Predictive Models and Monte Carlo Tree Search: A Pipeline for Believable Agents

1st Cristiana Pacheco
Queen Mary University of London
London, UK
c.pacheco@qmul.ac.uk

2nd Diego Perez-Liebana
Queen Mary University of London
London, UK
diego.perez@qmul.ac.uk

***Abstract*—Developing and assessing believable agents remains a sought out challenge. Recently, research has approached this problem by treating and assessing believability as a time-continuous phenomenon, learning from collected data to predict believability of games and game states. Our study will build on this work: by integrating this believability model with a game agent to affect its behaviour. In this short paper, we first describe our methodology and then the results obtained from our user study, which suggests that this methodology can help creating more believable agents, opening the possibility of integrating this type of models into game development. We also discuss the limitations of this approach, possible variants to tackle these, and ideas for future work to extend this preliminary work.**

***Index Terms*—mcts, human-like, believable models**

I. Introduction

Believability can be a core element for games. Not only it can improve the player's immersion in a game but also their enjoyment and sense of challenge [1]. Previous research has argued this to be due to humans being less predictable and more capable of inspiring emotion [2], [3]. This suggests that having believable Non-Player Characters (NPC) in games could produce higher levels of fun and engagement. Many techniques attempt to build believable agents, by either imitating players or by hard-coding specific behaviours [4]. All provide different advantages—such as convenience or predictability—and disadvantages—the lack of adaptability, repetitiveness, and others [2]. There is also a diverse range of applied techniques to evaluate agent believability [1], [5]. Most define believability in terms of behaviour only, representing it as an overall phenomenon despite its complexity [3]. To address this, a new set of studies [3], [6] attempted to see believability as a time-continuous phenomenon. In these, the authors acknowledge that believability is a fuzzy concept, similar to affective states. As a result, affective computing has focused on modelling these concepts in a time-continuous fashion with ad-hoc tools [7]. These new methods have demonstrated the importance of the context as a predictor of character believability, with models achieving high accuracy at predicting if the opponent in a given state is believable or not.

The contribution of the study presented in this paper is a first step building on those studies. We present an initial approach that biases a Monte Carlo Tree Search (MCTS) agent with a believability model. This model is based on a time-continuous assessment of agent believability in a given game. This allows us to train a believability model to then integrate it in MCTS. We build on previous studies [3] that focused on human-like assessment and believability modeling, to develop an NPC based on this model.

II. Related Work on Believability

In game AI research, one of the ways the term 'believable' can be approached is as character believability [5]. This involves an agent, known to the observer for being computer-controlled, that behaves in a believable way to a person. In this paper, this is the definition we use. The development of human-like behaviour has spawned a wide and diverse range of attempts [4]. Popular options are the use of simpler techniques (such as Finite State Machines and Behaviour Trees), or more complex AI models such as Imitation Learning (IL) [4]. The former are often chosen due to its simplicity of implementation, while mimicking human play is a sensible approach. However, these models have been criticised for their lack of expressiveness, complexity scaling and evolution [2], [4], or for not being able to adapt well to new scenarios or modeling expected behaviour only partially [2]. This limits the applicability of these approaches to game production scenarios. Previous research has suggested that very simple AI methods might not achieve a high degree of believability [2], and their non-deterministic nature makes obtaining human-like behaviour a challenge. Due to this, and the fact that MCTS is inherently a reactive method, as it continuously makes new plans for every search [8], our study proposes its use for believable decision making. With regards to evaluating human-like behaviour, previous studies suggest evaluations that often exist as a *byproduct* of a developed agent. A major drawback of this approach is that it does not define a concrete protocol whose research focuses first on the evaluation itself [1], [3], directly correlating the believability of the character with the AI that controls it [2]. This has been shown not to be the case: factors like protocol variables (players' experience, camera perspective; [1]) or altering the game's environment

