

USING ADAPTIVE BEHAVIOUR TREES to EVOLVe PERSONALITY AI for NPCs IN video GAMES

By Isaac Dunn



[Date]

[Company name]

[Company address]

Contents

[Abstract 5](#_Toc145699562)

[Keywords 5](#_Toc145699563)

[Abbreviations 5](#_Toc145699564)

[Chapter 1 Introduction 6](#_Toc145699565)

[1.1 Introduction 6](#_Toc145699566)

[1.2 Research Background 7](#_Toc145699567)

[1.3 Motivation 7](#_Toc145699568)

[1.3 Research question 8](#_Toc145699569)

[1.5 Aims 8](#_Toc145699570)

[1.6 Objectives 8](#_Toc145699571)

[1.7 Deliverables 8](#_Toc145699572)

[Chapter 2 Literature Review 8](#_Toc145699573)

[2.1 Literature review 8](#_Toc145699574)

[2.1.1 Behaviour trees 8](#_Toc145699575)

[2.1.2 Adaptive Behaviour Trees 10](#_Toc145699576)

[2.1.3 State Machines 14](#_Toc145699577)

[2.1.4 Hierarchical Task Network 16](#_Toc145699578)

[2.1.4 Large Language Models 17](#_Toc145699579)

[2.1.5 OCC layer 18](#_Toc145699580)

[2.1.6 trait-based personality. 19](#_Toc145699581)

[2.1.7 Associative Memory Network 20](#_Toc145699582)

[2.1.8 Literature Overview 21](#_Toc145699583)

[2.2 Analysis of the problem/ Improvement 21](#_Toc145699584)

[Chapter 3 Research Methods 23](#_Toc145699585)

[3.1 Research Methods 23](#_Toc145699586)

[Chapter 4 Artefact 25](#_Toc145699587)

[4.1 Design of an Artefact 25](#_Toc145699588)

[4.1.1 Development plan 25](#_Toc145699589)

[4.1.2 Testing/ Validation Plan 28](#_Toc145699590)

[4.2 Artefact implementation 31](#_Toc145699591)

[4.3 Testing and Validation of the artefact 32](#_Toc145699592)

[4.4 Critical Evaluation 33](#_Toc145699593)

[Chapter 5 Conclusion and Future Work 35](#_Toc145699594)

[5.1 Conclusion 35](#_Toc145699595)

[5.2 Future work 36](#_Toc145699596)

[Chapter 6 Appendix 37](#_Toc145699597)

[6.1 Project implementation 37](#_Toc145699598)

[6.2 UI 38](#_Toc145699599)

[6.3 Stat manager 39](#_Toc145699600)

[6.4 resource manager 40](#_Toc145699601)

[6.5 Nodes 41](#_Toc145699602)

[6.6 behaviour tree 52](#_Toc145699603)

[6.7 genetic algorithm 57](#_Toc145699604)

[6.8 Test Data 58](#_Toc145699605)

[Chapter 7 References 59](#_Toc145699606)

[Figure 1: example of a behaviour tree in Unreal Engine 5 which as stated by (Iovino et al., 2022) will contain root, composite (sequence) and leaf (in purple) nodes which are executed through the tree from left to right. 9](#_Toc145699607)

[Figure 2: image showing how as stated by (Iovino et al., 2022) behaviour trees can have behaviours replaced to create an adaptive behaviour tree with new behaviours that can change how an AI agent can act. 10](#_Toc145699608)

[Figure 3: image showing how a genetic algorithm as stated by (Iovino et al., 2022) propagates to its children with 2 parent chromosomes (first blue, second red) crossing over to share genes with uniform crossover and then the child chromosome after mutation is applied. With a mutation rate of 10% and the genome containing 10 genes this causes 1 gene to mutate (green). 11](#_Toc145699609)

[Figure 4: image showing how behaviour nodes in a behaviour tree as stated by (Iovino et al., 2022) can be expressed as genes in a genetic algorithm's chromosome. 12](#_Toc145699610)

[Figure 5: Q-learning algorithm as stated by (Jang, B. et al. 2019) 12](#_Toc145699611)

[Figure 6: diagram showing an adaptive behaviour tree with the anti-bloat pruning removing poor fitness behaviours as mentioned by (Colledanchise et al., 2018) 13](#_Toc145699612)

[Figure 7: image showing a finite state machine for an enemy AI with behaviours (circles) being triggered by conditions (arrows) as mentioned by (Li et al., 2016) 14](#_Toc145699613)

[Figure 8: diagram showing a basic FuSM with arrows showing how it transitions between behaviours and the fuzzy values as mentioned by (Rozikin, Dijaya and Taurusta. 2021) 15](#_Toc145699614)

[Figure 9: diagram showing how a FuSM transitions between 2 behaviours with a crossover point as mentioned by (Rozikin, Dijaya and Taurusta. 2021) 16](#_Toc145699615)

[Figure 10: graph showing a decision tree generated from a HTN with hierarchies for each non-primitive task and the primitive tasks at the bottom of the tree as shown by (Ontanon & Buro, 2015). 17](#_Toc145699616)

[Figure 11: an example of ChatGPT integration in game dialog showing how the player (human) can interact with the AI and the AI with itself in persistent conversions, (Sharples, M. 2023) 18](#_Toc145699617)

[Figure 12: 5-factor personality as seen in the OCC model by (Popescu et al., 2014) with the heuristic value shown in red representing the magnitude of a personality factor from -1 to 1. 19](#_Toc145699618)

[Figure 13: diagram showing an AMN combined with a decision maker and emotional AI to create goal driven personalities in video game NPCs as used by (Spraragen, and Madni, 2014) 21](#_Toc145699619)

[Figure 14: diagram showing the research onion (Melnikovas, 2019) 23](#_Toc145699620)

[Figure 15: use case diagram of the proposed ABT system in a game world showing interactions between systems 27](#_Toc145699621)

[Figure 16: UML diagram of all systems for the artefact 28](#_Toc145699622)

[Figure 17: diagram showing the default BT nodes with a top node to choose between attacking, eating and idling behaviours 29](#_Toc145699623)

[Figure 18: questionnaire design part 1 30](#_Toc145699624)

[Figure 19:questionnaire design part 2 31](#_Toc145699625)

[Figure 20: image showing the completed artefact game world with NPCs using the ABT system 37](#_Toc145699626)

[Figure 21: NPC with UI for behaviours, hunger and awareness 37](#_Toc145699627)

[Figure 22: code for UI 38](#_Toc145699628)

[Figure 23: stat manager for the NPC 39](#_Toc145699629)

[Figure 24: code for resource manager 40](#_Toc145699630)

[Figure 25: base node 41](#_Toc145699631)

[Figure 26: selector node 42](#_Toc145699632)

[Figure 27: sequence node 43](#_Toc145699633)

[Figure 28: walking behaviour 44](#_Toc145699634)

[Figure 29: flee behaviour 45](#_Toc145699635)

[Figure 30: idle behaviour for wandering 46](#_Toc145699636)

[Figure 31:detection node 48](#_Toc145699637)

[Figure 32: attack node 48](#_Toc145699638)

[Figure 33: hunger node 49](#_Toc145699639)

[Figure 34: node for talking to other NPCs 50](#_Toc145699640)

[Figure 35: eating behaviour 51](#_Toc145699641)

[Figure 36: behaviour tree 56](#_Toc145699642)

[Figure 37: genetic algorithm 58](#_Toc145699643)

[Figure 38: responses to the questionnaire 58](#_Toc145699644)

[Figure 39: experiment data 59](#_Toc145699645)

[Figure 40: table of the reason for answers given in the questionnaire 59](#_Toc145699646)

# Abstract

The use of procedural content generation for story telling in video games is a growing field of research with one field in particular growing in prominence being personality and emotional AI. One gap in this research is how most applications of personality AI are static with little simulated character growth. This project will seek research various behaviour tree and state machine methods for a decision layer and various methods for emotional AI such as trait-based behaviours or OCC models. The artefact will use adaptive behaviour trees (ABT) using genetic algorithms with trait-based behaviours to create an emotional AI for Non-player characters (NPCs) for video games with emotional responses changing over time to allow for more diverse characters to be generated in games. The methods used for this project were a mix of experiments on what behaviours were changed and surveys on how realistic the AI felt to the participants to find if the research was successful in creating adaptive AI. Overall, the results from this research found that adaptive behaviour trees when using genetic algorithms were successful at adapting emotional AI creating a fluid and dynamic AI personality however, they were erratic and unpredictable which should be improved on in future works.

