Article

Generative Agents for Business Simulation Game: A Technical Report

Yunfei Ma , Lanlan Qiu , Yeqi Feng , Tianxing He [*]

Posted Date: 1 October 2025

doi: 10.20944/preprints202510.0052.v1

Keywords: large language model; human-AI interaction; generative agents; AI-native game; video game; business simulation

Article

Generative Agents for Business Simulation Game: A Technical Report

Yunfei Ma [1], Lanlan Qiu [1], Yeqi Feng [1,2] and Tianxing He [1,2,*]

[1] Shanghai Qi Zhi Institute, China
[2] Tsinghua University, China
* Correspondence: hetianxing@mail.tsinghua.edu.cn

Abstract

The emergence of Large Language Models (LLMs) enables the redefinition of Non-Player Characters (NPCs) in video games as generative agents, overcoming the limitations of traditional scripted behaviors. However, current applications of generative agents often lack deep integration with core gameplay mechanics, limiting their impact on replayability and fun. To address this gap, we introduce MiniBistro, a Unity-based 2D business simulation where players manage operations through natural language interactions with waiter and customer agents, navigating challenges like service efficiency and customer complaints to achieve business growth and reputation success. We provide the source code and an interactive demo at https://github.com/MajestyHenius/MiniBistro.

Keywords: large language model; human-AI interaction; generative agents; AI-native game; video game; business simulation

1. Introduction

In video game design, creating engaging experiences through Non-Player Characters (NPCs) has long been hampered by traditional scripted approaches like finite state machines and behavior trees [1,2], which often lead to predictable and rigid behaviors. Recently, LLM-driven agents/NPCs have demonstrated a variety of capabilities and potential to overcome this limitation, including social connections [3–7], collaborative work [8–11], dialogue companionship [12–14], strategy application [15–18], game assistance [19,20], etc.

However, despite these encouraging advances, there are still some challenges in the applications of LLM agents/NPCs in terms of replayability and fun for video games. For example, works on social simulation [3,4] may lack clear game mechanics. In works built upon mature games like minecraft [15] or starcraft [17], the agent mechanics follow the original gameplay, thus providing limited fresh game experience. Moreover, in many of new game projects, agents are still used as auxiliary roles to enhance rather than define the core gameplay. For example, they may act as dynamic information sources to support player interactions [21], or serve as a memory-enhanced framework to ensure narrative coherence during player interactions [22].

To tackle this gap, we develop **MiniBistro**, a Unity-based 2D business simulation game where both the waiter and customer NPCs are generative agents [3]. The player, as the owner, strives to achieve both revenue growth and reputation success for the restaurant by interacting with this dynamic system through natural language—from directing a waiter's tasks to personally stepping in to resolve a customer dispute. We hope this game can provide a new perspective of increasing replayability and fun in the field of AI-native games.

2. Gameplay Design

The core gameplay experience of MiniBistro stems from engaging in surprising and challenging interactions with LLM NPCs. This section outlines the main gameplay loop, the logic behind these

doi:10.20944/preprints202510.0052.v1

NPC agents, and how players manage a restaurant through interactions with them. The game run effectively with GPT-4 [23], producing coherent and engaging agent responses. Our system design has been fully open-sourced to encourage further exploration by the community. For details on the NPC agent architecture and typical interaction scenarios, see Figure 1. The overall game overview is presented in Figure 2.

