

CLASSIFIED

Fright_Beat

Making_Of Document

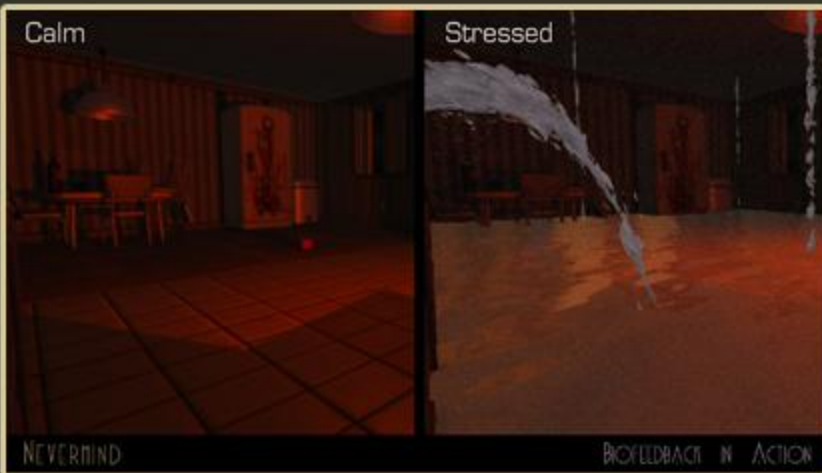


W012287L

Introduction - Project Summary

The focus of this Project was the Design & Development of a Horror Experience which makes use of Heart Rate Technology to deliver more effective scares & a more immersive experience.

Choosing the Horror Genre was the clear choice for this game, however, the Chase Horror Sub-Genre was chosen as when done right it can create potent dread and paranoia in the player while still being an engaging gameplay loop.



Having the Game be engaging by itself was a key consideration in its design. Games which have previously used Heart Rate Technology are usually walking simulators where the player can get easily bored and settle at a low HR.

The Hypothesis that motivates this project is that by adding more tense gameplay, the Heart Rate Integration can be even more effective than done before.

The project then became to create engaging gameplay which didn't rely on the Heart Rate Technology but Bolstered its Effectiveness. This is why a lot of the games that were used for research were games which had the player constantly tense & on the run.

Through this research, it was made clear that successful horror games relied on all types of design. The game had to include successful Audio Design and Level Design to build that tension and pay it off.

Because of this, the project was mainly focused around the Design elements rather than programming practises & efficiency.



W012287L

Introduction - Objectives

The Core Goals of this Project are outlined in the Game Design Document's Design Pillars. Heart Rate Integration is the main focus of the project. This includes both the method of integrating the Heart Rate Monitor as well as how the Game is designed to React.

The Successful Horror Elements & Replayable Gameplay are the necessary secondary objectives that must be hit in order for the Heart Rate Integration to be most effective.

Game Pillar	Explanation & Goals
Heart Rate Integration	<p>The core system which sets this project apart from its competitors is the Heart Rate Reading, this allows the game to adapt to the player's heart rate to make things more intense. This system will need to be noticeable and have a proper impact on the gameplay.</p> <p>Some Goals to meet this pillar are:</p> <ul style="list-style-type: none"> • Integrate the Heart Rate with the Active Game Elements like Enemy AI and Player Movement. • Integrate the Heart Rate with the Passive Game Elements like Sound Effects and Materials. • Create a range of different values so the experience is suited for the player's unique requirements (gender, age).
Successful Horror Elements	<p>This game is primarily a horror game first and foremost and the success of the horror elements is especially important as if the player isn't scared then the heart rate mechanics won't be as effective. Creating the necessary tension and fear for the heart rate mechanics to work effectively will require a good understanding and implementation of Atmosphere and Immersion.</p> <p>Some Goals to meet this pillar are:</p> <ul style="list-style-type: none"> • Create a building atmospheric soundtrack which will be affected by the heart rate. • Use lighting to create a tense environment around the player and act as affordance to objectives. • Learn & integrate effective jump scaring and climactic scares to act as a payoff for the tension.
Quick and Replayable Gameplay (arcade-like)	<p>Each Full Round of gameplay should be about 15 - 20 minutes in length, this is not only so the player doesn't have enough time to get used to the environment but also to add to the arcade-style of game. This arcade-like game flow which is like Pacman or more modern titles like Dark Deception focuses on short bursts of thrills instead of longer, drawn-out gameplay which can get stale fast.</p> <p>Some Goals to meet this pillar are:</p> <ul style="list-style-type: none"> • A game timer to push the player to complete the game as quick as possible for speedrunning/personal bests. <ul style="list-style-type: none"> • An Arcade Game Mode which includes additional pick-ups for a player score. • A smaller, condensed level design & objectives to fit the 15 - 20-minute expected range.

Each of these objectives has been definitively achieved to varying effectiveness. This can be seen through the MoSCoW Analysis which lists the targets which should be hit for the project to be a success.

