A PRELIMINARY INVESTIGATION INTO INTERACTIVE COMPUTER GAME MUSIC

By

Miles Warrington

Submitted in partial fulfillment of the
requirements for the degree of
Master of Music (Composition)
to the
Faculty of Humanities,
University of KwaZulu-Natal.

Durban, 2005.

TABLE OF CONTENTS

OUTLINE

This article will attempt to further understanding as far as music and computer games are concerned. More specifically, it will focus on “Interactive Music”. Interactivity in music has been developed and experimented upon since the early 1990’s, and has many other applications besides its use in computer games. For instance, there are “Interactive Music Performances” and “Interactive Compositions” etc. Such areas of interactivity will not be explored and are too broad a topic to mention in this article. For the purposes of explaining concepts and ideas, comparisons will be drawn with media such as film, as they have some important similarities with computer games but are perhaps better understood and on the whole more accessible. To this end a terminology section has been included to provide the inexperienced computer user some idea of concepts, tools and language used in the computer world.

INTERACTIVITY: A DEFINITION

Interactivity is a bi-directional response (Winkler describes it as a "two-way street") [Winkler, Todd. 1998. p 3], from both user and the computer game itself. This is achieved through an interface, which is comprised of the parameters as set out by the game as accepted by the user, which then elicits a response. So the interface is a point of translation between the computer game's virtual world and that as perceived and accepted by the user. This establishes a dialectic which will lead ultimately to the completion of the process or the game, as in this case. Interactivity is therefore the ability of a user to have a direct influence on a sequence of events and to modify their structure and form. Apropòs the computer game stimulating a response from the user through the interface, the computer game itself will respond to the influences or input, which is how it is designed to function. Interactivity, therefore cannot exist as a uni-directional input and response. It is by its very nature bi-directional.

In terms of traditionally linear media, such as photographs, music performances or recordings, books etc., the interface is a subjective emotive response from an individual to these media. In other words, it is a form of cultural discourse. The response to a Beethoven symphony for example, will depend on what the listener brings to the hearing. One person may have a vast referential background to the work and therefore respond in a very intimate or even critical manner to the performance and the work. On the other hand, a different person may remain unresponsive to the experience, lacking the referential background.

Conversely, in a non-linear medium, such as a non-linear computer game (both linear and non-linear forms exist - see chapter on "Linear and Non-Linear Media"), the interface at this point in time is principally a manifestation of the computer game designer's intent in terms of sound and visuals etc., and is reliant on intervention and both physical and mental participation and commitment on the part of the user. In this way, they have the ability to make the user feel essential or even indispensable in the process, which of course he/she is. It is clear to see from this, that there is a direct relationship between non-linearity and interactivity. In other words, interactive media are by their very definition non-linear.

It is possible, however to imagine a future where the arts media will colonise the new ground won by computer games. Conversely one can see the opposite happening and a new process developing, for example, the interactive film "I'm Your Man."[Ryan, Marie-Laure. 2001], leading to a new synthesis in the media generally. This is an example of a film being made into a kind of game, with interactive parameters.

Interactivity is, therefore, pervasive in all the media and is becoming more so in particular when referring to computer games. Books too can exhibit this quality. The "Choose Your Own Adventure"[1] series by Edward Packard for example, is an instance of a formerly linear form in the process of becoming non-linear and interactive. This may be achieved by asking the reader at the end of each section through a series of multiple choice questions, how the story should continue, and then being referred to different pages. A different example of the "interactive book" may be found in Ryan's book on "Narrative as Virtual Reality", in which she describes Neal Stephenson's book "The Diamond Age", under the heading "Dream of the Interactive Immersive Book." [Ryan, Marie-Laure. 2001]

Stephen Wilson outlines the differences between interactive media and non-interactive media as follows [Wilson, Stephen. 1993. p 2]:

- The adjustment/ choice process is not optional; it is structured into the events to the extent that some will not proceed without viewer actions.
- The choice process is externalized so that the nature of user action is obvious.
- Some forms of interactive events attempt to control the process of choice by specifying elements such as the timing of choices and the array of decisions available.
- Non interactive events such as novels, movies and the like imply strongly a preferred linear sequence even if internal adjustments such as those described earlier are being made. Most forms of interactive events avoid a suggested sequence.

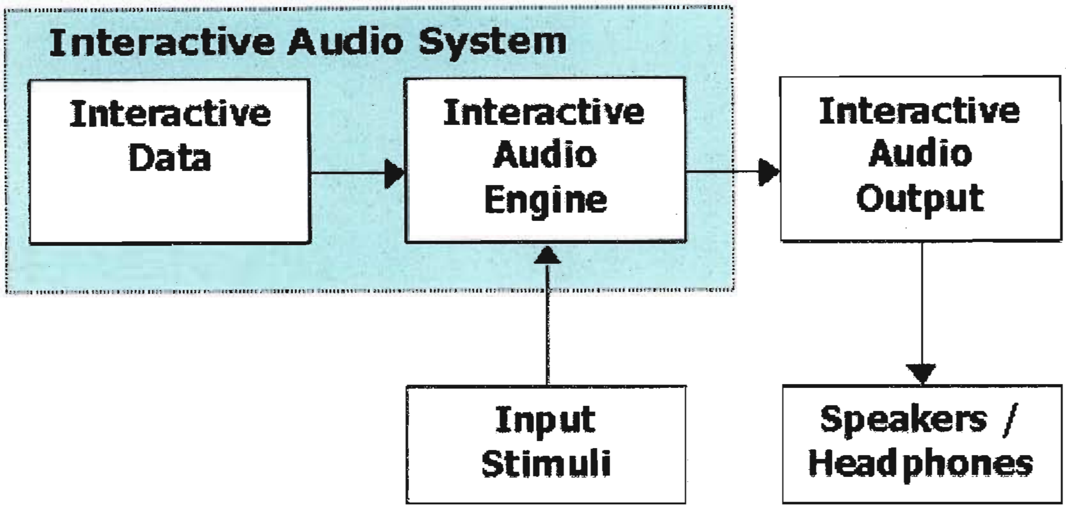
[1] There are many titles in this series by Packard, through several publishers from the 1970's through to the 1990's. For example: "Journey to the year 3000". Bantam Press, 1987.

In further consideration of the definition of interactivity, one could look at the distinction between creation and consumption. Poets, directors, editors, writers, composers and the like, have always had, by the very nature of their work, the ability to control and shape their creations. They decide upon the series of events occurring in the medium. The consumer – the readers, movie-goers, etc., on the other hand do not have this ability to shape the outcome. However, it has been suggested by Stephen Wilson, "that it is useful to conceptualize the experience of interactive multimedia as the attempt to break down this distinction by providing authoring opportunities to the consumer." [Wilson, Stephen. 1993. p 2] Todd Winkler clarifies this statement in another way, when writing of computer games thus: "Interactive music techniques add realism and elements of participation to screen-based works, creating effective soundtracks by using compositional algorithms that change continuously in response to a story or in response to a user's actions."[Winkler, Todd. 1998. p 296]

From this, we can clearly see that the medium of the computer game is well suited to providing the consumer some opportunity to control and shape the outcome of the experience. But what is required here, is a deeper understanding of the importance of making *music* interactive as the importance of interactivity in the graphic interface in computer games, is easily understood. This article will attempt to further understanding in this regard.

We see that the computer becomes the ultimate tool for the realisation of the interactive experience. This is so, as interactive multimedia demands the capability to exhibit visual and audio information simultaneously, and more importantly, allow for user input to change this information through an input device. The last two decades have shown huge leaps in the ability of the technology to reproduce photo quality pictures and cd quality sound. Important at this point, is to explain what exactly "interactive music/audio" is and how this is implemented in a computer game. To give a basic idea of how interactive music or audio in general is created in a computer game scenario the following diagram extracted from notes compiled in a group report from "Project Bar-B-Q" 2003 at the Eighth Annual Interactive Music Conference", helps to clearly display the components of an interactive audio system, which would occur in a computer game:

Fig. 1.[1]



Referring to the above diagram, an interactive audio system consists of an "interactive audio engine" (the workings of which will be described under the chapter "Interactive Audio Engines" later on in this article), and "interactive data". "Interactive data" is the actual game play programming that is coded to respond to user input or "input stimuli" that will in turn be sent to the "interactive audio engine" where depending on what the "input stimuli" is viz., scene change, fading or cross-fading, sound file choice, sound layering etc, the appropriate sound file(s) will be sent to the "interactive audio output", where it will be converted into a final mix before being sent to the speakers or headphones.

Exploring the concept of interactivity further, leads me on to discuss the concept of immersion.

"Immersion" describes a concept whereby the user or participant theoretically experiences the environment in a realistic as possible way by gaining stimulus through as many senses as possible. Simon Waters in an article entitled "Beyond the acousmatic: hybrid tendencies in electro-acoustic music", highlights the relationships between digital technologies, interactivity and immersion as identified by Roy Ascott: "Immersion relates to the scientific insight that the observer is always an implicit participant in the observation being made which translates in compositional terms into a concern with the sensory immersion of the subject (listener) in the work." [Emmerson & contributors. 2000. pg 62]

[1] "What is interactive audio?", Eighth Annual Interactive Music Conference, Project Bar-B-Q, Group report: What is Interactive Audio? And what should it be? Fat Labs 2003. http://www.projectbarbq.com/bbq03/bbq03r1.htm

This describes the importance of the “reality” of the experience. The more “real”, the better the response is from the player to the game. If for example there are multiple scenes in a game that require different environmental sounds, then these sounds or ambiences are determined by both user decision to be in that scene and by what is being projected on the screen. What I mean by this, is that what the user hears, is usually what the user sees. A good example would be the constant background noise one hears in a city environment: the bustle of traffic and the noise of people talking, walking, laughing, shouting etc., all contribute to that background noise. The ambience is therefore determined by what background graphic the user sees on the screen. In other words, if for example the user's avatar[1] (also possible in the 1st person) is standing in a busy street, the background noise appropriate would be car sounds, the general drone of traffic and of course the inherent sounds of lots of people. This type of sound is therefore reliant on the “set” – the graphical interface that the user plays in. The background over which the game takes place or scene is what the user hears and sees. The user is therefore “immersed” in the environment.

So the concept of “real” in a virtual world is an important determining factor in the definition of interactivity and immersion. Taking this one step further, we can combine “virtual” and “real” to form “virtual reality”. This is defined as an “interactive, immersive experience generated by a computer.”[Ryan, Marie-Laure. 2001] “Virtual Reality” computer games are however not all that accessible to the general public – the equipment for the head gear and so on is extremely expensive, as are the applications. Some computer games do however offer the user some form of VR by combining some of the techniques to create immersion and interactivity (explained in more detail in chapter to follow), and in so doing provide a “virtually real” experience for the user – “The purpose of VR technology is to connect the user to a simulated reality. It applies to novels, movies, drama, representational paintings, and those computer games that cast the user in the role of a character in a story…”[Ryan, Marie-Laure. 2001]

It is possible to see from this, that interaction has two clear aspects: either the user's actions affect that of the computer's output, or the computer's actions affect the user's input. This appears to be two-fold in the game environment, and what affects the one, has an indirect or direct affect on the output from the other. This becomes very clear in situations

[1] See terminology.

where the plot of the game requires the gamer to do certain things correctly in order to move further to completion. It is almost as if there is a 'dual-interaction' taking place. This proves that one of the inherent qualities of interactivity is the bi-directional response as explained earlier in this chapter.

"Note that in both of the above categories of interaction - electronics affected by the performer or performer affected by the electronics - can be mixed, and the boundaries can become blurred. In any case, the categories should be sufficiently broad to encompass all that currently fits under the heading of interactive." [Garnett, Guy, E. 2001. p 24]

LINEAR AND NON-LINEAR MEDIA

Media can be broken up into two categories, namely those that are linear and those that are non-linear in nature. Examples of the former could include linear computer games, photographs, movies, books and a music concert; and examples of the latter could include interactive music performances, interactive books (as explained earlier) and specifically of course for our purposes, non-linear computer games.

On the subject of linear media, it is possible, that a book for example may be opened onto any page, a video recorder fast forwarded to any point and the music skipped to any given point, making these seemingly highly linear mediums, non-linear. Yet, the response of the medium itself to the consumer's input is limited, in that the actual material doesn't change in accordance with the input. Instead, we are merely jumping along a kind of 'timeline'. In non-linear media, such as some computer games, this is quite different.

Computer games do occur in both forms of linearity and non-linearity. Games such as 'Tetris' or Microsoft's 'Solitaire' for example, are instances of linear computer games. In these games, the user has limited control over the outcome because of the way the games are designed. They are comprised of levels which succeed each other in a linear order and the user only has control over very specific tasks the game gives them. The games give the user a fixed 'set' of choices and do not provide a myriad of choices or options available in non-linear forms.

To clarify the differences, a closer look at the constituents of film music is required. In the film music industry, there are two types of sound/music. They are referred to as diegetic and non-diegetic sound. If the sound source of a character or object appears within the space of the film, it is known as diegetic sound. For example, a character picks up a violin in a scene and starts to play it. The violin sound we hear playing is diegetic sound. Non-diegetic sound on the other hand, refers to sound sources that do not make reference to the space of the film. A good example of this would be music or sound that enhances the action in the film. Claudia Gorbman describes this type of music in the following way: "Music creates an emotional springboard for the audience. It provides on a plate what emotion the audience should be feeling."[Gorbman, Claudia. 1987. p 73]

Computer games too have both diegetic and non-diegetic sound. Both the background ambience and music one hears is non-diegetic. Only the difference here is, it is non-linear, in that it will change based on random inputs from user interaction. So, whereas non-diegetic (ambient) music in film doesn't change at will from third party intervention, computer game ambience does. Unlike the non-diegetic sound source in films, here it is therefore interactive as it is determined by user input.

On the other hand, sound effects, clue music and dialogue are diegetic sound sources in computer games. Again, the distinct difference here is that they are non-linear, in that their production is stimulated and manipulated by user input. In the computer game "Uru" for example, at the beginning of the game when the user is in a forest ravine with many dead leaves on the floor, the crunching sound of foliage underfoot stops the second you decide to stop walking.

We now notice a distinct difference between the two genres here: in film, the user sees the picture and hears the music which feeds emotion; in games: the user sees the picture, hears the music but it *makes you do* i.e. it has the added ability to create action outside of itself and beyond its own boundaries. This is the very nature of interactivity.

Keeping these in mind, the differences between non-linear computer games and linear film become more apparent. Film is a literary, not a musical form. Music in films interacts contrapuntally - both the picture and music are independent parts of the whole, merely interacting with each other. We find music that suggests something apparent on the screen and contra, music that is suggestive of something not apparent. In computer games, the music doesn't merely suggest, it actually states. An example may be found in the game "Tron 2.0", when the player encounters danger, the music suddenly becomes very fast and intense. The player can of course decide to leave the area quickly and the tense music fades out and the scene music returns. In other words, it has the ability to now become more vital in its role with the pictures. It should be noted at this point though, that it is very difficult for music to actually describe something. It does however, allow for associations, especially when combined with pictures in film or graphics in computer games and it helps to contribute to the user's sense of ambience or atmosphere. Claudia Gorbman in her book on film music *Unheard Melodies*, talks about this relationship between image and picture as follows: "If we must summarize music-image and music-narrative relationships in two words or less", she says, "mutual implication is more accurate."[Gorbman, Claudia. 1987. p 15] This underlines the idea that the image in film is no longer thought of as autonomous. This 'mutual relationship' between music and picture is then seen to be even more apparent in computer games.

Due to some computer games non-linear nature, both music and picture rely on an external input. This type of interaction by its very nature, makes the music respond to the change in graphic very dramatic, and at the same time may provide invaluable information to the player. Non-linearity thus opens up many new interesting creative opportunities for the composer. There is the element of indeterminate user-input, so in non-linear games, the composer has to allow for the indeterminacy. In film, this is clearly not the case, and the composer has to merely follow the linear unfolding of images. This brings me to discuss the construction of music in linear and non-linear media.

MUSICAL CONSTRUCTION IN LINEAR AND NON-LINEAR MEDIA

In this section I will draw comparisons between the construction of music in film as a linear medium and non-linear computer games as a non-linear medium.

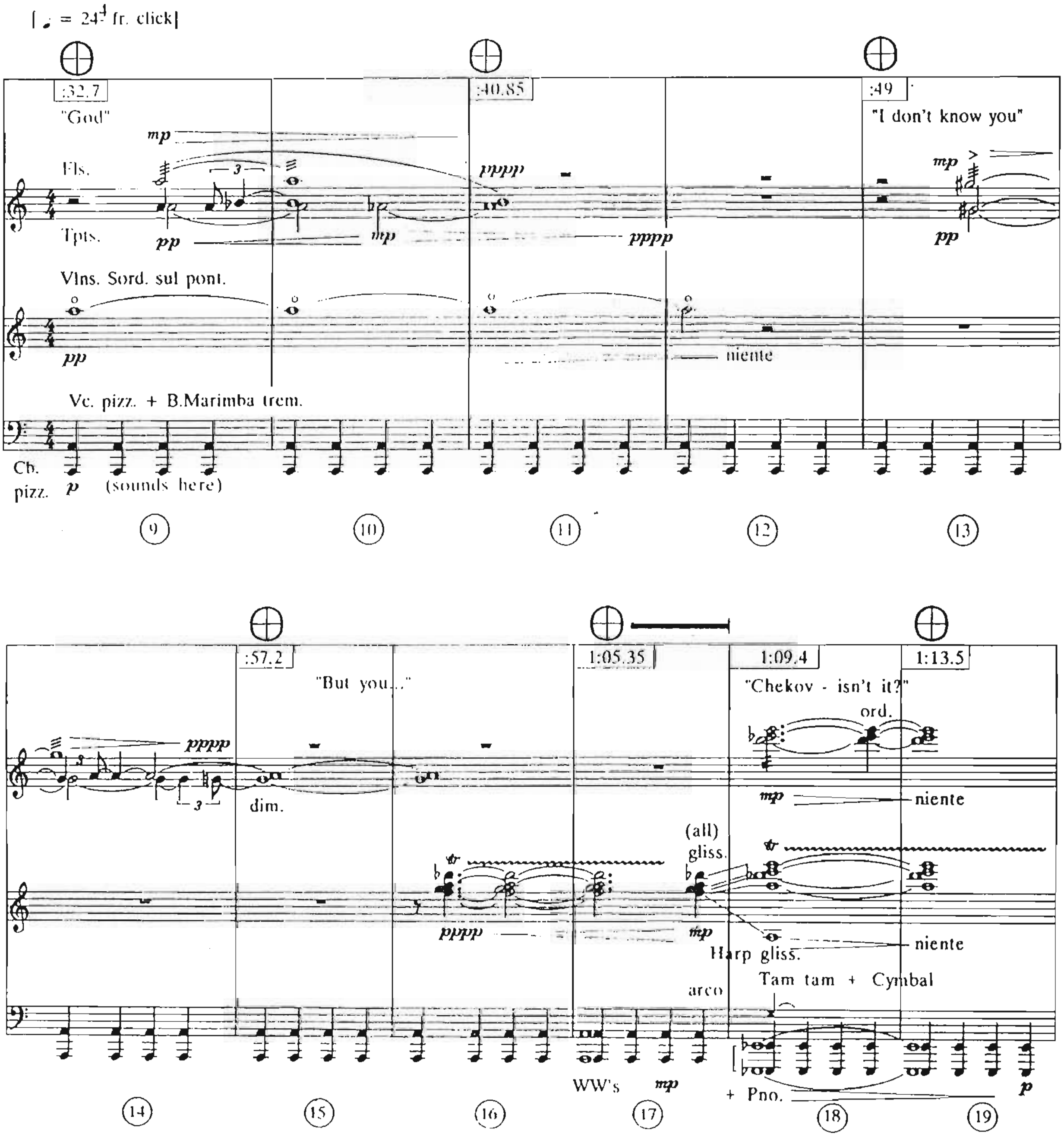
LINEAR FILM MUSIC CONSTRUCTION

In film music composition there is a very close working relationship between the composer and the director, the two interacting on many different levels. One such level would be an agreement on feeling or mood. In a tense scene, such as a climber climbing up a steep cliff face, we naturally hear tense or nervous music. Alexander Courage is quoted as saying "I've always said you're putting an emotional hook into the nose of the audience and moving them around this way emotionally while the picture's going on." [Karlin & Wright. 1990. pg 132] Similarly, David Raksin, film composer, has written that the purpose of music in films is "to help realize the meaning of a film." Aaron Copland on the other hand has said that a composer can do no more than "make potent through music the film's dramatic and emotional value."

Film in its traditional linear format has a beginning, middle and end, which is set, defined and unchangeable. Therefore, the construction of music in this case, is reliant on this nature of the medium. In other words, it is composed with a set sequence of events in mind. The composer here is bound by this sequence of events that make up the plot/story of the film. In no way is the sequence of events possibly going to be manipulated in screening by 3rd-party intervention. So, construction of music here, although determined by strict guidelines from the plot narrative, is similar to musical construction of a programmatic musical work. Programmatic refers to music that has a story or narrative associated with it such as the score to a set libretto or text. The music is thus constructed using the sequence of narrative events, and reflecting them as intended to be played from start to finish of the event sequence. The composer writes his ideas down, scores them as per the narrative and allows them to flow into each other in one direction only – from beginning to end.

To help illustrate the construction of music in linear film, the following example of a film score shows how the composer is affected by the linear sequence of events unfolding in the space of a linear film. [Karlin & Wright. 1990. pg 133]:

Fig. 2. Extract from score to the "Wrath of Khan", by James Horner



Looking at this score, one can see the dialogue inserts which are inserted as they occur in the film. The position of these insertions is determined by the time track indicated by the numbered boxes every 2 bars or so. This illustrates how important timing is in determining the position of elements such as dialogue, acting cues and music. The music here is written to slot in with these indications, thus demonstrating the reliance on a fixed time-line or sequence of events in linear film composition. This also highlights the programmatic nature of linear film composition, by pointing out the score's reliance on the written narrative.

