A Survey of Application of Reinforcement Learning in Video Games

Achintya K J[2]
Department of Computer Science Engineering
B.N.M. Institute of Technology
Bengaluru, India
23cse066@bnmit.in

Krishna Karanth G.[3]
Department of Computer Science Engineering
B.N.M. Institute of Technology
Bengaluru, India
23cse088@bnmit.in

***Abstract*—Reinforcement learning (RL) has become a pivotal method in video game development, facilitating the creation of intelligent and adaptable AI systems that enhance both gameplay mechanics and player engagement. This survey delves into the application of RL in video games, concentrating on areas such as Procedural Content Generation(PCG), non-player character (NPC) behavior design, and tailored gaming experiences. By examining recent advancements and relevant case studies, this paper identifies significant trends, challenges, and opportunities for integrating RL into game design. It aims to offer valuable insights for researchers and developers, underscoring the potential of RL to drive the evolution of interactive entertainment.**

***Index Terms*—Reinforcement Learning, Video games, Non-Player Characters**

I. Introduction

In recent years, significant advancements in Artificial Intelligence (AI), particularly in Reinforcement learning (RL), have garnered considerable attention. RL, a branch of machine learning, focuses on training agents to make sequential decisions by interacting with their environment. One intriguing application of RL that has captivated researchers is its use in video games. As game technologies evolve and player expectations grow, traditional programming approaches often struggle to address the complexity and variability of modern gaming scenarios. Rule-based AI systems, in particular, tend to lack flexibility and adaptability, making them less effective in dynamic game environments and when responding to player behaviors. [1]

Reinforcement learning offers a solution by emulating human learning processes, enabling game agents to enhance their performance and adaptability through continuous interaction. RL agents learn optimal strategies by observing the game environment, selecting actions based on the current state, and receiving rewards or penalties as feedback. Over time, through trial-and-error learning, agents refine their decision-making, mastering game rules and strategies. This adaptive approach allows agents to handle both predictable scenarios and unforeseen situations, demonstrating exceptional robustness and flexibility.

The application of RL in video games has yielded impressive results. For instance, in complex maze games, RL agents can discover optimal paths while avoiding obstacles and opponents through exploration. Similarly, in competitive games, agents can analyze opponents' strategies, identify weaknesses, and improve their chances of winning. Despite these advancements, there remains a gap in comprehensive literature that systematically explores this field.

This paper aims to bridge that gap by examining the application of RL in video games. It will focus on the evolution of RL in electronic gaming, comparisons between RL-based AI and traditional game AI, real-world examples of RL in the gaming industry, and the potential future of RL in gaming. Additionally, the paper will analyze various RL algorithms employed in game-playing, highlight key achievements, and discuss their broader impact on the gaming industry.

II. What is RL?

Reinforcement learning (RL) operates as a framework for self-directed learning, grounded in the principles of Markov Decision Processes (MDP). In RL, an agent interacts with its environment and receives feedback in the form of rewards or penalties based on its actions. RL algorithms are typically classified into three main categories: Value-based methods, Policy-based methods, and Actor-Critic methods. [2]

- Value-based Methods: Algorithms like Q-learning and value iteration fall under this category. These methods use a value function to estimate the "value" of each policy and select the one that maximizes rewards. Unlike heuristic search, the value function in RL learns directly from the environment. The performance of these methods heavily relies on the accuracy of the value function.
- Policy-based Methods: This category includes algorithms such as Policy Gradient and REINFORCE . Instead of assessing the value of each policy, these methods focus on evaluating the potential of policies. This allows the agent to occasionally explore options that may initially seem suboptimal but could lead to better long-term rewards.
- Actor-Critic Methods: These algorithms combine elements of both value-based and policy-based approaches. The Actor determines the strategy, while the Critic evaluates it using value functions. Examples include Deterministic Policy Gradient and Soft Actor-Critic (SAC). This dual-approach enables the agent to optimize its policy

while maintaining a balance between exploration speed and precision in strategy refinement.

