Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/54/20241484

AI applications in video games and future expectations

Xinhe Tian

Electrical and Electronic Engineering, University College of London, Gower Street, London, WC1E 6BT, United Kingdom

zceexti@ucl.ac.uk

Abstract. In recent years, artificial intelligence (AI) techniques have been extensively applied in various industries, including the gaming industry. The integration of different AI models into games has brought about a new era of game development, optimization, and overall gaming experience. This paper presents an overview of several AI algorithms that have been utilized in games, discussing their applications and providing predictions for the future of AI in gaming. It outlines how these algorithms have been incorporated into games to improve various aspects of gameplay. The application part of the paper focuses on game content optimization, specifically from the perspective of enhancing player experience. It examines areas such as player interaction and in-game elements, explaining how AI algorithms have been used to optimize these aspects and create a more immersive gaming experience. Furthermore, the future expectations section revolves around further advancements in optimizing the gaming experience and incorporating AI into the game design and development stages. It also discusses potential improvements in hardware that can better support AI integration in games. Finally, the article concludes with a summary and expectations for the future development of AI in gaming, highlighting the potential benefits and advancements that can be achieved through the continued use of AI algorithms.

Keywords: AI applications, game, algorithms, AI interactions.

1. Introduction

Artificial intelligence refers to the machine achieves human-like behaviors to analyze and respond to objective things. The technology contains a wide range of domains, not only the algorithms, engineering, and mathematics in the machine itself, but also subjects researching human beings like sociology and psychology. Game AI is the application of artificial intelligence techniques in the game related areas to design the game with more vivid elements, improving the motivation, entertainment, and attraction of the game. For instance, non-playable characters (NPC) in game presents human intelligence in response to the player's actions. Game AI was first appeared in board games to calculate the most likely winning strategy and acted as either an assistant or opponent role for the player. The technology started to flourish in late 20th century, the famous game "Pac-Man" was one of the works in 1980s. Nowadays the technique has been extended to more types of games, like the shooting, adventure and role play game. Today's game AI mainly focus on providing a highly realistic experience during the game. The simulation is built on the visual environment and interactive components in the game. Models are iterated towards a more human intelligence direction. In recent year games, AI can successfully preform basic human emotion, behaviors and expression (e.g., if the player attacks an NPC, NPC will fight back). However, in most games, there still exits vast number of machine irrational responses or lack of response in player

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/54/20241484

interactions. The problem is primarily caused by the defects in training models and the high technique costs of for the companies. This paper is written for making a basic introduction to the relative concepts of game AI. The game industry has taken a considerable market in the whole economy and join of AI into this area is inescapable. Game AI has greatly improved the playability of the games. Algorithms that have already been utilized and the application aspects will be introduced in this article to provide a reference for the future development. Also, future techniques that are still under development and are possible to be used are mentioned.

2. AI algorithms used in game

In this chapter, 5 different AI algorithms (reinforcement learning, imitation learning, Finite State Machine(FSM), Fuzzy State Machine(FuSM) and Genetic Algorithm(GA)) will be introduced. Reinforcement and imitation learning describe 2 machine learning methods. FSM, FuSM and GA use various ways to model the behaviors of the system.

2.1. Reinforcement learning

In game applications, reinforcement learning can be applied to make game characters (agents) learn decisions and behaviors based on their environment. It focuses on the problem of learning action plans, matching states and situations with actions that maximize long-term cumulative benefits. [1].

In reinforcement learning, the Markov Decision Process (MDP) is used to model sequential decision-making in a dynamic game system. The MDP consists of four main components: state, action, rewards, and policy [2].

State refers to information included in a certain state represented by the current environment observed by the agent. In reinforcement learning, algorithms for the state are constructed based on the expectation of rewards achieved in a given time.

Action is the output of reinforcement learning. After an observation for the environment, the agent will take a certain move from a set of possible discrete actions. The action will influence the current environment.

Rewards and punishment mechanism is one of the most important principles in machine learning. Once a decision is made, the system will give feedback accordingly. Different levels of feedback obtained are decided by the corresponding choice. The next decision made by the machine is affected by the previous feedback.

Policy is one of the most important features to impact the learning outcome. It will process the information acquired from the current state and decide which action to take. The machine aims to iterate an optimized policy to maximize the rewards.

