2019 IEEE Games, Entertainment, Media Conference (GEM)

Programming with Affect: How Behaviour Trees and a Lightweight Cognitive Architecture Enable the Development of Non-Player Characters with Emotions

Shakir Belle
Dept. of Computer Science,
Mathematics and Physics
University of The West Indies
Bridgetown, Barbados
shakir.belle@mycavehill.uwi.edu

Curtis Gittens
Dept. of Computer Science,
Mathematics and Physics
University of The West Indies
Bridgetown, Barbados
curtis.gittens@cavehill.uwi.edu

T.C. Nicholas Graham
School of Computing
Queen's University
Kingston, Ontario, Canada
nicholas.graham@queensu.ca

***Abstract*—There is little support in game development tools for implementing emotion and mood in non-player characters. In this paper, we present a tool that integrates behaviour trees into an emotion modeling framework, based on ALMA, to add emotion and mood to NPCs. We introduce two new nodes to traditional behaviour trees, the Emotion Adder, which triggers emotions in NPCs and the E-Selector, which incorporates the NPC's mood in the decision-making process. This EMoBeT framework arms developers with a tool that will assist them in integrating and manipulating psychologically valid moods and emotions of NPCs using the familiar behaviour tree model.**

Keywords—non-player characters, emotion, mood, games, psychologically-based behaviour, behaviour tree

Acknowledgement

We would like to express our thanks towards the Canada-CARICOM Leadership Scholarships Program for making our collaboration and by extension the entire project a possibility.

I. Introduction

This paper introduces a novel, psychologically-based emotion/mood modelling system for non-player characters (NPCs). To present vibrant and believable behaviour, NPCs should react to stimuli such as danger, insults, complements, or humour, through visible modification of their mood. Current technologies for programming NPC behaviour do not, however, treat mood as a first-class property: specifically, they do not provide mechanisms for modifying an NPC's mood, or for allowing the NPC's current mood to influence its behaviour.

To address this problem, we present EMoBeT, an extension of behaviour trees, that allows direct programming of mood. Through examples, we show how EMoBeT can be used to model an NPC's mood over time, to influence that mood in response to stimuli, and to guide behaviour based on current mood. EMoBeT is implemented via a framework which is based on the ALMA layered model of affect system [10]. Through a fast, custom implementation of ALMA, it is possible to implement a rich model of emotion that can be computed in real-time on standard gaming computers.

Emotion is generally defined as a human's instantaneous reaction to an experience, while *mood* is the longer-term, more stable mental state of a human [30]. Based on this, *personality* is then defined as a long-term affective state.

EMoBeT can be used to tackle issues such as the impact of emotions and mood on an NPC's choices, and to simulate decision-making in social situations, such as modeling the effects that peer pressure has on behavior. It can also be used in games to create more relatable NPCs.

EMoBeT extends the behaviour tree model with emotion and mood nodes to enable NPCs to make emotion-based decisions. We choose to extend behaviour trees because they are frequently used by game developers to model NPC behaviour. The EMoBeT framework has been extended with new behaviour tree node types providing the ability to manipulate emotions within an NPC, change its mood, and use its mood to make decisions.

The paper is organized as follows. In Section II, we survey existing approaches to modeling emotion in NPCs. Section III introduces our extensions to behaviour trees to allow programmers to treat mood as a first-class property of NPCs. In Section IV, we then present the architecture of EMoBeT, showing how NPC's emotions, mood, personality and memory are modeled, and how these are tied into a behaviour tree system. Finally, Section V provides examples of the use of EMoBeT's behaviour trees in scenarios where NPCs' moods are impacted by the player's behaviour, which in turn affects the NPC's actions.

II. Related Work

Researchers have worked to create more vibrant and believable NPCs with richer behaviours within games. The solutions that they have arrived at follow different perspectives but reach a similar conclusion: if games are to improve, the NPCs also need to be better. Popescu et al. state that if an emotional aspect were present in NPCs, there would be an increase in the variations of NPC behaviour that could create more interesting games [11]. Hudlicka states that focusing on producing more believable NPCs in games by using emotions would be a step toward more engaging games, both for entertainment and serious gaming purposes [12].

Emotions play a key role in determining how individuals behave, and this should be the same for NPCs [13]. An emotional connection between NPCs and players is needed to create more compelling experiences for the player [6]. Current games accomplish this through techniques such as cinematics, non-playable sequences, and rich dialogue, while few games feature NPCs whose behaviours change based on emotion [6].

We view systems that provide NPCs with emotion, mood or personality, through two perspectives: (i) the way the system is implemented and; (ii) the ease with which game developers can use the system in the creation of games. We explore four main approaches that game developers have used to add emotion and or mood to NPCs: (A) affective gaming; (B) behaviour trees [14], (C) black box systems [11] and; (iv) approaches based on cognitive architectures [15].

