Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/15/20230829

How AI evolved with game and implementation of modern AI in game

Bohan Shen

School of Computer Science and Technology, Beijing Jiaotong University, Beijing, 100000, China.

20722015@bjtu.edu.cn

Abstract. For a long time, game was a relatively unrecognized area by academic community, which lacks detailed and sufficient discussion. But with the growth of game industry, game AI has become a heated topic in recent years. As an important and evolving application of AI, there is a need to better discuss the application and future improvement of game AI technologies. This paper introduced history and breakthrough of AI made in game area. And made discussion centered on current implementation of some popular approaches for game AI, followed by the possible future of these technologies. Some new implementations like procedural content generation were then covered to further discuss future implementations of AI in game area. All in all, hot spots and development prospects of this research topic were prospected to enlighten future development of game AI.

Keywords: game AI, supervised learning, reinforcement learning, deep learning.

1. Introduction

AI has been accompanying the game since the appearance of computer games. Eventually, the task of artificial intelligence is to accomplish tasks comparable to human. From the extending content of the game that provide entertainment for players to the AI that can perform better than the best players from human. The challenge of developing game AI has brought advance in computational intelligence, reinforcement learning and other AI methos.

The game design can be described as a process to build and provide player with well-orchestrated game content for players, including virtual characters' behavior, sounds, game mechanics, timing of different events and the entities that directly interact with the player [1]. In recent years, with the involvement of AI in more and more fields of game development, like AI generated video, model, voice and even codes, AI is now not simply the program that controls characters to react to players' movement. The game AI can be defined as the artificial intelligence that helps generate, manage and provide game content for game designers and players. With the growing complexity and diversity of games, AI techniques will be indispensable in the future game development.

While the industrial game production has a strong requirement of commercial and practical effect, the academic game AI methods mainly focus on advanced, but non-scalable approaches with limited paybacks. This gap results in the limited interconnection and exchange between research and industry during last decades and also the lack of recognization by the academic field and general public. However,

DOI: 10.54254/2755-2721/15/20230829

some researchers argued for strengthen in game AI research and justified this research field [2]. Main arguments include:

- Problems in game can be considered as simplified tasks of real world. The research about game AI can help improve AI algorithms works in reality and provides an easier modeling way compared to modeling reality.
- By solving problems in game AI modeling can inspire how to solve problems in reality.

These arguments regard game as a form to simulate the reality. However, while game has been considered as an excellent field to evaluate the performance of AI systems, the increasing proportion of virtual world in people's life and users' higher requirement for the performance of game AI also requests for more systematic and academic development of game AI.

This paper mainly focuses on approach to AI algorithms in games. Introduce some innovations and breakthroughs appeared in the process of developing game AI. Then introduce how different AI techniques are implemented in modern games. Finally, having a further discussion about other implementation of AI in game industry, like procedural content generation and then take a further look into their future.

2. Backgrounds

Artificial intelligence has gained huge progress in recent years. From the early success in board games like AlphaGo to the recent popular area of various game play AIs, the evolution of algorithm makes researchers paying increasing attention to fields of video games with more variate and requirement of human-like performance, challenging the current AI research works [3]. In this section, we will first go through the development of game AI, focus on main breakthroughs in last decades to see how AI evolved with games. And then introduce the modelling problem of game AI at current stage to further discuss AI techniques and game types in the next section.

2.1. Game AI Evolution

At early stage, with limited hardware performance and immature AI model, finite state machine was implemented as a computation model that can simulate sequential logic and some computer programs. Situations in reality were predefined as input for the state machine to generate reaction as output like the figure 1 shown below.

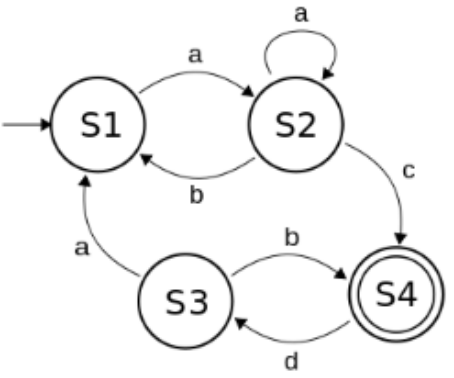

Figure 1. Finite state machine.

