Stress Management through Biometric Gaming

Gareth Kennedy

BSc. Software Design 2016

Athlone Institute of Technology

Declaration

I hereby certify that the material, which is submitted in this report towards the award of BSc. Software Design, is entirely my own work and has not been submitted for any academic assessment other than part fulfilment of the above named award.

Future students may use the material contained in this report provided that the source is acknowledged in full.

Signed…….

Date……

Abstract

Biometric gaming is a new advancement in gaming that attempts to improve the immersion in gaming. Along with the recent push in virtual reality (VR) technologies, biometric technologies are quickly being implemented alongside VR in an increased effort to immerse the player more thoroughly in the game world.

Due to the sudden popularity with biometric gaming, developing an application that could be used; not only for recreational purposes, but also medical purposes could show the benefits of biometric gaming as a viable game mechanic. This simulation will hopefully showcase biometrics as more than a novelty gimmick and will help pave the way for the use of biometrics in future games or for use in medical rehabilitation.

By creating a simulation that has been built around the use of biometrics as a game mechanic; the user will hopefully experience a fun and immersive game that can, over time, help the user manage their stress levels.

The results gained from this project should help prove the viability of biometric gaming in both the medical and recreational fields.

From the obtained results, a better understanding of biometric gaming can be obtained. It is hoped that readers of this thesis will see the possible pitfalls of implementing biometrics into a game and can avoid them in good time.

Acknowledgements

Firstly, I would like to thank my supervisors Dr. David Scott and Mr. Karol Fitzgerald. They were constantly at hand to help in the production of this thesis and proved invaluable in their knowledge. Their doors were always open if I had any questions or problems and no problem was too big or small.

Secondly, I would like to thank my fellow classmates. I have learnt so much from you all and have made some lifelong friends along the way. You have earned my upmost respect due to your extremely strong work ethic and down to earth attitude.

I would also like to thank Alan and Calum for “top quality banter” throughout my time in Athlone Institute of Technology.

Lastly, I would like to thank my family for their continued support throughout my time studying. I would not be here without you.

Thank you all.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Problem Statement

Stress is extremely prevalent in today's society and can lead to an abundance of health problems. According to the Irish Congress of Trade Unions, "90% of voluntary sector workers consider their job to be stressful" [1]. Games are often used as an escape in order to help one forget about their everyday stresses. A game where stress is implemented as a gameplay mechanic would allow the user to develop tools to help manage their stress levels.

Immersion is a vital part to a successful game. Generally, the more input a user has to a game, the more invested and immersed they are. Games rarely take feedback from the user; Games are supposed to react to what the user does. Considering this, it would be worthwhile to create a game that reacts to some biometric reading from the player in an attempt to deepen the immersion.

1.2 Proposed Solution

I propose a game that helps players develop the tools they need to help manage their stress levels in day to day life. The idea is to introduce players to a game environment where the game is significantly easier if the player can keep calm. Stress would be detected using some form of biometric detection device. The player would have to control their stress levels and keep calm in order to get through the game.

1.3 Possible Contributions from Work

The intention of this project is to help a user control their stress levels through biometric gaming.

This project could be used in the medical field in order to help patients learn to recognise when they are stressed and help them develop tools to control these stress levels. It could also be sold as a commercial game due to its interesting mechanics, providing a highly immersive gaming experience to the user.

1.4 Thesis Outline

Chapter one of this thesis will give a brisk overview of the problem, possible avenues that were pursued in solving the problem and a general outline of the thesis as a whole.

Chapter two of this thesis will focus on the background research that was undertaken for this project. It will look at topics such as target demographics, different methods of implementing AI, biometrics as a game mechanic and different types of suitable biometrics for gaming.

Chapter three will showcase the system design for the simulation. An in depth walkthrough will be given for the project, detailing all aspects and providing visual examples for each.

Chapter four will show the tests that were performed for the project, notably the feedback from user acceptance testing and the results of requirements testing in order to determine if biometric gaming is viable for monetisation.

Chapter five will discuss the results of the project in detail and will give recommendations for future iterations of the project.

1.5 Research Aims and Objectives

The main objective of this project is to see if the use of biometrics is viable as a game mechanic.

Secondary objectives this project hopes to obtain are to see if biometrics help deepen a user's immersion in the game world. It is also hoped that this project will help show that biometric gaming could be used in the medical sector for medical rehabilitation purposes.

Chapter 2: Background Research

2.1 Introduction

This project integrates many different components in order to immerse the user. This section will outline the different types of research that was undertaken for this project. The use of biometrics as a gameplay factor will be investigated and the suitability of different types of biometrics. Other games that use biometrics will be investigated on how and why they are used based on their genre. AI (artificial intelligence) will be thoroughly discussed from its history to different implementations in today's games. Lastly, different software programs will be analysed for suitability with the project.

2.2 Gaming

The first "video game" was created by Willy Higinbotham at Brookhaven National Library in 1958 [2]. Known as "Tennis for Two", it was the first insight into what video games could become. Around 1966, engineer Ralph Baer explores the idea of creating interactive games the work with a television. A year later, Baer successfully creates two interactive television (TV) games. Baer's TV game is licensed by the company Mangavox who, in 1972, release "Odyssey", the first home video game system [3]. After these keystones, gaming begins to quickly advance with companies like Atari, Namco, Nintendo and Sega quickly developing their own consoles and games. Today, gaming is considered its own media and, from a 2009 survey done by the Entertainment Software Rating Board (ESRB), the gaming industry amasses more than ten and a half billion dollars in revenue [4].

2.3 Demographic

This project is aimed to target the highest majority of gamers. According to a 2015 survey published by the Entertainment Software Association (ESA), the average age of someone who plays games is thirty five years old [5]. This game would cater to that age group as they would have an interest in their health and wellbeing at this age than a younger demographic. The concept of biometrics in gaming is also a unique and rarely seen idea that would interest a large variety of gamers.

2.4 Biometrics

Biometrics, by definition, is the application of statistical analysis to biological data. The term biometrics is derived from the Greek words "bio" meaning life and "metric" meaning to measure. Biometric technology is often used in the security sector for

identification purposes [6]. Fingerprint or retina scanners are common examples of biometric devices used in this sector.

2.4.1 Types of Biometrics Suitable for Stress Analysis

2.4.1.1 Respiratory Rate

Respiratory rate could be used as a biometric for stress analysis. Respiration rates generally increase when the body is being affected by stress factors. Breathing rates often increase when a person is panicked or put in an undesirable situation.

2.4.1.2 Vocal Analysis

Vocal analysis is also a viable choice for a stress analysis biometric. The pitch of a person's voice generally rises when under stress. Lie detectors are a form of vocal stress analysis.

2.4.1.3 Heart Rate

Heart rate is an extremely good candidate for stress analysis. The correlation between heart rate and stress levels is well documented and research involving heart rate often involve "acute laboratory stressors as opposed to chronic stress" [7].

2.5 Stress in Gaming

Stress is seen as your body's reaction to some sort of duress. Hans Selye refers to stress as "the non-specific response of the body to any demand for change" [8]. Gaming, like most other hobbies, are performed to help one forget about their problems, lowering their everyday stress levels. However, due to the immersive and competitive nature of games, the user may become frustrated and even more stressed from playing a game. For example, a study showed a 'teenager had repeatedly bitten his tongue due to stress experienced during gaming" [9]. Immersion is a huge factor in stress in gaming. If a user doesn't feel challenged or doesn't care about the outcomes of the game, they won't stress over failure or risk. There has been a lot of research into stress related gaming and its pros and cons. For example Hancock stated that "supplementing context-relevant stress in game training shows promise for enhancing individuals' motivation to succeed" [10, p.1].

2.5.1 Monitoring Stress

Stress has many signs and symptoms that all wildly vary in severity and ease of detection. Biometrics are often used to help assess ones level of stress usually due

to a direct correlation between the biometric and stress itself. Stress is believed to come in three different forms: acute, episodic and chronic [11].

2.5.1.1 Acute

Acute stress is stress caused by a short-term stress factor. Acute stress is the primal "fight or flight" response to immediate stress factors. Acute stress generally isn't considered harmful.

2.5.1.2 Episodic

Episodic stress is when acute stress occurs periodically in ones day to day life. Unlike acute stress, episodic stress can be harmful to one's mental health.

2.5.1.3 Chronic

Chronic stress is stress called by long-term stress factors that can be very harmful in the long run.

2.6 Artificial Intelligence

AI (Artificial Intelligence) is a considerably large part of the project. If the game was to become predictable and exploitable, the user would adapt and would likely feel little to no stress, therefore ignoring the whole stress management idea that the project is aimed at. AI would need to feel realistic and organic in order for the user to feel immersed in the game world. The user should fear getting spotted by the AI and actively try not to. Good AI is also mandatory for an immersive gaming experience. "The Artificial Intelligence in a game is perhaps one of the most influential ingredients for enabling a game player to suspend disbelief long enough to become properly immersed into the gameplay" [12].

