Modelling Environmental and Temporal Factors on Background Characters in Open World Games

by

Tony Cullen, B.Sc. (Hons)

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Master of Science in Computer Science

(Interactive Entertainment Technology)

University of Dublin, Trinity College

September 2015

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Tony Cullen

August 28, 2015

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Tony Cullen

August 28, 2015

Acknowledgments

Firstly I would like to thank my supervisor Dr. Mads Haahr for all his advice and support provided throughout this dissertation.

I would also like to thank my family for their support and encouragement throughout the entire year.

Tony Cullen

University of Dublin, Trinity College
September 2015

Modelling Environmental and Temporal Factors on Background Characters in Open World Games

Tony Cullen
University of Dublin, Trinity College, 2015

Supervisor: Dr. Mads Haahr

The goal of this dissertation is to explore methods for increasing the believability of "background" characters in open world games. While the background characters do not influence the game's main storyline, they play a vital role in making the game world feel more believable to the player. These characters often follow scripted behaviours, carrying out the same tasks each day unless otherwise influenced by the player. Furthermore, the environmental and time-related cycles of the game world, such as day/night cycles and weather cycles, have no impact on the behaviours and actions of these characters. This project proposes an emotional model which drives the behaviours of these background characters through both the tasks they carry out and the environmental and temporal factors of the game world. A State Manager is used to determine the next task for the non-player characters (NPCs) based on their current emotional state and the traits they possess. Traits, e.g. nocturnal or cold-blooded, are introduced to accommodate the different races found in open world games, such as

humans, lizards, and orcs. Having the environmental and temporal factors influence the behaviours of these races differently adds a further element of believability to these characters. The model is tested in a game environment within the Unity game engine and shows potential, as the characters' actions are determined by their emotional state, as opposed to the scripted behaviours typically found in open world games. With further work, this model could have potential for use in open world games.

Contents

Acknowledgments **iv**

Abstract **v**

List of Tables **x**

List of Figures **xi**

Chapter 1 Introduction **1**

1.1 Open World Games 1
1.2 Challenges Of Open World Games 3
1.3 Motivation 4
1.4 Objectives 5
1.5 Dissertation Roadmap 6

Chapter 2 State Of The Art **8**

2.1 Believability 8
2.1.1 Believable Agents 9
2.1.2 Believable Models In Academia 12
2.2 Current AI In Open World Games 13
2.2.1 S.T.A.L.K.E.R 14
2.2.2 Fable II 15
2.2.3 Oblivion, Skyrim, And Fallout: New Vegas 15
2.3 Sensing In Games 16
2.4 Control Theory 18
2.4.1 Open-Loop Control System 18

2.4.2 Closed-Loop Control System 19
2.5 Cycles in Games 19

Chapter 3 Design **22**
3.1 Previous Work 22
3.2 Defining The Model 22
3.2.1 Global Time Manager 23
3.2.2 Modelling Cycles 23
3.2.3 Emotional States 27
3.2.4 Tasks 28
3.2.5 State Manager 29
3.2.6 Traits 31
3.2.7 High Level Architecture 32

Chapter 4 Implementation **34**
4.1 Platform Selection 34
4.2 Implementing The Model 34
4.2.1 Creating The Cycles 34
4.2.2 Integrating The Model 39
4.3 Extensibility Of The Model 41
4.3.1 Cycle Manager 41
4.3.2 Tasks 41
4.3.3 Tunable Parameters 41

Chapter 5 Evaluation **43**
5.1 Improvement Over Existing Approaches 43
5.2 Support For Various Character Races 45
5.3 Efficiency Of Implementation 45
5.4 Impediments 46

Chapter 6 Conclusion **47**
6.1 Contributions 47
6.2 Future Work 48
6.2.1 Improving the Current Model 48

6.2.2 Integration 49
6.2.3 Extending the Model 49
6.3 Final Thoughts 49

Appendices **50**

Bibliography **51**

List of Tables

3.1 Cycles 24
3.2 Temperature Ranges 25
3.3 Emotional States 27
3.4 Tasks 29
3.5 Traits 31

4.1 Particle System Parameters 36

5.1 Nord Simulated Day 44
5.2 CPU Usage Time 46

List of Figures

2.1 Open-Loop Control System [33] 18
2.2 Closed-Loop Control System [33] 19

3.1 State Manager Task Selection 30
3.2 High Level Architecture 33

4.1 Day/Night Cycle Sunrise 36
4.2 Weather Cycles 37
4.3 Lunar Cycle 38
4.4 Enchanted Forest During A Full Moon 39
4.5 Model Visualisation 40

Chapter 1

Introduction

The topic of this dissertation concerns the believability of "background" characters in open world games. These are non-player characters (NPCs) which do not play a role in the game's main storyline, but rather those which exist in making the game world feel much more believable to the player. In most open world games, the background characters follow a repetitive schedule in which they carry out the same tasks over and over again at specific times. Furthermore, the environmental and time-related cycles of the game world, which are used to improve upon the believability of the world itself, have no impact on the behaviours and actions of these characters. In order to improve upon the behavioural fidelity of the background characters in open world games, these issues must be addressed. This dissertation proposes a model which drives the behaviours of these background characters based on environmental and temporal factors inherent of the region in which the NPC inhabits. It is hoped that this model would provide a more realistic and engaging gameplay experience by removing the predictability typically found in current NPCs.

1.1 Open World Games

An open world game is one in which the player has the ability to freely roam a virtual world and is given the freedom to choose how and when to complete their objectives [40]. Typically, these open world games games provide a large environment that the player can explore, allowing them to interact with and receive quests from NPCs that populate

the world. Open world games gives the player a sense of freedom and control in that they can play the game how they want [9]. In contrast to traditional games, where the levels are designed to be linear and the player is constrained to areas by invisible barriers, open world games provide the player with a much more interactive experience as a result of the freedom it employs. The actions and events in open world games are also typically scripted, taking some of the control away from the player.

The first open-world game was developed in 1984 and was called *Elite*. In Elite, the player controls a space ship and can travel across the galaxy to countless planets in order to trade cargo [40]. Players also were able to upgrade their ships and fight as or against space pirates. Elite provided the player with the freedom to choose their own way to gain capital, with no one right way to the play the game [6].

Other key titles in the open world genre include *Fallout 3* [44] and *Grand Theft Auto (GTA) III* [19]. Grand Theft Auto III, the third game in the *Grand Theft Auto* [18] series, was released in 2001 and built upon its predecessors by moving the series into 3D. This allowed for the player to truly visualise and explore the city of the game, as well as providing the player with the freedom to carry out missions using weapons and vehicles of their choosing. Prior to this game, with very few exceptions, the game world was static and there was normally minimal interaction between the game world and the player. Some examples of these limited interactions between the game world and the player, up to that point, included "falling blocks","rising spikes" or "swinging ropes" [40]. Grand Theft Auto III changed this by having the environment play a role in the players gameplay experience. Brian Baglow describes the environment accordingly, saying that "the environment is not merely the setting for the action, but is an active part of the overall gameplay, which affects and reacts to the player as they progress" [40]. The successors of this game, *Grand Theft Auto IV* [20] and *Grand Theft Auto V* [21], expanded this open world city model and introduced a considerable number of activities that the player can engage with such as scuba diving or base jumping.

In 2008, Bethesda Game Studios released Fallout 3 an open world role-playing game which drops the player into a post-apocalyptic future, as a result of nuclear holocaust, where they have the freedom to explore a large-scale environment known as the Capital Wasteland. The Capital Wasteland contains over one hundred and fifty locations that the player can explore, including towns, cities, and monuments. The magnitude of the

world meant that the player could invest hundreds of hours in the game carrying out the activities available and choosing how to complete them. This includes completing quests, exploring the wasteland for items and loot, discovering hundreds of locations and unique characters and fighting off the various hostile groups throughout wasteland such as mutants, mercenaries and robots. Fallout 3 was among the games which helped advance open world gaming into new territories [40] and won numerous awards as a result of its open-ended gameplay [42].

1.2 Challenges Of Open World Games

One challenge of open world games is attempting to build a large-scale environment without hindering the games performance. In order to make the environment believable, it must be populated with buildings, objects and characters. This is not only time-consuming but also means that the resources available are limited. One consequence of this is limited AI for NPCs. The NPCs in open world games, specifically the background characters who are non-essential to the games main story, cannot have complex AI because of the performance issues that would arise. There can be dozens of these characters on screen at a given time, and so the NPCs actions are typically scripted and their interaction with the environment is limited.

Another challenge of open world games is immersing the player within the game world. Linear games often utilise storytelling as a means of immersion, in which cut scenes and scripted events are used to tell a story that the player helps the player become engaged with the game. Since open world games provide the player with the freedom to play the game how he/she wants to, it is often difficult to develop a storyline without restricting this freedom. Some open world games have tried to introduce branching storylines, where the actions of the player have a consequence on the games ending. *Fallout: New Vegas* [16] is one example of an open world game which attempted to introduce branching storylines. Depending on the decisions of the player, such as carrying out quests in a certain manner or siding with particular factions, these will dictate the outcome of the story. This gives the player an even greater sense of freedom, as their actions have consequences and they can decide how the story unfolds. Despite the use of branching storylines in open world games, the storytelling is generally limited and is not enough to immerse the player in the game world. Some

firmly believe that open worlds tell their own story and a storyline is not such an important feature. For example, Ed del Castillo believes that "the future of video games isn't in storytelling, but in story-playing" and that "we have the opportunity to show variety of choice, to let the player play the game he or she wants to play, and experience what they want to experience" [40]. As a result of the limited storytelling in open world games, the virtual environment, which refers to all the elements which make up the game world including buildings, objects, weather, day and night cycles and the NPCs which inhabit it, plays a vital role immersing the players within the game world.

1.3 Motivation

In open world games, the goal of the environment and its NPCs is to help make the player become fully immersed into the virtual world by enhancing the world's believability. The term "immersion" in games is used to describe the degree of involvement [10, p.1]. Three different levels of involvement were identified in games. Engagement is the first level of involvement where the player invests time, effort and attention into the game. In order to for the player to become engaged in the game, they must overcome the barrier of gamers' preference. If they do enjoy the style of the game, they are unlikely to even engage with it [10, p.2]. For players to reach the second level of immersion, engrossment, they must overcome the barrier of game construction, where features of the game can directly affect the players emotions. To reach the final level of involvement, the barriers of empathy and atmosphere must be overcome. This level of immersion is called "total immersion". To achieve total immersion, the player must empathise with main character or team. Furthermore, the atmosphere must be truly believable and be able capture the players attention. In total immersion, the players gain a sense of presence within the game and are cut off from reality [10, p.3].

In order to create a believable NPC, it is necessary to associate believability with respect to the NPC's role in the game. If the character is essential to the games main storyline, they should generally possess a higher degree of complexity and believability as opposed to those who are not, such as the background characters. An example of complex AI can be seen in the character Ellie in *The Last Of Us* [14]. Ellie not only acts as a companion to the player by assisting the player in combat, but also

forms a relationship with the player. This is done in various ways throughout the game including the use of ambient animations and interesting conversation tracks as the game progresses [15]. While this type of complex AI helps immerse the player in the game, it wouldn't be suitable for the non-essential background characters, such as farmers and shopkeepers, found in open world games. These background NPCs are added in order enhance the players believability of the virtual world, but do not influence the main story of the game. As a result of their role in the game, and to prevent any performance issues, their actions and events are often scripted, preventing any dynamic behaviour from these NPCs. If the NPCs are predictable and act based on scripted behaviours, it can take away from the gameplay experience for the player.