A. *Affective Gaming*

The ability of games to extract behavioral cues from the player is a significant factor in providing a better gaming experience. Game developers and researchers have devoted considerable resources to developing *affective behaviours* in NPCs [11,12]. NPCs that are able to respond to the player's behavioural cues have greater believability and behavioural consistency, which enhances the player's sense of immersion [13]. With affective gaming, the focus is on detecting the player's psychophysiological state, and not on modelling the NPC's emotional behaviour. While this approach leads to more believable NPCs, it does not address the issue of modelling psychologically-based emotions in NPCs.

B. *Behaviour Trees*

Champandard describes behaviour trees as a directed acyclic graph that is made up of various types of nodes [16]. The behaviour tree is traversed from the topmost node downwards. The various node types can return one of three states after being executed: *successful*, *failure,* or *running*. Behaviour trees have been widely implemented in commercial games [17], tested in games such as StarCraft [18] and are also built into the popular Unreal Engine [19].

Behaviour trees have become an integral part of action selection in commercial games because of their scalability and extensibility [20]. Johansson and Dell'Acqua propose a number of behaviour tree extensions in [14], one of which is a new type of selector node called the *emotional selector*. The *emotional selector* node works by sorting its children nodes into an order based on three factors: the planning effort, the risk involved, and the time it takes to perform the actions. This approach to decision-making is good for games where the NPC must make tactical decisions in role-playing games (RPGs); but it is not designed for games where more social interactions are required. Additionally, the formulae used in the behaviour trees are based on an intuitive, instead of psychological approach.

C. *Black Box Systems*

Black box systems are those where the developer only has to be concerned with input and output. One example of a black box system that adds emotion to NPCs is the GAMYGDALA emotion engine [11]. GAMYGDALA is an AI sub-component that can identify the suitable emotion for the NPC to express based on the current event it is experiencing. It is considered to be a "black-box appraisal" engine because game developers, without knowing about emotional appraisal, will still obtain psychologically-plausible emotions within their games. This system has a psychological foundation in emotion theory based on the Ortony, Clore & Collins (OCC) emotion model [21]. The OCC model provides a structure of emotions and the conditions that elicit these emotions, as well as variables which may impact the intensities of the emotions. The authors indicate that the engine still has the ability to perform efficiently in a game and should be easy for game developers to use.

D. *Cognitive Architectures*

Cognitive architectures are a library of tightly-integrated systems that are designed to implement intelligent behaviour [22]. They are built in an effort to create programs that could solve problems from multiple domains and adapt to various situations – essentially, to model human cognitive processes [15, 23]. Even though cognitive architectures were not developed to be used in games, mature cognitive architectures such as SOAR and ACT-R have been integrated into games and game engines [24, 25].

Emotion is one of the key competencies that cognitive architectures model, and using this, a developer may be able to create emotional NPCs. Djordjevich et al. describe a system that uses a full cognitive architecture called the Sandia Human Embodiment and Representation Cognitive Architecture (SHERCA) [26]. They implemented the system in a training game called *Ground Truth: Toxic City* and noted an increase in the realism of the training game when comparing the use of non-emotional NPCs to the SHERCA-based NPCs. The NPCs were found to select the emotions that were expected of them.

Given the work done to date, games do not to make full use of the capabilities that a cognitive architecture offers, and this tends to lead to wastage of implementation and processing resources. Smart et al. indicated that they faced integration challenges with the ACT-R cognitive architecture because the programming language used was not one typically used in gaming [25]. Cognitive architectures also require significant resources to run, which can negatively impact the runtime performance of the game.

E. *Summary*

Each approach to providing NPCs with emotion and or mood has advantages and disadvantages. While behaviour trees are easy to understand and use, they may be limited in their usefulness with certain types of games. With black box approaches, by design, the game developer does not truly understand the system and does not have the ability to easily tweak behaviour or add additional functionality. Lastly, cognitive architectures can simulate affect with greater psychological accuracy; however, integration as well as game performance - where speed and timeliness are vital - remain significant challenges.

