

The Progress and Trend of Intelligent NPCs in Games

Haodong Du^{1,a,*}

¹*School of Software, Huazhong University of Science and Technology, No. 1037 Luoyu Road,
Hongshan District, Wuhan, Hubei, China*

a. 3174410168@qq.com

**corresponding author*

Abstract: With the rapid development of artificial intelligence, non-player characters (NPCs) in games have become increasingly intelligent. NPCs undoubtedly enhance players' gaming experience by allowing for better adjustments to game difficulty and narrative development through the optimization of NPC behavior logic. However, research on intelligent NPCs in games remains underdeveloped, highlighting the need for more systematic studies. To address this issue, we review the literature on intelligent NPCs and summarize the progress and trends of intelligent NPCs in games. We compare classical game NPCs with intelligent NPCs and conclude that continued development is essential for advancing intelligent NPCs. We should refine the scoring standards for NPC behaviors based on the content of different games, encouraging AI to exhibit human-like behaviors and enhance authenticity. Moreover, we should create an ideal environment for AI training to make the process more efficient and the results more effective. Furthermore, we propose the goals and challenges in the current development of NPCs. Finally, we summarize the progress and trends of intelligent NPCs in games to provide insights for future researchers.

Keywords: intelligent NPCs, Non-player Characters, AI in games

1. Introduction

Nowadays, games can be recognized as a fascinating art form due to the effort and painstaking conception of the designers behind them [1]. To improve games, artificial intelligence (AI) in the gaming industry is developing at an unprecedented rate. Among these, game NPCs with intelligent characteristics have become a major research focus, garnering significant attention from many researchers. However, in academia, there is still a notable lack of systematic analysis and research on various types of game NPCs. To address these issues, we conduct an in-depth review of the development of game NPC behavior logic and make a detailed comparison of different representative decision-making methods for NPCs. Our aim is to fill gaps in this research area and provide a reference for subsequent studies.

2. Literature Review of Game NPC

2.1. Classical Game NPC

The decision-making methods of traditional game NPCs include several approaches. The vast majority of them select predefined behaviors. Two of the most representative methods are decision-

making based on finite state machines (FSM) [2] and decision-making based on behavior trees (BT) [3]. The game industry began in the 1960s, and finite state machines were one of the main algorithms in early games. As a result, they have existed for nearly 60 years, making them a relatively primitive approach [4]. With the development of games, problems arose. When a game has an excessive number of states, it becomes difficult to manage them all with FSM. As a result, the behavior tree was introduced to solve such problems [5]. Obviously, both of them seem a bit old-fashioned. Nevertheless, there are still excellent games that use these methods to deliver engaging experiences to players.

Today, the gaming industry is evolving rapidly, and the market size is also expanding. These factors have prompted game developers to continually improve and optimize game NPCs. Since the 21st century, various types of games have emerged, driving the evolution of artificial intelligence in gaming. After the well-known game Pac-Man [6] established the basic AI behavior model for NPCs, games such as Street Fighter, Final Fantasy, Dark Souls, and the GTA series—featuring distinctive AI to support NPC behaviors—were released. NPCs in these games utilized behavior trees, state machines, and decision tables to respond to players. However, these rigid decision-making methods have several shortcomings. Clearly, in the evolving gaming landscape, traditional NPCs can no longer meet players' demands for more realistic and engaging experiences.

2.2. Intelligent Game NPC

With the continuous development of artificial intelligence, several algorithms have emerged, including:

1. Supervised learning [7]: In supervised learning, algorithms use labeled training data for learning. Labels provide correct output information about input information. Common algorithms include linear regression, logistic regression, decision trees, support vector machines, and random forests.

2. Unsupervised learning: Unsupervised learning processes unlabeled data and discovers hidden patterns, structures, or relationships in the data. Common algorithms include clustering, principal component analysis, and association rule learning. Compared to supervised learning, it can be an option when you have a large dataset and labeled data is unclear [8].

3. Semi-supervised learning: Combining learning with a small amount of labeled data and a large amount of unlabeled data is more suitable for practical use.