As technology advances, so does our ability to improve upon the conventional scripted NPCs to create more realistic ones. As a result of limited computational power, the audiovisual aspects of games have often taken precedence over the AI components, resulting in simplistic AI [54]. This makes AI even more important for immersion into the game world. As the NPCs begin to look and sound more real, their simplistic behaviour becomes even more evident and takes away from this element of realism. With advances in computational power, there is now an opportunity to enhance the AI found in games, making them more believable to the player.

The main motivation for this project is to improve upon the scripted nature of background characters in open world games. By having them perceive the environment they inhabit and have it play a role in their actions, it can improve the overall gameplay experience for the player.

1.4 Objectives

The goal of this dissertation is to develop a model for increasing the believability of background NPCs that populate the virtual environment in open world games. The NPCs this dissertation focuses on are the background characters that are non-essential to the games main storyline, but are still required to enhance the players gameplay experience. The model will build upon the past work of previous IET students John Fallon [17] and Tiarnán McNulty [32], who also looked to improve upon the behavioural fidelity of background characters in open world games. The model aims to achieve the following objectives:

1. Offer a tangible solution for increasing the believability of background characters by having environmental and temporal factors drive their behaviours.

2. Produce a generic implementation for the various character types found in open world games.

3. Provide an efficient implementation, so it does not affect the game's performance.

The primary goal of this project is the first objective, which looks to improve the believability of NPCs that currently exist in open world games. By having these characters perceive and react to environmental and temporal factors, it adds to the sense of realism in the game world. If the environmental and temporal cycles have no affect on the behaviours of these background characters, this does not truly represent human-like behaviour and it can take away from their believability. The second objective of this model is to implement it so that it can be applied to the various races of characters typically found in open world games. For example, in *The Elder Scrolls V: Skyrim* [45] there are ten main races which exist in the world, such as Nords (humans), Argonians (lizards) and Orcs [23]. Having the environmental and temporal factors influence the behaviours of these races differently would add a further element of believability to these characters. An example would be with the Argonian race. Since this race is a type of lizard, they could be modelled as being cold-blooded. As a result, temperature, which is influenced both based on the current time of day and the current weather, would affect them more than other races where as rainfall wouldn't have as much influence on them. Finally, the last objective is to produce an efficient model. In open world games, there is generally a large number of NPCs that populate different regions and many of these can be visible at any given time. As a result, it is important that this model does not hinder the performance of the gameplay.

1.5 Dissertation Roadmap

This dissertation is structured as follows:

- **Chapter 2** of this dissertation provides an overview of the current state of the art in industry, including the current approaches to making NPCs more believable.

- **Chapter 3** presents the design for the model which aims to increase the believability of background NPCs in games by having environmental and temporal factors drive their behaviours.
- **Chapter 4** presents a prototype implementation for the model that builds upon the past work of previous IET students John Fallon [17] and Tiarnán McNulty [32].
- **Chapter 5** presents an evaluation of the implemented model, assessing the extent to which the project objectives were achieved.
- **Chapter 6** will conclude this dissertation, summarising both the contributions of this project and any future work which could be undertaken.

Chapter 2

State Of The Art

The purpose of this chapter is review the state of the art in areas of work related to this project. First, the notion of believability and believable agents is discussed, along with research carried out on believable NPCs in academia. This is followed by a brief overview of the current AI that is used to drive background characters in open world games. The next section provides details on sensing in games, and how it is used to make characters aware of their environment. After this is a discussion on control systems and how they are used to manage the behaviours of a system. A control system presents a possible approach to managing a system consisting of temporal and environmental cycles which are used drive the behaviours of NPCs. Finally, the last section talks about the temporal and environmental cycles found in games and how they are used to help immerse the player into the game environment. Furthermore, it discusses the use of circadian rhythms as a general approach to modelling temporal cycles.

2.1 Believability

Although the notion of believability in open world games is particularly difficult to define as it is both subjective and domain dependent, it essentially means maintaining a "suspension of disbelief" among the player. With respect to background characters in games, we can determine these to be believable if they fulfil their role in the environment in a realistic manner.

2.1.1 Believable Agents

The concept of a what defines a believable agent has been studied in various fields including literature, theatre, film, and other media. Although there is no single definition that can determine what makes a believable agent, it is often regarded that in order for a character to be believable, it must maintain a suspension of disbelief among its audience [7]. Thomas and Johnston, animators of Disney, believe that the goal of a believable character is to produce the illusion of life [49]. According to Bryan Loyall, a believable agent suspends the disbelief of an audience by providing "a convincing portrayal of the personality they expect or come to expect" [30]. Although these definitions can be applied to believable agents in almost all applications, video games provide an added layer of complexity as the player can typically interact with the agents in the environment [48]. Livingstone suggests that a believable character depends both on the game being played and the role of AI in the game and poses the question "should AI be indistinguishable from a human player or should it try to be a better role-player?" [29].

A major issue in open world gaming is the predictability of background characters. When agents are predictable, consistently carrying out the same tasks in a repetitive manner, it can destroy the illusion of life [48]. By adding unpredictability to these characters, but not so much as to make their actions random, goes a long way in improving upon the believability of these characters [11]. There has been a number of different criteria outlined for developing believable agents. Warpefelt and Stråål studied and analysed the NPCs in open world games with a view to identify the current weaknesses in NPC behaviour. The goal of this study was to outline the issues of NPCs in open world games with the intention of achieving immersion-strengthening behaviour. They developed a list of six anti-heuristics which should be followed to make NPCs more believable. By avoiding these weaknesses of current NPCs found in games, it could enhance the believability of these characters [53]. The six anti-heuristics are as follows:

1. Ensure that the NPC always knows everything that is happening in the world. It should be omniscient!

2. Ensure that the NPC is seemingly unaware of things that it should feasibly be aware of.

3. Ensure that the NPC is seemingly unaware of what others are doing that could affect the NPCs, its friends or the environment.

4. Ensure that the NPC is seemingly unaware of actions performed that directly involve or affect it.

5. Ensure that the NPC always reacts in such a way that it makes its present situation worse.

6. Ensure that the NPC, through lack of reaction, never improves its situation.

In another criterion, Bryan Loyall [30] outlines a list of requirements that he believes a believable agent should follow, based upon the analysis of artists and experience with agents.

Personality

Personality is defined as "all of the particular details — especially details of behaviour, thought and emotion — that together define the individual" and is regarded as the most important requirement for believable agents. To create a believable agent, they must have a personality and portray individuality similar to those found in traditional characters.

Emotion

Emotion is a fundamental requirement in that artists often refer to as the attribute which gives life to the character. In order for an agent to be appear as being believable, they must have emotional reactions and expressions that are true to their personalities.

Self-Motivated

Another requirement for believable agents is that they are self-motivated. Autonomous agents must not only react based solely on external stimuli, but also perform actions of their own accord - the result of their inner drives and desires.

Change

In order for an agent to be believable, it must have the ability to change and grow. While change and growth are essential, it should be done so in a manner

that is reflective of the character's personality, as opposed to a random change. The main challenge of this requirement is being able to remove the predictability of characters by adding change and growth to an agent, while keeping true to their characteristics.

Social Relationships

Social relationships between characters is another requirement for creating a believable agent. The social interaction between characters plays a vital role in making the characters seem alive. These interactions are influenced and dependent upon their relationship. As a result, it is not simply enough to define a relationship between characters as "friends" or "enemies". The relationship between characters must be represented through the use of detailed behaviours and interactions. These behaviours need to be both character and relationship specific, as no two characters or relationships are the same.

Consistency of Expression

Consistency of expression is a basic requirement for believability. All agents have a multitude of ways to express themselves. This includes the use of facial expressions, posture, and gestures. For a character to maintain believability, it must combine the various avenues of expression to convey the message that is appropriate for the personality, feelings, situation, thinking, etc. of the character. The consistency of expression is vital in order to maintain the suspension of disbelief.

The Illusion of Life

The final requirement for believability is the illusion of life, which is actually a collection of requirements. In presenting an illusion of life, the agent must firstly appear to have goals. Secondly, for the character to be convincing, they must be able perform actions in parallel and complete their goals concurrently, as believable characters should be able to perform multiple activities at the same time. Agents must be also be both react and respond to changes in their environment. The speed in which the agent reacts and responds to these changes must done in a range that is deemed believable. Characters should also appear "situated" in that "they change what they are doing and how they do it in response to

the unfolding situation". Another requirement is that an agent must be resource bounded, in that they posses physical and mental limitations. Believable agents should also exist in a social context and be aware of the culture and other social aspects of world in which they inhabit. Characters ought to be "broadly capable" in which they can think, sense, act, have emotions, and more in the world that they exist. The final requirement of the illusion of life is that the agent's capabilities are seemingly well-integrated. The illusion of a believable character can be destroyed if the boundaries between capabilities of the agent can be seen by the player or viewer. An example of this might be seen if the agent suddenly comes to a halt when processing sensory data.

2.1.2 Believable Models In Academia

There has been extensive research in the area of modelling believable background characters in open world games. Some examples include implementing these characters with emotions or equipping them memory.

Emotional Models

An emotional model is one in which the NPC makes its decisions based on their current emotional state as opposed to a pre-defined behaviours e.g., the characters eat when they are hungry instead of eating at a specific time.

Chaplin et al presented an emotional model for NPCs in role-playing games, where the model is implemented with emotions e.g., anger and happiness. These emotions are modelled using drives such as energy, warmth, and sociability. Furthermore, the Iterated Prisoners Dilemma (IPD) is used to simulate interactions between agents [13]. By making characters' decisions based on their drives and emotions, it makes them exhibit human-like behaviour and improves upon the believability of current NPCs.

Past IET dissertations, including Fallon [17] and Mullaly [35], have explored the use emotional models specific to the background characters in open world games. These characters use emotional states to determine their behaviours. Traits were also introduced which gave the NPCs a unique personality. An example of a trait that can be given to an NPC is the energetic trait, where the character has more energy than typical NPCs [35]. By utilising an emotional model, the actions of the NPCs are influenced

by their emotional states, as opposed to pre-designated actions.

Although these works enhanced upon the believability of current NPCs in games, the models did not allow environmental and temporal factors to affect the emotions or behaviours of the NPCs. Such models, however, provide interesting approaches to developing believable agents and could be extended to incorporate these factors. For example, the current weather could affect the characters mood, such as rain making the NPC upset.

Memory Models

There have been many recent attempts to provide NPCs with memory so their interaction with the environment can be improved. Examples include the work carried out by Li et al [27], Lim et al [28], and McNulty [32]. The model proposed by Li et al aims to advance current NPCs by having them use sensors and memories to perform more interesting interactions with the objects and agents of the environment. The model also supports various psychological activities such as forgetting concepts and creating false memories. Lim et al presented a complex model for developing intelligent NPCs which takes into account multiple elements including perception, motivation, emotions and memory. This model focuses on developing NPCs for an educational role-playing game which deals with the cultural-awareness problems for children. McNulty introduced a memory model for background NPCs in open world games that allows the characters to remember and recall the state of the virtual world around them. This memory model can be used to create interesting behaviours, such as a hunter remembering the best locations to hunt animals based on past experiences.

