2024 2nd World Conference on Communication & Computing (WCONF)
Raipur, India. Jul 12-14, 2024

Unity ML-Agents: Revolutionizing Gaming through Reinforcement Learning

Umesh Raut
Department of Computer Engineering and Technology
MIT World Peace University
Pune, India
umesh.raut@mitwpu.edu.in

Prathmesh Galchhaniya
Department of Computer Engineering and Technology
MIT World Peace University
Pune, India
prathmeshmg@gmail.com

Anish Nehete
Department of Computer Engineering and Technology
MIT World Peace University
Pune, India
anishnehete@hotmail.com

Rohan Shinde
Department of Computer Engineering and Technology
MIT World Peace University
Pune, India
roshinde24@gmail.com

Avaneesh Bhoite
Department of Computer Engineering and Technology
MIT World Peace University
Pune, India
avaneesh.bhoite@gmail.com

Abstract **- At the vanguard of AI, Reinforcement Learning (RL) is transforming sectors and pushing the limits of human-computer interaction. In the world of gaming, RL has become a powerful force that presents unheard-of chances to improve user experiences, have an impact on game creation, and even transform the gaming industry. Even while players are immersed in well constructed worlds and challenges through gaming, the strict and scripted style of traditional game design has its drawbacks. Gamers frequently want for dynamic, erratic, and adaptable encounters that can compete with those provided by live opponents. Here is where RL steps in, bringing with it the power to create intelligent agents that can interact with virtual surroundings to learn, adapt, and change. This study explores the relationship between RL and gaming, looking into how RL techniques are being used to revitalize this multibillion-dollar sector. The paper explores the core ideas of reinforcement learning and reveals the mechanisms underlying this paradigm. We explore the nuances of RL algorithms and techniques, with a focus on the gaming industry, enabling developers to create non-player characters (NPCs) who pose distinct and customized challenges to players.**

Keywords - Reinforcement Learning, Artificial Intelligence, MLAgents, Unity, Blender, Gaming, Non-Player Characters

I. INTRODUCTION

Video games have captured the hearts and minds of millions of people in the ever-changing world of interactive entertainment, but they have also become a vibrant laboratory for cutting edge AI research. The distinction between the real and the imaginary is becoming increasingly hazy in the age of digital technology, and the people who inhabit these virtual worlds—known as NPCs—have become essential to the complete immersion that modern video games provide. A game's replay value, general appeal, and quality are largely determined by the actions, intelligence, and flexibility of its NPCs. It follows that the pursuit of improving NPC skills has led to a sizable body of research focused on the creation of sophisticated NPC learning strategies.

This paper explores the fascinating and rapidly developing field of NPC learning in the gaming arena. Academics can now experiment with AI algorithms in a unique and large-scale sandbox produced by video games. This allows for the creation of sentient, adaptable NPCs that can react to the actions of players and even learn from their experiences, in addition to static, pre-programmed creatures. The historical background of NPC development in video games, the critical role that NPC learning strategies play in influencing game narratives and experiences, and the state-of-the-art approaches that have advanced our ability to create NPCs that can actually learn from and adapt to the virtual environments they live in will all be covered in this paper. We aim to provide a comprehensive overview of the achievements made in this dynamic field and the cognitive abilities and adaptability of our NPCs in the complex virtual environments of racing video games by examining the rich tapestry of past and current research.

Expanding on the more general talk about NPC creation, we now focus on a particular use of AI in the context of racing games. Chan, Marvin, Chan, Christine, and Gelowitz's study [17] focuses on using AI techniques to create a racing simulator game for cars. The International Journal of Computer Games Technology paper provides a thorough analysis of the use of AI to create a realistic and immersive gaming experience. The writer's research different AI techniques and how the game's design incorporates them. This research advances knowledge about AI's role in the gaming industry by providing insights into the development process. The study gives significant information for game creators and AI enthusiasts interested in the convergence of technology and gaming.

The following sections will explore the historical background of NPC development, how important NPC learning techniques are to the development of game narratives, and the state-of-the-art approaches that are bringing us closer to developing NPCs that can truly learn and evolve within virtual worlds as we embark on this exploration. We will also look at a particular case study on AI in racing games to show the real-world uses of our research results.