# Keywords

Adaptive behaviour trees

Personality AI

Video Game NPC

Genetic algorithms

Narrative generation

# Abbreviations

BT – behaviour tree

ABT – Adaptive Behaviour Tree

GA – Genetic Algorithm

NPC – Non-Player Character

LLM – large language model

HTN – hierarchical task network

FSM – Finite State Machine

AMN – associative memory network

FuSM – Fuzzy State Machine

AI - Artificial Intelligence

L-OCC – Layered OCC

# Chapter 1 Introduction

## Introduction

Personality AI is a wide field of AI development that has many use cases as stated by (Yorita, A. *et al.* 2019) from chatbots which can create sentences with a generated personality attached which can be used for the service industry or used in the care sector to talk and provide support for people suffering from loneliness or people with disabilities. Another use case is in the field of procedural content generation which is a constantly evolving field in various industries like film or the video games industry in the case of this study which can use procedural content generation for generating stories in a game with one field being emotional AI for the use of generating stories during gameplay this can be seen in games like Dwarf Fortress which according to (Short, T.X. and Adams, T. 2019), AI agents in the game have simulated emotions and actions to engage the player in the Gameworld which overall can make up the personality of a given AI agent.

In this research report will be an investigation into the various kinds of behaviour trees and state machines for the use of decision-making AI before looking into the various forms of emotional AI layers. Below is the list of subjects that will be covered in the literature review…

* Behaviour trees
* Adaptive behaviour trees (ABT)
  + Genetic ABTs
  + Q-learning ABTs
  + Anti-bloat optimized ABTs
* State machines
  + Finite state machines
  + Fuzzy state machines
* Large Language models
* OCC emotion layer
* Trait based personality systems

An artefact will then be developed using the ABT method with a trait-based personality AI layer and tested to see if the emotional AI changes from actions in the game world by comparing the artefact to a BT via a survey on users’ opinions.

## 1.2 Research Background

Emotional AI in video games has previously been achieved with various methods such as behaviour trees or fuzzy state logic as seen with (Popescu et al., 2014), to simulate emotions and how the AI interacts with the Gameworld or other entities creating a simulated personality. One issue in the field of personality AI is how this AI is often static with emotional states being unchanged rather than a constant developing character as seen with (Belle, S., Gittens, C. and Nicholas Graham, T.C. 2022) where the AI is able to simulate personalities, but these personalities are unable to evolve as there is no mechanism to allow it with how the AI agent reacts to the world emotionally and how it chooses to interact based on those emotions remaining static. This field not only has use cases for video game AI but also may have some uses for personality chatbots as seen with (Yorita, A. *et al.* 2019) where chatbots can use simulated personalities to better interact with people making this field of research useful for personality AI systems that may be used with a chatbot that is able to develop alongside the user adapting certain personalities traits and mannerisms to better compliment the user’s personality.

As previously mentioned, according to (Short, T.X. and Adams, T. 2019) games like dwarf fortress use personality AI as a form of procedural content generation with small character stories being created through emergent systems and character behaviours which can make an otherwise non-story driven game have more personal stories attached as explained by (Ampatzidou, C. 2019) with emergent gameplay being used to immerse a player better with many intertwining systems forming into realistic scenarios with no scripted story elements requirement.

Another area of research would be into methods of simulating emotions for an AI agent with an emotional layer which according to (Dai et al., 2020) can be created using an OCC layer or according to (Short, 2021) using a trait-based system to simulate the behaviours that the decision-making AI would utilize with differing definitions of emotions in AI it would require looking into these methods. Both the emotion layer and the decision layer of the AI agent could be combined using a Large Language Model (LLM) to generate the behaviour tree, emotional actions and adapt it to the conditions of the game world. Overall, the literature review will seek to expand and evaluate these ideas.

## 1.3 Motivation

Many video game developers when creating stories with their games must either write a story for the game or utilize narrative generation techniques to develop a game’s story with the written story taking more time to write but can have a better quality of story than narrative generation techniques. A system that can provide the options of generating complex characters while still being easy to edit is a motivation for this study to further the field of narrative generation in video games and provide more realistic or entertaining generative stories to users. While this will not revolutionise the field it will provide an extra technique for game AI which can have uses in games that utilize procedurally generated characters to help generate more adaptive AI that can react to the game world.

## 1.3 Research question

The research question for this research is whether an adaptive behaviour tree can be used to create believable personality AI which this research project hypothesises that it is possible.

## 1.5 Aims

This project aims to use adaptive behaviours trees to create an emotional AI that can evolve to influences in the outside world creating evolving AI personalities.

## 1.6 Objectives

* Research into emotional AI methods
* Design an NPC AI using research results.
* Have NPCs emotional behaviours adapt to outside events in a game world.
* Test and validate artefact.
* Evaluate and find conclusions to the research.

## 1.7 Deliverables

* Research into existing approaches to emotional AI and interaction
* Artefact NPC AI using an emotional AI technique.
* Testing and validation of artifact
* Full report containing the research, artefact, validation, and conclusions of the research.

# Chapter 2 Literature Review

## 2.1 Literature review

### 2.1.1 Behaviour trees

Behaviour trees (BT) according to (Iovino et al., 2022) are used often in games programming as well as robotics to create behaviour driven AI in a modular and readable way with the tree-like structure aiding in that readability as seen in Figure 1 with the behaviour tree having a structure that shows the behaviours and flow of the tree and how it switches between behaviours. BTs are made up of numerous nodes which change how a behaviour is activated in the AI agent. These nodes can make up the whole tree starting with a root node which is where the tree starts. Any node can have children’s nodes which branch out into various functions for the AI agent with the final child node or the leaf node being the code for the desired behaviour which can give functionality to the AI agent such as follow or attack behaviours. All of these nodes have a success, running or fail criteria which are triggered in different node types in different ways which are known as composite nodes such as with a selector node which will run each child node one at a time until it finishes which it will then return a success, sequence node which will run each child node until a given child returns a fail which will then cause the sequence node to fail also, other node types include repeat which simply repeats a behaviour node until the child node returns a fail. Overall behaviour trees according to (Sekhavat, 2017) have the benefit of making the structure of an AI agent easier to understand whilst being reactive and predictive making it ideal for video games which is why it is commonly used in many games for creating relatively advanced AI agents. The drawback for BTs however is that without modification to the model the behaviours can not be switched for other behaviours making the model static and unusable for this project.

A screenshot of a computer

Description automatically generated

Figure 1: example of a behaviour tree in Unreal Engine 5 which as stated by (Iovino et al., 2022) will contain root, composite (sequence) and leaf (in purple) nodes which are executed through the tree from left to right.

### 2.1.2 Adaptive Behaviour Trees

#### 2.1.2.1 What are adaptive behaviour trees.

As previously mentioned, BTs have limitations that make it hard to generate new behaviours with similar issues in other state machine methods like FSMs and FuSMs as mentioned by (Colledanchise et al., 2018) Adaptive behaviour trees (ABTs) are a variant of BTs with all the same functionality of a behaviour tree with the previously mentioned BT node system giving a control flow for the behaviours. ABTs differ from BTs by using methods to adapt the behaviour tree under certain conditions as seen in Figure 2. These adaption techniques can be created using a variety of methods such as genetic algorithms (GA) or Q-learning.

A diagram of an object

Description automatically generated

Figure 2: image showing how as stated by (Iovino et al., 2022) behaviour trees can have behaviours replaced to create an adaptive behaviour tree with new behaviours that can change how an AI agent can act.

#### 2.1.2.2 genetic algorithms

The genetic algorithm method according to (Iovino et al., 2022) is a system that attempt to mimic evolution with encoded genes that could be using for a variety of systems, in this case the ABT’s node structure with the genes making up the order and type of behaviours as seen with (Iovino et al., 2022) system where certain behaviours for an AI agent such as moving right is set as a gene in the GA. How the GA evolves is by using a mixture of gene crossover and mutation where in this case the genes of an AI agent’s new BT is mixed with 2 parents’ genomes with half of the genes coming from one parent and half from the other to create a new child genome. This helps to mimic environmental pressures in a simulation with successful parent genes being capable of propagating more child genomes which can spread a successful strategy to a higher portion of the population. As mentioned by (Iovino et al., 2022) this, however, can eventually lead to stagnation as strategies will eventually become mostly like one another. According to (Iovino et al., 2022) to mitigate this mutation is also added to provide entirely new genes in the population which can be added either on creation of a new genome or during the lifetime of a pre-existing genome which can lead to new strategies emerging from the population. These mutated genes according to (Iovino et al., 2022) occur via a mutation rate which set the proportion of genes that will be mutated on the genomes creation with a mutation rate of 10% affecting 1 in 10 genes. This can be seen in Figure 3 which shows how the data within a genetic algorithm can change over time via crossover to children with the example also showing how mutation can mitigate stagnation in a limited population by randomly changing the genes in a child’s chromosome to ones not present in the population in this example the gene holding the number 2. Another important factor of genetic algorithms is the optimal mutation rate and crossover as according to (Hassanat et al., 2019) a high mutation rate can lead to changes happening too quickly which may make it more difficult for the genetic algorithm to adapt as it changes too randomly whereas due to how a genetic algorithm adapts a mutation rate that is too low will require more generations within the algorithm for more noticeable changes to occur. A similar issue occurs with the method of crossover with multiple types of crossovers having different effects with one-point crossover according to (Hassanat et al., 2019) is where the genome for each parent is split in 2 places with half the genome from each parent going to the child to share strategies for solving a task with other methods like multipoint crossover which according to (Hassanat et al., 2019) is similar to one-point crossover however it splits in more than one place sharing smaller strategies and uniform crossover which according to (Immanuel and Chakraborty, 2019) is where each gene is selected from each parent randomly which increases the diversity of children at the cost of established strategies from the parent being passed down.

