Man*, the player-character can be moved in four directions. These are two examples of simple game mechanics that define how the player's agent can be moved in the virtual world of the game.

According to Salen and Zimmerman (2004: 316-317), the core mechanic of a game is the mechanism through which players make meaningful choices in the game:

> A core mechanic is the essential play activity players perform again and again in a game. Sometimes, the core mechanic of a game is a single action. […] In a trivia game, the core mechanic is answering questions. In Donkey Kong, the core mechanic is using a joystick and jump button to maneuver a character on the screen. However, in many games, the core mechanic is a compound activity composed of a suite of actions. In a first-person shooter game such as Quake, the core mechanic is the set of interrelated actions of moving, aiming, firing, and managing resources such as health, ammo, and armor. Baseball's core mechanic is composed of a collection of batting, running, catching, and throwing skills.

Video games have player-controlled agents that perform actions in the virtual world of the game. In many cases, those agents can be called 'player-characters'. Salen and Zimmerman suggest in the quote above that the core mechanic of basketball is the same in a basketball game taking place in the real world and a basketball video game. The actions the player performs while playing a basketball video game are very different from the actions a basketball player performs in the real world. Nevertheless, the mechanics of the game of basketball are similar in these two instances because in the video game the player-character performs actions similar to those of a basketball player in the real world, including batting, running, catching, and throwing. It is important to distinguish between what the player does physically (i.e., player actions) and what the player's agent can do in the world of the game (a part of the game's mechanics).

To summarize, video games are similar to other games, for example, in that they are state machines and they are played within a practical and emotional 'magic circle'. However, video games have characteristics that other games do not have. For example, video games are sustained by computer processing power and memory. Video games are played on electronic displays with input devices such as joysticks. In video games, the players control characters that interact with the virtual worlds of the games. The video game system may control opponents and collaborators of the human players, or act as a referee. Through computer networks, people in different parts of the world can participate in the same game at the same time.

4.5 The Theme of Music-Making in Video Games

The present study is not the first to address the question of the similarities between play and music-making (see Blaine 2005; Denis & Jouvelot 2005; Hindman 2006). For example, Huizinga (1980 [1949]: 42) briefly discusses the similarities between play and making music:

> Making music bears at the outset all the formal characteristics of play proper: the activity begins and ends within strict limits of time and place, is repeatable, consists essentially in order, rhythm, alteration, transports audience and performers alike out of "ordinary" life into a sphere of gladness and serenity, which makes even sad music a lofty pleasure. In other words, it "enchants" and "enraptures" them. In itself it would be perfectly understandable, therefore, to comprise all music under the heading of play. Yet we know that play is something different, standing on its own. Further, bearing in mind that the term "playing" is never applied to singing, and to music-making only in certain languages, it seems probable that the connecting link between play and instrumental skill is to be sought in the nimble and orderly movements of the fingers.

It should be noted that Huizinga does not take into account spontaneous music-making, like humming and tapping rhythms, but considers making music as a special activity with a clear beginning and end.[50]

Like play, performing and listening to music can be an enrapturing experience, as Huizinga suggests in the quote above. Performing music on a musical instrument can be similar to playing a video game in the sense that both the performer and the player enter a feedback system (in the case of games, see Salen & Zimmerman 2004: 218-219, 315-316). In musical performance (especially improvisation), what the performer has performed probably affects his subsequent actions and the future actions of the other performers. Something similar can be found in single-player and multiplayer games. Few games progress linearly, but the player has freedom to experiment with alternative approaches while pursuing the goals of the game (Costikyan 2002: 20). In a similar way, a performer is confined by the framework of the piece of music he is performing, but the performer can to some degree improvise within that structure. The range of improvisation depends on the musical style and the context of the performance. For example, there is little room for improvisation in the performance of western classical art music. Both the player and the performer must think ahead of the actions to be performed in the future within the game session or the musical performance.

[50] In addition to the observation Huizinga (1980 [1949]: 42) makes about the manipulation of musical instruments being called 'playing' in certain languages, the reproduction of music from sound recordings is referred to in English "playing records" and the devices are called "record players", "cassette players", "MP3 players", and so forth. Salen and Zimmerman (2004: 302-303) discuss the various ways the terms 'play' and 'playing' have been used from the "act of creating music" to "activating a process", and from stalling to fooling someone. Juul (2003: 20-21) discusses some of the issues arising from the different meanings that the terms 'play' and 'game' have in various languages.

Many video games are such that players play them with “nimble and orderly movements of the fingers” on a game controller. There are certain patterns of fingerings for performing music with certain musical instruments. Perhaps there are related fingerings for playing certain video games with specific controller devices. This involves questions about real-time musical performance rather than nonlinear multitrack composing, which is the focus of this study. Thus, I will not explore this issue further here.

Music and sound effects have been a part of video games since the 1970s.[51] Music and sound (including sound effects and virtual acoustic environments produced with reverb and other spatial effects) appear to be an important part of the appeal of video games because they contribute to the immersion into games. Since the earliest video games, sound has been used as a way to emphasize the action in the game and to give feedback to players. Sound effects tell about the events occurring in the game, and in some video games, such information is also conveyed by the background music. Some video games have adaptive soundtracks that change smoothly according to what is happening in the game, rather than through cuts or cross-fades between fixed music clips.

Music contributes to the mood and the aesthetic experience of a video game. In some games, the player can choose the background music. In recent years, some recording artists have licensed their music to video game soundtracks and have recognized video games as a way for reaching new audiences with their music. There are some video games in which the background music influences the events in the game. One such background music responsive game[52] is *Vib-Ribbon*.

The theme of music-making can be found in some video games. Järvinen defines ‘theme’ as the subject matter that provides a meaningful context for everything that occurs in a game. Theme has several layers, such as game mechanics and audiovisual style. Theme consists of a setting (time and place) and a motivational psychological element. These may be adopted from the conventions of fiction, sports, and so forth. The theme of a video game is embodied in the literal and verbal rhetoric of the game. Rhetoric includes, for example, the names and descriptions given to actions in the game. Theme contributes to rendering a game coherent. Theme produces different expectations of a game and contributes to interpretations of it. Adopting a new theme can be a way to make old game mechanics appear fresh. Even if two games appear different because of their themes, they may have the same game mechanics beneath the surface appearance. (Järvinen 2003: 75-76.)

Pitch-control is a special game control method that refers to video games that respond to the pitch of the player’s whistling, singing, or humming (see Hämäläinen et al. 2006 [2004]). To name one example, in *Staraoke*, the pitch of the melody of the background music forms a path on the screen. Pitch is represented on the vertical axis and the progression of time on the horizontal axis. Thus, the viewpoint to the game world

---

[51] The musicologist Karen E. Collins (2006) has presented an overview of video game audio and music technology from the early 1970s onwards.

[52] For an overview of background music responsive video games, see Holm et al. (2006). For discussions on designing sound-based games, see Roden & Parberry (2005); Friberg & Gärdenfors (2006 [2004]).

is from the side. The player must guide his character through the path by singing the melody correctly.

One example of a video game in which music is the result of the players' actions is *BLiX* for Shockwave-enabled Web-browsers (Sweet 2004: 307-309). Another example is *Rez*, which is a third-person shoot'em up game with electronic drum sounds instead of conventional weapon sounds. When the player shoots, the drum sounds are replayed so that they rhythmically match the background music. It could be said that in video games such as *BLiX* and *Rez* the player's actions contribute to the musical soundtrack during the course of a game session.

Rhythm games are probably the most common type of video games that involve the theme of music-making. Wolf (2001e: 130) defines the rhythm game genre as follows:

> Games in which gameplay requires players to keep time with a musical rhythm. These games may employ a variety of controllers beyond the usual video game hardware, including controllers simulating drums (as in *DrumMania*), turntables (as in *Beatmania),* guitars (as in *Guitar Freaks*), or even maracas (as in *Samba de Amigo*).

In the examples of rhythm games mentioned by Wolf, sample files are triggered in real-time. Blaine (2005: 32) calls into question the effectiveness of the music-making experience in rhythm games and notes that instead of a creative input there is the reflexive following of the prompts dictating what buttons to press and when. Blaine (ibid.) argues that without special controllers, like the turntable controller for *Beatmania*, the playing experience is much less fun.

In some video games, it is an explicit part of the objectives of the game to make music in different ways. Next, I describe seven such games: *Beatmania*, *Amplitude*, *Fluid*, *Gitaroo Man*, *SingStar, Get on da Mic,* and *100% Star*.

4.5.1 Beatmania

Several *Beatmania* game titles (also known as *Bemani*) have been developed for various platforms like arcade machines and hand-held devices. One of these titles was developed by Konami for PlayStation and released in Europe in 2000. A special controller device for this version of *Beatmania* was available for some time. This device has five keys in the shape of the keys of an organ keyboard. The controller also features a "turntable" disc about the size of a CD that can be rotated clockwise and counterclockwise simulating the scratching that deejays do with vinyl records. The PlayStation version of *Beatmania* can also be played with a regular game controller.

Beatmania is one of the examples mentioned by Wolf (2001e: 130) in his definition of the rhythm game genre. The repeated action the player performs in *Beatmania* is to trigger sample files with sounds like drum beats, short guitar riffs, vinyl scratching sounds, and brief vocal exclamations.

There are six vertical bars (tracks) on the screen. One button on the game controller (or key and turntable on the special controller) corresponds to a track. Sample files are represented with small blocks called “notes”. Blocks move vertically from the top of the screen to the bottom. The blocks are aligned to the six bars on the screen and a button is assigned to each bar. The player must press the corresponding button when a block is at a specific vertical position on the screen. This action will trigger the sample file and increase the player’s score. The sample file that each button triggers varies during the course of the song. The player is given constantly feedback on how well he is doing through exclamations like “Great!”, “Good”, “Bad”, and “Poor”. The game becomes increasingly difficult over time as blocks fall closer to each other. If the player does not “catch” enough blocks (i.e. press correct buttons at correct times), he will not progress to the next level with a different piece of music. There are at least seven songs to play through.

A single player can compete against the game system or two players can compete against each other. The competition is indirect: The players’ actions do not influence (neither help nor hinder) each other. The winner of a game session is the player with the higher score.

4.5.2 Amplitude

Amplitude is the sequel to *Frequency*. Both titles are rhythm games developed by Harmonix for PlayStation 2. *Amplitude* was released in 2003. *Amplitude* features adaptations of about 25 songs from artists like David Bowie, Garbage, blink 182, Herbie Hancock, Run-D.M.C., Pink, Slipknot, and Weezer.

Although both are in essence rhythm games, *Amplitude* looks very different from *Beatmania*, because the world in *Amplitude* appears three-dimensional and has “depth”, whereas in *Beatmania* the game area is a two-dimensional plane. There are also short video clips and animations scattered around the game world of *Amplitude*.

The difference in the theme between *Amplitude* and *Beatmania* is that in *Amplitude* the player controls a spaceship that moves along a curved path. The player must shoot with the spaceship at gem-shaped cells that pass beneath three shooting marks that are aligned side by side. Buttons L1, R1, and R2 on the game controller correspond to these shooting marks. The cells represent sample files and are located on six tracks comprising a piece of music. When a cell is hit, a sample file is triggered and the energy released from it is added to the energy of the player. The cells’ appearance on the screen is synchronized to the beat of the music. Cells are grouped in sequences that are typically two measures long.

When the player manages to shoot all the cells in a sequence, the music on the track can be heard for some time. If the player misses a cell, the track becomes muted, and the player must try again. The musical content of each track varies between the songs and the different parts of a song, but generally they include vocals and instruments such as synthesizer, drums, bass, and guitar.

A game level is divided into parts that follow the structure of the piece of music playing in the background. The parts are called “verse”, “chorus”, and “break”. There can be several of these parts in a piece of music. Between each part is a checkpoint, and passing it increases the energy of the player. If the player misses several notes, he loses energy in the game. A game session ends when the player has lost all his energy.

There is a character called "FreQ", who is somehow associated with the spaceship. The FreQ character shows with an animation what type of track is chosen at the moment. The player can customize the appearance of his FreQ character.

The two main modes are "game" and "remix". In the "game" mode, there are four difficulty levels called "mellow", "normal", "brutal", and "insane". A characteristic of the "game" mode is that the player must collect energy in order to keep the game session going. In the "game" mode, the player gets a score for his actions. There are single player and multiplayer modes to choose from. *Amplitude* also features playing against other people through a network connection.

In the "game" mode, the player occasionally gets a power-up as a reward for managing to repeat the patterns on a track. A powerup can be used only once to assist the player in the game. Single player, multiplayer and network play powerups include "autoblaster" (the contents of a measure is cleared immediately), "freestyler" (jamming with scratching type of effect and getting points in the process), and "score doubler" (doubles the score that can be obtained from eight bars). Single player and multiplayer powerup is "slo-mo" (that slows down the speed of the music for a while). Multiplayer and network play powerups are "bumper" and "crippler", which can be used to cause trouble for other players.

In the "remix" mode, a mix of the song is defined in real-time by inserting sample files (by shooting with the spaceship) into one track at a time. The six tracks are the same as in the game mode. The remixed song can be stored for listening to it later. Energy is not measured in the "remix" mode, and thus remixing ends only when the music-maker so decides. The game's tutorial addresses the player as "DJ".

Amplitude features three sound processing effects called "chorus", "stutter", and "delay", which can be applied in to modify the sample files on a track. "Delay" is an echo-type effect (see Section 3.6.5) and "chorus" is like the effect defined with the metaterm 'chorus' (see Section 3.6.3). "Stutter" appears to be a type of tremolo effect.

The mix made while playing through a level (song) can be "replayed" (without the need for player input) after reaching the end of the level. Mixes made in the remix mode can be played as game levels.

4.5.3 Gitaroo Man

Gitaroo Man is in essence a rhythm game similar to *Beatmania* and *Amplitude*. In other words, the core action is that the player must press the correct buttons at correct times. *Gitaroo Man* features the additional challenge that in some modes the player must also follow the pitch of notes that moves up and down with the thumb-stick on the DualShock 2 controller. In some modes the player must also point the thumb-stick towards the direction from where the notes move towards the player-character. The buttons the players must press and the curving pitch lines are shown on the screen. The timings of when the player must press the buttons occur to the beat of the background music.

What differentiates *Gitaroo Man* from *Beatmania* and *Amplitude* is that the theme of making music is strongly bound to the story of the game. The theme presents a "musical battle" between Gitaroo Man and his opponents. The backstory is told in the printed user's guide as well as in an animation at the beginning of the game. The story unfolds through animated sequences between game levels.

The story of *Gitaroo Man* is about a boy named U-1, the player-character, who in the beginning has little self-confidence. U-1 learns that he is the heir to the title of Gitaroo Man, a hero who uses the power of his Gitaroo instrument to fight the forces of evil. An alien race called the Gravillians is attempting to take over the universe. U-1 must travel around the universe and fight these grave villains. While confronting these ordeals, U-1 learns to believe in himself.[53]

The fight between Gitaroo Man and each opponent is divided into three stages called "charge", "battle", and "final". When the player's actions are successful, they generate a score. When one opponent has been defeated, the game moves on to the next opponent. The background music is different during the fight with each opponent.

According to the game journalist Chris Kohler, Keiichi Yano, the game designer of *Gitaroo Man*, had the idea of jazz performance as in some way battle-oriented because of performers take solos from each other, and occasionally attempt to outdo the other musicians. One starting point in the design of the game was to visualize jazz "battles" in the style of shooting games like *Combat*, with laser beams coming out of the guitars. (Kohler 2005: 161.)

4.5.4 Fluid

Fluid was developed by Sony for PlayStation and released in 1996. In the product package, the subtitle of *Fluid* is "Interactive Sound Lab". In other words, *Fluid* is presented as an environment for experimenting with music-making.

The player controls a dolphin that swims in different underwater and fantasy worlds. The viewpoint to the apparently three-dimensional worlds is from behind the dolphin. There are three stages, namely "cruise stage", "groove editor", and "silent space".

There are twelve worlds with sample files in different musical styles. There are altogether c. 600 sample files. The player gets to a new world and can access a new set of sample files by swimming long enough during the "cruise stage". "Cruise stage" also functions as a "jam" mode, where four different sounds can be triggered in real-time on top of the song made in the "groove editor". The sounds (such as piano and strings) are triggered by pressing cross, triangle, square, and circular buttons on the game controller (several sounds can be triggered simultaneously) and the pitch of the sounds is changed by pressing the directional buttons.

[53] Kohler (2005: 162) notes that U-1 is a bilingual pun. In Japanese it is shorthand for the first name Yuuichi. In English, U-1 is pronounced almost like the phrase "you won".

In “groove editor”, the player makes pieces of music from “sound patterns” that are in essence looped sample files. The contents of the “sound patterns” are not programmable. Sample files include drum patterns, guitar riffs, bass lines, and so forth. There are eight tracks for the sample files. *Fluid* features some aspects of nonlinear multitrack composing, including selecting and modifying sample files in real-time with echo and reverb effects, defining the volume of each looped sample file, defining the pan position, applying the “autopan” feature, and applying vibrato (called “modulation”) to the sample files. Tracks can be muted and unmuted in real-time. The tempo of the song can be adjusted. Two alternative setups for the eight tracks can be defined, and it is possible to switch between these setups during playback by pressing the select button on the game controller. These setups can be saved.

“Silent space” is in essence a gateway between “cruise stage” and “groove editor”.

4.5.5 SingStar

SingStar (London Studios 2004) for PlayStation 2 is a karaoke game. Karaoke singing as such does not constitute a game. Of course, there have been plenty of karaoke competitions where people rank the performances into some order so a winner can be declared. In *SingStar*, the game system rates the singer’s performance. *SingStar* features a reverb effect and an optional video input for EyeToy video camera. Two microphones are included in the product package. *SingStar* features 30 popular songs and accompanying music videos. Extension sets with more songs and videos are also available.

The lyrics and an approximation of the melodic line of each song are shown on the screen. The accuracy of the player’s performance is measured through note onsets and pitch level that is apparently rounded to the nearest tone or semitone, depending on which of the three difficulty levels has been chosen. A representation of the player’s singing is shown on the screen as a line on top of the melodic line of the original melody. The accuracy of the player’s singing is shown on the screen in almost real-time. The more accurate the player’s performance, the higher score he gets. There is no scoring in “freestyle” mode.

SingStar offers several modes for the player. In one mode, the player’s voice can be transformed, which is suitable for certain types of songs. The original vocals can be lowered in volume or be entirely muted when the player starts singing. *SingStar* has several multiplayer modes, including a mode in which two players sing simultaneously into their microphones. The winner in one of the multiplayer competition modes is the player who gets the highest total score from three songs.

Through the EyeToy video camera accessory, the player can see himself on the screen. Video effects are used to transform the picture from the video camera so that it appears rather like a (1970s) music video. When a player sings certain musical notes defined as “golden notes” correctly, the “pose for paparrazi” feature is activated and a short video clip of the player's performance is recorded to an album, from which it can be viewed later.

"Star maker" mode offers some degree of fantasy or role-playing even if there is no player-character as such. The player performs at clubs and attempts to become a star in a world called "Harmony City". The more the player performs, the more "buzz" he creates. Because of "star maker" mode and the optional video camera feature, *SingStar* can be interpreted as an "idol" or a "pop star" simulator (similar titles include *Fame Academy*) rather than as a straightforward karaoke application for PlayStation 2.

4.5.6 Get on da Mic

Get on da Mic is a karaoke rap game or "rapaoke" as it is called in the game. A microphone is delivered with the game. *Get on da Mic* was developed for PlayStation 2 by Artificial Mind and Movement, Inc. and was released in 2004.

The songs in *Get on da Mic* are cover-versions of 40 hip hop songs. The volume level of the original vocals (however, not by the artists who originally made the songs known) can be diminished or entirely muted. Some of the lyrics contain certain explicit words that have been censored.

The lyrics are shown at the bottom of the screen. The player must rap the lyrics with correct timing, or as the user's guide states this: "Respecting the beat is high priority." The temporal accuracy of the player's rapping is shown on the screen. The player-character earns money when the player raps successfully. Money can be spent on housing, vehicles, jewels, clothing, electronics, luxury items, and acquiring new moves in the player-characters' performance animations.

Player-characters include the likes of Baby T, Malik West, Tricia Allen, Big Stacks, Kenia, and Flow-On. These characters' backgrounds are from different parts of the U.S. and thus are intended to represent different hip hop scenes. Additional characters are unlocked when the player makes progress in the game.

There are several single player and multiplayer modes with different levels of difficulty. "Career" mode involves a certain amount of fantasy or role-playing as the player-character progresses from the levels of (bathroom) mirror, to block, party, contest, mix tape, finals, showcase, album, launch party, video, and finally to world tour. "Freestyle" mode is for rapping one's own lyrics on top of the prefabricated songs or beats (parts of songs). Battle mode is a direct competition between players. There are also modes for practicing.

Get on da Mic supports the use of the EyeToy video camera, so that video images of the player performing can be shown on the screen.