2.6.1 History of AI

The history of AI can be said to begin in 1950, when British mathematician Alan Turing wrote a paper known as "Computing Machinery and Intelligence". Turing simply asked the question, "Can machines think?" Upon this question, the Turing test was built. The Turing test involved two humans and one computer. A is a computer that attempts to answer questions given to it by player C in a human-like fashion. Player B, the human, also answers questions given to him/her by player C. Basically, player C, the interrogator, is trying to figure out which player, A (the machine) or B (the human) is the human going by the answers he/she receives. The answers are

given to C through a screen and keyboard so that the machine wouldn't have to render words as audio.

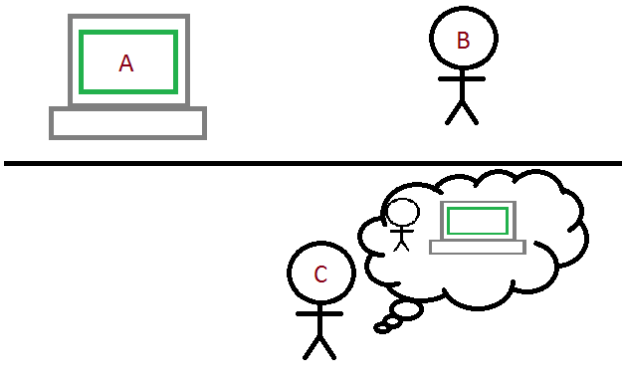

Figure 2.6.1: Turing test. Image adapted from Saygin, 2000 [13].

Upon completion of his research, Turing shifted his question from "Can machines think?" to a more meaningful "Are there imaginable digital computers which do well in the imitation game"? Turing's question become an important part of artificial intelligence's philosophy and is seen as the foundation on which AI has been built [13].

2.6.2 Neural *Networks*

An Artificial Neural Network (ANN) is a mathematical model based on the structure of biological neural networks. ANNs consist of many artificial neurons, each of which contains a mathematical function. Artificial neurons themselves are split into three layers. The first "entrance" layer takes in inputs and "weights" them, meaning that each input is multiplied by a weight. The second layer adds all weighted inputs and bias of the neuron together. In the last layer, the weighted inputs and bias is passed through a transfer function (also known as an activation function) [14].

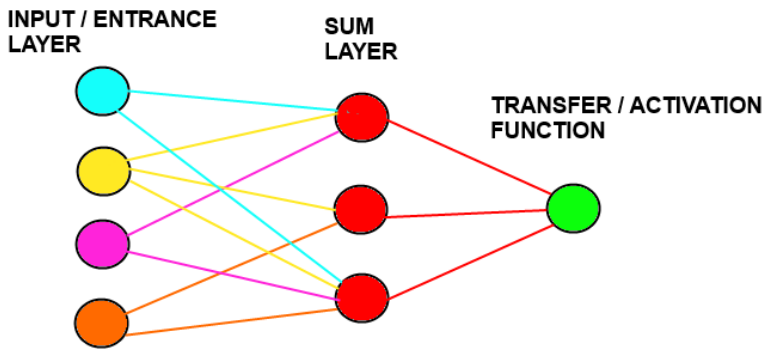

Figure 2.6.2: Neural network example

2.6.3 Behaviour *Trees*

Behaviour trees are a graphical representation of a particular object's AI. They are used to control behaviour of characters in a game environment. The tree starts at the parent/root node and it evaluates its children. These children are ordered based on their priority and are set to running if their conditions are met. When a child node is set to running, it returns the behaviour and the behaviour is played out in-game. The tree then starts back at the parent node and begins to evaluate the children again. Behaviour trees first became popular due to games like Halo 2 utilising them. Halo 2 had an average of sixty different behaviours arranged in four layers and was widely praised for its well implemented AI [15].

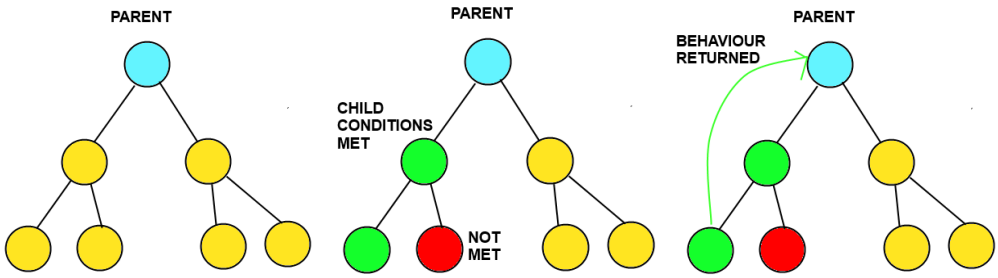

Figure 2.6.3: Behaviour tree walkthrough example

2.7 Possible AI Software Solutions

The Unity game engine doesn't contain its own tools for the creation of behaviour trees or ANN. Because of this, third party software will need to be used. Below are a few examples of third party AI software that work efficiently in the Unity game engine.

2.7.1 RAIN AI

RAIN is a complete AI engine that is widely used in many of today's game studios. It provides a full suite of tools that make AI creation and control extremely easy with no license fees or other costs.

Figure 2.7.1: RAIN AI logo

2.7.2 Behave 2

Behave 2 is a drag-drop AI plugin for Unity. It offers a clean interface and many features for enhanced modification of behaviour trees. Sadly, it does have a hefty price tag in order to use it for your own projects.

2.8 Suitable Genres for Stress Related Gaming

Games have many genres which are suitable candidates for stress related gaming. I believe the best genre would be one that builds up slowly, allowing the user to learn the ropes but can then quickly become stressful in order to test the user. If the user can remain calm they will quickly reap the benefits.

2.8.1 Stealth

Stealth games would work extremely well with stress related gaming. Acute stress factors are extremely prominent in the stealth genre and the players fight or flight response will often be put in use. One would have to keep calm in order to remain undetected. The area and patrols would be observed by the player before making any actions. Upon getting caught, the player would have to react quickly and calmly in order to get away. The ability to remain calm upon being detected is often what makes players good at stealth games. The recovery of the situation (being caught) is arguably more important than remaining undetected throughout. The user will generally always feel pressured to not get caught thus providing a baseline level of stress to work off.

2.8.2 Fighting

There are few games that can rival the increase in adrenaline and stress that fighting games provide. Fighting games are often extremely competitive and matches are

generally fought down to the wire. Players who can remain calm on a miniscule of health will often come out on the winning side due to the opponent getting greedy and slipping up.

2.8.3 Horror

Horror games would work well in terms of reading a user's stress levels. The constant sense of foreboding would fall under both the acute and episodic stress factors, forcing the player to adapt quickly. This genre has had success in the biometrics field with games like Nevermind appearing. Nevermind is a horror game that becomes scarier as the user's heart rate continues to rise. Nevermind has received many awards for its innovative use of biometric such as IndieArcade 2016 and Games for Change 2014 [16].

Figure 2.8.3: Nevermind game logo

Chapter 3: System Design

3.1 Introduction

The aim of this project is to help prevent symptoms caused by chronic stress factors by giving the user tools to manage acute and to a lesser extent, episodic stress factors. This project will use an immersive simulation based on the stealth game genre in order to test if the user is prone to being affected by acute stress factors.

The requirements, dependencies and assumptions of this project will be stated. The architecture and the interaction between devices will be showcased.

This section will outline the steps taken to design and implement a simulation of a stealth game that modifies its difficulty based on biometric readings taken from the user.

3.2 Requirements

3.2.1 Assumptions and Dependencies

Assumption 1	The simulation will be developed for Unity version 4.1.3's game environment.
Assumption 2	The user is wearing a Bluetooth compatible heart rate monitor.
Assumption 3	The Bluetooth dongle is connected to the device that is running the simulation.
Assumption 4	The user is able to set up the Bluetooth connection between the Bluetooth dongle and the heart rate monitor.
Dependency 1	The versions of Bluetooth on the heart rate monitor and the dongle are compatible.

Table 3.2.1: List of assumptions and dependencies.

3.2.2 Functional Requirements

This section will cover the functional requirements for this project.

The format for functional requirements is as follows:

Requirement Number	A unique number used to identify a requirement.
Description	Provides a short description of the requirement.
Rationale	An explanation as to why the requirement has been included.
Dependencies	Identify other requirements (by requirement number) that this requirement relies on.

Table 3.2.2: Layout for requirements.

The categories for the requirements are as follows:

1000	Player Controlled Character.
2000	Heart Rate Biometric.
3000	Enemy AI.

Table 3.2.3: Categories into which requirements will fall under.

