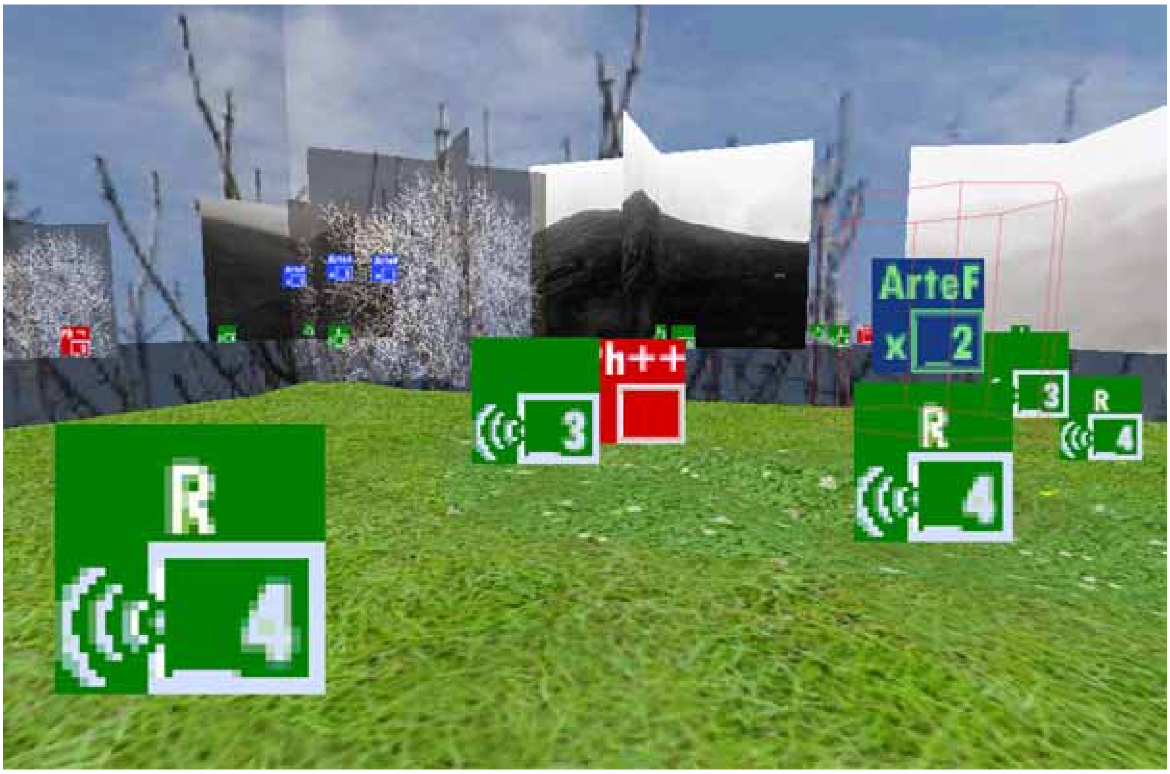
School of Computing and Mathematics
MSc Computer Aided Graphical Technology Applications



Andrew Paterson

Developing an archaeological method for authoring sound in a 4D space

Submitted in partial requirements for the degree of MSc CAGTA
Sept 2001

Supervisor: Clive Fencott

i) Acknowledgments

I would like to thank firstly Dr. Gerhard Eckel, Coordinator of the LISTEN project at GMD, Bonn, plus Larry Sider, Director of the School of Sound, London, for their support and invitation to attend the First LISTEN Experts Workshop in Glasgow, April 2001. These stimulating two days consolidated the main impetus and argument of this thesis.

Deserving thanks also is Deborah Long, Research Coordinator at Kilmartin House, Centre of Archaeology and Landscape Appreciation at Kilmartin, Argyll, Scotland. A two-week period of residence at the Centre towards the end of June introduced my research to the study of archaeology. Although the books I required were actually in Glasgow, my thoughts gestated at this remarkable place.

Much gratitude must go to my project supervisor, Clive Fencott, for his support and committed belief in what is admittedly an unconventional approach for a graphical technology application.

ii) Abstract

The ambition to create immersive audio-augmented environments raises problematic issues of time in a spatial environment, highlighting the disparity between temporally framed sound and interactive agency. Initial research into interactive virtual environments reveals that the user creates their own individual narrative as a result of their movement through the space, accumulating fragments of narrative to create meaning. Such a process suggests a parallel to the actions of the archaeologist: charting and gathering items of interest from an excavation, to create an interpretation of history or agency at the site. This thesis identifies archaeological stratigraphy as a practice that records complex spacio-temporal paradigms, and explores its relevance as a methodology to assist sound design in a 4D environmental space. Based upon the premise that the author of a time-based soundscape creates a database of narrative fragments and clues for the user to find, the paper proposes a distinct design process that reverses the excavation procedure, and re-imagines stratigraphical layers as successive phases of sound in the present tense.

The Unreal 3D engine and editor are chosen as a test environment. Several multi-functional classes are designed and programmed using UnrealScript - similar to a combination of Java and C++. These classes offer a variety of options for playing sounds, manage the passage of phases, and allow for interactive agency. Both theory and code are applied to construct a creative virtual environment, illustrating the arrangement of consecutive and consequential sounds into data-lists. Ultimately this 'archive' is placed within the Editor environment, to create an experimental soundscape.

Experience of production infers that computer-generated stratigraphical sequences are a useful step towards flexible creation and visualisation of relativity between units of narrative content. Likewise the multi-functional classes fulfill the purpose of translating these relations in the terms of agency, although 'real' time remains to be addressed satisfactorily. It is recognised that further temporal relations have to be identified beyond the present approach that accounts for sounds two nodes of existence – the beginning and the end. The lack of malleability of the record during run-time, indicates next step research is necessary to consider programmable behaviours or AI attributed to the sound actor, and how such features would develop narrative potential.

However, the overriding opinion gathered from the investigation is that although the metaphor of archaeological record should be treated as a state of transference and modification, rather than empirical and static, it represents a structure of narrative potential that can be released by the interactor's agency.

iii) Contents

iv) List of figures

1 Introduction

> "The space of landscape is at once cultural and natural, connecting values, modes of perception and representation, experiences, artefacts, histories, natural histories, dreams, identities, narratives, memories in networks of cultural ecology. Everything that goes with living in a place. Though historically layered and composed as tracks and traces, landscape is beyond simple conceptions of depth and surface, beyond the linearity, chronology, narrative and physical cartography. Lived meaningful inhabitation, of varying time length and subject to varying degrees of fragmentation and loss through time, landscape is a multi-temporal and complicated, folded *cultural* topography." [Shanks01: 293]

During our experience of everyday environments, a combination of visual, aural, tactile and smell indicators confirm the sense of presence. Our continued 'presence' in an environment is confirmed to us by constant monitoring of the changes in the different sensual feedback, determining a certain spatial position of relativity in the environment. The recognition of changes in each of the sensual information allows us to imagine a duration or habitation of the environment over a relative time period.

The aural sense is a receptor for sound, produced by the vibration of matter in space and time. Sound gives us information about the physical structure of things in our environment, such the material of the ground we walk upon, and also the space in which they are heard, for example, high reverberation as an indicator for a large cavernous space. The question of 'where' a sound is located is relative to the body of the listener. Hence the resonance of a space or environment indexes the movement of the hearer in the environment [Lerner97: 3]. It can also be argued that hearing, as the vibration of sound entering our body, monitors the reverberation of the space or environment, and the decay of the sound in time, indexing our body with a sense of time
[Lerner97: 2].

Throughout human evolution and history, it is only within the relatively recent 150 years that it has been possible to record sound, and replay it within a different environment, location and time period. For this reason, within the construction of virtual spaces in 3D computer environments, sound is recognized as a vital agent in the illusion of 'being', acknowledging its capacity to actualize presence of one space in another; in a virtual environment, actualizing the physical space of the user in the virtual space of the simulation. However, due to the dominance of graphics technology in the development of virtual environments, sound is often used as an augmentation to the visual interpretation of the space. This approach has led to the dominance of visual objects and structures defining the interactive and immersive experience, rather than sound's spacio-temporal indexes of presence.

Increasing success in digital binaural sound rendering, head-related transfer functions[1], and surround sound systems, has provoked creative and imaginative research proposals to develop virtual and augmented environment research focused specifically on the immersive spacio-temporality of sound. The LISTEN project, coordinated by Dr. Gerhard Eckel of GMD[2], Germany, aims towards producing immersive audio-augmented environments that will "provide users with intuitive access to personalised and situated audio information spaces while they naturally explore everyday environments" [Eckel01]a. This will be achieved by augmenting the physical environment with a dynamic soundscape, experienced by the users in real time through motion-tracked, wireless headphones.

A key feature of the LISTEN project is to allow an interactive audio environment, which records the movement of the user and intelligently adapts accordingly, allowing the option of avoiding repetitive audio loops. The project presents the notion that a virtual or augmented space contains, not only spatially mapped sound samples, played depending upon the user's position, but also that these samples may change depending upon the number of visits to that position. The interactive nature of the sound environment means that a history or memory of occupation within the environment accumulates, as the user moves and experiences the space, and that each sound plays, therefore, according to the accumulated history.

The objective of GMD is to develop a virtual reality based authoring tool for the design of the audio augmented environments. In consideration of design process, the spatial positioning of sound sources within a XYZ coordinate system is relatively straightforward, using established graphical mapping techniques, applied in a computer-generated representation of the space. Yet, how can a 'history' (of sounds within the space) be accounted for in interactive computer environments? History occupies a further dimension, that of time, dividing the continuum of experience into different stages and events. What methods are necessary to design and manage the dimension of history, and its component stages? At the First LISTEN Experts Workshop[3], such questions were raised asking 'how can LISTEN articulate and integrate various aspects of time: the time it takes for someone to explore the space, the time it takes to hear something unfold, the time frame of the content?'.

Archaeology, concerned with the occupation and engagement of people within spatial and time dimensions, can, in hypothesis, offer different approaches to the relationships between these

[1] Binaural sound rendering mimics the human acoustic system by making recordings using micro-phones embedded in a dummy head. Head-related transfer functions, or HRTFs, are mathematical functions based upon the sounds position, which can applied to the sound wave and take into account of the cues used by humans to localize the sound.

[2] German National Research Centre for Information Technology, based near Bonn. www.gmd.de

[3] Held at the LightHouse, Glasgow, 9-10 April 2001.

factors within virtual or augmented environments. For example, the empirical method of an archaeological excavation records, in great detail, the traces of human engagement in one site, over a period of time, while a theoretical approach engages with the 'archaeology of a place and landscape', attempting to understand how an environment may have been *perceived* by people – what *meaning* it held for them - at different phases of occupation. Within recent years the issue of agency in the representation of archaeological record, has been a topic of debate, interpreting the excavated space in the terms of occupancy and inhabitation.

Hence this master thesis will explore the discipline of archaeology, seeking assistance for the development of a design methodology managing spatial and time-relative mapping of sound in virtual and augmented environments. The thesis is divided into the following chapters of engagement:

The following chapter (2), introduces underlying key concepts - authorship, narrative potential, agency and sound within interactive virtual environments – while also defining features of an augmented sound environment such as that proposed by the LISTEN project.

Chapter 3 describes the archeological process and method of recording excavations known as 'archaeological stratigraphy', from the recording of deposits to the creation of a relative-time sequence of occupation. Chapter 4 acts as a crossover section, interpreting the archaeological theory as relevant to its possible use in the authoring of spacio-temporal sounds within virtual and augmented environments.

Further, as authorship is the main focus of this investigation, Unreal Editor software is introduced as a virtual environment design tool in chapter 5. Operating within the limitations of this environment, the chapter proceeds to describe custom-made classes designed, and programmed using object-orientated UnrealScript. The critical success factor for the design of these objects are that they can be placed within the virtual environment to play sounds, manage the passing of time, and allow for interactive agency.

Chapter 6 illustrates the proposed methodology with a creative virtual environment entitled 'Garden Monologue', combining a visual navigational space with an interactive soundscape of noise, voice and musical elements. As a production process highlights the strengths and weaknesses of a method, analysis, adaptation and further developmental issues are discussed to bring the thesis to conclusion.

2 Background concepts.

2.1 Narrative potential

Virtual environments are developing a technical and creative sophistication whereby much of the research in the medium is being diverted from technical 'how?' questions to creative 'why?' questions. Some of these new answers being returned are that virtual environments offer story-telling and narrative potential, although there are mixed opinions regarding how the seemingly disparate factors of interactivity and narrative can be integrated [Jull98] & [Meyer95: 236], and often depend upon insights into computer games plots and literary theory.

New interpretations of what narrative is, and how it is experienced within interactive media, are necessary to reconcile the above factors. Structural analysis of narrative by [Barthes88] has been recalled to identify underlying components that may offer clues toward how to approach this new interpretation, highlighting relevant characteristics such as: different levels of meaning, or individual basic units which sit at levels of action or discourse, while also facilitating the linear development of narrative structure; also a confusion of *consequence* - the result of an event, and *consecution* - what follows in succession [Fencott01]. Continuing from the same source, the transformation of the afore-mentioned basic units into meaning, and therefore a sense of narrative, is suggested as "conscious work" in virtual environments by the nature of *participation* within it.

However, perceptual opportunities, referring to attributes ascribed to an object, such as attractors, rewards and connectors [Fencott99], can be interpreted as the basic components of narrative within virtual environments in the terms of agency[4]:

> "The organization of these [perceptual opportunities] into a perceptual map allows us to consider their configuration in terms of larger structures – such as routes, choice points, challenge points and retainers – that represent the narrative potential of a virtual environment. Narrative potential can be seen as both the degree to which such structure can accumulate to form a meaningful experience and the degree to which content preserves its meaning over the course of the narrative rather than being overwhelmed by the pleasures of agency" [Fencott01].

By considering the similarities and differences between traditional narratives, interpreting the components defined by [Barthes88] above, and perceptual opportunities in a first-person scenario, certain conclusions regarding narrative potential in virtual environments have been presented which suggest that:

---

[4] Agency refers to the formulation of plans of action, and perceived consequence when put into practice.

Agency and narrative should be integrated at the most basic level of units; agency can be rewarded with small sections of sequential narrative; an integrated flow of narrative development (i.e. no separate levels) is possible; the use of language in the form of conversation fragments and tone of voice, greatly increases the connotation, characterization and level of meaning within the narrative [Fencott01].

2.2 Sound design and meaning in interactive virtual environments

An extension of the debate above, audio, and in particular musical forms within virtual environments, has to confront the apparent contradiction of interactivity, or agency, and a traditionally strong linear medium. Discussing the necessary process to tackle this issue, in regards to game environments, one new-media audio designer states that:

> "The sound designer should start with a macro structure in mind, containing a series of aims, concepts and themes that he wants to incorporate into the game's audio. Each scrap of audio need not necessarily work in isolation, but instead would fit into a bigger picture to create an overall impression. (Like the symphony, the whole would be greater than the sum of its parts.) At the micro level, elements of musical variety would be generated by a musical AI" [Weir00].

Mirroring the comment above in the previous section regarding narrative potential, this commentator continues by suggesting:

> "If an interactive narrative is one in which the user actively participates in weaving a story out of key fragmentary "clues", then perhaps this idea could be extended to the audio. The user could be viewed as the conductor, ordering the musical material [*being noise, voice, music, sound effect*] into something meaningful for that game-play" [Weir00].

In an attempt to illustrate this possible approach, and offer a tangential base to the following section regarding audio-augmented environments, we will continue by looking briefly at two seminal virtual environments; one from the games sector focusing heavily on sound – 'Thief' - created by Looking Glass Studios in 1998; while the other, 'Ephemere' by Char Davies, also completed in the same year, considers the generation of 'meaning' in an immersive virtual reality environment.

'Thief - the Dark Project'[5] was regarded as a significant break from its first-person role-playing contemporaries, by offering an alternative game design-concept based upon stealth, rather than fast-paced adrenaline-fuelled shooting. Described by some reviewers at the time as "more like living an exciting chapter in someone's life"[6], the game user took control of the

[5] The Thief series continued with part II – The Metal Age also by Looking Glass Studios in 2000,

[6] RPGVault Website, March 1999. http://rpgvault.ign.com/features/reviews/thief.shtml

protagonist in an action/adventure scenario of thievery, set in a grim fantasy world. Pertinent to this thesis investigation, it has been argued that sound plays a more central role in this game design concept than any other first-person game. This acclaim is the result of an ambition by the Project Director, Greg LoPiccolo, to create an aural environment where the sound both enriched the space and was an integral part of the game-play [Leonard99].

The ambition was realised in the design process in several ways. Firstly sound was the primary medium with which the AI characters communicated their location and internal state to the player and the game player:

> "Considerable work went into constructing sensory components sufficient to permit the AIs to make decisions purely based on the world as they perceive it. This allowed us to use player sounds as an integral part of game play, both as a way that players can reveal themselves inadvertently to the AIs and as a tool for players to distract or divert an AI. Moreover, AIs communicated with each other almost exclusively through sound. AI speech and sounds in the world, such as the sound of swords clashing, were assigned semantic values. In a confrontation, the player could expect nearby AIs to become alarmed by the sound of combat or cries for help, and was thus encouraged to ambush opponents as quietly as possible" [Leonard99].

For example, as it was the desired aim of the game to sneak around in the night darkness, and avoid contact with the AI characters, usually guards. Their occasional mutterings of words, snoring (if sleeping), or footsteps gave the player an awareness of their location. Likewise the action of movement, walking on different 'materials', such as metal or wood, created different volumes of noise, alerting the guards to your presence: '*I thought I saw someone there'* says the guard - the sample of sound chosen using a weighted random selection from generic themed resources, such as '*something's not right*', depending upon the state of the AI character.

This concept could operate by designing an extra 'layer' of level design, giving details about how sounds would propagate in real-time through the space, for example through windows, open/closed doors and between buildings. This meant that the sound design allowed the player and the AI's to perceive sounds more as they are in real life and have a better understanding of location within the virtual environment.

At the game-playing level, the combination of these factors made the user's experience incredibly immersive, despite what at the time were slightly sub-standard 3D graphics, and the general darkness of the vision. Hiding in shadows, the game-player projects their actions easily into the narrative of being a thief, darting from shadow to shadow, picking up on fragments of sounds heard from above, through the door etc., to decide what to do next, where to go, and essentially what the story is to be.

At a completely different level of immersion, Char Davies's 'Ephemere' project, a continuation of 'Osmose' (1995), offered a single-user to be enveloped in a virtual reality environment which explores themes of ephemerality of being within time and space, and nature as metaphor for transience:

> "Within the work are recurring archetypal elements suggesting a co-equivalency between the chthonic presences of the interior organic body and the subterranean earth, whose meanings and behaviours are dependent on the behaviour of the participant and spatial/temporal context" [Davies99].

The interface system utilises a head-mounted display, and a custom-built navigational system to allow the user, using slight body movements and breathing action, to move through the space. 30 graphical objects and 30 localised sound objects that recall mortal fleshy body of organs, blood and bone (referring not only to human but all bodies), are distributed though a spacio-temporal sequence of realms relating landscape, earth and body:

	winter >	spring >	summer >	autumn	
Landscape:	dormant >	blooming >	leafing >	falling leaves>	dust
Earth:		germinating >	fruition >	decay >	
Body:		eggs >	bones >		
	r i v e r				

[Davies99].

The immersant's experience is dependent upon the length of time spent in each zone, whether it be landscape, earth, or body, and will witness the different temporal phases through the transformation of both visual and aural content. In the case of the sound elements:

> "While the visuals pass through subtle changes of visibility and non-visibility, light and shadow and in the case of landscape, progress from the relatively literal to the abstract, the sound is also in a state of flux. Interactive and localised in three-dimensions, it flows between melodic form and mimetic effect in a state somewhere between structure and chaos, adapting moment by moment to the changing spatio-temporal context and the immersant's behaviour" [Davies99].

Furthermore there is a subtle inter-responsibility between selected elements and the user, which hold iconographic status such as seeds, rocks and river, and respond to the immersant's proximity, slow movement or continued gaze.

A constantly changing perceptual field, consisting of 3D transparent objects, means that there is a constantly fluctuating semiotic association between sounds, objects, and spatial arrangement. The participant navigates through the space according to their attraction and mood, and although it is possible to stay for the entire period in one realm, it is likely they will traverse between them interacting with concrete and abstract shapes at different stages of their development.

