A report submitted in partial fulfilment

of the regulations governing the award of

the Degree of

BSc. (Honours) Computer Science with Games Development

at the University of Northumbria at Newcastle

Project Report

Creating a video game where the day/night cycle is based upon relative psychological factors and attention span research

General Computing Project

Peter David Stuart

2021 / 2022

Authorship Declaration

DECLARATIONS

I declare the following:

(1) that the material contained in this dissertation is the end result of my own work and that due acknowledgement has been given in the bibliography and references to **ALL** sources be they printed, electronic or personal.

(2) The Word Count of this Dissertation is 23,225.

(3) that unless this dissertation has been confirmed as confidential, I agree to an entire electronic copy or sections of the dissertation to being placed on the eLearning Portal (Blackboard), if deemed appropriate, to allow future students the opportunity to see examples of past dissertations. I understand that if displayed on eLearning Portal it would be made available for no longer than five years and that students would be able to print off copies or download.

(4) I agree to my dissertation being submitted to a plagiarism detection service, where it will be stored in a database and compared against work submitted from this or any other School or from other institutions using the service.

In the event of the service detecting a high degree of similarity between content within the service this will be reported back to my supervisor and second marker, who may decide to undertake further investigation that may ultimately lead to disciplinary actions, should instances of plagiarism be detected.

(5) I have read the Northumbria University/Engineering and Environment Policy Statement on Ethics in Research and Consultancy and I confirm that ethical issues have been considered, evaluated and appropriately addressed in this research.

SIGNED: Peter David Stuart

Acknowledgements

I wish to acknowledge my supervisor Dan Hodgson for his guidance throughout the whole process of this dissertation. I could not have completed this project to anywhere near the standard I have without his support during crucial stages.

Thank you, Dan.

Abstract

This report focused on the creation of a video game as a proof-of-concept for the evaluation of day/night cycle length. A day/night cycle is a concept in video games where the passage of time is used to change the environment around the player. To aid this experimentation, a literature review was performed to gain the foundational knowledge needed in order to create the game's requirements.

To begin, the literature review looked at the impact different factors can have on the human attention span as well as how this can be observed. This literature proved vital in the testing phase where the testing methods were directly influenced from these reports. Further, a review of prior games featuring day/night cycles was conducted. The length for the product of this report was taken directly from the most effective games in this domain.

In order to design the game, a list of functional and usable requirements were created to ensure the product was palatable and enjoyable. These requirements were achieved and thus the game was deemed an entertainment product as well as a proof-of-concept. The further design phase saw diagrams and models drawn up to ensure the implementation was complete and effective.

The game itself, entitled *Half-Blood*, was successfully implemented using the prior designs. Algorithms consisting of the physical movement of the Sun and also the tracking of in-game time ensured that the main aim of the product was complete. The AI of the enemy characters was then directly influenced from the in-game time and attempted to kill the player as long as the Sun was down.

Finally, testing was completed to assess the effectiveness of the product and to prove the implementation of a day/night cycle was accomplished. The testing found that the shorter the cycle length, the more the player enjoyed and felt engaged playing. Recommendations of the software used to achieve these aims were made and planned future work was discussed.

To conclude, this project has studied attention spans and ultimately a game has been implemented effectively based on the findings of these studies. It is believed that outside the domain of video games, this research could be effective when trying to engage anyone via the use of shorter, more concise segments.

List of Contents

Chapter 1 - Introduction

1.1 Background to the Project

The primary aim of this project was to delve into how psychological factors can affect a gamer's experience. The studies found in this report noted that while playing, the player becomes oblivious to how a game's changing environment is affecting their mental state. There is much interest in the area of study surrounding enjoyment based on designer's choices. The method of this design choice that has generated the most intrigue is that of day/night cycles.

Day/night cycles are a method of game design used to alter the in-game environment through the passage of time. They are perhaps best known for their use in open world environments as a means to replicate real life and suspend player's disbelief. However, this report aimed to prove that day/night cycles can also be used as tools to generate engaging gameplay by adding challenges during specific time "states". The term states will be used throughout this study to refer to instances when a game's clock ticks over into a time where environmental changes are clear. For example, a game could tick over from a dawn state, to a day state, to a dusk state, and so on.

The purpose of this work will be to prove that these changes in state can be used to generate a change in player immersion. The reason for this study of work to be carried out is due to the prolonged personal interest in video games that use a cycle of time. It has always been thought that these cycles could mean something more than what they show on a surface level, and thus this study was undertaken to prove how powerful of a tool they truly are.

1.2 Proposed Work

The aims and objectives for this report were initially laid out in the Terms of Reference (see Appendix A). This list of criteria allowed for this project to be planned from a high level in order to monitor the progress being made throughout. The objectives specifically directly correlated with the structure of the report which led to a successful chain of paper chapters; this connection was found in section 12.1.2. The aims of this paper are as follows:

1. Investigate potential optimum day/night cycle for games

2. Create a game demo product as proof of concept

These aims were deemed necessary due to the context of the report. An investigation for the potential of an optimum cycle length was required to be able to create a proof-of-concept demo that featured a day/night cycle. The investigation could then be resumed in order to evaluate the game demo's test results.

The subject of the work produced would fall into the category of a game demo. Although a game demo, it must be noted that the scope of this project ensured that the game would be used primarily as a proof-of-concept. Therefore, any additions that would turn this demo into

a fully fledged video game, would negatively impact the effect that this product would have as an experimental piece of software. In order to prove the second aim of this investigation, it was imperative that the software be fully functional and usable so that human participants could effectively test the game and potentially prove any hypothesis.

The limitations encountered when attempting to implement this experimental game were few and far between. However, during testing the limitations of not being permitted to access specialist equipment or face-to-face contact negatively impacted the data collected from this phase. Nevertheless, the literature review of this paper tackled the issues of not having concrete data, and therefore this concern was accounted for. To conclude, the lack of scientific equipment could actually be seen as a positive because it encouraged a more defined study technique and a higher level of questioning to counteract the absence of theoretical data.

Finally, the proposed work of this project has been planned from the initial Terms of Reference (Appendix A) set out at the beginning of this report. This document acted as a guideline of work and ensured all goals of the project were attempted and ultimately completed. To complete the work, a choice of the required tools and techniques were selected in chapter 5. The main tool choice used to drive this project was the Unreal Engine (Epic Games, 2022). This game engine allowed for in depth, highly efficient tools that were used to build the world's environment as well as the backend artificial intelligence. From this development came testing that consisted of a software and user section respectively. Both sections aimed to prove functionality, usability, and overall success of the product created in relation to the primary aims of the project. Furthermore, evaluation was then carried out that encapsulated the product and also the process as a whole. To summarise the project, a conclusion was created that gauged the success of the report and its aims. Ultimately, the project closed with recommendations of the software and process used, while also providing plans for future work.

Section 1 - Research and Planning

Chapter 2 - Literature Review on Attention Spans and Cycles of Activity

2.1 Introduction - What is an Attention Span

The main focus of research carried out for this report will be on the topic of attention spans. The dictionary definition of an ‘attention span’ (2021) is as follows: “the length of time that someone can keep their thoughts and interest fixed on something”. Upon reviewing specialist literature on the topic of attention spans, it has been noted that there has been extensive research performed in the domain to identify factors such as: length, types and influences. Therefore, it should first be noted that the expert literature of Mateer and Sohlberg has been used as the foundation and benchmark for all other research in the domain.

2.2 Different Types of Attention Spans

This chapter will be breaking down every factor regarding attention span, this will start with a clear explanation of the many different *types* of attention span. In the Journal of Clinical and Experimental Neuropsychology (Mateer and Sohlberg, 1987), five clear differing types of attention span were identified.

- Focused Attention: The ability to respond discretely to specific visual, auditory, or tactile stimuli.
- Sustained Attention: The ability to maintain a consistent behavioral response during continuous or repetitive activity.
- Selective Attention: The ability to maintain a cognitive set which requires activation and inhibition of responses dependent upon discrimination of stimuli.
- Alternating Attention: The capacity for mental flexibility which allows for moving between tasks having different cognitive requirements.
- Divided Attention: The ability to simultaneously respond to multiple tasks.

The clear disparity between attention span types is useful as these terms will be used throughout the paper and it is integral the reader understands the differences between each type.

2.3 What Can Affect an Attention Span

There has been intense research on the potential influences of attention span. The potential for internal and external factors affecting attention spans is crucial to my investigation as this data will then need to be taken into account from participants.

2.3.1 Biological Sex

Many studies from different academic backgrounds were investigated to find if there was a significant differential in span times between male and female participants. Firstly, a study performed on behalf of the Institute of Sport Science and Innovations (Solianik et al., 2016) found that “current study showed no sex differences in the mean values of cognition”. However, there have also been studies that find that men and women perform quite differently in attention span tests. This can be seen in a widely cited journal article (Merritt et al, 2007) where it was found that men and women “show differential responses in an endogenously cued visual selective attention task”. As seen, this study was focusing on specifically the *selective* attention span type. The study involved using cues to distract the participants to recognise if their attention was disrupted. Studies involving valid and invalid cues come from the work of Posner who used a paradigm that has been used in several attention based studies since (Posner, 1980). An example of a trial’s paradigm involving a valid cue can be seen in Fig 1. (Posner and Cohen, 1984) whereas Fig 2. shows the same model but for an invalid cue created by the author. Interestingly, the female participants suffered “increased costs to an invalid cue” whilst the male participants actually “benefit from an invalid cue”. Therefore, it can be seen that a subtle difference can be significant in results.

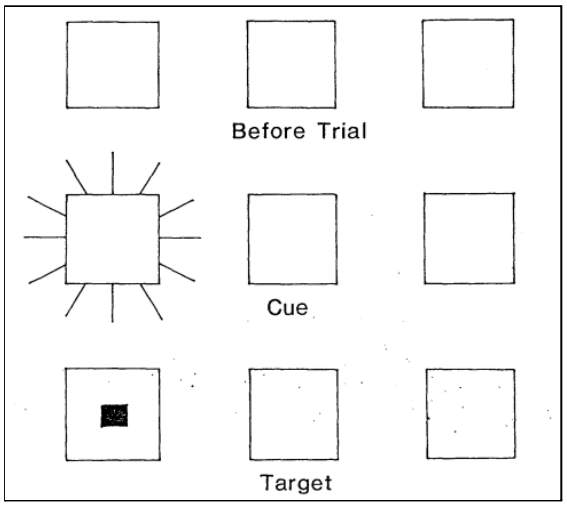


Fig 1. Valid Cue
(from Posner and Cohen, 1984)

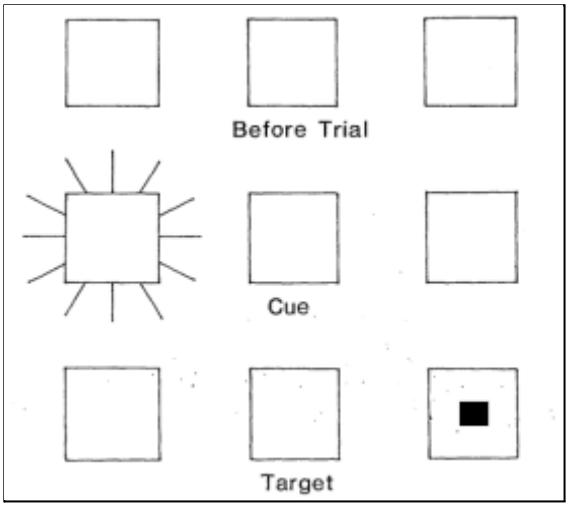


Fig 2. Invalid Cue

As the above test showed, men were slightly better at selective attention in this controlled environment. On the other hand, it should be noted that there could be a type of attention where women preside over men. This research was heavily inspired by the common saying that ‘women are better than men at multitasking’. Thus the research carried out led to a psychology article (Stoet, et al., 2013) which asked the question “Are women better than men at multi-tasking?”. The conclusion of this study found that “men and women performed the individual tasks with the same speed and accuracy” but that “mixing the two tasks made men slow down more”. To further aid this thesis it was also found that “women performed considerably better in one of the tasks measuring high level cognitive control”. To summarise, it can be said that multitasking falls under the attention *type* of divided due to the

fact it is “the ability to simultaneously respond to multiple tasks” and ‘multitasking’ (2021) is “a person's or product's ability to do more than one thing at a time”.

On the contrary, a third study (Flôres, 2016) has shown that there is little difference when comparing male and female participants as there was “no significant differences between the gender”. Regarding this trial, it was however seen that the external factor of focus type held more of an influence on the participants. To summarise, the data found regarding differences in attention due to sexual biology can be seen as ignorable due to the negligible disparity and the inconsistency between results in different types of attention (selective and divided). Therefore, biological sex will not be used as a variable in this study.

2.3.2 Environment and Focus

Concerning the previously mentioned test (Flôres, 2016) it was found that external factors played a bigger role than sex/gender. When focusing on a particular task, the mind will be using two different methods to produce the best outcome for the task at hand. These two methods are called distal external focus (DEF) and proximal external focus (PEF).

The definitions of both DEF and PEF can be best defined by two other pieces of literature in this domain. Firstly, a study looking at improving skilled golfers’ swings (Bell and Hardy, 2009) defined the DEF as being attentive to the “trajectory of the ball”, and PEF as being attentive to the “movement of the club”. To further aid in defining these complex descriptors, a study into improving novice dart players’ throwing performance (Mckay and Wulf, 2012) had some players focus on where they were aiming on the board “i.e. bull's eye” (DEF), and others focusing on the “trajectory of the dart” (PEF).

Now definitions have been explained clearly of the differing ways to focus, it can be seen that these methods of focus have been understood to be important variables when testing performance. This study in particular will be looking at how gamers hold their attention and therefore any important variables that can affect their performance should be accounted for.

2.3.3 Age

When thinking about cognition and memory, the factor of age is always one of the primary indicators that someone may be lacking in these neurological functions. However, can it always be assumed that older people have poor memory, or that younger children have extremely short attention spans? Research shows that on the subject of memory and attention the stereotype that exists about age is true, but not to the extent as portrayed by the media.

Memory

Firstly, two studies involving memory based experiments (Dobbs and Rule,1989) (Lindenberger, Marsiske, and Baltes, 2000) found that once the participant crossed the 60+ threshold then the results incurred significant detriment. For example, in an exercise of recalling 16 random words from memory in a series of different conditions (e.g. sitting, standing, walking, etc.), the participants in their 20s and 40s massively outperformed the participants in their 60s (Lindenberger, Marsiske, and Baltes, 2000). When standing, the 20s recalled an average of 13.3 words, 40s 12.1, and 60s 9.6 respectfully. When walking an aperiodic track, the 20s still recalled a respectful 10.5 words on average. However, in these

conditions the 40s recalled 7.9 and the 60s only 5.8. See Fig 3. for all results of this experiment.

	Encoding condition							
	Sitting		Standing		Walking, oval track		Walking, aperiodic track	
Age group	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Young	13.0	2.9	13.3	2.4	12.9	2.8	10.5	3.6
Middle-aged	11.7	3.2	12.1	3.0	9.7	3.3	7.9	3.6
Old	9.5	3.0	9.6	3.2	7.2	3.8	5.8	3.2

Note. $N = 140$. Maximum score = 16.

Fig 3. Mean Number of Correctly Recalled Words as a Function of Track Type, Encoding Condition, and Age Group (Lindenberger, Marsiske, and Baltes, 2000)

To further support the claim that age has a severe implication on memory, another study found that the results of a working memory task discovered that “significant declines were found between the ages of 60 to 69 and 70+ years” (Dobbs and Rule,1989).

Attention

Due to the intent of this report, it would be more appropriate to focus more specifically on attention spans. As noted above, there are several categories that attention spans can be broken down into. When researching how much of an impact age has on participants’ cognitive functions, studies were found that measured “divided attention” and “selective attention” respectively.

Divided Attention

When performing experiments based on the division of attention, it is key to monitor the participants’ ability to simultaneously respond to multiple tasks (Mateer and Sohlberg, 1987). There have been two studies selected that performed experiments expertly in this domain (Salthouse, Rogan, and Prill, 1984) (Ponds, Brouwer, and Van Wolffelaar, 1988). Firstly, Salthouse et al. (1984) conducted investigations based on prior research in the study of attention by Somberg and Salthouse (1982). The 1984 paper took the use of attention-operating characteristics (See Fig 4. for an example) from the 1982’s to further prove cognitive detriment in relation to the advancement in age.

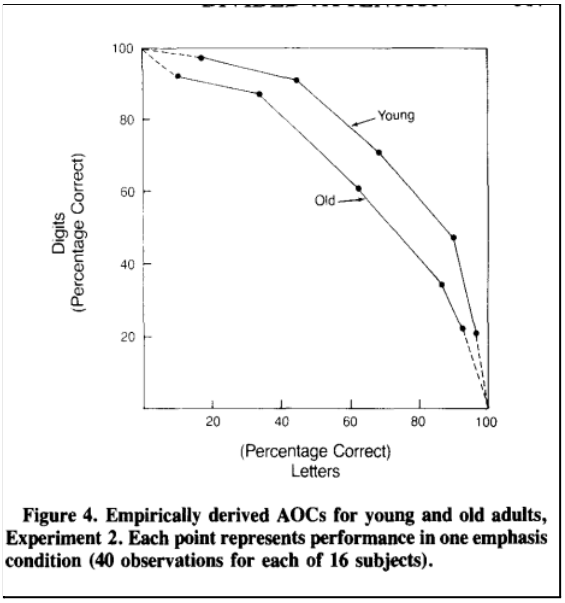


Figure 4. Empirically derived AOCs for young and old adults, Experiment 2. Each point represents performance in one emphasis condition (40 observations for each of 16 subjects).

Fig 4. An attention-operating characteristic taken from one of the experiments performed by Salthouse et al. (1984)

Selective Attention

The plethora of distinctive attention studies is further shown by two papers focused on the effects age has on selective attention. As with the Posner and Cohen's study regarding attention based on biological sex (1984), the two studies discussed here also aim to prove correlation between human factors and selective attention. Firstly, Maylor and Lavie (1998) attempted several experiments with differing set sizes in an attempt to identify how much an effect age has on the brain. The results found that when the "set size was small" then the effect the invalid cue had on the participants was "much greater for the older participants than for the younger ones". However, once set sizes became large, the "distractor" (invalid cue) affected the younger group slightly more than the older demographic. Therefore, the exact effect age has on selective memory is unclear due to the disparity in results. To support the claims made in the former experiment performed by Maylor and Lavie, Hasher and Zacks (1988) sought out to prove that there is an age-related inhibitory decline. This paper observed the highest quality literature available, during this primitive time of investigations into attention, and focused primarily on the "ageing and discourse comprehension". The findings from this study were extremely decisive, they found that "the efficiency of the inhibitory processes that underlie selective attention is reduced" in the context of ageing. This means that the inhibitions of the participants were directly affected based on their age, further proving that age is an incredibly useful variable when monitoring attention.

Conclusion

To conclude, these highly regarded studies form the opinion that age is the most prominent factor when analysing any brain activity. Therefore, during the playtesting stage of this study each form filled in will ask the participant for their age. This will be the only personal variable being collected due to its undeniable affect on the brain, as well as this it must be noted that the testing phase will require all participants to be over the age of 18 so the factor of age is important in more ways than one.

2.3.4 Conditions with Detrimental Effects on Attention Span

These conditions all listed above have been described as causing slight differences in attention span, such as men and women producing moderately different results. However there are many conditions, diseases and disorders that can greatly change someone's attention span and therefore must be identified in the earliest stage of planning. For instance, one study (Solianik, et al., 2016) made sure that the participants did not have any of the following: "no medications that could affect cognitive function; no diseases or disorders that could affect cognitive performance". Upon reading this paper it became clear that the study performed in this paper should also have exclusion criteria as to not create anomalous results.

From this realisation, more research into what could constitute *exclusion criteria* was performed. It was found that the following have a major implication on attention and cognition:

- Medication (National Health Service, 2018)
- Drugs
- Alcohol
- Disorders (e.g. ADHD)
- Injuries (specifically head trauma)

When it comes to the testing phase of this study it will be made clear that if the participant has any of the above criteria then they will be indiscriminately excluded. This is simply for the congruence of data and not for the defamation of character in a prejudicial way.

2.4 What is the Average Attention Span

The length of the average human attention span is an area of psychology and cognitive neuroscience that is widely debated. For instance, one study found that for an adult a time of 5-6 hours was a good estimate (Cornish and Dukette, 2009), but there is no empirical evidence to support this. This is the main issue when deciphering a human's attention span as it is not always black and white and attention comes in many forms as noted prior.

The main reasons for the lack of a clear understanding of average attention span are:

- Difficulty in measuring the span
- Difference in testing environment
- Task dependence

2.4.1 Task Dependence and Misinterpretations Surrounding Average Attention Spans

Following on from the discovery that attention span still remains incredibly difficult to measure and monitor, it was said that task dependence is one of the primary reasons for this issue. Remembering back to the five key *types* of attention: focused, sustained, selective, alternating, and divided. These directly correlate to the idea that any specific task will require one or perhaps many types of attention and thus a clear measurement is impossible to generate as each type of attention is so different from one another.

As well as the issue of task dependence, it has been widely reported in the world that human attention span is rapidly dwindling and is now less than that of a goldfish (McSpadden, 2015) (Ryssdal, 2014). Some insights (Microsoft, 2015) have even looked to measure the fall in human attention span and map it against what is believed to be a goldfish's, as seen in Fig

5. below. Nevertheless, it must be asked that is this concept of a dwindling average attention true and, furthermore, can this estimate of goldfish's attention be trusted or is it just a trivial myth?

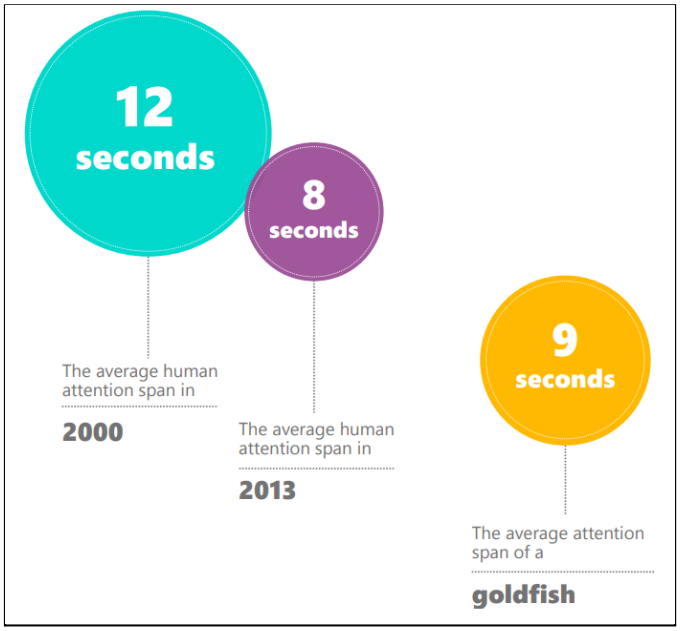


Fig 5. Human attention span vs goldfish attention span (from Microsoft Canada, 2015)

One BBC News report by Maybin (2017) aimed to debunk this exact statistic used by Microsoft Canada. The report focused on how Maybin interviewed expert Professor Felicity Huntingford who has "spent almost half a century studying fish behaviour". The data noted from Huntingford detailed how over a century of studies in fish memory actually found that fish can actually perform complex memory challenges.
To summarise, this study will be directly impacted by the lack of a clearly agreed method of attention span measurement, as well as the misinterpretations of attempted measurement studies such as the goldfish case listed prior.

2.5 Why This Study Relies on Attention Span Literature

This study will use the plethora of attention span literature to act as a foundation for understanding how attention can be affected and monitored during rigorous testing. The reason why this preliminary chapter has been so expansive is due to the fact it is a topic that has a large breadth of studying, whilst also being heavily scrutinised by many authors in the domain. Therefore, as an outsider to the topic of psychology, the researcher felt it necessary to delve into all different aspects in the domain of attention span literature.

Specifically, this study will heavily rely on all researched areas as the next chapter will show a better demonstration as to how these psychological experiments will directly correlate to gaming. The study that this report will go on to perform will encapsulate all knowledge of attention spans and will answer a clear hypothesis of "what is an ideal day/night cycle time for a video game to have". This is due to the belief that a video game's day/night cycle will be heavily influenced by the attention a gamer is willing to put into their experience before becoming bored or finding the tasks monotonous.

To summarise the primary study, the author will be creating a video game with a day/night cycle that will first and foremost use attention span knowledge to generate exciting gameplay for the player. The realisation that the *cycle* that many games use is simply a facade to hide a clear change in gameplay difficulty, revealed that the principal factor in *cycle* creation should be a clear understanding of how the gamer pays attention depending on the cycle they are currently in.

2.6 How Attention Spans Can Be Observed

There are many different methodologies used in previous studies to observe how the participants exhibit changes to their attention. Below are some of the methods used in the domain of attention span measurement and a justification for the method that will be chosen in this paper.

2.6.1 Method 1

The first method chosen is from a paper (Nacke and Lindley, 2008) that used three variables to track a player's gameplay experience. The variables used in this paper were: boredom, immersion, and flow. The apparatus used to measure these three variables were:

- Facial EMG
    - This was used to record muscle activity
- Galvanic skin response
    - This was used to measure the impedance of the skin
- Video recording
    - This was used as a validation tool to see if psychological data could be visually inspected on playback
    - It also allowed a more human look at how the player moved their body during gameplay
- Game experience survey
    - This was used to allow the player to note how they felt the experience went from their perspective
    - It also allowed for any non-physical responses to be noted by the examiner

After assessing the resulting findings that were generated from these apparatuses, it could be seen that each apparatus individually produced skewed visions. However, when looking at the combined results altogether they formed an extremely useful dataset. Although this study showed the clear usefulness of physical equipment, the study performed in this paper will not allow for physical interaction with the participants due to the ongoing pandemic. On the other hand, the use of a camera and survey can easily be recreated by using webcams and online forms in a totally virtual environment.

2.6.2 Method 2

An investigation of visual attention in FPS computer gameplay (Sennersten and Lindley, 2009) attempted to find if a player "attends to the target opponent while shooting at them" and also if the player "targets the nearest opponent". Unlike the plethora of apparatuses in the previous study, this study utilised only one apparatus: an eye tracker. This eye tracker was used to monitor where the player was looking and then this was measured against the control variables, such as: opponent distance.

The results from this study were critically held against the original hypothesis and it was found that the data matched with a high level of accuracy. For instance, the test to see if the player looked at their opponent fully before firing “appears to be correct in about 88% of initial encounters”. An 88% majority shows how the hypothesis chosen was a sensible estimate to how players attend to their targets before firing. The other section of the study also found that “an average of 77% of cases the closest opponent will be looked at and shot first”, therefore sporting another impressive overwhelming majority in favour of the researcher’s hypothesis.

After reviewing this paper in full and looking in depth at how the data was gathered and presented, it was concluded that being able to see the player’s eye movements specifically can help to generate extremely useful data. However, this study will unfortunately not be making use of eye trackers due to the prevention of face-to-face contact. This could be seen as a missed opportunity if the data gathered from this study is not as supportive as the methodology used by Sennersten and Lindley. To summarise, the lack of a clear data gathering tool like an eye tracker could have serious implications to the quality of data generated in this study but this will be covered in more detail during the testing phase in chapter 8.

2.6.3 Justification for Chosen Method

The methods this paper will be utilising during the study period will be influenced heavily by the previous studies noted above. The lack of professional equipment and direct human contact could potentially decrease the accuracy and quality of the data from past studies, nevertheless the scope of this paper is far smaller in scale than that of the previously mentioned studies.