Requirement Number	1001
Description	The player character will be able to move by some sort of user input.
Rationale	Allows player to control their character in order to play the game.
Dependencies	

Requirement Number	1002
Description	The player will be able to control the scene camera using some sort of user input.
Rationale	Allows player to control the camera so they can look around corners or over walls.
Dependencies	

Requirement Number	1003
Description	The player will be able to enter a crouching state in order to decrease the risk of detection.
Rationale	Allows player to sneak past enemies.
Dependencies	1001, 1004

Requirement Number	1004
Description	The player will emit sounds upon movement.
Rationale	Allows enemies to detect a moving player through audio cues.
Dependencies	1001, 1003

Requirement Number	1005
Description	The player will be able to interact with in-game objects.
Rationale	Allows player to affect enemy patrols by creating distractions.
Dependencies	

Requirement Number	1006
Description	Entering a crouching state will decrease the amount of sound the player character emits.
Rationale	Basic stealth game functionality. Make it harder for enemies to hear you.
Dependencies	1004

Requirement Number	2001
Description	The game will be able to read in a value for some form of biometric.
Rationale	This will allow the game to change its difficulty.
Dependencies	

Requirement Number	2002
Description	The player's base heart rate will be taken before the simulation begins.
Rationale	This tailors the difficulty to the individual due to heart rate average varying among individuals.
Dependencies	

Requirement Number	2003
Description	The difficulty will increase or decrease depending on the increase or decrease in heart rate.
Rationale	Difficulty can be changed depending on the user's stress levels which are measured through heart rate.
Dependencies	

Requirement Number	3001
Description	The enemies will patrol an area.
Rationale	Gives player opportunities to get past enemies.
Dependencies	

Requirement Number	3002
Description	The enemies will be able to detect the player by line of sight.
Rationale	Basic stealth game functionality.
Dependencies	

Requirement Number	3003
Description	The enemies will chase the player upon seeing the player.
Rationale	Basic stealth game functionality.
Dependencies	3002

Requirement Number	3004
Description	The enemies will be able to detect the player through sound the player character makes.
Rationale	Basic stealth game functionality.
Dependencies	1004

Requirement Number	3005
Description	The enemies will turn to the source of noise emitted by the player.
Rationale	The enemy will investigate the area from where the sound was emitted in an attempt to find the player.
Dependencies	3003

Requirement Number	3006
Description	The enemies will be able to interact with their environment.
Rationale	Adds personality and realism to the enemies.
Dependencies	

Requirement Number	3007
Description	The enemies will return to their patrolling state upon failing to find the player after investigating an area.
Rationale	Ensures enemies will not investigate an area indefinitely.
Dependencies	3001, 3002, 3003, 3004

Requirement Number	3008
Description	The enemies will switch from a chase state to an investigate state upon losing site of the player.
Rationale	It would be unrealistic for an enemy to know where a player is if the player moves out of site.
Dependencies	3003, 3005

Requirement Number	3009
Description	The enemies will return to patrolling or investigating upon failing to catch the player whilst chasing them.
Rationale	Adds personality and realism to the enemies.
Dependencies	3008

3.3 Architecture

This section will provide an overview of the architecture of the simulation. How the heart rate monitor and the simulation will interact will be shown and discussed in detail.

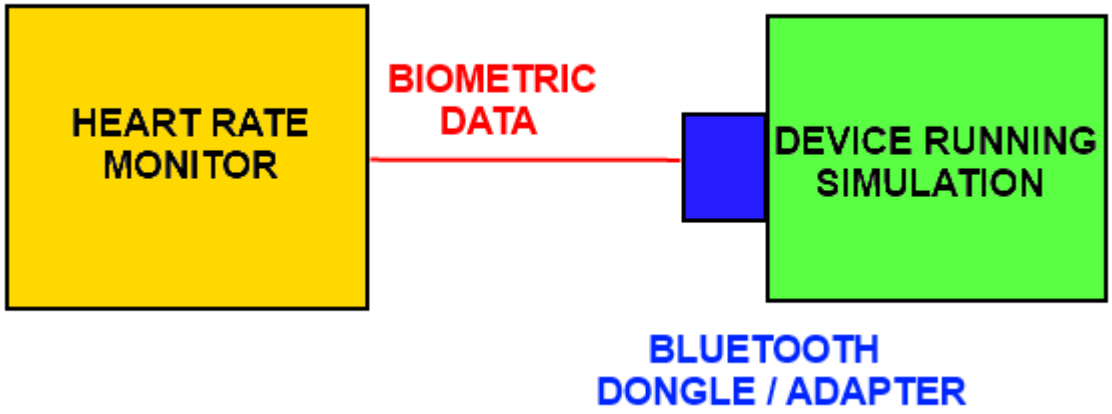

Figure 3.3.1: High-level architecture of the project

The architecture for the project is clean and simple. Using a Bluetooth connection between the Bluetooth compatible heart rate monitor and the Bluetooth enabled device, the biometric data is sent to the simulation so it can use it to adjust the difficulty of the AI.

3.4 Design

3.4.1 Introduction

This section will outline the design decisions that I made concerning many aspects of the project and why I made these decisions.

3.4.2 Biometric of Choice

For biometrics, I decided to go with heart rate. Heart rate monitors would be the most widely available and cheapest biometric devices to acquire. Heart rate is shown to be in direct correlation with stress levels when it comes to acute stress factors.

Regarding respiration rate; upon further research, it was deemed that “there was no significant correlation between chronic hassles frequency and baseline respiration rate” [7]. Respiration rate would only be suitable if used in tandem with heart rate to provide a more thorough data set. It is also important to note that controlled

breathing is considered a very good tool to manage stress levels and respiration rate could be tied into this [17].

As for voice stress analysis; I believe it wouldn't work in a commercial setting. Voice stress analysis would require the player to be asked questions which they would vocally answer whilst having their voice being recorded. The questions would need to be asked whilst the user is playing the game. This would cause distress in the user anyways as it would be difficult to focus on playing the game and answering questions at the same time. Voice stress analysis just seems too clunky to work efficiently due to background noises being picked up by the device.

3.4.3 Genre of Choice

In terms of genre, I decided to go for stealth. I believe stealth games to be the genre of choice in a game to do with stress management. Stealth games, when first played offer the best learning curve compared to other genres. For example, players could practice controlled breathing after a particularly stressful enemy encounter in order to prepare for the next section of the game. Fighting games can aggravate new users due to information overload and a high skill ceiling while horror games would lose their 'fear factor' over time. If the user was coming back to play the game multiple times, horror games would lose their element of surprise and therefore wouldn't stress the user at all. In horror games, it is also difficult for the user to help adapt to the stress factors. In stealth games, the dangers would be telegraphed beforehand and you would avoid them to the best of your ability. Horror games don't offer this luxury thus making it hard for the user to adapt.

So overall, in terms of ease of control and longevity; I believe the stealth genre would allow the user to adapt to the game and give better, more realistic results over time.

3.4.4 AI

Regarding AI, I had a choice between the use of ANN or behaviour trees. I decided to use behaviour trees due to the reasons outlined below.

ANN, while an extremely popular implementation of AI today, could be considered overkill for this project. ANN would require its own memory source in order to efficiently run as they are very computationally expensive. They would learn from frequent use and become more efficient over time. While this would be an interesting addition to the project; it isn't needed to achieve the goals of this project.

Behaviour trees are the most widely used AI solution for the current generation of

videogames. They offer a clear, concise, visual representation of a character's AI. While they don't learn and adapt like neural networks, they are far less computationally expensive. Realistically, behaviour trees offer the functionality I need at a fraction of the computational cost.

3.4.5 Choice of Third Party AI Solution

For the third party AI solution, RAIN AI seemed to be the optimal choice. It offered all the tools needed for the project with a clean, user-friendly interface. Due to a limited budget, this free to use software was extremely useful and got the desired results in good time after some research into the basics.

3.5 Implementation

3.5.1 Introduction

This section will give a detailed breakdown of the simulation and how it works. AI, character movement and interaction with the biometric along with other game mechanics will be explained in detail.

3.5.2 Character Movement

Character movement is done through the use of blend trees in Unity. Blend trees allow one to mix two or more animations together, often allowing a smoother transition from one animation state to the other by "filling in" the gaps between the two. "Blend Trees are used for allowing multiple animations to be blended smoothly by incorporating parts of them all to varying degrees" [18]. For the simulation, using parameters along with a simple C# script, the character can blend between an idle and running animation in order to give a smooth and realistic sprint animation.