MDP is a decision-making algorithm to construct a model of sequential decisions in a dynamic system. In this process, state transition probabilities only depend on the current state strategy which includes all the information learned from the past decisions (Figure 1).

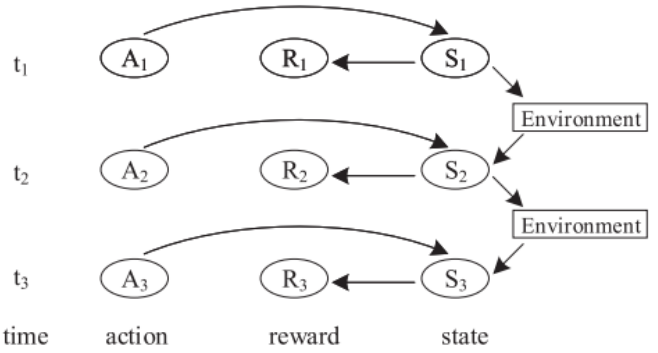

Figure 1. MDP flow chart [3]

DOI: 10.54254/2755-2721/54/20241484

The value of the current state can be quantized via algorithms like Bellman equations. To maximize this value, it calculates the sum of the rewards of the current state and a discount value that decrease the rewards of the next state. The value function is below:

$$V(s) = max_a(R(s,a) + \gamma V(s')) \quad (1)$$

where a denotes action, s denotes state, γ denotes discount factor

2.2. Imitation learning

Imitation Learning is a machine learning approach that aims to learn performance strategies from demonstrations, particularly in the context of games. Demonstrations are typically presented as state-action trajectories, where each pair represents the action to be taken at a given state [4].

2.2.1. GAIL (Generative Adversarial Imitation Learning). GAIL is a technique that employs generative adversarial training to imitate the performance of an expert model and closely approximate the expert's action distribution. In the context of gaming, GAIL constructs a reinforcement learning framework and utilizes specific algorithms to optimize the cost function.

2.2.2. BC (Behavior Cloning) BC is a supervised learning method commonly utilized in game settings. It learns the state-to-action mapping from expert trajectories, requiring a substantial amount of example data. However, the efficiency of BC may be hindered by the scarcity of appropriate examples and the high difficulty involved in data search.

These algorithms find application in game environments, enabling the learning of performance strategies by imitating expert demonstrations. In the game context, GAIL aims to bridge the performance gap between the agent and the expert, while BC leverages supervised learning to train game agents using expert trajectories. These methodologies provide game developers with effective means to train intelligent agents and enhance the overall gaming experience.

2.3. FSM

Finite State Machine (FSM) has been widely applied in game development to assist in controlling and interacting with game characters and objects. FSM is a mathematical model that describes the behaviors of an object in specific states and the transitions between states. It consists of a finite set of states and rules that define the transitions between these states. In games, each character or object can be defined as an FSM. It possesses a set of distinct states such as "walking," "idle," or "attacking." Based on the current state and input conditions, the FSM determines the next state. The transitions between states are typically achieved through trigger conditions and events.

The application principles of FSM in games are as follows:

1. State: The state of a game object is a fundamental component of an FSM. States can represent various behaviors or situations, such as a character's movement, attack, or defense states. Each state has specific behaviors and attributes associated with it.

2. Transition Condition: Transition conditions are the conditions or events that trigger state transitions. When specific conditions are met, the FSM switches from the current state to the next state. For example, when a character receives an attack, the state may transition from "idle" to "injured."

3. Action: Actions are the behaviors or tasks associated with states. They describe the specific operations that the FSM should perform in each state. For instance, in the "walking" state, a character may perform movement actions by adjusting speed and direction.

4.Transition Graph: The transition graph is a visual representation of an FSM. It consists of states and uses arrows to indicate the transition conditions between states. The transition graph helps developers understand and manage the states and behaviors of game objects intuitively.