Overall, this according to (Iovino et al., 2022) can be incorporated into a BT by allowing the specific behaviours in a BT to be expressed as the genes with a given AI agent having a genome as seen in Figure 4. This overall show how ABTs using GAs can function but also with the addition of having crossover and mutation can show how new behaviour trees can be created using GAs with whole new strategies forming from the combined systems. This genome according to (Xie et al., 2021) can be given constraints that can limit how the GA can mutate allowing for more control over the algorithm that can be useful in an emotional model to stop the AI from switching between behaviours in a way that could seem unrealistic.

A grid of numbers and red and blue numbers

Description automatically generated

Figure 3: image showing how a genetic algorithm as stated by (Iovino et al., 2022) propagates to its children with 2 parent chromosomes (first blue, second red) crossing over to share genes with uniform crossover and then the child chromosome after mutation is applied. With a mutation rate of 10% and the genome containing 10 genes this causes 1 gene to mutate (green).



Figure 4: image showing how behaviour nodes in a behaviour tree as stated by (Iovino et al., 2022) can be expressed as genes in a genetic algorithm's chromosome.

As the proposed system will require NPCs with pre-generated personalities as seen with (Liu et al., 2019) with this being used to solve a specific problem where it is appropriate for a GA’s initial population to have the chromosomes predefined but also for the chromosome of the NPC to change not just on creation but throughout its lifetime, this will require a modified version of genetic algorithms with one method proposed by (Nitisiri et al., 2019) where the mutation rate is not just applied on generation but also throughout its lifetime to enable the GA to adapt to changing conditions more effectively.

#### 2.1.2.3 Q-learning

Q-learning or reinforcement learning according to (Jang, B. et al. 2019) is an algorithm where a fitness value system is used to find the failure magnitude of a given attempt to solve a problem this could be the problem also seen in (Colledanchise et al., 2018) where a character needs a strategy to move from point A to point B. Q-learning will in this case have a heuristic value tied to the distance to its target position and will then adapt the algorithm using the learning rate to adjust the values used to trigger the behaviours on a BT with the example from (Dey and Child 2013) of an enemy fleeing the player when health is low with the health trigger value changing by the learning rate before it gets to its desired behaviour using the algorithm seen in Figure 5. While this method can adapt the ABTs by changing its conditional values this however can not change the structure of the ABT and the behaviours used which can make the behaviour more static than what is seen with ABTs which use genetic algorithms however this also has the benefit of being more predictable making it easier to understand how this AI will interact with it environment whereas the GA method will lead to a wider number different BT layouts as the method is able to swap out the behaviours and their triggers.



Figure 5: Q-learning algorithm as stated by (Jang, B. et al. 2019)

#### 2.1.2.4 Anti-bloat optimization

As stated by (Colledanchise et al., 2018) ABTs using genetic algorithms have issues with bloating where uncontrolled complexity of the behaviour tree can lead to the ABT structure being progressively larger which can cause issues with performance as the ABT will likely have unused or rarely used behaviours so while it is not necessary, anti-bloat measures should be considered for the project. As also mentioned by (Colledanchise et al., 2018) these measures can be implemented in various ways. A simple solution according to (Colledanchise et al., 2018) is to limit the size and depth of the ABT i.e. if it requires more than 5 composite nodes to execute a behaviour then prune the nodes or allow only a maximum of a number of nodes. While this is a simple solution it is not ideal as it removes behaviours without context for what function they could serve with crucial behaviour nodes required for a more complex set of actions being removed and a limited depths also can limit the complexity of the tree overall. This can be rectified by using a fitness test on the ABT which evaluates the fitness of each node on the tree based on how used a node is with nodes with no possible use case or functionality having a low fitness score. These low fitness nodes are then removed along with any child nodes essentially pruning the ABT without removing important behaviours and keeping the ABT smaller than without any anti-bloat control although this method may remove important nodes that are not used often. This can be seen in Figure 6 with a sequence of behaviours contain common behaviours like eating and sleeping and an unlikely behaviour like searching for gold with every successful execution adding to the fitness value. As the diagram shows the find gold behaviour has a low fitness score meaning the behaviours in likely impossible or redundant so when pruning is required i.e. the behaviour tree has reached max size then the behaviour is removed to lower the size of the tree optimizing it overall.

A diagram of a sequence of steps

Description automatically generated

Figure 6: diagram showing an adaptive behaviour tree with the anti-bloat pruning removing poor fitness behaviours as mentioned by (Colledanchise et al., 2018)

Overall ABTs created by using a BT and combining it with an adaptive method like genetic algorithms and optimised using a fitness based pruning model creating an adaptive and optimised.

### 2.1.3 State Machines

#### 2.1.3.1 Finite State Machines

As behaviour trees are similar state machine which according to (Li *et al.,* 2016) are a type of AI which will run a given behaviour based of a given state. There are various similar models of state machine with finite and fuzzy state machines. According to (Li *et al.,* 2016) a finite state machine (FSM) is a type of state machine where a state is given with a trigger to activate a state for example if an enemy gets too close to the player this can be the trigger for a state machine making a behaviour execute. This can be seen in Figure 7 with the conditions executing certain behaviours with the diagram showing a basic game enemy AI where the AI will default at a wander/idle behaviour to search for the player until the player is found at which point they will transition to the attack behaviour with this new behaviours transitioning either back to the wander if the player is lost or the flee behaviour is their health is low which they will then heal and transition back to wander effectively allowing for an AI agent to change states to meet the conditions in the game world. This type of AI was used often in games due to its simplicity however like BTs it is static making it not ideal for the project while lacking functionality as behaviours are executed procedurally and therefore will have a natural priority system for executing behaviours.

A diagram of a diagram

Description automatically generated

Figure 7: image showing a finite state machine for an enemy AI with behaviours (circles) being triggered by conditions (arrows) as mentioned by (Li et al., 2016)

#### 2.1.3.2 Fuzzy State Machines

The alternative to FSM is a fuzzy state machine (FuSM) which is used to remedy the functionality issues with finite state machines. This is due to how FuSMs work according to (Rozikin, Dijaya and Taurusta. 2021) where similar to FSMs behaviours are triggered using variables however they differ from FSMs by utilizing a method known as fuzzy logic where multiple behaviours can be activated to a degree for example as seen in Figure 8 where an enemy may either move towards the player or shoot the player however it can only shoot when nearby or around 30 metres however, fuzzy variables are applied with a threshold variable applied of 0.3 on the transition to attack and 0.7 on the transition to move to allow behaviours to activate while the previous behaviour still active as seen in Figure 9 where there is a crossover between the shoot and move behaviours that allow the AI to effective move and shoot at the same time essentially triggering 2 or more behaviours at once with the threshold values represent the low and high values for the behaviour condition to activate. This allows for more reactive emergent behaviours to be created a behaviour can blur between each other making them more lifelike while mitigating the limitations of functionality that FSMs have however, this can make behaviours less predictable and harder to develop as behaviours can interact in an exponential number of ways making more complex AI unreliable while also having the same issues with readability seen in FSMs.

A diagram of a diagram

Description automatically generated

Figure 8: diagram showing a basic FuSM with arrows showing how it transitions between behaviours and the fuzzy values as mentioned by (Rozikin, Dijaya and Taurusta. 2021)

A diagram of a game

Description automatically generated

Figure 9: diagram showing how a FuSM transitions between 2 behaviours with a crossover point as mentioned by (Rozikin, Dijaya and Taurusta. 2021)

As mentioned by (Postnikov, E.V. et al. 2019) FuSMs can be made to be adaptive in the same way as BTs with the use of Q-learning by using a fitness algorithm to using a heuristic value for fitness to find the error amount with the example from (Postnikov, E.V. et al. 2019) using a virtual football simulation with the error rate being the distance a ball is from the goal for the system to train on similar to what would be seen in ABTs using Q-learning. This error system would then change the transition states parameters i.e. the thresholds for changing between 2 state as seen in Figure 9 so that the transition between shooting and moving change thus affecting the behaviour of the AI as a whole with the amount the threshold changes depends on the learning rate of the Q-learning system. The use of FuSMs and Q-learning has the potential to allow FuSMs to adapt to their environment making them viable for this project however the issues of reliability and readability as previous mentioned by (Rozikin, Dijaya and Taurusta. 2021) could make this less viable than the ABT method.