Later behavior tree was developed, using a tree-like structures to create and perform AI behaviors. But just like finite state machine, the cost of design limited their input possibilities and also scope of application. Therefore, most early research on game AI was focused on classic board games like checkers and chess as they have strictly constrained rules and clear input [4]. The complexity of these games keeps people's interest for thousands of years and set the goal for AI to beat the best human player.

DOI: 10.54254/2755-2721/15/20230829

The first breakthrough was made by Gerald Tesauro in 1992, he developed TD-Gammon using temporal difference learning to beat human players in backgammon [5]. Later in 1997, Deep Blue from IBM defeated the grandmaster of chess, marking the new development stage of game AI [6]. And in 2017, the viectory of AlphaGo to the world champoin Ke Jie declares game AI's success in mastring classic board games [3]. With the development of AI algorithms, classic board games are now relatively easy for game AI. Video based game with features of much more input possibilities, information asymmetry and the need of interacting with real person like a human has got the attestation of game AI research.

Started by Google in 2014, AI was trained to play classic video games [7]. Benefited from early research done on classic board games, the research on video games made rapid progress. Then in 2019, Google's game AI beat professional players of RTS games and game AI built by OpenAI beat professional players of MOBA games [8].

2.2. Modeling of Game AI

It could be seen that as a branch of AI, game AI gained huge successful over last ten years with the development and evolution of AI. But it is worth mentioning that approaches modeling and improving game AI is still quite different from other implementation of AI.

From the universal view, as a strong interactive activity, game must be understood so that it can be played by user and so does the AI. However, compared to other popular field of AI like computer vision and language mode, the implementation of game AI does not necessarily have to understand the current situation or extract useful information from the game, playing game is a process that need AI to keep engaging in the game environment and make decision to help it reach certain game purpose or performing like a real human based on the environment. For this point of view, game AI is actually a decision model that handles its choice of action and input environment, the process of making decision is the main issue concerned while building game AI.

The dominant work procedure of game AI is to divide the interaction between AI agent and the input environment into discrete steps [2]. Every time AI enters a new step of running, it makes decisions based on the input get from the environment and its decision, together with other possible action from other AI or player will lead the environment to a new state. Then AI will repeat strategies considering current state as new input environment of its step of running. This process is the intelligence of game Artificial Intelligence. Based on model constructed, there are also algorithms helps to evaluate decisions made by AI and help AI to choose the action with best evaluation in the whole process.

3. AI techniques and implementations

3.1. AI techniques

Based on the modeling method of game AI mentioned above, there are many approaches for building game AI. It is worth having more discussion on their implementation and possible improvements.

3.1.1. Reinforcement Learning. Reinforcement learning is a training method trains game AI to make decisions based on rewards and punishments received from the environment. In reinforcement learning for game AI, game AI will be given a specific goal, such as defeating the player, and then interacts with the game environment to achieve that goal, there will be certain standard of measuring AI's decision. As the AI takes actions in the game, it receives rewards or punishments based on the outcome of those actions and the measuring standards.

Over time, game AI can learn from these rewards and punishments, and uses that information to make better decisions in the future. The reinforcement learning algorithm continually updates the AI's decision-making process based on the outcomes of its actions, allowing it to improve its performance over time. Reinforcement learning can assign game AI with complex and adaptive behaviors. For example, AI using reinforcement learning could learn to adapt to the player's strategies and change its

DOI: 10.54254/2755-2721/15/20230829

own behavior accordingly. This can result in a more engaging and challenging game experience for the player.

However, reinforcement learning can be computationally intensive and requires a significant amount of training data to achieve good results. Additionally, designing a game environment that is well-suited for reinforcement learning can be challenging, as it requires careful consideration of the rewards and punishments that the AI may receive. From recent research, combining deep neural networks with reinforcement learning may have remarkably performance in many generes of games, and its application could be extended to wider domain including level design and automated balancing in game [9].

