Great Expectations:

Designing Game Environments that Operate Against Player Schema, and the Influence of this on Perceived Levels of Fear

Programme Definition Document

© Peter Howell, 2011

Supervisor: Dr. Dan Pinchbeck

Table of Contents

I. Introduction

Within the games industry, the umbrella term of *Level Design* encompasses a wide range of skills and these skills are themselves from varying backgrounds. Technical skills are required to produce the art assets and scripts, architectural knowledge is required to produce convincing, believable game environments and an understanding of more traditional topics, such as colour theory, is required to give a game level maximum impact and provide the player with an enjoyable experience.

An understanding of human psychology and, specifically, player psychology are also highly important, yet they are somewhat less obvious candidates when asking the question *what skills are required of a level designer?*

It is perhaps more apparent to see psychology at work in level design by looking at a specific type of game – the horror game, or as it is more usually called, the survival horror game. This is a genre that has been in existence in some form or another since the very earliest games were being developed, with the earliest considered horror game being *Haunted House* (Atari, 1981) on the *Atari 2600* system. In the current generation of gaming, the genre has many well known franchises such as *Resident Evil* (Capcom, 1996-2010) *Silent Hill* (Konami, 1999-2009) and *Fatal Frame* (Tecmo, 2002-2008). The genre is also very much in use in independent development with *Penumbra* (Frictional Games, 2007), *Amnesia: The Dark Descent* (Frictional Games, 2010), *Hydrophobia* (Dark Energy Digital, 2010) and *The Path* (Tale of Tales, 2009) all receiving predominantly positive critical response.

These games all use an array of techniques designed to elicit a 'fearful' response from players. Fear creates psychological stress, which in turn causes the body to secrete a high dosage of adrenaline giving "a ... rush that is as addicting as any drug" (Koltz, 2004). The adrenalin rush is the pleasurable physiological response which helps engage players and keep them playing. As Weaver puts it in a developer commentary, "Fear is arousing…arousal is attractive [to audiences]" ("Irrational Behaviour Episode 5: What Are We Afraid Of", Irrational Games, 2010).

However, 'fear' in and of itself is far too broad a category for what is a complex brain function. There are many different types of fear; paranoia, anxiety, terror, dread and shock, as well as numerous variants on these, and each provokes a subtly different psychological and physiological response from the body. This poses some interesting questions.

Are players particularly enticed by a specific *type* of fear? For example, do some players enjoy a more visceral, immediate, 'shock' type of scare whilst others prefer a more thought provoking, psychological feeling of dread or anxiety that builds up slowly over time? Do players even consciously think about these differences or simply consider themselves 'frightened' in the most general sense? (Howell, 2010b).

Perhaps the most relevant question with regard to creating commercially successful games is how far can an intensely 'fearful' atmosphere carry a game? For example, will a player continue to play a game that perhaps is lacking in other areas of gameplay simply because they are enjoying the experience of being scared? Comparing critical reviews to players' reviews, there is some evidence that this could be the case. For example, critical reviews of *Ju-on: The Grudge* (XSeed Games, 2009) were widely negative. The game has an average critic score of 39% yet has an average user score of 60% (Metacritic, 2010), with some users rating it at 100%. Many of the user reviews cite the horror elements as being the best aspect of the game, with one review specifically stating that "Although the gameplay and controls are a bit dull, the scares and the graphics is what makes up for it" ("Axel", 2009). Whilst user reviews cannot be cited as 'scholarly' references, they are an important source of information when referring to video games. It is, after all, the end user that is the important audience for developers.

The intent of this project is to explore the techniques used by designers to generate fear and the tools, skills and methods used to implement them successfully in a game to elicit the desired response from the player.

It will endeavour to initially determine the main reasons that players enjoy these types of games through primary research as well as comprehensive reading and analysis of player responses to particular games within the horror genre. Sources of information available from developers and post-mortem documents will also be used to build up a picture of the development side of horror game design.

The project will utilise the body of research to inform the creation of an artefact, in the form of a custom level design that will incorporate the most powerful and successful techniques for creating fearful responses from players. Created using the *Unreal Development Kit* (Epic Games, 2009), this artefact will then be play-tested and evaluated by a number of players to analyse its design and whether it succeeds in providing an entertaining, scary experience.

II. Project Context

The worth of studying level design with relation to these games and their impact on the psychological state of the player is two-fold – personal skills development and relevance to the current state of the industry.

From a skills development perspective, this project will enable, through the creation of an artefact stemming from initial research and analysis, the learning and improvement of a wide range of abilities. Creation of a level, using proprietary tools will expand on knowledge of more mainstream software used in previous work. The assets required will also need to be produced using an industry standard pipeline which will call for a range of technical abilities, such as low and high polygon modelling, various texturing techniques such as normal and parallax occlusion mapping and the smooth implementation of the assets into the toolset. In a more general sense, skills in specific software packages will be able to be continually improved, such as *Photoshop* and *3D Studio Max.*

In addition to the technical skills that will be improved upon, a more detailed and wide reaching knowledge of level design theory will be developed. This project will enable the investigation and understanding of areas of level design complimentary to those already studied in a previous research project (Howell, 2010a) such as the importance of lighting and ambience.

This improved skill set and broader knowledge will improve employability and ability to work within industry; however the specific knowledge gained with regard to manipulating player responses to fear and frightening situations will be particularly beneficial. The franchises mentioned previously have all achieved very large sales figures with the original *Resident Evil* (Capcom, 1996) reaching worldwide sales of 5.05 million units (VGChartz, 2010). It is clearly a genre that is rewarding to have a deep understanding of. In 2006, Castle (2006) cited by Perron (2009) stated that “today, horror is commercially viable… horror [is] a virtual ‘sure sell’ for youth oriented films, television programmes and video games”. This statement can still be seen to be true four years later, with successes such as the Saw franchise making impressive capital.

The genre has broad appeal and elements of it are easily transferable to other genres, such as first-person-shooters and action-adventures, which can all benefit from sections of game play inspired by horror. For example, in the latter stages of *Uncharted: Drake's Fortune* (Naughty Dog, 2007) the player must navigate an abandoned Nazi bunker infested with mutated Spanish looters. The atmosphere is very reminiscent of the research laboratories in *Resident Evil*, yet this game otherwise could not be described as containing any other elements suggestive of a horror theme.

This possible cross-application of techniques between genres means that being able to work successfully with the techniques explored in this project will have benefits throughout a career in the industry.

III. Research Corpus

i. The Elements of Fear

As previously stated, fear in and of itself is comprised of numerous categories. For the purposes of this study it is necessary to define with specificity the emotions and responses being investigated. The very naming of the survival horror genre of games is itself misleading. If one considers what is being defined by the term ‘horror’ and compares that to what is most often presented in these games there is clearly a mismatch between description and gameplay. The supposed horror presented in many, although not all, survival horror games could more accurately be described as terror. Varma (1966, cited in Fahraeus, 2010) states: “The difference between Terror and Horror is the difference between awful apprehension and sickening realization: between the smell of death and stumbling against a corpse”. Many of these games create their gameplay through suspense and apprehension; building a sense of impending threat rather than presenting that threat directly to the player.

One could even argue that the true ‘horror’ of these games is only truly realised in the final ‘reveal’; when the culmination of many hours of suspenseful gameplay and ambiguity is brought together into the end of game boss fight and any remaining narrative questions are answered. “Terror is the possibility of the horrible; ...it is characterised by "obscurity" or indeterminacy where horror is the realization of that which is dreaded” (Radcliffe, 1826 cited in Fahraeus, 2010).