979-8-3503-2277-4/23/$31.00 ©2023 Crown

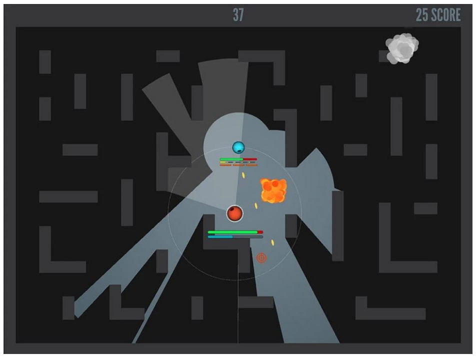

Fig. 1. Screenshot of MAZING from [10]. Player is blue, opponent is red. Fields-of-vision and auditory systems are showcased in grey. Bullets (yellow) and bombs (orange) are also present.

(number of enemies and their placement; [9]) affect the agent's perceived believability.

Studies often choose to represent believability as a high-level concept attributed to an agent for its entire gameplay session [5]. Research has attempted to challenge this by providing a low-level technique instead [3], [6], where the assessment is done moment-to-moment and data is collected throughout the session. The authors show which specific gameplay features are correlated with believable behaviour, modelling continuous believability predictions [3], and establishing the correlation between continuous and discrete methods [6]. These are the techniques and models that form the base of our study.

III. Background: MAZING, PAGAN and MCTS

This section provides an overview on the game (MAZING), the tool used to annotate game sessions for the believability model (PAGAN) and the decision making algorithm that will form the basis of our proposed final approach (MCTS).

A. The MAZING game

MAZING is a 2-player top-down shooter game that takes place in a maze. The player's goal is to defeat his opponent by shooting at it or throwing bombs. The opponent is a computer-controlled agent which tries to chase and catch the player. On collision, the player loses and the game resets. This is not a symmetrical game: opponent and player have different abilities and properties. For example, the enemy moves faster but it does not have any weapons or abilities. However, it has sensory systems—field-of-vision and auditory system—to detect the player. When it does not know where the player is, it moves randomly through the maze. When it knows where the player is, it chases it following the shortest path. The game can be seen in Figure 1 and full details can be found in [10].

B. PAGAN and Data Processing

We collect annotations through the *Platform for Audiovisual General-purpose ANnotation* tool (PAGAN) [7]. This is an online tool developed to collect affect annotations. It allows users to perform different types of annotation, out of which we use two: `RankTrace`, which is based on ordinal affect annotation (users indicate how believable a model is in an unbounded range); and `BTrace`, which is a binary labelling method that allows users to indicate whether a moment is believable or not [7]. In our previous work, the MAZING game was integrated with PAGAN so participants could play it online and annotate their session for every moment [3], [6]. Participants were requested to annotate the opponent's believability for two sessions. They were either assigned `BTrace` or `RankTrace` as their tool. The obtained data consists of moment-to-moment assessments for a total of 55 features which include player, agent and gameplay related telemetry [10]. More details can be seen in [3], [10]. The data is aligned and resampled by discretising it to a 3 second interval with 1 second lag, in order to account for the annotators' reaction time [10].

C. Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) [8] is a highly selective best first search method that balances exploration and exploitation of different actions in game states. The algorithm does this by building an asymmetric game tree, which grows towards the most promising parts of the state space. MCTS is an iterative algorithm, and each iteration performs four steps: *selection*, *expansion*, *simulation* and *backpropagation*. Details of this popular algorithm can be found here [8].

On every iteration, one game state (found at the end of the *simulation* step) is evaluated to indicate how much of an advantage position it is for the player. This could be either a win/loss factor (1.0 or 0.0, respectively), or a `state evaluation function` that provides a continuous value for the state (with values closer to 1.0 considered better for the agent, and viceversa).

IV. Method

This section details the method employed. We show how the believability model is trained (Section IV-A) with Random Forests, allowing us to determine if a given state rates high or low in believability. We then describe how this trained model is integrated in MCTS (Section IV-B). Section V describes how we evaluate the effect of the trained model in the believability of the agent.

