Journal of
Instrumentation Technology & Innovations

ISSN: 2249-4731 (Online)
ISSN: 2347-7261 (Print)
Volume 14, Issue 2, 2024
DOI (Journal): 10.37591/JoITI

http://engineeringjournals.stmjournals.in/index.php/JoITI/index

Research JoITI

Mario Ai Model Using Gaming Reinforcement Learning

Pooja Bharat Raskar[1,*], Rahi Vilas Thale[2], Nikita Saindane[3], Krutika Satish Shinde[4]

Abstract

It is essential for research on computational and/or artificial intelligence (CI/AI) applied to games to have relevant games to apply AI algorithms to. This is pertinent. It doesn't matter if one is studying how to use CI/AI techniques to test and improve AI (e.g., games provide challenging yet scalable problems which engage many central aspects of human cognitive capacity) or how to use CI/AI techniques to improve games (e.g., player satisfaction modeling, procedural content generation, and the creation of believable and interesting bots). Using reinforcement learning techniques, this paper proposes a thorough framework for teaching an intelligent agent to play Mario, a beloved video game. To improve game performance, the suggested model integrates the Proximal Policy Optimization (PPO) algorithm with the prospective integration of the Deep Q-Learning (DQL) algorithm. Setting up Mario in a gaming environment, starting the game, and preparing the game state are all part of the setup process. By utilizing vectorization and grayscale approaches, the agent is able to depict the game environment more effectively. To improve our method, we take cues from seminal academic articles like "A Survey of Deep Reinforcement Learning in Video Games" and "System Design for an Integrated Lifelong Reinforcement Learning Agent for Real-Time Strategy Games." The research looks at PPO and DQL's individual effectiveness as well as comparing their performance to reveal their advantages and disadvantages when it comes to teaching an AI agent to overcome the obstacles in the Mario game.

Keywords: AI Model, game reinforcement learning, proximal policy optimization, artificial intelligence, Mario game.

INTRODUCTION

***Author for Correspondence**
Pooja Bharat Raskar
E-mail: poojabr20comp@student.mes.ac.in

[1]Student, Department of Computer Engineering, Pillai HOC College of Engineering and Technology, Rasayani, Maharashtra, India
[2]Student, Department of Computer Engineering, Pillai HOC College of Engineering and Technology, Rasayani, Maharashtra. India
[3]Assistant Professor, Department of Computer Engineering, Pillai HOC College of Engineering and Technology, Rasayani, Maharashtra, India
[4]Student, Department of Computer Engineering, Pillai HOC College of Engineering and Technology, Rasayani, Maharashtra, India

Received Date: April 11, 2024
Accepted Date: April 15, 2024
Published Date: April 28, 2024

Citation: Pooja Bharat Raskar, Rahi Vilas Thale, Nikita Saindane, Krutika Satish Shinde. Mario AI Model Using Gaming Reinforcement Learning. International Journal of Microelectronics and Digital Integrated Circuits. 2023; 9(2): 1–6p.