The chosen methods will be:

- A live webcam feed of the player during gameplay
    - This will be synchronised with a shared screen of the gameplay
- A short semi-formal interview to gain instant insight into how they felt the testing went
- A questionnaire to be completed within a week of testing that will be used for quantitative data

The reason for the chosen methods lie solely in the availability of equipment and scope of the project. The evaluation of how the testing could have been improved will be seen in more detail within chapter 10. To conclude, the methods chosen have the backing of past expert testing but the reliability of the data could be poor, nevertheless this unreliability has been accounted for should it arise later in this paper.

Chapter 3 - Review of Day/Night Cycles in Domain Media

3.1 Introduction - What Is a Day/Night Cycle in the Context of Video Games?

Within the video game domain a day/night cycle is a device used to showcase the passage of time in the virtual world the player inhabits. Many different games have tackled this device in a variety of distinct ways, such as using it to “add atmosphere to a game, establish a time limit, play a role in gameplay rules” (Zagal and Mateas, 2010). Therefore, it can be said that a day/night cycle is only ever used as a tool to generate exciting gameplay.

3.2 How Day/Night Cycles Are a Justification for a Change in Difficulty

Continuing on from the previous point, day/night cycles work best when they are used to aid the gameplay experience. Taking this incite and linking it back to the review of attention span literature, this study will be taking the novel approach that *video games use day/night cycles as a façade for a clear change in difficulty*. The reason the researcher is taking this stance is because the prior research into attention and cognition revealed that a player will require a dynamic playing experience to keep their attention state high.

When disregarding the terms *day* and *night*, the cycle can generally be regarded as a cycle of two differing states. The terminology used between each state becomes redundant, especially in instances where a game does not switch in a binary fashion, but instead has several states such as dawn, day, dusk and night. This kind of cycle is much more common due to its gradual incline and decline in player stimulation, in fact some games can have so many *states* that the easing in and out becomes difficult to notice. This can all be down to how the creator wants to allow for a change in difficulty for the player, if they wish to have it very gradual then they can add in more *states* and vice-versa for a steep change in difficulty.

Regarding difficulty changes, a player will want a game to challenge them so they do not get bored, while at the same time not becoming too challenging that the tasks turn into a grind. The state that the player would wish to immerse themself in is called a *flow state.* The term flow state was coined by the regarded psychologist Csikszentmihalyi (1990) and has been used in the video game sector as “a state at the boundaries between engagement and immersion, of being totally absorbed in meeting a constantly unfolding challenge” (Lindley and Sennersten, 2006).

Therefore, a change in difficulty is necessary within games to keep the player's mind at that highest state of immersion. This study will be looking at how day/night cycles are these changes in many cases, and how a game can be made that incorporates this ideology.

3.3 How Day/Night Cycles Can Be Utilised Effectively

When assessing different day/night cycles in the medium of games it has become clear that a good day night cycle would keep the player engaged throughout both aspects of the cycle. A good cycle would also allow for the suspension of disbelief for why the behaviours of the environment change over time. This requires a majority of assets within the players boundaries to have a particular use on either side of the cycle to give new meaning to areas.

On the other hand, a poor day/night cycle would not take advantage of the passing of time. This could be due to AI being unaffected, leading the player to question why a cycle was installed as a feature. Anything that allows for the player to question their surroundings or argue with the creator's decision making shows poor game design.

Therefore, when attempting a day/night cycle the creator must build the world and levels with purpose as to not generate negative reactions. This will be useful during the designing phase of this project due to the insight into what makes a day/night cycle effective.

3.4 Different Categories of Day/Night Cycles in Games

When researching in this particular domain it was found that each game could fit into a particular category due to their cycle usage. The different types of day/night cycle usage will be broken down by the researcher into the categories:

- Integral to gameplay - This is in instances where the *day* and the *night* times are so different in abilities or challenges that it is the main gameplay feature of the game.
- Added challenge - Similar to the above category, this category links the cycle to the gameplay but not so much that the player feels like they are in a completely different world/character.
- Added realism - This category is seen mostly in large, expansive open worlds where the feature is simply to remain realistic to the world as we know it. Mainly used in games where Earth is the primary setting.
- Added opportunities - This category could also feature a few challenges, but the main difference being that the player is more intrigued to explore the same area in both *day* and *night* in order to gain more opportunities rather than find new challenges.
- Aesthetic only - This category is perhaps the least interesting as the changes seen when transitioning from *day* to *night* are only in the reduction and addition of brightness seen in the game world.

As well as, a category that will not be taken into account but will still be discussed:

- Real time clock - When the clock of the game world matches the system's clock exactly. This method will not be accounted for due the unlikelihood of a player being able to play through multiple cycles in one sitting.

Below are some examples of how an array of games have used the different types of day/night cycles (as defined above) to aid the gameplay experience. The examples chosen have been analysed in terms of cycle length (time taken for a 24 hour cycle to complete), and how effective the cycles are.

3.4.1 Cycle Category 1 - Integral to Gameplay - *Don't Starve* and *Knight Lore*

Examples Chosen

The games chosen for this category are *Don't Starve* (2013) and *Knight Lore* (1984).
Firstly, *Don't Starve* uses the incoming nightfall as pressure on the player to generate a campsite because the game kills you if you have no light source during this time. Therefore, this is an example of how day and night are complete opposites as the player cannot survive

in one without the right tools and even still the gameplay is vastly different to when in day time.

Secondly, *Knight Lore* is similar in the regard that gameplay completely alters, however the results are not failure threatening. Instead of generating a game over scenario based on time, the in-game time directly changes gameplay styles. This is due to nighttime changing the player into a werewolf. As a werewolf the character is attacked by more enemies than when in human form, however this negative is counteracted by the ability to jump higher when in werewolf form which allows for the completion of certain puzzles (Hunt, 2010).

Examples Cycle Length

The length of the *Don't Starve* cycle is best shown visually (See Fig 6.). Here it can be seen that the length of the day, dusk, and night sections are different lengths. The actual time for each section has been broken down by a user in the dedicated *Don't Starve Wiki* (2022) (see Fig 7.) and it can be seen that the length deviates each day. This is an interesting choice from the developer to vary gameplay and replicate the real-life shift in daylight hours. The full cycle always remains at 8 minutes.

Fig 6. GUI representation of the in-game clock ('Day-Night Cycle', 2022)

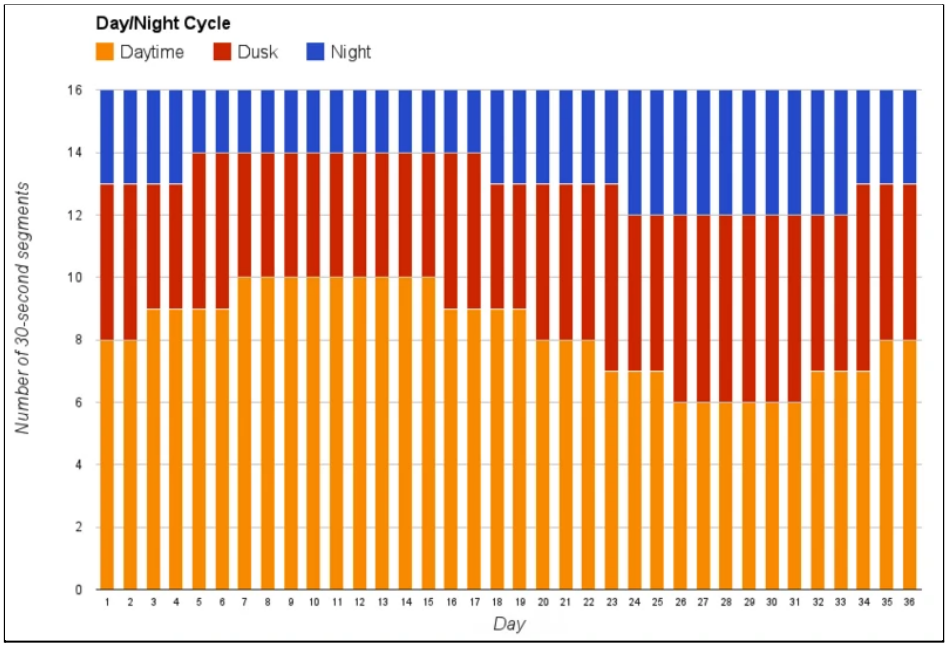


Fig 7. 30 second segments that make up the total cycle ('Day-Night Cycle', 2022)

On the other hand, *Knight Lore* uses a more inflexible approach of making every day an exact time with the length of day and night being made identical. The exact time for this

cycle is 1 minute (RZX Archive, 2013). Therefore, it can be said that video games that use similar cycle styles can be vastly different in cycle time as a whole, and also the dynamism of the cycle length based on game progression.

The Strengths and Weaknesses of How the Day/Night Cycle Is Utilised

The strength of the cycles used in these examples are that they push the player forward in progression and present them with engaging challenges.

However, the biggest weakness of both games are that they put the player under too much pressure too quickly. This can be too demanding for a player that is unskilled and may crumble under the time constraint.

3.4.2 Cycle Category 2 - Added Challenge - *Minecraft* and *Dying Light*

Examples Chosen

The examples chosen for this cycle category were perhaps the challenging as night is usually a tool used in games to invoke fear and danger. However, it was decided that this category should feature games that add a much larger challenge to the player than simply trying to scare them, while also not changing the emphasis on gameplay to the extent that category 1 above showed. The first example is *Minecraft* (2011) which has the added challenge of adversaries which only appear during the night hours. This creates a tactical choice for the player as they can stay out and fight these enemies for rewards, or they can stay inside and sleep till daytime.

Similar to *Minecraft*, other games usually offer a reward to the player for facing the extra challenge. For example, *Dying Light* (2015) uses the incoming nightfall to encourage the player to rest indoors as a break from engagement. If the player disagrees then they are put into a world of danger where the enemies are much stronger than themselves with increased abilities. This method of game design encourages the player to do a plethora of tasks in one space of time, with the night acting as a rest bite if they are safely indoors. However, if the player wishes to stay outside in the dangerous conditions then they will be rewarded with double experience points and in-game loot drops that are only available at this time.

Examples Cycle Length

The length of Minecraft's day/night cycle is directly correlated to its “tick speed”. A tick speed is the number of times the game loop is run in one second (Nystrom, 2021). *Minecraft* has a standard tick speed of 20 per second which means that the length of the one in-game cycle is 24,000 ticks (24 in-game hours) / 20 (to account for the amount of ticks in a second). This equation equals 1200 seconds, or 20 minutes for one full cycle. *Minecraft* breaks its cycle into four sections: day, dusk, night, and dawn. The lengths for each section is presented in real time as follows:

- Day - 10 minutes (exactly half)
- Dusk - 50 seconds (no enemies spawn but the player can sleep)
- Night - 8 minutes 20 seconds (enemies spawn, player can still sleep)
- Dawn - 50 seconds (enemies take sun damage)

Although *Minecraft* displays a slightly longer day time than night, the cycle length in *Dying Light* is much more top-heavy in regards to the distribution of day and night time. The day time lasts a staggering 64 minutes while the night only lasts for 7 minutes, as noted by the lead game designer (Dunning, 2014). The choice for such a difference in time is due to the extreme change in difficulty.

The Strengths and Weaknesses of How the Day/Night Cycle Is Utilised

The strengths of this cycle category is that the player is encouraged to venture into new challenges with the incentive of reward. This reward can also be rejected in return for a safe haven for the player to regroup for the next enemy attack. This risk and reward system offers the player choice and defines the gaming experience based on their real-life personality. Another strength is that this cycle should keep the player's immersion at the maximum because a player will choose to go inside or stay outside based on their current level of engagement. If they are lacking engagement then they will seek to increase this through challenge, whereas if they are burnt out then going inside will be the best option to maintain immersion.

The primary weakness of this cycle category is that it can alienate players of differing skill groups from each other. For instance, a player who does not fancy their chances will constantly stay inside during night hours and remain at a steady level. However, a skilled player may always seek an added challenge resulting in a much faster progression. In order to prevent this alienation, the designer must ensure that the risk and reward is balanced so the player's play style does not heavily affect their progression speed.

3.4.3 Cycle Category 3 - Added Realism - *Grand Theft Auto V* and *Metal Gear Solid V*

Examples Chosen

The examples chosen for this cycle category could have been a number of high quality, high budget open world games. The two chosen were chosen for their exceptional reputation in the domain of third person open world games. Firstly, *Grand Theft Auto V* (2013) is widely regarded as one of best and most significant video games ever released and its portrayal of a living, breathing world is just one of its excellent features. It uses its cycle to replicate its already stellar interpretation of Los Angeles by changing the environment subtly around the player. These changes can range from traffic increasing during "rush hour" times, pedestrian types being more common than others, and certain establishments opening and closing. Although all these changes seem insignificant individually, collectively they add a depth of reality not seen before in gaming.

The other example in this domain is *Metal Gear Solid V* (2015) which showcases another vast open world. The main features of this game's cycle is to invest players into the stealthy atmospheric world. The game makes use of its stealth genre and matches its day/night cycle to improve this. It does this by featuring dynamic weather conditions that are affected by the in-game time, for instance thunderstorms could only exist during night hours and sandstorms in day (see Fig 8.). As well as this, the passage of time is an added realism for the player as realistic weather conditions will immerse them in the stealth mindset to use darkness and distractions against the enemy.

Fig 8. Realistic weather conditions shown in *Metal Gear Solid V* (2015)

Examples Cycle Length

Through playtesting it can be determined that the length of *Grand Theft Auto V's* cycle is 48 minutes. The constant changing of the environment and NPC behaviour makes it incredibly hard to find where each "state change" of the cycle is. Perhaps a game of this magnitude no longer has the need for states and the cycle itself is fully dynamic.

The playtest of *Metal Gear Solid V* found the cycle to last around six hours. Therefore, this game must feature a similar passage system as *Grand Theft Auto V* due to its slow progression in the passage of time.

The Strengths and Weaknesses of How the Day/Night Cycle Is Utilised

The strengths of utilising this cycle category in a day/night cycle oriented game is due to the added depth it gives the game world. In the opinion of the author, if the player believes that the world they are in could function independently from their involvement then the game designer has excelled in immersing the player.

The weaknesses of this cycle is that the player may not notice or care for the subtle changes to the world they are inhabiting. Instead the changes may frustrate the player as night time could bring about unfavourable changes. For example, the night brings about a reduction in vision and potentially the spawn chance of their favourite vehicle drops dramatically.

3.4.4 Cycle Category 4 - Added Opportunities - *Far Cry Primal*

Example Chosen

The example chosen for this category of cycle is *Far Cry Primal* (2016). Much like other open world games developed by Ubisoft, this game features the added opportunity to collect rewards and find hidden features otherwise unavailable in the reverse time state. This is an encouragement for players to venture to the same area multiple times to monitor the change in scenery. This could also be a choice from the developer solely to increase playtime and not risk the game being completed too quickly.

Example Cycle Length

The cycle length for *Far Cry Primal* is unlike any game play tested before. The day/night cycle does not finish after a set time. Unlike games reviewed prior, the cycle does not have a top-heavy day time like *Dying Light*, nor does it dynamically change the length of state segments like *Don't Starve*. Instead, *Far Cry Primal* features a clock that increases speed based on player movement. The findings through playtesting found that the cycle length was directly impacted by the rate the player moved. This is an incredibly novel approach to cycle lengths and could be viewed as ground-breaking in the aim to maintain player immersion. The reason for this is that the player will only be subject to difficulty change if they are actively looking to progress themselves through the story i.e. traversing rapidly.

The Strengths and Weaknesses of How the Day/Night Cycle Is Utilised

The strengths of this particular cycle is that the player dictates their own speed of progression. For example, if the player wished to take their time with quests then the game environment would match their slow traversal. This is a strength as it directly opposes the negative choice made by cycle category 2 to alienate players of differing skill levels.

However, a weakness is that the developer should be padding out the game more fairly with rewards for the player. Permitting the player to certain rewards based on time constraints and not skill can be a discouragement for progression which in turn can lead to boredom. However, if the player does not choose to wait for the change in time then they will be missing rewards laid out for developers. Therefore, the player suffers a negative no matter the choice they make during this cycle category.

3.4.5 Cycle Category 5 - Aesthetic Only - *Forza Horizon 5*

Example Chosen

The example for this category is open world, driving simulator *Forza Horizon 5* (2021). The opportunity to review a game only recently released will give the opportunity to assess how day/night cycles are being created in the modern age. Through playtesting it has become clear that this game's two primary focuses are: the quality of driving controls and the realism of the driving area. The game's setting is a fictitious, modern day Mexico and so the quality of a day/night cycle would help to suspend the player's disbelief when traversing the vast world before them. However, although the believability of the day/night cycle is second to none, the added features it brings are futile. It is inferred that the addition of such a cycle is for aesthetic only as "night races" can be accessed during day time with the game artificially changing the time as the race begins.

Example Cycle Length

The cycle length of *Forza Horizon 5* lasts for just over 90 minutes. The cycle is split into a heavily disproportionate day and night time. When equating the percentages of this cycle it was found that approximately 7.1% of the cycle is set in the night state. This is the lowest percent of all the games reviewed, with the *Dying Light* cycle approximating at 9.9% making it the next proportionally shortest night state in comparison to day.

The Strengths and Weaknesses of How the Day/Night Cycle Is Utilised

The strengths of this cycle category is that it harmlessly engrosses the player in a world that replicates the natural passage of time. The increased length of the day time could be seen as a strength as it allows for better visibility for the majority of the game time.

However, the reverse argument is that the night time is too short and the noticing of this from players means that the intended suspension of disbelief from developers has been shattered. The outcry for an extended night time has been vocalised through various online forums (PixelatedDread, 2022) (Skyline_R32_187, 2022). The reason for these outcries is that the aesthetic of the game is at its peak during these night hours.

3.4.6 Other Popular Brief Examples

Cycle Category 6 - Real Time Clock - *Animal Crossing* and *Pokémon*

The last category to be discussed briefly is the use of real time clocks as seen in the Nintendo games *Animal Crossing: Wild World* (2005) and *Pokémon Diamond Version* (2006). These games both feature in-game clocks that are entirely reliant on the device's clock. This allows for encounters only available at specific real life times. This can extend from specific character appearing, to access of certain areas being permitted. This method of cycle allows for an in-depth simulated experience between the player and the game world.

3.5 Conclusions

The influence of the cycle categories will be seen in the design and implementation stages of this report where the cycle category will be selected from one of the many reviewed above. The experimentation aspect of this report will also require the use of multiple cycles to be tested to find the most optimal method of cycle. In order to directly monitor player engagement the cycle times will be split into: quick, medium, and slow with the medium being the control time and the other two being half and double this. During implementation a control time will be selected based on ongoing functionality testing to see how time best relates to the size of the play area.

To conclude, the review of both literature and video games in the research domain has been extremely beneficial to now establish requirements for the product. The reliance on this literature will allow for a product to be created with a day/night cycle that is based on the relevant psychological factors through attention span research. The flow state knowledge gained through studies of immersion will be useful in maintaining player engagement while playing the game demo.

Chapter 4 - Establishing Requirements

4.1 Requirement Specification

4.1.1 System Purpose

The system's purpose is to experiment how a day/night cycle can affect a player's attention span.

4.1.2 System Scope

The system itself will be housed in a proof-of-concept video game entitled, *Half-Blood*. The game will feature vampires that only come out to attack the player at night.

4.1.3 User Experience

The user experience is intended to be relaxed and calm during the day. However, during the night the player should experience fear and engagement.

4.1.4 User Interface

The system is being designed primarily as a PC game. The tool choice being Unreal Engine (Epic Games, 2022) allows for the distribution of Windows, Linux, and console applications. The choice to release only for Windows has been chosen due to its popularity as an operating system, and also the ease of packaging for this OS in comparison to others.

The system itself will be able to be downloaded and opened on any competent computer due to the current small scale of the project. However, a mid to high range computer will be required to run the game to an acceptable frame rate.

4.1.5 System Requirements

High-Level Functional Requirements

Below in Table 1. are the functional requirements of the product and their corresponding priority. Each function has been presented in a high-level format to best clarify the desired outcome. The priority of each function has been taken from the MoSCoW methodology (Clegg and Barker, 1994) and the functions have been ordered by this method.

Table 1.

No.	Functional Requirement	Priority
1	Day/night cycle	Must
2	Multiple lengths of day night cycles	Must
3	Intuitive AI	Must
4	Theme appropriate for gameplay	Should
5	Detailed enemies	Could

6	Aesthetic architecture	Could
7	Menu screens	Could
8	Multiple levels and/or multiple maps	Won’t

Detailed Functional Requirements

Below in Table 2. are the details of each functional requirement and the sub-requirements that they break down into further. These sub-requirements will also have their own priority which will allow for better time management when tackling each function individually.

Table 2.

No.	Functional Requirement	Priority
1	Day/night cycle	Must
1.1	Track the current time	Should
1.2	Dynamically adjust lengths of each time state	Could
2	Multiple lengths of day night cycles	Must
2.1	Ability for user to choose from a set of times	Must
2.2	Ability for user to choose any time they like for the cycle length	Could
3	Intuitive AI	Must
3.1	Behaviour dependent on in-game time	Must
3.2	Sense of free-will	Should
3.3	Interaction with each other	Could
4	Theme appropriate for gameplay	Should
4.1	Vampiric weapons and attacks	Should
4.2	Sound effects and ambient sound	Could
5	Detailed enemies	Could
5.1	Animation variety	Could
5.2	High quality, distinguishable character models	Won't
6	Aesthetic architecture	Could
6.1	Enemy living space	Could
6.2	Detailed building with functioning insides	Won’t

7	Menu screens	Could
7.1	Display start screen	Could
7.2	Display death screen	Could
7.3	Display win screen	Could
7.4	Display pause screen	Won’t
7.5	Display settings screen	Won't
8	Multiple levels and/or multiple maps	Won’t
8.1	Levelling system	Won't
8.2	Story mode	Won't

Usability Requirements

Table 3.

No.	Usability Requirement Description
1	Menu buttons work as expected, always send user to required area
2	Controls are responsiveness and movement matches input
3	Attacks are properly tracked and any contact made results in damage
4	Game runs smoothly due to efficient code, no crashes
5	Enemy AI always follows the intended path on behaviour tree
6	Graphical user interface is the appropriate size and colour for the user to see at all times, while not acting as a distraction
7	Game can be ran and opened on lower end machines to prove conciseness of the code

4.2 Justification of Requirements

The justification for the outlined requirements has been based on the studies conducted in the prior literature review. The requirements have been sorted based on criteria of functionality and usability. In order to differentiate the importance of each requirement, and therefore how much effort to expend in each, the requirements were ranked using the MoSCoW system. This will allow the correct amount of effort and time for each requirement in order to maintain a balanced workflow.

Functional requirements 1, 2 and 3 were deemed necessary to create a game that could be tested based on its day/night cycle. Therefore, the day/night cycle itself is a core component

to the game because it directly affects the research into understanding how the player reacts to differing lengths of said cycle. Continuing, this is why the other functional requirement of allowing for multiple lengths of day/night cycles was classed as a must.

Functional requirement 4 noted that the theme of the game should be appropriate for the gameplay to allow the player to be immersed. Therefore, the game will feature a vampire theme to make sure the player is not questioning why the game requires a day/night cycle. This can be linked back to Chapter 2 on the topic of *cycle categories*, in this case the final product would fall into the categories 1 (integral to gameplay) and 2 (added challenge). This category has been selected because it will have the most impact on the player and their decision making. For instance, if the game was created to be in category 3 (added realism) then, with the time and technical constraints, it would be challenging to create a game that the player would notice a change in gameplay. A game such as GTA V requires an incredible amount of minute details to slightly alter as day turns to night, and vice-versa, to create a realistic atmospheric change. There are not enough resources or time to create an effect similar to this.

Functional requirements 5, 6, and 7 mention that the game could become a better product through the introduction of detailed enemies. This suggests that the enemies need to resemble vampires but don't necessarily need to be detailed in such a way that would take up processing power. It would also be a poor use of time as the resources and energy could be better used in making the day/night cycle more optimised. In implementation, it will be made clear that the enemies are the opposition of the player through use of colour. In addition, the physical structures in the game will be designed in the same way as the enemies where they will feature the barebones in terms of shape but not in terms of high quality textures and materials. As long as the player has a good understanding as to what each building is meant to represent then they require no more detail. Finally, the UI of the game could be developed in a way that keeps the player involved, however it does not need to be treated as an integral game component.

Finally, functional requirement 8 was rejected due to the game being a proof-of-concept so there will be no need to have multiple levels or any hint of a progression system. Also due to the studying of the players, multiple levels would be too hard to track their attention over a longer period of time. As well as this, only one area will be accessible during the gameplay demo. This will limit the player to one location where all of the action can take place. Potentially populating a large area, or multiple areas, will be too demanding for a project of this size and it also will give the player too much to explore. The game will be created for the purpose of demoing a single day/night cycle so adding an explorable world will deter the player from engaging with the action elements which will be brought about by the changing in the cycle.

To conclude, the usability requirements have been established to ensure a smooth user experience. The game can operate alone with the just the functionality requirements in place, however the user experience will be damaged if their usability has not been accounted for.

Chapter 5 - Tools and Techniques

5.1 Game Development Tools and Language Justification

When developing a 3D third-person game, the two best options for intuitive design are Unity (Unity Technologies, 2022) and Unreal Engine (Epic Games, 2022) as noted by multiple sources (Kean, 2022) (Tyler, 2022). To decide which tool would be the most effective in producing a working demo under the time constraint, a list of requirements would need to be drawn up.

The list of requirements for the game development tool are:

- How expensive it costs
- Familiarity with the software in the past
- User friendly interface for design
- Ability to sell the experience to the user on a small scale
- Graphical prowess
- File size and ability to run on lower end devices

Firstly, Unreal Engine was compared against these list of requirements and instantly it was noted how effective a tool it is. The price, familiarity, and user friendliness are all answered faultlessly by this piece of software. Furthermore, the tools that Unreal boasts mean that the user experience can be enhanced through high quality graphics and physics engines. The worlds that have been created in Unreal all share the same 3D graphical precision unavailable in other free engines.

However, it would be unfair to not inspect Unity to see if it should be selected as an alternative. Firstly, the C#, language that Unity is built on, allows for object-oriented benefits such as inheritance and polymorphism. In addition, the similarity in text-based coding to previously used languages such as Java would make the conversion to a new syntax a smooth process. Most importantly, the stripped back base code allows for a modular experience. This means that the user can pick and choose what they wish to add meaning no bloatware can exist in the final product's file size. Finally, the lack of a payment required would also strip away any risks associated with trialling the product before use.

Nevertheless, the benefits that come with Unreal's component architecture means that the entire program is decoupled. As well as this, the realism tools that Unreal has built-in means that the positives that Unity possesses are simply negated by the prowess than Unreal boasts in effective game development tools.

In conclusion the game development tool chosen will be used in order to create the game demo to the highest level. Therefore, after weighing up the alternatives, Unreal Engine will be used to ensure a high quality 3D game is created that the player will be immersed in enough to suspend their disbelief. It is also an appropriate tool due to the experience the author has using the engine. As well as this, the engine's usability and user-friendliness when in the development stage will be integral to creating a fully usable proof-of-concept video game.

5.2 Preliminary Design Ideas

To describe the design process in the most effective way a game design document will be drawn up as there is no clearer way to emphasise or explain design decisions taken. Along with this documentation, sequence diagrams, flow charts and class diagrams will be created to allow a more visual representation of how the game will flow.