In describing the creative process of construction and design of Ephereme, Davies states:

> "The process has resembled the creation of a virtual opera, consisting of the development of a myriad of visual and aural elements, whose various comings and goings must be calculated in relation to each other, the progression of the work and the immersed participant, in real-time. " [Davies99].

However, without the interaction of the participant, the environment is essentially an arrangement of visual objects and sounds, evolving through a constrained passage of time. It is the individual pattern of movement of each immersant that is channelled at the end of the experience into a personal understanding, and thus the constituted meaning of the work.

2.3 LISTEN: audio augmented environments

The LISTEN project, which started in January 2001, is a research project funded by the European Commission in the context of the Information Society Technology (IST) programme[7], and is to be developed by a consortium of institutes and technology companies[8]. As noted above, LISTEN aims to "provide users with intuitive access to personalised and situated audio information spaces while they naturally explore everyday environments". Furthermore, "A new form of multi-sensory content is proposed to enhance the sensual, emotional and pedagogical impact of a broad spectrum of applications ranging from art shows to marketing or entertainment events" [Eckel01]b.

A range of features has been identified which will characterize the users experience of the proposed technology: augmentation, immersion, interaction, sauntering, and content. It would be useful to overview each of these proposed features, highlighted in [Eckel01]b to introduce the possibilities inherent within augmented environments consisting solely of sound, and the potential for narrative or information presentation.

2.3.1 Augmentation

A key difference between the audio environment proposed by LISTEN and that of a VE (either in a CAVE or screen based environment) is that the system augments an environment, which exists outside the confines of the computer apparatus. The example cited is a museum or art

[7] LISTEN - Augmenting everyday environments through interactive soundscapes, Fifth Framework Programme, Creating a user-friendly information society (IST), Contract no.: IST-1999-20646.

[8] Namely, GMD (German National Research Centre for Information Technology), IRCAM (Institute de Recherche et Coordination Acoustique/Musique), AKG Acoustics GmbH, IEMW (Vienna University of Technology), and the City of Bonn Museum of Modern Art.

gallery, whereby upon entry the visitor receives a set of light headphones that allows the visitor to access the soundscape. The sound is 'layered' within the real space, so that different sound scenarios mapped to different areas of the museum, supplying for example, environmental sound-clips, voice narrative and relevant audio accompaniment of artists' work, historical objects, and ambiguous text pieces. In this form the augmented audio would take the form of an informative guide. A different application could be sound or music as an indicator of mood – the energy of jazz, changing into an area of calm reflective piano, and could be indeed part of a complex sound concept by an artist. Different zones or situations within the space are navigated by the movement of the visitor, the changing aural environment merging with the indexes from the visual, tactile and olfactory senses. "The visit to the museum becomes an intuitive, sensual, bodily experience" [Eckel01]b.

A key issue in interpretation of the augmented environment is whether an object (or focus) is supported by the audio - the visual, tactile or olfactory sense augmented by the aural sense. Does the augmented reality present the user with an 'elaborated' space, whereby the interpretation of the subject is enhanced by the audio knowledge extension, or is it counter-pointed with poetic association? Or contrary, when the spatially positioned sound is presented exclusively on its own, does the audio information, minus subject, inspire the user to use his or her imagination to imagine or 'visualise' the subject, and its relationship with the larger environment?

It is argued that the core function of the audio augmented space is defined by either the *presence* or the *absence* of the subject, though a creative tension could be created utilising both of these forms in the same space.

2.3.2 Immersion

As the spatialised soundscape surrounds the user of the headphones, the illusion of immersion in a world of sound would be complete. The system would react to the movement of the user, emphasising the duality of being part of the environment and an independent agent within it. Until the user removes the headphones and disengages with the augmented aural sense, the user is continually immersed in the soundscape, removing any critical distance or reflection of the experience. Depending upon the nature of the headphones - semi-open or enclosing - the experience could be aurally inclusive or individual and solitary.

2.3.3 Interaction

When the user enters the motion-tracked space, they then become the interactor and the computer system activates the relevant sound sources via a databased collection of audio files, through to the wireless headphones. What is important to recognise is that the interactive audio environment would have a memory - "it registers the repetition of an action and reacts immediately with offering new audio sources" [Eckel01]b. This feature facilitates agency within the audio environment and offers the potential for narrative and personalised guidance through the space.

2.3.4 Sauntering

A further feature or interpretation of the LISTEN is that of a contemplative thought-inducing experience, comparing the bodily/aurally immersion with the experience of wandering or sauntering. The free movement and choice of time spent in a certain acoustic zone would allow you to meander through the space, succumbing to the temptation of different routes and sounds. Being an experience which effects the whole body, it

> "generates a higher sensibility for contemplative incentives, that are inscribed in the sound- and image-landscape. The inspiring sauntering makes you more sensible, affects all your senses. Playfully you receive the enrichment of stimulating new insights" [Eckel01]b.

2.3.5 Content

The elements of construction for an immersive soundscape are borrowed from the conventional model within sound design – noise or sound-effect, spoken word (voice over and dialogue), atmosphere, and music. Drawing upon the well-known formats of audio production and presentation, including radio feature, radio play, film music, and the sound textures of an interactive computer game, the LISTEN project would augment the collected sounds into a spatial environment. Depending upon the sequence of positions and directions taken by the user, an individual narrative of experience would evolve and be translated into a personalised soundscape. The interpretation of the experience would depend upon the dynamic combination of the above elements, for example descriptive tropes, such as metaphor, metonymy and synecdoche, via spoken word, poetic association of noise and sound effect, and emotions from different musical elements.

A creative application could be that poets, theatre directors, and sound artists map, into a spatial environment, words, sound, music, so that a visualisation process can be formed by the user of a word sculpture, a script with spatial interpretation, or a geography of sound. Furthermore if the soundscape referred to the mixture of both absence and presence of subject, as defined in the section above:

Artists could offer a hundred or more different interpretations of the one installation; Narrative could be a sculptor's tool, not just the preserve of the writer or film-maker; sculptures could be constructed from the matter of opinion; A collision of viewpoint and response could be an interactive process; Poets could take you on a physical journey through their thoughts; Dark corners of the mind may literally exist; Subversive voices in your ear may warp how you navigate and feel emotionally as you move through a given space; Heritage curators could present installations of imagined sites of interest, without having the exact visual artefacts, allowing the user to imagine the time and place in their mind; Artefacts become virtual environments; Memories may be mapped in geographical space, as in psychological therapy; Thoughts could be organised into an operating system.

2.4 Author and Interactor

In the previous sections, regarding interactive environments and the proposed features of LISTEN, references were made to the user of the system as the 'interactor' immersed within the environment, navigating and thus constructing an 'individual narrative'. Who is the author of this narrative? In discussing the content of the augmented audio, it was suggested that curators, artists, poets, sculptors, and theatre directors could be responsible for the content of the databased audio files, which are fed to the user's headphones in response to real-time interaction within the space. What is the difference between the user who constructs an 'individual narrative' during their experience, and the person who is responsible for the content of the medium, often regarded as the author? Stated by Murray:

> "There is a distinction between playing a creative role within an authored environment and having authorship of the environment itself. Certainly interactors can create aspects of digital stories…with the greatest degree of creative authorship being over those environments that reflect the least amount of pre-scripting. But interactors can only act within the possibilities that have been established by the writing and programming…all of the interactor's possible performances will have been called into being by the original author" [Murray97: 152].

Therefore it can be emphasized that:

> "The interactor is not the author of the digital narrative, although the interactor can experience one of the most exciting aspects of artistic creation – the thrill of exerting power over enticing and plastic materials. This is not authorship but agency" [Murray97: 153].

By creating an 'individual narrative' within the system, the user then is activating their ability of agency within the soundscape - the ability to chose where to be and so what to hear - rather than authoring the components of the narrative. The 'original author' is thus the programmer and the supplier of sound. As the construction of a virtual or augmented environment involves the use of computers, its form of authorship is deemed to be *procedural*:

> "Authorship in electronic media is procedural. Procedural authorship means writing the rules by which the texts appear as well as writing the texts themselves. It means writing the rules for the interactor's involvement, that is, the conditions under which things will happen in response to the participant's actions. It means establishing the properties of the objects and potential objects in the virtual world and the formulas for how they will relate to

> one another. The procedural author creates not just a set of scenes but a world of narrative possibilities."

> "In electronic narrative the procedural author is like a choreographer who supplies the rhythms, the context, and the set of steps that will be performed. The interactor, whether as navigator, the protagonist, explorer, or builder, makes use of this repertoire of possible steps and rhythms to improvise a particular dance among the many, many possible dances the author has enabled. We could perhaps say that the interactor is the author of a particular performance within an electronic story system, or the architect of a particular part of the virtual world, but we must distinguish this derivative authorship from originating authorship of the system" [Murray97: 152-153].

The layers of construction suggested in Murray's simile of a choreographed dance - the environmental setting, the context, the rhythms and the steps – are all different stages of planning for the experience as a whole. A procedural order, interpreted as the depth of engagement in the authoring process, is necessary not only for the elaboration of narrative components, but the management of them over a period of time.

3 Archaeological method and theory

3.1 Representations and the archaeological record

Beyond the unique spatio-temporal aspect of its data, archaeology can be approached as the understanding of the past, through the representational metaphor of an 'archaeological record'. This record is treated as the "independent and material representation of certain events and processes, and it is normally asserted that archaeology can only study those aspects of the past which are capable of material representation in this record" [Barrett00: 62].

The purpose of the record satisfies three aspects for analysis. Firstly it transcribes the material (remains) in order to produce an "archive of representations", consisting of a variety of textual comments, images and material samples. The following text is a sample of the information gathered for the context descriptions of a site, i.e. a description of the excavated ground and whatever is found within it (bag numbers are in brackets):

> "wall (1). Probably construction debris for wall (1). Phase II.
> Under (1) Over (13)
> Finds:
> 324 Bone (rodent)(13)
> Dark brown or black charcoal-flecked soil with occasional large boulders.
> Filling a bedrock hollow beneath wall (1). T. up to 25 cm. Runs most of the length of wall (1).
> Phase I.
> Under (12) (1) Same as (11) (5)
> Finds:494 Soil sample 497 Soil sample [7] 498 Soil sample [6] 501
> Soil sample [1] 503 Soil sample [4] 505 Soil sample [5] 506 Soil sample
> 531 Slag 532 Bone (rodent) 533 Stone 534 Ore 595
> Ore 596 Furnace bottom 597 Furnace lining
> (14) Dark brown charcoal-flecked soil in a cleft in bedrock to west of wall (1).
> Phase I.
> Over bedrock Under (10)
> Finds:
> 436 Hearth bottom 514 Soil sample
> (15) Dark brown soil beneath northern part of wall (6). T. c.10 cm. Phase I.
> Same as (13) Under (6) Over bedrock
> (16) Dark brown/black peaty loam with charcoal and gravel. Max. T. 20 cm.
> Filling hollow beneath wall (1). Underlain by natural yellow and grey clay.
> Phase I.
> Under (13) Over bedrock
> Finds:496 Soil sample 513 Soil sample [9] 518 Soil sample [9] 58
> Iron concretions 589 Burnt bone 590 Stone" [Campbell00].

Accuracy is of great importance as it secures the objectivity of 'exactly what is there'. Secondly, the material record represents past events and processes at the site in question, analysing the material origins and the processes that have had an effect since, although the dynamics of change are seen as too ambiguous to offer observational evidence, and so are not part of the report. Lastly, it analyses the meaning of events of the record in the terms of historical process. In other-words, the remains are inadvertently a representation of the historical process, as they are the consequences of it, and can be interpreted as the basis of indication for the activities that may have taken place at the site of record. [Barrett00: 62].

The creation of this archaeological record, and its stages of development, forms an archival process that begins, after a period of research about the site, with the excavation. Documenting, with scientific precision, modern archaeology records the spatial dimensions and stratigraphic layers of the excavation site using plans and sections, plus a detailed account of finds (artefacts) with photographs and illustrations. The following stage aims, using a relativity diagram called the 'stratigraphic sequence', or Harris matrix, and radiocarbon dating of suitable artefacts, to create a time sequence of the site, identifying 'phases'. This sequence then is subject to comparative research and analysis, of which the results are written into an excavation report, and possibly published. Usually this process ends with the deposition of the portable finds, accompanied with a published report, with a public body:

step 1. RECORDING >site notes
>sections
>plans
>the finds/artifacts*

step 2. PHASING >stratigraphic sequence
step 3. DATING >artefact analyses*

step 4. PHASING >phases/periods: illustrations & descriptions

step 5. WRITING REPORT >comparative research*
>excavation report

step 6. INTERPRETATION >museum
>published report
>archives

* related processes regarding artefacts. [Harris79: 104].

In accordance to the focus of this investigation – seeking relevant methods for application into virtual environment design – it is the steps (1-4) that represent the recording of three-dimensional data, and the relativity of time phases, that may hold clues to a suitable method. Hence the following section will proceed to present an overview of the well-known method of representation of spatial and temporal information – archaeological stratigraphy.

3.2 Archaeological Stratigraphy

Geology, and the layers of rock and soil beneath the Earth's surface, is not surprisingly the original inspiration for the recording of the excavation site:

> "All forms of stratification are the result of…cycles of erosion and deposition. Sedimentary rocks, for example, accumulate on the seabed from particles of other eroding formations and other detritus…These mud layers eventually become solid stone and may then be uplifted and themselves subject to erosion. The process of stratification is thus a cycle of erosion and accumulation, neither one without the other.
>
> On a much smaller scale this process also takes place on archaeological sites. In addition to the natural forces behind this process such as the climate or flora and fauna, the activities of people cannot be ignored. Since mankind learned to dig, it has been the major force in the process of stratification. For whatever purpose, the digging up of the earth will eventually result in the making of new strata. The process of archaeological stratification is thus an amalgam of natural patterns of erosion and deposition interlaced with human alterations of the landscape by excavation and building activities" [Harris79: 32].

Thus with a relative chronology of a site, by studying the layers of deposits and interventions made by people on the site over time, a representation and interpretation of the historical process and activity of the site may be obtained.

As an excavation of a site ultimately destroys the stratigraphic sequence by digging it up, in modern times, archaeologists have become much more sensitive to the importance of empirical evidence, than their pioneering counterparts in the first half of the 20th Century. Increasing use of technology in analysis has also meant modern archaeologists have had to develop systematic methods to record ever increasing amounts of data.

Edward Harris (1975, 1977) was one of the first archaeologists to analyze the stratigraphy of a site on a systematic basis. He proposed visualizing, diagrammatically, the relative chronology of a site by identifying all the stratigraphic relations between each deposit excavated. The subsequent document of his proposition – '*The Principles of Archaeological Stratigraphy*' [Harris79] has since been a key text within Archaeology, and the resultant diagram is known as a Harris Winchester Matrix (NB it is not a mathematical matrix). It is to this source that this paper now turns to overview the basic principles. The reader should note that the diagrams included to illustrate the principles of stratigraphy are scanned unchanged from the source to maintain their archaeological context.

There are, within the archaeological interpretation of stratigraphy, several different types of 'strata' that can be identified. Firstly, layers of material deposited or accumulated over one another horizontally, much in the nature of geology, are known as '*natural strata*'. Perpendicular features, such as pits, which have disrupted the *natural strata*, negative features that have cut away the layers, are known as a '*feature interfaces*'. In comparison, '*man-made* *layers*', as the name suggests are defined as human deposition for a determined function, such as a road, and '*upstanding strata*', which refer to strata which are consolidated in one site for a long period of time, such a walls, complicate the pattern of layers and their interpretation [Harris79: 36-8].

For purposes of this interpretation into stratigraphy, it is not necessary to delve much deeper (if you excuse the pun) into specific identification, through examples, of these different classifications. What are more relevant are the *recorded* attributes that have been applied to assist their interpretation.

3.2.1 Recording

As noted above, the recording of an excavation dig involves detailing large amounts of empirical information about a spatial area, and its different stages of excavation in depth. Written information, photographs and illustrations have traditionally been the currency of the archaeological record, although GIS and mobile computing are presently (and with increasing potential [Castlef92]&[Johnson97]) pushing the methods of the archaeological record. Regardless of the medium, there are certain basic attributes to represent, utilizing plan and section diagrams.

Up until recently, archaeologists relied upon the 'section' to understand all matters relating to the stratigraphy of a site. An archaeological section is a drawing of a vertical soil profile, as presented when cutting down through a mass of stratification. Usually all sides of the exposed surface are recorded and co-related. 'Stylized' diagrams illustrate the *boundary contours*, which define the spatial limits of each unit of stratification in both the horizontal and vertical dimensions. Hard drawn edges mark the distinction between one deposit and another, and include layer numbers, numbered as each different stratum is encountered (see figure 1).

The plan diagram, however, since the introduction of modern methods of open-area excavation, has courted shifting interest away from section diagrams. Plans are records of surfaces, not the records of cutting as section diagrams are (figure 2). Although plan diagrams can also indicate *boundary contours*, their usefulness lies in representing that which a section cannot. As many strata layers of different sizes overlap, superimposed over each other, only a part of the boundary contour will appear on the topological surface at a certain period in its development. Thus another representational indicator is used, a *surface contour*, to show the topological relief of the surface of a layer. Using a series of spot heights or elevations, these form conventional contours on a plan map only.

From the combination of dimensions offered by both the boundary and surface contours, the *volume* (and *mass*) of a stratigraphic unit may be determined if desired. Of course, this volume will include any artifacts or finds that exist within the deposits, and so therefore is the contextual reference for comparison or analysis of the artefact at a later date.

Both the stylized section and plan diagrams, by marking distinctions between the different units of strata, show the various *interfaces* between the different bodies of the strata. *Interfaces* can be interpreted as relationships "based upon the physical contact of the units of stratification and notions of relative time" [Harris79: 46]. There are several relationships that may exist between different units of stratification as illustrated in (figure 3).

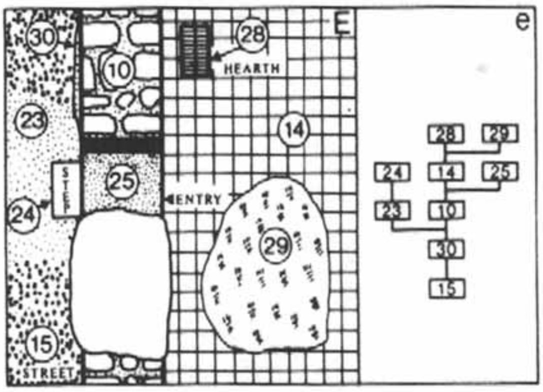


figure 1. Example of stylized section diagram (after [Harris79: 104]).

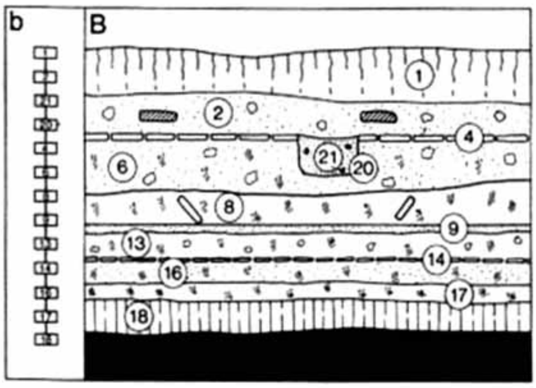


figure 2. Example of plan diagram with interfacing line and layer numbers (after [Harris79: 106]).