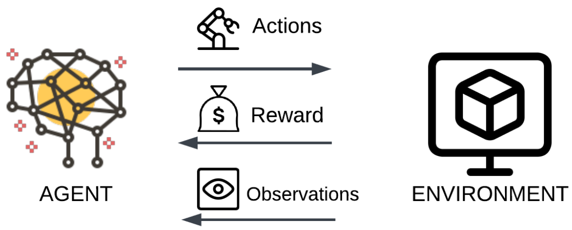

Fig. 1. Working of a RL

III. Evolution of RL in video games

In early days of game development, Game(Enemies/NPCs) AI was primarily based on systems with rules because everything was hard-coded in, and rigidity was typically the only option. Most of the time, these systems revealed themselves to be inflexible and expected not to change face-to-face with the increasingly complex features of modern games. RL completely changed the game. [1]

Deep reinforcement learning (DRL) is an important jump into AI. It allows agents to learn very complicated behavior in extremely complex environments. Consider a character in a real-time strategy game: it can manage resources, make sophisticated strategies, and even outperform human players. Such abilities are not inherent; they are learned through trial and error.

RL has also been used beyond competitive gaming. It's changing the way game worlds are constructed. PCG supported by RL enables games to tailor to each player's unique preferences. Levels can be adjusted as the player plays, so a level can become continually challenging but never impossible.

Another field of interest is multi-agent systems where different AI entities interact between each other, be it cooperatively or adversarially. Dota 2 and StarCraft II are two examples of games where RL has been shown to produce agents that cooperate in ways that feel almost human (sharing resources, strategizing together, adapting on-the-fly to opponents).

The evolution of Reinforcement Learning (RL) in the video game field marks a significant transition from basic rule-based approaches to advanced AI-based methodologies. Initial implementations were mainly focused on game-specific rule optimization of RL or were primarily limited to simple reward-based mechanisms to improve player engagement. Techniques such as Deep Q-Networks (DQN) and policy gradient approaches have allowed learning agents to develop complex strategies and demonstrate human-like behaviors in competitive games.

In 2016, a revolution was seen when DeepMind's AlphaGo could prove that RL learned complex strategies through deep neural networks. Those of RL used a wide variety of different gaming applications-now-from real-time strategy games, first-person shooting games, and simulation environments to complex multi-agent learning systems. These learning mechanisms, generally, are likely to become general-purpose and provide, or at least promise, flexible risk-adjusted responses to complex shifting interactive scenarios while allowing interchange with multiple different gaming environments to address faults of previous narrow, game-specific algorithms.

IV. Famous Algorithms in RL

A. Value-Based Methods

1) Deep Q-Networks (DQN): DQN employs deep neural networks to approximate Q-values for high-dimensional inputs, such as raw pixel data. DQN's success stems from two key innovations:

a) Experience Replay: This method improves learning by randomly sampling past experiences, reducing the impact of sequential data and improving generalization.

b) Target Network: A separate network is maintained for generating target values, significantly reducing training instability caused by moving targets.

2) Advanced DQN Variants: Several improvements to the base DQN architecture have emerged:

a)Double DQN addresses value overestimation by separating action selection from value estimation.

b) Dueling DQN introduces an innovative neural network architecture that separates state value estimation from action advantages, enabling more refined value assessment.

c) Rainbow DQN represents the culmination of multiple improvements, integrating various enhancements into a unified, high-performance framework.

3) Distributional Reinforcement Learning: Rather than learning expected values, distributional RL predicts the distribution of rewards:

a)C51 improved RL performance using a more sophisticated approach to value estimation.

b) QR-DQN advanced the field with quantile regression techniques.

c) IQN introduced implicit quantile networks, offering enhanced flexibility and superior performance over prior methods.

B. Policy Gradient Methods

1) Actor-Critic Methods: These stabilizes training by integrating a policy network (actor) with a value function network (critic).

a) A3C (Asynchronous Advantage Actor-Critic) conducts parallel training with multiple agents to accelerate learning.

b) GA3C merges A3C with GPU acceleration for improved scalability.

2) Modern Policy Optimization: a) PPO (Proximal Policy Optimization) has become a standard in the field, offering reliable performance through constrained policy updates.

b) SAC (Soft Actor-Critic) introduces entropy-regularized policies, particularly effective in continuous control tasks.