Having appropriate games to apply AI algorithms to is crucial for research on computational and/or artificial intelligence (CI/AI) applied to games. This is relevant. Regardless of whether one is researching how to use CI/AI techniques to improve games (e.g., player satisfaction modeling, procedural content generation, and the creation of believable and interesting bots) or how to use games to test and improve AI (e.g., games provide challenging yet scalable problems which engage many central aspects of human cognitive capacity).In recent years, the intersection of artificial intelligence (AI)and gaming has garnered significant attention, offered not only entertainment value but also served as a fertile ground for research and experimentation in AI algorithms and methodologies. Since each game presents a unique set of obstacles, no single game will ever be able to fulfill all initiatives and directions within this rapidly expanding field of study. Nonetheless, the community stands to benefit greatly from standardizing a small number of publicly accessible games that allow for the straightforward and equitable comparison of rival CI/AI techniques [1]. A "perfect benchmark game" would need to meet a lot of requirements. It should assess a variety of intriguing cognitive skills, ideally ones that other benchmark games on the market do not adequately evaluate. In other words, it should either have a naturally steep learning curve or be "easy to learn but hard to master," meaning that it should be able to distinguish between players and algorithms at all skill levels. It should be visually beautiful, simple to comprehend, and generally relevant to the audience. It should be enjoyable to play. The input/output space, also known as the policy representation, should be broad enough to allow for the easy application of various CI/AI techniques [2]. The technical foundation is equally crucial. The benchmark game implementation ought to function uniformly across all systems on the main computer platforms that are currently and will likely be accessible in the near future. If the software is not easy to install and the application programming interface (API) is not intuitive, then anyone with a basic understanding of programming should be able to set up a basic solution in less than five minutes. If not, many researchers will opt to use their own benchmarks because they are more familiar with them. The implementation must have little computing overhead and be scalable to real-time performance multiplied hundreds or thousands of times [3]. In particular, this last need is crucial for integrating learning algorithms into the game. One of the prominent areas within this intersection is the development of AI agents capable of playingand mastering video games autonomously. The renowned performer Maio stands out among the plethora of video games as a tough and iconic arena for AI testing. This project explores the field of reinforcement learning in gaming by concentrating on AI creation. the field of reinforcement learning in games by concentrating on creating an AI model that can play Mario on its own. The goal is to teach an AI sssagent to navigate the dynamic and challenging levels of the Mario game using reinforcement learning techniques, displaying intelligent behaviors similar to those of human players. Over the years, several artificial intelligence (AI) projects have focused on Mario, a well-known character in video games [4]. The Mario game environment is a perfect platform for testing the capabilities of AI agents since it offers a rich and dynamic setting full of obstacles, adversaries, and riddles. Earlier attempts to create artificial intelligence (AI) models for playing Mario have mostly depended on manually developed heuristics or rule-based systems, which frequently find it difficult to adjust to the game's complexity and variety. Reinforcement learning is a field within machine learning that focuses on how agents should behave in a given environment to maximize a concept known as cumulative reward [5]. It provides a potential framework for training AI agents to discover the best behaviors by trial and error. Reinforcement learning allows the AI agent to iteratively improve its decision-making process, eventually leading to expertise in the game, by rewarding desirable behaviors and penalizing unfavorable ones. In order to determine which reinforcement learning algorithms are most effective in training the AI agent for Mario gaming, this research will investigate a number of them, such as Q-learning, Proximal Policy Optimization (PPO), and Deep Q-Networks (DQN). The advantages and disadvantages of these algorithms in the context of gaming reinforcement learning will be assessed by analysis and testing, offering insights into their suitability for meeting the difficulties presented by the Mario environment. Using the Python programming language and well-known machine learning libraries like TensorFlow and Open AI Gym, the process entails implementing the AI model [6]. With the use of these tools, an artificial intelligence agent may be placed in a Mario simulation, interact with the game world, get feedback on its actions, and gain experience [7-11].

LITERATURE SURVEY

Research Paper	Year	Publisher's Name
Variations on the reinforcement learning performance of Blackjack [12]	9th August 2023	Avish Buramdoyal and Tim Gibbie
System Design for an Integrated Lifelong Reinforcement Learning Agent for Real-Time Strategy Games [13]	8th December 2022	James Smith, Michael Boran, Sahan Joshi
Towards Playing Full MOBA Games with Deep Reinforcement Learning [14]	31st December 2020	Deheng Ye, Guibin Chen, Sheng Chen and Wei Yang
A Survey of Deep Reinforcement Learning in Video Games [15]	26th December 2019	Kun Shao, Zhentao Tang and Yuanheng Zhu

International Journal of Microelectronics and Digital Integrated Circuits
Volume 9, Issue 2
ISSN: 2456-3986

PROPOSED SYSTEM

In order to accomplish the main goal of building an AI agent that can become proficient in the Mario game world through interaction and learning, a number of essential elements and techniques are included in the suggested system for building the Mario AI model utilizing gaming reinforcement learning. Create a virtual Mario world with specially designed simulators or game emulation frameworks like OpenAI Gym.. Make sure that every aspect of the environment—including obstacles, monsters, power-ups, and level structures—exactly mimics the dynamics, mechanics, and difficulties of the original Mario game. Examine and choose appropriate reinforcement learning strategies to train the artificial intelligence agent within the Mario environment. Think about techniques like Proximal Policy Optimization (PPO), Q-learning, Deep Q-Networks (DQN), or variations designed for the unique needs of game reinforcement learning Create the architecture for the AI model, which consists of neural network elements that represent the policy or value function of the agent. Use the Python programming language and appropriate machine learning libraries, such as PyTorch or TensorFlow, to implement the AI model. Link the AI model to the Mario simulation environment so that actions and observations can communicate back and forth.