III. IMPLEMENTING EMOTION AND MOOD THROUGH THE EMOBET FRAMEWORK

We now show how EMoBeT's behaviour trees can be used to add emotion and mood to NPCs and influence their in-game behaviour. Three requirements influence the operation of our framework. These are *influence*, *persistence* and *interface*. First, the emotions felt by the NPC must be able to *influence* its behaviour and decision-making processes. We accomplish this by using mood as a mediating element. In the framework,

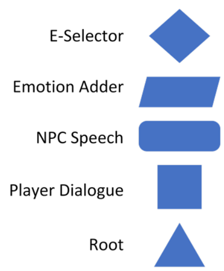

Figure 1A: Behaviour tree nodes

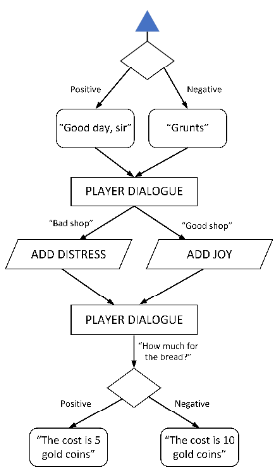

Figure 1B simple behaviour tree example

the emotions that have been triggered by external forces are combined to obtain an average emotional state. This emotional state impacts the mood of the NPC, which in turn exerts its influence over the decision-making process. As a result of this design decision, gameplay influences the mood and decision-making process of the NPC, since all emotions are triggered by events within the game.

The second requirement is *persistence*. The emotions felt by the NPC should not just be a single event that disappears after being experienced; instead, emotions should persist for further use as needed. The EMoBeT framework stores all emotions that the NPC experiences and groups them to form the emotional state of the NPC. The emotional state is present for as long as the NPC exists with the framework providing the facility to track emotions of multiple NPCs as the game progresses.

Our final requirement was to use behaviour trees to *interface* with the EMoBeT framework. This provides game developers with a familiar tool that can assist them in understanding the framework.

A. Behaviour Tree Extensions

We have extended the traditional behaviour tree formalism by incorporating two new nodes, *Emotion Adder* and *E-Selector*.

1) The Emotion Adder Node

The Emotion Adder, as the name implies, allows an emotion to be added to the NPC. For example, in Figure 1B, an Emotion Adder node is used to add the distress emotion to a shopkeeper NPC who is told by a player that his shop is bad. Another Emotion Adder node is used to add joy to the NPC's emotion in response to being told that his shop is good.

More specifically, the developer provides a) the *set of possible emotions* that can be generated from the Emotion Adder and; b) the respective *triggers* for the possible emotions. The ***set of possible emotions*** that can be defined within the Emotion Adder is based on the OCC model of emotion. The ***triggers*** are links to the events (parent nodes) that caused the emotion.

The Emotion Adder follows a Player Dialogue node. A Player Dialogue node presents a dialogue choice to the player. In Figure 1B, the choice is between insulting or praising the NPC's shop. The Emotion Adder accepts the choice made by the Player Dialogue node and generates a new emotion to add to the NPC. Emotion Adders must hold the possible emotions that can be generated by the player's choice as well as the choices that will trigger the emotions. Emotion Adder nodes have a single child.

2) The E-Selector Node

The E-Selector node is used to select an NPC action based on the NPC's current mood. For example, in Figure 1B, an E-Selector is used to allow the NPC shopkeeper's emotion to guide the prices the NPC offers to the player; if the NPC has a positive mood, it charges 5 pieces of gold for a loaf of bread, while charging 10 gold pieces if the NPC has a negative mood.

The E-Selector node checks the current mood of the NPC and then chooses the child that best fits the NPC's state. There are eight possible moods that are split into negative and positive moods. E-Selector nodes have exactly two children, representing the opposing (positive/negative) moods. The E-Selector chooses the child whose label matches the NPC's current mood state.

3) How the Added Nodes Function in the Behaviour Tree

These two new node types work with more traditional behaviour tree nodes to provide the NPC with emotional responses. The emotions of the NPC are triggered by actions of the player. The player chooses a course of action and the Emotion Adder invokes the next step in the process. The choice that the player makes triggers the corresponding emotion stored within the Emotion Adder, creating the emotion with all relevant information – including the trigger. The EMoBeT framework then processes the new emotion and updates the mood of the NPC. At some point in the future – not necessarily even in the same behaviour tree – an E-Selector accesses the current mood of the NPC and based on this, decides which action the NPC should take. Action-selection criteria are generally split into positive or negative categories to help with the selection process. The action node selected then carries out the respective stored actions.

IV. EMoBeT Framework Architecture

The Emotion Modelling Behaviour Tree (EMoBeT) framework leverages behaviour trees to create an easy-to-use tool for game developers to modify NPCs' mood and to use this mood to guide NPCs' behaviour. As described above, this is accomplished by adding two node types to the behaviour tree. This extended behaviour tree formalism is coupled with a light-weight version of the ALMA cognitive architecture to make it possible to have the benefits of simulating mood and