The only possible way of affecting the linearity of this kind of example is through repetition. Should the viewer decide to repeat the viewing of a particular scene and rewind or forward and play that scene over again, the same music heard previously will be heard, but the music itself will remain unaltered, the only change being that of the macro-compositional structure in the seating in which the repetition takes place.

NON-LINEAR COMPUTER GAME MUSIC AND SOUND ELEMENT CONSTRUCTION

Composing music for a non-linear medium requires a number of similar and also differing techniques from writing music for a linear medium. Similarities include the requirement that the composer be able to establish what the general feel or ambience in the medium is. Is it mystical? Is it futuristic or is it a morality scenario?

In non-linear computer-game-music construction, there are usually several types of sound that make up the overall sound environment. These could include (depends on what is required in any particular game), ambiences, music, sound effects, and dialogue. As previously explained, creating a successful "interactive" sound world would mean having to successfully combine all these elements and make them morph and change according to user input.

The ambiences and musical elements are constructed as loops, unless some musical or ambience sounds are intended to be played only once or twice, in which case they will be segments layered on top of the looping element. The loops play continuously and help to support any musical themes or emotive phrases which are layered on top of this loop in a random fashion to avoid any form of familiarity of the audio developing with the user. Loops are a necessity in games because the composer has no clear idea of how much time a user may spend in each scene or section. The solution then is therefore to create fairly long loops that don't contain too many specifically identifiable elements to again avoid familiarity breeding with the player.

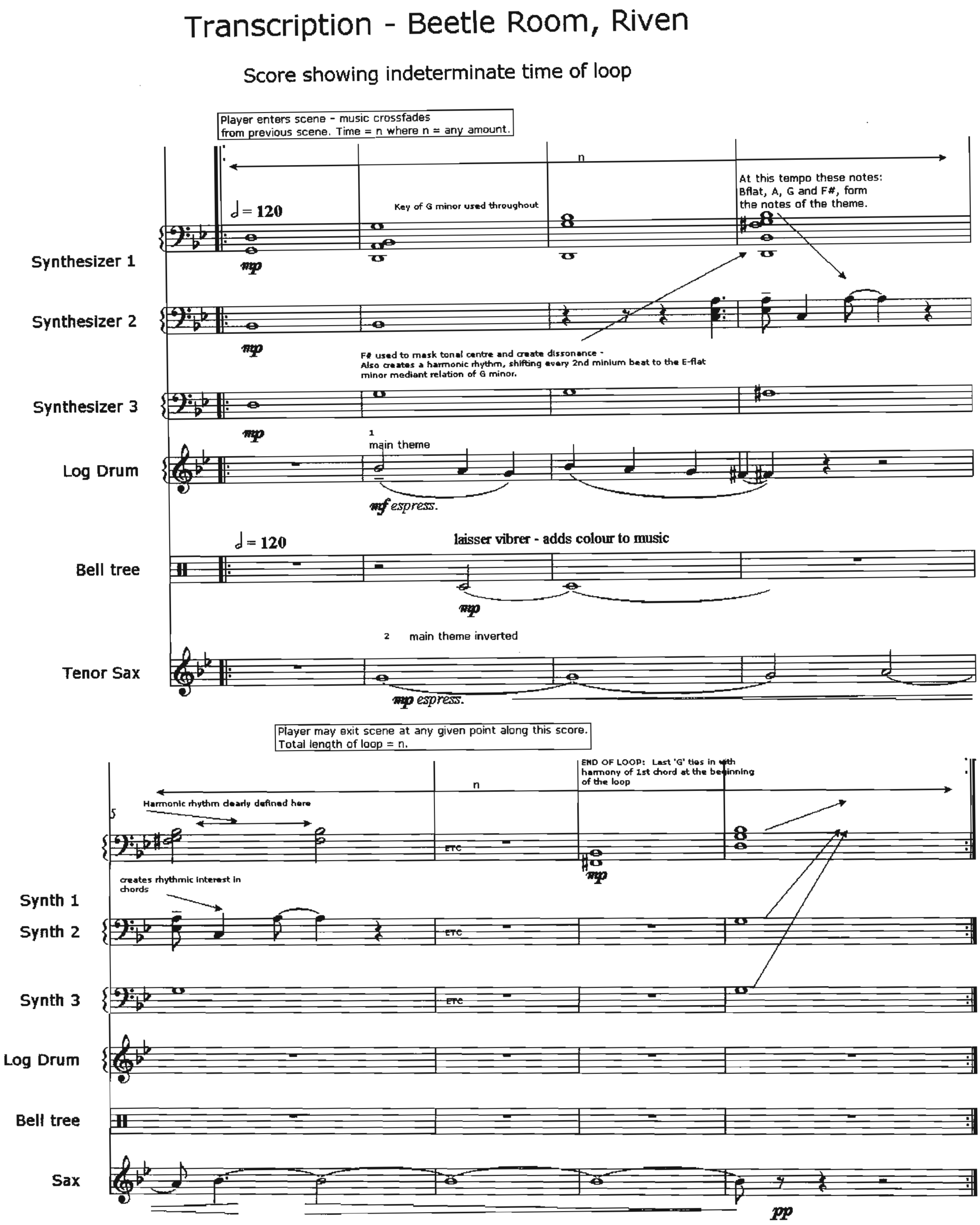
The loop and the elements should also make a clear reference to the scene to aid in providing sensible emotive stimulants.

In further consideration of game ambiences, they are present to help "immerse" the user in the game environment. The sound that you hear playing may indeed reflect what you see on the screen, but it is likely to be less two-dimensional than in linear film ambience, in that it may appear to be coming from outside the scene, acting as clues or references to the plot itself. Without the gamer taking note of these references, he/she may find it very difficult to complete the game.

The music in non-linear computer games must be able to move in a random sequence completely unreliant on a 'timeline' - it has no specific duration or mid-point. I say mid-point, as there is clearly a defined beginning and end to a game, but the player may re-enter the game at any given point, and actually end up further from completion based on decisions made, thus creating the ambiguity of a fixed mid-point. It is clear to see that the very interactive nature of the computer game – the idea that the consumer chooses a unique path through the material, enhances the non-linearity of the medium. The composer in this instance, must therefore write music that is capable of adaptation and seamless mergence from scene to scene. Because of the advent of indeterminate player input in non-linear games, the musical material of each scene should remain in the same key, to allow for homogony of sound between scene changes. Musical elements should also be kept down to a minimum, as too many repeated over and over, will also wear thin very quickly to anyone playing the game.

Depending on the design features of the audio engine responsible for the sound implementation in a game (audio engines are looked at in more detail later on), the composer may have the advantage of composing a few musical elements of short length, that themselves could have hundreds, if not thousands of permutations. The gamestate could in this case call for permutations based on a series of parameters, such as tempo, key, instrumentation and effects, to be played back when required. These elements would need to be constructed in a way to allow for all these permutations and at the same time hold thematic relevance. This does have the effect of making the construction of a musical engine complicated. Harland explains: "We have to generate a dynamic musical engine that must accomplish what a film score does for dozens of hours of unpredictable gameplay."[Harland, Kurt. 2000] (To illustrate the indeterminate time of any given scene, I created the following score from transcribing the music in one of the scenes from *Riven: The Sequel to Myst* – See FIG. 3)

Fig. 3. Time Score from Riven Transcription


Transcription - Beetle Room, Riven
Score showing indeterminate time of loop
Player enters scene - music crossfades from previous scene. Time = n where n = any amount.
n
♩= 120
Key of G minor used throughout
At this tempo these notes: Bflat, A, G and F#, form the notes of the theme.
mp
Synthesizer 1
Synthesizer 2
F# used to mask tonal centre and create dissonance - Also creates a harmonic rhythm, shifting every 2nd minium beat to the E-flat minor mediant relation of G minor.
Synthesizer 3
1
main theme
Log Drum
mf espress.
♩= 120
laisser vibrer - adds colour to music
Bell tree
2 main theme inverted
Tenor Sax
mp espress.
Player may exit scene at any given point along this score. Total length of loop = n.
END OF LOOP: Last 'G' ties in with harmony of 1st chord at the beginning of the loop
5
Harmonic rhythm clearly defined here
creates rhythmic interest in chords
ETC
Synth 1
Synth 2
Synth 3
Log Drum
Bell tree
Sax
pp

The line "n" on the score indicates the length of time spent in the scene by the player. The amount of time is indeterminate. This means that at any point, the player may decide to exit the area, which means that the composer must allow for this by ensuring a smooth change with the next scene at any point during the playback of the music for the current scene the player is in. This may be compared to the film score previously, which highlights the non-linearity of the example here and vividly shows the differences in approach to time in the two media.

Furthermore, it is possible to enhance the user's experience in non-linear computer games, through the employment of the techniques discussed under the heading of "Interactivity: A definition." Todd Winkler puts it this way: "Continuous control over musical parameters adds a level of variety and interaction not usually seen in commercially available titles."[Winkler, Todd. 1998. p. 297] This is further expanded on in Winkler's discussion on screen-based works: "In addition, constrained random values can be used to create endless variations automatically so that the music is different with each viewing." [Winkler, Todd. 1998. p. 297] These "constrained random values" are what I have referred to as elements. Winkler talks about them as being constrained, in other words unchanging or fixed. However, it is now possible to achieve un-constrained values, by making the elements that are randomly called up, change within themselves. For example, you create a short musical motif or element that is to be called up randomly but will be repeated at some stage. Instead of the original element being repeated as an exact replica of the original, it itself will now be different, and different each time it is called up. This will further be explained under the heading "Interactive Audio Engines."

In computer games, the outcome of scene repetition with regards to music can be very different linear media scene repetition. "Repetition seems inevitable in these situations as the user stumbles back to previous scenes." [Winkler, Todd. 1998. p. 297]

However, this does not have to be the case. In titles that exhibit very well constructed interactive music, each time the player goes back to a previous scene for example, the music might change due to certain changes the player has made within that period of the game. Winkler explains: "The success of the commercial title '*Myst* demonstrates how two devices can make this repetition bearable and even interesting: Firstly, the images and sound can be complex and highly detailed, so that on the 2nd or 3rd look, new aspects are discovered; secondly, upon returning to scenes that appeared to be static, images and sounds can be transformed to effect changes in the story." [Winkler, Todd. 1998. p. 297]

This occurs as a direct result of user interaction. This may be possible in film for example, only here the consumer has no power over the changes made.

Further methods of creating interactive music in our non-linear medium of computer games, could include the use of input devices (keyboard, mouse, joystick, etc), to effect changes in the audible output through the physical input of the user. Winkler understands the importance of this and explains it thus: "...the timing of decision making, mouse movements, or keyboard strokes, could be linked to control the tempo of the music, or clicking and dragging an object downward might cause the pitch to go lower or the timbre to get darker." [Winkler, Todd. 1998. p. 297] This kind of user input is obviously not possible in film. It may come close with the music or sound mimicking action on the screen, but is not controlled by the 3rd or 2nd person. Another aspect of timing in computer games, an aspect not possible as a parameter for interactivity in linear media, is the ability for non-linear media to effect the level of intensity in a scene by virtue of a user's decision to remain in that scene. "Parameter changes could also be controlled by the time spent within a single scene, so that the music would intensify as the viewing continued." [Winkler, Todd. 1998. p. 297] The linear composer by comparison is forced to create music based on pre-determined time frames that will never change.

For the purposes of further understanding the applications and full nature of musical construction in non-linear media, I have included a diagram that highlights possible interactive pathways and their relevant outcomes in non-linear media. (See Fig. 8, p. 65) Looking at this diagram, we notice that the programming of an interactive event such as a non-linear computer game will have a similar set of 'guidelines' to those making up the plot of such game. It also aids in explaining the reasons for specific techniques used in the construction of non-linear music. It highlights the multi-path possibilities in hypermedia which form the basis for interactive media construction. This means that should music be written for such media, it too will be affected by these multiple pathways. This implies that on a macro level, the combined sound of any one sitting will always be unique.

Immersion of the user in the game may further be enhanced by sound effects. Sound effects in computer games are in a way analogous with foley sound in film. Sound effects are sounds such as footsteps, door openings/closings, machine sounds etc. [Stevenson, Robert. 2000. pg. 2] Often the actual sounds these objects or devices make is recorded and then used as is in the medium. This further enhances the "realism" of the environment by presenting the user with noises one would hear if you were actually *in* the environment. They can be highly interactive, as in some cases they are decidedly reliant on user input for rendering. Using the same example as described in the previous chapter, walking down a forest path in the game “Uru – Ages Beyond Myst”, one hears the crunching of leaves underfoot as you walk on a forest path. The second you stop walking – the sound stops.

Dialogue is another sound element found in some computer games. Computer games that require dialogue use it as a means of communication between characters. The dialogue sound files, like all the other sounds, are pre-recorded and called upon when needed. The difference here is that instead of being used often, as for example, sound effects and certain musical elements such as loops, random elements and clue sounds, dialogue is only called upon in very specific instances i.e. when it is necessary to communicate with a person or character. This is however, more often than not, presented in the form of a soliloquy as a medium of instruction or story/plot exposure from one of the game characters in the first person narrative. In other instances, there is the opportunity to provide the player with a text-based input that could be responded to with a recorded dialogue running through an audio engine capable of advanced text recognition and thereby provide a suitable response to user questions etc. e.g. text and audio dialogue interaction in “gammaKhozi” [Amory, A. VLS. 2004] Sometimes however, dialogue does not occur in the form of recorded voice and merely appears as a text-based input and response. Dialogue in either case is interactive, as the player is initiating a response from the interface based on the choice to be where the dialogue occurs and the player could then respond to the dialogue in an appropriate way.

In consideration of some other aspects of non-linear sound design, as recently as 2003, a paper exploring teaching methods of spatial design, has pointed out the need for game creators and composers alike, to further explore and implement even better designs with regard to collaborative environments. A collaborative project between the universities of Auckland and RMIT University, Melbourne, was initiated to test the idea of creating a virtual environment that uses both sound and space to extend educational methods for spatial design education and for design and architectural practice. Basically, the idea was to create a new educational tool that focused on non-linearity as opposed to linear media such as rendered images, linear animation or sound recording. The game created was called “Memory Games”. It focused on users creating their own spatial design around the Melbourne CBD, using sound to enhance the spatial design. The sound referred to here is not ‘music’ per se, but sound effects that represent the implied surroundings created in the virtual space. Musical phrases, elements or even individual notes as well as different instruments - electronic and acoustic, representing a vast array of textures and timbres could just as easily be used instead of, or in conjunction with, natural sound effects. One of the key elements of this game is described as: “the relationships between sound and visual texture and their role to evoke memory and furnish new narratives.” [More, Gregory. 2003]

This seems to be a very important development as it allows for the design of a non-linear medium that is capable of “real time manipulation of sound and image where there is no distinction between editing environment and the presentation output.” [More, Gregory. 2003] To me, this is a kind of hands-on interactive medium. In other words, it is a game environment teaching design that is organic and realistic, in that it provides the designer with an environment that is extremely evocative and almost “natural”. More further clarifies this: “The sound overlaps and colours the visual, triggering alternate interpretations of geometry and surface, simultaneously informing the raw sound sample and resulting in some cases where the layered visual and soundscapes operate as “abstract machines of stratification.” [More, Gregory. 2003]

The latter seems to me to represent a trend towards the design of screen-based works that resemble a kind of improvised composition - the user designs a space and creates sound for the space and the two factors in turn affect the design and sound and vice versa.

On the other hand, certain linear works can represent an ‘implied’ interactive pathway. Only in this situation, the experience of the user is very personal and limited to his/her personal emotions. More describes the differences between non-linearity and linearity in this regard thus: “It could be argued that the choices made by a listener/viewer when confronted with a complex polyphonic orchestral composition, or the dense visual textures found in action painting, constitute unique pathways through a work, similar to those in a non-linear environment.” [More, Gregory. 2003]

This can easily be verified through personal experience of such works, where very often each new listening or viewing would create new experiences for the user and would differ vastly from user to user. In the computer game scenario, the presence of the person can be represented by an avatar and therefore the experience can be shared by many individuals.

With current technologies and new technological developments, the evolution of such non-linear environments will most definitely be a very exciting medium to watch. More describes this as: “the full potential of new learning environments combining virtual collocation and communication, with vivid aural and visual datascapes and based on non-linear forms of organization is still to be explored.” [More, Gregory. 2003]

In a final word from the latter and very informative article, More quotes Antonia Vizcaino’s description of Umberto Eco’s notion of the “open work” or non-linear medium: “(‘The poetics of the open work’), which considered the way modern music (Stockhausen, Berio, Boulez), modern writing (Mallarmé, Joyce), modern art (Calder, Pousser) in relation to modern science (Einstein, Bohr, Heisenberg) now produce ‘works in movement’ and ‘open works’, works whereby the addressee becomes an active element in bringing a work to provisional completion, or where the work itself brings openness to the fore.” [More, Gregory. 2003]

Non-linearity is thus the preferred form of media creation for the computer game, as the computer itself allows for this “openness” by virtue of the ability for the medium to incorporate user interaction. It is now necessary to consider and chronicle the development and design of some important gaming applications, hardware etc., since it started to become available for public consumption.

INTERACTIVE GAME MUSIC

There are many varied different types of computer games available today. These include arcade, game console, PC, Sony PlayStation, internet games and more, all exhibiting differing approaches to style, presentation, theme, target market etc. There are many that have been around for quite a number of years now, such as, Pac-Man, Tetris, Space Invaders, Super Mario Brothers and the like, as well as those long since forgotten, such as the text based work seen in the Zork series, where there is no audio whatsoever, apart from the odd onomatopoeia indicated by the program. In order to provide a contextual picture and background information regarding the development of commercial computer game titles and interactive game music in general, I have included a short history of this development, focusing on the important creations and discoveries, highlighting and illustrating the simultaneous development of interactive music or sound as the case may be: (The timeline was created with aid of Glenn MacDonald's "A brief timeline of video game music.") [MacDonald, Glenn. 2004]

1972:

The year in which the 'Magnavox Odyssey' was released. Designed by Ralph Baer in the 60's initially for the US military as a skills training device, it was released 8 years later into the commercial world. It became the first video game console running on an analog system. It contained twelve games, the most popular being 'Ping-Pong'. There was no sound whatsoever. This system was succeeded by 'Pong', developed only two years after the commercial release of 'Odyssey', by Nolan Bushnell. This first-ever arcade game had one up on the 'Odyssey' – a sonar blip that played back every time the ping-pong ball was batted back and forth.

1974:

This year saw the release of the very popular 'Simon Says'. The game console played back patterns of 4 different lights each accompanied by its own tone. The player then had to repeat what was seen and heard. Each time a successful repetition was performed by the player, another note/colour or two was added to the sequence. In a crude way, this console game could be construed as being the first ever 'interactive game audio' device. It was interactive in the sense that the user had to memorise what was played back and then perform the identical performance back to the machine.

1978:

Taito produces 'Space Invaders'. The game had large appeal due to its soundtrack, which for the first time was adaptive to the graphics on the screen. As enemies approached, the music increased in tempo, adding to the urgency of the moment intended to create panic in the player.

1979:

Atari develops 'Asteroids' with added sound effects over the primitive but adaptive audio track similar to 'Space Invaders'. 1979 was also the year in which the Atari 5200 was developed, along with the release of 'Pac-Man'. The latter game plays back energetic loops comprised of many beeps and single tones, which intensify as danger gets near or if a level is completed.

This was also the year which saw the release of the arcade game 'Q*bert'. This was the first instance of artificial voice creation intended to make the game character speak. It was achieved by programming random numbers into the voice chip whenever the character died. Should the player accidentally make the character fall off his pyramid perch, a 'thunk' sound was played back. This was pioneering in terms of player input creating an indeterminate audio response in relation to timing from the game.

1985:

'Tetris', developed by Russian programmer Alex Pajitnov, became an all time favorite amongst the global gaming community. The soundtracks that accompany the game graphics are effective yet constantly repeating loops of Eastern European folk dance tunes. The music changed from level to level, but was only interactive on a purely subliminal level, in that it engaged the player in the game by creating a 'hypnotic' effect.

1985 was also the year in which Nintendo released 'Super Mario Bros.' This game contained the first specifically composed audio track, by composer Shigeru Miyamoto. Specifically composed music refers to music that underscores the narrative of the medium – i.e. not using currently existing music composed for an unrelated topic or performance. An article on game history describes Miyamoto's music in this case: "Constantly shifting tone to match the action onscreen, Miyamoto's sound design achieves a new kind of synthesis with the gameplay. Try playing the game with the sound off, and you'll quickly miss those music and sound cues--for example, the exact timing of your immunity power-up wearing off. With the Super Mario Bros. soundtrack, video game sound design begins to move in a new direction, away from cinematic conventions and toward something altogether new." What MacDonald describes here as 'shifting tone', refers to music that changes to describe the current action on screen. The composer achieved this by making sure that when called up by the gamestate, the music for any specific area added emotional impact to the scene. 'Cinematic conventions', also refers to an attempt to emulate film music's ability to enhance action on the screen.

1987:

Composer Nobuo Uematsu writes music for Square's 'Final Fantasy'. As with the 'Super Mario Bros.' release in 1985, which attempted to create a cinematic feel as earlier described, the music here *was conceived* in a cinematic vein, attempting to emulate the emotional potency one finds in film music and proving to be very successful to this day with both gamers and game historians.

1991 to 1993:

Multi-channel audio streaming was developed by Nintendo with the release of their 'Famicon' product. The system allowed for 8 separate channels of audio playback. This initiated many advances with regards to game music development. Allowing for simultaneous multiple channel streaming, meant that the game audio produced became much richer in terms of texture and overall sound impact. Game composers could utilize this advance, by creating layered music or sounds.