4.5.7 100% Star

100% Star was developed by Teque Software for PlayStation. The game was released in 2001. *100% Star* is best described as a pop star manager simulation. The goal in the game is declared in the product package as to "[…] make the next super band that will take the world by storm." The user's guide also suggests that reaching the "Lifetime Achievement Award" is the ultimate goal. There is no player-character as such.

The game starts with the player choosing one or more artists as protégés. The player can design the appearance of each protégé. The player has a limited amount of money to allocate to different tasks. Each protégé can be trained in skills such as "vocals", "dancing", "fitness", "mood", and "charm". One of the tasks is to make pieces of music from a set of sample files.

Pieces of music are made in the "Music Editor" before they are "recorded". Recoding songs costs money. "Pre-written tracks" (songs) can also be purchased and then be "recorded", however, these songs cannot be heard. There is a limited number of pre-written songs, and the user's guide explains that "[o]nce they've all gone it's down to your creative juices to get an album out!" Thus, the player may be compelled to make his own music after some point during the course of the game.

The maximum length of a piece of music is 60 bars. The meter is fixed at 4/4 and the tempo cannot be adjusted. There are twelve tracks: "Drums", "Bass", "Synth", "Vocal", "Guitar", "Strings", "Percussion", and "Misc 1" to "Misc 5". The track types are fixed, so only certain types of sample files can be placed to each of them. There are in total fewer than 200 sample files, including, for instance, 42 drum, 22 bass, 22 guitar, and four string sample files. Sample files are identified only by their numbers. Sample files on the different tracks are downmixed automatically to a fixed balance. Thus, making new pieces of music is fairly straightforward, as only the selecting and arranging of sample files have to be considered.

When pieces of music have been recorded, they have to be released. The player chooses what songs are released in albums or on singles and at what times during the game session. The player can design the sleeves of the released records from a set of prefabricated graphics. The player also has to book concert performances for his protégé. The player must consider the size of the venue and the price of the tickets. The player may gain or lose money from the concerts. The concert performances of the songs made by the player are shown as 3D animations. The success of the player's protégés is shown on the record charts, representing how well their singles or albums have sold.

A major problem in *100% Star* is that it is difficult for the player to perceive what choices have what consequences. The player can easily get the feeling that his music-making has no influence on the success of his protégés. According to Björk and Holopainen (2005: 201-202), an illusion of influence makes players feel that their actions in the game have an impact on the outcome, regardless of whether this is the case or not. There is not much illusion of influence in *100% Star*.

5. Analysis Part 2: Comparing Video Games to Music-Making Software

5.1 Research Questions and Method

The purpose of the second part of the analysis is to answer the following question: What similarities and differences are there between playing video games and making music with the three music-making software titles? I approach this question from two angles. First I define the elements video games consist of, and identify the similarities to and differences from music-making software. Second I examine how these elements make possible the activity of play in music-themed and other video games, and explore what similarities and differences there are between playing these games and making music with music-making software. Whatever is shown not to overlap between them can be introduced into the design of future video games.

The method is to compare music-making software to video games through different definitions of video games. I have organized these definitions into three topics: Formal elements, dramatic elements, and video games as systems.

The basis of the formal elements of video games has been defined by Fullerton et al. (2004). I extend their definitions with arguments from Costikyan (2002), Juul (2003), Wolf and Perron (2003), Salen and Zimmerman (2004), and others.

The dramatic elements of video games have been defined by Fullerton et al. (2004). I extend their definitions with arguments by Costikyan (2002).

Video games have been defined as systems, for instance by Costikyan (2002) and Juul (2003). This topic also includes definitions of video games as state machines (Juul 2003), video games as simulations (Frasca 2003), and playing as mimicry (Caillois 1961). In addition, I consider here the differences between paidia and ludus as approaches to playing (Frasca 2003; Perron 2003).

These three topics enable me to compare various aspects of video games to music-making software, and playing to music-making. Of course, they do not exhaustively cover all possible definitions of video games. For instance, the social aspects of playing and music-making, cheating in playing, and playing as performance (see Eskelinen & Tronstad 2003) are not considered because they are not essential to finding answers to the research questions set out in Chapter 1.

The comparison material includes seven music-themed video games: *Beatmania*, *Amplitude*, *Fluid*, *Gitaroo Man*, *SingStar, Get on da Mic,* and *100% Star*. These have been developed for either PlayStation or PlayStation 2, so all of them can be played with the latter game console. Occasionally I also compare music-making software to characteristics of other video games including *Pac-Man*, *Theme Park World*, *Star Wars Bounty Hunter*, *Space Invaders*, and *SimCity*.

5.2 Similarities and Differences in Formal Elements

As discussed in Section 4.4, video games can be interpreted as software systems implemented with program code. Fullerton et al. (2004: 107) define a system as "[…] a set of interacting elements that form an integrated whole with a common goal or purpose." Video games consist of a set of elements that, when the system of the game is set in motion, create a dynamic experience in which the players engage (ibid.).

Fullerton et al. (ibid.: 24-30, 43-80) define the formal elements of video games as 'players', 'objectives', 'procedures', 'rules', 'resources', 'conflict', 'boundaries', and 'outcome'. Some of the formal elements are programmed to the game software. The remaining formal elements are outside the game software.

Throughout Section 5.2, I examine the similarities and differences between the formal elements of video games and music-making software. I summarize the findings in Section 5.2.9.

5.2.1 Boundaries

By 'boundary', Fullerton et al. refer to Huizinga's concept of 'magic circle', which is a temporary world within the real world where the rules of the game apply. Most games are closed entities, and boundaries indicate what belongs to the game. Fullerton et al. consider boundaries as practical and emotional. Boundaries are practical because they show when and where a game is taking place. Boundaries are emotional because whatever happens between the players during a game should not damage their relationships. (Fullerton et al. 2004: 76, 78-79.)

The physical and conceptual boundaries to a game must be crossed so that the activity of play can occur. Video games for consoles are typically played at fixed locations, such as living rooms or bedrooms in front of a television. Hence, some practical preparations are needed to begin playing a game, including getting to a specific place and loading the game software. However, this is not the case with all video games because beginning to play, for instance, a game with a mobile phone can be done extempore with few practical preparations.

Like games for PlayStation 2, music-making software titles are used in a certain place (wherever the game console is located) and with regular equipment (game controllers, television, and possibly amplifiers and loudspeakers). There is a substantial conceptual boundary to cross when beginning to make music with music-making software, as the things to consider during nonlinear multitrack composing are not mundane to most people.

Are there similarities between the ways the boundaries are crossed in video games and the ways music-making software titles are taken into use? Video games make invitations like: "Play as the legendary Gitaroo Man and save the planet from the clutches of the evil Gravillian family!" (text from product package of *Gitaroo Man*), and "You play Takt, a young conductor attempting to recruit the best musicians to put on a MEGA-CONCERT to save the concert hall" (text from product package of *Mad Maestro*). In other words, video games offer characters for the player through which to interact with the game. Games also set goals to achieve or conflicts to resolve. Music-making software titles present the following invitations in their product packages: “Become the greatest producer, director, DJ and remixer in the whole universe – with the MAGIX music maker!” In *eJay Clubworld*: “Eight real-world clubs and locations are waiting for you: each with a different and current music style for your creativity to explore.” In *MTV Music Generator 2* the slogan is: “Prepare for sound. Prepare for music. Prepare to generate.” In other words, music-making software titles present virtual spaces that are to some degree similar to multitrack studios. *eJay Clubworld* enables the music-maker to explore creativity, whereas games set goals to achieve or conflicts to resolve. They give access to sounds and tools the music-maker has otherwise no access to. Nonlinear multitrack composing is for most people a special activity.

5.2.2 Rules

Rules are a constitutive element of every game. According to Juul (2003: 50), games are tied to the processing of rules and thus games are immaterial. For example, it is the sets of rules that separate two card games such as poker and solitaire from each other, rather than the material props with which these games are played. When playing a game, players commit to follow the rules.

According to Juul (ibid.: 54), “[…] games contain a basic duality between their rules and their representation. Rules provide the dynamical aspect of a game.” To name an example of rules and their representation, “[…] the rules of chess govern the movement of the pieces; the representation of chess is the shape and color of the pieces” (ibid.). Rules “[…] *relate to selected and easily measurable aspects of the game context* […]” and include “[…] a specification of what aspects of the game and game context are relevant to the rule” (ibid.: 61-62).

The rules of a video game may be explained in the printed user’s guide or within the game itself. The player may not have to know all the rules in detail in order to play. Even if the rules are not explicitly explained, the player can gain some understanding of them while playing the game (Costikyan 2002: 19).

Fullerton et al. (2004: 64-68) describe three types of rules in video games: Rules defining objects and concepts, rules restricting actions, and rules determining effects. Through rules games create their own objects and concepts rather than inheriting them from the real world (ibid.: 65). Objects, for instance, are the components of a game. Concepts include the ways goals are declared important. Concepts may also overturn the laws or social rules of the real world by defining actions that are irresponsible or illegal in the real world as desirable in the game.

As part of defining concepts, restricting actions, and determining effects, rules define rewards and penalties. Rewards (e.g., in the form of resources) are given for making progress, whereas penalties (e.g., in the form of additional obstacles) are imposed, for instance, for failing to overcome an obstacle or achieving some sub-goal (Pearce 2002:

113). Rules influence the behavior of the player if he knows that he will get a reward from doing certain things and a penalty from failing to do certain things.

Rules define the actions the players can take and should take. For instance, the rules of the video game *Pong* stipulate that the racket can be moved only vertically and that the player must not let the ball past the racket. The first rule is represented in the graphical user interface of *Pong*. The latter rule is represented as the consequences of the player missing the ball.

In *Amplitude* and *Beatmania*, rules define the time frame within which the player must press a button in order for that action to be considered valid and be rewarded with an increase in the score. The exact durations of the time frames are not explained to the players. The rules of *Amplitude* define special features (like “autoblaster” and “slo-mo”) that can be used a limited number of times after the player has earned them. The player is likely to perceive access to these special features as a reward for his successful actions. By contrast, in music-making software, all features are available at all times but in different windows or menus. There are no rewards and penalties in music-making software.

In music-making software, it is not rules that define objects and concepts (Fullerton et al. 2004: 65-67) of what can be done with the software. I have argued earlier that the features of music-making software can be used to realize nonlinear multitrack composing techniques, which are concepts that are defined outside the software.

Juul divides rules into information rules, outcome valorization rules, and game state rules. Juul also makes a distinction between global rules that are used throughout a game, and local rules that apply only to a part of the game. (Juul 2003: 67.)

In video games, outcome valorization rules “[…] define what outcomes are considered positive, and what outcomes are considered negative.” (Ibid.) The outcomes of using music-making software are pieces of music. Music-making software titles do not valorize one musical outcome from another, and thus do not influence the decisions of the music-maker. The music-maker decides if the outcome is what he set out to do. I will later discuss the implications of the possible outcomes not being assigned values by the music-making software titles.

Game state rules “[…] cover the basic aspects of the game state - what players can and cannot do; what happens in response to different actions; the affordances and limitations that the game sets up.” (Ibid.: 66.) What the music-maker can and cannot do has been defined in the software code of the three music-making software titles as a part of the implementation of these titles. There are no grounds for claiming that music-making software titles have game state rules.

In music-making software, the closest thing to rules restricting actions are the restrictions in the features. For instance, the pattern sequencer “hyper generator” in *eJay Clubworld* is monophonic. One consequence of this is that it is not possible to program patterns with chords in *eJay Clubworld* (whereas patterns with chords can be programmed in *MTV Music Generator 2*). Rather than something defined by a rule, this should be interpreted as a characteristic of this feature or as a technical restriction.

Information rules “[…] govern what information gets passed to each player about the game state; determining whether it is a game of perfect or imperfect information” (ibid.:

67). *100% Star* is one example of a game with imperfect information because it is not explained to the player what choices made by him influence the success of his protégé. It is a part of playing this game to attempt to figure this out. In music-making software, information is not deliberately hidden from the music-maker.

5.2.3 Objectives

The objectives define what the player sets out to do in the game. The objective acts as a motivational force for the player (Wolf 2001e: 115).

Fullerton et al. define ten categories of game objectives. These are ‘capture’, ‘chase’, ‘race’, ‘alignment’, ‘rescue’ or ‘escape’, ‘forbidden act’, ‘construction’, ‘exploration’, ‘solution’, and ‘outwit’ (Fullerton et al. 2004: 53-60 adapted from Redl et al. and Parlett). This set of objective categories has limitations. For example, it does not include evidently the objective in rhythm games like *Gitaroo Man* and *Amplitude* (however, ‘capture’ could be interpreted as the objective in these games).

Goals are related to objectives. For instance, the goal of a game may be to find an object, and thus the objective may be exploration. Some games have a single goal, and other games have multiple goals, some of which may be sub-goals that have to be achieved before the player can attempt to reach the main goal. Sub-goals may also be more loosely connected to the main goal. In many games, the main goal is to achieve victory and thus bring the game session to an end in a way favorable to the player. However, achieving the main goal does not necessarily imply the end of the game. If a video game is fun to play, it will probably be played again even if the player has completed it by achieving the final goal.

Costikyan argues that when starting a game, players commit to behave as if achieving the goals of the game is important to them. Thus, the goals defined by the game guide the player’s behavior. (Costikyan 2002: 12.) Goals defined by the game are said to be intrinsic. However, there may be players who ignore the intrinsic goals and are satisfied with exploring the world of the game, possibly coming up with their own goals in the process.

In many games, reaching an explicit win-state condition is defined as the main goal (ibid.). Reaching that state may occur when the player does something that can be unequivocally measured by the game system. For instance, this could be performing a certain task as fast as possible, collecting the greatest number of certain objects, performing some task the most times with the resources available, etc. In some games there is no victory condition. For example, in *Pac-Man* there is simply an end condition that is reached when all lives have been lost (Björk & Holopainen 2005: 19).

One way to look at video games is to interpret the main goal as the core and that everything else is in the game to make the pursuit of that goal a struggle. In some video games there are things that either assist the player to achieve the goal or hinder the player from reaching the goal.

Some video games present several goals for the player to pursue, or the goals may change during the course of the game. For example, in *WarioWare, Inc.: Mega Microgame$*, the player must figure out by trial and error, or draw on previous experience to discover what the goal is in each mini-game and how to achieve it. In a similar way, the music-maker may approach music-making software by figuring out how to use the features, and what to do with them.

The objective of *Beatmania* is to repeat certain patterns as closely as possible. The reward for doing this correctly is points that measure how well the player has done in reaching the intrinsic goal of *Beatmania*, which is to get a score as high as possible. This objective could be labeled "call and repeat", rather than the musical performance concept of "call and response" because there is no room for variation in the "response" (in fact, making variations is penalized). In *Beatmania*, the player can trigger samples at any time he chooses, however, this is not the objective of the game because the player gets points only when pressing specific buttons at specific times, thus contributing to reaching the intrinsic goal of *Beatmania*. The game considers valid only nearly exact repetitions because such quantitative data is easy for the software to measure. The objective is similar in *Amplitude* with the difference that the player can select each time from a couple of alternative patterns (tracks with different musical material) the one he wants to attempt to repeat and unmute.

In *Gitaroo Man*, the objective is to defeat the opponents by repeating sounds by pressing correct buttons at correct times, and by following the pitch of some notes with the thumb-stick. These actions are defined as "charge" phase (for gaining power), "battle" phase with alternate "attack" and "guard" modes, and "final" phase for finishing off the opponent. Failing to repeat sounds and following their pitches exhausts the power of the player-character. Succeeding in this exhausts the opponent's power. The objective is to defeat all opponents.

The objective in *SingStar* is to match an existing framework (the timing and pitch of a melody) by singing. The objective is somewhat simpler in *Get on da Mic*, where the player must rap and match only the timing of the lyrics. The objective in *100% Star* is to manage resources (where to invest money and within what time-frame) and to get a pop band to the top of the charts. In *100% Star*, making new music is a byproduct of the player's pursuit of this goal.

As mentioned earlier, Fullerton et al. list ten categories of game objectives: Capture, chase, race, alignment, rescue or escape, forbidden act, construction, exploration, solution, and outwit (Fullerton et al. 2004: 53-60 adapted from Redl et al. and Parlett). How do these objectives apply to making music with music-making software?

Objectives that cannot be found in music-making software are capture, chase, race, rescue or escape, forbidden act, and outwit. Potential overlapping objectives include alignment, solutions, exploration, and construction, which I examine next in more detail.

Fullerton et al. (ibid.: 55) define the objective of 'alignment' as to "[…] arrange your game pieces in a certain spatial configuration or create conceptual alignment between categories of pieces." In a sense, there are similar mechanics in music-making software, where sample files are placed in spatial configurations (that form pieces of music) through icons representing them in the multitrack window. However, in music-making software there are no configurations defined by the software according to which the sample files should be aligned.

Could ‘solution’ be an objective in music-making software in the sense of looking for a solution to how to make a piece of music by using the available sample files and features? According to Fullerton et al. (ibid.: 58), ‘solution’ games are about solving a problem or puzzle sooner or more accurately than the competing players. Making music with music-making software does not involve striving for a solution like solving a puzzle, because there cannot be one single solution to the “problem” of making a piece of music, or of how to exploit the potential of the software.

According to Fullerton et al. (ibid.), ‘exploration’ games are about exploring areas represented in the game. Exploration may be combined with other objectives. Making music with music-making software could be interpreted to involve exploring the virtual space of the software and its potential for making music, that is, exploring the features in the software by finding out how to use them, what sample files there are, and so forth. There are no competitive objectives combined to this exploration. Hence, the exploration of the virtual space of music-making software may appear aimless.

The purpose of music-making software is to make music (or music videos) because this is essentially what their features are for. This purpose is similar to the objective of construction that is defined by Fullerton et al. to: “[…] build, maintain, or manage objects; this may be within a directly competitive environment. […] Games with a construction objective often make use of resource management or trading as a core gameplay element.” (Ibid.) On the level of abstraction, something is indeed constructed in music-making software: Pieces of music are built through discrete components looking like blocks (visual representations of sample files or patterns).

There is some similarity between a video game with a construction objective such as *Theme Park World* and music-making software: Something is constructed (a theme park with rollercoasters or a piece of music) and then the player or music-maker experiences this construction (rides the roller coaster through a 3D visual representation or listens through the piece of music). The important difference is that in *Theme Park World* the park requires management, that is, constant input from the player in order to keep it in a good shape. It is also necessary to construct new attractions in order to continue to draw customers, who bring the money needed to maintain the park. Such constant input is not required in music-making software in order to keep the song from changing or deteriorating in any way. There is no reason intrinsic to music-making software to keep constructing (extending or altering) a piece of music. In *Theme Park World*, the player has a limited amount of money to allocate to different tasks, so money is a resource to be managed. There are no limits in music-making software to what can be done with the available sample files and the other features. Hence, there is no resource management in music-making software.

Some video games do not define explicit goals. For instance, role-playing games such as *Ultima Online* and *EverQuest* allow a number of goals to be set, and players choose from these the goals that most attract them. (Costikyan 2002: 12-14.) Of the music-themed games, *Fluid* has player-definable goals because it is up to the player to choose what kind of music to make. The same is true in the remix mode of *Amplitude*.

Costikyan (ibid.: 13) considers *SimCity* a good toy because it provides players with a wide choice of goals. In a way similar to *SimCity* and role-playing games like *EverQuest*, it is up to the music-maker to define his goals: What kind of music to make with music-making software. As the making of the song progresses, the initial goal of making a certain type of music may change. Such a change of self-defined goals is also possible, for instance, in *SimCity*. A person making music with music-making software or being engaged in *SimCity* is not rewarded in any way by the software if he reaches the goal he has set himself. In other words, such goals are not intrinsic to the software.

The similarity between video games without explicit intrinsic goals and music-making software is that the latter allows the music-maker to set certain goals: To choose what type of music to make. Such goals emerge by making use of the features of the software.

With regard to goals, the important difference between video games and music-making software is that the goals are not defined by the software. Music-making software titles do not define what kind of music should be made with them (albeit the style of the included sample files do to some degree influence the music made with them), nor does it observe what the music-maker does, and does not reward or penalize the music-maker for doing or not doing certain things.

5.2.4 Outcome

In many games, the outcome of a game session is measurable and unequal: Some players win and others lose. The outcome may be measured, for instance, by a score, the statistics of the players' actions, examining how many of the sub-goals the players have achieved, or how many of the objectives have been completed. (Fullerton et al. 2004: 79.)

Juul (2003: 32-34) defines the outcome of a game as variable, quantifiable, and valued. A game is interesting when its outcome cannot be known beforehand, in other words, when the outcome is variable. The variability of the outcome depends partly on who plays the game (ibid.: 37). Juul (ibid.: 33) notes that "[i]f players always achieve a draw or if a master player plays his/her best against a beginner, it does not really work as a game *activity*." In other words, there may be little point in playing if the outcome is known for certain beforehand. Games have features such as handicaps to even out skill differences and ensure that the outcome is variable. (Ibid.)

Quantifiable outcomes are designed to be beyond discussion and ambiguity (ibid.: 34). If the actions of the player involve qualitative aspects, these have to be transformed into a quantifiable form in order for the outcome to be unambiguous. This is the case, for example, in some sports, which have judges to transform qualitative performances into a quantitative outcome (ibid.: 34 fn. 12).