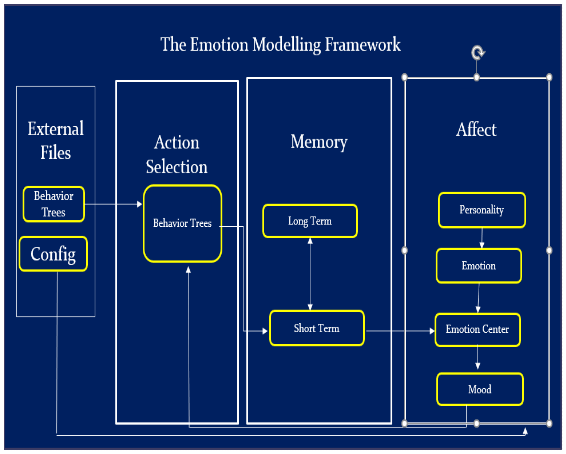

Figure 2: The framework architecture

emotion without the cost of a heavy-weight system that would be too resource-intensive to run on typical gaming hardware. In the following subsections, we provide an overview of the subsystems that make up the Memory and Affect components of the architecture. The Action Selection component is then discussed in section IV.

A. The Personality Subsystem

Any NPC created using EMoBeT is given a personality based on the five-factor model of personality – Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism [27, 28]. The personality can either be defined by the developer or assigned randomly. The personality of the NPC is responsible for its initial or default mood. The personality also effects emotion intensity. For example, having a highly neurotic personality will cause the intensities of negative emotions to be higher when compared to having a low-neurotic personality.

B. The Emotion Subsystem

Emotions are short-term affective states that are produced by the NPC's interactions with the world. Their purpose in EMoBeT is to influence the NPC's mood, which in turn influences the decisions the NPC makes. The emotion itself contains multiple parts, the most important being: (i) the type of emotion, (ii) the intensity of the emotion, and (iii) what triggered the emotion. The intensity of the emotion is responsible for the size of the impact the emotion will have on the NPC.

C. The Mood Subsystem

In the EMoBeT framework, mood is defined as a mid-term affective state that is also the deciding factor for all NPC decisions. We adopt Gebhard's definition of mood [10], based on the three values of Pleasure (P), Arousal (A) and Dominance (D). Each of the three values range from -1 to 1, with +P and –P being pleasant and unpleasant, +A and –A for aroused and unaroused, and +D and –D for dominant and submissive respectively. In combination, the PAD values generate eight octants in 3D space: Exuberant, Dependent, Relaxed, Docile, Hostile, Anxious, Disdainful or Bored (Table 1).

TABLE 1.

MOOD OCTANTS IN PAD SPACE

Mood Octants			
+P+A+D	Exuberant	-P-A-D	Bored
+P+A-D	Dependent	-P-A+D	Disdainful
+P-A+D	Relaxed	-P+A-D	Anxious
+P-A-D	Docile	-P+A+D	Hostile

Mood also has its own intensities: slight, moderate and full. The intensity of the mood is determined by how far it is from the center point of the mood space – the origin (0, 0, 0) in 3D space. The default mood is generated based on the personality of the NPC and is changed by the impact that emotions have on it. Emotions are also mapped to a point in PAD space. Each emotion that the NPC "feels" is stored and combined to form the virtual emotion center. The virtual emotion center also has a point in PAD space and this point along with the current mood's PAD value determines how the mood is updated. When the current mood is between the center point and the virtual emotion center (the PAD of all the combined emotions in the NPC) the current mood moves towards the emotion center. In the situation where the emotion center is between the zero point and the mood, the mood is pushed further into the octant within the PAD space. This example of the mood going further into the octant represents the NPCs mood becoming more intense as it experiences emotions that align with the current mood. The level of intensity in the emotion center determines how much the current mood moves, be it towards or away from it.

D. The Emotion Center

The emotion center is vital to the computation of moods in the ALMA system. It has a PAD value just like the mood and is used with the current mood of the NPC and the center point in PAD space to determine the next mood of the NPC. If the emotion center is found between the current mood and the center point, the future mood will be pushed further into its current octant. If it is located anywhere other than between the current mood and the center point, the mood will shift towards the emotion center. The reader is referred to Gebhard for details on the function of the emotion center [10].

E. The Memory Module

The purpose of the memory module is to give the NPC the ability to recall prior interactions and the emotions that resulted from those interactions. Memory is divided into long- and short-term and has the ability to decay. The decay function is based on the Ebbinghaus forgetting curve [29] to provide a better model for simulating loss of memory. The decay function manages the intensities of the emotions stored within the NPC's memories. Each interaction that an NPC has generates an emotion and a response. These emotions and the situation which triggered the emotions, along with the parties involved, are then stored within the NPC's memory. These memories come into play when the NPC encounters a player it has already met. When interacting with such a player, it will recall the strongest (the highest intensity) memory it has for the player, bringing to the surface and triggering the emotions associated with that memory, which then effects the current mood of the NPC.