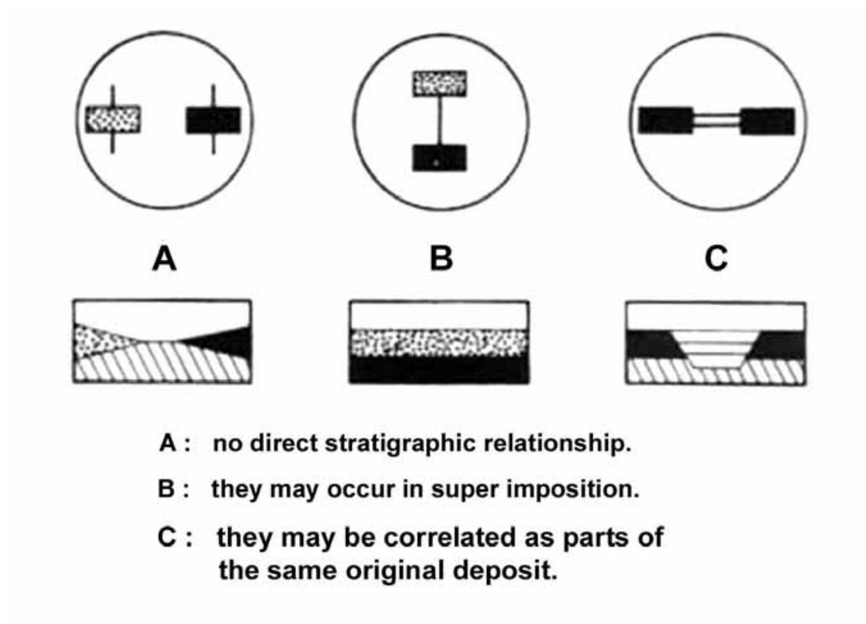


figure 3. Stratigraphic relationships (after [Harris79: 46]).

The use of computers for the recording of these relationships in recent years, has meant that the defining conditions have been developed further on a number-logic basis, extending the number of different forms to five states [Herzog90] (figure 4).

Each unit of stratification will have a position in relation to its neighbouring unit, according to the states described above, and thus a relativity or sequence can be determined assisting interpretation of the history of a site.

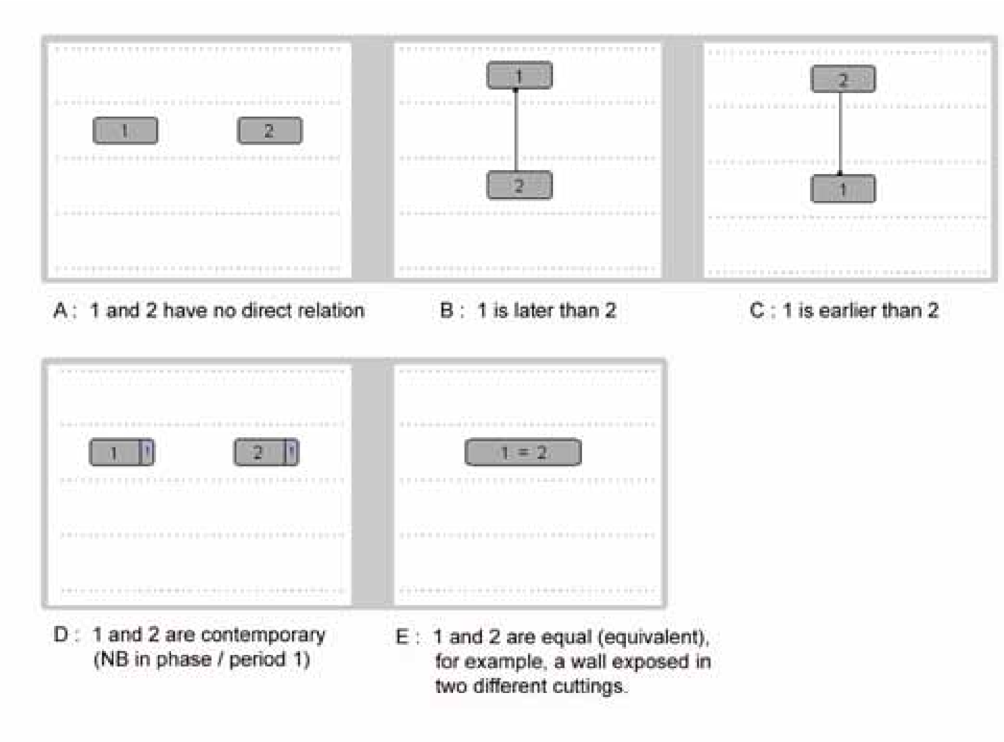


figure 4. Stratigraphic relationships (after [Herzog90]).

3.2.2 Phasing and Sequencing

The stratigraphic sequence of the site is defined by [Harris79: 86] as the sequence of deposition of strata on a site, through the course of time. Due to the complex nature and excavation of archaeological sites, the sequence represents the "translation of those physical relationships into abstract sequential relationships" [Harris: 86]. Using the rules described above, these relationships can be abstracted into a schematic diagram – the Harris Matrix.

The process of creating a Harris Matrix is illustrated in (figure 5). Initially the super-positional (earlier/later) relationships and correlations (contemporary/equality) are drawn for all the layers within the site. For example, unit 3 is positioned over units 5, 6, 7 and 9; meanwhile units 7 and 8 are connected due to their equality, as the foundation trench, unit 6, would have destroyed the gap in between. The next stage is to create a diagrammatic version of the section by combining all these physical relationships. All superfluous relationships are eliminated from the diagram using the following 'Law of Stratigraphical Succession':

> "Any given unit of archaeological stratification takes its place in the stratigraphic sequence of a site from its position between the undermost of all units which lie above it, and the uppermost of all those units which lie below it, and with which it has a physical contact; all other superpositional relationships being regarded as redundant" [Harris79: 96].

The intention of this method is that the stratigraphic sequence mirrors the process of excavation, where each stratum is removed in reverse order to that in which it was deposited. As the excavation proceeds, and each layer is removed, its number can be placed in its stratigraphic position on a growing diagram.

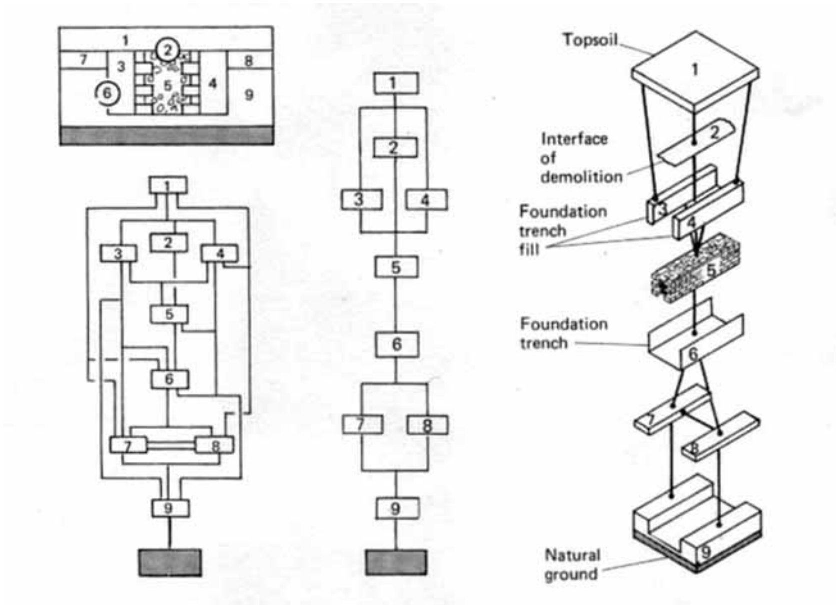


figure 5. Constructing a Harris Matrix (after [Greene83: 65]).

The division of a stratigraphic sequence into periods may take place during the excavation process or afterwards in analysis of the sequence, depending upon the relationships discovered between the deposits. The grouping of units into separate periods or phases can be made interpreting distinct differences in the building, use of the land, or that which can be identified by analysis of artifacts or dateable remains (using for example carbon dating methods). The periods can be numbered in reverse order (the first being the earliest) or named in describing fashion. For example, in the case of [Campbel00], the site phases are divided into historical periods, such as 'Middle Iron Age fort', 'Early Medieval fort', 'Extended Early Medieval fort', 'Abandonment and decay'. Further, each phase has been subdivided into three distinct states:

A – Structural (base),
B – Occupational (accumulative),
C – Infill / Residue (resultant).

Section diagrams, with the ability to show the complete history of the deposition on the site, are often used as an accompaniment to delineating the periods of a site, dotted lines indicating the perceived beginning and end of each period (see figure 6).

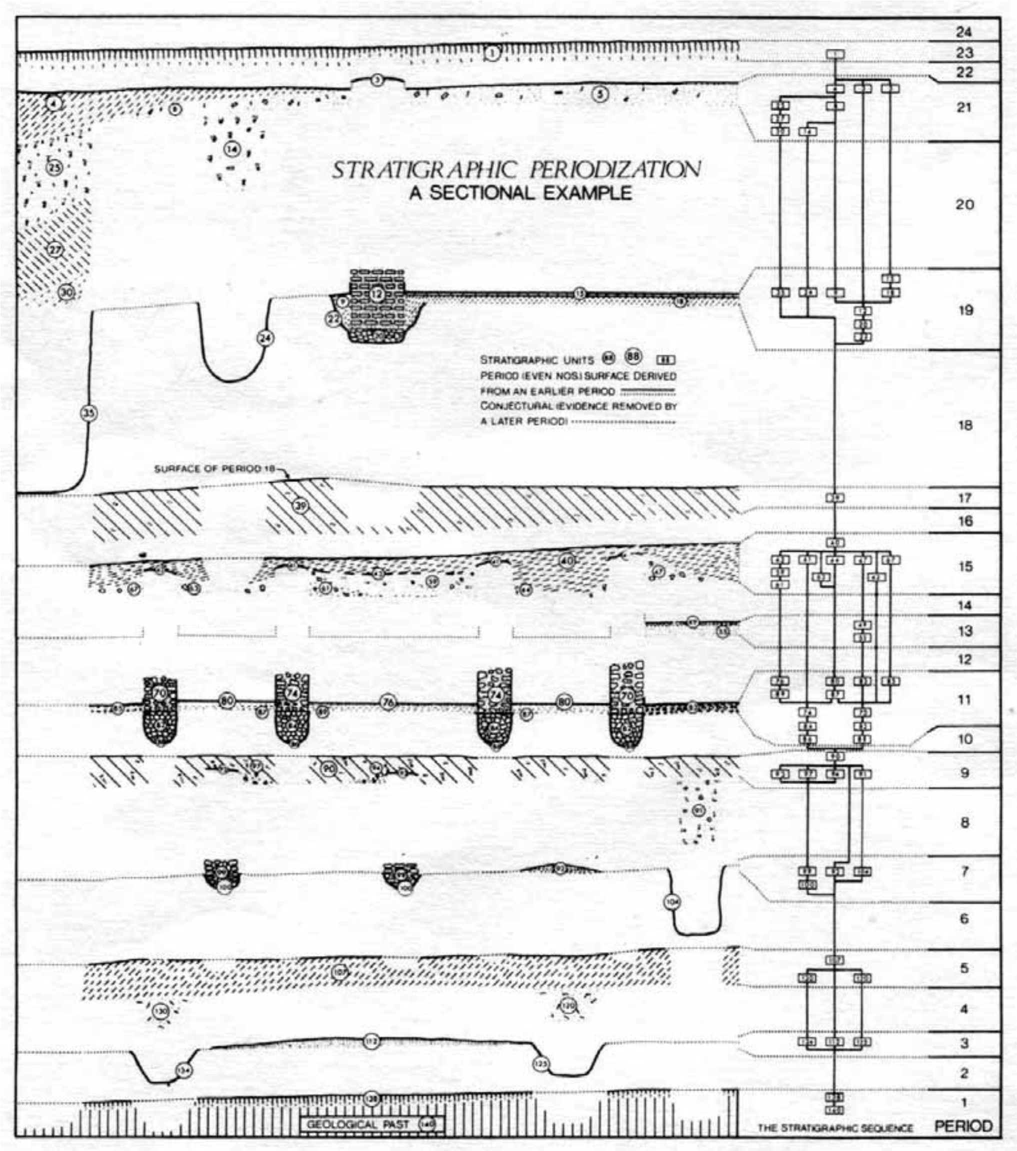


figure 6. Perceiving periods (after [Harris79: 90]).

3.2.3 Artefacts, dating and chronology

So far the emphasis of this overview in creating an archaeological record has lain in the recording of the stratigraphic deposits. Of course within these deposits are more rewarding finds – remains of both natural and human origin, which would give cultural, environmental and chronological indicators to both the sequence and topological character of the site. Artefacts are the traces of human activity and occupation from the site:

"*110*	Slag. Site 3 (31)
111	Slag. Site 3 (31)
112	Furnace lining 5g Site 3(31)
112/1	Fragment of vitrified lining with heavy vitrification on exterior. A globule of slag adheres to outer surface. Fabric C3, glossy/grey. Size 3 cm. Probably from near a tuyère. Site 3(31).
112/1	Crucible, two joining lid sherds. Form C. Fabric C2b. Knob abraded and mostly missing. Size 45 mm. Estimated D. 70 mm. Site 3(31) sieving.
112/2-7	Crucible, six bodysherds. Forms? Fabric C2a. Sizes 28, 25, 18, 18, 15, 14 mm. Site 3(31) sieving
112/8	Crucible, bodysherd. Form B. Fabric C2a. Size 27 mm. Site 3(31) sieving
112/9	Crucible rim sherd. Form ?C. Fabric C2c. Size 20 mm. Traces of lid attachment. Site 3(31)
112/10	Crucible, rim sherd. Form ? Fabric C2a. Size 20 mm. Site 3(31) sieving
113	Mould with ingate/feeder Site 3(31)
113	Mould Site 3(31) 6 fragments with keying
113	Mould Site 3(31) indet. object
113	Mould Site 3(31) 44 frags
114	Slag & iron ore. Site 3 (31) sieving
115	Flint. Site 3 (31) sieving
116	Slag. Site 3 (31) sieving
117	Bone. Site 3 (31) sieving.
118	Iron joiner's clamp. 25 x 16 x 9 mm. Site 3 (1)" [Campbell00].

Artefacts are recorded with their three dimensional position, the z-dimension placing the object at the level of its find according to a fixed datum such as sea-level. Fixed in space, the artefact is then fixed in time, by its association to the stratigraphic layer that it was found in. A single artefact or natural object found in a deposit has three different dates: a date of *origin*, i.e. when it was made, a *time range of use*, and a date in which it was *deposited* [Harris79: 97]. These dates can be calculated by a range of scientific dating methods, such as radiocarbon dating, and historical comparison with other finds.

A difficult task then follows for the archaeologist - dating the layer in which the artefact is found - by identifying indigenous finds within the layer to make comparison with other sources. It is noted that:

> "The aim of all such artefactual studies is in part to give a date to the individual layers and interfaces, phases and periods. By this means, the relative stratigraphic sequences can be tied to the chronology, in years, of human history. Without the chronological markers which are provided by their contained remains, the stratigraphic sequences of archaeological sites have little historical or cultural value" [Harris79: 98].

4 Interpretation

4.1 Applying the stratigraphical process to sound design

Now that the concept and method of archaeological stratigraphy has been introduced to the reader, it is necessary to identify which aspects of it may be useful for the authoring of virtual/augmented environments.

If we recall, the distinction between the author of an interactive environment and the interactor, it is the author, or programmer, who would initially have a general concept of the narrative or experience, creating different component parts and details. For the archaeologist, as we have seen, this process is reversed – the details are recorded, their relativity gauged and chronology investigated, resulting with an interpretation of the history of the occupied site. A narrative is supposed from the combined interpretation of fragmentary artefacts over a period of time.

The user of an interactive virtual/augmented environment acts in a similar fashion, exploring for, finding, and interpreting virtual 'artefacts' of interest, which may lead to an understanding of the experience. If the author is creating an environment for the interactor to experience, through their occupation over time, a design methodology such as the following, a reversal of the creation of an archaeological record, suggests itself as being a useful guidance:

step 1.	RESULTANT EXPERIENCE	> desired interpretation, meaning > function
step 2.	REPORT	> genre/style > summary or synopsis (history) > narrative elements > presence or absence of subject > degree of immersion
step 3.	PHASING	> defining of distinct phases/periods of development
step 4.	DATING	> identification of 'artefactual' objects which develop > historical, cultural, environmental clues to narrative
step 5.	PHASING	> arrangement of stratigraphical units into a stratigraphic > sequence of relativity (i.e. before, after, contemporary, > equal relationships)
step 6.	RECORDING	> sketching of plan for each phase, indicating boundary contours > numbering of different units. > chart the finds/artifacts in each unit.

4.1.1 Resultant experience

In the archaeological model, this stage is the conclusion for the study of an excavated site, the summary of knowledge and interpretation. For the purposes of this paper, it is the first step in the design process, and can be understood as the desired interpretation or meaning gained from immersion in the virtual/augmented environment. From a designer's point of view, what is the desired impression made upon the user – are they informed? Emotionally moved? Entertained or stimulated? Indeed the immersive experience may have a function, such as augmenting a physical space or object with information. However, if the soundscape is a sound art piece or interactive narrative, the resultant experience will be closely dependent upon the genre or style.

4.1.2 Design report

The report is understood as the overview document, defining the concepts and components of the environment. As a design report it will elaborate upon the determined genre of the audio environment, and how this relates to how it is experienced. For example, an ambient soundscape for relaxation may be smooth and gently changing with few unexpected sound objects, while a spoken-word based environment is constructed from clear changes in narrative. The soundscape may be inspired either by a poem, a film scene, an audio-walk in the city, or a history of art book. Each form transmits information in a different way, and will be interpreted, experienced, within the context of the medium. A contextual pointer or historical background could be provided for the user before immersion into the environment.

The report stage could present an experiential synopsis, identifying different narrative elements to be experienced, both linear and non-linear. It should also identify the degree of immersion into the soundscape – is it an augmented sound environment, such as LISTEN? Or is the experience placed within a graphical gaming environment? Both present different design issues and consider the relevancy of the absent/present subject. What sounds are represented in other senses, such as visual objects or smell? Where/when does the sound point to the absent subject, to be constructed in the user's imagination?

4.1.3 Phasing

In archaeological terms this meant the defining of distinct phases or periods. Distinctions in phase can be identified during the excavation process or afterwards in analysis of the stratigraphic sequence. The archaeological understanding of phases is based upon their layering in depth, as illustrated in a section diagram, and it is regarding the factor of 'depth' that the metaphor has to be modified for application in virtual or augmented environments.

When the end of one phase, and the start of another, is identified on an archaeological section diagram, essentially an occupied surface is identified, which has for whatever reasons been abandoned, built upon, or adapted, and lies below the present surface. Therefore each phase has been occupied at different depths (and so at different times) in the history of the site. The history of the site stretches over long periods of time, often hundreds and thousands of years.

The time an interactor occupies a virtual environment can operate on two levels, particularly if the environment involves a sense of narrative:

> "a relative narrative time - a consequential ordering of events - and a continuous and pervasive real time which, quite literally ticks mercilessly away at the bottom of the right hand corner of the screen" [Fencott01].

It is a design issue for the author, what type of time each phase represents - whether each indicates a distinction of *relative narrative* time or *continuous real* time. For example, each phase could represent a different chapter in narrative occupying the site, a changing soundscape due to different agency or navigation choices, or represent literally the passing of 360 seconds as the site is occupied. Ideally it would be possible to have a choice of either at disposal to assist narrative potential.

Regardless of what each phase represents, due to the interactor's continued occupation of the site, the successive phases have to be conceptualized, not as depths in layers at the site, but as successively changing overlays. A useful model of reference for conceptualizing spatio-temporal data has been developed by [Johnson97] for temporal GIS systems, in which a history of features[9] are modeled as a series of 'snapshots' on a space-time cube[10]. Imagine that the stratigraphic sequence of sound passes down through the virtual or augmented site, so that the earliest phase, and its associated sounds, is initially synchronized with the occupied site. As there is progression through the sequence, the synchronized section of the sequence disappears, conceptually downwards, so that it is replaced by the sounds associated with the following phase, and so on. The conception of layers in this manner maintains the metaphor of stratigraphy, and the usage of a Harris matrix diagram to assist the design process.

A consideration, in the application of phasing to the sound design, is to what extent the interactor can identify that there has been a change in phase, or that a sound belongs to a particular phase? The answer relates to the content application and function of the soundscape. It may be undesirable to indicate the change, allowing the sound content of the space to subtly indicate change, for example, different clusters of atmospheric noise. The

---

[9] Continuously present objects on the surface of a landscape, but with changing parameters over time.

[10] A space-time cube, has spatial dimensions situated on the x and y-axis, and time on the z-axis.

change of phase, in contrast, could explicitly play sound as an indicator for the passage of time, much in a similar way to the cut-scenes of skyline used in the computer game 'Shenmue' to mark the passing of each day to the next [Fencott01].

Generally, narrative content in virtual environments is based upon the premise that the interactor is always situated in the run-time present tense. This certainty, however, may be blurred by creative assignation of tenses to the language and spoken word, for example, using differences in the tense of description to describe the interactor's relationship to their environment.

