Machine Learning Methods for Emulating Personality Traits in a Gamified Environment

Georgios Liapis
Aristotle University of Thessaloniki
GR
gliapisa@gapps.auth.gr

Anna Vordou
Aristotle University of Thessaloniki
GR
vordou.ann@gmail.com

Ioannis Vlahavas
Aristotle University of Thessaloniki
GR
vlahavas@csd.auth.gr

Abstract

Personality traits are regarded as a significant factor of competency for job candidates, for example, evaluating the capacity to work efficiently within a team. However, there is a gap in the traditional assessment system for these cases since they typically rely on self-answered questionnaires that are biased or easily exploitable. Artificial Intelligence techniques can fill this gap by generating objective data to define standard personality template profiles, utilizing trained Reinforcement Learning agents. In this paper, we propose a gamified framework that employs Machine Learning methods to emulate personality traits based on the players' play styles, with the purpose of creating standard team profiles. The OCEAN Five personality model is used as a basis for this attempt, which characterizes personality as a synthesis of the five components: openness, conscientiousness, extraversion, agreeableness, and neuroticism. After generating gameplay data through self-play, we examine how various personality qualities, actions, and modes of communication impact the team performance of the agents, with respect to the different personality traits. Results indicate that the personality traits of the agents individually and as a team do impact their performance and efficiency. This can be used as a methodology for creating efficient individual bot agents or teams of agents in many game environments.

CCS Concepts

• **Computing methodologies → Multi-agent reinforcement learning**.

Keywords

Machine Learning, OCEAN 5, Gamified Environment

ACM Reference Format:
Georgios Liapis, Anna Vordou, and Ioannis Vlahavas. 2024. Machine Learning Methods for Emulating Personality Traits in a Gamified Environment. In *13th Conference on Artificial Intelligence (SETN 2024), September 11–13, 2024, Piraeus, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3688671.3688757

SETN 2024, September 11–13, 2024, Piraeus, Greece

ACM ISBN 979-8-4007-0982-1/24/09
https://doi.org/10.1145/3688671.3688757

1 Introduction

As technology advances, new opportunities are created to accomplish new milestones in various scientific domains. Gaming has changed and improved in many ways and can be combined and used in other sectors. Human resource management and education, for example, utilize game-based approaches [7] to enhance and expand their effectiveness by providing new experiences, opening up new chances for learning, and analyzing their employees. One such example of gaming is Escape Rooms (ER), either real-life or virtual, which is one fast-expanding game type that employs similar gamification tactics and collaboration [16]. The purpose of an ER is for a player or team to solve physical and mental puzzles and riddles within a certain time limit to leave the room. Companies employ ER for team bonding and assessment of individual players and the team as a whole since strong collaboration and excellent communication are critical for a team's success.

There have been attempts in real-life ER to quantify team effectiveness and performance, but the existing methods and tools are mostly restricted to a questionnaire the players fill out after the room is completed. This causes issues with evaluation because the measurements are skewed and dependent on each player's biased view of the game [9]. Even if a person is assigned to record certain data, it is hard to maintain track of each player's movement, activities, and conversation throughout the full ER game.

The development of an ER game that gathers and analyzes the play style of each player and the team as a whole can be a new, engaging, and effective solution to handle such difficulties. However, there is a need for numerous gameplay data with various types of player behaviors so that analysis of the results can reveal the features that are unique to each team. To do this, we developed Deep Reinforcement Learning agents that generate data using self-play while emulating personality traits based on the OCEAN Five personality trait model.

More specifically, we propose a multi-agent system, that solves an Escape Room game, while each agent shows specific behaviors and emulates a distinct trait based on their custom reward function using Deep Reinforcement Learning (RL). This allows us to compare each player's personality attributes based on their gameplay style. As a result, a team profile is generated that includes information about a range of measurements and attributes as well as their effectiveness, based on the combinations of the personalities of the agents.

This paper presents a multi-agent system in a 3D digital Escape Room as a simulation of team effectiveness based on the behaviors and personalities of each team member regarding their team traits, like Agreeableness and Extraversion. More specifically, we propose custom reward functions so that each agent acts and emulates behaviors based on the specific Extraversion and Agreeableness personality traits.

We focus on these specific traits because their behaviors and rewards are directly influenced by how the agents cooperate. This includes aspects such as communication, where agents share information or signals, and physical interactions like pushing, which affect how they navigate shared spaces. In contrast, traits like Openness primarily focus on an agent's interactions with the environment itself, such as exploring new areas or engaging with objects independently. By concentrating on traits that emphasize inter-agent cooperation, we can better understand and develop systems where teamwork and collaborative strategies are essential. This distinction allows us to create more dynamic and interactive multi-agent systems, laying the groundwork for more complex and varied behaviors in future research.

