iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XII Month of publication: December 2025

DOI: https://doi.org/10.22214/ijraset.2025.76473

Improving Non-Playable-Characters (NPC) in Games Using Reinforcement Learning

Suraj S J[1], Dr. Sankhya Nayak[2], Shreyas G Hegde[3], Sujay Bharadwaj K[4], Ullas L M[5]
Department of Computer Science & Engineering, Jawaharlal Nehru New College of Engineering,

Abstract: Non-Player Characters (NPCs) play a critical role in shaping player experience in modern digital games, yet traditional NPC behaviour is largely driven by static, rule-based logic that lacks adaptability and realism. Such approaches often result in predictable and exploitable gameplay, limiting long-term player engagement. This project presents Neo NPC, a reinforcement learning–based framework for developing adaptive NPC behaviour in a 2D fighting game environment. The proposed system leverages offline reinforcement learning to train intelligent NPC agents that exhibit progressively sophisticated combat strategies across predefined difficulty levels, namely Easy, Medium, and Hard. The study focuses on designing a structured training and deployment pipeline that decouples model learning from real-time gameplay execution. Gameplay environments are modelled to generate state-action-reward trajectories, which are used to train NPC policies using the Proximal Policy Optimization (PPO) algorithm. Trained policies are periodically evaluated, versioned, and categorized into difficulty tiers based on quantitative performance metrics such as win rate, damage efficiency, and survival time. These validated models are then integrated into the game engine through a modular inference layer, enabling real-time decision-making without modifying core game logic. Experimental results demonstrate clear behavioural differentiation across difficulty levels, with higher-tier models exhibiting improved defensive responses, reduced vulnerability to repetitive player strategies, and increased action diversity. Human playtesting further confirms that the adaptive NPCs provide a more challenging and engaging gameplay experience compared to traditional scripted opponents. The proposed approach highlights the effectiveness of reinforcement learning in producing scalable, reusable, and intelligent NPC behaviour. Future extensions of this work include online adaptation, multi-agent self-play, and transfer of trained models to more complex game environments.
Keywords: Game AI Agent, Reinforcement Learning, QLearning, Neural Networks, Deep Learning, Game AI

I. INTRODUCTION

Neo NPC – Interact, Adapt, Evolve is an innovative project that redefines the role of Non-Playable Characters (NPCs) in video games and interactive simulations. Traditionally, NPCs have operated using rigid, scripted behaviours—responding to player actions through predefined rules and conditions. This often results in predictable, repetitive, and emotionally detached interactions that limit player immersion and reduce the dynamic nature of gameplay. As player expectations rise and virtual environments become more complex, the need for intelligent, responsive, and adaptive NPCs has become a crucial frontier in game development and AI research.

Despite their ubiquity, the design of NPCs has traditionally followed a rigid, rule-based architecture, relying heavily on hand-crafted scripts and decision trees. While effective for basic functionality, such systems struggle to keep pace with player expectations for intelligent, adaptive, and emotionally engaging characters. These static NPCs often fail to reflect the complexity of human behaviour, resulting in gameplay experiences that, although visually stunning, lack depth and believability at the character interaction level.

Neo NPC seeks to challenge and redefine these conventions by developing a new generation of intelligent NPCs that can exhibit learned behaviour derived from machine learning models. At the core of this project lies the belief that NPCs should not merely exist as reactive entities with pre-written responses but as agents whose actions are influenced by patterns learned during training. Instead of relying on static scripts, the system uses pre-trained reinforcement learning models developed over different training durations.

These versions—labeled Easy, Medium, and Hard— allow the NPC to demonstrate progressively complex behaviours without requiring real- time self-evolution. The project aims to close the gap between player and character interaction by making NPCs feel more alive, aware, and contextually responsive within the scope of their trained capabilities.

II. AIMS AND OBJECTIVE

The design of Non-Playable Characters (NPCs) in modern games still relies heavily on scripted behaviour, decision trees, and finite state machines, resulting in predictable and non-adaptive interactions that break player immersion. The problem addressed in this project is the lack of NPCs capable of demonstrating learned, adaptable behaviour without manual rule programming. Neo NPC aims to bridge this gap by integrating reinforcement allowing NPCs to exhibit more realistic and varied responses within gameplay while maintaining computational feasibility.

