Proceedings of CONF-SEML 2025 Symposium: Machine Learning Theory and Applications
DOI: 10.54254/2755-2721/158/2025.TJ23324

The Exploring of AI Applications in Game Development

Yiyang Cui

Faculty of Information Technology, Monash University, Melbourne, Australian
ycui0036@student.monash.edu

Abstract: Recent advances in artificial intelligence technologies have begun to transform the gaming industry, especially in the areas of player-character interaction and narrative development. Traditionally, game stories and character relationships are predefined through scripted dialogues and sequences, which requires developers to invest a lot of time and effort. However, AI-driven approaches such as large language models (LLMs) and deep learning techniques offer a dynamic alternative that can enable more flexible, player-driven interactions and adaptive AI behaviors. This paper comprehensively reviews the current role of AI in game design from multiple perspectives, including applications in multi-agent interaction, procedural level and game content generation, and game development process optimization. In addition, this study explores the advantages and limitations of AI technology, coping with technical challenges, and ethical issues that may arise during the implementation of AI. The results are intended to provide a reference for the future application of AI in game design and provide recommendations for coping with emerging risks.

Keywords: Artificial Intelligence, Game design, Language Models, Aiethics, AI-driven Game Development.

1. Introduction

In the traditional game industry, when it comes to the interaction between players and the NPC and the advancement of the game plot, the script is usually written in advance, and then converted into a visual script by the developer in the game engine. This method can usually bring coherent and clear character relationships to players. In contrast, this method requires creators to design and modify a performance script a lot, and the content of the story will not change after completion. With the development of AI technology in recent years, some games have also used AI for the first time instead of predetermined scripts. For example, the large language model (LLM) can realize the free dialogue between the NPC and the player, and deep learning can increase the challenge brought by adversarial artificial intelligence to players. Designers expect AI to better improve the flexibility of the plot and the immersion of players, while reducing the workload of developers for designing scripts.

Therefore, it is necessary to study whether the current AI can meet the expectations of developers, whether players are satisfied with the introduction of AI, and the ethical risks of more intelligent AI in the future. This article will evaluate the current effectiveness of AI through the collected data and player feedback, and study the ongoing or completed AI participation projects to predict and suggest the development of AI in the next three years. In addition, this article will also explore the adaptability of current AI models in different game types, analyze the specific role of various AI tools in the development process, and propose several strategies to deal with technical and ethical challenges, providing a reference for subsequent research and practice.

2. Applications

The application of artificial intelligence technology in the field of games has undergone many years of evolution. Early programmatic content generation (PCG) laid the foundation for automated design. The programmatic content generation and machine learning framework (PCGML) proposed by Summerville et al. automatically generates content such as level structure and map layout by learning from existing game data, providing a theoretical and practical basis for the subsequent application of more advanced and generalized AI technology [1].

In recent years, large language models (LLMs) represented by Transformer have gradually become an emerging tool for game design. LLMs can efficiently understand and generate natural language, greatly reducing the technical threshold for game creation and promoting the important transformation of game design from "code-driven" to "natural language-driven"[2].

2.1. AI-driven NPC interaction

The most common game is Player VS Environment (PVE), where human players compete or cooperate with NPCs to achieve a certain goal, usually through behavior trees, finite state machines, etc. NPCs will execute the "simple reflex agent" mechanism and jump from one state to another when a certain condition is met [3]. This traditional method is mature and reliable, but it makes the NPC's behavior formulaic and lacks flexibility. Players usually only feel fresh at the beginning of the confrontation. At the same time, the more complex reaction mechanism will greatly increase the workload of developers.

The introduction of deep learning and neural networks allows NPCs (such as agents) to no longer rely on preset scripts. For example, in combat games, GCN allows each agent to exchange information with adjacent agents and learn each other's behavior. It uses the attention mechanism to extract interactive features, so that the AI characters in the game can show more natural, stable and strategic interactive behaviors in scenes such as collaboration, conflict, and resource competition. In this way, NPCs will no longer fight each other, thereby increasing the confrontation with players [4].

LLM has an unparalleled advantage in dialogue, so it is more handy for language games. Tabletop role-playing games (TTRPGs), also known as group games, have a human host (GM) during the game to expand the original story to a certain extent based on the background settings and player choices. Because of the randomness and rich playability, they are loved by many players. In contrast, although online role-playing games can also provide similar experiences, they are not as exciting as desktop games because they rely only on scripts and fixed scripts. However, the introduction of LLM has improved all of this. Take SceneCraft as an example. This framework uses the NPC settings, story backgrounds and scene dialogues provided by developers, and uses LLM to generate dialogues with branching structures, emotions and body descriptions, and then maps them into existing Unity animation resources and converts them into multi-branch scripts, thereby providing players with a highly dynamic and personalized interactive experience [5]. Similar to TTRPG, there are also love text adventure games (AVGs), which mainly use text combined with character portraits and sounds to attract players. In such games, falling in love with NPCs in the game is the core gameplay, which tests the creator's ability to write scripts and plots. However, there are not many interactive links in the game process, and the direction of the story is mainly determined by the player's choice of plot branches. In the early days, there were ways to interact with NPCs through emotion recognition, finite state machines and combined with player input text, but the NPC's response to emotional changes was relatively templated [6]. Today's LLM may have better performance and be more humane.

