

‘INTERACTIVE MUSIC’

Rob Bridgett for Computer Arts Magazine 2002

Abstract

What is Interactivity and why should it bother us linear musicians? The music we compose has more often than not been solely based on the traditional narrative structure of beginning middle and end, whether this is a three-minute pop song or a more large-scale orchestral work - the notion that the performance is going to be heard in exactly the order in which it was composed is one that we all feel comfortable with. Yet, increasingly the way we structure our music is changing, most of us have written tracks that are loop based for use on the internet and have provided several musical 'spot effects' that will work within that piece of music for when a button is pushed etc. Also the tools we use in terms of sequencers and editors are breaking down the notion of linearity in music composition in that we can take any section of our composition and move it easily around within the composition, via the copy, cut, paste method. Indeed the way that we use sequencers to copy and paste chunks of music and effects means that most of us composers who already use these tools and techniques will probably find no difficulty in preparing these pieces of music for use in an interactive game or application. It is conceiving of how these sections will function in the application and the preparation of the end material that is the only general barrier to production.

So why bother with these ideas of interactivity? The notion of music being self-generative and self-modifying is in fact an ancient one, the Aeolian, harp an instrument similar to a wind chime which creates random notes when blown in the wind, or a more common example the ringing of church bells, which have a specified number of different bells (or notes) which are played in a random and different order every time they are heard. So these techniques and approaches have been around for a long time, and we have probably not really been aware that they are familiar to us. Unfortunately a lot of the current writing and thinking about interactive music tends to be rather overly academic which does put up a rather intimidating barrier to those of us who just want to produce great sounding music, whether it's for games, web or any other interactive application.

Applications / Media and their influence over musical structures

Film and TV have influenced the structure of music for the last century, theatre and opera before them. Interactive applications and entertainment are now set to re-write the rulebook. Certainly within the last 100 years, film has been the single most important influence on narrative music, having been responsible for commissioning some of the most interesting, provoking and memorable music of the last century. Although the influence of film music can be felt heavily within the new wave of games consoles through increased use of 'orchestral' soundtracks, the actual structure of the music and the purposes for which the pieces are written is distinctly different. The reason for this re-structuring of music is due to the modular and interactive nature of the new media of game and web.

The first and most commonly used type of interactive music is still loosely a narrative approach, in that it has any number of beginnings, middles and ends but these can be played in a different order each time dependant on game-play or user interaction.

This type of interactive score incorporates several narrative modules for several purposes. These different modules can be thought of in simply defined terms as either narrative, continual, evolving or resolving.

A narrative piece of music is a piece in which plays straight from beginning to end without the need to be interrupted by user input. A good example of this would be a cut-scene or FMV movie in a game, as these sections work in a predictable and linear fashion, the old rules of film composition can be applied: the exact timings and lengths of events are known.

The second type of module, the continual, is basically a piece of music that needs to keep playing until a user input interrupts it. A good example would be a simple music loop on a menu awaiting input, or a particular section of a game in which the length of time for that section is an unknown parameter dependant upon many other factors.

The third state, which is an evolving module, is more complex. In narrative terms it is an unknown quantity, the length of time this music is to play for could be anything from 10 seconds to an hour, the same as for the simple continual loop, however there may be many other factors which influence this piece of music. A good example would be a combat situation within a game. Entering the combat would trigger the continual music, but then variables of how well or badly the player is doing in combat need to give audio feedback to the player and modify the music. If the fight is joined by two more enemies, the music will need to become more intense, and if the player runs to the side of the arena away from the fight to recover for a few moments, the music needs to reflect this less intense period of activity. This evolving sound can be achieved through manipulation of any number of musical variables, from pitch, timbre, orchestration physical modelling or effects filtering. And finally when the combat is over we need to smoothly get out of the combat music and back into a non-combat state, which brings us onto the final module, resolving.

The resolving module is a transitional piece, usually only played once which can be played at any time during the looping continual or evolving states and which signifies an end to the particular piece of music. If the player was successful in combat the combat music could fade out while the triumphant ending to the piece is played over the top, if the player was unsuccessful then a more tragic piece could be played. What you hear and when you hear it is therefore totally dependant upon the way the user interacts with the system.