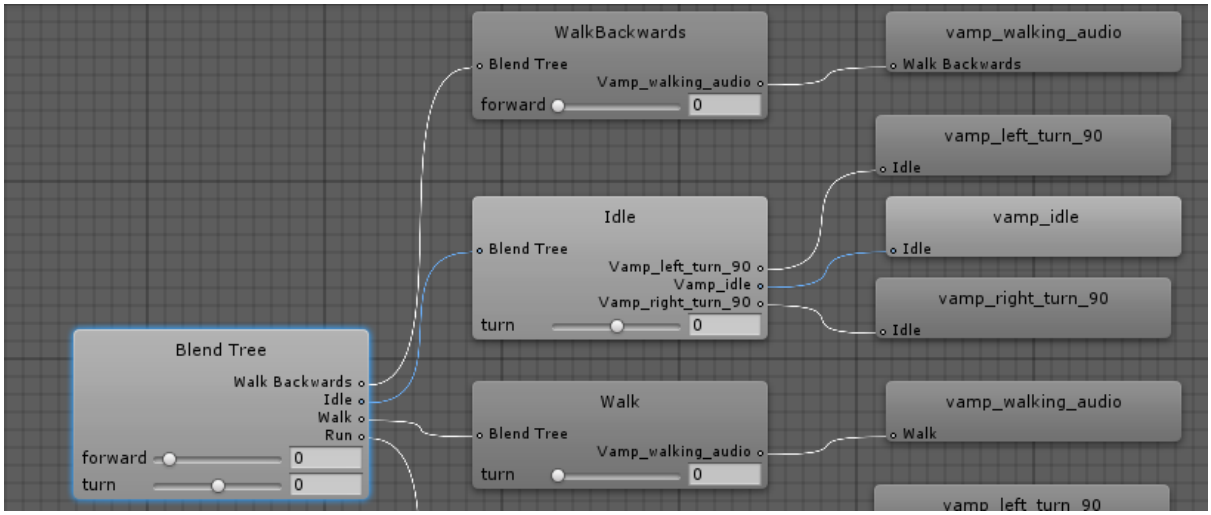

Figure 3.5.2.1: Example of blend trees

The severity of the blend can then be controlled by certain parameters, for example, the walk blend uses the turn parameter with left being minus one and right being positive one. Upon pressing the left or right keys, the animation will slowly blend from the ordinary forward walk to the left strafe or right strafe respectively.

```
private void turning()
{
    anim.SetFloat ("turn", Input.GetAxis("Horizontal"));
}

private void move()
{

    anim.SetFloat("forward", Input.GetAxis("Vertical"));
}
```

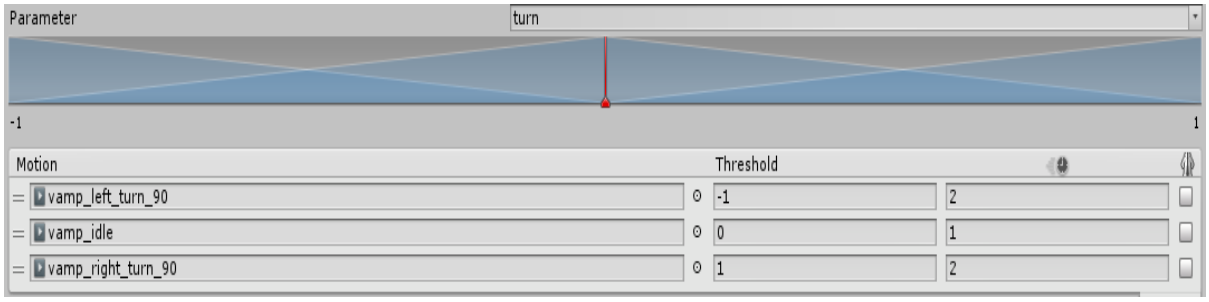

Figure 3.5.2.2: Visual representation of blend tree thresholds

Animation events were also used to help add to character movement. Animation events were added to all the movement animations so that a footstep noise would play every time the player controlled character took a step. I decided to add to this by making the footsteps quieter if the character was sneaking. Animation events involve setting “keyframes” in the animation and have a method be called every time the frame of that animation is reached.

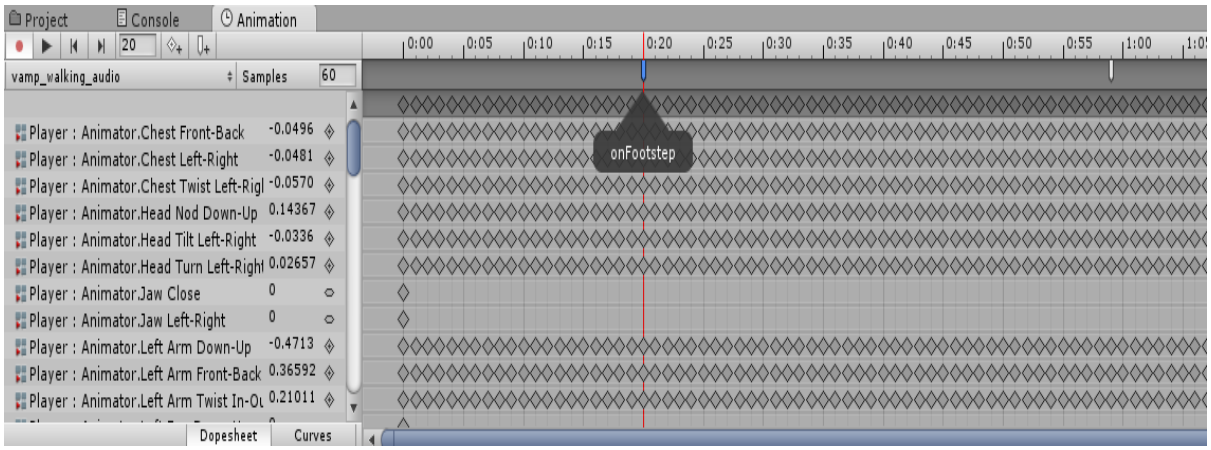

Figure 3.5.2.3: Animation event inserted into a frame of the animation

As we can see from the above example, the onFootstep method is called whenever this frame of the animation is played.

```
public Animator anim;
public AudioClip[] footstepSoundArray;
public AudioSource audioSource;

// Use this for initialization
void Start ()
{
    if (anim == null)
    {
        anim = GetComponent<Animator>();
    }

    if (audioSource == null)
    {
        audioSource = GetComponent<AudioSource>();
    }
}
void onFootstep()
{
    audioSource.clip = footstepSoundArray[Random.Range(0 , footstepSoundArray.Length)];
    audioSource.Play();

    //If player is crouching.
    if (anim.GetBool ("crouch") == true)
    {
        audioSource.pitch = 0.5f;
        audioSource.volume = 0.5f;
    }
    else
    {
        audioSource.pitch = 1.0f;
        audioSource.volume = 1.0f;
    }

}
```

3.5.3 AI

For AI, I decided to focus on the core functionality that AI needs for a stealth game while also adding in some unique interactions in order to give characters more personality. I believe that having enemies interacting with the environment will deepen the immersion thoroughly, investing players into the game world more.

3.5.3.1 Patrol and Chase

Patrol and chase are arguably the two most important functions that enemies must have in stealth games. This functionality was achieved using simple behaviour trees.

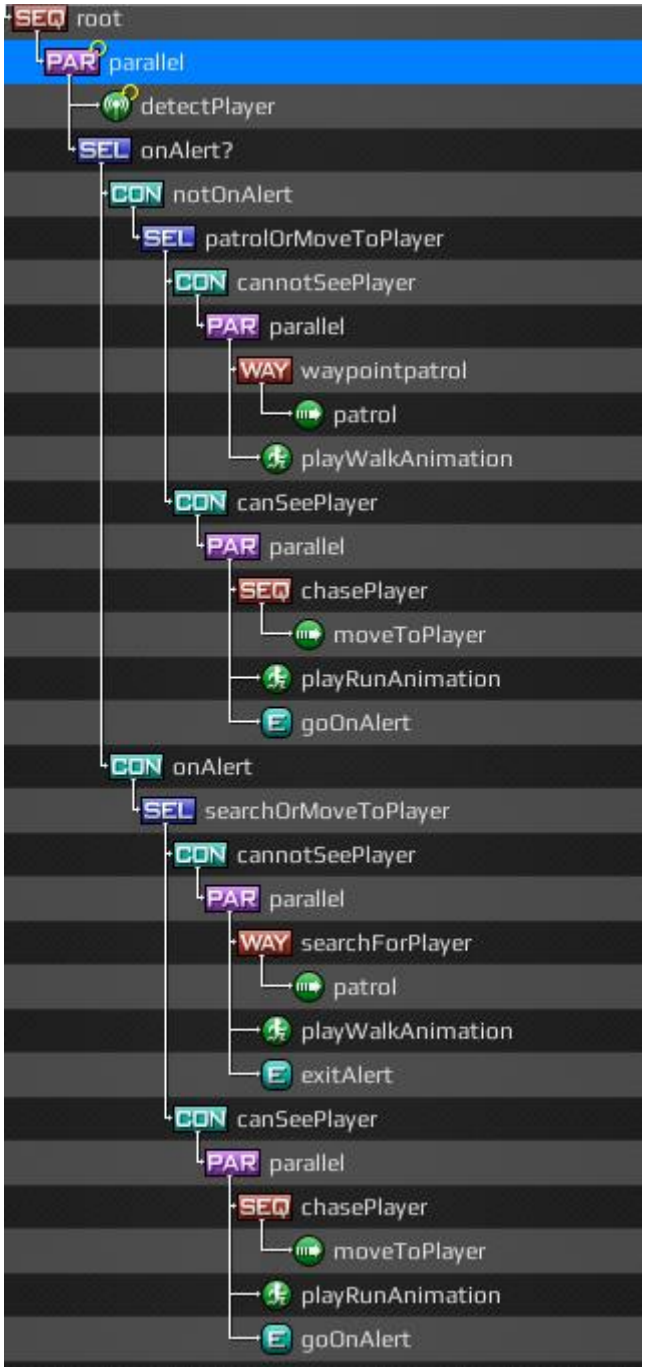

Figure 3.5.3.1.1: Patrol and chase behaviour tree

As one can see from the above image, the enemy will always attempt to detect the player. If the enemy is not on alert, they will simply continue to patrol their route whilst keeping an eye out for the player. If the enemy happens to spot the player, the alert Boolean will be set to true and they will begin to chase the player. If the enemy loses sight of the player, they will continue to search for the player as long as they are on alert. The enemy returns to their patrol after a period of time.