These documents will be used to assist with creating clear structures that the game must follow during the implementation stage. As well as this, they will be used to evaluate how closely the design was followed and if the technical challenges of implementation negatively impacted the vision of the design itself.

Section 2 - Design, Implementation and Testing

Chapter 6 - Design Process

6.1 Flow Chart of AI Reacting to Time

Due to the AI of the enemies being the driving force behind the day/night cycle, a flow chart describing the flow of their behaviours will be imperative to the implementation of a behaviour tree.
The flow charts below (Figs 9.-12.) show each of the different states the enemy can be in, the individual actions based on these states has been simplified to keep the diagram concise and also due to the numerous ways each task can be programmed.

Dawn Section

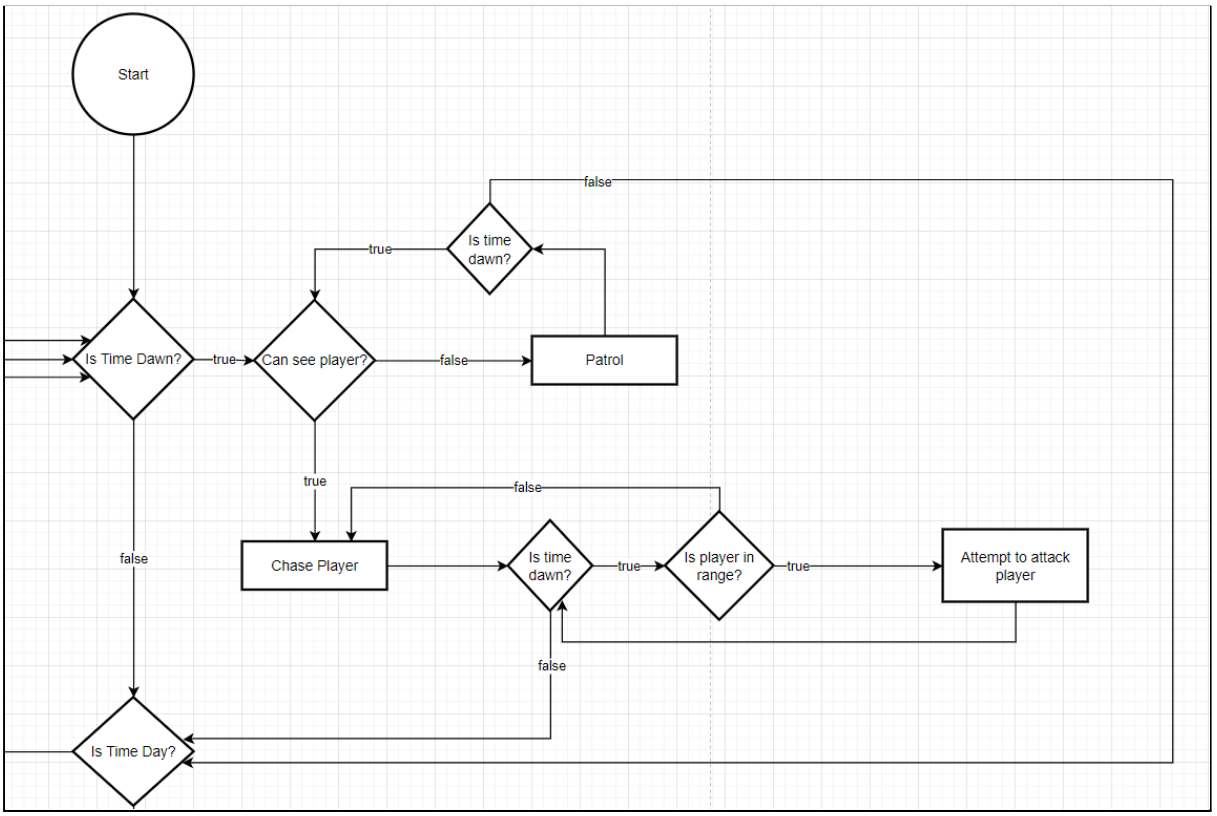


Fig 9.

Day Section

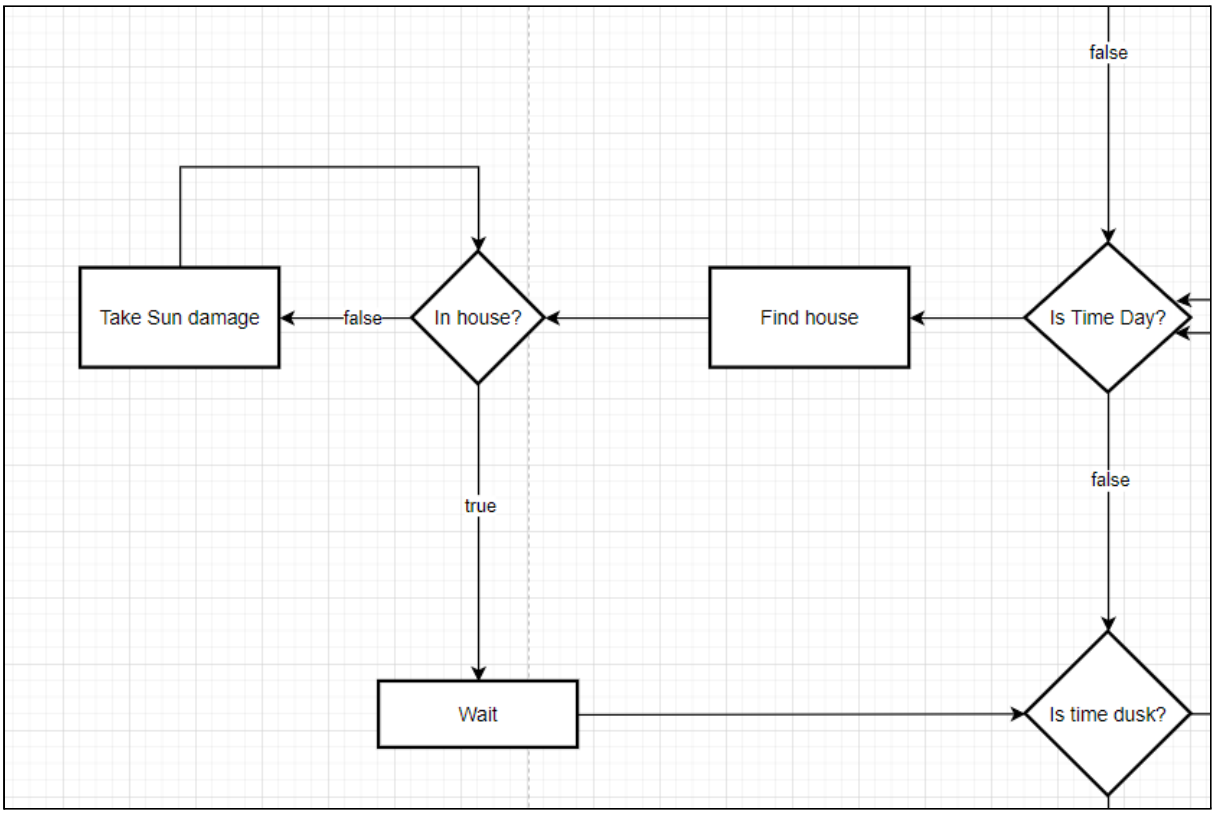


Fig 10.

Dusk Section

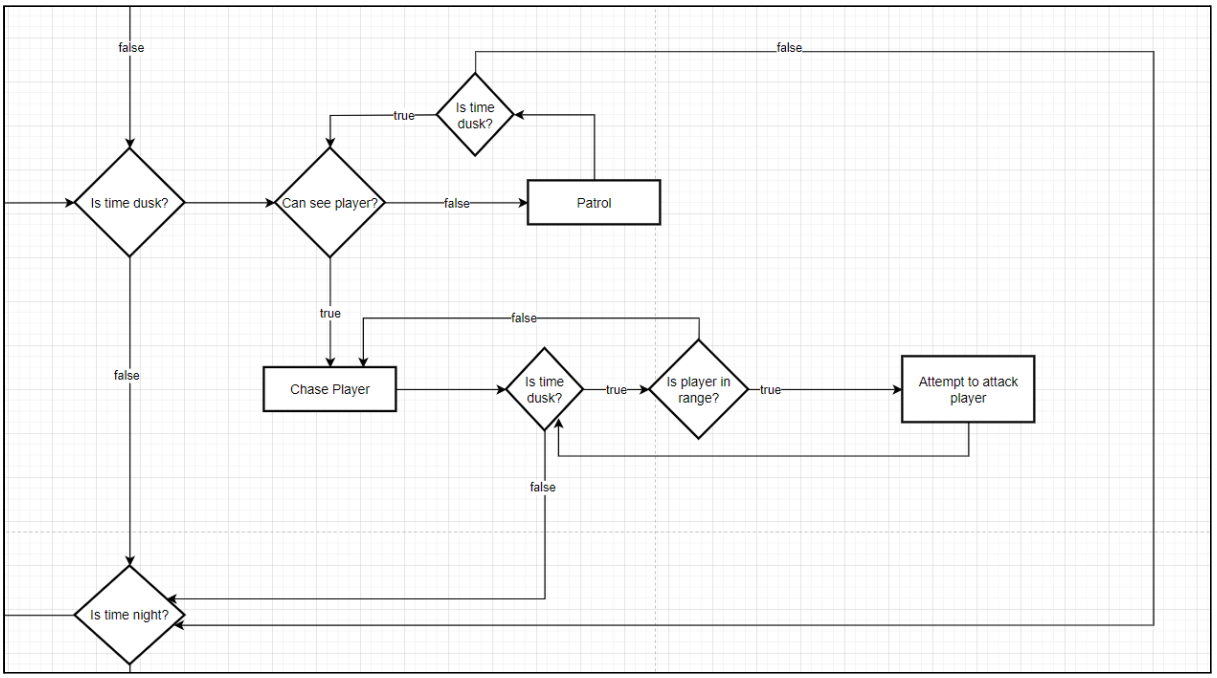


Fig 11.

Night Section

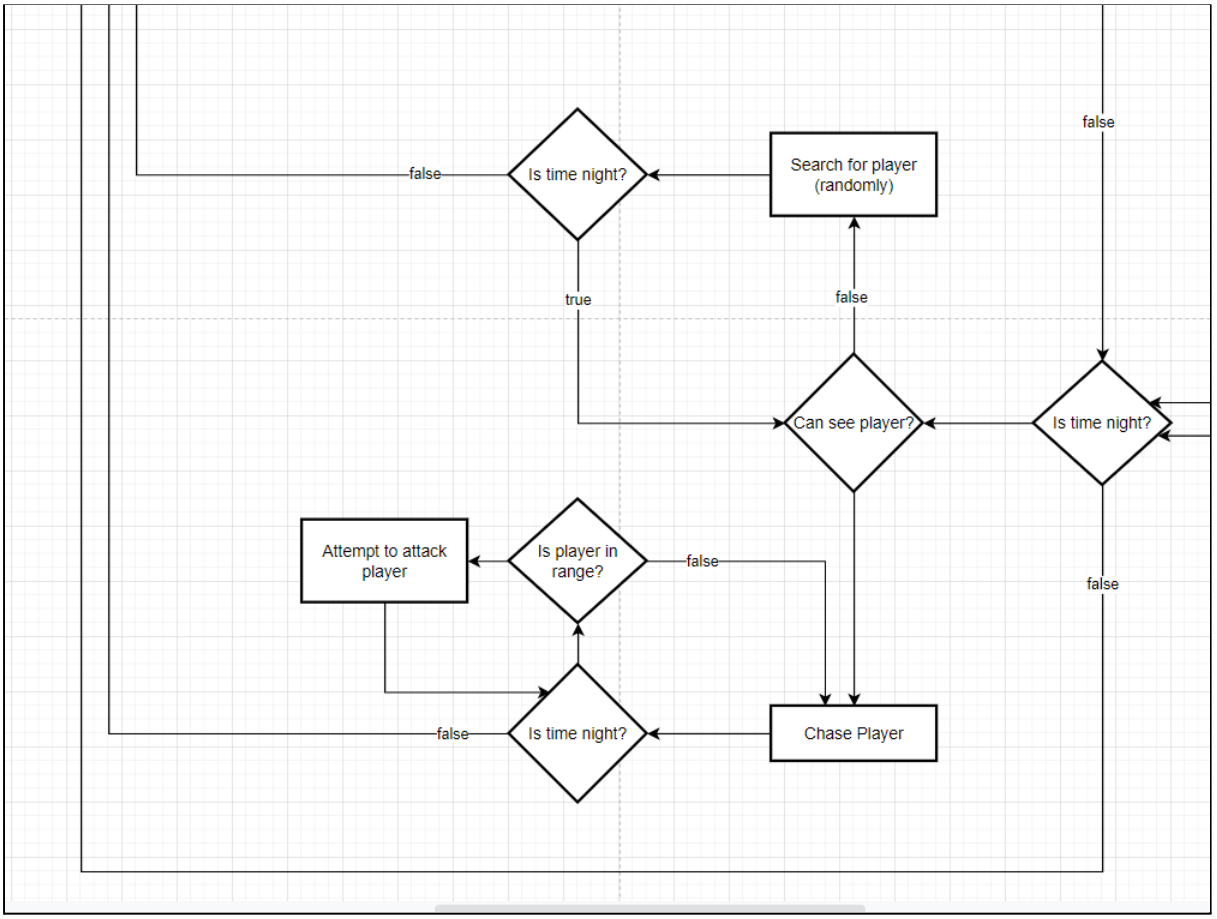


Fig 12.

6.1.1 Description of AI

To replicate the above flow chart, the AI will make use of the “AI Controller”. The AI Controller is a feature of Unreal that acts as the brain of the pawn it is controlling. This is an extremely useful tool as it allows each enemy to have a shared instruction set, but their behaviours are all independent of each other. For example, the enemies will all have a “Can See Player” variable that is either true or false. However, this variable could be different between each enemy depending on the eyeline they are currently holding.

6.1.2 Behaviour Tree

One of the advantages of Unreal is that it makes use of behaviour trees. Behaviours of AI can not only be changed using this tree, but also the level of detail that Unreal shows the developer with this tool makes it an outstanding feature. For example, Fig 13. shows which branch of the tree is being run by the AI and why it is passing this branch’s particular requirements.

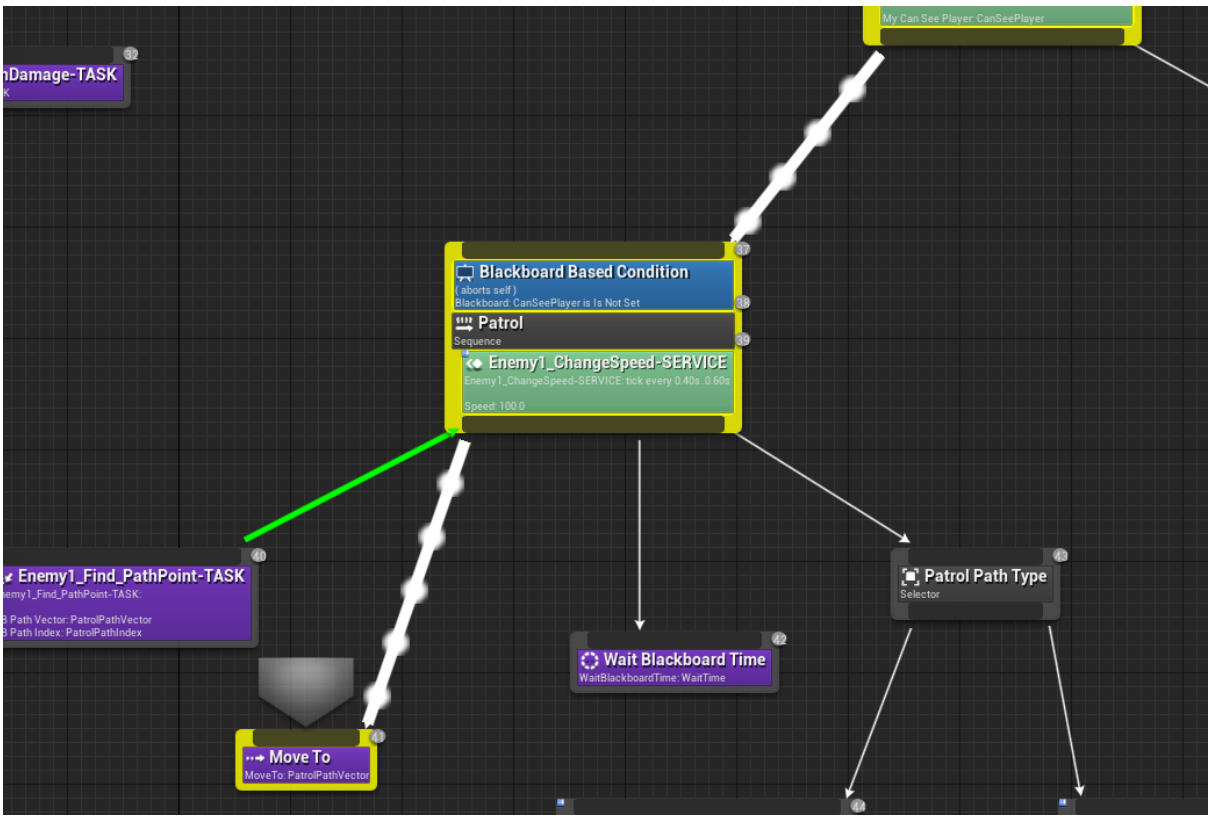


Fig 13. Example of a behaviour tree (taken from game during planning phase)

6.2 Class Diagram

A class diagram has been created to plan for inheritance and association between classes in the Unreal Engine. The variables and functions have been shown alongside their visibility, passed values, and data types. (see Fig 14. below).

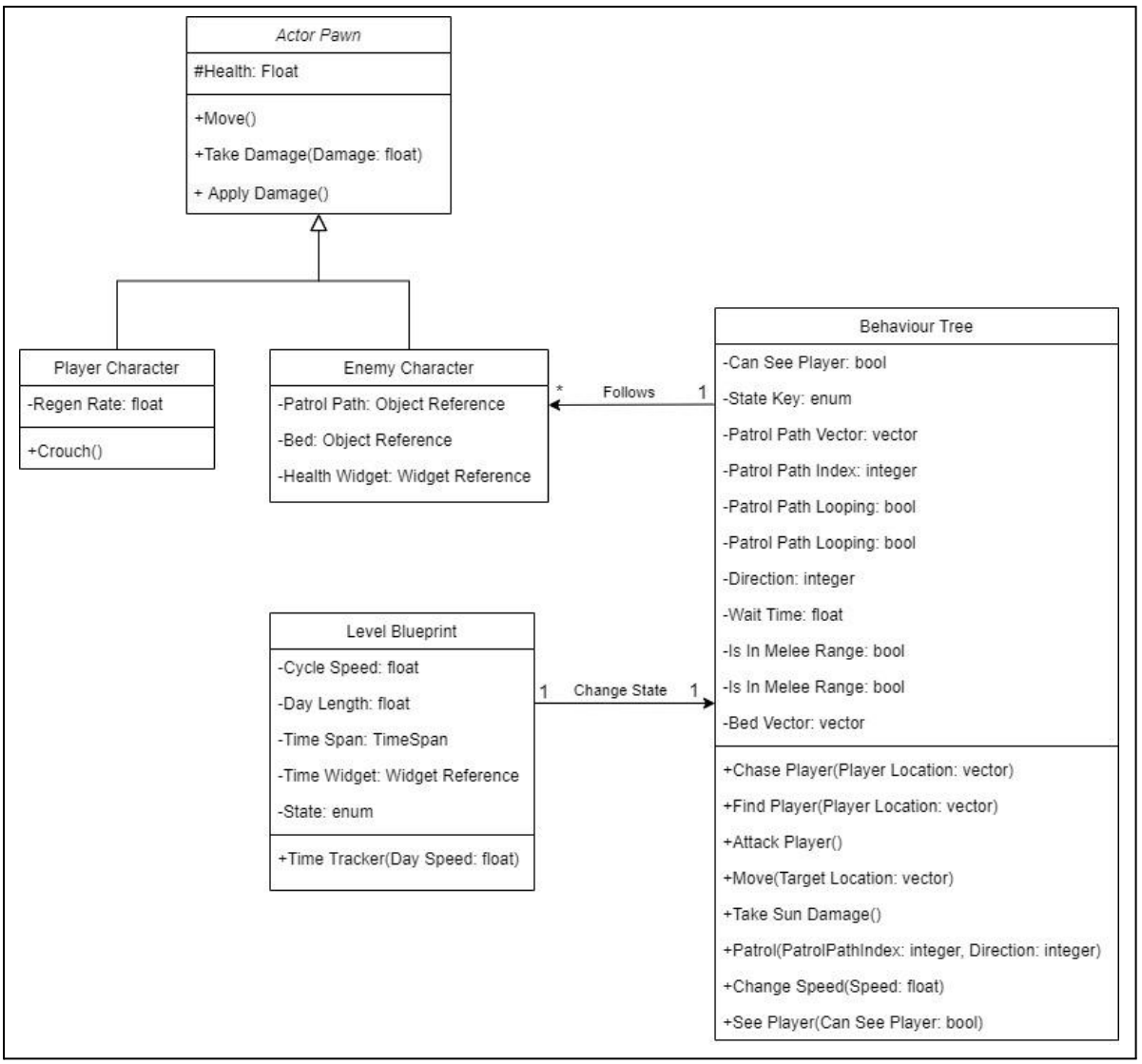


Fig 14. Class Diagrams of the player, enemy, and the level itself

6.3 Game Design

Full Game Design Document can be found in [Appendix B]

6.3.1 Game Overview

The game has the simple premise of defeating enemies while trying to stay alive. The complex feature is that the enemies' behaviours change based on a day/night cycle. This behaviour has been programmed by using behaviour trees to dictate a state based on the current in-game clock's time.

The game has no win condition or clearly defined end goal for the player to aim for. This is because the game is a proof-of-concept showing how a player's mental state can be affected based on the differing speeds of a cycle. The player could potentially "win" by killing all the enemies, however during testing there will not be enough time to do this over the course of one cycle.

If the player dies then the game is over and will need to be restarted. There are many ways to prevent dying, for example the player can use stealth to hide behind rocks and slowly pick off each enemy one by one. They can also use a combination of light and heavy attacks to counter the enemies attacks. Finally, they can hide away after combat to slowly regenerate health at a constant rate.

6.3.2 Interface Design

Controls

The default controls for *Half-Blood* have been selected as they make use of the industry-standard PC controls for movement, a WASD format will be used for directional motion. On PC the attacking controls will make use of the mouse as this will be in hand by the user because it is also used for adjusting the camera. Additional keys used for quitting and pausing will be assigned to the “escape” and “P” keys respectfully. See Fig 15.

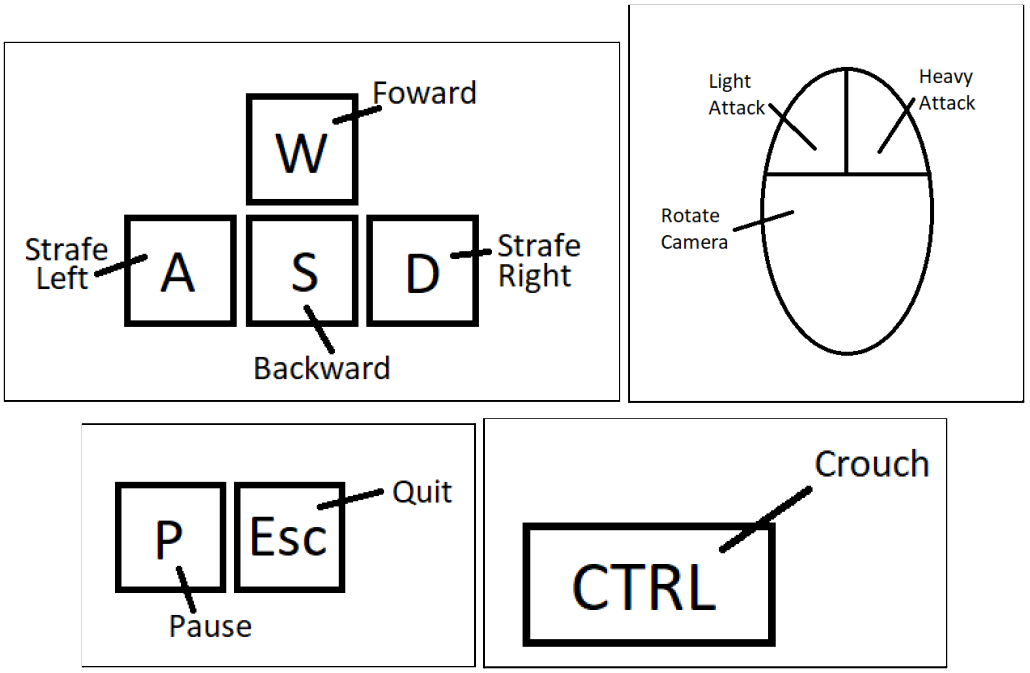


Fig 15. Keyboard and Mouse control scheme

Unreal Engine also has the feature to allow for multiple types of controller support, for simplicity a PS4 controller will be used to show the button mapping (See Fig 16.). The option for a controller means that the user testing will flow much smoother due to the flexibility in choice the user will have when playing.

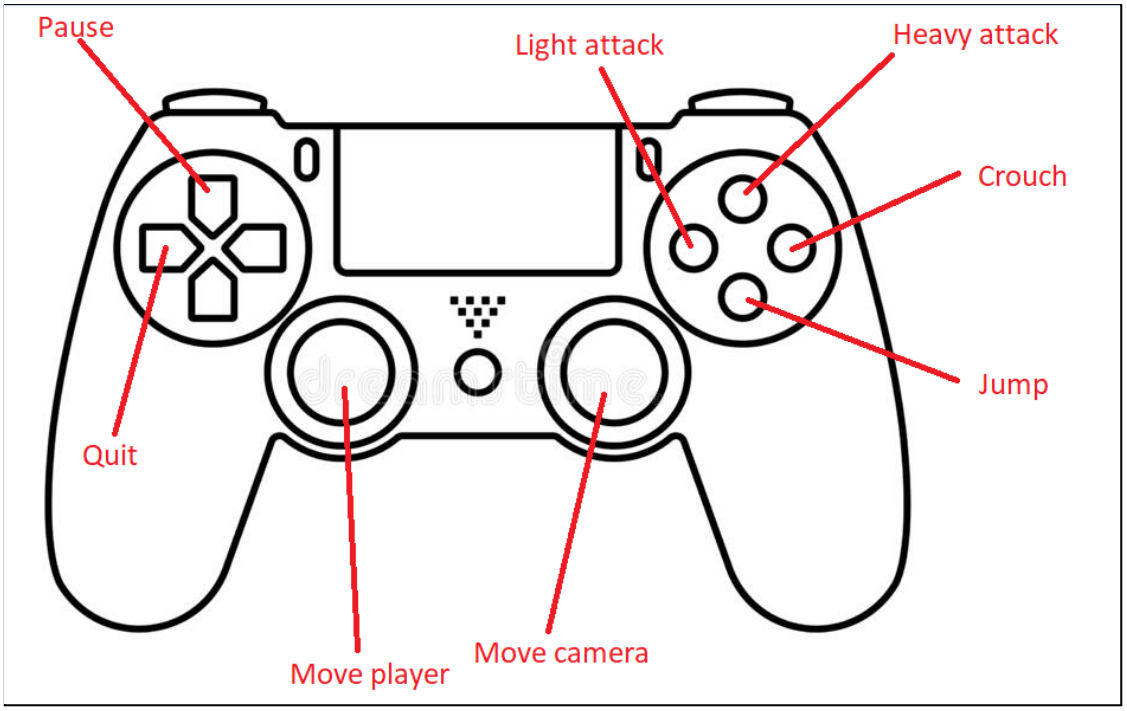


Fig 16. Gamepad control scheme

HUD

The HUD will be largely empty in order to allow for undistracted gameplay. The need for large amounts of information displayed would not be as beneficial for the user as a large view of the game world. The only information that is useful enough to the player, to justify being displayed at all times, is the current time in-game and the player's health. These will both be displayed in the game's top corners to allow for an unobstructed view.

