Balancing Emergence and Controllability of Dynamic-Generative Games through Human-AI Co-Authorization

Minji Kim*
minji.kim@hcs.snu.ac.kr
Seoul National University
Seoul, Korea

Abstract

The rapid advancement of Generative AI has introduced new opportunities in game design by enabling dynamic content generation during runtime. While dynamic-generative games offer rich and adaptive player experiences, designing such games presents unique challenges: ensuring contextual coherence and mitigating the uncertainty of the generative models. We present OzWon, a dynamic-generative game co-directed by a human Game Master (GM) and a generative AI agent. OzWon is a text-driven role-playing game in which players freely interact with the game by declaring their actions through text inputs. The LLM-based system interprets players' actions, generates output dialogues, and systematically progresses the game scenario. OzWon introduces a dedicated runtime authorization framework that allows the GM to efficiently supervise and revise dynamically generated content. We conducted a user study involving 22 participants to observe how GMs and the players engage with the dynamic-generative gameplay under our framework. Our qualitative findings identify four distinctive authorization strategies, revealing how users perceive and behave in co-directing the dynamic-generative gameplay with the AI system.

CCS Concepts

• **Human-centered computing** → **Empirical studies in HCI**.

Keywords

Game Design, Generative AI, Human-AI Interaction

ACM Reference Format:
Minji Kim. 2018. Balancing Emergence and Controllability of Dynamic-Generative Games through Human-AI Co-Authorization. In *CHI '26 Workshop: ACM Conference on Human Factors in Computing Systems, April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The recent advancement of generative AI has led to the emergence of entirely new forms of digital games, where the content is dynamically generated on demand at runtime. Instead of providing a set of pre-defined choices, the dynamic-generative game system enables players' spontaneous behaviors by generating responsive content at runtime, providing high player agency and personalized experience. However, designing a plausible player experience in dynamic-generative games poses a unique challenge due to an inherent technical limitation of generative AIs. It is virtually impossible to guarantee the quality of the runtime-generated components. Ensuring that generated content aligns with the game's context requires dedicated logical and contextual constraints, which degrade output quality in high chance [3, 4]. Moreover, generation delays caused by multiple rounds of validation and refinement severely interrupt the player experience.

In this work, we explore designs to leverage human–AI co-authorization to overcome the aforementioned challenges of dynamic generative games while providing engaging authorizing experiences. We present Oz-in-Wonderland (OzWon), a text-driven role-playing adventure game co-directed by a human Game Master (GM) and a Large Language Model (LLM) agent. The LLM agent functions as the core system, not only producing narration and dialogue but also executing system behaviors such as changing scenes, updating player states, and playing interactable events. The player progresses through the game by freely declaring their actions through text inputs, and the system generates the corresponding outputs in a turn-based manner.

Inspired by the TableTop Role-playing Game (TRPG), we incorporated GM user as a unique role who participates in the gameplay along with the player. The GM supervises the runtime-generated content and improvisationally co-directs the scenario. Before the generated output is delivered to the player, the GM pre-reviews and revises the output, fixing potential errors and modifying content in cases to direct gameplay in a plausible flow. By developing OzWon, we investigate design strategies for i) supporting users in properly authorizing the dynamic-generative games and ii) directing generative models to provide engaging game content under human-AI co-authorization.

To examine the authorization behaviors and user experiences during actual gameplays, we conducted a user study involving 22 participants (11 GMs and 11 players), where each GM-player pair played an exemplary scenario through the OzWon system. The majority of participants enjoyed the gameplay and were able to successfully progress through the scenario, even when the generative system produced critical errors, including logical inconsistencies and hallucinations. Through in-depth observation and qualitative analysis, we identified distinct authorization strategies of GM participants and players' varying expectations addressing both emergence and controllability. Based on our study results, we discuss broader design implications for supporting runtime authorization for dynamic-generative systems and providing a satisfying player experience in generative games.