### 2.1.4 Hierarchical Task Network

A Hierarchical task network or HTN according to (Ontanon & Buro, 2015) is a decision-making architecture which can create goals for an AI by breaking them down into smaller tasks with primitive tasks which are actions an AI can make and non-primitive tasks which are the tasks the AI must complete to achieve its goal. These tasks are then turned into a multi-layered decision tree as seen in Figure 10 which dictates the order the behaviours will be activated like BTs but the commands can come from a higher authority AI agent which allows it to control multiple AI agents. This could be seen with an example of a logging camp with an AI needing to make wood as the goal, this can first be broken down to gathering and processing as the non-primitive goal with gathering furthering broken to the more primitive behaviours of finding, chopping, and hauling the tree back to the camp. This would involve the AI for the logging camp controlling the workers from the camp to complete these actions.

Like BTs, HTNs are non-adaptive so the tasks broken down would likely be the same making the order of actions the HTN the same for each task. According to (Rehakova & Neruda, 2018) to remedy this, the HTN could be combined with a genetic algorithm with the genes from the GA affecting the primitive tasks decomposed from the HTN. Overall, the use of HTNs with GA could be used to create an adaptive AI however the top-down nature of the method is more useful for controlling multiple AI agents rather than a single agent which this project will rely on making the extra functionality no more beneficial to the project than other decision AI methods like BTs and other state machines.

A diagram of a mathematical model

Description automatically generated

Figure 10: graph showing a decision tree generated from a HTN with hierarchies for each non-primitive task and the primitive tasks at the bottom of the tree as shown by (Ontanon & Buro, 2015).

### 2.1.4 Large Language Models

Other methods for AI personalities can be with the use of Large Language Models (LLMs) with systems like chat-GPT being an option for generating the behaviour reaction and action required for this project. This can be seen with (Sharples, 2023) where an AI like Chat-GPT can be used in video game creation and design. With this idea chat-GPT could be used to create a behaviour tree and restructure in real-time to simulate this changing behaviour. LLMs like Chat-GPT can also be used to generate dialogue according to (Rapp et al., 2021). This dialogue can be made in response to what the user says or does in the game world and can be generated to respond to the situation given also, according to (Rapp et al., 2021) this dialogue can be created to mimic emotional behaviours which would overall be able create an adaptive emotional AI system as seen with the diagram in Figure 10 where an LLM assisted NPC AI can have interact with each other and the player having persistent conversations and interactions that influence the AI. There are however issues with this approach that can make it unviable for real-time AI as according to (Bender et al., 2021) LLMs require large amounts of data to operate with the dataset of GPT-3 being 570 gigabytes. This makes it difficult to use in videos games as it can make LLMs slow to process data effectively which can therefore make the AI less reactive. Other issues with this model can also occur according to (Sallam, 2023) as an issue with LLMs like Chat-GPT known as hallucinations where the AI can create inaccurate information which affects the overall quality of the dialogue outputted to the user. Overall, these issues make LLMs less effective for game AI then more conventional AI such as BTs but can provide both the logical and emotion layer of a personality AI.

A diagram of a diagram

Description automatically generated

Figure 11: an example of ChatGPT integration in game dialog showing how the player (human) can interact with the AI and the AI with itself in persistent conversions, (Sharples, M. 2023)

### 2.1.5 OCC layer

With the use of ABTs and other state machines they can provide the logic for executing behaviours but for the emotion/ personality system an emotional layer needs to be developed to simulate the emotional behaviours of the AI agent as well as how they interact and react to the game world. One such method of achieving this as mentioned by (Dai et al., 2020) is the layered OCC model or L-OCC where emotions are put into 5 categories as seen in Figure 11 for emotion types, these types being openness, extraversion, neuroticism, conscientiousness and agreeableness each of these categories are populated with related emotion types with a positive and negative effect determined by a heuristic value i.e. within openness can have the happiness emotion but the magnitude of this emotion can be positive or negative based on the heuristic value with negative values representing the magnitude of sadness the AI agent is simulating. Other methods can include the OCC component model which according to (Popescu et al., 2014) used in the emotion engine Gamydala to create various emotions and belief programmed which are expressed when an action intersects with the belief layer causing an emotional response in the AI. With the OCC emotions can be used as triggers for a state machine like a BT or ABT triggering an emotional behaviour.

A diagram of a diagram

Description automatically generated

Figure 12: 5-factor personality as seen in the OCC model by (Popescu et al., 2014) with the heuristic value shown in red representing the magnitude of a personality factor from -1 to 1.

### 2.1.6 trait-based personality.

As stated by (Short, 2021) a method of simulating emotion can be created using a system of a series of different personality traits that can be applied to the NPC with varying effects to simulate the emotions and interactions an AI agent may make with a game world. This can be seen in games like Darkest Dungeon or Crusader Kings 2 as according to (Short, 2021) this can be traits that allow for specific behaviours under certain conditions like an NPC having behaviours that represent traits like wrathful and have behaviours triggered by the trait for example the a wrathful NPC may be more likely to attempt to fight another NPC when annoyed allowing the system to function as the behaviours. This overall leaves the traits open ended in how they can represent a behaviour allowing for game developers to plan how the NPC will interact with the game world making it useful for story generation unlike in other emotional AI methods like LLMs which can act in ways that are more unpredictable as previously stated by (Sallam, 2023) so the trait based system overall would give more control to developers in how their AI can act.

According to (Ferro et al., 2013) a trait-based personality system can be combined with the OCC model to allow for an effective classification and development of each emotion type as proposed for the OCC model as used in systems like (Popescu et al., 2014) in the GAMYGDALA emotion engine. Issue with this model when however is evaluating what traits may be entertaining or realistic to the user as according to (Short, 2021) combining a trait based personality system with the OCC model can lead to behaviours that are not entirely obvious to the user as behaviours can be developed in a way that is difficult to interpret however (Ferro et al., 2013) shows that the trait system and OCC model can provide a wide possibility of behaviours to be used in the project which can be considered more realistic to the user but without user input and surveys this can prove difficult to prove.

### 2.1.7 Associative Memory Network

An Associative Memory Network or AMN as stated by (Spraragen, and Madni, 2014) is a system used in emotional AI models for keeping track of short- or long-term goals for video game NPCs. These memories are stored by certain events the NPC will perceive in a game world but can be recalled triggering behaviours and emotions on the decision layer. The memories can be recalled with either associative memories which simulate non-conscious memories or deliberative memories which simulate conscious memories with associative memories being triggered by the events the NPC will see with a heuristic score based on how similar the event is to their memory being used to find the strongest event to recall which overall can be used to simulate. The deliberative memories on the other hand are recalled in a step-by-step process to form a plan for the NPC’s goals. This system as stated by (Spraragen, and Madni, 2014) could be beneficial to allow the NPCs to simulate wants and needs creating more realistic AI in long term use. Another benefit of this system is that the decision and emotion layers can simply be switched out with the memory pool acting as a storage of information which for example a BT or any other state machine could use to fulfil the requirements for the behaviour to trigger.

A diagram of a working memory

Description automatically generated

Figure 13: diagram showing an AMN combined with a decision maker and emotional AI to create goal driven personalities in video game NPCs as used by (Spraragen, and Madni, 2014)

### 2.1.8 Literature Overview

Overall, the research has shown that an adaptive behaviour tree like the one used in (Georgeson, 2016) can be used to trigger trait based emotional behaviours in a structure that can mimic a personality. The research has also shown that ABTs have more benefits than other methods with FSMs, FuSMs and BTs lacking the adaptiveness to create an adaptive AI personality. While LLMs can create adaptive behaviours however as previously mentioned LLMs are too large, slow, and unpredictable for this application with FuSMs also having issues around predictability which overall effects game development and debugging of behaviours. HTNs like state machines also are incapable of adapting without the use of techniques like GAs however, they are more suitable for multiple agents to be controlled at once rather than a single agent. The emotional behaviours overall should be modelled after the Gamygdala system proposed by (Popescu et al., 2014) with traits allowing for easily defined, easy to develop and model with the OCC model providing a guideline for what behaviours should be developed to simulate a full artificial personality however the trait-based system is also appropriate for use to simply showcase emotional behaviours working with the ABT. The use of the AMN would also be beneficial to the project however it will not be necessary to create an adaptive emotional AI.

## 2.2 Analysis of the problem/ Improvement

Using the knowledge gained from the literature review the research gap in the field of emotional AI using adaptive behaviour trees has been identified with the research done for this project there is a distant lack of adaptive emotional AI with various methods such as BTs recreating emotional AI like (Agliata et al., 2022) but these have remained static or unpredictable in the case of (Rapp et al., 2021). Research into ABTs have shown that adaptive behaviour in video games is possible but the extent of this research has only been within the use of problem-solving AI as seen with (Iovino, M., et al. 2022). Some method like LLMs according to (Rapp et al., 2021) can create personalities that react and adapt to input however LLMs have many issues with speed affecting real-time AI and accuracy that make readable and debug-able design as seen in video games difficult to create. Others like HTNs can provide a similar level of functionality to BTs. As stated by (Ontanon & Buro, 2015) HTNs can effectly create their own trees while also according to (Rehakova & Neruda, 2018) having the ability to be made adaptive using GAs giving it similar functionality to an ABT however the hierarchical tasks are useful for multiple AI agents making the defining features of adaptive HTNs or HTNs redundant.