A. Training a Believability Model

We build upon previous work which performed the annotation of the data, by collecting moment-to-moment believability annotations on *MAZING* [3]. The available dataset [3] consists of 800 entries for `BTrace` and 741 for `RankTrace`. Each entry represents a state with a total of 55 features, including *opponent's behaviour*, *general gameplay* features and *player's behaviour*. Gameplay logs and subsequent believability annotations are treated in an ordinal fashion. We trained new models using this data, using preference learning with random forests yet again [11]. By training a machine learning model on these states and respective human annotations, it effectively allows us to build a believability model that classifies any given state in *MAZING* as believable or not. Upon providing a state, the model retrieves a 'believability score' for that state.

The choice of learning algorithm is based on previous evidence that shows that this technique is more robust than simple classification approaches [3], [11]. We used preference learning based on a *pairwise transformation*. The model then provides a relative relationship between data points, resulting in a binary classification: high or low believability. In this pairwise transformation every pair of data points $(x_i, x_j) \in X$ and corresponding label $(y_i, y_j) \in Y$ creates two new data points and assigns them new labels based on preference. For example, if $y_i >_\epsilon y_j$ (x_i is preferred to x_j), it creates $x' = x_i - x_j$ and $x'' = x_j - x_i$ and assigns $\lambda' = 1$ and $\lambda'' = -1$ as labels to them. This transformation is applied in sequence given the time-continuous nature of the dataset.

The binary classification of ranks is done with random forests (RF).[1] RFs are a type of machine learning algorithm where decision trees are randomly initialised and the output is the *mode* of the trees' predictions. These have been used before to model human data given their ease to train and robustness in other applications [11]. Only two parameters are changed and tested: the number of trees—ranging from 64 to 100—and the number of leaves—from 10 to 25. They are used for both `BTrace` and `RankTrace`; and per fold—training on all participants' data except for one, and testing on that one participant's data for each participant. Our best model was achieved with `BTrace` on a threshold of 0.25, with 73.3% accuracy on average and a 97.4% accuracy on the best fold. Thus we integrated this one with MCTS. To our knowledge, this is the first time a believability model has been developed with the purpose of applying it to alter the decision making of an algorithm.

B. Agents: Biased and Unbiased MCTS

We use two different agents: biased and unbiased MCTS, to play as opponent. The difference between these two agents resides on the `state evaluation function`. Both agents share other parameters, such as the iteration budget (400), the exploration constant (C=2) and the use of macro-actions [12] of size 5. Both MCTS agents use a state evaluation function, which we define as Φ_U and Φ_B for the Unbiased and Biased MCTS agents respectively. We first define a common term, $v(s)$, which encapsulates the fitness of a state s for the agent. It is defined as the linear combination of three features ($d(s)$: distance from the agent to the player, $h(s)$: agent's own health; $c(s)$: proportion of collisions with walls occurred per time step), normalised to a $[0, 1]$ range, and their respective weights with values determined empirically ($w_d = 0.8, w_h = 0.1, w_c = 0.1$) to obtain a competitive agent. The Unbiased MCTS function is defined as $\Phi_U(s) = v(s) + I(s) \times K$. $I(s)$ is an identity function, equal to 1 if the agent won the game in state s, and 0 otherwise. K is a high positive number (set to 10^6), thus this term is used to reward wining states. Given this state evaluation function, the complete decision making of the Unbiased-MCTS agent is solely focused on choosing actions to win the game with no knowledge of believability. In contrast, Biased-MCTS, uses another factor in the state evaluation function: $\Phi_m(s)$, which is given by the `BTrace` model. The Biased MCTS function is defined as $\Phi_B(s) = v(s) \times \alpha + \Phi_m(s) \times \beta + I(s) \times K$. $\Phi_m(s)$ is returned by the believability model and provides a probability for how confident it is that the state s is believable. We set $\alpha = \beta = 0.5$ to give equal weight to both parts of the evaluation. Thus, the decision making process of Biased-MCTS results from a combination of the fitness of the state s, which leads to winning the game, and the behavioural model evaluation, leading to believable game states.