In 1993, Cyan released the first game in the famous Myst series which started to set the benchmark level in terms of immersion. The soundtrack was recorded using a live orchestra and bought a new level of cinematic realism with it, by doing exactly what film music does, in that it emphatically addresses the environment the gamer is in. The sound effects too were produced and implemented as in film, further adding to the realism.

1994:

A very important year for game music development and recognition. It saw the sixth release of Square's 'Final Fantasy' series. MacDonald describes it as follows: "A great example of Uematsu's brilliance, this soundtrack demonstrates the increasing sophistication of video game music. Character-specific leitmotifs recur throughout gameplay, and the sheer variety of styles employed is audacious. Uematsu is deservedly compared to film composer John Williams. (The game's soundtrack would ultimately place first in GameSpot's Readers' Choice of the all-time greatest video game soundtracks.)"

1995:

CD quality audio streaming through a 24 multiple channel sound processor was released by Sony Corporation in this year. It was also the first to allow built-in digital effects playable back in real-time. One of the games analysed in "Gaming Applications – Commercial Titles", later on in this article, makes use of these features very thoroughly. It allows for greater interactive sound implementation by giving the opportunity to playback many different sound files at once that could be triggered through gameplay. The resultant effect was that game developers, including composers, now had for the first time, the opportunity to explore and develop the concept of immersion of the player in the game environment, by creating as realistic as possible sound design.

1997 to 2004:

In 1997, Cyan released the sequel to '*Myst*-'*Riven*', which saw more interactive elements coming to the fore as far as the sound is concerned. Orchestral music with themes capable of merging in a seamless cinematic fashion, but fully non-linear, set new standards in recognition of this game music's ability not only to enhance mood and atmosphere, but also provide clues to the player. (The music is analysed in more detail in the chapter on "Gaming Application – Commercial Titles – *Riven: The Sequel to Myst*, later on in this paper.)

The same year produced the release of "The Legend of Zelda: Ocarina of Time" This Nintendo game was the first title to incorporate user music composition into the gameplay. MacDonald explains: "In the game, you use the ocarina, a kind of flute, to teleport, open portals, or summon allies. There's also a musical puzzle in which you must follow the bass line of a song to make it through the Lost Woods."

Sega Games developed sophisticated voice recognition software for their Dreamcast[1] game 'SeaMan' in 2001, which allowed the user to have a two-way conversation with game characters on "everything from politics to baseball."

The Sony PlayStation2 was also released in 2001, with new technology allowing for the development of the following game as described by MacDonald: "Released in Japan and the UK only, the truly strange 'Vib-Ribbon' takes the relationship of music and gameplay in an entirely different direction. Playing the rabbit like creature Vibri, you must navigate levels that are themselves determined by the music track that's playing. Moody mope-rock equals slow and steady; frantic techno equals fast and furious. The kicker is that you can pop your own audio CDs into the PlayStation to generate entirely new levels based on the tempo of the music."

The years between 2001 and 2004, has seen the advent of the expansion of technologies developed in 2001 and the release of many titles exhibiting more and more interactive music. These include, relating to my own analysis, "Riven: The Sequel to Myst"and "Uru: Ages beyond Myst" for PC, "Grand Theft Auto III" for Sony PlayStation2, and "Tron 2.0", also for PC platform.

Expanding on the above timeline, the earlier games were of necessity primitive, limited as they were by technological and financial constraints of the time. This had the affect of making the music appear inadequate. What it did do though, was make vivid the necessity for audio development in computer games, and provided the basis for analysis based on comparisons with today's technologies. By this I mean, it proved useful to compare the methods and tools used to create the music for early titles to establish what changes have been made with regards to technology for the implementation of music in current titles and what this ultimately means for the music.

Therefore, beginning with the early titles from the 1980's, such as 'Pac-Man', 'Tetris', 'Asteroids' and the like, in many cases, if not all, there is a feature allowing for the selection of different tunes that one could select from a control panel. But these tunes would play as a long loop, never changing in the background, apart from the odd sound effect being

[1] A powerful 128-bit central processor and super intelligent sound processor, which has a 32-bit RISC CPU developed in 1999 by Sega, for their Dreamcast Console product.

http://www.gamespot.com/gamespot/features/video/vg_music/p6_02.html layered on top of the background loop. This might change from level to level, but nevertheless, it looped endlessly. Generally, some of the songs were well crafted as the producers of the games on the whole wanted composed music to be played, as if listening to a radio, attempting to create a more entertaining environment. But after a while this gets tedious - Kurt Harland has said, "Good songs that repeat themselves over and over are a kind of psychological torture that only our modern culture could produce."[Harland, Kurt. 2000]

Further highlighting the expectation of completed music by the producers, commercial titles were required to be released with every element of the title to be of a very high quality. At the time, this meant that the producers wanted completed music or "songs" that would play with the game. This type of music was and is certainly non-interactive and linear in nature. Harland clarifies this, "In 1994, reviews were still ready to criticize the soundtrack if it didn't sound like music in the traditional sense." [Harland, Kurt. 2000] As per reasons explained under the heading "Musical Construction in Linear and Non-Linear Media", music for interactive titles is not going to sound like "finished" music if you had to listen to it from the start of play to the end. There are far too many parameter changes to allow for this.

The use of "finished" music worked quite well for many years, until games became more complex with regards to thematic material and overall design, which required developers to think of ways to further enhance the game play experience. This becomes evident when looking at the development of computer games over the last 10 years or so. The early titles such as the ones mentioned above, had two-dimensional graphical interfaces, often in monochrome colour, combined with very simple technology compared with today's standards and therefore had very simple means of providing audio playback. The audio was invariably monophonic and produced by tiny oscillators often built into the game console or produced by a small PC speaker as a series of beeps. Over time, with the advancement of technology, the production of audio was enhanced through the development of the "audio engine" – see chapter "Interactive Audio Engines". Interestingly, the development of technology in this regard, was directly proportional to the requirements and demand from the computer market. The consumer wanted better entertainment, and better entertainment meant better visuals, better content and realistic audio. So as the technology developed, it became possible for game designers to develop games with complex plots, high levels of playability and immersive environments, pointing the gaming world in the direction of the 'real' and at the same time, providing competition for the film industry. This is especially true for games that have a fantasy, mystical or puzzle-solving theme. Games such as *Myst*[1] and *Riven: The Sequel to Myst*,[1] are good examples of such games. (These and other titles will be analyzed in more detail under the heading "Gaming Applications – Commercial Games.")

As touched on earlier, film music reflects the scene on the screen - i.e. a murder scene would have tense, suspense filled music, a marriage scene would have wedding music and so on, game music should allow for the same association between image and sound, only with the availability of 'interactive software', game music may have variations within each theme, so that if a scene is repeated by the user (a decision in a puzzle game for example, to go back a scene or two), the theme for that particular scene may not sound the same again.

It is however pointed out by Alexander Brandon that: "Developing and establishing a theme before introducing variation, while not absolutely necessary, adds a great deal. It familiarizes the player with the game in a more intimate way than constant, pervasive variation can." [Brandon, Alexander & Miller, Mark. 1999]

Alexander Brandon and Mark Miller, further emphasise this point by describing what they term "thematic development" [Brandon, Alexander & Miller, Mark. 1999]. They point out that thematic development is "crucial to the successful realization of real-time, interactive music." [Brandon, Alexander & Miller, Mark. 1999] In other words, the idea of thematic development, of buildup and resolution, is very important to the successful implementation of an interactive title.

Thematic development refers to a constantly changing material source that is directly affected by a storyline, plot or as in game music, a player's decisions. In terms of music, this is done by composing in such a way, that musical themes constantly change, thus avoiding any boring repetition. This in conjunction with interactive audio engines[2] can be applied very successfully with the result of a truly immersive environment. It is exactly this that separates ordinary game music from 'interactive' game music.

[1] Cyan Productions, 1997

[2] See Chapter on Interactive Audio Engines

So, through the development of new technologies, the capabilities of games to achieve interactivity in terms of both the gameplay and the music became more apparent. It is clear now that designing interactive music will require the composer to adopt new methods of writing music to allow for the successful implementation of interactive, non-linear music. So what are the parameters the game composer needs?

In conclusion here, highlighting the key issues identified through the research relating to "Interactive Music: A Definition", "Linear and Non-Linear Media" and "Musical Construction in Linear and Non-linear Media", several elements pertaining to non-linear musical construction were extracted and investigated. These included:

- As simplicity is paramount, short musical elements must be composed and played back in a repeating pattern with differing lengths between repetitions to avoid familiarity with the gamer.
- The latter may also allow the composer to write a simple musical element, but that which could have thousands of different permutations and therefore further decrease the risk of familiarity with repetition whilst still maintaining continuity with the background loop and any previously played musical material.
- The requirement for smooth transitions from scene to scene, encourages the composition of background loops in similar musical keys.
- The properties of non-linearity and the definition of interactivity and immersion, require the composer to develop techniques and ways of implementing these concepts. Music and ambience must blend and compliment each other in such a way so as to avoid conflict within each sound element and should be both designed in such a way as to enhance the environment the gamer is playing in.
- Identifying with the latter environment will enable the composer to further enhance interactivity by creating links with the plot itself. This means for example that should a player come across an important object in the game, the composer could write in musical clues that would be played back identically when the player is later exposed to the puzzle the initial clue made reference to.
- Knowledge of the audio engine and its features is instrumental in composing music for non-linear computer games. This is made obvious in the analysis of the construction of music in non-linear media as pointed out in the chapter "Musical Construction in Linear and Non-linear Media." Interactive music composition is therefore dependant on the audio engines interactive capabilities and features.

Features and applications of various audio engines will be investigated in the chapter “Interactive Audio Engines.”

INTERACTIVE AUDIO ENGINES

An interactive audio engine is a piece of software designed specifically to process commands from the gamestate[1] that are sound or music related. Game developers use the audio engine as a tool to make possible the audio in a game environment. The audio engine would then call upon a set of sounds or sound files to be played back as and when they are called upon by the gamestate. The game audio engine therefore works in tandem with all the other essential components that make up a computer game such as: [Stephens, Noel. 2001]

1) A Renderer
2) A physics engine
3) A particle system
4) Some form of a cinematic system
5) Some form of an entity manager
6) Some form of sound handling routines
7) Tools to interact with the above said systems
8) *anything else you might be able to think of*

With reference to the above list, for the purposes of this article, I will focus on points 6 and 7, and specifically anything to do with audio.

Note:

The personal computer (PC) world uses several branded tools to help realize the production and implementation of multimedia applications. In fact, they are numerous in quantity and it would be impossible to list them all. It is for these reasons that I will just list a few and then talk about their role with any specific game audio engines. Incidentally, it should be noted at this point, that all computers require some sort of software component to display video and stream audio. Microsoft uses an application called Direct X, whereas Macintosh uses Apple QuickTime. It is important to note that these 'built-in' (usually coming bundled with the operating system) applications, allow the game engines to perform their tasks through the computer's operating system.

[1] See Terminology.

The history of PC audio engines shows that the technology was slow in developing to where it is today. This as a direct result of the technology employed by the hardware present in the average PC. Many PC's manufactured right up to and including 1997, had very basic sound cards in them, capable of performing very ordinary audio playback functions. This was also the case in the open market. This was of course extremely limiting to game developers at the time, as their products had to suit the technology available. This meant that game audio composers employed by such developers were similarly limited in what they could do musically in a game. It was in 1999 that soundcard manufacturers cottoned on to the gaming market to provide consumers with more effective equipment.[1]

This meant that across the industry, especially in the gaming world, developers and the like could achieve much more in their design and overall product quality and capability. It was as a direct result of this, that game developers could design game audio engines that better suited their needs and could successfully be implemented with current PC hardware technologies.

These soundcards or hardware audio engines, will be discussed in more detail under "Other Audio Engines", but firstly, I will discuss a product created by Microsoft called DirectMusic Producer. It was developed in late 1999 to facilitate an audio engine for game developers that could be used with any sound hardware system present on the consumer's PC. It was conveniently packaged with the latest DirectX software version as the "note" describes.

Secondly, I will talk about the sound engine designed by the Virtual Learning Space Project (VLS) – VLS Sound Script Editor, which is adapted and based on some integral features of the commercially available stand-alone sound engine FMOD.

The reasons for doing analysis of these two engines, is that DirectMusic Producer comes packaged with the latest DirectX version for PC and the engine allows for individual element manipulation by user input in a gamestate and therefore has the ability to adapt spontaneously. It is also a MIDI based engine, in that the audio is triggered via MIDI data, which is a feature not employed by many interactive audio engines. The second engine

[1] See http://www.soundblaster.com/scripts/review_frame.asp?url=http://www.gamersdepot.com/rev_sblive_plat_a.htm for a review of the SoundBlaster *Live! Platinum* card which highlights some of the critical features developed at that time for better interactive user experience. Review is dated 25th October 1999.

(VLS Sound Script Editor) will be analysed, as it will be used in the creation of the music for the Portfolio of Compositions related to this paper.
And finally, I will briefly list and discuss other game audio engines that are of interest with regards to interactive music composition.

DIRECTMUSIC AND DIRECTMUSIC PRODUCER

DIRECTMUSIC

DirectMusic (referred to hereafter as DM) is software designed to be integrated with the Windows operating system. It runs as an audio engine, providing the base over which DirectMusic Producer (referred to hereafter as DMP) can work. It also works in tandem with DirectX components. These two applications provide the necessary tools to enhance computer potential with regard to the accompaniment of music and sound to a variety of computer applications, including: [Microsoft. 1999][1]

- Web sites
- Computer games
- Live multimedia presentations
- CD-ROM multimedia applications
- Audio examples placed in word documents, spreadsheets, or other files created in a Windows-compatible application
- Any other multimedia experience on a computer not listed above.

DM therefore provides the necessary platform to allow for access to the sound capabilities of a Windows based system. In the help file on DMP, Microsoft developer Tom Hays describes the abilities of DM as follows: "As the author of the music content, you do not work directly with any of the DM components in the operating system. Instead, you use the DMP application to create music files that play through and are processed by DM. This takes advantage of the flexibility and power the DM components provide."[Hays, Tom. 1998]

DMP allows the composer to create scenarios in which the music can be easily and flexibly manipulated, depending on what the inputs are. For example, it can simply play

[1] References to Microsoft here and after refer to general comments released by the company and do not relate to any specific comments made by Microsoft employees.

back a MIDI file at any chosen tempo, with any combination of sound effects, other MIDI files and files with built in variability. DM is then used to control and process the information provided by the composer in DMP. What happens here now, is that the timing and manipulation of any music output is linked to the actions of the user. These include mouse clicks, mouse movements, keystrokes or touching the screen if a touch sensitive screen is the input source.

Finally, and more importantly, DM provides the composer with the ability to use his/her own sounds, completely independent of the sounds found in a tone generator or sound module. Here, DM has the ability to behave like a sampler. The Interactive Audio Special Interest Group (IA-SIG) created the standard of Downloadable Sounds (DLS) which allowed people using PC sound cards and software synthesisers to create their own set of sounds as opposed to being limited by a fixed sound palette such as that found in a tone generator. DM uses the DLS standard. Ultimately, what this can achieve as far as the composition and design of music for a computer game is concerned, is that when the final product is pressed to cd, it means that all the sounds required for the game can be loaded instantaneously at the start of the game into the DM engine and recalled at any time when needed during the game.

DIRECTMUSIC PRODUCER

DMP is the application that the composer uses to create files that are eventually processed by DM. DMP therefore contains the material that the consumer would interact with, once the finished product is placed in the application such as a computer game or web page.

It is at this point, that various technicians, such as software developers and programmers making use of DMP, will write the necessary code that will allow for the integrated performance of the music with DM running on the operating system.

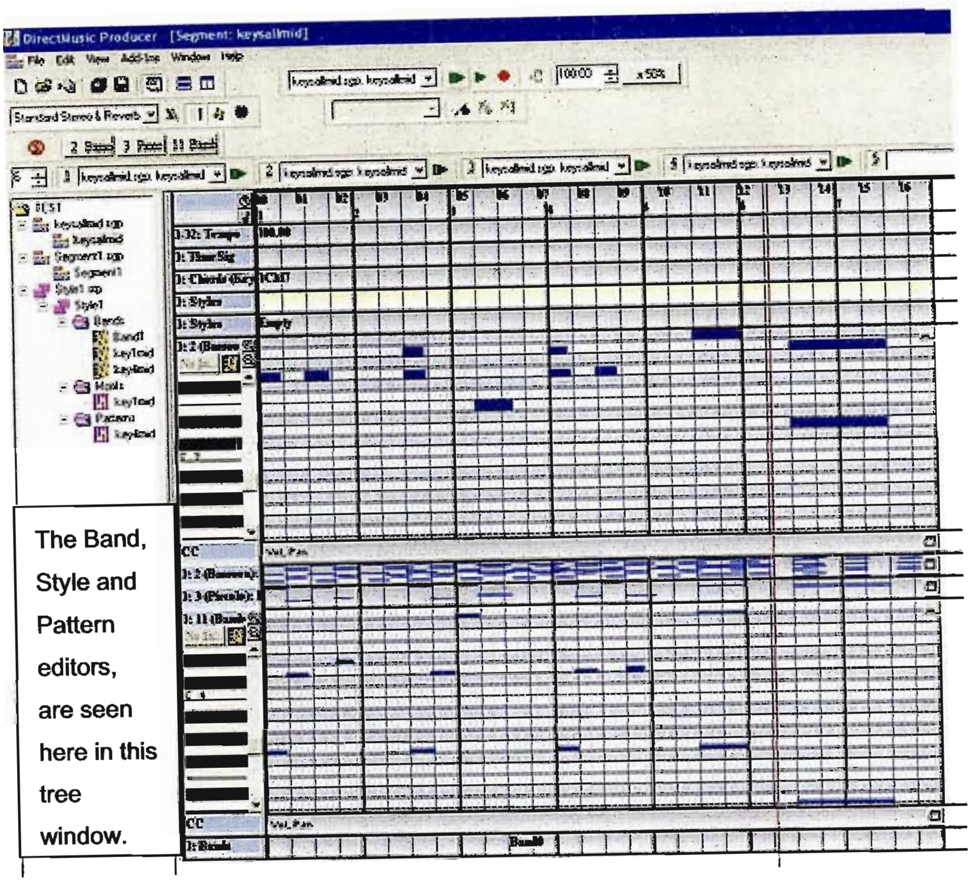
Furthermore, DMP allows the composer, to use either existing MIDI sequences or to create new MIDI files which may be used in combination with the DLS to create the required sound.

DMP has the ability to allow the composer to incorporate increased variability in the music. By this, I mean there are many options it can follow, such as different chord choices, alternate melodies, changes in rhythm, and increased or decreased intensity.

Interactivity in DMP is achieved in a number of ways. By linking the timing built into an application (such as a computer game) or by linking any action made by the computer user such as moving the mouse or pressing a key, DMP will respond to those stimuli in the programmed way. The program allows for instant movement from one chord or melody set to the next, which allows for smooth transitions from scene to scene (in a game) or page to page (in a web application.) [Hays, Tom. 1998] To allow for these interactive features, DMP provides the composer with what is called a "Style" and "Style Editor". [Microsoft, 1999] FIG. 4 below shows what this window looks like. A "Style" is comprised of three different elements. They are "Bands", "Motifs" and "Patterns". [Microsoft, 1999] The "Pattern Editor" [Microsoft, 1999] is the central work area for creating "Styles" - see FIG. 5. below. Hays describes this as "a sequencer that allows for real-time recording of MIDI data directly into the editing window." [Hays, Tom. 1998] The composer can chose between either piano roll information or traditional notation to be displayed. A controller track is also available for entering or drawing in information for pitch bend, volume, pan, etc.

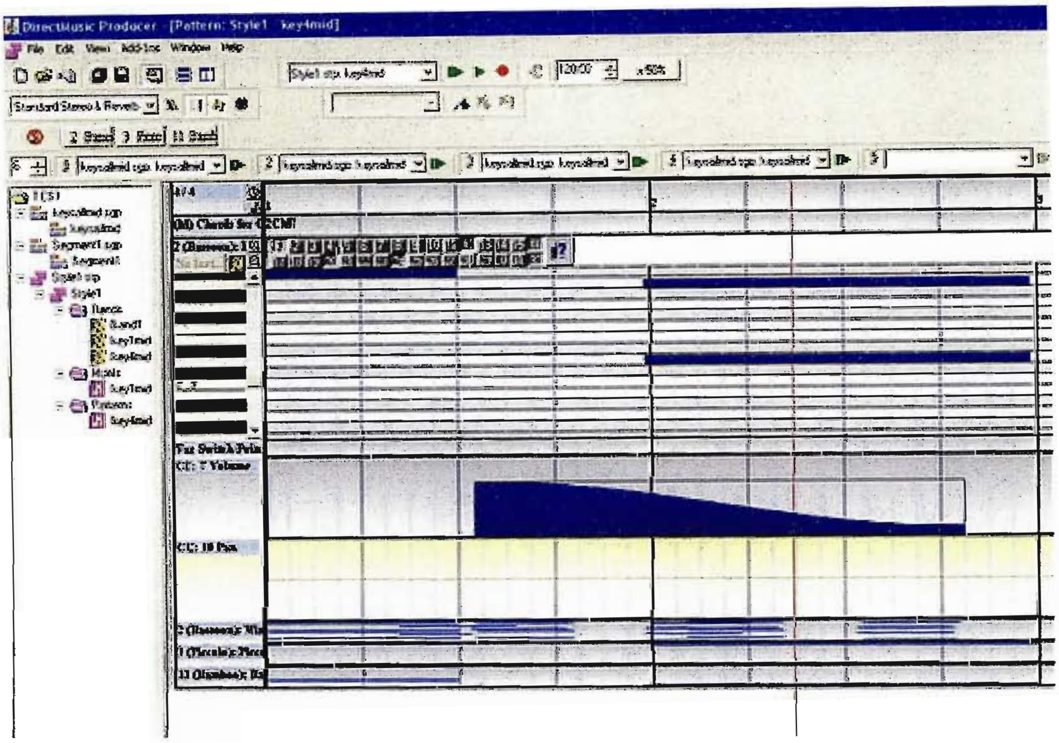
This is very important, for it allows the composer to create multiple variations of the original segment, through the implementation of "Patterns" of the segment. This adds a high level of interactivity to the music, because as described in the chapter on "Musical Construction in Linear and Non-Linear Media", it allows for real-time manipulation of individual musical elements with constant variation each time a repetition of the data occurs. It essentially allows the composer to write a single adaptable musical phrase in MIDI format which can have thousands of permutations if not even more. Above the piano roll information in FIG 5., there are small numbered tabs which allow for "performance data" [Microsoft, 1999] to be entered and shared between the patterns. In a non-linear computer game environment, these patterns can then be programmed to be called up as "randomized modules". [Hays, Tom. 1998]

Fig. 4. Screen-print of DMP Segment Window.