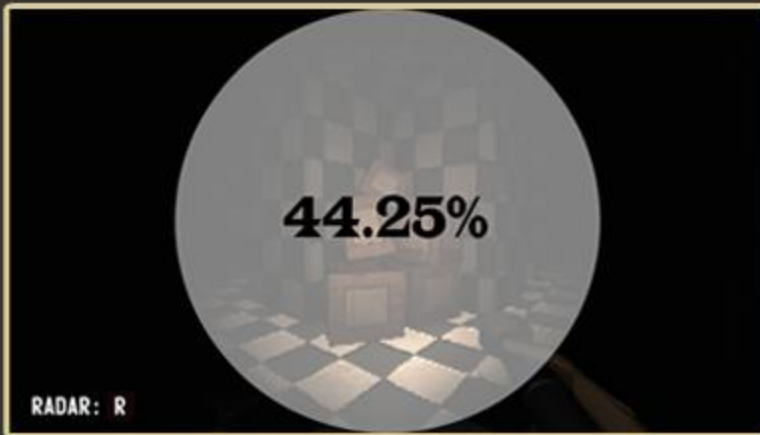
The MoSCoW Diagram extends to all disciplines of games development, the focus for this project was the Game Design of an Emergent Technology, which is why the Technical Design Elements Must be Included while the Art wasn't as Vital to the success of this Project.

We will see how each of these Targets are hit throughout the Document.

Step Name	Completed?
Create a tool which connects a Fitness Band with unreal Engine 5	
Create an Enemy AI which works based on data from the fitness band	
Design a simple maze-like level for the game	
Design a standard chase game mode	
Create a detailed design document for the project	
Design the game to be sufficiently scary and play to horror tropes	
Integrate both controller and keyboard controls	
Model and animate a simple monster model for the enemy	
Create a Technical Design Document to explain Implementation	
Create an opening cutscene to calm the player and read their rest rate	
Implement a difficulty increase as the player advances through the game	
Record a Trailer and Gameplay Video for the game	
Record original sound effects for the game	
Create a point-system for the arcade mode	
Create a complex enemy model & animations	
Add secondary objectives & puzzles into the maze	
Design an Arcade mode which includes additional items	
Online/Local Multiplayer	
Player Character Voice Acting	
World Leaderboard	

W012287L

Research - Horror Design



First-Person Flashlights are a common light source in all first-person horror games, they usually have a cone size that illuminates about half the screen, usually 50 - 60%. The flashlight in It Steals only illuminates 44% of the screen, however instead the light reaches quite far which works very well with the corridor-heavy level design of the game.

Sound Design is one of the best tools for generating immersion in the level. For example, in the project there is a lot of running around the level, meaning footsteps needed to be included. Increasing the reverb of the footsteps sound, it makes the environment around the player seem bigger and more imposing.

Sound Designers have well documented these small tricks in Both Articles and Youtube Tutorials which were vital to this project's sound production



One Common Technique in Lighting horror games is called Tenebrist Lighting. This is a method used in Paintings where there is high contrast between the Light and Dark of an Image. The Dark in these Images is usually Pitch Black Dark with the shadows being a framing device to show the subjects as powerful and mysterious.

This is used in games, especially for games with Stalker Enemies like this project where the game has to make them seem powerful and keep them hidden in the dark shadows & corners.



W012287L

Introduction - Gameplay Research

The control scheme of a horror game needs to be quite basic as Immersion is key to a horror game's success, if the control scheme is awkward and unnatural then it will take the player away from that immersion.

One constant between a lot of the horror games is being able to toggle the flashlight and crouch, while this wasn't included due to the enemy not reacting to light, future development could try to incorporate more sneak gameplay than simply running.

LETHAL COMPANY

Input	Keyboard	Gamepad
Movement	WASD Keys	Left Analogue Stick
Jump	Space	Bottom Face Button
Run Toggle	Left Shift	Left Bumper
Crouch	Left Control	Right Analogue Stick Press
Scan	Right Click	Right Shoulder Button
Look	Mouse Movement (XY)	Right Analogue Stick
Interact	E Key	Right Face Button
Use Item	Left Click	Right Bumper

ZARDY'S MAZE

Input	Keyboard	Gamepad
Movement	WASD Keys	Left Analogue Stick
Run Toggle	Left Shift	Left Analogue Stick Press
Recharge Flashlight	Right Click (Hold)	Left Bumper (Hold)
Look	Mouse Movement (XY)	Right Analogue Stick
Attack	Left Click	Right Bumper
Flashlight Toggle	Right Click (Press)	Left Bumper (Press)
Jump	Spacebar Key	Bottom Face Button
Interact	E Key	Left Face Button

DARK DECEPTION

Input	Keyboard	Gamepad
Movement	WASD Keys	Left Analogue Stick
Run Toggle	Left Shift	Left Analogue Stick Press
Map Toggle	Spacebar Key	Bottom Face Button
Look	Mouse Movement (XY)	Right Analogue Stick
Interact	Left Click	Right Face Button

