

Intelligent Agent and NPC Behavior Modeling: From Traditional Methods to AI Driven Interactive Game Design

Jiansong Zhou^{1,a,*}

¹*Maynooth International Engineering College, Fuzhou University, Fuzhou, China*

a. 832304209@fzu.edu.cn

**corresponding author*

Abstract: In recent years, artificial intelligence (AI) related technologies have been widely applied in the gaming industry, bringing enormous commercial value. Research has shown that non-player characters (NPCs) embedded with AI technology are more intelligent, popular among players, and can make them feel challenged and amused. Major game companies have established their own AI studios to invest energy in designing more intelligent NPCs, making the NPCs deployed in games increasingly superior. However, there is still a lack of comprehensive induction and summary of the behavior modeling design methods for NPCs. Therefore, this article focuses on providing a more comprehensive summary of this field, by searching and comparing relevant examples in existing game and research results of NPC design, summarizing and generalizing different types of methods, then analyzing and comparing the advantages and disadvantages of traditional algorithms and AI algorithms, as well as their usage scenarios, and proposing potential methods for the future in the end.

Keywords: Reinforcement learning, artificial intelligence, game design, NPC behavior design.

1. Introduction

In recent years, the scale of the artificial intelligence industry has shown a rapid upward trend. The "White Paper on the Development of the Global Artificial Intelligence Industry" shows that the global scale of the artificial intelligence industry will reach 707.8 billion US dollars in 2023, a year-on-year increase of 19.3%. As a large number of industries begin to deploy AI, the gaming industry, as a typical representative, can be said to have achieved amazing results. Currently, AI has become a new standard for research and development in the gaming field. Since 2016, many game companies represented by Tencent in China have started to establish AI studios to achieve more intelligent gaming. As NPC design is an important part of game development, manufacturers are also committed to achieving more complex and intelligent NPCs to attract more players.

According to a survey, 77.4% of surveyed players believe that NPCs using AI technology are more interesting and challenging. They believe that AI plays an important role in games, and over 96% of respondents are more willing to play games that use AI technology. 77.4% of respondents consider AI as an important factor when purchasing games.[1] It can be seen that how to make NPC's behavior decisions more intelligent has become a key issue to consider in game development.

Since the development of the gaming industry, game designers have adopted various methods to control the activities of NPCs. NPC behavior currently involves two major technologies: pathfinding navigation and behavioral interaction. Based on different periods and game types, it can be roughly

divided into three categories: 1. Traditional algorithms composed of behavior trees and automata, Just Cause 3: There is a system that selects priority positions so that NPCs can execute tasks most effectively based on factors such as player position, obstacle presence, and distance from allies.[2] 2. Based on Reinforcement Learning (RL) and decision tree algorithms, Cyberpunk 2077 is a representative work of a single player development world game, in which NPCs use decision trees to handle various situations, including interactions with players and reactions to events that occur in the game world. 3. Improved hybrid algorithms based on different game characteristics. For example, the first AI program based on supervised learning, JueWu-SL, was proposed for the MOBA game Honor of Kings. It was implemented in a neural network in a supervised and end-to-end manner, and demonstrated a competitive level comparable to high-level players in the standard 5v5 game.[3]

Due to the fact that most existing articles focus on technology improvement and research, a comprehensive summary of this field is still lacking. Therefore, this article summarizes and compares the above technologies, cites relevant existing examples for improving the design of different types of games, and proposes potential directions that are currently in the early stages. I hope to provide some references for professionals in related technologies.

2. Mainstream technical methods

2.1. Traditional algorithm design approach

NPCs have always been developed in traditional algorithmic forms in linear games. NPCs often serve as guides for players' tasks, and after completing related tasks, they no longer appear in the player's subsequent gameplay. Traditional algorithms often consist of three parts: firstly, the visual cone, which constructs a sensory area in front of the NPC's eyes (Figure 1.). [2] For example, in Tom Clancy's The Division, the opponent's visual sensors are more complex. In addition to a basic visible area that resembles an elongated hexagon, the enemy also has a larger hexagon that simulates the surrounding field of view. At the same time, space is placed behind to simulate sensations, or light is emitted from the NPC's eyes and reflected to perceive the player's position and status (Figure 2.). [2] Sound perception has a similar function, but with different media and processes. Next is the behavior decision algorithm, mainly consisting of behavior trees and automata, which determine how the behavior state of the NPC will transition based on the current state of the NPC and the conditions of the transition process.