Menu Screens

To avoid time being spent in the wrong areas and also to avoid venturing out of scope for a proof-of-concept demo, there will be extremely limited menu screens that will be primarily used for breaking up instant transitions.

The first menu will be a start screen, that breaks the transition between starting the game up and instant gameplay. This screen will only have the two buttons: start or quit. This will also be the place to explain the control scheme in a simple bullet point list as well as an image similar to Fig 16. that explains the controls on a gamepad.

The only other menu will be the game over screen that will inform the player that they need to quit the game in order to restart. Restarting the game via a quit command will require less out of scope programming, this is because the enemies will need to be reset to their original health and positions. Also, the game will not be very intensive so a restart should require very little load time.

6.3.3 Character Design

Due to the detail in foliage that Unreal Engine allows for, it would be best to keep the characters easily visible while also easily distinguishable from each other. The choice of a default white for the player and red for the enemy seemed suitable. In addition, different body types will further enforce the disparity between the player and enemies. The green, grassy backdrop will allow for both the player and enemies to remain visible at all times.

Player Design

For simplicity the player will be a solid white mannequin with a different body type to the enemies. The player will also always have a stake in their hand ready to use on the enemies (see Figs 17. and 18.).

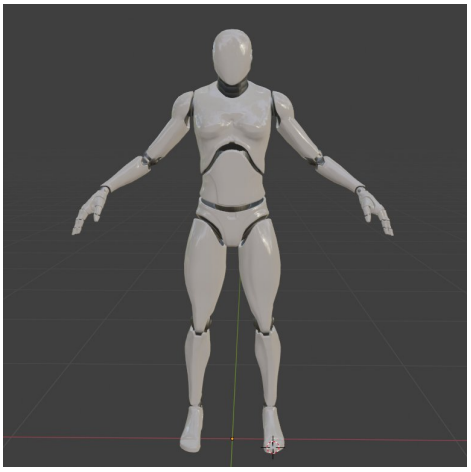


Fig 17. Player character model

Fig 18. Stake weapon model

Enemy Design

The enemies will have a larger body build to the player and be coloured bright red. They too will be holding a stake in their hand to attempt to kill the player with (see Fig 19. below).

Fig 19. Enemy character model

6.3.4 Map Design

The map was designed to allow for interesting combat scenarios. This is due to how the buildings are placed and how their distance to each other is far enough that the player won't be swarmed, but also close enough to allow for the player to be pincered if not careful.

The map also allows enough room for enemies to have patrol paths that are set to be around their homes, this is so the enemy can easily run into their house as day comes.

6.3.5 User Experience

The user experience is a key part of design that should be thought about during every stage of development. When implementing each requirement there should always be moments of reflection that question whether the user experience is being affected positively or negatively. Furthermore, the idea that the changes in time state should affect player engagement should be directly correlated to the entire user experience.

6.4 Algorithm Design

The basis of this proof-of-concept is the day/night cycle that constantly affects the world. Therefore, extensive research into how the time of an Unreal project can be manipulated while also affecting the Sun of said level. Of the five reputable tutorials watched (see bibliography), each one of them followed the same pattern of manipulating the games primary light source's (the Sun) Y rotation, or more commonly "the pitch".

6.4.1 Pseudocode Overview of Day/Night Cycle Algorithm

The algorithm for accomplishing a full cycle can be broken down into:

- The actual light source change
    - Adjusting the pitch of the Sun
    - Checking if the sun is going from down to up (sunrise) or up to down (sunset)

    - Loop through the sequence based on the direction the Sun is going
    - Change light source intensities based on time
- Time tracking
    - Get the speed the cycle is going
    - Multiply by 240 and add to the current time (initial time set to 12pm as exact middle when Sun is the highest point)
    - Truncate this time into a standard days, hours, minutes time system
    - Update player's clock widget with this standard time
- Enemy AI impact
    - Get the current hour in the time tracker
    - If the hour is between 5am and 7am then set current state to the "Dawn" state in all of the enemies' brains
    - If the hour is between 7am and 5pm then set current state to the "Day" state in all of the enemies' brains
    - If the hour is between 5pm and 7pm then set current state to the "Dusk" state in all of the enemies' brains
    - If the hour is between 7pm and 5am then set current state to the "Night" state in all of the enemies' brains

Chapter 7 - Implementation

The implementation of *Half-Blood* has been developed entirely within the Unreal Engine using the visual scripting programming method.
As development advanced, many iterations of the game were created in order to test functionality at each key stage. The key stages of development were:

1. Physical day/night cycle implementation
2. Trackable in-game time
3. AI is sent the in-game time constantly
4. AI patrol, chase, hide, and attack behaviours
5. Health system for AI and player
    a. Different attack types for player only
    b. Animations for attacks
    c. Behaviours for attacks
6. Crouching (stealth) system for player
7. Map design touch up
    a. Foliage
    b. Cover
    c. Houses
    d. Invulnerable starting area

Once each key stage was functional then focus shifted to the next key problem. If a latter stage of development impacted an earlier stage then focus would shift back to that stage until no issues between stages arose.

7.1 Implementation Plan

To adhere to a set plan, the above stages of development were further mapped into an implementation plan (this can be seen in Table 4.). This plan was used to ensure all areas of the requirements were met and that the product did not deviate from its intended purpose. The plan also meant that the scope for the product was kept to the maximum available within the restraints met, i.e. time.

A game log will be appended after each sitting to ensure development is both useful and recorded. This log will be reviewed against the implementation plan to evaluate the usefulness this plan held and how effective the author's development skills were at accomplishing the tasks set out.

Table 4.

Phase of development	Targets attempted	Requirements met	Milestones hit
1	Physical day/night cycle implementation	1	The Sun's Y pitch or "roll" is manipulated based on a speed given to the system. This speed can be derived from the user based on how long they would

			like the day to be (in seconds).
2	Trackable in-game time	1	The in-game time is tracked with a time tracker function that finds the time based on the Sun's position. This time is also used to send the enemy AI the correct state.
3	AI is sent the in-game time constantly	2, 3, 4	The AI behaviour tree is sent the time through the time tracker function. Depending on the hour, the behaviour tree is sent the corresponding state.
4	AI patrol, chase, hide, and attack behaviours	3, 5	The enemy AI behaviour tree features many functions to actively patrol, search for, and ultimately attack the player. These behaviours are linked to the in-game time and only take effect if all conditions are met.
5	Health system for AI and player	3, 4, 5	The AI features a widget placed above its head that reacts to the player or Sun damaging them. This widget is only displayed when the player is within range. The player's health system is regenerative and impacted by enemy attacks. The health bar is permanently displayed in the top right corner of the screen.
6	Crouching (stealth) system for player	3, 4	Player is able to hide behind rocks and can successfully sneak around enemies without detection.
7	Map design touch up	6	Bumps, foliage and textures have been added to add believability to the world. Mountain ranges to block player traversal outside the map boundaries were also added.
8	Menu system added for game over and starting screen	7	The player can enter or exit the game with the main menu screen. The player is greeted with a game over when upon dying, they can quit the game from this screen.

7.2 Algorithm

7.2.1 Prerequisites

Due to the nature of the product being a playable game demo, it was first imperative to set up a landscape. In fact in the first sitting of the game a playable area was laid out with a player start location chosen for testing purposes. The map made use of Unreal's example map which was designed to host third-person games. The play area was expanded beyond what was originally given, the size of the game was the first key choice in game design as it would directly correlate to how the day/night cycle times would affect the user experience. Of the games reviewed in the day/night cycle domain, many were large open world environments and those that weren't were instead procedurally generated. However, due to the scope of this product the map scale will have to be drawn down to a reasonable size. A square with sides of length 10,000 units each was chosen after completing functionality testing during the early stages of development due to being big enough to explore in the given time, but not large enough to invoke boredom through travelling. To note, 1 unit is the equivalent of cm with the scale used in this project,

The enemy characters designed in 6.3.3 were then instanced (See Fig 20.) from Unreal *blueprints* in order to populate the blank landscape. Instancing is a method of object-oriented programming utilised by Unreal blueprints, the blueprint system is a "scripting system based on the concept of using a node-based interface to create gameplay elements from within Unreal Editor" (Epic Games, 2022b). In short, any entity can be created with a set of scripts built using nodes (See Fig 21.) that can be placed countless times in a scene. This method of blueprinting the enemies was crucial as each enemy needed to follow the same set of instructions in relation to the game world's time.

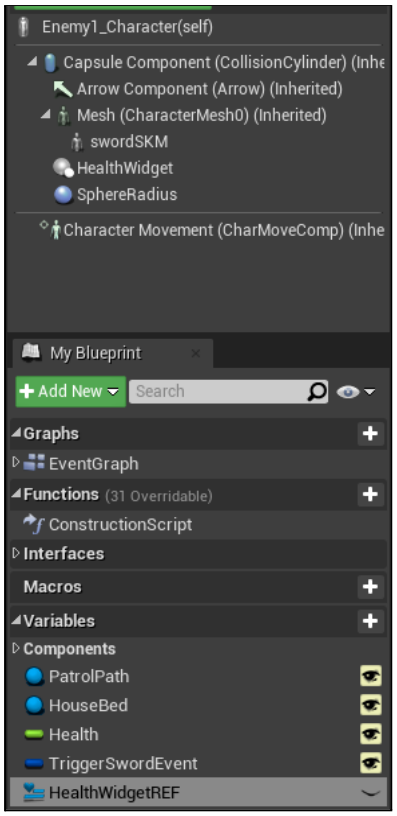


Fig 20. Showing the variables and components that each enemy character is instanced with

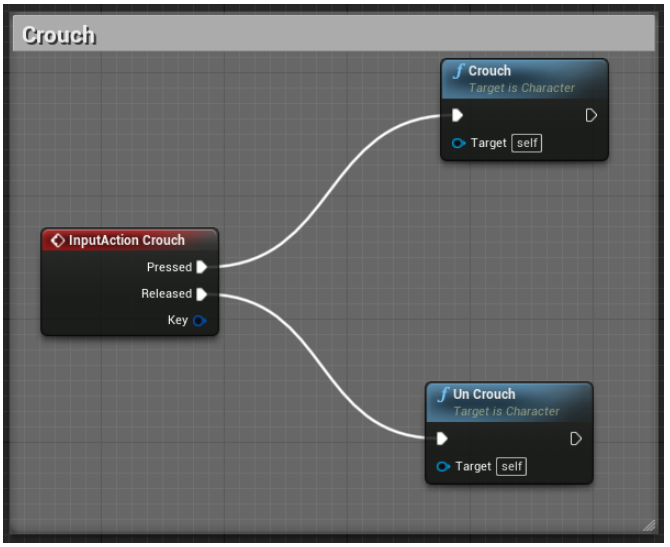


Fig 21. Showing the node based system used, this the alternative to using the C++ language

The first session of development ended with the enemies' houses being placed in a template fashion, and then research on how to incorporate a day/night cycle within Unreal Engine commenced.

7.2.2 The Cycle Algorithm - Physical

Now the landscape had been initialised, the algorithm for producing a working day/night cycle would need to be completed as it is the framework on which the rest of the game functions. Once this extremely high priority functionality of the game is completed then the enemy's will have data fed to them to dictate their behaviours. Before data can be passed, first a physical system needs to be created in which the Y axis of the in-game Sun is manipulated. Many tutorials in this domain of game development were reviewed in depth and most were useful helping understand the meaning behind the code. However, only one creator's tutorials (Matt Aspland, 2021a) (Matt Aspland, 2021b) were chosen based on the thorough explanation given in their videos, all others are listed in the bibliography as they collectively helped gain insight into the topic.

The cycle's algorithm can be broken down into key areas:

- Changing the pitch of the Sun (Fig 22.)
- Identify if the pitch is changing going into sunrise or sunset (Fig 23.)
- Loop through sequences to update a timeline (Fig 24.)
- Change light source intensities (Fig 25.)

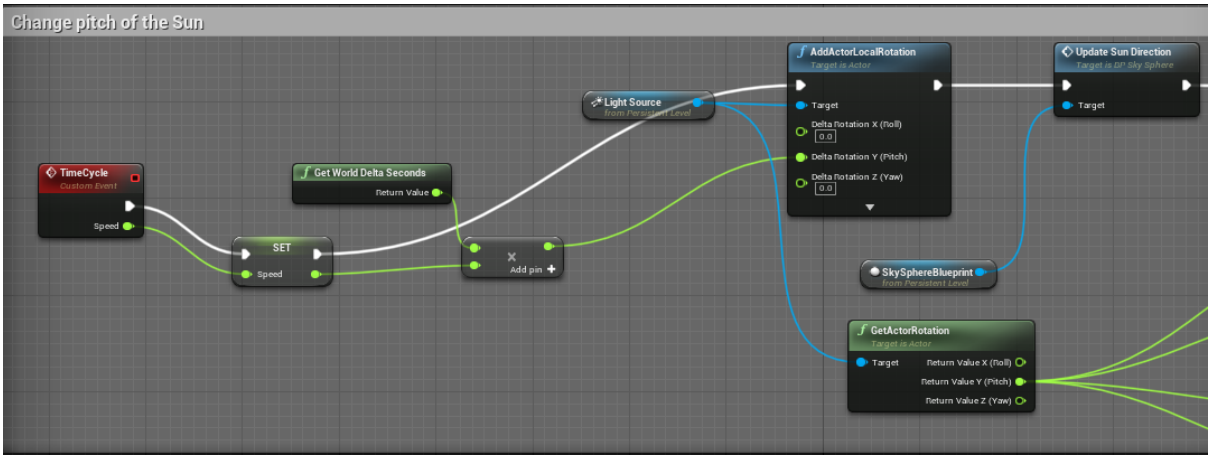


Fig 22. Showing how the light source's rotation variable is being manipulated by the World Delta Seconds (real time) and a given speed. This then updates the Sun Direction (top right) and the Y value of the rotation is got (bottom right)

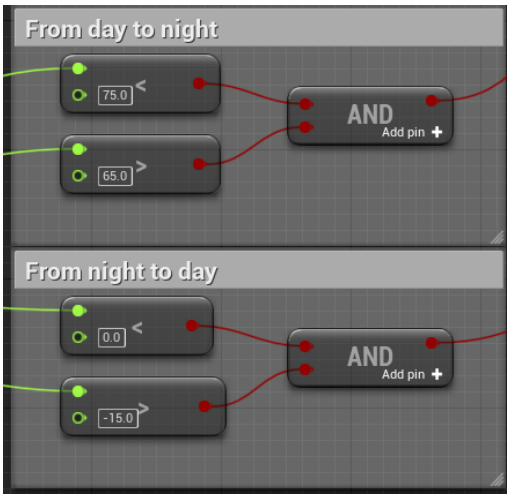


Fig 23. Showing how the Y value of the light source's rotation is used to identify if the game is going from day into a sunset (top), or if it's going from night into a sunrise (bottom)

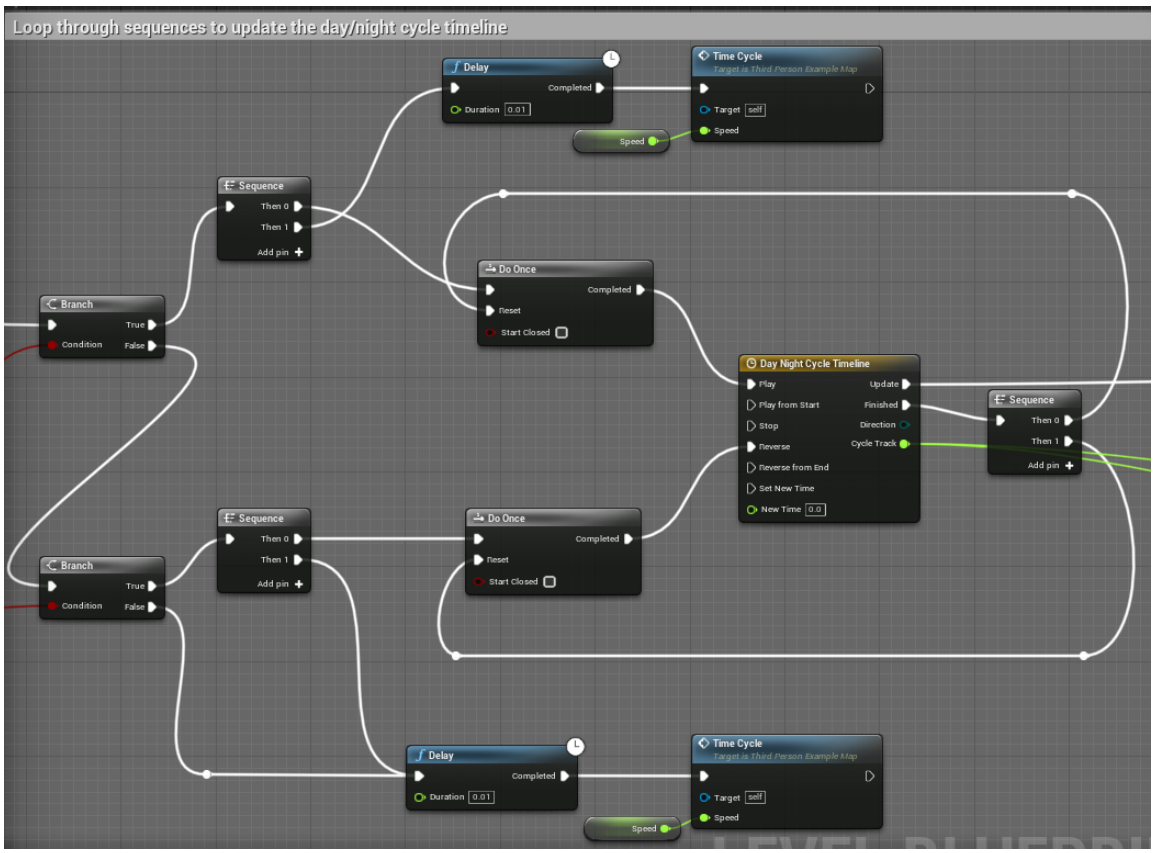


Fig 24. Showing how if the time is going from day into night (top half) then a sequence commences where a timeline is played until completion where it will then do the reverse. If the time is instead going from night to day (bottom half) then the sequence will be the reverse of the timeline. Each sequence ends with the time cycle being called with the given speed being fed in again.

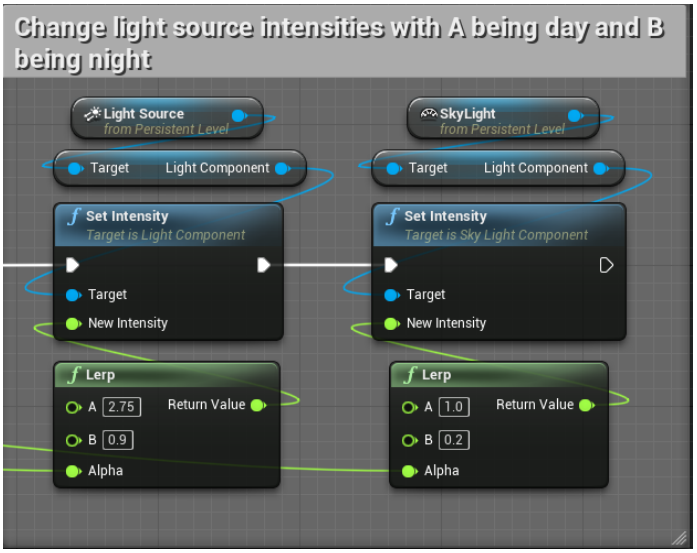


Fig 25. The end product is both the light source (the Sun) and the skylight (the overall brightness of the level) having their intensities set to a value based on the time given. The values 2.75 and 1.0 were chosen for the brightest it could be, and 0.9 and 0.2 for the darkest. The alpha (where on that scale the light intensity should be) is dictated by the value sent from the timeline.

By using this algorithm, a cycle of the Sun going down into sunset and then back up again into sunrise has been accomplished. However, there is no way the AI can be manipulated as there is no variable or function to describe what time it is within the timeline.

7.2.2 The Cycle Algorithm - Timer

As described above, there is nothing to track the passage of time. Therefore, a function will be created that allows not only the AI to be fed the time, but also the user via the use of widgets. Firstly, a custom function is created that will run as soon as the cycle starts. This function will be given the speed that the cycle is going and it will multiply this by 240 to turn this into a 24 hour system (See Fig 26.)

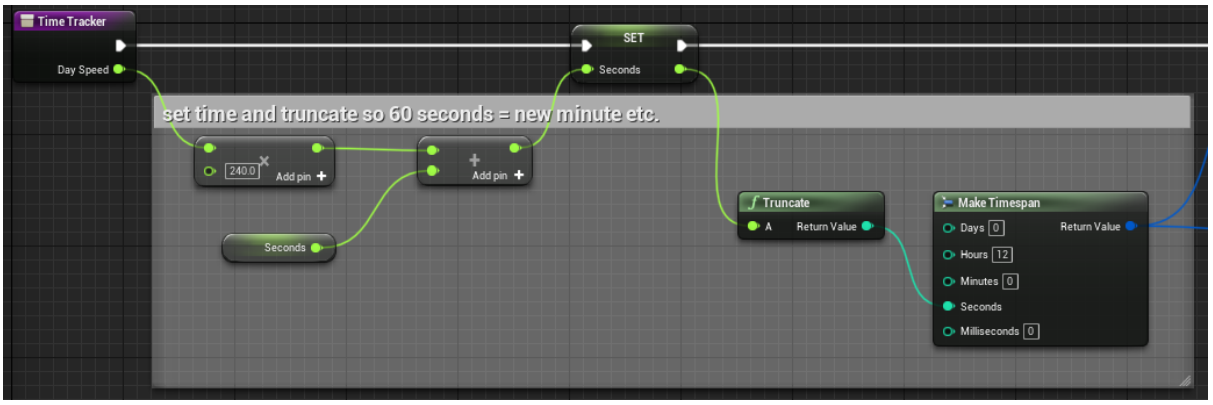


Fig 26. Showing how the time tracking function turns the speed into a time by using a "seconds" variable that is then truncated into a timespan. The timespan is initialised to start at 12:00, as this is when the Sun is at its peak.

The information for how truncation can be used to create a timespan was knowledge gained from a tutorial by the same user as the physical day/night cycle (Matt Aspland, 2021b) and also from mathematical concepts covered through studies in computer science.

This time is then sent to the user's time widget. The widget tool is a feature of Unreal that allows for an overlay of information on the screen, this is described in more detail in section 7.3.3 on the GUI.

Speed Equation

As seen above the algorithm for creating a day/night cycle, and then the time tracker function, both feature a speed variable. This variable is a float type, meaning it can represent any rational decimal number. Although the tutorials used for creating all the functional aspects of the cycle, unfortunately the equation given for how the speed variable can be found was incorrect (Matt Aspland, 2021a, 14:13). The equation given stated that if the player wished to input a day length (in seconds) then this value would be multiplied by 100 and then this total multiplied by 5. However, this was instantly disproved as incorrect when drawing the equation out as a formula triangle. When removing the 100 the formula can be expressed in a triangle formula (See Fig 27.)

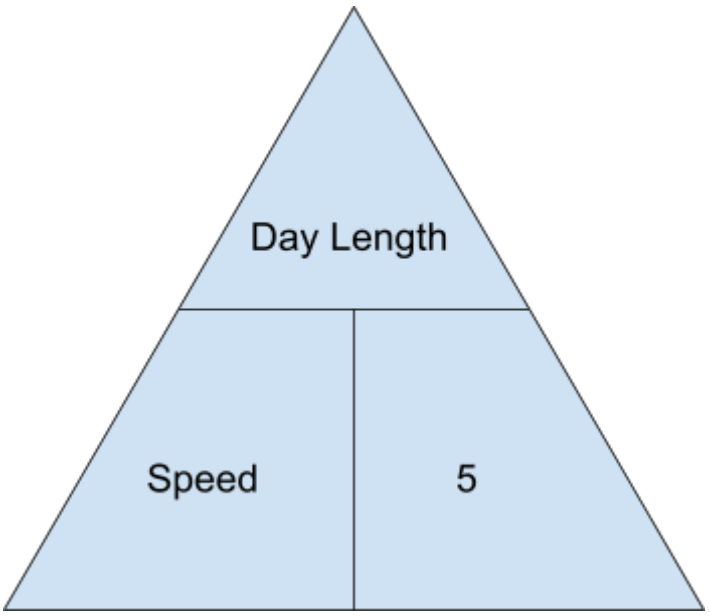


Fig 27. Formula triangle dictating how to obtain each variable used in the speed equation.

Now the correct formula had been achieved, the game was play tested with this new equation. However, it became clear that the equation was still slightly wrong due to the cycle finishing 1.5x longer than expected. Therefore it was understood that the solution was to change the 5 and into 7.5 as 5x1.5=7.5. Once this change was made, both equations were functioning as intended (See Fig 28.).

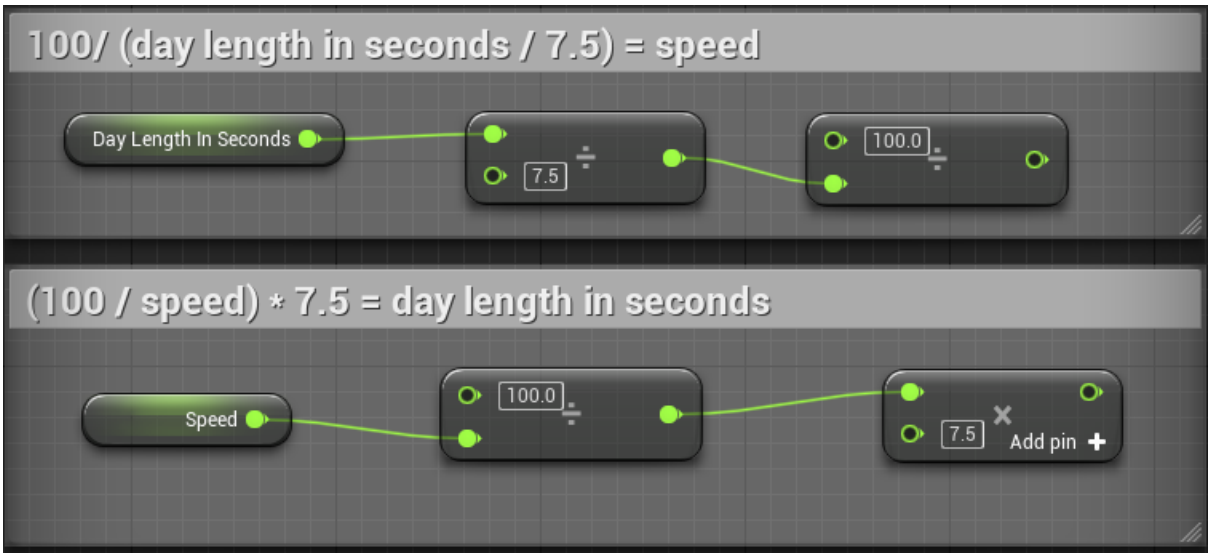


Fig 28. Detailing the fully correct formula for generating either the speed or the day length in seconds

Upon finding the correct equation, development into how the user can select different cycle lengths could commence. This will be imperative to accomplishing functional requirement 2 as it is what the majority of the user testing relies on.

