Using Bayesian Networks to Support Believable Expression of Emotions in Games

AFONSO VIEIRA, Instituto Superior Técnico, Portugal

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 o 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.

CCS Concepts: • **Computing methodologies** → **Intelligent agents**.

Additional Key Words and Phrases: Believability, Emotions, Bayesian Networks, Synthetic Characters, Games

1 INTRODUCTION

Emotions portrayed by game characters tend to be scripted, non-organic and reactive, i.e. 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. As a way of improving the coherence in the expression of emotions, not only the relationship between the player and their character but also the character's interaction with the game's environment should have an impact on the feel of a game by showing some feedback on the game character during the gameplay.

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), will improve the **Character Believability** in an adventure game. The testing was done in two scenarios: one without using the model, using a purely reactive character and the other with the model. Considering the said concept as a measuring manner, as the model gets more complex in the second scenario, it is expected that the **Character Believability** levels get higher with this added complexity, notably in the areas of *Behavior Understandability* and *Behaviour Coherence*. The second scenario will, then, reflect a higher level of character believability and thus be the better option to adopt for game design purposes. The hypotheses were tested in a game made from scratch alongside a master thesis' colleague (João Patrício).

2 RELATED WORK

Character Believability has been discussed since the nineteenth century and its interpretation has been ever-changing. From poetry [Coleridge et al. 1983], to Disney Animation[Thomas and Johnston 1981] and 3d computer animation [Lasseter 1987]. Carnegie Mellon researchers working on the *OZ* project made a substantial contribution of animated characters which could be utilized to create realistic bots with "the illusion of life" [Bates 1994] [Loyall 1997] [Mateas 1999]. Ortony [Ortony 2003] offered a more emotion-focused concept for believable agents. This believability criteria among others, offer AI designers guidance for creating systems that enable believable characters.

Metrics for **Character Believability** have been proposed. For example, in [Gomes et al. 2013] a metric was proposed to incorporate many believability factors into the overall sense of believability. These are: Behavior Coherence, Change With Experience, Awareness, Behavior Understandability, Personality, Emotional Expressiveness, Social, Visual Impact and Predictability.

In the case of this work, the **character believability** was shown through the character's **emotions**. The definitions for the latter are manifold [Kleinginna and Kleinginna 1981]. However, most theorists consider it to be "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." [Scherer et al. 2010].

These are important to account for virtual agents to behave as if they were real [Lee et al. 2006] [Bates 1994] [Oliveira and Sarmento 2003]. Nonetheless, emotions can appear alongside others, i.e. they can be **ambivalent**. "Happiness" can correspond to various combinations of pleasure, delight, amity, among other as seen in [Eladhari and Sellers 2008].

Anticipation is a key aspect that can influence the emotion one can manifest as preparing for upcoming events allows planning of behavioral strategies and action preferences that ensure survival in an ever-changing environment [Erk et al. 2006]. 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.

Moreover, **mood** greatly affects the emotions one can feel as well as their intents. When in a good mood a subject 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 - they will feel more positive emotions. 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.

Affective models can be seen as appraisal theory approaches [Scherer et al. 2010]. Appraisal theory presupposes that all emotions come in largely through the subject's interpretation of events. Appraisal theory applications have been done in the past [Dias and Paiva 2005] [Ochs et al. 2010]. These could model a range of the emotions, including coping mechanism to deal with specific goals and individual personality and also are an excellent examples of how consequences of events, object characteristics may all contribute to increased believability.

The model from Pimentel [Pimentel 2016] 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. Different ways of encounters take place with the objects as the game unfolds, leading to different evaluation interpretations for the same object. Based on the **Emotivector** [Martinho 2007], the sensorial input is split into several groups, according to what the agent expects and its valence: expecting a punishment and receiving a reward prompts the model to return happiness and satisfaction. Pimentel's approach served as a foundation for this work.

When constructing a model, it is important to calculate, what the character would expect to happen, provided the current world status. Given their versatility, one opted to use Bayesian Networks (BNs). A BN is a probabilistic graphical model that represents a set of variables and their conditional dependencies through a Directed Acyclic Graph (DAG) [Murphy 1998]. If a causal probabilistic dependence occurs between two random variables in the graph, a directed edge connects the two respective nodes [Murphy 1998], while the directed edge from a node A to a node B indicates that the random variable A causes the random variable B.