This research paper has the following main objective:

1. To design and implement a modular NPC framework that integrates reinforcement learning, neural networks, and behaviour -driven AI, enabling characters to exhibit learned behaviours based on pre-trained models rather than scripted logic.
2. Ensure that NPCs appear lifelike, context-aware, and responsive within the scope of their trained behaviours, enhancing player immersion through believable and varied interactions.

III. LITERATURE SURVEY

The literature survey reviews recent advances in adaptive NPC design, emphasizing reinforcement learning (RL), deep learning, and hybrid AI frameworks in gaming. It highlights the transition from rule-based systems to intelligent, context-aware NPCs capable of evolving behaviour dynamically based on player interaction.

A. *Natalia Khan et al. (International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2023). "AI Chatbots in Gaming Experience Transformation." [1]*

This paper investigates the application of AI chatbots in gaming to enhance non-linear storytelling and interactive dialogue. It reviews implementations focusing on context awareness and dynamic narrative generation, identifying both computational limitations and potential integration with reinforcement learning systems. The study emphasizes how AI can offer richer gameplay through emotionally adaptive dialogues, though challenges remain in maintaining coherence over extended sessions. It also highlights the role of sentiment analysis in shaping dialogue responses and outlines avenues for integrating voice-based interactions.

B. *Kun Shao et al. (Computational Intelligence and Neuroscience, 2022). "A Survey of Deep Reinforcement Learning in Video Games." [2]*

This survey categorizes and analyzes Deep Reinforcement Learning (DRL) methods like DQN, PPO, and A3C across various game genres. It outlines algorithmic strengths, implementation challenges, and potential improvements in areas like sample efficiency and multi-agent coordination, but lacks original empirical benchmarks. The paper serves as a foundational reference by tracing the evolution of DRL in gaming and underlines the importance of transfer learning and real-time adaptability in future systems. It also reviews open-source toolkits used in DRL research and notes the scarcity of standardized evaluation environments. This gap affects reproducibility and makes performance benchmarking difficult across diverse gaming environments, limiting cross-study comparisons.

C. *Wemade Next Co. Ltd. et al. (NVIDIA Technical Report, 2025). "AI-Powered NPCs in MMORPG: Asterion Case Study." [3]*

This case study presents an adaptive AI boss character named Asterion from the MMORPG MIR5. It utilizes cloud-based learning and NVIDIA's Avatar Cloud Engine to dynamically adapt strategies based on player behaviour, though its reliance on proprietary hardware limits broader access. The system demonstrates how real-time personalization of NPC actions can improve engagement but also highlights the trade-off between computational power and availability. Asterion's behaviour evolves across player encounters, enhancing replay value and immersion.

D. *Tejal Kadam et al. (International Journal of Scientific Research in Engineering and Management, 2020). "Reinforcement Learning Bot for Mario." [4]*

This paper details the development of an RL-based bot for Super Mario, using reward shaping to navigate game levels. The bot demonstrates emergent behaviour and learns effective strategies autonomously. Limitations include high variability and limited generalization. The authors suggest enhancements through hierarchical reinforcement learning to enable more structured task completion. The study also comments on scalability issues for more complex platforming environments. Additionally, it emphasizes the importance of balancing exploration versus exploitation in sparse-reward settings. The training process relies on episode-level feedback to optimize movement sequences. The implementation showcases how lightweight DRL frameworks can be applied to retro-style games with constrained action spaces.

E. Natalia Curado Carneiro (International Journal of Advanced Computer Science and Applications, 2021). "FSM-Based Behaviour Modeling for Mario." [5]

The study models Mario's behaviour using Finite State Machines (FSM), illustrating simple transitions between actions like walking and jumping. While effective for predictable behaviours, FSMs lack scalability for complex, adaptive decision-making in games. It also notes the rigidity in modifying FSMs when new states are added, making them less ideal for open-ended environments. The approach remains useful in constrained, highly controlled gameplay loops. This structure allows for clear debugging and deterministic outputs, which is advantageous during early-stage prototyping. However, the absence of memory or learning limits responsiveness to dynamic game contexts. The paper suggests FSMs are best suited for classic arcade or puzzle games with minimal player variability.

