Using Bayesian Networks to Support Believable Expression of Emotions in Games

José Afonso Baptista Vieira

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Carlos António Roque Martinho

Examination Committee

Chairperson: Prof. Luís Manuel Antunes Veiga
Supervisor: Prof. Carlos António Roque Martinho
Member of the Committee: Prof. João Miguel de Sousa de Assis Dias

DECEMBER 2021

Acknowledgments

To everyone who helped me be where I am, I thank you with all my heart.

Abstract

As emotions portrayed by game characters tend to be scripted, using a Bayesian Network to predict what can happen in a game environment was hypothesised to support more believable expression of emotions. Tests were conducted using an adventure game build from scratch applying a model to anticipate what can happen to a game character versus a reactive model. Although one found no evidence on how this model can better a reactive model, it was found evidence of the importance of control hardware in improving a character's believability, both in terms of comprehending what it is thinking or feeling and also in terms of understanding what it expects will happen next. Although not statistically significant, the variable relating to the environment the character is put - which was a dimension treated when creating the solution - approached significant values. These, on the other hand, may not be completely accurate, meaning that if features like the character's animations were improved, the findings might return different numbers from those shown.

Keywords

Believability; Emotions; Bayesian Networks; Synthetic Characters; Games.

Resumo

Como as emoções demonstradas pelas personagens de jogos tendem a ser fabricadas, foi formulada a hipótese que usando uma Rede Bayesiana para prever o que acontecerá num abiente de jogo poderá suportar uma expressão de emoções mais credível. Foram conduzidos testes num jogo de aventura feito de raíz e usando um modelo anticipatório do que poderá acontecer a uma personagem versus um modelo reativo. Embora não tenha sido encontrado qualquer indício de que este novo modelo podesse superar o modelo reativo, os resultados indicam que o hardware usado pelo jogador poderá ser um fator importante para melhorar a credibilidade de uma personagem, tanto em termos de compreender o que esta está a pensar ou sentir, como também de perceber o que esta espera acontecer no futuro. Embora nao seja estatiscticamente significante, a variavel relacionada com o ambiente envolvente da personagem - uma dimensão trabalhada na criação solução - encontra-se próxima de significância. Por outro lado, os resultados podem não ser completamente precisos: se algumas características tais como as animações da personagem fossem melhoradas, talvez os resultados encontrados tivessem sido diferentes dos expostos.

Palavras Chave

Credibilidade; Emoções; Redes Bayesianas; Personagens Sintéticas; Jogos.

Contents

List of Figures

List of Tables

Acronyms

API	Application Programming Interface
AI	Artificial Intelligence
BN	Bayesian Network
DAG	Directed Acyclic Graph
FAtiMA	Fearnot AffecTIve Mind Architecture
GAEM	Gameplay Aware Emotional Model
GIF	Graphics Interchange Format
JSON	JavaScript Object Notation
NPC	Non-Player Character
OCC	Ortony, Clore and Collins's
PC	Portable Computer

1

Introduction

Contents

1.1 Motivation

To the present day, games have been evolving to better present themselves as a work of art and enjoyment. Many of which, serve as a statement on the lives of gamers and as such, the industry continues to put forward many ways to innovate and create new and interesting ideas and mechanics to better adapt to the current standard of games. This is supported, on some cases, with an affective loop - when game characters respond to the player's interactions in a manner which seems to understand the users' emotions and creates a cycle of dynamic behaviours and responses. This is an important basis as games with affective behavior get gamers both more engaged and attached [1].

It is apparent that emotions are vital in various fields, ranging from education to entertainment. This becomes obvious as emotions can affect and/or be behind one's rationale and conduct. Not only that but emotions can be the result of the relation between anticipating an event and seeing how the event happens in the real world. Considering this, games are not left apart and part of the enticement created around them may be explained by the games' ability to elicit emotional responses. Moreover, games when played involve different interactions with not only the game environment but also between the player and the character(s) they are controlling and these can happen simultaneously. The area of study these belong is denoted as **Affective Computing**. It is the study and development of systems and devices that can recognize, interpret, process, and simulate human affects.

Emotions felt by the game characters are also key factors when considering good narratives. If a character is able to express their emotions convincingly (believably), then this allows to deepen the emotional value of the game and the relation between it, the game characters and the players of the game. This is because as one can interpret their behaviour as believable - convincing or realistic, acting naturally as if it had intents or beliefs - a better connection can be established between them and the players.

Ways of determining if characters can present a believable behaviour are already present [2]. For example, crucial aspects like *Behavior Understandability* and *Behaviour Coherence* can be measured (for instance, using questionnaires) to determine **Character Believability**. Also, ways of creating computations models which can represent the emotional state of a character can be made, for example using Bayesian Networks (BNs) as they are versatile.

One important aspect regarding believability is the concept of **situatedness**. Situatedness is a theory that asserts that the mind is ontologically and functionally connected with environmental, social, and cultural variables. As a result, psychological functions are best understood as a result of the agent's direct interaction with the environment. One essential tool for situatedness is anticipation. If the character understands the world around them, they can form expectations to what might happen to them and from that point formulate an emotion accordingly. This will be the basis of this work as this is important to form a believable character.

Therefore, as a way to offer a better gameplay experience, one will have to identify key aspects on how the players perceive their characters behaviour and situatedness and if that behaviour can be labelled as believable.

1.2 The Problem

The problem with current games relies on the fact that emotions portrayed by game characters tend to be scripted, non-organic and reactive.

Take, for example, an adventure game[1]. After venturing into the woods and covering enormous distances, the hero discovers a treasure chest which does not harm him and that contains a new and better sword. After being hit by a similar trap chest before, our hero can't help but feel frightful when he sees another that has caused him damage, but he is also relieved that nothing awful has happened this time. The hero could feel a little more confident the next time he encounters a similar situation.

In most games, this is not an example of how the character feels. It is very common to see predetermined emotions being portrayed in game characters which can break not only the emotional impact of a certain scenario but the immersion of a player in a game's world. Additionally, these do not take into consideration if the said emotions affect how the character is perceived and how they can dull the characters behaviour - they rely on characters reacting to events happening only at the moment and not taking into account what might cause a greater impact or not. Additionally, these emotions can feel unpleasant if not properly integrated. Why are these emotions fabricated and scripted and not more fluent and aware of the character's surroundings? Also, why not use a system which can portray emotions based on what the characters anticipate as the outcome of the events they are put on?

As a way of improving the coherence in the expression of emotions, not only the relationship between the player and their character but also their interaction with the game's environment should have an impact on the feel of a game and show some feedback on the game character during the gameplay. Character Believability can then assess the level of which players feel their characters act in a more veracious manner and therefore feel better connected to the latter and the game.

1.3 Hypothesis

With this is mind, creating a computational model of emotions which takes into consideration not only what happens considering other types of events (such as dealing with anticipation) but also what happens to the character (which is the traditional way of triggering emotions), would improve the **Character**

[1] Adventure games are video games in which the player assumes the role of a protagonist in an interactive story driven by exploration and/or puzzle-solving

Believability in an adventure game. Particularly, it would improve the *Behavior Understandability* and the *Behaviour Coherence*.

In order to obtain a higher level of dependence between the various behaviors of the character, and what influences them, the hypothesis was worked on two different scenarios:

- The character only reacting to situational events - ranging from health gain/loss or the opening of a treasure;
- The character reacting to events taking into account what it expects to happen - the character will have a way to predict what might happen (expect the outcome to favor them or not) and based on that react to the current scenario more precisely.

Considering the said concept as a measuring manner and as the model gets more complex in the second scenario, it was expected that the **Character Believability** levels get higher with this added complexity, notably in areas mentioned above. The second scenario would, then, reflect a higher level of character believability and thus be the better option to adopt for game design purposes.

1.4 Contributions

The following contributions were made:

- Literature review in works regarding Emotions, Character Believability.
- Development of a computational model for believable emotions according to expected scenarios happening in-game using BNs.
- Support of believable expressions in virtual characters in games.
- Development of a co-op game alongside a master thesis' colleague (João Patrício) where the hypothesis of the work was tested on.
- Implementation of the model in an actual game and use case study.
- Adequate evaluation of the emotional model with users through gameplay and questionnaires.
- Contribution in the Affective Computing area: by bringing great immersion and believability to the game experience, the user experience can be improved with Affective Computing [1]

1.5 Outline

In the next few chapters, a comprehensive view of character believability will be given - the meanings of character believability in the sense of autonomous agents followed by its dimensions. Moreover, along with some notes on psychology and the elements that come hand in hand with emotion, one can begin by describing what emotions are. This document will also include the existing state of the art and associated studies on assessment collection theories and emotional models. In addition, a closer look will be taken at BNs, some theoretical context will be given, as well as the existing work for this method and some rationale behind this probabilistic model's decision. Finally, a full questionnaire which will inspire the questionnaire that will be used to measure the level of character believability in the character of the game being implemented.

Additionally, the solution approached will be shown. Starting with an overview of the challenges one faced and how the solution came to be, followed by a more in-depth analysis on every aspect of the architecture and the thought process behind it - from concepts regarding its Input and Output to more abstract notions like Magnitude and Mood. Next, the evaluation will be displayed, including the testing scenario which one used to assess the model - with its description and tool choice - and the implementation of the model going a step by step in the making of it. Furthermore, the test methodology is delineated: how the initial tests were done, what the users were questioned about and how one dealt with issues regarding the user reports. Moreover, the results are presented, beginning with the sample analysis, followed by the comparison between this model and a reactive model and the results' discussion. The document ends with the conclusions and the future work.

2

Related work

Contents

In this chapter several key concepts will be introduced, such as: Character Believability, Emotions, Bayesian Networks and a way of assessing Character Believability by presenting a review of the literature and related work on these concepts. The aim is to establish the basis for a thorough grasp of the solution.

The purpose of this project is to understand and label correctly emotions to different scenarios enhancing the believability of a character, so one will start with the section regarding Character Believability. Its dimensions

Furthermore, the work will be followed by introducing the concept of emotions: the definition, how can one feel different emotions at the same time, how can someone expect certain emotions to happen, in conjunction with how these change according to certain situations and how they can be computed.

Following this section, an understanding of how BNs operate will be provided through some theoretical context and a display of the recent work will be done.

Finally, to assess character believability, a questionnaire will be addressed as it is vital for the final evaluation of the model and how the latter can affect gameplay experience.

2.1 Character Believability

As a way to assess how truthful the character's behaviour is using the model to be implemented, character believability will be used as a way to measure the degree experienced by every player. Throughout this next section, the definitions of character believability will be given, further accompanied by its dimensions and, lastly, a questionnaire which was be used to inspire the questionnaire presented to the players to calculate the intensity of character believability in the game implemented.

2.1.1 Introduction to Character Believability

The term *suspension of disbelief* was created by the English poet *Samuel Taylor Coleridge* around the turn of the nineteenth century [3]. The phrase refers to the mental condition in which a poem's reader might accept a supernatural, or merely romantic, persona as genuine, regardless of unusual features. Coleridge states his intention to write in a "semblance of truth" and to arouse the reader's imagination in *Lyrical Ballads* [4], clouding what would appear to be unrealistic at first sight and without context.

The phrase has developed since Coelridge's *Biographia Literaria*: the notion of character has been expanded to a fictitious setting, and the term now covers any creative form, not only poetry. Animation is one such creative medium that, in the hands of *Walt Disney Studio* artists in the 1930s, experienced significant technical and artistic advancement. In the influential *The Illusion of Life: Disney Animation* [5], Thomas and Johnston would outline the animation concepts taught by these artists. They discuss how animated figures may create the impression of being alive, having motivations, and thinking and responding appropriately in this book. Later on, Lasseter realized that the lessons of traditional animation might be applied to 3d computer animation [6].

In the 1990s, computer scientists working on autonomous agents began to look at how the aesthetic concepts of animated characters could be utilized to create realistic bots with "the illusion of life". Carnegie Mellon researchers working on the *OZ* project made a substantial contribution in this regard [7] [8] [9]. Ortony [10] offered a more emotion-focused concept for believable agents. He believed that the way agents perceive events and how this appraisal affects their emotional state should be consistent. This believability criteria among others, offer Artificial Intelligence (AI) designers guidance for creating systems that enable believable characters.

2.1.2 Dimensions of Believability

The measures' ultimate objective is to determine perceived character believability. Directly asking an audience how credible a character is, on the other hand, might be a tricky issue. Unless the audience is aware of the concept of **illusion of life** presented earlier, the answer will most likely not represent it. As a result, Gomes et. al. [2] proposed a metric that incorporates many believability factors into the overall sense of believability. Participants are questioned about more objective features of the agent in this way. The elements of credibility presented in their work were:

- **Behavior Coherence** - Coherence, according to Ortony [10], is a critical component of believability. The audience will observe the character's behaviour rather than its internal state, allowing them to be questioned about the former's coherence.

- **Change With Experience** - Loyall refers to the agent's change [8]. It's connected to Mckee's concept of story event, which is a substantial shift in a character's life value [11] in the context of interactive narrative. A classical plot arch requires these events to be present.

- **Awareness** - Agents should demonstrate how they view the world. Lester and Stone's situated liveliness [12] as well as Loyall's reactive and responsive elements [8] may be translated to this dimension.

- **Behavior Understandability** - Participants must be able to build a model of an agent's behavioural motives, according to Ortony's concept of believability [10]. Furthermore, as Bates [7] points out, an agent's behaviours must match what it is thinking and how it is feeling. This last line might be interpreted as: the agent's actions must convey what the participant believes the character is thinking about in situations where the thought process is not clearly presented. However, in order for this to happen, the audience must be able to construct a mental model of the character. As a result, the participant must be aware of the character's behaviour.

- **Personality** - Almost every definition of believability includes the concept of personality. Participants should be able to recognize the agent's behaviour features that characterize it as an individual, that make it unique, according to Loyall's definition [8].
- **Emotional Expressiveness** - The degree to which a character's feelings are expressed. Loyall [8] and Ortony [10] both discuss the idea of emotion.
- **Social** - Participants should be able to recognize character social relationships [8].
- **Visual Impact** - Is the degree to which an agent captures our attention, as stated by Lester and Stone as a believability booster [12].
- **Predictability** - Lester and Stone also emphasize the relevance of unrecognizable behaviour patterns, especially in the context of long-term interactions. Furthermore, Ortony [10] cautions against the negative impact predictability might have on believability when considering variability. Ortony did say, however, that a total absence of predictability might damage behaviour coherence and, as a result, believability. As a result, believability is harmed by both severe predictability and extreme unpredictability.

2.2 Emotions

Since the character will feel different emotions according to not only their current situation but also present and past events, it is important to tackle this intricate concept. Its definition will be covered, alongside some notes from psychology, followed by the aspects which come hand in hand with emotion, current state of the art and related work regarding appraisal selection theories and emotional models.

2.2.1 Definition of Emotion

Trying to define what emotions are can be a difficult task. It has been a topic widely studied in the psychological field but as for a definition, there is not an exact definition. There have been studies on disparate definitions which then were compiled into distinct categories as it can be seen in the work of Kleinginna and Kleinginna [13]. These categories were related to basic psychological theories which they supported such as adaptive, affective, cognitive, amongst others. However there is a consensus of the view that most theorists consider as seen in the 2010 book "A Blueprint for Affective Computing: A Sourcebook and Manual" [14]. This consensus can be seen as "a bounded episode in the life of an organism, characterized as an emergent pattern of component synchronization preparing adaptive action tendencies to relevant events as defined by their behavioural meaning and seeking control precedence over behaviour."

2.2.2 Ambivalence

As stated in [15]: researches about virtual agents with emotions have been conducted [7] [16], and have shown the indispensability of emotions to account for these to behave as if they were real. Current emotion theories [17] [18] applied to virtual agents discussed about only the presence of a single emotion state at a time. However, this is not the case for most living beings: one can be at the same time, for example, frustrated and sad. The concept of emotional agents has been introduced and developed during the last 25 years, but mixed or ambivalent emotions [15] have been an important analysis topic in other areas such as cognitive science and psychology [19] for a very long time.

In [20], it is referred that emotions also have different time spans: some of these affective states appear and fade quickly, while others are much longer lasting. Giving the example of surprise vs. empathy, one can already understand that empathy is very long lasting comparing with surprise. And this allows one to understand that there is a much more complex set of emotions present in emotional being which go beyond just “happy” or “sad”. One example shown in [20] is that: “happiness” can correspond to various combinations of pleasure, delight, amity, satisfaction, empathy, and joy (an incomplete set of inexact names for “happy” emotions moving up the Maslovian hierarchy [1] from physical to peak experiences).

As this is true, there is a problem in showing this ambivalence in synthetic characters. It may be difficult to express more than one emotion at a time and have the players clearly identify all the emotions the character is feeling. So, one will follow the standard procedure which calculates all the emotions and output the most intense one.

2.2.3 Anticipation & Confidence

After discussing what emotions are and how they can happen at the same time as others, the focus now will be on key aspects which can influence the emotion one can manifest. Anticipation is one of them. It has a major influence on the emotions an animal can feel. Using a Darwinian perspective, preparing for upcoming events allows planning of behavioral strategies and action preferences that ensure survival in an ever-changing environment [21].

An example can be drawn from [22]: “If I am walking in the woods and, suddenly, ‘something’ ahead on the path lets out a loud roar, my heart races, my muscles tense, I ‘feel’ afraid and ready to run away”. It is possible to see that the subject has a biological reaction to the unexpected. They are anticipating something bad can happen to them which then creates an emotional response: they begin to feel afraid.

Confidence is also a key factor in influencing our emotions. A simple definition can be found in the work from J.M. Barbalet [23]: “Confidence can be described, therefore, as an emotion of assured

[1] A theory in psychology proposed by Abraham Maslow in his 1943 paper “A theory of Human Motivation” in Psychological Review. Often portrayed in the shape of a pyramid, with the largest, most fundamental needs at the bottom and the need for self-actualization and transcendence at the top. In other words, the theory is that individuals’ most basic needs must be met before they become motivated to achieve higher level needs.

expectation [...] Confidence is the feeling which encourages one to go one's own way: confidence is an emotion of self-projection". So, one of confidence's trigger is related to the subject's familiarity with either the appearance or the interaction with an artefact, entity, amongst others. Making the bridge between anticipation and confidence, if the subject anticipates a certain event, a level of confidence is created, whether it be low (for example: if the subject does not know what to expect or it knows its outcome might be bad), high (for instance, if the subject knows what to expect) or something in between.