3.1.2. Supervised Learning. Supervised learning is another approach that involves training using labeled data. Training AI on a set of labeled examples. For example, AI may be trained to recognize and respond to a specific type of player behavior with dataset of labeled examples that show how it should respond to players' certain behavior and then tested on a separate set of data to evaluate its performance.

AI model trained by supervised learning can be used to have specific behaviors or decision-making processes. Advantages of this approach include that it can be relatively fast and efficient compared to other machine learning techniques. And within limited game conditions, like MOBA games or FPS games focusing on separated rounds of game play, it can have relatively better performance [10]. However, there are still challenges for supervised learning in game AI. On the one hand, it requires a large amount of labeled training data to achieve good results, which can be time-consuming and expensive to obtain. On the other hand, it could be difficult to ensure that training data can accurately reflects the game environment. And AI model trained on a specific set of labeled examples may not perform well in situations outside of that training data, leading to unexpected or undesirable behaviors [8].

Overall, supervised learning can be a useful tool in game AI for creating NPCs with specific behaviors or decision-making processes. However, it must be used carefully and with consideration of the specific challenges and limitations of the game environment.

3.1.3. Deep Q Networks. In addition to neural network for Reinforcement Learning, Deep Q Networks (DQNs) is almost the most widely used approach for game AI [11]. It is mainly implemented to learn optimal policies for decision-making in games. The basic idea behind a DQN is to use a deep neural network to approximate the Q-function, which is a function that takes state and player action as input, and action of AI as output based on the expected reward for taking certain action in that state.

To be more specific, the state could be the current game environment or different parameters of the character controlled by AI, while the action could represent a move or decision that the AI can make. And the expected reward would be the expected outcome of taking that action in that state, the outcome is usually calculated and measured by scores to show determine the gain or loss of AI. The DQN works by updating the Q-function based on a loss function that measures the error between the predicted Q-values and the actual Q-values. The actual Q-values are calculated using a process called experience replay, which involves storing the AI's experiences (i.e. state, action, reward, and next state) in a memory buffer and randomly sampling from this buffer during training.

Eventually, the use of DQN could help AI learn to make better decisions over time as it gains experience and refines its Q-function approximation. This can lead to more challenging and realistic gameplay, as well as more competitive AI opponents for human players. For example, Deep Q Networks were used for AI competition and visual fighting games on Atari2600 games [12] (see Figure 2).

DOI: 10.54254/2755-2721/15/20230829

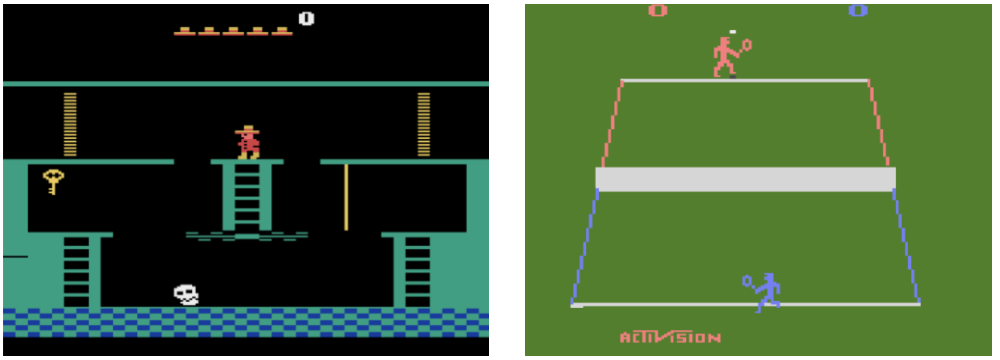

Figure 2. Atati2600 games used for Deep Q Networks training.

3.2. Strategies for implementing game AI

It could be seen that game AI has become indispensable part of modern game content, and as a complex and evolving approach, game AI needs strategies for its implementation. In this context, there are several strategies for implementing game AI that can be used to create engaging and complex gameplay.