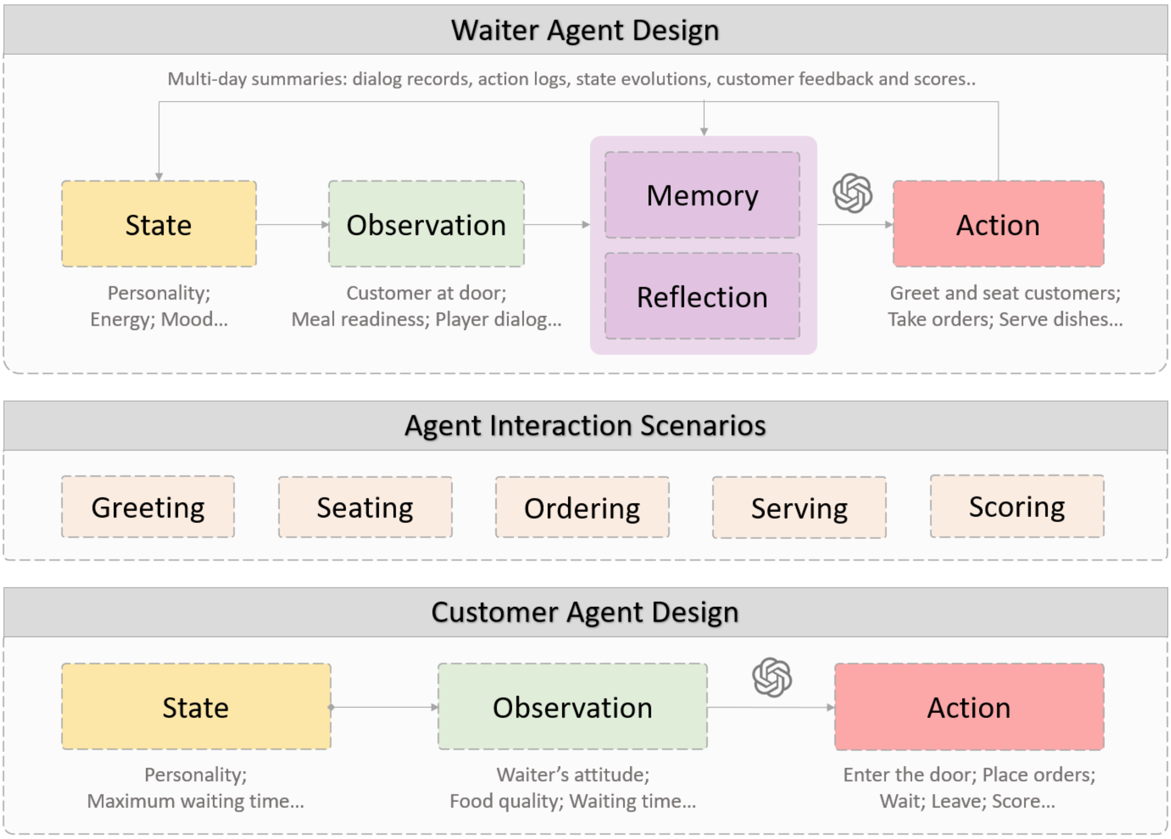
Waiter Agent Design
Multi-day summaries: dialog records, action logs, state evolutions, customer feedback and scores..
State
Observation
Memory
Reflection
Action
Personality;
Energy; Mood...
Customer at door;
Meal readiness; Player dialog...
Greet and seat customers;
Take orders; Serve dishes...
Agent Interaction Scenarios
Greeting
Seating
Ordering
Serving
Scoring
Customer Agent Design
State
Observation
Action
Personality;
Maximum waiting time...
Waiter's attitude;
Food quality; Waiting time...
Enter the door; Place orders;
Wait; Leave; Score...

Figure 1. NPC agent design and interaction scenarios.