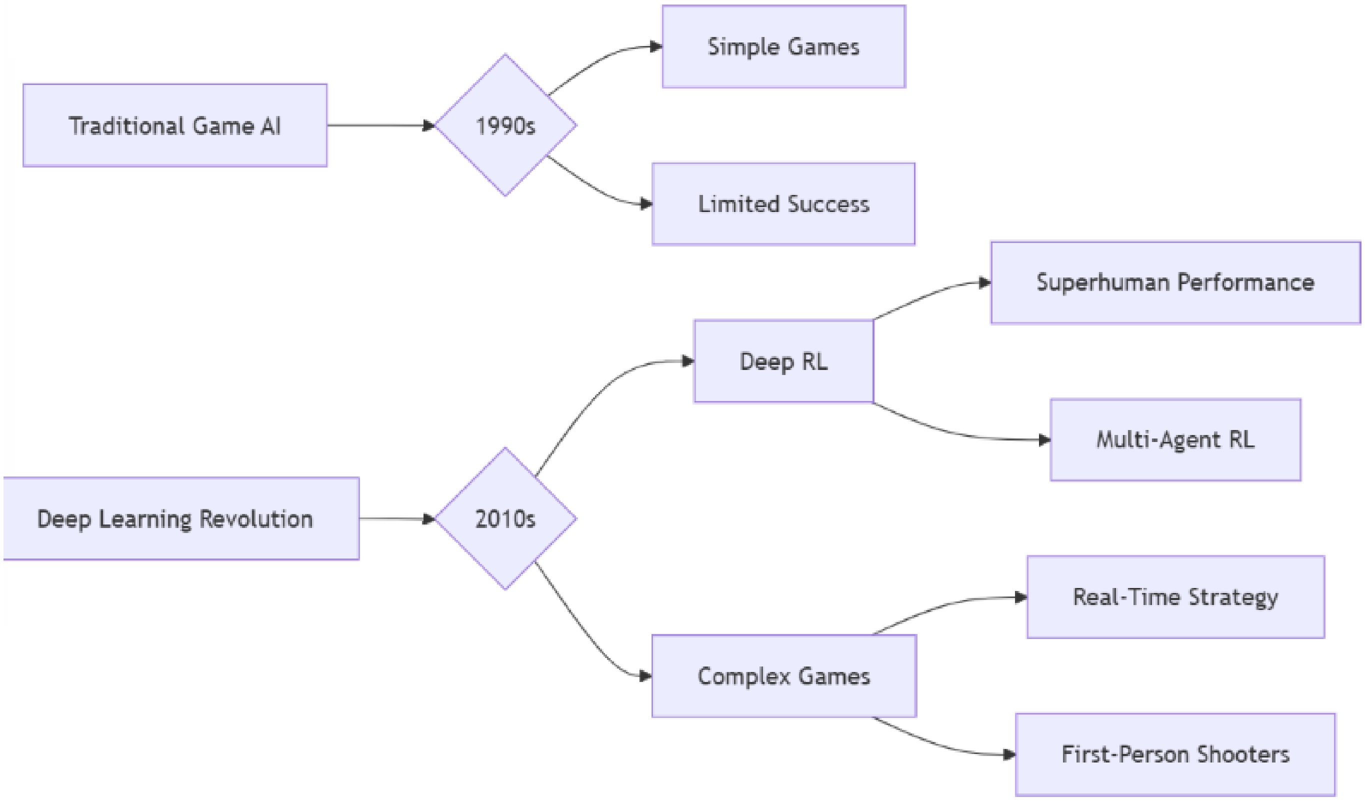

Fig. 2. Evolution of AI in game

C. Model-Based Methods

1) Tree-Structured Approaches: TreeQN and ATreeC demonstrate the power of structured planning in learned state spaces, offering improved sample efficiency over model-free approaches

2) Advanced Planning Models: 1) MuZero represents a breakthrough in model-based RL, achieving exceptional performance across diverse domains through learned dynamics models.

2) World Models showcase the potential of internal simulation for sample-efficient learning.

D. Hybrid Approaches

Recent work has focused on combining the strengths of different RL paradigms

- Reactor merges actor-critic methods with off-policy updates..
- IMPALA scales learning through distributed training and importance weighting
- PGQ synthesizes policy gradient and Q-learning approaches

V. SOME APPLICATIONS OF REINFORCEMENT LEARNING IN VIDEO GAMES

A. Deep RL: Overcoming the Limitations of Traditional NPC Navigation

The challenge of non-player character (NPC) navigation remains one of the most critical aspects of modern video game development, directly impacting how players experience and interact with game worlds. Traditionally, navigation methods like the navigation mesh (NavMesh) have been essential in game design, providing a structured way to represent traversable environments. The traditional methods for navigation often face challenges when it comes to more complex mechanics like jetpacks, grappling hooks, or teleportation. These features can make implementation tricky and limit scalability.