F. Pedro Almeida et al. (Applied Sciences, 2024). "Reinforcement Learning as an Approach to Train Multiplayer First-Person Shooter Game Agents." [6]

This paper evaluates deep reinforcement learning strategies—Curriculum Learning (CL), Behaviour Cloning (BC), and hybrid models—for training FPS agents in Unity. Using systematic matchups between bot teams, it found CL outperformed BC by 23.7% in win rates. Hybrid approaches had inconsistent outcomes due to training instability, but the work presents clear pipelines for real-time AI development. It also emphasizes the role of spatial awareness and reaction latency in agent behaviour modeling. The research highlights the challenge of balancing aggression and defense through adaptive reward structures. It proposes future refinement through multi-objective optimization and domain randomization. These insights contribute toward building robust, competitive agents suitable for commercial multiplayer environments.

G. Shouren Wang et al. (arXiv preprint, April 2025). "Enhancing Player Enjoyment with a Two Tier DRL and LLM Based Agent System for Fighting Games." [7]

This work introduces a Two-Tier Agent system combining DRL-based playstyle agents with an LLM-based hyper-agent that selects suitable opponents based on player feedback. The design enables adaptive matchups in Street Fighter II and boosts user experience, with gameplay showing up to 156% improvement in advanced technique execution. However, the setup adds complexity and lacks transparency in LLM decision-making. It further outlines the potential to scale this system into open-world or role-playing games. The authors emphasize the importance of continual learning cycles informed by player input to personalize difficulty and maintain engagement. The system also introduces a dynamic matchmaking layer, reacting to player proficiency trends in near real-time. Despite its promise, the hybrid approach raises questions about reproducibility, evaluation metrics, and ethical tuning of adaptive difficulty.

H. Chen Zhang et al. (IEEE Transactions on Games, Early Access, 2024). "Advancing DRL Agents in Commercial Fighting Games: Training, Integration, and Agent-Human Alignment." [8]

This paper presents Shūkai, a DRL agent deployed in Naruto Mobile, trained via Heterogeneous League Training (HELT) to generalize across 400+ characters. The system employs human-aligned reward mechanisms and diverse agent architectures. Results show a 22% boost in training speed and effective performance across characters, though the approach’s complexity and reliance on manual tuning are noted as drawbacks. It also emphasizes the need for continuous feedback loops to maintain balance as new characters are added. Additionally, the work explores modular policy learning to facilitate incremental updates as the game evolves. The integration pipeline accounts for real-time combat variability, ensuring agents respond fluidly to unpredictable human inputs. However, scalability concerns emerge when extending the framework to less structured or procedurally generated combat environments.

I. Iveta Dirgová Luptáková et al. (Sensors, 2024). "Playing Flappy Bird Based on Motion Recognition Using a Transformer Model and LIDAR Sensor." [9]

This research replaces image-based input with LIDAR data in Flappy Bird and uses Transformer models to enhance temporal understanding. Performance improved dramatically over CNN baselines, aided by a risk penalty mechanism.

The study, while innovative, is constrained by limited applicability in broader visual games and higher computational costs of Transformers. The findings suggest future integration with lightweight sensor fusion techniques. Additionally, the system demonstrates robustness to environmental noise, making it promising for accessibility-driven game adaptations. However, the setup requires precise calibration and stable input feeds, limiting ease of deployment. The work opens avenues for motion-controlled interfaces that bypass traditional visual pipelines.

J. Maciej Świechowski et al. (Artificial Intelligence Review, 2023). "Monte Carlo Tree Search: A Review of Recent Modifications and Applications." [10]

This survey covers enhancements in MCTS, such as UCT variants, parallelization, and neural integrations. It is especially useful for planning-intensive games like Go but has limited direct application in NPC control. The review is comprehensive but lacks empirical comparison across different games. The paper also identifies challenges in real-time deployment—such as latency constraints in RTS games—and emphasizes the need for memory-efficient modifications like dynamic node pruning to enable broader applicability. It also highlights hybrid approaches that combine MCTS with deep learning for better generalization. Moreover, the authors point to growing interest in using MCTS within procedural content generation. Despite the theoretical focus, the paper outlines several future research paths for adaptive and scalable search systems.