II. LITERATURE REVIEW

A review of the literature on the application of deep learning to video game play is provided by Justesen et al. (2020). They examine how AI is affecting video games, discussing advances in deep learning methodologies. The main goal is to demonstrate how neural network-based training methods improve gaming performance by teaching agents to excel in a variety of video games[1].

Zhang and Zhao (2019) review the literature and analyze data-based reinforcement learning algorithms in non-zero sum games with uncertain drift dynamics. They explore the difficulties in modifying reinforcement learning for scenarios in which drift dynamics are not known beforehand. This work illuminates methods for handling uncertainty in game dynamics, which makes a substantial contribution to the field of reinforcement learning research[2].

Shao, Zhu, and Zhao (2019) concentrate on micromanagement strategies utilizing curriculum transfer learning and reinforcement learning in the real-time strategy game StarCraft. The authors examine how AI can be used to improve micromanagement choices in intricate gaming environments, emphasizing the importance of curriculum transfer learning and reinforcement learning in the creation of engaging gameplay[3].

Ni and Paul (2019) review the literature and discuss how reinforcement learning can be used to solve security issues in smart grids. They address the evolving risks to critical infrastructure by concentrating on a multistage game framework, which advances our knowledge of how reinforcement learning can strengthen smart grid security[4].

Jent examines how the Unity Engine was used to create the independent platformer game "Stranded Away," focusing on key elements of game design and development. With a focus on user experience and Unity Engine utilization, the analysis offers insightful knowledge for independent game developers looking to maximize Unity's game design capabilities[9].

With a focus on techniques and technical aspects of game development, Singh and Kaur provide a thorough examination of game production using the Unity Game Engine. The study provides crucial information for developers and enthusiasts interested in using Unity to create interactive gaming experiences, highlighting the potential of the Unity Game Engine to streamline development processes and improve game design[10].

In a novel study, Kudek and Sužnjević apply AI to combine robotic control for a chess game inside the Unity Game Engine. This study looks at how robotics and gaming can work together, exploring ways to use AI to enable automated chess moves and enhance the gaming experience[11].

Sindhu, Annabel, and Monisha emphasize the use of AI algorithms to improve gameplay. This study clarifies the relationship between AI and game development by showing how AI innovations boost user interactions and gameplay elements[12].

Factaritz et al. present an Asynchronous Advantage Actor Critic (A3C) reinforcement learning architecture for end-to-end car control in a racing game using RGB camera inputs, falling into the field of artificial intelligence and game content creation. Their work highlights the bot's generalization capabilities in racing mode and demonstrates effective learning methodologies, good performance on a variety of tracks, and potential real-world driving applications[5].

In their examination of deep reinforcement learning in Match-3 games, Kamaldinov and Makarov (6) offer an open-source testing platform for different RL algorithms. Their study tackles the complexities of Match-3 games and provides essential information about AI-driven improvements to gameplay[6].

Using Deep Q-learning, Mnih et al. [7] achieved results on Atari 2600 games that were superior to human performance. Their novel method involves using Atari 2600 games as a crucial benchmark for evaluating deep reinforcement learning algorithms, and training algorithms to maximize scores based on pixel inputs and game scores.

A thorough analysis of the RL algorithms DQN and A2C is conducted by Torrado[8]. Their research emphasizes the variability of RL success across different games by examining training environments, performance comparisons, and the unique characteristics of RL versus planning-based agents.

Blaskovic, Zuzic, and Orehovacki look into the factors that affect user experience and development and have an impact on the costs associated with making driving video game soundtracks. Using simulations carried out by machine learning techniques, the study builds a computer-aided design (CAD) tool that helps with the creation of driving video game tracks and provides recommendations[15].

The study by unidentified authors looks at the geometric design of roads in the context of driving simulators through video game simulations, emphasizing affordable solutions. They investigate the use of video game simulations, specifically Assetto Corsa and rFactor2, as research tools to assess road geometry design and lower costs related to research simulators[16].