With valorization of outcome, Juul means that some of the possible outcomes are defined by the game as better than others. Outcomes may be positive or negative, and positive outcomes are typically harder to achieve. The valorization of outcome can be done by statement, by instructions, by assigning some actions a higher score than others, by defining only one way of progressing and making something happen, or implicit from the setup of the game. (Ibid.: 34.)

Juul (ibid.: 34-35) notes that players are attached to the outcome:

> The attachment of the player to the outcome is a psychological feature of the game activity. A player may actually feel happy if he/she wins, and actually unhappy if he/she looses. Curiously, this is not just related to player effort: a player may still feel happy when winning a game of pure chance.

The outcome of playing through one level of *Amplitude* is a special mix of a piece of music, which can be stored and replayed without need for further actions in the game. The outcome from one game session of *Amplitude* is also a score. Because *Amplitude* keeps track of the score, the outcome of one game session can be compared to all other game sessions as superior, inferior, or equal. Thus, the outcome is measurable and valued. The outcomes of the special mixes cannot be compared in this unambiguous way.

The outcomes of making music with music-making software are variable in the sense that it is unlikely that a music-maker could envision beforehand precisely how a piece of music will turn out and how it will sound. The outcomes of using music-making software are not quantifiable or valued because there are no winners, losers, or scores. This music-making activity has no distinct ending. Music-making software titles do not keep score, measure the outcomes of the music-making activity (the pieces of music), or have predefined end conditions the music-maker should strive to achieve. A piece of music is complete when the music-maker or the people around him so decide. A piece of music may be superior or inferior to another only if the music-maker or the people around him consider so. The software does not make this judgment of assigning values to the possible outcomes of music-making.

5.2.5 Procedures

Procedures are the methods of play and the actions a player takes to achieve the objectives of the game. Fullerton et al. describe procedures as “who does what, where, when, and how”. Games have procedures such as starting actions, ongoing procedures, special actions whose availability is conditional on other elements or the game state, and resolving actions which bring a game session to an end. Video games also have system procedures that define what the game software does “when and how”. (Fullerton et al. 2004: 60-64.)

What Fullerton et al. define as procedures are to some extent similar to player activity as defined by Wolf and Perron, who consider ‘player activity’ as the core of the video game playing experience. Games require player activity, and little will happen in a video game if the players do nothing. Player activity has mental and physical aspects. (Wolf & Perron 2003: 15.) Player activity manifests as player actions.

Every video game has an interface to its procedures. Wolf and Perron consider the interface as a junction point between the player and the game. Wolf and Perron describe the interface as a portal through which player activity takes place. Player activity takes place as input to the game through an interface, as a continuous response to what the player sees on the screen and hears from the loudspeakers. The dimensions of player activity are to some degree confined and quantified by the interface. (Ibid.) The physical constraints of the interface shape the procedures (Fullerton et al. 2004: 64).

The interface to a video game includes input devices (e.g., game controller, alphanumeric keyboard, microphone, and video camera), output devices (e.g., screen and loudspeakers) and onscreen graphical elements (e.g., characters, a pointer, and UI elements like sliders and buttons), which invite and enable player activity (Wolf & Perron 2003: 15).

The same input and output devices for PlayStation 2 are used for playing games and making music with music-making software. A notable exception is *Beatmania*, which can be played with a special controller. When *MTV Music Generator 2* is used with the USB sampling device, a microphone can be a part of the music-maker's interface just as it is a part of the player's interface in *SingStar* and *Get on da Mic*. As became apparent in Sections 3.3 to 3.5, the DualShock 2 controller is a versatile input device for nonlinear multitrack composing with the three music-making software titles.

When considering procedures, it is essential to distinguish between what the player does physically (player actions) and what the player's agent can do in the world of the game (a part of the game mechanics). Game mechanics is an established concept among game designers and scholars. Fullerton et al. do not discuss game mechanics as a part of the procedures, even if game mechanics can be considered a part of the ongoing procedures. A core mechanic is the mechanism through which players make choices in the game (Salen & Zimmerman 2004: 317).

The ongoing procedures (player actions) in *Beatmania* are the following: The player must press the correct button or rotate the turntable disc on the controller within a short time frame indicated on the screen. If the player succeeds in doing this, he is rewarded with some points and the music sounds good because the player triggered a sample file in rhythm with the music. The game mechanic in *Beatmania* is that a sample file is represented by a falling block (called "note") that is "caught" by pressing the correct button at the correct time. When the player presses the correct button at the correct time, the block disappears from the screen, the sample file is triggered, and the player gets some points. This is repeated over and over again until the player has made too many mistakes or the song ends.

In *SingStar*, the player must sing the correct notes into a microphone within the framework of a predefined melody of a certain piece of music. The notes that the player sings are shown as graphics on the screen on top of the graphics of the melody the player is supposed to sing. The game mechanic is to match the melody (represented by the graphics) through singing. In other words, the player action is singing. Although the player is supposed to sing certain lyrics as well, the game apparently recognizes only the pitch and timing of whatever the player sings. The game mechanic is similar that in *Gitaroo*

Man, even if the player actions are different (singing vis-à-vis pressing buttons and matching pitch with the thumb-stick). However, in *Gitaroo Man* repeating the patterns correctly also causes damage to the opponent and failing to do so causes damage to the player-character (a loss in the resource of health). This way, the actions the player makes through the game mechanics of *Gitaroo Man* are connected to the resource of health (preventing the player-character from losing power), progression (beating the opponent and getting to the next level), and score (valorized outcome).

In real-time musical performance, there is the framework of an existing piece of music. In improvisational performance, too, there is some framework within which the performance must fit. The musical performance is compared to this framework and it may be “faithful”, “innovative”, have mistakes, and so forth. The patterns in rhythm games offer a similar point of comparison, so the outcomes of playing rhythm games are quantifiable and valorized. There are no such points of comparison and hence no measurable outcome in music-making software.

The music-making actions taken with music-making software can be defined on the basis of the five-phase nonlinear multitrack composing process as selecting or recording sample files, modifying them, arranging them, and mixing them into a piece of music. Game mechanics corresponding exactly to these actions cannot be found in the music-themed video games, with the exception of *Fluid* and *100% Star*, which involve selecting and arranging sample files. However, there are similarities to game mechanics in video games with the objective of construction, that is, to select and place prefabricated components on the screen one-by-one, and to manipulate their properties one at a time, and ultimately build immaterial objects from these components.

5.2.6 Resources

In video games, resources are assets such as natural and economic resources, which are exploited in order to accomplish the goals of the game. Resources have utility and scarcity. One aspect of playing a game is to consider and manage the resources, because the resources are the means to achieve the goals. Fullerton et al. define the following resources as formal elements of video games: ‘lives’, ‘units’, ‘health’, ‘currency’, ‘actions’, ‘objects’, ‘terrain’, and ‘time’. (Fullerton et al. 2004: 68-74.) Next, I examine how resources influence music-making in the music-themed video games.

The resource of currency enables music-making in *100% Star*. In *Get on da Mic*, the player can use money to buy clothes and other accessories for his character. Currency is not a resource in *Get on da Mic*, because it does not have a utility related to achieving the goal of the game (to score as high as possible), and has no influence on the music-making in the game. Instead, it is a way of keeping score.

A game session of *Gitaroo Man* ends when the player-character has lost all power after taking too many hits. In other words, health is a resource that increases when the player makes music by pressing the correct buttons at the correct times, and decreases when the player fails to do so. The player can continue making music as long as the player-character has any energy left.

Fullerton et al. (2004: 69) define 'units' as follows:

> In games in which the player is represented in the game by more than one object at a time, you generally have unit resources to manage […]. Units may be all of one kind, as in checkers, or a number of different types, as in chess. Units may keep the same values throughout the game, or they may upgrade or evolve, as in real-time strategy games. Units may be finite […], or they may be renewable, as in games that allow players to build new units over time. When units are renewable, they often have an associated cost per unit.

Sample files in *Magix Music Maker* and *eJay Clubworld* could be interpreted as one type of unit. Each sample file is a unit with values or properties that can be modified. There is no limitation (like associated cost) to how much the properties can be modified. Pieces of music are made from this unit and each unit has different musical characteristics. The same can be said of patterns in *MTV Music Generator 2,* which is a type of higher level unit assembled from the unit of sample files.

An object can be considered a resource when it has utility and scarcity. Objects enable the player to accomplish the objectives of the game. (Ibid.: 72.) In *Amplitude*, objects include the special features (like "freestyler" and "score doubler") the player has to collect before they can be used. In *100% Star*, objects include the prefabricated songs, record cover art models, etc. Because the goal in music-making software is to make music, sample files and the other features could be considered as resources for achieving this goal. However, these are in no way restricted in music-making software. Hence, they cannot be considered as resources the way there are resources in video games.

Actions, including moves and turns, are resources when there are limitations to how they can be used in a game. When actions are resources, players must plan the use of the actions, moves, and turns carefully in order not to waste them. (Ibid.: 71-72.) Actions are not a resource in music-making software because the music-maker has all the features (actions) infinitely available.

Resources may be bound to certain locations in the virtual world of a video game. These include the resource of terrain that can be taken into use only when the player's agent occupies that part of the game world. (Ibid.: 73.) Jenkins and Squire (2002: 64) consider video games as 'contested spaces', suggesting that the conflict between players is over the virtual space.[54]

Terrain as a resource cannot be equated with the environment in games such as *Gitaroo Man*. The environment is a backdrop where the events of the game occur. If the environment in *Gitaroo Man* could be used, for instance, as an eroding shield with which to take cover, it would be a resource. Terrain has some use for the player in the struggle to reach the goal of the game, while an environment is a backdrop that may contribute to the story of the game or communicate the theme of the game.

---

[54] For an overview on designing virtual worlds for video games, see Bartle (2004).

When time is limited, it becomes a resource. Wolf (2001d: 89) defines two types of timers used in video games, namely ‘hourglass’ and ‘stopwatch’. In a game with an hourglass timer, the player must do as much possible of the given tasks before running out of time. The player may be rewarded with some additional time to the hourglass by accomplishing certain tasks. In a video game with a stopwatch timer, the player must attempt to complete certain tasks in as little time as possible. Time is not a resource in the case of the stopwatch timer, but a way to quantify one aspect of the performance of the player during the course of a game session.

In *Amplitude*, music-making in one game session lasts until the player’s “energy” diminishes completely. In *Amplitude,* “energy” is essentially the same as the resource of health as defined by Fullerton et al (2004: 70). Energy diminishes over time if the player continuously fails to repeat the patterns. Hence, the resource of time can also be found in *Amplitude*. These two resources enable the music-making. In addition, certain features that also affect music-making (including “slo-mo” and “freestyler”) are resources because they can be used only for a limited time or a limited number of times.

The resources of lives, health, and currency obviously cannot be found in music-making software. Time is not a resource in music-making software because there is neither a stopwatch nor an hourglass timer. The resource of units can be found in music-making software, as sample files or patterns are the basic building blocks of a piece of music.

In Section 3.7, I observed that the virtual spaces of the three music-making software titles are adjacent spaces displayed one at a time (Wolf 2001c). Music-making software titles do not present virtual worlds like, for instance, the worlds in *Fluid*, which are three-dimensional environments, but they are not *interactive* three-dimensional environments (ibid.) because the player’s actions do not change the contents of the world. There are some attempts in *eJay Clubworld* to represent the clubs as a virtual space the music-maker operates in. This is done, albeit unconvincingly, with the video clips shown before a club is being loaded to the game console. At best, these video clips suggest the “mood” of each club and contribute to the theme of deejaying in clubs. The resource of terrain cannot be found in music-making software.

Music-making software titles do not have resources in the sense video games do. There are no resources in *Fluid,* either, because sample files and the other features can be used without restrictions. In *Fluid*, the sample files are just grouped into different levels. The player has to keep the dolphin swimming for some time in order to get to new levels with new sets of sample files. There is no proper struggle in Fluid to obtain new sets of sample files.

5.2.7 Conflict

Several writers argue that conflict is at the heart of games. For example, according to Fullerton et al. (2004: 107) a game entertains its participants by “[…] creating a structured conflict and providing an entertaining process for players to resolve that conflict.” Salen and Zimmerman (2004: 80) define games as systems in which players engage in an artificial conflict. I noted already earlier that this conflict is artificial because it is created for the purpose of players attempting to resolve it during the course of the game.

Conflict is a key element in many games. Resolving the conflict in a video game requires actions and decisions on the part of the player. The attempts the player makes to resolve the conflict partly cause the dramatic element of challenge (see Section 5.3.4). Conflict can be set up in many ways, for example, through a direct competition between players that resolves in victory or defeat. Conflict requires a resolution such as complete or partial victory, or defeat. If the outcome of a game session is undecided, the conflict has not been resolved. The conflict is often reinstalled at the beginning of a new game session. Thus, each resolution to the conflict is only temporary as long as the player is committed to play the game again.

Fullerton et al. (2004: 74-75) define three sources of conflict in games, namely ‘obstacles’, ‘opponents’, and ‘dilemmas’. Obstacles may be set up by the other players during the course of the game, or they may exist by design in the world of the game. A game may unfold as a path from one obstacle to the next. The distinction between obstacles and opponents may not always be clear-cut. Because opponents may be characters controlled by artificial intelligence, they can be interpreted as obstacles the player must overcome through defeating them. The conflict in a game may be over the resources instead of a direct conflict between the players.

Dilemmas are about making choices, such as about choosing the lesser of two (or more) evils. According to Costikyan, the player considers the situation the game is in every moment, including the objectives, the available resources, and the opposition. The player tries to decide the best course of action to achieve the goals. (Costikyan 2002: 12.) For instance, by making a sacrifice, the player may gain something in the future during the course of the game.

The conflict in some games is explained to the players through the dramatic element of story, while in certain video game genres, such as shoot ‘em up, the conflict is a convention so there is no need to explain its reasons. For instance, people playing *Space Invaders* do not really need to know the reason for the invasion in order to commit to fight back. The same is true of many sport games, in which conflict in the form of competition is a convention.

In *SingStar* and *Get on da Mic,* conflict emerges from the difference between what the player is supposed to sing or rap (the example of the existing songs), and how the game software rates this performance. In *Beatmania* (two-player mode) and *Gitaroo Man*, the conflict is presented as a direct conflict between the player and the opponent. In *Gitaroo Man* each instance of conflict (fighting each opponent) is resolved through music-making actions.

In the single-player mode of *Amplitude*, the main source for conflict is that of avoiding running out of energy in the game world. Playing *Amplitude* also involves the conflict of continuously facing the dilemma of what track to choose from those available, and there is very little time to make these decisions. The choice may be based on the musical outcome the player is pursuing. Occasionally, *Amplitude* presents some tracks as more desirable than others by offering a bonus award for correctly repeating the pattern in those tracks, thus influencing player behavior.

If we want to find dilemmas in music-making software, then they are about the choices made while making a piece of music: Selecting the appropriate sample files, effects, mixing levels, and such. Nevertheless, these "dilemmas" are not connected to resources as they are in many video games. There are no intentional obstacles inherent in music-making software. Obstacles arise in music-making software when the music-maker is for some reason unable to use certain features. There are no opponents in music-making software because there are, for instance, no resources for which to compete.

5.2.8 Players

When defining 'players' as a formal element in video games, Fullerton et al. refer to different interaction patterns between players and the game. The interaction patterns of a video game have been defined by game designers and have been programmed into the game software. Fullerton et al. define seven player interaction patterns: Single player versus game, multiple individuals versus game, player versus player, unilateral competition, multilateral competition, cooperative play, and team competition. (Fullerton et al. 2004: 44-49 adapted from Avedon.)

Most of these interaction patterns define conflict or competition: This *against* that. As I noted earlier, there is no competition in music-making software. Hence, no similarities to player interaction patterns can be found in music-making software. Cooperative play could be one option in music-making software because theoretically several people could participate in the music-making. However, in practice the graphical user interfaces of music-making software titles are designed for a single music-maker because, for example, there is only a single pointer and selections can be made from only one window or menu at a time. In the jam modes in *MTV Music Generator 2* and *eJay Clubworld*, several people can co-operatively trigger samples in real-time with several game controllers, but this is not nonlinear multitrack composing and is thus outside the scope of this study.

5.2.9 Summary of Formal Elements

The similarities and differences between the formal elements of video games and music-making software can be summarized as follows. There are certain similarities between the formal elements of boundary, objectives, and procedures. Both video game players and music-makers cross a boundary, either to the special worlds of games or to nonlinear multitrack composition.

There are similarities with nonlinear multitrack composition and video game objectives of construction and exploration. However, there are no video games with exactly the same procedures as in the three music-making software titles (e.g., selecting, sampling, modifying, arranging, and mixing). *Fluid* and *100% Star* feature selecting and arranging sample files into pieces of music.

The following formal elements of video games cannot be found in music-making software: Conflict, players, rules, resources, and quantifiable outcome. The outcomes of making music with music-making software are variable, but not because of rules stating so. Instead, the variable outcomes in music-making software are a result of the actions of the music-maker and the great number of possible combinations that can be constructed from the materials and features in each music-making software title. I am not stating the obvious here (that each piece of music is different), but suggesting that there might be an excitement of uncertainty when beginning to make a piece of music similar to that experienced when commencing to play a new game level. In making music with music-making software, the outcome may be valorized by the music-maker, but not by the software.

Music-making software titles do not have resources in the sense that video games do. The management of resources is one aspect of playing a game, as resources are a means for achieving the goals of the game.

Making music in music-making software does not appear as a conflict for the music-maker to resolve the way games present conflicts for the player to solve.

The formal elements in themselves do not make a software into a game. It is how these elements are structured into a system that renders them into a game when this system is put in motion through input from a player. In the next section, I examine the dramatic elements that contribute to giving the formal elements meanings for players.

5.3 Similarities and Differences in Dramatic Elements

Fullerton et al. (2004: 30-31) argue that while some players may be engaged by abstract challenges, most players need something else to draw them to the game. Some video games use dramatic elements for engaging players with the formal systems of the game (ibid.: 91). Dramatic elements allow players to make emotional connections with the game (ibid.: 30-31). Fullerton et al. (ibid.: 30-35, 81-103) define the following dramatic elements: ‘premise’, ‘character’, ‘story’, ‘challenge’, and ‘player types’. Next, I examine music-making software through these dramatic elements. I summarize the findings in Section 5.3.6.

5.3.1 Premise

According to Fullerton et al. (2004: 91), premise “[…] establishes the action of the game within a setting or metaphor.” Premise enables players to become emotionally engaged in the game (ibid.). Premise concretizes the abstract system concepts of a video game, and makes a game emotionally appealing (ibid.: 93).

For example, the premise in *100% Star* is to “[…] take your own customized band to number one […]” (text from the user’s guide). In *Get on da Mic*, it is to “[r]ap your way from the street to the stage […]" (text from the product package). These premises are linked to music-making.

Introduction videos shown when starting *MTV Music Generator 2* (with deejay David Morales) and *eJay Clubworld* (with deejay Carl Cox) could be considered as contributing to the premise. In a sense, they are similar introductions to the theme and subsequent actions as are movie clips in video games like *Gitaroo Man*. The claims in the product packages of the three music-making software titles could be interpreted as suggestions on the premise:

- "MAGIX music maker is pure creativity. […] Become the greatest producer, director, DJ, and remixer in the universe. […] 100% FUN" (*Magix Music Maker*)
- "Prepare for sound. Prepare for music. Prepare to generate. Whether you're a superstar DJ or just starting out, MTV Music Generator 2 is all you need to create killer tunes in minutes." (*MTV Music Generator 2*)
- "Make your own hit without any musical training in a matter of minutes. Eight real-world clubs and locations are waiting for you: each with a different and current musical style for your creativity to explore." (*eJay Clubworld*)

These exaggerated premises promise creativity, and also "fun" in the case of *Magix Music Maker*, rather than challenge or playfulness. In particular *MTV Music Generator 2* and *eJay Clubworld* are presented more like quick-and-easy-to-use[55] music-making tools than as video games. In fact, the product package of *Magix Music Maker* also states that it is "More than a game!"

5.3.2 Character

According to Fullerton et al. (2004: 94), characters in video games "[…] are agents through whose actions a drama is told. By identifying with a character and the outcome of their goals, the audience internalizes the story's events and empathizes with its movement toward resolution."

Player-characters are the player's agents in the world of a game. In many video games, the player directly controls one or more characters. To name a straightforward example, in *Pac-Man* the player-character moves in the direction towards which the player pushes the joystick. In *Amplitude*, the agent is a spaceship, even if it is suggested that the player controls a human-shaped FreQ character (whose appearance can be defined to some degree by the player). It is through the spaceship that the player interacts with the game world, and not through the FreQ character. The FreQ character seems just an elaborate display of what type of track the player has chosen. The FreQ character renders the music-making slightly more concrete and involves the player perhaps a little more in

[55] For a discussion on the discourse of "easy-to-use" musical instruments, see Théberge (1997: 20-40).

the music-making. In the online mode of *Amplitude*, the FreQ character represents a player's actions to other players.