The contribution is:

- A multi-agent system that measures team efficiency based on personality
- A reward mechanism for emulating human behaviors in a dynamic, gamified environment

In this work, section 2 examines the main topics of our work and section 3 analyzes relevant work on personality trait implementations and multi-agent systems. The approach of our study and the results of the agent's training are provided in section 4, and in section 5 some inferences are reached as a result.

2 Background

In this section, we examine the two main components of our system: the personality characteristics model and the Reinforcement Learning (RL) theory.

2.1 Personality model

We have incorporated the OCEAN Five personality characteristics model and the acronym stands for Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism, and it is one of the most well-known models in the field [11].

Personality evaluation is an issue with various aspects such as temperament, emotion, and mentality; thus, the actual number is debatable. This has resulted in significant studies by various scientists that analyze the features upon which the OCEAN Five personality model is based. Following OCEAN-Five, various changes and new models were produced, like the Psychopathic Personality Inventory (PI) [21]. These new models are based on the OCEAN Five model, with differences in the subcategories of each feature. However, we utilize the Ocean Five model since it is more widely used, widely recognized, and generally.

2.2 Reinforcement Learning

Reinforcement Learning is a sub-field of Machine Learning that entails teaching an agent to learn through contact with an unknown environment. Generally, the agent learns to choose the best action to maximize a reward function in a particular circumstance by weighing the opinions of other actions [19]. As a result, the agent uses input based on prior experience to optimize its reward and therefore learns from its actions and experiences. Each following state and, hence, all future rewards can be affected by the actions selected in addition to the immediate reward [20].

An RL model is defined as a Markov Decision Process (MDP) and is represented as a 5-tuple $M = (S, A, p, \gamma, R)$, where S is the state space, A is the action space, and p is the environment dynamics function, γ is a discount factor, and R is the reward function [2]. In our agent system, we do not define the dynamics function or use the discount factor, while the observation and action space as well as the rewards are analyzed in section 4.

It must also be noted, that multiple autonomous, cooperating agents known as "multi-agent systems" share a common environment, which they observe with the help of sensors and respond to using actuators. Although the agents in a multi-agent system can be equipped with pre-designed behaviors, they frequently need to pick up new behaviors and actions online, which leads to a gradual improvement in the performance of the individual agent or of the multi-agent system as a whole [4].

2.3 HiDAC Simulations

For our work, we used the High-Density Autonomous Crowds (HiDAC) [8], a high-density crowd simulation system that handles the simulation of behaviors as well as their path-finding in a museum as a baseline for our rewards.

More specifically, after completing each episode, each agent is awarded based on the associated characteristic it emulates, depending on the trait on which they are being trained. So, they are rewarded based on how the HiDAC defines an agent's behavior. The personality of an agent pi is a 5-dimensional vector, with each dimension represented by a personality component, Psi_i. A Gaussian distribution function Nu with mean mui and standard deviation $sigma_i$ is used to simulate the distribution of personality variables in a group of individuals:

$$\pi = \langle \Psi^O, \Psi^C, \Psi^E, \Psi^A, \Psi^N \rangle \tag{1}$$

$$\begin{gathered} \Psi^i = (\mu_i, \sigma_i^2) \text{ for } i \in \{\text{O, C, E, A, N}\} \\ where\ \mu \in [0, 1], \sigma \in [-0.1, 0.1] \end{gathered} \tag{2}$$

The overall behavior β of an individual agent is a function of the different behaviors that it shows in the game and is defined as:

$$\beta = (\beta_1, \beta_2, ..., \beta_n)\,,\ \text{where } \beta_j = f\,(\pi)\,,\ \text{for j} = 1, ..., \text{n} \tag{3}$$

because each attribute varies, $Psii$ can have both positive and negative values.

Furthermore, a specific behavior may also exist in more than one personality dimension, depending on the positive/negative influence. For example, impatience can only be identified as a negative quality if a person is non-agreeable or low conscientious [12], leadership affects high-extroverted and conscientious people [10] and panic affects neurotic and low-conscientious individuals [5].

3 Related Work

Modeling personality features is a difficult issue, and past work has relied on simulation and statistical approaches [8]. The HiDAC simulation system delivers individual distinctions by assigning specific psychological and physiological traits to each participant. There is also evidence that conversational agents may be built to display personality through nonverbal behaviors like body movement and facial expressions, as well as verbal behaviors like dialogue selection and voice change [18]. These works demonstrate that modeling behavior in simulated settings or certain scenarios is possible and is used as a base for our models, however, in our work, we integrate these qualities in a dynamic gaming environment. More specifically, the main elements of the escape room (like the buttons or the keys) are changing in every new episode. This allows us to generate more variable gameplay data and create a more generic standard profile for the participant.