CHI '26, Barcelona, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

2 Related Work

2.1 Dynamic Content Generation

For decades, game design has explored dynamic content generation to enhance player agency and enrich gameplay by providing a variety of interactive content during playtime. Procedural Content Generation (PCG) is a prominent approach to automatically generate multiple variations of context-specific game content [29]. Beyond efficiently generating extensive content, PCG is widely adopted to provide a variety of content during runtime. Rogue [12] is a representative game that provided dynamically generated content at an early stage, where the level structures, enemies, and items are all randomly generated at each playthrough.

Recent advances in AI techniques have enabled real-time generation of sophisticated multi-modal components involving design artifacts and interactive behaviors. Generative models enable systems to produce assets adapted to various game contexts without a dedicated training dataset, allowing runtime generation of multi-modal assets including 3D meshes and textures [10, 25, 26, 35]. Moreover, LLMs provide systems with the ability to deeply reason about dynamic scenes, player interactions, and game mechanics, facilitating the generation of design elements including narratives and quests [2, 20, 31, 37]. Multiple studies further extend the scope of dynamic content through interactive behavior generation. Park et al. [24] provided an initial generative framework where the multiple LLM-powered agents dynamically plan their behaviors and spontaneously interact with the game environment in runtime. LLMR [9] presented a holistic Mixed Reality framework for runtime generation and modification of interactive 3D scenes. GROMIT [14] developed a runtime behavior generation system where users can dynamically create interactions within the virtual scene. DreamGarden [11] presented a system that hierarchically plans and generates an interactive scene from a high-level user prompt. Such works involve multiple sequential LLM modules to interpret user input, understand scene context, plan for the generated structure, generate multi-modal components for the output (including codes and assets), and iteratively debug the final output.

However, runtime generation still reports high rates of failures and requires large latency in the LLM-driven iterative refinement process. Previous work reported an average latency of about 90 seconds for a single prompt in asset generation, with even longer delays around 170 seconds for processing sequential prompts [9], which would substantially degrade player experience in a game context. Our work aims to provide a practically satisfying user experience with dynamic generation by involving real-time refinement through human-AI cooperation. We investigate designs to efficiently support GM users in verifying and revising the AI generation.

2.2 Authorizing Generative AIs

Despite its transformative ability in providing extensive content on demand, dynamic generation involves an inherent risk of uncertainty that the designers cannot completely control the content generated at runtime. Studies frequently report that LLMs have a high chance of hallucination and errors when the logical complexity of the context grows, which is also a common issue in game content generation [27]. Diverse research tackles such limitations through technical improvements. Retrieval-Augmented Generation (RAG) [19] and Chain-of-Though (CoT) [33] methods are well-known techniques to minimize uncertainty in LLM reasoning. Constrained generation approaches improve the reasoning capability of the LLMs by providing strict generation constraints to the models through dedicated decoding [4, 5]. Several studies present iterative LLM module structures to identify errors or self-verify the result [15, 16, 30, 34].

Recent research in the HCI domain investigates systems to support users in authorizing AI-generated results. Diverse studies explore interface designs that help users understand the generation space and efficiently realize their design intents through iterative refinement processes, including adjusting parameters and partially modifying the output [9, 11, 17, 32, 36]. Research including Guzdial et al. explored mixed-initiative level design system that enables users to sequentially achieve their design goals in a turn-based co-creation procedure with AI agents [1, 8, 13]. Reza et al. provided a writing interface that enables users to rapidly explore multiple generative variants and refine their creation [28]. Lu et al. presented a visualized interface to support users in properly predicting runtime behaviors and authorizing complex interactive narrative space co-generated with an LLM agent [20].

Following such a line of work, OzWon applies a runtime authorization interface to the dynamic-generative game system. We delegate complex decisions occurring in runtime generation to human users, considering that such context-specific judgment lies beyond the capability of the current AI techniques. We separated the roles between the GM and the player user, encouraging GM users' manual authorization to be performed in a playful manner while ensuring players engage in the gameplay without being aware of the AI performance.