It is open for debate whether artefacts or stratigraphical units are understood to be the closest comparative to the basic units of narrative or perceptual opportunities in virtual environments. The accumulation of relative contextual units, which form a meaningful whole or history, may indeed be the component parts of a narrative. Or contrary, like the material deposit in which a valuable artefact is found, the units may be only a setting for the narrative fragments to reside within. If this later scenario is the case, then artefacts may be the identifying historical, cultural or environmental tropes of the narrative within the site; the artefact as ambient sound that assigns reference of activity, or offer a key clue to the story.

4.1.4 Recording

Each sound, whether of strata or artefact type, has basic characteristics which would have to be determined as part of the soundscape design:

Location:	XYZ coordinate position in space.
Volume:	Simply the amplitude of the sound.
Radius:	If Omni-directional (or unobstructed by barriers) this would be the radius of a sphere. Essentially radiating from one point outwards, the radius marks the fall-out distance when the sound can no longer be heard.
Pitch:	Pitch or frequency of sound.
OneShot or *Loop*:	Play once or loop.
Phase:	Which phase of occupation does the sound exist (play)?

Furthermore, artefactual sounds may be positioned that have dateable properties, similar to the qualities of a material artefact, with an origin date (phase), a period of 'use', and date of its deposit or destruction. Although it exists within a certain phase or context, it is independent of the stratigraphic sequence. Sound 'artefacts' would have similar attributes as above, with several further that define their position in time:

StartDelay : Time since phase started that artefact appears.

Existence: Length of time that artefact exists before destroyed.

It can be presumed that both the design of virtual graphical environments and audio augmented environments, would take advantage of 3D graphical design software. Hence the author of a soundscape can map both using the plan (top) view and side/front views. The combination of these views would assist planning the radius of each sound.

However the mapping of sounds, considering each phase and borrowing methodology from archaeological stratigraphy, can be recorded on a plan diagram, showing their location at the phase beginning.

4.2 Enabling agency and an individual narrative

Allowing the interactor a sense of agency - that their actions affect and shape the environment they are present within - appears at odds with, and strains the usage of, the stratigraphical record as a useful metaphor in virtual and augmented environments. The stratigraphical metaphor fixes each distinctive (sound) component of the site into a set position to be enacted in consecutive real or relative time, essentially disregarding the unique features and potential of these new forms of media.

As noted earlier in this thesis, creating an individual narrative involves the ability to chose where to be and so what to hear. If the designed soundscape considers the number of times the interactor visits each sound position, and plays the consecutive sound, complex individual narrative situations can occur.

Therefore the positioned sounds need to have not only a radius in which the sound can be heard, but also a *collision volume*, which defines the trigger zone in which the sound plays. Two different types of trigger sound would satisfy both the active and passive listener: One that plays the subsequent sound only, unless triggered again by colliding with its collision volume again; And another which satisfies passive listening by proceeding through sounds in time while the interactor is within the collision volume. Regardless of the type of trigger, each sound would only play if its assigned phase value matches the current phase.

To further the development of agency potential, a *phase trigger*, which has its own collision volume, facilitating jumps to a specified phase, or changes the phase either up or down the sequence, could be applied to the space. This feature would allow the interactor to revisit previous stages already experienced, reverse the narrative completely, or to navigate within a non-linear narrative.

5 Application in virtual space

Although essentially this thesis is inspired by the LISTEN project, and the proposed features of audio augmented environments, the technology for augmented everyday environments is still in the early stages of prototyping, and is not available to experiment with using test design methodologies. However, the archaeological theory can be applied within a virtual environment, whereby the spatio-temporal sound design would augment the visual aspect of the space. As a testing and development space, this environment is suitable and also useful for more experimental applications of content.

A screen-based virtual environment compromises the sense of immersion, dependent upon computer-generated visuals and the quality of sound card and audio output facilities available to the running computer. However, the author has decided to use the Unreal 3D engine, which facilitates real-time 3D graphics of good real-time game standard[11]. The engine, in regards to sound, can play .WAV files (mono, 8 bit or 16 bit) and facilitates 3D positioning of sound, although at a software level this is experienced as stereo panning. By default the engine has 16 channels for sound effects and 48 for music files, however for the purposes of this application, all 64 sound channels are enabled for sound files. A key argument for using the Unreal engine above others is the distributed editor package.

5.1 Unreal Editor: A tool for authoring interactive real-time environments

The Unreal Editor (UnrealEd2) is distributed as a 'beta' version (updated by patches[12]) with Unreal Tournament, the popular PC first-person-shooter game by Epic Games. Due to its semi-official release status, there are no manuals or help files for this program, although like other games of similar genre (e.g. Quake) there is an extremely lively ‘mod’ (~ification) scene on the Internet, with many amateur or semi-professional websites offering tutorials and scripting tips.

Although many of the features of the editor program are obviously geared towards designing and constructing game levels for the ‘death-match’ sequences of a 'shoot-em-up' first-person-view tournament, it is possible to import your own models, textures and sounds, render atmospheric lighting, and crucially create custom features, using the engine's own scripting language - UnrealScript, an object-orientated code, similar to a combination of Java and C++.

It is the aforementioned adaptability, flexibility and future prospects of development, which is promoting the Unreal editor to become a popular research tool for content design, semantics,

---

[11] The 3D graphics engine is licensed by Epic games, and is used in their Unreal Tournament and Deus Ex games titles. For further information, http://unreal.epicgames.com

behaviour and psychology within virtual environments. Naturally there are certain limitations of using the Unreal engine: The lack of a stand-alone 'player' for the environments means that it can only run on a PC, and one in which a copy of the 'Unreal Tournament' game is installed. Also a large amount of modification is necessary to eliminate the 'first-person-shooter' context of both the pre-game user interface, and certain aspects of the game-play.

Regardless of the limitations, the decision to use Unreal Editor reflects the focus of this thesis – towards the creative author of the soundscape - as designing and constructing the virtual environment within Unreal Editor involves using a visual, not programming, interface.

The graphical user interface is similar to other 3D modelling packages, offering top, side, front and perspective view-ports (see figure 7). It operates with a method of using 3D ‘brushes’ of basic geometric shapes and imported models to subtract the ‘navigational space’ from the ‘solid’ grid. In this manner the virtual space is first scooped out as a negative volume and is then added to create complexity and variable interior architectural space. 'Actors', for example triggers, navigation pointers or lights are selected from the Actor Class list and added to the constructed space, placed at a 3D position in one of the view-port windows, and their properties edited (see figure 8). The combination of these actors, define the agency, content, and features of the virtual environment.

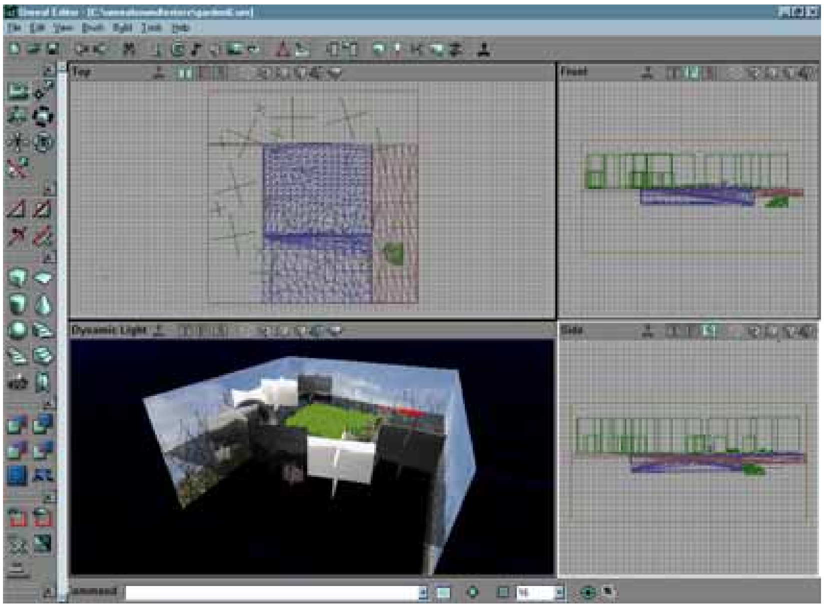


figure 7. Screenshot - Unreal Editor graphical windows.

[12] Author has used patch 436 update.

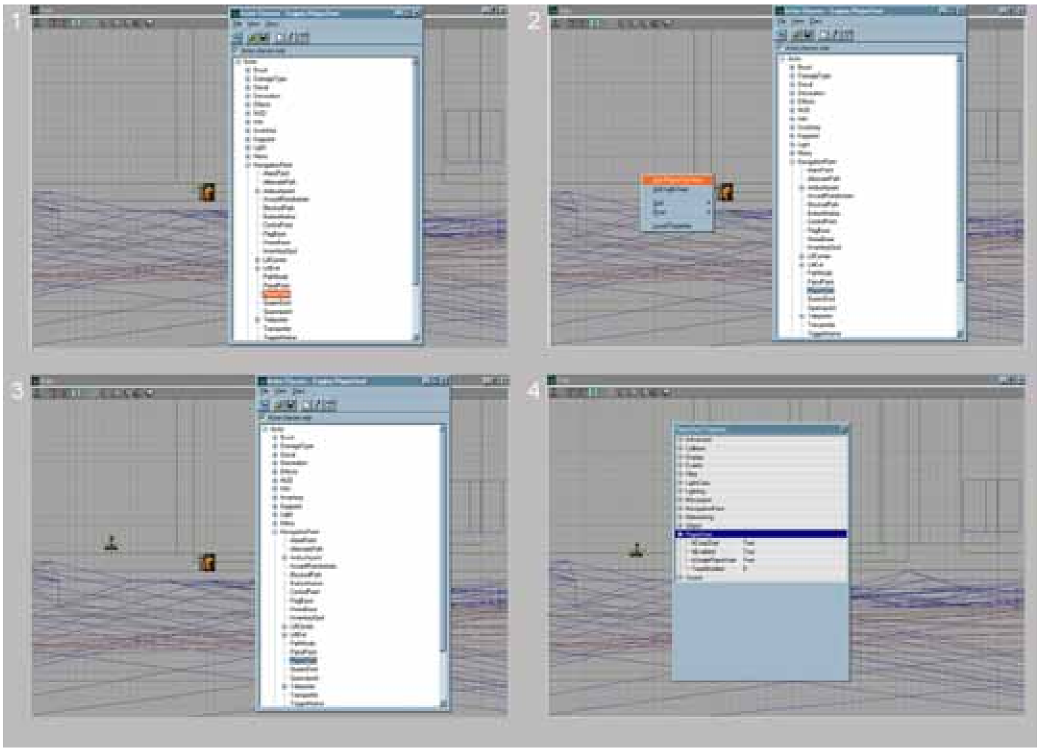


figure 8. (1) Selecting an Actor, (2) & (3) Adding it to a position in the side -view,
(4) Displaying the Actor's properties.

5.2 Scripting archaeological sounds

The critical success factors for applying the archaeological approach discussed previously are associated with an actor that allows flexible recording (placement and assignation) of sound files in the 3D environment. The engineering of an object-oriented class should engage with the characteristics highlighted in [5.1.4], namely *volume*, *sound radius*, *pitch*, its type – does it play once (*one shot*) or *loop*, and *phase* of occupation that the sound exists (plays) within.

It is important that the actor manages both *consecutive* and *consequential* dynamism, in other-words linear sequencing of sounds, and non-linear interactivity of change. To allow interactivity, and the sensation that the interactor is experiencing the sound as a result of their movement, or continued presence in a certain position, there has to be two different types of sequence. Firstly, in the conventional sense, it is necessary to have a linear progression through a sound 'history' of a place. This can be understood, both by the author and the interactor, as time 'rolling over' or continuing in that certain position. Secondly, it is useful to have a sound type, which rewards the interactivity of movement within the space by the user, and is adaptable regarding repeated visits to the same position. For example by playing different sounds each time it is encountered. Regardless of the type, each 'archaeological' sound actor must have a collision detection scheme defining when (and where) the sound will be heard.

As each sound is assigned to exist in a certain phase of development or narrative, it is critical that the sounds are checked against the value of the current phase, and that there is the opportunity to flexibly trigger the phase onto the next, either linearly or non-linearly, depending upon the content. Alternatively a further option may be to ascribe time lengths to each phase, and that it changes uncontrollably.

Essentially from the perspective of the author and the design process, the 'archaeological' sound actor should offer the maximum flexibility for scripting active and passive components of the narrative of the site, and openly allow the assignment of all these factors within Unreal Editor's 'Actor properties' dialogue.

5.2.1 Existing actors, methods and structures using UnrealScript

Unreal Editor contains several Actors and features that handle sound. On almost all occasions where sound effects are played during scripting and run-time, the Unreal Script function `PlaySound` is used:

```
// Sound slots for actors.
enum ESoundSlot
{
    SLOT_None,
    SLOT_Misc,
    SLOT_Pain,
    SLOT_Interact,
    SLOT_Ambient,
    SLOT_Talk,
    SLOT_Interface,
};

// Play a sound effect.
native(264) final function PlaySound
(
    sound    Sound,
    optional ESoundSlot Slot,
    optional float  Volume,
    optional bool   bNoOverride,
    optional float  Radius,
    optional float  Pitch
);
```

[Quoted from the 'Actor' class] [13]

The variable called `ESoundSlot` can be ascribed by the designer to the sound, indicating which type of sound is being played. Each actor has eight sound slots, and so can play up to 8 simultaneous (overlapping) sounds. By associating each sound a slot, this allows the games engine to decide if the sound is cut off and overridden (NB/ the Boolean variable `bNoOverride` setting). `SLOT_None` is a special setting, which indicates that the sound should not be interrupt any other sounds, and allows many sounds to overlap each other [Sween99]a.

[13] Unreal Engine, © Epic Games.

Furthermore, common to each actor is its ability to play an 'Ambient Sound', either 'OneShot' or 'Looping' at its location in 3D space. The ambient sound has certain editable parameters: volume (amplitude), pitch and radius within the editor interface, declared in the 'Actor' class as the following:

```
var(Sound) sound  AmbientSound;  // Ambient sound file.
var(Sound) byte   SoundRadius;   // Radius of ambient sound (0 - 255).
var(Sound) byte   SoundVolume;   // Volume of ambient sound (0 - 255).
var(Sound) byte   SoundPitch;    // Sound pitch shift, 64.0=none.
```

[Quoted from the 'Actor' class] [14]

There are no code examples of how the ambient sound plays, and as it uses a different type of variable (i.e. bytes) from the PlaySound function, it can be guessed that it operates at different level within the games engine.

Two actors which extend from the keypoint class, DynamicAmbientSound and TriggeredAmbientSound offer the nearest comparison to the aims of this thesis, and assisted consideration of relevant scripting features and structure.

DynamicAmbientSound allows up to 16 sounds, assigned in the UnrealEd actor properties window, to be held in an array, which are played at random during run-time. There are two functions that operate this class, one which starts at the beginning of the game play, BeginPlay, that checks the number of sounds entered into the array by the author, and via a random number generator chooses one to play using a basic PlaySound function call. The function then assigns the array number to a variable entitled lastSound, as a memory of what was played last, and calls the Timer function with randomised delay added. Various checks are then made within the Timer function as to what sound is consequently played:

```
i = Rand(numSounds); // Which sound should be played?
while( i == lastSound && bDontRepeat && numSounds > 1 )
        i = Rand(numSounds);

PlaySound (Sounds[i]);
lastSound = i;
```

[Quoted from 'DynamicAmbientSound' class] [15]

TriggeredAmbientSound, meanwhile, replays a sound if triggered via collision with an associated, but separate trigger. Using also the PlaySound function, it allows the author to ascribe a sound to the actor, and uses the variables of the AmbientSound property:

```
if (bInitiallyOn) {
    PlaySound (AmbSound,, Float(SoundVolume)/128,, Float(SoundRadius)*25);
    bIsOn = true;
```

[Quoted from 'TriggeredAmbientSound' class] [16]

---

[14] Unreal Engine, © Epic Games.
[15] Unreal Engine, © Epic Games.

Of note, this actor uses different states to determine its function, such as `TriggerToggle`, `OnWhileTriggered`. In the first state, the sound is turned off if triggered, and turned on when triggered again. While In the `OnWhileTriggered` state, the instigator must stay within the trigger's radius for the sound effect to continue playing:

```
state() OnWhileTriggered
{
    function Trigger( actor Other, pawn EventInstigator )
    {
        bIsOn = true;
        bPlayedOnce = false;
    }
    function UnTrigger( actor Other, pawn EventInstigator )
    {
        bIsOn = false;
    }
Begin:
    bIsOn = false;
}
```

[Quoted from 'TriggeredAmbientSound' class] [17]

Using brackets after the state declaration, i.e. `state()`, means that it is editable in Unreal Editor, and so possible for the author to choose the state of the actor in the design process.

However both of these examples offer no opportunity to manage a sequence of sounds. `DynamicAmbientSound` operates on the premise of randomness and allows no characteristics to be applied to the sound; while `TriggeredAmbientSound` offers agency to the interactor, but needs a separate trigger to be associated with the actor to play it, and then only allows one sound file to be attached to it.

5.2.2 ArchaeSoundTrigger

The following class proposes an object-oriented solution to allow the management of consecutive and consequential development of sound, and conceptually is based upon the startigraphical unit. It is constructed with scripting inspiration from the combination of elements featured in the previous section, and the interactive characteristics of the `Trigger` class. The following section will elaborate upon aspects of the class, although an unbroken listing of the code can be found in [Appendix A].

The basic premise of management of 'history' in the class, is based upon an array list of 16 sound data bundles, whereby '0' represents the earliest and '15' the latest[18]. This 'rack' of

---

[16] Unreal Engine, © Epic Games.
[17] Unreal Engine, © Epic Games.
[18] The number of sounds can theoretically be increased, depending upon the desired division of the soundscape into sequential components, although this author decided that it became

numbers in the Actor properties window (figure 9) is, admittedly, a reversal of the archaeological convention discussed earlier, whereby the earliest strata is visually and diagrammatically at the bottom of the pile. However, unless a customised interface is made also, it is an idiosyncrasy that will have to be forgiven and accounted for.

There is in fact two different arrays - `SRelate` and `SSound` - constructed using the `StructSound` and `Relativity` structures. Each number in the array therefore consists of:

```
    struct StructSound
    {
        var() sound sound;  // sound sample.
        var() byte  volume; // between 0 and 255. 'byte' descriptor used due
to
                            // easy sliding interface in editor.
        var() float radius; // radius of heard sound. 'float' descriptor used
                            // to maintain continuity with collision radius
        var() float pitch;  // 0.5 = half pitch, 1.0 = normal etc.
        var() bool  bLoop;  // looping sound or one shot.
    };

    struct Relativity
    {
        var() int   phase; //  phase in which the sound exists (plays)
        var() float pause; //  length of time (secs) between samples,
                           //  (used in RollOver state).
    };

    var() Relativity SRelate[16]; // Which phase & pause each sound is
    associated
    var() StructSound SSound[16]; // What to play (must be at least one sound)
```

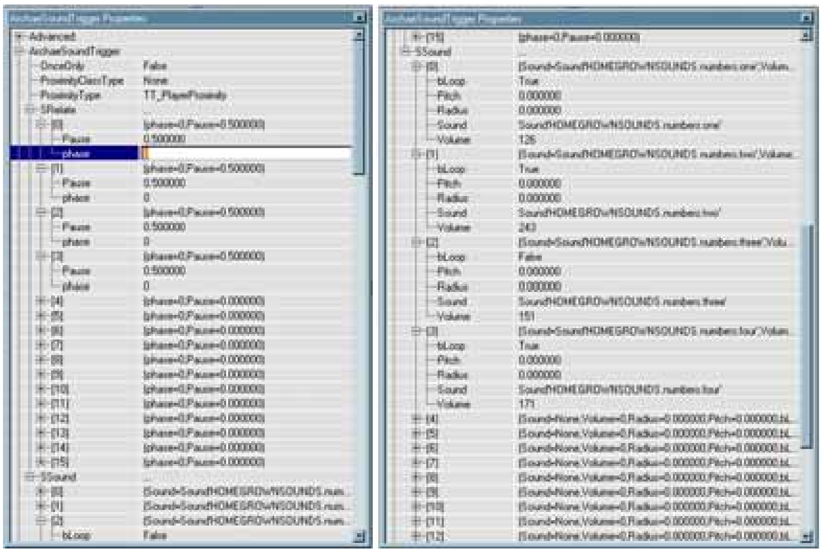


figure 9. Sound Data Arrays in 'ArchaeSoundTrigger' class, showing the two screenshots of SRelate and SSound.