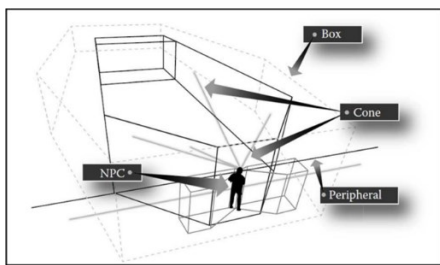


Figure 1: Visual Space Construction of NPCs.[4]



Figure 2: NPCs perceive player position through visual cones [4]

Finally, there is the navigation and pathfinding of NPCs, whose behavior often depends on the state of the environment. When players have not yet participated in the tasks involved, NPCs often have predefined navigation maps for movement, with different points in the map representing the corresponding activities performed by NPCs at that position. When NPCs participate in tasks, they often need to go to destinations that are not in the pre-defined navigation map. In this case, pathfinding algorithms are needed to help NPCs quickly reach their destinations. Currently, the most popular

pathfinding algorithm is the A* algorithm, which calculates the cost of each path to find the optimal route (Figure 3.). Although the above algorithms have been able to complete NPC activities on a two-dimensional level, they are still unable to meet NPC activities in three-dimensional space, such as vertical building or mountain surface climbing. To address this issue, Navigation mesh (Navmesh) was designed for improvement. It consists of a series of polygons that represent the areas where the character can walk. This enables the 3D map to be two-dimensional, and the A* algorithm can be used to find suitable paths on the 2D plane (Figure 4.).

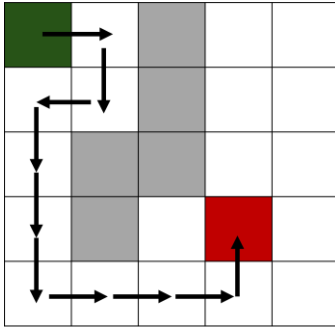


Figure 3: A* Algorithm Movement Diagram.

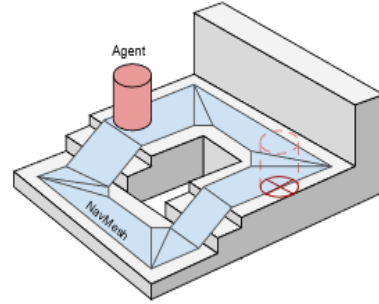


Figure 4: Schematic diagram of Navmesh walkable area.[5]

2.2. Hybrid algorithm enhancement

In the early 2000s, many games represented by StarCraft began using RL to improve NPC performance, but there were many flaws. With the improvement of hardware performance, more and more companies combined it with other algorithms to improve many problems. Here are two examples of games designed to improve NPC navigation, pathfinding, and action behavior.

The first example is Tencent's collaboration with the University of Science and Technology of China to improve game Arena Breakout AI. [6] The game revolves around large-scale real-time online material competition as its core gameplay, and the biggest challenge for NPCs is to make them move reasonably on a large map. By establishing a navigation network with materials as heat and using other game related variables related to materials as parameters for RL, this cannot meet the requirements. To cope with the challenges faced at any time, adversarial AI is trained separately to switch states and engage in confrontations at any time during the process of reaching the destination. In order to improve the performance of AI, reward training for human like behavior and continuous self-adversarial enhancement have been added. The following summarizes the prominent improvements in this example, starting with the reward design:

$$r = r_T + r_d + r_{aux} \quad (1)$$

Divided into final reward r_T , navigation reward r_d , and auxiliary reward r_{aux} . The final reward and navigation reward are used to encourage NPCs to engage in confrontation with enemies, representing the outcome of the confrontation and the distance to the enemy, respectively. Auxiliary rewards are used to reward NPCs when they perform more human like player actions, under this constraint, making their behavior more like that of ordinary players.[6]

Furthermore, Navigation Mesh and Shooting rule enhanced Reinforcement Learning (NSRL) has improved the shortcomings of Navmesh. NSRL combines Navmesh and shooting rules to enhance the performance of Deep Reinforcement Learning (DRL). This design allows agents to maintain a balance between precise aiming and shooting of targets in global navigation. NSRL uses DRL models to predict when to enable navigation grids, providing diverse behaviors for gaming AI.[6] However,

Navmesh may lead to performance issues in large scenarios as it requires the creation of a large number of meshes and may not handle complex terrain and obstacles well (Table 1.).