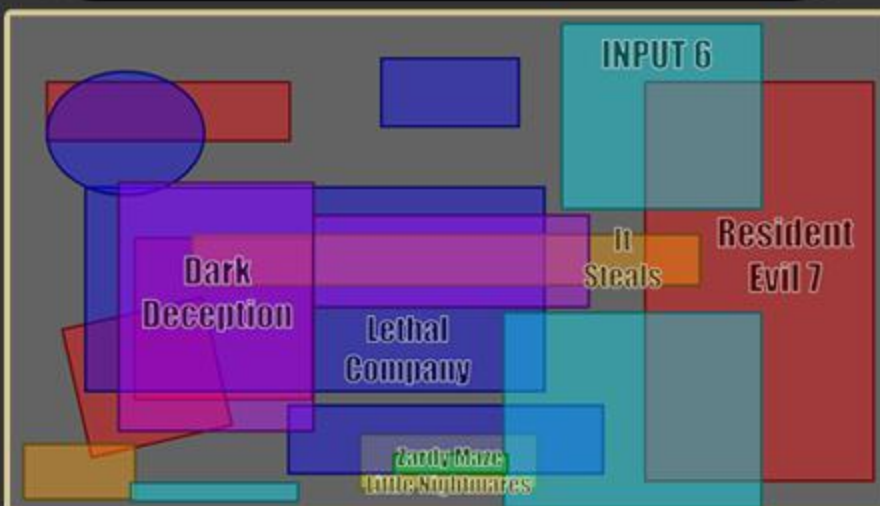
RESIDENT EVIL 7

Input	Keyboard	Gamepad
Movement	WASD Keys	Left Analogue Stick
Run Toggle	Left Shift	Left Analogue Stick Press
Use Heal Item	Control Key	Right Shoulder
Look	Mouse Movement (XY)	Right Analogue Stick
Interact	E Key	Bottom Face Button
Aim Weapon	Right Click (Hold)	Left Bumper
Attack	Left Click	Right Bumper
Crouch	C Key	Right Analogue Stick Press

A HUD Layout Document was produced to analyse how UI is displayed throughout the researched games' HUDs. While all together they take up a lot of the screen, for most of these screens they are hidden unless you press a button or perform an action.

A trend with UI in horror games is that less is more, the less UI on the screen at once is better as it allows the player to become more immersed in the world.

Due to the nature of Horror UI trying to stay as out of the way as possible, most Horror Games make use of Minimalist UI. Minimalism is defined as Eliminating excess and strategically placing the remaining elements.



Play Game

Alien: Isolation

Survivor Mode

Play Main Campaign

W012287L

Development - Heart Rate Implementation 1

ANT+ Was the Original method that would be used to connect the game with Heart Rate, it has been used in other games which make use of Health Technology such as Bring To Light which uses the player's Heart Rate. ANT+ is a system which allows the user to access data which is collected by supported fitness devices.



While Researching the ANT+ Systems to decide whether it was right for this project, I got comfortable with ANT+ Software such as SimuANT and ANTware. These are the connecting software which act as a bridge between the Device and the Computer.

ANTware in particular is key to connecting products to ANT+ Systems as developers can configure the sockets and channels that data are sent through.



After additional Research into the ANT+ systems it became clear that ANT+ was not suitable for this project. While it is the Industry Standard for Fitness Data, it required Industry funds to buy both the Developer Kit as well as the two ANT USBs which were required to work with the SDK.

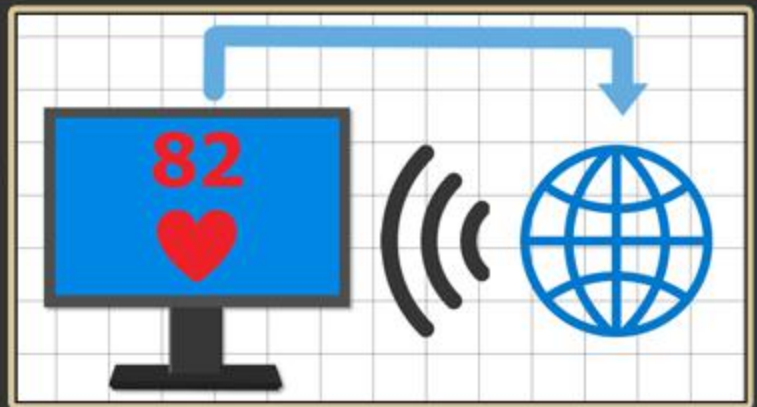


W012287L

Development - Heart Rate Implementation 2

An API, also known as an Application Programming Interface, is defined as "a software intermediary that allows two applications to communicate". What that means for this project is that a server collects the Heart Rate Data and the Data is then taken from the Server by the computer through a web request.