2.2.4 Emotions & Mood

Mood greatly affects the emotions one can feel as well as their intents. As it was shown in [20]: readily understanding a character's mood is useful for understanding character motivations and interactions. It defines the nature and strength of the emotions a character "feels" in different contexts. If a certain subject is in a good mood then they will be more prone to have positive views about the world that they live in and the emotions felt will be in agreement with this and thus this will be a subject who feels more positive emotions. On the contrary, if a subject is in a bad mood then they will be prone to have negative views about their environment and a predominance of negative emotions will be felt.

This is important to be portrayed by the game character as this can serve as a way for the players to understand the overall feeling of the character based on what happened in the past. So, if, for example. the majority of events that happened to a character are negative events, i.e. events that lead to a negative outcome, then, the character might be more in a sad mood.

2.2.5 Computable Emotions

As a way to create a richer experience for gamers, as stated in the motivation section, it is fundamental then to discuss the state of the art approaches to affective modelling. Scherer, Länziger and Roesch's work on affective computing [14] categorizes affective models categories into five separate general categories: appraisal theory approaches, anatomical approaches, rational approaches, communicative approaches, and lastly, dimensional theory approaches. Although all of the categories are important and are viable methods for various scenarios, for this work the main focus will be on appraisal theory approaches. This choice comes from the fact that appraisal theory presupposes that all emotions come in largely through the subject's interpretation of events. Appraisal theory refers to an examination of the how good or bad an object or state of affairs is for the well-being of one. It does not, however, account for non-reflective emotions, which do not seem to require any appraisal.

Some appraisal theories oppose this issue by claiming that appraisal is not a mechanism that is conscious, deliberative, analytical and gradual, but rather unconscious, automatic and swift [24] [25]. The intuitive appraisal is differentiated by some researchers, as it is seen as a separate implicit type of evaluation that happens without reflection. A good portion of emotional models are based on the appraisal theory in the field of computer science, helping agents to synthesize emotions and communicate them as coherently and humanly as possible.

Based on the preferences, aims, standards, beliefs and behaviors of the agent, the appraisal assessments are often subjective. All in all, the philosophy of cognitive appraisal presents the reasoning that underpins emotional expression. This can, in fact, be laid altogether with the concept of anticipation and confidence of certain events as it is best used in connecting awareness with emotion.

One interesting appraisal theory application is Fearnot AffecTIve Mind Architecture (FAtiMA) [26] which implements the Ortony, Clore and Collins's (OCC) model (the emotional classification in [27] which sets forth that emotion is structured into the categories of Fortunes-of-others, Prospect-based, Well-Being, Attribution and Attraction, or more largely grouped into consequences of events, actions of agents or aspects of objects). FAtiMA applies this model by collecting appraisals based on the emotional force in a scale from -10 to 10. Together with goal structure and perceived events, FAtiMA could model all of the emotions inside OCC, including coping mechanism to deal with specific goals and individual personality.

Another good example which uses the OCC model and the implementation of appraisal theory in video games is in the model from the work "Simulation of the dynamics of non-player character's emotions and social relations in games" [28]. This is an excellent illustration of how consequences of events, object characteristics may all contribute to increased believability. This model on the believability of the Non-Player Characters (NPCs) and seeks to improve the overall experience through their personality, their social relations and their roles. In the model, it was used extroversion alongside with neuroticism. Extroversion can be seen as how sensitive one is to positive emotion as opposed to neuroticism which tells how much one is sensitive to negative emotions. The OCC then modeled the following emotions: joy/distress, hope/fear and relief/disappointment (which supports the idea that a limited version of the OCC was used). Furthermore, there was an implemented component (emotional decay) used to regress the emotional state to that of a neutral after a set amount of time.

One last example can be seen in the the work from Pimentel [29]. The goal of the Gameplay Aware Emotional Model (GAEM) was to enhance play-through experience by developing a dynamic emotional model in a particular game context that matches the character of the player. Based on current and past events, their model evaluated the current situation in order to construct a model based on expectations, where the character could experience 6 distinct emotions (along with a neutral one).

This was done based on the evaluation of the character for any event or object and the consequence of that event or the contact with that object. That is, the emotional state is characterized by the contrast between what is occurring in the world at the moment and the probability of the same phenomenon. There is an appraisal selection aspect of the GAEM that will assign an appraisal type that may be either positive, neutral or negative. Different ways of encounters take place with the objects as the game unfolds, leading to different evaluation interpretations for the same object. The emotion is chosen in line with the following table, after the appraisal selection, along with how the event is unexpected.

	Less than expected	Within Expectation	More than Expected
Good	Distress	Happy	Hope
Bad	Relief	Sad	Fear

Table 2.1: Mapping between stimulus and emotional responses

This was based on the Emotivector [22]. In it, sensations can be modelled dynamically in order to incorporate both anticipation and expectation. In this approach, the sensorial input is split into several groups, according to what the agent expects and its valence: showing an increase in a certain positive emotion or receiving a better reward than that expected and the model returns excitement. In a similar way, showing a decrease of a positive sensation or even a worse reward than that expected and the model returns distress. Likewise, a higher punishment than that expected makes the model returns torment; if a lower punishment is given then it returns relief. It is also possible to make other categories such as expecting a punishment and receiving a reward prompts the model to return happiness and satisfaction.

Pimentel's approach offers an appraisal model based on both anticipation and expectation using Martinho's *Emotivector* and what these can do to influence emotions in virtual agents. Given how much their work relates to this one, it is, by this means, that this emotional model stands as the foundation for the model to be presented.

2.3 Bayesian Networks

When constructing a model, it is important to calculate, under such conditions, what the character would expect (or not expect) to happen, provided the current world status. Therefore, one opted to use BNs to model that aspect as these offer some practical advantages (for example, their versatility - better explained in the next sections). For that matter, the introduction to some theoretical background of BNs will be given as well as recent applications of this technique and some reasoning behind the choice of this probabilistic model.

2.3.1 Theoretical Background

A BN is a probabilistic graphical model that represents a set of variables and their conditional dependencies through a Directed Acyclic Graph (DAG) (also referred to as a Bayes Network, belief network, or decision network). It reflects the causal probabilistic relationship between a sequence of random variables, their conditional dependencies, and a compact representation of the distribution of joint probability [30].

It is comprised of two main components: a DAG and a set of conditional probability distributions. The DAG is a set of node-represented random variables. For health measurements, for example, a node could be a health domain and the node's states may be the potential responses to that domain. If a causal probabilistic dependence occurs between two random variables in the graph, a directed edge connects the two respective nodes [30], while the directed edge from a node A to a node B indicates that the random variable A causes the random variable B. Since a static causal probabilistic dependency is defined by the directed edges, cycles are not allowed in the graph. For each node in the graph, a conditional distribution of probability is specified. In other words, for every possible consequence of the preceding causal node(s), the conditional probability distribution of a node (random variable) is defined.

Consider the following example for illustration purposes. Suppose in a mid-game scenario, one tries to defeat a monster of a certain type, but instead the monster defeats the character one is controlling (this is defined as an observation/evidence). One would like to know which of the potential causes of the character's defeat is more plausible. Just two potential causes of this misfortune are suspected in this simplistic illustration: having a sword or not and the character being already wounded or not. The corresponding DAG is depicted in Figure 2.1

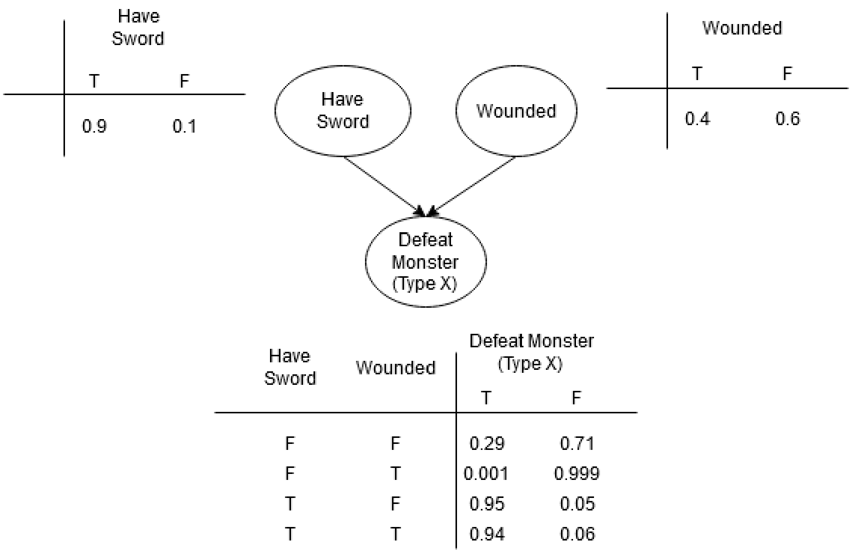


Figure 2.1: A Simple BN with conditional probability tables.

The two causes are believed to be independent in this banal illustration (there is no edge between the two causal nodes), although this assumption is not generally necessary. BNs are able to collect as many causal interactions as possible to explain the real-life situation credibly, unless there is a cycle in the graph. Because a DAG is a hierarchical system, the usage of words such as parent, child, ancestor, or descendant for certain nodes is unambiguous [31]. Both "Have Sword" and "Wounded" in Figure 2.1 are ancestors and parents of “Defeat Monster (Type X)”; “Defeat Monster (Type X)” is analogically a descendant and a child of both “Have Sword” and “Wounded”. The aim is to measure the posterior conditional probability distribution of each of the potential unknown triggers given the evidence observed, so, in a conditional probability the statement is of the following kind: given the event *Evidence*, the probability of the event *Cause* is x. The notation for this statement is:

$$P[Cause|Evidence] = x \tag{2.1}$$

It should be observed that Equation (2.1) does not mean that the probability of *Cause* is *x* whenever *Evidence* is true. This means that if *Evidence* is true, and *Cause* is unrelated to anything else known, then $P[Cause] = x$.

Nevertheless in reality, provided the cause, one is always able to get just the converse conditional probability distribution of observing the evidence:

$$P[Evidence|Cause] = x \tag{2.2}$$

The whole definition of BNs is based on the Bayes theorem, which allows one to express the conditional distribution of cause probability given the evidence observed using the converse conditional probability of observing evidence given the cause, this yields:

$$P[Cause|Evidence] = P[Evidence|Cause]\frac{P[Cause]}{P[Evidence]} \tag{2.3}$$

Where *Evidence* is an **Event/Observation**, *Cause* is a **Cause/Hypothesis**, $P[Cause|Evidence]$ is the **Posterior probability** and $P[Evidence|Cause]$ is the **Likelihood function**. $P[Cause]$ represents the **Prior probability**.

Provided the node’s parents, every node in a BN is always conditionally independent of all its non-descendants. Therefore, given their parents, the joint probability distribution of all random variables in the graph factorizes into a set of conditional probability distributions of random variables. Thus, one may construct a full probability model by defining only the distribution of conditional probability in each node [31].

With the example already given in Figure 2.1, the model identifies which kind of monster the character is facing (through its type) and other properties associated with an interaction with this monster - having a sword and already being wounded. As the parent nodes are binary (are only True or False), the resulting child node will contain every combination of its parents’ boolean value. This will prove useful, as it can easily present all of the possible outcomes and thus one may, then, work on interpreting these results and understand if given a situation is expected or not.

Upon this, an update can be made to the network. Through *Belief Propagation*, it is possible for children to propagate their beliefs to parents and vice versa. So feeding it with new information makes the next prediction more precise regarding what has been just evidenced.

There are 3 concepts when regarding the *Belief Propagation*:

- **Likelihood** - which holds information about observations of children.
- **Priors** - the probabilities of certain events which are already known in the beginning.
- **Belief** - the posterior probability after one observed certain events.

These are then used in the messaging done to propagate the belief.

A message to a parent takes all incoming messages into account, regardless of whether they have been sent by children or parents (with the exception of the parent receiving the message), and takes into account the probabilities given certain parents' values. A high-probability variable setting thus forwards incoming messages better than low probabilities. An incoming message is determined by the conditional probability of the setting of the message.

The intuition behind the message to the children is close to the message to a parent. All incoming messages are taken into consideration (all the information one can get is considered) and the aggregate is then forwarded to the next node.

With this, it is possible to undergo the process of *Parameter Learning* (which uses data to learn the distributions of a BN). Conditional distributions also include parameters that are unknown and must be estimated, e.g. via the maximum likelihood approach, from data. Provided unobserved variables, direct maximization of the likelihood (or of the posterior probability) is often complex. The expectation-maximization algorithm, which alternates computing expected values of the unobserved variables conditional on observed data, is a classical approach to this problem, with maximizing the complete likelihood (or posterior) assuming that expected values previously computed are correct. This process converges on maximum likelihood (or maximum posterior) parameter values under mild regularity conditions.

The choice of using BNs lies mostly on the fact of its versatility and low information cost, meaning that even with a low number of updates to the network, good results can be expected. As the network tries to predict what will happen taken into account what has been observed in the past, one can then emulate different "backgrounds" with a simple tweak of the values of the parameters in the networks - mirroring the observations of specific events and making the network to assume certain values.

2.3.2 Recent Work

BNs are a very versatile way to predict values, it is because of this reason they can be used for a variety of different subjects.

One example from this is the 2014 paper from Koromila et al. which tries to predict the environmental risk of a possible ship accident [32]. This was achieved by constructing and using a basic Bayesian model whose tables of conditional probabilities were described by an expert. They also showed the use of the BNs implemented in the context of two true use cases.

Another example is the work from Vlek et al. (2013) [33] which models crime scenarios in BNs. The approach described incorporates two well-known techniques for dealing with legal evidence: probabilistic reasoning in the form of BNs and narrative. By exploring the potential outcomes about what could have occurred, the holistic nature of the narrative tends to find all relevant variables in an event. In the development of BNs, they have built upon the work of authors who suggested using legal idioms. For dealing with scenarios, they have introduced a scenario idiom and a merged scenario idiom and defined a method for systematically constructing the entire BN for a case. For the holistic view of situations, they have thus enhanced the systematization begun by the aforementioned authors.

One last example, this one being more emotionally driven: *The analysis of driver's behavioral tendency under different emotional states based on a Bayesian Network* by Liu and Wang [34]. The findings of this research have shown that emotion is an important factor affecting the behavioral decision-making of the driver. In various emotional states, there are major variations in the driving behavioral pattern. For example, high speed driving was related to frustration and disgust, slow speed driving was linked to anxiety and helplessness, among others.

In addition, it is possible to extend the study findings to a vehicle safety alert system, thereby improving the accuracy of driving behaviour prediction. The findings also lead to the perception of human-vehicle activity and the reduction in risky driving actions induced by negative emotions.

Current studies, however, do not apply BNs to the emotional side of in-game characters. This is an underlying reason to rely on the versatility of this probabilistic model and to apply it to a different scenario.

2.4 Assessing Character Believability through Questionnaires

Following what was stated in the the work from Gomes et. al. [2], likert scales are frequently used to measure individual subjective impressions, thus they proposed to utilize one scale for each dimension (except for emotional expressiveness that can be tackled separately). The statement's range boundary values would be classified as "Totally Agreeing" or "Totally Disagreeing". The templates that can be used in questionnaires are as follows:

- **Awareness** - $< X >$[2] perceives the world around him/her.
- **Behavior Understandability** - It is easy to understand what $< X >$ is thinking about.

[2]This field is replaced by the name of the character currently being analyzed.

- **Personality** - $< X >$ has a personality.
- **Visual Impact** - $< X >$'s behavior draws my attention.
- **Predictability** - $< X >$'s behavior is predictable.
- **Behavior Coherence** - $< X >$'s behavior is coherent.
- **Change With Experience** - $< X >$'s behavior changes according to experience.
- **Social** - $< X >$ interacts socially with other characters

By asking participants what emotions they thought the character was mostly conveying in particular circumstances it would be enough to measure emotional expressiveness. A multiple choice test may be used to measure this, with each option corresponding to a basic emotion such as anger or fear [35]. A higher value would indicate a higher frequency of correctly detected emotions in this situation. By accurate, it is meant that it is in accordance with what the system was attempting to communicate. Previous research [36] utilized these scales, with anecdotal evidence that users comprehended the questions.

Finally, if all of the following requirements are met, the premise that a character controlling system A creates a higher sense of believability than a system B is supported:

- The predictability values of system A are not considerably closer to one of the rating extremes (totally agree or totally disagree) than those of system B.
- Except for predictability, no dimension in B is considerably greater (higher agreement) than in A.
- System A has a higher score on at least one dimension (excluding predictability), or system B has a predictability rating that is considerably closer to one of the extremes than system A.
- Character emotion identification is more accurate than chance ($\frac{100\%}{\textit{number of expressions}}$).

2.5 Discussion

The relevance of this work was analyzed in the previous sections, where one started with the idea of Character Believability and its usefulness in order to test how the model to be applied could enhance the game experience if the dynamics of relationships between a character and its environment were enhanced.

In addition, appraisal theory, which is a generally known form of endowing agents with synthetic emotions based on appraisal, was covered. The OCC is one of the most used appraisal models in Computer Science because it facilitates the production of emotions in a wide spectrum of contexts.

This model makes it possible to produce emotions based on appraisal rules with considerable versatility (given one has to program them). It was introduced how the OCC influenced GAEM at the end of this segment. Emotions are computed in GAEM based on expectations that are affected by past events and that will affect the character's appraisals. As one plans to apply this concept to increase the character believability, this model will serve as a basis to this work - keeping in mind that the modulation component which will allow for the anticipation of the world state will use BNs.