S. G. Yang along with others investigate the use of deep reinforcement learning (DRL) approaches to improve autonomous vehicle navigation, with a special focus on path-tracking tasks. Previous techniques to path-tracking depended on classical control algorithms such as PID controllers, which had difficulties in dealing with complicated settings. The study makes innovative contributions to algorithm design and training approaches, with the goal of overcoming these problems and advancing the field by presenting a personalized RL-based methodology for improving unmanned vehicle navigation[18].

G. Xu proposes an innovative method for path planning that combines deep learning via reinforcement (DRL) using perturbed fluids. The study contributes to the developing field of path planning strategies, where standard methods frequently struggle in dynamic and uncertain settings. Previous research has investigated a variety of machine learning with physics-inspired algorithms for path planning, demonstrating the potential effectiveness of such hybrid techniques. By merging DRL using perturbed fluids, the study advances the features of path planning algorithms, providing resilience and adaptability in dynamic contexts that standard methods may lack[19].

J. Su contributes to the increasing body of research on the Traveling Salesman Problem with reinforcement learning. Being a classic combinatorial optimization problem, it is determining the shortest path that visits a set of cities exactly once before returning to the origin. Deep reinforcement learning has lately emerged as a viable approach to combinatorial optimization issues due to its capacity to learn intricate decision-making policies. The research provides a new way for improving the efficiency of RL algorithms in TSP solution by mixing diverse representations. This approach is inspired by a variety of representation learning strategies, with the goal of improving the efficiency and efficacy of TSP solutions, which have important practical uses in transportation, logistics, and routing optimization[20].

The study, Safe Reinforcement Learning for Autonomous Lane-Changing: Considering the Collision Time and Lane Boundary Constraints[21], advances the area of self-driving vehicles by addressing safety problems in lane-changing maneuvers. With the growing interest in self-driving vehicles, ensuring secure and effective lane changes is critical. Previous research has investigated a variety of ways for achieving safe lane shifts, including rules-based methods and methods based on machine learning. R. Jia and colleagues provide a new method to improve the reliability and security of automated lane-changing systems, thereby contributing to the evolution of autonomous vehicle technology.

W. Wu and X. Zhang's research focuses on the difficulty of coordinating many unmanned aerial vehicles (UAVs) within challenging situations. Previous studies in UAV swarm control investigated a variety of methodologies, including standard control techniques and optimization algorithms. However, traditional techniques might be unable to adapt to dynamic challenges and shifting circumstances in the environment. The research presumably proposes a novel technique to improve the communication and adaptation of drone swarms in dynamic as well as static multi-obstacle environments using reinforcement learning[22].

I. Thammachantuek's contributions to the field of self-driving cars by assessing the effectiveness of different algorithms based on machine learning for decision-making tasks. Previous studies regarding autonomous driving investigated various decision-making strategies, such as rule-based systems as well as machine learning approaches. However, comparative examination of machine learning techniques especially designed for making decisions in autonomous vehicles is scarce. The paper's thorough comparison is likely to provide an understanding of the strengths and drawbacks of various machine learning algorithms, thereby boosting the development of autonomous vehicle decision-making systems[23].

S. Chiba and H. Sasaoka's study explores transfer learning for autonomous driving, with a focus on model car racing. In order to improve the autonomy of these model cars, this research chiefly investigates the integration of reinforcement learning techniques, demonstrating the usefulness of artificial intelligence in gaming situations. Future developments in reinforcement learning algorithms suited for instantaneous decision-making in fast-paced racing situations may encompass this work's purview, which could result in enhanced autonomous driving systems for both practical and recreational uses[24].

The deployment of autonomous cars using machine learning techniques—including reinforcement learning methodologies—is covered in the work by S. Perla, N. N. K., and S. Potta. This research shows how artificial intelligence and gaming ideas are coming together to create autonomous systems that can navigate real-world settings. The use of AI to autonomous driving demonstrates technological advances in the field and offers safer and more effective modes of transportation. Subsequent investigations in this field might concentrate on optimizing algorithms to facilitate better decision-making in intricate situations, ultimately leading to the extensive integration of self-driving cars with enhanced functionalities and dependability[25]

Proximal Policy Optimisation (PPO) is the subject of a study on model-based reinforcement learning that is presented in the work by Y. Sun, X. Yuan, W. Liu, and C. Sun. This work investigates how model-based techniques might be combined with PPO to increase the effectiveness and sample complexity of algorithms for reinforcement learning. Such research aims to improve learning and decision-making processes in dynamic contexts by demonstrating the convergence of artificial intelligence and gaming. Model-based reinforcement learning has important applications in robotics, gaming, and autonomous systems, among other domains. Subsequent pursuits could entail enhancing algorithms and expanding applications throughout other sectors to enhance efficiency and relevance[26].