2.2 Current AI In Open World Games

The AI used in open world games differs greatly to other genres. Due to its large scale environment, both in terms of the worlds size and the amount of NPCs that inhabit it, there are limitations on the complexity of the AI. Having complex AI for all the NPCs is not a viable option due to the performance issues that would arise. As a result, these NPCs usually consist of a simple finite state machine for their states and path finding to navigate through the environment. While this provides efficiency, it does

result in repetitive behaviour among the NPCs as they will perform the same tasks over and over again. With the advances in technology, developers are attempting to create much more believable and interesting NPCs by making them more dynamic. Some modern open world games which have done so include Skyrim, *Fable II* [46] and *S.T.A.L.K.E.R* [55].

2.2.1 S.T.A.L.K.E.R

S.T.A.L.K.E.R is a series of open world first person shooter game developed by GSC Game World. The first game of the series was developed in 2007. In the S.T.A.L.K.E.R series, the player explores a region called 'The Zone'. This area is the result of a second explosion at the Chernobyl Nuclear Power Plant, which causes strange changes to the surrounding area. The NPCs in this game use a system called A-Life. The goal of A-life in S.T.A.L.K.E.R is that the characters in the game live their own lives and exist all the time, not only when they are in the players field of view. The character AI consists of three layers:

1. collection and processing of information
2. making decisions
3. a set of low-level controllers

The character AI collects and processes information using receptors. Each character has three types of receptors: visual, sound and damage and uses these receptors to choose interesting parts of the world such as threats to the NPC and objects that can be picked up. In order for the NPCs to make decisions and plan their next actions, A-Life uses GOAP. The properties of world are continually queried and the planner is used to create the current plan based on these properties. A set of low-level controllers are also used which are responsible for elements such as movement, sound, animation and controlling objects [12]. Combining these three layers gives the player a sense that the NPCs have human-like behaviour.

2.2.2 Fable II

Fable II is an open world game, developed by Lionhead Studios in 2008, where the player and his/her companion dog complete quests throughout the land of Albion. The AI characters for Fable II followed three rules:

1. Not aggravate the player
2. Focus around the player
3. Look after itself.

The NPCs of Fable utilise an opinion system of the player. Depending on the deeds the player does throughout the game, it will influence the NPC's opinion of them. The villagers' detailed opinions of the player include three spectrums which are hate/love, frightening/funny and ugly/attractive [5]. These opinions can be influenced in different ways such as changing appearance or giving them gifts. The Fable opinion system is used to try and immerse the player in the game, by having them feel like the world is aware of their actions.

2.2.3 Oblivion, Skyrim, And Fallout: New Vegas

The system used in all three of these open world games is called Radiant AI. It was first present in *The Elder Scrolls IV: Oblivion* [43], the predecessor to Skyrim. Using this system in Oblivion went a long way toward making the NPCs act in realistic ways. The NPCs would carry out daily activities such as sleeping, eating or working their respective roles such as farming the land. Despite this advance in making NPCs more believable, the characters were limited to around five tasks and there was no real dynamic behaviour [8].

Fallout: New Vegas [16] also employed the use of Radiant AI. It improved upon the Radiant AI used in Oblivion through two key concepts: karma and reputation. In Fallout, karma is used to reflect the status of how the player is received by the inhabitants of the Wasteland. The player can be viewed as being good, neutral, or evil depending on the choices he/she makes when progressing through the game. The NPCs will behave differently to the player on their karma, such as having different dialogue options [1]. Reputation is also used alongside karma. This is most notably

used among the different factions in the Wasteland. Depending on the good or bad deeds the player does for these groups of people, either fame or infamy will be gained. The player's reputation with any given group is a measure of the combined total of fame and infamy that has been earned. Based on the player's reputation with these factions, they may receive benefits or drawbacks such as cheaper merchandise or the faction might hire "thugs" to attack the player [3]. Karma and reputation brought an interesting new aspect to the players gameplay experience, as they needed to be more aware of their actions due to the consequences that may arise in the game world.

For Skyrim, the Radiant AI technology was improved in many ways. Firstly, each NPC performs tasks that make sense in their environment. To further improve this, the cities were populated with utilities such as mines, farms or mills so that the NPCs had believable tasks with respect to their environment. Another improvement of the Radiant AI is that the NPCs are more aware of the player. For example, if the player drops an item an NPC who sees the player do it might ask "Why did you do that?" or if the player commits a crime against an NPC, they might hire thugs to teach you a lesson. Not only does this system improve the NPC's behaviours, it also produces radiant or dynamic quests for the player and have the player fight enemies suitable for their skill level [2, 8].

Despite the advances in Radiant AI, there are still a number of issues which take away the believability of the AI. One such issue is that the NPCs are fully scheduled, as they will perform a specific action at a given time. At each hour of the day, they are given a specific task to perform and this never changes. Furthermore, temporal and environmental factors do not have any effect on the NPCs decision making process. This does not exhibit human-like behaviour and takes away from the believability of these characters.

2.3 Sensing In Games

Sensing is a method used in game by AI in order to gain knowledge about the environment. The three main senses that are used in games are sight, touch and sound. The sense of smell could also potentially be used, but it is relatively unexplored in games. The first and most often used sensory modality is sight. In order to represent sight, a view cone of 60° is generally used, which is the area in which the NPC has visual

perception. A number of factors often affect this sight cone, including line of sight, distance and brightness. The second sensory modality is touch. Touch is modelled in games by a physical collision, where a character is notified by a collision with another. The final sense is sound. NPCs use sound in order to detect noises within a given distance, such as the player walking or objects being dropped [34].

Sensing is a key component in making NPCs environmentally aware. Environmental awareness for background characters refers to their ability to perceive and/or react to the changes in external environment. Currently, background characters have a limited awareness of their environment in open world games due to the added complexity and performance issues that can arise. Attempts have been made to improve the believability of background NPCs in open world games by making them aware of the environment they inhabit. Ijaz et al presented a model which aims to improve upon current NPCs through incorporating believable conversation in games. The NPCs attempt to integrate these conversations by being aware of objects and other NPCs in the environment and using this in their conversations [24]. While the NPCs utilises the environment for conversational use, the environment does not play a role in the characters' decision making. Ibrahim et al produced a model which used visual perception for NPCs so they could perceive their surroundings. Information is stored about objects in the environment and when the NPCs see this object they receive knowledge about it [31]. Despite the fact that this can be used to produce a more dynamic NPC whose behaviour is not limited to pre-defined actions, it focuses on objects in the environment as opposed to the temporal and other external factors such as weather and the time of day. There are many applications of sensing in games. In Skyrim, the NPCs make use of all three of sensory modalities, in order to produce a more interactive experience with the player. They use these senses for various reasons, such as preventing the player from sneaking up and/or stealing up from them. Sensing adds an extra layer of realism in games as it helps provide a virtual character with the illusion of life. While NPCs in most open world games utilise different senses, they are usually only with respect to the player and often these NPCs do not perceive or react to other elements in their environment.

2.4 Control Theory

Control theory/engineering is concerned with control systems that are used to manage the behaviours of a system or device. These systems iteratively interact with the environment through the use of a sense-plan-act paradigm [39]. Most open world games generally utilize a finite state machine as a control system in order to manage NPCs decision making process. Despite being an efficient method, it also unwieldy and often results in unrealistic NPC behaviour. Furthermore, finite state machines are typically not suitable for dynamic environments. In order for a control system to be able to handle a dynamic environment, the NPCs must be able monitor the environment and be ready to change their plans and actions to suit the changing world [26]. Through its use of the sense-plan-act paradigm, a control system could be used to manage a system involving temporal and environmental cycles which are used drive the behaviours of NPCs. There are two types of control systems: open-loop control systems and closed-loop control systems.

2.4.1 Open-Loop Control System

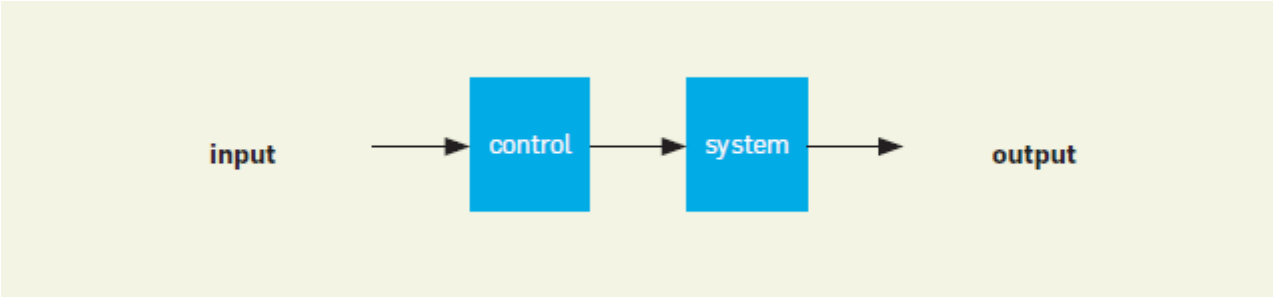


Figure 2.1: Open-Loop Control System [33]

In an open-loop control system (non-feedback system), the control input to the system is determined using only the external input without considering the systems output conditions (figure 2.1). Since these systems do not take into account feedback from the systems output, no comparison can be made between the desired and actual output values. As a result, errors that may arise cannot be corrected, such as external disturbances to the system. Open-loop systems are typically used in systems which require no feedback [33, 51].

2.4.2 Closed-Loop Control System

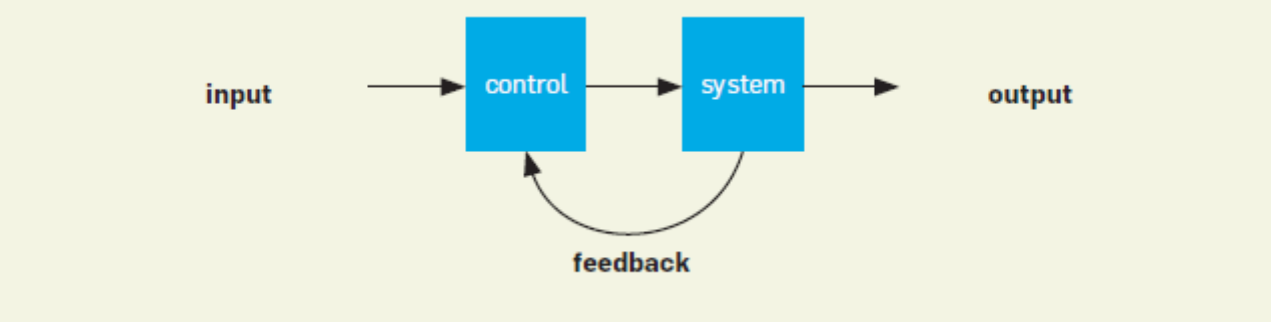


Figure 2.2: Closed-Loop Control System [33]

A closed-loop control system, or feedback control system, is a system which takes into account the current output of the system and feeds this information back to the controller. This approach utilises a feedback system to determine the control input based on a combination of the measurements received from the system's output and the external input. Using this method allows the system to by self-adaptive by constantly correcting and adjusting the system under control. Although closed-loop systems have an extra layer of complexity by having one or more feedback paths, it also has many advantages over open-loop systems, including its ability to minimize the effects of external disturbances on the system as well as reducing errors by automatically adjusting the systems input [33, 50].

2.5 Cycles in Games

Most open world games employ cycles to assist in immersing the player into environment. The primary goal of these cycles is to increase the believability of the world, by allowing the player to associate the virtual world with the real one.

Day/Night Cycles

Day/night cycles are used in almost all open world games. These cycles enable the use of time in the virtual world, which can be used for various aspects of the game.