Furthermore, one got insight from BNs and how they operate to credibly illustrate the real-life situation - they can gather as many causal associations as possible, unless there is a cycle in the graph. With this in mind, it is possible to shape a solution to the problem if the non-existence of a cycle condition is met. Also, any node in a BN is always conditionally independent of all its non-descendants if the node's parents are provided. The joint probability distribution of all random variables in the graph then factorizes, given their parents, into a series of conditional probability distributions of random variables. Thus, by specifying only the distribution of conditional probability in each node, one can create a complete probability model. Additionally, one can control the game mechanics to allow for more credible models through their phenomena. These models are expected to be relatively simple, with different nodes for every type of prediction (for instance, prediction regarding enemy encounters will have a different node to the one found in, say, treasure spotting). Regarding the recent work for this section, one showed the versatility of BNs and how they can apply to different fields of research and yield considerable results.

Finally, it was presented the dimensions that the questionnaire would cover through the likert-type scale questions (and others) were presented to be answered by the players.

3

Model

Contents

The model was created with the objective of increasing not just character believability, but also the whole gameplay experience by accurately modeling and communicating the player character's emotional state. To achieve this aim, it was necessary for the model to be aware not only of what was occurring in the game world in real time, but also of what had happened earlier - there would be a requirement to go through all of the information that can be gathered from a game state and convey an emotion from it.

Taking the example from the hero given in the beginning of Section 1.2, when given in narrative form, this short example is straightforward to grasp, but when attempting to model the hero's behavior dynamically, many difficulties arise: What should he eventually feel? Should he be relieved that he wasn't hurt or terrified by the sight of another chest? How may this interaction predict how he would react to future trap encounters? What if he came upon an enemy that he could almost surely kill in close proximity? How much did he suffer as a result of the prior chest trap? Would he have died as a result of this? Or did this not have a significant influence in the long run? - what could he possibly be thinking?

These questions can be answered in a fashion that does not place excessive demands on ordinary Portable Computers (PCs) and is designed to function alongside most games with minimum intrusion into the game's fundamental structure. It operates by presenting a possible emotional appraisal every game frame depending on current and prior data, but being independent of the game design. As a result, the network between these is straightforward. The game is in charge of giving the model pertinent information in the form of **stimuli** (much like the stimuli us humans receive from the world to process information). The model is then responsible for communicating an appropriate **emotion**[1] to the character modulation module - which is the work from a master thesis' colleague (João Patrício) - which in turn sends the character modulation back to the game. This relationship can be seen in Figure 3.1.

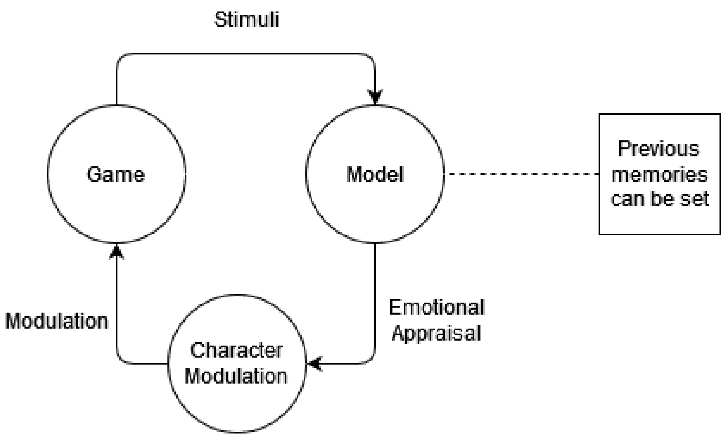


Figure 3.1: Game, Model and Character Modulation relationship.

Taking the same example from the hero and the sword in Section 1.2, as the character formulates a

[1] Which contains the **stimulus** it is reacting to, the character's overall **mood** and the **magnitude** of the emotion. These will be covered in depth in the upcoming sections.

stimulus based on seeing a chest, it remembers that in the past the chest was in fact a trap and hurt him - the model sends an emotional appraisal of feeling **fearful** to the character modulation module. The character then opens the chest (the model receives a stimuli for opening a chest), and is now presented with receiving a better sword. As this event is different from what relied in the model (or the character's memories), it sends an emotional appraisal of feeling **relieved** to the character modulation module.

Stimuli can be divided into three different categories:

- **See Stimuli** - which are stimuli perceived through proximity i.e. what the character is currently seeing. Although they do not alter the internal state of the character, these are important to the overall appraisal of the emotion that is being calculated. Examples of these are: seeing enemies or chests.

- **Chest Interaction Stimuli** - That are perceived through actions, in this case referring to opening of chests. Unlike the stimuli referenced before, these alter the internal state of the character and what it remembers for future events. They're all about how a character's interaction with the environment impacts how the avatar feels about a certain object or thing, whether it's positive, negative, or a combination of the two.

- **Combat Stimuli** - Much like *Chest Interaction Stimuli* but only regarding in-combat scenarios such as receiving damage or damaging the enemy. This implies a more thorough examination of the game because it's not just a matter of listening for changes in a single variable; it's also a matter of examining the game state to determine what constitutes an episode or a full interaction that culminates in a reward or punishment, as well as how this episode or interaction alters the avatar's perceptions of the object or thing in question. Another important aspect is the "continuity" of the action in which the affective state of the character will reflect the combat progress and the character appraisals.

Using the earlier example of the hero in the wilderness, the *See Stimulus* would be the hero seeing the new chest, but a *Chest Interaction Stimulus* would be the entire opening of the chest itself, which would reduce our hero's negative view of chests.

In order to let the player or the game designer further personalize their character before ever starting to play - making the experience much more believable - and because few games begin with game characters who are completely devoid of memories and emotions (usually implying some sort of past or personality), it is also absolutely essential to implant "artificial" memories, tweaking the model itself, in order to create a cushion or pillow for the character's emotions in order to achieve better responses. If the designer wants, they are able to setup the BN - which can be tweaked to embed these memories - according to their preference. The character may, for example, start the game with recollections of past openings of chests leading to hurting the character and thus making it feel frightful of seeing chests. It can also, for instance, recall of being successful on killing a certain type of enemy, making the character feel hopeful when seeing those types of enemies.

The output of the emotions will differ if the character is seeing something (Passive Emotion), or if the character interacted with some element (Active Emotion).

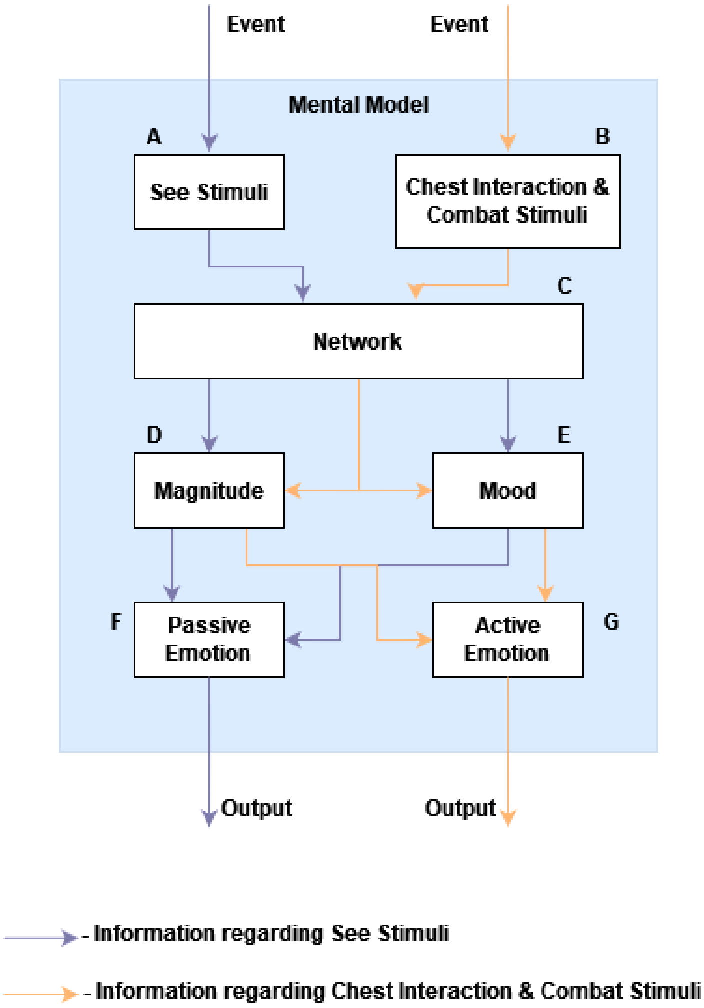


Figure 3.2: An overview of the Mental Model.

Using, once again, the example from the hero, the See stimulus of seeing a chest will be created by the model (A). By utilizing the see stimulus type of seeing the chest, the stimulus is linked with an appraisal that is kept within the model (C). Furthermore, all their different degrees of confidence are analyzed on each game frame (or at a predetermined refresh rate), and the most intense one at that time is picked to formulate an appraisal and be passed to the next module as an expression (D). The confidence value can be seen as an assessment of the character towards the outcome of that stimulus. If it feels that a certain event is almost sure to happen given a certain stimulus, then the confidence value is high. It ultimately selects the stimulus which it is more confident about its outcome (D). Then the overall mood of the character will be determined (if everything it knows about the current and past situations makes the character feel afraid or relieved), as well as the intensity with which it feels it (E). Finally, using both an appraisal and a confidence factor, one may create an acceptable emotional reaction (F).

Combat stimuli and Chest Interaction Stimuli (B) both update a particular object/entity's perception (C). Not only that but they also compare the character's perception state before and after the stimulus and produce an appraisal based on this comparison (D). If the hero was afraid of the chest because it might hurt them, knowing the chest they just opened was not a trap (did not hurt him) and it also gave them a new sword, then it would be logical to be more hopeful towards chests so now its mood becomes more positive (E). Maybe the next ones can lead the hero to more treasure/loot. It then outputs an emotion based on what happened (G).

3.1 Input & Output

As stated, the game sends events to the model to which then the model creates either See Stimuli, Combat Stimuli or Chest Interaction Stimuli. These stimuli are only interpretations made in the character's head of things that happen around it, much like the stimuli us humans receive are human-like interpretations of the world around us.

These stimuli contain information regarding what just happened: stimulus always contain a source (the cause/the origin of the stimulus) and if it was already processed by the network (more detailed in upcoming sections).

See Stimuli can be subdivided into seeing enemies or chests - the sources for these are the enemy/chest's position. As enemies can have different types it must be ensured that the character can have a different perception of the different types of enemies. So, the enemy type is also present in the stimulus for seeing enemies.

Regarding other types of stimulus, Chest Interaction Stimulus are simple and only contain the basic information for a stimulus, previously mentioned (as they do not require anything else). Combat Stimulus are subdivided into 4 different stimulus: getting hit by the enemy, attacking the enemy, getting killed by the enemy, defeating the enemy. From these, only the stimulus for getting hit by the enemy contains additional information: the damage received. A summarized explanation of this can be found in the table below.

		Stimulus Content	
		Source	**Additional Information**
See Stimuli	**Seeing Enemies**	Enemy's Position	Enemy Type
	Seeing Chests	Chest's Position	-
Chest Interaction Stimuli		Chest's Position	-
Combat Stimuli	**Getting Hit By Enemy**	Enemy's Position	Enemy Type; Damage Received
	Attacking an Enemy	Enemy's Position	Enemy Type
	Getting Killed By Enemy	Enemy's Position	Enemy Type
	Defeating the Enemy	Enemy's Position	Enemy Type

Table 3.1: Different types of stimuli and their content.

When it comes to output, an example of an emotional response would contain the following information *Emotion =* {*Stimulus = seeing a chest*[2], *mood = 0.9*[3], *magnitude = 0.6*[4]}.

3.2 Processing

3.2.1 See Stimuli

See Stimuli are kept in the character's internal state. At every frame, the character searches for nearby entities to which it knows that when interacting with them it will change its internal state (chests that are still not opened or enemies that are alive). The character's *range of sight* can be chosen at will by the developer. Therefore, the character can only sense entities that are within its range of sight and those that are not blocked by other objects in the environment. This means that if there is a chest behind a big rock, the character will not be able to see that there is a chest there.

3.2.2 Chest Interaction & Combat Stimuli

Among with See Stimuli, other types of Stimuli are also kept in the character's internal state. These are all of the Stimuli the character receives that are not through seeing, such as receiving damage or attacking an enemy. These stimuli are also kept but this time with the conduct of being processed as a whole - as many attacks, for example, come from the same enemy and need to be addressed to calculate the estimated damage the character expected versus what was the damage it actually received. This is better understood in Section 3.2.7.B. These of course can be increased with added functionality, for example, more combat stimuli that refer to specific situations, more interaction stimuli with different objects, amongst others.

[2]Which contains the source for the chest
[3]For now lets consider this value as the character feeling very good
[4]It is a fairly strong emotion

3.2.3 Network

In order to have some *Decision Making* process, the character needs to store the information about everything it knows inside its memory. As this model will play with uncertainty (often referred as a *stochastic environment*), the character need to form anticipation to what they feel is the outcome of a certain action. This was done using a BN.

Each type of enemy, treasure and situation in the game can have their impact on the character so for every single element in the game there is a certain emotion that can be expressed. As stated in Section 2.3.1, BNs are able to collect as many causal interactions as possible to explain the real-life situation credibly, unless there is a cycle in the graph. Not only that, but because parent nodes are binary (either True or False), the resultant child node will include every possible boolean value of its parents. This will be beneficial since it will quickly display all of the possible outcomes, allowing one to concentrate on analyzing the data and determining whether or not a certain circumstance is expected. Also it is important to understand that *Belief Propagation* allows children to communicate their beliefs to their parents and vice versa. As a result, giving it fresh data makes the next estimate more exact in terms of what has just been shown.

The decision to use BNs is based mostly not only on their versatility but their low information cost, i.e. that even with a small number of network updates, good results may be expected. As the network tries to anticipate what will happen based on what has been seen in the past, different "backgrounds" may be emulated by changing the values of the network's parameters - reflecting particular event observations and causing the network to assume certain values.

There are a considerable number of libraries which offer an implementation for BNs, that are made to handle lots of data at the same time. One chose to use *Bayes Server*. Although its name can lead to misinterpretations, *Bayes Server* offers the possibility for a local setup. It is a technology which is used in a variety of different fields ranging from Aerospace to Health, Finance and other advanced sectors. It supports C# integration, which is the most used language in the game-engine that one used to implement the game (Unity[5]). Among many things, it offers *Online Learning* which enables the user or Application Programming Interface (API)[6] developer to update the distributions in a BN each record at a time. As a result of the API used not offering a graphical visualization of the network, one implemented one from scratch using Unity's Graph View, which is an experimental API. Although visualizing the structure of a BN is optional, it is a great way to understand a network.

[5]Unity is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005. https://unity.com/

[6]An API simplifies software development and innovation by enabling applications to exchange data and functionality easily and securely

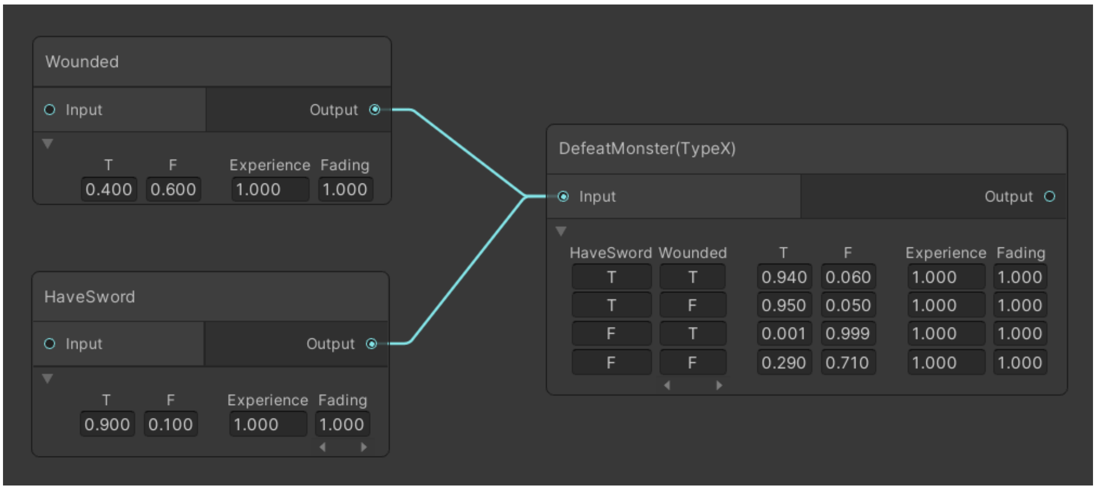


Figure 3.3: The example from Figure 2.1 using Unity's Graph View.

The meanings for every aspect in Figure 3.3 will be explained in detail later in this section. Thanks to said implementation, it is possible to open the network as an attribute of the character (in Unity) and change the values according to what the designer wants. The network and its values are saved as a file and, on the start of the game, a copy is made to ensure the original is not overridden. Additionally, as the two APIs do not store data in the same way, it is crucial to do a translation from the data that is stored in a file (Unity's Graph View), to the data that can be interpreted by *Bayes Server* in the beginning of the game and the other way around every time the network is updated (in order to see a graphical depiction of the changed values).

The conceptual network that supports this work can be as complex and interconnected as the designer sees fit. It can start to be as in Figure 3.3 and then continue to grow in similar ways. For example: what can the character expect from chests? Considering the items the character has it can expect what is about to come from a chest. Another example is what can the character expect from a horde of enemies? Considering different types of enemies and given how many of them there are, it can expect to be very wounded at the end of the combat or not. So, the work is as scalable as the designer sees fit.

One opted for a first implementation of the network using only simpler concepts (using no edge dependencies, for example) in order to test the impact that the network can have at this point - allowing to build a baseline for a future evaluation of a more complex implementation of the network. The implementation can be seen in Figure 3.4.