Causes are believed to be independent when there is no edge between the two causal nodes, although this assumption is not generally necessary. Because a DAG is a hierarchical system, the usage of words such as parent, child, ancestor, or descendant for certain nodes is unambiguous[Spiegelhalter 2002].

The whole definition of BN 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.

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**.

One may construct a full probability model by defining only the distribution of conditional probability in each node[Spiegelhalter 2002]. Any 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 aid in understanding if given a situation is expected or not.

Through *Belief Propagation*, children can propagate their beliefs to parents and vice versa using the **likelihood**, **priors** and said **belief**. It is then possible to undergo the process of *Parameter Learning* which uses data to learn the distributions of a BN. The expectation-maximization algorithm is a classical approach to the direct maximization of the posterior probability.

The choice of using BNs lies mostly on the fact of its versatility, as seen in recent works from trying to predict the environmental risk of a possible ship accident [Koromila et al. 2014], to modeling crime scenarios [Vlek et al. 2013] and even an analysis of driver's behavioral tendency under different emotional states [Liu and Wang 2020]. BNs also offer low information cost - even with a low number of updates, good results can be expected. As the network tries to predict what will happen taken into account the past, one can then emulate different "backgrounds" with a simple tweak of the values of the parameters - mirroring the observations of specific events.

Relying on these factors, BNs were applied to the emotional side of in-game characters as current studies lack this approach.

The work from Gomes et. al. [Gomes et al. 2013] was followed, as a way to measure the impact of the model.

3 MODEL

The model approach is centered on six main concepts – Stimuli, the Network, Confidence, Mood, Magnitude, Emotional Output. Excluding "Confidence", these are arranged as in Figure 1.

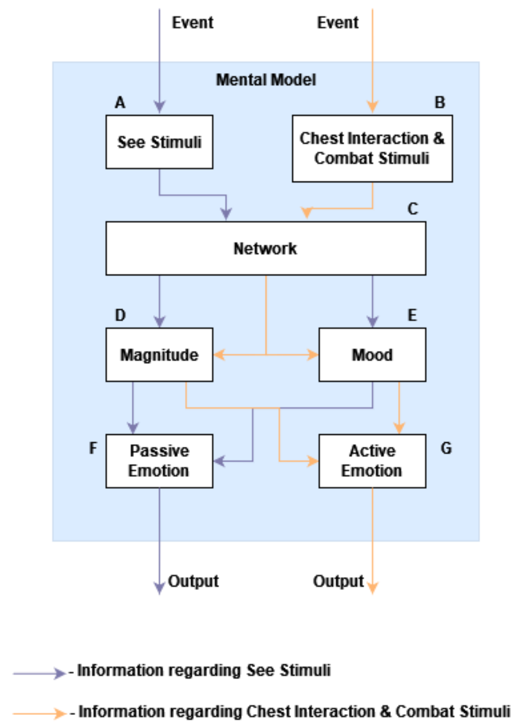

Fig. 1. An overview of the Mental Model.

The character formulates stimuli from the game's environment and outputs either a Passive or Active Emotion containing the Mood, the Magnitude of the emotion and the stimulus.

3.1 Stimuli

Stimuli are interpretations made in the character's head of things that happen around it. They 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 (explained later). There are See Stimuli - when a character sees a chest or an enemy - Chest Interaction Stimuli - when it opens a chest - and Combat Stimuli - when it receives or gives damage to an enemy or when it dies by or kills an enemy.

The character's internal state is where See Stimuli are maintained. At every frame, the character looks for surrounding elements with which it is aware that engaging will cause it to modify its internal state, such as chests that have yet to be opened or alive foes. Other forms of stimuli, in addition to See Stimuli, are retained in the character's internal state. These stimuli are likewise saved, but this time with the intention of being processed as a whole. For example, if a character sees an enemy, and then engages in a battle, the enemy attacks will be used to formulate stimuli and these will be stored in order to compute the anticipated damage the character expected versus the actual damage received.