The notions that structure in composition can be built in several modules is something that can be seen not only in games or interactive applications, but has been taken up within live TV and game shows. In the British TV show 'Millionaire' for example the background music is structured using a continual module and then resolving piece of music. A tense 'loop' of music is played while the contestant ponders the answers and then when the answer is revealed another 'finale' piece of music is played over this. Now, because no-one knows the exact timing of when the contestant is going to give the answer the looping tension-building music must continually play without foreseeable end, the only real ending to this music happens when the other piece of music which is played when the answer is revealed, again there are two possible types of ending for this resolution, the correct answer and the wrong answer and musical equivalents for these have been composed.

So here we have an example of music being composed and structured in two distinct modules, each with a distinct purpose.

Modular Interactive Music Making

As we are now aware of the modular nature of interactive music, we can begin to think of ways in which these modules can be designed in order that they work together. There have until now been very few applications which allow the composer to work directly in the interactive format, but recently Microsoft's Direct Music Producer (DMP) has emerged, a free application which allows interactive content to be developed for Games or the Web, or any interactive medium.

DMP is designed specifically for building interactive content and to this end it is structured around building modular variations, intro and outro modules, and variations on themes all of which are responsive to user input or defined or scripted. Indeed this application contains some very clever ways in which an interactive score can be built up and create smooth transitions between the various modules.

One of the more interesting features are the ability to program variants on things like 'one shot' sound effects, so for every footstep or sound event, you can have up to 32 variations of the same event, these can be played in order from first, in order from random, in random order, with no repeats and by shuffling. This feature can also be applied to the music that can be written within DMP.

One of the main ways of creating music within DMP is via creation of Downloadable Sounds (DLS instruments) which bear a striking similarity to software samplers in that you can take any sound or groups of sounds and map them onto a keyboard. This enables you to create any number of instruments, a sampled piano for example, and to control this via MIDI score within the main composition window. This brings us onto one of the main flaws of DMP in that it only really supports general MIDI information, unless you are using a DLS instrument which are unfortunately very memory intensive within interactive applications. Unfortunately the current expectations of the gaming and web audience is to hear fully orchestrated and professionally recorded audio on a par with film scores, and this is where DMP becomes tricky to handle if you are intending to manipulate pre-recorded modules of music. Even though the ability to stream in conventional Wave data does exist it is still incredibly difficult and not very intuitive to fade between sections of music preview the results in real-time. What you in fact have to do to get looping wave data to work properly is to assign the loops as DLS instruments occupying one key on the keyboard, and by the time you start working with lots of different modules things become extremely unmanageable and memory intensive. As an extension to the problems with DMP's manipulation of wave data, the ability to manipulate timbre is also seriously lacking from the application.

Overall the application is very difficult to use and feels like it is in very early beta stage, as there are so many features that the program lacks, and it does seem to have been developed for a purely General MIDI driven set up. However Microsoft must be praised on the development of this tool as it is more than a leap in the right direction for musicians wanting to create music for interactive applications.

Generative Music

A second approach, and one less reliant upon the familiar modular narratives of film and linear music is that of generative music. This bears more in common with the Aeolian harp and church bells mentioned earlier, in that some basic parameters are set and then the music is allowed to evolve randomly, never the same from one performance to the next.

This approach is often considered an overly mathematical way of composing music, however the control over all the parameters is still totally within the grasp of the composer. These generative sections of music can either be built over time in a linear fashion, or can be employed using a continual loop. The generative score can also be thought of as an extension to the works and ideas put forward by John Cage on aleatory music, and more recently Brian Eno who has worked extensively with what is currently the best known of the interactive music applications, SSEYO's KOAN.

SSEYO KOAN

SSEYO have recently won an Interactive BAFTA technology award for their KOAN plug-in, and the application that allows you to create content for this platform is equally as intriguing and exciting. Established as far back as 1986, this application has been geared towards presenting the user with a simple and effective way of creating ever changing, self-generating music. The number of parameters that can be set by the musician are vast, and include all the basic musical values we are all familiar with such as scale, chord progressions and harmonic rules, and even refreshingly, timbre.