There are no agents like player-characters in music-making software. However, there are games such as *Beatmania* and *100% Star*, where the agents are not so obviously characters.

5.3.3 Story

A video game may begin by telling a backstory (which the player's actions cannot affect) and then a story unfolds as the player progresses in the game. Backstory gives a setting and context for the conflict in game. It explains the motivation of the characters' actions. (Fullerton et al. 2004: 97.) The backstory may also explain the premise of the game to the player. In video games based on movies, the story and action in the game may draw on the story and the events in the movie.

One way to tell the story is to insert animated story chapters between game levels. Such story chapters can explain events in past game levels (the consequences of the player's actions) and provide motivation for upcoming game levels (by setting up a new conflict or proclaiming a sub-goal). The progression of the story may be influenced by the choices the player has made during the course of the game. (Ibid.)

The story can be linked to the theme of the game. This is the case in *Gitaroo Man*, where the story progresses through cut-scenes between the game levels. The story of *Gitaroo Man* involves the player-character discovering in himself the hidden skill of making music. In career mode, *Get on da Mic* presents the story of a struggling hip hop artist attempting to become successful in the world of the game. However, a story is not essential to games, and not all video games feature a story. This is the case with *Amplitude*, *Beatmania*, and *Fluid*. There is no story in any of the three music-making software titles.

5.3.4 Challenge

Fullerton et al. (2004: 83) suggest that the challenge in video games invites players to use their physical, mental, social, and other skills. Costikyan argues that challenge is an essential element in all games. Players want the games to challenge them and playing a game should be *a struggle* towards a goal. Games may not be fun if they are not challenging because players may not feel any sense of accomplishment if they achieve the goals too easily. (Costikyan 2002: 14-17.)

According to Costikyan (ibid.: 12), "[t]he basic transaction we make with games is to agree to behave as if achieving victory is important, to let the objective guide our behavior in the game. There's little point, after all, in playing a game without making that basic commitment." In other words, Costikyan argues that a player makes a commitment to struggle to pursue the objective (goal) of the game. To do this, they must be somehow motivated to commit themselves to the challenge of the game. Dramatic elements motivate players to make that commitment.

Struggle may occur through a direct competition with other players, or when the player tries to overcome the obstacles posed by the game (ibid.: 15).

The element of challenge is easy to pinpoint in some video games. For example, *Gitaroo Man* offers a lot of challenge in the sense that in the game it is quite difficult to beat even the first opponents. In comparison, *Fluid* contains very little challenge because there is no competition over a resource, and there are no time limits. The only challenge in *Fluid* is to decide what kind of music to make.

Do music-makers make a similar commitment with music-making software as video games players do to pursue the goals of the game? Music-makers obviously invest time and effort in this activity.

The objective or goal in music-making software is probably more the music-making activity itself, rather than the pursuit to completing a piece of music. Of course, such a goal cannot be reached unequivocally. It is difficult to define when a piece of music is complete. The challenge music-making software offers is a process rather than a sequence of sub-goals (in the sense of, for instance, first making the rhythmic section, then the supporting section, and so forth).

Video games reinforce a player’s commitment by offering sub-goals the player can achieve by investing some effort and then giving the player rewards for so doing. Sub-goals make the main goal appear achievable. Music-making software titles do not offer rewards throughout the music-making process the way video games do to sustain interest over time.

Challenge is not intentionally programmed into music-making software. Regardless of whether we interpret music-making software as some type of game or as tools for making music, using their features should be as effortless as possible. Of course, in practice there may be various hindrances, if not obstacles in the sense of a formal element. Such hindrances are caused, for instance, by the music-makers’ limited skills, a confusing GUI, or inconsistent terminology. Such hindrances (or more precisely, usability issues) are not implemented intentionally to the software the way obstacles in video games are.

5.3.5 Player Types

The potential for play engages players emotionally in games (Fullerton et al. 2004: 87). Fullerton et al. (ibid.: 90) discuss ten potential player types, and argue that each player type has different needs and agendas. The player types are ‘competitor’, ‘collector’, ‘achiever’, ‘joker’, ‘director’, ‘storyteller’, ‘explorer’, ‘artist’, ‘performer’, and ‘craftsman’.

Certain games are designed to encourage specific types of play (ibid.). Probably very few games can provide something for all player types. Hence, game designers must choose what player needs and agendas they want to appeal to with the game they are designing. In the following, I attempt to answer the following question: What types of player needs and agendas could making music with music-making software titles fulfill?

Of the potential player types Fullerton et al. (ibid.) discuss, the following two cannot be considered to apply to music-makers using music-making software: Competitor (as I discussed earlier) and collector (because there is nothing to collect, and the sample files are by default organized into sets).

An explorer is curious and seeks boundaries in the world of the game (ibid.). In music-making software, exploration could involve finding out what the features can be used for, rather than exploring the limits of the virtual spaces of the software titles (because there is no challenge to find those limits).

A player of the artist type is driven by creativity, creation, and design (ibid.). These aspirations definitely appear to be a part of the agenda for using music-making software. In comparison, in *100% Star*, the player can make music from prefabricated sample files, design the physical appearance of the pop stars, and design the appearance and content of the singles and albums released in the world of the game. The music, the physical appearance of the player's protégé and the contents of the released music may or may not assist in some way in achieving the main goal of *100% Star*, which is to get the protégé to the top of the charts. In video games, the agenda of an artist may be an addition to a more competitive agenda, just as it is in *100% Star*. In music-making software, there is no competitive agenda combined with the artistic agenda.

A craftsman wants to build, craft, or puzzle things out (ibid.). As discussed in the case of video games with the construction objective, fulfilling these aspirations is possible to some degree with music-making software.

A performer likes to put on a show for other people (ibid.). This could apply to aspects of making music with music-making software in real-time, for example, mixing. In other words, a player with the agenda of a performer would present his music-making as a show perhaps in some ways similar to the real-time performance of a deejay.

A storyteller likes to create worlds of fantasy and imagination, and then to live in those worlds when playing (ibid.). There is no opportunity for this in music-making software.

A player of the joker type does not take the game seriously, and plays for the fun of playing rather, for instance, than for striving for victory (ibid.). A music-maker with this agenda could be interested to try the features of music-making software without the aim of completing a piece of music.

A director wants to be in charge and to direct the play of other human players (ibid.). This is obviously not possible in music-making software.

An achiever plays for the purpose of reaching new levels of achievement (ibid.). The achievements may be represented in different ways in games, for instance through level, experience points, or military ranks. In *Gitaroo Man* and *Amplitude*, getting to a new level is rewarded with a new piece of music. An achiever could use music-making software to learn multitrack composing techniques or for accomplishments in making new pieces of music. However, music-making software titles do not keep track of such achievements nor acknowledge them.

5.3.6 Summary of Dramatic Elements

Making music with music-making software titles bears similarities to some of the dramatic elements of video games defined by Fullerton et al. (2004). These include the player types of explorer, artist, craftsman, performer, and possibly achiever. However, the similarities are not crucial because many other activities can also fulfill these needs and agendas. The absence of the dramatic elements of character, story, dramatic arc, and intrinsic challenge from the activity of making music with music-making software is more significant, because their absence suggests that music-making software titles are better interpreted as tools than as video games.

The challenge in music-making software is different from the challenge in video games. The premise in music-making software involves creativity, rather than a competition or response to a challenge as in games. Music-making software titles do not present the music-maker with any reason or motivation to make a piece of music. Video games have intrinsic ways of motivating players, such as presenting a conflict for the player to resolve, and giving feedback and rewards for playing. In music-making software, all motivation must come from the music-maker himself or as encouragement from people around him.

Fullerton et al. (ibid.: 94) consider the characters in video games as agents through whose actions and tribulations drama is conveyed. There is often a conflict between some of the characters. Fullerton et al. (ibid.: 100-101) write:

> In a game, the rising action is linked to both the formal and dramatic systems. This is because games are usually designed to provide more challenge as they progress. Games that also have well-integrated dramatic elements will intertwine those elements with the formal system so that as the challenge rises, the story develops. […] Once the conflict is set in motion, it must escalate for the drama to be effective. Escalating conflict creates tension, and in most stories, the tension in a story gets worse before it gets better, resulting in a classic dramatic arc.

Meaningful conflict is designed to keep players from accomplishing their goals too easily and to draw the player into the game emotionally by creating a sense of tension regarding the outcome. Conflict is one way to introduce drama into a game because it forces the player to take action. (Ibid.) I have already acknowledged the absence of conflict in music-making software.

The dramatic arc can be a tool for game design, that is, one way for game designers to structure the intensity of action throughout the game. When the events of a game follow the classic dramatic arc there is an exposition (introduction), followed by rising action or tension (e.g., increase in difficulty or the amount of action), climax (e.g., confronting the boss monster), and resolution. (Ibid.: 100-101.) In some games, such structuring of action occurs on each game level as well as in the game as a whole. If the player fails in the climax stage, for instance, to defeat the boss monster, it is probable that he will not get the feeling of resolution in the dramatic arc[56] but that the game will just end abruptly. Such dissatisfaction may be a strategy utilized by game designers to motivate the player to try again, rather than to discourage him from playing that game ever again.

In some video games, there is a clear conflict between the player-character and the opponents (ibid.: 102). As I have argued earlier, this is the case in *Gitaroo Man*, where the action of fighting each enemy to some degree follows the dramatic arc. There are alternative charge and battle phases, and a climax phase for finishing off the enemy. The animated clips between opponents act as moments to give the player a breather.

The process of making music with music-making software does not follow a dramatic arc because the music-making activity is not defined by the software to proceed according to a specific order. In addition, as I argued earlier, there are no player-characters whose aspirations and ordeals could evoke drama in music-making software.

Fullerton et al. (2004) do not discuss context, which is an important aspect influencing how players make emotional connections with a game. Game designers can design the player's experiences to some extent, but the experiences of playing a game are only felt in specific contexts. A context is the framework within which, among other things, the players can or cannot make emotional connections to the game. People play video games in different contexts (such as alone or with other people), they have different attitudes to the theme and so forth, and hence their experiences may be very different. Game designers can do little to control the context of each situation in which their games will be played.

To summarize, there are only trivial similarities between the dramatic elements of video games and making music with music-making software.

5.4 Similarities and Differences in Games as Systems and Simulations, and the Different Types of Play

In Section 4.4, I discussed video games as systems and state machines. Juul draws from the work of the computer scientists Bran Selic, Garth Gullekson, and Paul T. Ward when defining the six aspects of a state machine. These are the initial state, the possible input events, the set of possible states that the machine can be in, the state function that maps states and input into states (what happens in response to a given input), the output function that maps states and inputs to outputs (what the machine outputs at a given time), and the output events (what outputs the machine can produce). (Juul 2003: 57.)

On the basis of this definition, music-making software and various software tools can also be interpreted as state machines: The behavior of a music-making software title or a word processor is defined through algorithms determining what output a specific input produces in a specific state. The rules of a video game are implemented in the game software in a similar way through algorithms. In games, rules specify limitations (what cannot be done) and affordances (what can be done) (ibid.: 55). The same could be said

[56] Alternatively, it could be considered that the resolution in this case is a dissatisfying one, in a similar way as in a movie ending in the failure of the protagonist.

of all software as they allow certain things to be done, and do not allow other things to be done. For instance, it is not possible to edit a video clip with word processor software.

According to Juul (ibid.: 58), "[w]hen you play a game, you are interacting with the game state via its possible input events - the input events allow the players to exert effort in order to make the game end with a desired outcome." In many video games, the desired outcome is defined as reaching the final goal of the game. Games are special state machines because of rules that define end conditions. For example, in *Pac-Man* a game session ends when all lives have been lost. The player cannot keep playing even if he wanted to, but has to start a new game session from level one and the score of zero points. This rule brings challenge to the game. There are no end conditions in music-making software, and a music-making session ends only when the music-maker so decides. Music-making software titles do not declare when a piece of music is complete.

In some games, rules define the output function so that the player may not be informed of all aspects of the game states (ibid.: 59). Such imperfect information is not feasible with tools. Music-making software titles mostly feature perfect (complete) information, so the music-maker does not have to make guesses or estimates. One exception is the graphical icons of sample files and patterns in the multitrack view of the three music-making software titles. The sample files and patterns are not represented visually in a way that would facilitate music-making with them. Another exception is that the music-maker has to estimate how different filter settings will modify the sample file. However, these are usability issues rather than instances of imperfect information as in games.

Costikyan (2002: 12, 20) argues that games shape player behavior through defining goals. A change in game state can involve a change in the goals or sub-goals. For example, in *Pac-Man*, when the player has reached a power pill, the roles are reversed for a brief time so that the hunters become the hunted. In some video games, the actions the player can do vary between game states. In *Gitaroo Man*, there are discrete charge, battle, and final modes (game states). A change of mode may lead to a change in sub-goals. In charge mode the player has to fill up his health meter, while in battle mode the player must weaken the opponent's health, and in final mode the objective is to finish off the opponent.

In video games, discrete states (seen by the player as, for instance, different tasks, levels, quests, and missions) can be used to phase the things the player has to consider and the actions he has to take. This way, the player does not have to consider all possible sub-goals and actions simultaneously. In some games, the order of the tasks is predefined so that the player must complete one task before he can move on to the next task.

The important difference between the systems of video games and the systems of music-making software is that video games shape the behavior of the players by defining goals, proclaiming certain actions to be more valuable than others, valorizing outcomes, and presenting certain tasks to accomplish before it is possible to move on to the next task. Music-making software titles do not shape the behavior of the music-makers in these ways. Music-making software titles give music-makers tools and materials they can use as they see fit. No past actions by the music-maker restrict or extend the availability of the features in music-making software because the same features are available at all times.

The music-maker may be attached to the outcomes of music-making, but not because the possible outcomes have been valorized by music-making software. Nevertheless, this does not imply that making music with music-making software could not be some form of play.

Earlier I emphasized that play and game are not the same thing. So, what type of play can making music with music-making software be?

5.4.1 Paidia and Ludus

The game and media researcher Torben Grodal (2003: 193) points out that one problem with the term 'game' is that it is defined loosely in many discussions. This problem became apparent earlier when discussing *SimCity*, which is not a game in the sense that it does not have goals and victory conditions defined by the software. Thus, it is appropriate to ask if *SimCity* is 'played' or 'used'? Are people engaged in *SimCity* 'users', 'players', or perhaps 'gameplayers' or 'gamers'? This question is best approached through the concepts of ludus and paidia, which I discussed in Section 4.3.

Ludus is a form of play with rules that define winners and losers, while paidia is a form of play that does not do this (Frasca 2003: 229-230). In video games of the ludus type, the software acts as a referee by enforcing the rules and proclaiming the winners and the losers. In a ludus type of game the player makes a commitment to strive to become the winner. A game session of a ludus game comes to an end when a winner and a loser have been produced. A ludus game may also come to an end when a resource such as time or lives has been exhausted, and the player's performance during the game session has been measured in some way. Most video games are of the ludus type because they produce explicitly winners and losers, or because one outcome is valorized above another. Score, time, the amount of resources spent, and level are examples of data that allows making comparisons between game sessions. When a video game system keeps track of the score, time, resources spent, and level, one can compete against himself or others.

Video games of the paidia type are more open-ended than ludus games. The ludus type of games provides fitting endings such as succeeding or failing to reach the goals defined by the game. In contrast, there are no fitting endings to paidia play and no specific results are expected from it. (Ibid.: 230.) *SimCity* is often mentioned as an example of a paidia game, although it is fair to call into question if it is appropriate to call *SimCity* a video *game* because it is of the paidia type. Costikyan (2002: 12) reiterates how Will Wright, the designer of *SimCity*, has described it as a 'software toy' rather than as a game:

> [Wright] offered a ball as an illuminating comparison: It offers many interesting behaviors, which you may explore. You can bounce it, twirl it, throw it, dribble it. And, if you wish, you may use it in a game: soccer, or basketball, or whatever. But the game is not intrinsic in the toy; it is a set of player-defined objectives *overlaid* on the toy.

Costikyan argues that *SimCity* is a good toy because, like a ball, it enables several kinds of actions in the pursuit of player-defined goals. *SimCity* is similar to many video games in that it creates a virtual world that can be manipulated by the player (or user). However, *SimCity* has no intrinsic goals or victory conditions. A *SimCity* player (or user) may set goals for himself, for instance to build a certain type of city. *SimCity* is interesting to play (or to use) repeatedly because it enables a wide choice of goals to be set. (Ibid.: 12-13.)

Because *SimCity* does not end with victory or defeat, or when a goal defined by the software has been reached - even if the player (or user) has set a goal himself, the software is not aware of it - *SimCity* is not a ludus game. However, Costikyan (ibid.: 13) argues that *SimCity* is a game "[…] when a user plays it *as* a game […]". For example, a ball is not a game in itself, but can be used as a prop for different games. In a similar way, *SimCity* can be used as an instrument or a tool (for lack of a better word) for certain games that require the player to keep track of whether or not he achieves the goals he has set himself.

I will use the terms 'paidia play' and 'ludus play' to distinguish between these two types of playing whenever that distinction is important to make. 'Ludus play' is carried out for the sake of pursuing goals defined by the game. The goals (if any) in 'paidia play' are set by the player, and the player may change these goals during the course of paidia play. Hence, there are 'paidia players' and 'ludus players' (rather than 'ludus gamers'). A 'paidia player' may define the goals himself and play to achieve these goals. A 'paidia player' may also play without any set goals. In a similar way, I make a distinction between a 'paidia game' and a 'ludus game'.

Until now, I have not considered the role of the player. The term 'game' can be problematic because it is not possible to dictate how someone will play a video game. Even video games of the ludus type can be played with the paidia approach, for instance by ignoring the intrinsic goals. It is another issue how meaningful paidia play is in such cases. Nevertheless, it is conceivable that a video game can be used for ludus play if the player makes a commitment to pursue the goals defined by the game, and that the same game can also be used for paidia play by ignoring the intrinsic goals. Hence, I use 'playing' as an umbrella term for both paidia and ludus approaches. It is appropriate to call a paidia type of software a 'game' when "[…] a user plays it *as* a game […]" (ibid.). Some video games enable both ludus and paidia in different modes or as approaches to playing the game. In other words, these ludus games do not penalize the paidia player for his approach. For example, there are separate ludus play and paidia play modes in *Amplitude*.

5.4.2 Games as Simulations and Mimicry in Play

Video games are special kinds of systems because they can model the appearance and behavior of other systems such as machines. In other words, video games can be simulations. Frasca (2003: 223-225) argues that video games are the first complex simulational media for mass audiences. Through video games, simulations of different vehicles, for instance, have become accessible to many people.

Video games have the potential to be simulations because they can present environments for experimentation (ibid.). Experimentation is encouraged in video games because the real-world consequences of playing (and making mistakes) are mitigated (Gee 2003: 62-63). The game designer Chris Crawford (2005 [1982]) argues:

> Conflict implies danger; danger means risk of harm; harm is undesirable. Therefore, a game is an artifice for providing the psychological experiences of conflict and danger while excluding their physical realizations. In short, a game is a safe way to experience reality. More accurately, the results of a game are always less harsh than the situations the game models.

When video games simulate reality, Crawford argues, they provide safe ways to experience reality. However, Crawford continues, games are not entirely lacking real-world consequences: “The penalties for losing a game can sometimes be a significant deterrent to game play. Losing to another person always entails some loss of dignity. This may be an attraction of computer games there is less shame in losing to a computer.” (Ibid.) Juul (2003: 35) notes that games can optionally have real-world consequences, which may be negotiated on a play by play, location by location, or person to person basis.

Different types of simulations can be implemented as video games. Interfaces of different machines can be recreated with visual representations on a two-dimensional screen and generic game controllers. However, to create a convincing simulation requires more than representing the controls of a machine or some other system on the screen. Frasca (2003: 223) argues that the essence of simulation in video games is to simulate behavior:

> […] to simulate is to model a (source) system through a different system which maintains (for somebody) some of the behaviors of the original system. […] The key term here is “behavior.” Simulation does not simply retain the - generally audiovisual - characteristics of the object but it also includes a model of its behaviors. This model reacts to certain stimuli (input data, pushing buttons, joystick movements), according to a set of conditions.

To a certain input, a simulation produces an output similar to the output of the system it simulates. This way, using a simulation can involve exploring different options, and experimenting with what the outcomes of certain actions may be (ibid.).

Wolf defines two types of simulation as video game genres, namely ‘management simulation’ and ‘training simulation’. In management simulations, the player must balance the use of certain resources to construct and maintain a community (e.g., a city) or institution (e.g., a soccer team), while dealing with internal or external forces challenging the existence of the community or institution. In addition to entertainment, management simulations can have educational purposes. (Wolf 2001e: 126.) The purpose in training simulations is to simulate realistic situations for the purpose of developing skills such as steering. Training simulations range from realistic simulations used for training airline pilots and tank drivers to simplified approximations of such simulations intended for the purpose of entertainment. (Ibid.: 134.)