3 System Overview

OzWon is a text-driven role-playing adventure game that progresses in a turn-based manner. The player and the GM participate in the session together through dedicated clients. The player client consists of a text input interface, a dialogue window, and a 2D scene image (Figure 1-A). At each turn, the player declares their behavior through a text input, which is sent to both the GM client and the LLM-powered generator server. The generator takes the player's input along with the current scene information, the overall scenario, and the ongoing game progress.

Upon the generation, the output is first sent to the GM client. The GM reviews it and decides whether to accept or revise (Figure 1-C). For the revision, the GM may directly modify the text or regenerate the entire output (Figure 1-D, details in Section 4.3). As the GM finalizes the revision and confirms the output, the player receives the result. The player client displays narrations and dialogues, while executing system events according to the output (Figure 1-B). For instance, when the player declares *"I will take the cookie"*, the narration *"You put the cookie in your pocket"* is displayed, a *cookie* item pop-up appears on the screen, and the player's inventory is updated with the *cookie* item. The generative system primarilly produces outputs to direct the player into a predefined scenario. The scenario is constructed through the GM client before the gameplay.

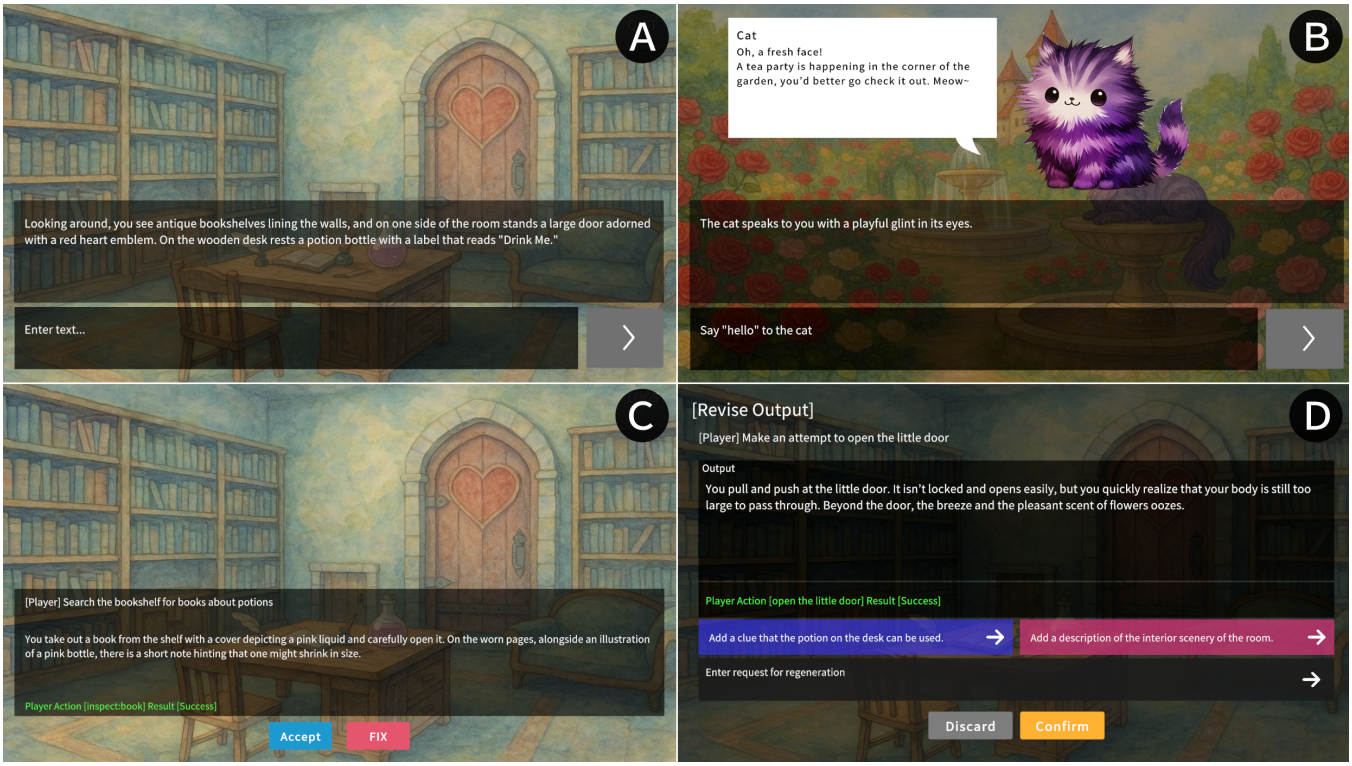

Figure 1: Player and GM clients' screens. A) Idle player client interface, displaying narration and a text input interface. B) NPC portrait and dialogue bubbles are displayed when the player is talking with NPCs. C) The generation output is sent to GM client, waiting for them to decide whether to accept or revise. D) Revision menu of the GM client. The GM is given three options: directly modifying the output text, running automatic regeneration based on AI-generated suggestions (blue and red buttons), or manually entering a request prompt for the regeneration.

4 System Design

4.1 Generative Modules

The generator consists of two consecutive LLM modules: the output generator and the validator. The generated output consists of narrative output, system output, and suggestions for the GM. The **narrative output** includes the narration and the NPC dialogues that are directly delivered to the player. The **system output** consists of the system-interpreted keywords, including player action, target of the action, action result, and triggered events. Once the output is generated, the validator module verifies system-level errors.

The generative system of OzWon interprets the gameplay context organized by scene-level units where the player is currently located. The scene defines the list of interactive components (NPCs, items, and events) associated with the scene. Each component includes a short descriptive text and an instruction for the generator agent. Figure 2 illustrates an example of the component structure.