K. Muhammad Bambang Firdaus et al. (Journal of Physics: Conference Series, 2023). "FSM for Retro Arcade Fighting Game." [11]

The paper designs an FSM-based combo system in Godot for the game Brawl Tale. It highlights predictable transitions and responsive controls, validated through structured development cycles and user testing. However, its deterministic nature restricts adaptability. The study recommends FSMs primarily for games emphasizing mechanical precision over emergent behaviour. It also notes that FSMs simplify debugging and maintenance in tightly scoped environments. The system's modular state design improves clarity for developers working in collaborative teams. Despite limitations, FSMs remain practical for latency-sensitive arcade-style combat systems.

L. Natalia Curado Carneiro et al. (International Journal of Interactive Multimedia and Artificial Intelligence, 2022). "Integrating FSM and Reinforcement Learning for Hybrid NPCs." [12]

A hybrid framework combines FSM for high-level behaviour control and RL for fine-tuned decision-making. The platformer testbed demonstrates faster training and context-aware behaviours, though managing boundaries and reward shaping introduces design complexity. This framework shows promise for balancing predictability and adaptability in commercial games, particularly in genres like stealth or RPGs where nuanced agent responses and structured progression are vital. The study highlights how FSM structures can guide exploration, reducing erratic RL behaviour during early training. It also outlines how state transitions can be dynamically adjusted based on learned patterns. However, achieving smooth coordination between FSM logic and RL policies requires careful architecture design and testing.

M. Afzal Hussain et al. (Indonesian Journal of Electrical Engineering and Computer Science, 2023). "A Survey on Unity Game Engine: Architecture, Features, and Its Applications." [13]

This survey outlines Unity's system architecture, component model, and cross-platform deployment. It covers core areas like scripting, rendering, and asset workflows, but lacks discussion on AI or RL integration. The paper serves as a foundational reference for Unity's structure and capabilities. It also includes comparisons with Unreal Engine in terms of toolchain modularity and learning curve. Additionally, it notes Unity's strengths in mobile game development, lightweight runtime efficiency, and superior integration with VR/AR platforms.

IV. RESEARCH METHODOLGY

This research adopts a reinforcement learning–driven methodology to design, train, validate, and deploy intelligent Non-Player Character (NPC) behaviour in an interactive game environment, referred to as **Neo NPC**. The objective is to develop NPC agents capable of exhibiting adaptive, skill-scaled behaviour corresponding to predefined difficulty levels (Easy, Medium, and Hard), while maintaining reproducibility, modularity, and real-time performance suitability. The methodology integrates offline reinforcement learning training with runtime inference in a game engine, ensuring a clean separation between learning and execution.

The complete methodology is structured into the following steps:

1) Environment Design and State Representation:

A custom game environment was developed using **Pygame**, serving both as the playable game client and as the reinforcement learning environment. The environment adheres to the Markov Decision Process (MDP) formulation, defined by the tuple ⟨S, A, R, T⟩

State Space (S): The state representation includes normalized numerical features describing the current game context, such as:

- Relative positions and distances between player and NPC
- Health levels of both agents

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

- Current action states (attack, block, jump, idle)
- Cooldown timers and action availability flags

These observations are structured to be deterministic, fixed-dimensional, and compatible with standard RL policy networks. Action Space (A): A discrete action space is defined, covering all NPC capabilities, including movement, defensive actions, basic attacks, and special actions. Each action maps directly to an executable in-game behaviour.

This structured environment design ensures consistency between training and inference and enables reproducible policy learning.

2) Reward Function Design:

A carefully shaped reward function was used to guide policy learning toward desirable combat behaviour. Rewards were assigned based on:

- Positive rewards for dealing damage, successful defense, and strategic positioning
- Negative rewards for taking damage, unnecessary idling, or repeated ineffective actions
- Minor penalties for excessive action repetition to discourage exploitative strategies.