Music-making software could be interpreted as simulations in the sense that they are simplified remediations of multitrack studio equipment and music utilities for PCs. Music-making software titles allow music-makers to perform actions of music-making similar to what they could do in a multitrack recording studio if they had access to a studio and knew how to use its particular setup of equipment.

According to Frasca (2003: 223-224), simulations generate signs similar to the systems they simulate:

> A flight simulator or a simple toy plane are not only signs, but machines that generate signs according to rules that model some of the behaviors of a real plane. […] the flight simulator allows the player to perform actions that will modify the behavior of the system in a way that is similar to the behavior of the actual plane.

What Frasca argues about simulations simulating the behavior of the original system also applies to multitrack studios and music-making software. The behavior of the equipment in a multitrack studio is predictable: Specific devices produce audible outcomes the music-makers can expect. Certain aspects of the functionality of a multitrack studio are simulated in music-making software. The audible output that music-making software titles produce is similar to the output of certain devices in a multitrack studio. Some of the devices of a multitrack recording studio are represented visually in music-making software in a similar type of two-dimensional approximations as, for instance, the controls of an airplane in a flight simulator video game. Thus, it can be argued that the behavior of music-making software approximates the behavior of some aspects of the entity of the multitrack studio.[57] The three music-making software titles feature enough of the fundamental functionality of multitrack studio equipment to make music with nonlinear multitrack composing techniques.

According to Frasca (ibid.: 233), simulation may be of paidia and ludus types. In some video games that are simulations, there are valued and quantifiable outcomes, so these are ludus games. To name a few examples of simulations that are ludus games, in a flight simulator a route can be flown in less time than ever before, or a record number of enemy tanks can be destroyed in a tank simulator. Possibly all simulations can be used as tools for paidia play if the person using them chooses to do so.

It seems obvious that a simulation is effective as an experience (as a substitute for the real) when the user behaves so that what is simulated is important to him. The digital media scholar Janet Murray points out that the pleasurable surrender to an imaginative world of a novel or a movie has been described as “willing suspension of disbelief”. However, Murray notes that when people enter the fictional world of a novel or a movie, they do not simply suspend a critical faculty, but also begin to exercise a creative faculty. People actively create belief because they use their intelligence to reinforce the experience rather than to call into question the reality of the experience. (Murray quoted in Perron 2003: 242.) This argument can be applied to playing simulation type of video games. For instance, in a flight simulator having all controls represented on the screen and

[57] Alternatively, music-making software could be interpreted as second-order simulations, that is, simulations of music utilities for PCs, which in turn can be regarded as simulations of multitrack studio hardware.

giving feedback to the player through a force feedback equipped controller may not be enough for the simulation to be an effective experience. Using or playing a simulator also requires some creation of belief. The player or user of the simulator should care for the outcome, for instance, whether the airplane he controls lands successfully or crashes, as if his pride (at least) was at stake.

Simulation and the creation of belief are related to mimicry. Mimicry in the sense of pretending to be someone else is one aspect of play. Mimicry allows one to become partly someone else for the duration of play: People say, for instance, that children play astronauts or cowboys. Caillois (1961: 19) describes mimicry in play as follows:

> One can also escape himself and become another. This is *mimicry.* […] Play can consist […] of becoming an illusory character oneself, and of so behaving. One is thus confronted with a diverse series of manifestations, the common element of which is that the subject makes believe or makes others believe that he is someone other than himself.

Caillois' definition of mimicry applies equally well to stage actors as to people playing. Mimicry while playing video games is by no means a straightforward matter, but a complex issue beyond the scope of this study.

Perron (2003: 242) considers mimicry and role-playing central to the experience of video games. In video games where the player is committed to pursue the goals defined by the game, the player controls a character and cares for the fate of the character, because when the character loses, the player also loses. In some video games, the player-character may die so that everything the player has gathered with great effort in the world of the game is lost. The player cares about the outcome despite the possibility to "press any key to play again", because he has invested time and effort in it. There may also be some form of emotional attachment to the player-character, in a similar way as there may be emotional attachment to characters in various forms of fiction.

Depending on the theme of the video game, the player can mimic some aspects of different characters. For instance, the characters in a game set in the *Star Wars* universe are different from the characters in a game set in the world of *Gitaroo Man*. The characters in a video game and the situations they get involved in should be interesting so that players are keen to interact through the characters with the game world and possibly the other players.[58] Video games declare, for example, that "[…] you are Jango Fett, the most fearless bounty hunter in the galaxy […]" (text from product package of *Star Wars Bounty Hunter*), "[…] you play Takt, a young conductor […]" (text from product package of *Mad Maestro*), and "[p]lay Gitaroo as U-I against a different opponent […]" (text from product package of *Gitaroo Man*). In other words, these games invite the player to mimic some aspects of being a fearless bounty hunter (shooting and fighting), a young conductor (conducting an orchestra), or fighting opponents by playing the Gitaroo as U-I. By playing video games with paidia and ludus approaches, players can approximate the actions, for instance, that astronauts and airplane pilots perform.

---

[58] Designing characters is one aspect of game design. For an overview on character design, see Sheldon (2004).

Mimicry involves the player identifying to some degree with the aspirations and tribulations of the player-characters. The player must act (in the sense of behaving and taking actions) as a fearless bounty hunter in order to make it in the grim world of *Star Wars Bounty Hunter*. In *Get on da Mic*, the player may identify with the ambition of the player-character to become a successful hip hop artist in the world of the game. In *100% Star*, the player may identify with the aspiration of the pop star manager to get his protégé to the top of the charts. There are no player-characters to identify with in music-making software.

Mimicry may involve performing actions one is otherwise not able to do. For instance, in a flight simulator the player may not mimic an airplane pilot, but nevertheless performs actions similar to what piloting an airplane requires. *100% Star* is a pop star manager simulator in the sense that in the game the player performs actions like managing resources and planning future releases and concerts, which may well be a part of the work of a real-world pop star manager.

Music-making software titles attempt to evoke creation of belief, for example, with their introduction videos that suggest that the music-maker has access to recording studio tools or is about to perform at various clubs around the world. Music-making software titles can be interpreted as simulations because they enable music-makers to pretend to be deejays or recording artists by presenting approximations of the tools and materials deejays and recording artists use. This applies in particular to people with little experience of nonlinear multitrack composing. They make music while being engaged in this simulation. However, offering similar tools is not enough for mimicry to occur. It is far from obvious exactly how the music-makers know how to mimic deejays and recording artists. They may gather parts of this knowledge from sources like sound recordings, concert videos, or written texts, that is, from anywhere in the semiotic domain of multitrack composing.

To summarize: Simulation as a way of exploring possibilities and mimicry as an aspect of play are areas of overlap between playing video games and making music with music-making software.

5.4.3 Summary of Games as Systems and Simulations, and the Different Types of Play

Can music-making software be used as instruments or tools for some form of play? As Costikyan (2002: 12-13) argues with the case of *SimCity*, software other than video games can be used for ludus type of play, but the player himself (or some other person) must keep track of the goals and outcomes, and enforce the rules defining the ludus play.

One observation when comparing ludus video games to music-making software is that in the latter, there are no outcomes valorized by the software, there are no winners and losers, there is no score, and there are no end conditions. This is also the case in *Fluid*, which is a never-ending quest, as is actually suggested in the printed user's guide.

Fluid is best interpreted as a software toy that enables music-making through the approach of paidia play.

In contrast to *Fluid*, *Amplitude* has a quantifiable and valued outcome (a positive outcome is getting a score as high as possible). In *Amplitude*, there is a winner or a loser even in the single-player mode because every game session will be superior or inferior to another. Because it has a scoring system, *Amplitude* is undoubtedly a ludus game. In *Beatmania*, *Gitaroo Man*, *SingStar*, and *Get on da Mic*, making music is also a part of the struggle towards an outcome valorized as positive: To achieve a better score than ever before or to directly beat the competitors. In these games, the outcome is quantifiable because the player's accuracy in time, triggered sounds, and pitch is rated as a score.

In *Fluid* and the three music-making software titles, it is up to the music-maker to determine what to do and to decide when a piece of music is complete. This is one more finding justifying the interpretation of music-making software as tools or software toys (because of the low technical quality of the sample files and effects) rather than as games.

Music-making software titles are open-ended similarly to the way Frasca (2003: 230) defines paidia games. Like *SimCity* and *Fluid*, the three music-making software titles allow a number of goals (different musical outcomes from a music-making session) to be set by the music-maker. There are no endings the way there are in ludus games, which end when a certain state (end condition) has been reached. This absence of endings is caused by the conceptual problem that it is very difficult for a software system to define when a piece of music is complete if there are unlimited resources (such as sample files and time) available for making it.

Quoting Lalandé, Frasca defines 'paidea', and, in effect, 'paidia' as "[p]*rodigality of physical or mental activity which has no immediate useful objective, nor defined objective, and whose only reason to be is based in the pleasure experimented by the player"* (Lalandé quoted in Frasca 2006 [1999]).[59] Even if paidia play does not have an immediate useful objective, paidia play may still be an approach to making music with nonlinear multitrack composing techniques. Fullerton et al. (2004: 88-89) argue that play can be interpreted as a state of mind, as a process of experimentation, and as a special kind of approach to trying new things. The essence of play is that it is neither serious nor obligatory. Thus, paidia play is a potential approach to making music with music-making software.

If the three music-making software titles are successful in inviting music-makers to take a playful approach, they can be used as "playgrounds" that provide the tools and materials for making music through the approach of paidia play. This is an important similarity between playing video games and making music with music-making software. Because of the same hardware (PlayStation 2 console, input and output devices), people familiar with video games may consider music-making software as tools or toys for paidia

---

[59] Frasca uses the term 'paidea' in his article "Ludology Meets Narratology: Similitude and Differences Between (Video)Games and Narrative" (Frasca 2006 [1999]), and the term 'paidia' in his later article "Simulation versus Narrative: Introduction to Ludology" (Frasca 2003). It is apparent that Frasca means the same with these two terms.

play and approach music-making with them as a form of play as they do with all the other titles for PlayStation 2.

The three music-making software titles enable paidia play as an approach to music-making. Ludus play is a more difficult approach to making music with music-making software because it requires the music-maker to define fixed goals and to ascertain if these goals are achieved.

In the following chapter, I summarize the similarities and differences between video games and music-making software, and conclude this study by suggesting what should be taken into consideration in the design of future video games that involve nonlinear multitrack composing.

6. Conclusions and Design Foundations

6.1 The Key Similarities and Differences

One of the findings in the previous chapter was that there are similarities between characteristics of music-making software and a number of the formal elements of video games. For example, the ways in which music-making software titles invite music-makers to cross over the boundary to music-making are similar to the ways in which some video games invite players to the world of the game. Parallels to the video game objective ‘exploration’ can be found in music-making with music-making software. There are similarities to the actions in video games with the objective of ‘construction’, that is, selecting and placing prefabricated components on the screen, manipulating their properties, and making immaterial objects from these components.

Of the dramatic elements of video games, making music with music-making software has similarities to certain player types. Making music with music-making software can be exploration in the sense that the music-maker explores how to make music with the features of the software title. Making music can also appeal to the ‘artist’ and ‘craftsman’ player types. A player with the agenda of ‘performer’ could be attracted to the real-time aspects of music-making, such as mixing in *Magix Music Maker* and *eJay Clubworld*. Making music with music-making software potentially appeals to the ‘achiever’ player type, because several things can be achieved in music-making, for instance, completing a piece of music that combines several musical styles.

Music-making software can be interpreted as simulations and the three titles make some attempts to invite the music-maker to mimic (emulate) the actions of a recording artist or deejay. Thus, the three music-making software titles can be interpreted as playgrounds for making music with the approach of paidia play.

However, the aforementioned similarities do not justify interpreting music-making software as games. Next, I summarize the key differences between video games and music-making software by showing how different activities of music-making have been intertwined with the activity of playing in the seven music-themed video games. Table 6.1 shows musical activities enabled by the music-themed video games, their means of inviting play, affordances and limitations concerning music-making, and ludus game elements. I discuss these in detail in the following sections.

Video game	Music-making activities	Means of inviting play	Affordances and limitations concerning music-making	Ludus game elements
SingStar	Singing (karaoke)	Appealing to player agendas of artist and competitor Potential for role-playing in “Star maker” mode	Affordances: c. 30 pieces of music Limitations: the player can make a limited number of mistakes. The player’s singing is quantified.	Quantifying the player’s actions within a framework of timing and pitch, and showing the result immediately. Scoring the result. Positive outcome: to get a score as high as possible. Resources: time, number of allowed mistakes Conflict ending in victory or defeat
Get on da Mic	Rapping (karaoke)	Appealing to player agendas of artist and competitor Potential for role-playing in “career” mode	Affordances: c. 40 pieces of music Limitations: the player can make a limited number of mistakes. The player’s rapping is quantified.	Quantifying and valorizing the player’s actions within the framework of timing. Scoring the result. Positive outcome: to get a score as high as possible. Resources: time, number of allowed mistakes Conflict ending in victory or defeat
Beatmania	Triggering sounds in real-time	Appealing to player agenda of competitor	Affordances: c. 7 pieces of music. The player can trigger sounds by pressing six buttons on the controller. Limitations: the player can make a limited number of mistakes	Quantifying the player’s actions within a framework of timing and six buttons. Scoring the result. Positive outcomes: to beat competitor in direct competition and get a score as high as possible. Resources: time, number of allowed mistakes Conflict ending in victory or defeat
Amplitude	Triggering and modifying sounds in real-time and composing on a lattice in real-time	Appealing to player agendas of artist (in “remix” mode) and competitor	Affordances: c. 25 pieces of music and effects that can be applied in real-time. Player can trigger sounds by pressing three buttons on the controller. There are six tracks to choose from. Limitations: the player can make a limited number of mistakes	Quantifying the player’s actions within a framework of timing and three buttons. Scoring the result. Positive outcomes: to get a score as high as possible and gain access to new songs. Resources: time, energy, and special features like power-ups Conflict ending in victory or defeat
Gitaroo Man	Triggering sounds in real-time	Appealing to player agenda of competitor	Affordances: c. 12 pieces of music. The player can trigger sounds by pressing four buttons on the controller. Limitations: the player can make a limited number of mistakes before the game ends	Quantifying the player’s actions within a framework of timing and pitch. Scoring the result. Positive outcomes: to defeat each opponent and get a score as high as possible. Resources: time and energy Conflict ending in victory or defeat
100% Star	Nonlinear multitrack composing (selecting and arranging)	Appealing to player agendas of artist and competitor	Affordances: c. 200 sample files Limitations: the player has a certain amount of money in the beginning. Money is needed to make music.	Positive outcomes: to get the player’s protégés to the top of the charts and get the “Lifetime Achievement Award” Resources: currency and quantified abilities of the protégés. Managing these resources.
Fluid	Triggering and modifying sounds in real-time	Appealing to player agendas of artist and explorer	Affordances: c. 600 sample files and some effects Limitations: sample files are organized into sets which become available one at a time	Positive outcome: to get to the final level and thus gain access to all sample files (however, this does not bring *Fluid* to an end)

Table 6.1. Summary of the seven music-themed video games.

6.1.1 Music-Making Activities

In *SingStar* and *Get on da Mic* the framework of music-making is karaoke, i.e. singing or rapping on top of background music that is fixed, with the exception of the original vocals, which can be muted.

Sounds (sample files) are triggered in real-time while playing the rhythm games *Beatmania, Amplitude,* and *Gitaroo Man*. This is to some extent similar to “finger drumming” on sampling drum machines such as Akai MPC60, which are equipped with small pads for triggering sample files in real-time. Deejays and musicians have used such sampling drum machines both in the studio and in live performances. The core of rhythm games is essentially a variant of the *Simon* type of electronic games introduced in the 1970s. The timings of when to press buttons are based on the beat and other characteristics of the background music. When playing rhythm games, the player enters the feedback loop of both the game (including the constantly updated progress made towards the goals) and a kind of real-time musical performance (triggering sounds that are heard on top of the background music).

During a game session, the background music in *Beatmania* appears to lack something and thus the player is invited to fill this musical void by triggering sample files. In *Amplitude*, an increasing number of tracks become muted if the player does nothing or continuously fails to press the correct buttons at the correct times. This “deterioration” of music in *Amplitude* invites the player to act to keep the music sounding good.

There are some aspects of nonlinear multitrack composing in *Fluid* and *100% Star*. In *Fluid*, sample files are selected, replayed as loops, and modified with effects like echo and reverb. Some sounds can be triggered and modified in pitch in real-time on top of the music playing in the background. *100% Star* features certain stages of the nonlinear multitrack composing process, namely selecting and arranging. *100% Star* does not enable mixing or modifying sample files.

6.1.2 Means of Inviting Play

Before a game can be played, the player must physically and mentally enter the magic circle of the game (Salen & Zimmerman 2004: 95-96). I noted earlier that the physical boundary is similar in music-making software to video games. A video game can invite the player in several ways to cross the conceptual boundary into the world of the game. These ways include offering a challenge, presenting a conflict to resolve, promising fun and fantasy, and inviting role-playing.

Get on da Mic invites players to the game partly through its elements of role-playing or fantasy, including the exaggerated appearance of the player-characters and the peripherals that can be bought for them in the world of the game. Thus, *Get on da Mic* is not simply a karaoke application. The same can be said of *SingStar* because of its “Star maker” mode with fantasy or role-playing elements, and the option for the player to see himself on the screen through a video camera, and the “pose for paparazzi” feature.

In the case of ludus play, entering the magic circle involves players taking on a lusory attitude, which includes accepting the rules of the game (Suits 2006 [1978]: 185, 188-190), and with them, the affordances and limitations defined by these rules. For Salen and Zimmerman (2004: 99), a lusory attitude is "[…] the state of mind required to enter into the play of a game." Taking on a lusory attitude involves, for instance, behaving during the course of a game as if the goals of the game are important. Committing to resolve a conflict presented by the game (e.g., the threat of the Gravillians in *Gitaroo Man*) is another example of players assuming a lusory attitude.

Music-themed video games invite players to cross the boundary into the magic circle, for instance by appealing to player types (agendas) like 'artist' and 'competitor'. In *Beatmania* the goal is both to beat the competition (another player, or a previous personal score) and in the process, contribute to the music. In *Beatmania*, there is not much appeal to an artistic agenda because there are few musical choices to make and anything else but precise repetition is penalized. In *Amplitude*, the player can choose which track to attempt to unmute next and apply sound modifying effects. In the tutorial, the player is addressed as "DJ". In other words, in *Amplitude* the artistic player agenda is combined with the competitive agenda.

6.1.3 Affordances and Limitations

According to Juul (2003: 55), the rules of a game specify affordances (including the actions players can take) and limitations (including the actions players cannot take). I next discuss how the affordances and limitations in music-themed video games influence music-making.

In *SingStar*, *Get on da Mic*, *Amplitude*, *Beatmania*, and *Gitaroo Man*, the player is provided with a number of prefabricated songs on top of which the games take place. Songs are a part of the affordances of these games.

The player in *Amplitude*, *Beatmania*, and *Gitaroo Man* is provided with a certain number of buttons on the game controller to trigger sample files with. The sample files that can be triggered are defined by the game system. In other words, the player is at the same time limited in the variety of sample files he can trigger and the times when he can trigger the sample files.

The player's singing in *SingStar* is quantified in time and pitch, and the player's rapping in *Get on da Mic* is quantified in time. In other words, a qualitative performance of singing or rapping is transformed into something that is measured, and this enables the game systems to make comparisons between different performances. This means that the player is limited in what these game systems consider legitimate singing or rapping.

In *Amplitude*, *Beatmania*, *Gitaroo Man, SingStar*, and *Get on da Mic*, the player's music-making is limited by the number of mistakes allowed. If the player makes too many mistakes, the game brings the game session to an end.

The player is provided with about 600 sample files and a couple of effects as a means for making music in *Fluid*. The limitations in *Fluid* are that sample files are organized into sets with different musical styles which become available one at a time. The player has to keep the dolphin swimming long enough to get new sample files to make music with. The player is also given the opportunity to trigger samples when the dolphin is swimming in the “cruise stage”, and perhaps this is what the player is supposed to do, in other words, to “jam” while waiting to get access to new sample files.

In *100% Star*, the player is provided with approximately 200 sample files to make music with. The limitation is that it costs money in the game to make music and get the music released (so its success in the game world is defined), and the player has a restricted amount of money at the beginning. The player must balance the use of money between making music and other areas like improving the skills of his protégés.

Most of the limitations in video games are artificial, like the number of lives or mistakes allowed, and are designed to create tension, which is a part of the experience of playing. There are restrictions in the features of music-making software as a result of which, certain things cannot be done with them. However, these cannot be interpreted as limitations in the same sense as the limitations in video games. In video games, there are affordances and limitations for the player to exert effort in order to reach the goals defined by the game. In music-making software, the music-maker has access to features with certain restrictions. By using these features, the music-maker exerts effort in music-making to reach a musical outcome defined (more or less precisely) by himself, and not by the software.