4. Deep learning: Including both supervised and unsupervised learning, it uses deep neural networks to learn complex patterns and representations in data. Deep neural networks are composed of multiple layers of neurons and can automatically extract features from raw data and perform advanced abstraction and reasoning. Deep learning shows promising prospects in visual recognition, but it still faces challenges, including poor interpretability [9]. Dian Lei and his colleagues conducted a deeper exploration and explanation of the black box in deep learning [10].

5. Large language models (LLMs): Deep learning models with a large number of parameters and strong computing power. These models are trained on large-scale data and can learn rich knowledge and complex patterns, with high performance and generalization ability. LLMs are effective at processing human language and responding to it, making the application of LLMs to game NPCs a potential approach. As Humza Naveed et al. have thoroughly studied and introduced LLMs in their paper, A Comprehensive Overview of Large Language Models, it provides a valuable reference for understanding the capabilities and applications of LLMs in this context [11].

6. Reinforcement learning (RL): Allows an agent to interact with the environment and learn an optimal strategy method, which can obtain the maximum accumulated reward in the long-term interaction process. The main algorithms include Q-learning, deep Q-network, and policy gradient algorithms, etc. In the context of game NPCs, RL can be utilized in several aspects. NPCs can learn optimal behaviors through RL algorithms to adapt to different game situations and player actions. In

the paper Deep Reinforcement Learning: An Overview, Yuxi offers a solid foundation for understanding the fundamental concepts [12].

With the development of artificial intelligence technology, game developers are increasingly dissatisfied with previously inflexible NPCs and are beginning to explore the application of machine learning to NPCs in games. In the early 21st century, some researchers began to apply reinforcement learning to the behavior control of game NPCs. Initially, researchers often used reinforcement learning algorithms to train agents to play games. Although this research mainly focused on game player agents, it inspired the application of reinforcement learning to NPC behavior.

Later, research on NPCs has primarily focused on three aspects: 1. Behavior learning; 2. Strategy learning; and 3. Interaction between NPCs. Behavior learning enables NPCs to perform various tasks in different environments through self-learning, such as attacking, fleeing, waiting, etc. Strategy learning enables NPCs to make judgments based on their accumulated experience when facing complex choices, thus obtaining results that are closest to the target effect. The interaction of NPCs puts NPCs and players on an equal footing in the game, rather than all NPCs merely serving the players. From the perspective of players observing NPCs, this enhances the authenticity and enjoyment of the game.

With the development of deep learning technology, researchers began combining deep learning with reinforcement learning to improve the performance and intelligence of intelligent NPCs. Deep reinforcement learning algorithms can automatically learn feature representations in games, enabling NPCs to better understand the game environment and make decisions.

At this stage, famous deep reinforcement learning algorithms, such as Deep Q-Network (DQN) [13], Policy Gradient Algorithm [14], Proximal Policy Optimization (PPO) [15], and Trust Region Policy Optimization [16], are widely used in the behavior control of game NPCs. These algorithms have achieved remarkable results in some games, demonstrating the huge potential of intelligent NPCs.

3. Analysis of Game NPC

3.1. Analysis of Classical Game NPC

Pac-Man, as a classic game, exhibits several remarkable characteristics.

First, the ghosts' states are clearly divided into three main types: patrolling, chasing, and evading. Moreover, several conditions govern the transitions between states. For instance, when the distance between Pac-Man and the ghosts decreases or when Pac-Man's state changes significantly, the ghosts transition from the patrolling to the chasing state. When Pac-Man becomes invisible and the ghosts lose their target, or when a power pellet confuses them, the ghosts will return from the chasing to the patrolling state. Once Pac-Man consumes a power pellet, the ghosts immediately enter the evading state. After remaining in the evading state for a certain period, they will return to the patrolling state for reassessment. These behavioral patterns are clear and demonstrate a degree of intelligence. Furthermore, the strategic placement of power pellets significantly enhances the game's appeal. Power pellets are typically placed in key positions, requiring players to weigh risks and rewards and decide whether to take risks to obtain them, adding strategic and challenging elements to the game.

Second, the rules of Pac-Man are simple, and its environmental design is relatively basic. However, it is precisely this simplicity that allows the game to run smoothly without occupying excessive space or consuming significant time, even when repeatedly traversing the bottom of the behavior tree. This characteristic ensures that the game runs smoothly on various devices, whether an old game console or a low-powered computer.