Reward shaping was intentionally conservative to avoid policy overfitting while encouraging gradual behavioural improvement. This approach supports stable learning and smoother difficulty progression across models

Event / Condition	**Reward Value**	**Q-Learning Role**
Successful attack (damage dealt)	+1.0	Immediate reward
Successful block / defense	+0.5	Immediate reward
Strategic positioning	+0.2	Shaping reward
Damage received	−1.2	Negative reward
Unnecessary idle action	−0.1	Penalty
Repeated ineffective action	−0.3	Penalty
Excessive action repetition	−0.2	Penalty
Match win (terminal state)	+10.0	Terminal reward
Match loss (terminal state)	−10.0	Terminal penalty

3) Reinforcement Learning Algorithm Selection:

A Q-learning–based approach was selected due to its conceptual simplicity, discrete action suitability, and interpretability for game AI behaviour analysis. In particular, Deep Q-Networks (DQN) enable learning optimal action–value functions directly from gameplay experience, making them well-suited for discrete combat actions such as movement, attack, defense, and evasion. The learning objective is to approximate the optimal Q-function

$$Q * (s, a) = E[rt + \gamma a' \max Q * (st + 1, a')]$$

which represents the expected cumulative reward of taking action $a$in state s. Experience replay buffers and target networks are employed to stabilize learning and prevent divergence. Neural Q-networks are implemented using multilayer perceptrons and optimized using Adam-based stochastic gradient descent.

4) Data Collection and Offline Training

Training is performed entirely offline to avoid impacting runtime performance. The training pipeline consists of:

- Generating gameplay transitions via self-play, scripted opponents, and ε-greedy exploration
- Storing state–action–reward–next-state tuples in a replay buffer
- Sampling minibatches from the buffer to iteratively update the Q-network Episodes are executed under deterministic physics and fixed time steps, ensuring repeatable experiments and controlled reward dynamics. Offline training allows large-scale experimentation and hyperparameter tuning without modifying the live game environment.

5) *Checkpointing* and Difficulty Labeling

Model checkpoints are saved at fixed training intervals corresponding to distinct learning stages. Each checkpoint is evaluated using objective performance metrics, including:

- Win rate against baseline scripted agents
- Damage efficiency (damage dealt vs. damage received)
- State–action coverage and behavioural diversity

Based on empirical evaluation results, checkpoints are categorized as:

- Easy: Early Q-networks with high exploration and suboptimal value estimates
- Medium: Partially converged policies exhibiting basic defense and counterplay
- Hard: Near-converged Q-functions demonstrating consistent strategy selection and optimal responses This enables difficulty scaling driven by learning progression rather than manual rule tuning.

6) Model Validation and Evaluation:

Selected checkpoints undergo validation using:

- Automated evaluation matches against fixed strategies
- Stress testing with repetitive player behaviours
- Human playtesting to assess perceived intelligence and fairness

Q-value distributions, learning curves, and policy entropy are analyzed to verify convergence and behavioural stability. Only models meeting predefined performance thresholds are approved for deployment.

7) Model Serialization and Version Control:

Validated Q-learning models are serialized along with structured metadata, including Network weights and architecture configuration, Observation encoding and discrete action mappings, Training steps, exploration parameters, and reward definitions, Evaluation metrics and integrity checksums All model artifacts are stored in a version-controlled repository with immutable releases to ensure reproducibility and rollback safety.

8) Runtime Integration and Inference:

During gameplay, the NPC Model Manager loads the selected Q-network corresponding to the chosen difficulty. For each game loop iteration:

- The current game state is encoded as an observation
- The Q-network computes Q-values for all valid actions
- The action with the highest estimated value is selected
- The action is executed through game mechanics and animations Inference latency is minimized to satisfy real-time gameplay constraints.

9) Logging, Monitoring, and Iterative Improvement:

Runtime sessions generate logs capturing selected actions, state frequencies, win/loss outcomes, and anomalous behavior. These logs are used to refine reward functions, state representations, and exploration strategies in subsequent offline training cycles, enabling continuous improvement without modifying core gameplay logic.

V. RESULTS AND DISCUSSION

The developed reinforcement-learning-based Neo NPC system demonstrates strong capability in producing adaptive and difficulty-scaled non player character behaviour within a 2D fighting game environment. The trained models exhibit clear behavioural differentiation across Easy, Medium, and Hard difficulty levels, validating the effectiveness of the proposed offline training and model stratification pipeline. Quantitative evaluation and qualitative gameplay analysis confirm that the learned policies generalize well across varied player strategies while maintaining real-time performance constraints.