This is the differentiation that will be applied to this study: the *terror* of suspense and of not being able to predict, and the *horror* of a visceral shock or disgusting reveal of an enemy or monster. *Fear,* as an emotional response, encompasses both of these specificities.

ii. The Reality of Fear

When discussing the idea of 'fear' in its most general sense, one must make a differentiation between fears experienced in a real situation and fears experienced through a medium such as video games, or any other type of media for that matter.

The reason for this is that in a real life situation, fear is often caused because there is a threat to the person's wellbeing. There is never a *direct* threat to the player of a game, and so fear must be generated by giving the player something they are implicitly afraid of losing. This could be as literal and tangible in the game world as health, ammunition or other vital resources. It could merely be a threat of losing *progress* – for example, by not allowing the player to save their game before a boss fight – by failing, the player knows they will have to replay a section they had already completed.

Wells, of Irrational Games, discussing *System Shock 2* (Irrational Games & Looking Glass Studios, 1999) states that "If [the player] knows a game can't hurt [them], you have to hurt the player within the game… People agreed that *System Shock 2* was pretty scary [because] you only had two bullets at any given time" ("Irrational Behaviour Episode 5: What Are We Afraid Of", Irrational Games, 2010). In this game, the player was forced to consider every bullet fired, as ammunition was incredibly scarce and weapons broke easily. This, of course lead to the player needing to make a 'fight or flight' decision when faced with combat situations lending a tense and stressful edge to game play. A similar mechanism can be seen in *Amnesia: The Dark Descent* in which the player has two key resources; tinderboxes and lantern oil. Both help illuminate areas (by lighting candles and the player's oil lamp respectively), which in turn helps prevent the player avatar from losing his sanity. They too are scarce and must be used wisely. However, it could be argued that it is the avatar's *sanity* – which is a measurable resource – is that which the player may be most afraid of losing as it makes them more vulnerable to enemy attacks.

iii. The Explicit versus the Implicit

Having differentiated between real life and media, there is yet one aspect of fear that permeates all guises that it may present itself in; implicit and explicit fear.

Phobias and phobic responses have been played on by media producers countless times. Probably the most referenced phobia is a phobia of spiders. In film, this has been seized upon with films such as *Arachnophobia* (Marshall, 1990) and books, such as *Harry Potter and the Chamber of Secrets* (Rowling, 1998). Similarly games, including *Resident Evil* often incorporate spiders or other arachnoid enemies. An article written by staff at gaming magazine *Edge* suggests that "[relying] on learned fears: personal fears and established iconography [is not a good method of evoking a fearful response]" ("How to Make Fear", 2006, para. 7). It could be argued that – while appealing directly to phobic responses could be considered a 'cheap' method of eliciting fear – it may still have merit when used with brevity in appropriate situations such as those requiring an element of shock or surprise.

Phobic responses are an *explicit* method of generating fear; the viewer or player can see the thing that is making them scared. However, it is the *implicit* methods which are cited my many designers and researchers as being the most effective. These methods generate fear by preventing the subject from seeing the source of their fear. As Rouse (n.d, cited in Perron, 2009) writes: "In Horror, the way the audience fills in the blanks will be far more disturbing than anything the writer can come up with". Games are able to use, with great effectiveness, methods which 'get under a player's skin'; techniques that cause them a sense of *unease* rather than out and out terror. Other media is capable of this also, however the intrinsically interactive nature of the gaming medium makes it a particularly useful platform.

Take, for example, a highly cliché horror convention; darkness. Few works of horror exist in which darkness is not used as a key tool for creating fear. "Darkness limits our perception, creating space for tension and doubt to flourish" (Pinchbeck, n.d, cited in Perron, 2009). The audience's imaginations create their own fear from this limitation.

Of course the same effect can be achieved by different means; the heavy fog in *Silent Hill* (Konami, 1999) similarly limits perception without resorting to darkness throughout the game, although other areas use darkness also.

Other forms of media lead the audience through an environment regardless of how they feel towards it, but in a game, the player must force themselves to continue forward even if they really do not want to, as it is the only way that they can make progress.

iv. The Terror of the Unnatural or Impure

> A sense of decay – especially of something apparently comforting and normal, or something we imagine to be pure, such as childhood or the tenderness between two people – can be profoundly disturbing, on a level for which people often have no words. (“How to Make Fear”, 2006, para. 15)

Presenting something in a different context to that in which an audience may expect to see it can be highly effective at creating the aforementioned sense of unease. Take for example, *Dead Space* (EA Redwood Shores, 2008); enemies known as Lurkers are formed from infected and mutated human babies. Similarly, in *Max Payne* (Remedy Entertainment, 2001) a drug-induced dream sequence sees the player having to navigate a surreal environment, following a trail of blood accompanied by the sound of a crying baby in the distance, which gets closer as the player progresses. Both of these examples focus on the corruption of childhood and innocence, but clearly show how alienating a familiar or ‘pure’ idea or object can create discomfort and unease in an audience’s mind. There is a very apparent category error, presenting something an audience is very much used to in a distorted or dark fashion.

It is even possible to portray in a twisted manner something which could already be described as such. Take for example a classic horror monster; the zombie. Zombies have a sense of interstitiality due to their appearance in different forms across all periods of horror literature and media reaching back to their origins in voodoo practices. Because they are so well defined in the genre, to retain their ability to scare they must be presented in a new, unpredictable way. A good example of this is the Crimson Head zombie mutations in *Resident Evil*. The slow, lumbering enemies that players are accustomed to are suddenly much quicker and much more powerful. This forces the player to adjust their tactics, meaning they cannot become 'comfortable' with their ability to cope with the game's challenges; feeling comfortable within a game is something which can destroy any sense of terror a player may have.

A more overarching example of the idea of 'corruption' can be seen in *Silent Hill* which has two realities: the real world and an alternate dimension known as 'Nowhere'. Nowhere is a hellish, transfigured version of the real world intended to portray the horrors and demons within the protagonist's psyche. It is a corruption of the entire initial game world.

v. Knowing the Avatar; Not Knowing the Rules

A pilot study carried out prior to this project (Howell, 2010b) asked participants for their feedback on their experience of playing the game *Amnesia: The Dark Descent*. Two key things that were noted by participants were having an empathy or understanding of the character they were playing as, and knowing very little about the game and how the game worked.

These two aspects are in actuality inextricably interwoven. A simple way to show this is to look at a particular theatrical technique – dramatic irony. A dramatically ironic situation is created when the audience (or player) knows more than the character/s on screen (or in game). This is often the case in games; mechanisms such as maps showing the location of enemies, or pointers directing the player to their next objective are, unless presented as a diegetic part of the game world, additional information that the character would not have access to. The result is that the player knows the game's rules, and loses a level of empathy or identification with their avatar; it becomes a process of the player manipulating an on-screen avatar, as opposed to the player *being* that avatar.

Amnesia: The Dark Descent ensures that players are not given any more information than their on-screen presence. No back-story outside of what the avatar knows is presented to the player, and it unfolds in a completely diegetic manner (through flashbacks, journals, memos and the like). As stated by a participant in this pilot study, this "make[s] you more aware and [makes] you think as the character would" (Howell, 2010b). Working on the same level as the avatar, being in the same frame of mind, heightens both immersion and tension, leading to more intense feelings of fear. These feelings are then further compounded by not knowing what the game is going to present the player with next; a classical fear of the unknown.

vi. Creating a Universal 'Unknown'; Designing against Tradition

A fear of the unknown may be a widely accepted horror staple, but it suffers from being highly ambiguous, and ambiguity is a problem when attempting to define mechanisms and design techniques. By looking at this idea through the theory of *schemas* however, it is possible to define the 'unknown' in a more tangible sense. As defined by Bartlett (1932, cited in Hayes, 1998) a schema is "an active organisation of past reactions, or of past experiences, which must always be supposed to be operating in any well-adapted organismic response"; in relation to players, this means each person playing a game will have expectations of it, based on their previous experience playing similar games.