The API that ended up being used in this project is called Pulsoid, its software which is primarily used by livestreamers to show their heart rate on stream. Pulsoid operates through a mobile app that connects to the HR device and uploads the data to a server, which can then be accessed through the API by the client (the game)



```
int main()
{
    std::string result;
    CURL* curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_DEFAULT);
    curl = curl_easy_init();
    struct curl_slist* headers = NULL;
    std::string token = "18dbf30-5081-4d75-96a9-799717437300"; // your actual token
    if (curl) {
        std::string token_header = "Authorization: Bearer " + token; // access token is keyword
        headers = curl_slist_append(headers, token_header.c_str());
        curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);
        curl_easy_setopt(curl, CURLOPT_URL, "https://dev.pulsoid.net/api/v1/token/validate");
        curl_easy_setopt(curl, CURLOPT_URL, "https://dev.pulsoid.net/api/v1/data/heart_rate/latest");
        res = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
    curl_slist_free_all(headers);
    curl_global_cleanup();
    std::cout << result << "\n";
}
```

Microsoft Visual Studio Debug Console

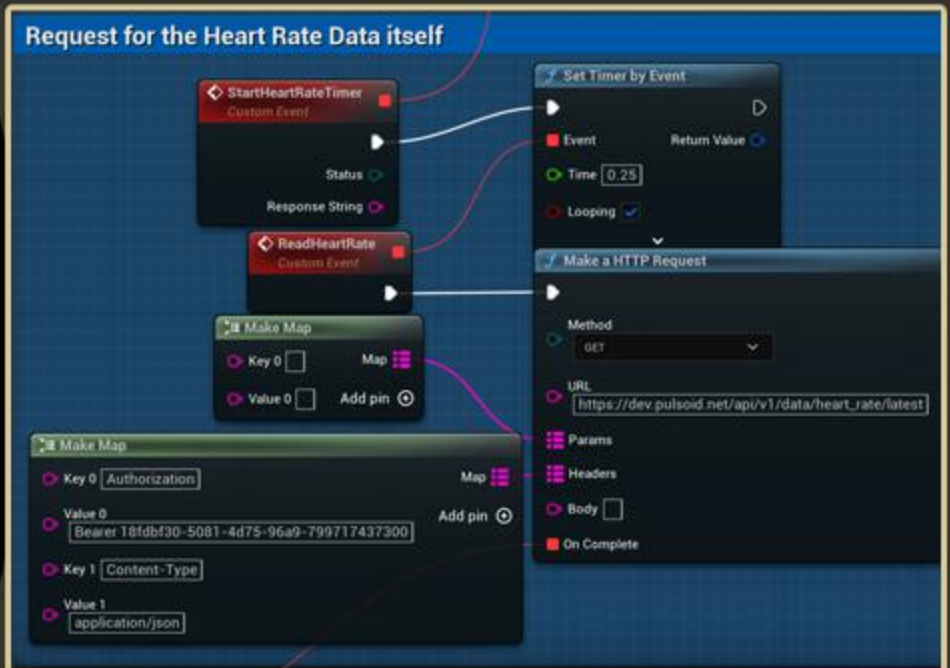
```
{ "measured_at": 170063819505, "data": { "heart_rate": 104 } }
```

D:\level 6\VP\Projects\Uear-HeartRateReadingC++\Uear-HeartRateKil...hPulsoid\win4\Debug\HeartRateWithPulsoid.exe (process 611)

I began developing with Pulsoid by trying to create my own Unreal Plugin, this was using Curl which is another software that allows for HTTP requests in Visual Studio.

However, after searching, the Plugin aptly named "HTTP for Unreal" was bought and implemented to bridge the gap between the Heart Rate Monitor and Unreal Engine.

Due to Pulsoid being developed for streaming, it doesn't require any maintenance after heart rate device is put on, so the player can fully focus on the game.



W012287L

Development - Gameplay & Managers

This game is managed through the Game Mode and the Two Game Managers, these being the Heart Rate Manager & The Gameplay Manager. The Game Mode is used to allow actors to communicate with these managers, such as when an actor requires any Heart Rate Variable, they use Interface Events to signal to the Game Mode to send them that data.



The Heart Rate Manager encapsulates all the Heart Rate reading functionality into a single actor, including the HTTP Requests. It gets the Resting Heart Rate from the Main Menu & calculates the Highest Heart Rate by adding the difference between the Resting Heart Rate and the Standard Heart Rate (80) by 100.

Calculate HR Upper Bound



The Gameplay Manager's purpose is to manage the objectives & Game States, this includes checking whether the End Game sequence should start with the Hatch Spawning and the Enemy becoming even more Aggressive. The Gameplay Manager is communicated with more directly than the Game Mode with each Objective having a reference to the level's Gameplay Manager.