To speed up learning and enhance generalization abilities, start the training process by letting the AI agent interact with the Mario environment simulation. Release the trained AI agent for use in production or real-time settings for use in real-world scenarios or additional testing. To provide smooth user or game environment interaction, integrate the AI agent with gaming platforms or systems. Through the agent's interactions, gather experience data such as observations, actions, rewards, and next states. To optimize for long-term cumulative rewards, use the chosen reinforcement learning algorithm to change the AI agent's policy or value function based on the experience gathered. To balance exploration and exploitation, fine-tune training parameters including learning rate, exploration rate, and discount factor. Test the trained AI agent extensively in the Mario simulation environment to see how well it performs. Track important performance indicators like level completion efficiency, cumulative prizes, success rate, and completion time. Proposed System Architecture is shown in Figure.1

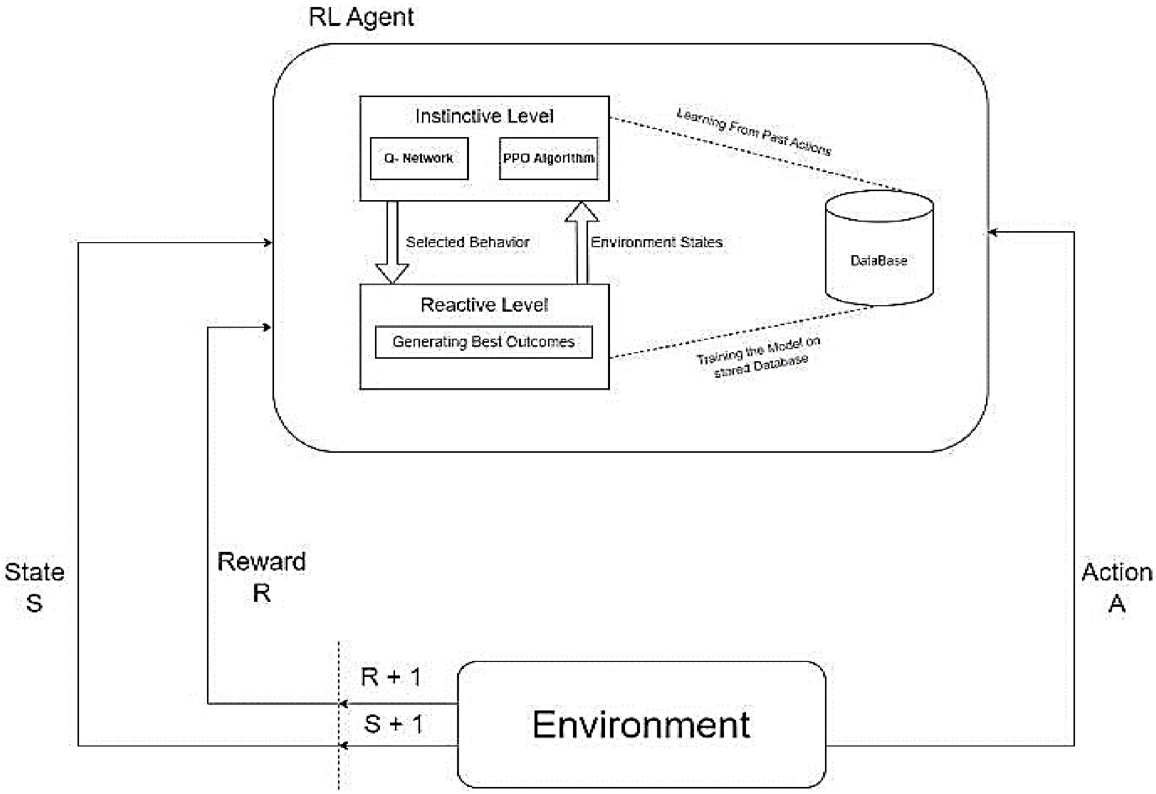

Figure 1. Proposed System Architecture.

To evaluate the AI agent's efficacy and competency, compare its results to those of human players or pre-established benchmarks. Analyze the behavior of the trained AI agent in detail, looking for trends, techniques, and places that could use better. Make iterative improvements to the AI model and training procedure based on the knowledge gathered from assessment and analysis. To improve performance and robustness, try different model topologies, training techniques, and reinforcement learning algorithms. To expedite learning and enhance generalization skills, incorporate strategies like curricular learning, transfer learning, or ensemble methods. For practical uses or additional research, deploy the trained AI agent into production or real-time situations. To provide smooth user or game environment interaction, integrate the AI agent with gaming platforms or systems. Track the AI agent's performance in practical situations and refine deployment tactics to optimize usefulness and efficiency. Flow Diagram of proposed system is shown in Figure 2.