In addition, a multi-agent system based on decision trees and data from common surveys was developed as a model for diagnosing or identifying personality types [17]. It is used to identify a person's dominant personality type, to position someone in the proper job, or to pinpoint a person's reactions depending on their specific traits. The primary distinction between this approach and our work is that it collects data using conventional self-administered questionnaires, which does nothing to lessen the bias of the results.

Furthermore, there has been research on creating adaptive agents in serious game environments. In this work, a multi-agent system was built and integrated into the SIMFOR project, a serious game regarding crisis management [15]. The agents are built based on the Belief Desire Intention (BDI) deliberation model and include editing options to aid in a crisis scenario construction. As a result, the designer may set the agent's actions, as depicted in a crisis management scenario case study. Though this approach shows sufficient generality to be applied in other serious games training situations, our multi-agent system is built to set a standard methodology for NPC creation and not contribute to general crisis management scenario construction.

Moreover, there have been previous attempts at using a game environment as a simulation for modeling behaviors [14], which analyzes how a single agent can emulate simple in-game behavior related to its movement based on its openness personality trait, while it navigates in a simple room. The work of [13] also showcases how an escape room can theoretically measure specific gameplay data and metrics regarding a player's personality using specific puzzles and riddles. Our work implements a multi-agent system to set a reward function methodology for teams, measuring efficiency, in a gamified environment while exhibiting complex behaviors based on multiple personality traits.

4 Methodology

The gaming mechanics are presented in this section. We also establish the parameters for evaluating personality traits and explain how Deep RL agents work.

4.1 Environment

We implemented the agents and the multi-agent system in Unity and used the example assets to build our 3D Environment. We created a simplified Escape Room environment, where the agents have to find a button, that will reveal a key in order to unlock the final door. We must also note, that our environment is dynamic, meaning that the buttons, the keys, and the door are not always in the same position, but spawn randomly on each episode while

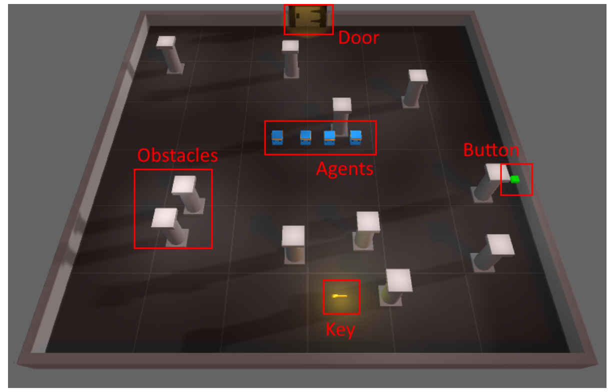

Figure 1: Environment

there are obstacles inside the room that are randomly generated each time, as we can see in Figure 1.

The agents at the beginning were trained to learn to solve the Escape Room game first by identifying the buttons and keys required for level completion and then we trained the agents to emulate the Extraversion and Agreeableness traits and the corresponding behaviors since these two are related to a team of people and their relationships. We do not include the other traits because it is out of the scope of this multi-agent system study.

4.2 Action Space

The agent's actions are discrete and selected during gaming, whereas its rewards are predetermined based on the trait the agent emulates. The agent's action space consists of several distinct capabilities. For movement, the agent can move forward and backward, with each movement action represented as a Boolean variable, allowing the agent to either initiate or halt the movement. For rotation, the agent can rotate to the left and right, with each rotation action also being a Boolean variable that determines whether the agent is currently rotating in a given direction. Additionally, the agents can adjust their speed between two modes: normal and fast, enabling them to navigate the environment more efficiently depending on the situation. They also have the ability to perform a gesture action by pointing out locations of objects of interest within the room, which helps in communicating with other agents or highlighting key areas. Furthermore, the agents can push each other as a result of their movement, which occurs naturally when one agent's movement causes physical contact with another agent and is not a discrete action. By combining these actions, agents can navigate and interact with their environment and each other effectively, facilitating complex behaviors and cooperative tasks.

It must be noted, that we also monitor specific information for each agent's gameplay style, like their movement, if they interact with each other (push each other), and with the environment, like if they pick up keys or press buttons.

4.3 State Space

We use ray casts for the observations, which is a physics function that projects a ray into the environment scene and returns a Boolean result if a target is successfully intersected. Ray-cast observations

(RayPerceptionSensor) are utilized by agents to receive observations from the environment in the Unity implementation. Before making a judgment using the observation vector, the agent class invokes this method.

The agent is observing the surroundings and gathering information about objects with specific tags, including, other agents and/or NPCs (Non Playable Characters), walls, doors, buttons, and keys. The agent projects one (1) ray in front of it and seven (7) more in each direction. This allows it to have complete knowledge of its surroundings.

Furthermore, each agent takes into account information from the environment, such as vector observation, for buttons, keys, and doors and their current state, e.g. whether they are pressed, found, or unlocked. For example, when an agent picks up the key, all the other agents get this information too through their observations.