The results which can be heard by visiting SSEYO's home page (<http://www.sseyo.com/>) are very refreshing in that they are in no way like general midi, this is due mainly to KOAN's use of built in FM synthesis, all of whose parameters may form part of the interactive nature of the music.

There are several main features within KOAN that range from predefined to algorithmically defined.

1) Rhythmic voice. This value offers a great deal of interactive scope with some of the parameters being 'phrase pitch range', 'phrase gap range', 'phrase length' , etc

2) Ambient voice. This involves similar constraints as the rhythmic voice, yet allows longer more ambient notes to be generated, as well as being capable of freely ignoring the rhythm track.

3) Follow voice. This follows, as the name suggests, activity in a previously defined voice. But you could set further parameters such as following the lead-line at an interval of a minor third, or to do this four notes behind to create a canon-like structure.

4) Repeat bar. Simply effects how 'repeat' functions.

5) Fixed pattern. This feature will basically play a pre-defined midi sequence, however it can also be told to behave in a generatively adaptive manner or follow adaptive occurrences in other voices.

6) Listen voice. Again this simply takes another pre-defined voice and generatively creates an accompaniment to that line based on whatever rules you wish to define for it.

One of the downsides of the power of the KOAN application is that after some time the lists of numbers and parameters can get extremely confusing. However a quick and easy way to get started is to make the KOAN application generate a random sound set-up, which you can then tinker about with to get the hang of what parameters change what sounds.

Aesthetic Problems

Interactive music is normally viewed from a technical rather than an aesthetic perspective. This means that composers have needed to work in ways constrained by the software that they are using, rather than being able to purely realise musical ideas without being limited by the sounds used by the hardware. Although many of the methods of generating interactive music are still reliant on the purely modular approach, what is required is an outlook based more on a halfway point between this modular approach and the generative approach.

Getting beyond general MIDI

The problem with many of the interactive music creation applications currently available is that they rely heavily on general MIDI or native computer MIDI sound banks from which they draw their sounds. This is of course bad for the person who is using the end application or game, as the end results can sound not a million miles away from the works of John Shuttleworth.

Ways to circumvent this do exist, such as utilisation of the DLS libraries (or downloadable sounds) within Direct Music Producer, which allow the mapping of any particular WAV sound across the virtual MIDI keyboard in order that the synthesiser can then use these sounds. This method is flawed at present due to the vast amount of memory that such DLS sound files take up. By using greatly compressed file formats such as MP3 or Ogg Vorbis, these timbral flaws can begin to be eliminated.

Where next? The Future of Interactive Music.

This is a brief look at apps and advancements on the horizon for interactive composers, where they may be heading and what would make life easier for linear systems to adapt the notions of interactivity into their make-up.

In terms of future developments for interactive music production and implementation, a number of advancements are still sorely in demand. The built-in dedicated audio processing hardware, which is native to the XBOX console, is a step in the right direction for real time effects filtering and manipulation. Other consoles need to make these real-time advancements possible; DirectSound on the PC has been making these advances in terms of CDROM and PC applications. However it is a severely neglected element of interactive web software such as Macromedia Flash and Shockwave in that there is no real time DSP effects such as simple reverb and delay. The introduction of these effects would immediately increase the scope with which interactive and real time sound and music can be filtered and altered, whilst dropping memory size and download time. The limitations of sound-object action-script for Flash are also too clearly shown up for the musician who is not a gifted programmer.

Another major advancement in terms of real-time modified timbre is that of physical modelling whereby the timbre of a synthesised sound is radically altered based on LFO or other real-time alteration of the sounds, the Native Instruments Absynth module gives one some idea of how this kind of sound manipulation works when

tied to a particular parameter, and the creative scope for an application that allows this to be simply and effectively executed is quite vast.