3.2 Network

In order to have some *Decision Making* process, the character needs to store the information about everything it knows inside its memory and as this model will play with uncertainty, the character needs to form anticipation to the outcome of a certain action. This was done using a BN. To implement it, one chose to integrate the Bayes Server Application Programming Interface (API)[1] . Among many things, it offers *Online Learning* which enables the user or API developer to update the distributions in a BN each record at a time.

The conceptual network that supports this work can be as complex and interconnected as one sees fit. One opted for a first implementation using only simpler concepts to test the impact which the network can have at this point - building a baseline for future work. The implementation can be seen in Figure 2.

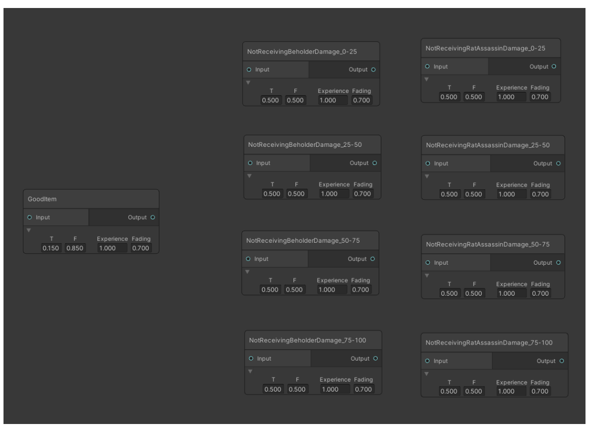

Fig. 2. Implementation of the Network using Unity's Graph View API[2]

As such, it is possible to open the network as an attribute of the character and change the values at will. The network and its values are saved as a file and, at the start of the game, a copy to not override the original. The two APIs stored data differently and a translation was done from that which is stored in a file (Unity's Graph View), to the data read by *Bayes Server* in the beginning of the game (for the network to update throughout it) and the other way around every time the network is updated (in order to see a change in the values).

"Good Item" displays how likely the next treasure contains a good item, i.e. a better sword. The damage expected to receive from a character is split into 4 equally valued parts (25% each). 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.

[1]https://www.bayesserver.com/
[2] https://docs.unity3d.com/ScriptReference/Experimental.GraphView.GraphView.html

It can also be seen Experience and Fading values. This is because Online Learning of discrete nodes requires the use of Experience tables and optionally Fading tables. Experience is the degree of prior knowledge about the associated probabilities is reflected in the value. Fading means the significance of prior information diminishes over time and more importance is put to recent information.

3.3 Confidence

As one is dealing with probability values, these are not certain. So, **confidence** is a value which represents how confident the character is that an outcome will happen. As these are an evaluation of the network, the closer a node's probability gets to 1, the more confident the character will become. This means confidence values range from 0 to 0.5. For simple nodes with no parents, it can be calculated easily by subtracting 0.5 from the probable value. For child nodes the value is calculated differently. 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 child node and dividing by the total number of nodes. So the calculation is as follows:

$$C_{ChildNode=True} = \frac{C_{Parent1} + C_{Parent2} + ... + C_{ChildNode}}{N_{Parents} + 1} \quad (1)$$

Where C means *Confidence*. This calculation was devised in order for the character to chose which event to react to. This allows for the character to focus on the events that it is more confident about.

3.4 Mood

Considering a window of the past events of the character, if the majority of these are positive ones, then the character's mood leans towards **happiness**, otherwise it will lean towards **sadness** (using a continuous scale from -1 to 1). It can be seen as a spectrum (from "sad" to "happy").

An overall assessment of what the character knows is done: a weighted average of the overall confidence of the character is done. The confidence values are added when the outcome is good - for example, when the chance of getting a good item from opening a chest -, subtracted otherwise and the result is divided by the sum of all weights. When considering its health condition, the character takes its current health percentage and then subtracts the possible damage percentage it knows an enemy is likely to give. The confidence value is added to the mood if this result is greater than 0% and subtracted otherwise. When in-combat, it takes the enemy's current health percentage and uses it as a progress of the battle - when 100% then the battle as only begun, 0% means the battle is 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 reacts differently to stronger versions of enemies. This is done inducing a bias towards these which is defined a priori. 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 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. The confidence value is added or subtracted the same way as before.