We employed the GPT-5-mini model [23] for the generator and GPT-o4-mini [22] for the validator after iterative prototyping cycles. For the image generation (utilized at scenario construction and playtime component generation), we adopted the DALL-E 3 model [21]. In order to minimize generation latency, we streamlined the generator module to include only the generation and validation stages. The full generation process of our architecture takes around 30 seconds.

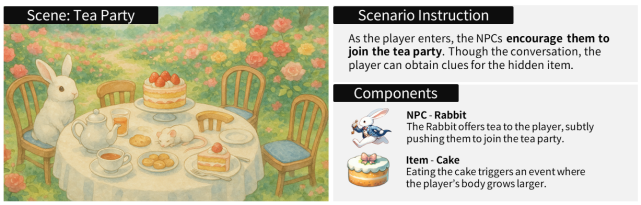

Figure 2: An example scene component utilized in the OzWon system. A scene includes a background image sprite, a default description, a list of interactive components (e.g., NPCs, items, events), and the instructions for the generator module.

4.2 Player Action Interpretation

For each player input, the system identifies the type of attempted player action, its target, and the resulting outcome. The outcomes are generated based on the scenario and the gameplay context. The generator primarily determines action results following the scenario-defined instructions; if a specific intended action is provided in the scenario, the system produces the predefined results. Otherwise, the system accepts logically feasible actions within the context and generates corresponding results. Implausible or contextually impossible actions are evaluated as failure. When the system detects

any action that conflicts with the scenario, it requests the GM's judgment.

4.3 Authorization Interface

As illustrated in Figure 1-C, the GM can choose to either confirm or revise the generated output at each turn. Choosing Accept delivers the generated output directly to the player, while choosing Revision transitions to the dedicated revision menu. Exceptionally, when the generator requests GM judgment, the GM is forced to review and revise the result.

For the revision, the GM is provided three revision options: direct modification, automatic regeneration with suggestions, and manual regeneration with revision requests. **Direct modification** allows the user to manually edit the narration and dialogue texts. **Automatic regeneration** provides two automatically regenerated suggestions. Specifically, one of the suggestions guides the player toward the next intended behavior in the scenario, while the other one encourages a diversified player action independent of the scenario. **Manual regeneration** enables the user to prompt their own revision request to the generator. Regeneration proceeds in a similar manner to the initial generation, involving output generator and validator modules, typically taking about 30 seconds in generation.