7.2.3 How the AI Reacts to the Cycle Change

Now the cycle is fully functional, the time is linked, and the length of the day can be dictated by a variable; focus will be shifted into AI functionality. During the designing process, it was clear that behaviour trees would be used to seamlessly change between the desired responses of the enemies. However, it was unclear how best to send the time to the enemies.

The use of a Boolean data type would be viable if the game relied on just a day state and a night state, however due to the amount of states exceeding 2 a different approach would have to be taken. Instead, an Enum data type was chosen due to its usefulness in having a set of confirmed values, called enumerators. These enumerators would be labelled: dawn, day, dusk, and night respectively.

State Enum

When using an Enum it is crucial that every aspect of the program is using the same values. The room for error when referencing the Enum data is completely removed by Unreal due to its ability to create an Enum that can be referenced anywhere with all the values set from its creation (See Fig 29.).

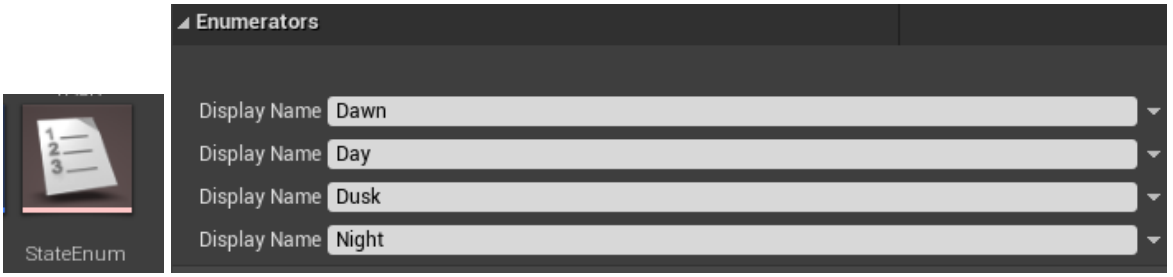


Fig 29. Showing the Enum data set that will be used throughout the program. The Enum has been given the name “StateEnum”

Sending Enum to AI

With a universal Enum setup, the task was then to communicate a change in state to the brain of each enemy. At first this was incorrectly performed by calling all the actors of the class “enemy” and changing a variable within their construct. However, when trying to dictate a behaviour change in the behaviour tree, it became clear that the tree had no access to this private variable.

The solution was found to be in giving the behaviour tree a “blackboard key” that directly corresponded to the StateEnum created earlier. A blackboard key is a feature of Unreal that allows for a variable to be changed for all members who possess the same AI (Epic Games, 2022c). The way this blackboard key was changed was by utilising the truncation of the time, as seen earlier in 7.2.2, and from this timespan the hours was taken. The hour was then put into several branches that had a condition based on whether the hour fell between certain predetermined values. These values were as follows:

- Dawn - 4:01am-6:00am
- Day - 6:01am-5:00pm
- Dusk - 5:01pm-7:00pm
- Night - 7:01pm-4:00am

Behaviour Tree

To begin, enemies were given a simple print string that returned the time it currently was, being either dawn, day, dusk or night (See Fig 30.).

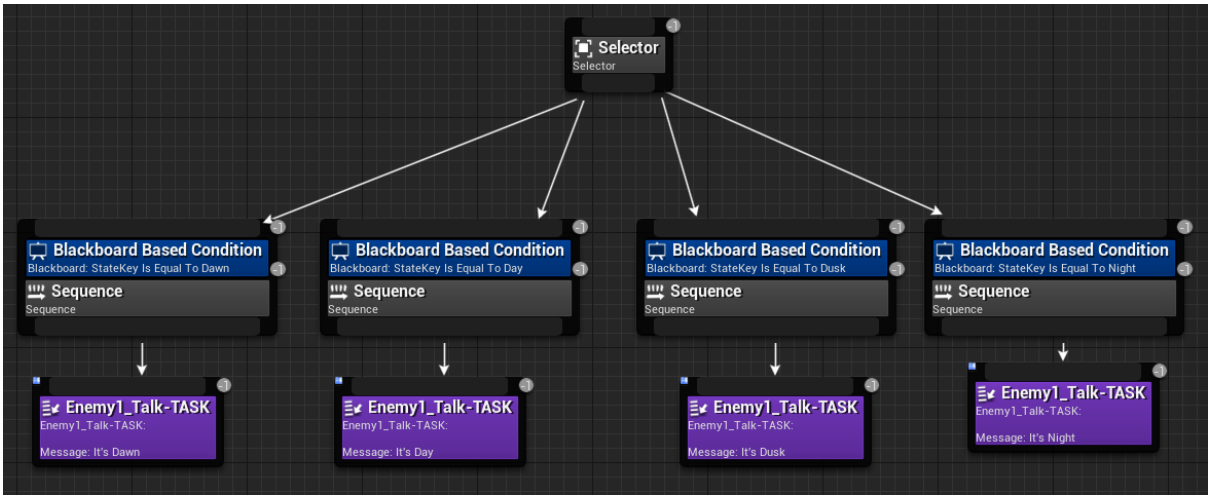


Fig 30. Showing how a behaviour tree works with a selector choosing a sequence and then a sequence performing tasks.

To best explain how the behaviour tree works, it can be thought of as a series of gates. For example, the tree attempts to run tasks from left to right meaning if the time was currently dusk then the first two nodes of the tree would instantly fail to run. However, once the tree lands on the third node, which has the same value as the current time of day, it enters this tree and again attempts to run from left to right. If a task on the left fails then it goes back up to the top of this current branch and tests the next branch (usually the right of this same branch).

Behaviour Tree Compared to Design Process

As the behaviour tree inside Unreal “flows” through in a gate-like format ,similar to a flow chart, it would be beneficial to compare the flow chart of state changes (section 6.1) to the behaviour tree with the processes depicted in a tree format.

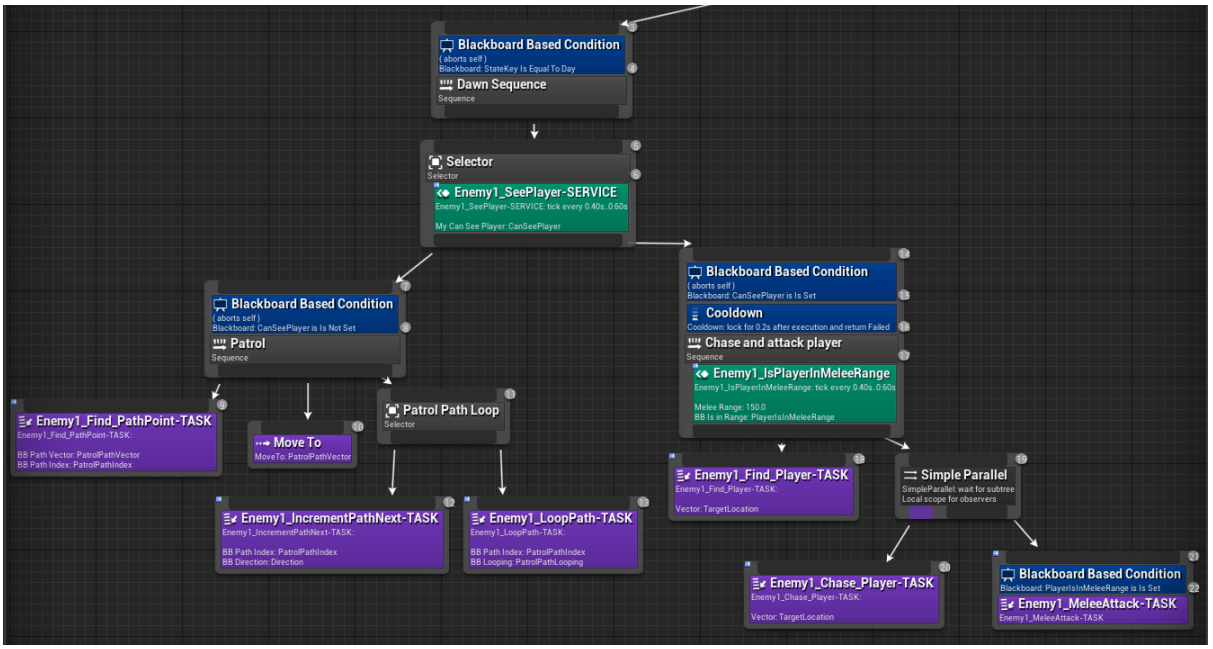


Fig 31. Showing how the enemy decides between patrolling or chasing the player based on conditions, services, and range.

The dawn and dusk trees are perhaps the most useful to observe. This is because they hold the most condition based tasks. Before explaining how the tree functions it is best to first identify the tree's different features, this is best described in Table 5. seen below.

Table 5.

Operator	Ability	Ability in Fig 31.	Colour
Selector	Decides which branch to go down.	Choose the left branch if the enemy is not seen, or the right if they are.	Grey
Sequence	Performs a series of tasks directly below when run.	The patrol sequence on the left runs the find_pathpoint_task and the patrol path loop selector. The chase and attack player sequence on the right runs the find player, chase player, and potentially attack player tasks.	Grey
Task	A task is any given set of instructions.	One example of a task in this tree is the chase player task, this task finds the player's location and repeatedly sets the enemy to go to this location.	Purple
Service	A method checked repeatedly.	The two services running in this branch are checking if the player is seen and if the player is within melee range.	Green

Simple Parallel	This runs the task on its direct bottom left with high priority, while also running the task to its bottom right simultaneously.	The only simple parallel used here is run when the player is within range of the enemy. When they are both the chase and melee attack tasks are run in parallel.	Grey and purple
Decorator	This is used to check if a certain condition is true before running. There can be many conditions required for each branch to be executed.	The top sequence uses a decorator to check if the time is in fact dawn. The other decorations are used to check if the player is seen (taken from the see player service), and if the player is in melee range (taken from the player in melee range service). There is also a cooldown decorator that is used to delay branch operations for a few frames so as to not overload the program with requests.	Blue

Day

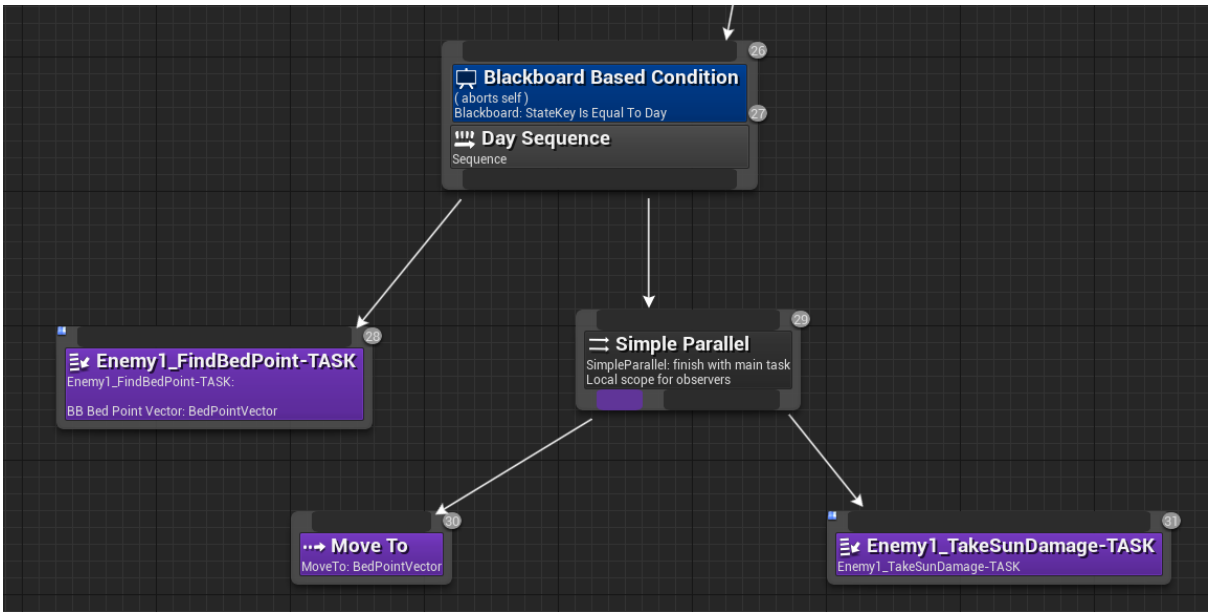


Fig 32. Showing how during the day the enemy finds their bed and then attempts to move to their bed while taking sun damage if they are not inside.

The day behaviour, like the flow chart from the design phase, is the simplest behaviour tree. It features only 3 tasks, 1 parallel, 1 sequence, and 1 decorator. The Sun damage task applies a damage total of 5 for every second that the enemy is not overlapping with the radius of their designated bed.

Night

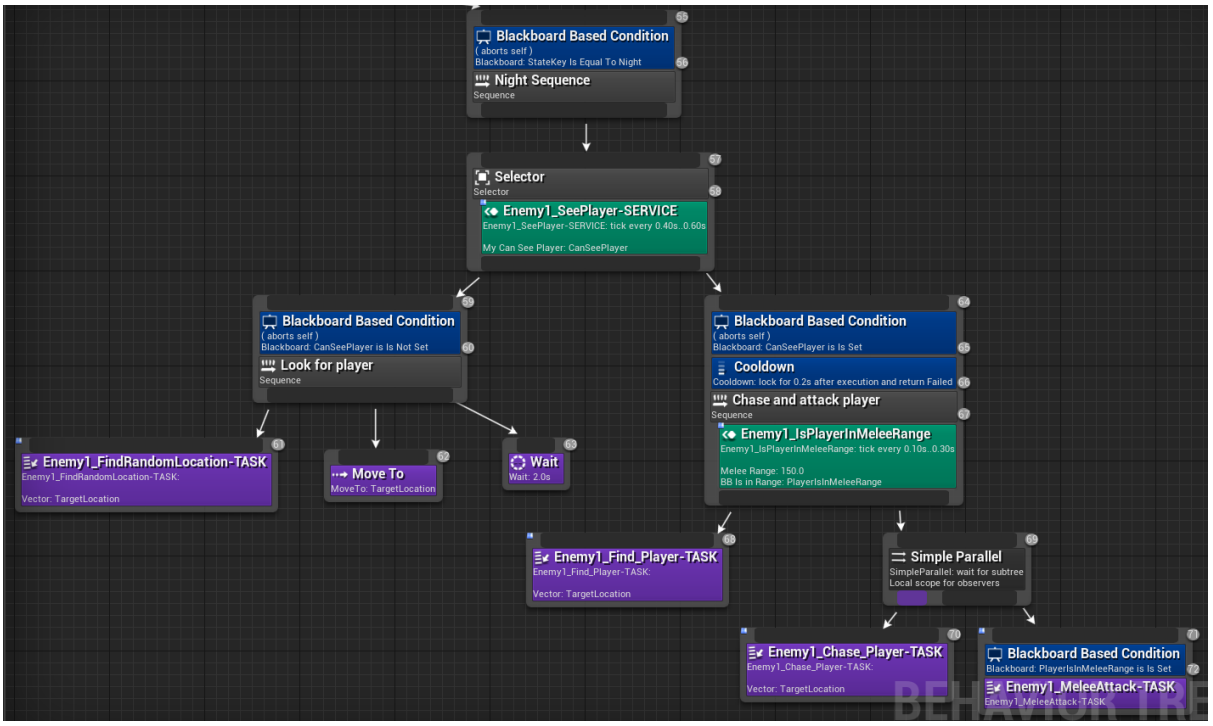


Fig 33. Showing the night behaviour tree

The night behaviour tree is a more aggressive version of the dawn and dusk. In this section the enemy actively seeks out the player instead of simply patrolling their designated patrol path. This is done by finding a random point in their navigable radius and moving to this point, this process occurs every 2 seconds (as seen in Fig 33. above). Once the enemy finds the player, it chases and attacks them in the same way as the dawn and dusk times.

7.3 Game

7.3.1 Map

The map in *Half-Blood* was implemented by referring to the design documentation in section 6.3.4, as well as following the specifications laid out in the game design document in Appendix B. The environment itself was created using Unreal Engine's scene editor, more specifically the landscaping tool, foliage placer, and geometry assets.

Firstly, the map area was built up using the landscaping tool which allows for manipulation of the ground to create peaks and troughs. This tool was used primarily to give the game a textured look as opposed to a flat land with no depth.

Secondly, the foliage tool of Unreal allows for a random disparity in assets placed within the world. For example, the placement of rocks, grass, and shrubbery can be set to different ratios to produce a world that feels truly random while still generating all the assets needed by the creator. This was done by setting the amount of big rocks, used for cover, to low while keeping the amount of short grass high. The outcome was a world with enough cover places for the player to take advantage of and enough grass to suspend disbelief for the player.

Finally, the basic 3D geometry assets that Unreal provides were used to create the houses and starting area. The ability to set geometry to either “additive” or “subtractive” allows for houses to have doors by using a rectangular cuboid in a door shape and setting the mode to “subtractive”. This box, although invisible due to being subtractive, can be used as an artificial invisible wall to block the player entering during the day time. This works by a simple collision system where the player’s model cannot pass through this, but any other game actor can, including enemies.

7.3.2 Player and Enemies

The models and character design of both the player and the enemy have followed the specifications laid out in section 6.3.3. The player character in *Half-Blood* was developed in the model of the default third person character provided by Unreal in the example map. The model was slightly adjusted for gameplay reasons. Firstly, the model was changed from a masculine figure into a smaller feminine character in order to create a disparity between the enemy and player builds. There are many components that both the player and enemy possess, there are also specific features that help define the differences between both. Below are listed some of the key characteristics that make up the player and enemy characters.

Shared Components

Both the player and the enemies share the following:

- A weapon
    - This is seen as a stake in the players hand
    - If this stake make contact with the enemy then the enemy takes damage
        - This also applies to the enemy attacking the player
- Capsule component (used to see if the enemy can see the player seen around the player in Fig 34. and 35. below)
    - This shrinks during crouching to allow the player to cover behind rocks
- Health
    - Set to 100 initially (In Fig 34. it is seen that this variable is public by the open eye next to it, this means other functions can freely change the value of this variable)
    - Taken away based on damage type given to player or enemy
        - The enemy specifically takes damage to the Sun whereas the player does not
- Attack animations
    - This includes a shared heavy attack
    - The player can use a quick attack which is a speed manipulated heavy attack that the enemy does not have access to

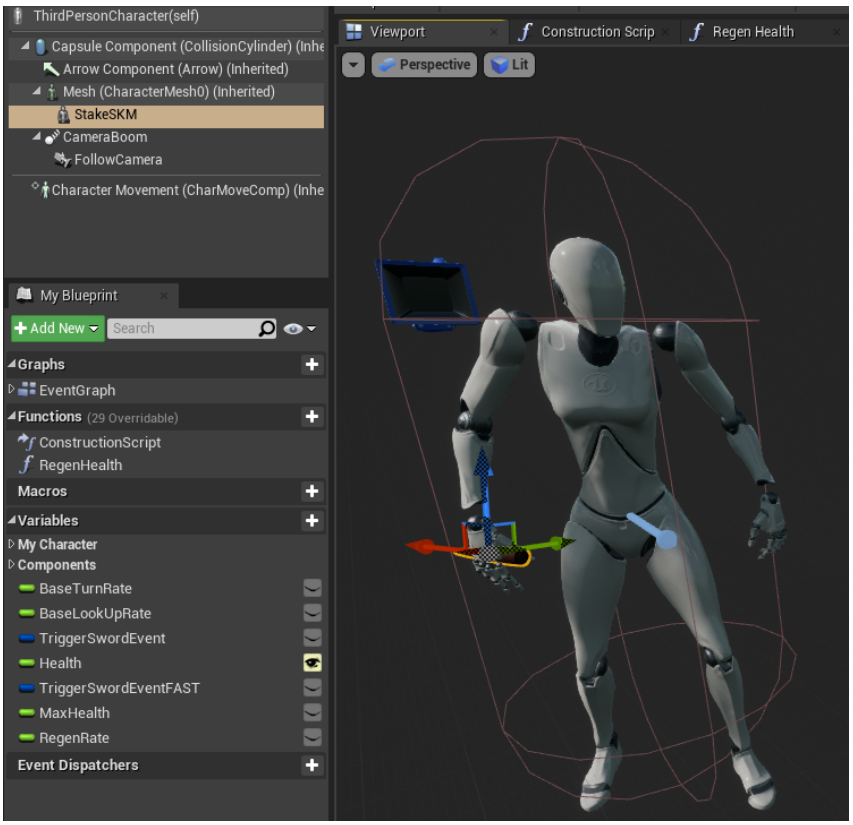


Fig 34. Showing the components, functions, variables, character model, and model additions of the player.

Player Specific Components

When looking at the player and the components specific to their character, the largest differences to the enemy character is the regen health and crouching functions. The regen health function allows the player to slowly regenerate health that is lost during battles, the reason for this function is so the player is always at a slight advantage to the enemies. Another key reason for allowing the player to regenerate over time is due to the gap in action during the day time, this would be a perfect point to allow the player to rebuild and regroup for the next nighttime attack.

The crouching function is also integral to creating a slight, but not overwhelming, imbalance in player and enemy abilities. It allows the player to use the in-game environment to perform tactical manoeuvres on the fly. The method allowing for stealth comes from the player's capsule component (as seen in Fig 34. and 35.) that directly communicates with the enemy's cone of vision. As seen below in Fig 35., the capsule shrinks even smaller than the player, so the benefit of the doubt is always given to the player so they are always sure that a rock will be protecting them from enemy vision.

Fig 35. Showing the difference between the capsule component while standing and while crouching

Enemy Specific Components

The enemy and all its features can be seen below in Fig 36. The most prominent features of the enemy are within its variables, specifically the patrol path, house bed, and health widget reference.

The patrol path is a reference to a given patrol path within the game world, an enemy can be given any patrol path in the game world during initialisation. The enemy can also be given no patrol path and so has no set movement path during times when the patrol path is called. More interestingly, the enemy can also be given the same patrol path as another enemy and they will each follow the path depending on their current position.

The house bed is also a reference to a given bed that is placed inside a house. This bed is just like the house asset as it is made with geometry assets, however it is a blueprint so it holds extra data such as its vector location. The bed is primarily used to test whether the enemy is placed within its house. If the enemy's body is not overlapping with the bed's radius then the enemy will proceed to take damage from the Sun when it is day time. This is seen best in section 7.2.3 for how the AI reacts to state changes.

Finally, the enemy has a health widget reference which corresponds to a widget placed above their head. This widget is only visible when the player walks within a certain distance of the enemy. This health is a percentage from 0-100, when the enemy takes damage it is directly seen in this widget. The widget is always facing the player so a sneak attack from behind will allow for the player to see the health reduce.

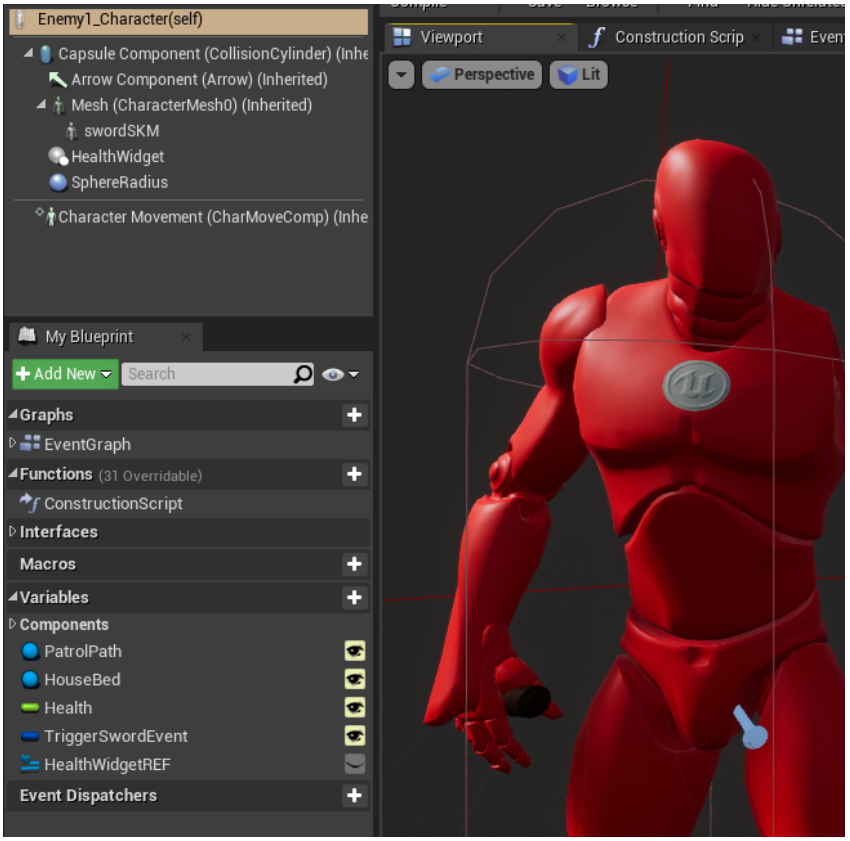


Fig 36. Showing the components, functions, variables, character model, and model additions of the enemy character.

7.3.3 GUI

The elements implemented for the GUI were directly influenced by the designs in section 6.3.2. Below are the menus and HUD with a brief description on the functionality of each element, and also how each system has its own individual purpose.

Main Menu

Fig 37. Main Menu

The main menu (see Fig 37.) is the opening screen on start-up. Its purpose is to act as a stop-gap between starting the game and entering gameplay. The functionality of this screen is to let users read through the control scheme for either of the supported controllers. The screen also features a start and quit button that either enter the game world or quit to the desktop.

The choice to have the controls be a part of this screen instead of their own designated screen was simply to save on processing power for the user. It is understood that due to being made in Unreal, an engine that gives the user many features to work with, the game may not perform as well on lower-end systems. Therefore, it is crucial to make sure these systems do not run unneeded assets that will cause slowdown.

Game Over

Fig 38. Game Over Screen

The game over screen (as seen in Fig 38.) is used to bridge the gap between death and exiting the game. This screen only appears under the condition that the player's health reaches 0. Once their death animation plays this screen pops up and the player only has the option to quit. The purpose of quitting the game instead of a potential restart is because all the assets and functions work through an event called "event begin play" which is only triggered on the game starting anew. The need for a full game restart is also not too demanding as the player's machine will face no impact from restarting as the application will be fully packaged and ready to play again with no major hitches.

HUD

Fig 39. HUD at start of game

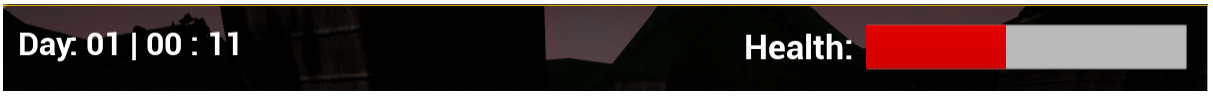


Fig 40. HUD during gameplay

The HUD, as seen in 2 different states above in Figs 39. and 40., has been laid out in the abstract way detailed in design section 6.3.2. The day is in an easy to read format with clear text that is always visible in either day or night. The health is also instantly recognisable as a percent bar that reduces on damage from enemies.