Multi-objective exploration in the context of Proximal Policy Optimisation (PPO) is discussed in the study by N. D. H. Khoi, C. Pham Van, H. V. Tran, and C. D. Truong. This work explores how PPO algorithms might be optimized to concurrently take into account a number of goals, like maximizing rewards and minimizing risks or expenses associated with exploration. This work advances artificial intelligence in decision-making processes, especially in dynamic and unpredictable situations, by incorporating multi-objective exploration methodologies. Subsequent paths could entail honing multi-objective optimisation techniques to improve the flexibility and resilience of reinforcement learning algorithms in a range of contexts, such as autonomous systems, robotics, and gaming[27].

III. METHODOLOGY

This section explains the development process used to create the kart racing game that is based on reinforcement learning. The research's objective was to develop an autonomous agent that could navigate a challenging racecourse, arrive at certain checkpoints, and use reinforcement learning to acquire the best driving practices. Figure 1 shows the methodology of the game.

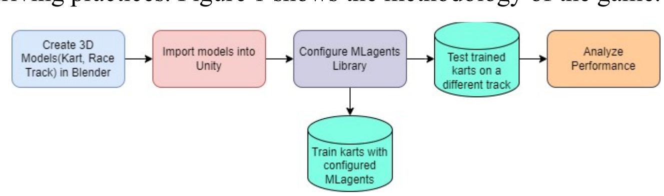

Fig. 1. Methodology

A. *Unity Game Engine:*

The game was made using the Unity game engine, a commonly used platform for generating interactive 3D applications. The Unity engine (Fig. 2) provides a comprehensive framework for producing simulations and games, allowing for the integration of complicated physics, graphics, and AI components.

Fig. 2. Screen displayed on starting the game

B. *Game Architecture:*

The game architecture comprises several interconnected scripts and components. Key components include:

- KartController: Responsible for handling the kart's movement, acceleration, steering, and collision detection.
- Checkpoint and CheckpointManager: Define the checkpoints in the racing circuit. The CheckpointManager controls timing restrictions, records the current checkpoint, and initiates activities when checkpoints are reached.
- AutomaticCameraSystem: Manages the game cameras, switching viewpoints dependent on the kart's progress through checkpoints.
- KartAgent: This applies reinforcement learning using the Unity ML-Agents framework. It describes the activities, incentives, and relationships between the agent and its surroundings.

C. *Reinforcement Learning Framework:*

To train the autonomous agent, the Unity ML-Agents toolkit was used. It offers a user interface via which Unity games can use reinforcement learning algorithms. The KartAgent script derives from the ML-Agents Agent class and provides the essential methods:

- Initialise(): Configures the agent at the start of every training session.
- OnEpisodeBegin(): Starts the kart at the beginning location and resets the surroundings.
- CollectObservations(): This function compiles pertinent environmental data, like the distance to the next checkpoint, and feeds it to the neural network so that it can make decisions.
- OnActionReceived(): Controls the steering and acceleration of the kart by processing commands from the neural network.
- Heuristic(): Enables the agent to be manually controlled for testing purposes.

D. *Sensors and Observations:*

The agent sees the surroundings via vector observations. The agent can determine its position with relation to the track thanks to these observations, which also include the normalized vector between the kart and the next checkpoint. By feeding the neural network with the observations, the agent is able to make decisions more easily.

E. *Reward System:*

The purpose of the incentive system is to promote desired behavior. Positive incentives are given to the agent for finishing the race and getting to checkpoints in the allotted time. When players deviate from the path or take longer than expected to reach a checkpoint, they receive negative rewards. The rewards are carefully calibrated to incentivise the agent to learn optimal race strategy.