So, the final state space consists of the ray-cast arrays, which project a total of fifteen (15) rays that each check the 5 tags analyzed above and the three (3) vectors from the environment state, with the final size of our state being seventy-eight (78).

The observation values are ray-cast Boolean for each tag and Boolean for each vector. As a result, the agent gains knowledge from its sensors and its environment. This allows it to gain awareness of its environment and surroundings. Throughout the training, rewards were assigned based on interactions with each other, as explained in the next section.

4.4 Rewards

A common difficulty with many MDPs is that rewards are often relatively sparse, providing non-zero rewards to the agent for only a few states and making the learning procedure difficult. This phenomenon occurs in an Escape Room setting as well, because the most significant rewarding events occur only after all the agents escape, making it hard for each one to grasp the chain of events that led to it accurately.

For our work, the rewards are broken into two parts. The first part is related to the ER game and how the agents' team learns to solve it, and the second one is the custom reward functions for the agents to behave in specific ways. These two kinds of rewards are monitored separately since the first one is the reward for the whole team and shows if the team manages to solve the escape room and escape. The second reward is given to each agent and is related to the behaviors it exhibited, and is discussed in the next paragraphs.

More specifically for the ER game, we implemented proper reward functions for each one of the agents, when they reach specific checkpoints in the progress of the room, like picking up a key, and a final team reward when all the agents escape.

We choose these rewards so that the agents will learn and have the ultimate goal of escaping, not just finding the key or opening the door. So we crafted the proposed rewards as a + 0.4 reward for each agent when they see a key or unlock a door a +1 if they manage to escape, and the team reward as +100 if all the agents escape the room. In the future, a more sophisticated reward-shaping approach will be used.

For the next phase of our implementation, we have formulated mathematical equations based on the HiDAC system's metrics. We utilized their systems architecture and have made changes to their way of interpreting behaviors for our system. Based on the formulas of Table 1, we created new reward functions, by combining different kinds of behavior by setting the corresponding Ψ during the training. For example, we get the mean speed of the agent or how many times it collided and pushed other agents and we calculate the corresponding behavior and reward them depending on the Ψ we have set.

Table 1: Traits to rewards, based on actions and characteristics

Personality Trait	Behaviors (Original)	Reward (custom)
Extraversion	Leadership	0.3 * mean speed * Ψ^E
	Communication	1 if num of communication actions used >= Ψ^E >= 0.5
	Impatience	0.3 * 2 * $\Psi^E - 1$ if Ψ^E >0
	Pushing	1 if num of push actions used >= 0.3 *Ψ^E >= 0.5
	Personal Space	-
	Walk speed	Max walk speed + 1
	Gesture	Num of correct gestures * 10
Agreeableness	Impatience	0.3 * $(1 - \Psi^A)$ if run each step
	Pushing	1 if num of push actions used >= 0.3 * (1- Ψ^A) >= 0.5
	Right Preference	0.3 *(Times right/time) * Ψ^A
	Wait Radius	-
	Wait Timer	-

The rewards are given at the end of each episode, based on how it interacted with the environment. We are able to know every interaction between the agent and the environment by keeping track of the tags of each item it interacts with by utilizing colliders.

Since each agent is trained while showing specific behaviors, we expect the rewards and data to fluctuate since the behavior can be unstable or create tension between the team players, especially if they are trained in opposite traits and show different behaviors. For instance, consider two agents, one extrovert and one introvert. Since their rewards are based on speed, each will move faster than the other.

4.5 Training Methodology

As we stated in the previous section, we created a simple multi-agent system in that each agent tries to achieve specific tasks, such as pressing buttons and opening doors and the goal is for all the agents to escape.

Then, each agent in the team is trained on specific personality traits, alongside specific behaviors, based on the previous reward model, while each one tries to achieve the same tasks as before, and the team's objective is the escape of all the agents.

We implemented the MA-POCA (MultiAgent POsthumous Credit Assignment) algorithm, from the ML-agents package [3], which learns a centralized value function to estimate the group of agents' expected discounted return and a centralized agent-centric counterfactual baseline to achieve credit assignment [6].

4.6 Training Results

In this subsection, the results of the agents' training are shown, alongside discussion and conclusions.

To train the agents that emulated human behavior, we started by training a simple multi-agent system to solve the ER, so that we could set the training hyperparameters. In the following figures, the X-axis shows the steps of the training, and the Y-axis the rewards or the time for each episode.