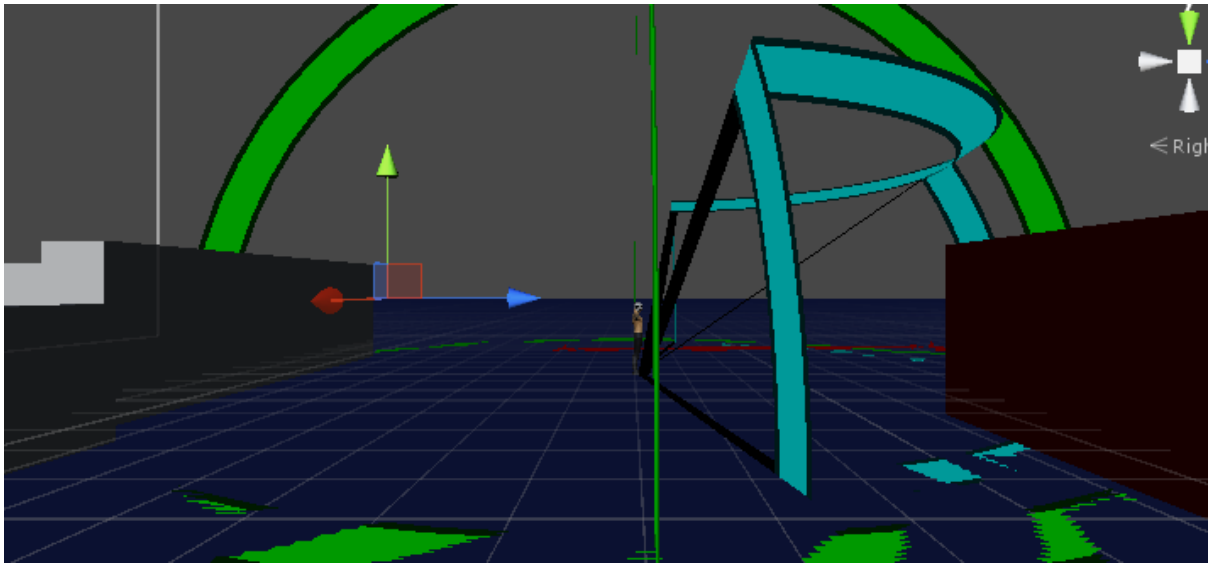

Figure 3.5.3.1.2: Enemy's sight radius shown in blue

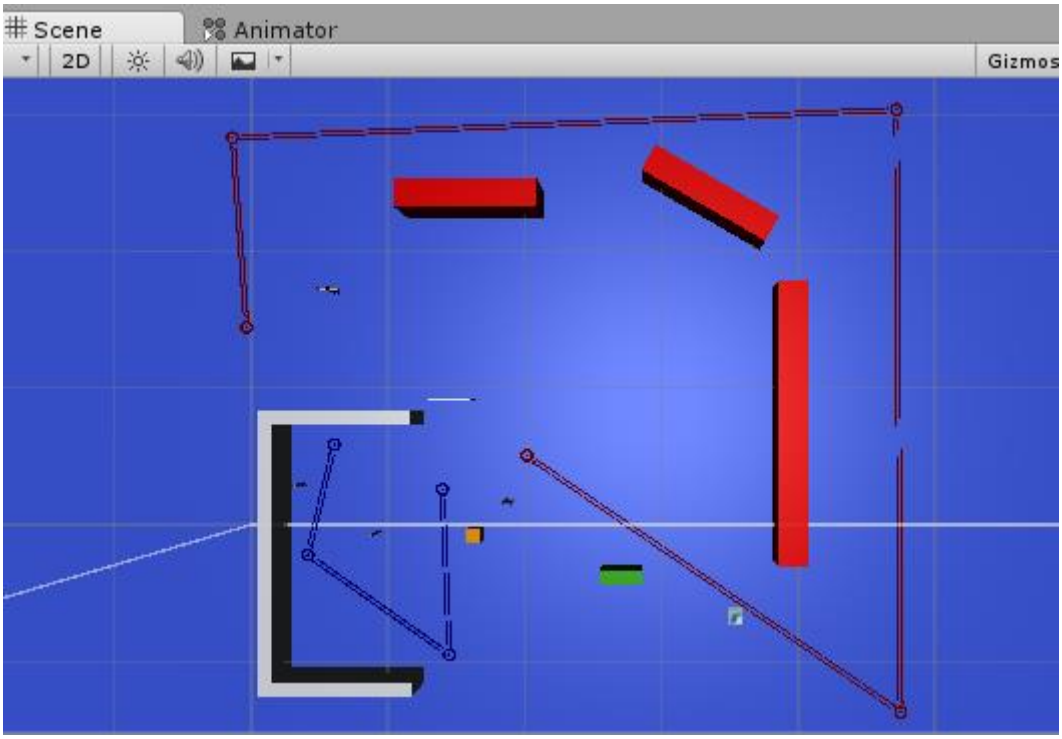

Figure 3.5.3.1.3: Patrol routes that enemies follow

3.5.3.2 Detect Player using Sound

As mentioned earlier, the player character emits noise whilst walking or running. An enemy that could sense the player by use of sound was created in order to increase the challenge for the player. This functionality was implemented using behaviour trees similar to patrol and chase, however, the parameters were now audio entities

and not visual ones. So basically, if the player emitted a sound whilst inside the hearing radius of the enemy, he would turn and then start chasing the player.

Figure 3.5.3.2.1: Noise detect node of the behaviour tree

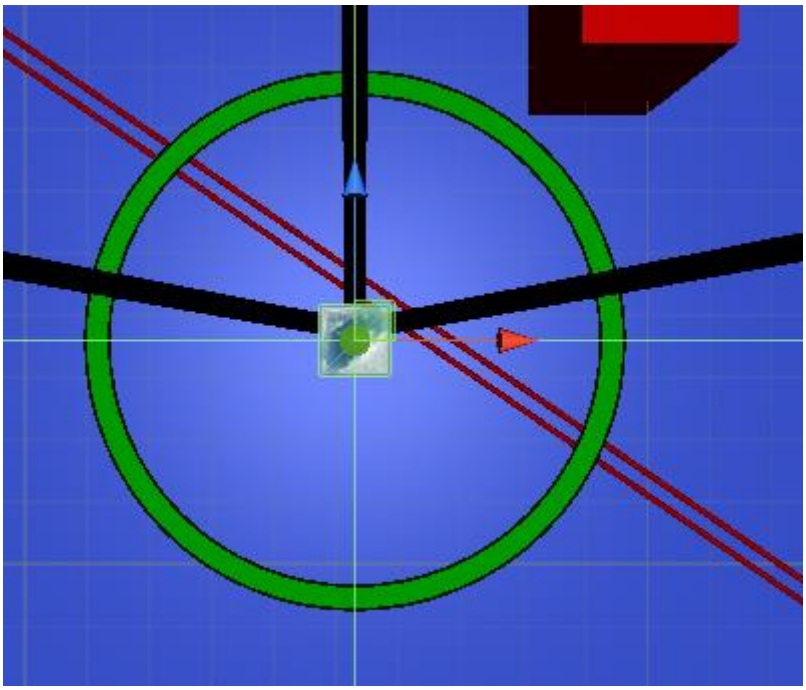

Figure 3.5.3.2.2: Hearing radius of the enemy shown in green

The hearing radius can be set to only pick up on sounds of certain strengths. For example, when the player is crouching and sneaking by the enemy, the noise of the footsteps from crouching isn't picked up by the enemy.