Table 1: Comparison of structural performance between NSLR and Navmesh.

	NSLR	Navmesh
Algorithm composition	DRL & Navmesh	Static grid generation & path planning algorithm
Operation process	Using Navmesh or other tactical decisions based on dynamic predictions of the environment	Focusing on path planning, lacking support for complex operations such as tactical behavior
Resource Consumption	Dynamic use to reduce resource waste	Global usage wastes a significant amount of resources

Different from two dimensional (2D) rendering of three dimensional (3D) maps, another AI agent WILD-SCAV uses a depth map (which records the distance of each pixel in the scene relative to the observation point (usually the camera). In 3D space, depth maps provide a way to convert 2D images into 3D information by assigning a depth value to each pixel, which represents the straight-line distance from that point to the camera. Create a global 3D occupancy grid map using game variables such as proxy location and camera direction.[7] The article also mentions the use of Simultaneous Localization and Mapping (SLAM) method to generate occupancy maps, which is a common technique in the fields of robotics and autonomous driving, used to simultaneously construct environmental maps and locate agents themselves.

The second example is Blade & Soul, which features real-time 1v1 combat as its core gameplay. Its main challenge is how to manipulate NPC behavior and plan complex action state combinations to achieve the level of ordinary or even professional players. This type of game requires consideration of many skill combinations, cooldown time, and multi-directional movements. A new self-play learning method has been introduced: Self Play Curriculum Learning, which improves the generalization ability of agents by creating agents with different combat styles (attack, defense, and balance) and placing them in the same opponent pool to compete. [8] And the use of data skipping technology greatly saves the waste of resources in NPC action calculation. [8] An AI agent for the game was developed based on the following design process. As shown in (Figure 5):

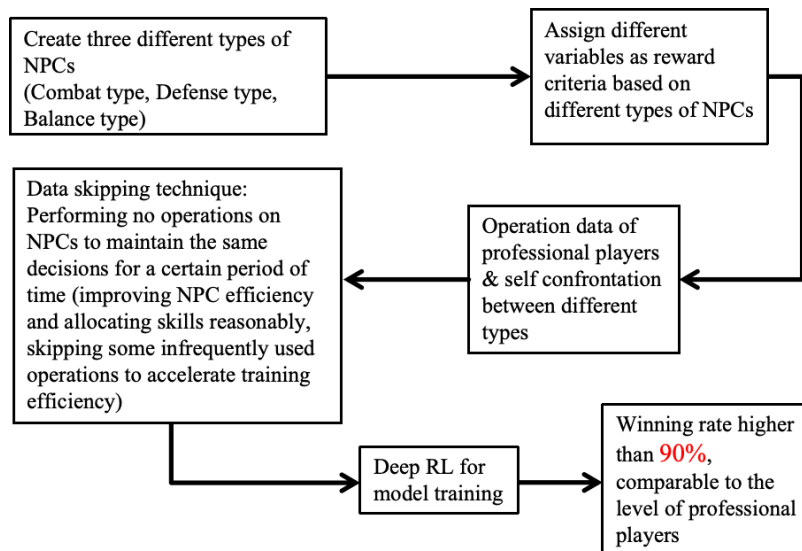


Figure 5: Brief Introduction to the Creation Process of Blade & Soul AI Agent

The results of the experiment are satisfactory, with an NPC winning rate of up to 90%, comparable to professional players, and able to meet players' pursuit of challenge.

Due to limited space, this article only provides specific examples of improvements for two different aspects. However, it can be seen that combining different decision-making methods can create a more comprehensive and effective NPC decision-making approach.[9] This is clearly reflected in NPC's decision-making ability, adaptability to complex environments, resource utilization, generalization ability, and scalability. Here are a few examples cited, but they will not be elaborated further. Abiyev et al. proposed a decision-making method that combines fuzzy logic and neural networks for robot agents in simulated robot soccer games. [9] Dey and Child combined the behavior tree and reinforcement learning proposed method QL-BT to facilitate AI practitioners in deciding when to execute each branch of the logical tree.[10]

Finally, summarize and compare the above algorithms as shown in Table 2.

Table 2: Comparison and Induction of Different Algorithms.