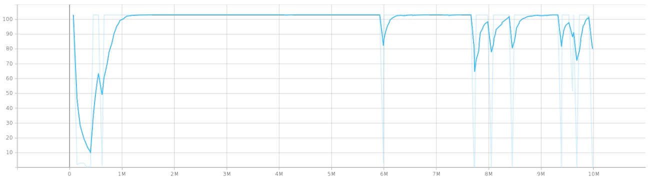

Figure 2: Default multi-agent team

Table 2: Hyperparemets and best values

Hyperparameters	Values
batch size	128, 254 , 512, 1024
buffer size	64000, 128000, 256000, 10240
learning rate	0.005
hidden units	256, 512
number layers	2

The results of the best default (without emulating behaviors) multi-agent team are shown in Figure 2, and as we can observe it learns to solve the room most of the time, while sometimes some agents did not escape as we can observe from the drops of the rewards. This is related to the dynamically changing environment.

Regarding the hyperparameters, we tuned with the batch size, the buffer size, the learning rate, the hidden units, and the number of layers.

The final parameters that were tested are the ones shown in Table 2, with the underlined ones being the best.

All the agent teams were trained for 10 million steps, which took around 8 hours each for each team, and for their assessment we looked into the group rewards, each episode length, and in some cases their behavior metrics.

Following that, we started to implement the reward functions regarding the behaviors and train agent teams with different kinds of behaviors and traits. For better results, each agent had only one personality trait. The teams consisted of 4 agents and since the possible combinations with four types of personality (Extrovert, Introvert, Agreeable, and Non-Agreeable) were numerous, we present some of the more standard.

In the first four teams we trained, all the teams consisted of agents that showed only one of the above personalities. This way we could set a base of how agents with the same personality traits would perform.

As we can see in Figure 3, both the team with only extrovert agents (orange) and introverts (pink) learn to solve the room quite fast but they have drops of rewards (meaning that not all agents managed to escape). This can happen because sometimes they show impatient behavior and do not cooperate efficiently.

Regarding their mean episode length, we must note that the introverts (pink) take more time to finish the room, since based on the bibliography they tend to move slower, as shown in Figure 4.

In Figure 5, the team with only agreeable (red) agents takes some time to learn to solve the room but they have some drops of rewards, while, the non-agreeable agents (cyan), were a bit less efficient but

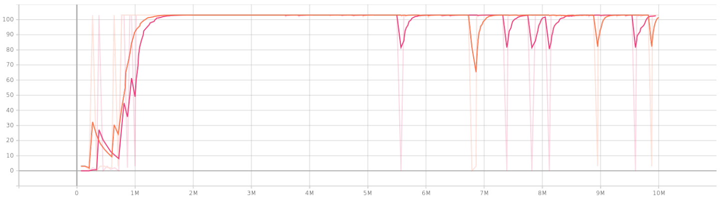

Figure 3: Extrovert (orange) and Introverts (pink) agents team rewards

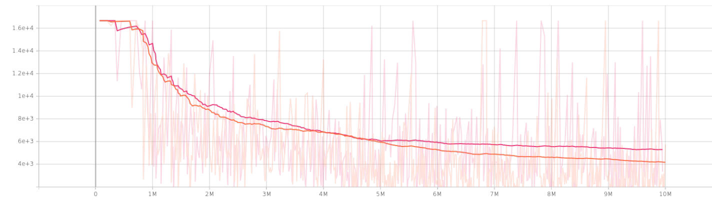

Figure 4: Extrovert (orange) and Introvert (pink) episode length

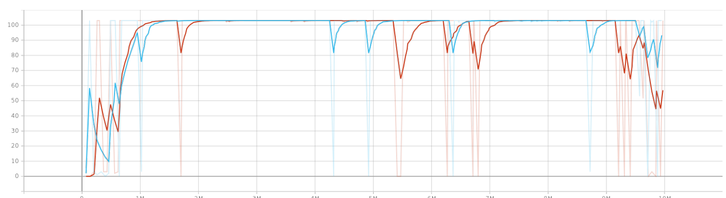

Figure 5: Agreeable (red) and Non-Agreeable (cyan) agents team rewards

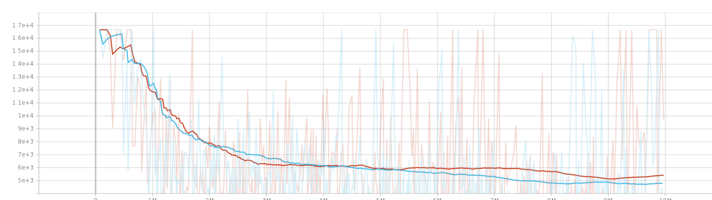

Figure 6: Agreeable (red) and Non-Agreeable (cyan) episode length

learned to solve the room a bit earlier but show some fluctuation of the results, related to their impatience.

We must also highlight that both agreeable (red) and non-agreeable agents (cyan) teams need almost the same time to solve the room but with a fluctuation after some time, as depicted in Figure 6. This seems to happen since their behaviors are not correlated with their speed but with how patient they are.