Anything that falls into these expectations is a 'known'; the player expected it, and was duly presented with it. To create an 'unknown' that is applicable to a broad range of players, a game must make use of design tools and game mechanisms in ways that do not conform to generic schemas. Presenting the player with something they were not prepared for, or presenting something they *were* prepared for in a different, unexpected way, can heighten uneasiness and suspense.

> We don't have to take these given golden rules about the way things work; they've developed because that's the way the market has developed. (Pinchbeck, 2010)

While a universal 'unknown' may not be possible – it is unlikely that every single player will have the same basic set of schemas – it is very possible to purposely design in a way that goes against generic tradition for the type of game being created. This will then have the effect of unnerving or disorienting the majority of players.

vii. Creating a 'Toolkit' for Fear

Whilst the idea that a set of techniques can be theorised and then used to equal effect throughout all horror games is naïve – the placement and context of a technique within a game world will have an effect on how it is perceived by the player – it is clear that there are defined methods and considerations that are widely accepted as being key to creating fear successfully. Those that have been defined thus far by this research – making the player fear loss of something, balancing explicit and implicit threats to keep the player on edge, and warping their perception of 'safe' or 'pure' themes into something unnatural and disturbing – are some of the most powerful and widely used. Designing against player expectations is similarly a very powerful technique.

This is not to say that less methodical approaches will not work well in the proper circumstances; there are times when having an enemy jump out of a darkened corner directly behind the player will be a highly scary and effective mechanism. Perron (2004, cited in Roux-Girard, n.d) sums up well the use of such techniques:

> To trigger sudden events is undoubtedly one of the most basic techniques used to scare someone. However, because the effect is considered easy to achieve, it is often labelled as a cheap approach…compared with another more valued one: suspense.

This approach will become cheapened and less effective the more frequently it is used; or example, *Doom 3* (id Software, 2004) was highly criticised for making incredibly heavy use of 'monster-closets' from which to spring attacks on the unsuspecting player.

> They're quite literally waiting for the player to pass by, setting off a scripted trigger, then their door opens automagically without noise and they shoot you in the back…made me jump back in the seat and I needed to take a break just two hours into the game. After eight hours of that, however, the feeling of fear is replaced with irritation. (Wojnarowicz, 2004)

So whilst certainly not the only method, the general rule of successfully scaring an audience is to make them scare or disturb *themselves* through their own imaginations, not through something specific that the designer has created.

The design tools at a designer's disposal for implementing fear in games vary depending on the technology available and the platform being developed for. To give a general overview however, the key aspects are lighting, shadows, music, on-screen and off-screen sound effects and physical level structure. It is possible to clarify the uses of these tools by loosely grouping them into two main categories: Indirect Tools and Direct Tools.

a. Indirect Tools

The design mechanisms that fit into this category are completely or predominantly devoid of direct player interaction. They create the game space inhabited by the player and give it meaning. For example, light and shadow play an obvious role in horror games as they hide sections of the environment, deadly monsters or life-saving ammunition and powerups. They also serve to limit the player's forward vision, keeping them alert and tense. However, this lighting is passive (if one does not consider heavily dynamic lighting) and thus is not actively engaging the player.

In actuality, one could consider a lighting artist's main role when working on a horror title as being responsible for the placement of *darkness* rather than the placement of light. As Burke (1998 [1757], cited in Niedenthal, n.d) states; "darkness itself on other occasions is known by experience to have a greater effect on the passions than light". The absence of light could be argued to be far more influential than its presence.

A game's soundtrack is the second key indirect tool; whilst for the majority of cases it is not a diegetic part of the virtual game space it has a great influence over a player's mindset during game play.

When considering music however, one must not underestimate the power of silence. Admittedly, the presence of music can increase immersion, heighten tension and put a player 'on edge'. For example, instrumental pieces utilising a combination of sharp and flat notes playing in relative discord can heighten the unease of a situation, with the inharmonious nature of the sound being transferred to the player's mental state. The use of silence refers back, once again, to maintaining implicitness. The silence may suggest to a player that they need to be listening for nearby threats, or that a threat will appear very soon. "The silence…puts the player on edge…increasing the expectation that danger will soon appear. The appearance of the danger is, therefore, heightened in intensity by way of its sudden intrusion into silence" (Whalen, 2004, cited in Roux-Girard, n.d). Silence can therefore also be used to mislead a player into perceiving a non-existent threat, keeping them alert and engaged.

b. Direct Tools

Tools in this category are either directly interacted with by the player, have a tangible existence in the game space or are triggered by the player's own actions.

Music is predominantly ambient, but can be triggered by player progress to specifically elicit certain emotional responses, such as triggering 'panic' music when an enemy gives chase.

The other element of the soundtrack – sound effects – is somewhat different to music in a game. Their absence would cause the game to feel unrealistic meaning any sense of fear is in danger of feeling fake. Their nature and perceived origin are highly effective tools for a designer to utilise. The majority of sound effects are diegetic, being able to be attributed to in-world objects such as machinery and creatures. Again they can be player triggered to provide a shock in their own right or to accompany a visual reveal in order to intensify it.

Technological advances mean that modern gaming is able to draw on surround sound to deliver a highly immersive audio landscape for players to experience. The relevance of off-screen sound was noted by Roux-Girard (n.d, cited in Perron, 2009) discussing *Alone in the Dark: The New Nightmare* (Darkworks, 2001) who states that "[the game] makes great use of off-screen possibilities. Sometimes used to give prior knowledge of incoming threats, these off-screen sounds contribute towards building quite a deal of apprehension". Off-screen sounds were effective even prior to the advent of surround sound systems and thus can only become more influential with their inclusion.

The structure of a level and the way that it is played through by the player can themselves be used to create fear. With specific regard to games played from a first-person perspective which is the focus of this study, directing the player's vision (or indeed purposely limiting it) can lead to heightened terror or suspense. For example, if a player is for whatever reason forced to proceed down a narrow hallway, the designer can be relatively certain that their attention will be focused on the end of the hallway. The designer is then free to fill that space with something 'fearful'; be that a shadowy figure of an enemy, or an unusual event, such as candles inexplicably being extinguished. Level structure can be considered an 'event tool' as the player *must* navigate it in order to make progress and it can then serve to dictate more explicit events – such as extinguishing candles and triggering enemies as mentioned above.

The process of leading a player to face these 'fear bottlenecks' in itself serves to elicit terror, even if the designer does not actually include anything in that space at the end of the hallway. In many cases, if a player knows that the only possible way to proceed is through the dark, foreboding corridor they will do an excellent job of scaring themselves by imagining what horror may attack them from the darkness. If nothing happens, that tension is not fully released, so the player must carry it with them further through the game, heightening their suspense and unease. The longer the designer can build up that tension without providing a way for the player to release it, the greater its effect. Of course, building it up for too long will eventually become tiresome – both mentally and physically – for the player, so balance is essential.

viii. The Limit of Deconstructing Design Tools

The tools outlined in the previous two sections are those which a level designer has direct manipulative control over. However, games are by far much more complex than a collection of audiovisual signposts placed within an interesting virtual environment. A game's narrative, for example – a tool that cannot easily be classified as *Direct* or Indirect – has an influence on how the game is read by a player and in turn how that player plays it. A game's artistic style, will similarly

Ideally, from a scientific standpoint, these broad groupings would be further distilled and refined in order to create a clear set of specific mechanisms which could be implemented and tested individually. However, a scientific approach such as this is not necessarily the most practical manner of looking at a topic as intrinsically complex and interdependent as fear.