5 User Study

5.1 Settings

5.1.1 Participants. We recruited 11 GMs and 11 player participants through an online survey posted in online game communities and social networks. Since the GM and player roles differ, we separately recruited participants for each group. All participants were aged between 21-34 (mean=26.82, SD=3.65). 13 participants identified as female, and the remaining 9 as male. During recruitment, the participants submitted a survey measuring their level of experience with generative AIs (e.g., LLM usage) and games in a 5-point Likert scale, and their previous TRPG experiences. Every participant possessed a certain level of experience in playing digital games. Additionally, we checked whether the GM and player in each session had a preexisting acquaintance. All participants were South Korean nationals, and both the play sessions and interviews were conducted in Korean. We translated the narrative texts from the play sessions and the interview data into English in the paper. We detail the participant demographics in Table .

5.1.2 Procedure. The study was conducted online via Zoom sessions. Each session lasted for 75 minutes. During the first 15 minutes, the GM participates in the session alone. Prior to the experiment, the GM participants were given the game scenario and freely took time to review it. After joining the session, the GM was guided to decide the expected scenario progression, instructed about the authorization interface, and familiarized themselves with the experimental tasks through a short tutorial.

After completing the GM tutorial, the player enters the session, and the two participants engage in 45 minutes of gameplay. Players were encouraged to complete the scenario in a free manner. The GMs were guided to authorize AI outputs to properly support the player in completing the game. The GM was set free to decide whether and how to revise the AI outcomes. Participants were allowed to freely communicate and coordinate strategies for the gameplay. However, we restricted GMs from directly revealing the scenario to the player. Finally, we conducted a 10-minute semi-structured interview and instructed each participant to complete a post-study survey about their gameplay experience and system usability. The entire session was recorded with consent.

5.1.3 Game Scenario. We provided a single pre-defined scenario in the study to ensure consistency. The provided scenario features an adventure game themed around *Alice-in-Wonderland* [7] composed of five scenes. The player could clear the scenario by completing eight key events. Through an internal pilot run within the research group, we estimated the scenario completion time to be 15 minutes. However, this turned out to be a mistake, since only four pairs succeeded in clearing during the 45-minute session.

5.2 Results and Findings

5.2.1 Measurement and Analysis. To investigate participants' gameplay progression and authorization patterns, we conducted a comprehensive analysis of system logs and interpreted user behaviors. For the qualitative analysis, we conducted a thematic analysis [6] of the interview transcripts and the open-ended responses from the post-study survey. The first author led the analysis procedure and conducted open coding of the interview transcripts. Subsequently, two authors with over five years of HCI research experience identified the resulting themes through multiple rounds of discussion.

5.2.2 Overall Gameplay Trends. The play session proceeded for 20.64 turns on average (SD=4.59, MIN=15, MAX=29). The GMs revised about half of the results, accepting 9.27 (SD=3.82) turns and revising 11.36 (6.33) turns on average. We detail the values in Table 2. We observed substantial variance in both the total number of turns and the time spent per turn across sessions. The average time spent per turn was 113.98 seconds (SD=61.31), with an average of 36.42 seconds (SD=9.10) of generation delays. The GM's result revision process took 71.54 seconds on average (SD=53.85).

Among the participants, 4 pairs cleared the scenario (P2, 6, 7, and 10) while the others failed to proceed to the end. In general, the pairs who cleared the scenario proceeded a larger number of turns than the average: P6, 7, and 10 proceeded 25, 29, and 27 turns each. Exceptionally, P2 completed the scenario in 17 turns.

5.2.3 GMs' Authorization Strategies. We identified four representative strategies in the GMs' authorization. Below, we describe the key criteria of GMs for choosing revisions and the methods they employ.