Collected Objective

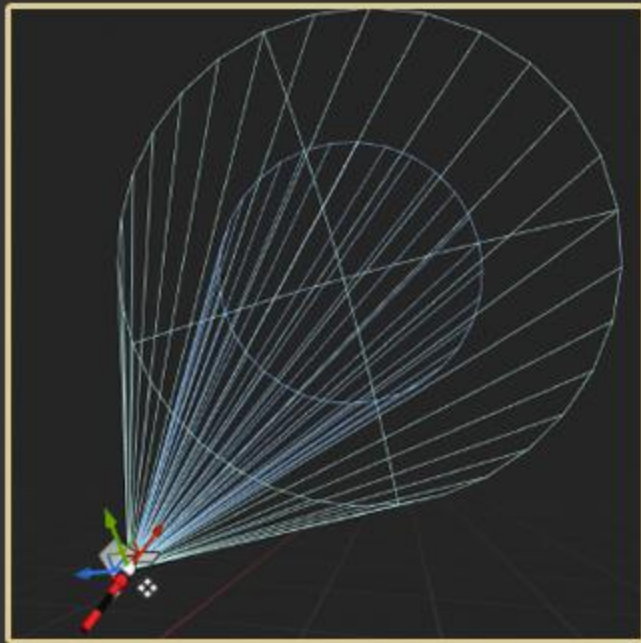
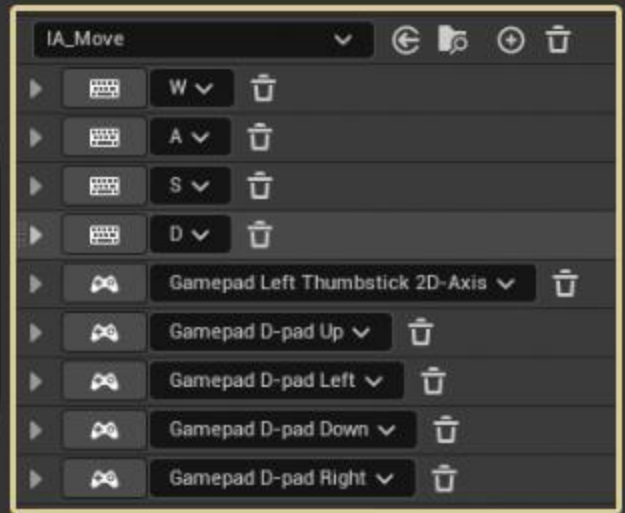


W012287L

Development - Player Control

One of the Should Have points in the MoSCoW Analysis was to have the game work with both Keyboard & Gamepad. This was quite important as a lot of the games used as references took the time to implement both input methods.

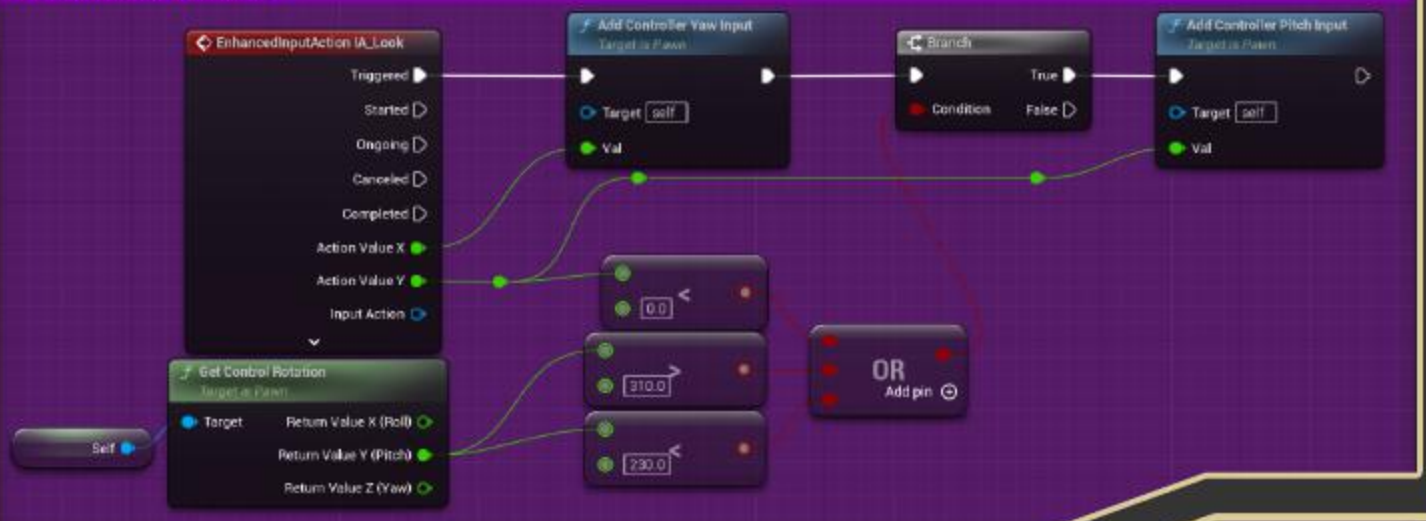
Unreal's Enhanced Input System made this quite easy to implement with the Input Actions being mapped to both Keyboard & Gamepad Input Methods.



As stated in the Research Section of this project, the Flashlight is an especially important part of any First-Person Horror Game. To ensure that the player's flashlight wouldn't reveal everything, a Lightmap Texture was used to dim the light and give the light's rays a realistic look which added to the immersion of the game. In addition, the player's flashlight had slight lag on the camera so horror reveals would have more build-up.

Small Tweaks were made to the standard player controls to make the game more realistic and have better game feel, one of these details was clamping the Y rotation to ensure the player cannot look directly up or down.