F. *Training Process:*

During the training phase, episodes are run iteratively so that the agent can pick up knowledge from its interactions with the surroundings. Reinforcement learning techniques are used to update the neural network of the agent, optimizing its policy over time to maximize the cumulative reward. To improve learning efficiency and stability, training parameters including learning rate, discount factor, and neural network design were adjusted.

G. *Evaluation and Validation:*

The ability of the trained agent to stay on the track, navigate through checkpoints, and finish laps within the allotted time was used to evaluate its performance. The evaluation approach involves testing the agent in various track configurations to measure its flexibility and generalization skills. Performance parameters such as completion rate, lap times, and adherence to the track were assessed to validate the effectiveness of the trained agent.

IV. FRAMEWORK

The fundamental technologies(Fig.3) serving as the sophisticated abstraction programming tools responsible for constructing the gaming system mentioned in the preceding section can be outlined as follows.

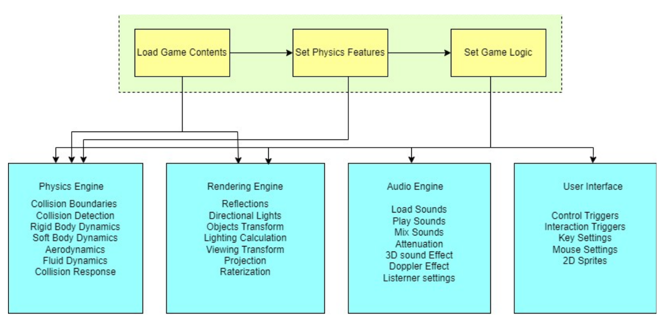

Fig. 3. Framework

A. *Unity Game Engine:*

Unity is a game creation engine that can generate programs for Windows, Mac OS X, Linux, and mobile platforms. Unity's features include physics calculations and graphic rendering, freeing up game developers to focus on the artistic and developmental portions of gaming software. The Unity engine allows developers to choose how realistic physics are implemented into game objects, including their position, scaling, and integration. Fig. 4 displays a snapshot of the agents being taught.

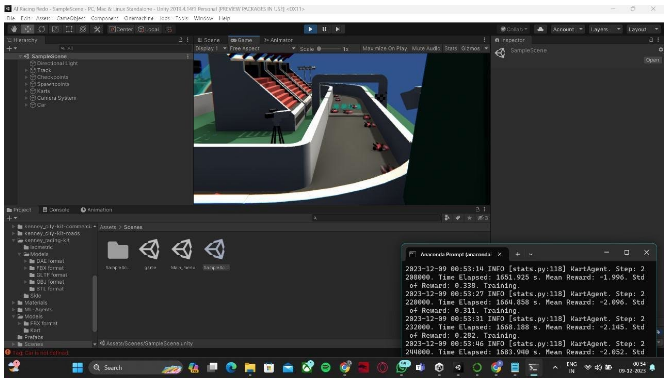

Fig. 4. Snapshot of agents in training

B. *Visual Studio Code-*

One powerful tool for creating games in Unity is Visual Studio Code, or VS Code. Using C# or other supported languages, you may quickly and effectively build, modify, and debug game scripts thanks to this integration. VS Code enhances the development process in various ways.

C. *Programming Languages and Data Structures-*

C# served as the main programming language for controlling object behavior via scripts. Because C# is object-oriented and class-based, it was the preferred language.

D. *Unity ML Agents Framework-*

Unity ML-Agents module gives a robust framework for constructing a dynamic racing game. Using this library, game developers may improve the racing experience and construct clever NPCs.. Developers may train NPCs to negotiate tracks efficiently via reinforcement learning, increasing their competitiveness.

IV. IMPLEMENTATION

Pressing the "W" key to accelerate, the "S" key to stop, the "D" key to steer right, and the "A" key to steer left are the keys the player uses to enter commands to the player car's controller. In order to claim victory, the player must complete a full circuit of the track. The main menu screen of the game is depicted in Fig 5. Fig 6 shows the game in action while its agents are being trained. The karts driven by non-player characters are moving in the opposite direction. The game's components, including the player-controlled race car and the NPC karts, are depicted in Fig 7.