The application of FSM in games offers several advantages. Firstly, it provides a simple and intuitive approach to managing the states and behaviors of game objects. By defining clear states and conditions, developers can have better control over the game logic and interaction. Secondly, FSM is flexible and

DOI: 10.54254/2755-2721/54/20241484

extensible. It allows for easy expansion and modification of the behaviors of game objects by adding new states and transition conditions. This flexibility makes the game development process more adjustable and maintainable. Lastly, FSM is effective in handling complex nonlinear behaviors. With well-defined behaviors and transitions for each state, developers can implement complex game logic and interactions more easily without getting lost or confused.

In conclusion, FSM is a mathematical model used in game development to control and interact with game objects by defining states, transition conditions, and actions. It offers an intuitive and flexible approach to managing the states and behaviors of game objects, making game development more controllable and extensible. By properly applying FSM, developers can create more exciting and diversified game experiences.

2.4. FuSM

FuSM (Fuzzy State Machine) and AI fuzzy logic algorithm are commonly used techniques in game development for implementing intelligent behaviors of characters or NPCs. The principles of their application are as follows:

Fuzzy state machine is an extension of FSM. FSM only has a single current status, FuSM has a superposition of different status [5]. Traditional state machines are based on discrete states, where each state has deterministic transition conditions and actions. In contrast, a fuzzy state machine introduces the concept of fuzzy logic, making the transitions between states fuzzy and continuous. In a fuzzy state machine, each state has a fuzzy value that represents the degree or confidence level of the current state. By applying the AI fuzzy logic algorithm, a set of inputs can be fuzzified and the fuzzy value of each state can be calculated based on a set of rules. Depending on the magnitude of the fuzzy value, the current state can transition to other states with a certain degree of fuzziness. This fuzziness can be defined through fuzzy control rules, such as fuzzy AND, fuzzy OR, and so on.

By using a fuzzy state machine, behavior decisions of characters can be defined more flexibly, allowing them to adapt to changing circumstances. For example, in a combat game, a character can determine its current state based on inputs such as health points, number of enemies, and weapon status, and then perform state transitions and actions according to a set of predefined rules. Compared to FSM, FuSM only works in games that can exist multiple current states. In an FuSM, the current state is decided by a value named “activationlevd” assigned to each state instead of whether a certain value is met or not. In League of Legend, the players choose different roles before the game starts, and they obtain different skills in the game. As the skills have various “activationlevd”, every player has various behaviors and states in game. FuSM games are normally supposed to be more interesting compared to FSM games for the unpredictable future behaviors of the players.

AI Fuzzy Logic Algorithm: AI fuzzy logic algorithm refers to the application of fuzzy logic in artificial intelligence decision-making problems. In games, AI fuzzy logic algorithm is commonly used for character decision-making, such as enemy actions or item usage (Figure 2).

The AI fuzzy logic algorithm relies on a set of fuzzy rules and fuzzy sets. It fuzzified input and output data, performs fuzzy inference on these data, and produces decision results. The basic components of the fuzzy logic algorithm include fuzzification, fuzzy inference, and defuzzification.

In the usual Boolean logic, there are only two possibilities about the “truth”, which is either true (1) or not true (0). In fuzzy logic, there is a “degree of truth”, which means the value of “truth” can be a number between 0 to 1. Human emotion recognition is complex as the physiologic behaviors cannot be judged by only “true” or “not true”, (e.g. respiration, impulse rate). The features are often located in the intermediate values. Fuzzy logic makes vague input values available and resemble human reasoning. It is adept in coping with human-like complicated problems. With those advantages, this algorithm can be utilized to train models recording human emotions.

DOI: 10.54254/2755-2721/54/20241484

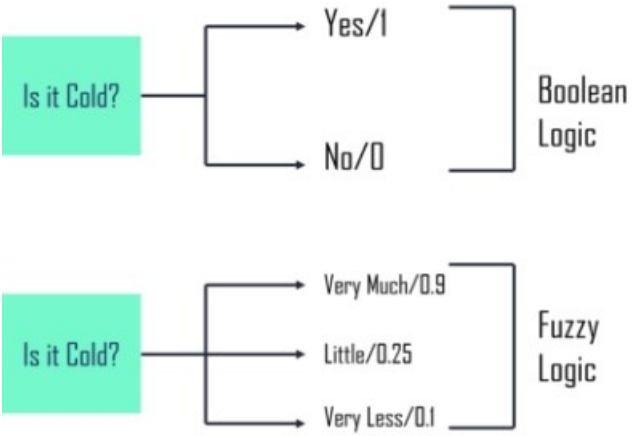

Figure 2. Fuzzy logic compared to Boolean logic [6]

In games, the AI fuzzy logic algorithm can be employed to solve complex decision-making problems, such as enemy attack decisions. By defining a set of fuzzy rules and input variables (e.g., enemy distance, health points), fuzzifying the data, and performing inference,a fuzzy output result (e.g., whether to perform a close combat attack, which skill to use) is obtained. Finally, defuzzification is applied to convert the fuzzy result into specific actions.