This weighted average only takes into consideration what the character knows about its environment until that moment.

3.5 Magnitude

In this work extrinsic factors were taken into account, such as a feeling of surprise when something unexpected happen. As such, magnitude, can be seen as both the intensity of a certain emotion or its unexpectedness: events which happen often lack novelty and thus are less impactful - the character gets used to it.

This is calculated taking the character's memories into account and contrasting it with what happened in the world. This is done subtracting the confidence of the before and after state of a certain event. The difference will reflect how unexpected the event was. Confidence values range from 0 to 0.5 but magnitude ranges from 0 to 1 so the output is doubled.

When the character is killed or kills an enemy, the confidence is not the only information used - it also takes into account the expected damaged from an enemy, as this can be different at the end of the battle from what it was in the beginning. So, the end result is the absolute value of subtracting the damage of the before and after state. As there are 4 different nodes for damage from an enemy and result must be between 0 and 1 then the character subtracts the damage expected(%) from the damage received(%).

When only seeing an enemy or chest, the magnitude is the confidence multiplied by 2, as this is only an assessment of the character's memories and its expectations.

3.6 Emotional Output

The character emits a **Passive Emotion** when seeing enemies or chests (which do not change its internal state and are outputted at every frame) but also emits an **Active Emotion** when killing or getting killed by an enemy, when it attacks or is attacked by an enemy and when it opens a chest (which changes its internal state). The character will not output a **Passive Emotion** when 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).

When calculating a **Passive Emotion**, the character sees (within a specified range) enemies or chests which it will react to. Then it searches for the path in the network (from leaf to root node - either killing an enemy or opening a chest) which holds a higher confidence value plus a default confidence margin. If it is deciding between chests which to react to, the closest is chosen. Within that default confidence margin, values are treated as if they were the same. The final value for it was 0.1 as nodes with the same set of updates would differ around that value.

This will be outputted and can be labelled as: **Relief** - when *Magnitude = 0, Mood = 1*; **Distress** - when *Magnitude = 0, Mood = -1*; **Hope** - when *Magnitude = 1, Mood = 1* and **Fear** - when *Magnitude = 1, Mood = -1*.

When calculating a **Active Emotion**, it is possible to understand, for example, when the character is injured from the enemy through Unity Events. The character formulates a stimuli based on what happened and stores this stimuli until it gets killed or kills that enemy - only then it knows the total damage percentage the enemy inflicted. When this happens, the character sums all the damage received by that enemy for later comparison and outputting the emotion. When the character is killed by an enemy, it stores this stimuli as being given 100% of damage. After the damage is summed, the stimuli are discarded.

The state of the network is stored, the network is updated and the after state is compared to the before stare. The data from the network is copied to Unity's Graph View and an **Active Emotion** is outputted.

4 EVALUATION

4.1 Game Description

The game consists on a main character who sees its friends fainting at a table and ventures into the wilderness to seek vengeance, killing enemies along the way and opening treasures for loot.

The game developed starts with the controls being given to the player. It is possible to control the character, making it move in every direction, jump, attack (with a sword) or defend (with a shield). There are checkpoints in the game where the character returns after being killed. There are two types of enemies: Rat Assassins (which have a dagger) and Beholders (that hover above the ground) - both with a normal and stronger version. The character can interact with levers which allows it to access different rooms of the game. The enemies chase the character as they see it.

Fig. 3. The second room the player encounters.

4.2 Implementation

Firstly, the game world which the testing would be done in was created. After making a moving character and a camera that would follow it, 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

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.

the character would be able to traverse and finally have a playable game.

Next, the integration with the Bayes Server API was done following the API documentation.

The visual depiction using Unity's Graph View API was coded from scratch to better understand what was the architecture one was going for. The two APIs were connected allowing them to copy values from one to the other - the visual API was able to add nodes, change their values and choose which nodes connected to one another before running the game and upon running the game, this information would be sent to the Bayes Server API in order for the online learning to be done.