The Band, Style and Pattern editors, are seen here in this tree window.

Fig. 5. Screen-print of Style Editor in DMP.



DirectMusic Producer - [Pattern: Style1 key4mid]
File Edit View Add-Ins Window Help
Standard Stereo & Reverb
4/4
CC: 7 Volume
CC: 10 Pan

In the tutorial file on DMP, Microsoft describes the above mentioned characteristics of the program, in the following scenario: “For example, the appearance of a villain in a CD-ROM game might be accompanied by a particular music theme, and the theme could vary depending on the circumstances, such as the game player's character being in peril. Each time the player passes through a doorway, a special theme could be played on top of the background music already playing. As another example, a multimedia CD-ROM about international travel could have a different style of music accompany each country or city, with smooth transitions from one style to another as the user explores. In addition, each mouse click could have a special sound effect or motif dedicated to it."[Hays, Tom. 1998]

It is clear to see from this, that with the tools offered by DirectMusic Producer, the ideal interactive experience can be realized. There are however, some limitations. Firstly, DirectMusic Producer has only very recently come out of Beta testing and there are still some very clumsy interfaces and the program is not very easy to use. There is still the limitation of real-time effects processing. In other words, the desired effect one wants on specific sounds cannot at this stage be done from inside DMP. This means that the composer has to add the DLS sounds already with added effects, which can often make the files larger than is necessary. If Microsoft could develop a set of plug-ins for DMP that control effects on the DLS side of things, there would probably be no need for any other game sound engine.

VLS AUDIO ENGINE

The purpose for including a detailed description of this audio engine is that it relates directly to the project I worked with for the creation of my "Portfolio of Compositions".

Background and rationale of the VLS (Virtual Learning Space) sound engine

The purpose of the VLS Audio Engine is to allow mood and selection of music and sound effects within VLS game scenes. A three dimensional virtual reality environment requires the facility to play sound which can be panned to allow audio cues and sound effects to match the game player's current orientation/view. These are in line with the requirements for interactive music/sound creation as outlined in the chapter on "Interactive Game Music."

VLS sound script editor

Usage

The sound script editor is a small program that mimics the actions of the virtual sound engine. Here the composer is required to write lines of script that inform the audio engine of what to do and when. The program is useful, in that it provides a tool for auditioning the commands for each scene which can then be copied and pasted into the scene editor and later made into a game archive ready for playback in the actual gamestate.

Operation

The operation of the script editor requires the writing of script commands. All the commands are outlined as follows:

Sound Commands[1]

COMMAND	SYNTAX	RESULT and description
PlayEffect	(value, loop) value - id of sound in effects dll loop - true | false	Plays a sound effect, such as a door slam or footstep etc. Controlled by commands in the gamestate when specific sounds are required. This is achieved by placing a 'hotspot' over the area, e.g. a door, which when clicked on, will tell the engine to play the effect.
PlayEffectXY	(value, x, y, loop) x - screen x co-ordinate y - screen y co-ordinate	Similar to 'PlayEffect', only this command is initiated by cursor position on the screen in the gamestate. Useful when the cursor slides over an object or area that requires an effect to be played without having to click on it. Enhances interactivity when using realistic sound effects.
StopEffect	(value) value - id of sound effect currently playing	Cancels the effect currently playing. Useful command if a loop is specified, such as in 'PlayEffectXY(loop – true), where it is necessary to cancel the sound when moving away from the object or area.
PlaySound	(name, loop) name - name	Command for the background loop. Here the composer is required to enter the filename of the

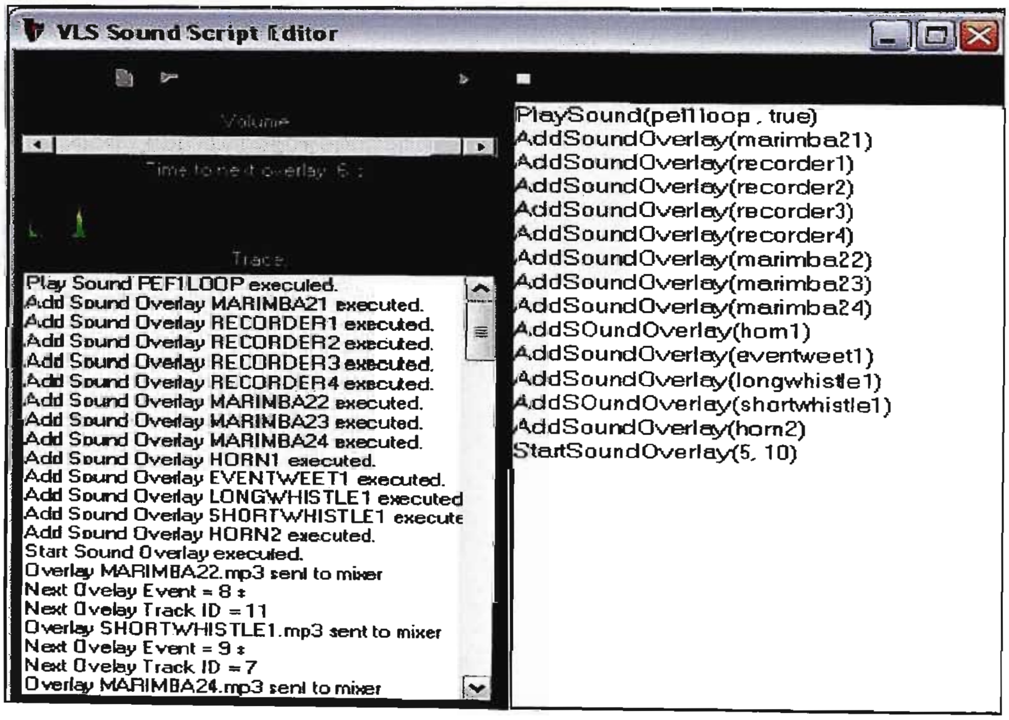
[1] Compiled with the aid of notes from a text document explaining the commands and bundled with the VLS software. Virtual Learning Space Project. Amory, Alan. Nov. 2002

	of mpg file to play loop - true | false	background loop to be played in a particular scene – different files are created for each scene – and then specify it to be looped. In this case, it is necessary to specify a 'true' value for the loop, as the player may spend any amount of time in any particular scene.
PlaySoundCrossFade	(name, loop) name - name of mpg file to play loop - true | false	Important command allowing for the smooth transition of background loops from one area to another. As highlighted in the chapter on "Interactive Game Music", this is an important interactive feature.
AddSoundOverlay	(name) name - name of mpg file to play	This command allows the composer to specify musical elements to be layered on top of the background loop in a random fashion.
RemoveSoundOverlay	(name) name - name of mpg file to play	Command to remove a specific overlay from the audio engine memory.
StartSoundOverlay	(min, max) min - minimum time (in seconds) to wait before playing next overlay max - maximum time (in seconds) to wait before playing next overlay	Here the composer specifies the playback timing of the musical elements described above in 'AddSoundOverlay'. A maximum and minimum time is specified which allows for random playback of musical elements, which enhances the opportunity for the composer to prevent the player from becoming familiar with a recognizable repetition pattern of the musical elements. It is also possible here to have up to 10 or so musical elements specified for each scene which will be played back completely randomly. The composer therefore has to take great care in ensuring the homogeneous quality of each element and the background loop. Very important interactive

		element.
ClearSoundOverlays	()	Stops all the overlays specified above.

The commands listed above are then auditioned in the VLS Script Editor programme. This enables the composer to test the sound files created to determine their cohesive abilities with one another and to ensure that they are relevant to the scene they are intended to be played back in. Once this is done and an acceptable result is found, the script lines are copied and pasted into the VLS Portal Game Engine (an editor programme with the scene variables, graphics and other files as they would appear in the game with the sound script), where they are saved into the game archive. I have included screen-shots of the VLS Script Editor (Fig 6.) and the VLS Portal Game Engine (Fig 7.) to illustrate firstly the auditioning phase, and then the archive creation phase with the scene graphics and other material:[1]

Fig. 6.

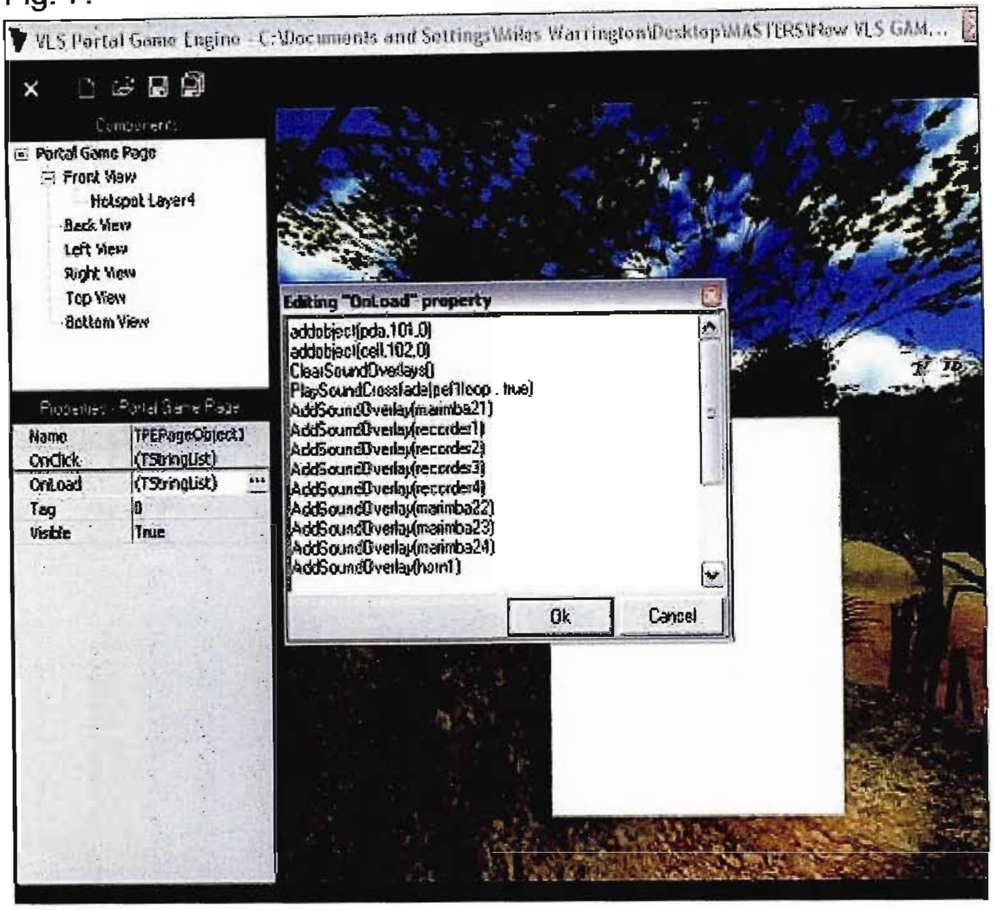


```
Play Sound PEF1LOOP executed.
Add Sound Overlay MARIMBA21 executed.
Add Sound Overlay RECORDER1 executed.
Add Sound Overlay RECORDER2 executed.
Add Sound Overlay RECORDER3 executed.
Add Sound Overlay RECORDER4 executed.
Add Sound Overlay MARIMBA22 executed.
Add Sound Overlay MARIMBA23 executed.
Add Sound Overlay MARIMBA24 executed.
Add Sound Overlay HORN1 executed.
Add Sound Overlay EVENTWEET1 executed.
Add Sound Overlay LONGWHISTLE1 executed
Add Sound Overlay SHORTWHISTLE1 execute
Add Sound Overlay HORN2 executed.
Start Sound Overlay executed.
Overlay MARIMBA22.mp3 sent to mixer
Next Ovelay Event = 8 s
Next Ovelay Track ID = 11
Overlay SHORTWHISTLE1.mp3 sent to mixer
Next Ovelay Event = 9 s
Next Ovelay Track ID = 7
Overlay MARIMBA24.mp3 sent to mixer
```

```
PlaySound(pef1loop , true)
AddSoundOverlay(marimba21)
AddSoundOverlay(recorder1)
AddSoundOverlay(recorder2)
AddSoundOverlay(recorder3)
AddSoundOverlay(recorder4)
AddSoundOverlay(marimba22)
AddSoundOverlay(marimba23)
AddSoundOverlay(marimba24)
AddSOundOverlay(horn1)
AddSoundOverlay(eventweet1)
AddSoundOverlay(longwhistle1)
AddSOundOverlay(shortwhistle1)
AddSoundOverlay(horn2)
StartSoundOverlay(5, 10)
```

[1] 'Virtual Learning Spaces' VLS - Professor Alan Amory (co-ordinator) School of Life and Environmental Sciences, University of Kwa-Zulu Natal, Durban. 2001.

Fig. 7.


VLS Portal Game Engine - C:\Documents and Settings\Miles Warrington\Desktop\MASTERS\New VLS GAM...
Components
Portal Game Page
Front View
Hotspot Layer4
Back View
Left View
Right View
Top View
Bottom View
Properties - Portal Game Page
Editing "OnLoad" property
Ok
Cancel

Name	TPEPageObject1
OnClick	(TStringList)
OnLoad	(TStringList)
Tag	0
Visible	True

```
addobject(pda,101,0)
addobject(cell,102,0)
ClearSoundOverlays()
PlaySoundCrossfade(pef1loop , true)
AddSoundOverlay(marimba21)
AddSoundOverlay(recorder1)
AddSoundOverlay(recorder2)
AddSoundOverlay(recorder3)
AddSoundOverlay(recorder4)
AddSoundOverlay(marimba22)
AddSoundOverlay(marimba23)
AddSoundOverlay(marimba24)
AddSoundOverlay(horn1)
```

VLS sound engine components:

The VLS engine uses the features of a standalone commercial audio engine called FMOD. The implementation of the VLS sound engine includes a subset of features included with the FMOD sound engine. Features not implemented include:

- Mixing of sound effects in software mixer in preference to hardware mixing via Direct Sound – instead of using a built-in mixer that is found on most sound cards, the mixing is done through a virtual mixer built into the engine itself.
- Interpolating mixing facility – allows for different audio files to be inserted with the current file playing.
- Volume ramping and click removal when there are sudden pan or volume changes.
- Software 3D engine with support for logarithmic volume attenuation and audio panning. The 3D engine allows for parameters to be set within a 3D graphic display space. In other words, the engine allows for multiple inputs from a 3D space to be used for the triggering of audio files.
- Support for Microsoft Wavetable Synthesizer.
- WAV Loop point support.

Potential enhancements to the VLS sound engine include exposing the following FMOD facilities:

- Stream synchronisation.
- Flange, chorus, reverb, equalisation and echo.
- Programmable DSP[1] engine allows programmable effects on music and sound effects output.

Therefore, the most significant discoveries that determined how I would implement my music for the VLS game are as follows:

i. The audio engine allows loops to be performed either as a background sound or as an overlay to the background loop.

ii. For the purposes of overlaid sounds, the looping facility is not a necessary feature, as the overlays are designed to imitate random environmental sounds. The overlay facility proved useful in this regard when designing the sound environment in the jungle – i.e. bird sounds were added as overlays.

iii. The puzzle themes needed to be coded separately from the other sounds, as it was only possible to incorporate the sounds in the puzzle dll[1] files themselves. This meant it was impossible to test the puzzle sounds in the gamestate prior to the coding.

iv. Resource management of the sound engine prevented frequent background loop changes i.e. from one portal file to the next. It was necessary then to compose and design only 3 or 4 changes to the background per scene.

v. The sound engine does currently not allow for individual element manipulation by user input. This meant that once the sounds were created and archived into the game, they would be called up when needed, but will not change within themselves. This meant that to achieve "realism" I had to circumvent this problem by adding random pan and volume envelopes to each recorded sound element prior to

[1] See Terminology.

mixing all the audio intended for each scene. This caused the sound elements to behave as if being altered from the gamestate.

The features currently available in this engine therefore do not allow for real-time musical adaptability. In other words, once the elements are composed and scripted into the game archive, they cannot be further manipulated by user interaction. One of the goals of achieving absolute musical interactivity is the combination of basic interactive elements, such as the ability of the software to change the music based on scene shifts, but to also adapt to specific and indeterminate user input. For example, the ideal situation would have been the ability for the engine to allow for the composition of just one musical phrase that could be used as a single macro-element, and then have this element change to user input, by allowing for countless permutations to be played back when called upon - much like the features described in the analysis of DMP above.

OTHER AUDIO ENGINES

There are quite a few game developers that have designed their own unique audio engines to meet specific requirements of that particular design team and project. The lists that follow provide some insight into game developers choosing to do this and the reasoning behind their work. (VLS is also an example of this). Research has revealed many such examples however I have chosen some specific ones which I think are pertinent to the successful realization of 'Interactive Music' in computer games. Apart from this, it was necessary to identify some engines used in the gaming industry to provide me with some insight into what technology is available to composers designing music for interactive applications and to determine what features aid in achieving interactivity.

Former Atari Games technology director, Brad Fuller, former chairman of "The Interactive Audio Special Interest Group, IA-SIG", Mark Miller and "The Interactive audio Journal" editor Alexander Brandon, have outlined some of the most significant interactive audio engines in terms of their interactive capabilities. They can be divided into two categories: hardware and software. The discussion below relates to engines designed post 1994.

Hardware:

One of the first engines was called “C.A.G.E.” – Configurable Audio Generation Engine, after composer John Cage and was one of the first key standalone audio boards developed to work with the main board of the PC. This board was a DSP board (Digital Signal Processor) that consisted of a music and speech sequencer and an operating system of its own. It allowed for simultaneous playback of hundreds of channels. It was this very device that led to the development of current stand alone PC audio cards. Some of these PC audio cards included the “Aureal Vortex 2 (Diamon Terratec)” This card was one of the first stand-alone PC soundcards to allow for the use of DLS sounds as described in the chapter “DirectMusic and DirectMusic Producer. This was an important development for it allowed composers to use their own sounds instead of relying on General MIDI sound banks.

Another card was the “Creative Labs SB Live!”. This PC soundcard was developed by Creative. They developed their own method of allowing composers to choose their own sounds. They called it the “Soundfont” format. This is similar to DLS. Here banks of sounds could be created using any sound the composer wishes to use. The banks are then loaded into a memory bank on the audio card itself and the sounds are called up through MIDI triggering. A composer could then package a sound set with a computer game. User input could then be tagged with code specifically designed to generate MIDI information which could be sent to the card, changing bank information, channel, patch etc.

The “Yamaha DS2 YM744” also made use of DLS. It shipped with PC motherboards as a motherboard-based accelerator. This meant that the PC consumer didn’t have to purchase a sound card as an add-on to the computer equipment. Further to the use of sound cards in desktop and laptop PC’s , standalone video game consoles were also available since the early 1990’s. These units were dedicated only to computer games. In terms of using DLS, the most significant so far is the “Xbox”. This fairly new video game console was released by Microsoft in late 2001 and developed with industry leaders such as INTEL and NVIDIA. The system includes the most powerful dedicated audio processor available to a games console on the market together with its own hard disk to enable audio data caching. It makes extensive use of DirectMusic and DirectSound software.

Software: (list compiled with material from “Project Bar-B-Q Group Report” - see footnote[1])

Beatnik(www.beatnik.com)

An interactive audio system allowing pre-designed responses for sound and music cues to play in real-time response to end-user input. It gives interactive audio creators an opportunity to compile and compress different media types and design them to react in a predictable, cohesive way in response to unpredictable user and/or system input. Initially designed for Web audio, Beatnik technology is now used primarily for audio in wireless devices such as mobile phones. According to the company, "Beatnik offers core audio engine technology, application programming interfaces (APIs) and tools that allow the development of audio-enabled applications, services and content delivery for the mobile world."

Mixman(www.mixman.com)

This consumer software automatically aligns triggered sounds to the beat, so no matter how rhythmically challenged you are, you can create music that sounds good. It can be played from a computer keyboard or a proprietary USB controller that resembles DJ turntables.

AbletonLive(www.ableton.com)

This Mac/Win program is a performance-oriented looping and arranging tool, in contrast to "offline" editing programs such as Sony Acid and BitHeadz Phrazer. Live automatically scales sounds to fit a song's tempo and key in real time.

RuntimeMixingSystem

A system that can take N channels of audio input and mix them into N channels of output based on the states of each input channel at time T.

RemixingSystem

As interactive data, songs include musical information plus data parameters used to modify the song on the fly. The user actions act as input stimuli to modify the pre-defined audio stream.