6.1.4 Ludus Game Elements

Ludus play becomes possible when the rules of a game define quantifiable and valorized outcomes (e.g., being the winner or gaining as high a score as possible), and the player commits to exert effort in order to achieve such an outcome. This is the case in *100% Star*, *SingStar*, *Get on da Mic*, *Amplitude*, *Beatmania*, and *Gitaroo Man*.

Conflict emerges between players (and players and AI-controlled characters) when they have been assigned opposite valorized outcomes. For instance, in *Beatmania* there can be only one winner at the end of each game session. In *Gitaroo Man*, the goal is to defeat every opponent controlled by the game.

In *Fluid*, music is made by selecting sample files and arranging them into pieces of music. Levels restrict the number and style of sample files available. A new set of sample files becomes available when the player progresses from one level to another. However, little effort is required to progress from one level to the next. It seems that all that is required of the player is to wait long enough while the dolphin swims ahead. The only ludus game element in *Fluid* could be the goal of getting to the final level (that is, to gain access to all sample files), but there is no struggle to achieve this and apparently *Fluid* does not end this way. The struggle in *Fluid* is similar to what there is in music-making software: At the outset beginning to learn to use the music-making features, and then confronting the “dilemma” of what kind of music to make. There is no valorized outcome in *Fluid* as one resulting piece of music is not considered superior or inferior to the others.

In *SingStar, Get on da Mic, Beatmania, Amplitude,* and *Gitaroo Man* the goal is to repeat successfully specific patterns by singing, rapping, or pressing buttons on the game controller. These patterns are based on the melody, drum section, bass line, or some other part of a piece of music. These patterns are a point of comparison for evaluating the player's performance. Karaoke and rhythm games are straightforward to implement as ludus games because they feature precise patterns that the player attempts to follow as closely as possible. This way the outcome of a game session is quantifiable, valorized, and variable (a player cannot repeat the patterns even in the same song precisely in the same way during every game session). A game session ends when the player has made too many mistakes. In music-making software, there is no framework to make comparisons with, and thus to quantify and assign a value to what the music-maker has done.

Beatmania, Amplitude, and *Gitaroo Man* can also be played with the audio turned off. Having no audio does not necessarily prevent the player from achieving the goal of getting a good score. Listening to the music may or may not facilitate playing these games.

Earlier I noted that it is not justified to claim that making music with music-making software has variable outcomes in the sense of variable outcomes produced by games. In addition, music-making software titles do not have valorized and quantifiable outcomes. One way to turn music-making into ludus play is to define the means of music-making as resources, as is the case in *100% Star*, where music is one of the several resources for the player to manage. In *100% Star*, making music (including "recording" a song) costs money (the resource of currency) that can be invested in other things that may as well contribute to the success of the player's protégés and achieving the ultimate goal of the game. In *100% Star*, music is a resource to manage because the player must consider what songs to make, what songs to release, and so forth.

When playing a game, the player considers the situation, opposition, and the resources that are available every moment (Costikyan 2002: 12). In the music-themed video games examined (with the exception of *Fluid*), there is a layer of music-making and a layer of ludus play. In rhythm games like *Beatmania* and *Amplitude*, these two layers are nicely combined. In *100% Star*, music-making is not well intertwined with ludus play because the player cannot know what music-making actions influence his success in the charts of the game. Music-making software titles do not have opponents' actions to anticipate or resources to manage, so the music-maker's considerations are different from the considerations of playing a ludus game.

6.2 Play during Multitrack Composing

Earlier I noted that in many games rules impose artificial limitations on what the player can do. Submitting one's music-making activities to the restricted features of music-making software could be a form of play, especially if the music-maker also has access to more versatile multitrack studio equipment. Is it appropriate to build constraints into a creative activity like music-making? Should not the making of music be without limitations? Brian Eno is one artist working with the latest multitrack studio technology who sets limitations to what he can do in order to be able to work with this equipment, which would otherwise offer so many options as to excessively occupy his attention (see Eno 2005 [1999]). Thus, setting artificial limitations to what one is able to do can be a playful approach to a creative activity like music-making.

Making music with music-making software has the potential to be a form of play in one more way in addition to what has been discussed so far. As I noted earlier, a sample file may function as a sonic signifier that represents the source it was recorded from and thus represents whatever the source represents. Théberge (1997: 204) argues that much of pop music "[…] has retained the sense of identity that sounds carry with them – their ability to act as a kind of referent for the object that is the source of the sound – thus leading to an aesthetic fundamentally based in collage and ironic juxtaposition." To name an example, sounds of vintage electronic instruments can give musicians a form of direct sonic, and in some cases iconic access to the music of the past (ibid.: 120). Taylor (2001: 152) notes that in hip hop music, sampling has been used as a way to express homage to musical forebears, or to establish a kind of musical community by using sounds from music of the past. Taylor (ibid.: 153) argues that "[…] almost all hip hop sampling practices are involved with making meanings, meanings that make sense to the musicians and to their fans." Gracyk (1996: 79) describes one early example of what could be interpreted as a playful approach to music-making in the recording studio: "[…] [the musician] Todd Rundgren abruptly stops the flow of [the album] *Something/Anything* (1972) to play a "sounds of the studio" game, alerting listeners to the presence of tape hiss and bad editing." In setting up a similar type of hide and seek game, the musician Wendy Carlos writes in the liner notes of her "all-digital" album *Switched-On Bach 2000* (1992) that there is actually on the album one sound originating from an analog synthesizer, and challenges listeners to pinpoint it.

Music-makers may use certain sample files because of the meanings the sampled sounds evoke (Taylor 2001: 154). With special samples, music-makers can suggest different eras and places, pay homage to musical forebears, and they can "visit" in their imagination the eras and places the sampled sounds invoke. The means for modifying sample files (e.g., low-fidelity sampling, reverb, and echo) also produce sonic signifiers and can be used to refer to specific eras or musical forebears. Thus, music-makers can be engaged in a form of play when they refer to certain eras, places, events, people, and concepts with their music.

As discussed in Section 3.9, fan activities described by Jenkins (1992a; 1992b) can be interpreted as a form of playing with the signifiers appropriated from various media texts. Through sampling (and scanning images, digitizing video, and so forth), fans can with software tools almost "hold in their hands" signifiers appropriated from media texts. These signifiers can be transformed, recontextualized, and juxtaposed with other signifiers. Many fans take pleasure in making intertextual connections across a variety of media texts (Jenkins 1992a: 36). Fan fiction, filk songs, fan-made music videos, and remixing are examples of playful activities of producing media texts based on and referring to other media texts. Jenkins (ibid.: 33, 171) notes that there is an ongoing struggle over the meanings of media texts when fans attempt to suggest alternative meanings. Through remixing, music-makers can contribute to the struggle over the meanings, for example, of certain pieces of music or the repertoire of an artist. With music-making software like *MTV Music Generator 2*, the framework for this activity is playful rather than serious.

According to Frith (1996: 244), in particular in popular music, "[d]igital storage has reinforced the feeling both that music is never finished (with every cut the starting point for a variety of mixes) and that it is never really *integrated* – we are always aware of the parts that make up the (temporary) whole." If people perceive, as Frith suggests, that a piece of music consists of discrete parts and is never finished, they could interpret this as an invitation to make new mixes themselves. Remixing can be personalization of music, for example, because a remix differs by definition from the original version, and because the music-maker can recognize the changes *he* has made to an existing piece of music.[60] In particular *MTV Music Generator 2*, which enables sampling, can be a tool for remixing. Thus, enabling sampling and remixing are two design foundations of future video games that involve nonlinear multitrack composing. In the following section, I conclude this study by discussing other design foundations.

---

[60] The ethnomusicologist Marc Perlman's (2003) discussion on people appropriating commodities like pieces of audio equipment offers insight also to understanding remixing as personalization of music. Perlman discusses how some people personalize audio hardware like amplifiers by modifying them. Perlman suggests that when people operate with commodities like audio equipment, personal emotion is imbued into those commodities. Following the economic anthropologist James Carrier, Perlman defines this phenomenon as 'appropriation', which "[…] is the work of imprinting the identity of the owner on the thing owned." A commodity is converted through appropriation into a personal possession. Commodities are appropriated through rituals of acquisition and display. Appropriation may involve physical or symbolic alteration of the commodity. As some people put a lot of effort choosing what to purchase (e.g., avoiding the influence of advertisements, fashion, and peer pressure), commodities are also appropriated through the exercise of choice when acquired. (Perlman 2003: 349-351.) Selecting material to sample has parallels to appropriation through the exercise of choice. Like the appropriation of a commodity, remixing can involve the symbolic alteration and display (sharing with others) of a piece of music.

6.3 Design Foundations of Video Games that Involve Nonlinear Multitrack Composing

I commence this section by reiterating the definition of 'game'. Juul (2003: 30) defines the classic game model as follows:

> *A game is a rule-based system with a variable and quantifiable outcome, where different outcomes are assigned different values, the player exerts effort in order to influence the outcome, the player feels attached to the outcome, and the consequences of the activity are optional and negotiable.*

Juul (ibid.: 172) suggests that the classic game model is one way of structuring interaction and can be used for many purposes and for creating a range of experiences. Because the model defines valued outcomes, it describes ludus games.

The key differences between video games and music-making software become apparent through Juul's definition of the classic game model. As I noted earlier, the activity of making music with music-making software is not restricted by rules that define quantifiable and valued outcomes. With music-making software, the music-maker does not exert effort to influence the outcome to become such that is defined by someone other or something else than himself.

What would it imply for music-making if ludus game rules were introduced into the activity of nonlinear multitrack composing? To begin with, according to Juul's model, these rules have to define outcomes that are variable, quantifiable, and valued (ibid.: 32-34).

Reaching an outcome valorized as positive would be defined as a goal of the game. The ultimate objective in nonlinear multitrack composing is to make a piece of music and this outcome may or may not be valorized by the music-maker. It is difficult to implement a game system that would automatically quantify and valorize an outcome like a piece of music. However, in a multiplayer online game, other people could rate the music, so some pieces of music (outcomes of music-making activities) would be valorized above others. Alternatively, the musical outcome does not have to be valued but the utilization of the means for making it could be assigned values. This is, in turn, connected to variable and quantifiable outcomes.

How can variable and quantifiable outcomes be applied to nonlinear multitrack composing? As I noted earlier, the musical outcomes of making music with music-making software are surely variable. However, the outcomes are not variable in the sense that in most games, the player should not be able to win every time. It is difficult to express in a quantifiable way how successful a musical outcome is, but the utilization of resources for making that musical outcome can be measured. One such resource could be currency: The player would have to pay in the world of the game for the use of the music-making tools and materials. In addition, time can be defined as a resource and the use of time could be measured with a stopwatch or hourglass timer. One way to introduce conflict into a game is to define a competition over the resources. In nonlinear multitrack composing this could be a competition for special sample files or patterns.

For the reasons I discussed earlier, future video games that involve nonlinear multitrack composing should enable user sampling. In a ludus game, user sampling could have a cost associated with it, and this could be defined in the game as, for example, a “licensing fee”. Alternatively, using sampling would limit the use of some other feature so the music-maker would need to consider how the use of sampling limits future options. Resources could also be more abstract like a pool of “musical energy” that every music-making action reduces.

A major difficulty in combining nonlinear multitrack composing with ludus play is assigning positive and negative values to the outcomes of music-making. Juul (ibid.: 34-35) notes that players are attached to the outcome of a game so that a player will be the winner and “happy” in the event of a positive outcome, and the loser and “unhappy” in the event of a negative outcome.

In Juul’s model, ‘negotiable consequences’ mean that a game can be played with or without real-life consequences. This characteristic applies not so much to video games as to other games. Nevertheless, players can agree among themselves what real-time consequences playing a video game have.

‘Player effort’ means in Juul’s model that players struggle to reach an outcome valorized as positive. In the case of ludus play, the valued outcomes are defined by the game’s rules. A fundamental design problem is how to intertwine the layer of effort in nonlinear multitrack composing with the layer of effort in ludus play. Next, I offer some suggestions how to approach this problem.

According to Juul (ibid.: 168), games are ‘themable’, meaning that “[a] game can always be changed from one setting to another […] [and] a set of rules can be assigned a new fictional world without modifying the rules […].” This argument can be validated by comparing *Amplitude* to its predecessor, *Frequency*. The rules in these two video games are very similar, but *Amplitude* appears different because of its science fiction theme. The theme of *Amplitude* involves the player in shooting at cells with a spaceship. This theme is rather disconnected from music-making. However, according to the designers of *Amplitude*, this theme helps novice players to understand what they are supposed to do (ibid.: 163). In a similar way, an appropriate theme can present nonlinear multitrack composing as easily approachable for novice music-makers. Thus, different themes should be considered for future video games that involve nonlinear multitrack composing. When beginning to design the theme, the experiential goals of the game have to be identified: What music-making and playing experiences should the game enable? How to combine artistic activities with competitive activities in the theme?

Video games that involve nonlinear multitrack composing could be designed, for example, around the theme of a record producer or record label manager. To make this a ludus game, there could be a cost associated with the different stages of the nonlinear multitrack composing process. In the beginning, the player would construct a studio with the components of his choosing (like sample file libraries, and software versions of synthesizers, drum machines, and effects devices). When making music, studio time would have to be paid for, as would accessing existing songs that could be remixed by the player. The problem in implementing such a game remains how to develop a software system that determines how successful the pieces of music become in the world of the game (assuming that a positive outcome of the game is defined as getting a song to the top of the charts). In other words, the problem is how to make the musical outcome quantifiable (this was the major flaw in *100% Star*).

In the video game *The Movies*, the player is in charge of a movie studio. In the game, AI-controlled characters like actors, directors, and screenwriters are resources. The abilities of these virtual characters are measured and the player can indirectly attempt to improve their abilities so that they would perform better in what they do (e.g., acting, directing movies, and writing screenplays). This approach of the player having indirect control of the abilities of virtual characters could be used in a record producer or record label manager type of game. The player would hire musicians for recording sessions. What the virtual musicians perform is defined vaguely or precisely by the player. The performances of the virtual musicians would be generated with algorithms producing, for example, MIDI data. If the virtual musician is highly skilled, he produces more complex or varied music than an average virtual musician. The player may take a more hands-on approach in the phases of music-making and modify the music "performed" by the virtual musicians. The player would manage resources including deciding where to invest money, for example, in improving the skills of the virtual musicians, new studio equipment (new virtual instruments and sample files), licensing songs for remixing, and so forth. Thus, the resources invested to make music are quantifiable. One more interesting aspect of *The Movies* is that the game can be played in a mode that follows the historical periods of cinema. For example, when the game is set in the early 1920s, the player has at his disposal the movie-making technology of that era. This idea could be adapted to a record producer or record label manager type of game so that when the game is set, for example, in the early 1960s the player has only a four-track recorder available. As the game progresses to a new era, the player gains access to studio equipment with more features. In addition, the player could choose special scenarios like "Detroit 1987" and see what kind of music he can make when he has access only to certain equipment, like software versions of a Roland TR-909 drum machine, a TB-303 synthesizer/sequencer, and a Yamaha DX100 synthesizer, and low-fidelity sampling like on an Ensoniq Mirage.

Another approach concerning the theme is to consider alternative ways of representing visually the actions of choosing components (selecting or recording sample files), modifying the properties of these components (sound design), and combining these components (arranging and mixing) into a virtual acoustic space (sound spatialization) of a piece of music. For instance, Gibson's (1997) three-dimensional visual approach to mixing could be exploited instead of the current two-dimensional visual representations of sliders and knobs. One option for the theme is an abstract game with nonrepresentational graphics and an objective not organized as a narrative (see Wolf 2001e: 117). The thematic approach could be cartoonish (even bordering on the absurd) as in *Electroplankton* for Nintendo DS (which, however, is not a ludus game as it lacks resources and valued outcomes). Sample files and the ways of selecting, modifying, and arranging them could be defined in the theme as resources. Time could be defined as a resource. Thus, the resources used to make music would at least in part determine the quantifiable and valued outcomes. The music-making activity could also include some aspects of real-time performance, like initially defining the mixes in real-time, and later fine-tuning them in non-real-time.

To summarize, the fundamental design problem in combining nonlinear multitrack composing with ludus play is that the outcomes of music-making are difficult to define as quantifiable and valued. It can be difficult to implement a game system that would rate music automatically, that is, to transform a qualitative thing into something quantifiable. It is more straightforward to combine nonlinear multitrack composing and paidia play. As I noted earlier, paidia play is a possible approach to music-making with the three music-making software titles.

6.3.1 Further Considerations

Future video games that involve nonlinear multitrack composing should enable music-makers to realize the nonlinear multitrack composing techniques discussed in Section 3.8, namely sampling, sound design, programming, mixing, sound spatialization, and remixing. Music-makers can contribute to the semiotic domain of multitrack composing, for instance, by sharing their knowledge of the techniques and by making their music available to others. Built-in Internet connections and officially maintained online communities are not obligatory for the future games because music and information can be shared on the Internet in other ways.

One starting point for the present study was the theory of remediation as defined by Bolter and Grusin (1999), who suggest that media hybrids can be developed by adapting characteristics of different media. However, as I noted in Section 2.8, the remediated version of a medium may include social problems and inequalities that have been a part of the earlier medium. People who have been discouraged from using the earlier form of a medium may also be discouraged from using the remediated form. The music sociologist Mavis Bayton (1990 [1988]) notes that male musicians have dominated the use of recent music technology, and women have less frequently been encouraged to learn and use it. According Théberge (2001: 13), the use of music software among women is limited, even among those who are skilled with computers and musical instruments. These concerns expressed by Bayton and Théberge should be taken into consideration in the design of future video games that involve nonlinear multitrack composing. Designers should identify the obstacles that have led to the inequalities in the use of recent music technology and music software, and design new interfaces and approaches to music-making that level out these obstacles. Here the approaches to designing video games that appeal to women, as suggested by Ray (2004) could be of help.

One more topic to consider in the design is that future video games that involve nonlinear multitrack composing should make it easy for players to learn the semiotic domain of multitrack composing so that they become literate and producers in the domain. Scholars including Gee (2003) and Squire (2006 [2002]) argue that video games can be good learning tools for various subject matter. Video games facilitating learning is a broad topic. Thus, next I consider briefly if some of the key principles of video games that facilitate learning can be found in the three music-making software titles, and suggest how these principles could be adapted as a part of the design foundations.

Certain video games are special in that learning how to play them is integrated into the game so that there is no distinction between learning and playing (Gee 2003: 120). Gee argues that video games are good learning tools when they adapt to the skill-level of the player and so offer the proper level of challenge at all times (ibid.: 71). This way learning is continuous. According to Gee, video games that facilitate learning offer the player in the early episodes a ‘concentrated sample’, meaning that these games initially focus on the fundamental actions, interactions, and objects and tools the player has to master. (Ibid.: 122, 135, 137.) Through concentrated samples, players get a “taste” of the semiotic domain of the video game and get to practice the essential actions of that game (ibid.: 122, 137). In the three music-making software titles, all features are available right from the beginning. This can be confusing to novice music-makers, as there may be too much to do, remember, and consider. Thus, giving the music-maker a concentrated sample (e.g., a subset of features and sample files) in the beginning should be considered in the design of future video games that involve nonlinear multitrack composing.

A lot of the learning that takes place through playing video games involves transfer. One example of transfer is the case of a student who applies something he has learned in biology to a new problem he faces in social studies. (Ibid.: 124.) Precursor domains are another example of transfer. Gee (ibid.: 48) suggests that simulations like *SimCity* can be precursor domains for sciences that use computer-based simulations as a research method. In other words, Gee suggests that skills and attitudes acquired in the semiotic domains of video games can be transferred for use in other semiotic domains. Squire (2006 [2002]) points out that research on transfer suggests that playing video games develops skills that are useful only in very similar contexts. Thus, the procedures of future video games that involve nonlinear multitrack composing should not be too distant from the procedures of making music with multitrack studio equipment.

A simulation type of video game may give a player a “taste” of a semiotic domain because the player encounters terminology, signs (e.g., control panel icons), situations, goals, feelings, and values similar to those in the domain that is simulated (Gee 2003: 97). The three music-making software titles can be interpreted as precursor domains for the semiotic domain of multitrack composing. By using music-making software, novice music-makers can learn to make music with nonlinear multitrack composing techniques, and in the process, learn some of the terminology, signs, and situations associated with the semiotic domain of multitrack composing. Obviously, this does not mean that these music-makers could walk into any multitrack recording studio and use the equipment effortlessly, but music-makers can nevertheless get a “taste” of the semiotic domain of multitrack composing. Future video games should exploit the opportunities of providing a “taste” of the semiotic domain of multitrack composing by, for instance, using an established terminology and visual representations of user interfaces of existing hardware and software.

In video games that facilitate learning, the player is given information both on-demand and just-in-time. According to Gee, this way the player can best understand the information given to him and exploit it in the game. (Ibid.: 138.) In music-making software, the stages of making a piece of music are explained in the tutorials, but otherwise the music-maker is not given much information on-demand or just-in-time. In other words, the music-maker is more or less left on his own. This is an obstacle to learning the semiotic domain of multitrack composition with the three music-making software titles, and this issue should be addressed in the design of future video games that involve nonlinear multitrack composing.