Look Rotation Control



W012287L

Development - Level Design



Synty's Sci-Fi Pack was chosen for this game not only due to it fitting the theme of the level & game very well. It was also chosen as there would be a lot of Maze Walls in the Level, if the maze walls weren't optimised this could really impact performance, but as the assets were low-poly they could be used without affecting the pace of the game.



The Models themselves were not the only visual elements implemented into the world, as the low-poly artstyle was great for optimisation but on its own, it is too simple to effectively generate fear. Because of this, Decals were created to put on the walls, their opacity being controlled by the player's Heart Rate. A Post-Process Volume was also implemented to give the level a fog-filled grainy appearance



W012287L

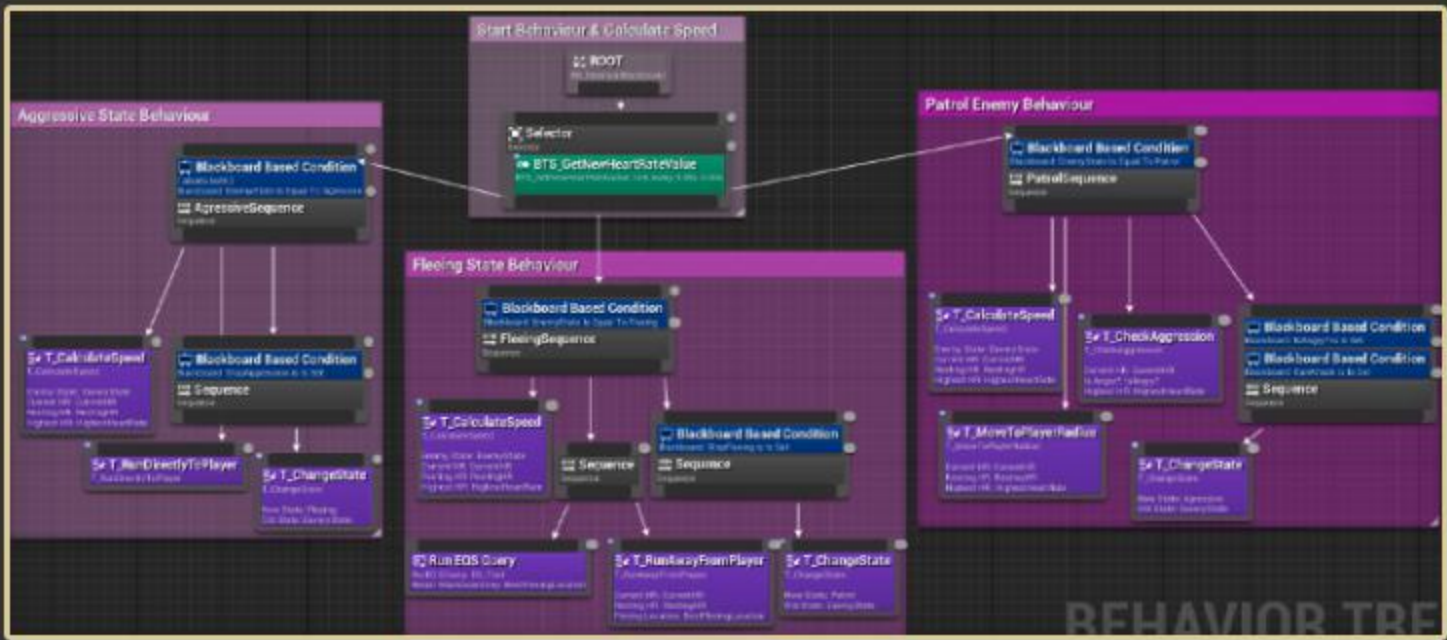
Development - AI Behaviour 1



When designing the AI for the Stalker Enemy, it was clear that a State Machine was suitable for the enemy as the behaviour of the enemy will be decided through the Heart Rate value of the player.

The enemy has three states, the first is the Patrol State where the enemy will move to an area around the player, with the radius he can move to being decided by the player's Heart Rate.

Then there is the aggressive state where the enemy will directly chase the player. the final state is Fleeing where the enemy picks a location in the opposite direction to the player and moves to it, before returning to the Patrol.



Unreal doesn't have Direct Support for State Machines, instead Unreal uses a Behaviour Tree Model which can be adapted into a State Tree using Conditions. This Behaviour Tree runs a "Service" before selecting a state, with this service getting the Player's Heart Rate and checking it against the Maximum Heart Rate.