[1] “What is interactive audio?”, Eighth Annual Interactive Music Conference, Project Bar-B-Q, Group report: What is Interactive Audio? And what should it be? Fat Labs 2003. http://www.projectbarbq.com/bbq03/bbq03r1.htm

RolandV-Synth(www.v-synth.com)

This sampling keyboard is an interactive composition and performance tool that features real-time pitch-shifting and time-scaling based on the player's actions.

KorgKarma(www.korg.com,www.karmalab.com)

This synthesizer keyboard generates idiomatically correct musical parts based on player's actions.

HyperInstruments(www.media.mit.edu/hyperins)

Musical systems developed at the MIT Media Lab that augment the expression capabilities of musicians. By using sensors to measure extra human parameters (input stimuli), the sound (the data for the system) is changed based on these parameters.

GameAudioSystems

The capability to establish a set of rules and procedures for how audio assets are played back and controlled during playback in response to unpredictably timed player input and system-generated events, all in accordance with the audio creator's creative design.

MadPlayer(www.MadWaves.com)

The MadPlayer is a handheld composition and performance device. A MadPlayer song includes musical information and data parameters used to (re-)compose a song and modify it on the fly. MadPlayer engine algorithms take user actions and pseudo-random number series as input stimuli to modify the pre-defined audio stream; thus, the music produced has been modified in response to input stimuli.

MilesSoundSystem(www.radgametools.com/miles.htm)

This software acts as a virtual sound library with many different capabilities in terms of sound triggering and output. It can work in tandem with Directsound or as a standalone audio engine. It has been widely used as an audio engine in many games, including titles from the *Myst* series. Developed by John Miles in 1991 as a driver for soundcards and MSDOS.

The availability of all these and many other interactive audio engines seems to come hand in hand with the demand from the audio community as a whole as well as the consumer world-wide. Directly related to this is the demand for high-end, good quality products from consumers encouraging developers in the gaming and PC media industry as a whole to produce products that need to meet the requirements of the aforementioned consumer. It has taken a while for interactive software and hardware to develop since the first audio chips used in electronic equipment such as the TI 5220 from Atari, that was made by its use in the famous “Speak & Spell” products in the early 1980’s, to today’s sophisticated virtual products such as DirectMusic and Dolby Digital 5.1 surround sound systems. But the result is we now have the means to create truly adaptive and interactive audio for a wide range of electronic applications and media, the applications of which I will list in the conclusion to this article.

GAMING APPLICATIONS

Specific commercial titles and other applications relevant to interactive music composition within the paradigm of computer games, will be looked at in this section. I have chosen some titles from the *Myst*[1] series, *Tron 2.0*[1] and *Grand Theft Auto III*[1].

The reason for choosing these titles stems from research into titles with different entertainment purposes. The *Myst* titles fall under what is termed "1st /2nd person, puzzle fantasy games", whereas *Tron 2.0* falls into the category of "action adventure". *Grand Theft Auto III* is different yet again, in that this is purely an "action game" involving the interaction of an avatar in the 1st person with the gamestate. I have therefore chosen these games because they are quite different in many respects one to another.

In relation to the definition of interactivity and the description of the properties of non-linear media, the theory here, is that all titles wishing to create an immersive environment for their intended consumer, should by default, make use of interactive music in some form or another. Computer games are by their very nature 'interactive media' and should therefore have 'interactive music'. I say should, as this is important to engage the user in such a way that he/she has to pay close attention to everything going on in the gamestate, *including* the music, of course, created in such a way as to prevent the person turning the sound off. This also ties in with the premise that most commercial game titles today are produced and conceived in a similar way to commercial film titles. In fact, in the last 5 years, the two media have become quite competitive in the entertainment sector and are now seen as 'rivals.'[MacGann, Neal. 2003. p.5] Analysis at this point will help to define the explanation of "Interactive Music" at the beginning of this paper and in so doing, highlight techniques and tools used to compose interactive computer game music.

The analysis was approached by firstly asking several questions about the game music. Answering such questions required the use of several tools to achieve analysis, as well as problem solving to overcome some technical issues with analyzing stereo audio data. These included:

[1] See Bibliography – Software – for publishing information.

- Critical examination of what the music is doing at any given point.
- Why is the music doing what it is doing at that point?
- How did the composer achieve this? This included an analysis of the tools the composer used to implement the music, which involved research into the architecture of the game software viz. audio engines, sequencers, studio equipment etc.
- Recordings of the game audio as the games were played – here two computers were used – one on which the game was played and another used to record the audio simultaneously.
- The recorded audio was then transcribed with the aid of a piece of software called "Amazing MIDI"[1] which roughly converts audio data into MIDI data. (It proved essential here to compare the original audio with the MIDI data at all times, to determine what was going on in terms of independent musical lines in order to isolate them, as the software converted the data into a monophonic line of MIDI data.)
- The analysis presented several problems in terms of timing with the sequencer track and required real-time recordings to be played in via a MIDI Control keyboard to input the correct melody or musical line.
- Importation of the MIDI data into Sibelius Music Software[2] to be scored and notated correctly.
- Analysis of the transcriptions to determine what the composer did when and why.

[1] Arakisoftware, 2003 http://www.pluto.dti.ne.jp/~araki/amazingmidi/

[2] Sibelius Software Limited, 2001 http://www.sibelius.com

VARIOUS TITLES IN THE MYST SERIES: "RIVEN: THE SEQUEL TO MYST", "MYST III: EXILE" AND "URU"

RIVEN: The Sequel to MYST[1] *(with music by Robyn Miller) & Myst III: Exile (with music by Jack Wall.)*

Analysis of the game identified several key elements of the sound design in Riven. They are as follows: 1. Specific area music, 2. Clue music and 3. Puzzle completion music.

Specific Area Music

Each area in the game has its own unique music with differing themes which can be described individually. For example, the first area the player visits is a large rotating room - what I have called "Beetle Room". Upon entering, you hear music with a tenor saxophone and log drum melody, playing above suspended strings and synthesiser chords made into pads. The theme melody consists of 4 pitches, namely B flat, A, G and F sharp. The melody is very short and interspersed with long sections of relatively indefinable sounds made up of chords supporting the initial melody. This helps to maintain subtlety and prevent musical saturation for the player i.e. musical material is specifically designed to avoid the player becoming familiar with the music. The melody is also inverted the second time one hears it, creating interest without blatant repetition. It appears that at this stage the music is not meant to provide the player with any clues, but certainly adds atmosphere. Analysis also revealed that very careful attention was paid to the fading in and out of the music (upon entering or exiting the area as the case may be), with the environmental sounds playing in the background. In conjunction with this, the sound for this area takes the form of a 2 minute loop. Being fairly long, the player would have to spend quite a while in the area before saturation level is reached. This is an important tool in the successful composition of music for computer games, as highlighted earlier in the article.

Further playing brings one to four other immediately available areas without having to complete puzzles. Here, the composer has developed the ideas one first hears in the "Beetle Room". The sustained chords from the pads used to support any thematic material,

[1] Cyan, Inc. and Broderbund Software Inc. 1997

remains identical in pitch and harmonic variation. Again, this is vital to creating continuity and thus avoiding obvious changes.

Clue Music

Near the end of *Riven: The Sequel to Myst*, the player has to memorize a four note bell melody played by a bedside clock in the villain's bedroom. This is essential as it is a code for an elevator door which has to be opened later on in order to rescue someone. Without doing this, it is impossible to complete the game successfully. This is an instance of direct musical interaction, where the player has to make a conscious effort to listen to what is going on. Interestingly, if the player exits the game at this point and comes back later, the previous melody doesn't work, and he/she has to play the bell melody again, and this time it *is different*. This is yet another method of creating interactive musical elements. Apart from that, this game in the series has less interactive elements than the other titles in the series.

The music does however on the whole, form a wonderful soundscape[1] to the cinematic feel of the gamestate.

Puzzle Completion Music

What has been referred to as 'reward music'[Wall, Jack. 2002.] in later chapters of the *Myst* series, such as *Myst III: Myst in Exile*, this type of music is non-existent in *Riven*. The music described here is essentially clips that play when the player successfully completes a task. In *Riven* however, all the player hears when puzzles are completed, are sound effects relating to the puzzles at hand, given that the puzzle-solving process takes on the form of virtual physical motions or tasks.

Research into the composition of music in general for all the titles in the *Myst* series, has revealed a trend to develop and expand on the 'cinematic' concept as seen in *Riven*. For example, in comparison to the music for *Riven,* the music for *Myst III - Myst in Exile,* as explained by Jack Wall, composer of the music for the game, was approached in a different way to the music of *Riven*. He wanted the freedom of melody writing and the possibility of

[1] See terminology.

using as many instruments as he liked, including a full orchestra. In the previous titles, Miller only used one synthesiser for both the first two titles in the series. In his approach to composing the music for *Myst III: Exile*, Jack Wall stated: "I wanted the music to have as much 'purpose' as possible - not just be tied to areas within the game, but also to have a level of randomness and interactivity to it."[Wall, Jack. 2002] What he means by this, is composing and implementing music in the game that reacts to user input through stimulus from the game.

This happened on several different levels in this game: Firstly every time a player completes a puzzle, the theme from that stage or "age" is played to help the player realize the importance of the completion of such a task. Secondly, if at any stage the player moves into areas requiring live cinematics or action, music making reference to the cinematics i.e. a puzzle object moving would have music associated with it, to enhance the experience and lastly, the general background music or what is termed "In-Game music" by Wall in this example, changes within each scene itself to retain the user's attention through the audio. This, as described earlier, is paramount to interactive computer game music. Jack Wall then goes on to list the ideas he had to enhance the game and the musical interactivity: [Wall, Jack. 2002]

- Reward Music - For achieving certain important puzzles within the game. These would be the Age Themes for the three "Lesson" Ages in the game used as thematic material for that age also.
- Orchestra and choir - For all live action and cinematics.
- In-Game Music - Similar style to the music in *Myst* and *Riven.* Tied specifically to Ages and areas, but also played in a random, ever-changing, interactive fashion.

Indeed, all these elements (with the exception of puzzle completion music) highlighted by Wall are evident in *Riven.* To show some of the techniques used by the composer, I have included transcriptions of some of the music to *Riven.* , These are hoped to highlight some of the techniques employed in writing music for such applications. (See Figs. 9 thru 13 in the transcriptions section for scores and analysis of some of the main areas of the game.)

TIM LARKIN and THE MUSIC FOR 'URU' – the latest chapter in the MYST SERIES

Uru - the latest title by UbiSoft, is a continuation of the *Myst* adventure. To begin with here, I have included an interview with Mr. Tim Larkin in the United States, composer and sound designer for *Uru*, to shed some light on his techniques for composing the music for the game. I asked him various questions which he generously answered, (MW – Miles Warrington, TL – Tim Larkin):

MW: In composing the music for 'Uru', how did you approach the overall sound design[1] – was it a matter of approaching it as one would in a film or was it a different approach?

TL: You have to start out with the same basic concept in mind, which I think is the same no matter what medium, and that is to create good sound and music. From that point on, there are several departures. You need to be more transparent at times, and interactive at times. Themes need to react in a way that allows them to be repetitive without being redundant. When you do decide to draw attention to the music or sound, it's usually special case moments like underscoring a section of linear game play, or speech in the case of URU.

MW: What 'interactive' elements did you use – i.e. did you focus on 'cue themes' to point the player in a certain direction and give clues or did the music simply change depending on an obvious mood created by the images?

TL: The music in URU underscores more of the environments and exploration. There are several re-curing themes however, such as the Yeesha theme and the Cleft that were intended to tie various scenarios together. As for giving the player direction, this game doesn't require musical direction, as it's more about taking your own time to explore and uncover story.

[1] See Terminology

MW: In composing music for a non-linear environment, the 'background loop' or sound needs to be simple to avoid too sudden changes between scenes. Did you solve this problem by keeping them simple or by using a different technique?

TL: The loops are designed to seamlessly loop into themselves, and in spots to resolve into another piece with minimum impact. This is usually attained by using a similar musical palette or style. Also the very entrance of the music can be subtle enough as to not draw attention to itself until it's well into the piece.

MW: Did you analyze any music from other computer games, and if so what techniques/approaches did you use to gain any information from them and what games?

TL: I tend to listen to as many other games as I can, and see what's successful and what isn't. I can't think of a particular example that led me to write for URU in the manner that I did.

MW: Did you create elements that remain the same but are played back randomly or did you write elements that themselves changed constantly? (This will depend on the sound engine design and capabilities.)

TL: I did write quite a few pieces that take place randomly. They would usually be in a particular style for a particular place. I would write maybe 5 cues to play at random time intervals that I could input the longest possible length and shortest length between each piece.

MW: If you had to analyze the music for 'Uru', what would be the key elements you would extract from your analysis, apart from focusing on any obvious mood elements?

TL: I think that in this genre of game, the mood elements are the most important. I'm trying to set up a sense of mystery and the unknown, as well as underscoring environments for the most part, so the key component is to create the sense of what isn't seen, or what underlying story might be hidden. I want to make you feel that you are in an ancient place at times that inhabited a lost culture. Those types of suggestions are done with the musical palette and style.

MW: Did you do any live recordings with musicians or was most of the music sequenced and recorded in a studio using samplers etc.?

TL: I used several live players that dubbed over samples. The live players were guitar, Duduk, percussion, trumpet and voice. The rest is sampled.

MW: Did the sound engine design allow for you to be as free as possible in the creation of the music or was it limiting in some ways and if so, which ways?

TL: It was limited in the sense that I was unable to clock anything or recognize markers within sound files. That is being remedied for the next iteration of the engine.

MW: Were you collaborated in the creation of the music engine i.e. were any suggestions you might have had for the design of the engine taken into account by the project team?

TL: I was to a point, although all of my suggestions were unable to be implemented.

MW: Did you use DirectMusic Producer or MAX, to generate the desired playback together with an independent engine in the game or separately or none at all?

TL: None at all, yet.

MW: As a composer for interactive environments, have you come across or do you know of any relevant published literature on the subject of composition for such media? I ask this as I have searched high and low, only to find 99% of the info from the net!

TL: I know that there are several books that tend to focus more on the business of game audio, but none that actually deal with only the composition techniques.

MW: What specific tools did you use to create the music for 'Uru' and other titles in your portfolio? I.e. equipment, software etc. Did they differ depending on the requirements from the game developers?

TL: I use the same tools for the sound production no matter what style or medium. I use ProTools for sound design and mastering, and Digital Performer for music composition. I also have quite an extensive sample library that keeps the sounds fresh and diverse for various projects.

MW: Did you explore extreme compositional parameters i.e. with regards to modern compositional techniques with tempo, pitch, writing across the bar, dynamics etc., or did you employ traditional compositional parameters/techniques?
As a modern/contemporary composer in the 'classical' sense, I would personally be disappointed with any 'commercial' boundaries placed on the project. This leads me on to ask: Were you limited in your expression by the producers or were you free to explore compositional parameters at your will?

TL: I was given total freedom with URU. If anything, I was encouraged to compose outside the box. That was something that I was extremely thankful for, and a situation that I tend to work better under.

MW: Finally, as a successful composer for interactive titles, what advice would you give to composers attempting to write successfully for interactive media?

TL: Start out writing good music. That is the key to being successful. Many up and coming composers don't know what is good and what isn't, and as a result, they put stuff out that doesn't meet a high standard. Not everything I write is good, but I know that, and realize that if it isn't then I either fix it, or scrap it. I can't believe some of the music that I hear in games that gets by the producers. It amazes me how uneducated the production teams can be at times. I think that this might be due in part to the infancy of the game industry itself. In time, I do believe that the standard will be set as high as the film industry due to maturity and experience. There are quite a few great soundtracks being produced now, and I think the bar is constantly being set higher.

I wish to thank Mr. Larkin for his contribution in this matter.

The points above reiterate the findings of my analysis in terms of environmental sound creation and the music underscoring the theme or story. As in *Riven,* Larkin points out that there isn't a necessity for musical direction in the game, but rather the employment of simplicity in musical material and writing it in a way as to enhance the exploration/adventure qualities of these games. It is perhaps easier for a composer in this instance to create music based on a purely abstract environment - not making any reference to the world we live in. Yet anything that makes reference to existing human environments in a game, should be made reference to also in the music. Again this last point emphasises the 'immersion' and 'cinematic' concepts described earlier. (Figs. 14 and 15 in the transcriptions section show some details of the *URU* score.)

MUSIC FOR TRON 2.0 [1] AND GRAND THEFT AUTO III [2]

Tron 2.0:

Tron 2.0 is the first of the "action/adventure" games I chose to analyse. This game is a sequel to the original movie "Tron" in which you play the role of a computer script in a virus over-run computer system.

Analysis – examination of orchestration, harmony and rhythm of the music revealed several key components. Firstly, a limited number of musical elements were composed based on a set of musical parameters. These parameters consist of a minor chord in the key of "a" minor with an added 6th in the chord (F-sharp), arpeggios made up of the notes, chords made up of the same notes and ostinato rhythms with variations. Other parameters include the use of different instrumentation upon repetition of an element. In terms of "adaptive" or interactive audio techniques, the use of a single musical idea with the possibility of many permutations is in line with such techniques. This is highlighted by the fact that the elements in this game are analogous with those described by Tom Hays in the chapter on "DirectMusic Producer" as "randomized modules". [Hays, Tom. 1998] Further to this, the music in the game appears to adapt to user input as the gameplay is completely determined by user response and changes very specifically to what is happening in the gamestate. For example, there is no time limit to any action that needs to be performed and one encounters completely random isolated enemies and tasks to be performed at any

[1] Buena Vista Interactive. http://bvinteractive.go.com/

[2] Take-Two Interactive Software for Sony PlayStation PS 2. http://www.rockstargames.com

given sitting. The musical elements are thus called upon at any given time and are chosen by the engine depending on their action or task specific properties.

Added to this, is the fact that there are no long looping musical elements in this game. Instead, the designers created the aforementioned elements to be used instead of looping melodies. It is also for this reason that the musical elements appear to adapt to user interaction because they occur only during specific actions. Supporting these elements however, are background loops of chords played subtly by string and synthesiser pads, providing harmonic support and texture to the user determined overlays.

Interestingly, the elements themselves are in some instances reworkings of Wendy Carlos' original music for the film. (See Fig. 16 in transcriptions section for further details.) What was described as a strange but very effective score to the movie, the music was played using mostly electronic instruments, such as the Moog[1] and GDS[2] synthesisers. The music in the game too makes extensive use of synthesised electronic music.

Grand Theft Auto III:

This game is different from the other games I chose to analyse in that the music is played back as a "virtual radio station". The player is cast as a freelance thief whose task is to steal cars and work for the city mafia. In doing so, the player spends most of the gameplay in vehicles of differing kinds. Each one however is equipped with a radio. The player is given the option of choosing between 10 different stations. They range from talk radio programmes, jazz, reggae, rock, pop, dance and even an opera channel. Each channel has its own DJ, who speaks from time to time announcing adverts, play-listing schedules and so on.

Each channel has approximately 25 minutes of recorded material in MP3 format, that is looped once it is completed. This is long enough as there are a total of 10 channels, making the total length of available audio approximately 250 minutes. The result is that the player doesn't notice the repetition in any one sitting, which on average would last perhaps 1 hour.

[1] See http://www.synthony.com/vintage/memorymoog.html for details.

[2] CRUMAR General Development System (GDS) and Synergy Synth. http://www.synthony.com/vintage/dkigds.html

The music does not make any environmental references in this case, but is employed as a method of entertainment – just as one would choose to do when listening to the radio when driving in reality. The music is not original in the sense that it was composed for the game. Instead, the game developers have licensed existing audio from record labels, featuring the artists heard in the game. For example, in the "opera" channel, which consists mostly of very well known and popular tenor and soprano arias from the operas of Verdi and Puccini such as 'Rigoletto', 'La Boheme', 'La Traviata', 'Madam Butterfly', etc. Unfortunately, no indication is given anywhere of the artists involved in these extracts. With regards to the popular music radio stations, easily identifiable where tracks by 'Toto', 'Lora Branigan', 'Michael Jackson', 'Duran Duran' and the like, together with a lot of unfamiliar tracks with relatively unknown artists. What all this music does do though, is make potent the era the game is set in – the early 1990's, by making available to the player sounds one would hear if in fact you were living in that time.

With regards to the dialogue, the DJ's make reference to the city, by advertising products and events one sees posted on billboards and signs. Another instance of the dialogue making reference to the action on the screen, is with regards to the weather. Every so often, weather reports are given and these make direct reference to the weather one sees on the screen. Dialogue is also used as a method of transferring information to the player when tasks are assigned by the mafia bosses. In this way, the dialogue in this game is more referential to the gameplay environment than the music.

The main focus of this title is the action. Unlike titles from the *Myst* series for example, intimate exploration and puzzle solving is not the object of the game, and therefore the game is not conceived as a cinematic work involving the player as a character in a movie with a specific plot that could have various outcomes. Instead, the player's actions are determined by the action and tasks given, which follow a strict event schedule and timeline.

This characteristic has circumvented the need for 'cinematic' music and the developers therefore chose not to create original composed music, but rather existing music with the option of choice (through the radio stations) for entertainment purposes. The key concept here is thus entertainment and action and not 'emotive' or 'immersion' through music of the player in the gamestate. The music only makes a loose reference to the era the game is set in. The action in the game is therefore far more important than the music in this title –

drawing the attention of the player away from the audio. There are however very realistic sound effects that mimic the action on the screen and thus contribute in some way to the 'immersion' concept.

So, we have gone from computer generated tones in real-time in the late 1980's and early 1990's to virtual radio stations playing existing music to fully conceived and produced scores, that actually could be symphonic, just as movie scores are produced. What follows again, is a trend in the gaming industry to produce products in line with the film industry in terms of cinematics, experience and immersion. Thus the music for computer games is evolving and has come a long way since its inception.