A structuralistic approach to game design, whereby mechanisms and methods are reduced down to the smallest component parts, is an approach which works well when looking at technical aspects of that game and its environment design. "Game design is extremely structuralist in this sense; all spaces are built around play-testing and studying how people respond to them" (Jeffries, 2010). However, when one attempts to apply a structuralistic framework to game design in relation to its creation of *emotion* or player *response,* the process is faced with similar problems as faced by other fields such as psychology and literature when applying structuralism; namely that there is a finite limit the deconstruction can reach before the process becomes moot.

In the case of fear and the process of eliciting it from a player, the primary problem is that which is being studied is *perception*. This, by its very nature is subjective. "The designer is intrinsically aware of everything about the reality that they're in and how it works. They perceive the space in a totally different manner from a person who is playing it" (Jeffries, 2010). For a designer to break the process and elements of game design down is both impractical and unnecessary – it is the relationship and interplay between all of the various parts of a game's design and the 'whole' that they create that is the key concern as this is what a player will experience.

This realisation can be further supported by previous research that attempted to study the effect of game world lighting on a player's decision making process (Howell, 2010a). This study concluded that:

> ...while lighting has the potential to enhance entertainment value by making more intelligent, diverse use of it, it is first necessary to determine its place with relation to other game world factors and also in relation to the expectations of the player... The interaction between a player's preconceptions and what the game offers seems to result in the experience the player takes away from playing, rather than attempting to separate any one feature over another. (Howell, 2010a)

This evidence suggests that the current study would benefit from taking a far more functionalistic view of game design, purposely aiming to create a game environment that is strengthened by the interplay between its component parts in order to elicit both terror and horror in the player. The reaction of players in relation to this interplay of components would then become the focal point of the research, looking at the success or failure of taking functionalism and applying its rules to game design.

ix. Tools for an Original Design for the Testing of Theoretical Mechanisms of Fear in a Functionalistic Design

This project intends to test the theories outlined through the creation of an original game level design which utilises them and evaluating player responses. This design will be rooted in functionalistic, holistic theory and incorporate the mechanisms outlined previously in this research corpus.

To this end, it was necessary to select appropriate tools and technologies to realise an effective horror environment. A vast selection of possible game engines and toolkits are available that are all, to some extent, viable options for this project; however many do not comply with the specific design requirements or prove too complex to make completion of the work in the given time period possible.

An initial shortlist of possible technologies was evaluated to determine the most plausible options; this list comprised the *Unreal Development Kit* or *UDK,* the *Source Engine* (Valve Corporation, 2010), *Unity 3* (Unity Technologies, 2010) and the *Amnesia Toolkit*.

Using *UDK* would have been advantageous for numerous reasons. A lot of experience has already been gained with the tools, meaning the learning curve before being able to create high quality work would be much shorter than with other software. The *UDK* is also highly user-friendly with regard to the importing and use of custom assets, as well as having a high quality lighting engine capable of dynamic real-time lighting which would be beneficial in the horror genre. The BSP (Binary Space Partition) editor is also extremely useful for blocking out full levels in very short time frames to test initial game flow and level structure.

Its limitations however are that, by default the game play presented within *UDK* is that of fast-paced, action oriented first-person-shooter (FPS) games. Whilst it is entirely possible to create other game play styles with this engine (*Tom Clancy's EndWar* (Ubisoft Shanghai, 2008) for example utilises it to create a Real Time Strategy game), the level of custom scripting required would be high. The two key adjustments that would need to be made to make it a viable tool for this project would be altering the speed of the player's movement and using a 'filter' effect on the camera to make the game look 'rougher' – the *Unreal*

Engine tends to make models look 'shiny' and exaggerated, which is not the graphical style that is being sought for in this project.

The *Source Engine* is more viable in the sense that it would take less intensive scripting in order to convert the style of game play to suit a horror situation. The engine is clearly capable of producing the desired environmental results, such as *Left 4 Dead*'s (Turtle Rock Studios & Valve South, 2008) dark, zombie-infested locales. The default game play the engine is designed for is again FPS, however the speed of play is less of that presented by *UDK.*

However, with no prior knowledge of the asset pipeline for the engine, and only minimal knowledge of the editing tools, there would be an intense period of learning before production of the artefact could begin. Whilst the final result could be high quality the risk of not completing the project within the time frame allowed is too great for this to be used as a basis.

Unity 3, being a development environment rather than a pre-built engine would allow for the greatest flexibility. Allowing the creation of any game using imported custom assets and able to be programmed using *JavaScript* or *C#*, amongst others, it is a very powerful software kit.

Due to this initial 'raw' state however, too much scripting would be required along with a very high number of custom assets that would need to be created. Therefore, whilst possibly able to produce a very particular, defined end result it is again not viable within the timeframe.

The *Amnesia Toolkit* is a set of proprietary tools developed by Frictional Games to produce *Amnesia: The Dark Descent* and have been made publically available for mod creation. This does therefore limit the game play style to the slow paced, non-combat first person exploration presented in this game. However this style is a very good basis upon which to design, test and evaluate specific mechanisms for research.

The engine allows the use of imported assets but also provides the assets created by Frictional Games, meaning that is may be possible to reuse some of these in order to speed up the production pipeline; for example, a level can be quickly blocked out using these assets before a greater time investment is made to produce original assets for the final level design.

The disadvantage of this engine is that it is still being developed, meaning the development environment may not be as stable as more mature engines. The software will also need to be learned, as no prior knowledge is held. Lastly, for the purposes of creating a full custom level, a large number of custom assets will be required compared to what would be required using other engines. The assets supplied were created for specific uses in the original game and may not be as flexible as those created by, for example, Epic Games which were intended to be reusable and multipurpose.

Given the advantages and disadvantages of all these tools, some further research into *UDK* has been carried out to identify the extent of scripting required to make the changes needed for this project. Implementing them should be feasible with minimal assistance required, and thus *UDK* is the proposed toolset for continuing with this project.

IV. Skills Forecast

In order to successfully complete this project it will be necessary to improve the understanding held in some key areas, as well as learn new technical, organisational and time management skills. This chapter will outline these areas, and the resolution of these learning requirements will be covered in the next chapter.

i. Technical Skills

The project will require a substantial artefact to be produced, to industry standard, with the aim of being assessed and critiqued by a range of players. Whilst many of the required technical skills have been acquired throughout previous academic and personal experience, there are still areas that will need to be explored in more depth. The creation of a game level will require original art assets to be produced, using a combination of skills.

3D Modelling is one of the key skills that will be required, and is something that a lot of prior experience has already been gained in. A large body of modelling work has been completed primarily focussing on technical or mechanical subject matter, as well as a small number of assets specifically designed for in-game real time use. However, of this prior experience much of it has been for projects with more lenient restrictions on polygon counts. Working for game engine assets that will be used in a large real time environment will require much more attention to detail when designing topology and constructing meshes to ensure optimisation.

Leading directly on from this requirement, normal maps will need to be used in order to achieve industry standard results with regards to optimisation and appearance. This is again an area in which previous experience is held, although there is scope for improving in this area. The correct method of baking normal data from high detail to low detail models in *3D Studio Max* will need to be learned as this is a process which has not been used before. Knowledge of normal map creation and application tools will need to be improved upon, which in turn will require more advanced knowledge of methods of correctly applying UV unwraps to models.

To achieve an industry standard result, 2D texture painting skills in tools such as *Photoshop* will need to be improved upon; however, this is an ongoing requirement as an artist in order to continually improve and as such is not a specific requirement for this project as much as it is a career requirement.

In addition to the skills required for the artistic element of the artefact, some basic scripting will also be required. An understanding of level scripting is already held, due to creation of previous artefacts using level editing toolsets such as *UnrealEd*, as well as designing scenarios for a number of 2D role-playing-games. However, more advanced or complex scripts may be necessary to achieve the desired effects or implement particular game mechanisms for this project.