Performance evaluation was conducted through automated playtesting and controlled gameplay scenarios. The Easy-tier models, derived from early training checkpoints, showed limited strategic awareness and higher vulnerability to repetitive player actions, achieving win rates in the range of approximately 30–45%. These models frequently relied on basic movement and attack patterns, resulting in higher damage intake and shorter survival times. Such behaviour aligns with the intended design goal of providing an accessible and forgiving experience for novice players.

In contrast, the Medium-tier models demonstrated notable improvements in combat effectiveness and defensive behaviour. These models achieved win rates between 50–65%, accompanied by longer survival times and reduced damage taken. Medium-tier NPCs exhibited partial adaptation to repetitive player strategies, often responding with blocking or evasive actions after repeated exposure. This indicates successful learning of intermediate-level tactical behaviour through extended training and exposure to diverse gameplay scenarios.

The Hard-tier models achieved the highest performance across all evaluation metrics. These models consistently recorded win rates exceeding 75% in controlled simulations and displayed significantly improved damage efficiency. Compared to Easy-tier models, Hard-tier NPCs showed an average reduction of approximately 40% in damage taken against repetitive player attacks, highlighting their ability to counter predictable strategies through repositioning, defensive timing, and action variation. Human playtesting further confirmed that Hard-tier NPCs were perceived as challenging and less predictable, enhancing overall gameplay engagement.

Behavioural analysis revealed a progressive increase in action diversity and decision quality with training maturity. While Easy-tier models exhibited high action repetition and reactive behaviour, Hard-tier models demonstrated more balanced action selection and reduced exploitability. The observed reduction in repetitive-action vulnerability confirms the effectiveness of reward shaping and action-penalty mechanisms used during training.

The robustness of the Neo NPC system is further attributed to its structured offline training methodology and checkpoint-based evaluation strategy. Periodic checkpointing and objective difficulty labelling enabled consistent difficulty scaling without reliance on handcrafted rules. This approach ensured that difficulty progression was driven by measurable performance metrics rather than subjective tuning.

From a system integration perspective, runtime inference performance remained within acceptable limits, with model inference consistently completing within 2 milliseconds per frame on CPU. This confirms that the separation of offline training and real-time inference successfully preserves gameplay smoothness while enabling intelligent NPC behaviour. The NPC Model Manager reliably loaded and validated model checkpoints using stored metadata and integrity checks, supporting seamless deployment of multiple difficulty versions without modifying core game logic.

Overall, the results demonstrate that reinforcement learning provides a viable and scalable alternative to traditional rule-based NPC design. The Neo NPC framework successfully produces adaptive, reusable, and difficulty-aware NPC behaviour that enhances player experience and replayability. Compared to scripted opponents, the learned NPCs exhibit superior adaptability and realism, establishing the proposed system as a strong foundation for future research in game AI. Potential extensions include online adaptation, multi-agent self-play, and transfer learning to more complex game environments.

VI. CONCLUSION

The Neo NPC system successfully demonstrates how reinforcement learning can be used to create adaptive, realistic, and scalable NPC behaviour in game environments. By training the agent through repeated interaction and evaluating multiple checkpoints, the project delivers three balanced difficulty levels—Easy, Medium, and Hard—without relying on scripted patterns or fixed decision trees. The modular architecture allows trained models to be integrated seamlessly into the game engine, enabling NPCs to react intelligently to player actions and exhibit more dynamic combat strategies. The results show clear improvements in responsiveness, adaptiveness, and player challenge compared to traditional rule-based NPCs, validating the effectiveness of the proposed design and implementation. The project also demonstrates the advantages of using a modular pipeline, where training, evaluation, and deployment remain independent yet well-connected stages. This separation ensures that updated models or improved training methods can be integrated without altering the core game logic. As a result, Neo NPC remains flexible for future enhancements and scalable across different game genres and mechanics Furthermore, the comparative results between rule-based NPCs and reinforcement- learning-driven NPCs show a clear improvement in challenge, engagement, and unpredictability. The system proves that even with limited inputs and simplified environments, reinforcement learning can generate believable behaviours that significantly elevate the overall gameplay experience. Neo NPC, therefore, acts as a strong foundation for more advanced AI-driven game designs.