V. Experimental Study

A. Design and Setup

The objective of this preliminary study is to evaluate if there is a clear preference between our two agents in terms of believability. The study was explained to the participants, with consent and data collected through a survey. A total of 39 individuals participated in this study, which took place in person and on a laptop. Due to the importance of having experienced players as judges [1], [5], having experience with video games was a requirement. Each participant played the game twice for 1 minute: once against Unbiased MCTS and once against Biased MCTS. The order in which they are played is randomised. When the participants finish, they're given both videos to watch in the order they played before the exit survey. The use of videos follows the findings in [6], to reduce the chances of challenging participants' memory. Participants are then asked which video they prefer in terms of their opponent's believability (with believability defined as "your opponent is playing like a human would in the given situation"). The options were: *First video*, *Second video*, *Both* and *Neither*. This is followed by the opportunity to comment on the opponents that were faced—this was an open ended question which allowed for any type of information to be given.

B. MCTS vs Biased MCTS

The results show the preference for each agent in terms of believability. The majority (19) of our participants chose the biased MCTS as more believable than the other agent. 'Neither', 'Both' and 'Unbiased MCTS' where chosen 6, 5 and 9 times, respectively. These results suggest a preference for the believability-biased MCTS. A Chi-Square for Goodness of Fit Test (χ^2 value 12.59 and p-value 0.0056) rejects the null hypothesis that the participants would not be able to distinguish between the two agents, showing the effects of including the believabity model in biased MCTS.

We suspect the biased MCTS, given its believability model, provides a wider range of behaviour as half of its goal is to be believable rather than just winning. After all, previous research [3] was showing a higher correlation with believability when there was diversity of interaction. This seems to also be the case when investigating the reasons behind participants' choices. The majority (17) of the judges that picked the biased MCTS mention 'dodging bullets and bombs' as reasons

[1] We used Microsoft's NimbusML's *FastForestBinaryTrainer*, v1.7.1.

for their pick. In contrast, MCTS was observed to chase the player 'at all costs' 12 times in total. The remaining choices came down to a mixture of the previous reasons followed by the perception of those 'still not being enough' to distinguish them. This maintains previous research conclusions that believability is not straightforward and related to the observers expectations [2].

VI. Discussion and Limitations

This is, to our knowledge, the first study to explore a design where the assessment of believability came first, its modeling being trained on annotated data, to then be used to influence an agent's decision making. Our results support the idea that biasing an agent with a believability model could potentially output a more believable character.

Despite this contribution there were obstacles encountered. The lack of existing libraries and available content for integrating a model in Unity/C# complicated the engineering extensively. A drawback of choosing MCTS for the agent is that it needs a forward model (i.e. a simulation of the real game to build the search tree), which for existing games in unity often (as it was the case) means reimplementing most of the game. MCTS also requires searching for an action in real-time, creating multiple copies of the game state, which impacts both time and memory during execution. This, added to the use of an external library to use the believability model, created a slower game where FPS would drop to 10 at times. Some participants highlighted this as a downside, as it can potentially influence play-style, enjoyment and even opponent behaviour throughout the game. Whether this problem with FPS affected the results of the study or not, it is hard to tell. It is also worth observing that, while the majority of the participants were able to identify as most believable agent the one with the believability model, 50% of the them selected one of the other three options (none, both, unbiased MCTS). This clearly shows that there is room for improvement and more convincing agents can be achieved. Despite these issues, we are confident that the study carried out for this paper shows the potential of the proposed approach, and opens a line of research for this kind of models that aim to achieve believable agents using models trained with continuous annotations.