F. Categorizing the EMoBeT framework

The EMoBeT framework spans three of the categories of systems identified in Section II: systems that use behaviour trees for NPC modelling; systems that use affective measures to facilitate more stimulating game play; and black box systems where the developer only has to take into consideration the inputs and outputs. EMoBeT draws on aspects of GAMYGDALA [11] and Emotional Behaviour Trees [14]. Unlike Emotional Behaviour Trees [14], whose approach is not based on psychological theory, the EMoBeT framework is based on the Five Factor Model [28] for its personality implementation and the Ebbinghaus forgetting curve for the memory decay [29]. Additionally, our approach to decision making differs from Emotional Behaviour Trees [14], which focuses on finding optimal responses to a given situation, while our approach focuses on simulating human-like emotions and having these emotions effect the decision-making process regardless of whether it is an optimal outcome for the NPC.

The aim of the EMoBeT framework is to be a tool that contains components of a cognitive architecture that would be useful to games – such as emotion, mood, personality and memory – in a more light-weight implementation so that NPC responses are not pre-scripted responses. Having the cognitive architecture driven by the NPC's mood adds a layer of unpredictability aimed at creating a more engaging NPC. This, combined with prior experiences of the NPC impacting its current mood, provides even more unscripted variations that effect its behaviour.

V. Scenarios and Evaluation

As discussed in section IV, EMoBeT was designed to meet three requirements: persistence of emotions and mood; the ability for emotions/mood to influence NPC behaviour; and the ability for the gameplay to modify NPC emotion/mood. Through examples, we now show that our framework meets these requirements to provide the desired functionality.

Figure 3 is a sample behaviour tree that incorporates our new node types. This example shows the usefulness of the EMoBeT framework in a gaming example, by demonstrating the ability of the framework to provide psychologically-based emotions and moods that impact a persons' behavior.

Figure 3 Behaviour Tree Scenario 1 – Father & Son conversation

Table 2.

BEHAVIOUR TREE SCENARIO 1 TABULAR FORMAT

Player Dialogue Steps and Options		
1	Player Choice +ve	**"You had a good game today son"**
	Player Choice -ve	"We need more training for you boy"
	NPC Emotion +ve	**Joy**
	NPC Emotion -ve	Disappointment
	NPC Reaction +ve	**"Thanks dad I did my best"**
	NPC Reaction -ve	"That is normal, if you came more…"
2	Player Choice +ve	**"I'll be there more"**
	Player Choice -ve	"I can't make the next one."
	NPC Emotion +ve	**Satisfaction**
	NPC Emotion -ve	Disliking
	NPC Reaction +ve	**"You are the best"**
	NPC Reaction -ve	"ughh, whatever"

This implements a simple game scenario where the player interacts with an NPC and the player's actions and choices affect the emotional state of the NPC. The player is presented with decisions throughout the interaction with the NPC. Each scenario is represented by its own behaviour tree. The scenarios aimed to show how the player's interactions with the NPC affected the NPC's mood and ultimately its decision making. The scenario also shows that the NPC's mood persists throughout the life span of the NPC.

Table 2 gives a condensed version of the behaviour tree in Figure 3 that also shows the player and NPC selections. The blue-bolded text represents the choices the player makes, the emotions triggered by these choices and how the NPC reacts to the choices. We use the tabular version of the behaviour tree to discuss the results of the other scenarios that made up our experiments.

Table 3 shows the mood change of the NPC as the behaviour tree represented by Table 2 is executed. In this behaviour tree the player plays the part of a father talking to his son after a basketball game. The player has the choice to either compliment or berate the NPC son. As shown in Table 2, the player selects a compliment, eliciting the emotion of *joy* from the NPC instead of disappointment When the emotion of *joy* is triggered within the NPC, it causes the previously empty Emotion Center to be filled with the PAD value representing that emotion. Filling this value into the Emotion Center moves the mood of the NPC in a positive direction, which in this case reinforces the positive octant that the mood was in, causing the NPC to have a positive reaction towards the player.

Next, Figure 3 shows that the player was presented with another choice, either assuring the son that he will be attending this next game or telling him that he can't make it.

The player choses to say that they can make the game, eliciting the emotion of satisfaction from the NPC (Table 2). This emotion causes the pleasure and dominance in the emotion center to rise, while arousal falls. This is expected, as the PAD value of satisfaction is (0.3, -0.2, 0.4), which after merging with the previous emotion center, accounts for the respective changes.

TABLE 3.

BEHAVIOUR TREE SCENARIO 1 MOOD CHANGE OUTCOME.