VII. ACKNOWLEDGEMENT

On presenting the Major Project Phase-1 report on "NEO NPC using Deep Learning". We feel great to express our humble feelings of thanks to all those who have helped us directly or indirectly in the successful completion of the project work. We would like to thank our respected guide Dr. Sankhya N Nayak, Associate Professor, Dept. of CS&E, and our major project phase 1 coordinators Mr. Vedananda D E, Assistant Professor, Dept. of CS&E and Mrs. Radhika S K, Assistant Professor, Dept. of CS&E, who helped us a lot in completing this task, for their continuous encouragement and guidance throughout the project work. We would like to thank Dr. Jalesh Kumar, Professor and Head of the Dept. of CS&E, JNNCE, Shivamogga and Dr. Y Vijaya Kumar, the Principal, JNNCE, Shivamogga for all their support and encouragement. We are grateful to the Department of Computer Science and Engineering and our institution Jawaharlal Nehru New College of Engineering and for imparting us the knowledge with which we can do our best. Finally, we would also like to thank the whole teaching and non-teaching staff of Computer Science and Engineering Department.

REFERENCES

[1] Natalia Khan, N. Sabahat, "AI Chatbots in Gaming Experience Transformation," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 9, Issue 1, 2023.
[2] Kun Shao, Yufeng Yuan, Jie Zhang, Dongbin Zhao, "A Survey of Deep Reinforcement Learning in Video Games," Computational Intelligence and Neuroscience, Vol. 2022, Article ID 7061722, 22 pages.
[3] WeMade, NVIDIA, "AI-Powered NPCs in MMORPG: Asterion Case Study," NVIDIA Technical Report, 2025.
[4] Tejal Kadam, Archisha Chandel, Akanksha Dubey, Rahul Patil, "Reinforcement Learning Bot for Mario," International Journal of Scientific Research in Engineering and Management, Vol. 4, Issue 9, 2020.
[5] Natalia Curado Carneiro, "FSM-Based Behaviour Modeling for Mario," International Journal of Advanced Computer Science and Applications, Vol. 12, No. 10, 2021.
[6] Pedro Almeida, Vítor Carvalho, Alberto Simões, "Reinforcement Learning as an Approach to Train Multiplayer First-Person Shooter Game Agents," Applied Sciences, Vol. 14, Issue 4, 2024.
[7] Shouren Wang, Zehua Jiang, Fernando Sliva, Sam Earle, Julian Togelius, "Enhancing Player Enjoyment with a Two Tier DRL and LLM Based Agent System for Fighting Games," arXiv preprint, arXiv:2504.07425, April 2025.
[8] Chen Zhang, Qiang He, Yuan Zhou, Elvis S. Liu, Hong Wang, Jian Zhao, Yang Wang, "Advancing DRL Agents in Commercial Fighting Games: Training, Integration, and Agent-Human Alignment," IEEE Transactions on Games, Early Access, 2024.
[9] Iveta Dirgová Luptáková, Martin Kubovčík, Jiří Pospíchal, "Playing Flappy Bird Based on Motion Recognition Using a Transformer Model and LIDAR Sensor," Sensors, Vol. 24, No. 2, 2024.
[10] Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, Jacek Mańdziuk, "Monte Carlo Tree Search: A Review of Recent Modifications and Applications," Artificial Intelligence Review, Vol. 56, 2023, pp. 841–907.
[11] Muhammad Bambang Firdaus, Alan Zulfikar Waksito, Andi Tejawati, Medi Taruk, M. Khairul Anam, Akhmad Irsyad, "FSM for Retro Arcade Fighting Game," Journal of Physics: Conference Series, Vol. 2632, 2023.
[12] Natalia Curado Carneiro, Tejal Kadam, Archisha Chandel, Rahul Patil, "Integrating FSM and Reinforcement Learning for Hybrid NPCs," International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 7, No. 3, 2022.
[13] Afzal Hussain, Abdul Rauf, Mian Rizwan, "A Survey on Unity Game Engine: Architecture, Features, and Its Applications," Indonesian Journal of Electrical Engineering and Computer Science, Vol. 29, No. 2, 2023.