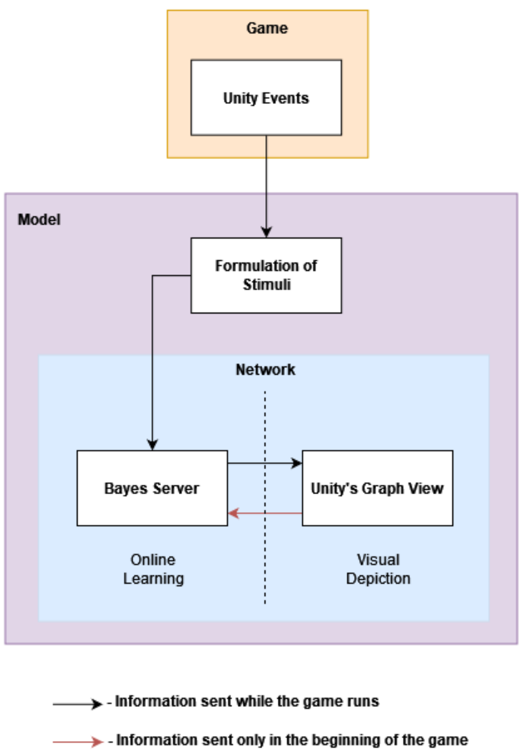

Fig. 4. The architecture and flow of information between the game through Unity Events, the Bayes Server API and Unity's Graph View API.

As a way for the the game to broadcast events which in turn updated the network, Unity Events were used. This was done as one was 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. The values from the network needed to be decoded 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 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.

4.3 Test Methodology

The tests were divided into 2 parts: the pilot tests (with a focus group) and the final tests. The pilot tests were created to validate if everything was in order to then be able to progress to the final tests. The control group was allowed to ask questions to the developers (as these were present in an internet call) and for the final tests were done only by sharing the same questionnaire's link which contained the game build but were not accompanied by the developers.

Users were questioned about their gaming preferences and demographics (to help with the characterisation of the sample). 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.

Testers played 3 game versions. One which all affective processing was either pre-scripted or reactive (version 1, reactive model), one which contained the model (version 2, non-reactive model). To test both models, 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:

- 1 - It is easy to understand what the character is thinking or feeling.
- 2 - The world around the character influenced the character behaviour.

- 3 - The character behaved in a predictable and coherent manner.
- 4 - It is easy to understand what the character thinks will happen next.

These measured **Behavior Understandability**, **Change With Experience**, **Predictability**, **Behaviour Coherence** and also the model's ability to predict what would happen in the future and if this was apparent to the players.

They had 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) and 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, users had to upload a zip file which contained files with information retrieved from their playing in order to perform a "manipulation check" - (if what the users reported was what actually happened during their playthrough).

In the control group an outlier was discovered which could not detect some character's emotions. After deliberation, two more people were introduced to the control group and did not conduct the same unexpected results. The testing continued to the next step.

5 RESULTS

Using IBM's SPSS to analyze the testing outcomes, a total of 31 individuals answered the questionnaire. 4 users were removed from the study (leaving 27 in total) as these could not clearly discern positive expressions from negative ones, and thus their understanding of the game could be biased. Additionally, ordinal demographic data was transformed to its numeric counterpart.

Of the total 27 testers, 24 identified themselves as male, 2 as female and 1 as Non-binary. 20 of which were 22 or 23 years old. The average test user schedules time to play video games (20 players, 74.07%) and enjoys the adventure game genre (23 players, 85.19%). Additionally not a single user plays video games very rarely and is not familiar with these games. 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. 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. All of the reported valences were according to what the character was feeling being it more positive when showing **Relief** (Mean = 3.52, Std. Dev. = 0.7) and **Hope** (Mean = 3.93, Std. Dev. = 0.73) or more negative with **Distress** (Mean = 1.67, Std. Dev. = 0.62) and **Fear** (Mean = 1.7, Std. Dev. = 0.823).