2.5. GA

Genetic Algorithm (GA) is a heuristic search algorithm based on the principles of biological genetics. It is commonly used in game applications to effectively search for and optimize solutions to problems. In application, GA starts by randomly generating a set of initial solutions as the initial population. Each individual in the population is evaluated based on the evaluation criteria of the problem, resulting in a fitness value (Figure 3). Higher fitness values indicate better solutions.

Next, through the selection operation, a subset of superior individuals is chosen as parents to produce the next generation. The selection operation ensures that the fittest solutions have a higher chance of being retained.

Then, through the crossover operation, a pair of individuals from the parent generation is selected, and their genetic information is exchanged and combined to generate new individuals. The crossover operation allows for the synthesis of favorable traits from different individuals, increasing the probability of obtaining better solutions. In addition, to introduce diversity into the search space and explore new regions, GA applies the mutation operation. The mutation operation introduces small random changes to the newly generated individuals, enabling the search to escape local optima.

Finally, the population is updated by merging the newly generated individuals with the original population. The process of selection, crossover, mutation, and updating is repeated until a termination condition is met, such as reaching a certain number of iterations or finding a satisfactory solution. In the context of gaming, GA can be applied to optimize game performance, generate intelligent character behaviors, and design level difficulty, among other uses. By emulating the process of evolution, GA adapts to the requirements of the game environment, providing better solutions and enhancing the gaming experience. It has a wide range of applications and serves as a powerful tool for solving complex problems and searching for optimal solutions.

GA can be utilized to deal with more complex targets. It automatically accumulates the experience and adjust the methods globally. The algorithm scores each individual and choose the elites to create an optimal new generation. In strategy games, its iterations efficiently evolve the winning strategies and can be used by the player or the computer acting as an opponent for the player.

DOI: 10.54254/2755-2721/54/20241484

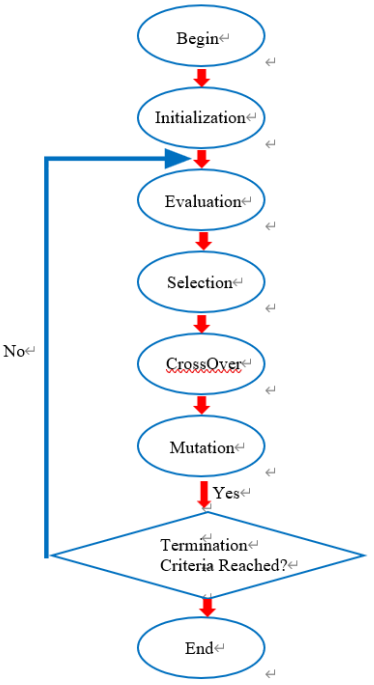

Figure 3. GA procedure diagram [7]

3. AI used to optimize player experience and game content

3.1. Detecting rendered image glitches

Some video games use simulated scenes and objects. For those games, they have a relatively high requirement for the graphs presented and are normally accompanied with issues of textures of the graphs. Without proper detection before the game starts, missing or corrupting of the graphs can ruin the game experience of the player. DCNNs (Deep Convolutional Neural Networks) can be used to recognize patterns. Training appropriate DCNNs with adequate data including faulty images is an effective method to detect glitches [8].

3.2. Game balancing

Matches different skill level (beginner, intermediate, expert) player with various level of tasks automatically [9].

In PVP games, matches similar skill and proficiency level players with each other.