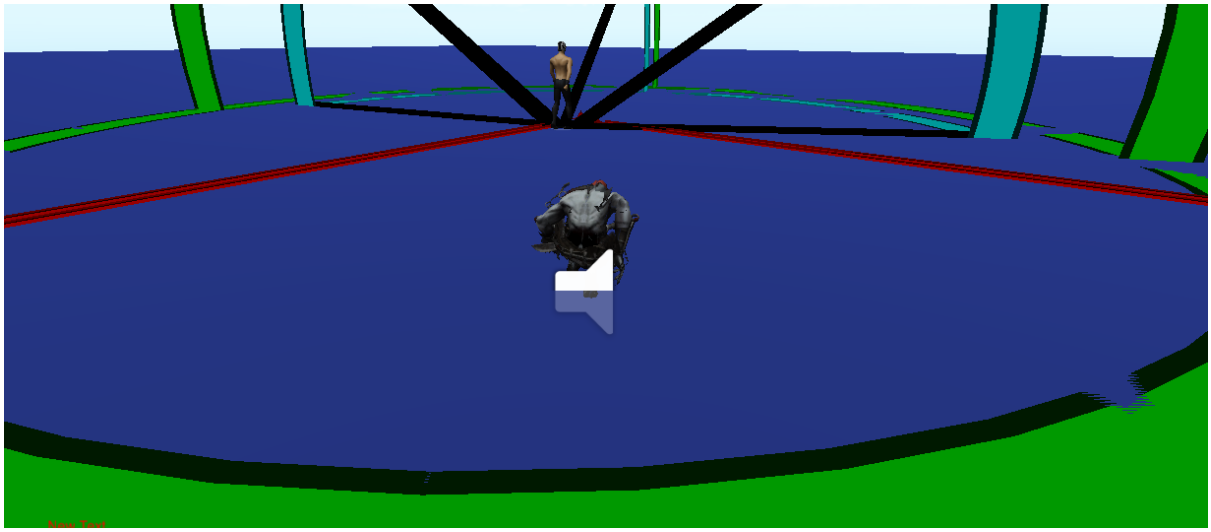

Figure 3.5.3.2.3: The player is sneaking inside the enemy's hearing radius undetected

3.5.3.3 Enemy Interacting with Environment

Interaction with the environment was also a vital part for immersive and realistic AI. In order to provide this scenario, I created a scene where an enemy patrols nearby a light source. The player can run over to the light switch and turn it off. Once the light is off, the enemy will stop patrolling the area and make his way to the light switch in order to turn the light back on. The player can use this as a distraction in order to get by the enemy if need be.

```
public class lightSourceOnOff : MonoBehaviour
{
    public GameObject referenceToLightSource;
    public AIRig lightCheckingAi;
    public bool lightOn;
    private bool collisionsOn;

    // Use this for initialization
    void Start ()
    {
        lightCheckingAi = GameObject.Find ("Robot Kyle").GetComponentInChildren<AIRig>();

        if (referenceToLightSource == null)
        {
            referenceToLightSource = GameObject.Find("LightSource");
        }

        lightOn = true;
        collisionsOn = true;
    }

    // Update is called once per frame
    void Update ()
    {
        lightState();
    }

//Tell the light whether it's on or off.
void lightState()
{
    referenceToLightSource.GetComponent<Light>().enabled = lightOn;
}


//Change the lightOn bool to its opposite.
void switchLightState()
{
    lightOn = !lightOn;
    lightCheckingAi.AI.WorkingMemory.SetItem<bool> ("isLightOn", lightOn);
}

IEnumerator disableCollisions()
{
    collisionsOn = false;
    yield return new WaitForSeconds(1);
    collisionsOn = true;

}

IEnumerator OnCollisionEnter(Collision other)
{
    if (collisionsOn)
    {
        if (other.gameObject.CompareTag("Player") || other.gameObject.CompareTag("Enemy"))
        {
            Debug.Log (other.gameObject.tag);
            switchLightState();
            //Disables collisions for some amount of time.
            yield return StartCoroutine("disableCollisions");

        }
    }
}
```

As seen from the above code snippets, we are getting a reference to the AIRig component found on the enemy. The AIRig component is found in the Rain AI developer kit and is used to handle all AI functionality through Rain. We are grabbing

the Boolean variable “isLightOn” from the AI and changing it to the state that the light is currently in; true being on and false being off.

Some collision detection is also handled here using co-routines. This collision detection code was used because, in unity, sometimes two of the same collisions can be registered in quick succession meaning the light goes from on to off to on again and this is not what is wanted. So once the player collides with the light switch, collisions are turned off on the light switch for one second to avoid the multiple collisions problem. After the one second wait, the collisions are turned on once again.

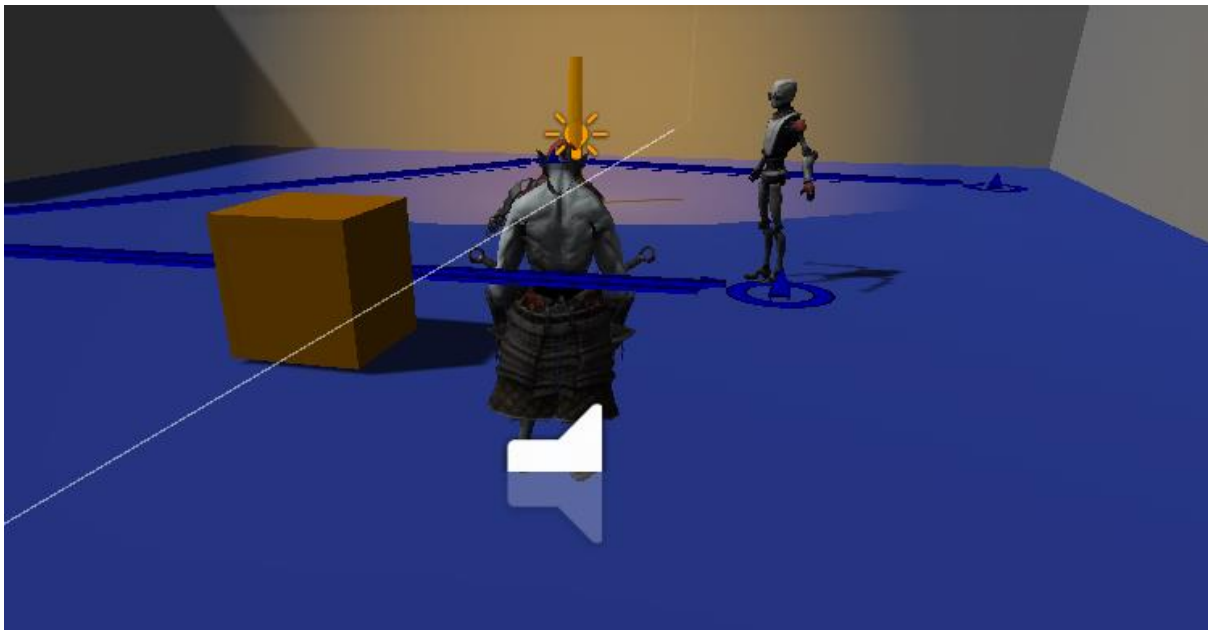

Figure 3.5.3.3.1: The enemy patrols the area whilst the light is on

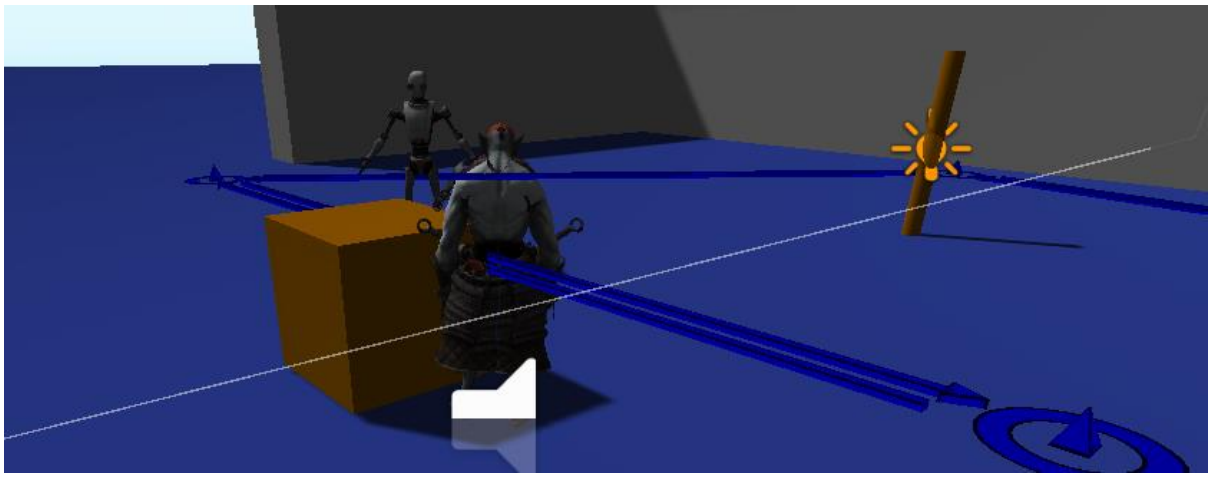

Figure 3.5.3.3.2: The enemy goes to investigate the light switch when the light is off