7.3.4 Win/Loss Conditions

The win/loss conditions in the current iteration of the game demo are extremely limited.
There is no win condition because the aim of the game for the player during testing is to simply play through the cycles as an experience. It is not an expectation of the player to

want to win in this proof-of-concept, but instead to want to witness a change in gaming behaviour through the advancement in time.
The loss condition is evident in this demo phase however. Although the player is not expect to kill all the enemies,

7.3.5 Finishing Touches

The finishing touches of the game consisted of ironing out any remaining bugs and ensuring all usability requirements were met. The slight squeaking of variables made sure that all functions ran as intended with numbers, such as the distance required to see the player, being slightly increased before deployment. The use of music and sound effects could have been established here, but the copyright issues and lack of sound design knowledge prevented this from coming to fruition.

7.4 Summary of Implementation

To summarise, this chapter has concluded with the implementation of the desired system in accordance with the requirements and design laid out. The following chapter will look to test this system with more routine and critical software testing, as well as a set of user testing. The results from these tests will measure the success of the implementation phase and the data will be analysed in a later chapter.

Chapter 8 - Testing

In this chapter the product will be put under scrutiny by both software testing and user testing. Throughout implementation testing has been performed on an extremely low-level, on-the-fly basis. However, dedicating a chapter of this paper to testing in excruciating detail in terms of functionality, performance, and usability will allow for an evaluation of the product as a whole.

8.1 Software Testing

8.1.1 Plan

To properly test the software, a clear plan should be in place in order for the testing to be effective. Firstly, the testing has been divided into functionality testing and performance testing. This division will help keep results separate and ensure the quality of testing in each section is completed to the highest standard. Each set of data that has been tested is tabled and fully accessible in the appendices. The testing equipment used is the industry standard and testing has been thorough across all sections.

8.1.2 Functionality Testing

Functionality Testing Results can be seen in Appendix D

To test the software's functionality, both white-box and black-box testing methods were used frequently. White-box testing was specifically used when testing the AI, as the game itself would not always give a perfect indication of how the AI was reacting to its current situation. Therefore, running the game and taking a look at how the behaviour trees were executing worked as an excellent indicator to their current state of operation.

Furthermore, the functionality testing was broken into: basic gameplay mechanics, combat and stealth, enemy AI, and menu systems. A minimum of four test cases for each section were selected based on the importance to the rest of the system. Prerequisites and player actions were noted in order for the test case to work. Further, an expected result matched with the actual outcome were displayed to monitor if functionality was working as intended.

The analysis of this testing shows that every function tested works as intended. This shows that the requirement documentation created has been followed and completed to a high standard. However, this can also conclude that the requirement specifications were insufficient as every task was easily accomplished; perhaps more complex requirements were needed in order to test higher level programming skills.

8.1.3 Performance Testing

Performance Testing Results can be seen in Appendix E

Performance testing was completed as a means to see the efficiency of the code when attempting to scale the game to much larger magnitudes than originally planned for. The reason for this is to judge the maintainability of the code and see how the performance is impacted based on its scalability. When creating this game demo it was not imagined that

over 100 enemies would all be active at once, however if the code is highly efficient then the performance should not suffer.

The method for testing involved the use of a frames per second (FPS) monitor, *Fraps* (Beepa Pty Ltd, 2013). *Fraps* allows for segments of games to be recorded for a set time, returning the minimum, maximum, and average FPS for this time period. A short time of 10 seconds was chosen to monitor the impact each change had on the performance. The minimum playable value was deemed to be 60 FPS, as any lower and the game appear "choppy". Six test cases were chosen off increasing value and the results were tabled.

The analysis of the resulting data shows that the efficiency of the cycle algorithm results in no hindrance upon increasing the speed. For example, for the final test case the speed is set to 10,000 which means the day length is an extremely quick 0.075 seconds, however the program still performs amicably.

On the other hand, the performance testing of increasing the number of enemies resulted in a much more expected outcome. Increasing the enemies from 1 to 10 resulted in a stable 90 FPS average across the first three test cases. However, upon increasing to 20, the program began to falter. 50 enemies took the program below 60 FPS for the first time, with a recorded minimum of 58 FPS. Finally, 100 enemies was the limit for playability as the program recorded a low of 27 and a high of 50 FPS. Therefore, it can be concluded that the enemy AI is too demanding as several "brains" attempt to perform tasks simultaneously. The attempted solution for this could be to assess each AI task and look to reduce casting, alternatively a spawning system with a limit on enemies alive at the same time could also be looked into.

8.2 User Testing

User Testing Plans and Results can be seen in Appendix F

8.2.1 Plan

Before displaying the user testing plan it is crucial to first explain why this product was tested with users. The reason for user testing lies within the literature review where it was theorised that human attention is key when looking to make an engaging product. Therefore, varying lengths of day/night cycles were created for each user to test with the resulting data from observations and surveys being analysed. The other reason that user testing is necessary is due to the list of usability requirements created, as these need to be tested by a third party to eliminate bias.

Before each user test, the tester was informed of the nature of the product and background lore, such as not being able to enter the vampire houses as the player is also part vampire.

8.2.2 Observational Results Summarised by Cycle

During testing, the observations made during the think aloud study were paired with an interview to generate qualitative data. This qualitative data has since been compiled and analysed. The most prominent and most interesting points have been summarised below, categorised by the cycle length talked about.

Cycle 1 - 7 Minutes

The control cycle was the player's first experience with the game and so the observational and think aloud data were much different than the two cycles that came after. Firstly, the players all performed the same first action of running around the environment and using cover to test the crouching functionality. As the night fell they all started to vocalise how the game's darkness was creating a rise in tension. However, once the enemies started to exit their houses the players all took very different approaches. Some players attempted to attack as soon as the enemies exited, some stayed hidden and observed the enemies patrolling, and some attempted to use stealth to pick off one at a time. Once the first 24 hours were over the players were told that they could quit which most did, however some insisted that they wished to complete the game.

Cycle 2 - 3.5 minutes

The next cycle was the quicker cycle, note that no players were told the exact lengths of each cycle, which instilled the most engagement. They all instantly noticed the increase in speed and felt overwhelmed as the enemies rapidly started to engage them. Compared to the first cycle, many found it infuriating that they could not kill as many enemies due to the shorter space of time.

Due to the speed increase, most players opted to stay in the game when being told that they could quit. This showed a clear engagement as most participants stopped communicating once their game hit the peak night hours. Although this did not provide any vocal data, it did allow an inference that their immersion had peaked so much that they were no longer focused on the study.

Cycle 3 - 14 minutes

Finally, the lengthiest cycle was tested. Instantly, players noticed the change in speeds from the quick cycle. Unlike previous cycles this cycle was the only one where players died from the attacks during the night hours. This occurred when players tried to use the same tactics as the quick cycle, however this time the enemies did not return to their house as quickly. Once other players noticed how it was easier to be killed in this cycle, the increase in crouching was exponential.

When asked how long each participant felt the cycles lasted, the answers were all extremely underestimated. For instance, one participant felt that the 14 minute cycle was only 6 minutes. Similarly, the answers for the control cycle resulted in an average of 5 minutes. Whereas the answers regarding the quick cycle surprisingly resulted in accurate answers around the 3 minute range.

This cycle saw a 100% quit rate when told that the game had reached 24 hours. It was clear that the engagement of this cycle was lacking compared to the others. The questionnaire data below was used to support the claims made here in a more quantitative manner.

8.2.3 Questionnaire Results General Summary

Once the hands-on section of testing was completed, the tester was subjected to an online questionnaire. This questionnaire was used to also generate some quantitative data that

could be easily charted and further scrutinised. The questions asked were purely numerical as the observation and interview had already allowed for qualitative data.

Firstly, the breakdown of the 8 participants' ages shows a range of 18 to 52 with a mean age of 27.4.

The average gaming skill rating by each participant is 2.4, which shows how this study accomplished the task of choosing people with some understanding of video games. The results of this question could come down to the participant's humbleness to not choose answers at the top end of the spectrum.

The response to a clear difference in cycles lengths had a 100% approval. The participants clearly noted a change while playing and this answer followed through into the questionnaire results.

The response to whether the instructions given were clear also brought a 100% approval. This not only proves that the participants fully understood the game they were playing, but also that the usability of the system was accomplished.

Finally, to summarise the non-cycle related questions, 75% of participants found the damage totals fair from both the player and the enemies. Of the 25% that found the totals unfair, it was noted that there was a call for a percentage number to complement the percentage bar. Further, a call to increase enemy damage and decrease health regeneration for fairer gameplay was another criticism. Therefore, further tests in future will look to find the perfect regen rate and enemy damage calculations.

8.2.4 Questionnaire Results Summarised by Cycle

Section 8.2.3 evaluated the questions focused on the usability and functionality of the game. However, the purpose of this study was to understand what made a video game's day/night cycle effective. Therefore, the data from the questions relating to the individual cycles was further scrutinised in more detail.

Cycle 1 - 7 Minutes

The results regarding the control cycle see it as the most fair and most average. The data to support this is that it:

- Ranked second in terms of enjoyment with 37.5% of the vote
- Ranked first in terms of fairness with 50% of the vote
- Scored 6.9/10 for engagement

Cycle 2 - 3.5 minutes

The quickest cycle brought in data suggesting that it is the most enjoyable and engaging. However, participants note that it is not the most fair cycle suggesting that it is potentially overwhelming. The data to support this is as follows:

- Ranked first in terms of enjoyment with 62.5% of the vote
- Ranked second in terms of fairness with 37.5% of the vote
- Scored 8.9/10 for engagement

Cycle 3 - 14 minutes

Finally, the longest cycle scored the lowest results in every cycle based question. The data suggests that this cycle was too long for the size of play area allowed. The majority of video games reviewed in chapter 3 boast longer cycle lengths than this, however the play area of these games are either much larger or entirely procedurally generated. The results from this survey regarding cycle 3 are as follows:

- Ranked last in terms of enjoyment with 0% of the vote
- Ranked last in terms of fairness with 12.5% of the vote
- Scored 5.3/10 for engagement

8.2.5 Further Comments

The end of the questionnaire also featured an extra comments box. This box was used to find any more nuances that the tester wished to mention that may have been forgotten from the interview stage. The outcome from this section mainly resulted in testers voicing their opinion on how *Half-Blood* could be improved. This mainly focused in the combat area and perhaps the need for more tactical choice. The most commonly repeated points were:

1. More accurate combat system
2. More hiding spaces or a more diverse array of hiding spaces
3. More interesting gameplay during the non-active hours

Section 3 - Evaluation and Conclusions

Chapter 9 - Evaluation of the Product

The product created was a game called *Half-Blood*. This game was implemented using the requirements created in both the Terms of Reference (Appendix A), and the documentation from Chapter 4. The product is evaluated based on the criteria laid out in previous planning and designing phases.

9.1 Fitness for Purpose

9.1.1 Meeting of Requirements

The functional requirements of the product were ordered by the priority they should be given. This prioritisation allowed for the more crucial aspects of the game to be completed which would then lay the foundations for the less useful aspects to be developed. For example, the game must have a day/night cycle and intuitive AI before the architecture and character design could be considered useful additions.

Functional Requirements

Must Requirements
1. Day/night cycle
2. Multiple lengths of day night cycles
3. Intuitive AI

Every requirement (1-3) labelled as a “must” was fully implemented.

Should Requirements
4. Theme appropriate for gameplay

Requirement 4 was labelled as a “should” and was therefore implemented soon after the highest priority items were in place.

Could Requirements
5. Detailed enemies
6. Aesthetic architecture
7. Menu screens

Requirements 5 through 7 “could” be implemented according to the documentation. Therefore, they were performed during the end of development with less an emphasis more on the core functionality than any minor detailing.

Won’t Requirements
8. Multiple levels and/or multiple maps

Finally, requirement 8 was rejected due to the scope of the project. It was decided that even after all other requirements were implemented, this requirement should be omitted and that testing should take precedence.

Summary

Due to the completion of all approved criteria, the product created could be classified as a playable demo. This means that the product has gone beyond a proof-of-concept piece of software used to evaluate player's psychology, but can now be played as a source of entertainment.

Functionality Testing results can be seen in Appendix D

Usability Requirements

All the usability requirements established were met by this product. All the controls and UI work perfectly because they were tailored directly to player responsiveness. The player can interact with the game as they like and there have been no reports of crashes from both individual testing or playtesting.

9.1.2 Quality and Future-Proofing of Implemented Functionality

The quality of the implemented functionality cannot be denied due to the above evaluation. However, the future-proofing and sustainability of the functions should be criticised. In an honest evaluation, the lack of dynamic variables used in many of the game's algorithms creates poor coupling and cohesion. On the other hand, it must be noted that the use of blueprints (classes that allow for polymorphism) has reduced the replication of code and decreased coupling.

To measure the sustainability of the code, a scenario in which a programmer takes over this established codebase was imagined. In this scenario it is believed that the programmer would find the code, in its current state, to be easily changeable to suit their needs. For example, if the programmer wanted to move the game into a different map then the functional blueprints could easily be transported across with no issue. Therefore, the combined quality and resilience of the current codebase should be praised when evaluating the product created.

9.2 Build quality

9.2.1 Terms of Reference and Literature Review Specification

The quality of the final build has been affected from the opening stages of this project. Firstly, the terms of reference, which influenced the literature review of attention spans and domain media, was the primary source of requirement specification. However, once the requirement specification began it became clear that more gameplay features would need to be added to compliment the day/night cycle algorithm. For example, the addition of a crouching ability was only realised during the designing phase when thinking about the playing environment. The addition of a feature like this, that came about after the initial ideas formed from domain media, is not a negative as it only increases the playability of *Half-Blood*.

9.2.2 Requirements and Design Specification

Although the additions listed above were not negative, it was still unfortunate that the requirements specification had to change due to an oversight. The same cannot be said for the designing phase of development. This phase saw no changes in functionality and usability requisites and the final build quality was positively impacted by this decision. The UML-standard class diagrams and flowcharts were the primary design tools used and they were followed expertly to create a product that fully matched the documentation. The pseudocode also acted as a solid foundation to the predominant day/night cycle algorithm, this was then incorporated using the visual scripting techniques in the Unreal Engine. To summarise, the distinctly outlined design process led to a successful end product.

9.2.3 Overall Programming Quality with Visual Scripting

Visual scripting ensured that *Half-Blood* remained fully modular. The tools given by Unreal allow for developers to make use of polymorphism and de-coupled coding concepts that, in this context, created a cohesive codebase. The component based architecture of the engine also allowed for character models to possess different assets such as patrol paths and health totals. This specific model of architecture was one of the reasons given during tool selection when compared to the object-oriented approach that Unity's C# language utilises.

When programming with this well designed engine, official documentation and reputable tutorials were followed to maintain clean and consistent code. Coding conventions from text based languages have been used inside the visual paradigm to replicate the maintainability seen in past programming endeavours. The use of comments to explain functions and their complex sub-components was used to allow for the system to be understood by anyone from a non-expertise background.

9.2.4 Test Plans and Results

The testing phase allowed for the build quality to be put under extreme scrutiny. The software testing particularly analysed the comprehensiveness of the code, while the user testing helped to identify any other minor issues that went unnoticed.

The performance testing was completed with the focus of identifying how the game ran during differing intensities. The outcome of the performance testing was an indication that, on a gaming PC, the game's build was easily runnable on high frame rates even with the addition of multiple enemies. Therefore, this shows the scalability of the product and the high standard of build quality.

However, the results from user testing demonstrated that the combat functions featured some bugs that were not noticed during functionality testing. The evaluation of this sector has since led to more testing with different variables and altered hitboxes to ensure that if this product was to be iterated again, it would be of a much higher standard in terms of the code effectiveness.

Chapter 10 - Evaluation of the Process

The project process in its entirety was a large scale task that involved the reviewing of past domain literature and then the implementation based on said literature. The research and planning phase formed the core of the ideas that would be later built on in the design phase. These designs would then be implemented and tested until a clear analysis of the success of the project could be undertaken. Below is an evaluation of how different factors have affected the success of the process as a whole and how changes could have been made to achieve a higher standard of work.

10.1 Measures of Success

10.1.1 Learning Process

Regardless of the project's success, the amount of knowledge gained from this process is undoubted. To begin, the literature review allowed unbridled insight into player psychology and how the mind is influenced by its surroundings. Paired with this, the review of games in the domain of changing cycles opened an entirely different angle into game design and how developers purposely create environments that lock the player into an immersed state. The skills required for game design have also massively improved, specifically using UML to plan and then implementing this within a game development tool such as Unreal Engine. Therefore, the learning process that this project has enabled has developed useful skills and opened the door to exciting media.

10.1.2 Meeting Objectives

In the earliest stages of the process a plan was enacted in bullet point form to create a high-view of criteria that must be met over the course of the project. This plan was created in the earliest documentation for this process, the Terms of Reference (see Appendix A). By evaluating the completeness of each objective, it can be inferred how successful the process was in a generalised sense.

Table 6.

No.	Objective	Proof of Completeness
1	Literature review on attention spans and cycles of activity	Chapter 2
2	Review of day/night cycles of different games	Chapter 3
3	Establishing requirements	Chapter 4
4	Creating design documentation for the game	Chapter 5 & 6 and Appendix B
5	Creating the game	Chapter 7 and Appendix C

6	Testing the game (software testing + play testing)	Chapter 8 and Appendix D, E & F
7	Evaluation of the product	Chapter 9
8	Evaluation of the project as a whole	Chapter 10
9	Personal reflection	Chapter 11

10.1.3 Analysis of Time Management

When taking on a project of this magnitude, that has several interlinking phases, it is crucial to manage time effectively. This is the primary reason that a GANTT chart was produced when establishing the terms of reference. This GANTT chart was designed to be followed to a tee, not just so all tasks were completed but also so that each task was assigned the correct amount of time. In an honest reflection, this GANTT chart was adhered to up until the implementation phase where the software development tool's complexities negatively affected the speed of the project. The challenges of using an engine that was not mastered was mentioned in chapter 5 when choosing the development tool. Therefore, the schedule's dissonance was unfortunate even if the potential holdup was identified early in the project. To improve the effectiveness of later projects, a more generous time should be allowed when using non-mastered software, or alternatively a more knowledgeable tool should be used.

10.2 Potential Improvements

This project process can be taken and the negatives can be evaluated in order to improve later projects. Therefore, the potential improvements that this project could have featured are evaluated and the positive impact they could have had will be discussed.

10.2.1 Poor Testing Apparatus

The most obvious improvement, that would have given this project a much more solid conclusion, is by using better testing apparatuses. Better testing equipment would present a more defined conclusion due to the increased accuracy and greater test data generated. From the literature review it became clear that the testing performed in the examined papers produced astonishing data. The data from these expertly crafted tests led to detailed conclusions that were backed up by the scientific evidence. On the other hand, this project did not have access to the specialist equipment that was used in these excellent reports. If a project in attention observation were to be undertaken at a later date then professional apparatus, such as facial EMG, would be booked in advance.
To also note, the prevention of face-to-face contact during the duration of this project potentially disengaged testers and the observation data was undoubtedly negatively affected.

10.2.2 Lack of a Development Methodology

Before development began, the research into a clearly defined development methodology was unfortunately excluded. In hindsight, this exclusion was an extremely poor decision due to the chaotic development cycle that ensued. With a clearly laid out plan of iterations this

project had the potential to have a clean implement and test rotation. Instead the programming and analysis felt disjointed from each other which led to minor bugs making their way into the released product. For the future an array of methodologies will be assessed, depending on the project and team size, to ensure a coherent lifespan of the project.

10.3 Legal, Ethical, Professional, and Social Issues Discussed

10.3.1 Ethical Issues

This project was deemed a medium risk when consulting with the ethics department during the proposal stage. The university protocol insisted that the nature of this project would feature no risks other than the use of human participants. In order to avoid a high risk project, the choice to test remotely was taken as a precaution. The ethical issues involving the data of these human participants was negated due to the lack of information taken. Each participant was paired with a number and their age, this created a barrier between the information given and any traceable details. The participants were also given a participant information sheet and a consent form that informed them of all the risks associated with this project and how they could withdraw at any time. Therefore, any potential issues that could arise from an ethical standpoint have been addressed well in this project.

10.3.2 Social Issues

As well as potential ethical issues, this project could have also faced a social issue. This was entirely down to the context of the project background. Due to the project's aim involving attention spans, it was important to exclude any participants that may skew data due to underlying conditions. During the literature review in section 2.3.4, it was found that many conditions can negatively affect the brain in relation to attention spans. To prevent discrimination upon testing, the advertisements for this project testing phase clearly identified the list of exclusion criteria and why it was necessary for this list. To further prevent prejudice and maintain congruent data, the exclusion criteria was again repeated in the participant information sheet.

Chapter 11 - Personal Reflection

11.1 Lessons Learned

Personal reflection is an important part during the closure of a long-term project as it allows for a more intimate critique of yourself. From the origin of this project to its conclusion, many lessons have been learned and the key elements that make a successful report have been identified through trial and error. In my opinion, the mistakes made during this project have only opened the doors to more successful projects in the future as the lessons learned have been invaluable. This paired with the new found knowledge in such intriguing domains have made for a wonderfully eye-opening experience.

11.1.1 Subject Matter

Firstly, the subject matter of this report took me into a world of intense data-driven analysis on the human mind. The papers read during the preliminary stages of this report have changed the way I think about human attention and what can directly affect it. The continuation of reviewing games featuring day/night cycles only made me more intrigued into how game developer's harness the data taken from these studies, and then use it to create engaging worlds for the player to explore. There is much to be said when analysing game psychology that this report could not do justice with the scope it is set in. That being said, I will certainly be researching this domain further and it would be highly encouraged that the reader does too.

11.1.2 Project Process

I learned a lot about the process a project goes through and how to ensure success at every stage. Starting with a project's inception, I would encourage the use of a clear structure as this helped me keep on track with the objectives and deliverables expected of me. This plan ensured that I kept an equilibrium between all tasks so each section of development was treated fairly. If this plan was not followed, or worse not established, then there would be a disproportionate amount of effort administered to each task resulting in an unbalanced final product.

Chapter 12 - Conclusions and Recommendations

12.1 Conclusion

12.1.1 Aims and Success

In conclusion, this project's success can be determined entirely by the fulfilment of the initial aims set out. These aims were conceptualised in the Terms of Reference (see Appendix A) and are as follows: "investigate potential optimum day/night cycle for games" and "create a game demo product as proof of concept". The first aim, to investigate potential optimum day/night cycles, is believed to have succeeded due to the breadth of research performed in both chapter 2 and 3. The only negative from this objective's success is how this mass amount of research did not lead to a definitive answer. Perhaps a different aim should have been created that focused more on whether an optimum cycle can even exist. It is believed that many factors, such as the map size and the availability of tasks open to the player, can dictate how long a cycle should be in any given game. However, the second aim tackled in this project has been accomplished to its full extent. The proof-of-concept game demo is a fully functioning piece of media that has been used to test against the former aim. The results formed from this demo shows how this aim has been fulfilled and confirms the overall success of the project as a whole.

As noted earlier, this project unfortunately failed to find an optimised length for a day/night cycle. However, this is still more of a success than the original aim and title of this project. The original title was as follows: "Creating the perfect day/night cycle for a game by researching the contributing psychological factors ". This choice to change this report's aim of creating a "perfect" day/night cycle into simply investigating the potential for an optimum cycle is a choice that positively affected the outcome of this paper. If this paper was to try to create a perfect cycle then the project would have been a failure as it has been proven that there is no perfect length throughout this report. Therefore, the inaugural stages of a project of this scale are integral to the success, as a poor research aim can destroy the expected result.

12.1.2 High View of Chapters

When evaluating the report in its entirety, each chapter's purpose has been assessed to determine how the project progressed at each stage. Initially, the literature review gained an understanding into attention spans, how they could be affected, and most importantly how they can be measured. The further analysis of games in the domain of day/night cycles proved to be an excellent pairing with the literature review. The combined findings from these chapters would help to create the requirement framework that *Half-Blood* would be built upon.

The requirement outline and design phase were excellent uses of planning that allowed the implementation phase to flow smoothly. Had it not been for the UML and game design documentation, the implementation could have not only staggered the entire report, but led to failure in meeting the primary project aims. Nevertheless, the design section instead accelerated the remaining report due to its conciseness which caused the implementation to be finished in good time and to a high standard. The use of functional and usable

requirements was integral to the success as it created achievable and measurable targets that were aspired to during development. During the evaluation it was clear how the structure of this report created a chain-like framework in which the success of one section directly impacted the success of the one that followed. This could be seen as a negative because if one section is a failure then the rest of the project collapses. However, from the evidence shown, this project veritably prospered from this structure due to the level of effort spent in each chapter.

The implementation stage focused on the creation of the game demo, *Half-Blood*. This was the proof-of-concept shell that housed the mechanics researched in the past chapters. *Half-Blood* showcases the experimentation of three different cycle times. Although a product made to be purely experimental, the effort of using Unreal Engine to create an interesting video game was still a valid practice to encourage player engagement. The quality of the implementation has been shown by the accomplishment of both the usability requirements, and the prioritised functional requirements. The negatives from this stage lie primarily with the features not identified in the requirement and design stage. The largest criticism from players was the lack of a complex combat system with tactical choice. The reason this was not planned in the design phase was due to the already complex algorithms that made up the cycle itself and the AI affected by the cycle time. *Half-Blood* does allow the player to switch between light and heavy attacks, it also allows the player to hide behind large objects so this area of gameplay was not a complete oversight. Although a negative, it is clear that the point made previously about a project decline knock-on effect has been proven by this blunder.

Once development was completed, a detail-oriented testing phase commenced. Chapter 8 focused on software testing, consisting of functionality testing and performance testing, and user testing. The tests that *Half-Blood* undertook to verify its functionality were laid out in table form with the expected result matched to the actual result. The performance testing was carried out on a mid-range gaming PC to rate the game's scalability in its current state. Both sections of the software testing found the game to be performing at a satisfactory level, these results can be found in Appendix D and E. The user testing phase was used to answer the questions established in the research and planning section. Users were submitted to a full cycle play through of three differing lengths unknown to them, several study methods were then used to infer data from each playtest. The results found that the quickest cycle was the most enjoyable and that the one that the participants felt most engaged in. This shows a clear correlation in engagement and enjoyment. However, the consensus for the fairest cycle was found to be medium speed, the control cycle time inspired by many other games in this domain. The plan and all the results from user testing can be found in Appendix F.

Finally, the product created and the ongoing project process have been scrutinised meticulously in the evaluation chapters. These chapters act as a moment of personal reflection where it is identified how prosperous the report has been at accomplishing the primary aims set out.

12.2 Recommendations and Future Work

Due to the closure of this project, it would be beneficial to compose a list of recommendations and make a plan of future work if the contents of this project are to be taken further.

Firstly, the largest recommendation that this paper offers is the breadth of resources used to form the literature review. These papers formed the foundation of this study and aided in designing a game demo that has been enjoyed by many of the testers. The literature focusing on what can affect an attention span and also how attention spans can actually be measured were fascinating introductions to such an interesting domain of research. The preconceptions of human attention are put under such scrutiny such that the myths we've grown to take as fact, are dismantled and proven as factually incorrect. Perhaps the most interesting study was one which proved that goldfish can in fact solve complex memory problems, contrary to what is believed in popular media (Ryssdal, 2014). Furthermore, a recommendation must be made for the Unreal Engine as it served as the primary development tool throughout the entire implementation phase. Without a piece of software capable of such high levels of abstraction and high-level features, this project would not have been able to exceed in creating such an enjoyable gaming experience. Not only does the software allow for incredible games to be built, but the intuitive UI makes for the development process to be an enjoyable and pleasurable experience in itself.