Recent advances in Deep Reinforcement Learning (Deep RL) offer a promising solution to these limitations. Unlike the predetermined pathfinding of NavMesh, Deep Reinforcement Learning enables NPCs to acquire navigation strategies autonomously by interacting with their surroundings. Using advanced algorithms like Soft Actor-Critic (SAC), Deep RL allows NPCs to adapt to large, intricate, and ever-changing maps. A notable real-world application of this technology appeared in the AAA title ”Hyper Scape,” where Deep RL-powered navigation achieved impressive results on maps far larger than those typically studied in academic research. The system demonstrated a 90% success rate in complex scenarios involving advanced movement mechanics, significantly outperforming traditional NavMesh implementations. [4]

The successful implementation of Deep RL navigation in ”Hyper Scape” represents a turning point in game AI development, demonstrating that research-driven approaches can effectively scale to meet industry demands. The fact that makes Deep RL stand out is how it enables NPCs to ”see” and interact with their surroundings. It uses tools like 3D maps and 2D depth visuals to give the NPCs a sense of spatial awareness, allowing them to make smarter decisions in real time.

The main advantage of Deep RL over NavMesh is that it is far less labor-intensive. Developers don’t need to manually tweak maps or add navigation links for advanced abilities. Instead, NPCs can learn on their own, adapting to changes in the environment as they occur. This speeds up development and improves the overall gaming experience, making the world more immersive and dynamic.

In summary, Deep RL revolutionizes NPC navigation by providing scalable, adaptable, and efficient solutions. As games continue to evolve with more complex movement mechanics and larger worlds, Deep RL’s ability to provide adaptive, efficient navigation solutions positions it as a potential successor to traditional methods, promising more dynamic and engaging player experiences. [4]

B. Simulated Policy Learning (SimPLe): Enhancing Efficiency in Atari Games

One notable application of the reinforcement learning field is in video games. Simulated Policy Learning (SimPLe) is such an example of reinforcement learning-based approaches wherein model-based play of Atari games is achieved using video prediction models. This is unlike other model-free approaches which usually require thousands of environment interactions before a competitive level can be glossed over. SimPLe is said to be able to learn the game from around 100,000 interactions or about 2 hours of play and this count is no doubt quite impressive in terms of sample efficiency trained with a world model predicting future frames and rewards of the game periods.

The program spans an assortment of 26 Atari games, but singles itself from the rest especially on those that require total and extensive exploration like Freeway and Pong. Stochastic video prediction models can handle the uncertainties in dynamic systems because of their new use of discrete latent variables. SimPLe further claims that model-based RL discovered under conditions as contrasted to few attempts will be best configured for data-efficient tasks, but also introduces a new frontier on the application of reinforcement learning to interactive game situations in visually rich environments. The increasing importance of the field relative to more and more demanding game problems will continue to promise lowering computational costs. [5] [9]

C. APPLICATION OF RL IN FPS GAMES

Reinforcement learning techniques embedded into first-person shooter games have shown huge improvement in the autonomous agent’s performance in very complicated, semi observable 3D environments. In contrast to traditional AI programming of game agents, RL allows learning adaptive strategy from agents by letting it directly engage with its game environment. While traditional approaches to RL have shown more promising results in 2D environments, such as Atari games, these fail in FPS settings where unique challenges exist. Examples of the challenges include navigation of very complex 3D maps, partial observability and mastering skills like locating and defeating enemies, resource collection, and spatial awareness. To tackle this, Deep Recurrent Q-Network (DRQN) architecture was used, which incorporates Long Short-Term Memory (LSTM) networks for enabling the agent to keep track of and use the history of game states over several frames.

During training, agents use high-level information, visible enemies, or items, to achieve greatly improved feature detection and faster learning. Convolutional networks captured higher levels of abstraction in representations of game scenarios using this co-training method.

The division of game sections into navigation and action, under which a method is divided and conquered, forms the backbone of this approach. Each program was trained with a dedicated network to trim down somewhat of the monumentality of the task, thus engendering more focused learning. For instance, an action network optimizes exploration and item collection, while an action network focuses on combat strategies. This modularity prevents inefficient behaviors, such as camping, and supports faster, more targeted training.