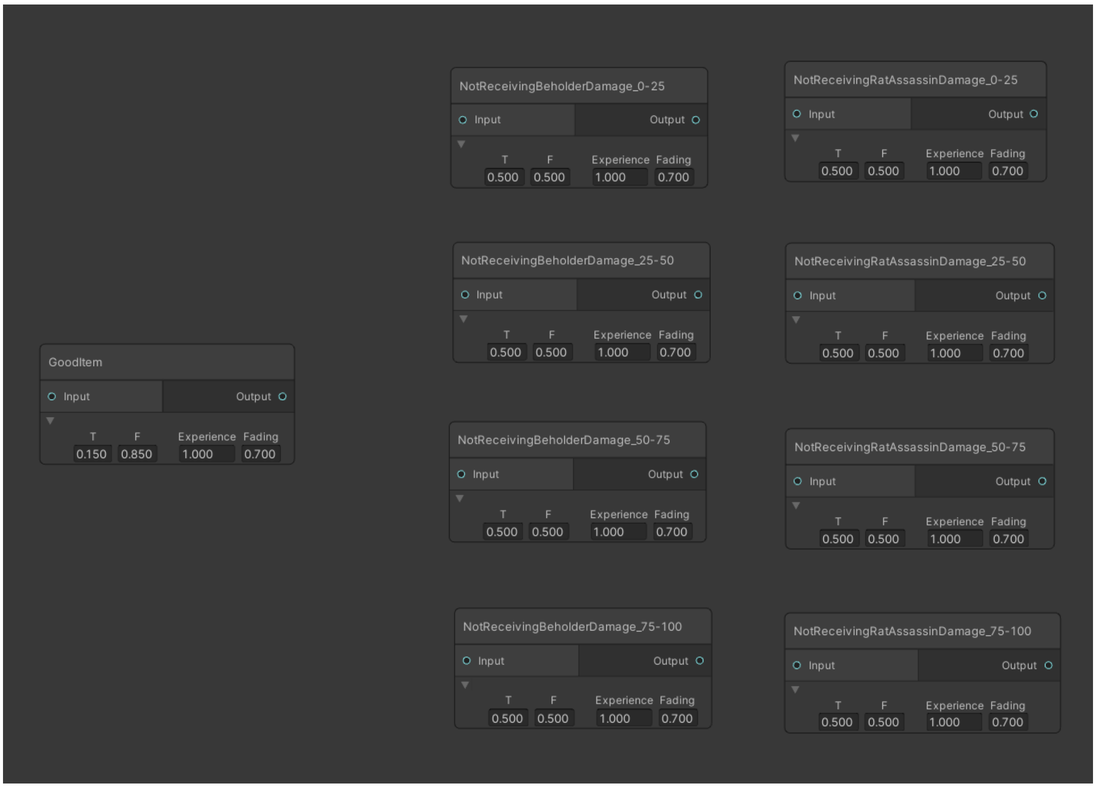


Figure 3.4: Implementation of the Network.

The image shows 9 different nodes. One opted for explicitly naming the nodes so that their **True** value represents a positive outcome and its **False** value indicates a negative outcome. The leftmost one is called “Good Item” and it displays how likely the next opening of a treasure contains a good item, i.e. a better sword, a better shield, among others. It was also decided that for chests, the character would have a negative background towards opening them (it can be seen that the most likely outcome when opening a chest is not receiving a good item, i.e., the probability of receiving a good item is only 15%).

Then, arranged in columns, it is possible to see 2 sets of 4 nodes, showing the most likely damage percentage not to be received by 2 different types of monster (Beholder and Rat Assassin). The damage percentage is split into 4 equally valued parts (From 0% to 25%, 25% to 50%, 50% to 75% and 75% to 100%). Throughout the game, these values will be changed according to what happened to the character, yet this time each set of 4 nodes must ensure that only one node is false at a time. This happens because the character can not expect to be receiving between 0% to 25% and 75% to 100% of damage at the same time from the same enemy. This dependency was not ensured by the causal dependencies between nodes using edges, but through code.

It can also be seen in both Figure 3.3 and Figure 3.4 Experience and Fading values. This is because Online Learning of discrete nodes requires the use of Experience tables and optionally Fading tables.

Before online learning can be conducted, any distribution that one desires to update must have an experience table associated with it. Each parent combination for a node must have an experience value. The degree of prior knowledge about the associated probabilities is reflected in the value. A number of 1000, for example, indicates that there is some certainty in the related probability values, but a value of 1 indicates that the probabilities are uncertain. A 0 experience value implies that no adaptation for that parent configuration should be done. A Dirichlet Distribution [7] is created when the experience values are multiplied by the probability values for a node, and it is utilized in the online learning process. This is the same as using a standard fully Bayesian approach and is used internally by the API.One chose to have an experience value of 1 for every node so that when updating the network, all of the values would be updated with values that would be more according to the present experience of the character. Nonetheless, if the designer wants, a higher experience value can be given to any node, in order to ensure that the character will not be so much drawn to change its perception of things by present experiences.

A fading table, in addition to an experience table, can be added to allow prior values to be given less weight. i.e. the significance of information that happened before diminishes over time. For each parent combination, a fading value between (0, 1] is needed. A number of 1 indicates that no fading is used, whereas lesser values (such as 0.99 or 0.9) indicate that fading is used. Upon some tests, it was decided that a value of 0.7 would be ideal to every node. This value represents that the past accounts for 70% of what the character feels about a certain outcome (and 30% represents the weight of what just happened). This means that with few (1-3) updates, the view of the character towards a certain outcome may change to its polar opposite. This also allows for the game designer to establish different values for different scenarios, making the character taking into account more or less of what it already knows about that scenario.

3.2.4 Confidence

Now, it is important to understand the concept of **confidence**. As it can be seen in Figure 3.3 there are 3 different nodes. If one would to predict the most probable outcome - in this case, if the character can defeat the monster - how could this be done? One simple but effective way is to analyse the **True** and **False** values of its parents combinations and return a value. In this example, the most probable outcome for *HaveSword* is **True** (90% probability) and for being *Wounded* is **False** (60% probability). So, if one wants to know the value for *DeafeatMonster(TypeX)*, it will be given by looking at the table (from the network) for this node in the **True False** row, which yields a result of **True** (95% probability).

However, as one is dealing with concepts of probability, these values are not certain, i.e. if the character is wounded then all of these predictions will not hold any particularly useful value. Instead, it

[7] Dirichlet distributions are commonly used as prior distributions in Bayesian statistics. The Dirichlet distribution is the conjugate prior of the categorical distribution and multinomial distribution.

is possible to attribute a **confidence** value at each time the character analyses the possible outcome for a node. By doing this, it may understand what the most probable outcome given its confidence about it.

In the example from the figure, the confidence value for the *HaveSword* node should be higher than the one from *Wounded* node. This is because the probability for the former to happen is greater than the latter to not happen ($0.9 > 0.6$). So it is possible to stipulate that the closer a binary value gets to 1 (100% probability of happening), the more confident the character is about the outcome of that node (as its experience in the past show that every time that node is calculated/updated, it showed a stronger response to that binary value). On the other hand, the closer a binary value gets to 0.5, the more uncertain the character is about what is about to happen (as there is a 50-50 chance of either being **True** or being **False**). This way, the range of values of confidence that the character can have fluctuates from 0 to 0.5 (being 0 the less confident - when there is a 50/50 chance - and 0.5 the most confident - when the outcome of a node is 100% probable of happening). This can be calculated easily by subtracting 0.5 from the probable value of the node (the one above 0.5). Therefore, from the example shown, it is possible to understand that the most probable outcome for *HaveSword* is **True** (with 90% probability and 0.4 of confidence) and for being *Wounded* is **False** (with 60% probability and 0.1 of confidence).

In the case of the last node in the network (*DeafeatMonster(TypeX)*), it is a child node. As this is the case, one is now trying to understand the confidence value of the path that leads to this node, i.e. the confidence values of the parents must be taken into consideration additionally. This is the same as saying: in order for the outcome of this node to happpen, one (or both, or neither) of the parents had to happen. So, the confidence of this outcome can be calculated through knowing the confidence of its predecessors, adding the values to the confidence of the current node (which is known to be 0.45 as for the example it shows 95% probability of happening) and dividing by the total number of nodes (which is 3). So the calculation is as follows:

$$\begin{aligned} C_{DeafeatMonster(TypeX)=True} &= \frac{C_{Wounded} + C_{HaveSword} + C_{DeafeatMonster(TypeX)}}{3} \\ &= \frac{0.1 + 0.4 + 0.45}{3} \\ &= 0.31(6) \end{aligned} \tag{3.1}$$

Where *C* means *Confidence*.

The character, then, uses this notion to chose which event to react to. It focuses on the events that it is more confident about.

3.2.5 Mood

Now that the concept of confidence is understood, it is possible to begin to explain how the mood of the character is calculated.

3.2.5.A Mood Calculation

As seen in Section 2.2.4, mood is useful for understanding a character's motivations - upon selecting the output, considering a window of the past events of the character, if the majority of these are positive ones, then the character's mood can lean towards **happiness**, otherwise it will lean towards **sadness** (using for example a continuous scale from -1 to 1).

As it can be seen as a spectrum (from "sad" to "happy"), it is important to refer to the character's memories when trying to define (or establish) what can it be feeling at a certain time. As such, an overall assessment of what the character knows is done, i.e. taking a look at everything the character knows (from survival chances to loot of chests) it is possible to perform a weighted average of the overall confidence of the character. This weighted average is calculated differently depending on the situation and only takes into consideration what the character knows about its environment meaning that, for example, if the character has not seen a certain type of enemy, then it will not take into consideration the confidence for defeating that enemy simply because in the perspective of the character that enemy does not exist.

For every appraisal, the mood is initialized at 0 so as to analyse at to always have a "clean slate" to know what the character is feeling at a certain time. So if the outcome of a certain event is more likely to be bad, then the confidence for that event is subtracted from the mood and vice versa. In the end, the mood is then divided by the total sum of the confidence values, leaving it with a value between -1 and 1.

This allows for an appraisal of the character's emotion at a given time and it can be mirrored to what happens in real life with human emotions. It can happen that one has a terrible memory associated with a certain event and a truly positive about another one and it would only be natural to take these two into consideration when evaluating how one is feeling. It could be argued, however, that having two strong opposite stimuli can make the character take both into consideration and react with a neutral appraisal when typically one could consider making the character concentrate more on a specific event (eventually with some attenuation) which was not the case.

When character analyses its health condition, it has to understand if the current health is high enough to survive in the wilderness. It can happen that an health percentage of 50% can be enough for the character to feel confident in some cases (for example, in the beginning of the game where the enemies are easier to defeat) but in others it can feel less confident (for instance, after defeating some tougher enemies). Because of this the appraisal is done differently: the character takes its current health percentage and then subtracts the possible damage percentage it knows an enemy is likely to give and assesses if this yields a value above 0%. The confidence value is added to the mood if the character's health is greater than 0% and subtracted if this is not the case.

3.2.5.B In-Combat Mood

Nonetheless, the character has the ability to understand that the present is ever-changing and so at a certain point, the view one once had can be completely different from what it is now. It is for this reason that the character is also able to predict the near future when engaging in combat. The character decides if it is in combat or not if it can see the enemy or not.

It takes the enemy's current health percentage and uses it as a progress of the battle - when the enemy's health percentage is at 100% then the battle as only begun, but when the enemy's health percentage reaches closer to 0% then the battle is almost over. Doing this, and taking into account how much damage it has received from that enemy, the character then calculates (using the rule of three) how much health will it have when the battle is over. The character knows how much damage it has received and given to an enemy because it stores these stimuli. This will be explained in Section 3.2.7.B.

As there are different types of enemies and each enemy type has a **normal** and a **stronger** version (which only appears after seeing a normal version of the enemy), when confronted with the stronger version of the enemy (which it had seen before) the character can have a different appraisal about this enemy type, if the designer wants. This is done inducing a bias towards these stronger versions which is defined by the designer. This bias is a value from 0 to 1 which represents how much additional damage the character will think this newer version deals. It can, for example, be set to 0, in order not to introduce any bias.

So, when calculating the amount of health it thinks it will have at the end of the combat, it subtracts this bias to it, leaving it with the final result. This is as if the character thought to itself "Oh, I've seen this enemy before! Only this time it appears to be stronger... I should be more careful, it should deal more damage than the others.".

If the health is more than 0%, i.e. if the character is alive (meaning it can defeat the enemy), then the confidence value is added to the mood. Otherwise, the confidence value is subtracted.

3.2.6 Magnitude

Systems like these can consider extrinsic factors - such as a feeling of surprise when something unexpected happens - but also intrinsic factors - like the ability to do something with great importance to the agent's objectives. Both can make the character feel strong emotions. In the case of this work, only extrinsic factors were taken into account.

Magnitude, then, can be seen as both the intensity of a certain emotion or its unexpectedness. Giving an example, if the hero is used to defeat a monster of a certain type, although the mood can become quite high (reaching closer to 1), it begins to lack novelty as this happens frequently. As it was said, it becomes “used to it”. So if the hero comes across that type of enemy and effectively kills it, then this event will not yield a significantly new emotion, it becomes less intense, with less magnitude. Contrarily, if the hero is used to defeat that certain type of monster and then, surprisingly, it perishes, then this must have a greater impact on the overall emotion the character is feeling. The character might ask itself “How can this have happened? I have never lost a fight with this monster!” and become more uncertain, more **afraid** of dealing with this type of enemy.

Therefore, to calculate the intensity of the emotion it is important to analyze how were the character’s memories towards that event. If it was confident about something happening and the opposite happened then this perfectly illustrates an example of an unexpected outcome. For this reason, the absolute value of subtracting the confidence of the before and after state of a certain event, indicates the shift in the confidence the character has towards a certain outcome. As this work’s output is sent to the module João Patrício developed, it was decided that Magnitude values should be between 0 and 1. The end result of this equation is a value between 0 and 0.5 so it is then multiplied by 2.

In the case of the enemy killing the character or the character killing the enemy, the assessment is made differently. As there are 4 different nodes that tell which damage the character is more likely to receive, the confidence value may not be the only source of information. The damage the character thinks it will receive at a certain point can be different from the damage it effectively receives after a combat has ended. It is because of this that it is important to analyze the before and after states as values of damage. So, the equation changes to the absolute value of subtracting the damage of the before and after state. As the result has to be a value between 0 and 1, and as there are 4 different nodes to represent the damage expected, one opted to use these as quartiles. If the character received 75% of damage and expected to receive 25% then the difference is done as such:

$$\begin{aligned} M_{DamageReceived(TypeX)} &= |DamageReceived(TypeX) - DamageExpected(TypeX)| \\ &= |0.75 - 0.25| \\ &= 0.5 \end{aligned} \tag{3.2}$$

Where *M* is the *Magnitude* of the emotion.

In the scenario of only seeing an enemy or a chest, a magnitude value must also be calculated. For this particular event, the confidence of the outcome is only multiplied by 2, as there is only made an assessment of the past of the character i.e. of its memories and what it expects to happen.

3.2.7 Passive & Active Emotions

As the character may be seeing many entities at the same time, it can develop a different appraisal for each and every individual entity, yet it only outputs the appraisal to which it is more confident about the outcome. This is represented as a **Passive Emotion**. Passive Emotions are outputted at every frame and are only assessments based on what the character thinks it might happen given what happened in the past - they do not alter the internal state of the character. On the other hand, an **Active Emotion** is an emotion that alters the character's internal state. Changes in the character internal state are reflected in the networks probabilities. It is only outputted when something in the environment around the character changes (when Chest Interaction and Combat Stimuli are created).

Picking up the example from before with the hero in the wilderness, it has now come across several chests to which all of them contained new items and increased the survival chances of the character. Now, the hero feels very confident that nothing in their path can hurt them or make them feel less confident. For their surprise, they see another chest and begin to output a Passive Emotion regarding the seeing of the chest. This emotion contains the following information *Passive Emotion =* {*Stimulus = seeing a chest, mood = 0.9, magnitude = 0.6*}. Upon opening the chest, the character is greeted with a trap and loses some health points. The following Active Emotion can be outputted *Active Emotion =* {*Stimulus = opening a chest, mood = -0.6, magnitude = 0.5*}.

Upon outputting an Active Emotion related to opening of chests and killing an enemy, the character can no longer output Passive Emotion towards them. The character will not react anymore to seeing chests that have been opened (as it already reacted to opening them) nor to enemies that have perished (as these disappear and do not hold any more value).

3.2.7.A Passive Emotion

Moving on into Passive Emotional output, they are predictions based on what the character believes will happen based on what has happened in the past - they do not change the character's internal state.

For this to happen, the character knows, at a given time, what are the most probable outcomes. This is because its internal state allows the character to search for nodes in the BN which will be probable events, i.e. nodes that are leaves. Having known which of them are leaves, it is then possible to search for their parents, their parent's parents and so on, reaching the root nodes. Through these root nodes, the confidence calculation can be done and the leaf nodes then will be checked to see which of them hold any significance value, given what the character is experiencing. To do this, the character iterates through its See Stimuli. All of these See Stimuli now have an equivalent leaf node with a confidence value attached, meaning the character can understand, by seeing a chest or an enemy in their range of sight, what is the most probable outcome related to seeing them. As mentioned earlier, seeing chests will make the character predict if they contain a good item or not while seeing enemies will make the character predict if it is able to kill them or not. If one were to change the structure of the BN, the character could, then, predict different results. These See Stimuli that now have an equivalent leaf node will be iterated and chosen which one is the character feeling more confident about its outcome. The comparison is done using the following metrics:

- If the character is currently comparing the See Stimulus which (until now) it feels more confident about with a See Stimulus of seeing an enemy, then, for the latter to be chosen as the one the character feels more confident about, its confidence value must be higher than the former plus a default confidence margin.

- The same is done when comparing with See Stimulus of seeing a chest, only now, if both being compared are referring to chests, then the confidence is the same for both of them, so the character chooses the closest to it.

The default confidence margin - which is also a parameter that can be changed by the developer - is a margin given so that the comparisons can have range to be equal. As one is dealing with numbers that can have multiple decimal places, it is important to know that these, when compared, can be misinterpreted. If one were to compare two confidence values that are close by a factor of, for example, 0.0001, then it makes more sense to deal with these values as if they were the same and not one greater than the other. It is because of this that within that default confidence margin, values are treated as if they were the same.

The final value chosen for this margin was 0.1, so, for example, if a confidence value for an outcome was 0.3, for another outcome to be taken into account as different it would have to have a confidence value of 0.4 or more. This is because after some testing, and considering that the network was settled to change an outcome with few updates (1-3), values which on average were updated the same amount of times, would have values which were close to each other. These were most of the time close by a factor of 0.1, hence the value chosen.