Fig. 5. Start Screen

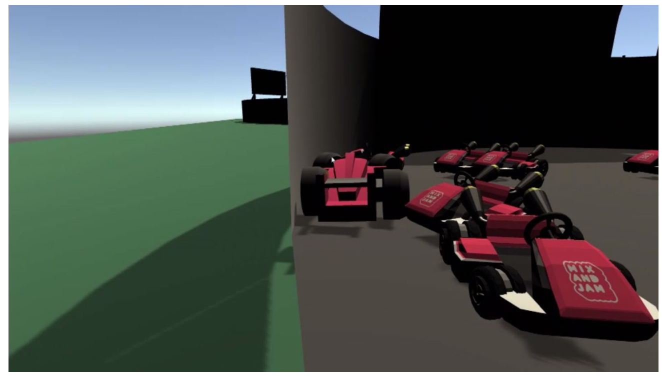

Fig. 6. Screenshot of game during training

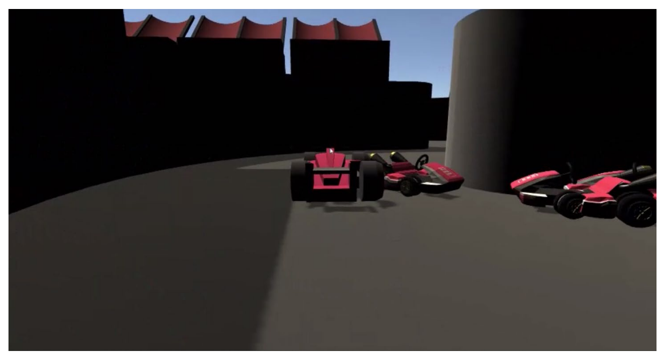

Fig. 7. Screenshot of game during Gameplay

V. RESULTS

With the model being trained in four and a half hours,we achieved stellar results with the agents demonstrating not only accuracy but also incredible speed. Figure 8 visually represents the environmental parameters observed throughout the training process.

Our model gained substantial intelligence after ~4 million iterations and was driving with optimal speed and accuracy after ~10 million iterations. The results obtained demonstrate the high efficiency of the Proximal Policy Optimization (PPO) algorithm, showcasing its effectiveness in various reinforcement learning applications. PPO is notable for its efficiency in utilizing samples, stability during training, adaptability to various environments, ease of implementation, and robustness across different hyperparameter

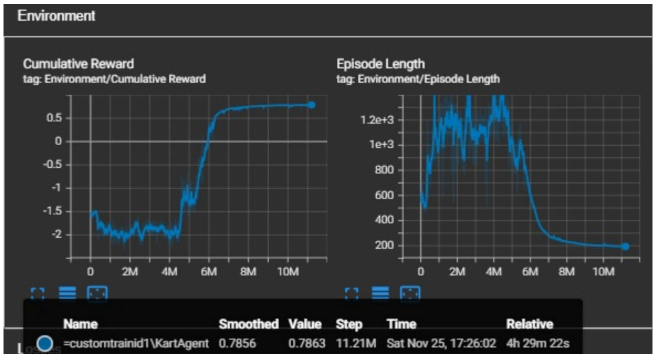

Fig. 8. Environment parameters during agent training

configurations. Its capability for parallelization and consistent performance across diverse domains further solidify PPO as a dependable choice. Additionally, PPO seamlessly integrates with deep neural networks, offering an effective solution for navigating the exploration-exploitation trade-offs inherent in complex scenarios.

When compared to traditional reinforcement learning techniques such as Q-learning, SARSA (State-Action-Reward-State-Action), and policy gradient methods, PPO outperforms them by effectively managing the balance between exploration and exploitation, resulting in stable training and consistent robustness across diverse environmental conditions. PPO's efficient use of samples, as well as its seamless integration with deep neural networks, considerably improve its adaptability and scalability for reinforcement learning applications.

V. FUTURE SCOPE

Unity proved instrumental in successfully developing a racing vehicle game, incorporating physics, vector computation, and waypoint system components for realistic racetrack navigation. Leveraging Unity for reinforcement learning (RL) transforms NPC behavior, offering dynamic, personalized gaming experiences. RL algorithms enable NPCs to adapt to player tactics, enhancing realism. Racing games provide valuable RL training settings for autonomous driving scenarios. Multi-agent RL fosters NPC cooperation, dynamic level creation, and AI-driven game design for compelling racing experiences. RL advancements in racing games support autonomous NPC racing leagues and AI research, with potential applications in VR and AR games, promising heightened immersion. In the future, we intend to combine RL techniques and AI shortest path algorithms for even more accuracy and immersive NPC behavior. Our configuration right now doesn't allow us to use some of the other models which hampered our testing, so we plan to expand on that as well.