Emotional models have also been discussed with LLMs as mentioned by (Sharples, M. 2023) being capable of producing near limitless content but the issues with hallucinations and system requirements make it unreliable for real-time usage and Behaviour trait systems as mentioned by system as used by (Sallam, 2023) and (Short, 2021) which while are simplistic and may not be an accurate representation of an emotion but the emotions that need to be developed can be defined using the trait for expressing emotions which are useful to developers as they are predictable, more easily expressed and modular making them ideal for a behaviour tree which can execute one behaviour at a time under a certain condition and the simplicity can make it easier to develop a behaviour that a user can recognize.

With the gap identified and researched the artefact should be developed using an adaptive behaviour tree with genetic algorithms to act as the decision-making process for the NPC AI and the use of trait based emotional behaviours with the various emotional behaviours to be developed individually to mimic simple emotional behaviours.

Overall, from what has been seen in research there is no exclusive tools required to create the project, but some have been seen to be beneficial as shown by (Georgeson, J. 2016) with game engines like Unity being used to create a game environment to test and showcase the findings. As also mentioned by (Georgeson, J. 2016) game engines like Unity can provide useful resources which aide game developers such as an editor and graphical user interface which can help ease development of features and make it easier to show the behaviours for the project. With this in mind the artefact will be developed in Unity.

# Chapter 3 Research Methods

## 3.1 Research Methods

To effectively and validly research this topic the research onion methodology will be used as stated by (Melnikovas, 2019) which is a method of planning research with a total of 6 layers overall to help build a framework of research on a given subject which can be seen in the diagram...

A diagram of a diagram

Description automatically generated

Figure 14: diagram showing the research onion (Melnikovas, 2019)

The layers of the onion represent the different elements of research starting with the research philosophy on the outermost layer which makes up the philosophy of the research such as pragmatism or positivism which serves to lay the foundations for the research with the philosophy of why the research and any ethical restraints are important to the researcher. According to (Saunders et al., 2021) philosophies like but not limited to interpretivism where meaning is subjective with no 1 single solution or answer for any problem making it more ideal foe qualitative research. Another form is positivism where any idea is measurable as true or false allowing for structure ideal for quantitative research. According to (Saunders et al., 2021) Pragmatism can be defined as a complex philosophy which investigates the effects of different ideas with the best one for a given situation being the ideal method which may be ideal for a mix of methodologies. There are many other examples of philosophies that can be used in this research.

The 2nd layer is the research approach which is how the research theory is made using methods like deduction where research is made using a pre-existing theory to test a hypothesis i.e., Does carbon cause global warming? Where you would test this hypothesis by collecting data on the research subject using the theory to prove the hypothesis which is used in fields with pre-existing theories and practices that can be built on. Inductive theory on the other hand begins with data collection which is then analysed to create the theory which is useful for fields where research is limited. Overall, these approaches like inductive can be used to define if the research is around a new or little-known field of research or if the research is simply an improvement on existing methods which deductive would be a beneficial approach.

The 3rd layer is methodological choice which details the data collection method for proving a hypothesis this can be methods such as collecting data as either quantitative or qualitive which can be data that is easily stored as a variable or data that is more quality focused, respectively. These can used one or more methods of collection or even mixing the methods to gather the data.

This then leads to the 4th layer which is the research strategy which is the specific methods of data collection these can be quantitative data collection methods like experiments or surveys or qualitative methods such as case studies.

The 5th layer is time horizons which is the time scale for the study with methods like cross sectional which is where data is collected over a short period of time which can provide data about current events or research for shorter studies. Longitudinal is another method of data collection is where data is collected over a wide period to collect data from multiple periods which is a useful method for studying changes over time.

The final layer being the techniques and procedures is on methods of how the data will be physically or digitally collected i.e., the content of surveys and case study interviews and who or what the data will be collected from as well as whether the data will come from a primary source which involves the researcher collecting the data themselves or secondary data where the data is already available for use from another research project.

Overall, the research onion method allows the researcher to think about how the research will be planned out with what methods and for what reason. For this reason, this research project will use the research onion to guide what research methods will be used and why they will be used.

Due to the nature of narrative AI quality being based on a use case for video games which means that the research into this field could be seen as nuanced requiring a philosophy that is open ended ruling out philosophies like positivism. Overall, the pragmatism philosophy matches the open-ended nature of the research. During the literature review it was found that there was very little research in this field which makes inductive research more appropriate. For the choice of methodology, a mixed method would be beneficial as the research can be both quantitative and qualitative which can be used to determine the popularity of a feature in a game as well as track how the AI changes which can be shown using the survey and experiment methods to see how the AI interacts with both the game world and the users. This study will then be completed in a fixed period using cross sectional studies with the survey completed over the course of 3 days and the experiment being complete in a total of 3 1-hour tests. The data that will be collected will be the opinions of the users of which AI they believe is more realistic between an AI agent using the simple behaviour tree or the agent of an adaptive behaviour tree to compare the differences of opinion between the 2 AI designs. The experiment data collected will be how the behaviour trees have adapted over the 1-hour period which how much of a specific behaviour has been added or removed. The data collected will be from primary sources due to lack of any secondary research on the subject limiting any longitudinal research due to time constraints.

To summarise the research will be in the following points,

* Pragmatic philosophy
* Inductive research
* Mix of qualitive and quantitative data
* Survey and experiment
* Cross sectional
* Primary research

# Chapter 4 Artefact

## 4.1 Design of an Artefact

### 4.1.1 Development plan

With the research gap in the problem analysis identified and the research methods being decided on, the artefact can be designed to solve the gap in research found from the literature review with a ABT system. The artefact will use the ABT method (Colledanchise et al., 2018) to create behaviour trees for the AI agents while as suggested by (Iovino et al., 2022) using a GA to adapt the BTs with the genome of the GA containing nodes which BTs are made up of. The emotional layer will be using the trait-based behaviour system to create the emotional layer as seen with (Short, 2021) and (Sallam, 2023) to define the behaviours of an NPC and make them easily identifiable to potential users. The nodes should not only have the BT composite node like Selector or Sequence nodes for functionality but various behaviour nodes which they NPCs will use which according to (Iovino et al., 2022) will add functionality to the ABT to allow the decision layer to function.

As the research is into the development of an AI system for a game, an environment must be set up to facilitate this system and act as a showcase for how this system would work. While this system would not need to be a fully developed game it would be beneficial to add certain features for the AI agents to interact with which would involve creating simple NPCs to inhabit the world alongside interactable like berry bushes for the NPCs to eat from or the ability for NPCs to talk or fight each other for example. With this idea many behaviours will be planned for development such as the following behaviours Attacking which will attack the target when nearby, detection which searches for a target, walking which simply moves the NPC from point A to B, eating which will eat a berry from a nearby bush, talking where the NPC selects a target NPC to talk to. Idling which is like walking but the NPC moves to a random position and fleeing where the NPC will walk away from a target.

For the design, some diagrams have been made to plan out the development of the project with a use case diagram showing how the events in the game world will affect the NPC’s behaviour trees by triggering specific behaviours i.e., danger nearby may cause the fear trait to activate forcing the NPC to run away from the danger. These events could then trigger a mutation in the ABT genome switching a behaviour out for another with the BT within the ABT being regenerated with this new behaviour potentially affecting the NPC overall. This behaviour mutation for the NPCs in the artefact will be triggered by the NPCs attacking each to simulate a traumatic event which then causes the ABTs in the NPCs to mutate. For more details into how the artefact will be implemented an UML diagram has also been created to show the specific functions and variables in each class as well as how the classes will interact with each other to help aide in the creation of this system.

A diagram of a diagram

Description automatically generated

Figure 15: use case diagram of the proposed ABT system in a game world showing interactions between systems

A diagram of a data flow

Description automatically generated

Figure 16: UML diagram of all systems for the artefact

### 4.1.2 Testing/ Validation Plan

As previously mentions in Chapter 3 the research onion method was used to decide on the validation methods for the project with the artefact validation using a mix of quantitative and qualitative data to prove the findings from the implementation with the use of a survey and an experiment to test the artifact not only functions but can work as an improvement on behaviour trees. To test and compare this effectively both will start will a standard behaviour tree with a set of behaviours that would be typically used for an NPC as seen in Figure 17 with a default tree that only allows the NPCs to interact with the game world in only simple ways with the top node being a selector for 3 different behaviours with the first priority behaviour in the selection being an attack sequence which contains the leaf nodes for detecting, moving to and attacking the target with the sequence node used means that if any of these leaf nodes fail then the attack behaviour has failed. Similarly, the next behaviour being the eating behaviour which also uses a sequence with the leaf nodes of checking hunger, walking to food, and eating food. Finally, a simple idle behaviour. This overall will act as simple BT to adapt in the testing of the artefact as it provides basic functionality to the NPC which can than evolve new behaviours and better show the changing personalities that it will simulate.