In many video games, there are multiple ways to progress and solve the problems in the game. Players can experiment with different approaches and choose those that suit best their style of thinking, solving problems, performing actions, and learning. According to Gee, this is highly motivating for both sustaining interest in the game and for learning. (Ibid.: 81, 108.) This principle of multiple approaches is intrinsic in the nonlinear multitrack composing process, as there is no preferred order of approach or single solution to make a piece of music. Future video games that involve nonlinear multitrack composing should not force music-makers to follow predefined paths.

People often learn from their mistakes. Video games allow players to make mistakes and the consequences of making mistakes in video games are not severe (see Section 5.4.2). *Magix Music Maker* and *eJay Clubworld* allow very few "mistakes" to be made because most combinations of sample files sound good together rhythmically and harmonically. On the other hand, *MTV Music Generator 2* allows perhaps too many "mistakes" to be made as it enables programming patterns freely on a chromatic pitch lattice, so it is possible to program patterns that sound unpleasant or chaotic. Future video games that involve nonlinear multitrack composing should perhaps allow some imperfections in the music made with it so the music-maker can gain experience when contemplating his mistakes. In addition, if there is a risk for making mistakes and failure, every success may be experienced more strongly.

In the semiotic domains of video games, meaning, thinking, and learning are linked to multiple modalities that may be mutually supportive in communicating similar or different meanings, each of which contributes to communicate a larger whole. (Ibid.: 108-109.) With the three music-making software titles, learning the semiotic domain of multitrack composing takes place through multiple modalities such as sound, graphics (sample file waveforms, notes on a pitch lattice), and written language (the terminology). For example, becoming able to edit a sample file through its visual representation (a waveform graphic) requires learning to understand the relationship between the visual representation and the sound. Programming patterns requires learning to understand, for example, how note pitches and the progression of time are represented visually. Making music with music-making software requires multimodal literacy, but the primary source of information that enables this activity is the modality of sound.

According to Gee (ibid.: 49), learning a semiotic domain involves becoming able to participate at some level in the affinity groups associated with that domain. It is plausible that people using music-making software can become associated with some affinity groups within the semiotic domain of multitrack composing, for example, by participating in discussion forums on the Internet. Learning a semiotic domain necessitates reflecting on and getting feedback from one's actions in the domain (ibid.: 73, 107). In video games, sources of feedback include responses from the game software and comments from people knowledgeable about the semiotic domain one is learning. The three music-making software titles do not give any feedback on the music-maker's actions. In video games, positive feedback can be given when some of the player's actions (or the successful results of these actions) are valued above other actions. In other words, giving feedback can be connected to actions that contribute to achieving (or failing to achieve) a valued outcome. This means that in the design of future video games that involve nonlinear multitrack composing, game designers have to define which actions are to be assigned positive values and which are not. In other words, the designers have to define which music-making actions are valuable within the framework of the game.

It is plausible that a player should expect a video game to teach him in certain ways. Therefore, the principles that video games employ to facilitate learning, as suggested by Gee and others, should be considered in the design of future video games that involve nonlinear multitrack composing. This is also a topic for further research.

The next step in exploring how to combine playing video games with nonlinear multitrack composing is to define a game design and implement a software prototype on the basis of it. By playtesting the prototype, data can be gathered about how people perceive such a game/music-making hybrid, and how they use or want to use its features (see Fullerton et al. 2004: 157-316). In addition to playtesting, ethnographic methods like participant observation would be advantageous in ascertaining the playtesters' experiences of the prototype. The data collected would be analyzed and the resulting information utilized to revise the game design and in the implementation of a new prototype, which in turn would be playtested. Game design is an iterative process of playtesting, evaluating, and revising (ibid.: 197).

Earlier I proposed the personalization of music as one interpretation for the remixing phenomenon. This includes, for example, adding some personal touches to an existing piece of music. The remixed pieces of music become personal because the music-maker recognizes his idiosyncrasies (including, e.g., the choices made while making the music and the reasons for these choices) when listening to the remixes made by him. Making music can be a means of self-expression and self-reflection. Thus, learning to make music with the means of nonlinear multitrack composing is a form of empowerment. As nonlinear multitrack composing techniques are used extensively to make music, obtaining practical knowledge (see Chanan 1994: 27-31, 250-287) of these techniques can be interpreted as a part of learning (digital) media literacy. Further research is required to achieve a detailed understanding of the play-oriented amateur music-making phenomenon outlined in this study: Why do people make music and what meanings do they draw from it?

References

Books, Articles, and Online Publications

Abercrombie, Nicholas & Brian Longhurst (1998). *Audiences: A Sociological Theory of Performance and Imagination*. Trowbridge, Wiltshire: Sage.

Aho, Marko (2005). “Soitinten simulacra, eli musiikkia kyborgien makuun.” In Outi Ampuja & Kaarina Kilpiö (eds.): *Kuultava menneisyys: suomalaista äänimaiseman historiaa*. Historia mirabilis 3. Turku: Turun historiallinen yhdistys ry. Pages 217-233.

Bartle, Richard (2004). *Designing Virtual Worlds*. Indianapolis, IN.: New Riders Games.

Bartlett, Bruce & Jenny Bartlett (1998). *Practical Recording Techniques*. Second Edition. Boston, MA: Focal Press.

Bayton, Mavis (1990 [1988]). “How Women Become Musicians.” In Simon Frith (ed.): *On Record: Rock, Pop, & the Written Word*. New York: Pantheon. Pages 238-257.

Björk, Staffan & Jussi Holopainen (2005). *Patterns in Game Design*. Hingham, MA: Charles River Media.

Blaine, Tina (2005). “The Convergence of Alternate Controllers and Musical Interfaces in Interactive Entertainment.” In *Proceedings of New Interfaces for Musical Expression (NIME’05) Conference*, May 26-28, 2005, Vancouver, Canada. Pages 27-33.

Bolter, Jay David & Richard Grusin (1999). *Remediation: Understanding New Media*. Cambridge, MA: MIT Press.

Caillois, Roger (1961). *Man, Play, and Games*. Glencoe, New York: The Free Press.

Cary, Tristram (1992). *Illustrated Compendium of Musical Technology*. London: Faber and Faber.

Chadabe, Joel (1997). *Electric Sound. The Past and Promise of Electronic Music*. Upper Saddle River, NJ: Prentice Hall.

Chanan, Michael (1994). *Musica Practica: The Social Practice of Western Music from Gregorian Chant to Postmodernism*. London: Verso.

Chanan, Michael (1995). *Repeated Takes: A Short History Of Recording and its Effects on Music*. London: Verso.

Colbeck, Julian (1988). *Keyfax 3*. London: Music Maker Books.

Collins, Karen E. (2006). “Video Games Audio [work in progress].” http://www.dullien-inc.com/collins/texts/vgaudio.pdf [Accessed 2.3.2006]

Copier, Marinka & Joost Raessens (2003). “Level Up.” In Marinka Copier & Joost Raessens (eds.): *Level Up*. *Digital Games Research Conference.* 4-6 November 2003. Utrecht University. Pages 8-10.

Costikyan, Greg (2002). “I Have No Words & I Must Design: Toward a Critical Vocabulary for Games.” In Frans Mäyrä (ed.): *Computer Games and Digital Culture Conference Proceedings*. Tampere: Tampere University Press. Pages 9-33.

Cox, Christoph & Daniel Warner (2004). “Glossary.” In Christoph Cox & Daniel Warner (eds.): *Audio Culture: Readings in Modern Music*. New York: Continuum. Pages 409-417.

Crawford, Chris (2005 [1982]). *The Art of Computer Game Design*. http://www.vancouver.wsu.edu/fac/ peabody/game-book/Coverpage.html [Accessed 27.12.2005]

Cunningham, Mark (1998). *Good Vibrations. A History of Record Production*. Bodmin: Sanctuary Publishing Limited.

Cutler, Chris (2000). “Plunderphonics.” In Simon Emmerson (ed.): *Music, Electronic Media and Culture.* Aldershot: Ashgate. Pages 87-114.

de la Motte-Haber, Helga (2002 [2000]). “Soundsampling: An Aesthetic Challenge.” In Hans-Joachim Braun (ed.): *Music and Technology in the Twentieth Century*. Baltimore: The John Hopkins University Press. Pages 199-206.

DeMaria, Rusel & Johnny L. Wilson (2002). *High Score! The Illustrated History of Electronic Games*. Berkeley, CA: McGraw-Hill/Osborne.

Denis, Guillaume & Pierre Jouvelot (2005). “Motivation-Driven Educational Game Design: Applying Best Practices to Music Education”. In *Proceedings of Advances in Computer Entertainment (ACE) Conference*, Valencia, Spain, June 15-17, 2005. Pages 462-465.

Dobson, Richard (1992). *A Dictionary of Electronic and Computer Music Technology*. Oxford: Oxford University Press.

Durant, Alan (1990). “A New Day for Music? Digital Technologies in Contemporary Music-Making.” In Philip Hayward (ed.): *Culture, Technology and Creativity*. London: John Libbey & Co. Pages 175-196.

Eisenberg, Evan (1988). *The Recording Angel: Music, Records and Culture from Aristotle to Zappa.* London: Pan Books.

Emmerson, Simon (ed.) (2000). *Music, Electronic Media and Culture.* Aldershot: Ashgate.

Eno, Brian (1998). “Foreword.” In Mark Cunningham: *Good Vibrations. A History of Record Production*. Second Edition. Bodmin: Sanctuary Publishin Limited. Page 17.

Eno, Brian (2004). “The Studio as Compositional Tool.” In Christoph Cox & Daniel Warner (eds.): *Audio Culture: Readings in Modern Music*. New York: Continuum. Pages 127-130.

Eno, Brian (2005 [1999]). “The Revenge of the Intuitive.” *Wired* 7.01. http://www.wired.com/wired/archive/7.01/eno.html [Accessed 27.12.2005]

Eskelinen, Markku & Ragnhild Tronstad (2003). “Video Games and Configurative Performances.” In Mark J. P. Wolf & Bernard Perron (eds.): *The Video Game Theory Reader*. London: Routledge. Pages 195-220.

Fikentscher, Kai (2003). ““There’s not a problem I can’t fix, ‘cause I can do it in the mix” On The Performative Technology of 12-Inch Vinyl.” In René T. A. Lysloff & Leslie C. Gay, Jr. (eds.): *Music and Technoculture*. Middletown, CT: Wesleyan University Press. Pages 290-315.

Finn, Mark (2002). “Console Games in the Age of Convergence.” In Frans Mäyrä (ed.): *Computer Games and Digital Culture Conference Proceedings*. Tampere: Tampere University Press. Pages 45-58.

Finnegan, Ruth (1989). *Hidden Musicians*. Cambridge: Cambridge University Press

Frasca, Gonzalo (2003). “Simulation versus Narrative: Introduction to Ludology.” In Mark J. P. Wolf & Bernard Perron (eds.): *The Video Game Theory Reader*. London: Routledge. Pages 221-235.

Frasca, Gonzalo (2006 [1999]). “Ludology Meets Narratology: Similitude and Differences Between (Video)Games and Narrative.” http://www.ludology.org/articles/ludology.htm [Accessed 2.2.2006]

Friberg, Johnny & Dan Gärdenfors (2006 [2004]). “Audio Games: New perspectives on game audio.” In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology,* June 3-5, 2004, Singapore. http://www.sitrec.kth.se/bildbank/pdf/Friberg_G%C3%A4rdenfors_ACE2004.pdf [Accessed 2.2.2006]

Frith, Simon (1996). *Performing Rites: On the Value of Popular Music*. Oxford: Oxford University Press.

Frith, Simon (2001). “The Popular Music Industry.” In Simon Frith, Will Straw & John Street (eds.): *The Cambridge Companion to Pop and Rock*. Cambridge: Cambridge University Press. Pages 26-52.

Frith, Simon & Lee Marshall (eds.) (2004). *Music and Copyright*. Second Edition. Trowbridge, Wilts: Edinburgh University Press.

Fullerton, Tracy, Christopher Swain & Steven Hoffman (2004). *Game Design Workshop. Designing, Prototyping, and Playtesting Games*. San Francisco, CA: CMP Books.

Gee, James Paul (2003). *What Video Games Have To Teach Us About Learning and Literacy*. New York: Palgrave Macmillan.

Gerrish, Bruce (2001). *Remix. The Electronic Music Explosion*. Auburn Hills, MI: EMBooks.

Gibson, David (1997). *The Art of Mixing: a Visual Guide to Recording, Engineering, and Production*. Auburn Hills, MI: Mix Books.

Gould, Glenn (2004 [1966]). “The Prospects of Recording.” In Christoph Cox & Daniel Warner (eds.): *Audio Culture: Readings in Modern Music*. New York: Continuum. Pages 115-126.

Goodwin, Andrew (1990 [1988]). “Sample and Hold. Pop Music in the Digital Age of Reproduction.” In Simon Frith (ed.): *On Record: Rock, Pop, & the Written Word*. New York: Pantheon. Pages 258-273.

Goodwin, Andrew (1991). “Rationalization and Democratization in the New Technologies of Popular Music.” In James Lull (ed.): *Popular Music and Communication*. Second Edition. Newbury Park, CA: Sage. Pages 75-100.

Goodwin, Andrew (1997). “Drumming and Memory. Scholarship, Technology, and Music-Making.” In Andrew Herman et al. (eds.): *Mapping the Beat. Popular Music and Contemporary Theory*. Malden, MA: Blackwell Publishers. Pages 121-136.

Gracyk, Theodore (1996). *Rhythm and Noise: An Aesthetics of Rock*. Durham, NC: Duke University Press.

Grimshaw, Mark (1998). "Remix! Where's the Original?" In Tarja Hautamäki & Helmi Järviluoma (eds.): *Music on Show: Issues of Performance*. Tampere: University of Tampere Department of Folk Tradition. Pages 129-131.

Grodal, Torben (2003). "Stories for Eye, Ear, and Muscles: Video Games, Media, and Embodied Experiences." In Mark J. P. Wolf & Bernard Perron (eds.): *The Video Game Theory Reader*. London: Routledge. Pages 129-155.

Gronow, Pekka (1996 [1983]). "The Record Industry: The Growth of a Mass Medium." In Pekka Gronow: *The Recording Industry: An Ethnomusicological Approach*. Tampere: University of Tampere. Pages 37-59.

Gronow, Pekka (1996). *The Recording Industry: An Ethnomusicological Approach*. Tampere: University of Tampere.

Gronow, Pekka & Ilpo Saunio (1998). *An International History of the Recording Industry*. Guildford and King's Lynn: Cassell.

Hawkins, Erik (2004). *The Complete Guide to Remixing*. Boston, MA: Berklee Press.

Herz, J. C. (1997). *Joystick Nation: How Videogames Ate Our Quarters, Won Our Hearts, and Rewired Our Minds*. Boston, MA: Little, Brown, and Co.

Hindman, David (2006). "Modal Kombat: Competition and Choreography in Synesthetic Musical Performance." In *Proceedings of New Interfaces for Musical Expression (NIME'06) Conference*, June 4-8, 2006, Paris, France. Pages 296-299.

Holm, Jukka, Juha Arrasvuori & Kai Havukainen (2006). "Using MIDI to Modify Game Content." In *Proceedings of New Interfaces for Musical Expression (NIME'06) Conference*, June 4-8, 2006, Paris, France. Pages 65-70.

Holopainen, Jussi & Aki Järvinen (2005). "Ludology for Game Developers – an Academic Perspective." In Steve Rabin (ed.): *Introduction to Game Development*. Hingham, MA: Charles River Media. Pages 51-68.

Huber, David Miles & Robert E. Runstein (1997). *Modern Recording Techniques*. Fourth Edition. Boston, MA: Focal Press.

Huizinga, Johan (1980 [1949]). *Homo Ludens. A Study of the Play-Element in Culture*. London: Routledge & Kegan Paul.

Hämäläinen, Perttu, Teemu Mäki-Patola, Ville Pulkki & Matti Airas (2006 [2004]). “Musical Computer Games Played by Singing.” In *Proceedings of the 7th Int. Conference on Digital Audio Effects (DAFx'04)*, October 5-8, 2004, Naples, Italy. http://www.tml.tkk.fi/~tmakipat/pubs/dafx2.pdf [Accessed 2.2.2006]

Jenkins, Henry (1992a). *Textual Poachers: Television Fans and Participatory Culture.* London: Routledge.

Jenkins, Henry (1992b). “Fan Culture and Popular Media.” In Lisa A. Lewis (ed.): *The Adoring Audience: Fan Culture and Popular Media*. London: Routledge. Pages 208-236.

Jenkins, Henry & Kurt Squire (2002). “The Art of Contested Spaces.” In Lucien King (ed.): *Game on: The History and Culture of Videogames*. London: Laurence King Publishing Ltd. Pages 63-75.

Jones, Steve (1992). *Rock Formation: Music, Technology, and Mass Communication*. Newbury Park: Sage.

Julien, Olivier (1999). “The Diverting of Musical Technology by Rock Musicians: The Example of Double-Tracking.” *Popular Music* Vol 18/3. Cambridge University Press. Pages 357-365.

Juul, Jesper (2003). *Half-Real: Video Games between Real Rules and Fictional Worlds*. Ph.D. dissertation. IT University of Copenhagen.

Järvinen, Aki (2003). “Making and Breaking Games: a Typology of Rules.” In Marinka Copier & Joost Raessens (eds.): *Level Up*. *Digital Games Research Conference.* 4-6 November 2003. Utrecht University. Pages 68-79.

Katz, Mark (2004). *Capturing Sound: How Technology Has Changed Music*. Berkeley, CA: University of California Press.

Kealy, Edward R. (1990 [1979]). “From Craft to Art. The Case of Sound Mixers and Popular Music.” In Simon Frith (ed.): *On Record: Rock, Pop, & the Written Word*. New York: Pantheon. Pages 207-220.

Kent, Steven L. (2001). *The Ultimate History of Video Games. From Pong to Pokémon: The Story Behind the Craze That Touched Our Lives and Changed the World*. Roseville, CA: Prima Pub.

King, Geoff & Tanya Krzywinska (2002). “Computer Games / Cinema / Interfaces.” In Frans Mäyrä (ed.): *Computer Games and Digital Culture Conference Proceedings*. Tampere: Tampere University Press. Pages 141-153.

Kohler, Chris (2005). *Power-Up: How Japanese Video Games Gave the World an Extra Life*. Indianapolis, IN: BradyGAMES Publishing.

Kress, Gunther & Theo van Leeuwen (2001). *Multimodal Discourse. The modes and media of contemporary communication.* London: Arnold.

Landau, Sidney I. (1989). *Dictionaries: The Art and Craft of Lexicography*. Cambridge: Cambridge University Press.

van Leuween, Theo (1999). *Speech, Music, Sound*. Hong Kong: Macmillan.

Lunenfeld, Peter (2003). "The Design Cluster." In Brenda Laurel (ed.): *Design Research: Methods and Perspectives*. Cambridge, MA: MIT Press. Pages 10-15.

Lysloff, René T. A. (2003). "Musical Life in Softcity. An Internet Ethnography." In René T. A. Lysloff & Leslie C. Gay, Jr. (eds.): *Music and Technoculture*. Middletown, CT: Wesleyan University Press. Pages 23-63.

Lysloff, René T. A. & Leslie C. Gay, Jr. (eds.) (2003). *Music and Technoculture*. Middletown, CT: Wesleyan University Press.

Marks, Aaron (2001). *The Complete Guide to Game Audio: For Composers, Musicians, Sound Designers, and Game Developers*. Lawrence, KS: CMP Books.

Manning, Peter (1993). *Electronic and Computer Music*. Second Edition. Oxford: Clarendon Press.

Manovich, Lev (2001). *The Language of New Media*. Cambridge, MA: MIT Press.

Manovich, Lev (2006 [2002]). "Avant-garde as Software." http://www.uoc.edu/artnodes/eng/art/manovich1002/manovich1002.html [Accessed 1.2.2006]

May, Derrick (2000). "Interview." In Peter Shapiro (ed.): *Modulations. A History of Electronic Music: Throbbing Words on Sound*. Hong Kong: Caipirinha Productions, Inc. Pages 126-127.

McAllister, Ken S. (2004). *Game Work: Language, Power, and Computer Game Culture*. Tuscaloosa, AL: University of Alabama Press.

Middleton, Richard (1990). *Studying Popular Music*. Milton Keynes: Open University Press.

Millard, Andre (2002 [2000]). “Tape Recording and Music Making.” In Hans-Joachim Braun (ed.): *Music and Technology in the Twentieth Century*. Baltimore and London: The John Hopkins University Press. Pages 158-167.

Miranda, Eduardo Reck (2000 [1998]). *Computer Sound Synthesis for the Electronic Musician*. Oxford: Focal Press.

Moog, Bob (1984 [1976]). “Modulation.” In *Synthesizer Basics / By the Editors of Keyboard Magazine*. Milwaukee, WI: Hal Leonard Publishing Corporation. Pages 31-32.