3.2.1. Clear Design. Implementing a successful AI in a game starts with having a clear plan of what the AI needs to do and how it fits into the game design. A well-defined AI design can help identify what kind of AI is required, what behaviors it needs to exhibit, and what kinds of data it will need to process. While implementing game AI, well designed network structure would be the basement for future AI development and there should be serious consideration about acceptability of possible input that may added in the coming development. A poor designed network can lead to both undesired results and limit of reusability in future development.

3.2.2. Use of existing frameworks and tools. There are many AI frameworks and tools available to help simplify the development process. These can include libraries for pathfinding, behavior trees, an5o 06uyyurrrrrrrry d decision-making, which can help the reduction of the need for complex code and improve the efficiency of development process.

3.2.3. Balancing simplicity with complexity. The AI should be well structured and simple to implement and maintain, while being complex enough to create engaging and challenging gameplay. Finding the right balance between these two aspects is essential to the success of the game AI. This issue becomes more important when considering the limit of users' hardware devices. To be more specific, being too complex means the higher threshold for user community while being not simple enough may fail to provide game content that can attract users for long-term play.

3.2.4. Testing and iteration. AI development is an iterative process, and it is essential to test early and often. This can involve creating test environments and scenarios to see how the AI performs, and making necessary adjustments. Continuous testing and iteration can be helpful for improving the AI's performance and ensure that it meets the design goals.

4. Procedural content generation in game production

4.1. Advantages of procedural content generation

Procedural content generation for game (PCG-G) is a subfield of computational creativity that focuses on the automatic generation of game content using algorithms and machine learning techniques. It can be used in game development to create game content, such as levels, landscapes, and characters algorithmically instead of designing them manually. The use of PCG-G in games has become more prevalent in recent years due to the increasing demand for personalized and dynamic game experiences.

DOI: 10.54254/2755-2721/15/20230829

This has led to the integration of AI algorithms into PCG-G systems to create more sophisticated and intelligent game content. According to Barriga and Nicolas (2019), approaches of PCG-G can be classified into two kinds, traditional Search-Based methods and Machine Learning methods [13].

Traditional methods mainly depend on Pesudo-random Number Generators, Gnerative Grameers, Fractals and Noise with advantages of efficiency and easiness [14]. However, as the complexity of video game is growing exponentially with the prohibitive content and cost of games in recent years, there raises needs for more intelligent and creative approaches for generating game content to provide personalized and dynamic game experiences. And this has led to the integration of AI algorithms into PCG systems to create more sophisticated and intelligent game content.

Although most AI solutions are designed for classification and prediction problems, including game AI models mentioned above, there are few ways work relatively for generating game content, like recurrent neural networks (RNNs) and Generative Adversarial Networks (GANs).

Take recurrent neural networks, for example. RNNs work by using feedback loops in the network architecture, allowing the network to maintain an internal state that can be updated based on previous inputs. This makes RNNs well-suited for generating game content that is dependent on previous actions or events in the game. Neural networks can be trained on existing game content to learn the patterns and structures of the game, and then generate new content that is similar in style and quality as inputs are given to the network cumulatively.

For GANs' approach, AI algorithms can analyze and learn from existing game content to try to generate and improve the performance of network. In this process, one algorithm will generate content based on users' requirements as input, and another algorithm evaluates the quality of the generated content, providing feedback to the generator algorithm to improve its output [15].

4.2. Controversy and future implementation of procedural content generation

While procedural content generation for game offers many benefits to game developers and players, it also raises some concerns and controversies that need to be addressed.

One major concern is that PCG-G could lead to games that lack creativity and originality. Critics argue that relying too heavily on PCG-G could result in games that feel repetitive and formulaic, with little room for innovation and surprises. And another concern is that PCG-G could perpetuate biases and stereotypes if the algorithms used to generate content are not carefully designed and tested. For example, if an AI algorithm is trained on a biased dataset, it may generate content that reflects those biases, perpetuating harmful stereotypes and reinforcing existing inequalities.