3.5.4 Altering Difficulty through the Biometric

Altering difficulty is a vital part of this project. Using a base heart rate variable (this would be taken from the user before they start playing the game), anything below the base decreases difficulty and anything above the base increases difficulty.

```
void setDifficulty()
{
    float heartRateDifference = currentHeartRateCache - baseHeartRateCache;


    //If the difference is less than the base heart rate.
    if (heartRateDifference < baseHeartRateCache)
    {
        Mathf.Abs(heartRateDifference);
        //Big difference.
        if((heartRateDifference/baseHeartRateCache) <= 0.6)
        {
            difficulty = 1;
        }
        //Medium difference.
        else if ((heartRateDifference/baseHeartRateCache) <= 0.8)
        {
            difficulty = 2;
        }
    }
```

```
//If the difference is greater than the base heart rate.
else if (heartRateDifference > baseHeartRateCache)
{
    Mathf.Abs(heartRateDifference);
    //Big difference.
    if((heartRateDifference/baseHeartRateCache) <= 0.6)
    {
        difficulty = 5;
    }
    //Medium difference.
    else if ((heartRateDifference/baseHeartRateCache) <= 0.8)
    {
        difficulty = 4;
    }
}
//The difference in heart rate can be considered negligable.
else
{
    difficulty = 3;
}
```

As seen from the above code, we simply get the absolute vale of the difference between the current heart rate minus the base heart rate. The difficulty is calculated using this difference. Depending on the size of the difference, the difficulty is then set. The size of the enemies hearing radius is increased or decreased depending on the current difficulty level that the player is facing. There are five difficulties with one being the easiest and five being the hardest.

3.5.5 Camera Control

Camera control has been implemented so the player has the option to peer around walls in order to spot patrolling enemies. This is done so the player can assess the situation before diving in and possibly getting caught.

```
if(target != null)
{
    //If right mouse button down.
    if(cameraButtonDown)
    {
        x += Input.GetAxis("Mouse X") * xSpeed * 0.02;
        y -= Input.GetAxis("Mouse Y") * ySpeed * 0.02;

        var rotation : Quaternion = Quaternion.Euler(y , x , 0);
        var position : Vector3 = rotation * Vector3(0.0 , 0.0 , -walkDistance) + target.position;

        myTransform.rotation = rotation;
        myTransform.position = position;

    }
    else
    {
        //cameraSetup();
        x = 0;
        y = 0;
```

```
// Calculate the current rotation angles
var wantedRotationAngle : float = target.eulerAngles.y;
var wantedHeight : float = target.position.y + height;

var currentRotationAngle : float = myTransform.eulerAngles.y;
var currentHeight : float = myTransform.position.y;

// Damp the rotation around the y-axis
currentRotationAngle = Mathf.LerpAngle(currentRotationAngle, wantedRotationAngle, rotationDamping * Time.deltaTime);

// Damp the height
currentHeight = Mathf.Lerp(currentHeight, wantedHeight, heightDamping * Time.deltaTime);

// Convert the angle into a rotation
var currentRotation : Quaternion = Quaternion.Euler(0, currentRotationAngle, 0);

// Set the position of the camera on the x-z plane to:
// distance meters behind the target
myTransform.position = target.position;
myTransform.position -= currentRotation * Vector3.forward * walkDistance;

// Set the height of the camera
myTransform.position.y = currentHeight;

// Always look at the target
myTransform.LookAt(target);
```

As shown in the above code, once the “camera button” (in the case of this simulation, the right mouse button) is held down, the camera can be rotated around the player character so they can see their surroundings clearly.

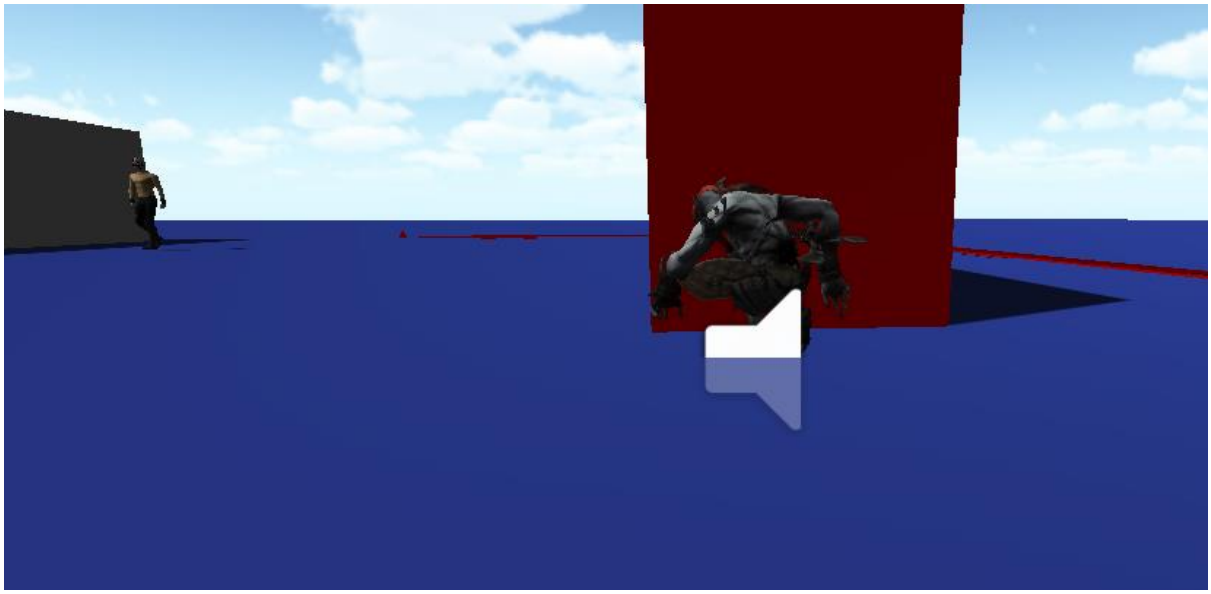

Figure 3.5.5.1: Camera controls used to peer around corner

3.6 Assets

The main character's animations and model are taken from mixamo.com whilst the enemy models that are used are taken from the unity asset store and are royalty free. The sound clips used in the project are also royalty free. I would like to thank the creators of these assets for allowing me to use these high quality models, animations and sound effects in my project. Any other assets used in the project were all self-created and may be used in future iterations of the project.

Chapter 4: Testing and Evaluation

4.1 Introduction

This section will go through the testing performed for this project. The testing process revolved around both requirements testing and user acceptance testing. The aim of software testing is “to demonstrate to the developer and the customer that the software meets its requirements” [19, p.538]. These tests were implemented in order to show the project achieved the goals it had set out to achieve. These tests were formed in order to see if biometrics are a viable game mechanic and to see if the user enjoyed the gaming experience.

4.2 Testing

4.2.1 Requirements Testing

Requirement Number	1002
Description	The player will be able to control the scene camera using some sort of user input.
Result	Pass.

Requirement Number	1003
Description	The player will be able to enter a crouching state in order to decrease the risk of detection.
Result	Pass.

Requirement Number	1004
Description	The player will emit sounds upon movement.
Result	Pass.

Requirement Number	1005
Description	The player will be able to interact with in-game objects.
Result	Pass.

Requirement Number	1006
Description	Entering a crouching state will decrease the amount of sound the player character emits.
Result	Pass.

Requirement Number	2001
Description	The game will be able to read in a value for some form of biometric.
Result	Fail.

Requirement Number	2002
Description	The player's base heart rate will be taken before the simulation begins.
Result	Fail.

Requirement Number	2003
Description	The difficulty will increase or decrease depending on the increase or decrease in heart rate.
Result	Pass.

Requirement Number	3001
Description	The enemies will patrol an area.
Result	Pass.

Requirement Number	3002
Description	The enemies will be able to detect the player by line of sight.
Result	Pass.

Requirement Number	3003
Description	The enemies will chase the player upon seeing the player.
Result	Pass.

Requirement Number	3004
Description	The enemies will be able to detect the player through sound the player character makes.
Rationale	Pass.

Requirement Number	3005
Description	The enemies will turn to the source of noise emitted by the player.
Result	Pass.

Requirement Number	3006
Description	The enemies will be able to interact with their environment.
Result	Pass.

Requirement Number	3007
Description	The enemies will return to their patrolling state upon failing to find the player after investigating an area.
Result	Pass.

Requirement Number	3008
Description	The enemies will switch from a chase state to an investigate state upon losing site of the player.
Result	Pass.

Requirement Number	3009
Description	The enemies will return to patrolling or investigating upon failing to catch the player whilst chasing them.
Result	Pass.