EMOTIONS ELICITED		JOY	SATISFACTION
Pleasure	0.495	**0.528364**	**0.561728**
Arousal	0.115	**0.131682**	**0.148364**
Dominance	0.268	**0.276341**	**0.284682**
Mood	Exuberant	Exuberant	Exuberant
Mood Intensity	Moderate	Moderate	Moderate
Emotion Center Pleasure		**0.4**	**0.484261**
Emotion Center Arousal		**0.2**	**0.16456**
Emotion Center Dominance		**0.1**	**0.276452**

Although the arousal in the emotion center decreased, it was still positive overall (Table 3), which led to all three values of the mood showing a positive increase. Since the NPC is still in a positive mood state, it shows a positive reaction, reflected by experiencing the positive reaction. (Table 2), rather than compensating for a negative emotion with a high-sugar meal.

Table 4 shows the tabular format of one of the behaviour trees that make up Scenario 2 that was used to test the framework. This behaviour tree represents a scenario where the player and the NPC are out to choose lunch and are having a conversation at the same time. The first node, the Player Dialogue node, contains a set of actions that the player can choose from, which elicits different emotions from the NPC.

The player can choose to either compliment the NPC or severely chastise it for the outcome of a meeting. The children of this Player Dialogue node are both Emotion Adder nodes, specifying the emotion that will be triggered within the NPC depending on the choice the player made. If the player was complimentary toward the NPC, this triggers the emotion of pride in the NPC. If the player was critical of the NPC, resentment is added. The E-Selector nodes follow the Emotion Adder nodes, and it is these that select the NPC's reaction based on its current mood. After the NPC reacts, the player then has another opportunity to positively or negatively interact with the NPC through a second Player Dialogue node. Again, based on the interaction, either Reproach or Satisfaction emotions are triggered in the NPC, leading the NPC either to select either healthy or unhealthy food.

Table 5 shows the mood changes of Scenario 2. The first column of Table 5 shows the mood of the NPC before the scenario starts. Note that this column is populated based on prior interactions with the player character and its previous scenario mood. The second column shows what happens when the emotion *Pride* is triggered within this NPC. Each of the emotion centers' PAD values increase at this point because the emotion pride was added. With the current emotion center being positive in both pleasure and arousal, and the previous mood also being positive in pleasure and arousal, the new pleasure and arousal are pushed further in the

Table 4.

BEHAVIOUR TREE SCENARIO 2

Player Dialogue Steps and Options		
1	Player Choice +ve	**"You did good work... today"**
	Player Choice -ve	"That was terrible, simply dreadful!"
	NPC Emotion +ve	**Pride**
	NPC Emotion -ve	Resentment
	NPC Reaction +ve	**"Thank you, I'm glad it went well"**
	NPC Reaction -ve	"Did I?"
2	Player Choice +ve	"I always had faith"
	Player Choice -ve	**"You do usually mess up"**
	NPC Emotion +ve	Satisfaction
	NPC Emotion -ve	**Reproach**
	NPC Reaction +ve	Healthy food choice
	NPC Reaction -ve	**High-sugar meal**

positive direction. On the other hand, though the dominance value in the emotion center has become more positive, it was still a negative value. The dominance value of the pre-scenario mood was also negative, which causes the dominance value for the new mood to decrease. These changes cause the mood intensity to increase from being slightly dependent to moderately dependent.

The third column in Table 5 shows that the addition of the Reproach emotion to the Emotion Center causes the pleasure value to drop while both the arousal and dominance values rise. The fall in pleasure causes the Emotion Center value to become negative, which is on the other side of the center point in PAD space. This in turn pulls the pleasure value of the new mood into the negative range. As both the arousal and dominance values in the emotion center rise, guiding the arousal and dominance of the new mood to be more positive than the previous mood.

The dominance value, although pulled in a positive direction, is still negative, since the emotion was not intense enough to change it; however, the overall change in the mood of the NPC moves from being moderately-dependent to slightly anxious, which is classified as a negative mood. This means that in Table 4, the NPC moves from a positive mood state where it would have selected a healthy food to a negative mood state where decided to select unhealthy food.

Table 5.

MOOD CHANGES FOR BEHAVIOUR TREE SCENARIO 2

EMOTIONS ELICITED		Pride	Reproach
Pleasure	0.197881	**0.202856**	**-0.19599**
Arousal	0.14233	**0.207651**	**0.369175**
Dominance	-0.38621	**-0.69228**	**-0.19112**
Mood	Dependent	Dependent	Anxious
Mood Intensity	Slightly	Moderate	Slightly
Emotion Center Pleasure	0.002983	**0.095391**	**-0.21392**
Emotion Center Arousal	0.039162	**0.186023**	**0.255497**
Emotion Center Dominance	-0.18351	**-0.12521**	**0.013943**

Table 6.