A diagram of a diagram

Description automatically generated

Figure 17: diagram showing the default BT nodes with a top node to choose between attacking, eating and idling behaviours

The experiment will simply be gathering the quantity of behavioural changes by counting each mutation in the genome of the behaviour tree to show if the system is functional. This can be done with 3 1-hour tests with a population of 25 NPCs with a mutation rate of 10%. This test also can serve to evaluate the current system developed and the emotional behaviours added to the project as the behaviours could influence the results as they may change the way the NPCs interact with the world and each other.

For the survey it will be designed with several questions that will show a 1-minute video clip of the ABT after 1 hour of evolution and a BT working to compare the changes the NPCs will undertake from an ABT to the BTs which should act more static. The participants will view both clips and will be asked their opinions on which system they believe is more believable and if they felt that the NPC’s AI in the ABT was evolving with the actions shown in the video clips. These video clips will be obtained with the BT clips being simply gathered from the game world whereas the ABT video clips will be gained after a 1-hour test. The layout of this questionnaire can be seen in Figure 18 & Figure 19 with one question for which video showed the more realistic AI and another follow up question into why the participant chose their answer. To make sure the results are not biased the participants will not know which option is for the BT or the ABT helping to give more accurate personal opinions over which system made more realistic NPC AI.

Overall, both tests will show whether or not this system is not only functional but desirable to users showing how the artefact is valid to test the research as they may or may not consider the NPC AI to be valid or not and the experiment will show if certain behaviours are preferred with this system allowing for evaluations of the project to be made.

A screenshot of a video game

Description automatically generated

Figure 18: questionnaire design part 1

A screenshot of a video game

Description automatically generated

Figure 19:questionnaire design part 2

## 4.2 Artefact implementation

As previously mentioned in Section 2.2, Unity was used to implement the artifact as the extra tools for game development it provided helped to create game objects with graphics to aide in showcasing the features of the artefact and the scripting system allowing for features to be added to a game object. This overall has resulting in the project as seen in Figure 20 with a simple game world with a controllable player character as well as slime characters acting as the NPCs in the game world each with the ability to interact with each other and the player using an ABT to make decisions.

How this was developed will be broken down feature by feature starting with the NPCs which is a game object with several scripts with a simple UI script as seen in Figure 22 which has the purpose of showcasing what the NPC is doing which can be seen in Figure 21 with the currently executed behaviour being shown and the hunger or awareness levels being shown in progress bars. the NPC also had a stat manager script as seen in Figure 23 which simply held information about the NPC like heuristic hunger and health values which can trigger certain behaviours.

The NPC also had a behaviour tree script attached as seen in Figure 36 which contained several functions and systems to function starting with the various node scripts which begins with a base node as seen in Figure 25 which simply has a node status Enum for whether the node is currently running or has had a success or failure which has an abstract function for the behaviour returning the node status type. This abstract node is then expanded to create various other nodes with composite nodes like sequence or selection nodes which as previously stated will return a failure if a simple child node fails as seen in Figure 27 and will only return a success if a child node is successful as seen in Figure 26 respectively. These nodes have then been expanded to multiple nodes which will act as the behaviours the NPC AI will utilize which includes simple behaviours like walk (see Figure 28), hunger (see Figure 33), eat (see Figure 35), detection (see Figure 31) and idle but also emotional behaviours such as attacking/ fighting (see Figure 32), fleeing (see Figure 29), and talking nodes (see Figure 34).

The genetic algorithm as seen in Figure 37 which works by first initialising a BT using genes from the list of all node types which creates the default BT genome as previously mentioned in the design. Next is the mutation function which will swap out a gene in the genome with a random chance using the mutation rate of 10% to trigger functioning as the adaptive element for the ABT.

Some features have also been added extra from the designs which serve to help with the development of this system which are non-essential to the ABT which are the player object as seen in which is simply a way to view the game world for testing i.e. How the game world is seen in Figure 20. Another feature added that was not in the designs is a resource manager as seen in Figure 24 which is simply a script that holds references to all game objects which is used to help NPCs use these resources efficiently. One feature added which was not in the design is seen in the add gene function which is used to add node inside sequences and selector nodes which were not accessible without this extra feature. The add gene function allowed for all nodes to be affected by the GA.

## 4.3 Testing and Validation of the artefact

The experiment to test the ABT could evolve the emotional behaviours was conducted with the 3 tests being made and stored in the artefact as seen in the table below with a mutation count of each gene type as seen in Figure 39. This test overall showed that each of the behaviours were being mutated mostly even throughout the tests showing that all the behaviours were being expressed in the ABT. The results of these tests being used to create the example evolved personality which can be valid as seen in works around research about emotional AI like (Kreminski, *et al.* 2020) where the AI developed is investigated with a case study of how the AI performed.

For the questionnaire this was conducted online using Google Forms which as planned was conducted over the 7-day period with the proposed use of questions to find the opinion of the proposed system and see how it compares to conventual methods which in this case is a BT. This was a valid method to use as it has been used in other research into player sentiment like with (Carstensdottir, et al 2019) by using qualitative data like the detailed answers on question 2 to analyse player sentiment at various narrative-based methods in video games which required a qualitative review of a given system to find the faults and benefits they provide.

## 4.4 Critical Evaluation

The questionnaire after completion had a total of 19 respondents with 31.6% choosing Option 1 (BT) as the more realistic AI type and the majority of 68.4% for Option 2 (ABT) making the ABT method more popular than BTs with the test group used in the surveys.

For the second question asking for the reason for the answer to the previous question which while had less responses with only a total of 14 out of 19 total participants a majority who answered also voted for the AI using ABT over the BT with only 6 voting for Option 1 (BT) and the other 8 voting for Option 2 (ABT) however these results were more even overall make the detailed answers more inconclusive overall.

From the responses 2 answers [1,5] stated that the ABT was more erratic than the BT as the reason for their answer with another 4 answers [2,4,9,10] stating that the behaviours seemed more realistic. However, 3 answers [3,12,14] stated the ABT had more purpose to its actions. The final 5 answers [6,7,8,11,13] stated that the ABT seemed more fluid and natural as the BT NPCs all had the same behaviours running at the same. Overall, the results from this question show the reasons why the participants chose the answer they chose with the general opinion on the ABT being more natural and have more purpose for its behaviour.

The ABT overall was capable of adapting an emotional personality as seen from the experiment (see Figure 39) the results show that the ABTs were capable of adapting with any of the emotional behaviours given to the ABT with only around a 5% difference in lowest mutated behaviour (talk) and the highest (eat sequence) with a total of 413 and 436 mutations total showing that the NPC personalities were able to adapt and change over time with the questionnaire (see Figure 40) also showing this result with some respondents stating that the AI was more distinctive to each other than the BT. However, one issue with the project is how the ABT can regularly stop working or slow down the evolving process which can be seen in the experiment data (see Figure 39) with the first 2 tests having no more than 50 mutations of any behaviour during an hour of testing whereas the 3rd test had a much higher mutation count of over 300 mutations for each behaviour. This dramatic increase in mutations may be caused by an issue with the way the behaviours were set up with only the attacking behaviour causing mutations which means that in any NPC had the attack sequence behaviours or any behaviours that would allow the NPC to attack within the attack sequence mutated then this would disrupt the ability for the NPC to attack other NPCs changing this behaviour. This would affect the ABT as the number of NPCs with the attack behaviour declines the slower their personalities will adapt resulting in the different numbers of mutations between tests. This could be rectified with a more diverse set of behaviours which according to (Short, 2021) can help create more emergent behaviours and allow for different ways for the ABT to evolve for example the hunger behaviour could trigger a mutation if the NPC were hungry for a certain period or various other behaviours which could represent various other behaviours in an AI personality.

One issue with the project is with predictability as while as seen with the questionnaire (see Figure 38) a majority preferred the ABT over the BT, many disagreed which of the participants who filled the reason for their answer (see Figure 40) with 7% respondents saying the NPC AI was too erratic and a further 14% of respondents stating that the BTs AI was more realistic. As stated by (Duarte et al., 2020) adaptive AIs like ABTs when used to create NPCs for video games will often be less predictable but may be more realistic or human-like when used in games as the adaption can make the behaviours more fluid and different at the cost of more rigid and designed AI despite both the BT and the ABT starting with the default tree (see Figure 17). This overall explains the mixed responses with many saying the ABT acts “erratic” while others say the ABT is more human-like as adaptive AI will often have issues with predictability.