For example, it allows for events and characters behaviours to be scripted, to simulate interesting and realistic behaviours. Scripting allows background NPCs to exhibit human-like behaviours, by following a schedule. For example, the NPC performs actions during the day, such as working, and then sleeping at night. However, this approach has its limitations as the NPCs repeat the same actions at the same time each day, which does not truly represent human-like behaviour. Some games have attempted to utilise these day/night cycles to create more interesting behaviours such as in *Dying Light* [47], where the zombies which inhabit the environment become faster and stronger at night. This makes for a more interesting gameplay experience as the player has to be aware of the time of day in the game.

Despite attempts made to use these day/night cycles in games for improved behaviour for AI, they are mostly used for visual effects as opposed to providing sleeping patterns for NPCs. One possible solution for introducing sleeping patterns is through the modelling of circadian rhythms onto NPCs. In order to adapt to the alternation of day and night, most living organisms exhibit daily periodic oscillations in their biochemical or physiological behaviour, known as circadian rhythms, which are driven by circadian clocks [22, 25, 56]. A fundamental aspect of circadian systems is entrainment, which helps maintain a relationship between the circadian system, the timing of sleep and wakefulness, and environmental time (24-hour day) in order to remain awake during the day and asleep at night. These circadian rhythms are primarily influenced by the perception of light and darkness in the environment [25]. When we sense light, a signal is sent to our brain to help us remain awake. If we sense darkness, however, the body produces melatonin which makes us drowsy and in turn helps us to fall asleep. There are a number of factors which disrupt your circadian rhythm, such as seasonality. In winter, for example, the lack of morning light may result in trouble sleeping at night and then waking in the morning [38].

Dynamic Weather Systems And Seasonal Cycles

Dynamic weather systems, like day/night cycles, play an important role in immersing the player into the game world. Having a variety of dynamic weather effects, such as rain and snow, provides a more accurate representation of the real world, as opposed to the weather being sunny, for example, throughout the entire game. Seasonal cycles

are rarely used in open world games. One of the few games to implement this feature is *Shenmue* [41]. Shenmue utilised this feature to create a more realistic weather pattern, where the type of weather was influenced by the season. Although the NPCs behaviour was not affected by these seasons as they continued their daily scripted actions, it did allow for interesting behaviour. One example of this is that when it rains the NPCs will equip an umbrella, as if they are aware of their environment [4].

An underlying problem with these cycles is that they have the same problem as day/night cycles in which they are typically only used for visual effects, as opposed to influencing the behaviour of the NPCs that exist in the world. Having the NPCs perceive their environment is a possible solution to this problem, such as perceiving the current weather in order to determine their next action. This could be used to simulate interesting behaviours such as rain affecting the NPCs current mood or high/low temperatures affecting their ability to work.

Chapter 3

Design

This chapter presents a design for the project. The major components, of which the model is composed of, are discussed. Once the model and its underlying concepts have been clearly defined, some example NPCs are described to illustrate the model. Finally, a high-level architecture of the system is presented.

3.1 Previous Work

As mentioned in section 1.4, the model will build upon the work of previous IET students John Fallon [17] and Tiarnán McNulty [32]. The emotional model of Fallon provides an ideal starting point for the design of the model outlined in this project.

3.2 Defining The Model

In order to increase the believability of background characters in open world games, it is essential that they move away from following scripted routines. The model defined in this dissertation attempts to do this, by having the characters' decision making behaviours influenced by their everyday tasks and the various cycles found in the game world.

3.2.1 Global Time Manager

Almost all aspects of the model are time-dependent and so it is necessary to have a global time manager which maintains the in-game time. This time manager keeps track of a number of in-game elements including time of day, the current phase (dawn, day, dusk, night), the date, and seasonality. By keeping track of these elements, it allows for cycles to be modelled more realistically and dynamically. Furthermore, the game itself, and the characters in it, can use the time manager in order to determine when certain actions or events may occur.

3.2.2 Modelling Cycles

In open world games, cycles are often used for assisting the player in becoming immersed in the game world. Although they provide for a more engaging experience, this is typically only achieved through their visual effects. By having the cycles affect the background characters in the world, it results in an increased purpose for the cycles of these games. With this in mind, it is important to model cycles in a realistic manner. If the cycles are modelled using complete randomness, it will lead to unrealistic outcomes, e.g., sporadic changes in temperature or weather of the world. As a result, this could lead to unrealistic behaviours of the background characters. To overcome this problem, the cycles are dependent upon the global time manager, or related to the time manager in some way. This allows for cycles to be modelled in a time-dependent manner. For example, the temperature of the world will depend upon the current season. Additionally, modelling cycles as being time-dependent allows for unique behaviours during different times of the year. The list of cycles can be seen in Table 3.1, along with a description of each cycle and their dependencies.

Table 3.1: Cycles

Cycle	Description	Dependencies
Day/Night	Determines the transitions between day and night	Time of Day
Temperature	Determines the current temperature	Time of Day, Season
Weather	Determines the current type of weather	Temperature, Season
Lunar	Determines the lunar activity	Time of Day, Date
Environmental	Determines the effects of the region itself	Lunar

Day/Night Cycles

As outlined, all the cycles of the game world should be either time-dependent, or related to time in some manner. The day/night cycles are associated with time based on the current time of day, where the sun comes up during the day and sets at night. To achieve this, a directional light is used to represent the sun, and this directional light is connected to the time of day. With a complete rotation representing 360°, the 24 hour time of day is converted into a 0-1 scale, where 0 represents the start of the day and 1 represents the start of the next day i.e. midnight. By multiplying the current time of day (0-1) by 360°, this returns the correct angle of the directional light, based on the current time.

To have the day/night cycles affect the background characters, it is designed so that the current light intensity of the sun will affect them. The basic formula is:

$$Effect = -\frac{1 - currentLightIntesnity}{DampeningValue}$$

The dampening value is added so that the effect can be easily modified or balanced.

Temperature Cycles

To model temperature cycles in a realistic manner, it is designed so that is both dependent on the time of day and seasonality. Each season maintains a minimum and maximum temperature which it can reach. This provides flexibility as it allows for different temperatures each season. A temperature range is then selected, using probability, based on the current season and the current phase of the day. Given four

temperature ranges of high, mid-high, mid-low, and low, during the day the temperature range will be likely be either high or mid-high. Similarly, during the night, the temperature range will likely be either low or mid-low. Although the temperature range may be high during the day, for example, this does not necessarily result in high temperatures, as it bound by the current season. A high temperature in winter may only reach a value such as 5°. To calculate the values for the minimum and maximum temperatures, for each temperature range, the step size is required. This is achieved using the formula:

$$StepSize = \frac{maxTemp - minTemp}{NumOfTempRanges}$$

An example of the temperature range values can be seen in Table 3.2, where the season is winter, and the assigned minimum and maximum temperatures are -10° and 5° respectively. Using the formula for calculating the step size results in a step size of 3.75.

To model the effects of temperature on the background characters, a minimum and maximum is threshold is maintained. If the temperature falls below or above these thresholds, then the character will be affected. It is designed this way so the temperature will not affect the characters of the world too much. The formula used is:

$$Effect = \pm\frac{currentTemp - thresholdValue}{DampeningValue}$$

Similar to the effects of light, a dampening value is maintained so the effect can be easily modified or balanced.

Table 3.2: Temperature Ranges

Temperature Range	**Min Temp**	**Max Temp**
High	1.25	5
Mid-High	-2.5	1.25
Mid-Low	-6.25	-2.5
Low	-10	-6.25

Weather Cycles

The weather cycles of the world are designed so that they are dependent upon both seasonality and temperature. There are three main weather types that can occur: rain, snow, and sun (the absence of rain and snow). With a view to maintain realistic weather patterns, the model utilises probability. The probability of a given weather type occurring is determined by first assigning them a base probability, which relies on the current season. Following this, a probability modifier is added, which increases or decreases the base probability of each weather type. This is reliant on the current temperature of the world. An example result of this is that the season of winter, and the low temperatures it brings, will significantly increase the probability of snow arising.

Having the weather cycles affect the character is achieved by taking into account two factors: the current weather and whether or not the character is indoors or outdoors. This design means that characters who are indoors will not be affected by the effects of rain or snow, which will influence their decision making behaviours. For simplicity, both rain and snow will affect the characters based on a user-defined value.

Lunar Cycles

The lunar cycles are designed so that they indirectly affect the characters of the world. Based on the current lunar phase, e.g., full moon and half moon, it can lead to dynamic events being developed. This dissertation uses lunar cycles to affect parts of the environment, as opposed to directly affecting the characters.

Environmental Cycles

The notion of environmental cycles revolves around the idea that the region itself may change over time and affect the background characters in some manner. To showcase this, an enchanted forest was designed which affects the characters based on their proximity to the region. The effect is dependent upon the current lunar cycle, where the forest will only light up and affect the character during a full moon. To calculate the effect of the forest, the same formula that is typically used for the effect of grenades in games can be applied. This is where the effect falls off based on the distance to the centre of the event. The formula is as follows:

$$Effect = 1 - \frac{distanceToForest}{radiusOfEffect}$$

3.2.3 Emotional States

In order to model the effects of cycles on the characters, an emotional model is employed. This involves assigning the characters emotional states, such as mood and energy, and having these states being affected by both the actions of the character and the cycles of the world. To achieve this, each character is equipped with four emotional states: mood, energy, satedness and sanity, as shown in Table 3.3. Each state's value is limited between -1.0 to +1.0, meaning that any given state can range anywhere between fully satisfied to completely dissatisfied. Due to the simplicity of the state model, additional emotional states can be easily added at any time.

The decision to equip the characters with emotions, and its design, was strongly influenced by the work of John Fallon, who attempted to develop an emotional model for the background characters in the Skyrim. Initially, the design was to simply have the cycles influence the characters task selection. So, for example, if it suddenly started to rain, the character might immediately seek shelter. However, it was later decided that a more suitable design would be to equip the background characters with emotional states. This results in a less forcing model, while also providing an explanation behind choosing certain actions.

Table 3.3: Emotional States

-1.0	**State**	**+1.0**
Sad	Mood	Happy
Tired	Energy	Rested
Hungry	Satedness	Full
Insane	Sanity	Sane

In order to affect the emotional state values, a modify state value function was designed to allow both the cycles of the world and the tasks the characters can carry out (see Section 3.2.4) easily modify these values. Passed into this function are two parameters: the state being affected and the modification amount. Once these values are passed into the function, the state values are modified using the following formula:

$$Clamp(modificationState.Value + (modificationAmount * timeDifference))$$

Firstly, the modification amount that is passed in gets multiplied by a time difference, in order to spread out the value over a period of specified in-game time. This allows a value such as 0.2 be passed in, and the state which is being modified will gradually decrease over the specified period, e.g., one hour. This value then gets added to the current state value and is finally clamped between the values of -1.0 and +1.0.

3.2.4 Tasks

Each character maintains a list of tasks or actions that they can carry out. While most of these tasks were already available with the environment Tiarnán McNulty developed, some additional tasks, such as finding shelter, were introduced to add more variety. Each task the character carries out will either positively or negatively affect the character's emotional state values. With a view to maintain believable times that an activity can be carried out, each task is assigned an activity time. The purpose of activity times is to help maintain realism in the game world. If the characters sleep during the day or work at night, this can slightly take away from the believability of these characters. The use of activity times limits the periods of time in which activities can take place, while also encouraging a the character to select certain tasks. The complete list of available tasks can be seen in Table 3.4.