The individual editors within UDK function in a predominantly similar way to older versions of the Unreal Editor such as UnrealEd 3 (Epic Games & Digital Extremes, 2008) which is advantageous as an advanced understanding of this editor is already held. However, the node based material editor is something that is still being learned and this learning process will be aided through working on this project. Similarly, the process of importing assets into, specifically, *UDK* is not a process that has been carried out often previously and will need to be re-learned at the start of the development period.

ii. Project Planning and Management Skills

With previous experience of working on large projects, both personal and as part of a team in the position of project manager, much has already been learned with regard to organising and managing them.

Personal projects have included an academic dissertation which required a very similar organisational and scheduling structure as this proposed project and a project which required a sizeable end product to be presented to a client within a strict time frame.

Personal time management skills have been consistently improved throughout academic life and proven by the ability to take on additional work and projects outside of a full-time degree course, such as working for external clients including the United States Army.

Whilst this body of experience will be highly beneficial in structuring this project, there is definite scope for improvement with regard to documenting the organisation and management of projects. In past projects much of the project documentation was not formalised – such as in the form of a Gannt Chart or Critical Path Analysis – and this caused issues when attempting to analyse the success of the project plan both during and after its completion.

This will be a particularly important skill to improve, not just for the purpose of self analysis during this project but also for operating as part of a professional team when working in industry. Documentation is a key aspect in the smooth operation of a team as it aids communication and understanding throughout.

V. Learning Plan

Having identified the areas in which learning or further learning is required in the previous section, this section will identify the resources available to accomplish these learning objectives and outline the projected time frames required to complete them.

i. Asset Creation for Real Time Engines

The key skills outlined previously in this area can all be considered part of the wider topic of the *game asset creation pipeline*. This topic is covered at length by the second year BSc Computer Games Technology unit 'Commercial Asset Production and Environments for Real Time'. This is a yearlong unit, with the first semester focussing on character creation for games, from a low polygon model, through *ZBrush* to generate a high polygon model and finally taking this high polygon data and baking it onto the low polygon model.

While this process is being carried out on characters, the same process is applicable for creating normal map data for static meshes to go into a level. Due to this being taught during the first semester, it has already proved useful in highlighting considerations that need to be made at the start of the workflow, such as laying out topology correctly to avoid problems when baking out normal data. This unit will continue to be attended (including both lectures and workshops) in order to gain as much information about this topic as possible. The second semester focuses on environment creation for games – this topic will be taught at approximately the same time as this project will enter the development phase and so any problems that arise with regard to environment development pipelines can likely be discussed or solved during the workshops of this unit.

ii. Unreal Development Kit

While this is not a topic that no prior knowledge is held on, it is a very complex toolset containing a vast amount of functionality; thus, there are always going to be skills and functions that can be learned.

For the purpose of this project there will be specific areas of the toolset to which more attention will need to be paid, such as the advanced features of static and dynamic lighting and the engine's *LightMass* light mapping engine. This is a new lighting engine built for the latest version of the *Unreal Engine* designed to simulate real light behaviour much closer than previous iterations of the engine; for example, bounced light and colour transfer. This is a key tool in creating atmospheric environments – obviously an important feature of any 'horror'- based game.

There are two other important aspects of *UDK* that will need to be understood to a high level to achieve industry standard results – the node-based material editor, and the *Kismet* scripting system.

The material editor is used to create advanced materials and shaders using a variety of texture maps and other variables. This editor is designed to simplify the shader creation process, but a solid understanding of shader technology will be needed to take full advantage of it.

The *Kismet* scripting system is a graphical programming system that allows level designers to script in-game events and agent behaviour through a visual node network. Again this is a system intended to simplify the design process but a good knowledge of traditional programming is required to think in terms of routines and sub-routines in order to create working code systems.

The resources that will be used to assist in learning and developing skills in *UDK* consist of a variety of video tutorials from reputed sources such as *3DBuzz* and *Eat3D*. These videos offer in-depth knowledge on specific topics within the engine. In addition to these, there is a *UDK* workshop which runs within the University which will be attended as it will offer the opportunity to work hands-on with the toolset alongside others with experience. This is an opportunity to discuss issues and find solutions much more quickly than would probably be possible working alone. These workshops will be running throughout the semester so can serve as a constant learning resource as they are required.

iii. Personal Development and Professionalism

The main learning resource for this section has already been utilised. The 'Professional Development' unit on the MSc Computer Games Technology course included two lectures focussed on project planning which have provided information on methods such as critical path analysis. This information has enabled further, more directed investigation of project management techniques that will be beneficial during this project.

Within the University there are numerous contacts that can also be utilised for further assistance and guidance in these areas, such as the Enterprise department. While these are not immediately necessary, they have been identified as sources of information, should they become necessary.

A project of this type, involving the development of a large artefact, requires numerous intermediary milestones throughout the development cycle in order to keep the project on track and allow for iterative testing. Time management and scheduling is important for this, and this has also been covered in the lectures mentioned above.

The final aspect of the learning plan for this project is a more overarching one – professionalism. This incorporates all of the skills and tools that will need to be learned during the entire length of the project and will help in preparation for working in an industrial context. Scheduling a realistic project plan with built in milestones, working to a professional standard, using industry-level techniques and testing and evaluating a product with regard to its functional and design qualities; these are all vital to understanding an appreciating a professional work flow.

VI. Project Plan

The scale of the project is ambitious, but with efficient and realistic intermediary milestones built into the project schedule it is feasible within the time frame.

There will be contingencies built into this plan which will account for unforeseen problems or delays occurring, which will be outlined in this section also. A list of milestones throughout the project process can be seen in *Appendix II.*

i. Outline of Project Stages – Semester One

a. Proof-of-Concept

The proof-of-concept artefact is a small prototype design that is intended to display key aspects of the main project. This concept artefact serves two purposes; firstly, that the software is capable of producing the effects and mechanisms required by the artefact and secondly, that the required personal expertise is available to create these mechanisms. A detailed list of exactly which features are included in this proof of concept can be seen in *Appendix I*. The CD accompanying this document contains a video walkthrough of the artefact highlighting each feature in context, along with an in-depth description in the following section.

ii. Outline of Project Stages – Semester Two

a. Concept and Design

Time allocated: Two Weeks

This is a short but vital step in defining the structure and style of the artefact. This stage will also highlight at an early stage anything which may have been omitted from the initial learning plan highlighted in the previous chapter.

This phase will include the creation of concept art for key aspects of the level design and an overall blueprint plan for the level layout. The layout will include notations of key aspects that will need to be implemented such as focal points and event placements along with their triggers. This complete level overview will allow the creation of a list of assets that will need to be created (as well as a list of assets that will need to be sourced from external locations) and a list of key mechanisms that will need to be implemented.

The information generated in this phase will act as a *design bible* for the project. It will aim to provide as far as possible everything that the project requires and the style that the assets will be created in.

b. Level Block-Out and Initial Lighting Intensity Model

Time allocated: Two Weeks

Having formed an overview of the level design, this stage will take a 'breadth-first' approach to generate a fully blocked out level to test that the proposed structure is suitable for purpose. For example, how long does it take to move through the level? If, at this early stage, the level takes a reasonable amount of time to move through, then it can be predicted that with the addition of puzzles and other mechanisms that will slow player movement that the level will provide a substantial amount of play time.

Due to the nature of the artefact and its purpose to elicit fear in those playing it, the blocked out design will also include an initial lighting test.