Correcting errors in the generated output. The most common reason for revision was to correct the explicit errors in the generation. This included **sentence errors** where the narration or dialogue contains incorrect grammar or awkward expressions, and the **logical errors** where the agent misinterprets the player action or the context. The GMs mostly preferred to manually correct the explicit sentence errors, while they utilized regeneration to fix the logical errors. Interestingly, all participants (11/11) detected and corrected sentence errors in NPC dialogues, while the errors in narration were modified relatively less (6/11). We observed that all GMs accepted at least once a narration containing an explicit sentence error (e.g., inconsistent honorific expressions, syntax errors, or disclosure of system terms).

Table 1: Participant demographics. The GM and PL on the same row participated together in the session.

Id	AI Knowledge	TRPG Experience	Id	AI Knowledge	TRPG Experience	Prior Acquaintance
GM1	2	No	PL1	2	Player & Master	Yes
GM2	2	Player	PL2	2	Player	Yes
GM3	5	No	PL3	2	No	Yes
GM4	2	Player & Master	PL4	2	No	Yes
GM5	2	Player	PL5	1	No	Yes
GM6	5	Player	PL6	4	No	Yes
GM7	2	No	PL7	3	No	Yes
GM8	4	No	PL8	3	No	No
GM9	3	No	PL9	2	No	Yes
GM10	4	Player	PL10	4	Player	No
GM11	3	Player	PL11	2	Player	No

	Total Turns	Accepted Turns (%)	Revised Turns (%)	Regenerated Turns (%)
All	20.63	44.93	55.06	29.52
Cleared groups	24.5	38.78	61.22	38.78
Non-cleared groups	18.42	49.61	50.39	22.48

Table 2: General trends of GM authorization. The values denote the average number of turns and percentage of GMs' authorization decisions in each group. Note that the turns utilizing regeneration were also included in the revised turns.

Guiding player toward the scenario. The majority of the GMs (GM1, 3-11) frequently revised the outcomes to provide guidance for the player toward the intended scenario. Interestingly, the way to provide guidance varied across the participants. As an illustrative case, in the first scene, the player has to drink the potion on the desk, which makes their body shrink, and then enter the small door on the wall. In this scene, most GMs provided guidance to the player through their own unique approaches. GM1 added an explicit hint through regeneration, notifying that *drinking the potion might make your body smaller enough to enter the small door.* On the other hand, GM8 replaced the generator's direct suggestion—*drinking the potion might help*—with a more implicit cue: *The glass bottle on the desk glimmers in the sunlight*. GM4, instead of describing the potion through narration, added the description *you find an illustration of a pink glass bottle among the books on the shelf* when the player examined the bookshelf. They then induced the player to discover information about the potion through the book.

Encouraging player to explore. We observed that some of the GMs (GM4-6 and 8-11) also continuously revised the output to remove the scenario-related components and encourage players to explore more. This type of revision was carried out in two main ways. The first group of participants (GM4-5, 8, and 11) directly removed or replaced explicit scenario-inducing cues in the narration text. The others (GM6 and 9-10) utilized regeneration; interestingly, in these cases, they consistently began by manually removing the undesired cues and then specified what should be regenerated in their place.

Creatively modifying the output. Another frequent way of authorization among the GMs was to proceed with the scenario into their own creative intents by adding descriptions or interactions that do not exist in the original scenario. For instance, GM10 attempted to recreate the NPC's personality by requesting, *'let the heart queen speak in a southwestern dialect, and add a huge laughter at the end of the dialogue.'* Notably, several participants actively utilized manual regeneration (GM4, 6-7, and 9-11) to add their creative intent. In some cases, the GMs created original narratives in order to guide the player through the scenario. For instance, GM7 included the phrase *You hear the footsteps of trump soldiers approaching to catch you* to encourage the player to quickly exit a scene that had already been explored.

The major authorization strategies considerably varied between individual participants. For instance, GM4 and GM8 refrained from triggering the regeneration feature and manually edited most of the outputs. When asked for the reason, they responded that they chose to directly revise the output to minimize delays in each turn. GM3 added: *"If the generation time had been within 5–10 seconds, I probably would have actively tried the regeneration. But (in the study,) since the generation already took quite a while, I fixed (the results) directly to complete it faster."*. In contrast, GM9 and GM10 frequently utilized regeneration in most cases, even when correcting the sentence errors; GM9 utilized regeneration for every revision, and GM10 employed regeneration 19 times out of 21 revision instances.