VII. Conclusions and Future Work

This preliminary study explores how to use character believability modeling from time-continuous assessment to bias an algorithm's behaviour. We first train a elievability model using previously retrieved time-continuous believability assessment data. Then, we integrate this model with a decision-making algorithm, Monte Carlo Tree Search (MCTS), to bias action decisions to favour not only winning the game but also rewarding reaching believable states according to our trained model. Finally, we run a user study with 39 participants, who played against our believable MCTS agent and a baseline MCTS with no bias. Participants were asked their preference in terms of the bot's human-likeness and their opinions on their choices. The analysis of our data indicates a preference over the agent with a believability model, in support of the approach presented in this paper.

While this study shows that it is feasible to combine the MCTS algorithm with believability models to generate more believable agents, several lines of future work are available that could expand this work and address some of its limitations as described in this paper. For instance, it would be interesting to explore different weights for the state evaluation function described in the paper, and whether these weights should remain constant during the execution of the game. We could also investigate if these weights could depend on the history of game states; given that our results suggest that believability can be treated as a continuous factor. Finally, it would also be interesting to extend the analysis to include measures on the behaviour of the agents, how they differ (e.g. disparity in the decision making, or their dodging ability), and how this correlates to the comments and choices of the participants of our user study. Our work also opens doors to various new improvements and techniques, such as using other (model-free) Reinforcement Learning algorithms (which, once trained, have a lower memory and real-time consumption profile) or ML models for predicting believability other than Random Forests. Finally, it would be interesting to study if the proposed model is also able to create believable agents in other types of games, such as tabletop or turn-based games, where decisions are taken in a more discrete manner.

References

[1] C. Pacheco, L. Tokarchuk, and D. Pérez-Liébana, "Studying believability assessment in racing games," in *Proc. of the 13th international Conf. on the foundations of digital games*, 2018, pp. 1–10.

[2] F. Tencé, C. Buche, P. De Loor, and O. Marc, "The challenge of believability in video games: Definitions, agents models and imitation learning," *arXiv preprint arXiv:1009.0451*, 2010.

[3] C. Pacheco, D. Melhart, A. Liapis, G. N. Yannakakis, and D. Perez-Liebana, "Trace it like you believe it: Time-continuous believability prediction," in *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2021, pp. 1–8.

[4] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*. Springer, 2018, vol. 2.

[5] J. Togelius, G. N. Yannakakis, S. Karakovskiy, and N. Shaker, "Assessing believability," in *Believable bots*. Springer, 2013, pp. 215–230.

[6] C. Pacheco, D. Melhart *et al.*, "Discrete versus ordinal time-continuous believability assessment," in *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE, 2021, pp. 1–7.

[7] D. Melhart, A. Liapis, and G. N. Yannakakis, "Pagan: Video affect annotation made easy," in *Proc. of the Intl. Conf. on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2019, pp. 130–136.

[8] C. B. Browne *et al.*, "A Survey of Monte Carlo Tree Search Methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[9] E. Camilleri, G. N. Yannakakis, and A. Dingli, "Platformer level design for player believability," in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 2016, pp. 1–8.

[10] D. Melhart, G. N. Yannakakis, and A. Liapis, "I feel i feel you: A theory of mind experiment in games," *KI-Künstliche Intelligenz*, vol. 34, no. 1, pp. 45–55, 2020.

[11] G. N. Yannakakis, R. Cowie, and C. Busso, "The Ordinal Nature of Emotions: An Emerging Approach," *IEEE Trans. on Affective Computing*, 2018.

[12] D. Perez, E. J. Powley, D. Whitehouse, P. Rohlfshagen, S. Samothrakis, P. I. Cowling, and S. M. Lucas, "Solving the physical traveling salesman problem: Tree search and macro actions," *IEEE Transactions on CI and AI in Games*, vol. 6, no. 1, pp. 31–45, 2013.