There are various superb examples of how amazingly realistic the timbre of physical modelling can sound to be found on TaoSynth's homepage <http://web.ukonline.co.uk/taosynth/>

However, as this is only available on Linux, and demands a high understanding of code in order to implement, it is a long time until these features can be wrapped in a user interface that non-coder musicians can actually experiment with. The one thing this does point to is a not-too-distant future where physical modelling will form an ally with, and possibly replace altogether, using real samples, as physically modelled sounds can be controlled completely by the computer. Physical modelling also has great things promised for vocal synthesis - whereby all kinds of weird and never-before heard creatures and will be conjured forth, also perhaps opening a way to the holy-grail of voice-overs whereby the voice of a long dead actor, such as Orson Welles, could be given new lines of dialogue. This technology also has a sinister side when one considers the potential uses for vocal physical modelling, in that anyone would be able to mimic anyone else's voice within a computer... think of the scene in Terminator 2 when the T2000 mimics perfectly the voice of the boy's mother... it's quite frightening, but this is where current research is heading.

Other less sinister areas for future enhancement of interactive applications lie in the type of music data they can deal with. This again refers to the drawbacks offered by pure general-MIDI support for applications like DMP. Although the actual MIDI clock features are essential in creating timings for interactive modules to function within, by enabling support for music loops via programs such as Sonic Foundry's Acid, and support for these in various file formats, the timbral nightmares of general MIDI can begin to disappear and make way for use of real recordings.

Again on the subject of MIDI, the eventual evolution of DLS libraries will form something similar to what we currently know as Software Samplers such as Nemysys' GigaStudio. These will enable the MIDI functionality to become much more realistic, one can imagine the power of the GigaPiano made into a fully accessible DLS instrument on a games console, integrating elements of pre-scored modular themes with elements of random generative music, also combined with MP3 or lossless sound file compression to make the DLS samples so much more memory efficient. Of course there is ample scope for all of the applications I have described to become functional in one application, comprising modular arrangement of MIDI powered software samplers and looped sample information, sections of Koan's generative software synth, offering physical modelling and utilising real-time effects and third party plug-ins, so that eventually the interactive score will be a case of setting up the rules by which an application generates sound from every conceivable parameter.

Taking all these concepts to their logical conclusions one can foresee a time when the computer or application is creating every sound in real-time and the composers job is to merely create and define the parameters within which the music and sounds will function - and even this task may well eventually be best left up to the computer. However one feels that this will somehow be lacking in the emotional 'human' feel that we get from music, even in the harshest electronic compositions there is a human element, and until computers can effectively mimic these emotional cues to such a degree that we cannot tell the difference, music will still remain a strictly human endeavour.

Approaching Interactivity: Creating your own simple interactivity and generative scores.

Interactivity can be based on very complex rules, but it need not be any more complicated than creating a track in ACID or some other loop based sequencer and exporting each individual piece of the score in the modular forms described earlier. However, getting a musical composition that doesn't sound like it is continuously looping can be achieved quite easily through generative processes.

Generative audio need not be reliant on a complex set of rules, often very simple approaches generate mathematically and musically beautiful pieces. These techniques may also be applied as a starting point to traditional linear composition (aleatory composition).

One such approach would be to simply create five WAVs of varying lengths consisting of separate notes. These notes when played together may make up a chord, but when all these sounds are played simultaneously and looped, due to their differing lengths will play together at different time intervals. An example which is made from several different keyboard notes can be created quickly and simply using Macromedia Flash

This is simply done by making each note a separate Flash movie with differing loop timings from the other notes, there are five notes in all and they continually generate an ever changing chord - this can be achieved with any sound (even loops that fade in and out can be simply and effectively implemented within Flash) and any number of sounds all with fairly low memory usage, and as they are separate elements they don't have to all load up before anything is heard.

...possibilities

As can be seen the interactive nature of music and sound effects can be as simple or complex as we decide. The main problem is indeed one of conception of how the elements all function in relation to one another, and not one of production. The music we produce already on linear systems can be simply and effectively adapted for use in interactive systems. We will perhaps not see interactivity in every musical and performance situation, but we have already begun to undermine the artists carefully considered album running orders via use of random access CD playlists and MP3 playlists...perhaps it will eventually permeate every musical situation we are currently comfortable with.