ACKNOWLEDGEMENT

We would like to offer our deepest thanks to all individuals who played a key role in the effective completion of this paper as part of our 4th-year capstone project. First and foremost, we are profoundly appreciative of our project guide, Dr. Umesh Raut, for their steadfast leadership, mentoring, and vital insights throughout the study process. Their skill and attention substantially increased the quality of this job. We also offer our thanks to the teachers and personnel at Dr. Vishwanath Karad MIT World Peace University, whose assistance and resources set the framework for our project. Their dedication to academic quality has been important in our advancement as academics. This effort would not have been possible without the collaborative assistance and participation of all these people and institutions, and we are truly appreciative for their contributions.

REFERENCES

[1] N. Justesen, P. Bontrager, J. Togelius and S. Risi, "Deep Learning for Video Game Playing," in IEEE Transactions on Games, vol. 12, no. 1, pp. 1-20, March 2020, doi: 10.1109/TG.2019.2896986.

[2] Q. Zhang and D. Zhao, "Data-Based Reinforcement Learning for Nonzero-Sum Games With Unknown Drift Dynamics," in IEEE Transactions on Cybernetics, vol. 49, no. 8, pp. 2874-2885, Aug. 2019, doi: 10.1109/TCYB.2018.2830820.

[3] K. Shao, Y. Zhu and D. Zhao, "StarCraft Micromanagement With Reinforcement Learning and Curriculum Transfer Learning," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 3, no. 1, pp. 73-84, Feb. 2019, doi: 10.1109/TETCI.2018.2823329.

[4] Z. Ni and S. Paul, "A Multistage Game in Smart Grid Security: A Reinforcement Learning Solution," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 9, pp. 2684-2695, Sept. 2019, doi: 10.1109/TNNLS.2018.2885530.

[5] M. Jaritz, R. de Charette, M. Toromanoff, E. Perot, & F. Nashashibi. (2018). End-to-End Race Driving with Deep Reinforcement Learning. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2070-2075).

[6] I. Kamaldinov and I. Makarov. (2019). Deep Reinforcement Learning in Match-3 Game. In 2019 IEEE Conference on Games (CoG) (pp. 1-4).

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, & M. Riedmiller. (2013). Deep Q-networks. arXiv preprint arXiv:1312.5602.

[8] R. R. Torrado, P. Bontrager, J. Togelius, J. Liu, & D. Perez-Liebana. (2018). Deep Reinforcement Learning for General Video Game AI. In 2018 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 1-8).

[9] L. Blašković, A. Žužić and T. Orehovački, "Stranded Away: Implementation and User Experience Evaluation of an Indie Platformer Game Developed Using Unity Engine," 2023 46th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, 2023, pp. 92-97, doi: 10.23919/MIPRO57284.2023.10159769

[10] S. Singh and A. Kaur, "Game Development using Unity Game Engine," 2022 3rd International Conference on Computing, Analytics and Networks (ICAN), Rajpura, Punjab, India, 2022, pp. 1-6, doi: 10.1109/ICAN56228.2022.10007155.

[11] D. Kudek and M. Sužnjević, "Automated Robot Control for a Game of Chess in Unity Game Engine through Artificial Intelligence," 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 2021, pp. 1166-1171, doi: 10.23919/MIPRO52101.2021.9597142.