Moog, Bob (1984 [1978]). “Principles of Voltage Control Part I.” In *Synthesizer Basics / By the Editors of Keyboard Magazine*. Milwaukee, WI: Hal Leonard Publishing Corporation. Pages 41-42.

Mäyrä, Frans (2002). “Introduction: All Your Base Are Belong to Us.” In Frans Mäyrä (ed.): *Computer Games and Digital Culture Conference Proceedings*. Tampere: Tampere University Press. Pages 5-8.

Nazarian, Bruce (1988). *Recording Production Techniques for Musicians*. London; New York: Amsco Publications.

Negus, Keith (1992). *Producing Pop: Culture and Conflict in the Popular Music Industry*. London: Arnold.

Oswald, John (2004). “Bettered by the Borrower: The Ethics of Musical Debt.” In Christoph Cox & Daniel Warner (eds.): *Audio Culture: Readings in Modern Music*. New York: Continuum. Pages 131-137.

Pearce, Celia (2002). “Story as Play Space.” In Lucien King (ed.): *Game on: The History and Culture of Videogames*. London: Laurence King Publishing Ltd. Pages 112-119.

Pekkilä, Erkki (1996). “Etnomusikologia ja mediatodellisuus.” In Pirkko Moisala (ed.): *Etnomusikologian vuosikirja 1996*. Helsinki: Suomen etnomusikolginen seura. Pages 227-241.

Perlman, Marc (2003). “Consuming Audio: An Introduction to Tweak Theory.” In René T. A. Lysloff & Leslie C. Gay, Jr. (eds.): *Music and Technoculture*. Middletown, CT: Wesleyan University Press. Pages 346-357.

Perron, Bernard (2003). “From Gamers to Players and Gameplayers. The Example of Interactive Movies.” In Mark J. P. Wolf & Bernard Perron (eds.): *The Video Game Theory Reader*. London: Routledge. Pages 237-258.

Pinch, Trevor & Frank Trocco (2002 [2000]). “The Social Construction of the Early Electronic Music Synthesizer.” In Hans-Joachim Braun (ed.): *Music and Technology in the Twentieth Century*. Baltimore, MD: The John Hopkins University Press. Pages 67-83.

Pinch, Trevor & Frank Trocco (2002). *Analog Days. The Invention and Impact of the Moog Synthesizer*. Cambridge, MA: Harvard University Press.

Ray, Sheri Graner (2004). *Gender Inclusive Games Design: Expanding the Market*. Hingham, MA: Charles River Media.

Read, Oliver & Walter Welch (1976). *From Tin Foil to Stereo: Evolution of the Phonograph*. Indianapolis, IN: Howard W. Sams & Co.

Reynolds, Simon (1998). *Energy Flash: A Journey Through Rave Music and Dance Culture*. London: Picador.

Rhea, Tom 1984 [1979]. “What is Synthesis?” In *Synthesizer Basics / By the Editors of Keyboard Magazine*. Milwaukee, WI: Hal Leonard Publishing Corporation. Pages 3-5.

Risset, J. C. (1991). “Timbre Analysis by Synthesis: Representations, Imitations, and Variants for Musical Composition.” In Giovanni DePoli, Aldo Piccialli & Curtis Roads (eds.): *Representations of Musical Signals*. Cambridge, MA: MIT Press. Pages 7-44.

Roads, Curtis et al. (1996). *The Computer Music Tutorial*. Cambridge, MA: MIT Press.

Roads, Curtis (1996a). “Sampling and Additive Synthesis.” In Curtis Roads et al.: *The Computer Music Tutorial*. Cambridge, MA: MIT Press. Pages 115-155.

Roads, Curtis (1996b). “Sound Mixing.” In Curtis Roads et al.: *The Computer Music Tutorial*. Cambridge, MA: MIT Press. Pages 353-385.

Roads, Curtis (1996c). “Sound Spatialization and Reverberation.” In Curtis Roads et al.: *The Computer Music Tutorial*. Cambridge, MA: MIT Press. Pages 449-493.

Roads, Curtis (1996d). “Performance Software.” In Curtis Roads et al.: *The Computer Music Tutorial*. Cambridge, MA: MIT Press. Pages 659-701.

Roads, Curtis & John Strawn (1996a). “Digital Audio Concepts.” In Curtis Roads et al.: *The Computer Music Tutorial*. Cambridge, MA: MIT Press. Pages 5-47.

Roads, Curtis & John Strawn (1996b). “Introduction to Digital Sound Synthesis.” In Curtis Roads et al.: *The Computer Music Tutorial*. Cambridge, MA: MIT Press. Pages 85-113.

Roden, Timothy & Ian Parberry (2005). “Designing a Narrative-Based Audio Only 3D Game Engine.” In *Proceedings of Advances in Computer Entertainment (ACE) Conference*, Valencia, Spain, June 15-17, 2005. Pages 274-277.

Rose, Tricia (1994). *Black Noise: Rap Music and Black Culture in Contemporary America*. Chapel Hill, NC: University of North Carolina Press.

Salen, Katie & Eric Zimmerman (2004). *Rules of Play. Game Design Fundamentals*. Cambridge, MA: MIT Press.

Samagaio, Frank (2002). *The Mellotron Book*. Vallejo, CA: artistpro.com, LCC.

Schafer, R. Murray (1994). *The Soundscape: Our Sonic Environment and the Tuning of the World*. Rochester, Vermont: Destiny Books.

Schmidt Horning, Susan (2002 [2000]). “From Polka to Punk: Growth of an Independent Recording Studio, 1934-1977.” In Hans-Joachim Braun (ed.): *Music and Technology in the Twentieth Century*. Baltimore, MD: The John Hopkins University Press. Pages 136-147.

Schrader, Barry (1982). *Introduction to Electro-Acoustic Music*. Englewood Cliffs, NJ: Prentice-Hall.

Sheldon, Lee (2004). *Character Development and Storytelling*. Boston, MA: Thomas Course Technology.

Shuker, Roy (2002). *Popular Music. The Key Concepts*. London: Routledge.

Squire, Kurt (2006 [2002]). “Cultural Framing of Computer/Video Games.” *Game Studies* Vol 2/1. http://www.gamestudies.org/0102/squire/ [Accessed 31.1.2006]

Sterne, Jonathan (2003). *The Audible Past: Cultural Origins of Sound Reproduction*. Durham, NC: Duke University Press.

Straw, Will (2001). “Consumption.” In Simon Frith, Will Straw & John Street (eds.): *The Cambridge Companion to Pop and Rock*. Cambridge: Cambridge University Press. Pages 53-73.

Suits, Bernard (2006 [1978]). “Construction of a Definition.” In Katie Salen & Eric Zimmerman (eds.): *The Game Design Reader: A Rules of Play Anthology.* Cambridge, MA: MIT Press. Pages 172-191.

Sweet, Michael (2004). “Using Audio as a Game Feedback Device.” In Tracy Fullerton, Christopher Swain & Steven Hoffman: *Game Design Workshop. Designing, Prototyping, and Playtesting Games*. San Francisco, CA: CMP Books. Pages 307-309.

Tagg, Philip (1994). “From Refrain to Rave: the Decline of Figure and the Rise of Ground.” *Popular Music* Vol 13/2. Cambridge University Press. Pages 209-222.

Tasajärvi, Lassi (ed.) (2004). *Demoscene: The Art of Real-time*. Helsinki: Even Lake Studios.

Taylor, Timothy D. (2001). *Strange Sounds: Music, Technology and Culture*. London: Routledge.

Théberge, Paul (1997). *Any Sound You Can Imagine: Making Music/Consuming Technology*. Hanover, NH: University Press of New England.

Théberge, Paul (2001). “‘Plugged In’: Technology and Popular Music.” In Simon Frith, Will Straw & John Street (eds.): *The Cambridge Companion to Pop and Rock*. Cambridge: Cambridge University Press. Pages 3-25.

Théberge, Paul (2003). “Ethnic Sounds”: The Economy and Discourse of World Music Sampling. In René T. A. Lysloff & Leslie C. Gay, Jr. (eds.): *Music and Technoculture*. Middletown, CT: Wesleyan University Press. Pages 93-108.

Théberge, Paul (2004). “Technology, Creative Practice and Copyright.” In Simon Frith & Lee Marshall (eds.) *Music and Copyright*. Second Edition. Trowbridge, Wilts: Edinburgh University Press. Pages 139-156.

Tiits, Kalev (1997). “The Sound (Technology) of Music – ajatuksia äänisynteesin ja sointivärin problematiikasta.” *Synkooppi Opus 51* 3/1997. Pages 32-39.

Toynbee, Jason (2000). *Making Popular Music: Musicians, Creativity, and Institutions*. London: Arnold.

Trask, Simon & Tim Barr (1996). “Ferocious Beat.” In Chris Kempster (ed.) *History of House*. Kent: Music Maker Publications. Pages 49-56.

Truax, Barry (2001*). Acoustic Communication*. Second Edition. Westport, CT: Ablex Publishing.

Vail (1993a). “Buchla’s First Modular System.” In Mark Vail: *Vintage Synthesizers: Groundbreaking Instruments and Pioneering Designers of Electronic Music Synthesizers*. San Francisco, CA: Miller Freeman Books. Pages 97-101.

Vail (1993b). “Fairlight CMI.” In Mark Vail: *Vintage Synthesizers: Groundbreaking Instruments and Pioneering Designers of Electronic Music Synthesizers*. San Francisco, CA: Miller Freeman Books. Pages 190-195.

Vail (1993c). “Linn LM-1 Drum Computer.” In Mark Vail: *Vintage Synthesizers: Groundbreaking Instruments and Pioneering Designers of Electronic Music Synthesizers*. San Francisco, CA: Miller Freeman Books. Pages 252-256.

Walser, Robert (1993). *Running With the Devil: Power, Gender, and Madness in Heavy Metal Music*. Hanover, NH: University Press of New England.

Wanderley, Marcelo & Marc Battier (eds.) (2000). *Trends in Gestural Control of Music*. Paris: IRCAM. [Electronic publication]

Waters, Simon (2000). Beyond the Acousmatic: Hybrid Tendencies in Electroacoustic Music. In Simon Emmerson (ed.) *Music, Electronic Media and Culture*. Aldershot, Burlington: Ashgate. Pages 56-83.

Wikipedia (2005a). Entry: “bastard pop”. http://www.wikipedia.org [Accessed 27.12.2005]

Wikipedia (2005b). Entry: “the grey album”. http://www.wikipedia.org [Accessed 27.12.2005]

Winkler, Todd (1998). *Composing Interactive Music*. Cambridge, MA: MIT Press.

Wishart, Trevor (1996). *On Sonic Art*. Amsterdam: Harwood Academic Publishers.

Wolf, Mark J. P. (2001a). “Introduction.” In Mark J. P. Wolf (ed.) *The Medium of the Video Game*. Austin, TX: University of Texas Press. Pages 1-9.

Wolf, Mark J. P. (2001b). “The Video Game as a Medium.” In Mark J. P. Wolf (ed.) *The Medium of the Video Game*. Austin, TX: University of Texas Press. Pages 13-33.

Wolf, Mark J. P. (2001c). “Space in the Video Game.” In Mark J. P. Wolf (ed.) *The Medium of the Video Game*. Austin, TX: University of Texas Press. Pages 51-75.

Wolf, Mark J. P. (2001d). “Time in the Video Game”. In Mark J. P. Wolf (ed.) *The Medium of the Video Game*. Austin, TX: University of Texas Press. Pages 77-91.

Wolf, Mark J. P. (2001e). “Genre and the Video Game.” In Mark J. P. Wolf (ed.) *The Medium of the Video Game*. Austin, TX: University of Texas Press. Pages 113-134.

Wolf, Mark J. P. & Bernard Perron (2003). “Introduction.” In Mark J. P. Wolf & Bernard Perron (eds.): *The Video Game Theory Reader*. London: Routledge. Pages 1-24.

Software

The format for software references is: Title (platform or OS), developer (year of release).

ACID (Windows XP), Sony (2006).

eJay Clubworld (PlayStation 2), Unique Development Studios (2002).

GarageBand (Mac OS X), Apple Computer (2006).

Magix Music Maker (PlayStation 2), Magix Development (2001).

MTV Music Generator 2 (PlayStation 2), Jester Interactive (2001).

MTV Music Generator 3: This is the Remix (PlayStation 2), Mix Max (2004).

Music (PlayStation), Jester Interactive (1998).

Music 2000 (PlayStation), Jester Interactive (1999).

Music 3000 (PlayStation 2), Jester Interactive (2003).

Premiere (Windows XP), Adobe Systems (2006).

Video Games

The format for video game references is: Title (platform or OS), developer (year of release).

100% Star (PlayStation), Teque Software (2001).

Amplitude (PlayStation 2), Harmonix (2003).

BeatMania (PlayStation), Konami (2000).

BLiX (Shockwave), gameLab (2001).

Electroplankton (Nintendo DS), Nintendo (2006).

EverQuest (Windows 98), Sony Online Entertainment (1999).

Fame Academy (PlayStation 2), Monte Cristo (2003).

Fluid (PlayStation), Sony (1996).

Frequency (PlayStation 2), Harmonix (2001).

Get on da Mic (PlayStation 2), Artificial Mind and Movement, Inc. (2004).

Gitaroo Man (PlayStation 2), Inis (2002).

Mad Maestro (PlayStation 2), Desert Productions (2002).

The Movies (Windows XP), Lionhead Studios (2005).

MTV Celebrity Deathmatch (PlayStation 2), Big Ape Prod. (2003).

Pac-Man (arcade), Namco (1980).

Pong (arcade), Atari (1973).

Rez (PlayStation 2), UGA (2002).

SimCity (several platforms), Maxis (1989-).

The Sims (Windows 98), Maxis (2000).

SingStar (PlayStation 2), London Studios (2004).

Space Invaders (arcade), Taito (1977).

Staraoke (Windows XP), Intervisio (2003).

Star Wars Bounty Hunter (PlayStation 2), LucasArts (2002).

Theme Park World (PlayStation 2), Bullfrog Productions (2000).

WarioWare, Inc.: Mega Microgame$ (GameBoy Advance), Nintendo (2003).

Vib-Ribbon (PlayStation), NanaOn-sha (2000).

Tiivistelmä

Pelillisyys musiikin tekemisessä: tutkimus videopelien ja musiikinteko-ohjelmistojen yhtäläisyyksistä

Sony PlayStation 2 videopelikonsolilla käytettävillä musiikinteko-ohjelmistoilla tehdään musiikkia valitsemalla, muuntelemalla, ja yhdistelemällä digitaalisia äänitiedostoja niiden visuaalisten representaatioiden kautta. Tällainen menetelmä musiikin tekemiseen määritellään tutkimuksessa epälineaariseksi moniraitasäveltämiseksi (*nonlinear multitrack composing*). Lingvisti James Paul Geen teoriaa videopeleistä semioottisina alueina sovelletaan määrittelemään moniraitasäveltämisen semioottinen alue (*semiotic domain of multitrack composing*), johon sisältyy mm. tekniikat moniraitastudioiden laitteiden käyttämiseen musiikin tekemisessä ja näillä tekniikoilla tehdylle musiikille annetut merkitykset.

Tutkimuksen tarkoituksena on selvittää millaisia periaatteita tulisi soveltaa kun pelillisyys halutaan yhdistää musiikin tekemiseen epälineaarisen moniraitasäveltämisen tekniikoilla. Tutkimus taustoittaa tällaisten videopelien suunnittelua.

Tutkimusaineistona ovat musiikinteko-ohjelmistot *Magix Music Maker*, *MTV Music Generator 2* ja *eJay Clubworld*, sekä seitsemän musiikkiaiheista videopeliä. Tutkimusmenetelminä on kuvailla musiikinteko-ohjelmistoja ja videopelejä, vertailla musiikinteko-ohjelmistoja niiden yhteisen toiminnallisuuden selvittämiseksi, ja osoittaa moniraitastudioiden laitteiden funktiot määrittelevän metaterminologian kautta kuinka musiikinteko-ohjelmistoilla pystytään toteuttamaan epälineaarisen moniraitasäveltämisen tekniikat. Musiikinteko-ohjelmistoja vertaillaan videopeleihin ja epälineaarista moniraitasäveltämistä pelillisyyden eri muotoihin. Tutkimuksessa määritellään epälineaarisen moniraitasäveltämisen malli, jonka vaiheet ovat äänitiedostojen valitseminen, samplaaminen, muunteleminen, sovittaminen ja miksaaminen. Mallia käytetään jäsentämään tutkimuksessa tehtäviä vertailuja.

Tutkimuksessa osoitetaan miten musiikinteko-ohjelmistot ovat yhteensopivia moniraitastudioiden laitteiden kanssa. Musiikinteko-ohjelmistoilla pystytään toteuttamaan seuraavat epälineearisen moniraitasäveltämisen tekniikat: samplaaminen (*sampling*), äänisuunnittelu (*sound design*), miksaaminen (*mixing*), ohjelmointi (*programming*), äänen spatialisointi (*sound spatialization*) ja remiksaaminen (*remixing*). Tutkimuksessa määritellään näiden tekniikoiden taustat ja annetaan esimerkkejä siitä miten tekniikoita on hyödynnetty musiikin tekemisessä. Tekniikoiden määrittelyt pohjataan aikaisempaan kirjallisuuteen.

Tutkimuksessa vertaillaan musiikinteko-ohjelmistoja seuraaviin videopelien ominaisuuksiin: formaalit elementit, dramaattiset elementit, videopelit systeemeinä ja simulaatioina, sekä jäljittely pelaamisessa. Videopelien ominaisuuksien määritelmät pohjataan aikaisempaan kirjallisuuteen. Musiikin tekemistä videopeleillä ja musiikinteko-ohjelmistoilla tarkastellaan pelitutkija Gonzalo Frascaa seuraten käsitteiden *ludus* ja *paidia* kautta. *Ludus* määritellään pelaamiseksi, jossa on tavoitteena tuottaa esim. voittaja tai

ennätystulos. *Paidia* määritellään vapaammaksi leikkimiseksi, jossa ei ole tällaisia tavoitteita.

Musiikinteko-ohjelmistoille ja videopeleille osoitetaan seuraavat olennaiset yhtäläisyydet. Musiikinteko-ohjelmistot muistuttavat videopelejä joissa pelaaminen yhdistyy luomiseen ja kokeilemiseen. Kuten videopeleissä joissa on pyrkimyksenä rakentaminen, myös musiikinteko-ohjelmistoilla tehdään objekteja valitsemalla, muuntelemalla ja sijoittelemalla esivalmistettuja komponentteja. Musiikinteko-ohjelmistot tulkitaan avoimiksi simulaatioiksi eli eräänlaisiksi “leikkikentiksi”, jotka tarjoavat välineet ja materiaalit musiikin tekemiseen *paidia* lähestymistavalla.

Musiikinteko-ohjelmistoille ja videopeleille osoitetaan seuraavat keskeiset eroavuudet. Musiikinteko-ohjelmistoissa ei ole monien pelien tapaan yksiselitteisiä tavoitteita, joiden saavuttaminen päättäisi musiikin tekemisen. Musiikinteko-ohjelmistot eivät motivoi musiikin tekemistä samoilla tavoilla kun pelit motivoivat pelaajia (esim. palkkioilla ja mukauttamalla haasteellisuutta taitoja vastaavaksi), eivätkä anna pelien tapaan palautetta. Resurssien hallinta on keskeistä useassa pelissä. Musiikinteko-ohjelmistojen toiminnot ja äänitiedostot eivät ole resursseja, sillä niiden käyttöä ei ole rajoitettu. Musiikinteko-ohjelmistot eivät vaikuta musiikintekijän käyttäytymiseen kuten pelit vaikuttavat pelaajien käyttäytymiseen määrittelemällä tietyt pelaamisen lopputuloksista toisia arvokkaammiksi. Musiikinteko-ohjelmistoissa ei ole konfliktia kuten *ludus* tyyppisissä peleissä koska musiikin tekemiseen ei liity kilpailua eikä sen lopputuloksena ole voittajia ja häviäjiä, eikä pistein tai muuten määrällisesti mitattua tulosta. Täten musiikin tekeminen musiikinteko-ohjelmistoilla ei ole *ludus* tyyppistä pelaamista.

Tutkimuksen tuloksena esitetään suunnitteluperiaatteita uusille videopeleille, joissa epälineaarinen moniraitasäveltäminen yhdistyy pelaamiseen. Suunnitteluperiaatteet pohjataan musiikinteko-ohjelmistojen ja videopelien yhtäläisyyksiin ja eroavuuksiin. Pelitutkija Jesper Juulin määritelmää seuraten, *ludus* tyyppistä pelaamista ja epälineaarista moniraitasäveltämistä yhdistävien pelien sääntöjen tulee määritellä lopputulos joka on vaihteleva, määrällisesti mitattavissa oleva ja arvotettu. Tällaisiin lopputuloksiin päästään esim. määrittelemällä musiikin tekemisen välineet ja materiaalit resursseiksi ja mittaamalla niiden käyttöä. Suunnittelussa tulisi hyödyntää videopeleissä käytettyjä oppimista tukevia menetelmiä edesauttamaan epälineaarisen moniraitasäveltämisen oppimista.