---

increasingly difficult to scroll through the java interface in the Unreal Editor, the more divisions are made.

The 'bundled' information has been split up for two reasons - firstly for ease of comparison in the actor properties, and secondly later [see 6.2.3], when it becomes necessary to access the information 'remotely' from another actor, a structure containing all the attributes is too large for access. However the numbers refer explicitly to each other and it should be understood that each number of the array is synchronous with that in the other array.

The following variables all handle time relevant and repetition issues:

```
    var() bool OnceOnly; // Flag to indicate if there is an iteration through
the
                         // list of sound samples once or whether it plays
                         // through again.
    var bool bPlayedOnce; // Flag to indicate if sound list has played once.
    var bool bRollOn;     // Flag to indicate if Rollover state, i.e. the
sounds
                          //progress through the list when triggered.
```

There are two variables, `curSound` and `PhaseNow`, which represent the present tense, and are the variables that can be changed (iterated in either direction) to move through the sound data list:

```
    var int curSound;  // Which sound is to be played currently?
    var int PhaseNow;  // Phase updated incrementally.
    var int numSounds; // number of sounds available in the sound data list.
```

Literally acting as a limit to the sequence potential, the `numSounds` variable also indicates the limit of phases possible, presuming for the sake of ease that there are not more phases than successive sounds in the data-list.

Upon run-time, the `BeginPlay` function checks the number of sounds that have been entered in the authoring and design process with Unreal Editor, and initialises both `curSound` and `PhaseNow` to zero, the earliest position in the sequence.

Both states `PlaySoundRollOver` and `PlaySoundOneShot` operate in almost identical fashion, other than the assignment of the flag value `bRollOn` to either true or false. `PlaySoundRollOver` satisfies the linear passing (and presumption) of time through the sound data list. The main focus of both states is the '`if`' check within the `Touch` function that establishes the relevancy of not only the collision detection, but the current phase. If successful, the `Timer` function is called, after the sound's associated `pause`:

```
    state() PlaySoundRollover
    {   // Called when something enters the collision radius.
        function Touch( actor Other )
        {
            if ( (IsRelevant( Other ) &&
            // IF is the actor who collides with the collision radius the
            // correct trigger? defined in Editor as ProximityType
```

```
            (SRelate[curSound].phase == PhaseNow) &&
            // AND the 'present-tense' phase the same as current sound's
            phase?
            !OnceOnly)
            // AND it is not a OnceOnly play
            || // OR
            (IsRelevant( Other ) &&
            (SRelate[curSound].phase == PhaseNow) &&
            OnceOnly &&
            // AND it IS a OnceOnly play
            !bPlayedOnce) ) // but hasn't been played yet..THEN its ok.
            {
                bRollOn = true;
                if (SRelate[curSound].pause == 0)
                // if no value is entered default pause is 0.5 secs.
                    SRelate[curSound].pause = 0.5;

                SetTimer(SRelate[curSound].pause, False);
                // pauses and calls Timer function.
            }
        }
    }
```

Within the `SRelate` array, each sound data bundle has an associated `pause` value that can be utilised within this state as a mediator of pace, and temporal positioning of the sound. For example, if all 16 sounds spaces are filled, it may be assigned in many different temporal combinations: each sound existing in a different phase; a combination of groups in different clusters within the 16; or every one of the 16 sounds existing in the fifth phase.

After checking the current sound is at the end of the sequence and if it should repeat over, the `Timer` function has to consider two different scenarios - does the current sound loop itself or play only once? This feature has been added to allow ambient atmospheric sounds that suit repeated or longer listening to be layered within the spatio-temporal sequence with voice narration or effect sounds. As only the `AmbientSound` property in the Unreal Engine allows looping sound files, if the current sound's `bLoop` variable holds `true`, it is necessary to ascribe the current sound's characteristics (i.e. sound file, volume, radius and pitch) to the `AmbientSound`'s variables, maintaining continuity between different data types. To allow a developed option of using the one (looping) `AmbientSound` type as background to shorter, one-shot sounds, the `PlaySound` function is used when `SSound[curSound].bLoop == true`:

```
        if ( SRelate[curSound].phase != PhaseNow ) return;
        // if current sound's phase is no longer matching the phase – don't
        // play  any more sounds until they match again
        ….
        if (SSound[curSound].bLoop == false) // hence sound is 'OneShot'
        {
            AmbientSound = None;// presently empties any previously held sound
            // if no entry for volume is set to default 2.0
            if (SSound[curSound].volume == 0) Fvol = 2.0;
            else
                Fvol = float (SSound[curSound].volume)*2.0;

            // if no entry for radius, set to default of Actor's Collision
```

```
            // Radius

            if (SSound[curSound].radius == 0) Frad = CollisionRadius;
            else
                Frad = SSound[curSound].radius;
            // if no entry for pitch is set to default 1.0
            if (SSound[curSound].pitch == 0) Fpitch = 1.0;
            else
                FPitch = SSound[curSound].pitch;

        PlaySound (SSound[curSound].sound, SLOT_None, Fvol,, Frad, Fpitch);
        }
        curSound++;
        // After sound is played the current sound no. progressively increases
        // to access a new sound data array

        if ( (curSound == numSounds) && !OnceOnly ) curSound = 0;
        // if sound data list has reached end, and actor is not set to play
        Once // only next sound played will be at beginning

        if( (curSound == numSounds) && OnceOnly ) bPlayedOnce = true;
        // if sound data list has reach end, and flag OnceOnly is true then

        if ( bRollOn == true )
        // Otherwise check if PlaySoundRollOver flag is true.
        // If it is then do:
           SetTimer(SRelate[curSound].pause, False);
           //repeat the timer function with new current sound values.
```

What is important to understand from this code, is that the only value that is changing from that provided by the author in the UnrealEditor environment, is the `curSound` indicator, which opens access to the information held in the data array. To remind the reader of the metaphor used earlier in this text - that of the present-tense surface overlaying the occupied site, and its invisible replacement by movement downwards below the surface - `curSound` acts as the invisible surface always re-manoeuvring its position against the present-tense `PhaseNow`, the occupied site.

5.2.3 PhaseChange

To allow agency potential within the relative playing of sounds in the environment, a class called `PhaseChange` has been constructed that extends the `Trigger` actor class [for full listing of code consult Appendix A]. This actor has the characteristics of a trigger, using the `CollisionRadius` value as an indicator of the trigger zone, and similar to the `ArchaeSoundTrigger` checks the relevancy of the colliding Pawn or `Actor`. However the `PhaseChange` class offers a combination of changeable attributes to offer flexibility in its usage to change the current phase direction or narrative development. These attributes can be edited in the actor's properties dialogue, influencing the type of phase change and to what effect - either selected actors or all (see figure 11):

```
var() enum ERelative
{
    PD_Forward,  // Phase is incremented ++
```

```
        PD_Backward, // Phase is incremented --
        PD_JumpTo,   // Jump to phase selected

    } PhaseDirection;

    var() enum EPhaseChange
    {
        PC_TriggerSelect, // Tag and Event have to be the same for phase
    change
        PC_TriggerAll,    // Effects all 'ArchaeSoundTriggers' actors

    } PhaseChange;

    var() enum ELife
    {
        PL_Eternal, // Triggers for all time
        PL_ExistsInPhase, // Triggers only in ascribed 'ExistsInPhase' number

    } PhaseLife;

    var() int JumpTo;        // if JumpTo state, then where it will jump to
    var() int ExistsInPhase; // Which phase will the trigger exist?
    var() int MaxPhase;      // Maximum phase number is added
```

The class operates using the `Touch` function, checking if the touching actor is relevant, and if so, depending upon `Elife` type, that it exists in the present phase. The `Timer` function is called when true. A short delay of 0.5 seconds is set before the function operates, to allow for inconsistencies in the phase change that may take place as the interactor moves over the collision radius.

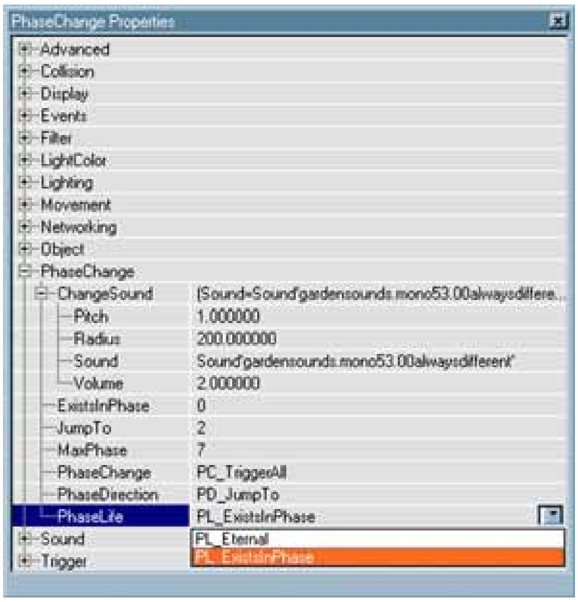


figure 10: `PhaseChange` Properties, highlighting `ExistsInPhase` state.

The actor essentially fulfills any change phase by accessing the `PhaseNow` variable in each `ArchaeSoundTrigger` and `APhaseNow` in `Artefact` classes respectively, changing the value within. This method has been used due to the object-orientated nature of the scripting language, and lack of ANSCII C-style global variables that would indicate the current Phase value. UnrealScript uses a command called `foreach` to allow the management and search of large numbers of actors in the environment, and in this case a further assignment called `AllActors (class BaseClass, out actor Actor, optional name MatchTag)` [Sween99]b, which iterates through all actors in the level. The second parameter in the `foreach` command is a variable assigned an actor on each iteration through the `foreach` loop. As the `PC_TriggerAll` and `PC_TriggerSelect` switch operates with little difference other than the `Event` and `Tag` check, the following code from `PhaseChange` class illustrates this operation:

```
function timer()
{
    local ArchaeSoundTrigger SS;
    //Sets up local ArchaeSoundTrigger variable called SS
    local Artefact AA;  // Sets up local Artefact class  called AA
    //These will be used within the 'foreach' loops as local 'version' of
    //found actors during iteration
    local int i;
    bChange = false; // initialises Flag to false before check
                    // NB/ this is checked at end of 'Timer' function
    switch ( PhaseChange )
    {
        case PC_TriggerSelect:

        if ( Event != '' ) // if event is not empty i.e. if any to change
        {
            foreach AllActors( class 'ArchaeSoundTrigger', SS )
            // for each iteration through every found ArchaeSoundTrigger
            // actor  in the level, assign its data temporarily to local
            // actor SS
            {
                if ( SS.Tag == Event )
                // if tag of actor is same as PhaseChange Event name then:
                {
                    if ( (PhaseDirection == PD_Forward) &&
                    // AND if  actor's phase is currently one below limit
                         (SS.PhaseNow < (MaxPhase-1) )
                        )
                    {
                        SS.PhaseNow++; // increase actor's current phase
                        PCPresent++; // increase internal current phase
                        bChange = true;  // change has happened
                        for (i = 0; i < SS.numSounds; i++)
                        {
                            if (SS.SRelate[i].phase == SS.PhaseNow)
                            {
                                SS.curSound = i;
                                break;
                            } // Above: checks through the actor's sound
                             // datalist for new starting position in list
                        }
                    } // end if
                } // end if tag
            } // end foreach
        } // end if Event
    } // end Switch
} // end function Timer
```

In accordance to the archaeological metaphor, `Artefact` actors may only exist with the existence of `ArchaeSoundTrigger` actors, and so the `APhaseNow` value attached to each `Artefact` actor is altered, if a change has been made in the `ArchaeSoundTrigger` iteration. The program procedure operates in the similar fashion as above, although it is not necessary to find the next sound data list position, as `Artefacts` exist independently of the sequence:

```
if (bChange)
{
    // Adjust the PhaseNow value on each Artefact actor
    foreach AllActors(class 'Artefact', AA)
    {
        if (AA.Tag == Event)
        {
            if (PhaseDirection == PD_Backward)  AA.APhaseNow--;
            if (PhaseDirection == PD_Forward)  AA.APhaseNow++;
            if (PhaseDirection == PD_JumpTo) AA.APhaseNow = JumpTo;
        }
    }
}
```

An issue of section [5.1.3] was to what extent the interactor can identify that there has been a change in phase, or that a sound belongs to a particular phase? During the design process, within the Unreal Editor environment, the author can assign a sound which will be played when the `PhaseChange` actor is triggered, and can be characterised by the same attributes as seen above, using the `StructSound` data structure:

```
var() StructSound ChangeSound;
```

At the end of the `Timer` function the following check is made; the `PlayChangeSound` function is called if a sound has been specified and an actual phase change has been made (Note that the sound is OneShot only):

```
if ( (ChangeSound.sound != None) && (bChange == true) )
    PlayChangeSound();

function PlayChangeSound()
{
    local float Fvol;
    local float Frad;
    local float Fpitch;

    // if no entry for volume is set to default
    if (ChangeSound.volume == 0) Fvol = 1.0;
    else
        Fvol = float (ChangeSound.volume);

    // if no entry for radius, set to default of Actor's Collision Radius
    if (ChangeSound.radius == 0) Frad = CollisionRadius;
    else
        Frad = ChangeSound.radius;

    // if no entry for pitch, set to default 1.0
    if (ChangeSound.pitch == 0) Fpitch = 1.0;
    else
```

```
            Fpitch = ChangeSound.pitch;

        PlaySound (ChangeSound.sound, SLOT_Misc, Fvol,, Frad, Fpitch);
    }
```

5.2.4 Artefact

Due to the debatable aspect of the role of the sound artefact within a soundscape composed of 'stratigraphic units', the construction of this class aimed to offer an alternative to the sequential features of the `ArchaeaSoundTrigger` actor, and reflect the temporal nature of an archaeological artefact's existence. As discovered in [3.2.3], an artefact has an origin date, a period of usage and after, a time of deposit. By approaching the ambiguous nature of sound-as-artefact, assigning these key attributes links the sound's existence firmly in a present tense of agency - the time when it can be heard, and then no longer.

In terms of an object-oriented class, and within the framework of phases already constructed, the existence of the sound artefact, i.e. when it can be heard, is dependent upon the question – does it exist within the current phase? A further issue is, how long can the sound be heard for?

The class allows the author the choice of design options for the sound in the actor properties dialogue, defining either a temporal or constant nature within a phase, or an overriding timeless one:

```
    var() enum ELife
    {
        AL_Eternal,  // Exists for all time
        AL_Constant,  // Exists for entire occupation of phase
        AL_Temporal, // Plays in relevant phase only for number of seconds
                     // ascribed in InExistenceTime variable, or if none
                     // entered, the length of the sound sample
    } ArteLife;


    var() int OrigInPhase; // Originates in phase
    var() float Delay;     // delay after phase begins that artefact exists
    var() float InExistenceTime; // length of time artefact is in existence
```

A `BeginPlay()` function at the start of game play initialises the `APhaseNow` variable to zero, the starting phase. If the reader recalls, this is the value that is altered by any `PhaseChange` actor. Of note, the `GetSoundDuration()` function is utilised to find a value for the `InExistenceTime` variable if the author has not included a desired length of time that the sound may be heard:

```
    if ((ASound.bLoop == false) && (InExistenceTime == 0))
        InExistenceTime = GetSoundDuration( ASound.sound );
```

At the time of writing this class [for full code listing see appendix A] operates in a similar fashion to the above examples, as an extension of the `Triggers` class. Although the intention was to create a class which leaned more towards temporality than interactivity, initial tests did not yield a successful 'ambient sound' style actor. Therefore the class functions operate in a similar fashion to the `ArcheaSoundTrigger` and the `PhaseChange` actors: if the interactor triggers a collision radius, and the ascribed phase of existence matches the current phase, the time function is called which plays the sound after a delay. As follows:

```
function Touch( actor Other )
{
    if( ( (IsRelevant( Other )) &&
        (OrigInPhase == APhaseNow) ) // correct phase for playing?
        || // OR
        ( (IsRelevant( Other )) &&
          (ArteLife == AL_Eternal) ) // Eternal option chosen
      )
    {
        if ((ArteLife == AL_Temporal) && (Playing == false))
            LifeSpan = InExistenceTime + Delay;
            // Lifespan is a seconds countdown before deposit
        SetTimer(Delay, false);
    }
}
```

Particular to this class is the use of the `LifeSpan` variable, to give the existence of the actor a life span in seconds before it is destroyed, thus taking no further part in future visits to that location.

5.3 Assisting icons

A complex soundscape, designed in Unreal Editor, would involve many actors placed around the environment, and in mind of the cluttering accumulation of brush lines and icons on the orthographic design windows, prove to be a complicated design space. This is particularly the case when one icon texture would represent actors, which have much flexibility, and therefore function in different ways. In the case of `ArchaeSoundTrigger` and `Artefact`, each actor has specific operational existence, i.e. they will only play sound if the current phase matches the current sound to be played on the sound data list, while the `PhaseChange` actor, can either increment upwards or downwards, or jump to a phase number between 0-15.

It would be useful for the author to have an understanding of the relative and interactive behaviour of each actor, assisting the design of spacio-temporal relationships between them. The Actor properties interface in Unreal Editor, extending `display`, allows assignment of a texture to the `Texture` variable for visibility purposes in the editor and, occasionally, runtime purposes. To assist the understanding of complicated arrangements of 'archaeological-style'

Sounds, a series of icon textures have been designed to indicate type of actor, starting phase of existence, and state of actor, replacing the default texture (see figure 11).

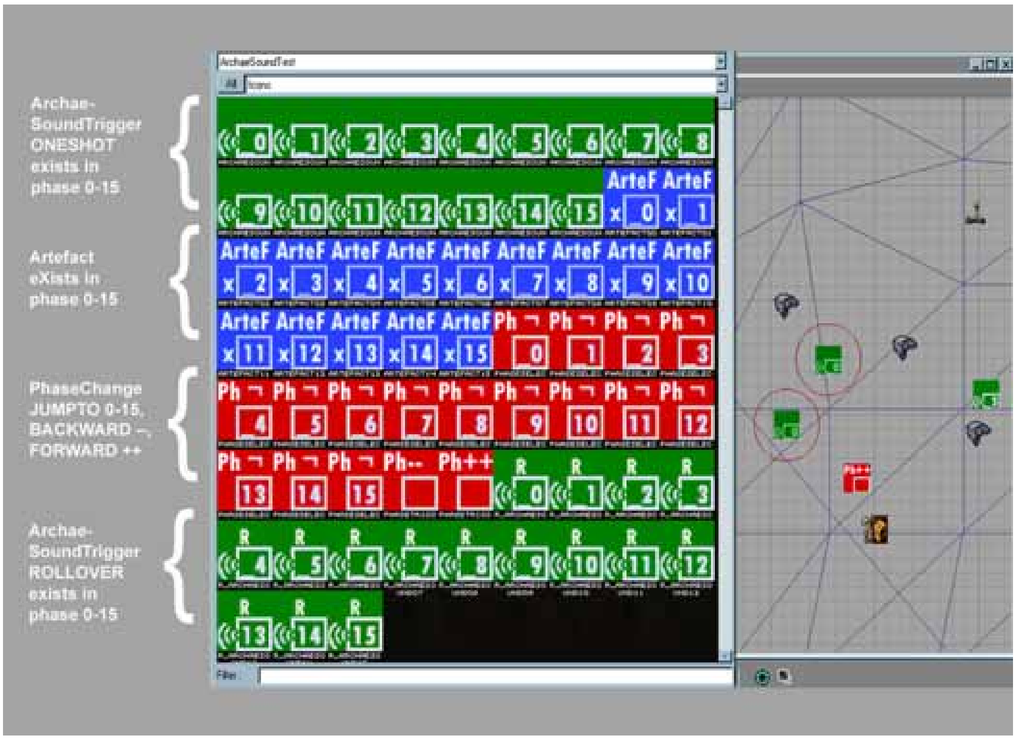


figure 11: Design and application of icons, to denote different states or Phase of existence.