Reward shaping was the most important point in making all of this happen. They introduced intermediate rewards for actions like item pickup, health loss, and ammo usage. In addition, frame-skipping methods optimize computation resources without sacrificing agent performance. [5]

Some evaluations conducted through the platforms, such as ViZDoom, have established the viability of such methods of RL. Agents trained with enhanced DRQN models achieve high kill-to-death ratios (K/D), surpassing human capabilities in both single-player and multiplayer scenarios. This approach was proven not only capable of learning but also really adaptable because improvements were clearer for generality across diverse games.

Trough everything done in gaming through RL, not only do we get closer to how AI can revolutionize gaming, but also there are appropriate lessons which can be drawn in applying it for real robotics and partially observable decision-making tasks. [6]

D. Reinforcement Learning to Enhance NPC AI in FPS Games

Reinforcement Learning (RL) is the revolutionary technology for making better Non-Player Characters (NPCs) for first-person shooters (FPSs). These problems are classified into the category of creating experience which is realistic and immersive game play. The problem of hard-coded behaviors has come along with predictable and fun boring behaviors which will make the player walk off from using the game. With the aid of RL games now train NPCs to behave dynamic, adaptive and realistic. It did make them a lot more believable.

Reinforcement learning such as this is most readily available in an FPS game when the algorithm for Proximal Policy Optimization is used to learn shooter NPCs. This learning reinforces changing of NPC strategies during actual play. That is, in the training phase, NPCs were exposed to different scenarios in a simulated environment and given a reward (positive) for hitting a target and another (negative) for failure to do so. Hence, iterative behaviors or learning will afford NPCs with a gradual process of learning decision-making and activities until almost human predictability and variation of activity in NPCs is achieved. RL-trained NPCs indeed realize aim in a much tense manner in contrast with deterministic ones, while their accuracy differs in the same manner as that of seasoned real human beings.

Further substantiating the believability of RL-trained NPCs, comparative analysis was carried out in a study where an RL-trained shooter NPC was locked in an SCM with a hand-coded same performance. The RL model showcased its superiority by dynamically changing the target, manner of the model according to movement and distance from it. The behavior of the hand-coded NPC was fixed and followed a repetitive pattern. Such a capacity adds to the perceived intelligence of the artificial NPC and verifies the solution to over-preciseness rates common in traditional NPC designs wherein perfect accuracies can make bots seem too inhuman and mechanical.

Furthermore, RL helps to refine the behavior of NPCs so that there is no need to put in different codes to handle cases. The learning agents learn by direct interaction and have the capacity to adjust to countless unforeseen game situations; hence it can save developers a lot of time and effort utilizing this method. This is more consequential in FPS games, where the immediacy of decision-making, many more players, and various modes of interaction all function to indicate that NPCs should work under countless conditions.

Similar to many other emerging fields in industries, reinforcement learning promises a lot, but it has its barriers such as requiring high computational resources and large triads of data for training. Advances in simulation technologies and RL algorithms are in the process of easing these hurdles to bring forth more efficient implementations. Adopting RL in design thus reaffirms the continuous redefining of NPC development and increased engagement and immersion for surfers in FPS games.

In summary, applying RL to FPS games for NPCs might just be the most important move toward having believable and engaging NPCs. RL can generate adaptive behavior that can be human-like and might break most of the remaining limitations for traditional NPC design. It promises to revolutionize the understanding of the development player experience in the industry.= [7]

VI. Proposed Methodologies

To deal with challenges in building adaptable and high-performing agents in video games, most especially in FPS environments, we hybridize Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC). This will hinge on utilizing joint strengths of the two algorithms to produce robust and versatile RL agents capable of mastering strategic and dynamic game mechanics.

A. Framework Overview

PPO is the most appropriate for the case; it is stable and efficient in learning discrete action spaces since it introduces a trust region constraint to avoid large updates that may destabilize learning. On the other hand, SAC excels in addressing continuous action spaces and in promoting exploration with entropy regularization that allows agents try out new discovered strategies in even very high-dimensional situations.