FLOWDIAGRAM

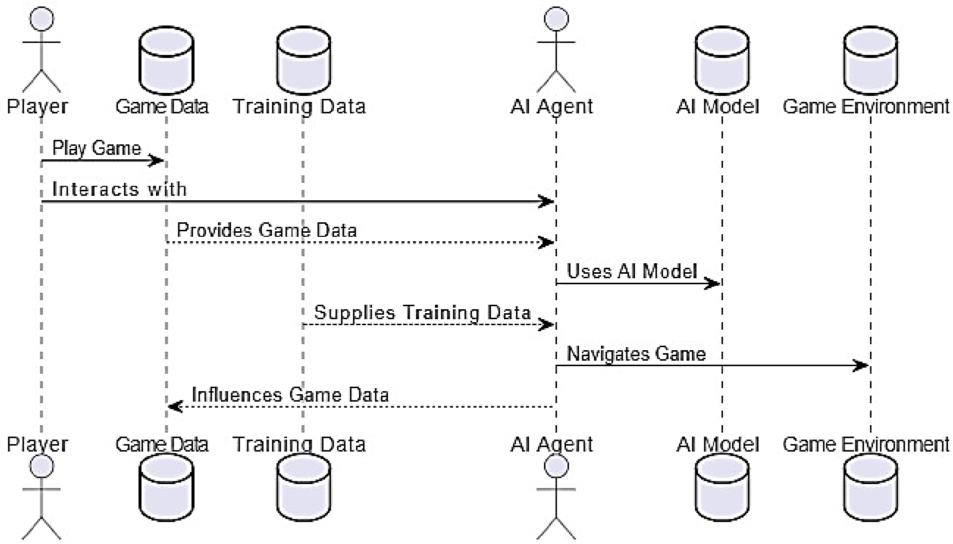

Figure 2. Dataflow diagram.

Advantages

1. *Adaptability:* Through trial and error, reinforcement learning enables the AI model to adjust and enhance its performance over time. The AI agent gains knowledge from its interactions with the Mario game world and modifies its tactics and behaviors to optimize gains.
2. *Generalization:* Reinforcement learning helps the AI agent generalize learnt methods across similar contexts by training it on a variety of Mario game levels and scenarios. This increases the AI agent's overall resilience and flexibility by allowing it to apply its knowledge and abilities to novel and unseen game scenarios.
3. *User Experience Enhancement:* By offering intelligent hints or suggestions, variable difficulty settings, and individualized coaching, AI-powered assistants or agents can improve the Mario gaming experience overall. This can produce more captivating and immersive gaming experiences and appeal to a wider spectrum of gamers, from beginners to experts.
4. *Competitive Gameplay:* In Mario game tournaments or challenges, AI agents based on reinforcement learning can face off against human players or other AI opponents. This pushes the bounds of gaming excellence, promotes skill growth, and creates healthy competition.

Disadvantages

1. *Complexity of the Game Environment:* The Mario game world is an incredibly intricate place to play, full of dynamic obstacles, level designs, and unpredictable adversary behavior. It can be difficult and computationally demanding to train an AI model to successfully navigate and grasp such a complex environment utilizing reinforcement learning approaches.

2. *Dependency on Environment Simulations:* Most AI model development for gaming uses simulated settings for testing and training. Performance gaps and difficulties applying newly learnt behaviors to real-world gaming settings might result from differences between the simulated and real game environments.
3. *Ethical Considerations:* There are ethical questions raised by the use of AI agents in gaming contexts, especially with regard to transparency, fairness, and the effect on user experiences. Reinforcement learning-trained AI models may display behaviors that human players find unfair or unpleasant, therefore careful design and assessment are required to address any potential ethical issues.
4. *High Dimensionality:* The Mario game's state space is a high-dimensional set of variables that includes the player's position, the positions of the enemies, power-ups, and environmental objects. Due to its large dimensionality, training requires more computational resources and makes learning optimal techniques and policies more challenging.

Future Scope

The Mario AI model's future application in gaming reinforcement learning is wide-ranging and multifaceted, with potential for interdisciplinary research, technical advancement, and societal effect. Researchers can open up new avenues for improving AI-driven games and influencing the path of interactive entertainment by adopting these future directions. In order to improve user comprehension and trust, future research might concentrate on making the AI model's decision-making processes more transparent and comprehensible.