Similarly, Anjum et al.'s GPT-NPC practice in Stardew Valley also revealed players' general recognition of AI-driven characters' richer and more expressive interaction methods [7].

Traditional linear narrative structures are gradually being replaced by AI-driven nonlinear narrative systems. The Word2World system allows designers to directly construct game worlds and plot content using only natural language descriptions, greatly reducing the threshold for content creation for non-programmers while enhancing the openness and flexibility of game plots [8].

2.2. Programmatic level and game content generation

As a popular puzzle game, Sokoban has simple rules that are suitable for trying the possibility of LLM level generation. Todd used the GPT-2 standard model as a basis and found through testing that LLM can generate novel and playable levels, even if the levels themselves do not have any language structure. Because of this, the model needs to learn Sokoban's level design from scratch. The number of training samples is positively correlated with playability, but it performs poorly when faced with levels with complex solutions and too many steps [9].

MarioGPT also achieved automatic design of Mario levels based on text prompts by fine-tuning the GPT-2 model. While ensuring the playability of the levels, it retained the diversity and creative space of the generated level styles [10]. In addition, generative models such as GAN (generative adversarial network) have also been applied in level layout and map terrain design. Volz et al. proposed applying GAN to Mario level structure generation. Through latent space mapping, designers can obtain more creative possibilities and controllability in the level generation process [11].

Traditional games often set the game difficulty to different levels, such as easy, normal, difficult and hell. However, this method often requires designers to carefully adjust to achieve the best effect, which will undoubtedly take a lot of time and testing. The introduction of dynamic difficulty adjustment (DDA) is regarded as a key mechanism for AI to improve player experience. By analyzing player behavior data in real time, DDA can maintain the balance of game difficulty without excessively reducing the difficulty. This mechanism not only improves the immersion and stickiness of the game, but also significantly improves player satisfaction and retention [12].

2.3. Game development process optimization

Dragon's Dogma 2 is a classic example of AI-driven NPCs in recent years, which has triggered a lot of discussion and thinking. The developers tried to create a world where NPCs have independent consciousness and their behaviors will be based on their own settings, even if there is no player interacting with them. The purpose is to enhance immersion and realism. All NPCs are operated by AI deployed locally.

Cloud deployment is different from local deployment. It does not require any CPU or other hardware performance. The computing power provided by the cloud is much higher than that of the local, which allows AI-driven NPCs to perform better [13].

In addition to its application in player interaction, AI is also widely used in the entire process of game production. As early as 2013, Riedl and Zook pointed out the great potential of AI in automated script generation, behavior simulation and game system tuning [14]. These applications enable development teams to perform prototype design, content testing and performance optimization more quickly, thereby significantly improving development efficiency.

In general, the application of AI in the game field has gone beyond a simple content generation tool and has gradually developed into a comprehensive technology that runs through the entire process of game development and involves multi-dimensional game interactions. In addition to the content mentioned in this article, various AI models have also shown their influence in different game types.

3. User feedback

The changes in the game experience brought by AI are very impressive. At the same time, in the development process, the impact of AI on production efficiency, as well as the actual experience of developers and players, have profoundly affected the development path of the game industry. AI can accelerate the production process and optimize the testing process, such as automatic script generation and repetitive testing, to achieve the effect of reducing costs and increasing efficiency [15]. From the questionnaire for developers, it can be seen that 72% of developers are positive about the narrative content and efficiency generated by AI [16]. Especially in terms of visual resources, art designers can use generative AI in early concept and prototype design to quickly obtain the expected results. The entire process saves about 30% of working time compared with the traditional model.

However, practitioners also generally expressed concerns about its uncontrollable side, especially the stability of output and the consistency of overall style. Developers may become more and more dependent on AI, which is another hidden worry [16]. In terms of specific applications, the current application of LLM in game design is mainly concentrated in the generation of text content and dialogue content [17]. Players' experience with LLM embedded in NPCs is generally acceptable. In a game project called Stardew Valley where ChatGPT is embedded, most players believe that LLM enhances interactive immersion and character personality expression, which is its main advantage. However, if there is inconsistency in logic or the performance deviates from the character setting during the interaction, the sense of being out of play will be stronger [18].