On the other hand, GM2 showed a high acceptance of the generated results, accepting 14 outputs among the total 17 turns, even when the narrative output included explicitly low sentence quality (e.g., the narration text reveals a system keyword to the player). Nevertheless, PL2 successfully completed the game in relatively few turns and reported in the post-session interview that they did not notice any significant issues in the narrative text.

5.2.4 Player Experiences. We were delighted that all 22 participants explicitly indicated that the play session was interesting and enjoyable.

Several groups (P3-4, P6, P8, and P10) even asked if they could play the session once again, apart from the experiment. The participants commonly appreciated a strong sense of player agency, emergent experiences, and high levels of immersion. GMs also expressed enjoyment in that the session felt more like a collaborative gameplay experience rather than completing a task. In particular, GM participants highlighted a distinct sense of experience in that they enjoyed the sense of creatively driving the scenario (GM6, 8-10) and providing a playful experience for the player (GM3 and 11): *"Being a game master seems even more fun than being a player. I can control the situation as I want, and the AI provides the initial suggestions so that I can keep (proceeding in the game) with enjoyment."* (GM9). However, a majority of GM participants also expressed common frustration related to generation delays. Five of them (GM4-6, 9-10) reported that they were unable to authorize the scenario as they wanted due to the concern about making players wait too long.

5.2.5 System Usability. Overall, users evaluated OzWon as having appropriate system usability and provided high scores on the System Usability Scale [18]; 83.64 for players and 74.32 for GMs on average. When we asked about the most critical aspect degrading the usability, the majority of the participants (6 players and 8 GMs) mentioned the long delay of the turn progression. Compared to player participants, GMs expressed greater difficulty with system usage, commonly noting that it was hard to revise outputs when the AI failed to produce intended results (GM3-4, GM6-7, and GM9).

6 Discussion

6.1 Effect of Employing Human Supervisor to Co-Direct the Dynamic-Generative Gameplay

We identified that the GM played a critical role in making highly context-specific decisions that are beyond the reasoning capabilities of the generative AI system. Our study showed an overall tendency that the GM user's performance significantly influences the success of the gameplay. The groups that cleared the game progressed a higher number of total turns (which implies they spent less time understanding and revising outputs), and more frequently utilized revision and regeneration compared to the groups that did not. GMs in successful groups were often good at recognizing the misleading caused by the sequential hallucination of the generative system and effectively guided the players back to the desired scenario.

6.2 Design Implications

6.2.1 Effects of multiple authorization methods. We suggest offering multiple authorization methods and tailored suggestions to better support users' varying authorization intents. The participants in our study commonly preferred regeneration with manual prompts instead of adopting the suggestions offered by the LLM-driven generator when they had specific regeneration intents. AI-generated suggestions were limited in supporting specific tasks, but were successful in providing effective alternatives when resolving explicit system-level errors that human supervisors face difficulties in revising.

6.2.2 Minimizing generation latency. The generation latency has a crucial impact on both GM and player experiences. In our study, most participants reported that generation latency (36.42 seconds per generation) was the primary source of frustration in their experience. This tendency was more remarkable among GM participants, as some of them attempted to complete authorization as quickly as possible, considering that the generation had already taken a long delay. GM9 stated that they would *"rather have shorter generation time in exchange for reduced quality"*, in that they could manually improve the final output if they were provided longer authorization time.

Unfortunately, within our current exploration, it is difficult to realize such a tradeoff below a certain latency. Generating outputs that are executable at the system level inherently takes a considerable amount of time, as it involves sequential logical reasoning involving understanding the game context and verifying the validity of the system-level output. We consider that further improvements in generation latency are constrained by the current technical capabilities of LLMs. Nevertheless, we emphasize that minimizing generation latency as much as possible is indispensable for ensuring both the quality of authorization and the player experience. Future works may explore dedicated methods to achieve an optimal balance between generation delay and output quality.