The player experience is strongly related to the difficulty levels to handle the game, especially in PVP (player versus player) games. It is crucial to match the player with a moderate level correspond to the skills to ensure the player will not be frustrated by too complex tasks. Unreasonable matches lead to decreasing motivation. AI can learn the rules and figure out the fittest matches through learning from an extreme large number of matches with itself. For single-player games, it will detect the feedback of players' states (e.g., lost HP amount, death) and adjust the current task level or giving recommendations for the suitable difficulty level for the players.

3.3. Creating game contents

To utilize AI to produce pictures, models are trained by using abundant proper images as an input and adding sentences describing features. Then large amounts of eligible images can be made in a short time. Creating texts and music have the same principle. AI can create a story via reading other similar texts. For example, Chatgpt can text a joke if it is requested to tell a joke. Those techniques are now widely used in game development for their convenience and efficiency to reduce the cost.

DOI: 10.54254/2755-2721/54/20241484

In games including pvp (person vs person), if one of the players exit or lose internet connection during the game, AI will manipulate the character of that player and continue the game to avoid destroying the experience of other participants. In some moba games like League of Legend, the players can choose whether AI control the character or not if his partner drop out.

3.4. Emotion Detecting

PEM (Player Experience Modelling) will optimize players' game experience through their physiologic performance during the game. It contains detection of emotional manifestations represented by physiologic behaviors like facial expression, respiration, pulse rate and gestures [9].

To achieve this, AEI (Artificial Emotional Intelligence) technique is included. This technology acquires human behaviors via hardware (sensors, computer camera, microphone, etc.). The computer will process those features with deep learning algorithms and compare them with other data to try to aware important emotions (anger, excitement, fear, etc.) from players' physiologic behaviors received. Fuzzy logic algorithm mentioned above is utilized in this area.

3.5. More intellective non-playing characters

This part presents more intelligent and human-like NPC to optimize players' game experience, including NPC emotion modelling and various flexible responses. To increase the flexibility of the game, the answers to the questions are made to be open-ended which means the responses of the NPCs will be unique to each player based on their various actions. After evaluating the meaning of the texts inputs by the player, NPC will give an individual reply to each player to simulate a realistic conversation.

FAtiMA Toolkit is a tool that can help create characters with emotional intelligence. Its creation mainly based on the nature of the character instead of the event happens (Figure 4). This provides reasonable responses of different characters with various backgrounds participating in the same scene. CiF-CK is also utilized to make agents realize their own conditions. The relative model is trained based on emotional changes, responses and actions to interactions in different social environment.

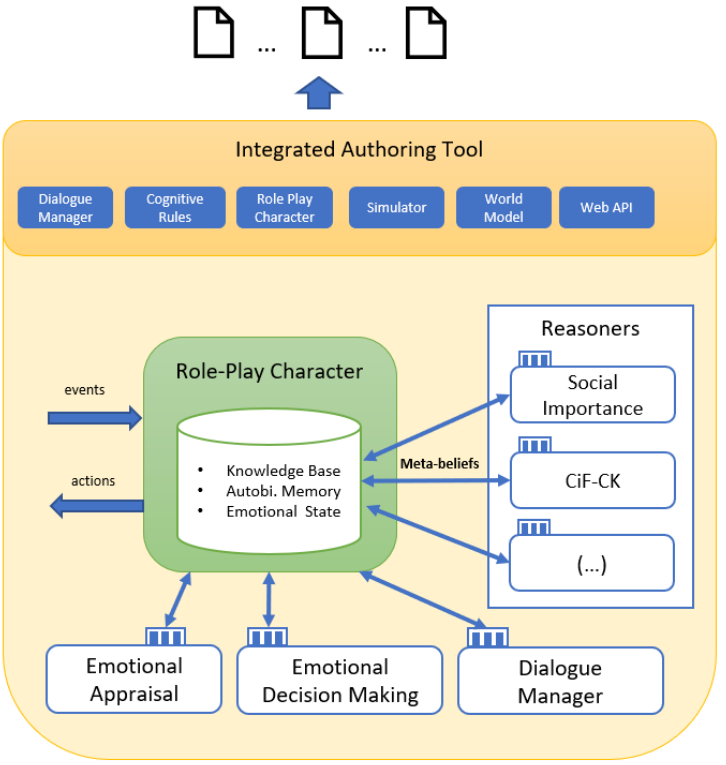

Figure 4. FAtiMA Toolkit components [10]

DOI: 10.54254/2755-2721/54/20241484

3.6. Interaction with human

Natural Language Processing (NLP) is a technique to understand human language and behaviors. AI will assess that information sent by the player and give corresponding reasonable and meaningful feedback. For example, in PVP games once the system has detected aggressive words sent by one of the players to his partner during the game, the player will face corresponding punishment (e.g., message ban) for his word attack to other players.