The next step is to train agents, that each one has a different kind of personality trait. We allocated the initial number of agents based on the 25 percent ratio of introverts to extroverts in a community [1].

The results show that even though the team with 3 extroverts and 1 introvert (green) cooperates greatly at first and is very stable,

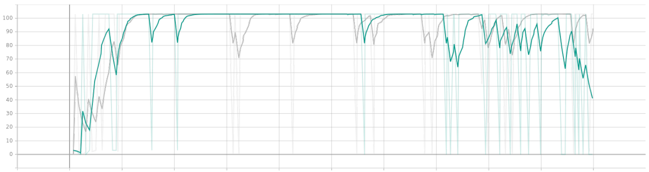

Figure 7: 3 Extrovert and 1 Introvert (green) and 3 Introverts and 1 Extrovert (grey) agents team rewards

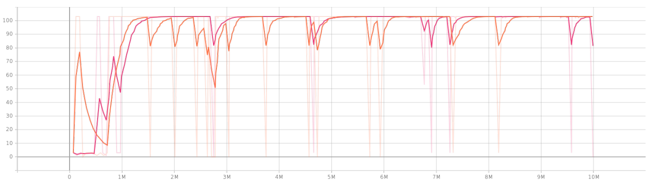

Figure 8: 3 Agreeable and 1 Non-Agreeable (pink) and 3 Non-Agreeable and 1 Agreeable (orange) agents team rewards

in the end, the team has stopped collaborating, as seen in Figure 7. This happened because the extroverts always rushed to push each other and get the rewards, while the introverts were more patient and did not engage with the other players. On the other hand, in a reverse personality traits team of 3 introverts and 1 extrovert (grey), the results are almost identical to the only introvert agent's team but slightly less efficient. This means that 1 extrovert does have a negative impact on 3 introverts but less than 1 introvert in 3 extroverts. This happens because, if the extroverts escape, they may wait for the introverted one, slowing down the team, while on the other hand, the extrovert will help the slow introverted team to finish more quickly and efficiently.

Following the same pattern, we created a team of 3 agreeable and 1 non-agreeable agent and the reverse.

As we can see in Figure 8, the 3 agreeable and 1 non-agreeable team (pink) starts with a drop in the rewards, and then it learns to cooperate but tends to be quite inefficient in the end. The agreeable agents, being patient and not pushing each other, contributed to efficient gameplay.

Respectively, the team with 3 Non-agreeable agents and 1 agreeable (blue), is closer to the one with only Non-agreeable agents, but their rewards fluctuate throughout the training.

The mean episode length of these four teams shows how they are compared to each other in Figure 9. It must be noted, that the team with the most extroverts in the end takes more time to finish since the introvert holds them back with its slow movement.

As we said before, personality traits and behaviors are interconnected and so the behaviors of each agent show are related to the personality trait they have been trained on. One of the main characteristic behaviors of the extraversion and agreeableness trait is how the agent pushes the others inside the room.

In Figure 10, we can observe that the team with more Non-agreeable and the one with more extrovert agents tend to push each other more than the other teams. This was expected since they show unstable behavior and tend to try to reach the goal first.

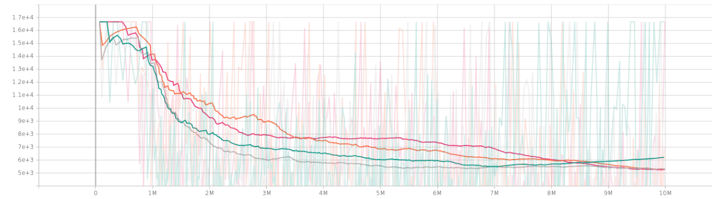

Figure 9: 3 Extrovert and 1 Introvert (green) and 3 Introverts and 1 Extrovert (grey), 3 Agreeable and 1 Non-Agreeable (pink) and 3 Non-Agreeable and 1 Agreeable (orange) team episode lengths

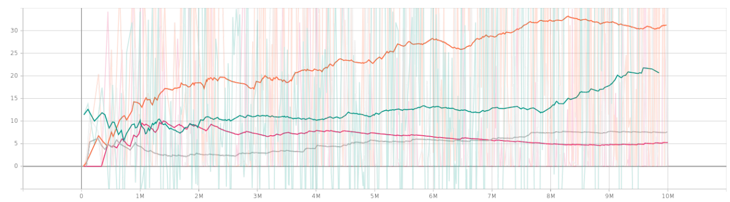

Figure 10: 3 Extrovert and 1 Introvert (green) and 3 Introverts and 1 Extrovert (grey), 3 Agreeable and 1 Non-Agreeable (pink) and 3 Non-Agreeable and 1 Agreeable (orange) pushing actions