BEHAVIOUR TREE SCENARIO 3

Player Dialogue Steps and Options		
1	Player Choice +ve	"Let's talk about this calmly"
	Player Choice -ve	**"Could you hold the attitude?"**
	NPC Emotion +ve	Gratitude
	NPC Emotion -ve	**Resentment**
	NPC Reaction +ve	"I am willing to, are you?"
	NPC Reaction -ve	**"I'm calm, speak for yourself"**
2	Player Choice +ve	"I see your point"
	Player Choice -ve	**"You never listen"**
	NPC Emotion +ve	**Satisfaction**
	NPC Emotion -ve	Reproach
	NPC Reaction +ve	**Healthy food choice**
	NPC Reaction -ve	High-sugar meal

Table 5 also shows that the mood of the NPC persists throughout the game as each entry in the table represents an interaction with the player and the mood only changed after the player had triggered an emotion in the NPC.

Table 6 shows the condensed behaviour tree for the third scenario. In this scenario, the player is having an argument with the NPC. The player can either choose to try to de-escalate the situation or continue in the same vein. The player chose to continue the argument. This triggered the emotion of Resentment in the NPC, which lowered all PAD values for the emotion center, and made the arousal and dominance values negative. That in turn lowered the mood of the NPC. Even though the mood values were lowered, it was not enough to change the mood of the NPC. Next the player chose to be rude to the NPC triggering the Reproach emotion. Again, the emotion center was lowered, and by extension the mood, but once more it was not enough to change the overall mood of the NPC.

However, the intensity of the NPC's exuberance dropped from moderate to slight, moving the NPC closer to a negative mood, but not enough to change its mood, so it made the choice to eat healthy

TABLE 7.

MOOD CHANGES FOR BEHAVIOUR TREE SCENARIO 3

Emotions Elicited		Resentment	Reproach
Pleasure	0.625314	**0.499279**	**0.373245**
Arousal	0.190067	**0.146429**	**0.100408**
Dominance	0.285713	**0.21806**	**0.145906**
Mood	Exuberant	Exuberant	Exuberant
Mood Intensity	Moderate	Moderate	Slightly
Emotion Center Pleasure	0.304943	**0.054747**	**0.054747**
Emotion Center Arousal	0.2	**-0.04775**	**-0.04775**
Emotion Center Dominance	0.004943	**-0.0966**	**-0.0966**

A. *Discussion*

The scenarios above provide a simple yet representative use of the framework. The behaviour trees interface with the cognitive architecture, providing the developer with a familiar tool to work with. The behaviour trees are binary, allowing the player only two choices of actions to take in response to the NPC's mood. This is the most basic form that the behaviour tree can take. The notation could be expanded to offer the player a wider range of options, resulting in a wider variety of possible emotions that can be triggered from an interaction with a player. The mood of the NPC is largely determined by the various emotions the NPC has experienced over its lifetime fulfilling our requirements that the emotions felt be abled to influence NPC behavior and that the NPC mood persist over its entire lifetime. The mood is only evaluated as either positive or negative in these examples, but as previously mentioned, there are eight different moods with varying intensities. Each mood can be mapped to varying reactions that can be elicited from an NPC. It is also possible to differentiate NPC reactions constructed on how intensely the moods are felt. This provides the framework with the ability to make subtle changes to the NPC reactions giving a higher sense of immersion and realism when compared to typical scripted NPCs.

VI. CONCLUSION

We have created a tool intended to assist in the creation of games involving believable non-player characters. The EMoBeT framework provides a psychological basis for emotional modelling with the additional benefit of using behaviour trees – a tool that is familiar to game developers. This reduces the cognitive load that comes with using a new framework in game development. For our future work, we will build richer scenarios in order to provide a better testbed for the framework. We will also implement it in the game to investigate its effectiveness at modelling helpful food choice scenarios. Finally, we will test the usability and performance of the system to determine if it is a viable approach for designing psychologically-based NPCs in commercial-grade serious games.

References

[1] T. Udo, C. M. Grilo, K. D. Brownell, A. H. Weinberger, R. J. DiLeone, and S. A. McKee, 'Modeling the Effects of Positive and Negative Mood on the Ability to Resist Eating in Obese and Non-obese Individuals', *Eat. Behav.*, vol. 14, no. 1, pp. 40–46, Jan. 2013.

[2] M. Macht, 'Characteristics of eating in anger, fear, sadness and joy', *Appetite*, vol. 33, no. 1, pp. 129–139, Aug. 1999.

[3] M. Macht and G. Simons, 'Emotions and eating in everyday life', *Appetite*, vol. 35, no. 1, pp. 65–71, Aug. 2000.

[4] J. Ashurst *et al.*, 'The Association among Emotions and Food Choices in First-Year College Students Using mobile-Ecological Momentary Assessments', *BMC Public Health*, vol. 18, May 2018.

[5] J. M. Kivikangas, 'Affect channel model of evaluation in the context of digital games', in *Emotion in Games*, Springer, 2016, pp. 21–37.