In regards to future work, there could be extensive work performed in both the process and the product. Firstly, the process could be extended with much more specialist equipment in the testing phase. With highly scientific equipment the data would form a much clearer picture into how the player was reacting on a subconscious level while play-testing. In addition, a longer time period for the project in a world where face-to-face contact was permitted would undoubtedly lead to a plethora of high quality data. On the other hand, the product itself could also see improvement with further development. The improvements would consist of a more exciting and levelled combat system with the potential for much more tactical choice. As well as this, the scope of the product into either a multi-level system or a larger playable world could be looked at as a test to the scalability of the current product. While the users enjoyed the differing attack types, the inclusion of a more combat-savvy AI would definitely introduce another level of depth to player immersion.

Finally, it is hoped that this project has been influential in intriguing others into the world of gaming psychology. The work in measuring attention spans within a gaming context will be vital for so many when designing immersive environments for the player. In addition, when stepping out of the gaming domain, the research performed in this paper could also be vital to many other areas. Upon researching attention span difficulties in other domains, the paper entitled "Attention during Lectures: Beyond Ten Minutes" (Wilson and Korn, 2007) questioned whether students' attention was at its peak for the duration of a lecture. The paper found that there was not enough "controlled research" to support this claim but that "students' attention does vary during lectures". Therefore, the controlled research performed in this project could be used in a different context to aid research in other sectors. This research of attention spans may have already been used to some effect as there has been somewhat of a recent trend of lecturers splitting up online lectures into shorter sections in

order to maintain student attention, this can be seen in many highly reputable institutions (LSA Learning & Teaching Technology Consultants, 2020) (Ferrero, 2021) (D'Lima, 2020).

References

‘Attention span’ (2021) Available at: https://dictionary.cambridge.org/dictionary/english/attention-span (Accessed: 27 November 2021).

Beepa Pty Ltd (2013) *Fraps* (3.5.99) [Computer Program] Available at: https://fraps.com/ (Accessed: 20 April 2022).

Bell, J.J. and Hardy, J., (2009),’ Effects of attentional focus on skilled performance in golf’, *Journal of applied sport psychology*, *21*(2), pp.163-177.

Clegg, D. and Barker, R., (1994), *Case method fast-track: a RAD approach*. Addison-Wesley Longman Publishing Co., Inc..

Csikszentmihalyi, M., 1990. *Flow: The psychology of optimal experience* (Vol. 1990). New York: Harper & Row.

‘Day-Night Cycle’ (2022) *Don't Starve Wiki*. Available at: https://dontstarve.fandom.com/wiki/Day-Night_Cycle (Accessed: 18 December 2021).

David Cornish, M. and Dukette, D., (2009), *The essential 20: Twenty components of an excellent health care team*. Dorrance Publishing.

D’Lima, D. (2020) *‘Chunking’ lectures into shorter, meaningful recordings for online delivery*. Available at: https://www.ucl.ac.uk/teaching-learning/case-studies/2020/oct/chunking-lectures-shorter-meaningful-recordings-online-delivery (Accessed: 27 April 2022).

Dobbs, A.R. and Rule, B.G., (1989), Adult age differences in working memory. *Psychology and aging*, *4*(4), p.500.

Epic Games (2022a) *Unreal Engine* (4.26) [Computer Program] Available at: https://www.unrealengine.com/en-US/download (Accessed: 4 April 2022).

Epic Games (2022b) *Blueprint Overview*. Available at: https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints/Overview/ (Accessed: 8 April 2022).

Epic Games (2022c) *Set Value as Enum*. Available at: https://docs.unrealengine.com/4.26/en-US/BlueprintAPI/AI/Components/Blackboard/SetValueasEnum/ (Accessed: 10 April 2022).

Ferrero, M.A. (2021) *How to Split Your Online Lecture to Keep Your Students Engaged*. Available at: https://mariangelf.com/how-to-split-your-online-lecture-to-keep-your-students-engaged/ (Accessed: 27 April 2022).

Flôres, F.S., Menezes, K.M. and Katzer, J.I., (2016), 'Influences of gender on attention and learning of motor skills', *Journal of Physical Education*, *27*.

Game Freak (2006) *Pokémon Diamond Version* [Video game]. Nintendo. Available at: https://www.nintendo.co.uk/Games/Nintendo-DS/Pokemon-Diamond-272354.html (Accessed: 17 December 2021).

Hasher, L. and Zacks, R.T., (1988), Working memory, comprehension, and aging: *A review and a new view. Psychology of learning and motivation*, 22, p. 219.

Hunt, S. (2010). "The Ultimate Hero: A Complete History of Sabreman". *Retro Gamer* (Issue #73), pp. 24–31.

Dunning, J. (2014) *Dying Light Developers Now Targeting 1080p/30fps on PS4*. Available at: https://www.playstationlifestyle.net/2014/12/10/dying-light-developers-now-targeting-1080p30fps-ps4/ (Accessed: 18 December 2021).

Kean, K. (2022) *The 5 Best Game Engines for Beginners in Video Game Development*. Available at: https://www.makeuseof.com/best-free-game-engine-for-beginners/ (Accessed: 4 April 2022).

Klei Entertainment (2013) *Don't Starve* [Video game]. Klei Entertainment. Available at: https://store.steampowered.com/app/219740/Dont_Starve/ (Accessed: 17 December 2021).

Kojima Productions (2015) *Metal Gear Solid V: The Phantom Pain* [Video game]. Konami. Available at: https://store.steampowered.com/app/287700/METAL_GEAR_SOLID_V_THE_PHANTOM_PAIN/ (Accessed: 17 December 2021).

Lindenberger, U., Marsiske, M. and Baltes, P.B., (2000), Memorizing while walking: increase in dual-task costs from young adulthood to old age. *Psychology and aging*, *15*(3), p.417.

Lindley, C.A. and Sennersten, C.C., (2006), July. A cognitive framework for the analysis of game play: tasks, schemas and attention theory. In *Workshop on the cognitive science of games and game play, the 28th annual conference of the cognitive science society* (Vol. 29).

LSA Learning & Teaching Technology Consultants. (2020) *Breaking Up Video Lectures*. Available at: https://lsa.umich.edu/technology-services/news-events/all-news/teaching-tip-of-the-week/breaking-up-video-lectures.html (Accessed: 27 April 2022).

Matt Aspland (2021a) *How To Create A Day And Night Cycle - Unreal Engine 4 Tutorial*. 22 April. Available at: https://www.youtube.com/watch?v=Zf7Sc1vW6ks (Accessed: 2 April 2022).

Matt Aspland (2021b) *How To Create A Day And Night Cycle - Unreal Engine 4 Tutorial*. 26 April. Available at: https://www.youtube.com/watch?v=veDYC3VLg04 (Accessed: 2 April 2022).

Maybin, S. (2017) Busting the attention span myth, *BBC*, 10 March. Available at: https://www.bbc.co.uk/news/health-38896790 (Accessed: 1 December 2021).

Maylor, E.A. and Lavie, N., (1998), The influence of perceptual load on age differences in selective attention. *Psychology and aging*, *13*(4), p.563.

McKay, B. and Wulf, G., (2012), 'A distal external focus enhances novice dart throwing performance', *International Journal of Sport and Exercise Psychology*, *10*(2), pp.149-156.

McSpadden, K. (2015) 'You Now Have a Shorter Attention Span Than a Goldfish', *Time*, Available at: https://time.com/3858309/attention-spans-goldfish/ (Accessed: 2 December 2021).

Merritt, P., Hirshman, E., Wharton, W., Stangl, B., Devlin, J. and Lenz, A., (2007), 'Evidence for gender differences in visual selective attention', *Personality and individual differences*, *43*(3), pp.597-609.

Microsoft. (2015), 'Attention spans', *Consumer Insights: Microsoft Canada*, Available at: http://pausethinkconsider.com/wp-content/uploads/2016/09/microsoft-attention-spans-research-report.pdf (Accessed: 29 November 2021).

Mojang Studios (2011) *Minecraft* [Video game]. Mojang Studios. Available at: https://www.minecraft.net/en-us/ (Accessed: 17 December 2021).

'Multitasking' (2021) Available at: https://dictionary.cambridge.org/dictionary/english/multitasking (Accessed: 29 November 2021).

Nacke, L. and Lindley, C.A., (2008), November. Flow and immersion in first-person shooters: measuring the player's gameplay experience. In *Proceedings of the 2008 conference on future play: Research, play, share* (pp. 81-88).

National Health Service (2018) *Treatment - Generalised anxiety disorder in adults*. Available at: https://www.nhs.uk/mental-health/conditions/generalised-anxiety-disorder/treatment/ (Accessed: 1 December 2021).

Nintendo EAD (2005) *Animal Crossing: Wild World* [Video game]. Nintendo. Available at: https://www.nintendo.co.uk/Games/Nintendo-DS/Animal-Crossing-Wild-World-270011.html (Accessed: 17 December 2021).

PixelatedDread. (2022) 'Night time in freeroam is too short in FH5', *Reddit: r/ForzaHorizon*, 20 January posted. Available at: https://www.reddit.com/r/ForzaHorizon/comments/s8ntzg/night_time_in_freeroam_is_too_short_in_fh5/ (Accessed: 24 March 2022).

Playground Game (2021) *Forza Horizon 5* [Video game]. Xbox Game Studios. Available at: https://store.steampowered.com/app/1551360/Forza_Horizon_5/ (Accessed: 17 December 2021).

Ponds, R.W., Brouwer, W.H. and Van Wolffelaar, P.C., (1988), Age differences in divided attention in a simulated driving task. *Journal of Gerontology*, *43*(6), pp.P151-P156.

Posner, M.I. and Cohen, Y., (1984), 'Components of visual orienting', *Attention and performance X: Control of language processes*, *32*, pp. 535.

Posner, M.I., (1980), 'Orienting of attention', *Quarterly journal of experimental psychology*, *32*(1), pp.3-25.

Nystrom, R. (2021), *Game Loop*. Available at: https://gameprogrammingpatterns.com/game-loop.html (Accessed: 18 December 2021).

Rockstar North (2013) *Grand Theft Auto V* [Video game]. Rockstar Games. Available at: https://store.steampowered.com/app/271590/Grand_Theft_Auto_V/ (Accessed: 17 December 2021).

Ryssdal, K., (2014), 'Goldfish have longer attention spans than Americans, and the publishing industry knows it', *Marketplace Business*, *11*.

RZX Archive (2013) *Knight Lore Walkthrough, ZX Spectrum*. 26 June. Available at: https://www.youtube.com/watch?v=7n7qtErhF-A& (Accessed: 18 December 2021).

Salthouse, T.A., Rogan, J.D. and Prill, K.A., (1984), Division of attention: Age differences on a visually presented memory task. *Memory & Cognition*, *12*(6), pp.613-620.

Sennersten, C.C. and Lindley, C.A., (2009), 'An investigation of visual attention in FPS computer gameplay', In *2009 Conference in Games and Virtual Worlds for Serious Applications* (pp. 68-75).

Skyline_R32_187. (2022) 'Extend the night time!', *Forza Motorsport Forums*, 10 March posted. Available at: https://forums.forzamotorsport.net/t/extend-the-night-time/539481 (Accessed: 24 March 2022).

Sohlberg, M.M. and Mateer, C.A., (1987), Effectiveness of an attention-training program. *Journal of clinical and experimental neuropsychology*, *9*(2), pp.117-130.

Solianik, R., Brazaitis, M. and Skurvydas, A., (2016), 'Sex-related differences in attention and memory', *Medicina*, *52*(6), pp.372-377.

Somberg, B.L. and Salthouse, T.A., (1982), Divided attention abilities in young and old adults. *Journal of Experimental Psychology: human perception and performance*, *8*(5), p.651.

Stoet, G., O'Connor, D.B., Conner, M. and Laws, K.R., (2013), 'Are women better than men at multi-tasking?', *BMC Psychology*, *1*(1), pp.1-10.

Techland (2015) *Dying Light* [Video game]. Warner Bros. Interactive Entertainment. Available at: https://store.steampowered.com/app/239140/Dying_Light/ (Accessed: 17 December 2021).

Tyler, D. (2022) *How to Choose the Best Video Game Engine*. Available at: https://www.gamedesigning.org/career/video-game-engines/ (Accessed: 4 April 2022).

Ubisoft Montreal (2016) *Far Cry Primal* [Video game]. Ubisoft. Available at: https://store.steampowered.com/app/371660/Far_Cry_Primal/ (Accessed: 17 December 2021).

Ultimate Play the Game (1984) *Knight Lore* [Video game]. Ultimate Play the Game. Available at: http://bbcmicro.co.uk/game.php?id=208 (Accessed: 17 December 2021).

Unity Technologies (2022) *Unity* (2021.2.13) [Computer Program] Available at: https://unity.com/download (Accessed: 4 April 2022).

Wilson, K. and Korn, J.H., (2007), 'Attention during lectures: Beyond ten minutes', *Teaching of Psychology*, *34*(2), pp.85-89.

Zagal, J.P. and Mateas, M., (2010), Time in video games: A survey and analysis. *Simulation & Gaming*, *41*(6), pp.844-868.

Bibliography

Bellotti, F., Berta, R., Gloria, A.D. and Primavera, L., (2009), Enhancing the educational value of video games. *Computers in Entertainment (CIE)*, *7*(2), pp.1-18.

Chen, J., (2007). Flow in games (and everything else). *Communications of the ACM*, *50*(4), pp.31-34.

Csikszentmihalyi, M. (1975) *Beyond Boredom and Anxiety: Experiencing Flow in Work and Play*. San Francisco: Jossey-Bass.

Dean Ashford (2018) *UE4 - Tutorial - Day/Night Cycle!*. 18 August. Available at: https://www.youtube.com/watch?v=6KKBcivkKH4
(Accessed: 2 April 2022).

Ellis, G.D., Voelkl, J.E. and Morris, C., 1994. Measurement and analysis issues with explanation of variance in daily experience using the flow model. *Journal of leisure research*, *26*(4), pp.337-356.

Igarzábal, F.A., (2019), *Time and Space in Video Games*. transcript-Verlag.

Igarzábal, F.A., (2020), Experiencing the passage of time in video games.

IJsselsteijn, W., Van Den Hoogen, W., Klimmt, C., De Kort, Y., Lindley, C., Mathiak, K., Poels, K., Ravaja, N., Turpeinen, M. and Vorderer, P., (2008), August. Measuring the experience of digital game enjoyment. In *Proceedings of measuring behavior* (Vol. 2008, No. 2008, pp. 88-89). Maastricht, the Netherlands: Noldus.

I See 3D (2021a) *Unreal Engine 5 - Create a day night cycle in 90 seconds*. 19 July. Available at: https://www.youtube.com/watch?v=UZSRtwIa3Ls
(Accessed: 2 April 2022).

I See 3D (2021b) *UE4 Day Night Cycle With Moon*. 3 October. Available at: https://www.youtube.com/watch?v=rQTdrBUw8a4
(Accessed: 2 April 2022).

Krähenbühl, P., (2018), Free supervision from video games. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2955-2964).

Khoshnoud, S., Igarzábal, F.A. and Wittmann, M., (2020), Peripheral-physiological and neural correlates of the flow experience while playing video games: a comprehensive review. *PeerJ*, *8*, p.e10520.

McDowd, J.M., (1997), Inhibition in attention and aging. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, *52*(6), pp.P265-P273.

Nitsche, M., (2007), September. Mapping Time in Video Games. In *DiGRA Conference*.

Rouse III, R., (2009), Match Made in Hell: The inevitable success of the horror genre in video games. *Horror video games: Essays on the fusion of fear and play*, pp.15-25.

Ryan Laley (2019) *Unreal Engine 4 Tutorial - Day / Night Cycle*. 20 June. Available at: https://www.youtube.com/watch?v=4ahk-4FNfow
(Accessed: 2 April 2022).

Sweetser, P. and Wyeth, P., 2005. GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)*, *3*(3), pp.3-3.

underscore (2018) *UE4 Tutorial: Day/Night Cycle (w/ Clock and Moon) (Request)*. 6 August. Available at: https://www.youtube.com/watch?v=x3xBtPIVWEs
(Accessed: 2 April 2022).

Appendices

A. Terms of Reference

KV6003: Individual Computing Project

Project Terms of Reference

Name: Peter David Stuart

Student ID: W18010680

Course of study: Computer Science with Games Development

Creating a video game where the day/night cycle is based upon relative psychological factors and attention span research.

Supervisor: Dan Hodgson

Second Marker: James Nicholson

General Computing Project

Project Title

Creating a video game where the day/night cycle is based upon relative psychological factors and attention span research.

Background to Project

This project has been chosen due to its intrigue surrounding the mental aspects of a gamer's experience. When playing games the player may become oblivious as to how a game's cycle is affecting their psychological well-being. I am extremely interested in the area of study surrounding enjoyment based upon how much action the player is subjected to at one given time. My personal favourite method of delivering changing environments and levels of action to the player is by using a day/night cycle.

To explain, a day/night cycle is when a game constantly cycles through changing environmental factors in a clock based system. In some instances the change of day to night and night to day is simply just for aesthetic purposes. However, in the more interesting cases, the change in time can produce completely different worlds and atmospheres for the player to explore.

For example, some popular games that make use of a day/night cycle are: Minecraft, Dying Light and Red Dead Redemption 2.

The nature of the intended product is a playable demo and research paper looking into player's enjoyment of games depending on how much "uptime" and "downtime" they are given. Following previously about cycles, I believe in the examples picked that their cycles are simply a facade for the game's intensity scale going up and down. For instance, in Minecraft during the day there is little threat to the player and this time is mainly spent building, crafting, mining and farming. On the other hand, once night falls the player is susceptible to mercenaries which try to attack and ultimately kill the player. The other example in Dying Light, faces the player against much tougher opposition once the game cycles into its night mode. Finally, Red Dead Redemption 2 is a game where the significance of the night falling is not as severe as the other two, however the increase in enemies and animals attacking the player becomes much more severe.

When looking at the attention spans in gamers it can be seen that through the use of more and more technology we, as a society, are reducing our attention spans to media and more specifically to video games. This piece of knowledge will be imperative to my creation of a well crafted cycle that will not leave the player bored or, even worse, stop them playing.

The problem domain this project will be working in is a personal discovery whereby I would like to gain knowledge for future projects as the idea of a rich day/night cycle is greatly intriguing. When carrying out this project it is important to research heavily into gamer's current experiences with critically acclaimed and also heavily criticised games to get an idea as to what makes player's the most mentally satisfied when experiencing a game's full cycle.

Much initial research has been performed and a solid list of references have been established which I will use as the primary areas of my findings. These will be listed below in the "sources of information" section. From these sources I found that there are many research papers in this area of study, however there are few academic endeavours into my novel research proposal of specifically day/night cycles. This is a good thing because it means I can use the abundant generic papers to gain a good understanding of attention

spans and gaming psychology separately and then create my own paper combining these two ideas together in a novel piece of research.

Proposed Work

The proposed work for this general computing project is to make a proof of concept game along with a research paper aided by the testing within the game. This game will not be overly bloated with aesthetic assets but instead it will be used only to review how a player feels when playing a game that cycles between high and low intensity gameplay. This would mean that the playing time would exceed no more than one hour due to the time limitations and so not to lose the core concept idea.

This project will be utilising all the skills developed throughout my time at the university and has ties very close to the specific course specific pathway of Games Development. During my time at the university I have sought to develop a game of significant purpose and this opportunity will be the greatest chance I have to show that off.

To make this proof of concept game I will be using the Unreal Engine. This game engine is something that I have worked with in the past and I thus seek to use my existing skills to create a product that will aid in my research. As well as this, I believe stretching my knowledge on this particular game engine will help me expand my abilities in the game creation field.

The scope of the game will be limited, however the core and concept of the game will stay true to the problem domain being examined. This is primarily so the central idea of the game is not lost in a much more demanding endeavour. By using Unreal Engine I can ensure that the scope and size of the game stay within my boundaries due to my understanding of the engine's mechanics.

After the game has been created I will playtest it myself to iron out bugs and errors. Once I am happy with the way the game looks and feels I will then instigate testing with willing participants. This will all take place online and will give me a greater understanding of how players approach a game with a day/night cycle. The testing phase will be split into observations, interviews and questionnaires to give me an all-round insight into how successful my creation is.

Aims of Project

1. Investigate potential optimum day/night cycle for games
2. Create a game demo product as proof of concept

Objectives

- Literature review on attention spans and cycles of activity
- Review of day/night cycles of different games
- Establishing requirements
- Creating design documentation for the game
- Creating the game
- Testing the game (software testing + play testing)
- Evaluation of the product
- Evaluation of the project as a whole
- Personal reflection

Skills

This project has been chosen due to my ability and interest in the game's industry as well as my chosen degree being computer science with games development. The expertise gained from the past 3 years of study have built up a solid foundation of computing based skills of which game's specific proficiency can be expanded upon. Below is a list of skills I will need for this project to come to fruition.

Skill needed for project	**Is the skill currently known?**	**Module taken to gain skill**	**How the skill will be acquired**
Research Based			
Creating an ethically sound project plan	No	KV5003 Human Computer Interaction. I created an ethics plan with a group in this module but not to the extent needed for an individual computing project	By completing the lectures and workshop relating to an ethics plan created for this module
High standard of academic writing ability	Yes	KF5042 Intelligent Systems	N/A
Ability find the highest quality references	Yes	KF5042 Intelligent Systems	N/A
Referencing ability	Yes	KF5042 Intelligent Systems	N/A
Game Based			
Gameplay testing	Yes	KF5012 Software Engineering	N/A

		Practice	
Understanding and documenting people's experience with my product	Yes	KV5003 Human Computer Interaction	N/A
Unreal Engine gameplay programming	Yes but will be improved as creation continues	KF5012 Software Engineering Practice	Using tutor tutorials as well as YouTube videos relating to the scripting aspect of the Unreal Engine
Understanding the process is which games are made	No	KF6015 Games Design	This will be improved through the lectures in this module
Asset creation	No	N/A	This skill will be improved via external work performed in programs such as blender
World building	Yes but could do with improvement	KF6015 Games Design	This will be improved through the lectures in this module

Sources of Information

I can split the bibliography of my initial research into two sections: literature on specifically video games and their psychological implications and then general attention span literature. I have selected the three best and most broad in scope papers in each of these sections to give my research a solid foundation initially.

Video game and Psychology Literature

Orosy-Fildes, C. and Allan, R.W., 1989. Psychology of computer use: XII. Videogame play: Human reaction time to visual stimuli. *Perceptual and motor skills*, *69*(1), pp.243-247.

King, D., Delfabbro, P. and Griffiths, M., 2010. Video game structural characteristics: A new psychological taxonomy. *International journal of mental health and addiction*, *8*(1), pp.90-106.

Russell, W.D. and Newton, M., 2008. Short-term psychological effects of interactive video game technology exercise on mood and attention. *Journal of educational technology & society*, *11*(2), pp.294-308.

Attention Span Literature

Bray, D.A., 2008. Information pollution, knowledge overload, limited attention spans, and our responsibilities as IS professionals. *In Global Information Technology Management Association (GITMA) World Conference-June.*

Pope, A.T. and Bogart, E.H., 1996. Extended attention span training system: Video game neurotherapy for attention deficit disorder. *Child Study Journal*, *26*(1), pp.39-50.

Antzaka, A., Lallier, M., Meyer, S., Diard, J., Carreiras, M. and Valdois, S., 2017. Enhancing reading performance through action video games: The role of visual attention span. *Scientific reports*, *7*(1), pp.1-10.

Resources

Below is a list of the hardware and software that will be used to complete this project to its maximum best capacity. Please note that all software used on home devices will be properly licensed.

- Gaming PC built to handle AAA games and high quality game engines
    - If this becomes unavailable then there are top quality machines available in the games lab of the university
- Unreal Engine software downloaded which will be used to produce the game
    - If this becomes unavailable then the games lab has this software pre-installed
- Internet access and word processor to complete the research and subsequent report paper
    - These are both available all around campus should my machine become inoperative

When showcasing my product at the university I will have the files in the cloud ready to transfer onto a capable machine within campus. This will also mean that I will have a backup of all my workings should anything become corrupt.

Structure and Contents of Project Report

Report Structure Chapters

1. Intro
2. Literature review on attention spans and cycles of activity
3. Review of day/night cycles of different games
4. Establishing requirements
5. Creating design documentation for the game

6. Creating the game
7. Testing the game (software testing + play testing)
8. Evaluation of the product
9. Evaluation of the project as a whole
10. Personal reflection
11. Conclusions and recommendations

List of Appendices

A. Terms of Reference
B. Requirements Specification
C. Design Documents
D. Source Coding Link GitHub repo
E. Test Plans and Results
F. Documents Used in Investigative Work

Assessment Criteria

Product

	Fitness for Purpose	Build Quality
General Computing Project	50%	50%

Deliverables for marking:

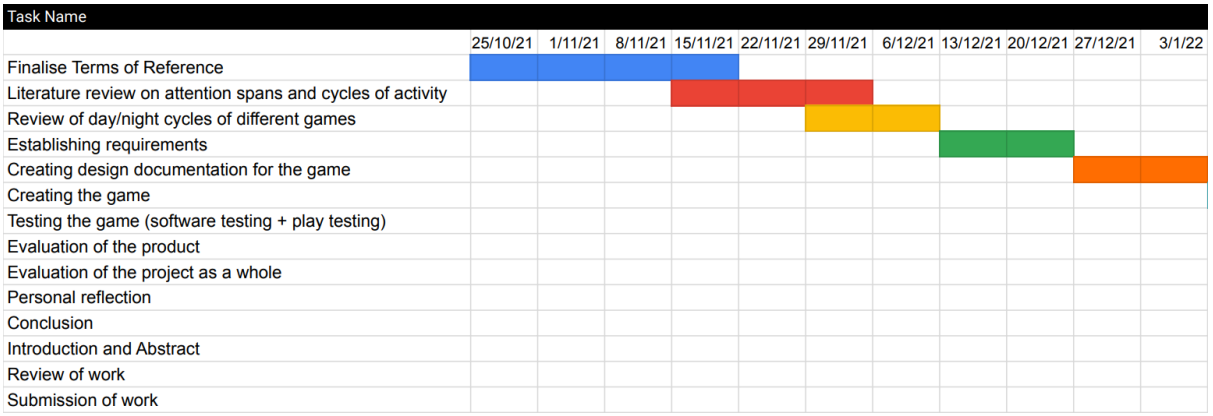
- Fitness for Purpose
    - Meeting of Requirements as identified during project
    - Quality of Functionality
    - HCI
- Build Quality
    - Requirements specification & analysis
    - Design specification
    - Code quality
    - Test plans and results

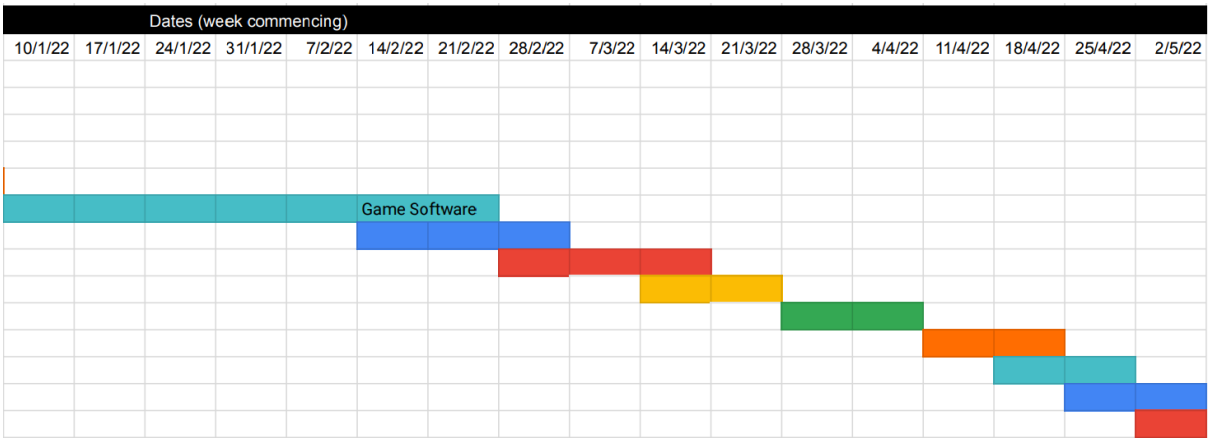
Project Plan

Due to program limitations the GANTT chart has been split in two, if this is hard to read then the link for it can be found below.