As stimulus which invoked the highest confidence of happening a certain outcome is now selected, the Passive Emotion can now be outputted containing in it the magnitude of the emotion, the overall mood of the character and the said stimulus. The methods for calculating these were discussed in previous sections.

As the magnitude is a value from 0 to 1 and has Mood ranges from -1 to 1, there are 4 different emotional appraisal extremes that were considered alongside João Patrício:

- **Magnitude = 0, Mood = 1**: is equivalent to the character feeling *Relieved*.

- **Magnitude = 0, Mood = -1**: means the character is *Distressed*.

- **Magnitude = 1, Mood = 1**: implies the character is *Hopeful*.

- **Magnitude = 1, Mood = -1**: represents the character is *Afraid*.

As Magnitude values represents the unexpectedness of an event, the emotions in which the magnitude is at 0 represent emotions which happened but are not a surprise for the character. **Hope** and **Fear**, however, show a reaction to a unexpected outcome: if the character is confident about killing an enemy, upon engaging in combat, if the character starts getting closer to die, it will feel afraid of the enemy. The opposite is also true: if the character feels its going to die and then gets closer to kill the enemy then it gets hopeful to kill the enemy. On the other hand, if the character knows it will die from a combat with a certain enemy and then that exact outcome happens, then the character will feels distressed about that fact. The same can be said the other way around.

3.2.7.B Active Emotion

An Active Emotion is one that causes the character's internal state to change. It is only outputted when anything in the character's environment changes (when Chest Interaction and Combat Stimuli are created). These all follow the same steps:

- **Formulation** of the stimuli
- **Processing** of the stimuli
- **Update** the network
- **Return** the emotion

Giving an example of the hero in the wilderness, it now enters in combat mode with a certain type of enemy which the character feels is going to result in their own death (it feels afraid of that enemy). Upon some fighting, the character is able to kill the enemy. The following explanation will contain both the in-combat procedure (for when the character is damaged/attacks the enemy) and the killing of the enemy.

Regarding the **formulation** of the stimuli, through Unity Events[8] it is possible to understand, for example, when the character is injured from the enemy. After this is detected, the character creates a new stimulus regarding what just happened (stimulus for getting hit by the enemy). This stimulus is stored for later use, in order to update correctly with what percentage of damage the enemy dealt to the character. This process is done until either the character kills the enemy or the other way around.

After the character killed the enemy, in order to **process** the stimuli stored (the stimuli for getting hit by the enemy), the character goes through all of the stimuli it received during that battle and sums all the damage received by the enemy for later comparison and outputting an emotion. If the character were to be killed by an enemy, the character would interpret this event as the same as losing all its health in its

[8]Unity Events are a way of allowing user driven callback to be persisted from edit time to run time without the need for additional programming and script configuration.

entirety i.e. this enemy dealt damage that is summed to 100% of the health of the character. This is done to avoid cases when the character was already low on health and then was killed by the enemy. The damage it received upon being killed could in fact be less than the amount the character was expecting to receive and return a more positive emotion instead of being frightened by just had died.

After the processing is done, all of the stimuli that were stored and, consequently, used for processing, are now discarded as they no longer prove useful.

Then comes the **update** of the network. As these outcomes are well known in the architecture of the network (leaf nodes), the process of identifying which nodes to update or not can be done the same way as it was explained in Section 3.2.7.A. The state of the network before updating (the amount of damage the character expected to receive from the fight with that type of enemy) is stored for later comparison with the updated network (the actual damage it received). Following the evidence collected by the character (killing an enemy and receiving a certain amount of damage) the network updates its distribution and experience tables taking into account the fading values that were associated which were talked about in Section 3.2.3. It is at this time that the data from this network is copied over to Unity's Graph View in order to see a graphical depiction of the changed values.

Finally, an emotion can be **returned** (containing its magnitude, the character's overall mood and its stimulus). The ways in which these are calculated were presented in earlier subsections. In this case, the emotion returned would have a high value for magnitude (as this was an unexpected event), the mood (which would also contain information about other events but would take this event into account as being something good) and the stimulus for killing the enemy.

Of course, further functionality may be added, following, for instance, the added stimuli referenced in Section 3.2.2.

3.3 Summary

This chapter opened with a broad overview of the mental model, emphasizing the key differences between the game and the model itself, as well as what each should communicate to the other. The game transmits all of the data it deems essential from an affective viewpoint to the model, which then sends all of the data it needs to appropriately portray that affective state to the character modulation module, which then returns the character modulation to the game.

Afterwards, we detailed the main concepts of the mental model and how they interacted with each other, leading to an analysis of the flow of data, as it comes from the video game. It starts with the input of events from the game which in turn are used by the character to formulate different types of stimuli - which can be used to form different types of emotion. These stimuli are then taken into account with what the character predicts will happen. This is possible through calculating the confidence values. These different types of stimuli can alter the internal state or they can also be just lookups to the network. When updating the network, the character analyses the new result with the former result. This results in the formulation of the Magnitude of the emotion. The character then assesses all of the network and through the confidence values assesses if the majority of the outcomes are positive or negative, resulting in the character's overall Mood. These are then key aspects to include in the output of the emotion - whether it is a passive (in the case of seeing a chest or an enemy) or active emotion (when the state of the world changed) - alongside the stimulus which originated the emotional response.

4

Evaluation

Contents

The model may be used in any avatar-based video game, but when it comes to testing it and deciding which video games to utilize it in, adventure games stand out as a logical choice. These are games that have a strong focus on the adventure elements, as the name indicates. This means that its major feature is the opportunity for players to control characters that face various difficulties. This makes them great candidates for testing the model, not just because of their avatar orientation, but also because believability is a highly valued feature in games that offer a storyline such as these.

The term "adventure game" comes from the 1970s text computer game[1] Colossal Cave Adventure[2], often known as Adventure, which pioneered a form of gameplay that was replicated by numerous creators and eventually established its own genre. Unlike the literary genre, which is defined by the subject it addresses: the action of adventure, the video game genre is characterized by its gameplay.

Most adventure video games attempt to be unique while putting the user in a familiar gameplay environment to make the game easier to play. Following this line of thinking, it is critical to follow these inherited gameplay features in order to test and apply the model in a situation that is comparable to games with which the user is already familiar.

4.1 Testing Scenario

To test the model, it is required a game to use it on and, of course, a user sample to test it on. Testing the model on a full, altered game might result in tests that need testers to be familiar with the game ahead of time, reducing the sample size available for testing. This means that testing would be limited to the game picked, which might be less than optimal. As a result, it was decided to make a little video game episode with the model running alongside it - this is the most efficient way because it eliminates unnecessary code while still offering a suitable testing situation.

As it was already stated, the chosen video game episode to assess the model was an adventure game with few controls and game mechanics.

4.1.1 Description

The controls are provided to the player at the start of the game (these controls are also shown in the pause menu). The player then takes control of the main character, who is standing in the woods next to a table. Despite the fact that the character is not alone, all of the character's companions at the table fall to the ground. The game is completely linear, and the player is free to explore whatever they want with no predetermined objectives. He is introduced to the following ideas while he plays the game.

[1] Text Computer Games, sometimes known as "interactive fiction," are software simulations of worlds in which players control characters and manipulate the environment using text instructions.

[2] Will Crowther created Colossal Cave Adventure on the PDP-10 mainframe between 1975 and 1977. With the aid of Don Woods, the game was enlarged in 1977, and other programmers produced game variations and ports to various platforms in the years that followed.

The **Main Character** - At this stage with an affective state to fear what comes from chests. It has 100% of health and if it drops to zero the character dies and is revived at the last checkpoint (its health is put back at 100%). It can move in any direction and jump in the air, attack enemies using a sword and defend using a shield.

(a) Walking in every direction.

(b) Jumping in the air.

(c) Attacking using a sword.

(d) Defending using the shield

Figure 4.1: The different actions the player can make the character execute.

Sword - The first sword the character is given, with it the character is able to attack enemies and progress through the game.

Better Sword - This sword is obtained through opening a chest and it deals more damage to the enemies than the original sword.

Shield - The shield given to the character in the game, it allows for defending against foes, in order to increase the character's chances of staying alive. **Checkpoints** - Which allow the character to return to a safe area after it dies. To activate them, the character must get near them.

Chests - Which may contain a better sword than the one given at the start of the game or may be a trap chest and induce damage to the character.

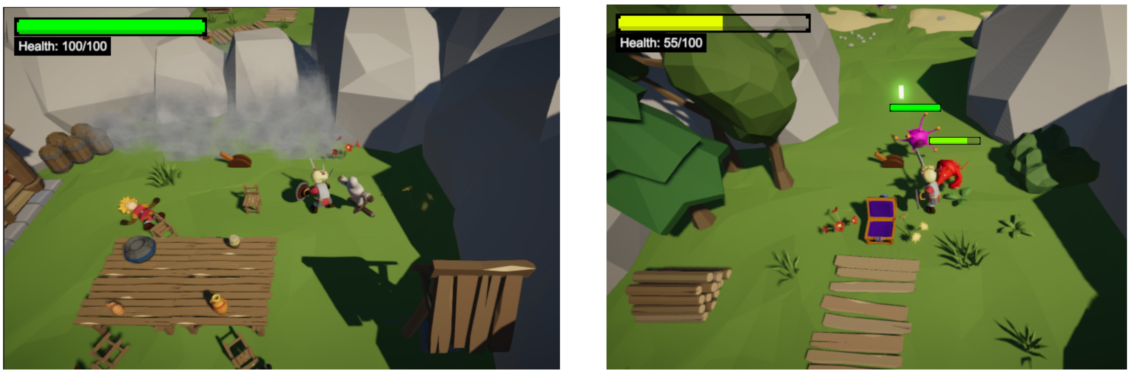
Rat Assassin - This is the first enemy type the player encounters in the woods. It is a weak monster, with low health points and only deals a little amount of damage. It has 2 different attacks: one stabbing and one swirl.

Stronger Rat Assassin - This is the first stronger version of an enemy the player finds in the woods. It is bigger and instead of the red scarf and dagger, this time they are white. While it also has 2 attacks as the normal version, they now deal a lot more damage.

Beholder - This the second enemy type the hero faces. It hovers above ground and deals more damage than the normal Rat Assassin seen before.

Stronger Beholder - The last enemy the player encounters and the most difficult to battle with. It is much bigger and instead of its normal purple skin and blue eye, this time they are black and green, respectively. It is the final boss and while its attack being the same as seen in the normal version, it is the enemy that deals the greater amount of damage in the game.

Miscellaneous - These are either props the character can interact with (which do not affect its emotional state) in order to progress through the game or simply to aid in the suitability of the environment. These comprise of: **levers** that open rock walls, a **target dummy** which allows the character to practice its attack skills, the character's **friends** that fall to ground in the beginning of the game, and finally the **dead soldiers** found near the final boss.



(a) The initial room the player is put in

(b) The second room the player encounters

Figure 4.2: The first 2 rooms, where there can be seen: (a) the players friends, the initial sword, the lever making the rock doors open and the target dummy, (b) the opened chest, the better sword, a beholder and a rat assassin and the checkpoint in green.

All of the enemies upon seeing the main character will go towards it and attack it until they either lose sight of the main character or it dies.

(a) The third room the player is put in

(b) The last room the player encounters

Figure 4.3: The last 2 rooms, where there can be seen: (a) 2 beholders and a rat assassin, (b) the normal and stronger versions of the beholders and rat assassins.

4.1.2 Tool Choice

Naturally, in order to apply the model, it makes sense to use an existing game production system that allows the model to operate in conjunction with it.

It was decided that Unity would be the best option. Not only Unity is a framework which both developers are very comfortable with but it also has a great source and amount of documentation and built-in tools. There is an active community full of answers if one wishes to understand further what can be done in the framework and to offer help. Regarding the BNs, as Unity uses mostly C#, one only needed to setup a Bayes Server .NET API from C#.

4.2 Implementation

When starting with the implementation, the first thing to be addressed was the game world which the testing would be done in.

It started with using assets found online for games in Unity. From seeing the assets, ideas started to emerge to what could be done to create a challenging experience while at the same time maintaining the player engaged in the game and aware of the character's emotions.

The initial stage was to make a moving character and a camera that would follow it. After that, it was just a question of building the game logic itself, including key bindings, movements, attacks, enemy AI, treasure findings, sound effects, particle system, a pause menu and so on. The last part was creating a game world which set the environment the character would be able to traverse and finally have a playable game.

Next it was time to integrate the game with the Bayes Server API, which took a fair amount of time to be able to do. This lies in the fact the documentation relied on the fact that who would read it already had thorough understanding of how online learning proceeded. So, to fully integrate the API onto the system one had to understand how online learning worked. It was possible to search in the API documentation to not only learn what it did but also how it this learning method was used alongside some tutorials online. This was, by far, the most challenging part of the implementation but fruitful nevertheless.

Furthermore, after knowing the Online Learning process was fully functional, it was time to give it a visual component, and that is where Unity's Graph View API was integrated. Although still experimental, it offered a way to visualize the BN and to better understand what was the architecture one was going for. The graphical depiction was coded from scratch and the connection to the Bayes Server API was done by allowing the two systems to talk to each other in order for them to copy values from one to the other. This means that one was able to add nodes, change their values and choose which nodes connected to one another before running the game. Upon running the game, this information would be sent to the Bayes Server API in order for the online learning to be done.

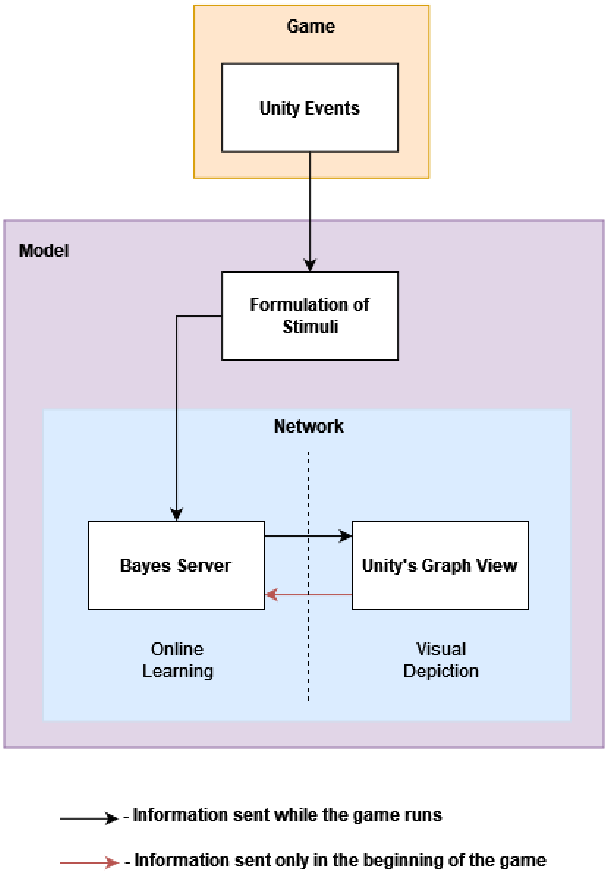


Figure 4.4: The architecture and flow of information between the game through Unity Events, the Bayes Server API and Unity's Graph View API.

After the two APIs were communicating with each other, it was time to build a way for the the game to broadcast events which in turn updated the network. For this matter, Unity Events were used and worked perfectly right from the very beginning. This is because Unity Events are a very viable when dealing with:

- **Content Driven Callbacks** - which means the system can know, at the time of the event, all the content it wants to know. This is useful, for example to know the type of enemy the character has just attacked.

- **Pre-configured Call Events** - This means that when something targeted as an event happens, it runs a certain function that is bound to it. For instance, when attacking an enemy formulate a stimulus.

- **Persistent Callbacks** - Used when one wants to persist data when the actor of the data was deleted. An example of this would be when an enemy dies, the callback for killing an enemy must be made but as the enemy dies then this data is maintained - it persists.

- **Decoupling Systems** - A term to indicate that two or more systems somehow work or are connected without being directly connected. This is exactly what happens with the game environment and the BN.

Following this, the character now formulated stimuli and updated the network. It was at this point that the model and the character modulation module were connected - making the output from the model to be the input of the module using placeholder Emotions.

After being connected, it was time to decode the values from the network into the emotions to be outputted. The mood and magnitude concepts were implemented around the same time and were thought throughout some time to fully represent a truthful appraisal from the network. These were then pipelined to be outputted to the next module.

During this stage there were 2 days in which 5 users were consulted (3 in one day and 2 in other). This was done in order to possibly ascertain whether or not a design decision should be kept or discarded in favour of another. The users would test the game (using either a gamepad or a mouse and keyboard setup) in a specific scenario and would both answer questions asked about the game as well as talk out loud their intents and character's reactions. After being consulted, the users reported the actions of the emotions of the character felt coherent and, thus, it was possible to continue to the next phase. The GANTT Chart for the implementation can be seen in Figure 4.5.

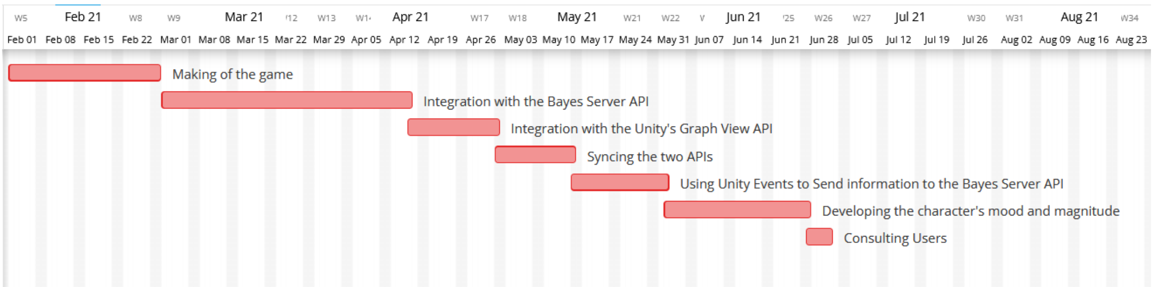


Figure 4.5: The GANTT chart of the implementation.

The implementation process described only covered the final solution one developed. Although not mentioned, many obstacles were found along the way which were then discussed and tackled in the most suitable way one found possible.

After creating the game scenario described to test the model, in order to achieve accurate findings, an adequate methodology was followed.