6 Design process and production analysis example: 'Garden monologue'

Now that the tools and components have been examined and constructed as necessary for the authoring of 'archaeological' sound in virtual environments, the following section highlights a creative utilisation of the custom-scripted classes in Unreal Editor entitled 'Garden Monologue'.

The virtual environment design follows the proposed design methodology introduced in section [5], and utilises the freely distributed Harris Matrix generating program ArchEd 1.0[19] for the creation of the stratigraphic sequences. The aim is to put the methodology into real practice, to analyse its effectiveness and flexibility for application, its limitations, and elaborate upon unconsidered problems relevant specifically to the creative process of virtual environment design.

6.1 Prelude

It is very difficult to imagine constructing a soundscape with no concept of the space it is to occupy. Although this may seem an obvious point, in the first instance, the construction of 'archaeological' layers' of sound involves much choice of content, recording, and processing of sound, before it is necessarily placed within spatial and/or temporal relationships. The design and concept of the soundscape is difficult to progress without an understanding of the environment. The navigation possibilities, visual objects, stimuli, perceptual opportunities within the environment may all be relevant considerations before any sounds are chosen, never mind structured relationally. For example – a visual texture of gravel, placed on the ground of the virtual space, has a suitably expected (or unexpected!) sound effect when it is walked on. Even in the case of a LISTEN-style environment, it is necessary to consider the space in which the interactor will move around and if there are any visual subjects related to a sound in a specific position. The construction, or at least knowledge of the virtual or augmented space must precede the sound design.

6.2 Objective and content themes

The intended experience of the example virtual environment is to create an impression of a 'memoryscape', merging interpretations of structure, content and function. 'Garden monologue' fits within the genre or style of a virtual artwork or poem, and is inspired the psychological excavation of a site containing frustration. Constructed from layers of descriptive, historical, poetic, fantastical and associative relationships to the personal landscape, the interactive experience aims towards giving the interactor an impression of 'unearthing' and 'unrevealing' meaning and non-meaning. However, references are made to remind and question the nature and reason for constructing such a space.

Spoken voice monologue forms the majority of the sounds played, at the present time of writing, though changing ambient sound and effects add atmosphere to the environment. The spoken word can be interpreted as a monologue 'stream' of consciousness, broken up and divided in time and space, to be interacted with. As example sounds, the spoken word is useful to test the creative application and flexibility of the relativity attributes of the ArchaeSoundTrigger class, because as content it offers the greatest challenge to consecutive and consequential aspects of narrative.

The visual aesthetic and characteristic of the environment is that of a removed space of landscape; a garden half-remembered or only of partial significance (see figure 12). The model textures are taken from the actual garden site, but are combined with abstract colours and occasionally arranged reminiscent to a landscape architecture concept sketch. A 'cut-out' section of grass lawn hangs in mid air, forming the largest identifier of the garden. One edge of the section breaks down into half completed construction work, steps hang in mid air either to be in existence, or now obsolete. Rose heads hang in mid air, while large free-standing black and white photos of hills populate the north and west space suspended around the section.

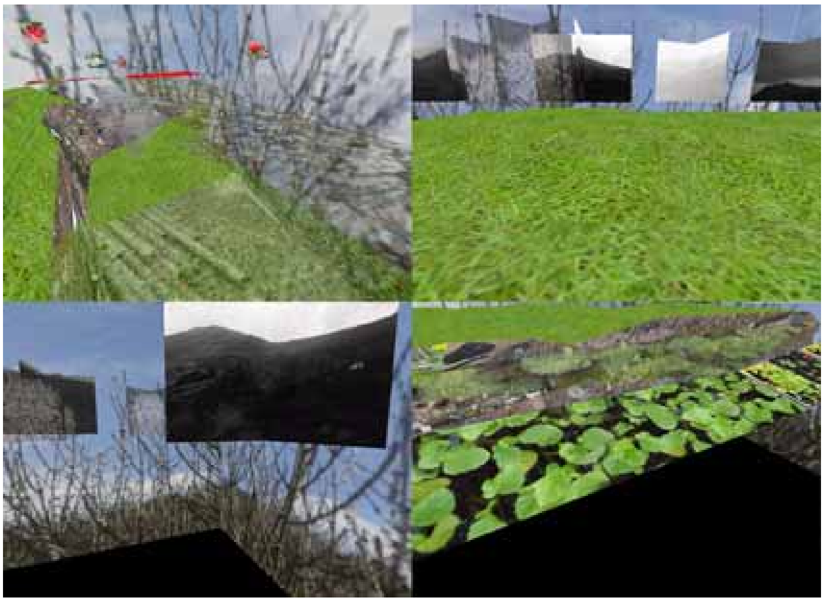


figure 12. Screen shots of 'Garden Monologue' at run-time.

It is intended that the interactor discovers with surprise the ability to walk on the same level in mid air around the section, and explore the space between and behind the spaces, mixing photo-figurative imagery with abstract black and whites 'fields' of colour. There is a mixture of spoken references to objects present in sight, for example 'the hills', 'the roses', 'the grass', and those which do not exist but are suggested by location, such as 'the (apple) tree', 'the west wall', and the 'deposit pools'.

---

[19] For further information: http://www.mpi-sb.mpg.de/~arche

6.3 Phasing and sequencing to data lists

Following the decision of theme, and the generation of textual content - to be recorded as spoken voice - it was necessary to arrange the different units of words and short phrases into bundles of consecutive sequences, that may form 'RollOver' temporal sequences, and those that take the form of consequential sounds, i.e. single 'OneShot' sounds. Also to consider were words that will play once only upon entering the collision radius, and those that reward the interactor's agency of returning to the same position.

The author decided to test the suitability of the ArchEd Harris matrix program to plan out a relational layout of textual narrative and sounds, with minimal pre-planning. This software allows the possibility of organising relationships such as those encountered in (figure 4) - earlier, later than, contemporary, equal - in an easily editable manner, and move around the arrangements as the network increases. However, the most obvious limitation of the program is that it is made for expansion downwards in the application window - from an archaeologists point of view - recorded as the strata is encountered, so that the latest is recorded first and the earliest recorded last. For a virtual environment/ narrative designer, this feature of the program opposes usefulness, as it is necessary to start not at the last temporal arrangement but the first. Therefore to maintain relativity, and allow the expansion forward in time of consequential arrangements, it is necessary to scroll to a starting point far down the document to allow space for development and placement of icons. Also worthy of note, ArchEd assigns contemporary relationships by attaching an accumulative number to both unit icons. If the order of assigning contemporary relationships is done haphazardly, it is more difficult to assign phase numbers at the later stage, as the increasing 'contemporary' number is no longer following relative relationships.

These points aside, the program proves to be useful in allowing creative interactivity in arranging relative sequences, with the option of using descriptive tags for each unit, rather than the default number option (figure 13). In the process it was possible to decide, during layout, contemporaneous groups that would be in existence at the beginning of each phase by attaching contemporary relationships between the desired units. By zooming out, an overview of the sequence and content balance may be seen, visualising the authored temporal map, and utilising the Harris matrix generator function, reorganise the layout to best suit its relativity. It should be noted, however, that this author did not find this functionality particularly useful when large unit names are used, as the reorganised map has to be reduced to an unreadable level, to accommodate the space necessary when many units are placed according to known relativity (figure 14).

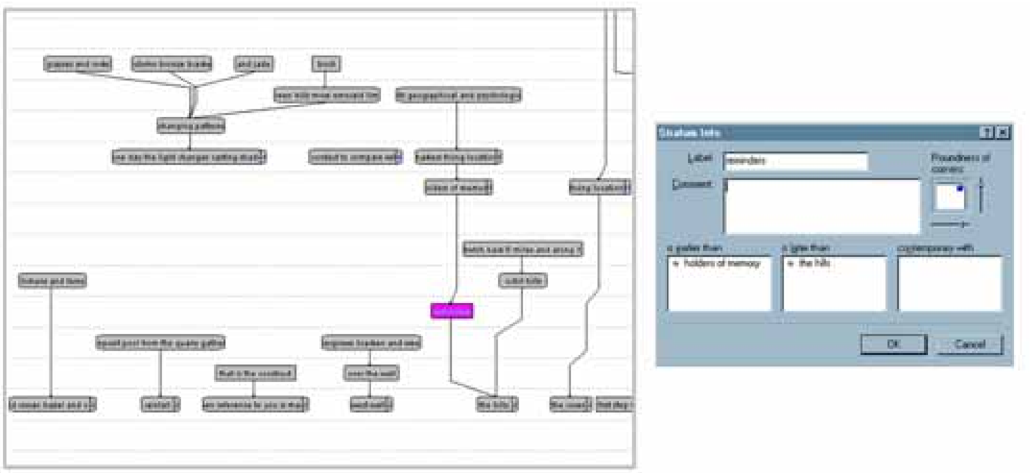


figure 13. Descriptive content tags (close up) in arranged sequences

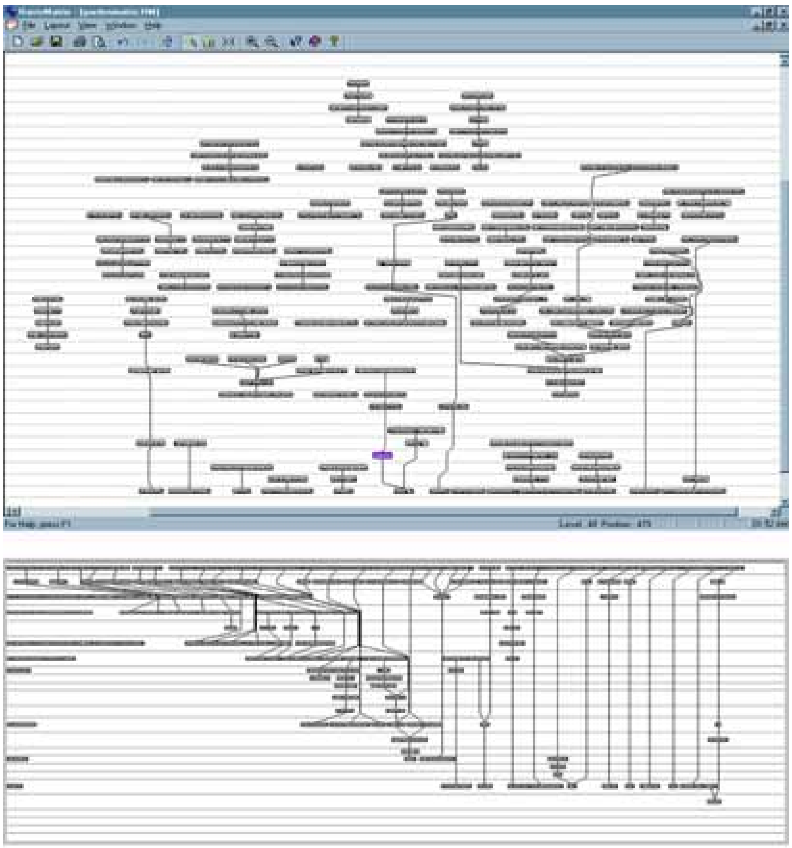


figure 14. Overview of temporal arrangement, and below the result of using ArchEd's Harris matrix generator function. [See appendix C].

Once the arrangement was complete to point of satisfaction, including the desired treads of consecutive words that are applied to follow each other, the threads were transcribed into a table using word processing software (figure 15). The word-list of threads was then recorded as voice monologues, and other sounds gathered and recorded into similar lists. This part of the process mimics the creation of textual recorded or archival information [3.1] gathered during the excavation of an archaeological site, and acts as a transitory archive to be applied into the virtual or augmented environment.

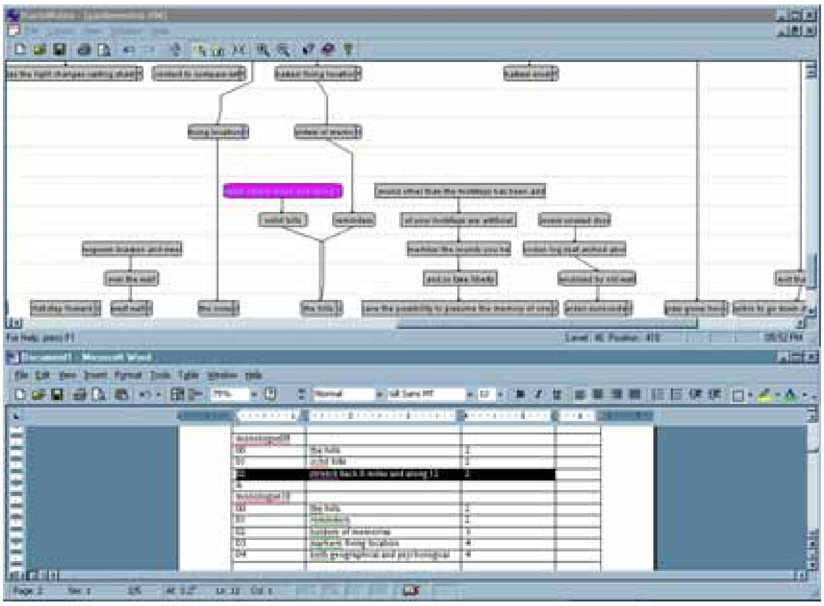


figure 15. Using Harris Matrix to create sequence and sound data list.

6.4 Transferring the record into Unreal Editor

Unreal Editor offers four different viewing windows – top, front and side orthographic views consisting of line drawings, plus a 3D perspective view showing lighting and textures on the objects within the scene. Specific to Unreal Editor, all views also show the icons representing authoring and interactive features applicable to the environment during run-time.

Although the orthographic views are useful for two dimensional measurement, manipulation and comparison of 'brushes', these views can often become rather cluttered. In the specific case of the 'Garden Monologue' map used within this study, the orthographic views show the polygon meshes of terrain model, which to a greater extent have transparent textures attached and are thus invisible within the run-time environment. Furthermore, a direct view, looking down from above, of the 3D projection view, does not offer the necessary overview suitable to use as a plan map of the space (figure 16).

In section [4.1.4], discussing how the archaeological method can be applied to spatial sound design, it is suggested that plan maps are most suited for the 'recording' of sounds using the archaeological method - i.e. identifying the spatial positions and radius of the sound at the beginning of each phase. However, upon placement of the sound 'actors', reference to visual cues within the virtual or augmented environment is often a necessary requirement to assist design, and co-relate sound with location. Therefore, both the plan map *and* the 3D texture window are utilised. When the sound 'actor' is placed at the desired location, the sound data list, which was compiled as a result of earlier relative sequencing, is transferred into the Relativity and StructSound data lists of the actor properties interface (figure 17).

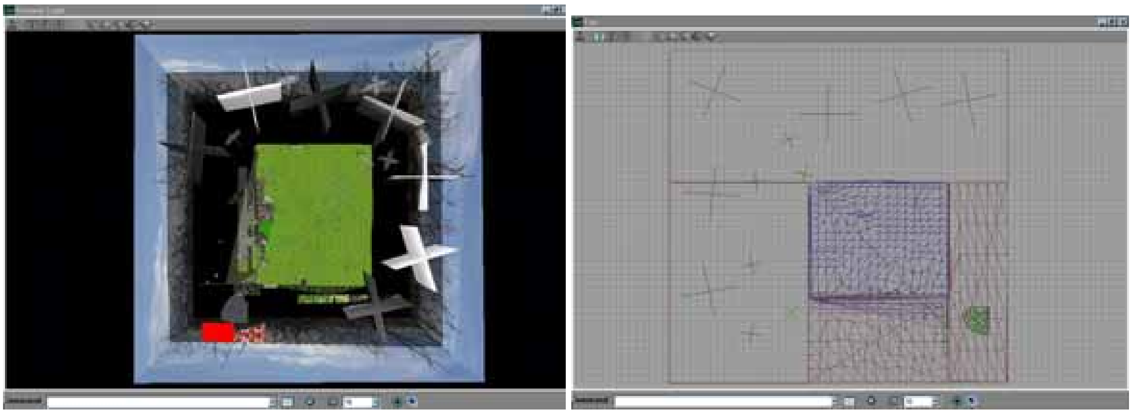


figure 16. Comparison of the 3D perspective view looking down, and the top orthographic plan view of virtual
environment in Unreal Editor.

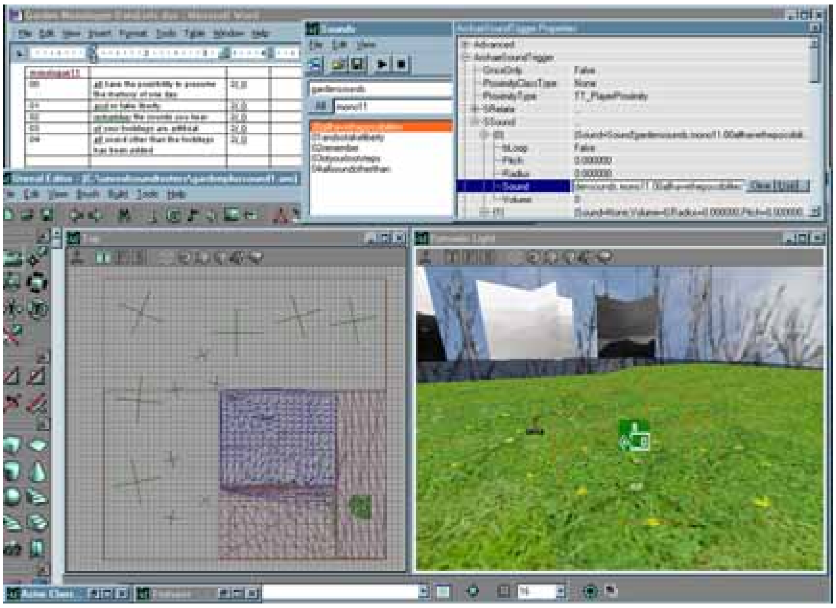


figure 17. Using a Sound data list to assist data entry into Unreal Editor.

As the placement of actors are made in the plan view, this author has assigned icon textures and group tags to segregate the ArchaeSoundTrigger actors into different groups, mimicking the charting of archaeological stratigraphical units onto different plans for each phase transition. By assigning each sound to a group name, a series of electronic phase 'plans' exist when groups are checked and unchecked visible, depending upon which phase is desired to be visible (figure 18 - Note in this figure that the collision radii have been turned on to assist planning the agency or narrative potential of each placed sound).

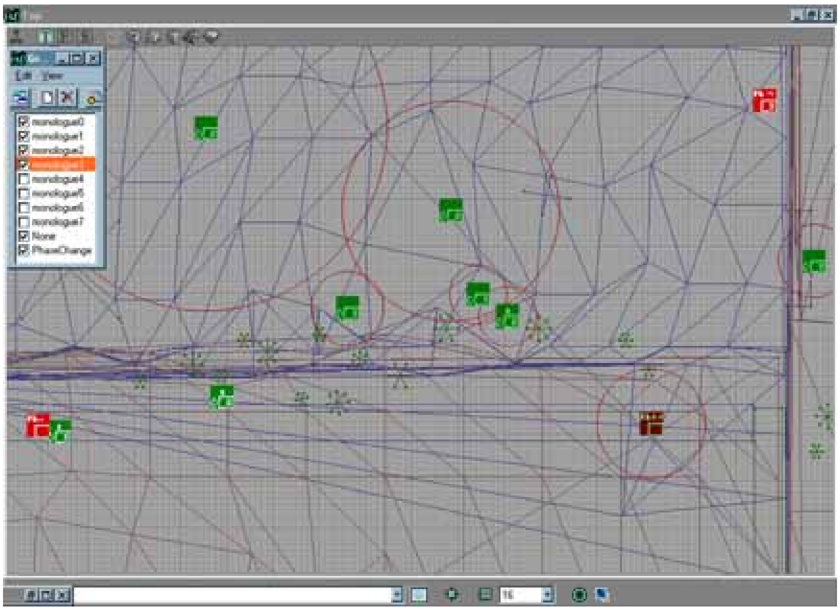


figure 18. Using groups, icon textures and radius to assist placement of 'archaeological' sound actors in the plan view.

Based upon a reversal of the archaeological stratigraphic methodology, it is at this stage the metaphor of the record breaks down, and veers off away from archaeology.