W012287L

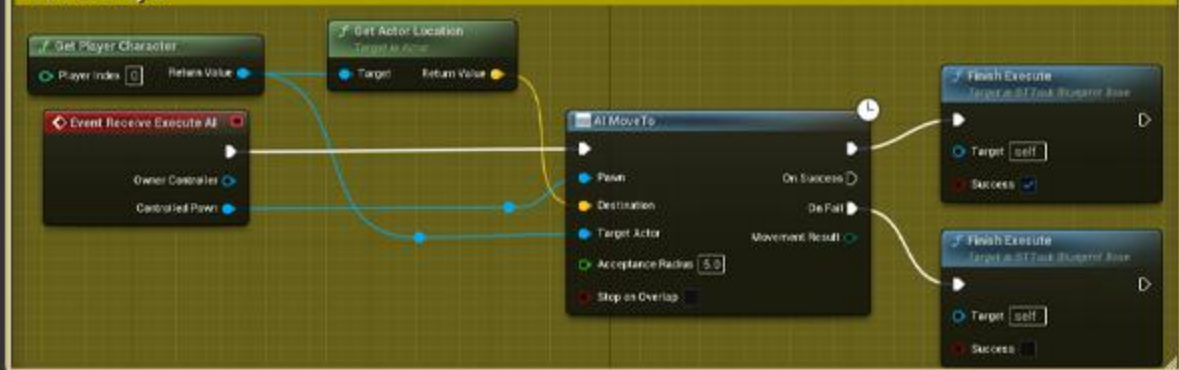
Development - AI Behaviour 2

The Main Functionality of the Enemy is focused on its Movement, as the Behaviour in what he moves towards is based on multiple factors. For example in the Move Around Player Task in the Patrol State, the Range that the random location is decided from is based on the Heart Rate of the player, and the Speed is also decided through a separate event previous to that one.

Move Around the Player in chosen radius



Run To Player



The Fleeing Movement is the most complex as it uses an Environment Query System. This generates a Grid of Locations and scores them on two criteria. The further they are away from the player, the higher the score, and the locations in sight of the enemy are disabled. The query then picks a location with a high scored and the AI moves to that location.

W012287L

Development - User Interface

In the effort to support both Keyboard & Controller, Common UI was implemented into this project. Common UI is an overhaul to the UMG Unreal System which allows for Controller Support, UI Stacks & Queues, and Style Blueprints which allow widgets to inherit from the same style and all have the same appearance. It was a challenge to get used to but was definitely worth it.



Common UI Works with Widget Stacks, these are a FIFO Structure where Widgets are pushed onto the stack and the widgets already on the stack are deactivated until the top widget is pulled from the Stack.

This allows for Seamless Menu Transitions, for example the Main Menu screen is pushed first, then either the Instructional Video or HR Reading Screen are pushed on top, transitioning the player into the game

One of the Deliverables that was outlined in both the Proposal Document & MoSCoW Diagram was to create an instructional video which would instruct the player on how to equip and start the Heart Rate Reader. It both explains how the Heart Rate Reading works and the game's objectives.



W012287L

Development - Sound Design

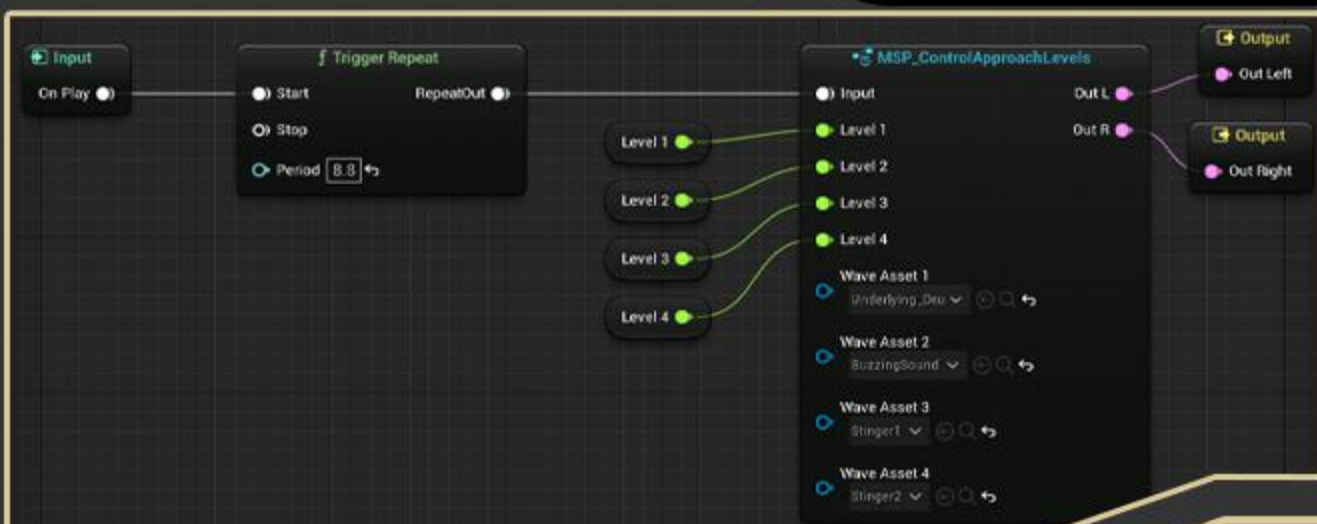
The Audio of this project posed a unique challenge as I wanted the Tempo of the soundtrack and atmospheric sounds to match with the player's Heart Rate to subconsciously raise the player's HR.

To do this, Unreal's Metasounds tools were used to construct sounds with the Trigger Repeat node, this node allowed for BPM values to be directly plugged into the repeater so sound effects would repeat at the same frequency as the player's Heart Rate. Metasounds was also vital for the other interactive sounds that were implemented in the game.