Particular lighting effects or designs in the final artefact may rely on complex shadow casting or tools such as shadow mapping and dynamic lighting; however, this stage can be used to create a base *lighting intensity* model, showing the placement and concentration of lighting throughout the level. This will highlight any areas where the balance between light and darkness does not fit the aim of the design. For example, while darkness is intended to create fear through restricted vision, too much of it will lead to player frustration and also negate the effect of architecture and other visual elements as they will be obscured from the player's perception. *UDK* has a level viewing mode that is specifically for viewing lighting intensity information, making this a very useful exercise to carry out early on in the project as it will inform modelling and lighting decisions further through the development cycle.

c. Asset Sourcing and Creation

Time allocated: Six Weeks

The development of this artefact will require a large amount of custom assets to be created, such as static meshes and dynamic meshes, as well as texture maps for them and scripts for events and mechanisms. Given the time scale for the project, and the fact that the entire project is being developed by an individual, it is a sensible compromise to source some assets from external suppliers. While the majority of assets will be custom made, base assets, such as simple brick or concrete textures or 'filler' objects, such as crates and barrels can be acquired royalty free from a number of sources.

After completing the concept and design stage and initial level block-out, a relatively comprehensive list of the assets required will be known. While there will always be additions to this as the project progresses, at this stage a library of assets can be collected from which to draw on later.

The assets that will be custom made will also be started during this phase. It is not practical to create *every* asset at this stage, as some may not be required if the level design is restructured or altered later. In reality, this stage is iterative and will be returned to as each development milestone is reached, with assets being created alongside the compiling of the level in the toolset.

This is reflective of an industry working situation, although of course in industry the level designer would not have to take on the role of mesh modeller and texture artist as well – at least not to the extent that will be required in this project.

d. Level Creation and Iterative Control Group Play-Testing

Time Allocated: Six Weeks

This phase of development runs alongside the above phase. It will include the initial creation of a more fleshed-out level, with more detailed art assets and implemented gameplay events. As assets are required, they can be created if they have not been already. This phase is similarly iterative in the same way as the Asset Creation phase.

As sections of the level are completed to a playable level, they will be play-tested by a small control group whose purpose is to find critical functionality-based errors. This testing ensures bugs are eliminated as early as possible, before they become too deeply embedded in the game. This is especially important for Kismet-based errors, as scripts containing bugs can create emergent problems later on if other scripts rely on them.

iii. Outline of Project Stages – Semester Three

a. Public Gameplay Study, Feedback Gathering and Analysis

Time allocated: Four Weeks

Once the level is completed to a beta standard it can be released for a public gameplay study. This will be carried out using University of Portsmouth students, and any other participants that are willing and available to play.

This stage will also include gathering player feedback on the level design, in a similar way to the research methodology used in the pilot study that preceded this project (Howell, 2010b). This feedback will provide a basis for answering the question of whether a holistic, functionalistic approach to level design is practical and effective, as well as providing a critique of the produced artefact itself. This critique will be beneficial personally when undertaking future design projects.

This stage falls during a holiday period, but this should not pose a problem as there are a range of participants that will be available even during holiday periods.

b. Actioning Feedback to Implement Improvements, Writing Analysis and Finalising Report

Time Allocated: Four Weeks

Changes made during this final stage will be dependent on the time remaining before the final deadline. Ideally, there should be at least four weeks for this stage. Any major problems can be either fixed or mitigated to reduce their impact (depending on how much work would be required to completely remove them from the artefact) and minor adjustments made.

The collating and writing up of all feedback data will be carried out during this stage and the written report finalised.

iv. Float Time Built into Project Plan

The plan outlined in this section leaves four to five weeks free between proposed completion of the project and the final deadline. This leaves this free time as a float period which can be added on to other sections should they require more time. Given the ambitious nature of the project, this float time will likely be required during the core development phases, such as Asset Creation and Level Building. The following section highlights in greater detail how specific risks will be controlled.

v. Justification of Methodological Approach to the Development and Evaluation of the Artefact

The iterative process chosen for this project is being used primarily due to its ability to mitigate risk during development and also as it provides a very early prototype that can be critiqued – this is very useful for fixing obvious issues before they cause bigger problems further down the development pipeline.

> The iterative process minimizes risks ... by combining both the well structured management techniques of the waterfall process and the early validation techniques of the evolutionary model. This process ... provides the flexibility necessary to accommodate a high dynamic range of technical alternatives. (Miller, Paradis & Whalen, 1991, p.520).

As stated by Miller et al. above, this approach is highly flexible also. As stated previously in this chapter, this project has the benefit of much more design freedom than a full size project would have. Thus, applying a production methodology that incorporates flexibility into its design seems the most appropriate course of action.

The ‘breadth-first’ approach – whereby the entire product is blocked out at the start of the development process – gives context to the overall design. However, once the process of iteratively creating each section of the level begins, a ‘depth-first’ approach will be taken;

each blocked out section will, in turn, be completed to as close to a finished standard as possible, before moving on to the next section.

The opening section of the level will be built initially, followed by the closing section. Once these two critical areas are completed, the rest of the level can be built section by section. This structure means that if time starts to run out near the end of the project, areas can be cut from the middle portion of the level without having a detrimental effect on the most important areas – the start and end sections.

The method of evaluating the artefact by means of small test groups followed by a large testing group is being used due to its similarity to industry testing procedures and also because it improves the project's efficiency by highlighting any issues encountered as early as possible.

vi. Risk Management and Mitigation during Development

The main risk in any project is running out of time in the schedule to complete the necessary work. As mentioned above, the approach being taken aims to mitigate any risk to the project from unforeseen issues, such as illness or hardware failure. By creating the starting area of the level and the finishing area of the level at the beginning of the project, the two vital locations that the level *must* contain will be in place. The sections connecting these points can be linked in any order which retains its coherence. So, for example, if two weeks were lost to illness, the section that would have been developed in those two weeks could, if necessary be cut from the level and the only real detrimental effect on the final product would be its length.

There are contingencies that can be called upon should other problems occur, such as not being able to create the required number of custom assets. In this case, as this is a non-commercial project, royalty free third-party assets can be called upon to fill any gaps that may occur.

With regard to the actual safeguarding of project data, data loss will be protected against through automatic backups to an external hard drive, and manual backups to at least one cloud storage server.

VII. Proof of Concept Artefact

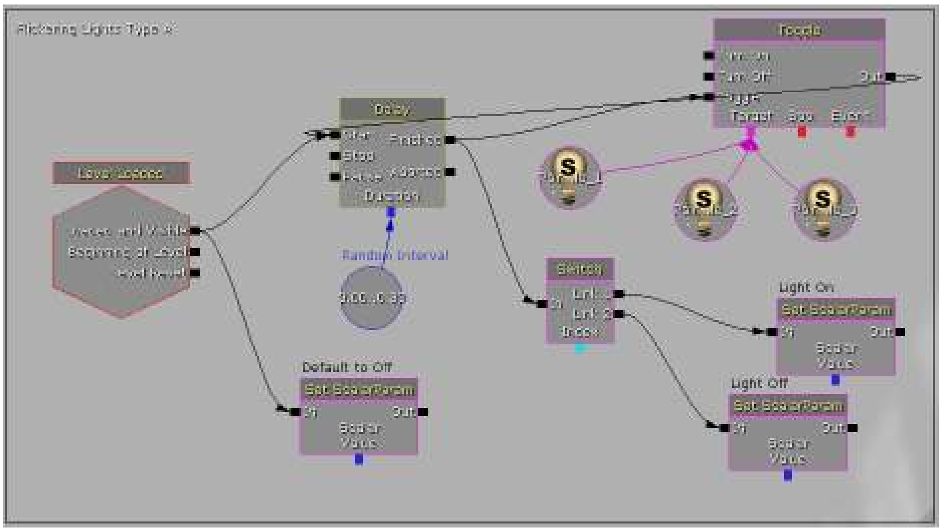
The proof of concept artefact was developed in order to test certain key mechanisms, functionality and personal ability relating to the chosen software to be used for this project – primarily, *3D Studio Max*, *Photoshop* and the *Unreal Development Kit,* or *UDK*. The artefact produced was a small game environment, built with the *UDK.* The CD accompanying this document contains both the raw map data as well as commentated and non-commentated video play-throughs of the environment, highlighting the key aspects. This chapter will reference timestamps within the commentated video to show what is being described.