Table 3.4: Tasks

Task Name	Activity Times	Mood	Energy	Satedness	Sanity
Blacksmith	Work	-0.2	-0.3	-0.2	0.0
Lumberjack	Work	-0.2	-0.3	-0.2	0.0
Woodcutter	Work	-0.2	-0.3	-0.2	0.0
StableWorker	Work	-0.2	-0.3	-0.2	0.0
Shopkeeper	Work	-0.2	-0.3	-0.2	0.0
Fishing	Work, Sleep, Free	0.2	-0.2	-0.2	0.0
Sleep	Sleep	0.1	0.4	-0.2	0.0
Shelter	Work, Sleep, Free	0.1	0.2	-0.1	0.0
Eat	Work, Sleep, Free	0.1	0.0	1.0	0.0
Idle	Work, Sleep, Free	0.1	0.3	-0.1	0.0
Talk To Self	Work, Sleep, Free	0.0	-0.1	-0.2	1.0
Wander	Work, Sleep, Free	0.2	-0.1	-0.2	0.0

3.2.5 State Manager

The State Manager is responsible for selecting the character's next task, based on a specified update frequency. After the states have been modified, by the character's current task and by the effects of the cycles, the state manager will evaluate the character's current emotional state. This results in a list of possible tasks that the character can carry out. A check is then made to ensure that certain tasks cannot be selected based on their activity times (see 3.2.4). This is achieved by comparing each task in the list of possible tasks to those in a list of unselectable tasks. If the task is found in the unselectable tasks, then it is removed as a possibility. From the list of remaining possible tasks, a task will be chosen, using probability, that will improve the character's lowest state. In selecting this task, it will also take into account the current conditions of the game world, and whether or not the task will actually improve the lowest state. To do this, it will go through the list of possible tasks and add the lowest state's current value to the amount that the task will affect that state (see Table 3.4). If the result is higher, then that task can still be selected, otherwise it is removed from the list of possible tasks. A high-level workflow can be seen in Figure 3.1, which shows how the

state manager chooses the best task for the character.

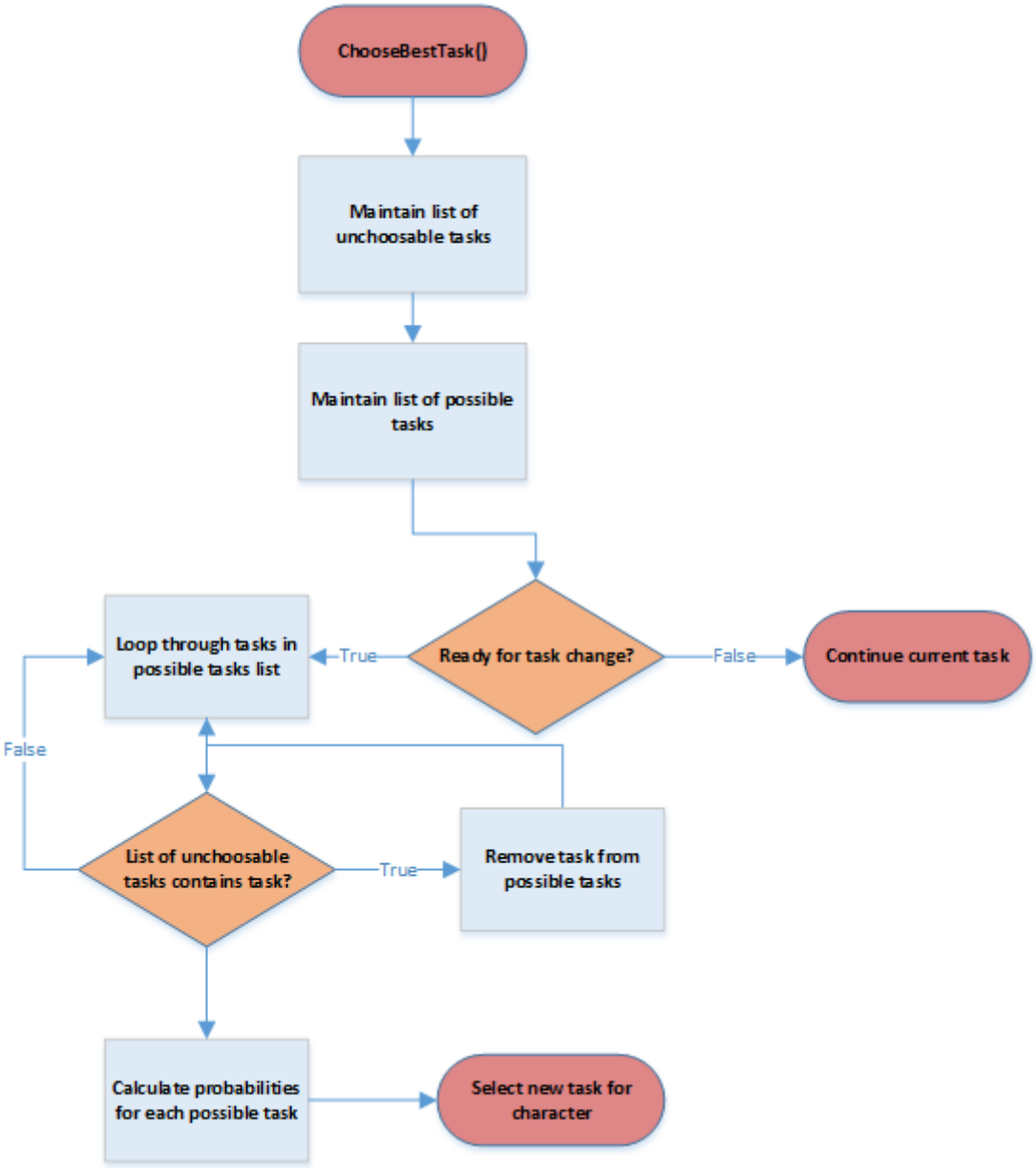


Figure 3.1: State Manager Task Selection

3.2.6 Traits

In most open world games, the world is populated with characters of various different races, such as humans, elves, and orcs. Having the environmental and temporal factors of the world influence the behaviours of these races differently adds a further element of believability to these characters. The use of traits allows the model to accommodate the characters of different races, by providing them with distinct characteristics. Each trait will determine whether or not a certain cycle of the world will affect the character. For example, the Nocturnal trait will result in a character sleeping during daylight and coming out at night. This leads to more dynamic characters that the player may encounter while playing the game. In order to model these traits, simple true or false boolean logic is followed. This is determined by whether or not the character is assigned the trait. Table 3.5 presents a list of possible traits the character can be equipped with, along with a brief description of these traits and the states which are affected, or unaffected, as a result.

Table 3.5: Traits

Trait	**Description**	**Effected State(s)**
Nocturnal	The NPC sleeps during daylight	Energy
Cold-Blooded	The NPC is affected more by temperature	Mood, Energy
Amphibious	The NPC is unaffected by rain	Mood
Chionophile	The NPC is unaffected by snow	Mood
Lycanthrope	The NPC is affected by lunar cycles	Sanity

Example NPCs

With the addition of traits to the emotional model, a number of different characters could arise in the game world. One example is a traditional "Nord" or human character, which might not have any traits associated with them. As a result, the cycles will have a normal affect on this character, e.g., rain and snow will drain their mood and the absence of daylight will drain their energy (to encourage them to sleep). Another example is an "Amphibious Argonian" (lizard) character, which is assigned the cold-blooded and amphibious traits. This results in high and low temperatures having greater affects on this character, but they will remain unaffected by rain. A final exam-

ple is a “Nocturnal Orc”, which is assigned the nocturnal and chionophile traits. With these traits, the character will be inclined to sleep during the day and carry out other activities at night, as well as remaining unaffected by snow. By simply adding a few traits to these characters, it gives a sense that the characters have unique personalities in comparison to each other. Furthermore, it could also lead to more interesting gameplay for the player, where they comes across different types of characters at different time periods of the game. For example, following a change in season, the player may come across different types of characters who have been assigned different traits and similarly meet distinct and interesting characters during both the day and night.

3.2.7 High Level Architecture

The high-level architecture of the model can be seen in Figure 3.2, which summarises the main requirements of the system. The effects from the both the character’s current task and the cycles of the world are passed into the state manager as a form of input. These values will modify the character’s emotional state values accordingly. After the states have been modified, the state manager will evaluate the character’s current emotional state, resulting in a list of possible tasks that the character can carry out. From this list, a task will be chosen using probability, while taking into account the current conditions of the game world. For example, if it is currently raining and the temperature is low, the character will be encouraged to select an indoor activity. Once a task is selected by the state manager, it is output to the character so they can carry it out.

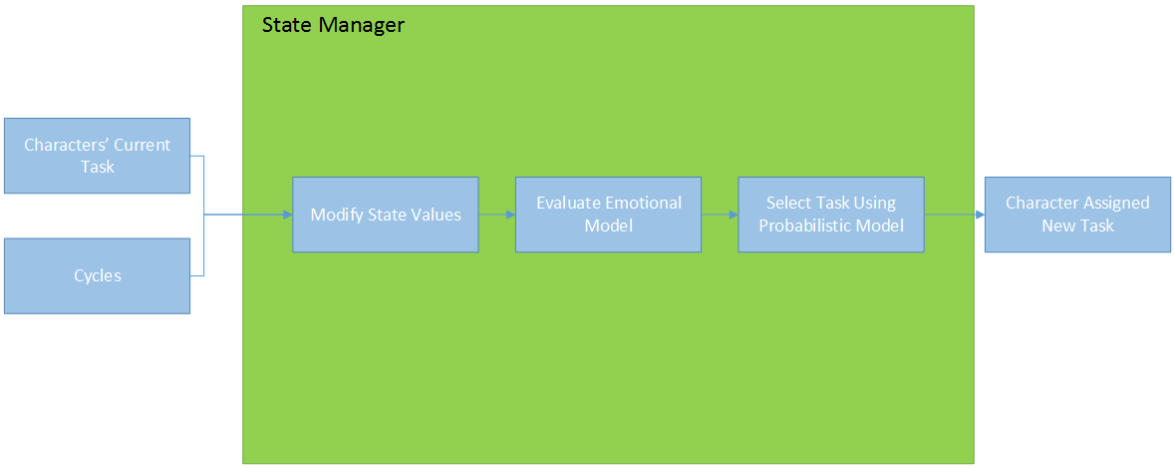


Figure 3.2: High Level Architecture

Chapter 4

Implementation

This chapter presents the implementation of a prototype, which is developed in order to demonstrate the concepts of the model. The choice of platform is mentioned, followed by a discussion on how the model was implemented.

4.1 Platform Selection

The platform selected for this dissertation was Unity3D [52]. Tiarnán McNulty [32] had previously developed and populated a village in the Unity game engine as part of his dissertation. By using this village as a starting point, it saved a lot time that would have been otherwise spent either creating an environment, or perhaps trying to integrate the model into Skyrim using their modding kit, the Creation Kit.

4.2 Implementing The Model

This sections concerns the implementation of the model and its components in order to visually demonstrate the concepts of the proposed model.

4.2.1 Creating The Cycles

By implementing the cycles of the game world visually, it allows for the concepts of the model to be seen more clearly. Additionally, it also helps in terms of debugging the model. For these reasons, the cycles were implemented into the game world. This

section outlines how the visual aspects of these cycles were implemented. The different cycles and their description can be seen in Table 3.1.