[12] R. M. Sindhu, L. S. P. Annabel and G. Monisha, "Development of a 2D Game using Artificial Intelligence in Unity," 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2022, pp. 1031-1037, doi: 10.1109/ICOEI53556.2022.9776750

[13] .Kavouras, I., Rallis, I., Doulamis, A., Doulamis, N. (2023). Evaluating the Feasibility of Fast Game Development Using Open Source Tools and AI Algorithms. In: Krouska, A., Troussas, C., Caro, J. (eds) Novel & Intelligent Digital Systems: Proceedings of the 2nd International Conference (NiDS 2022). NiDS 2022. Lecture Notes in Networks and Systems, vol 556. Springer, Cham. https://doi.org/10.1007/978-3-031-17601-2_13

[14] Aleena Chia (2022) The metaverse, but not the way you think: game engines and automation beyond game development, Critical Studies in Media Communication, 39:3, 191-200, DOI: 10.1080/15295036.2022.2080850

[15] Stokes, Mitchell, "Improving the performance of skeletal mesh animations in the Blender game engine" (2014). EWU Masters Thesis Collection. 187.

[16] .Santiago Higuera de Frutos, María Castro,Assessing sim racing software for low-cost driving simulator to road geometric research,Transportation Research Procedia,Volume 58,2021,Pages 575-582,ISSN 2352-1465,https://doi.org/10.1016/j.trpro.2021.11.076.

[17] Chan, Marvin & Chan, Christine & Gelowitz, Craig. (2015). Development of a Car Racing Simulator Game Using Artificial Intelligence Techniques. International Journal of Computer Games Technology. 2015. 1-6. 10.1155/2015/839721.

[18] S. G. Yang, E. H. Cho, J. Kim and S. -C. Lim, "Deep Reinforcement Learning-Based Path-Tracking for Unmanned Vehicle Navigation Enhancement," 2024 International Conference on Electronics, Information, and Communication (ICEIC), Taipei, Taiwan, 2024, pp. 1-4, doi: 10.1109/ICEIC61013.2024.10457143.

[19] G. Xu, J. Wang and H. Liu, "Path Planning Based on Deep Reinforcement Learning and Perturbed Fluids," 2023 China Automation Congress (CAC), Chongqing, China, 2023, pp. 1217-1222, doi: 10.1109/CAC59555.2023.10451060.

[20] J. Su, Z. Li and Y. Lu, "Deep Reinforcement Learning Algorithm Combining Different Representations to Solve the Traveling Salesman Problem," 2023 China Automation Congress (CAC), Chongqing, China, 2023, pp. 6046-6051, doi: 10.1109/CAC59555.2023.10452020.

[21] R. Jia, S. Lu, W. Xie, L. Dai, Q. Wu and J. Zhang, "Safe Reinforcement Learning for Autonomous Lane-Changing: Considering the Collision Time and Lane Boundary Constraints," 2023 China Automation Congress (CAC), Chongqing, China, 2023, pp. 3484-3489, doi: 10.1109/CAC59555.2023.10450529.

[22] W. Wu and X. Zhang, "Reinforcement Learning-Based Swarm Control for UAVs in Static and Dynamic Multi-Obstacle Environments," 2023 China Automation Congress (CAC), Chongqing, China, 2023, pp. 1387-1392, doi: 10.1109/CAC59555.2023.10450844.

[23] I. Thammachantuek, S. Kosolsomnbat and M. Ketcham, "Comparison of Machine Learning Algorithm's Performance Based on Decision making in Autonomous Car," 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), Pattaya, Thailand, 2018, pp. 1-6, doi: 10.1109/iSAI-NLP.2018.8693002.

[24] S. Chiba and H. Sasaoka, "Basic Study for Transfer Learning for Autonomous Driving in Car Race of Model Car," 2021 6th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand, 2021, pp. 138-141, doi: 10.1109/ICBIR52339.2021.9465856.

[25] S. Perla, N. N. K and S. Potta, "Implementation of Autonomous Cars using Machine Learning," 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, 2022, pp. 1444-1451, doi: 10.1109/ICECAA55415.2022.9936102.

[26] Y. Sun, X. Yuan, W. Liu and C. Sun, "Model-Based Reinforcement Learning via Proximal Policy Optimization," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 4736-4740, doi: 10.1109/CAC48633.2019.8996875.

[27] N. D. H. Khoi, C. Pham Van, H. V. Tran and C. D. Truong, "Multi-Objective Exploration for Proximal Policy Optimization," 2020 Applying New Technology in Green Buildings (ATiGB), Da Nang, Vietnam, 2021, pp. 105-109, doi: 10.1109/ATiGB50996.2021.9423319.