Compared to Pac-Man, Dungeon & Fighter shares similar characteristics. The state classification of monsters is clear and diverse, primarily including the normal, alert, attack, and berserk states.

When players approach, monsters transition from the normal to the alert state. If players continue to approach or take attack actions, monsters will enter the attack state. When monsters are subjected to specific attacks or their health drops to a certain threshold, they enter the berserk state. At the same time, guiding NPCs offer varying assistance or trigger different storylines based on players' progress. Although NPCs 'intelligently' provide different help or trigger different storylines according to the progress of players, these scenarios are all anticipated by game designers. The behavior of NPCs is pre-written in the code according to specific environmental conditions. The main reason for the popularity of the game is the comprehensive consideration and ingenious design of the producers. They anticipate the possible situations players may encounter and the reasonable reactions of NPCs in different situations. These designs by the producers make the behavior of NPCs seem very intelligent.

In general, after analyzing the original NPC behavior logic method, we find that game NPCs without the use of artificial intelligence have the following characteristics:

Advantages:

1. The states are simple and often switch only between a small number of predefined states.
2. The conditions for state transitions are simple.
3. The behavior is reasonable and carefully designed by the producer.
4. The storyline is guided, ensuring that the development often converges to the same result or a small set of results, which are the outcomes the producer intends for the players to achieve.

Disadvantages:

1. The storyline is linear. During the process of completing the storyline, players often have only a limited number of actions to choose from. This approach is also used to achieve result convergence.
2. The reactions of NPCs are repetitive. As players advance through the storyline, the behavior or dialogue of NPCs is often identical. This leads to exploration during the first playthrough, but when replaying the game, if special storylines are not triggered, the experience becomes much more monotonous. This is a primary pain point in these games. Once players become familiar with the various storylines and the fixed reactions of NPCs, the outcomes remain predictable, making subsequent playthroughs less engaging. As a result, when playing the game for the second time or more, the exploration, fun, and challenge are significantly diminished.

3.2. Analysis of Intelligent Game NPC

In their research "Research on Multi-NPC Marine Game AI System based on Q-Learning Algorithm," Fanmo Meng and Cho Joung Hyung conducted an in-depth exploration of the multi-NPC marine game AI system based on the Q-learning algorithm [17]. The research highlighted that traditional game NPC behavior trees suffer from problems such as complex manual coding and a lack of learning ability. Additionally, it analyzed the limitations of the Q-learning algorithm, which, while advantageous, is prone to falling into local optima. By introducing the simulated annealing algorithm to optimize the Q-learning algorithm, the performance was enhanced, and the NPC behavior tree was further refined, making its decision-making more closely resemble human thinking. This work provides an important reference and inspiration for research on intelligent NPCs.

MOBA (Multiplayer Online Battle Arena) games refer to a type of game with a standard 5v5 mode where players battle in real-time to protect their own defensive towers while pushing down the opponent's defensive towers through team strategies. Each hero in these games has multiple choices and gameplay methods in actual combat, and the behavior of each hero impacts the overall development of the game. As a result, the mechanics of MOBA games are quite complex. In the MOBA game Honor of Kings, a mode named "JueWu Human-Computer" was launched, allowing five real players to have real-time battles with trained AI. Once this mode was introduced, it received extensive attention from players. It marked the first time players experienced the "wisdom" of

artificial intelligence. Generally, players found that the behavior and decision-making of JueWu AI in the game surpassed most players, which made the difficulty of this mode very high and also signaled the success of JueWu AI. In fact, JueWu AI originated from an experiment applying AI to MOBA games. Ye Deheng, Chen Guibin, and others from institutions such as Tencent AI Lab and Tencent Timi Studio proposed JueWu-SL, an AI program based on supervised learning [18]. In the experiment, they used data from the top 1% of players in the game to construct a dataset, employing deep convolutional and fully connected neural networks. They set evaluation indicators such as win rate, kills, damage conversion ratio, damage taken, healing amount, and tower damage for a comprehensive assessment. The result was that JueWu-SL achieved a very high win rate when battling against artificial intelligence developed with other algorithms. Even the practical performance of JueWu-SL surpassed that of the top 1% of players. Their experimental results highlight the feasibility and superiority of supervised learning in training intelligent NPCs for MOBA games.