THE FUTURE

Determining the future of anything to do with technology is extremely difficult. However, based on current trends, one should see computer game music becoming even more effective in its attempt to create immersive and fully interactive environments. Some interesting applications of the concept of 'interactive music' have been entered into in this paper, such as the project on spatial design called "Memory Games" initiated and developed by universities of Auckland and RMIT University, Melbourne, Australia.

During my research for this paper, I have discovered other interesting possible applications of interactive music. For example, the design of interactive platforms utilizing the whole human body - in effect creating a virtual reality scenario for playing games. These include for example, a game called *Collide* - developed as an experiment in full body virtual interaction by Jonah Warren for his masters thesis on "Unencumbered Full Body Interaction in Video Games". [Warren, Jonah. 2003]

This opens up the possibility for the creation of environments where the music for such games could be stimulated by body movement through the technological interface. This is similar to a finger or mouse button click, but using unconventional input devices. It creates interesting possibilities for musical creation and composition by tapping into the paradigms of 'improvised' music triggered by random body movements. This is analogous with the well rehearsed experiments of Machover's "Data Glove" during the last twenty years, but the *Collide* project here adds a new dimension to body input by allowing for infinitely more input streams as the whole body is being utilized in a virtual 3D space. It is thus moving away from the conventional input of key strokes and mouse clicks by virtue of the added dimensionality of the input sensory setup. This in turn creates interesting compositional problems by the very nature of its randomness - how do you create musical sense from this or is musical sense really necessary and what musical elements do you compose that are played back in a random and spontaneous way to achieve reference to the environment the person is moving in?

This follows a trend of moving away from static or fixed musical elements in these media. With the advent of such technologies, it is impossible to achieve musical meaning by composing musical elements that cannot adapt and change spontaneously and randomly, as is the case with truly immersive computer games. As described earlier, Microsoft's 'DirectMusic Producer' is moving in this very direction.

The advent of the mergence of the computer game medium with film and vice versa, is opening up many possibilities for viewer participation in conventionally linear mediums by bringing non-linear concepts to the fore. It is thus possible to imagine a scenario where one day, technology could be available where the user can play a video in a home-theatre setup and through virtual linking devices actually get involved in the film as a character and change the outcome. In fact, this very scenario has been thought of and discussed in future possible scenarios by Kristoffer Gansing. For example, a new installation is described from the "*Future Cinema* exhibition at the ZKM, Karlsruhe, where the cinematic imagery after film is explored, mainly through different installations depending on direct physical input from the audience." [Gansing, Kristoffer. 2003] Very much like the advanced computer games currently available, this presents the composer with a whole new array of interesting and challenging problems. As in the previously imagined scenario, how do you create meaningful music, making reference to the plot and action, when there is always going to be completely spontaneous and random input affecting the linearity of the work?

Certainly some of the methods used to solve these problems have been discovered in the analysis of computer-game-music in this article and include advanced audio engines, capable of adapting to user input as well as the discovery of new and necessary compositional techniques to achieve interactivity. Emphasising the need for 'adaptive audio engine' advancement is one of the goals of the "Project Bar B-Q", as described in the chapter on "Interactive Audio Engines." The project is constantly pushing the technological envelope in terms of interactive audio engine development, with the goal to provide game developers and composers the tools to achieve complete immersion in the gameplay experience for the user. So ultimately, what the future holds in terms of computer game development and particularly for computer game music, as is the purpose outlined here, is difficult if not impossible to say.

The last year has seen the advent of the first ever symphonic computer game music concert, held in Germany in August 2003, at the Gewandhaus in Leipzig.[1] The concert was

[1] See http://www.gamasutra.com/features/20031124/boecker_01.shtml, for details on the concert.

a break through step in establishing the seriousness of computer game music from specific titles and composers world-wide. The establishment of a 'Game Symphony Orchestra' is one such future goal of the serious games community and organisers, in the hope that the celebration of the unique art of computer game music is maintained and developed. It is hoped that with such an orchestra, composers can have access to the resources of a symphony orchestra and develop and maintain a platform for recording and live performance of game music.

Interactive music is also reaching the 'final frontier'. For example, The Space Frontier Foundation[1], has initiated a new project on the exploration of Mars that has many ancillary projects attached to it. One such project includes possible future experimentation in terms of 'interactive' music making in space. Composer Morton Subotnick explains:

"Playing Music on Mars could be done with virtual instruments created on the computer. Taking advantage of what a computer can do for us, we can have the computer play the notes of the work while we interpret and conduct. We could also link several computers and perform as musical ensembles of conductor/performers. In addition, we can create music which is special to the Martian environment. One such musical composition might be a piece of music created on the computer and performed by body movements rather than a keyboard or computer mouse. We would create the 'interactive' music and image composition by building sensors that detect motion. Then, since the Martian gravity is less than Earth's gravity, we could control the computer-generated music and images by jumping and twirling like a dancer through the sensor space." [Subotnick, Morton. 2002] This is certainly a really interesting idea even though it is not directly referring to computer games, it is however analogous with the concepts and technologies developed for the *Collide* project entered into previously and could even one day be used for "interactive entertainment" purposes, such as computer games, for people on different planets.

To end here, I have extracted a quote from George Sanger, a game music developer and composer from Fat Labs Studios, which highlights the thinking in terms of game music development and techniques:

[1] A public organization, comprising of scientists, business people, academics and ordinary civilian space enthusiasts, 'dedicated to opening the space frontier to human settlement as rapidly as possible.' See http://www.space-frontier.org/History/ for details.

"No one person can see the future, or even the present, of this very complex field. It requires the help of the eyes, brains, and hearts of the entire community, just to get one's head around all of the marketing, artistic, hardware, software, interactivity, personality, and Internet issues. That being said, here is the certain, inevitable future of Game Music.

001--The user buys the game and installs it.

002--If he is happy with the music that shipped with the game, he is done. If not,

003--He clicks on "change music."

004--He and/or his program now have the opportunity to select a trustworthy GamePlay DJ from the Internet. They choose somebody.

005--If that DJ has posted a mapping file for the game in question, GOTO 07, else,

006--Another DJ is selected until a mapping file is found

007--Any Amount of Music is now mapped from the vast number of musical works on the Web to the various gamestates of the game by the expert (GamePlay DJ) of his (the User's or the Developer's) choice.

008--The music is either streamed in real-time, or downloaded and changed out invisibly in background.

009--The musician gets paid automatically, proportionally to the number of people he has entertained, the number of minutes he entertained them, and the amount of income generated by the game.

It's that simple, and I'll bet on it. The most fitting model for Game Music is not movies. It is radio." [Sanger, George. 1999]

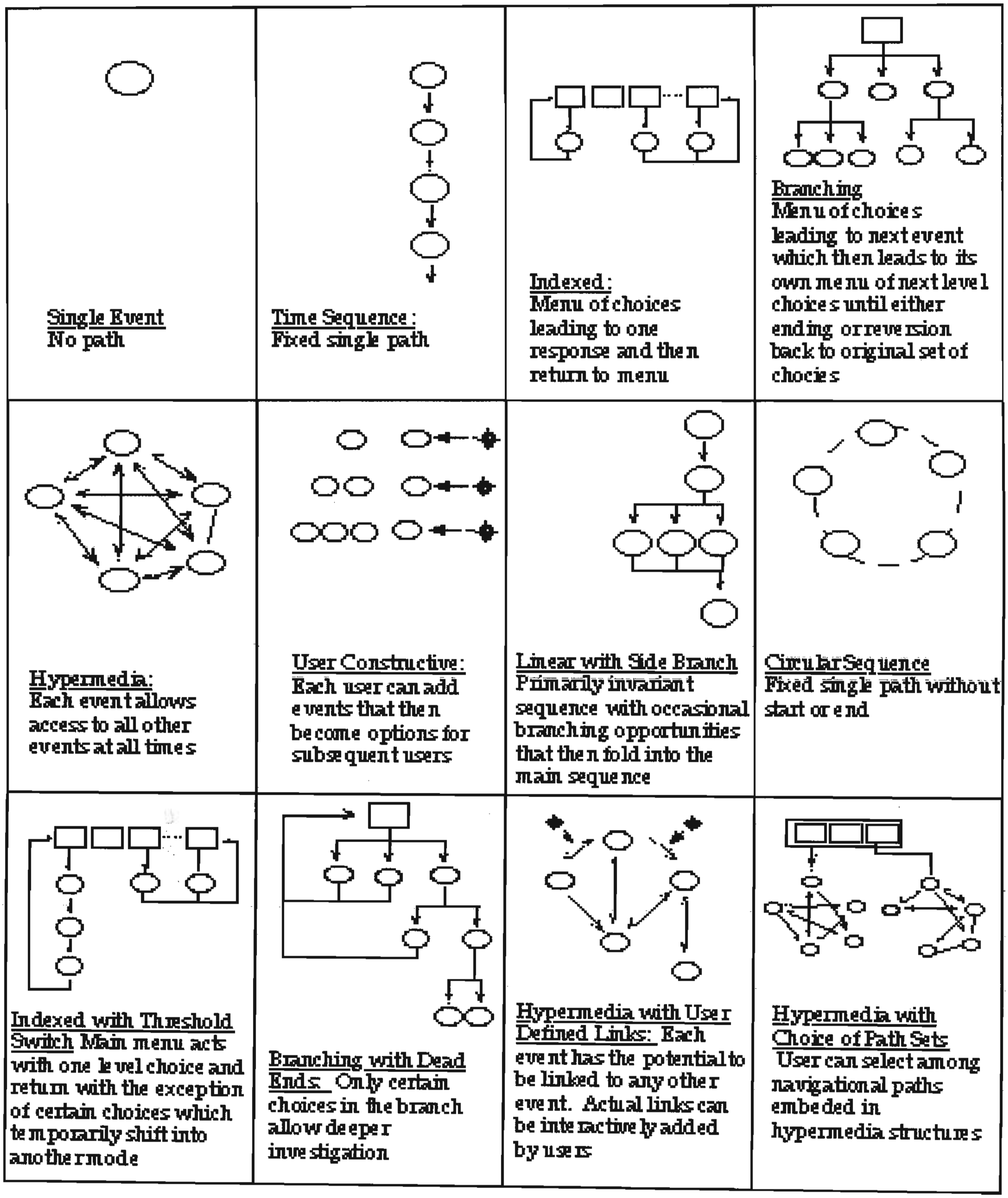
CONCLUSION

In this article, the basic concepts of interactive music, its purpose in computer games and the development of audio in computer games in general has been discussed. Compositional techniques used by game music composers have been discovered as well as a broad insight into the technology required to make interactive music possible, such as audio engines.

Examples of how the music for currently available commercial game titles was created and implemented have been given and comparisons between good and bad examples of games exhibiting different levels of immersion and interactivity have been made. The evolution of interactive game music has been explained and you have also seen how this has affected the overall level of product impact and content wealth.

My aim in presenting this paper was not merely to justify the existence of interactive music in computer games, nor to exhaustively chronicle its impact on computer games as whole, but rather to highlight the problems, techniques, tools and thinking in general, that is required of the composer when writing for this medium. Through my analysis here and discoveries, I hope to create interest amongst other composers in the genre and stimulate further interest in enhancing and developing compositional techniques and in so doing, better the music for the computer game today and in the future.

Fig. 8. www.mercury,sfu.edu - "The Aesthetics and Practice of Designing Interactive Computer Events."



Single Event
No path

Time Sequence :
Fixed single path

Indexed:
Menu of choices leading to one response and then return to menu

Branching
Menu of choices leading to next event which then leads to its own menu of next level choices until either ending or reversion back to original set of chocies

Hypermedia:
Each event allows access to all other events at all times

User Constructive:
Each user can add events that then become options for subsequent users

Linear with Side Branch
Primarily invariant sequence with occasional branching opportunities that then fold into the main sequence

Circular Sequence
Fixed single path without start or end

Indexed with Threshold Switch Main menu acts with one level choice and return with the exception of certain choices which temporarily shift into another mode

Branching with Dead Ends: Only certain choices in the branch allow deeper investigation

Hypermedia with User Defined Links: Each event has the potential to be linked to any other event. Actual links can be interactively added by users

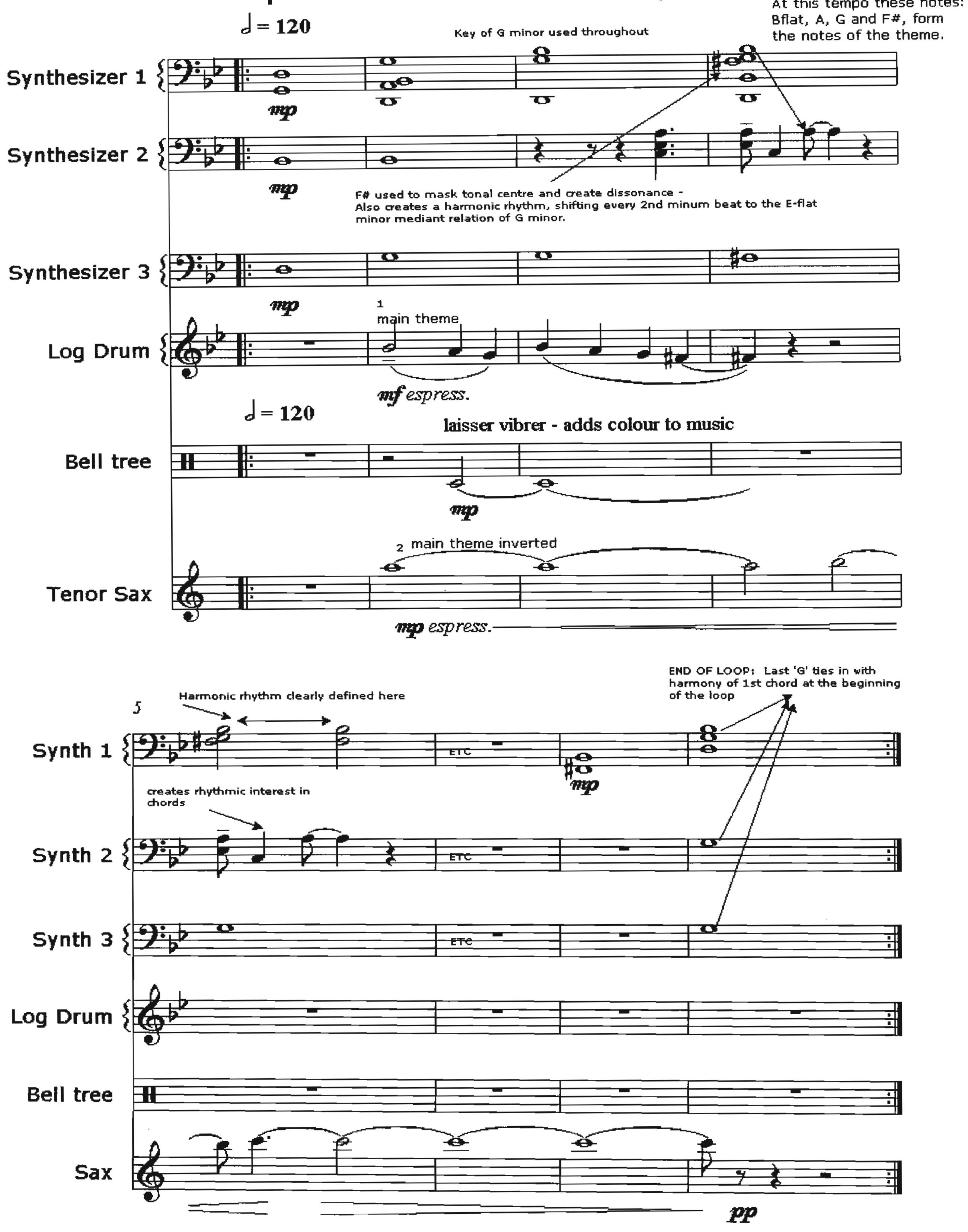
Hypermedia with Choice of Path Sets
User can select among navigational paths embeded in hypermedia structures

COMMERCIAL GAME MUSIC TRANSCRIPTIONS

1. TRANSCRIPTIONS OF MUSIC IN RIVEN: THE SEQUEL TO MYST:

Fig. 9.

Transcription - Beetle Room, Riven



At this tempo these notes: Bflat, A, G and F#, form the notes of the theme.

Key of G minor used throughout

F# used to mask tonal centre and create dissonance - Also creates a harmonic rhythm, shifting every 2nd minum beat to the E-flat minor mediant relation of G minor.

laisser vibrer - adds colour to music

Harmonic rhythm clearly defined here

creates rhythmic interest in chords

END OF LOOP: Last 'G' ties in with harmony of 1st chord at the beginning of the loop

Fig. 10.

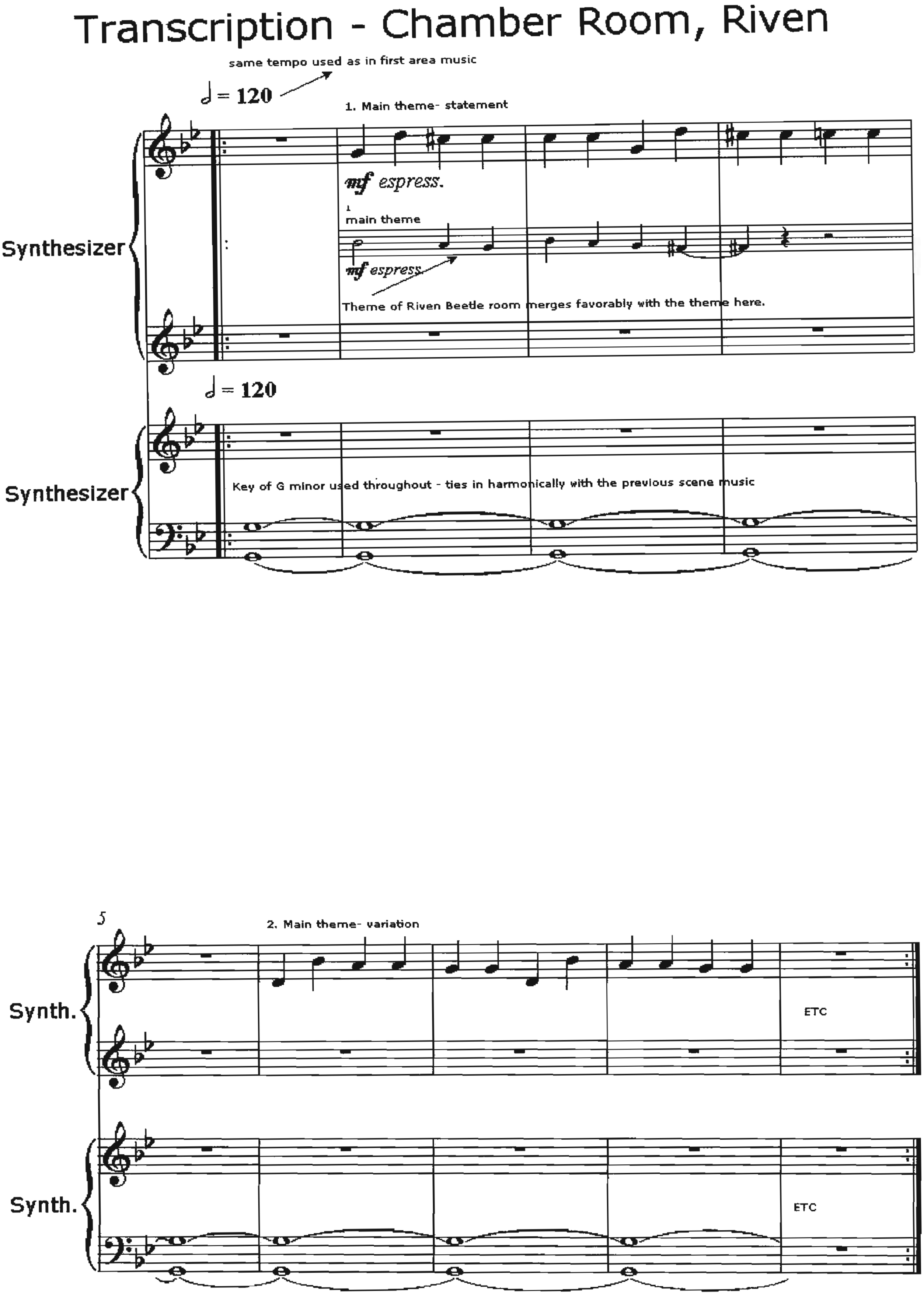

Transcription - Chamber Room, Riven
same tempo used as in first area music
♩= 120
1. Main theme- statement
mf espress.
1
main theme
mf espress.
Theme of Riven Beetle room merges favorably with the theme here.
Synthesizer
♩= 120
Synthesizer
Key of G minor used throughout - ties in harmonically with the previous scene music
5
2. Main theme- variation
Synth.
ETC
Synth.
ETC

Fig. 11.