Metasounds allows for Inputs & Parameters in the sounds so Blueprints can control aspects of a Metasound. An example of this is the Sound which plays as the Enemy Approaches, this soundtrack is built off four tracks with four Floats altering the volumes.

These Floats are then Altered by the Player Character based on the distance between the player and the Enemy.



W012287L

Evaluation - Testing Results

1. Please input a random 6 Digit number as your Unique Identifier

[More Details](#)

7
Responses

Latest Responses

"573936"

"221201"

"123456"

The Testing of this project was performed in a very professional manner due to the Health Data involved. Each test was done on-site in a quiet room with the testers reading the Information Sheet & Signing the Consent Form.

4. Was it apparent that the game was affected by your heart rate?

[More Details](#)

● Yes 3
● No 4



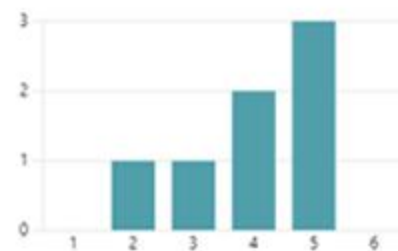
Unfortunately, more than Half of the testers seemed to Forget that Heart Rate was a core component of the gameplay.

Although from my follow up Qualitative question, testers made it clear they knew the Enemy & Camera were effected by Heart Rate, but they didn't effect the Gameplay enough.

2. On a Scale of 1 to 6, how effective did you find the Horror in this Game? (How scared were you?)

[More Details](#)

4.00
Average Rating



On the Other Hand, all the Testers had positive scored for the Horror & Gameplay Elements, with the Main Enemy & Sound Design being referenced often as successfully building the tension.

This was great to hear as a lot of the game's development focused on these Design Elements.

In other games which use Emergent Technologies, such as Before Your Eyes which uses the Webcam, the Emergent Tech is a core part of the experience which can't be ignored. Further development of this project should be focused on making the game more noticably reliant on HR



W012287L

Evaluation - Original Intentions

This project has been successfully crafted to hit all three design pillars that were set at the very start, with the Horror Elements in particular being the most successful and resonating with the testers. However, the original intention of this project was to craft a game which adapted to your heart rate, and while the implementation is there, it seems that the execution of these mechanics could have been a more vital part of the experience so players absolutely knew this game was controlled through Heart Rate.

5. What Elements of the Game stood out to you most as being effected by Heart Rate?

7 Responses

ID ↑	Name	Responses
1	anonymous	the FOV
2	anonymous	The sound Getting louder as the Enemy Gets Closer to me
3	anonymous	The screen change, the speed of the player
4	anonymous	Speed and Sound
5	anonymous	nothing
6	anonymous	Enemy
7	anonymous	the enemy chasing you



The Framework itself has been ironed clear of any bugs that were found during development, with the game being quite self-contained and easy to pick-up and play as stated in the Arcade-Style Gameplay Design Pillar.

I believe that as the project evolved it skewed away from arcade mechanics like a point system but this could still have been achieved in other ways. One example is the game Zardy's Maze, another maze-style chase horror game which allows the player to change the difficulty score of the enemies, this could have been achieved through altering the speed of the enemy based on the rating.



W012287L

CLASSIFIED

- 1a - *Nevermind*. (2015). PC. Flying Mollusk. Steam
- 1b - *Alien: Isolation*. (2014). PC. Creative Assembly. Steam
- 1c - *It Steals*. (2020). PC. Zeekers. Steam
- 2a - McGee, M. (2020) *How to design sounds for Horror games*, YouTube. (Accessed: 01 March 2024).
- 2b - Coster, A.D. (2016) *'A Man Singing By Candlelight'* Artsynet. (Accessed: 01 March 2022).
- 3a - Garmin. (2004) *The Wireless Sensor Network Solution - This is Ant*, (Accessed: 01 March 2024).
- 3b - ANT+. (2013) *ANTware*. Available at <https://www.thisisant.com/resources/antware-ii/> (Accessed: 01 March 2024).
- 4a - *Geometry Dash*. (2014). PC/Mobile. Rob Top Games. Steam/App Stores
- 4b - Stenberg, D. (1998) *C:url*. Available at: <https://curl.se/> (Accessed: 01 March 2024).
- 5a - SYNTY. (2014) *Polygon: Sci-Fi City Pack Game Assets, Low-Poly Models Store*. (Accessed: 01 March 2024).
- 6a - Unreal Engine, (2020) *AI with Behaviour Trees*. (Accessed: 01 March 2024).
- 7a - Unreal Engine, (2022) *Introduction to common UI | Inside Unreal*. (Accessed: 01 March 2024).
- 8a - *Before Your Eyes*, (2021), PC, Goodbye World Games, Steam
- 9a - *Zardy's Maze*. (2020), PC. Swanky Box. Steam
- 9b - *Ultimate Custom Night*, (2018), PC, Scott Cawthon, Steam