Sentiment analysis is used in NLP (Figure 5). It is designed to distinguish the properties of a word (e.g., positive, neutral, negative, joy belongs to positive, anger is negative). In machine learning, sentiment analysis is a classification problem. This can be achieved by converting text features (e.g., using python library: CountVectorizer) to vectors to input these vectors paired with property tags (e.g., python library: RandomForestClassifier) into the algorithm to train a model. Techniques like Word Embeddings can be used to extract features from the text. Word Embeddings present vectors in high-dimensional space referring the words and the space between the vectors represents the similarity of the words. Figure below shows a general process of machine learning sentiment analysis:

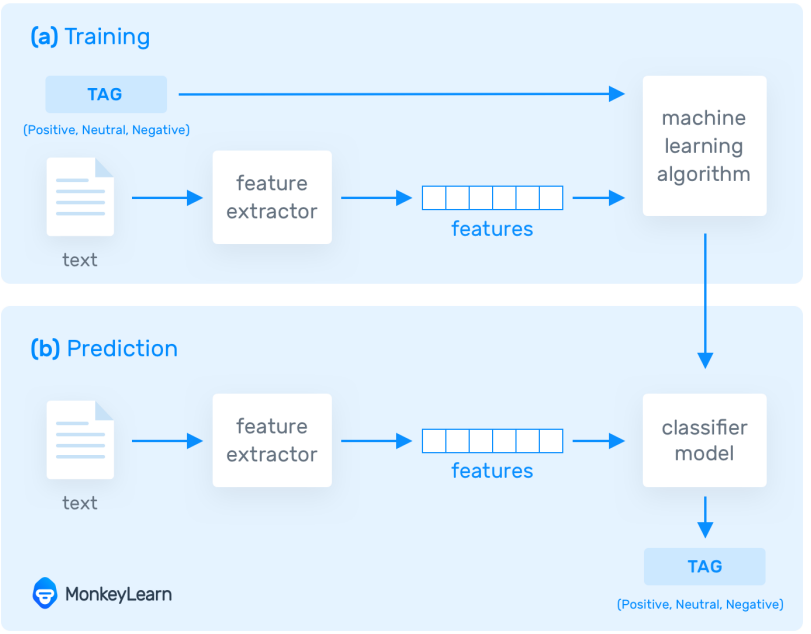

Figure 5. Sentiment analysis machine learning process [11]

The technique is also utilized in creating NPCs (evaluating meaning of the texts) mentioned before. NPCs that can understand human languages and give reasonable responses are more vivid.

4. Future expectations of AI in video games

4.1. Further intelligent to build a game

Decrease the difficulty of creating a game and even make customized game for each individual. Also, more automatic and intelligent iteration of a game.

AI is adept in predicting human favors based on their habits. For game designers, the machine can recommend players' preferred elements to help them in game development. In the future, AI may be able to keep building the game until the game ends according to the preference of each individual [12]. For example, in open world game, players have various liking degree towards different maps (e.g., forest, valley, sea), AI will construct customized map up-to-date based on each player's preference. When the game ends, the player can replay the game immediately as an expert instead of a beginner based on the

DOI: 10.54254/2755-2721/54/20241484

new iteration fitting the player's skill created by AI. With the using of this technology, the cost of game development will decrease and simultaneously improve the developing efficiency.

4.2. Highly intelligent human-like AI in game

AI is able to play the role of real players. Every character in game behaves like an existing human being. For players lack of accompany in game, AI will take the place of friends, partners and lovers instead. Nowadays AI can only figure out relatively basic human languages, emotion, and actions. In future the machine will be extremely close to real human. The potential shortcoming is the related moral problems.

For games in testing stage, highly human-like AI can play the role of the testing players and save the cost of testing player recruitment. Compared with real players, AI will be more efficient and accurate to analyze and repair the defect in game. With data collected in testing, further development strategies can be predicted more reasonably.