Transcription - Blue Tunnel, Riven

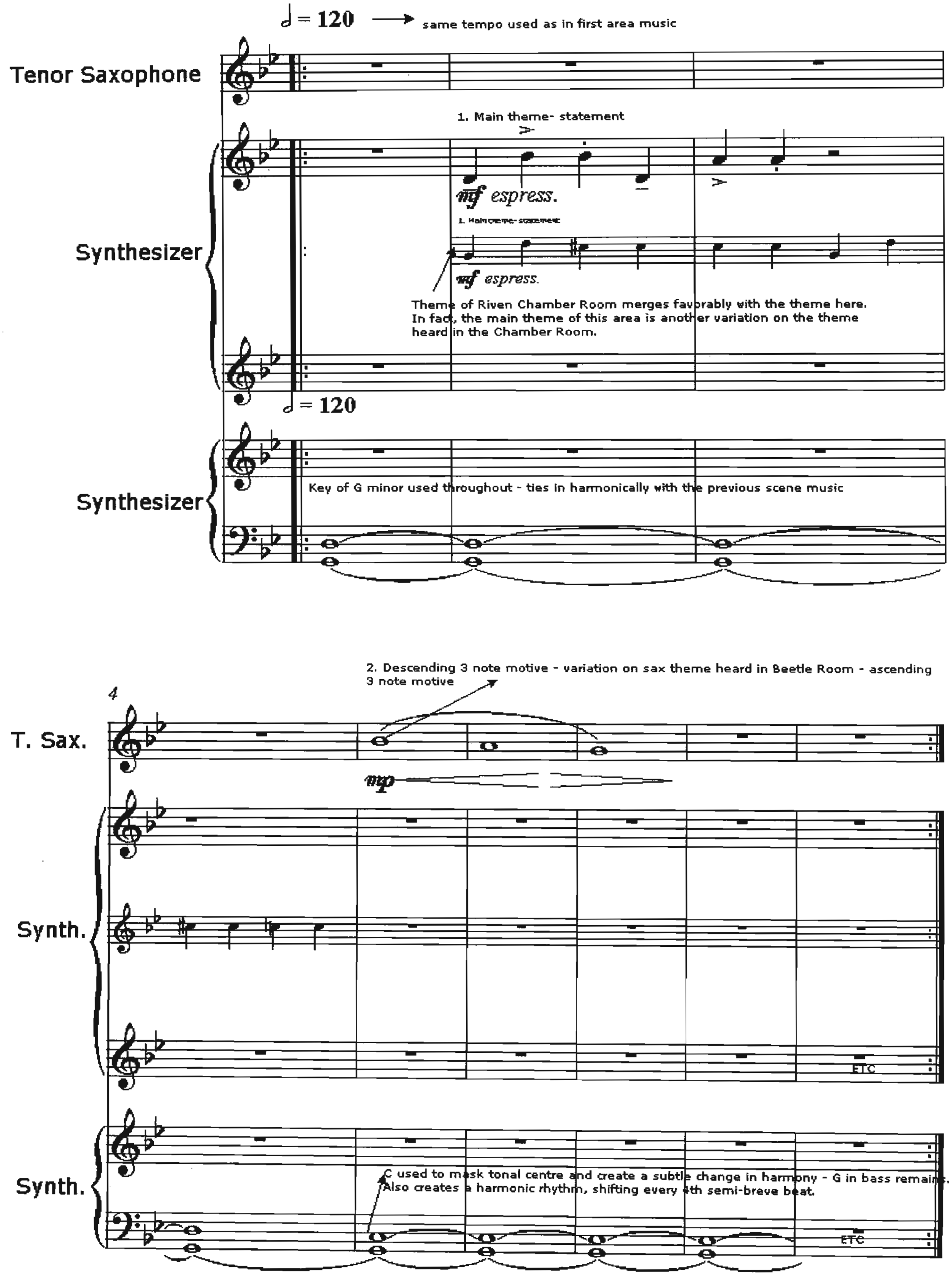

𝅗𝅥 = 120
same tempo used as in first area music
Tenor Saxophone
Synthesizer
Synthesizer
𝅗𝅥 = 120
1. Main theme- statement
mf espress.
1. Main theme- statement
mf espress.
Theme of Riven Chamber Room merges favorably with the theme here. In fact, the main theme of this area is another variation on the theme heard in the Chamber Room.
Key of G minor used throughout - ties in harmonically with the previous scene music
4
T. Sax.
Synth.
Synth.
2. Descending 3 note motive - variation on sax theme heard in Beetle Room - ascending 3 note motive
mp
ETC
C used to mask tonal centre and create a subtle change in harmony - G in bass remains.
Also creates a harmonic rhythm, shifting every 4th semi-breve beat.
ETC

Fig. 12.

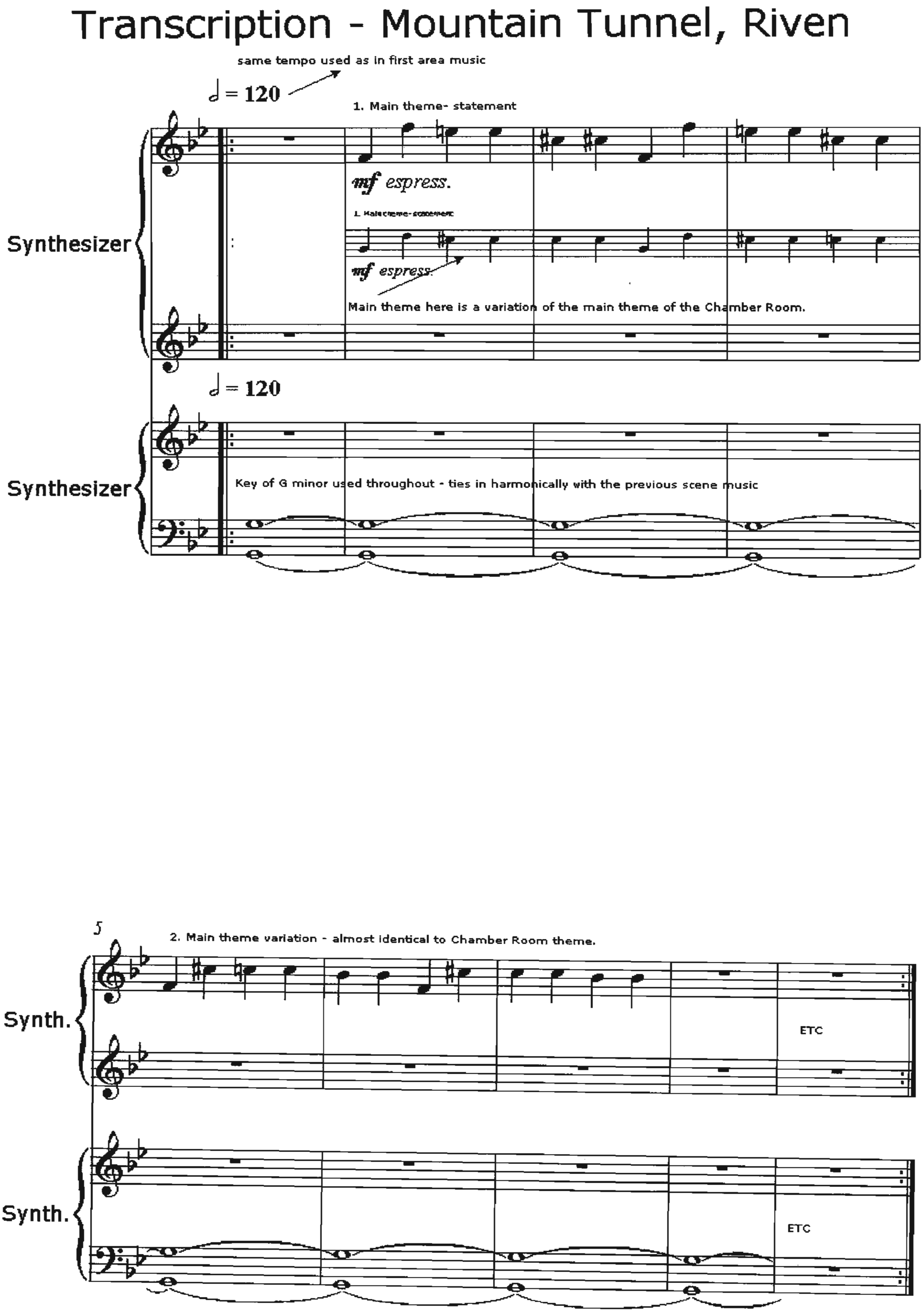
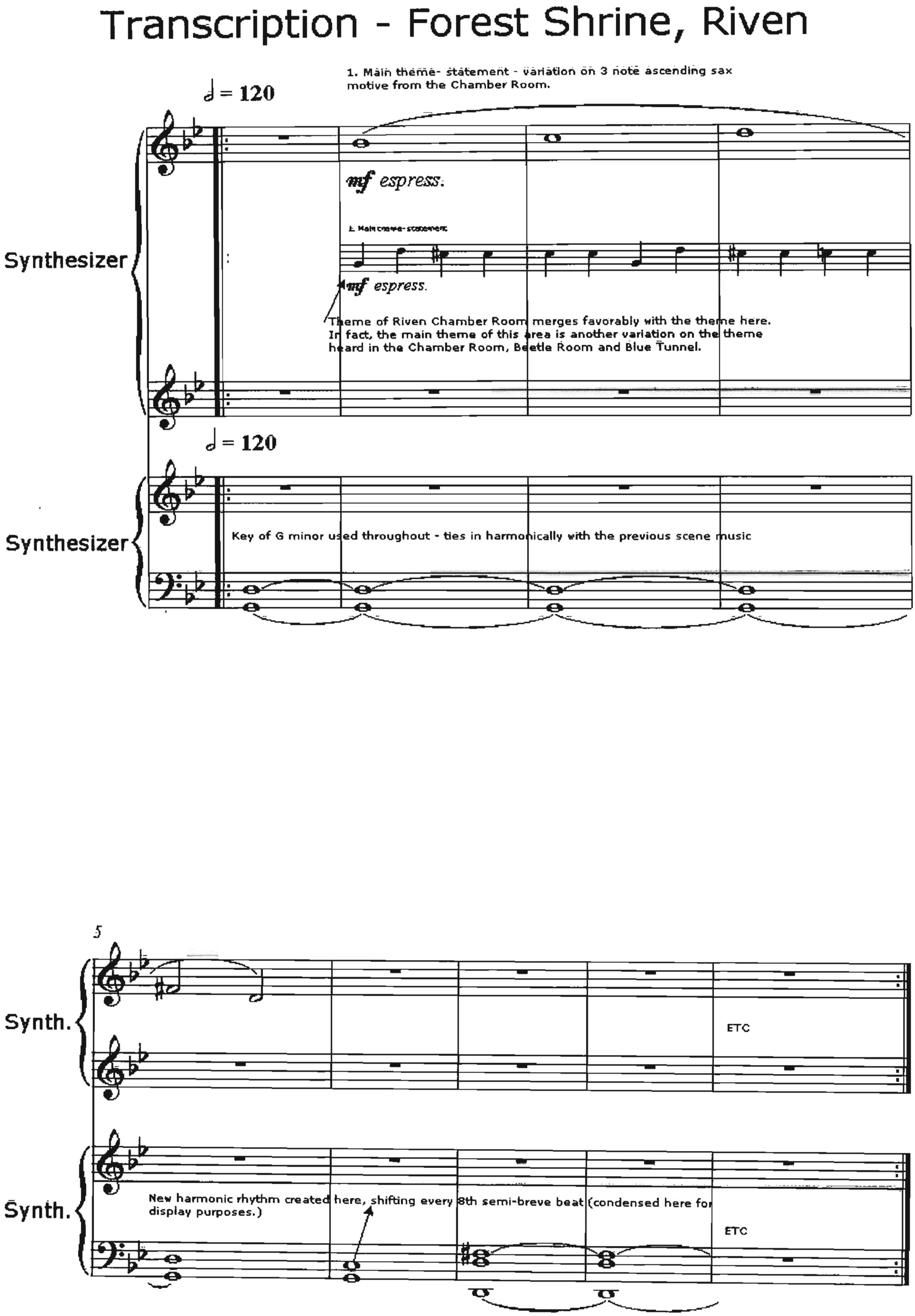

Transcription - Mountain Tunnel, Riven
same tempo used as in first area music
𝅗𝅥 = 120
1. Main theme- statement
mf espress.
1. Main theme- statement
mf espress.
Main theme here is a variation of the main theme of the Chamber Room.
Synthesizer
𝅗𝅥 = 120
Synthesizer
Key of G minor used throughout - ties in harmonically with the previous scene music
5
2. Main theme variation - almost identical to Chamber Room theme.
Synth.
ETC
Synth.
ETC

Fig. 13.



1. Main theme- statement - variation on 3 note ascending sax motive from the Chamber Room.

Theme of Riven Chamber Room merges favorably with the theme here. In fact, the main theme of this area is another variation on the theme heard in the Chamber Room, Beetle Room and Blue Tunnel.

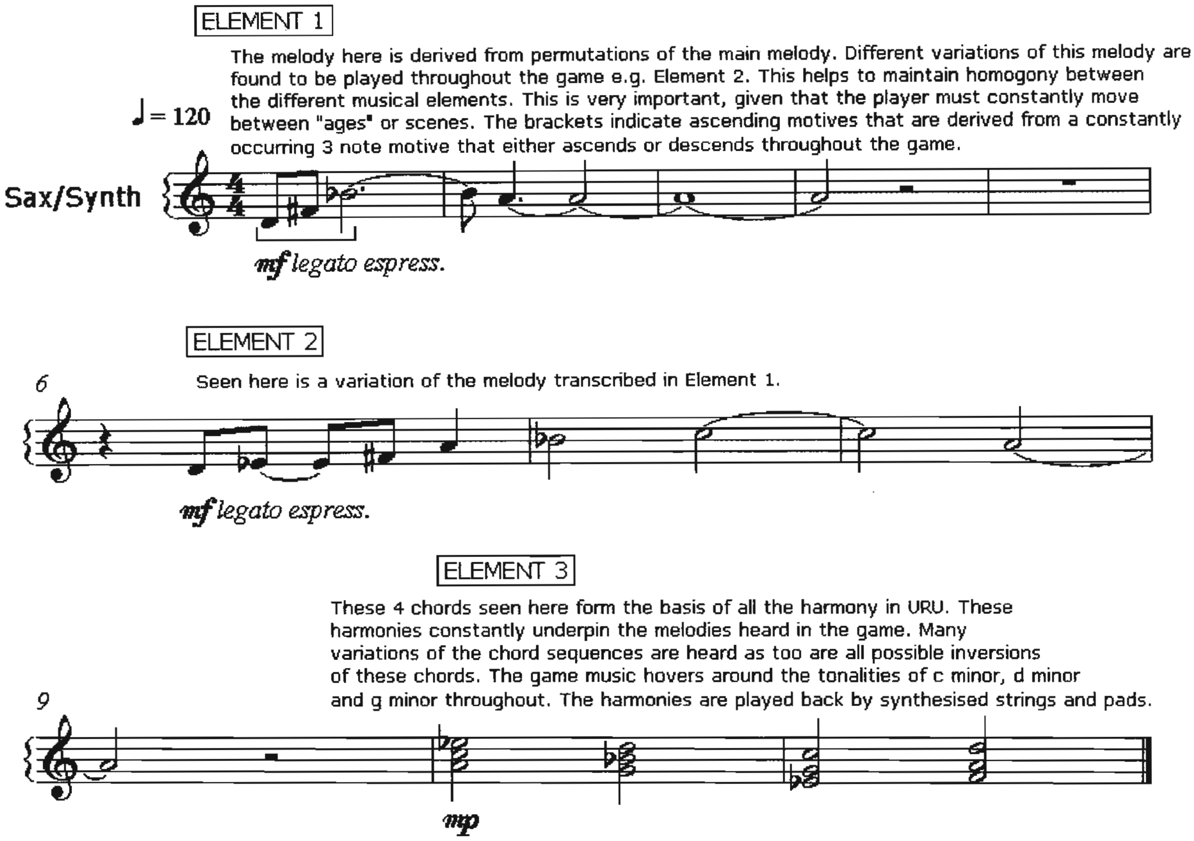
Key of G minor used throughout - ties in harmonically with the previous scene music

New harmonic rhythm created here, shifting every 8th semi-breve beat (condensed here for display purposes.)

2. TRANSCRIPTIONS OF MUSIC IN URU:

Fig. 14.

Transcription - Variations and other elements from URU



ELEMENT 1

The melody here is derived from permutations of the main melody. Different variations of this melody are found to be played throughout the game e.g. Element 2. This helps to maintain homogony between the different musical elements. This is very important, given that the player must constantly move between "ages" or scenes. The brackets indicate ascending motives that are derived from a constantly occurring 3 note motive that either ascends or descends throughout the game.

ELEMENT 2

Seen here is a variation of the melody transcribed in Element 1.

ELEMENT 3

These 4 chords seen here form the basis of all the harmony in URU. These harmonies constantly underpin the melodies heard in the game. Many variations of the chord sequences are heard as too are all possible inversions of these chords. The game music hovers around the tonalities of c minor, d minor and g minor throughout. The harmonies are played back by synthesised strings and pads.

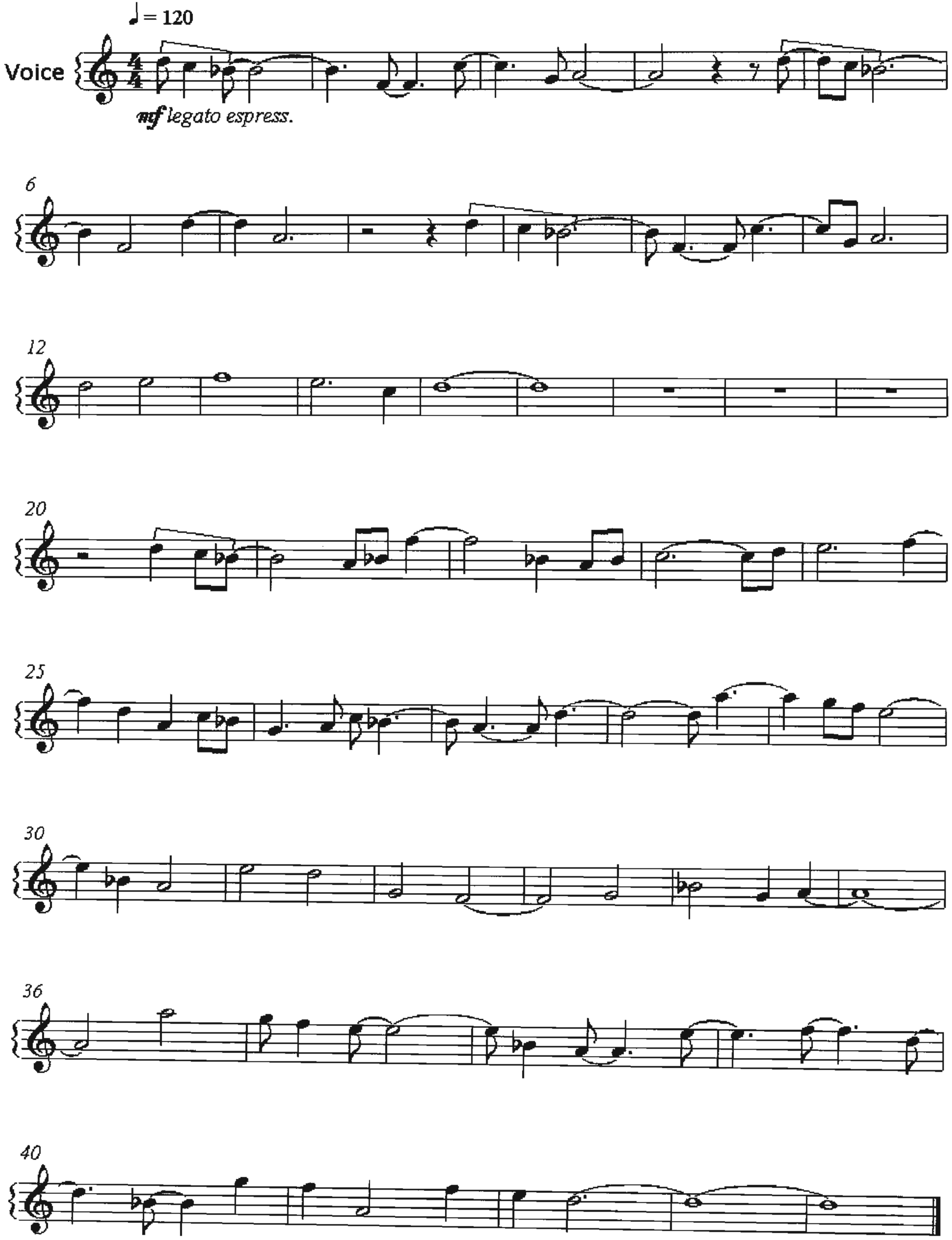
GENERAL NOTES:

1. As in the music to Riven, the composer has been careful to maintain thematic homogony between the musical elements, thus allowing for a seamless blend of music between scenes.
2. The difference of the music in Uru to that of Riven, occurs in the implementation of the elements themselves:
   - Each elements does not form part of a background loop that cannot change i.e. they are played back in the game as independent elements from the background loops when called upon by the gamestate.
   - The background loops themselves constantly shift and change in terms of instrumentation and are there to provide a harmonic base for the overlayed elements.

Fig. 15.

Transcription - Main Melody, URU

The melody seen here occurs in the main puzzle area of the game and where the game's resolution takes place. It is firmly rooted in the key of d minor, with modulations taking place from the centre key to g minor, F major and c minor. It contains the ascending and descending phrases found in the other melodies in the game. In fact, all the melodies are derived from the melody seen here. I have bracketed the ascending and descending motives which are derived from a constantly occurring 3 note motive that either ascends or descends throughout the game.



3. TRANSCRIPTIONS OF MUSIC IN TRON 2.0:

Fig. 16.