GANTT CHART LINK

https://docs.google.com/spreadsheets/d/1Vc7njQ4V6qg_BSThGypD4qYefcWhQcenb2jwPsEwsTY/edit?usp=sharing





This GANTT chart shows the tasks needed to complete each objective, the time taken (in hours) for each task, and finally when I aim to complete each task (in weeks).

Terms of Reference Review Form

This form is to be completed **by 2nd marker** after the TOR group review meeting and then sent to the student and the supervisor for signature. After the TOR group review, students should make any necessary changes to the TOR, online ethics and risk assessment forms (as required by your Supervisor and Second Marker), then submit the final approved TOR to Blackboard by the dates given in the module schedule whilst also completing the online ethics and risk assessment approval process.

Date of TOR Review: 29/10/21

Review Outcomes:

The topic is appropriate to the student's programme.	Yes
The project contains sufficient practical work using computing skills relevant to the programme.	Yes
An appropriate topic for the literature review has been identified.	Yes
An explanation of the contribution of the analysis/literature review to the project work has been given.	Yes

The TOR (select one):

● Accepted without changes.	X
● Needs the changes listed below.	
● Cannot be made satisfactory and a new topic is required.	

Ethics Draft PDF (select one):

● Has been reviewed and can be approved via the Ethics Online System.	
● Requires revision before approval can be granted via the Ethics Online System.	

● Has been reviewed; the project should be referred to the Faculty Research Ethics Committee (FREC) via the Ethics Online System.	
● The project has already been referred to FREC via the Ethics Online System.	
● Has not yet been provided.	X
● Other (please explain).	

Risk Assessment Draft PDF (select one):

● Has been reviewed and can be approved via the Ethics Online System.	
● Requires revision.	
● Is required but has not yet been provided.	
● Not required.	X

Changes required/identified issues:

Changes to proposed aim(s):

Changes to proposed objectives:
Changes to deliverables:
Changes to structure and contents of project report: Missing details about how playtesting will be organised and evaluated – no need to change the TOR at this stage but just make sure you think about this carefully.
Changes to project plan:
Resource issues:
Other comments:

Signatures: **Student: Peter Stuart**

Supervisor: Dan Hodgson

Second Marker: James Nicholson

B. Game Design Document

Half-Blood

Game Design Document

Abstract

A day/night cycle is used to instigate varying behaviours in enemy vampire AI. The player must survive as a half-blood in a land where vampires seek domination.

Peter David Stuart
KV6003 - Individual Computing Project

Introduction

Game Overview

Game Name: Half-Blood
Genre: Survival horror
Platform: PC
High view of the game: The player must survive a group of vampires as they attack during the night and hide during the day
Dimension: 3D
Perspective: 3rd Person
Target Demographic: 18+ (gore, horror , violence)
Tone of game: Scary, Survival, Relentless
Tags: Action, Horror, Survival, Day/night cycle, Third-Person, Single player, Stealth

Background Lore

The player has become a half-blood vampire after quickly treating themselves with an antidote in a vampire attack. Although lucky to escape becoming a full vampire, they have been left with some of the characteristics. They are still unaffected by sunlight however, they cannot enter someone's house without permission.
Plunged into the middle of a vampire lair, will they be able to make it out alive in the coming days?

Overview of Gameplay

The aim of the game in this proof-of-concept demo is to simply survive the attacks of the vampires at night. To prevent death during these night attacks the player must either hide from the enemies or attack and kill them one by one. As the day/night cycle advances, the enemies behaviour towards the player changes between: seeking an attack, patrolling and hiding.

Gameplay

The game starts with the player able to choose between three possible cycle lengths. Upon choosing one they are sent into a world of vampires, each with their own house, that will attempt to kill the player at night. The game starts at 12:00 to give the player a chance to regroup and plan their night attack.

The main aim is to survive the ongoing night attacks while also trying to pick off the attacking enemies. The player must use stealth and tactics to manoeuvre around the world without alerting too many enemies. If the player becomes surrounded then they will certainly die.

The win condition is to kill every enemy in the given world. The lose condition is dying to the enemy vampires.

The enemies work in states. During dawn and dusk hours they will patrol close to their homes in order to guard themselves and also to be able to quickly flee should the Sun come up. During the day they will hide from the Sun as this will periodically damage them. Finally, during night the vampires will actively seek out to kill the player by randomly searching random points around the map.

To kill the vampires, the player can use a combination of light and heavy attacks. The choice between these attacks will make for interesting gameplay options as one is slow with large damage output, and the other quick with weak damage output. However, the enemy can also attempt to kill the player. It is equipped with a single attack that will cut the player's health down significantly.

User Interface

Controls

The controls used are WASD for movement.
Left click on the mouse will be used for a light attack (faster to complete but less damage).
Right click on the mouse will be used for a heavy attack (slow to complete but more damage).

Due to the limited controls, controller support has been implemented. The left analogue stick is used for movement, the right for camera. The top face button for heavy attack, left for light attack, right for crouch, and bottom for jump. The directional pad is used to pause and quit the game with left and up being used respectively.

HUD

There are two aspects to the HUD that the player will need to know about in this iteration of the game.

- The time: The time is imperative to the player as it dictates the behaviour of the enemies. This will be shown in a “day: hour : seconds” format.
- Health: As the only lose condition for the player is to die, they need to be able to see their health as it goes down

Menus

The current menus will only be a barebones start screen and a game over screen.

- The start screen will simply have a start and quit button.
- The game over screen will only have a quit button.

Character Design

Player Design

For simplicity the player will be a solid white mannequin with a different body type to the enemies. The player will also always have a stake in their hand ready to use on the enemies.

Enemy Design

The enemies will have a larger body build to the player and be coloured bright red. They too will be holding a stake in their hand to attempt to kill the player with.

Map Design

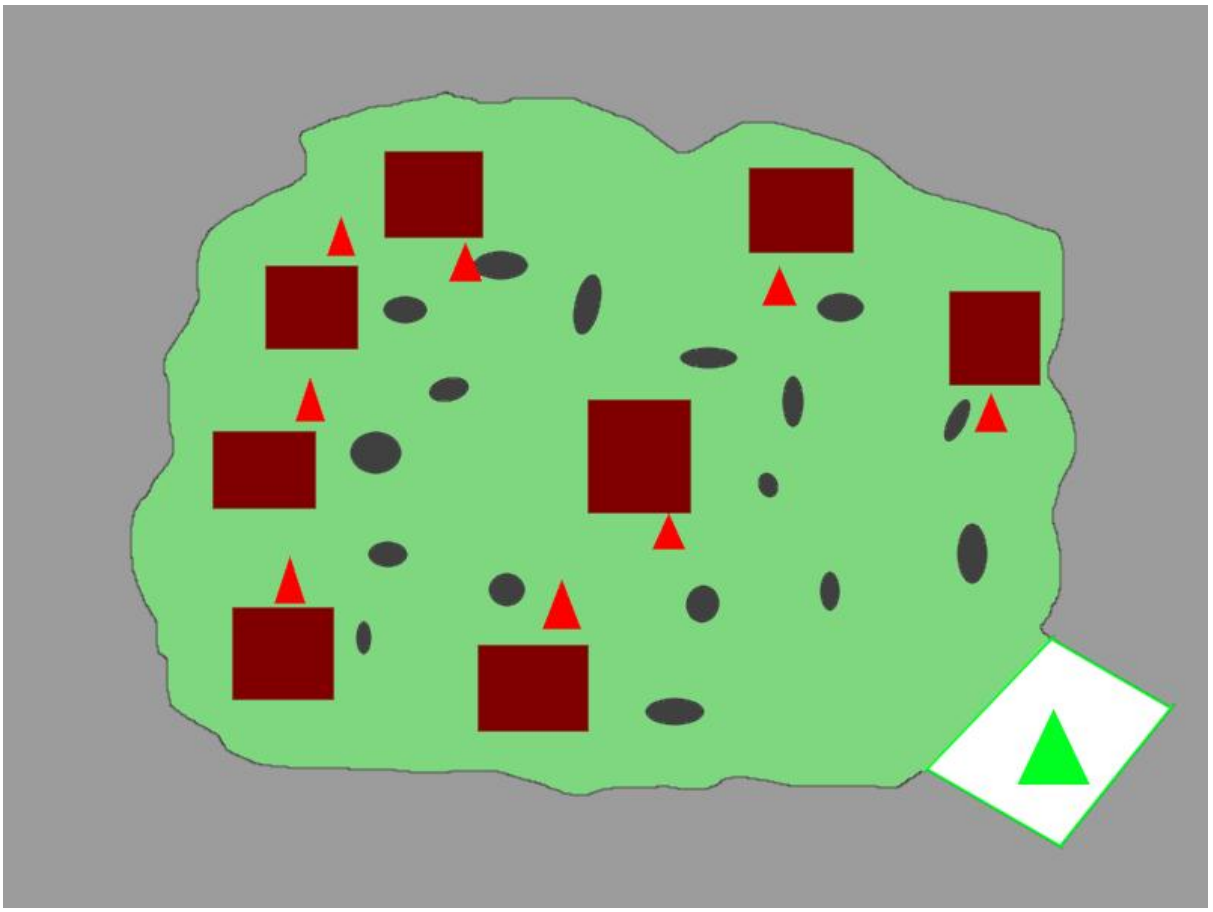
General Layout

The general layout is a village in the centre of a mountainous region. The mountains will be used as a barrier to trap the player in the play area.

Within the village will be an array of houses, each home to a vampire. They will be spread out enough so the player can move about and hide and not be overwhelmed by the amount of enemies.

Each enemy will have a patrol path that circles around their house area and so the player should not want to get too close.

Diagram



Key:

- Brown square: House
- Red triangle: Vampire
- Grey oval: Rock
- Green triangle: Player

- Grey border: Mountains
- Green middle: Grass

C. Source Code Link

Source Code

OR

https://livenorthumbriaac-my.sharepoint.com/:f:/g/personal/w18010680_northumbria_ac_uk/EnsEXWoWRttOtltia2yJpyYBS6w-mmTn7Ft_fLNY3MBnhA?e=CAZ962

Finished Product Link

Application

OR

https://livenorthumbriaac-my.sharepoint.com/:f:/g/personal/w18010680_northumbria_ac_uk/EpB9KNJoGldHhuy0Dl8Iq_cBJ28u8zrBJznSy7GS2UaunA?e=UvPOy7

D. Functionality Testing Results

Results are defined in terms of PC input for simplicity.

Test Case	Prerequisite(s)	Player Action	Expected Result	Actual Result
Basic gameplay mechanics				
Movement	Player is alive	WASD key presses	Player moves in the cardinal direction pressed	Functions as intended
Jumping	Player is alive	Space key press	Player jumps into the air	Functions as intended
Crouching	Player is alive	CTRL key hold	Player crouches while key is held, uncrouches when key is released	Functions as intended
Pausing	Player is alive	P key press	Game is paused and nothing in the game moves	Functions as intended
Combat and Stealth				
Player attacks enemy with light attack	Enemy is within range	Left mouse button is pressed	Enemy receives 15 damage points from the player	Functions as intended
Player attacks enemy with heavy attack	Enemy is within range	Right mouse button is pressed	Enemy receives 30 damage points from the player	Functions as intended
Enemy attacks player	Player is within range	N/A	Player receives 12 damage points from enemy	Functions as intended
Player stealth	Player is behind large cover	CTRL key is held	Enemy cannot see the player	Functions as intended
Enemy AI				
Enemy takes Sun damage	Enemy is not inside, time state is day	N/A	Enemy suffers 5 damage points for every	Functions as intended

			second they are outside	
Enemy patrols path	Enemy cannot see player, time state is dawn or dusk, enemy has a patrol path assigned to them	N/A	Enemy patrols up and down their patrol path	Functions as intended
Enemy chases player	Enemy can see player, time state is not day	N/A	Enemy finds player location and runs to it	Functions as intended
Enemy runs to bed	Enemy is not inside, time state is day	N/A	Enemy finds their bed location and runs to it	Functions as intended
Menu System				
Player presses Play on Main Menu	Game is launched	Left mouse click	Game begins	Functions as intended
Player presses Quit on Main Menu	Game is launched	Left mouse click	Game is closed	Functions as intended
Player presses Quit on Game Over	Player health is less than or equal to 0	Left mouse click	Game is closed	Functions as intended

E . Performance Testing Results

The following tests were performed using *Fraps* (Beepa Pty Ltd, 2013) on a mid to high end gaming PC, see specs below:

CPU	Intel i5 4690k
GPU	NVIDIA GTX 980
RAM	32GB DDR3
Storage System	SSD

Frame Rate

The cycle speed variable was increased to see the effect this had on the game's FPS. The **control FPS is 89.625**, this is with 8 enemies and a cycle speed of 1.8. This control data was chosen as it is the data that will be used for the final release.

Cycle Speed	Frame Rate per Second (FPS	Min	Max	Playable?
0.5	82.500	77	93	Yes
10	87.100	77	100	Yes
50	87.300	77	102	Yes
100	84.400	76	94	Yes
500	83.800	78	93	Yes
1,000	85.600	79	94	Yes
10,000	89.200	75	117	Yes

The enemies were then increased to see the effect this had on FPS. Note: the speed was set to 10 during these tests

Enemy count	Average FPS	Min	Max	Playable?
1	90.600	79	109	Yes
5	86.500	78	104	Yes
10	93.900	78	121	Yes
20	79.800	74	87	Yes
50	65.200	58	74	Almost
100	34.500	27	50	No

F. User Testing Plans and Results

User Testing Plan

The data gathering will be broken down into 3 specific areas.

Observation

Description

Observation - I will observe first hand how the participants react to my game by using a webcam or camera focused directly on their face and body. From the resulting movements of gazing it will be interpreted how they are feeling, this will also be paired with a “think aloud” commentary from them.

Interview

Description

Interview - The participant will then talk to me directly (online) about how they felt the experience went and fill me in on how they were feeling inside.

There will be no more than 5 questions asked and these questions will be based directly on the observations noted.

Questionnaire

Description

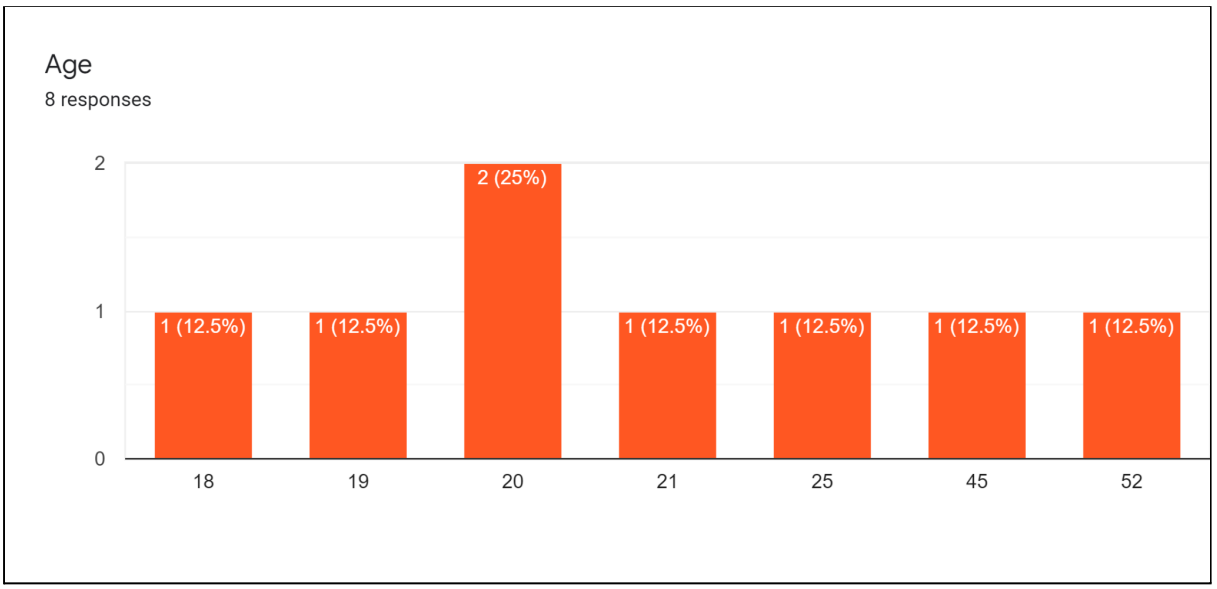
Questionnaire - The participant will then finally fill in a set of questions, that I will create during the game documentation stage, to help me gain some quantitative knowledge about their experience

Questions

1. Participant Number
2. Age
3. How would you rate your skills within gaming?
4. Could you tell a clear difference in cycle time between the 3 options?
5. In terms of enjoyment, which cycle was the best?
6. In terms of fair gameplay, which cycle was the best?
7. Did you find the damage totals from both you and the enemies fair?
8. If No to the previous question, how would you change the damage totals?
9. Did you find the instructions and controls clear?
10. How engaged were you during the quick cycle?
11. How engaged were you during the medium cycle?
12. How engaged were you during the slow cycle?

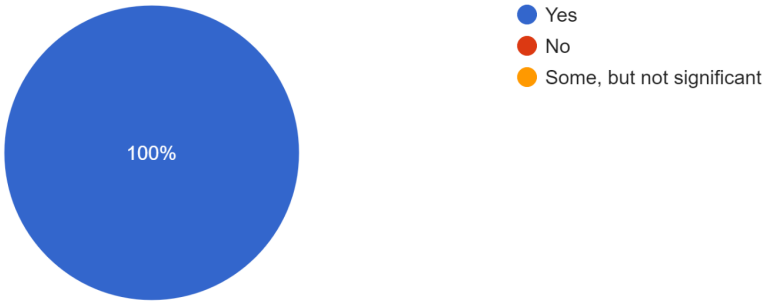
13. In no more than three points, what could be added or removed to improve your gameplay experience?

User Testing Results



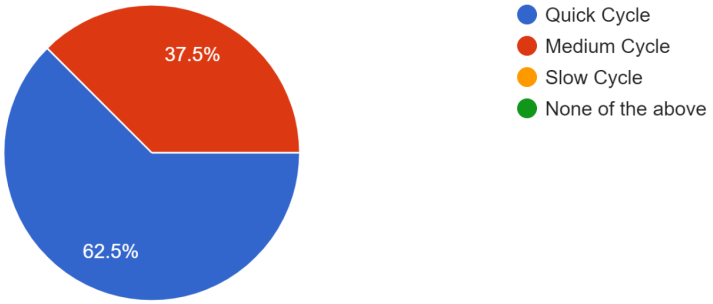
Could you tell a clear difference in cycle time between the 3 options?

8 responses



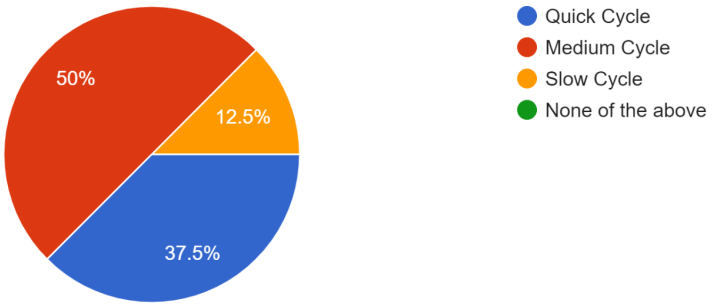
In terms of enjoyment, which cycle was the best?

8 responses



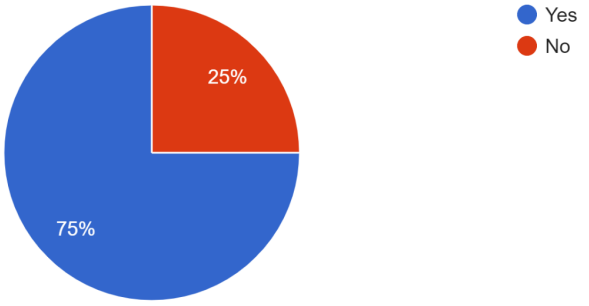
In terms of fair gameplay, which cycle was the best?

8 responses



Did you find the damage totals from both you and the enemies fair?

8 responses



If No to the previous question, how would you change the damage totals?

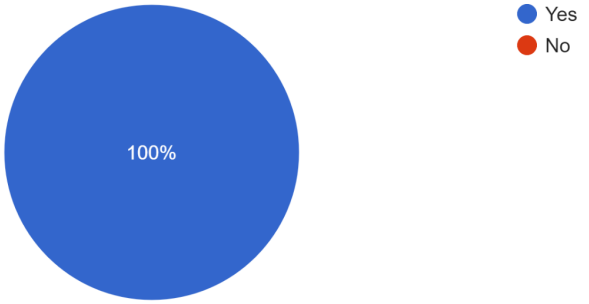
2 responses

I wish that there was a percentage number as well next to the health bar, so can see how well I am doing

Slow down health regeneration and increase damage from enemies

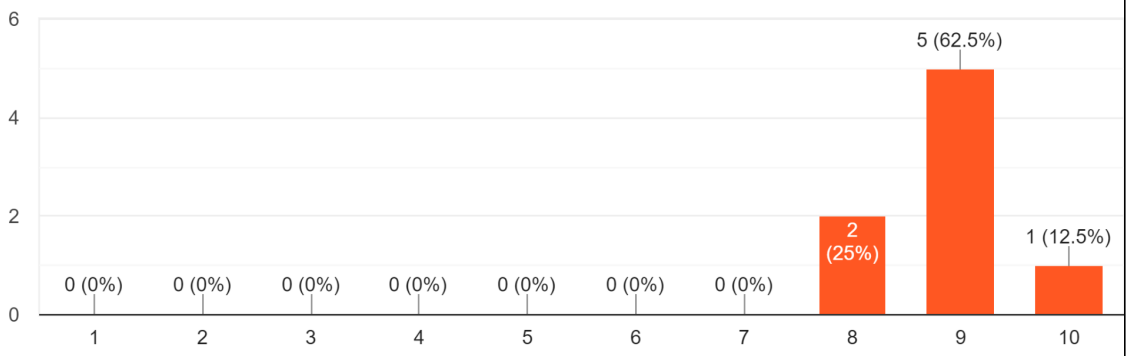
Did you find the instructions and controls clear?

8 responses



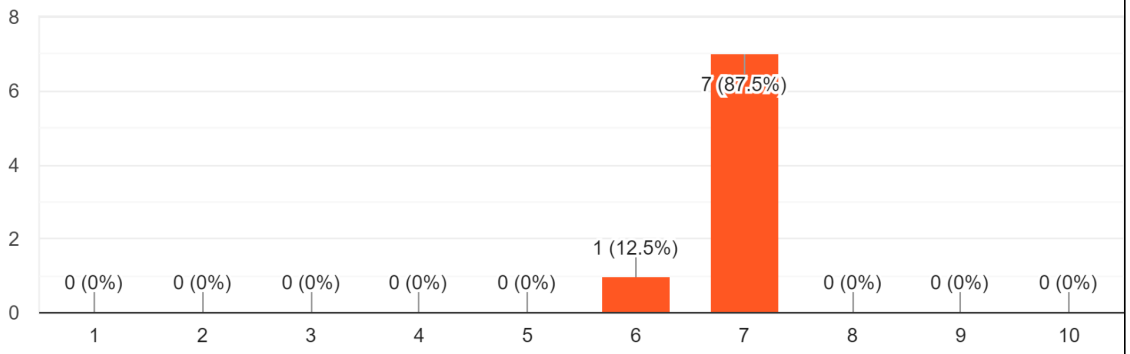
How engaged were you during the quick cycle?

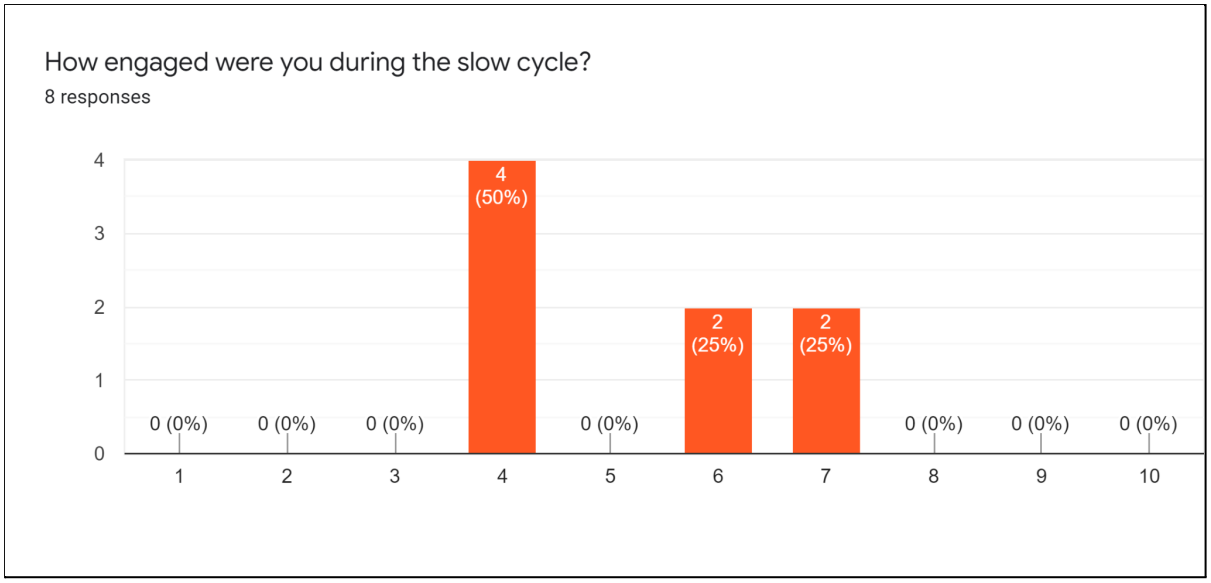
8 responses



How engaged were you during the medium cycle?

8 responses





In no more than three points, what could be added or removed to improve your gameplay experience?

8 responses

potentially more hiding places, maybe a couple big bolders or trees but might overcrowd the main area as its not huge anyway

sometimes the attacks weren't hitting the vampires properly, but apart from that I enjoyed playing :)

More fluent combat, sometimes it was hard to make contact with an enemy to deal damage

An explanation on what to do when you load Into the game

Sound effects would immerse me more in the gameplay, or some background music. Some more hiding spots would be great too to make each cycle last longer. Otherwise really good and engaging

Add gameplay options during daytime. Add in weapons and or equipment. Increase difficulty as you start the new cycle.

level of darkness too dark on the night time so can be hard to see what's going on.

Nothing to do during the daytime cycle, needs to have something like side tasks to be completed during the daytime or else players could lose interest. Maybe the ability to earn weapons or greater damage punches?

All Responses

[User Testing - KV6003 (Responses).xlsx]

OR

https://livenorthumbriaac-my.sharepoint.com/:x:/g/personal/w18010680_northumbria_ac_uk/EY-NAeRpsWpAoJ0yhJHYKTABDIAfEaNAxs77O8cS1qSTRg?e=euejR0