First-person shooter (FPS) games, such as Counter-Strike, are also extremely popular. In CS, the main gameplay revolves around managing the economy, purchasing guns and items. The terrorist side wins by killing all the police officers or detonating a bomb at the bomb site within the specified time, while the police must kill all the terrorists and ensure the bomb has either not been planted or has been defused. The map contains two bomb sites, with police and terrorists spawning at their respective points. Players must plan economically and cooperate tactically, making the game complex with many possible outcomes. The game also features a Deathmatch mode, where players spawn at random positions, and all other players are enemies. The rules for this mode are relatively simple. The AI performance in CS differs significantly across these two modes. Due to the simpler rules of the Deathmatch mode, the NPC only needs to move and shoot at other players within their line of sight. However, in the classic mode, which requires cooperation and strategy, the NPC appears somewhat clumsy. The article “Counter-Strike Deathmatch with Large-Scale Behavioural Cloning” provides a study on the performance of NPCs in the Deathmatch mode of CS [19]. To make the AI agent reach a high level of gameplay and exhibit a human-like gaming style, the researchers adopted a two-stage learning approach: 1) gathering a large amount of data from real-player games on public servers for training, and 2) fine-tuning the model on a small, clean dataset of expert demonstrations. This approach significantly increased data efficiency, making it much more effective than pure reinforcement learning (RL) algorithms. The results of the experiment show that when the collected data is relatively complex, providing an idealized classic environment for artificial intelligence has a significant accelerating effect on training.

Table 1: Comparison Analysis of NPC Algorithms in Games

No.	Research	Algorithm	Game
1	Application of Artificial Intelligence Techniques in Ms. Pac-Man Game: A Review	Monte Carlo Tree Search Algorithm	Pac-Man
2		FSM, Behavior Trees (Decision Trees)	Street Fighter
3		Program Design	Dungeon & Fighter
4	Supervised Learning Achieves Human-Level Performance in MOBA Games: A Case Study of Honor of Kings	JueWu-SL	Honor of Kings

Table 1: (continued).

5	Research on Multi-NPC Marine Game AI System Based on Q-Learning Algorithm	Q-Learning Based on Simulated Annealing	Multi-NPC Marine Game
6	Counter-Strike Deathmatch with Large-Scale Behavioral Cloning	Two-Stage Behavioral Cloning Methodology	Counter-Strike: Global Offensive

By comparing NPCs under traditional and new algorithms, we draw the following conclusions from Table I: NPCs under traditional algorithms exhibit single behaviors and repetitive actions in repeated games. Although this may detract from the realism of the game, human-designed language and action sequences are essential for guiding the main plot. On the other hand, new intelligent NPCs are better suited for player-versus-player games with standardized rules. The current challenge lies in seamlessly combining traditional and new algorithms to create NPCs that emulate human-like thinking, possess personality and emotion, and can freely interact with players based on experience and logic. While GPT is gradually approaching this goal, it still faces challenges such as the need for large datasets for training, reliance on real-life data rather than a game-specific context, and lack of plot-driven guidance, all of which require further development. At the same time, the ethics of AI in games must not be overlooked. When AI manages game NPCs entirely, it may use violent, pornographic, and other controversial content to attract more players. As AI becomes sophisticated enough to bypass developers' restrictions on interactive content, it could negatively impact the psychological well-being of teenagers and children. Therefore, it is crucial to focus on the ethics of AI and continue to monitor, regulate, and research its development [20].

4. Conclusion

The future of NPC behavior algorithm design in games should focus on training AI to simulate human game thinking rather than machine thinking. While utilizing large datasets for training, it is important to refine evaluation criteria and develop a comprehensive AI behavior scoring system that encourages AI to explore the optimal solution through continuous training. AI should be encouraged to be exploratory. It does not need to completely mimic the common patterns of human thinking in data; instead, making small, creative changes is more conducive to discovering the real overall optimal solution. Given the complexity of real-player game data, providing a relatively ideal environment for AI training is essential, one that can be derived from a better real-player game context. This design trend will help advance the development of NPC behavior algorithms, leading to a more immersive, realistic, and challenging gaming experience for players.