B. Steps of Methodology

Initial stabilization with PPO: Train the agent using PPO during the beginning stage of learning to stabilize the policy. PPO discovers essential behaviors quickly, focusing on maximizing reward structure and is best applied in discrete-semi-structured environments like maze solving for NPC pathfinding or weapon choice for FPS games. As policy divergence is practically avoided, agents learn the core objectives and mechanics very quickly. [9]

Refinement and Adaptation with SAC: Once the core policy is developed, move to training based on SAC. SAC has an exploration which is entropy based and thus encourages agents to try less obvious or riskier actions during this refinement phase, which is crucial for mastering some advanced, dynamic mechanics like:

Traversal systems that include jetpacking or grappling pose intricate controls requiring a smooth and uninterrupted performance in their use. Precision aiming and recoil should be combined with motor activity management and adaptive firing techniques. Real-time adaptation is adjusting to the capricious player behavior or environment changes, such as destructible objects or new obstructions. Dual-feedback Systems Incorporate a dual-reward system during training. PPO will maximize rewards for environment-driven events, which will keep the agent aligned with immediate game-play objectives like reaching destinations or completing tasks or even shooting enemies with great accuracy.

SAC is either maximizing the expected reward with supplementary entropy for long-term exploration or developing an agent's ability to discover a few unconventional but potentially better strategies. [9]

1) Dual-Feedback Mechanism: Increase the reward scheme during the training:

PPO maximizes all rewards from the environment alone, ensuring that the agent, at each instance of gameplay, directs itself towards goals, tasks, or attempted actions within it, for example, fighting. SAC makes agent long-term exploratory by maximal expected reward with added entropy for discovering the much unconventional but probably best ways of achieving the objective.

C. Application for FPS Games

1) Coordinations of team and tactics: Use PPO to assign strategic rolls like offense, defense or resource collection, and use SAC to refine tactical execution such as coordinating flanking maneuvers or efficiently using resources. Dynamic map navigation: The hybrid model will allow agents to navigate complex, multi-layered maps without having to develop new routes and dangers manually. Skill-based actions: Train agents to learn and adapt to advanced FPS mechanics like adjusting shots made while weapon recoil, performing precise jumps or parkour maneuvers, and making optimal selections from possible take cover points during the act of combat. [6]

VII. Conclusion

This research paper discusses how reinforcement learning (RL) is revolutionizing video games by enhancing gameplay mechanics and player experiences. It defines the evolution of RL through its process from traditional rule-based methods to advanced algorithms such as Deep Q-Network (DQN) and policy gradient and actor-critic methods, which are capable of developing game agents that are intelligent and adaptable. The applications of RL are listed in the likes of procedural content generation, NPC behavior development, and first-person shooter (FPS) games. It explains the handling of complex environments and dynamic interactions.

One of the most important contributions is the combined Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) hybrid approach towards fulfilling strategic and dynamic challenges typical of FPS games, a prime example of RL maturity that can develop agents with the adaptation capacity and behavior of the human player. The evolution of RL promises to redefine the gaming landscape in artificial intelligence as it approaches realistic and immersive gameplay. [1], [8], [9]

References

[1] X. Hou, “Exploring the Role of Reinforcement Learning in Video Game Environments,” Advances in Computer Science Research. Atlantis Press International BV, pp. 193–201, 2023.

[2] X. Fan, “The Application of Reinforcement Learning in Video Games,” Advances in Computer Science Research. Atlantis Press International BV, pp. 202–211, 2023.

[3] Shao, Kun, et al. “A Survey of Deep Reinforcement Learning in Video Games.” ArXiv (Cornell University), 23 Dec. 2019.

[4] Alonso, Eloi, et al. “Deep Reinforcement Learning for Navigation in AAA Video Games.” International Joint Conference on Artificial Intelligence, 9 Aug. 2021.

[5] Kaiser, Łukasz, et al. “Model Based Reinforcement Learning for Atari.” International Conference on Learning Representations, 30 Apr. 2020.

[6] Lample, Guillaume, and Devendra Singh Chaplot. “Playing FPS Games with Deep Reinforcement Learning.” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, 13 Feb. 2017.

[7] A. Servat and H. S. Mohamadi, ”Immersive Game Worlds: Using Deep Reinforcement Learning for Lifelike Non-Player Characters,” in Proc. Int. Serious Games Symp. (ISGS), Tehran, Iran, Dec. 2023.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, p. 529, 2015.

[9] “Rainbow: combining improvements in deep reinforcement learning,” in AAAI Conference on Artificial Intelligence, 2018

[10] S. John, W. Filip, D. Prafulla, R. Alec, and K. Oleg, “Proximal policy optimization algorithms,” CoRR, vol. abs/1707.06347, 2017.