5.1 The Developed Model vs the Reactive Model

One looked at how each of the Likert variables differs from one model to the other. Likert variables are ordinal in nature, 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.

Shapiro-Wilk Tests were done to test for normality. Only the fourth question in both versions did approximate from a normal distribution ($p_{V1} = 0.099 > 0.05, p_{V2} = 0.062 > 0.05$). One applied a Dependent T-test to both version yet there was no statistically significant improvement in the understanding what the character thinks will happen next ($p = 0.589 > 0.05$). A Non-parametric analysis was done to every other question using a Wilcoxon Signed-Rank Test yielding: **Q1** = ($p = 0.740 > 0.05$), **Q2** = ($p = 0.059 > 0.05$), **Q3** = ($p = 0.349 > 0.05$). These suggest that there is no statistical difference in: understanding what character was thinking or feeling, the world influence effect in the character's behaviour and the character's predictability and behavior coherence between the two models.

It was also considered to determine whether there was any correlation between the models and the sample's demographics. As other demographics were very polar, the player's control choice was the sub-group chosen to be analysed. The correlation between every demographics sub-group was run and there no statistical significant correlation between any of them, using Kendall's tau-b ($p > 0.05$). As there were no correlation, a Mann-Whitney U-test was done to understand if the control scheme used by the players influenced the character believability.

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 Developed: ($U = 38.5$, $p = 0.013$, $\mu = 4.89$, $Median = 5.0$)) 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 Developed: ($U = 43.5$, $p = 0.026$, $\mu = 4.56$, $Median = 5.0$)). The medians show no difference and the means increase in both cases. These results show that when using a gamepad to play, testers felt is was easier to understand what the character was thinking or feeling and also what the character thinks will happen next than when they played with a mouse and keyboard setup.

5.2 Result Validation

The gathered results were analysed one by one to understand if what the users reported was what actually happened in the game. However, there were many cases that these did not match. For example, there was a bias towards reporting more negative emotions (like fear) than what actually happened. One suspected it was because the animations were more apparent for this emotion.

5.3 Discussion

The results obtained were checked in order to guarantee that the participants could understand no less than the valence of the emotion (if it was negative, positive or something in between). However, the finer expressions of the emotions provided might not have been clear in some contexts presented and the players could have been 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.

6 CONCLUSIONS

Because game characters' emotions are often written, it was hypothesized that utilizing a Bayesian Network to forecast what may happen in a gaming setting would allow for more convincing emotional expression. The tests were carried out using an adventure game created from the ground up that used a model to predict what would happen to a game character against a reactive model.

Although there was no evidence that this model can outperform a reactive model, there was evidence of the importance of control hardware in improving a character's believability, both in terms of understanding what it is thinking or feeling and in terms of understanding what it expects to happen next. In some of the circumstances supplied, the finer expressions of the emotions provided may not have been evident and so the result carried could differ if other types of emotion expressions are chosen.

For future work, 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 multi-player 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.

REFERENCES

Joseph Bates. 1994. The Role of Emotion in Believable Agents. *Commun. ACM* 37, 7 (July 1994), 122–125. https://doi.org/10.1145/176789.176803

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

João Dias and Ana Paiva. 2005. Feeling and Reasoning: A Computational Model for Emotional Characters. In *Progress in Artificial Intelligence*, Carlos Bento, Amílcar Cardoso, and Gaël Dias (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 127–140.

Mirjam Palosaari Eladhari and Michael Sellers. 2008. Good Moods: Outlook, Affect and Mood in Dynemotion and the Mind Module. In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share* (Toronto, Ontario, Canada) *(Future Play '08)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/1496984.1496986

Susanne Erk, Birgit Abler, and Henrik Walter. 2006. Cognitive modulation of emotion anticipation. *The European journal of neuroscience* 24 (09 2006), 1227–36. https://doi.org/10.1111/j.1460-9568.2006.04976.x

Paulo Gomes, Ana Paiva, Carlos Martinho, and Arnav Jhala. 2013. Metrics for Character Believability in Interactive Narrative. In *Interactive Storytelling*, Hartmut Koenitz, Tonguc Ibrahim Sezen, Gabriele Ferri, Mads Haahr, Digdem Sezen, and Güven Çatak (Eds.). Springer International Publishing, Cham, 223–228.