The key factor for this lies essentially with the creative act. Contrary to the practice of archaeology, creative activity and design are not constrained scientifically or empirically in construction. The designer is unlikely to be replicating the exact relay of sounds for interpretation, in a manner of accuracy that the archaeologist would record the material deposits of a site. The record or sequence, applied for creative or narrative purposes is, due to its fabrication, free from the confines of empirical representation. Flexible and adaptable to change, the record, in this case, is reflective of the creative potential of the digital medium for construction, the nature of the authoring tool, and therefore also process. Creativity and design, especially in experimentation, often relies upon taking steps backwards, revising and adapting to experience.

It should therefore be presumed that the process of 'recording' in the design process is not absolutist, and may actually be better described as a period of 'transference and modification', according to run-time test experience in developing the environment. During the process of designing 'Garden Monologue', both content and stratigraphic relativity was adapted during the transferring stages of creating the sound data lists after sequencing. Responding to the experience of listening to interactive tests of the soundscape, variables such as radius, and sound samples have been 'tweaked', and either the matrix sequence or data list record, updating to suit.

6.5 Analysis of archaeological classes in design and real-time

If the reader recalls, the success criteria of the custom-build classes was to allow both *consecutive* and *consequential* dynamism. Or in other-words linear sequencing of sounds, and non-linear interactivity of change, including the maximum flexibility of form to the designer. How do the classes stand after following the design process into practice? What works and where are the present limitations?

As a basic information structure, the `SSound` and `SRelate` data lists of the `Archae-SoundTrigger` class satisfies the transference of relevant information characterising the sound from the sound data list. The separation of relativity from sound characteristics such as volume and radius, rather than being a hindrance, allows the designer to easily scan – overlook without much scrolling - the relationship between different sounds in the list, despite the annoying feature of the Editor to extend the branches of structures occasionally as a default.

However one aspect of the sound characteristic has been ill considered[20] - that being the *length* of time which each sound plays for. This becomes apparent during run-time, particularly when the `ArchaeSoundTrigger` is in the `RollOver` state, and the interactor is within the `collision radius`. As the class executes, it does not pause for the previous sound to finish what is being played, before pausing the allocated amount of time specified, and playing the following sound. Therefore sounds overlap each other depending upon the length and pause associated with each. Granted, this can have an interesting accumulative effect, but is problematic for continuing spoken sentences or other such tightly controlled samples. There is an UnrealScipt function, `GetSoundDuration(sound Sound)` that may be used, however this author has tamed and controlled overlapping sounds in the `ArchaeSoundTrigger` actors by noting the time length of each sound in the archival record [see Appendix B], and considering its addition to the following sound's pause time length during the design process. This method successfully solves the problem, but still allows accumulation to be a feature if desired.

A large part of the design process involves understanding and moving between the different phase plans, considering how agency and sound fragments may accumulate to form meaning. As each `ArchaeSoundTrigger` actor has been allocated to a group representing an origin of existence phase, it is possible to understand where within the temporal sequence the sound will be first triggered and so experienced. However, as some actors have a sound data list that contains a combination of different assigned phases, it therefore means that those sounds which don’t originate in the assigned 'base' phase are lost to the map. What would be very useful to account for this visual loss within the design would be a responsive system that checked the data list, and displayed the appropriate icon texture for the phase selected for view.

In general the `ArchaeSoundTrigger` class successfully offers a mixture of options to define characteristics to a series of sounds, and plays them in real-time dependent upon the current phase and the interactor's trigger of the collision radius. The different options available to the designer - looped, one-shot, rollover, sequential or once only - implemented in the 'Garden monologue' example - suggest the possible richness of sound collage or narrative that may be accumulated and layered.

What is apparent from run-time experience is the challenge of encouraging both agency and patience while using the `ArchaeSoundTrigger`, especially when in the `RollOver` state. Due to the nature of the Unreal Engine - fast movement, and quick viewing ability - it is possible for the interactor to walk straight through the collision radius and the sound radius, triggering the sound and cutting it out, moving on elsewhere, before the sound has had the chance to play out. Although quite obviously a factor of disembodied movement in a virtual environment (using a mouse and keyboard) this problem also has the potential to affect audio-augmented environments, whereby the user is actually walking around a space listening to fragments of sound. If a small collision circle is similar to the size of the sound's radius, it is likely that the sound will be cut off either quickly or soon after it has started, as the listener continues to move around[21]. A practical measure this author has found, especially for small radii, is that the sound radius should extend beyond the collision radius. Thus the audibility of the sound is sustained, even when the user is walking away from the source. Regardless, a culture of listening in both these environments has to be encouraged that rewards patience to remain in one position for a short period of time, and also tentative movements rather than mad gallops.

The `Artefact` actor, as discussed earlier, was intended to take the form of an ambient sound, operating out-with the necessity to be triggered. In present form the class needs further development, to release its metaphorical potential. The over-bearing reliance upon agency - i.e. the crossing of the collision radius - to present a phase-relevant sound to the interactor, reduces opportunities to plan temporal developments of narrative that the interactor may have no control over or relationship with; for example, the spatial movement of a particular bird song over the period of time in one phase. Ideally the 'artefact' sound would be activated upon entry to a new phase, and exist for a certain period of time, before disappearing from the site in present tense; thus acting as an attractor or retainer for its location [Fencott00].

---

[20] This short-coming has been the fault of focusing development tests on sound samples of spoken numbers - markers for correct sequencing and restraint.
[21] However, inverted, this problem could become a feature of the soundscape, whereby sound only plays when the player stops moving and waits.

The `PhaseChange` class, an object of agency and consequence, operates successfully to change the current phase counter in each temporal actor. The attached sound proves to be a useful indicator of change, though only if the sounds or spoken word/s used are constantly similar or markedly different from the others, so that the interactor learns to associate the potential change of soundscape with that sound. This actor embodies the apparent disparities between linearity, archaeology or history, and non-linearity, agency and individual construction of narrative. It is possible, depending upon the assigned state, to jump from beginning to middle, back and forward to the end of the sound data list, or even perpetually revolve through a loop of phases with little understanding of the full content. There is a danger that the interactor becomes frustrated with this situation, or indeed that they trigger all the `OnceOnly` and temporal `Artefact` actors, leaving few sounds left to interact with. These scenarios pose the question - 'should the phases also be controlled by an overriding real-time variable, in a similar manner to Davies's 'Ephemere' project?

To summarise, the application of the archaeological classes satisfactorily indicates the potential of integrating agency and narrative at the most basic level of units as suggested by [Fencott01] in section [2.1]. The authoring process accommodates a generous variety of options to manage the relative history of sound on repeated visits to the same position by the interactor. However, a practice of patience has to be implemented by the user, if their agency is rewarded by sequential narrative. All too often the aesthetic of game-like 3D virtual environments encourages the urge to navigate quickly round the space, cutting off or missing altogether sequential sound sequences. It is presumed that such problems are lessened in an audio-augmented environment application.

6.6 Issues for further development

Continuing from the previous section, certain areas of further development and necessary focus have been highlighted as a result of using stratigraphy and the archaeological record as a metaphor for placing spatio-temporal sound.

As suggested in the previous section, the present system as it stands treats the sound to be played as an instance, with one node of existence, rather than what is essentially an entity with two nodes, i.e. a beginning and an end. The inclusion of a second relational node - the terminal point of the sound - would involve additional permutations of contextual relationships. For example beyond earlier than, later than, and contemporary, it would be necessary to have defining categories which considered asymmetrical synchronisation and potential relationships between the beginning and end of two different sounds. During initial research into archaeological methods, one source [Holst00] considered the problem of temporal structures in medieval settlements, proposing categories that defined such relationships in light of ambiguous overlapping temporal evidence of occupation at a site. The approach aimed to identify relationships that would be useful for interpretative constructs and subjective evaluation. This author decided that it was important to lay foundations first using [Harris79]'s rather more objective approach, and suitably this limited implementation has highlighted the expressed need to further address the temporal occupation of sound and its relationships of relativity to other sounds that also have a temporal occupation of time.

Using the archaeological record as a metaphor has allowed this author to construct a structure, which although dynamic in movement through predefined divisions in time, it is essentially static in behaviour and location. For example the states of the classes cannot be altered once the environment is operating in real-time, and the sound actor is fixed into position. As next step research and development, it would be necessary to consider how programmable behaviours or AI can be attributed to the sound actor, and how such features would develop narrative potential.

6.7 On-going content creation

The creation of the soundscape for 'Garden Monologue' is an ongoing process, experimenting with different forms of content and spacio-temporal relationships. The challenge of interactive narrative, and 'archaeological' sound began with the use of spoken word. However it is the aim of this author to use this form as a starting point towards less literal forms, atmospherics and abstraction. Emphasising a previous point, the record is organic and adaptable, reflecting shifting associations and interests. It will be presented as accompaniment to the partial requirements of MSc CAGTA degree, in the week beginning 17th September 2001, at its then present stage of development.

7 Conclusion

The objective of this thesis was to explore and develop a framework to facilitate the authoring of sound with spatio- temporal relativity in virtual or augmented environments.

The archaeological record of a site is the result of a process, which charts the spatial and temporal relationships between stratigraphical units of material and the artefactual finds within it. At the beginning of the research period it was hypothetically presumed that archaeology, as a process of recording and interpreting the history of a site, could offer guidance for the author of a virtual/augmented environment. Similar objectives, i.e. to create a history of information - sound samples within the site - was the basis of this presumption.

Initial background study into interactive environments and mediums has raised awareness of the dialectic nature of agency and narrative, highlighting the concept that the interactor creates their own individual narrative from their own agency. This personalised interpretation of the site, based upon their exploration of the environment and the fragments of narrative gathered, strongly suggests a parallel to the action of the archaeologist, likewise charting and gathering items of interest from an excavation.

Based upon this premise, the author of the spatio-temporal soundscape creates a virtual 'archaeological' database of narrative fragments for the interactor to reveal via their agency in the site. By reversing the archaeologists' process of recording empirical information to interpret into historical relativity, and thus narrative, a basic procedure of design took form. This methodology was tested in the production example entitled 'Garden Monologue'. The following conclusions can be made regarding the successful and limiting aspects of the proposed methodology:

- The design and conceptualisation of a soundscape is difficult to progress without an understanding of the virtual or augmented environment. This is step one.

- Content should consider a culture of listening and asynchronicity, facilitating agency and rewarding patience. Note also that the dimension of time involves much more content and planning.

- By breaking down the content into different groupings or phases, sequences of events can be planned, using the metaphor of stratigraphy. This is imagined, without depth, in virtual/augmented environments as layers of successive *present tense* 'overlays of sound' occupying the site.

- Narrative development can usefully be managed by assigning sounds to exist in a particular phase. Each phase of the soundscape may contain the basic units of sonic narrative, namely consequential sounds, which reward *active* agency or movement, and consecutive sounds, which reward *passive* agency or standing still and listening.

- A computer Harris matrix program (such as ArchEd) used to create sequences of consecutive and sequential sound has been found to be useful for flexible creating / visualising temporal relationships between small sections of narrative content. Relative relations such as earlier-than, later-than and contemporary can be ascribed, assisting the transferring of sequences to databases for application into the authoring tool.

- However the Harris matrix and stratigraphical sequencing is ultimately limited in relevancy to sound, due to its treatment of basic units as single nodes of existence – sound needs two, a beginning and an end – therefore further temporal relations have to be identified which account for the length of time each sound plays for.

- Both a plan and 3D perspective view are necessary to place sounds with radii in a virtual or augmented environment, to take into consideration the relationship between sound content and place.

- Agency and interactivity are facilitated via movement and triggers, which can change the phase forward, backward or jump to a defined phase. Conflict arises between interactive temporal change and the sense of an over-riding 'archaeological' narrative.

- There is a need to develop the artefact metaphor for a non-interactive sound, to compliment agency, which exists solely in a time frame of existence. The artefact can be one or all of the following: a clue, a treasure, an indicator of function or activity, and a linkage to the cultural, historical environment.

- Contrary to the static archaeological record, the method applied creates a record which is, and should be, adaptable and flexible to change. The creative act is not often based upon scientific or strictly empirical recording. The record is created as a state of transference and modification.

As if in response to the last point, the following quote regards the issue of agency in reference to the archaeological record. Suitably, as a closing remark, it conveniently comments upon the value of perseverance in creating a structuring methodology for authoring sounds in virtual and augmented spaces:

> "Agency is always situated in structural conditions which facilitate its actions because agency requires a medium through which to work. Practice is therefore structured by the resources, which are its medium and its outcome. These resources extend from material and symbolic resources to traditions of execution and expression…
>
> …Structures should not be regarded as simply constraining or determinate, but rather as a field of possibilities reproduced by the practices which occupy that field" [Barrett01: 150].

8 References

[Barrett00] BARRETT, John C. 2000.
'*A thesis on agency*',
Agency in Archaeology, Ed. Dobres, M-C. & Robb, J., Routledge Press.

[Barrett01] BARRETT, John C. 2001.
'*Agency, the Duality of Structure, and the Problem of the Archaeological Record*',
Archaeological Theory Today, Ed. Ian Hodder, Polity Press.

[Barthes88] BARTHES, Roland. 1988.
'*Introduction to Structural Analysis of Narratives*',
The Semiotic Challenge, translated by Richard Howard, Basil Blackwell.

[Castlef92] CASTLEFORD, John. 1992.
'*Archaeology, GIS, and the time dimension: an overview*'
Proceedings of Computer Applications and Quantitative Methods in Archaeology, (CAA) 91, BAR International Series, Oxford.

[Campbel00] CAMPBELL, Euan & LANE, Alan. 2000.
'*Dunnadd: an early dalriatic capital*',
Oxbow Books, Oxford.

[Davies99] DAVIES, Char. 1999.
'*Éphémère: landscape, earth, body, and time in immersive virtual space*',
Chapter 7: projects, Reframing Consciousness, Edited by Roy Ascott, Intellect Books, Exeter.

[Eckel01]a ECKEL, Gerhard. 2001.
'*Immersive Audio-Augmented Environments*',
Proceedings of the 8th Biennial Symposium on Arts and Technology, Connecticut College, New London, CT, USA, 2001.

[Eckel01]b ECKEL, Gerhard. 2001.
LISTEN website, http://listen.gmd.de

[Fencott99] FENCOTT, Clive. 1999.
'*Content and Creativity in Virtual Environment Design*',
Paper presented at Virtual Systems and Multimedia '99 (VSMM'99), University of Abertay, Dundee, Scotland.

[Fencott01] FENCOTT, Clive. 2001.
'*Virtual Storytelling as Narrative Potential: towards an ecology of narrative*',
Paper presented at First International Conference on Virtual Storytelling 2001, Conference Centre, Avignon, France.

[Greene83] GREENE, Kevin. 1983.
'*Archaeology: an introduction*'
Batsford Academic Press.

[Harris79] HARRIS, Edward. 1979.
'*Principles of Archaeological Stratigraphy*',
Academic Press.

[Herzog90] HERZOG, Irmela & SCOLLAR, Irwin. 1990.
'*A New Graph Theoretic Oriented Program for Harris Matrix Analysis*',
Proceedings of Computer Applications and Quantitative Methods in Archaeology (CAA) 90, BAR International Series, Oxford.

[Holst00] HOLST, Mads Kahler. 2000.
'*Formalizing Fact and Fiction in Four Dimensions: a relational description of temporal structures in settlements*',
Proceedings of Computer Applications and Quantitative Methods in Archaeology (CAA) 00, BAR International Series, Oxford.

[Johnson97] JOHNSON, Ian. 1997.
'*Mapping the Fourth Dimension: the timemap project,*'
Proceedings of Computer Applications and Quantitative Methods in Archaeology (CAA) 97, BAR International Series, Oxford. http://www.timemap.net

[Jull98] JULL, Jesper. 1998.
'*A Clash between Game and Narrative: interactive fiction*
(*Why computer games do not tell good stories and why this is not a problem.)'*,
Paper from the Conference of Digital Arts & Culture 98, Bergen.

[Leonard99] LEONARD, Tim. 1999.
'*Postmortem: Looking Glass's Thief: the dark age*',
Article, Game Developer magazine, July 1999.

[Lerner97] LERNER, Sophea. 1997.
'*Resonating the Fragmented Body: sound design in new media*',
Lecture paper, International Society of Electronic Arts (ISEA) 97.

[Meyer95] MEYER, Kenneth. 1995.
'*Dramatic Narrative in Virtual Reality*'
Communication in the Age of Virtual Reality, Ed. F. Biocca & M.R. Levy, Hillsdale/NY: LEA Publishers, p. 219-258.

[Murray97] MURRAY, Janet H. 1997.
'*Hamlet on the Holodeck: the future of narrative in cyberspace*',
MIT Press,

[Shanks01] SHANKS, Michael, 2001.
'*Culture/Archaeology: the dispersion of a discipline and its objects*',
Archaeological Theory Today, Ed. Ian Hodder, Polity Press.

[Sween99]a SWEENEY, Tim. 1999.
'*Unreal Audio Subsystem*',
http://unreal.epicgames.com/Audio.htm

[Sween99]b SWEENEY, Tim. 1999.
'*UnrealScript Language Reference*',
http://unreal.epicgames.com/UnrealScript.htm

[Weir00] WEIR, Paul. 2000.
'*Satisfaction with Interaction: future prospects for interactive audio'*
Paper from the SoundingOff2 conference on new media audio, London.