CONCLUSION

This project has ventured into the intriguing nexus of artificial intelligence (AI) and gaming in an attempt to develop the Mario AI model using gaming Reinforcement learning. The goal is to create an intelligent agent that can interact with and learn from the iconic Mario game environment. This project has made great progress toward accomplishing its main goal by methodically investigating model design approaches, training strategies, and algorithms for reinforcement learning. Proximal Policy Optimization (PPO), Deep Q-Networks (DQN), and Q-learning are a few of the reinforcement learning algorithms that have been implemented and experimented with. These methods have yielded valuable insights into their suitability and efficacy for training the AI agent. The AI agent has shown remarkable adaptability and learning capability, gradually improving its performance and displaying intelligent behavior reminiscent of human players. In-depth discussions of important subjects including algorithm performance, generalization, exploratory trade-offs, interpretability, and ethical implications have been held throughout the project, emphasizing the complexity of AI-driven gaming environments and the wider social effects of AI use in gaming.

REFERENCES

1. Variations on the Reinforcement Learning performance of Blackjack, Avish Buramdoyal, Tim Gebbie, arXiv, 9 Aug 2023
2. System Design for an Integrated Lifelong ReinforcementLearning Agent for Real-Time Strategy Games, Indranil Sur, Zachary Daniels, Abrar Rahman, arXiv, 8 Dec 2022
3. Towards Playing Full MOBA Games with Deep Reinforcement Learning, Deheng Ye, Guibin Chen, WenZhang, Sheng Chen, arXiv, 31 Dec 2020
4. A Survey of Deep Reinforcement Learning in Video Games,Kun Shao, Zhentao Tang, Yuanheng Zhu, Member, IEEE, Nannan Li, and Dongbin Zhao, Fellow, IEEE, 26 Dec 2019.
5. Sah SK, Fidele H. Enhancing Mario Gaming Using Optimized Reinforcement Learning. InInternational Conference on Distributed Computing and Intelligent Technology 2024 Jan 4 (pp. 206–221). Cham: Springer Nature Switzerland.
6. Karakovskiy S, Togelius J. The mario ai benchmark and competitions. IEEE Transactions on Computational Intelligence and AI in Games. 2012 Feb 22;4(1):55–67.
7. Lee G, Luo M, Zambetta F, Li X. Learning a Super Mario controller from examples of human play. In2014 IEEE Congress on Evolutionary Computation (CEC) 2014 Jul 6 (pp. 1–8). IEEE.

8. Mohan S, Laird JE. Learning to play Mario. Tech. Rep. CCA–TR-2009–03. 2009.
9. Engelsvoll RN, Gammelsrød A, Thoresen BI. Generating levels and playing Super Mario Bros. with deep reinforcement learning using various techniques for level generation and deep q-networks for playing (Master's thesis, University of Agder).
10. Hou NC, Hong NS, On CK, Teo J. Infinite Mario Bross AI using genetic algorithm. In2011 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT) 2011 Oct 20 (pp. 85–89). IEEE.
11. Arzate Cruz C, Igarashi T. Mariomix: Creating aligned playstyles for bots with interactive reinforcement learning. InExtended abstracts of the 2020 annual symposium on computer-human interaction in play 2020 Nov 2 (pp. 134–139).
12. Buramdoyal A, Gebbie T. Variations on the Reinforcement Learning performance of Blackjack. arXiv preprint arXiv:2308.07329. 2023 Aug 9.
13. Sur I, Daniels Z, Rahman A, Faber K, Gallardo G, Hayes T, Taylor C, Gurbuz MB, Smith J, Joshi S, Japkowicz N. System design for an integrated lifelong reinforcement learning agent for real-time strategy games. InProceedings of the Second International Conference on AI-ML Systems 2022 Oct 12 (pp. 1–9).
14. Ye D, Chen G, Zhang W, Chen S, Yuan B, Liu B, Chen J, Liu Z, Qiu F, Yu H, Yin Y. Towards playing full moba games with deep reinforcement learning. Advances in Neural Information Processing Systems. 2020;33:621–32.
15. Shao K, Tang Z, Zhu Y, Li N, Zhao D. A survey of deep reinforcement learning in video games. arXiv preprint arXiv:1912.10944. 2019 Dec 23.