4. Limitations and ethics

4.1. Limitations

Although LLM has shown good performance in level generation, its essence is not to truly "understand" or "run" the level, but to splice fragments based on rule simulation. This method may lead to a gap between the clearance path envisioned by the designer and the actual feasible solution. Especially in complex level design, the expressiveness of LLM is significantly reduced, so it is still necessary to cooperate with other methods to verify and optimize the generated level [5].

In addition, although LLM has a strong language generation ability, it still has obvious shortcomings in its memory mechanism. When the input context conflicts with the internal memory of the model, the model may ignore the current context; and when the context is weakly related to the current generated content, it is easily interfered by the previous context. Such problems may cause inconsistencies in plot logic in text-based games, affecting the player's immersive experience [9].

In terms of actual deployment, there are also a series of prerequisites for the stable operation of the cloud LLM model. For example, LLM is extremely sensitive to changes in the game environment. To ensure that NPC behavior complies with the settings, it is necessary to rely on an external database to input game status information in real time to prevent it from generating non-compliant content. Even so, LLM may still experience "hallucinations" or logical breaks during operation. In addition, cloud deployment is also highly dependent on the stability and response speed of the network, which is also a key factor affecting the player experience.

Take "Dragon's Dogma 2" as an example. According to developer interviews, the NPC AI behavior in the game is heavily dependent on CPU performance, especially in urban areas with dense NPC populations. Players reported that the maximum frame rate in towns is only 30 frames, while in the wild it can reach 80-90 frames. Since AI mainly occupies the CPU rather than the GPU, it is difficult to significantly improve the frame rate even if the image quality is reduced [19]. Therefore, when deploying AI locally, the hardware conditions of the players must be comprehensively considered to find a reasonable balance between NPC expressiveness and system resource consumption. More efficient logic scheduling and behavior execution order may reduce the performance burden while maintaining expressiveness, thereby achieving behavioral performances that go beyond traditional scripts.

At the same time, cloud deployment is also accompanied by problems such as response delay, network dependency, and data security, which have become important practical challenges that restrict its large-scale application.

In addition, the quality of level content generated by GAN is highly dependent on training data. If the number of samples is too small, it is easy to cause repeated generated content and a single style. More importantly, GAN does not have the ability to understand game mechanics or player behavior, which may lead to inconsistent styles in some areas and affect the overall game experience.

4.2. Ethics

As AI can produce more and more rich and complex content, the corresponding ethical issues are gradually emerging. The most important is the rights of players. Players will have various emotional changes in the process of interacting with AI. For example, language games are more likely to touch the hearts of players. However, there are two hidden dangers in this process. First, such emotions will induce players to do certain behaviors, such as inducing consumption or excessive anthropomorphism, which will cause players to rely on AI. Second, the instability of the model will cause mental damage, such as some generated broken pictures or outrageous remarks. Players' privacy may also face the problem of leakage. For example, the use of cloud-deployed models requires collecting players' information, usually text, voice, etc. Game companies may analyze this data to get better feedback, but this will involve whether it violates the privacy of players.

Then there are the ethical issues that may arise when players interact with NPCs embedded with AI. For example, in Dragon's Dogma 2, players will kill some unimportant NPCs in order to solve the problem of low frame rate and lag caused by too many NPCs. Another phenomenon in this game is that NPCs will get a disease, but the sick NPC is set to kill other NPCs. These unconventional behaviors have not yet been characterized, but in the future, when NPCs embedded with AI perform more realistically, they may cause corresponding ethical issues.

The behavior and performance of AI are often related to the number of training times and the content of samples, especially for LLM models that mainly process text. The input parameters will seriously affect the content output by LLM. However, LLM requires more than 100 million or even billions of parameters, which will make it difficult to control the training content of the model itself and may cause bias problems. For example, the gender, racial discrimination or anti-human content of the sample itself will make it possible for LLM to output incorrect text, which will have devastating consequences for the overall game experience. Therefore, developers need to supervise and control the output content and carefully screen the training content. LLM deployed locally should establish relevant regulations, such as not allowing players to make local adjustments, so as to avoid biased and negative output content, resulting in unclear "output responsibility".