9 Appendices

A Custom-made UnrealScript class examples (full code listings):

```
//=============================================================================
// ArchaeSoundTrigger. For looping or OneShot sounds
//=============================================================================
class ArchaeSoundTrigger expands Triggers;

// Sound Type.
struct StructSound
{
	var() sound sound;	// sound sample
	var() byte	volume; // between 0 and 255
	var() float	radius; // radius of heard sound
	var() float	pitch;	// 0.5 = half pitch, 1.0 = normal etc.
	var() bool	bLoop; // looping sound or one shot
};

struct Relativity
{
	var() int	phase; // phase in which the sound exists (plays)
	var() float pause; // length of time in seconds between samples in rollover state
};

// Trigger type.
var() enum ETriggerType
{
	TT_PlayerProximity,	// Trigger is activated by player proximity
	TT_PawnProximity, // Trigger is activated by any pawn's proximity
	TT_ClassProximity,	// Trigger is activated by actor of that class only
	TT_AnyProximity,	// Trigger is activated by any actor in proximity
} ProximityType;

var()	Relativity SRelate[16];// What phase and pause each sound is associated with
var()	StructSound SSound[16]; // What to play(must be at least one sound)
var()	class<actor> ProximityClassType; // Type of class which triggers sound
var()	bool OnceOnly; // Iterates through the sound samples once or loops
var	int numSounds; // The number of sounds available
var	int curSound; // Which sound is to be played currently?
var	bool bPlayedOnce; // Flag to indicate if looped has played once
var	bool bRollOn; // Flag to indicate if Rollover state
var	int PhaseNow; // Phase updated incrementally
var	float Fvol; // volume
var	float Frad; // radius
var	float Fpitch; // pitch


function BeginPlay ()
{
	local int i;

	// Calculate how many sounds the user specified
	for (i=0; i<16; i++)
	{
		if (SSound[i].sound == None) //if there are no sounds entered break
		{
			numSounds=i;
			break;
		}
	}

	PhaseNow = 0;
	curSound = 0;
	if ( OnceOnly ) bPlayedOnce = false;
}
```

```
final function bool IsRelevant( actor Other )
{
     // trigger type
     switch( ProximityType )
     {
          case TT_PlayerProximity:
               return Pawn(Other)!=None && Pawn(Other).bIsPlayer;
          case TT_PawnProximity:
               return Pawn(Other)!=None && ( Pawn(Other).Intelligence > BRAINS_None );
          case TT_ClassProximity:
               return ClassIsChildOf(Other.Class, ProximityClassType);
          case TT_AnyProximity:
               return true;
     }
}

state() PlaySoundOneShot
{
     // Called when something touches the trigger.
     function Touch( actor Other )
     {
          if( (IsRelevant( Other ) &&
               (SRelate[curSound].phase == PhaseNow) &&
               !OnceOnly)
               ||
               (IsRelevant( Other ) &&
               (SRelate[curSound].phase == PhaseNow) &&
               OnceOnly &&
               !bPlayedOnce) )
          {
               bRollOn = false;
               if (SRelate[curSound].pause == 0)
                    // if no value is entered default pause is 0.5 secs
                    SRelate[curSound].pause = 1.0;

               SetTimer(SRelate[curSound].pause, False);
          }
     }
}


state() PlaySoundRollover
{
     // Called when something touches the trigger.
     function Touch( actor Other )
     {
          if ( (IsRelevant( Other ) &&
           (SRelate[curSound].phase == PhaseNow) &&
           !OnceOnly)
           ||
           (IsRelevant( Other ) &&
           (SRelate[curSound].phase == PhaseNow) &&
           OnceOnly &&
           !bPlayedOnce) )
          {
               bRollOn = true;
               if (SRelate[curSound].pause == 0)
                    // if no value is entered default pause is 0.5 secs
                    SRelate[curSound].pause = 1.0;

               SetTimer(SRelate[curSound].pause, False);
          }
     }

}
```

```
function Timer ()
{
        if ( SRelate[curSound].phase != PhaseNow ) return;

        if (SSound[curSound].bLoop == true)
        {
             AmbientSound = SSound[curSound].sound;

             if (SSound[curSound].volume == 0) SoundVolume = 128;
             else SoundVolume = SSound[curSound].volume;

             if (SSound[curSound].radius == 0) SoundRadius = byte(CollisionRadius)/33;
             else SoundRadius = byte(SSound[curSound].radius)/33;

             if (SSound[curSound].pitch == 0) SoundPitch = 64;
             else SoundPitch = byte(SSound[curSound].pitch) * 64;

             BroadcastMessage ( "sample number " $ curSound, false);
             BroadcastMessage ( "volume = " $ SoundVolume, false);
             BroadcastMessage ( "radius = " $ float(SoundRadius)*33, false);
             BroadcastMessage ( "pitch = " $ float(SoundPitch)/64, false);
        }

        if (SSound[curSound].bLoop == false)
        {
             AmbientSound = None;
             // if no entry for volume is set to default 2.0
             if (SSound[curSound].volume == 0) Fvol = 2.0;
             else Fvol = float (SSound[curSound].volume)*2.0;

             // if no entry for radius is set to default of Actor's Collision Radius
             if (SSound[curSound].radius == 0) Frad = CollisionRadius;
             else Frad = SSound[curSound].radius;

             // if no entry for pitch is set to default 1.0
             if (SSound[curSound].pitch == 0) Fpitch = 1.0;
             else Fpitch = SSound[curSound].pitch;

             //BroadcastMessage ( "sample number " $ curSound, false);
             //BroadcastMessage ( "volume = " $ byte( Fvol ), false);
             //BroadcastMessage ( "radius = " $ Frad, false);
             //BroadcastMessage ( "pitch = " $ Fpitch, false);

             PlaySound (SSound[curSound].sound, SLOT_None, Fvol,, Frad, Fpitch);
        }

        curSound++;

        if ( (curSound == numSounds) && !OnceOnly )
                curSound = 0;
        if( (curSound == numSounds) && OnceOnly )
                bPlayedOnce = true;

        if ( bRollOn == true ) SetTimer(SRelate[curSound].pause, False);
}
```

```
//=============================================================================
// Artefact.  Currently plays by trigger
//=============================================================================
class Artefact expands triggers;

struct StructSound
{
     var() sound sound;
     var() byte bLoopVol;
     var() float volume;
     var() float radius;
     var() float pitch;
     var() bool bLoop;
};

var() enum ELife
{
     AL_Eternal, // Exists for all time within phase
     AL_Temporal, // Plays only for as length ascribed in 'ExistsInPhase' number

} ArteLife;

// Trigger type.
var() enum ETriggerType
{
     TT_PlayerProximity,    // Trigger is activated by player proximity
     TT_AnyProximity,    // Trigger is activated by any actor in proximity
} ProximityType;

var()    StructSound ASound;
var      int APhaseNow; // Phase now
var()    int OrigInPhase; // Originates in phase
var()    float Delay; // delay after phase begins that artefact exists
var()    float InExistenceTime; // length of time artefact is in existence
var()    name Descript; // Name assigned to Tag
var      bool Playing; //
var      float vol; // volume
var      float rad; // radius
var      float pitc; // pitch



function BeginPlay()
{
     APhaseNow = 0;
     Tag = Descript;
     Playing = false;
     InitialState = 'active';
     if ((ASound.bLoop == false) && (InExistenceTime == 0))
          InExistenceTime = GetSoundDuration( ASound.sound );

}

final function bool IsRelevant( actor Other )
{
     switch( ProximityType )
     {
          case TT_PlayerProximity:
               return Pawn(Other)!=None && Pawn(Other).bIsPlayer;
          case TT_AnyProximity:
               return true;
     }
}

auto state() active
{
     function Touch( actor Other )
     {
          if ( (IsRelevant( Other )) && (OrigInPhase == APhaseNow) )
          {
               if ((ArteLife == AL_Temporal) && (Playing == false))
               LifeSpan = InExistenceTime + Delay;

               SetTimer(Delay, false);
          }
     }
}
```

```
function Timer()
{
         if (ASound.bLoop == true)
         {
              AmbientSound = ASound.sound;
              if (ASound.bLoopVol == 0) SoundVolume = 128;
              else SoundVolume = ASound.bLoopVol;

              if (ASound.radius == 0) SoundRadius = byte(CollisionRadius)/33;
              else SoundRadius = byte(ASound.radius)/33;

              if (ASound.pitch == 0) SoundPitch = 64;
              else SoundPitch = byte(ASound.pitch) * 64;

// BroadcastMessage ( "Artefact " $ Tag $ "  volume = " $ SoundVolume, false);
// BroadcastMessage ( "radius = " $ ASound.radius $ "  pitch = " $ ASound.pitch,false);
// BroadcastMessage ( "lifespan = " $ LifeSpan, false);

         }

         if (ASound.bLoop == false)
         {
              AmbientSound = None;
              if (ASound.Volume == 0) vol = 2.0;
              else vol = (ASound.Volume)*2.0;

              if (ASound.radius == 0) rad = CollisionRadius;
              else rad = ASound.radius;

              if (ASound.pitch == 0) pitc = 1.0;
              else pitc = ASound.pitch;

              BroadcastMessage ( "Artefact " $ Tag $ "  volume = " $ vol, false);
              BroadcastMessage ( "radius = " $ rad $ "  pitch = " $ pitc, false);
              BroadcastMessage ( "lifespan = " $ LifeSpan, false);

              PlaySound (ASound.sound, SLOT_None, vol,, rad, pitc);
         }
         Playing = true;
}
```

```
//=============================================================================
// PhaseChange.
//=============================================================================
class PhaseChange expands Trigger;

var() enum ERelative
{
    PD_Forward,       // Phase is incremented ++
    PD_Backward,      // Phase is incremented --
    PD_JumpTo,        // Jump to phase selected

} PhaseDirection;

var() enum EPhaseChange
{
    PC_TriggerSelect, // Tag and Event have to be the same for phase change
    PC_TriggerAll,    // Effects all 'ArchaeSoundTriggers' actors

} PhaseChange;

var() enum ELife
{
    PL_Eternal, // Triggers for all time
    PL_ExistsInPhase, // Triggers only in ascribed 'ExistsInPhase' number

} PhaseLife;

//struct NarraDetails
//{
//  var() bool LinearNarrative; // Either linear or non-linear narrative type
//  var() float PhaseTimeMins[16]; // Length of each phase in minutes
//};

struct StructSound
{
    var() sound sound;    // sound sample
    var() float  volume; // 0.5 = half volume, 1.0 = normal, etc.
    var() float  radius; // radius of heard sound
    var() float  pitch;   // 0.5 = half pitch, 1.0 = normal etc.
};

var() int ExistsInPhase;
var() int JumpTo;
var() StructSound ChangeSound;
var   bool bChange;
var() int MaxPhase;
var int PCPresent;

function BeginPlay()
{
    // local ArchaeSoundTrigger AST;
    // local int dl;
    //MaxPhase = 0;

    PCPresent = 0;

    // find the largest phase assignment to actors and
    // make it MaxPhase value limit
    // foreach Allactors(class 'ArchaeSoundTrigger', AST)
    // {
    //   for(dl = 0; dl < 16; dl++)
    //   {
    //       if ( AST.SRelate[dl].Phase > MaxPhase );
    //           MaxPhase = AST.SRelate[dl].Phase;
    //   }
    //   log( "MaxPhase = " $ MaxPhase);
    // }
}
```

```
// Non-linear functions
// Called when something touches the trigger.
function Touch( actor Other )
{
     // Is the actor the correct actor to trigger...
     if( (IsRelevant( Other ) &&
           (PhaseLife == PL_Eternal)) // YES if triggers all time
          || // OR
          (IsRelevant( Other ) &&
          (PhaseLife == PL_ExistsInPhase) && // YES if exists in phase
          (PCPresent == ExistsInPhase) )     // AND present counter matches Existence
       )
     {
          SetTimer(0.5, false);

          if( (bTriggerOnceOnly) && (bChange == true) )
               // Ignore future touches.
               SetCollision(False);
     }
}


function PlayChangeSound()
{
     local float Fvol;
     local float Frad;
     local float Fpitch;

     // if no entry for volume is set to default
     if (ChangeSound.volume == 0) Fvol = 1.0;
     else
          Fvol = ChangeSound.volume;

     // if no entry for radius is set to default of Actor's Collision Radius
     if (ChangeSound.radius == 0) Frad = CollisionRadius;
     else
          Frad = ChangeSound.radius;

     // if no entry for pitch is set to default 1.0
     if (ChangeSound.pitch == 0) Fpitch = 1.0;
     else
          Fpitch = ChangeSound.pitch;

     PlaySound (ChangeSound.sound, SLOT_Misc, Fvol,, Frad, Fpitch);
}


function timer()
{
     local ArchaeSoundTrigger SS;
     local Artefact AA;
     local int i;

     bChange = false;

     switch ( PhaseChange )
     {
               // Triggers all the actors in the level
               case PC_TriggerAll:

                    foreach AllActors( class 'ArchaeSoundTrigger', SS )
                    {
```

```
    // If forward direction i.e. upwards in number within limit
    //-----------------------------------------------------------
    if ( (PhaseDirection == PD_Forward) &&
          (SS.PhaseNow < (MaxPhase-1)) )
    {
         SS.PhaseNow++;
         BroadcastMessage ( "PhaseNow = " $ SS.PhaseNow, false);
         bChange = true;
         PCPresent++;

         for (i = 0; i < SS.numSounds; i++)
         {
              if (SS.SRelate[i].phase == SS.PhaseNow)
              {
                   SS.curSound = i;
                   break;
              }
         }

    } // end of if forwards...


    // down the number scale to zero (earliest phase)
    //-------------------------------------------------
    if ( (PhaseDirection == PD_Backward) &&
               (SS.PhaseNow > 0) )
    {
         SS.PhaseNow--;
         BroadcastMessage ( "PhaseNow = " $ SS.PhaseNow, false);
         bChange = true;
         PCPresent--;

         for (i = 0; i < SS.numSounds; i++)
         {
              if (SS.SRelate[i].phase == SS.PhaseNow)
              {
                   SS.curSound = i;
                   break;
              }
         }
    }  // end of else if backwards...


    // If JumpTo option, phase is now the JumpTo value
    // and if PlayedOnce = true, resets to false.
    //--------------------------------------------------
    if ( (PhaseDirection == PD_JumpTo) &&
          (JumpTo < MaxPhase) && JumpTo > 0 )
    {
         SS.PhaseNow = JumpTo;
         if ( SS.bPlayedOnce == true ) SS.bPlayedOnce = false;
         BroadcastMessage ( "PhaseNow = " $ SS.PhaseNow, false);
         bChange = true;
         PCPresent = JumpTo;

         for (i = 0; i < SS.numSounds; i++)
         {
              if (SS.SRelate[i].phase == SS.PhaseNow)
              {
                   SS.curSound = i;
                   break;
              }
         }
    }  // end of else if jumpto...


}// end of foreach...
```

```
        if (bChange)
        {
            // Adjust the PhaseNow value on each Artefact actor
            //-------------------------------------------------
            foreach AllActors(class 'Artefact', AA)
            {
                if  (PhaseDirection == PD_Backward) AA.APhaseNow--;
                if  (PhaseDirection == PD_Forward)  AA.APhaseNow++;
                if  (PhaseDirection == PD_JumpTo)   AA.APhaseNow = JumpTo;
            }
        }

    break;

    case PC_TriggerSelect:
        if ( Event != '' )
        {
            foreach AllActors( class 'ArchaeSoundTrigger', SS )
            {

                if ( SS.Tag == Event )
                {
                // If forward direction i.e. upwards in number within limit
                //---------------------------------------------------------
                    if ( (PhaseDirection == PD_Forward) &&
                          (SS.PhaseNow < (MaxPhase-1)) )
                    {
                        SS.PhaseNow++;
                // BroadcastMessage ( "PhaseNow = " $ SS.PhaseNow, false);
                        bChange = true;
                        for (i = 0; i < SS.numSounds; i++)
                        {
                            if (SS.SRelate[i].phase == SS.PhaseNow)
                            {
                                SS.curSound = i;
                                break;
                            }
                        }
                    }


                // If backwards direction i.e.
                // down the number scale to zero (earlieststph phase)
                //-------------------------------------------------

                    if ( (PhaseDirection == PD_Backward) &&
                               (SS.PhaseNow > 0) )
                    {
                        SS.PhaseNow--;
                // BroadcastMessage ( "PhaseNow = " $ SS.PhaseNow, false);
                        bChange = true;
                        for (i = 0; i < SS.numSounds; i++)
                        {
                            if (SS.SRelate[i].phase == SS.PhaseNow)
                            {
                                SS.curSound = i;
                                break;
                            }
                        }
                    } //end of if...
```

```
                        // If JumpTo option, phase is now the JumpTo value
                        // and if PlayedOnce = true, resets to false
                        //------------------------------------------------
                            if ( (PhaseDirection == PD_JumpTo) &&
                                 (JumpTo < MaxPhase && JumpTo > 0) )
                            {
                                SS.PhaseNow = JumpTo;
                                if ( SS.bPlayedOnce == true )
                                    SS.bPlayedOnce = false;

                        // BroadcastMessage ( "PhaseNow = " $ SS.PhaseNow, false);
                                bChange = true;
                                for (i = 0; i < SS.numSounds; i++)
                                {
                                    if (SS.SRelate[i].phase == SS.PhaseNow)
                                    {
                                        SS.curSound = i;
                                        break;
                                    }
                                }
                            }


                        }    //end of SS.tag...


                    }    // end of foreach

                    if (bChange)
                    {
                        // Adjust the PhaseNow value on each Artefact actor
                        //-------------------------------------------------
                        foreach AllActors(class 'Artefact', AA)
                        {
                            if (AA.Tag == Event)
                            {
                                if (PhaseDirection == PD_Backward) AA.APhaseNow--;
                                if (PhaseDirection == PD_Forward) AA.APhaseNow++;
                                if (PhaseDirection == PD_JumpTo)
                                    AA.APhaseNow = JumpTo;
                            }
                        }
                    }

                } // end of if Event...
            break;
    } // end of switch

    if ( (ChangeSound.sound != None) && (bChange == true) )
        PlayChangeSound();

}
```

B Extract from 'Garden Monologue' Sound Data Lists

title and number	collision volume (radius & height) R-rollover/OS-oneshot content	contemp /phase	pause secs	time length secs	loop	volume	sound radius	pitch
monologue01	500 X 40 OS							
00	look up look	2/ 0	1.0	1.55	f	2.0	500	1.0
01	the sky is blue	2/ 0	1.0	1.28	f	0.8	500	1.0
02	no sky grey with clouds	2/ 0	1.0	2.37	f	0.8	500	1.0
03	hole	5/ 3	1.0	0.64	f	0.5	500	1.0
04	appearing where they fall	5/ 3	1.0	1.50	f	0.8	500	1.0
05	look up look	5/ 3	1.0	1.55	f	2.0	500	1.0
06	no blue sky with clouds	5/ 3	1.0	1.96	f	0.8	500	1.0
monologue02	300 X 40 OS							
00	you are here	all	2.0	1.35	f	0.3	250	1.0
01	you will be here again	all	2.0	1.81	f	0.3	250	1.0
02	you were here	all	2.0	1.25	f	0.3	250	1.0
03	and said empty	all	2.0	1.66	f	0.3	250	1.0
04	you are here	all	2.0	1.45	f	0.3	250	1.0
monologue03	300 X 40 R							
00	rowan hazel and oak	2/ 0	2.0	2.18	f	0.8	300	1.0
01	lichens and ferns	2/ 0	4.0	1.67	f	0.8	300	1.0
monologue04	200 X 40 R							
00	rainfall	2/ 0	2.0	0.81	f	0.8	300	1.0
01	deposit pool from the quarry gathers	2/ 0	2.0	2.07	f	0.8	300	1.0
monologue05	40 X 40 R							
00	when reference to you is made	2/ 0	3.0	2.01	f	1.0	100	1.0
01	that is the construct	2/ 0	1.5	1.54	f	1.0	100	1.0
monologue06	40 X 40 OS							
00	first step forward	2/ 0	0.5	1.34	f	0.5	120	1.0
monologue07	300 X 40 R							
00	west wall	2/ 0	0.5	0.93	f	0.8	100	1.0
01	over the wall	2/ 0	1.0	1.20	f	0.8	100	1.0
02	overgrown bracken and weeds	2/ 0	1.0	2.17	f	0.8	100	1.0
monologue08	80 X 40 R							
00	the roses	2/ 0	1.0	0.78	f	1.0	100	1.0
01	fixing location	3/ 1	1.0	1.64	f	0.6	100	1.0
02	the roses look just as they do	6/ 4	1.0	1.68	f	1.0	100	1.0
03	filling the gap	6/ 4	3.0	1.36	f	0.6	100	1.0
04	rose	8/ 6	0.5	0.66	f	0.8	80	1.0
05	maybe white rose	8/ 6	3.0	1.32	f	0.8	80	1.0
06	rose red newer	8/ 6	3.0	1.34	f	0.8	80	1.0

monologue09	2000 X 500 OS							
00	the hills	all	1.0	1.20	f	0.7	2000>	1.0
01	ochil hills	all	4.0	1.03	f	2.0	2000>	1.0
02	stretch back 8 miles and along 12	all	1.5	2.93	f	1.5	2000>	1.0
&&								
monologue10	200 X 80 R							
00	the hills	2/ 0	0.5	1.20	f	2.0	200	1.0
01	reminders	2/ 0	0.5	1.02	f	0.5	200	1.0
02	holders of memories	3/ 1	0.5	1.46	f	2.0	200	1.0
03	markers fixing location	4/ 2	0.5	1.90	f	1.0	200	1.0
04	both geographical	4/ 2	2.0	1.51	f	1.0	200	1.0
05	and psychological	4/ 2	1.5	1.25	f	1.0	200	1.0
monologue11	120 X 200 R							
00	all have the possibility to presume the memory of one day	2/ 0	1.0	4.13	f	1.0	120	1.0
01	and so take liberty	2/ 0	2.5	1.79	f	0.4	100	1.0
02	remember the sounds you hear	2/ 0	2.0	2.06	f	1.0	100	1.0
03	of your footsteps are artificial	2/ 0	1.5	3.03	f	0.4	120	1.0
04	all sound other than the footsteps has been added	2/ 0	4.0	3.64	f	200	150	1.0
monologue12	150 X 200							
00	garden surrounded	2/ 0	0.5	1.14	f	1.0	200	1.0
01	enclosed by old wall	2/ 0	1.5	1.50	f	0.8	200	1.0
02	wooden log seat arched above	2/ 0	2.0	2.50	f	0.6	100	1.0
03	covers unused door	2/ 0	2.5	2.09	f	0.5	100	1.0
monologue13	400 X 200 OS							
00	grass grows here	2/ 0	4.0	1.35	T	128	150	1.0
01	grow here	5/ 3	4.0	0.94	T	200	150	1.0
02	used to be grass here	7/ 5	4.0	1.53	T	128	150	1.0
&&								
monologue14	250 X 200 OS							
00	grass grows here	2/ 0	4.0	1.52	T	200	150	1.0
01	grow here	5/ 3	0.5	1.04	f	1.0	150	1.0
02	grow while sittling down	5/ 3	2.0	1.65	T	128	150	1.0
03	keep standing up and sitting down here	5/ 3	0.5	2.34	f	1.0	150	1.0
04	face different direction each time	5/ 3	2.0	2.30	T	128	150	1.0
05	remove that which burdens	5/ 3	0.5	1.86	f	1.0	150	1.0