4.3 Test Methodology

The tests were divided into 2 parts: the pilot tests (with a control group) and the final tests. The pilot tests were created in order to validate if everything was in order with the game, the model and the character modulation module. Pilot testers would test the game and answer the questionnaire and were allowed to talk with the developers and ask questions (through an internet call) in order to tackle any issue found while testing. If then the results came out as expected, then the testing could advance to the next step.

This next step allowed one to only distribute the questionnaire's link to users allowing them to download the game, play it and answer the questionnaire. This is because one needed to ensure the players would understand how to do these steps without any exterior help as there was no possibility to be present with the players at the time of testing. However, players were given e-mail contacts and the possibility to join a Discord [3] server in order to ask any questions they had during the questionnaire. The final number of participants would comprise of both the pilot testers and the final testers as both tests were the same but done with different people.

Subjects were instructed to click on an internet link that sent them to a single questionnaire [4] that contained the game build to be downloaded and executed, as well as questions on the users' gaming preferences and demographics. Users were questioned about their gender, age, frequency of video game play, knowledge with the game genre, the importance they place on the characters' emotional

[3] Discord is a VoIP, instant messaging and digital distribution platform. Users communicate with voice calls, video calls, text messaging, media and files in private chats or as part of communities called "servers". Servers are a collection of persistent chat rooms and voice chat channels.

[4] https://shorturl.at/ksB12

expressions, and whether they would play the game using a gamepad or a keyboard and mouse configuration. Next, the players would be asked to answer 4 questions regarding what they thought was the valence of the expression demonstrated in a Graphics Interchange Format (GIF) image by the main character. This was done using a Likert scale from 1 to 5 meaning "Very Negative" and "Very Positive", respectively, with a neutral choice in 3.

They then proceeded to play the game. It was identical to the testing scenario in every way except for one very important difference: instead of using the model only (1), users were also asked to play a game version where all affective processing was done using standard techniques – either pre-scripted or reactive (2, the control scenario) - and a version to test the work from João Patrício (3). So, in total, there were 3 different versions of the game.

The users were not told which version was which, in order to not bias the results. All affective responses in the control scenario are pre-conditioned. The character was entirely reactive, displaying the most recent emotional appraisals in the same manner every time (so if it got damage it would feel fear, if it got a better sword it would get happy and so on).

As the three versions would be played by every participant, these had to be randomized for every new player that tested the game. This was done by getting the game to load a random sequence of the three versions and at the end of every playthrough, the game would display a version code which users could then select in the questionnaire which of the three they had seen and answer the questions for that specific version. Every version contained the same questions and both questions from this work and from João Patrício's were put in the same questionnaire in order not to create a bias in the players for some change that might have happened.

Regarding this work's questions, the users were asked to rate how much they agreed or disagreed with a series of statements on a Likert scale of 1 to 7, with a neutral choice in 4:

- It is easy to understand what the character is thinking or feeling.
- The world around the character influenced the character behaviour.
- The character behaved in a predictable and coherent manner.
- It is easy to understand what the character thinks will happen next.

These were based on the metric described in Section 2.4 [2]. The first question was asked in order to assess **Behavior Understandability**. If it was possible to understand the intents and feelings of the character by analysing their behaviour. The second question was done to determine if the model could demonstrate **Change With Experience**, if for different situations the charater reacted differently. The third question tackled **Predictability** and **Behaviour Coherence**, i.e. if it was possible to understand why the character behaved as it did. The last question dealt with the model's ability to predict what would happen in the future and if this was apparent to the players.

Nextly, the players were asked to report the frequency of the 4 different emotional responses in a Likert scale with a range from 1 (Never Happened) to 5 (Happened Frequently).

Not only that but users were also asked to describe one or two (one mandatory and one optional) situations in which their character expressed one of the expressions showed in the questionnaire and briefly explain what may have lead to it.

At the end of the questionnaire, users were asked to submit a zip file (which contained information about the playthrough using JavaScript Object Notation (JSON) [5] files) and to leave any comments they would like to (optional).

The JSON files contained information regarding the expression the character. This information was stored in different samples, in the same file, with an interval of 0.1 seconds. In every sample it was labelled the relative time to the beginning of the game that sample was created, the character's active and passive emotion and every event that might have happened (killing an enemy, opening a chest, amongst others). The samples also contained information for the work of João Patrício. These were collected in order to perform a "manipulation check" - to see if what the users reported was what actually happened during their playthrough.

The gameplay, as well as the questions themselves, were created to take as little time as possible while yet allowing for thorough assessment - typically, testing all three versions takes approximately 20 minutes. The tests were not timed, thus the estimate is based on observation of the small group of people who did the exam as a control group. The statements were designed to elicit user feedback on a number of critical factors that were judged significant enough to track and where the model could most likely outperform the industry standard – the findings of which can be seen in the following chapter.

The first step was, then, to find a suitable control group to compare the model against. The control group was created (with 5 people) and so, the pilot tests began. There was, however, a test outlier which appeared to not understand some of the character's behaviours at certain points. These were thought and were given a thorough inspection. Upon discussion with João Patrício, it seemed that the only difference regarding the testing was that the user was using a mouse and keyboard instead of a gamepad (which all the other users were using). After some deliberation, it was decided that two more people should be introduced to the control group to see if these conducted the same unexpected results or if we were on the presence of only an outlier. The testing was made and no other user experienced some trouble recognizing the character's emotions, allowing the testing to continue to the next step.

[5]JSON is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays.

4.4 Summary

Through this chapter one presented the testing scenario to test the model. In the description, one outlined the game features developed such as the gameplay mechanics, different rooms, items and actions. It was also presented the tool choice for the development of the game as well as why it was chosen.

The implementation of the work was presented, going through the creative process of creating the game environment as well as the game mechanics. The integration with the Bayes Server API was explained and how it was able to communicate with Unity's Graph View API. The architecture for the communication was also graphically depicted as well as why Unity Events were an important tool for the character to understand what was going on around it. After explaining the model was connected to the character modulation module, it was mentioned that the mood and magnitude were developed, thought through and pipelined to be outputted. Finally, it was told how subject tested the initial stage of the game and a GANTT Chart of the implementation was given.

In Section 4.3 it was explained how the tests were divided into 2 parts and how they came to be. An overview of the questionnaire was presented as well as the intentions behind every question. It was elucidated how information collected in the JSON files was done to perform a "manipulation check" and how both the gameplay and questionnaire were created to take little time as possible. In the end, a control group was found to compare the model and an outlier appeared but after two more people were introduced, the results came out as expected and the testing transitioned to the next stage.

The next chapter will contain the results found as well as a thorough analysis on them. The Non-Reactive Model will be compared to the Reactive Model in every dimension, and the "manipulation check" will be explained. After, a discussion of the results one collected will be made as well as their implications.

5

Results

Contents

The data was migrated into IBM's SPSS in order to analyze the testing outcomes, which involved, initially, a total of 31 individuals. Upon analysing the question regarding the valence of the expression demonstrated by the game character, 4 users were removed from the study (leaving 27 in total) as these did not understand what the character's expressions were and, thus, may not have fully understood the questionnaire at hand. Additionally, ordinal demographic data was transformed to its numeric counterpart.

5.1 Sample analysis

Of the total 27 testers, 24 identified themselves as male, 2 as female and 1 as Non-binary. The age and gender distribution can be seen in Figure 5.1.

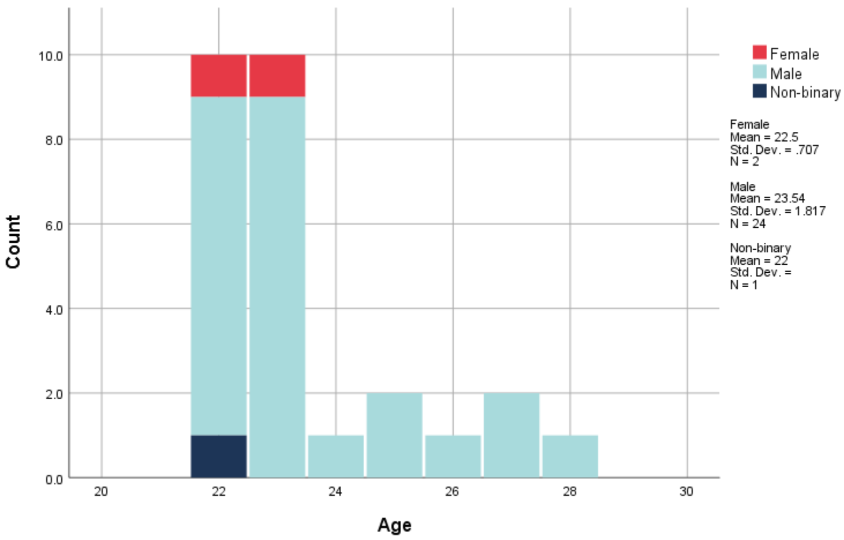


Figure 5.1: The Testers' Age and Gender distribution.

In terms of gaming habits and familiarity with the Adventure genre, the average test user schedules time to play video games (20 players, 74.07%) and enjoys the genre (23 players, 85.19%). Additionally not a single user plays video games very rarely and is not familiar with these games, which was interesting to see.

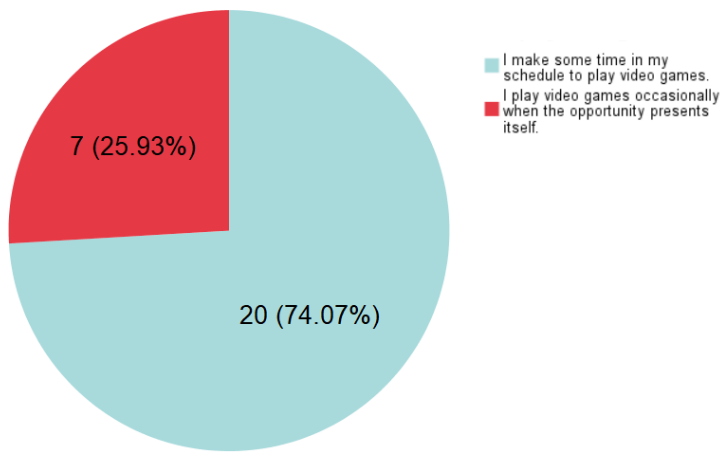


Figure 5.2: Player's gaming frequencies.

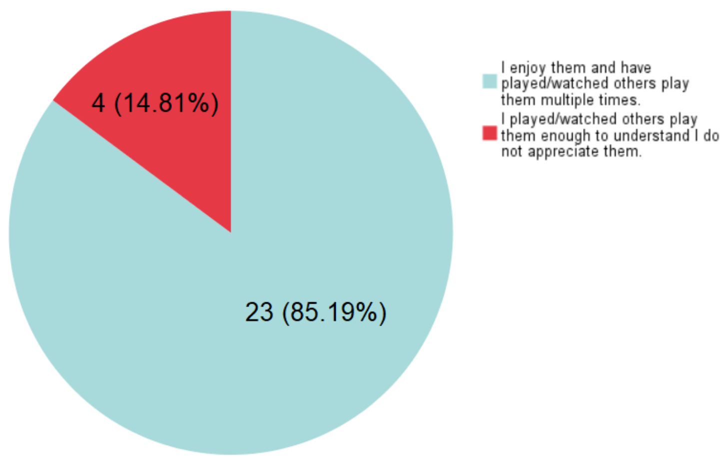


Figure 5.3: Player's familiarity with the Adventure Game genre.

Regarding if the players usually value the expression of emotions of the characters they control when playing games, 8 testers (29.63%) report that they value gameplay over the character's expression, 16 users (59.26%) said they equally value the two dynamics, and only 3 people (11.11%) addressed they value the character's expressions over the gameplay.

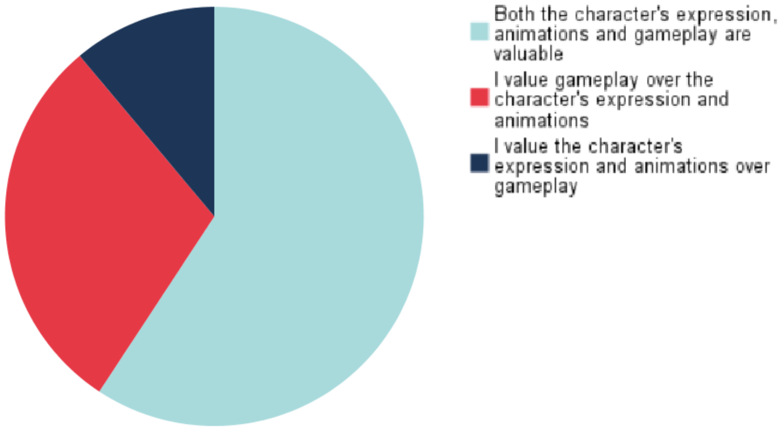


Figure 5.4: The Testers' value between character's expression and gameplay.

Regarding the way they chose to test the game, 11 testers (40.74%) used the keyboard and mouse setup while 16 (59.26%) used a gamepad.

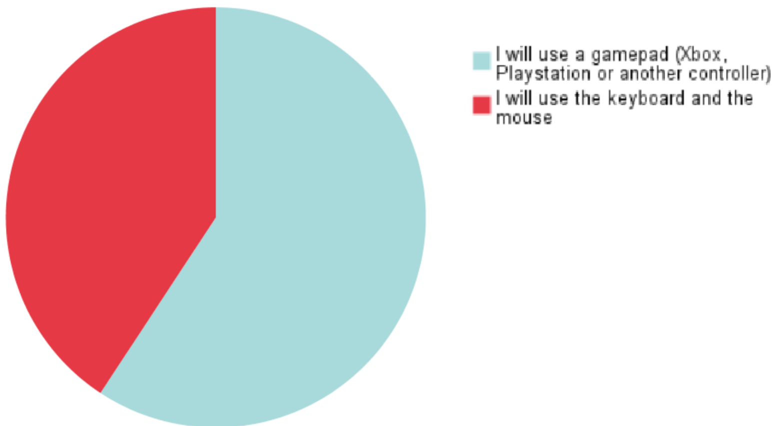
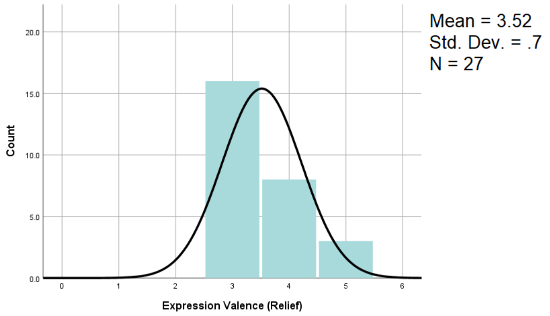
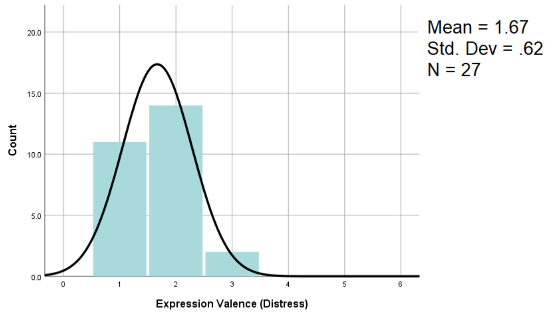


Figure 5.5: The Testers' controls choice.

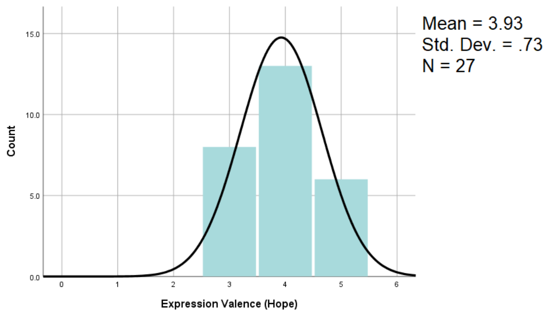
Finally, the distribution of the valence of the expression demonstrated by the main character through GIFs are as follow.



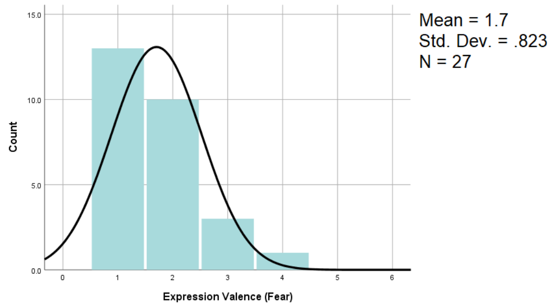
(a) Valence of the character feeling relieved.



(b) Valence of the character feeling distressed.



(c) Valence of the character feeling hopeful.



(d) valence of the character feeling afraid.

Figure 5.6: The user's reports on the valence of the character's expressions.

All of the reported valences were according to what the character was feeling being it more positive when showing **Relief** and **Hope** or more negative with **Distress** and **Fear**.

5.2 The Non-Reactive Model vs the Reactive Model

In this next section, one will look at how each of the Likert variables differs from one model to the next, starting with a short summary of the data's statistical descriptives. They are ordinal in nature, just like Likert variables, but because they are generally symmetric, it was assumed equal gaps between the ordinal values. By doing so, one may consider ordinal data as numeric in some domains, allowing one to make a wider range of statistical inferences.

Because the most robust tests require a normal distribution, The hypothesis of whether or not the data may approximate a normal distribution is tested as the first step in analyzing the connection between the models and the Likert data using the Shapiro-Wilk Tests. After examining how each variable varies from one model to the next, it is examined whether any demographics, in addition to the model, produce patterns within the Likert variables.