4.2.2 User Acceptance Testing

User acceptance tests were conducted on five, unbiased volunteers. Nielsen states that only five users are needed in order to discover eighty five percent of usability issues within an application [20]. Any more testers would yield negligible results. Three volunteers belonged to the eighteen to twenty five age groups whilst the remainder belonged to the thirty five and over age group. These volunteers used the application for a period of time and gave their feedback on the simulation regarding a few key categories. The mark for each category was given out of five, and any comments that were given were recorded and taken into account.

The feedback forms that the volunteers received are as follows:

User Number	The number given to identify the tester.
Immersion	How immersed the player felt. One being not very immersed and five being extremely immersed.
Use of Biometric	What the player thought of the use of the biometric. One being poorly implemented and/or not enjoyable and five being well implemented and/or very enjoyable.
AI	Did the player find the AI challenging or interesting. One being not challenging or interesting. Five being fairly challenging and interesting.
Comments	Comments that the player had about the project.

Table 4.2.2: Representation of user acceptance form

(See Appendix A for results)

4.3 Evaluation

The simulation satisfied most of the requirements that were laid out. The user acceptance testing showed favourable results towards the viability of biometric gaming. Obviously, the weak point is the emulated heart beat that was being used for testing purposes. This weakened the results overall as the simulation wasn't taking in the user's actual heart rate therefore weakening the immersion figures as a result. The results for AI were very impressive. Interestingly, one user debated that a third person game is harder to get immersed in but thought the stealth genre was a good choice.

Chapter 5: Conclusions

5.1 Introduction

This section of the thesis will outline what the project has achieved as a whole, how well these goals have been achieved and recommendations on possible improvements.

5.2 Reflection

Upon completion of testing, I believe that the project achieved most of its goals to some success. Sadly, due to time constraints and lack of funding, I was unable to procure a heart rate monitor in order to test the simulation to its fullest. Whilst an emulated heartbeat was used, I believe that a real heartbeat (from the user) would offer better test results and would help confirm my conclusions more thoroughly.

Whilst I was quite happy with the test results, a wider testing group with varying ages and backgrounds would give more thorough results that could be deemed more beneficial.

The tests show that biometric gaming appears viable as a game mechanic, and in fact does deepen immersion.

5.3 Recommendations

In terms of recommendations for future iterations of this project, I have a few suggestions that I believe would help improve the project greatly.

Firstly, acquiring a heart rate monitor in order to use with the simulation is of upmost priority. I believe the testing would benefit a lot from the use of a proper heart rate rather than an emulated one.

Secondly, I believe the implementation of more biometrics would be beneficial. The use of multiple biometrics in conjunction with each other would help retrieve a more varied data set and I believe this would also help deepen the immersion for the user.

Whilst the core AI functionality has been completed, I believe more complex AI interaction would help increase the immersion. Having the AI interact with its environment even more so would offer a more realistic and organic experience that would immerse the user in the game world.

In terms of immersion, a version of the project that utilises the Oculus Rift would be beneficial. I believe the Oculus Rift would improve the immersion even more for the user and that biometrics would become even more viable when used with the Oculus. However, I believe the simulation would have to be turned from a third person game to a first person game view if the Oculus was to be implemented.

In terms of testing, more testers for user acceptance testing would be beneficial. Whilst, according to Nielsen, five testers would be sufficient [20], I believe that a wider, more diverse testing group would refine the results even more so and obtain a more meaningful result. Also, it would be worthwhile to get testers from the medical field in order to get their opinion on the simulation from a medical standpoint.

Finally, if the simulation was to be made into a game; I strongly believe that a Bluetooth adapter and heart rate monitor should be paired with the game on sale. This would ensure that the game would work with the hardware that is provided rather than using a wide variety of varying heart rate monitors or Bluetooth adapters.

References

[1] Jim Clarke (2001). Stress in the Workplace. Available at: http://www.irishhealth.com/article.html?id=1241 [Accessed 04 April 2016]

[2] Kristen J. Nyitray (2011). William Alfred Higinbotham: Scientist, Activist, and Computer Game Pioneer. IEEE Annals of the History of Computing.

[3] Amanda Kudler. Timeline: Video Games. Available at: http://www.infoplease.com/spot/gamestimeline3.html [Accessed 05 April 2016]

[4] Entertainment Software Rating Board (2009). How much do you know about video games? Available at: http://www.esrb.org/about/video-game-industry-statistics.aspx [Accessed 05 April 2016]

[5] Entertainment Software Association. Industry Facts (2015). Available at: http://www.theesa.com/about-esa/industry-facts/ [Accessed 02 April 2016]

[6] Anil K. Jain, Patrick Flynn, Arun A. Ross (2008). Handbook of Biometrics.

[7] C. Schubert, M. Lambertz, R.A. Nelesen, W. Bardwell, J.-B. Choi and J.E. Dimsdale (2008). Effects of stress on heart rate complexity - A comparison between short-term and chronic stress.

[8] Hans Selye (1950). The Physiology and Pathology of Exposure to Stress.

[9] K. Siomos, G. Floros, D. Braimiotis, A. Lappas, I. Karagiannaki-Kastani (2010). Stress-related, self-inflicted wounds during online gaming.

[10] C. Morris, P. Hancock, E. Shirkey, Motivational Effects of Adding Context Relevant Stress in PC-Based Game Training. University of Central Florida, 2004.

[11] Katherine A. McGonagle and Ronald C. Kessler (1990). Chronic Stress, Acute Stress, and Depressive Symptoms.

[12] Darryl Charles. Enhancing Gameplay: Challenges for Artificial Intelligence in Digital Games.

[13] Ayse P. Saygin, Ilyas Cicekli and Varol Akman (2000). Turing test: 50 years later.

[14] Andrej Krenker, Janez Bešter and Andrej Kos (2011). Introduction to the Artificial Neural Networks, Artificial Neural Networks - Methodological Advances and Biomedical Applications, Prof. Kenji Suzuki (Ed.), ISBN: 978- 953-307-243-2, InTech, Available from: http://www.intechopen.com/books/artificial-neural-networksmethodological-advances-and-biomedical-applications/introduction-to-the-artificial-neural-networks.

[15] Isla. D. (2005). Handling complexity in the Halo 2 AI. In Game Developers Conference.

[16] Nevermind. Available at: http://nevermindgame.com/ [Accessed 10 March 2016]

[17] Luciano Bernardi, Joanna Wdowczyk-Szulc, Cinzia Valenti, Stefano Castoldi, Claudio Passino, Giammario Spadacini, Peter Sleight (2000). Effects of controlled breathing, mental activity and mental stress with or without verbalization on heart rate variability.

[18] Blend Trees. Unity Documentation (2016). Available at: http://docs.unity3d.com/Manual/class-BlendTree.html [Accessed 04 April 2016]

[19] I. Sommerville, *Software Engineering*. Boston: Addison-Wesley, 2011.

[20] Nielsen, Jakob, and Landauer, Thomas K.: “A mathematical model of the finding of usability problems,”Proceedings of ACM INTERCHI’93 Conference (Amsterdam, The Netherlands, 24-29 April 1993), pp. 206-213.

Glossary

Boolean – A data type with only two possible values: true or false. It is named after mathematician George Boole.

Co-routines - A function that can pause execution and return control to the main program but can continue at its last position upon receiving control again.

Learning Curve – The rate of a person’s progress in terms of gaming experience.

Oculus Rift – A virtual reality headset.

Unity - A game engine developed by Unity technologies.

Unity Asset Store – A library of assets hosted by the Unity game engine.

Variable - A storage location with an associated symbolic name.

Virtual Reality - A term commonly used to describe a computer-simulated reality.

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Networks
ESA	Electronic Software Association
ESRB	Entertainment Software Rating Board
TV	Television
VR	Virtual Reality

Appendix A: User Acceptance Testing

A.1 User 1

User Number	1
Immersion	Mark given: 2
Use of Biometric	Mark given: 3
AI	Mark given: 4
Comments	Hard to feel overly immersed in a third person game. Liked the stealth gameplay. AI was very good.

A.2 User 2

User Number	2
Immersion	Mark given: 3
Use of Biometric	Mark given: 4
AI	Mark given: 4
Comments	Biometric idea is interesting although I would like to see the game work with the heart rate monitor.

A.3 User 3

User Number	3
Immersion	Mark given: 4
Use of Biometric	Mark given: 2
AI	Mark given: 4
Comments	Really like the character movement. The increase and decrease of difficulty is very cool. Would like to see the game with heart rate monitor implemented.

A.4 User 4

User Number	4
Immersion	Mark given: 3
Use of Biometric	Mark given: 2
AI	Mark given: 4
Comments	Like how AI interacts with the player and the light source. Difficulty increase/decrease is interesting but needs heart rate monitor.

A.5 User 5

User Number	5
Immersion	Mark given: 4
Use of Biometric	Mark given: 3
AI	Mark given: 3
Comments	Solid game. Interesting mechanics. Would definitely play it if fleshed out more.