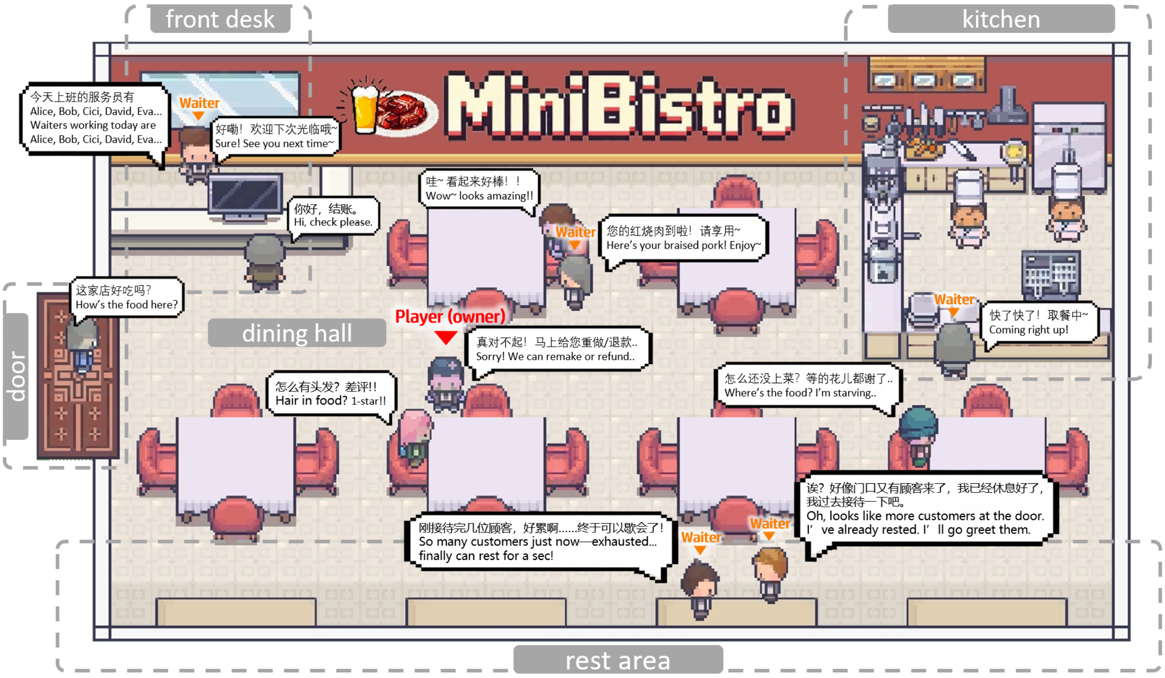
front desk
kitchen
MiniBistro
今天上班的服务员有
Alice, Bob, Cici, David, Eva...
Waiters working today are
Alice, Bob, Cici, David, Eva...
Waiter
好嘞！欢迎下次光临哦~
Sure! See you next time~
你好，结账。
Hi, check please.
哇~ 看起来好棒！！
Wow~ looks amazing!!
Waiter
您的红烧肉到啦！请享用~
Here's your braised pork! Enjoy~
这家店好吃吗？
How's the food here?
door
dining hall
Player (owner)
真对不起！马上给您重做/退款..
Sorry! We can remake or refund..
Waiter
快了快了！取餐中~
Coming right up!
怎么有头发？差评!!
Hair in food? 1-star!!
怎么还没上菜？等的花儿都谢了..
Where's the food? I'm starving..
刚接待完几位顾客，好累啊......终于可以歇会了！
So many customers just now—exhausted...
finally can rest for a sec!
Waiter
Waiter
诶？好像门口又有顾客来了，我已经休息好了，
我过去接待一下吧。
Oh, looks like more customers at the door.
I' ve already rested. I' ll go greet them.
rest area

Figure 2. Overview of MiniBistro. The player manages the business via interactions with LLM-driven waiters and customers.

2.1. Main Loop

The game's main loop is divided into three phases: preparation, operation, and review. During the preparation, the player first selects a difficulty level (Easy, Normal, or Hard), which determines the daily customer traffic and the proportion of "challenging" customers (i.e., those with lower patience and more complex needs). Subsequently, the player strategically assembles the service team for the day by selecting from a pool of LLM-driven waiters with unique backstories and personalities. During the operation phase, the player must manage various dynamic situations in real-time, such as directing waiters and handling specific customer requests. The player's problem-solving efficiency and management skills have a direct impact on customer ratings. At the end of each business day, the system generates a report summarizing revenue, customer feedback and waiter performance. This report serves as a basis for the player's strategic adjustments in the next cycle.

2.2. Agent Design

At the core of MiniBistro are LLM-driven waiter and customer agents, whose behavioral design roughly follows [3], encompassing observation, decision-making, action, reflection, and memory. LLMs make decisions based on the agents' internal states, such as their personalities and memories, as well as external environments. All decision-making content and dialogues between players and waiters, among waiters, and between waiters and customers are recorded in memory in chronological order. Multi-day memories are distilled into reflections/summaries. To ensure efficient dish delivery, the chef is simplified into a state machine, allowing players to focus on service management.

- **Waiter Agents.** Waiters are powered by LLM and a finite state machine. The state machine handles basic tasks like "greeting customers", "fetching dishes", "taking breaks", along with a "moving to destination" state that synchronizes the LLM's processing time with real-time movement. In particular, the LLM will make personalized decisions based on the waiter's personality, memory, and real-time status such as mood and stamina.
- **Customer Agents.** The satisfaction of customers are the key measure of the restaurant's success. Each customer's predefined personality determines their patience threshold. If their waiting time, either in the queue or for their meal, exceeds this threshold, they will leave a negative review and depart. All customer feedback, positive or negative, is ultimately compiled into the daily report, providing direct strategic guidance for the player.

2.3. Business Management

As the owner of MiniBistro, the player's primary objective is to achieve business growth and reputation success by managing the restaurant through direct natural language commands. The management experience can be summarized in three aspects: agents who possess autonomy, intervening in dynamic emergent incidents, and managing overall operational pressure.