Note on video footage: The CD contains four .avi files; an uncompressed version of the play-through with and without commentary, and compressed versions of the same videos compressed using the CinePak codec. It is recommended that for the best playback, video files are copied from the CD onto the local hard drive.

i. Dynamic Lights & Dynamic Materials

Video Timestamp: 00:23

The *UDK* lighting engine (*Lightmass*) is capable of rendering numerous real-time dynamic lighting sources without being detrimental to performance. The flickering light example shown is one such dynamic light source. To enhance the believability of the light, the emissive channel on the light mesh material is manipulated in real-time to match the flickering of the light source. This is achieved using a simple *Kismet* (*UDK*'s visual scripting language) sequence, as shown in the screenshot below.



Kismet sequence for flickering lights - the 'Toggle' function handles the light source while the 'Switch' function changes the emissive channel of the material, all based on a 'Random Interval' variable.

ii. High-Fidelity Baked Lightmaps

Video Timestamp: 00:30

The *Lightmass* lighting system also calculates accurate and high quality lightmaps for baking static lighting into a level. *Lightmass* calculates bounced light and radiosity using a photon-based lighting algorithm, meaning far fewer light sources must be placed by the level designer to achieve a realistic environment. The resolution of the generated lightmaps can be altered for individual surfaces, depending on the complexity of shadow being cast on them. For example, the shadow from the window shown in the video is using a resolution of 4, as opposed to the default 32, as this produces a cleaner shadow.

iii. Customised Screen Prompts for Interactive Objects

Video Timestamp: 00:39

This is a minor tweak, but it adds an extra level of customisation. This is achieved using a very simple *Kismet* sequence combined with some of the built-in functions of the *UDK.*

iv. Blended Cross-Fading of Music Tracks

Video Timestamp: 00:48

Music is a key design tool in all games, but is even more apparent in games revolving around fear, terror and horror. The ability to seamlessly cross-fade ambient music tracks will be highly useful in producing an evocative and immersive soundscape to accompany game play and respond to it in a dynamic way. In the video, the music change is triggered simply by the player walking through the door, but it could be triggered by anything; for example, when a player is close to death, the music could change to reflect this.

v. Real-time 'God Rays'

Video Timestamp: 00:56

UDK is capable of rendering different types of volumetric lighting. The method shown here uses a *Dominant Directional Light* projecting light shafts in real-time, which interact accurately with the spinning fan blade in the ceiling. This type of effect can really heighten the realism of a scene.

vi. High-Fidelity Dynamic Shadowing

Video Timestamp: 01:09

In the same way that *Lightmass* is capable of rendering high-quality static shadow maps, the shadows produced by dynamic light sources in *UDK* are, by default, very crisp. This can be used to great effect, as with the fan shadow in the video. This could also be used to pick out shadows of, for example, nearby enemies or monsters in order to elicit fear.

vii. Localised Fog

Video Timestamp: 01:24

Fog is particularly relevant to this project, as it is a horror staple. *UDK* is very flexible in producing these effects. In this instance, the fog has been used to represent smoke from a fire in a localised space, using a *Linear Half-space Fog Volume*; a standard *UDK* function. Fog can also be defined within a space contained inside any static mesh; this essentially means customisable fog volumes can be constructed to fit within any given area of a level. Standard distance and height fog are also very simple to set up, the latter of which is shown near the end of the video.

viii. Realistic Particle Effects

Video Timestamp: 01:31

Whilst this is an obvious feature in any modern game engine, the built in Particle Editor in *UDK* offers excellent flexibility, without the need to resort to external software packages such as *Adobe After Effects* to produce particle systems.

ix. Accurate Sound Attenuation using Reverb Volumes

Video Timestamp: 01:24

Reverb Volumes can be set up by the level designer and assigned various properties which dictate how sound should be heard from within and outside of them. For example, in the video, the fire alarm sound effect is muffled when outside of the smoke-filled corridor, and can be heard clearly when the player moves into the reverb volume contained within the corridor. It would also be possible to add an additional layer of realism by changing the volume of the sound effect depending on whether the door to the corridor is open or not. This would be implemented using a *Kismet* sequence.

Reverb volumes can also be used to control the actual type of reverb in an area; for instance, a very heavy echo in a cave.

x. Skeletal Meshes and Static Meshes Casting Shadows with Mesh Renderer Turned Off

Video Timestamp (Skeletal Meshes): 01:54
Video Timestamp (Static Meshes): 02:11

This was a functionality test to discover whether this effect could be produced convincingly within the engine. After numerous attempts, it was successful, however there are some caveats.

- Turning a mesh's renderer off produces warnings during the level building process. This does not appear to be problematic at this stage, but may be an issue as the effect is used in more complex ways.
- For shadows to be visible from far enough away to suit the intended game play purpose (i.e. as far away as one would typically expect to be able to see an enemy mesh) the default shadow draw distance had to be increased greatly. This is again, not a problem at this stage, but could cause performance issues in larger environments. This should be easily avoidable however with the use of sensible level streaming (the process of only holding in memory the level data for the area immediately surrounding the player's location).
- Static Meshes will only cast a crisp shadow when lit with dynamic lights. This could cause performance issues also if too many dynamic light sources are in use. This can again however be minimised with intelligent level design.

xi. AI Bot Spawning

Video Timestamp: 02:29

Spawning of AI opponents is handled through *Kismet* using *UDK's* built in *Actor Factory* function. This is used simply by determining an actor class to be spawned, and a path-node in the level at which to spawn it. To make use of custom actor classes, one can simply write a new class that extends the standard *UTPawn* class, and these can then be spawned in exactly the same manner.

xii. Custom SkyDome and Exponential Height Fog

Video Timestamp: 02:47

The SkyDome texture used is a custom asset that has been imported to *UDK*. This is a very simple and intuitive process that should cause no problems during development.

The *Exponential Height Fog* is an enhancement of the standard *Distance Fog* seen in many games, offering greater flexibility and the ability to use different types of fog and different heights. This is a very useful tool when creating the correct mood and ambience in a scene and thus is an important consideration for this project.

xiii. Additional Engine Modifications and Available Software Support

Video Timestamp: *Not shown explicitly*

Some additional changes were carried out to the base .ini files to reconfigure certain aspects of game play to suit the proposed type of game to be produced. Whilst not difficult to implement it is clear that a lot of time will be required during development for developing the custom *UnrealScript* components required. This was expected however, and will be very useful for enhancing personal ability in an area outside of art and design.

While creating this artefact, it has also been clear that there is a large wealth of knowledge online relating to nearly every aspect of *UDK.* This will prove invaluable during development, hopefully meaning that any problems can be solved relatively quickly.

The player's movement speed was halved in order to bring it more in line with standard first person games as opposed to fast-paced multiplayer type games. For the same reason, the player's ability to double jump and to dodge side-to-side was also disabled.

VIII. Bibliography

Atari (1981). *Haunted House.* [CARTRIDGE] Atari 2600. California: Atari Corporation.

"Axel" (October 15, 2009) User Review of *Ju-on: The Grudge* uploaded to and retrieved on November 1, 2010 from http://www.metacritic.com/game/wii/ju-on-the-grudge/user-reviews?sort-by=score&num_items=100.

Capcom (1996). *Resident Evil.* [DISC] PlayStation. Osaka: Capcom.

Capcom (1996 – 2010) *Resident Evil Series.* [MULTIPLE FORMATS] Multiple Platforms. Multiple Publishers.