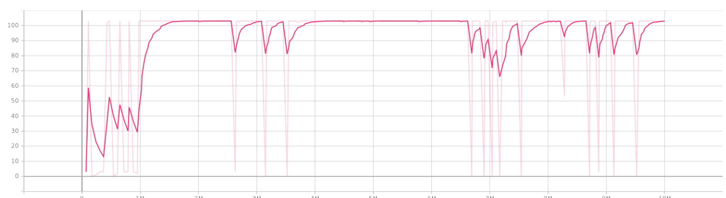

Figure 11: 1 Extrovert, 1 Introvert, 1 Agreeable and 1 Non-Agreeable agents team rewards

Finally, we created a team with agents of all personality traits, one extrovert, one introvert, one agreeable, and one non-agreeable. As we can observe in Figure 11, this combination shows the biggest drops in rewards, meaning that the agents did not manage to solve the room effectively. This is something expected since the agents show contrary behaviors and they cannot collaborate.

Based on the results, it is safe to say that the personality traits of the agents of each team do play a significant role in how the group operates, how efficiently it operates, and how quickly they manage to escape. This means that the reward functions and how they were set, can indeed replicate and emulate how the traits and behaviors are exhibited in real life based on logic.

More specifically, the agreeable agents in general, are the most efficient and fast in cooperating, and the introverts are almost the same effective but a lot slower in solving the rooms. Furthermore, when the team was constructed with agents of all the traits, the team had some difficulty collaborating, especially in the early stages.

So there is a basis for stating that team efficiency in the gamified environment, where the agents have to collaborate to finish a task, is related to the behaviors and traits each team member has. The results also show that the specific metrics and rewards we have set can indeed translate to behavior changes in the agents and help set a new reward methodology of creating agents with similar behaviors and set standards for other types of games as well.

Finally, we can say that the personality the agents have can play a great role in how the team works, how efficient it is, and how fast it can solve specific tasks.

5 Conclusions and Future Work

In this paper, we presented a multi-agent environment in the form of an escape room, with agents that emulate the OCEAN Five Personality Traits characteristics based on custom reward functions. The environment is dynamically changing and was developed in such a way that allows us to generate data regarding the play style of the agents, and their interactions with the room and between them.

The emulation of human behavior in a gaming environment is a complex endeavor since personality is a complex matter, making it difficult to quantify without a large number of distinct human players with specific traits and behaviors.

In our work, we used as a template of our agent's behaviors and personality traits the OCEAN Five model, which is one of the most accepted and has been modeled by the HiDAC simulation. More specifically, we implemented the extroversion and agreeable traits in our agents, since these two regard teams of people and not individuals.

So, based on custom reward functions, we developed multi-agent teams that tried to solve a gamified Escape Room environment while exhibiting specific behaviors and emulating a predefined personality trait. By doing so, we collected enough data so that we could compare the efficiency and speed of each team to the respective traits their team members had.

This led to the conclusion that, by utilizing RL agents and custom reward functions, human behaviors can be emulated by the agents. The training of multi-agent systems with specific hyperparameters and algorithms that reward both the team and the agents can indeed create teams with a variety of behaviors.

This leads to the conclusion, not only do the custom reward functions for the behaviors and traits emulate the equivalent in real life but also show that they have a major impact on the team and its efficiency. By conducting personality assessments of a company's employees, we can apply this knowledge in real-life scenarios to create highly effective teams. Using the results from our simulations of various personality traits, we can strategically assemble teams that optimize productivity. This approach not only aids the HR department in identifying the ideal personality traits for new candidates but also in making informed decisions about internal placements, ultimately enhancing overall organizational efficiency.

Lastly, this work can serve as a foundation for creating agents and agent teams capable of learning to solve tasks in various game environments using the same principles and reward structures. This allows for the development of NPCs (Non-Playable Characters) that can emulate diverse personalities and behaviors. Additionally, our goal is to develop more complex agents with an expanded set of actions and more sophisticated reward systems, enabling us to evaluate their performance across different types of game environments. A limitation of this multi-agent study is that we did not include other traits or explore more combinations of traits within a team, as it falls outside the current scope. Additionally, the simplicity of the reward system presents another drawback. However, addressing these aspects will be a focus of future work to enhance this methodology.