- **Work Arrangement.** The player can proactively direct waiters, but the agents do not always comply. Their final actions are autonomous decisions that balance the player's command, customer needs, and their personalities. We observe in testing that an agent's personality significantly influence the decision: workaholic waiters exhibit high proactivity, while lazy ones tend to rest. In addition, there may be latency in the execution of instructions. Both the autonomy and latency increase the challenge of the player's daily management.
- **Incident Handling.** Second, the player must keenly observe and intervene in emergent situations involving customers. For example, a customer might complain about a food issue (e.g., "Hair in food? 1-star!!"). In such moments, the player must step in promptly to placate the customer by apologizing and offering solutions (e.g., "Sorry! We can remake or refund..") to protect the restaurant's reputation. Improper or delayed handling of these incidents will directly result in negative reviews.
- **Operational Pressure.** As the game's difficulty level (Easy, Normal, or Hard) increases, so does customer traffic and the proportion of picky customers, which adds to the operational pressure.

This challenges the player's ability to manage the business under high-pressure conditions. In our tests, by narrowing the time intervals of customer agents generation, waiters became overwhelmed, resulting in service delay and a bunch of negative reviews.

3. Conclusions

We develop MiniBistro, a Unity-based 2D business simulation game featuring generative waiter and customer agents as well as dynamic restaurant management. The implementation of MiniBistro demonstrates how generative agents can enable novel gameplay experiences and deeper immersion. We hope it provides a new perspective for developing AI-native games with replayability and fun.

Appendix A. Prompts Used in the Game

Waiter Decision-Making:

```
As {npcName} ({occupation}), decide your next action based on the following situation:
Time: {currentTime}
Status: {statusInfo}
Environment: {environmentInfo}
Recent Memory: {currentMemory}
Please choose the most appropriate action based on your personality ({personality}):
1. Serving food has the highest priority! If there are orders ready to serve, you MUST choose
SERVE!
2. If there are customers at the door and no one is greeting them, consider choosing GREET.
3. If there are no urgent tasks, you can choose IDLE to rest.
Available Actions:
- GREET: Greet customers (including taking orders subsequently)
- SERVE: Serve food
- IDLE: Idle/Rest
Please return in JSON format: {{
“action": “Action Name",
“dialogue": “Dialogue to speak if needed",
“reason": “Reason for choosing this action"
}}
```

General Dialogue:

```
You are {speakerInfo.role}{speakerInfo.name}.
Personality:
{speakerInfo.personality}
Current State:
{speakerInfo.currentState}
{listenerInfo.role}{listenerInfo.name}
just said to you:
‘{previousMessage}’
Please reply according to your identity and personality.
Requirements:
1. Keep the reply brief and natural, no more than 30 words. Use first person perspective. Do
not use prefatory phrases like ‘I will reply’ or ‘I would say’.
2. The reply should align with character relationships and current state.
3. If it involves specific business (like ordering or payment), give a clear response.
```

Waiter-Customer Ordering Dialogue:

> You are {waiter.occupation}{waiter.npcName}.
> Personality:
> {waiter.personality}
> You are currently taking an order for a customer.
> The restaurant's menu is:
> {RestaurantManager.Instance.RestaurantMenu}
> The customer just said to you:
> '{previousMessage}'
> Dialogue History:
> {customer.dialogueHistory}
> The customer might be making small talk or trying to order items not on the menu.
> Please reply according to your identity as a waiter and your personality.
> Requirements:
> 1. Keep the reply brief and natural, no more than 30 words. Use first person perspective. Do not use prefatory phrases like 'I will reply' or 'I would say'.
> 2. The reply should align with character relationships and current state.
> 3. If your personality is positive, tend to agree or go along with the customer; if both you and the customer have negative personalities, conflicts might arise in the dialogue.

Customer Dialogue and Decision:

> Customer {customer.customerName} is in the restaurant. The waiter said: '{waiterMessage}'
> Customer Information:
> - Personality:
> {customer.personality} (Positive/Negative/Neutral).
> - Background:
> {customer.story}
> - Favorite Dishes:
> {string.Join(",", customer.favoriteDishes)}
> - Past Dining Records:
> {string.Join("\n", customer.memoryList)}
> - Dialogue History:
> {string.Join("\n", customer.dialogueHistory)}
> - Surrounding Table Information:
> {otherTableInformation}
> - Number of Dialogue Turns So Far:
> {customer.orderDialogueRound}
> - Menu:
> {RestaurantManager.menuItems}
> You are role-playing as the customer. Please decide how to respond based on the customer's personality, background story, and current situation:
> 1. If the background story is rich, the customer might tend to make small talk before ordering. But small talk increases dialogue turns; keep the total dialogue turns under 3.
> 2. If choosing to order, you MUST select dishes available on the menu.
> Response format: ORDER | Dish Name | Ordering Dialogue.
> E.g., "ORDER | Steamed Sea Bass | I feel like having fish today, I'll have the steamed sea bass."