Day/Night Cycles

With the introduction of procedural skyboxes in Unity 5, this allowed for the visual representation of the day/night cycles to be fairly easily implemented. There are a number of variables which can be modified to change the visual effects of the sky itself, including the atmosphere thickness and the sky's color by modifying its tint value. Additionally, the procedural skybox allows for the sun rotate across the world by creating a directional light and rotating it, where 360° represents a complete rotation. As the directional light is rotated, the procedural skybox, provided by Unity, automatically handles the visual effects of the sun. All that was required was modifying the variables of the sky, and modifying the light intensity of the sun based on whether or not it has set. If the sun is above the horizon, its light intensity is set to 1, otherwise it is set to 0. This results in the environment becoming darker at night, and brighter during the day. Figure 4.1 shows the visual sunrise of the game world.

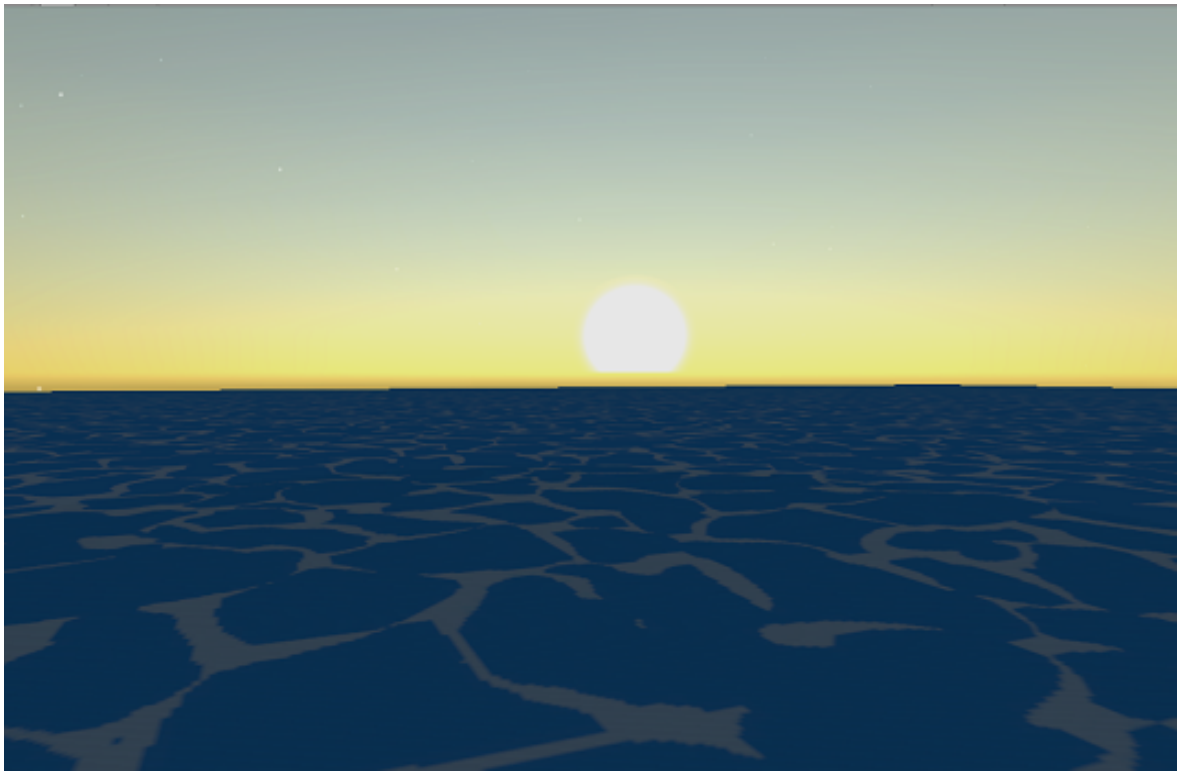


Figure 4.1: Day/Night Cycle Sunrise

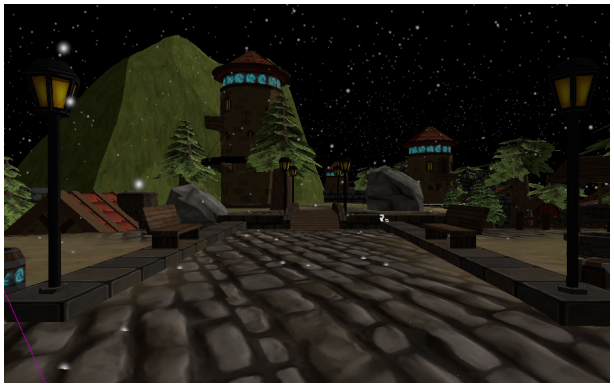
Weather And Temperature Cycles

As temperature cannot be modelled in terms of visual effects, or at least modelled with relative ease, a text based visualisation was implemented. To show the temperature cycles of the game world, the current temperature is output as text on the screen.

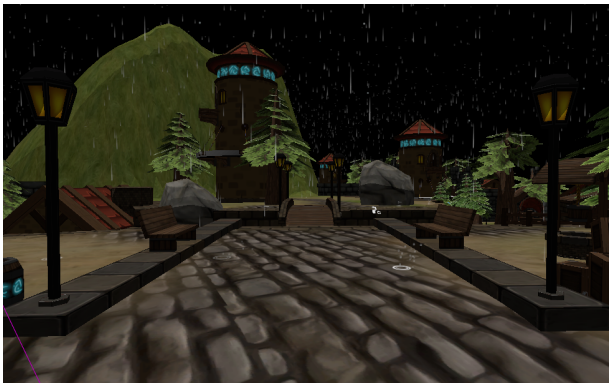
Both the rain and snow effects implemented for this project utilise Unity's particle system. This involves altering a number of parameters to get the desired results. Table 4.1 outlines some of the chosen parameters, and Figure 4.2 shows an in-game example of these weather effects.

Table 4.1: Particle System Parameters

Weather Effect	**Max Particles**	**Size / Shape**	**Sub-Emitters**	**Render Mode**
Snow	40000	Box, 500 x 300	None	Billboard
Rain	10000	Box, 500 x 300	Splash, Ripple	Streched Billboard



(a) Snow



(b) Rain

Figure 4.2: Weather Cycles

Lunar Cycles

Although Unity 5 introduced procedural skyboxes, which handles the visual effects of the sun, the same does not apply for the night sky. There are no lunar elements of the default skybox, and so the night sky had to modelled manually. To model the night sky, both stars and a moon were added to the world. For these components to rotate around the world, both of these effects utilise the same directional light calculation used for the day/night cycles. The stars' rotation is equal to the sun's rotation, but will only be visible during the night. The moon's rotation around the world is to equivalent to the negated sun's rotation. This results in the moon rising as the sun sets on the world. Figure 4.3 shows an in-game example of the night sky.

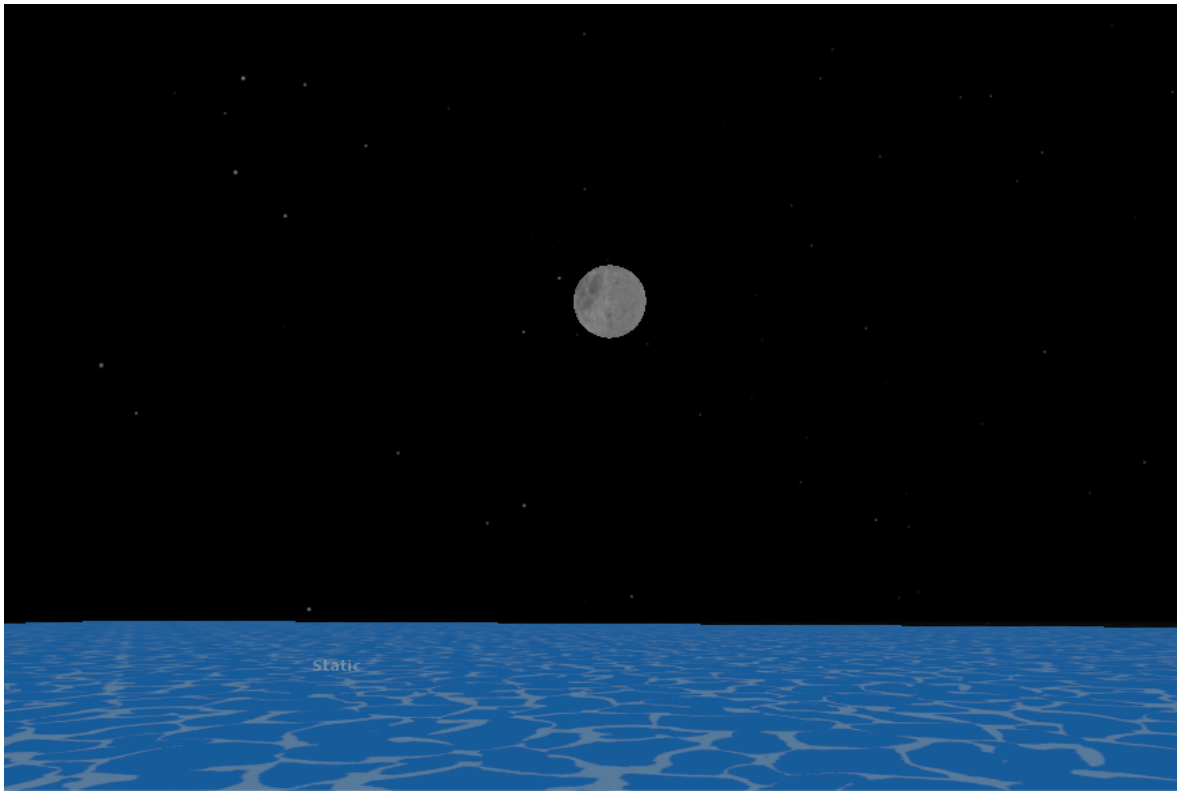


Figure 4.3: Lunar Cycle

Environmental Cycles

The idea behind environmental cycles is that the environment itself can affect the characters of the world. When a character enters a certain region, they would be affected in some manner. To demonstrate the effects of the environment, an enchanted forest was implemented. This involved adding a number of trees, bushes and particle systems to an area of the village. In the scenario of a full moon, the enchanted forest will light up using these particle effects. If the current lunar phase is not a full moon, the forest will not light up or affect the characters. Figure 4.4 shows the implemented forest during a full moon.