5.2.1 Statistical Descriptives

	N	Mo	µ	var.
It is easy to understand what the character is thinking or feeling.	27	5	4.81	2.54
	27	5	4.89	3.41
The world around the character influenced the character behaviour.	27	5	5.26	1.97
	27	6	5.70	1.83
The character behaved in a predictable and coherent manner.	27	5	5.15	1.52
	27	5	5.44	1.95
It is easy to understand what the character thinks will happen next	27	4	4.41	2.33
	27	5	4.56	3.26

Table 5.1: Likert variable's descriptive statistics presented with the Non-Reactive Model values below the Reactive Model

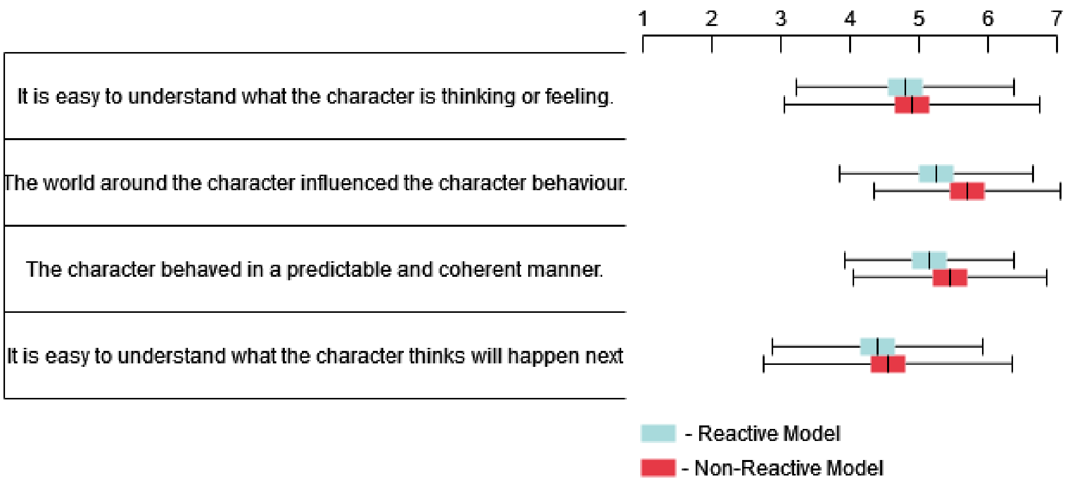


Figure 5.7: Chart of the Likert variable spread of the answers, showing for each an interval from $\mu - \sigma$ to $\mu + \sigma$.

In Figure 5.7, a brief review of the descriptives, indicates comparable findings in most areas, with a larger variance on the ease of understanding what the character is thinking or feeling and the ease of understanding what the character expects will happen next. Also, because the means in both models are similar, they focus shift more on the middle two assertions, which both relate to the character's behavioural component.

5.2.2 It is easy to understand what the character is thinking or feeling

This first question was asked as it was crucial to evaluate **Behavior Understandability**. Second, this element (in combination with others) served as a measure of the character's emotional believability; if the majority of testers gave a low score to this variable, the testing scenario would have had to be re-evaluated. However, this was not the case, with 63% of those surveyed responding positively, 15%

neutral/ambivalently, and 22% negatively. Both emotional models have statistical descriptives that are extremely comparable, implying that there is no difference from employing a reactive model.

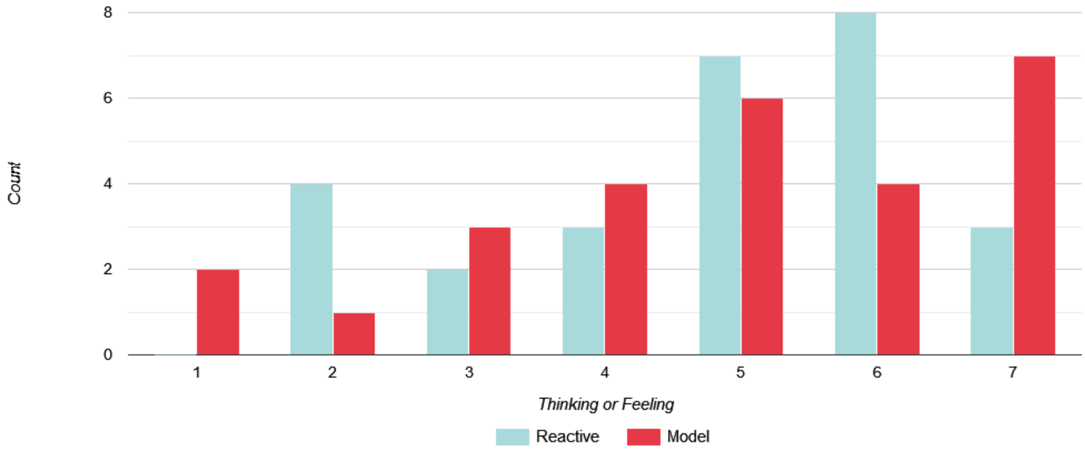


Figure 5.8: Thinking or Feeling variable grouped by model.

Both distributions have comparable histograms, with the reactive model having considerably more responses in the 5 and 6 values. To make sure that these responses fit within a normal distribution, on both of them, a Shapiro-Wilk test is run, which rejects the null hypothesis ($p_{V1} = 0.007 < 0.05, p_{V2} = 0.017 < 0.05$, Appendix A.1.1, Figure A.1) that the distributions approximate normal distributions.

Because neither distribution should be deemed normal at this point, a non-parametric analysis is done. A Wilcoxon Signed-Rank Test is a suitable alternative since the dependent variable is ordinal and the independent variable has two related groups (both groups played the reactive model and the non-reactive model) (Appendix A.1.2, Figure A.2). In this example, the test revealed ($p = 0.740 > 0.05$) that there is no statistically significant shift from one distribution to the other, implying that there is no statistical difference between the two models in terms of what the character was thinking or feeling.

5.2.3 The world around the character influenced the character behaviour

The next question was asked in order to assess if the users felt the character's behaviour changed with experience and at what level did it change. Unlike the preceding assertion, responses were less uniform, with 85% positive, and both the remaining 7-8% percent negative and neutral or indifferent.This difference may be seen in the descriptive statistics, as the prior statement had an average of 4.80 - 4.90, but this one had an average of 5.30 - 5.70. In addition, unlike the preceding variable, in the emotional model, mode moves from 5 to 6. When compared to the first variable, the variance is significantly smaller, but unlike the prior variable, there appears to be less polarization of viewpoints, as seen by the histogram.

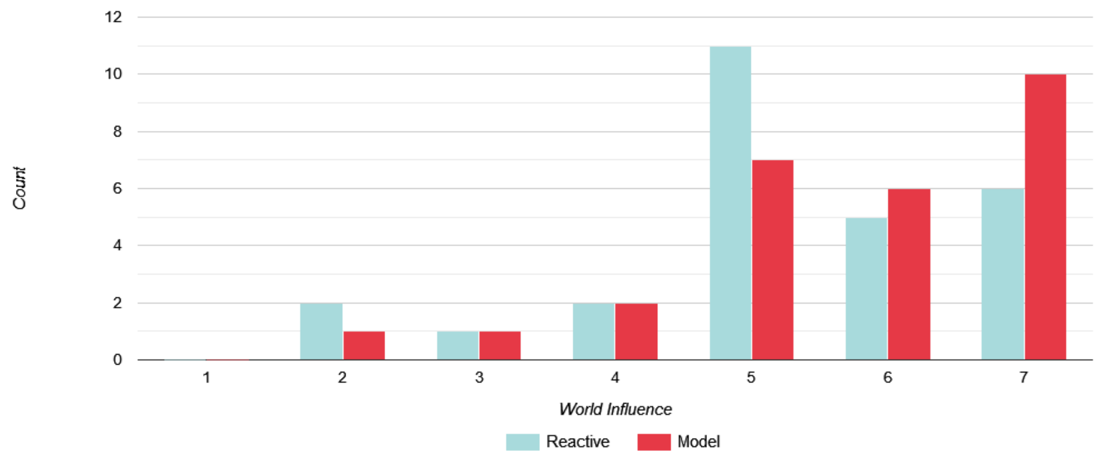


Figure 5.9: World Influence variable grouped by model.

A visual inspection shows two almost distinct distributions, with emotional model having a significantly larger amount of really positive responses. A Shapiro-Wilk Test is used on each of them to ensure that the answers fit inside a normal distribution. This test rejects the null hypothesis ($p_{V1} = 0.004 < 0.05, p_{V2} = 0.001 < 0.05$, Appendix A.1.1, Figure A.1) that the distributions approximate normal distributions.

A Wilcoxon Signed-Rank Test (Appendix A.1.2, Figure A.2) is used as before, and it reveals that, although close to being significant, there is no statistically significant shift from one data set to another ($p = 0.059 > 0.05$), implying that there is no statistical difference in the world influence effect in the character's behaviour between one model and the other.

5.2.4 The character behaved in a predictable and coherent manner

The third question was done in order to determine both **Predictability** and **Behavior Coherence** from the character - is the character's behaviour made sense regarding what happened to it. This section focuses on the immediate cause-and-effect relationship that may be seen in direct character actions as well as in relation to previous events. Not only did it show an increase in the variance, but also showed practically no noticeable shift in the average from the reactive to the emotional model (5.15 - 5.44, respectively).

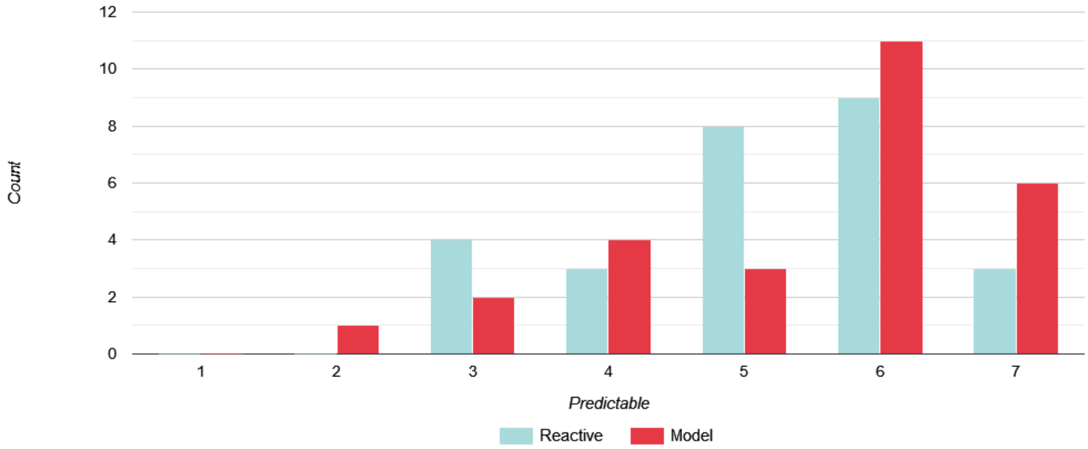


Figure 5.10: Predictable variable grouped by model.

The histograms of both distributions are similar, with the non-reactive model having significantly more responses in the 6 and 7 values. A Shapiro-Wilk Test is used on each of them to ensure that the answers fit inside a normal distribution. This rejects the null hypothesis ($p_{V1} = 0.011 < 0.05, p_{V2} = 0.002 < 0.05$, Appendix A.1.1, Figure A.1) that the distributions approximate normal distributions.

As previously, the Wilcoxon Signed-Rank Test (Appendix A.1.2, Figure A.2) is performed, and it indicates that there is no statistically significant shift from one data set to the next ($p = 0.349 > 0.05$), suggesting that there is no statistical difference in character predictability and behavior coherence between the two models.

5.2.5 It is easy to understand what the character thinks will happen next

The last variable to be examined is to determine the user's perception of the character's expectations. The variance increases considerably across the models (from 2.33 to 3.26 in the reactive and non-reactive models, respectively), while the mean does not change significantly (from 4.41 to 4.56).

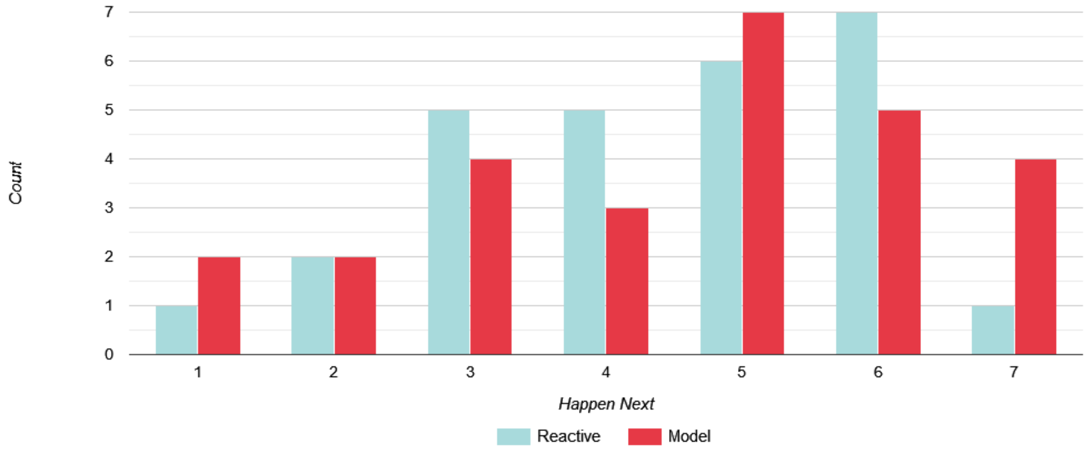


Figure 5.11: Happen Next variable grouped by model.

Both distributions have comparable histograms, but the reactive model has substantially more responses in the 6 value. A Shapiro-Wilk test, however, indicates that the data can be examined using a parametric test, as the result approximate from a normal distribution ($p_{V1} = 0.099 > 0.05, p_{V2} = 0.062 > 0.05$, Appendix A.1.1, Figure A.1).

In assuming a normal distribution it can be applied a Dependent T-test to both samples (Appendix A.1.3, Figure A.3). Due to the means of the two jumps and the direction of the t-value, we can conclude that there was no statistically significant improvement in the understanding what the character thinks will happen next ($p = 0.589 > 0.05$). The Wilcoxon Signed-Rank Test (Appendix A.1.2, Figure A.2) indicate the same results.

5.2.6 Demographics sub-groups

Apart from examining each statement in connection to each model, it was also considered to determine whether there was any correlation between them and the sample's demographics. Upon reviewing them, it was considered that the only sub-group which had enough data to be analysed and possibly yield significant results was the player's control choice. The other demographics presented were also very polar as the majority were dedicated male players who enjoyed the type of game in the test and valued both the character's expression and gameplay. The sub-group regarding which control scheme was used by the players was also the one who upon discussion with João Patrício seemed to yield different results, even in the control group from the beginning.

As only 11 testers (40.74%) used the keyboard and mouse setup and 16 (59.26%) used a gamepad, one decide to use non-parametric tests to determine or not if there was a significant statistical change in the results.

Firstly, it is important to understand if there are no dependencies between the demographics sub-groups. If, for example, every dedicated player used a mouse and a keyboard to test the game then the cause for the possible results can either be because of using that control scheme or because they are dedicated players. Kendall's tau-b determines whether there is a monotonic relationship between two given variables. A monotonic relationship exists when either the variables increase in value together, or as one variable value increases, the other variable value decreases. It also allows for the two variables to be measured on an ordinal scale, which they are.

The correlation between every demographics sub-group was run and there no statistical significant correlation between any of them ($p > 0.05$, Appendix A.1.4, Figure A.4).

As there were no correlation, a Mann-Whitney U-test is an appropriate alternative to understand if the control scheme used by the players influenced the character believability. This is because the dependent variable is ordinal and the independent variable has two separate groups (reactive model and non-reactive model).

Test Statistics[a]

	V1_ThinkingFeelingNumeric	V1_WorldInfluenceNumeric	V1_PredictableNumeric	V1_HappenNextNumeric	V2_ThinkingFeelingNumeric	V2_WorldInfluenceNumeric	V2_PredictableNumeric	V2_HappenNextNumeric
Mann-Whitney U	35.000	71.000	83.000	48.000	38.500	65.500	60.000	43.500
Wilcoxon W	101.000	137.000	149.000	114.000	104.500	131.500	126.000	109.500
Z	-2.681	-.877	-.256	-2.015	-2.487	-1.157	-1.442	-2.231
Asymp. Sig. (2-tailed)	.007	.381	.798	.044	.013	.247	.149	.026
Exact Sig. [2*(1-tailed Sig.)]	.008[b]	.422[b]	.827[b]	.050[b]	.013[b]	.272[b]	.178[b]	.026[b]

a. Grouping Variable: How will you play the game?

b. Not corrected for ties.

Figure 5.12: Mann-Whitney U-test results for the Likert variables.

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles 25th	50th (Median)	75th
V1_ThinkingFeelingNumeric	27	4.8148	1.59415	2.00	7.00	4.0000	5.0000	6.0000
V1_WorldInfluenceNumeric	27	5.2593	1.40309	2.00	7.00	5.0000	5.0000	6.0000
V1_PredictableNumeric	27	5.1481	1.23113	3.00	7.00	4.0000	5.0000	6.0000
V1_HappenNextNumeric	27	4.4074	1.52566	1.00	7.00	3.0000	5.0000	6.0000
V2_ThinkingFeelingNumeric	27	4.8889	1.84669	1.00	7.00	4.0000	5.0000	7.0000
V2_WorldInfluenceNumeric	27	5.7037	1.35348	2.00	7.00	5.0000	6.0000	7.0000
V2_PredictableNumeric	27	5.4444	1.39596	2.00	7.00	4.0000	6.0000	6.0000
V2_HappenNextNumeric	27	4.5556	1.80455	1.00	7.00	3.0000	5.0000	6.0000
How will you play the game?	27	1.41	.501	1	2	1.00	1.00	2.00

Figure 5.13: Descriptive statistics of the Mann-Whitney U-test results for the Likert variables.

As it can be seen in Figure 5.12 and Figure 5.13, the results seem to show that an important factor for the believability of the character is the hardware used to control it, at least in the context of the believability dimensions one explored. This can be seen in the dimension of understanding what the character is thinking or feeling (Reactive: ($U = 35.0$, $p = 0.007$, $\mu = 4.81$, $Median = 5.0$) and Non-Reactive: ($U = 38.5$, $p = 0.013$, $\mu = 4.89$, $Median = 5.0$)) and in the dimension of understanding what the character thinks will happen next (Reactive: ($U = 48.0$, $p = 0.044$, $\mu = 4.41$, $Median = 5.0$) and Non-Reactive: ($U = 43.5$, $p = 0.026$, $\mu = 4.56$, $Median = 5.0$)). The medians show no difference and the means increase in both cases.