[6] B. Ravenet, F. Pecune, M. Chollet, and C. Pelachaud, 'Emotion and Attitude Modeling for Non-player Characters', in *Emotion in Games*, Springer, 2016, pp. 139–154.

[7] A. Chowanda, P. Blanchfield, M. Flintham, and M. Valstar, 'Computational models of emotion, personality, and social relationships for interactions in games', in *The 2016 International Conference on Autonomous Agents & Multiagent Systems*, 2016, pp. 1343–1344.

[8] K. Karpouzis and G. N. Yannakakis, *Emotion in Games: Theory and Praxis*, vol. 4. Springer, 2016.

[9] M. Kim, ‘The artificial emotion model of game character through analysis of cognitive situation’, in *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, 2009, pp. 489–493.

[10] P. Gebhard, ‘ALMA: a layered model of affect’, in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 2005, pp. 29–36.

[11] A. Popescu, J. Broekens, and M. Van Someren, ‘GAMYGDALA: An emotion engine for games’, *IEEE Trans. Affect. Comput.*, vol. 5, no. 1, pp. 32–44, 2014.

[12] E. Hudlicka and J. Broekens, ‘Foundations for modelling emotions in game characters: Modelling emotion effects on cognition’, in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, 2009, pp. 1–6.

[13] M. Ochs, N. Sabouret, and V. Corruble, ‘Simulation of the dynamics of nonplayer characters’ emotions and social relations in games’, *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 4, pp. 281–297, 2009.

[14] A. Johansson and P. Dell’Acqua, ‘Emotional behavior trees’, in *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, 2012, pp. 355–362.

[15] I. Kotseruba and J. K. Tsotsos, ‘A review of 40 years of cognitive architecture research: Core cognitive abilities and practical applications’, *ArXiv Prepr. ArXiv161008602*, 2016.

[16] A. J. Champandard, ‘Getting started with decision making and control systems’, *AI Game Program. Wisdom*, vol. 4, pp. 257–264, 2008.

[17] D. Isla, ‘Building a Better Battle: HALO 3 AI Objectives’, *GDC Vault*. [Online]. Available: https://www.gdcvault.com/play/497/Building-a-Better-Battle-HALO. [Accessed: 27-Mar-2019].

[18] B. Weber, ‘Extending Behavior Trees’, *Extending Behavior Trees*, 05-Mar-2017.

[19] J. Busby, Z. Parrish, and J. Wilson, *Mastering Unreal technology*. Indianapolis, Ind: Sams, 2010.

[20] W. P. Subagyo, S. M. S. Nugroho, and S. Sumpeno, ‘Simulation multi behavior NPCs in fire evacuation using emotional behavior tree’, in *2016 International Seminar on Application for Technology of Information and Communication (ISemantic)*, 2016, pp. 184–190.

[21] A. Ortony, G. L. Clore, and A. Collins, *The cognitive structure of emotions*, Reprinted. Cambridge: Cambridge Univ. Press, 1999.

[22] P. S. Rosenbloom, J. Laird, and A. Newell, *The SOAR papers: Research on integrated intelligence*. Cambridge, MA: MIT Press, 1993.

[23] N. J. Nilsson, *The quest for artificial intelligence: a history of ideas and achievements*. Cambridge ; New York: Cambridge University Press, 2010.

[24] J. Laird, *The Soar cognitive architecture*. Cambridge,Mass. ; London, England: MIT Press, 2012.

[25] P. R. Smart, T. Scutt, K. Sycara, and N. R. Shadbolt, ‘Integrating ACT-R cognitive models with the Unity game engine’, in *Integrating Cognitive Architectures into Virtual Character Design*, IGI Global, 2016, pp. 35–64.

[26] D. D. Djordjevich, P. G. Xavier, M. L. Bernard, J. H. Whetzel, M. R. Glickman, and S. J. Verzi, ‘Preparing for the aftermath: Using emotional agents in game-based training for disaster response’, in *2008 IEEE Symposium On Computational Intelligence and Games*, 2008, pp. 266–275.

[27] P. T. Costa and R. R. MacCrae, *Revised NEO personality inventory (NEO PI-R) and NEO five-factor inventory (NEO-FFI): Professional manual*. Psychological Assessment Resources, Incorporated, 1992.

[28] R. R. McCrae and P. T. Costa Jr, ‘Adding Liebe und Arbeit: The full five-factor model and well-being’, *Pers. Soc. Psychol. Bull.*, vol. 17, no. 2, pp. 227–232, 1991.

[29] J. T. Wixted and S. K. Carpenter, ‘The Wickelgren power law and the Ebbinghaus savings function’, *Psychol. Sci.*, vol. 18, no. 2, pp. 133–134, 2007.

[30] P. Ekkekakis, “Affect, mood, and emotion,” Measurement in sport and exercise psychology, vol. 321, 2012..