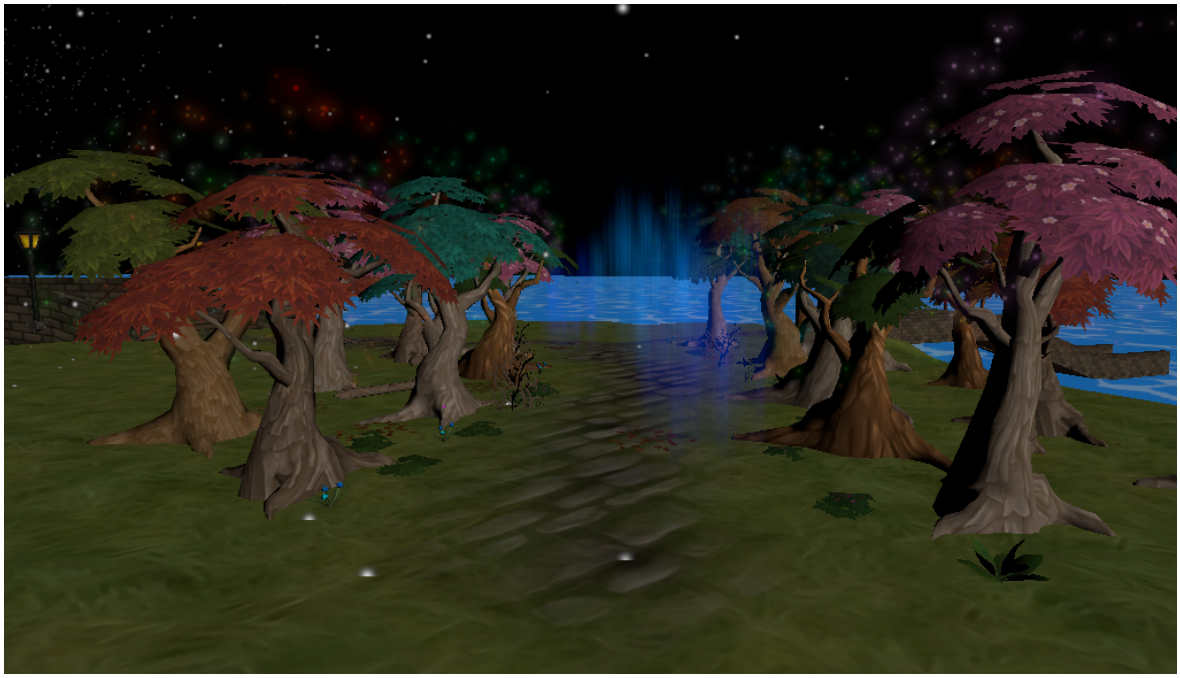


Figure 4.4: Enchanted Forest During A Full Moon

4.2.2 Integrating The Model

Since the model proposed in this dissertation is for use in open world games, the model was integrated into a Unity game environment developed by Tiarnán McNulty. Most aspects of the world were dependent upon the use of the memory model outlined in his project, where the characters carried out tasks and triggered certain events based on memory. It was realised, however, that the memory model conflicts with the model being developed in this project, and had to be disabled. Consequently, a number of features in the game world, such as the inventory system and the task or action system no longer worked, and had to be modified or reimplemented. Furthermore, the task selection model provided did not function correctly and had to be implemented from scratch.

While the memory model had a number of tasks developed, additional tasks were required to showcase the model in this project. This includes a shelter action, where the character will take shelter under certain conditions. To achieve this, each NPC is assigned a sphere collider, and when this sphere collider enters the collider of any

specified shelter location, the character will be marked as being sheltered. When the sphere collider leaves the location, they will marked as being outdoors. The character will choose a shelter location based on their proximity to each possible shelter location.

An important feature implemented was a way of visualising the overall model. This is important to being able to showcase the model, while also providing assistance in terms of debugging. To accomplish this, each character is assigned a series of graphics to display different information. Above the character, the current task is displayed. The emotional states, and their values, are displayed one side of the character as well as the current effects from cycles on the other side. While not as an important feature, each agent maintains an inventory of items. Based on the current task, they will equip the appropriate item, e.g., when working as a blacksmith they equip a hammer. Figure 4.5 shows an NPC who is currently working as a blacksmith.

Figure 4.5: Model Visualisation

4.3 Extensibility Of The Model

The following section details some of the implemented features which allow for extensibility of the model.

4.3.1 Cycle Manager

A cycle manager class was implemented which keeps track of and updates all the cycles in the game world. This is achieved by maintaining a list of cycles, which also allows new cycles to be added easily. Each new cycle created extends a base cycle class, and inherits the time manager as a result. While a generic solution to modelling cycles was not found due to time-constraints, an object-oriented approach is in place to achieve this in the future.

4.3.2 Tasks

To avoid having to manually code the affects of each action on each of the emotional states, a text file is maintained with these values. By maintaining the tasks and their effects in a comma-separated text file, it allows for extensibility, where new tasks and their effects can be easily added. Each line of the file follows the example format:

$$Sleep, Mood, 0.1, Energy, 0.4, Satedness, -0.2, Sanity, 0$$

where the first value is always the action, and the subsequent values are state and modification values.

4.3.3 Tunable Parameters

To allow for extensibility of the model, most variables developed in this project can be modified using Unity's inspector. By modifying these parameters, it is possible to improve / change upon the model in the future. Some tunable parameters include the following:

- **Time Manager:**
  - *Current Time/Date*: The current time and date of the world.

- *Day Length*: The length of a day.
- *Time Multiplier*: The speed of the time manager, which is mainly used for debugging purposes.

- **Cycles:**
  - *Effect Values*: The base effect value for the cycle, although this may be altered under certain conditions.
  - *Weather/Temperature Update Frequency*: How often to update the current weather and temperature of the world.
  - *Minimum/Maximum Temperatures*: The minimum and maximum temperatures of the world for each season.
  - *Lunar Frequency*: How often a full moon occurs in the game world.
  - *Enchanted Forest Radius*: The radius of effect of the enchanted forest.

- **State Manager:**
  - *NPC Tasks*: A list of tasks than an NPC can carry out.
  - *Traits*: A list of traits the NPC has.
  - *Minimum threshold*: A threshold used in certain circumstances where a new task will only be selected if a state value falls below this threshold.

Chapter 5

Evaluation

5.1 Improvement Over Existing Approaches

The primary objective of this dissertation is to step away from the scripted behaviours of background characters in open world games, in order to improve upon their believability. The model proposed in this dissertation achieves this by equipping the NPCs with emotional states. Having their decision making behaviour influenced by these states leads to more unique characters and gives a sense that these characters are both aware of their actions and of the different cycles of the world. Although it can be agreed that avoiding the static approach of using predetermined actions increases the believability of these characters, it is still difficult to determine if the model is believable or not. As the notion of believability is mainly subjective, some people may judge an NPC to be believable, whereas others may disagree.

A problem which could arise is determining whether or not player is believable from an in-game perspective. Without the graphic visualisations present in this model, such as the emotional states and the effects of the cycles, it may be difficult to understand why the character is carrying out a certain action. One possible solution to solve this is through the use of gestures and conversations, where the player can visibly see the player's mood or feeling or possibly interact with these character's to find out what the NPC is thinking or how they are feeling.

Tables 5.1 below outlines an example simulated day that was carried out by a Nord character, who has no traits assigned. The simulation is carried out during summer.

Looking at the schedule, it appears that the character lead a plausible day.

Table 5.1: Nord Simulated Day

Time	**Task**	**Lowest State**	**External Effects**
0:00	Sleep	Energy	Day/Night
1:00	Sleep	Energy	Day/Night
2:00	Sleep	Energy	Day/Night
3:00	Sleep	Energy	Day/Night
4:00	Eat	Satedness	Day/Night
5:00	Sleep	Energy	Day/Night
6:00	Idle	Energy	None
7:00	Idle	Energy	None
8:00	Eat	Satedness	None
9:00	Blacksmith	Energy	None
10:00	Blacksmith	Energy	Temperature
11:00	Blacksmith	Energy	Weather, Temperature
12:00	Idle	Energy	Weather, Temperature
13:00	Blacksmith	Mood	Temperature
14:00	Blacksmith	Mood	Temperature
15:00	Wander	Mood	Temperature
16:00	Eat	Satedness	Temperature
17:00	Idle	Energy	Temperature
18:00	Fishing	Mood	Temperature, Day/Night
19:00	Idle	Energy	Temperature, Day/Night
20:00	Shelter	Energy	Weather, Temperature, Day/Night
21:00	Idle	Energy	Rain, Temperature
22:00	Eat	Satedness	Temperature, Day/Night
23:00	Sleep	Energy	Temperature, Day/Night

5.2 Support For Various Character Races

The introduction of traits, presented in Section 3.2.6, allows the model to accommodate the characters of different races, by providing them with distinct characteristics. The traits were developed to specifically achieve this goal by determining whether or not a certain cycle of the world will affect the character. This leads to more dynamic characters that the player may encounter while playing the game. The example NPCs in Section 3.2.6 provides some examples of the different kind of characters which could arise using these traits. While more work could be carried out to further improve the support for different character races, the traits system provides an ideal starting point.

5.3 Efficiency Of Implementation

Maintaining an efficient implementation of the model is very important for open world games. As each region in an open world game often has a large number of NPCs, which can be visible at any time, the performance must not affect the gameplay.

Efficiency Of Design

With maintaining an efficient implementation in mind, the design and implementation of certain features changed greatly. One example is the design of calculating the affects of rain and snow on the characters. Initially, it was designed so that only a physical collision between the rain or snow and the NPCs would affect them. However, it was realised that the collision detection between each particle, e.g., 40000 individual particles for snow, and each individual NPC was far too expensive. As a result, it was decided that a boolean value is maintained to check whether or not it is raining or snowing. In addition to this, a boolean value is also kept to determine if the NPC is indoors or outdoors. This design choice greatly increases the efficiency of the model.

Runtime Performance

For a model to be deemed efficient, it should maintain minimal memory and processing requirements. Unity provides a profiling tool which shows the different memory and CPU requirements for the objects in the scene, on a frame by frame basis. While the

profiling tool breaks down the CPU requirements of each individual object in the scene, it does not do the same for memory requirements, as it only presents the total memory allocated. This is problematic, as it does not allow analysis of the model's memory requirements. As a result, only the CPU requirements analysis was carried out. Table 5.2 shows the results obtained using Unity's profiler. This takes into account the state manager updating and the affects of cycles. While the model runs efficiently, there are currently optimisation requirements for the State Manager. On some occasions, the CPU time required suddenly spikes a lot higher than average. Further analysis on the State Manager is required to solve this issue. Other aspects of the model, the global time manager and the cycle manager, were also analysed. The average result for these were approximately 0.07 ms and 0.04 ms, respectively, of CPU time.

Table 5.2: CPU Usage Time

Number of NPCs	**CPU Time (ms)**
3	0.14
6	0.34
12	0.56
24	0.9
48	1.53
96	3.6

5.4 Impediments

As mentioned in section 1.4, the implementation aspect of this project is built upon the work of previous IET student Tiarnán McNulty. The environment developed, and its NPCs, provided an excellent starting point to showcase the concepts of the project. However, a large amount of time was spent trying to understand how the system he developed works, most notably the memory model and the task selection model. As outlined in Section 4.2.2, a lot of changes were required to get the model proposed to work in the environment. The effort required in attempting to implement or reimplement a number of features, most notably task selection model, i.e., the State Manager, was substantial and took away from the main focus of this project.

Chapter 6

Conclusion

This final chapter presents a conclusion to the project, summarising the main contributions made by this project, outlining further work that could be undertaken and concludes with some final thoughts.

6.1 Contributions

Although further work may be undertaken, this section highlights the contributions of this project to date.

Building upon the work of past IET students John Fallon [17] and Tiarnán McNulty [32], this project contributes an alternative method for increasing the behavioural fidelity of background characters in open world games. Through the use of an emotional model, which is affected by both the tasks the background characters carry out and the various cycles inherent in the game world, it produces more interesting and believable background characters.

To demonstrate and visually represent the model, a prototype was created and integrated into the game environment created by Tiarnán McNulty. While further work is required to improve upon the model's prototype, the prototype successfully demonstrates the model.

The potential of this model has been highlighted with the creation of various NPCs with different traits, which leads to more dynamic and believable behaviour. The use of traits allows for the creation of unique individuals, while also taking into account

for the multiple races found in open world games.

In addition to improving upon the behavioural fidelity of background characters, the project contributes an improved use of cycles in open world games. The use of cycles in open world games are typically reserved for visual effects and by having the cycles affect the background characters in the world, it adds a further element of importance to the cycles of these games.