Algorithm Comparison	Traditional algorithm	Original RL/Decision Tree Algorithm	Modern RL Hybrid Algorithm
Mainstream methods	Behavioral Automata/Behavior Tree & Visual Light Cone	RL/Decision Tree	Rule based RL & in-game variables & heatmap & self play
Application scenarios	Small independent game (linear game)	Traditional strategy games, traditional 3A games	Large real-time adversarial game with complex combat operations
Advantage	Build simple and fast	Training is relatively easy	Targeted, with variable states and a sense of realism
Drawback	There are many state variables and omissions, making it difficult to adapt to changing environments	NPC intelligence is weak, affecting players' playability	Large amount of data, tedious training

3. Limitations and Shortcomings

Although NPCs currently exhibit some intelligence, they still face many challenges.

The AI for specific games cannot be generalized: Currently, most of these NPCs are designed by game designers, and NPCs often have no behavioral response during non-task stages, only a rigid model placed in the scene or repeating a few predetermined sentences. At the same time, the game cannot be easily transferred to other games or maps for use. This means that AI's generalization ability and adaptability are limited. [11]

In large-scale real-time adversarial games, NPCs still have a significant gap compared to human players, often only serving as mobile resource packs that cannot pose challenges and threats to players, resulting in the game becoming boring.

The inefficiency of computation: Many machine learning techniques require a large amount of computing resources during the learning and classification stages, which may be impractical in real-

time digital games. Game AI is often limited by the computing resources allocated to the game, and there are often various resource occupations during game execution, which may limit the complexity and performance of learning algorithms.[12] Games that require agents to plan strategies for longer periods of time typically involve complex sequential decisions that require remembering and utilizing a large amount of prior information to achieve long-term goals. Deep Q Network (DQN) technology still faces challenges when dealing with games that require complex long-term planning.[13]

Limitations of evaluation methods: Most AI evaluations are based on their win rate against a limited number of professional human players, which may be too crude and may exaggerate the actual level of AI. In addition, for games with incomplete information, existing evaluation methods may not accurately reflect the performance of AI.[11]

4. Suggestions for future research

Firstly, for the emotion engine: game developers define goals for each NPC, which are the states that the NPC wants to achieve. Developers label events in the game as having a positive or negative impact on these goals. According to the Ortony, Clore, and Collins (OCC) emotion model, evaluate events and generate emotions related to NPC targets. Track the goals and beliefs of each NPC, and update emotional states based on game events. [14]

It is based on psychological theory, and through emotional expression and behavior, makes NPC reactions more unpredictable and diverse. And it can process the emotional states of a large number of NPCs in real time, suitable for large-scale games. And it can be designed as an independent module that works independently from the AI system of the game itself, easily integrated into different games without the need for a deep understanding of emotional evaluation theory. At the same time, emotional computing can also be applied to dynamically adjust the difficulty of game NPCs to enhance players' active participation and motivation.[15]

Then switch to the brain: the player creates and trains an AI that can generate a "brain" tailored to different situations. Essentially, it is an artificial neural network trained through neuroevolutionary algorithms, allowing NPCs to switch between different scenarios to meet different behavioral requirements.[16] Although the application of "brain switching" and neural evolution in games is still in a relatively early stage, they have enormous potential to play an important role in future game design and artificial intelligence fields.

Simultaneously developing specialized frameworks and tools can help developers make NPC decisions more easily. Currently, some existing frameworks, such as AI Graph Editor and F.E.A.R SDK can better help facilitate the implementation of NPC decision-making.[17]

5. Conclusions

Due to fixed directions, differences in classification, and limitations in research scope, this type of research may miss out on some relevant major studies and specific concepts, as well as insufficient retrieval from databases and search platforms.

This article aims to provide a comprehensive summary of the relevant technologies for NPC behavior decision design, and analyze the design ideas, performance differences, and usage scenarios of traditional algorithms and reinforcement learning algorithms. At the same time, it proposes a promising technology direction that is still in its early stages. I hope to provide a reference for researchers to design research directions.

At present, games have become a testing ground and incubator for building universal AI agents, constantly driving the update and iteration of decision-making AI technology. The application of AI models in the motion behavior and decision-making of game NPCs is not only for entertaining players, but the design methods can also be applied to various fields such as autonomous driving, robot

development, image recognition, navigation and positioning. It can be said that the significance of studying the application of AI in games is self-evident.