The trait-based behaviour system was beneficial to develop as stated by (Short, 2021) trait-based behaviours allow for specific behaviours to be easily developed with them also being easily identifiable to users which can be seen in the artefact created as the trait-based behaviour allowed for a number of behaviours to be developed like an angry attack behaviour, fearful behaviour for fleeing or a sociable talking behaviour which could be simply added to the list of possible behaviours for the ABT to mutate with overall allowing for an ease of implementation of well-defined behaviours which was seen in the results of the questionnaire (see Figure 40) with many recognizing the behaviours being used such as the eating and talking behaviours for example however as each behaviour had to be developed specifically this may have been responsible for the low number of behaviours overall which contributed to the issues with adaptability with behaviours. Whereas the OCC system according to (Ferro et al., 2013) would provide an emergent system using a limited number of behaviours which could have resolved this issue as stated by (Short, 2021) the OCC system would make the behaviours harder to recognise which could in turn make the issues with the ABT’s erratic behaviour worse.

The GA was also implemented into the ABT with the predefined genome as suggested by (Liu et al., 2019) which was used to create the default BT to be used for the experiments. The use of the predefined genome worked well at making the NPCs using the ABT to interact with each other in ways that facilitate the mutation which would be more difficult if the initial behaviours for the ABT were fully randomised as this could lead to the issues previously mentioned with the mutations in the genome slowing due to the behaviours selected in the genome not allowing for other NPCs to be mutated likely resulting in the wide range of mutation counts presented in the experiment data (see Figure 39) as some NPCs may not be initialised with the behaviours required to for example have a NPC attack another NPC to cause mutation as the default behaviour has behaviours associated with detection, walking and attacking ordered respectively to allow for the successful use attack on the NPC. Overall, the use of the predefined genome was successful for an emotional ABT.

The anti-bloat method used according to (Colledanchise et al., 2018) was a simple depth-based limitation which was simply to implement into the project by checking the depth of a behaviour to be added in the mutation process and if the behaviour hit the threshold depth, then the behaviour would not be added. This had the effect of stopping the ABT from becoming too large to run during the experiments as the tree would be limited in scope requiring less behaviours each frame which would lessen the performance requirement for the project. This was used instead of the fitness-based model which would remove behaviours based on how often it was used however as also noted by (Colledanchise et al., 2018) this would reduce the ability for behaviours that would be rarely used to be expressed in the ABT’s genome hence the use of a depth-based limitation. This however was not ideal as well as according to (Colledanchise et al., 2018) it reduces the ability for the ABT more complex behaviours which may have been a factor for the ABT’s erratic behaviour discussed in the questionnaire (see Figure 40). Overall, the anti-bloat methods used improved performance at the cost of potentially more complex and emergent behaviours from generating.

# Chapter 5 Conclusion and Future Work

## 5.1 Conclusion

Overall, the research has found that adaptive behaviour trees can be used to create changing personalities in video game NPCs with the artefact created solving the research gap by creating an emotional AI that can adapt to events in a game world to simulate a personality changing overtime. however, the issues around unpredictability and solving the bloating of behaviour trees without sacrificing complex and emergent behaviours continue to be a challenge for ABTs.

With constraints of the duration of this research many ideas were affected such as the number of behaviours that the NPCs could utilize changing the way they interact in more diverse ways instead only 1 behaviour was responsible for affecting other NPCs ABTs also the lack of the proposed event system reduced the number of ways the AI could be adapted overtime.

Testing was done using a mix of experiments and surveys with successfully provided the research benefits of quantitative and qualitative research respectfully. While the testing did provide valuable insight into how the system performed the number of participants for the test was low potentially affecting the results however the opinions of what the users thought of the NPCs AI was useful in exposing some flaws and benefits to the system over more conventual emotional AI systems like BTs.

During this research it was also found that there are multiple kinds of AI capable of creating an adaptive personality such as state machines which can provide a more simplistic AI for smaller scope projects and adaptive HTNs that can be used for a top-down approach to adaptive personalities.

## 5.2 Future work

With this pilot study into ABTs it should be given more extensive testing for a larger number of participants in any future research as if the study had a larger population to test with more issues or potential improvements could be found in the detailed answers and the higher number of respondents could help to make the results of the study more meaningful.

In a future study extra features could be added to the ABT for study such as different anti-bloat techniques like fitness testing for behaviours use as previously mentioned. The proposed event system for in game events that could affect NPCs could also be added in future works to improve the opportunity for the AI to adapt. Another feature that could be added is more controlled adaption of behaviour as any behaviour can be mutated into another which could make the behaviour more unpredictable and unrealistic using methods of designing and controlling the behaviours to evolve into certain other behaviours could be another area of research in the future.

Research should also investigate the various other methods of decision AI to see if adaptive methods like GA and Q-learning can also be applied such as state machines like FSMs and FuSM as well as methods like HTNs as these could provide more options for game AI developers who overwise would not use a BT or ABT in any application of adaptive emotional AI. For emotional AI it could be beneficial to further look into OCC behaviour with the adaption affect thresholds for certain behaviours to be express in the heuristic values for the 5 factors.

Another feature that could be added in future work and research is memory-based AI like the AMN method which could easily be added into the current model as AMN also requires a decision layer and an emotional layer to function which would allow for AMNs to be easily embedded into ABTs and adaptive state machines which use these layers. This would provide a better ability for the AI to plan goals and to allow for more emergent behaviours to be expressed.

# Chapter 6 Appendix

## 6.1 Project implementation

A video game graphics of a video game

Description automatically generated

Figure 20: image showing the completed artefact game world with NPCs using the ABT system

A video game screen with a green object

Description automatically generated

Figure 21: NPC with UI for behaviours, hunger and awareness

## 6.2 UI

A screen shot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

Figure 22: code for UI

## 6.3 Stat manager

A screenshot of a computer program

Description automatically generated A screenshot of a computer program

Description automatically generated

A computer screen with white text

Description automatically generated

Figure 23: stat manager for the NPC

## 6.4 resource manager

A computer screen shot of a program code

Description automatically generated

Figure 24: code for resource manager

## 6.5 Nodes

A screen shot of a computer program

Description automatically generated

Figure 25: base node

A screenshot of a computer program

Description automatically generated

Figure 26: selector node

A computer screen shot of a program

Description automatically generated

Figure 27: sequence node

A computer screen with text on it

Description automatically generated

A screen shot of a computer program

Description automatically generated

Figure 28: walking behaviour

A computer screen with text on it

Description automatically generated

A screenshot of a computer program

Description automatically generated

Figure 29: flee behaviour

A computer screen shot of a program code

Description automatically generated

A computer screen shot of a program

Description automatically generated

Figure 30: idle behaviour for wandering

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen with text and numbers

Description automatically generated

Figure 31:detection node

A screen shot of a computer program

Description automatically generated

Figure 32: attack node

A computer screen shot of a program

Description automatically generated

Figure 33: hunger node

A screen shot of a computer program

Description automatically generated

Figure 34: node for talking to other NPCs

A screenshot of a computer program

Description automatically generated

Figure 35: eating behaviour

## 6.6 behaviour tree

A computer screen shot of a program

Description automatically generated

A computer screen shot of a program

Description automatically generated

A computer screen shot of a program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Figure 36: behaviour tree

## 6.7 genetic algorithm

A computer screen shot of a program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Figure 37: genetic algorithm

## 6.8 Test Data

A pie chart with text

Description automatically generated

Figure : responses to the questionnaire

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Attack | Eat | Idle | Flee | Talk | Walk |
| Test 1 | 32 | 46 | 35 | 36 | 45 | 33 |
| Test 2 | 43 | 40 | 47 | 35 | 41 | 41 |
| Test 3 | 353 | 350 | 337 | 347 | 327 | 351 |
| Total | 428 | 436 | 419 | 418 | 413 | 425 |

Figure 39: experiment data

|  |  |  |
| --- | --- | --- |
| ID | Reason for answer | Answer |
| 1 | The states seemed to change much more frequently in the second video which in terms of gameplay perspective would be quite disorienting, however this could be because of the text updating which may not be noticeable in a finished product. | Option 1 |
| 2 | The first option they are looking for food and second they just wait for it to regrow | Option 1 |
| 3 | the blobs in video 1 moved randomly and changed behaviour the same. in video 2 there was a clear motive for behaviour and movement was more purposeful | Option 2 |
| 4 | Transition is smooth. | Option 1 |
| 5 | I have chosen video 1 as the movement appears more natural and random with fewer intersecting characters at any time | Option 1 |
| 6 | Less uniform, can't notice patterns | Option 2 |
| 7 | The movement of 1 is too coordinated and didn’t feel organic | Option 2 |
| 8 | The jumps seemed more random whereas in the first, several were jumping at the same time | Option 2 |
| 9 | Option 2’s movement seemed a little static in comparison | Option 1 |
| 10 | Not 100% sure, the movement in option 1 seems more fluid | Option 1 |
| 11 | More randomised | Option 2 |
| 12 | More personality in option 2, they seem to have more thought rather than just mindless following of paths as is seen in option 1 | Option 2 |
| 13 | Action more distinctive | Option 2 |
| 14 | The AI seemed more calm, collective, and social with them being in set groups | Option 2 |

Figure : table of the reason for answers given in the questionnaire

# Chapter 7 References

Agliata, F. et al. (2022) Adding variety in npcs behaviour using emotional states ... - researchgate, ResearchGate. Available at: https://www.researchgate.net/publication/339644545\_ADDING\_VARIETY\_IN\_NPCS\_BEHAVIOUR\_USING\_EMOTIONAL\_STATES\_AND\_GENETIC\_ALGORITHMS\_THE\_GENIE\_PROJECT (Accessed: 26 July 2023).