6.2 Future Work

There are a number of different ways in which this project could be improved upon or extended with future work.

6.2.1 Improving the Current Model

There a number of ways in which the current model could be improved. Increasing the number of emotional states for the characters and adding more tasks that the characters can carry out would lead to more compelling behaviours.

More dynamic cycles could also be introduced. Although the weather and temperature are affected by the current season, the lunar and day/night cycles are currently unaffected. This could be fixed by having dynamic day and night lengths, and dynamic occurrence of full moons, which are dependent on the current season.

As the State Manager was implemented late (see Section 5.4), there were a number of aspects to the State Manager that could be improved. Tuning how much the tasks and external factors affect the characters' emotional states could increase their believability, as these values were not fully balanced. Furthermore, a more sophisticated method of selecting which task to carry out next could also be developed.

In order to enhance the believability of the characters, conversational tracks and gestures could also be added. This could help engage the player more as it could lead to interesting game play. For example, if it suddenly starts raining, you could watch as the character slowly gets frustrated through the gestures they make.

6.2.2 Integration

Despite the model proposed by this dissertation leading to more dynamic behaviour, as characters behaviours are influenced by their emotional state, the behaviours could be more interesting. One possible solution to this is through integration with other systems, such as the relationship system of Brendan O'Connor [37] and the quest system of Sarah Noonan [36]. While there are a number interesting behaviours that could be developed through the integration of these systems, one example might be that if the current lunar cycle is a full moon, the sanity level of the characters could be reduced. When the sanity level of the characters fall below a threshold, this could cause friends or clans to become hostile towards each other. As a result, this could lead to the development of a dynamic quest, where the player has to resolve the hostilities.

6.2.3 Extending the Model

With further improvements on the current model, it would be interesting to try and extend the model to a game such as Skyrim. This could be achieved by creating a mod for Skyrim using the Creation Kit, Skyrim's modding kit, and the Skyrim Script Extender (SKSE), a tool which expands the scripting capabilities of the Creation Kit. As Skyrim has day/night and weather cycles already modelled for visual effects, and the creation of characters using the Creation Kit, it wouldn't require a large amount of work to test out the model. Furthermore, the SKSE allows mods to be developed in C++, and since this model is developed in C#, it could easily be exported over.

6.3 Final Thoughts

The model proposed by this dissertation shows potential for use in open world games. Although the prototype developed for this project is limited in some areas, it does demonstrate the overall concept of this project. Taking the core concepts of this model and applying it to an open world game, such as Skyrim, could lead to more interesting and believable characters throughout the game world. Furthermore, with additional work, such as introducing facial gestures and interesting conversational tracks, it could provide the player with an overall more engaging gameplay experience.

Appendix

A disc with all the relevant files (e.g., Unity project, source code, etc..) is attached to the back of this dissertation.

Bibliography

[1] Karma. http://fallout.wikia.com/wiki/Karma. [Online; Accessed 18-May-2015].

[2] Radiant A.I. http://elderscrolls.wikia.com/wiki/Radiant_A.I. [Online; Accessed 17-May-2015].

[3] Reputation. http://fallout.wikia.com/wiki/Reputation. [Online; Accessed 18-May-2015].

[4] Shenmue. http://shenmue.wikia.com/wiki/Shenmue. [Online; Accessed 20-May-2015].

[5] Villager. http://fable.wikia.com/wiki/Villager. [Online; Accessed 17-May-2015].

[6] Matt Barton and Bill Loguidice. The History of Elite: Space, the Endless Frontier. http://www.gamasutra.com/view/feature/3983/the_history_of_elite_space_the_.php, 2009. [Online; Accessed 22-May-2015].

[7] Joseph Bates. The role of emotion in believable agents. *Commun. ACM*, 37(7):122–125, July 1994.

[8] Matt Bertz. The Technology Behind The Elder Scrolls V: Skyrim. http://www.gameinformer.com/games/the_elder_scrolls_v_skyrim/b/xbox360archive/2011/01/17/the-technology-behind-elder-scrolls-v-skyrim.aspx, 2011. [Online; Accessed 17-May-2015].

[9] Giant Bomb. Open World (Concept). http://www.giantbomb.com/open-world/3015-207/, 2015. [Online; Accessed 20-May-2015].

[10] Emily Brown and Paul Cairns. A grounded investigation of game immersion. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1297–1300, New York, NY, USA, 2004. ACM.

[11] Bobby D. Bryant and Risto Miikkulainen. *Evolving stochastic controller networks for intelligent game agents*, pages 1007–1014. 2006.

[12] Alex J. Champandard. A-Life, Emergent AI and S.T.A.L.K.E.R.: An Interview with Dmitriy Iassenev. `http://aigamedev.com/open/interviews/stalker-alife/`, 2008. [Online; Accessed 16-May-2015].

[13] David Joseph Chaplin and Abdennour El Rhalibi. IPD for Emotional NPC Societies in Games. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '04, pages 51–60, New York, NY, USA, 2004. ACM.

[14] Naughty Dog. The Last of Us, 2013.

[15] Max Dyckhoff. Buddy AI in the Last of Us. `http://www.gdcvault.com/play/1020364/Ellie-Buddy-AI-in-The`. [Online; Accessed 24-May-2015].

[16] Obsidian Entertainment. Fallout: New Vegas, 2010.

[17] John Fallon. Believable Behaviour of Background Characters in Open World Games. M.Sc. Thesis, University of Dublin, Trinity College, 2013.

[18] Rockstar Games. Grand Theft Auto, 1998.

[19] Rockstar Games. Grand Theft Auto III, 2001.

[20] Rockstar Games. Grand Theft Auto IV, 2008.

[21] Rockstar Games. Grand Theft Auto V, 2013.

[22] D. Gonze, J. Halloy, and A. Goldbeter. Deterministic and stochastic models for circadian rhythms. *Pathol Biol (Paris)*, 51(4):227–230, June 2003.

[23] IGN. Races. `http://ie.ign.com/wikis/the-elder-scrolls-5-skyrim/Races`. [Online; Accessed 23-May-2015].

[24] Kiran Ijaz, Anton Bogdanovych, and Simeon Simoff. Enhancing the believability of embodied conversational agents through environment-, self- and interaction-awareness. In *Proceedings of the Thirty-Fourth Australasian Computer Science Conference - Volume 113*, ACSC '11, pages 107–116, Darlinghurst, Australia, Australia, 2011. Australian Computer Society, Inc.

[25] Kenneth P. Wright Jr., Andrew W. McHill, Brian R. Birks, Brandon R. Griffin, Thomas Rusterholz, and Evan D. Chinoy. Entrainment of the human circadian clock to the natural light-dark cycle. *Current Biology*, 23(16):1554 – 1558, 2013.

[26] Aaron Khoo, Greg Dunham, Nick Trienens, and Sanjay Sood. Efficient, realistic npc control systems using behavior-based techniques. In *Artificial Intelligence and Interactive Entertainment II*, pages 46–51, 2002.

[27] Weizi Philip Li, Tim Balint, and Jan M. Allbeck. Using a parameterized memory model to modulate npc ai. In Ruth Aylett, Brigitte Krenn, Catherine Pelachaud, and Hiroshi Shimodaira, editors, *IVA*, volume 8108 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2013.

[28] Mei Yii Lim, João Dias, Ruth Aylett, and Ana Paiva. Agents for games and simulations. chapter Intelligent NPCs for Educational Role Play Game, pages 107–118. Springer-Verlag, Berlin, Heidelberg, 2009.

[29] Daniel Livingstone. Turing's test and believable ai in games. *Comput. Entertain.*, 4(1), January 2006.

[30] Aaron Bryan Loyall. *Believable Agents: Building Interactive Personalities*. PhD thesis, Pittsburgh, PA, USA, 1997. AAI9813841.

[31] Ibrahim M. Mahmoud, Lianchao Li, Dieter Wloka, and Mostafa Z. Ali. Believable npcs in serious games: HTN planning approach based on visual perception. In *2014 IEEE Conference on Computational Intelligence and Games, CIG 2014, Dortmund, Germany, August 26-29, 2014*, pages 1–8, 2014.

[32] Tiarnán McNulty. Residual Memory for Background Characters in Complex Environments. M.Sc. Thesis, University of Dublin, Trinity College, 2014.

[33] Erik Meijer and Vikram Kapoor. The responsive enterprise: Embracing the hacker way. *Queue*, 12(10):10:10–10:18, October 2014.

[34] Ian Millington. *Artificial Intelligence for Games*. Morgan Kaufmann, 2006.

[35] Niall Mullally. An Emotional Model for Background Characters in Open World Games. M.Sc. Thesis, University of Dublin, Trinity College, 2014.

[36] Sarah Noonan. Side Quest Generation using Interactive Storytelling for Open World Role Playing Games. M.Sc. Thesis, University of Dublin, Trinity College, 2015.

[37] Brendan O'Connor. A Relationship Model for Believable Social Dynamics of Characters in Games. M.Sc. Thesis, University of Dublin, Trinity College, 2015.

[38] Re-Timer. Circadian Rhythm. `http://re-timer.com/science/circadian-rhythm/`. [Online; Accessed 07-June-2015].

[39] Mazeiar Salehie and Ladan Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM Trans. Auton. Adapt. Syst.*, 4(2):14:1–14:42, May 2009.

[40] Jamie Sefton. The roots of open-world games. `http://www.gamesradar.com/the-roots-of-open-world-games/`, 2008. [Online; Accessed 20-May-2015].

[41] Sega. Shenmue, 1999.

[42] Bethesda Softworks. Fallout 3 Awards. `http://fallout.bethsoft.com/eng/links/fallout3-awards.php`. [Online; Accessed 22-May-2015].

[43] Bethesda Game Studios. The Elder Scrolls IV: Oblivion, 2006.

[44] Bethesda Game Studios. Fallout 3, 2008.

[45] Bethesda Game Studios. The Elder Scrolls V: Skyrim, 2011.

[46] Lionhead Studios. Fable II, 2008.

[47] Techland. Dying Light, 2015.

[48] Fabien Tencé, Cédric Buche, Pierre De Loor, and Olivier Marc. The challenge of believability in video games: Definitions, agents models and imitation learning. *CoRR*, abs/1009.0451, 2010.

[49] F. Thomas and O. Johnston. *Disney Animation: The Illusion of Life*. Abbeville Press, 1987.

[50] Electronics Tutorials. Closed-loop System. `http://www.electronics-tutorials.ws/systems/closed-loop-system.html/`. [Online; Accessed 01-June-2015].

[51] Electronics Tutorials. Open-loop System. `http://www.electronics-tutorials.ws/systems/open-loop-system.html/`. [Online; Accessed 01-June-2015].

[52] Unity. Unity3d game engine. `https://unity3d.com/`. [Online; Accessed 01-June-2015].

[53] Henrik Warpefelt and Björn Stråat. Anti-heuristics for maintaining immersion through believable non-player characters. In *fdg'13*, pages 455–456, 2013.

[54] Henrik Warpefelt and Björn Stråat. Breaking immersion by creating social unbelievabilty. In *Proceedings of AISB 2013 Convention*, Social Coordination: Principles, Artefacts and Theories (SOCIAL.PATH), pages 92–100, 2013.

[55] GSC Game World. S.T.A.L.K.E.R, 2007.

[56] Zhouyi Xu and Xiaodong Cai. Stochastic simulation of delay-induced circadian rhythms in drosophila. *EURASIP J. Bioinformatics and Systems Biology*, 2009, 2009.