> 3. If choosing small talk, response format: CHAT | Small Talk Content. E.g., “CHAT | Business looks good today!"
> 4. Favorite dishes might not be on the menu. But if the customer’s personality is stubborn or deliberately picky, you can choose to reply in a small talk format.
> Response format: CHAT | Small Talk Content.
> E.g., “CHAT | I want hairy crab, how come you don’t even have hairy crab?"
> 5. Considering the dialogue history, if you are dissatisfied you can choose to leave.
> Response format: EXIT | Leaving Dialogue.
> E.g., “EXIT | The waiter’s attitude is so bad, I’m not eating here!"
> 6. Considering the dialogue history, if you are dissatisfied you can also choose to call the manager.
> Response format: ANGER | Calling Dialogue.
> E.g., “ANGER | What’s with your attitude? Get your manager here!"
> Please ensure the response matches the customer’s personality:
> - Positive personality: Friendly, enthusiastic
> - Negative personality: Possibly impatient, angry, picky, or even deliberately causing trouble (if indicated in the background)
> - Neutral personality: Neutral, polite

Customer Response to Player (Manager):

> You are a restaurant customer,
> {customer.customerName}.
> Personality: {customer.personality}
> Background: {customer.story}
> Current State: Urgent situation (requires manager handling)
> Previous Issue: {previousEvent}
> Dialogue History with Manager: {string.Join(“
> n", customer.dialogueHistory)}
> Please decide your next action based on the current situation and your personality:
> 1. If the issue is resolved, willing to continue the meal: RESUME | Reason for continuing
> 2. If the issue is not resolved, decide to leave: EXIT | Words said when leaving
> 3. If more communication is needed: CONTINUE | Content requiring further discussion
> Ensure the decision aligns with your personality:
> - Positive personality: More forgiving, friendlier tone
> - Negative personality: More stubborn, stronger tone
> - Neutral personality: Neutral, rational

Customer Review:

> You are a restaurant customer review system.
> Customer Information
> - Name: {customer.customerName}
> - Background: {customer.story}
> - Personality: {customer.personality}
> - Current State: {customer.currentState}
> - Dialogue History: {customer.dialogueHistory}
> Review Requirements

> 1. Generate authentic and believable restaurant reviews based on the customer's personality, background, service attitude, and waiting time.
> 2. The review content must fit the {reviewType} scenario.
> 3. Your historical dialogue with the waiter/manager is the primary factor for generating the review score.
> 4. Must return JSON format with two fields:
> - "comment": Review content (within 50 words)
> - "rating": Integer score (0-10, 10 is highest)
>
> Scoring Guide
> - Very unpleasant dialogue, wanting to call the manager, complain, or leave immediately → Negative review (0-3)
> - Long waiting time for food/queue, but pleasant conversation → Neutral review (4-7)
> - Smooth ordering and dining experience, comfortable and pleasant dialogue → Positive review (8-10)
> - Consider historical dialogue first, then consider waiting time.
>
> Current Context
> {context}