Transcription - Music Elements in Tron 2.0

NOTES:

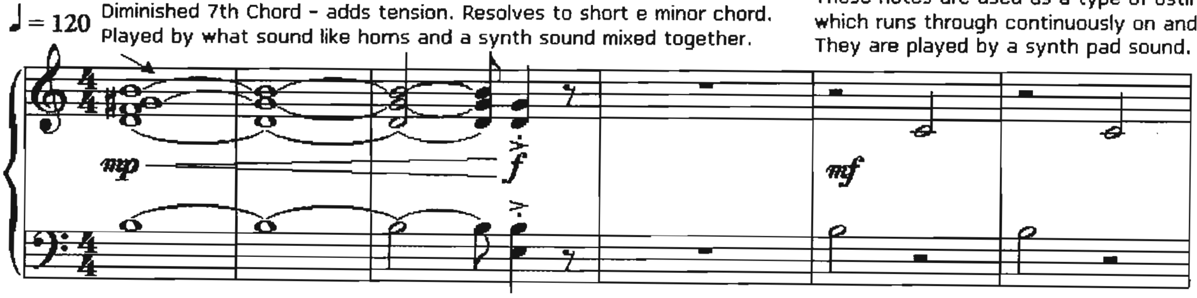
1. Throughout the game, the chords seen in element 1 are played back at different time intervals.
2. Above the chords, are layered each other element, depending on the gamestate. i.e. in approaching danger, elements 5, 6, and 7 are played back.
3. This is making use of few musical elements, with the outcome of the music consisting overall of 1 basic design - that of a minor chord starting on a and it's inversions with an added 6th. This makes up the tonality of the music in the game.
4. Because of this, each element is capable of being played back simultaneously and merging with one another.
5. Importantly to note in this game, is that there are no musical loops playing in the gamestate - instead, the designers have used these elements and variations to enhance the gameplay. This makes the music appear to "adapt" to user input from the gamestate. Also, the use of short musical elements that are all related, points to the adaptability of the original material (made up of the notes of the arpeggiated chord) which fits in with the design of interactive music and its parameters and principles.
6. The instrumentation used is very electronic - in line with the theme of the game.

ELEMENT 1

Diminished 7th Chord - adds tension. Resolves to short e minor chord. Played by what sound like horns and a synth sound mixed together.

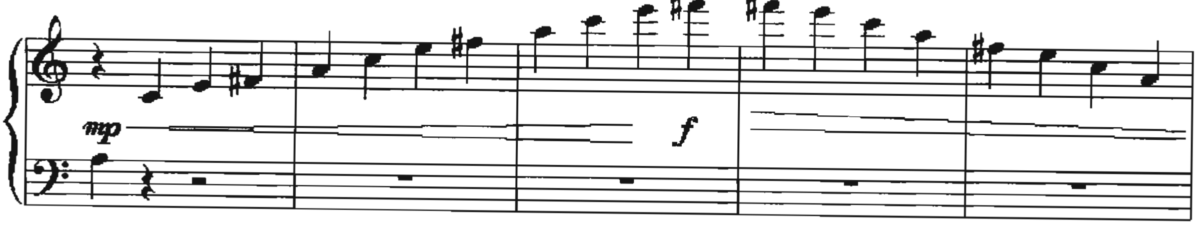
ELEMENT 2

These notes are used as a type of ostinato, which runs through continuously on and off. They are played by a synth pad sound.



ELEMENT 3

Seen here is a minor arpeggio with an added 6th. It runs up and down and is played back when danger occurs. The sound used is a type of sine synth.



ELEMENT 4

Seen here is a variation on the music of Element 2. The variation occurs before enemies are encountered. They are played by a synth pad sound.

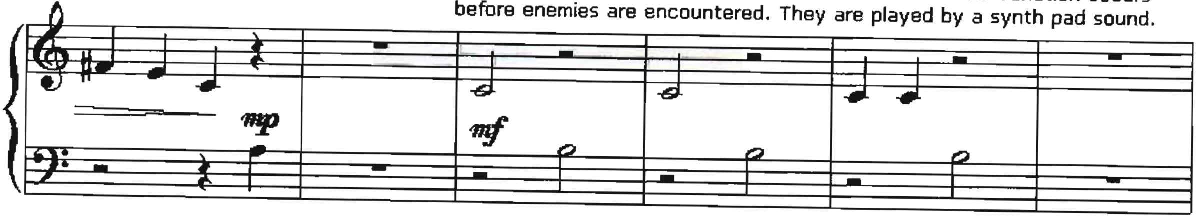
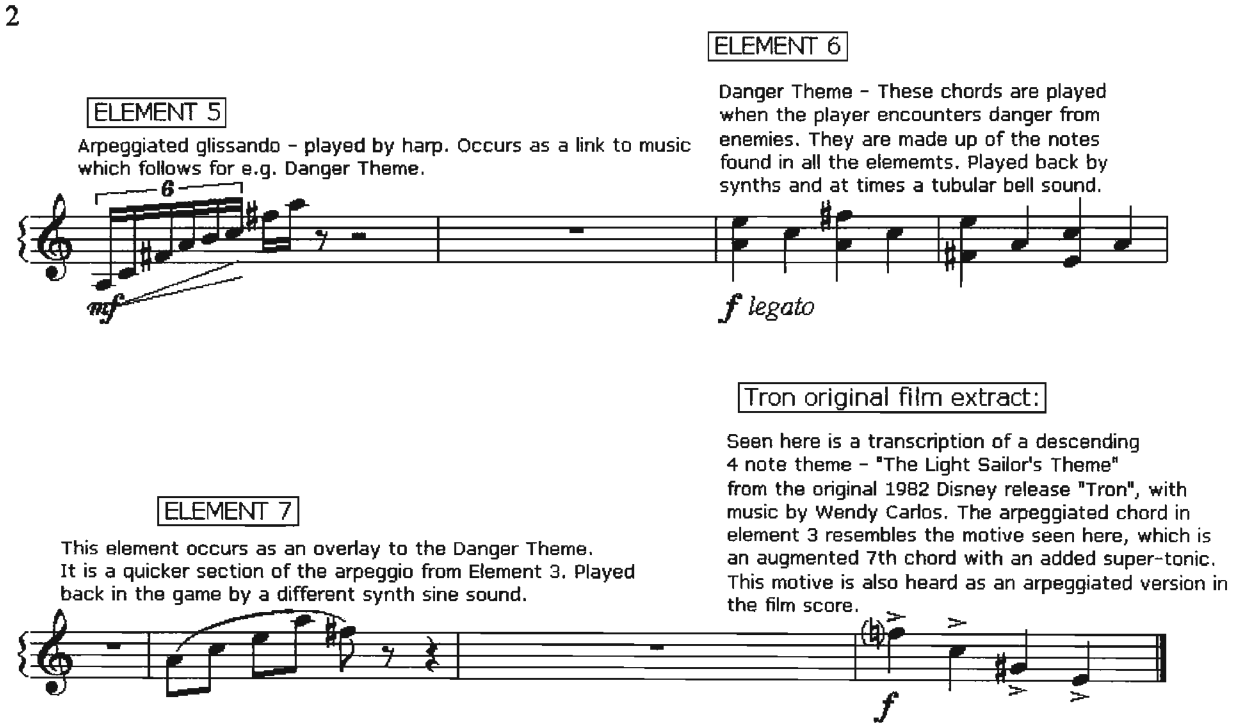


Fig. 16. cont.



ELEMENT 5

Arpeggiated glissando - played by harp. Occurs as a link to music which follows for e.g. Danger Theme.

ELEMENT 6

Danger Theme - These chords are played when the player encounters danger from enemies. They are made up of the notes found in all the elememts. Played back by synths and at times a tubular bell sound.

ELEMENT 7

This element occurs as an overlay to the Danger Theme. It is a quicker section of the arpeggio from Element 3. Played back in the game by a different synth sine sound.

Tron original film extract:

Seen here is a transcription of a descending 4 note theme - "The Light Sailor's Theme" from the original 1982 Disney release "Tron", with music by Wendy Carlos. The arpeggiated chord in element 3 resembles the motive seen here, which is an augmented 7th chord with an added super-tonic. This motive is also heard as an arpeggiated version in the film score.

TERMINOLOGY

Ambience: The macro sound referring to the environment a gamer is in. *Ambient sounds* on the other hand, are the individual components that make up the *ambience.* The *ambience* is therefore the sound/music that makes direct reference to the environment.

Audio Editor: A computer program specifically designed to create, edit, record and mix audio.

Avatar: A graphic rendering of a character in a computer game that is viewed in the 2nd or 3rd person.

Computer: For the purpose of this article, the word *computer* refers to either a desktop personal computer such as an IBM compatible computer or an Apple Macintosh or a laptop or I-Book. An electronic device that has the ability to store, retrieve, and process data, and can be programmed with instructions that it remembers. The physical parts that make up a computer (the central processing unit, input, output, and memory) are called hardware. Programs that tell a computer what to do are called software.

Consumer: Any person who uses the *media* for purposes of entertainment, education or commercial uses.

Dll - Dynamic Link Library. A feature of the Microsoft Windows family of operating systems that supports executable routines--usually serving a specific function or set of functions--to be stored separately as files with the extension .dll, and to be loaded only when called by the program that needs them. This saves memory during program execution and enables code reusability. [1]

DSP – Digital Signal Processing: DSP can be used to create equalization, compression, etc. of a digital signal. [2]

[1] http://www.google.co.za/search?hl=en&lr=&oi=defmore&q=define:DLL

[2] http://www.thewhippinpost.co.uk/glossary_A-G.htm

Effects: Sounds that mimic action. A door closing would sound like a door closing. This would either be recorded or manufactured using an *audio editor.* The sound effect of say water flowing would be created in the same way. These sounds that help create a realistic and natural feeling in a computer game are very similar to the foley sounds found in film.

Foley: A pre-recorded or simulated sound produced for a radio, television, film or theatrical program in order to suggest an actual sonic environment. The design of such sounds (often abbreviated as SFX), particularly when a complex acoustic environment is to be simulated, involves a thorough understanding of the structure of such a soundscape.[1]

Gamestate: term used to describe (whilst a game is running), the current activity of such a computer game.

Interface: A shared boundary where two or more systems meet; or the means by which communication is achieved at this boundary. An interface can be between hardware and hardware (such as sockets and plugs, or electrical signals), hardware and software, software and software, human and computer (such as a mouse or keyboard and display screen).[2]

MIDI: An acronym which stands for Musical Instrument Digital Interface. It is merely a standard language that let's instruments, devices, software and hardware from different manufacturers communicate fluently.[3]

Multimedia: Multimedia is communication that uses any combination of different media, and may or may not involve computers. Multimedia may include text, spoken audio, music, images, animation and video. The large amounts of data required for computer multimedia files makes CD-ROMs a good option for storage; but there are other ways of receiving multimedia communications, such as the World Wide Web. Multimedia programs are often interactive, and include games, sales presentations, encyclopedias, and more.[2]

[1] http://www2.sfu.ca/sonic-studio/handbook/Sound_Effect.html

[2] http://www.computeruser.com/resources/dictionary/dictionary.html

[3] http://vtg.org/cimonline/Glossary.html#S

Pad: Term used to describe a set of sounds designed to play back simultaneously on a synthesizer. 'Pads' could consist of one or more fixed sounds, forming part of a sound set on a tone generator or the sounds within the pad could be sampled and therefore created from any source and comprising of any sound imaginable.

Sampler: Also called a digital sampler. A type of synthesizer which derives its sounds from recording actual sounds (instruments or non musical sounds) and then storing them in computer memory, either floppy discs, hard drive, or recorded onto CD-ROM. They are used extensively for generating sound effects.[1]

Sequencer: Music software application for creating, editing and arranging music to create finished songs.[2]

Sound or Video Engine: computer software that is designed to run simultaneously with other software and provide streaming capabilities for audio or video.

Soundscape: An environment of sound (or sonic environment) with emphasis on the way it is perceived and understood by the individual, or by a society. It thus depends on the relationship between the individual and any such environment. The term may refer to actual environments, or to abstract constructions such as musical compositions and tape montages, particularly when considered as an artificial environment.[3]

Sound Design or Soundscape Design: A new interdiscipline combining the talents of scientists, social scientists and artists (particularly musicians). *Soundscape* design attempts to discover principles and to develop techniques by which the social, psychological and aesthetic quality of the acoustic environment or *soundscape* may be improved.[4]

[1] http://vtg.org/cimonline/Glossary.html#S

[2] http://www.thewhippinpost.co.uk/glossary_S-Z.htm

[3] http://www2.sfu.ca/sonic-studio/handbook/Soundscape.html

[4] http://www.sfu.ca/sonic-studio/handbook/Soundscape_Design.html

Synthesis/Synthesized: The electronic production of sound where no acoustic source is used. An electrical signal is produced which is the analog of a sound wave; that is, the voltage fluctuation in the signal represents that of the desired sound pressure variation. When this signal is fed to an amplifier and loudspeaker, the sound becomes an acoustic signal which behaves like any other sound.[1]

Tone Generator/Sound Module: A synthesizer without a piano keyboard. Since Midi allows one keyboard to literally play another, there is little reason to acquire more piano keyboards when wanting to expand your palette of sound choices. Buying tone modules is usually a bit cheaper than the keyboard version, and saves valuable space.[2]

[1] http://www2.sfu.ca/sonic-studio/handbook/Sound_Synthesis.html

[2] http://vtg.org/cimonline/Glossary.html#T

BIBLIOGRAPHY

(All internet sites accessed between 1st September 2004 and 25th September 2005, unless otherwise specified)

Internet Sites:

http://www.macopinion.com

http://www.geocites.com

http://www.mercury.sfu.edu *Accessed November 2002*

http://www.citd.scar.utoronto.ca *Accessed December 2001*

http://www.mushi.ingalp.org *Accessed September 2002*

http://www.altpop.com

http://www.soundtrackcentral.com

http://www2.sfu.ca/sonic-studio/handbook/

http://www.gamespot.com

http://www.gamasutra.com

http://www.gamemusic.com

http://www.gamedev.net

http://www.gameaudio.3dportal.com

http://xroads.virginia.edu/~HYPER/LEVIN/levtit.html http://www.InGaming.com

http://www.microsoft.com/directx

http://www.iasig.org

http://www.cyan.com

http://www.gamestudies.org

http://www.midi.org

http://www.vgyellowpages.com

Books:

Belton, John and Weis, Elisabeth. “Film sound, theory and practice.” Columbia University Press, New York, 1985.

Burt, George. “The art of film music.” Northeastern University Press, Boston, 1994.

Cook, Nicholas. ”Analysing Musical Multimedia.” Clarendon Press Oxford, 1998.

Emmerson, Simon, editor and contributors. “Music, electronic media and culture.” Ashgate, 2000.

Gorbman, Claudia, Unheard Melodies: Narrative Film Music. Indiana University Press. 1987.

Karlin, Fred & Wright, Rayburn. “On the Track – A Guide to Contemporary Film Scoring.” Schirmer Books. 1990.

Rowe, Robert. “Interactive Music Systems: Machine Listening and Composing.” Massachusetts Institute of Technology, 1993.

Ryan, Marie-Laure. “Narrative as Virtual Reality – Immersion and Interactivity in Literature and Electronic Media.” John Hopkins University Press, 2001.

Todd, Winkler. Composing Interactive Music: Techniques and Ideas Using MAX. Massachusetts Institute of Technology, 1998.

Articles:

Amory, Allan & Seagram, Robert. “Educational Game Models: Conceptualisation and Evaluation.” South African Journal of Higher Education. 17(2), 2003.

Brandon, Alexander and Miller, Mark. “A Direct Music Primer.” Interactive Composition Column 1.1 Interactive Audio Special Interest Group (IA-SIG). 1999. http://www.iasig.org

Brandon, Alexander, editor. “IA-SIG Newsletter.” The Interactive Audio Journal. Vol. 1. No’s. 1-7, 1999-2002. http://www.iasig.org

Carrol, Noel. “TV and Film: A Philosophical Perspective.” Journal of Aesthetic Education. Vol. 35. No. 1, Spring 2001.

Gansing, Kristoffer. “The Myth of Interactivity or the Interactive Myth?: Interactive Film as an Imaginary Genre.” Melbourne DAC 2003.
http://hypertext.rmit.edu.au/dac/papers/Gansing.pdf

Garnett, Guy, E. “The Aesthetics of Interactive Computer Music”, Computer Music Journal, 25:1. Massachusetts Institute of Technology, Spring 2001.

Harland, Kurt. Composing for Interactive Music. Gamasutra, 2000.
http://www.gamasutra.com/features/20000217/harland_01.htm

Hays, Tom. "DirectMusic Producer for the Masses." 1998. http://www.gamasutra.com/features/sound_and_music/ 19981106/directmusic_12.htm

Hutchison, Andrew. "Analysing the Performance of Interactive Narrative." Melbourne DAC 2003. http://hypertext.rmit.edu.au/dac/papers/Hutchison.pdf

MacDonald, Glenn. "A brief timeline of video game music." http://www.gamespot.com/gamespot/features/video/vg_music/

Miller, Mark. "Audio Trends." http://www.gamasutra,com/features/119991102/gameaudiosupp/trends_01.htm

More, Gregory. "Implementing Nonlinear Sound Strategies within Spatial Design: Learning sound and spatial design within a collaborative virtual environment."Melbourne DAC 2003. http://hypertext.rmit.edu.au/dac/papers/More.pdf [sic: pg 134 of article should reference http://www.themodernword.com/eco/eco_vizciano.html and not http://www.themodernworld.com/eco/eco_vizciano.html]

Myatt, Tony. "Strategies for Interaction in *Construction 3.*"Organised Sound. 7(2). Cambridge University Press, 2002.

Newton-Dunn, Nakano & Gibson. "Block Jam: A Tangible Interface for Interactive Music." Journal of New Music Research. Vol. 32. No. 4, 2003.

Ng, Kia. "Sensing and Mapping for Interactive Performance." Organised Sound. Vol. 7(2). Cambridge University Press, UK, 2002.

Noël, Carroll. "TV and film: a philosophical perspective." Journal of Aesthetic Education. Vol. 35 No. 1, 2001.

Rowe, Robert. "The aesthetics of interactive music systems." Contemporary Music Review, Vol. 18, part 3, 1999.

Sanger, George. IASIG Newsletter Volume 1 Issue 3 November 1999. MIDI Manufacturers Association 1994-2004. http://www.iasig.org/pubs/journals/jour0103.shtml

Stevenson, Robert. "The Art of Noise: Game Studio Recording and Foley." 2000. http://www.gamasutra.com/features/20000922/stevenson_01.htm

Subotnick, Morton. "Space Frontier Foundation Mars Site." 1997 – 2002. http://www.mars-frontier.org/show.php?page=/visions/quotes&section=visions

Truax, Barry. "The aesthetics of computer music: a questionable concept reconsidered." Organised Sound. Vol. 5(3). Cambridge University Press, UK, 2000.

Whalen, Zach. "Play Along – An Approach to Videogame Music". International Journal of Computer Game Research. Vol. 4 Issue 1. November 2004. http://www.gamestudies.org/0401/whalen/

Wilson, Stephen. "The Aesthetics and Practice of Designing Interactive Computer Events." 1993. http://userwww.sfsu.edu/~swilson/papers/interactive2.html

Yuille, Jermey. "Smearing Discontinuity: In-Game Sound." Melbourne DAC, 2003. http://hypertext.rmit.edu.au/dac/papers/Yuille.pdf

PhD and Masters Theses:

Bittanti, Matteo. "The Technoludic Film: Images Of Video Games In Movies (1973-2001)." Masters Thesis, San Jose State University. 2001. Accessed from: http://www.gamasutra.com/education/theses/20020501/bittanti_01.shtml

McGann, Neal. "Watching Games And Playing Movies: The Influence Of Cinema On Computer Games." Masters Thesis, Dublin Institute of Technology. 2003. Accessed from: http://www.gamasutra.com/education/theses/20040515/mcgann_01.shtml

Price, Charley. "Evolution of Interaction Schemes in Computer Role-Playing Games." Masters Thesis, Carnegie Mellon University. 2003. Accessed from: http://www.gamasutra.com/education/theses/20030929/price_01.shtml

Wanderley, Mortensen. “Performer-Instrument Interaction: Applications To Gestural Control Of Sound Synthesis.” PhD Thesis, IRCAM – Centre Pompidou, Paris, France. 2001. Accessed from: http://recherche.ircam.fr/equipes/analyse-synthese/wanderle/Gestes/Externe/kassel.pdf

Warren, Jonah. “Unencumbered Full Body Interaction In Video Games.” Masters thesis, Parsons School pf Design, New York. 2003. Accessed From: http://www.gamasutra.com/education/theses/20030915/jonah_thesis.pdf

Letters/Forum Discussions:

Stephens, Noel. Game Engines versus Rendering Engines – a letter to the editor of Gamasutra.com 26/11/2001. http://www.gamasutra.com/php-bin/letter_display.php?letter_id=114

Interviews and Interview Transcripts:

Interview with “Uru – Ages Beyond Myst” music composer Tim Larkin, conducted by Miles Warrington via email. November 2004.

Alexander Brandon interviews Guy Whitmore, June 2002. http://www.iasig.org/pubs/interview/guy_whitmore.shtml

Music4Games interviews Alexander Brandon. http://www.music4games.net/f_alexbrandon.html

Rob Sollars interviews Lennie Moore, composer and conductor. InGaming Magazine. http://www..InGaming.com

Software:

Amazing MIDI. Arakisoftware, 2003. http://www.pluto.dti.ne.jp/~araki/amazingmidi/

Cakewalk Pro Audio 8.0. Twelve-Tone Systems, 1999.
http://www.cakewalk.com/

CoolEdit Pro 2.0. Syntrillium Software Corporation, 2002.
http://www.syntrillium.com/

Grand Theft Auto III - Take-Two Interactive Software for Sony PlayStation PS 2.
http://www.rockstargames.com

Microsoft Direct Music & Direct Music Producer, 1999.
http://www.microsoft.com

Microsoft XP Professional. Microsoft Coroporation, 2001.
http://www.microsoft.com

Riven: The Sequel to Myst. Cyan, Inc. and Broderbund Software Inc. 1997.
http://www.cyanworlds.com/games/riven.php

Sibelius 2. Sibelius Software Limited, 2001.
http://www.sibelius.com

Tron 2.0. Buena Vista Interactive, 2003.
http://bvinteractive.go.com/

Uru: Ages Beyond Myst. Cyan, Inc. and Broderbund Software Inc. 2003.
http://www.cyanworlds.com/games/uru.php