Ampatzidou, C. (2018) ‘Reinventing the rules: Emergent gameplay for Civic Learning’, *The Hackable City*, pp. 187–203. doi:10.1007/978-981-13-2694-3\_10.

Belle, S., Gittens, C. and Graham, T.C.N. (2019) ‘Programming with affect: How behaviour trees and a lightweight cognitive architecture enable the development of non-player characters with emotions’, *2019 IEEE Games, Entertainment, Media Conference (GEM)* [Preprint]. doi:10.1109/gem.2019.8811542.

Belle, S., Gittens, C. and Nicholas Graham, T.C. (2022) ‘A framework for creating non-player characters that make psychologically-driven decisions’, *2022 IEEE International Conference on Consumer Electronics (ICCE)* [Preprint]. doi:10.1109/icce53296.2022.9730383.

Bender, E.M. et al. (2021) ‘On the dangers of stochastic parrots’, Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency [Preprint]. doi:10.1145/3442188.3445922.

Carstensdottir, E., Kleinman, E. and El-Nasr, M.S. (2019) ‘Player interaction in narrative games’, *Proceedings of the 14th International Conference on the Foundations of Digital Games* [Preprint]. doi:10.1145/3337722.3337730.

Child, C. (2016) *NPCs as people, too: The extreme AI personality engine - researchgate*, *NPCs as people, too: The extreme AI personality engine*. Available at: https://www.researchgate.net/publication/308265024\_NPCs\_as\_People\_Too\_The\_Extreme\_AI\_Personality\_Engine (Accessed: 03 July 2023).

Colledanchise, M., Parasuraman, R. and Ogren, P. (2018) ‘Learning of behavior trees for autonomous agents’, *IEEE Transactions on Games*, 11(2), pp. 183–189. doi:10.1109/tg.2018.2816806.

Dai, K. et al. (2020) ‘Injecting affective computing into online virtual training through FPS Games’, 2020 IEEE International Conference on Progress in Informatics and Computing (PIC) [Preprint]. doi:10.1109/pic50277.2020.9350814.

Dey, R. and Child, C. (2013) ‘QL-BT: Enhancing behaviour tree design and implementation with Q-learning’, *2013 IEEE Conference on Computational Inteligence in Games (CIG)* [Preprint]. doi:10.1109/cig.2013.6633623.

Duarte, F.F. et al. (2020) ‘A survey of planning and learning in games’, Applied Sciences, 10(13), p. 4529. doi:10.3390/app10134529.

Georgeson, J. (2016) NPCs vote! changing voter reactions over time using the extreme AI personality engine, arXiv.org. Available at: https://arxiv.org/abs/1609.05315 (Accessed: 26 July 2023).

Hassanat, A. et al. (2019) ‘Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach’, Information, 10(12), p. 390. doi:10.3390/info10120390.

Immanuel, S.D. and Chakraborty, U.Kr. (2019) ‘Genetic algorithm: An approach on optimization’, 2019 International Conference on Communication and Electronics Systems (ICCES) [Preprint]. doi:10.1109/icces45898.2019.9002372.

Iovino, M., Scukins, E., Styrud, J., Ögren, P. and Smith, C. (2022) 'A survey of Behavior Trees in robotics and AI', *Robotics and Autonomous Systems,*154, pp. 104096 Available at: 10.1016/j.robot.2022.104096.

Jang, B. *et al.* (2019) ‘Q-Learning Algorithms: A comprehensive classification and applications’, *IEEE Access*, 7, pp. 133653–133667. doi:10.1109/access.2019.2941229.

Kreminski, M. *et al.* (2020) ‘Why are we like this?: The Ai Architecture of a co-creative storytelling game’, *International Conference on the Foundations of Digital Games* [Preprint]. doi:10.1145/3402942.3402953.

Li, J. *et al.* (2016) ‘Interact with robot: An efficient approach based on finite state machine and mouse gesture recognition’, *2016 9th International Conference on Human System Interactions (HSI)* [Preprint]. doi:10.1109/hsi.2016.7529632.

Li, L. and Campbell, J. (2010) ‘Emotion modeling and interaction of npcs in virtual simulation and games’, *International Journal of Virtual Reality*, 9(4), pp. 1–6. doi:10.20870/ijvr.2010.9.4.2784.

Liu, Y. et al. (2019) ‘Multi-objective AGV scheduling in an automatic sorting system of an unmanned (intelligent) warehouse by using two adaptive genetic algorithms and a multi-adaptive genetic algorithm’, PLOS ONE, 14(12). doi:10.1371/journal.pone.0226161.

Meiring, G. and Myburgh, H. (2015) ‘A review of intelligent driving style analysis systems and related artificial intelligence algorithms’, Sensors, 15(12), pp. 30653–30682. doi:10.3390/s151229822.

Melnikovas, A. (2019) Towards an explicit research methodology: Adapting Research Onion Model for Futures Studies \* Journal of Futures Studies, Journal of Futures Studies. Available at: https://jfsdigital.org/articles-and-essays/2018-2/towards-an-explicit-research-methodology-adapting-research-onion-model-for-futures-studies/ (Accessed: 15 September 2023).

Nitisiri, K., Gen, M. and Ohwada, H. (2019) ‘A parallel multi-objective genetic algorithm with learning based mutation for railway scheduling’, Computers &amp;amp; Industrial Engineering, 130, pp. 381–394. doi:10.1016/j.cie.2019.02.035.

Ontanon, S. and Buro, M. (2015) Adversarial hierarchical-task network planning ... - ACM Digital Library, Adversarial hierarchical-task network planning for complex real-time games. Available at: https://dl.acm.org/doi/abs/10.5555/2832415.2832479 (Accessed: 09 September 2023).

Popescu, A., Broekens, J. and van Someren, M. (2014) ‘Gamygdala: An emotion engine for games’, IEEE Transactions on Affective Computing, 5(1), pp. 32–44. doi:10.1109/t-affc.2013.24.

Postnikov, E.V. *et al.* (2019) ‘Application of fuzzy state machines to control players in virtual soccer simulation’, *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)* [Preprint]. doi:10.1109/eiconrus.2019.8657109.

Rapp, A., Curti, L. and Boldi, A. (2021) ‘The human side of human-chatbot interaction: A systematic literature review of ten years of research on text-based Chatbots’, International Journal of Human-Computer Studies, 151, p. 102630. doi:10.1016/j.ijhcs.2021.102630.

Rehakova, L. and Neruda, R. (2018) ‘Utilization of genetic programming to solve a simple task network planning problem’, 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) [Preprint]. doi:10.1109/smc.2018.00619.

Rozikin, M., Dijaya, R. and Taurusta, C. (2021) ‘Education Game Indonesian old Museum Explorer using Fuzzy State Machine’, *Journal of Physics: Conference Series*, 1764(1), p. 012059. doi:10.1088/1742-6596/1764/1/012059.

Sallam, M. (2023) ‘CHATGPT utility in healthcare education, research, and practice: Systematic review on the promising perspectives and valid concerns’, Healthcare, 11(6), p. 887. doi:10.3390/healthcare11060887.

Saunders, M.NK. et al. (2021) ‘Chapter 4: Understanding research philosophy and approaches to theory development’, in Research methods for business students. 8th edn. Harlow, Essex: Pearson Education Limited, pp. 128–171.

Sekhavat, Y.A. (2017) ‘Behavior trees for computer games’, *International Journal on Artificial Intelligence Tools*, 26(02), p. 1730001. doi:10.1142/s0218213017300010.

Sharples, M. (2023) *Towards social generative AI for education: Theory, practices and Ethics*, *arXiv.org*. Available at: https://arxiv.org/abs/2306.10063 (Accessed: 09 July 2023).

Short, T. (2021) ‘Designing stronger AI personalities’, Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 13(2), pp. 111–117. doi:10.1609/aiide.v13i2.12973.

Short, T.X. and Adams, T. (2019) *Procedural storytelling in Game Design*. Boca Raton, New York: CRC Press.

Spraragen, M. and Madni, A.M. (2014) ‘Modeling of emotional effects on decision-making by Game Agents’, *Procedia Computer Science*, 28, pp. 736–743. doi:10.1016/j.procs.2014.03.088.

Xie, L., Chen, Y. and Chang, R. (2021) ‘Scheduling optimization of prefabricated construction projects by genetic algorithm’, *Applied Sciences*, 11(12), p. 5531. doi:10.3390/app11125531.

Yorita, A. *et al.* (2019) ‘Self-adapting chatbot personalities for better peer support’, *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)* [Preprint]. doi:10.1109/smc.2019.8914583.