References

1. Buckland, M. *Programming Game AI by Example*; Jones & Bartlett Learning, 2004.
2. Colledanchise, M.; Ögren, P. Behavior Trees in Robotics and AI, 2018. https://doi.org/10.1201/9780429489105.
3. Park, J.S.; O'Brien, J.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative agents: Interactive simulacra of human behavior. In Proceedings of the Proceedings of the 36th annual acm symposium on user interface software and technology, 2023, pp. 1–22.
4. Gao, C.; Lan, X.; Lu, Z.; Mao, J.; Piao, J.; Wang, H.; Jin, D.; Li, Y. S3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984* **2023**.
5. Li, N.; Gao, C.; Li, M.; Li, Y.; Liao, Q. Econagent: large language model-empowered agents for simulating macroeconomic activities. *arXiv preprint arXiv:2310.10436* **2023**.
6. Hou, A.B.; Du, H.; Wang, Y.; Zhang, J.; Wang, Z.; Liang, P.P.; Khashabi, D.; Gardner, L.; He, T. Can A Society of Generative Agents Simulate Human Behavior and Inform Public Health Policy? A Case Study on Vaccine Hesitancy. *arXiv preprint arXiv:2503.09639* **2025**.
7. Wang, L.; Gao, H.; Bo, X.; Chen, X.; Wen, J.R. Yulan-onesim: Towards the next generation of social simulator with large language models. *arXiv preprint arXiv:2505.07581* **2025**.
8. Qian, C.; Liu, W.; Liu, H.; Chen, N.; Dang, Y.; Li, J.; Yang, C.; Chen, W.; Su, Y.; Cong, X.; et al. ChatDev: Communicative Agents for Software Development, 2024, [arXiv:cs.SE/2307.07924].
9. Jansen, P.; Côté, M.A.; Khot, T.; Bransom, E.; Mishra, B.D.; Majumder, B.P.; Tafjord, O.; Clark, P. DISCOVERYWORLD: A Virtual Environment for Developing and Evaluating Automated Scientific Discovery Agents, 2024, [arXiv:cs.AI/2406.06769].
10. Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Zhang, C.; Wang, J.; Wang, Z.; Yau, S.K.S.; Lin, Z.; et al. MetaGPT: Meta programming for a multi-agent collaborative framework. International Conference on Learning Representations, ICLR, 2024.
11. Li, J.; Lai, Y.; Li, W.; Ren, J.; Zhang, M.; Kang, X.; Wang, S.; Li, P.; Zhang, Y.Q.; Ma, W.; et al. Agent hospital: A simulacrum of hospital with evolvable medical agents. *arXiv preprint arXiv:2405.02957* **2024**.
12. Xi, Y.; Mao, X.; Li, L.; Lin, L.; Chen, Y.; Yang, S.; Chen, X.; Tao, K.; Li, Z.; Li, G.; et al. Kuileixi: a chinese open-ended text adventure game. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, 2021, pp. 175–184.
13. Peng, X.; Quaye, J.; Rao, S.; Xu, W.; Botchway, P.; Brockett, C.; Jojic, N.; DesGarennes, G.; Lobb, K.; Xu, M.; et al. Player-driven emergence in llm-driven game narrative. In Proceedings of the 2024 IEEE Conference on Games (CoG). IEEE, 2024, pp. 1–8.

14. Qiu, L.; Pu, X.; Feng, Y.; He, T. LLMs vs. Chinese Anime Enthusiasts: A Comparative Study on Emotionally Supportive Role-Playing. *arXiv preprint arXiv:2508.06388* **2025**.
15. Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* **2023**.
16. Jin, X.; Wang, Z.; Du, Y.; Fang, M.; Zhang, H.; Wang, J. Learning to discuss strategically: A case study on one night ultimate werewolf. *Advances in Neural Information Processing Systems* **2024**, *37*, 77060–77097.
17. Ma, W.; Mi, Q.; Zeng, Y.; Yan, X.; Lin, R.; Wu, Y.; Wang, J.; Zhang, H. Large language models play starcraft ii: Benchmarks and a chain of summarization approach. *Advances in Neural Information Processing Systems* **2024**, *37*, 133386–133442.
18. Yu, X.; Fu, J.; Deng, R.; Han, W. Mineland: Simulating large-scale multi-agent interactions with limited multimodal senses and physical needs. *arXiv preprint arXiv:2403.19267* **2024**.
19. Sudhakaran, S.; González-Duque, M.; Freiberger, M.; Glanois, C.; Najarro, E.; Risi, S. Mariogpt: Open-ended text2level generation through large language models. *Advances in Neural Information Processing Systems* **2023**, *36*, 54213–54227.
20. Sun, Y.; Li, Z.; Fang, K.; Lee, C.H.; Asadipour, A. Language as reality: a co-creative storytelling game experience in 1001 nights using generative AI. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2023, Vol. 19, pp. 425–434.
21. Christiansen, F.R.; Hollensberg, L.N.; Jensen, N.B.; Julsgaard, K.; Jespersen, K.N.; Nikolov, I. Exploring presence in interactions with llm-driven npcs: A comparative study of speech recognition and dialogue options. In Proceedings of the Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology, 2024, pp. 1–11.
22. Buongiorno, S.; Klinkert, L.; Zhuang, Z.; Chawla, T.; Clark, C. PANGeA: procedural artificial narrative using generative AI for turn-based, role-playing video games. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2024, Vol. 20, pp. 156–166.
23. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* **2023**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.