5.2.7 Manipulation Check

It is important to address the fact that the results gathered from the users were reviewed and compared to what was really being shown in the game at the time of playing (using the data collected and stored in the JSON files). For instance, for every version of the game the players tested, there would be a chart created for the frequency of every passive and active emotion. These frequencies were calculated, at each time step of 0.1 seconds, given the value of the magnitude of that emotion. So if the magnitude of the emotion had a value closer to 1, then it would count as being more frequent then if it had, say, a value closer to 0. This was done as a way to represent if the emotion was apparent or not and its intensity. This means that the more apparent the emotions were (with magnitude closer to 1) then the higher impact on the frequency than those which were less apparent (magnitude closer to 0). The values from every sample were then averaged to have a value between 0 and 1, for better comparison between the different emotions. An example from the frequencies gathered can be seen in Figure 5.14, and in Figure 5.15 what the user reported for that playthrough.

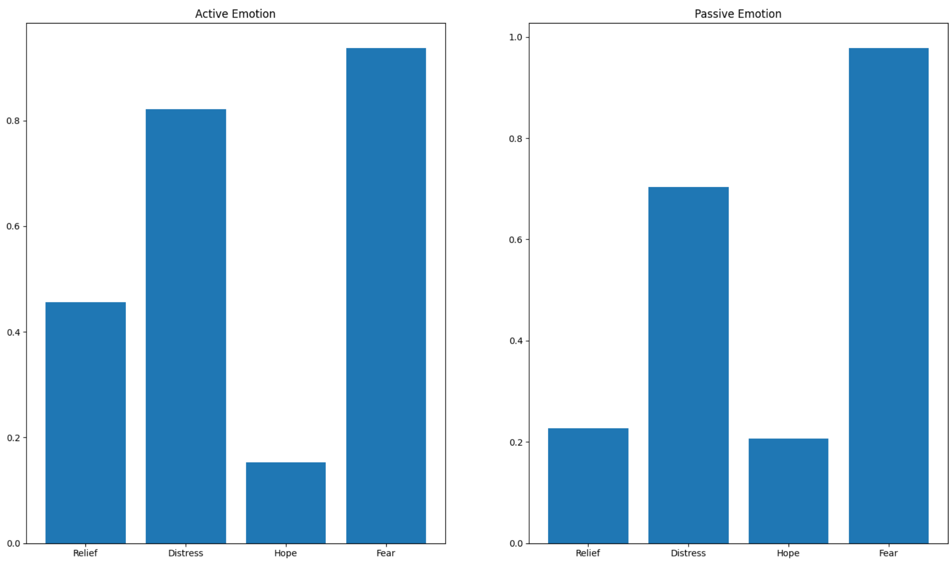


Figure 5.14: The frequencies gathered from a version playthrough.

	V2_Frequency Relief	V2_Frequency Distress	V2_Frequency Hope	V2_Frequency Fear
4	3	5	2	4

Figure 5.15: The frequencies reported from the users.

When comparing the user reports of the frequencies of the expressed emotions in their playthrough with what was effectively expressed by the the game, there were many cases that these did not match. In particular, there was a bias towards reporting more negative emotions (like fear) than what actually happened in-game. As the animations shown for this emotion were the most apparent, reviewing what the users reported when asked to describe one or two situation, this could be the explanation to why this might have happened.

Given that these were viewed and interpreted correctly by the vast majority of players in the beginning of the questionnaire, then, the only factor that changed was that this time the players were actually in-game and not explicitly rating the valence of the emotion the character is showing.

5.3 Discussion

From the sections before, it was seen that there was no improvement in using this non-reactive model to ease the understanding of what the character is thinking or feeling. It was expected that the model would lead to more believable characters, however a reactive model is naturally very simple in design, taking less effort to master/comprehend but, as a result, giving considerably less depth. Also, because the reactive model usually produced extreme examples of emotions (values of magnitude equal to 1 and mood equal to -1/1), the testers may be able to discern the feelings the character was presenting more clearly.

This model also did not improve the dimension of the world influence in the character's behaviour. Although the character reacts mainly to the same situations, yielding results very close to each other i.e. not being statistically significant, this variable was close to being significant which was a dimension treated and thought about when creating the solution.

As for the character's behaviour predictability and coherence, it was also not seen a major difference. This is less expected, since a character with a more human-like emotional model would have behaved in a more believable and predictable manner rather than the reactive, which was just reacting to circumstances on sight with no recollection of what occurred. However, it might be claimed that the knowledge of recalling what happened in the past was not immediately evident to the participants, leading them to believe that emotional appraisals were arriving out of nowhere.

Additionally, this model did not increase the ease to understand what the character thinks will happen next. This is also less expected as in the reactive model the character does not hold any memory from what happened and so can not predict the future. This is not true for the non-reactive. In this case, the character remembers past events and tries to anticipate what will happen in the next interaction with the entity, being it an enemy or a chest. Nonetheless, it can also be argued that what the character portrays was not in itself wrong yet this was not what it thought would happen next - which could have been misinterpreted.

Regarding the sub-groups, it was seen that there were no statistical significant correlations between them and that the results appear to suggest that, at least in the context of the believability dimensions examined, the hardware utilized to control the character is a significant element in its believability both in terms of understanding what the character is thinking or feeling and also understanding what the character thinks will happen next. This result, however, is not fully exact and can stem from other causes not known to date. As such, this is a component which deserves a better study in the future.

The animations presented by the game to the player may not be correctly demonstrating the pretended emotions as well as the players not being given a certain guidance to focus on the character's behaviour and expression. These, however, could introduce a bias which was not intended for the purpose of a thorough evaluation procedure.

If the character's animation were to be bettered, then the results might yield different values from those which have been shown. Not only that but also because, as the game includes many fighting scenarios, it may also be difficult for the users to correctly identify the emotion the character is feeling or even to be looking at the character attentively. In the questionnaire, the question was also asked in a likert scale, with the extremes being "Never Happened" and "Happened Frequently". However, these are more relative terms, i.e. what might have been frequent to a user may not be as frequent to another one.

6

Conclusions

This thesis started with asserting the importance of character believability. It was presented the criteria to be enhanced in the context of games and what its dimensions were. Emotions present in games were discussed and how they can appear in conjunction with others. One showed how they are formulated, how there are some factors that influence them and notes in psychology as well as some state of the art work regarding computable emotions. It was then introduced the model which one would be working on, how it can be of help to solve the problem at hand and related work in the field. Then, is was presented how it was possible to assess character believability through questionnaires.

Following that, one went through a general overview of the model, noting the key differences between the game, the model itself, and how each should communicate with one another. Following that, the model was dissected in further detail, including descriptions of how each stimulus is generated and how the network stores the character's memories. Confidence levels were also explained. how they are determined, as well as the mood and magnitude . Finally, how these are combined to produce either an active or passive emotion that is then passed to the following module.

One looked at how key Likert variables stayed constant across both models. One speculated that this was either due to the game's animations which, although tested in detail, may have somewhat not expressed the emotions pretended or because of the lack of information given to the players to be aware of the character's feelings (which could introduce an unwanted bias). It's also possible that the shift in focus while in combat circumstances played a significant role. As such, the control which the user plays may be a significant element in demonstrating the credibility of a character both in terms of understanding what it is thinking or feeling and also understanding what the it thinks will happen next. These, however, may not be fully exact, meaning that if features like the character's animations were to be bettered, then the results might yield different values from those which have been shown.

For future work, one can start by addressing the limitations from this work regarding the animations and how these portray the character's emotions. From that point, this work was made in order to be the basis of a more thorough model. This means that integrating more types of stimuli is a good starting point. Not only that but giving characters a personality could add more depth to the emotions felt by the characters. Additionally, the game is currently single-player only. It would be interesting to have a multiplayer version where characters would have a sort of affinity (or aversion) to one another. Exploring the concept of Theory of Mind could have an impact on the relations of said characters. These works could then impact other areas of study between virtual agents and humans such as empathy.

Bibliography

[1] E. Hudlicka, "Affective computing for game design," *4th International North-American Conference on Intelligent Games and Simulation, Game-On 'NA 2008*, pp. 5–12, 01 2008.

[2] P. Gomes, C. Martinho, A. Paiva, and A. Jhala, *Metrics for Character Believability in Interactive Narrative*. ICIDS, 11 2013, vol. 8230.

[3] S. Coleridge, J. Engell, and W. Bate, *Biographia literaria, or, Biographical sketches of my literary life and opinions*. London: Routledge and Kegan Paul, 1983.

[4] W. Wordsworth, S. Coleridge, and M. Schmidt, *Lyrical ballads: With a few other poems*. London: Penguin, 2006.

[5] F. Thomas and O. Johnston, *Disney animation: The illusion of life*. New York: Abbeville Press, 1981.

[6] J. Lasseter, "Principles of traditional animation applied to 3d computer animation," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '87. New York, NY, USA: Association for Computing Machinery, 1987, p. 35–44. [Online]. Available: https://doi.org/10.1145/37401.37407

[7] J. Bates, "The role of emotion in believable agents," *Commun. ACM*, vol. 37, no. 7, p. 122–125, Jul. 1994. [Online]. Available: https://doi.org/10.1145/176789.176803

[8] A. Loyall, *Believable Agents: Building Interactive Personalities*. Carnegie Mellon University, 1997.

[9] M. Mateas, *An oz-centric review of interactive drama and believable agents*. Carnegie Mellon University, 1999.

[10] A. Ortony, *Emotions in Humans and Artifacts, chapter On making believable emotional agents believable*. MIT Press, 2003.

[11] R. McKee, *Story: Substance, Structure, Style, and the Principles of Screen-writing*. Harper Collins Publishers, 1997.

[12] J. C. Lester and B. A. Stone, “Increasing believability in animated pedagogical agents,” in *Proceedings of the first international conference on Autonomous agents*. ACM, 1997, p. 16–21.

[13] P. R. Kleinginna and A. M. Kleinginna, “A categorized list of emotion definitions, with suggestions for a consensual definition,” *Motivation and Emotion*, vol. 5, no. 4, pp. 345–379, Dec 1981. [Online]. Available: https://doi.org/10.1007/BF00992553

[14] K. R. Scherer, T. Banziger, and E. Roesch, *A Blueprint for Affective Computing: A Sourcebook and Manual*, 1st ed. USA: Oxford University Press, Inc., 2010.

[15] B. P. Lee, E. C. Kao, and V. Soo, “Feeling ambivalent: A model of mixed emotions for virtual agents,” in *Intelligent Virtual Agents, 6th International Conference, IVA 2006, Marina Del Rey, CA, USA, August 21-23, 2006, Proceedings*, ser. Lecture Notes in Computer Science, J. Gratch, R. M. Young, R. Aylett, D. Ballin, and P. Olivier, Eds., vol. 4133. Springer, 2006, pp. 329–342. [Online]. Available: https://doi.org/10.1007/11821830_27

[16] E. Oliveira and L. Sarmento, “Emotional advantage for adaptability and autonomy,” in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS ’03. New York, NY, USA: Association for Computing Machinery, 2003, p. 305–312. [Online]. Available: https://doi.org/10.1145/860575.860625

[17] A. L. C. Bazzan and R. H. Bordini, “A framework for the simulation of agents with emotions,” in *Proceedings of the Fifth International Conference on Autonomous Agents*, ser. AGENTS ’01. New York, NY, USA: Association for Computing Machinery, 2001, p. 292–299. [Online]. Available: https://doi.org/10.1145/375735.376313

[18] P. J. Gmytrasiewicz and C. L. Lisetti, “Using decision theory to formalize emotions in multi-agent systems,” in *Proceedings Fourth International Conference on MultiAgent Systems*, 02 2000, pp. 391–392.

[19] J. Larsen, A. P. Mcgraw, and J. Cacioppo, “Can people feel happy and sad at the same time?” *Journal of personality and social psychology*, vol. 81, pp. 684–96, 10 2001.

[20] M. P. Eladhari and M. Sellers, “Good moods: Outlook, affect and mood in dynemotion and the mind module,” in *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, ser. Future Play ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 1–8. [Online]. Available: https://doi.org/10.1145/1496984.1496986

[21] S. Erk, B. Abler, and H. Walter, “Cognitive modulation of emotion anticipation,” *The European journal of neuroscience*, vol. 24, pp. 1227–36, 09 2006.

[22] C. Martinho, "Emotivector: Mecanismo afectivo e anticipatório para personagens sintéticas. ph.d. thesis," *Instituto Superior Técnico*, 09-2007.

[23] J. Barbalet, "Social emotions: Confidence, trust and loyalty," *International Journal of Sociology and Social Policy*, vol. 16, pp. 75–96, 12 1996.

[24] N. H. Frijda, "The place of appraisal in emotion," *Cognition and Emotion*, vol. 7, no. 3-4, pp. 357–387, 1993. [Online]. Available: https://doi.org/10.1080/02699939308409193

[25] R. S. Lazarus, "Progress on a cognitive-motivational-relational theory of emotion." *American Psychologist*, vol. 46, no. 8, pp. 819–834, 1991. [Online]. Available: https://doi.org/10.1037/0003-066X.46.8.819

[26] J. Dias and A. Paiva, "Feeling and reasoning: A computational model for emotional characters," in *Progress in Artificial Intelligence*, C. Bento, A. Cardoso, and G. Dias, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 127–140.

[27] A. Ortony, G. Clore, and A. Collins, *The Cognitive Structure of Emotion*. Cambridge University Press, 01 1988, vol. 18.

[28] M. Ochs, N. Sabouret, and V. Corruble, "Simulation of the dynamics of nonplayer characters' emotions and social relations in games," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 1, pp. 281 – 297, 01 2010.

[29] P. M. F. Pimentel, "Gameplay aware emotional model for virtual characters in video games," *Master's thesis, Instituto Superior Técnico, Portugal*, 2016.

[30] K. Murphy, "A brief introduction to graphical models and bayesian networks," in *http://www.cs.berkeley.edu/ murphyk/Bayes/bayes.html*, 1998.

[31] D. Spiegelhalter, "Bayesian graphical modelling: A case-study in monitoring health outcomes," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 47, pp. 115 – 133, 01 2002.

[32] I. Koromila, Z. Nivolianitou, and T. Giannakopoulos, "Bayesian network to predict environmental risk of a possible ship accident," in *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '14. New York, NY, USA: Association for Computing Machinery, 2014. [Online]. Available: https://doi.org/10.1145/2674396.2674463

[33] C. Vlek, H. Prakken, S. Renooij, and B. Verheij, "Modeling crime scenarios in a bayesian network," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*, ser. ICAIL '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 150–159. [Online]. Available: https://doi.org/10.1145/2514601.2514618

[34] Y. Liu and X. Wang, “The analysis of driver’s behavioral tendency under different emotional states based on a bayesian network,” *IEEE Transactions on Affective Computing*, pp. 1–1, 2020.

[35] P. Ekman, “An argument for basic emotions,” *Cognition and Emotion*, vol. 6, no. 3-4, p. 169–200, 1992.

[36] P. F. Gomes, C. Martinho, and A. Paiva, “I’ve been here before! location and appraisal in memory retrieval,” in *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, Taipei, Taiwan, 2011, p. 1039–1046, iFAAMAS, 201.

A

Annexes

A.1 The model vs the Reactive model

V1 refers to the Reactive model while **V2** refers to the Non-Reactive model.

A.1.1 Shapiro-Wilk Tests

Tests of Normality

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
V1_ThinkingFeelingNumeric	.213	27	.003	.889	27	.007
V1_WorldInfluenceNumeric	.242	27	.000	.875	27	.004
V1_PredictableNumeric	.200	27	.007	.896	27	.011
V1_HappenNextNumeric	.170	27	.045	.936	27	.099
V2_ThinkingFeelingNumeric	.154	27	.102	.904	27	.017
V2_WorldInfluenceNumeric	.201	27	.006	.850	27	.001
V2_PredictableNumeric	.284	27	.000	.867	27	.002
V2_HappenNextNumeric	.190	27	.014	.928	27	.062

a. Lilliefors Significance Correction

Figure A.1: Shapiro-Wilk tests.

A.1.2 Wilcoxon Signed-Rank Tests

Test Statistics[a]

	V2_ThinkingFeelingNumeric - V1_ThinkingFeelingNumeric	V2_WorldInfluenceNumeric - V1_WorldInfluenceNumeric	V2_PredictableNumeric - V1_PredictableNumeric	V2_HappenNextNumeric - V1_HappenNextNumeric
Z	-.332[b]	-1.885[b]	-.936[b]	-.783[b]
Asymp. Sig. (2-tailed)	.740	.059	.349	.434

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

Figure A.2: Wilcoxon Signed-Rank tests.

A.1.3 Dependent T-Test

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper			
Pair 1	V1_HappenNextNumeric - V2_HappenNextNumeric	-.14815	1.40613	.27061	-.70440	.40810	-.547	26	.589

Figure A.3: Dependent T-Test

A.1.4 Kendall's Tau-b

Correlations

			GenderNumeric	How frequently do you play video games?	Do you enjoy "Adventure Games" (e.g. Legend of Zelda, Don't Starve)?	How do you usually value the expression of emotions of the characters you control when playing games?	How will you play the game?
Kendall's tau_b	GenderNumeric	Correlation Coefficient	1.000	-.196	.351	.090	.088
		Sig. (2-tailed)	.	.310	.070	.628	.651
		N	27	27	27	27	27
	How frequently do you play video games?	Correlation Coefficient	-.196	1.000	.229	-.251	.025
		Sig. (2-tailed)	.310	.	.243	.184	.897
		N	27	27	27	27	27
	Do you enjoy "Adventure Games" (e.g. Legend of Zelda, Don't Starve)?	Correlation Coefficient	.351	.229	1.000	-.147	.079
		Sig. (2-tailed)	.070	.243	.	.435	.689
		N	27	27	27	27	27
	How do you usually value the expression of emotions of the characters you control when playing games?	Correlation Coefficient	.090	-.251	-.147	1.000	-.005
		Sig. (2-tailed)	.628	.184	.435	.	.978
		N	27	27	27	27	27
	How will you play the game?	Correlation Coefficient	.088	.025	.079	-.005	1.000
		Sig. (2-tailed)	.651	.897	.689	.978	.
		N	27	27	27	27	27

Figure A.4: Kendall's Tau-b