Castle, M. (ed.) (2006). *On writing Horror: A Handbook by the Horror writers of America.* Cincinnati: Writer's Digest Books.

Dark Energy Digital (2010). *Hydrophobia.* [DIGITAL DOWNLOAD] Xbox Live Arcade for Xbox 360. Reading: Microsoft Game Studios.

Darkworks, (2001). *Alone in the Dark: The New Nightmare.* [DISC] PC. Lyon: Infogrames.

EA Redwood Shores, (2008). *Dead Space.* [DISC] Xbox 360. Redwood City: Electronic Arts.

Epic Games, (2009). *Unreal Development Kit* (Build 58xx) [COMPUTER SOFTWARE]. Cary: Epic Games.

Epic Games & Digital Extremes (2008). *UnrealEd 3.* [DISC] PC. Cary: Epic Games.

Fahraeus, A. (2010). *Horror vs Terror.* Retrieved November 15, 2010 from http://womenandhorrorfiction.blogspot.com/2010/04/horror-vs-terror.html

Frictional Games (2007). *Penumbra.* [DISC] PC. Stockholm: Paradox Interactive.

Frictional Games (2010a). *Amnesia: The Dark Descent.* [DIGITAL DOWNLOAD] PC. Washington: Valve Corporation.

Frictional Games (2010b). *Amnesia Toolset.* [DIGITAL DOWNLOAD] PC. Helsingborg: Frictional Games.

Hayes, N. (1998). *Foundations of Psychology – An Introductory Text, Second Edition.* London: Routledge

Howell, P. (2010a). Light at the end of the Tunnel: A study of the influence of game world lighting in a First Person Shooter on a player's decision making processes. Unpublished undergraduate dissertation, University of Portsmouth, Portsmouth.

Howell, P. (2010b). *An exploratory study into implicit fear mechanisms in video games; how do they engage, immerse and affect the player's enjoyment of Amnesia: The Dark Descent?* Unpublished postgraduate research project, University of Portsmouth, Portsmouth.

How to Make Fear, (October 31, 2006). Retrieved October 30, 2010, from http://www.next-gen.biz/features/how-make-fear.

id Software (2004). *Doom 3*. [DISC] PC. Santa Monica: Activision.

Irrational Games & Looking Glass Studios, (1999). *System Shock 2*. [DISC] PC. Redwood City: Electronic Arts.

Irrational Games. (2010). *Irrational Behaviour Episode 5: What Are We Afraid Of.* [ONLINE]. May 3, 2010. Retrieved October 30, 2010, from http://irrationalgames.com/insider/irrational-behavior/irrational-behavior-episode-5/.

Jeffries, L.B. (2010) *Post-Structuralism in Video Games.* Retrieved December 13, 2010 from http://www.popmatters.com/pm/post/130849-post-structuralism-in-video-games/

Koltz, K.L. (2004) *Fear Not.* Retrieved October 11, 2010 from http://www.suite101.com/article.cfm/spirituality_health/111916.

Konami (1999)) *Silent Hill.* [DISC] Sony Playstation. Tokyo: Konami.

Konami (1999 – 2009) *Silent Hill Series.* [MULTIPLE FORMATS] Multiple Platforms. Multiple Publishers.

Metacritic (2010). *Ju-on: The Grudge for Wii.* Retrieved November 1, 2010, from http://www.metacritic.com/game/wii/ju-on-the-grudge.

Miller, F, Paradis, R. & Whalen, K. (November, 1991) *Iterative development life cycle (IDLC): a management process for large scale intelligent system development.* Paper presented at Third International Conference on Tools for Artificial Intelligence. Retrieved November 24, 2010 from IEEEXplore Database.

Naughty Dog, (2007). *Uncharted: Drake's Fortune.* London: Sony Computer Entertainment.

Perron, B. (ed.) (2009). *Horror Video Games: Essays on the Fusion of Fear and Play.* Jefferson: McFarland & Company.

Pinchbeck, D. (2010). *Game Lecture Series #5: Doing Development Lead Research in Games*. [ONLINE VIDEO] Retrieved December 17, 2010 from http://vimeo.com/channels/gls#12578274

Remedy Entertainment, (2001). *Max Payne.* [DISC] PC. Texas: Gathering of Developers.

Rollings, A. & Adams, E. (2003) *Andrew Rollings and Ernest Adams on Game Design.* London: New Riders

Rowling, J.K. (1998) *Harry Potter and the Chamber of Secrets.* London: Bloomsbury

Tale of Tales (2009). *The Path.* [Digital Download] PC. Belgium: Tale of Tales.

Tecmo (2002 – 2008) *Fatal Frame Series.* [MULTIPLE FORMATS] Multiple Platforms. Multiple Publishers.

Turtle Rock Studios & Valve South, (2008). *Left 4 Dead.* [DISC] Xbox 360. Bellevue: Valve Corporation.

Ubisoft Shanghai, (2008). *Tom Clancy's EndWar.* [DISC] Xbox 360. Paris: Ubisoft.

Unity Technologies, (2010). *Unity 3.* (Version 3.0.0f5) [COMPUTER SOFTWARE] San Francsico: Unity Technologies.

Valve Corporation, (2010). *Source Engine*. (Build 4335) [COMPUTER SOFTWARE] Bellevue: Valve Corporation.

VGChartz (2010). *gamrReview – Browse Games.* Retrieved October 9, 2010, from http://gamrreview.vgchartz.com/browse.php?name=resident+evil.

Whalen, Z. (2004) Play Along – An Approach to Videogame Music*. Game Studies*, 4 (1), available online at http://gamestudies.org/0401/whalen/.

Wojnarowicz, J. (2004) *DooM III Review.* Retrieved November 1, 2010 from http://www.firingsquad.com/games/doom_3_review/page2.asp.

XSeed Games, (2009). *Ju-on: The Grudge.* [DISC] Wii. Luton: Rising Star Games.

Appendix I

List of Features for Implementation in Proof of Concept Artefact

Feature	Justification
Small, Playable Game Environment	Shows technical ability with software
Custom assets imported and placed in environment	Shows knowledge of asset pipeline for both mesh and texture data
Implement different types of static and dynamic lighting effects in a testing capacity	Vital tool for 'horror' design, this shows both engine capability and personal capability to implement these effects
Implement dynamic background music in a testing capacity	Another vital tool for creating changing ambience in a 'horror' game, this shows both engine capability and personal capability to implement this
Implement a basic scripted sequence utilising Kismet scripting	Kismet will be used for everything interactive in the game environment and to handle background scripts. This will show personal capability with this system.
Implement static and rigged objects which cast dynamic shadows but have their mesh renderer disabled	This effect is a core concept for the design of the artefact, which is centred on only being able to see enemy shadows, not the enemies themselves. This shows the engines capability to handle this effect.
Implement a basic AI enemy	Shows personal ability with Kismet AI scripting

Appendix II

List of Project Milestones

Milestone	Latest Date for Completion
Proof of Concept Artefact	December 24th, 2010
Concept Art and Design Work	February 18th, 2011
Level Block-Out and Lighting Intensity Model	March 11th, 2011
Asset Creation and Third-Party Asset Sourcing	April 22nd, 2011
Level Building and Control Group Play-Testing	May 20th, 2011
Implement Changes Based on Play-Testing	June 10th, 2011
Public Play-Testing and Feedback Gathering	July 8th, 2011
Analysis of Feedback Data	July 15th, 2011
Final Adjustments and Improvements to Artefact	August 12th, 2011
Writing Final Copy of Report	August 19th, 2011
Float Time across Entire Project	***Approximately 4 – 5 Weeks***

Appendix III

Gannt Chart Breakdown of Initial Project Schedule