References

- [1] Leonardo, J. S., Brandon, N. M., & Neil, E. S., et al. (2024). *Analyzing AI and the Impact in Video Games*. 2022 4th International Conference on Cybernetics and Intelligent System, DOI: 10.1109/ICORIS56080.2022.10031590
- [2] Volodymyr, L., Mykola, T., & Andrii, Y., et al. (2023). *The Working Principle of Artificial Intelligence in Video Games*. 2023 IEEE International Conference on Smart Information Systems and Technologies, DOI: 10.1109/SIST58284.2023.10223491
- [3] Ye, D. H., Chen, G. B., & Zhao, P. L., et al. (2022). *Supervised Learning Achieves Human-Level Performance in MOBA Games: A Case Study of Honor of Kings*, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 33, NO. 3, MARCH 2022, DOI: 10.1109/TNNLS.2020.3029475
- [4] Blognetology. (2022). *Имитация разума: как устроен искусственный интеллект в играх*. Хабр. <https://habr.com/ru/company/netologyru/blog/598489/> (in Russian)
- [5] <https://docs.unity3d.com/cn/2020.3/Manual/nav-InnerWorkings.html>, 2024.10.26
- [6] Chen, Z., Huan, H., & Yuan, Z., et al. (2024). *Training Interactive Agent in Large FPS Game Map with Rule-enhanced Reinforcement Learning*, 2024 IEEE Conference on Games, DOI: 10.1109/CoG60054.2024.10645654
- [7] Fang, Z. Y., Zhao, J., & Zhou, W. G., et al. (2023). *Implementing First-Person Shooter Game AI in WILD-SCAV with Rule-Enhanced Deep Reinforcement Learning*, 2023 IEEE Conference on Games, DOI: 10.1109/CoG57401.2023.10333171
- [8] Inseo, O., Seungeun, R., Sangbin, M., et al. (2022). *Creating Pro-Level AI for a Real-Time Fighting Game Using Deep Reinforcement Learning*, IEEE TRANSACTIONS ON GAMES, VOL. 14, NO. 2, JUNE 2022, DOI: 10.1109/TG.2021.3049539
- [9] Abiyev, R. H., Günsel, I., Akkaya, N., et al (2016). *Robot soccer control using behaviour trees and fuzzy logic*. Procedia Comput Sci 102(August):477–484. <https://doi.org/10.1016/j.procs.2016.09.430>
- [10] Dey, R., Child, C., (2013). *QL-BT: Enhancing behaviour tree design and implementation with Q-learning*. In: 2013 IEEE conference on computational intelligence in games (CIG), IEEE. pp 1–8. DOI: 10.1109/CIG.2013.6633623
- [11] Yin, Q. Y., Yang, J., & Huang, K. Q., et al. (2023). *AI in Human-computer Gaming: Techniques, Challenges and Opportunities*, Machine Intelligence Research 20(3), June 2023, 299–317, DOI: <https://doi.org/10.1007/s11633-022-1384-6>
- [12] Leo, G., Darryl, C., Michaela, B., (2009). *Machine learning in digital games: a survey*, Springer Science+Business Media B.V. 2009, DOI: <https://doi.org/10.1007/s10462-009-9112-y>
- [13] Volodymyr, M., Koray, K., David, S., et al. (2015). *Human-level control through deep reinforcement learning*, LETTER, DOI: <https://doi.org/10.1038/nature14236>
- [14] Alexandru, P., Joost, B., & Maarten, V. S., (2014). *GAMYGDALA: An Emotion Engine for Games*, IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, VOL. 5, NO. 1, JANUARY-MARCH, DOI: 10.1109/T-AFFC.2013.24
- [15] Panagiotis, D. P., & Dimitrios, E. K., (2023). *Game Difficulty Adaptation and Experience Personalization: A Literature Review*, International Journal of Human–Computer Interaction, 39:1, 1-22, DOI:10.1080/10447318.2021.2020008
- [16] Daniel, J., Sebastian, R., & Julian, T., (2017). *EvoCommander: A Novel Game Based on Evolving and Switching Between Artificial Brains*, IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 9, NO. 2, JUNE, DOI: 10.1109/TCIAIG.2016.2535416
- [17] Muhtar, C. U., Kaya, O., (2023). *Non-player character decision-making in computer games*, Artificial Intelligence Review (2023) 56:14159–14191, <https://doi.org/10.1007/s10462-023-10491-7>