To mitigate this risk, game developers need to ensure that their PCG-G algorithms are designed with diversity and inclusion in mind, and that they are regularly audited to identify and address any biases. In terms of future implementation, PCG-G is likely to become even more prevalent in the game industry as technology advances and AI algorithms become more sophisticated. As PCG-G becomes more mainstream, we can expect to see more games that offer personalized experiences tailored to individual players, and more games that adapt and evolve based on player behavior and preferences. Additionally, PCG-G may become more integrated with virtual reality and other emerging technologies, providing even more immersive and interactive game experiences.

5. Conclusion

This paper mainly focused on the history and current implementation of game AI. Among which the history of game AI covered how AI technology evolved with game development and some important breakthroughs happened in this process followed by their corresponding implementations in game production in their own time. Then there was a brief introduction about modeling methods for game AI to introduce the system and approaches for implementing AI in modern games. Based on the introduced modeling methods, the paper expanded the game AI technology to some well-performed network structures that are widely used in game production and made a discussion about possible improvements for these network structures for further development. After that, there were some strategies concerned with implementing game AI to help the performance of AI in game production. Lastly, the heated spot

DOI: 10.54254/2755-2721/15/20230829

procedural content generation that appeared recent years were evaluated by its advantages and controversy about it, followed by a look at its implementation in the future. Overall, the paper focused on the history and current implementation of various game AI technologies and discussed them from multi-aspect of view, and formed own opinions to their future based on the progress made by them.

References

[1] Westera, W., Prada, R., Mascarenhas, S., Santos, P.A., Dias, J., Guimarães, M. and Georgiadis, K. 'Artificial intelligence moving serious gaming: Presenting reusable game AI components', 2020 *Edu. Infor. Tech.*, **25(1)**, 351+.

[2] Risi, S., & Preuss, M. From chess and atari to starcraft and beyond: How game ai is driving the world of ai. 2020 *KI-Künst. Intell.*, **34**, 7-17.

[3] Silver, David, et al. Mastering the game of Go with deep neural networks and tree search. 2016 *Nature* **529.7587**: 484-489.

[4] Lu, Yunlong, and Wenxin Li. Techniques and Paradigms in Modern Game AI Systems. 2022 *Algorithms* 15.8: 282.

[5] Tesauro, Gerald. Temporal difference learning and TD-Gammon. 1995 *Commun. ACM* **38.3**: 58-68.

[6] Campbell, Murray, A. Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. 2002 *Artif. Intell.* **134.1-2**: 57-83.

[7] Mnih, Volodymyr, et al. Playing atari with deep reinforcement learning. 2013 *arXiv preprint arXiv:1312.5602*.

[8] Berner, Christopher, et al. Dota 2 with large scale deep reinforcement learning. 2019 *arXiv preprint arXiv:1912.06680.*

[9] Oh, Inseok, et al. Creating pro-level AI for a real-time fighting game using deep reinforcement learning. 2021 *IEEE Trans. Games* **14.2**: 212-220.

[10] Ye, Deheng, et al. Towards playing full moba games with deep reinforcement learning. 2020 *Adv. Neur. Infor. Proce. Sys.* **33**: 621-632.

[11] Gunawan, Leonardo Jose, et al. Analyzing AI and the Impact in Video Games. 2022 *4th Inter.Conf. Cyber. Intell. Sys.*,1-9.

[12] Torrado, Ruben Rodriguez, et al. Deep reinforcement learning for general video game ai. 2018 *IEEE Conf. Comput. Intell. Games*, 1-11.

[13] Barriga, Nicolas A. A short introduction to procedural content generation algorithms for videogames. 2019 *Intern. J. Artif. Intell. Tools* **28.02**: 1930001.

[14] Zhang, Yuzhong, Guixuan Zhang, and Xinyuan Huang. A Survey of Procedural Content Generation for Games. 2022 *Inter. Conf. Cul.Orient. Sci. Tech.*, 1-10.

[15] Liu, Jialin, et al. Deep learning for procedural content generation. 2021 *Neu.l Comput. Appl.* **33.1**: 19-37.