5. Conclusions

Using LLM to generate plots and cooperate with the narrative experience manager is expected to further improve the narrative integrity of the story. The system will display multiple possible plot paths based on current events and story causality, and select the branch direction that meets the author's expectations and fits the player's behavior. Among them, the Narrative Experience Manager (EM) is more like a director, which can dynamically adjust the direction of the plot to ensure that the ending of the story is logically coherent and reasonable. Co-creative AI tools will play an indispensable role in future development. At present, although most tools are relatively mature in the generation of map, level and scene content, they are still insufficient in the design of AI behavior. The core requirements of developers for co-creative AI tools are: AI can be a creative collaborator, have a high degree of behavior and personality plasticity, provide clear behavioral visualization feedback, and give developers full control during the design process. In addition, it is necessary to further expand the evaluation dimensions of AI, such as social influence and user acceptance. Compared with the accuracy used to evaluate performance in the traditional sense, game developers are more concerned about the performance stability and role consistency of AI. For example, whether AI can continue to maintain its established personality settings and whether the content it generates has a negative impact on players are directly related to the immersive experience and user satisfaction of the game. Looking to the future, with the continuous evolution of AI technology and the continuous expansion of application scenarios, its influence in the game field is bound to expand furthe.

References

[1] *Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., Isaksen, A., ... & Togelius, J. (2018). Procedural content generation via machine learning (PCGML). IEEE Transactions on Games, 10(3), 257–270.*

[2] *Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.*

[3] *Jagdale, D. (2021). Finite state machine in game development. International Journal of Advanced Research in Science, Communication and Technology, 10(1).*

[4] *Jiang, J., Dun, C., Huang, T., & Lu, Z. (2018). Graph convolutional reinforcement learning. arXiv preprint arXiv:1810.09202.*

[5] *Sudhakaran, S., González-Duque, M., Freiberger, M., Glanois, C., Najarro, E., & Risi, S. (2023). Mariogpt: Open-ended text2level generation through large language models. Advances in Neural Information Processing Systems, 36, 54213–54227.*

[6] *Cavazza, M., Pizzi, D., Charles, F., Vogt, T., & André, E. (2009). Emotional input for character-based interactive storytelling. Proceedings of the International Conference on Interactive Storytelling, 1–10.*

[7] *Anjum, A., Li, Y., Law, N., Charity, M., & Togelius, J. (2024, May). The ink splotch effect: A case study on ChatGPT as a co-creative game designer. In Proceedings of the 19th International Conference on the Foundations of Digital Games (pp. 1–15).*

[8] *Nasir, M. U., James, S., & Togelius, J. (2024). Word2World: Generating stories and worlds through large language models. arXiv preprint arXiv:2405.06686.*

[9] *Todd, G., Earle, S., Nasir, M. U., Green, M. C., & Togelius, J. (2023). Level generation through large language models. In Proceedings of the 18th International Conference on the Foundations of Digital Games (pp. 1–8).*

[10] *Kumaran, V., Rowe, J., Mott, B., & Lester, J. (2023, October). SceneCraft: Automating interactive narrative scene generation in digital games with large language models. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 19(1), 86–96.*

[11] *Volz, V., Schrum, J., Liu, J., Lucas, S. M., Smith, A. M., & Risi, S. (2018, July). Evolving Mario levels in the latent space of a deep convolutional generative adversarial network. In Proceedings of the Genetic and Evolutionary Computation Conference (pp. 221–228). ACM.*

[12] *Li, D., Rawat, A. S., Zaheer, M., Wang, X., Lukasik, M., Veit, A., Yu, F., & Kumar, S. (2022). Large language models with controllable working memory. arXiv preprint arXiv:2211.05110.*

[13] *Gallotta, R., Todd, G., Zammit, M., Earle, S., Liapis, A., Togelius, J., & Yannakakis, G. N. (2024). Large language models and games: A survey and roadmap. IEEE Transactions on Games.*

[14] *Riedl, M. O., & Zook, A. (2013, August). AI for game production. In 2013 IEEE Conference on Computational Intelligence in Games (CIG) (pp. 1–8). IEEE.*

[15] *Filipović, A. (2023). The role of artificial intelligence in video game development. Kultura Polisa, 20(3),*

[16] *Jagli, D. R., Chandra, D. S., Dhanikonda, S. R., & Laxmi, N. (2024). Artificial intelligence usage in game development. Preprints. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4929567*

[17] *Yang, D., Kleinman, E., & Harteveld, C. (2024, August). GPT for games: A scoping review (2020–2023). In 2024 IEEE Conference on Games (CoG) (pp. 1–8). IEEE.*

[18] *Deng, C., Duan, Y., Jin, X., Chang, H., Tian, Y., Liu, H., ... & Wang, H. (2024). Deconstructing the ethics of large language models from long-standing issues to new-emerging dilemmas. arXiv preprint arXiv:2406.05392.*

[19] *Gamersky. (2024, October 19). Developers discuss performance issues in Dragon's Dogma 2: Related to NPC AI. Retrieved from https://www.gamersky.com/news/202410/1831886.shtml*