References

[1] [n. d.]. Collaborative narration of the past and extraversion - ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S0092656605001054
[2] [n. d.]. Theory and Practice of Distance Education. https://www.routledge.com/Theory-and-Practice-of-Distance-Education/Holmberg/p/book/9780415112925
[3] 2021. Unity ML-Agents Toolkit.
[4] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. 2010. Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1* (2010), 183–221.
[5] Tomas Chamorro-Premuzic and Adrian Furnham. 2008. Personality, intelligence and approaches to learning as predictors of academic performance. *Personality and Individual Differences* 44, 7 (May 2008), 1596–1603. https://doi.org/10.1016/j.paid.2008.01.003
[6] Andrew Cohen, Ervin Teng, Vincent-Pierre Berges, Ruo-Ping Dong, Hunter Henry, Marwan Mattar, Alexander Zook, and Sujoy Ganguly. 2022. On the Use and Misuse of Absorbing States in Multi-agent Reinforcement Learning. arXiv:2111.05992 (Jun 2022). http://arxiv.org/abs/2111.05992 arXiv:2111.05992 [cs].
[7] Darina Dicheva, Christo Dichev, Gennady Agre, and Galia Angelova. 2015. Gamification in Education: A Systematic Mapping Study. *Journal of Educational Technology & Society* 18, 3 (2015), 75–88.
[8] Funda Durupinar, Nuria Pelechano, Jan Allbeck, U. Gudukbay, and Norman Badler. 2011. How the Ocean Personality Model Affects the Perception of Crowds. *IEEE Computer Graphics and Applications* 31 (May 2011), 22–31. https://doi.org/10.1109/MCG.2009.105
[9] Panagiotis Fotaris and Theodoros Mastoras. 2019. *Escape Rooms for Learning: A Systematic Review.* https://doi.org/10.34190/GBL.19.179
[10] Robert R. Hirschfeld, Mark H. Jordan, Christopher H. Thomas, and Hubert S. Feild. 2008. Observed leadership potential of personnel in a team setting: Big five traits and proximal factors as predictors. *International Journal of Selection and Assessment* 16, 4 (2008), 385–402. https://doi.org/10.1111/j.1468-2389.2008.00443.x Place: United Kingdom Publisher: Wiley-Blackwell Publishing Ltd..
[11] K. L. Jang, W. J. Livesley, and P. A. Vernon. 1996. Heritability of the big five personality dimensions and their facets: a twin study. *Journal of Personality* 64, 3 (Sept. 1996), 577–591. https://doi.org/10.1111/j.1467-6494.1996.tb00522.x
[12] Nan Jiang, Mi Shi, Yilong Xiao, Kan Shi, and Barry Watson. 2011. *Factors Affecting Pedestrian Crossing Behaviors at Signalized Crosswalks in Urban Areas in Beijing and Singapore.* https://doi.org/10.1061/41177(415)138 Pages: 1097.
[13] Georgios Liapis, Katerina Zacharia, Kejsi Rrasa, Aikaterini Liapi, and Ioannis Vlahavas. 2022. Modelling Core Personality Traits Behaviours in a Gamified Escape Room Environment. *European Conference on Games Based Learning* 16, 11 (Sep 2022), 723–731. https://doi.org/10.34190/ecgbl.16.1.602
[14] Vlahavas I. Liapis G., Lazaridis A. 2021. Escape Room Experience for Team Building Through Gamification Using Deep Reinforcement Learning. *15th European Conference of Games Based Learning* (2021).
[15] M'hammed Oulhaci, Erwan Tranvouez, Sébastien Fournier, and Bernard Espinasse. 2013. A MultiAgent Architecture for Collaborative Serious Game applied to Crisis Management Training: Improving Adaptability of Non Played Characters. *The 7th European Conference on Games Based Learning* ECGBL 2013 (10 2013).
[16] Rui Pan, Henry Lo, and Carman Neustaedter. 2017. Collaboration, Awareness, and Communication in Real-Life Escape Rooms. 1353–1364. https://doi.org/10.1145/3064663.3064767
[17] Margarita Ramírez Ramírez, Hilda Beatriz Ramírez Moreno, Esperanza Manrique Rojas, Carlos Hurtado, and Sergio Octavio Vázquez Núñez. 2019. Multi-Agent System Model for Diagnosis of Personality Types. In *Agents and Multi-Agent Systems: Technologies and Applications 2018*, Gordan Jezic, Yun-Heh Jessica Chen-Burger, Robert J. Howlett, Lakhmi C. Jain, Ljubo Vlacic, and Roman Šperka (Eds.). Springer International Publishing, Cham, 209–214.
[18] Sinan Sonlu, Uğur Güdükbay, and Funda Durupinar. 2021. A Conversational Agent Framework with Multi-Modal Personality Expression. *ACM Trans. Graph.* 40, 1, Article 7 (jan 2021), 16 pages. https://doi.org/10.1145/3439795
[19] Michael Sutton. 2020. Gamification Success Stories (Part III): Overview of Michael Sutton's work in Game-Based Learning. (Sept. 2020).
[20] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement learning: an introduction* (second edition ed.). The MIT Press, Cambridge, Massachusetts.

[21] Katarzyna Uzieblo, Bruno Verschuere, Eva Van den Bussche, and Geert Crombez. 2010. The validity of the Psychopathic Personality Inventory-Revised in a community sample. *Assessment* 17, 3 (2010), 334–346. https://doi.org/10.1177/1073191109356544