4.3. Further virtual reality

Optimized hardware property, comfort level, interaction response and more real-life like environment.

For three-dimensional real-life imitation game environment, AI will provide higher flexibility for players. Players will be able to experience more actions personally in three-dimensional game space (e.g., flying, fighting). The world in game will have more interoperable choices (e.g., can chat with every character, interact with every object) to approach the feeling in real world. The hardware will be improved to maintain the virtual reality game for a longer period of time and more stable running state. Currently, an elaborate virtual reality requires relatively high cost due to the technique difficulties. In future it will be accessible to every player.

5. Conclusion

This article provides a brief introduction to AI algorithms, their relevant applications, and the potential future in gaming. Among the five algorithms discussed, a comparison is made between reinforcement learning and imitation learning. The former is based on interacting with the environment to learn from mistakes and attempts, with the objective of maximizing rewards, while the latter involves learning through expert demonstrations. FuSM is an extension of FSM, and GA is another distinctive model based on biological genetics. The applications primarily focus on game content and player experience. It extensively analyzes the impact of AI algorithms on various aspects such as game scene settings, content generation, intelligent optimization, and human-computer interaction.

Looking ahead, the integration of AI algorithms into gaming holds tremendous potential for future advancements. While reinforcement learning and imitation learning have already demonstrated their capabilities, we can expect further refinements and advancements in these fields. The extension of FSM with FuSM and the utilization of GA to construct different models show promising directions for enhancing AI capabilities in games.

In terms of applications, there is a growing interest in leveraging AI algorithms to optimize game content and enhance player experiences. This includes more sophisticated game scene settings, improved content generation algorithms that create dynamic and immersive environments, and the development of intelligent systems that can adapt and optimize gameplay in real-time. Additionally, advancements in human-computer interaction through AI-powered NPCs will continue to play a vital role in creating more engaging and interactive gaming experiences.

Overall, the rapid progress of AI algorithms in the gaming industry is paving the way for innovative applications and advancements. With ongoing research and development, we can anticipate even more sophisticated and intelligent AI systems in the future, revolutionizing the way games are designed, developed, and experienced.

References

[1] Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019). Reinforcement learning, fast and slow. Trends in cognitive sciences, 23(5), 408-422.

DOI: 10.54254/2755-2721/54/20241484

[2] Li lin (2021).Research on application of artificial intelligence in video games based on machine learning. Nanjing information engineering university.
[3] Wang, C., Lei, S., Ju, P., Chen, C., Peng, C., & Hou, Y. (2020). MDP-based distribution network reconfiguration with renewable distributed generation: Approximate dynamic programming approach. IEEE Transactions on Smart Grid, 11(4), 3620-3631.
[4] Zheng, B., Verma, S., Zhou, J., Tsang, I. W., & Chen, F. (2022). Imitation learning: Progress, taxonomies and challenges. IEEE Transactions on Neural Networks and Learning Systems, (99), 1-16.
[5] Wang Yuxuan (2019).Analysis of the development and application of game artificial intelligence. Science and Technology Communication (02):125-126.
[6] Edureka:What is fuzzy logic in AI and what are its applications https://www.edureka.co/blog/fuzzy-logic-ai/
[7] Lambora, A., Gupta, K., & Chopra, K. (2019, February). Genetic algorithm-A literature review. In 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 380-384). IEEE.
[8] Ling, C., Tollmar, K., & Gisslén, L. (2020, October). Using deep convolutional neural networks to detect rendered glitches in video games. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 16, No. 1, pp. 66-73).
[9] Westera, W., Prada, R., Mascarenhas, S., Santos, P. A., Dias, J., Guimarães, M., & Ruseti, S. (2020). Artificial intelligence moving serious gaming: Presenting reusable game AI components. Education and Information Technologies, 25, 351-380.
[10] Research Gate: available via license: Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International https://www.researchgate.net/figure/FAtiMA-Toolkit-Components_fig1_349787091
[11] Monkey Learn: Sentiment Analysis: A Definitive Guide https://monkeylearn.com/sentiment-analysis/
[12] Zhou fei, Li jiuyan (2020). Current status and prospects of artificial intelligence application in game development. China management informatization(23),183-185