References

- [1] X. Fan, J. Wu, and L. Tian, "A Review of Artificial Intelligence for Games," pp. 298–303, 2020, doi: 10.1007/978-981-15-0187-6.
- [2] D. Lee, S. Member, and M. Yannakakis, "Principles of Finite State Machines," *Methods*, vol. 84, no. 8, 1996.
- [3] L. Ruifeng, W. Jiasheng, Z. Haolong, and T. Mengfan, "Research progress and Application of Behavior Tree Technology," *BESC 2019 - 6th Int. Conf. Behav. Econ. Socio-Cultural Comput. Proc.*, pp. 19–22, 2019, doi: 10.1109/BESC48373.2019.8963263.
- [4] D. Jagdale, "Finite State Machine in Game Development," *Int. J. Adv. Res. Sci. Commun. Technol.*, no. October, pp. 384–390, 2021, doi: 10.48175/ijarsct-2062.
- [5] Y. A. Sekhavat, "Behavior Trees for Computer Games," *Int. J. Artif. Intell. Tools*, vol. 26, no. 2, pp. 1–28, 2017, doi: 10.1142/S0218213017300010.

- [6] P. Rohlfshagen, J. Liu, D. Perez-Liebana, and S. M. Lucas, "Pac-Man conquers academia: Two decades of research using a classic arcade game," *IEEE Trans. Games*, vol. 10, no. 3, pp. 233–256, 2018, doi: 10.1109/TG.2017.2737145.
- [7] T. Hastie, R. Tibshirani, and J. Friedman, "Springer Series in Statistics," *Elem. Stat. Learn.*, vol. 27, no. 2, pp. 83–85, 2009, doi: 10.1007/b94608.
- [8] B. Mahesh, "Machine Learning Algorithms - A Review," *Int. J. Sci. Res.*, vol. 9, no. 1, pp. 381–386, 2020, doi: 10.21275/art20203995.
- [9] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [10] D. Lei, X. Chen, and J. Zhao, "Opening the black box of deep learning," pp. 1–27, 2018, [Online]. Available: <http://arxiv.org/abs/1805.08355>
- [11] H. Naveed et al., "A Comprehensive Overview of Large Language Models," 2023, [Online]. Available: <http://arxiv.org/abs/2307.06435>
- [12] S. S. Mousavi, M. Schukat, and E. Howley, "Deep Reinforcement Learning: An Overview," *Lect. Notes Networks Syst.*, vol. 16, pp. 426–440, 2018, doi: 10.1007/978-3-319-56991-8_32.
- [13] M. Roderick, J. MacGlashan, and S. Tellex, "Implementing the Deep Q-Network," no. Nips, pp. 1–9, 2017, [Online]. Available: <http://arxiv.org/abs/1711.07478>
- [14] M. Ghavamzadeh and Y. Engel, "Bayesian policy gradient algorithms," *Adv. Neural Inf. Process. Syst.*, pp. 457–464, 2007.
- [15] W. Meng, Q. Zheng, G. Pan, and Y. Yin, "Off-Policy Proximal Policy Optimization," *Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023*, vol. 37, pp. 9162–9170, 2023, doi: 10.1609/aaai.v37i8.26099.
- [16] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 3, pp. 1889–1897, 2015.
- [17] F. Meng and C. J. Hyung, "Research on Multi-NPC Marine Game AI System based on Q-learning Algorithm," *2022 IEEE Int. Conf. Artif. Intell. Comput. Appl. ICAICA 2022*, pp. 648–652, 2022, doi: 10.1109/ICAICA54878.2022.9844648.
- [18] D. Ye et al., "Supervised Learning Achieves Human-Level Performance in MOBA Games: A Case Study of Honor of Kings," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 3, pp. 908–918, 2022, doi: 10.1109/TNNLS.2020.3029475.
- [19] T. Pearce and J. Zhu, "Counter-Strike Deathmatch with Large-Scale Behavioural Cloning," *IEEE Conf. Comput. Intell. Games, CIG*, vol. 2022-Augus, pp. 104–111, 2022, doi: 10.1109/CoG51982.2022.9893617.
- [20] D. Melhart, J. Togelius, B. Mikkelsen, C. Holmgard, and G. N. Yannakakis, "The Ethics of AI in Games," *IEEE Trans. Affect. Comput.*, vol. 15, no. 1, pp. 79–92, 2024, doi: 10.1109/TAFFC.2023.3276425.