Paul R. Kleinginna and Anne M. Kleinginna. 1981. A categorized list of emotion definitions, with suggestions for a consensual definition. *Motivation and Emotion* 5, 4 (01 Dec 1981), 345–379. https://doi.org/10.1007/BF00992553

Ioanna Koromila, Zoe Nivolianitou, and Theodoros Giannakopoulos. 2014. 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* (Rhodes, Greece) *(PETRA '14)*. Association for Computing Machinery, New York, NY, USA, Article 44, 5 pages. https://doi.org/10.1145/2674396.2674463

John Lasseter. 1987. Principles of Traditional Animation Applied to 3D Computer Animation. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. Association for Computing Machinery, New York, NY, USA, 35–44. https://doi.org/10.1145/37401.37407

Benny Ping-Han Lee, Edward Chao-Chun Kao, and Von-Wun Soo. 2006. Feeling Ambivalent: A Model of Mixed Emotions for Virtual Agents. In *Intelligent Virtual Agents*, Jonathan Gratch, Michael Young, Ruth Aylett, Daniel Ballin, and Patrick Olivier (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 329–342.

Yaqi Liu and Xiaoyuan Wang. 2020. The analysis of driver's behavioral tendency under different emotional states based on a Bayesian Network. *IEEE Transactions on Affective Computing* 1, 01 (2020), 1–1. https://doi.org/10.1109/TAFFC.2020.3027720

Aaron Bryan Loyall. 1997. *Believable Agents: Building Interactive Personalities.* Ph. D. Dissertation. Carnegie Mellon University, USA. AAI9813841.

C. Martinho. 09-2007. Emotivector: Mecanismo Afectivo e Anticipatório para Personagens Sintéticas. Ph.D. thesis. *Instituto Superior Técnico* 01, 1 (09-2007).

Michael Mateas. 1999. *An Oz-Centric Review of Interactive Drama and Believable Agents.* Springer Berlin Heidelberg, Berlin, Heidelberg, 297–328. https://doi.org/10.1007/3-540-48317-9_12

K. Murphy. 1998. A brief introduction to graphical models and bayesian networks. In *http://www.cs.berkeley.edu/ murphyk/Bayes/bayes.html.* UC Berkeley, Berkeley, USA, 1–19.

Magalie Ochs, Nicolas Sabouret, and Vincent Corruble. 2010. Simulation of the Dynamics of Nonplayer Characters' Emotions and Social Relations in Games. *Computational Intelligence and AI in Games, IEEE Transactions on* 1 (01 2010), 281 – 297. https://doi.org/10.1109/TCIAIG.2009.2036247

Eugénio Oliveira and Luís Sarmento. 2003. Emotional Advantage for Adaptability and Autonomy. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems* (Melbourne, Australia) *(AAMAS '03)*. Association for Computing Machinery, New York, NY, USA, 305–312. https://doi.org/10.1145/860575.860625

Andrew Ortony. 2003. *Emotions in Humans and Artifacts, chapter On making believable emotional agents believable.* MIT Press, Massachusetts USA.

Pedro Miguel Ferreira Pimentel. 2016. Gameplay aware emotional model for virtual characters in video games. *Master's thesis, Instituto Superior Técnico, Portugal* 01, 1 (2016).

Klaus R. Scherer, Tanja Banziger, and Etienne Roesch. 2010. *A Blueprint for Affective Computing: A Sourcebook and Manual* (1st ed.). Oxford University Press, Inc., USA.

David Spiegelhalter. 2002. Bayesian graphical modelling: A case-study in monitoring health outcomes. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 47 (01 2002), 115 – 133. https://doi.org/10.1111/1467-9876.00101

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

Charlotte Vlek, Henry Prakken, Silja Renooij, and Bart Verheij. 2013. Modeling Crime Scenarios in a Bayesian Network. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law* (Rome, Italy) *(ICAIL '13)*. Association for Computing Machinery, New York, NY, USA, 150–159. https://doi.org/10.1145/2514601.2514618