6.3 Limitations

6.3.1 Limitation in participant size and demographics. The number of participants was relatively small, which restricts the statistical robustness of the results. The system may have included language-specific limitations that could influence both the performance of the model and the participants' experiences.

6.3.2 Constraints in study settings. The constraints in the user study settings might have posed limitations in the results. The playtime provided during the experiment was relatively short, inducing many participants to primarily concentrate on completing the scenario within the allotted time rather than proceeding with the game in a free manner. This time pressure may have also introduced negative effects on the player experience.

7 Conclusion

We present OzWon, a dynamic generative game co-directed by a human Game Master and a generative AI agent. We demonstrate how human–AI co-authorization can serve as a viable design strategy to address these challenges, balancing system controllability with emergence to provide high player agency and improvisational richness. Our work not only illustrates the feasibility of creating engaging player experiences under these constraints but also offers design insights for future dynamic-generative systems that aspire to combine the creativity of AI with the discernment of human guidance.

References

[1] Alberto Alvarez, Steve Dahlskog, Jose Font, Johan Holmberg, Chelsi Nolasco, and Axel Österman. 2018. Fostering creativity in the mixed-initiative evolutionary dungeon designer. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*. 1–8.

[2] Trevor Ashby, Braden K Webb, Gregory Knapp, Jackson Searle, and Nancy Fulda. 2023. Personalized quest and dialogue generation in role-playing games: A knowledge graph-and language model-based approach. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–20.

[3] Anirudh Atmakuru, Jatin Nainani, Rohith Siddhartha Reddy Bheemreddy, Anirudh Lakkaraju, Zonghai Yao, Hamed Zamani, and Haw-Shiuan Chang. 2024. Cs4: Measuring the creativity of large language models automatically by controlling the number of story-writing constraints. *arXiv preprint arXiv:2410.04197* (2024).

[4] Debangshu Banerjee, Tarun Suresh, Shubham Ugare, Sasa Misailovic, and Gagandeep Singh. 2025. CRANE: Reasoning with constrained LLM generation. *arXiv preprint arXiv:2502.09061* (2025).

[5] Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2024. Guiding llms the right way: Fast, non-invasive constrained generation. *arXiv preprint arXiv:2403.06988* (2024).

[6] Virginia Braun and Victoria Clarke. 2012. *Thematic analysis.* American Psychological Association.

[7] Lewis Carroll. 2024. Alice in wonderland. *Drama & Theatre* 2024, 116 (2024), 39–39.

[8] Megan Charity, Ahmed Khalifa, and Julian Togelius. 2020. Baba is y'all: Collaborative mixed-initiative level design. In *2020 IEEE Conference on Games (CoG)*. IEEE, 542–549.

[9] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. Llmr: Real-time prompting of interactive worlds using large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–22.

[10] Sam Earle, Filippos Kokkinos, Yuhe Nie, Julian Togelius, and Roberta Raileanu. 2024. Dreamcraft: Text-guided generation of functional 3D environments in Minecraft. In *Proceedings of the 19th International Conference on the Foundations of Digital Games*. 1–15.

[11] Sam Earle, Samyak Parajuli, and Andrzej Banburski-Fahey. 2025. DreamGarden: A Designer Assistant for Growing Games from a Single Prompt. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–19.

[12] Graybeard Games. 1980. Rogue. https://store.steampowered.com/app/1443430/Rogue/. Accessed: 2025-10-11.

[13] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O Riedl. 2019. Friend, collaborator, student, manager: How design of an ai-driven game level editor affects creators. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.

[14] Nicholas Jennings, Han Wang, Isabel Li, James Smith, and Bjoern Hartmann. 2024. What's the Game, then? Opportunities and Challenges for Runtime Behavior Generation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–13.

[15] Junho Kim, Hyunjun Kim, Kim Yeonju, and Yong Man Ro. 2024. Code: Contrasting self-generated description to combat hallucination in large multi-modal models. *Advances in Neural Information Processing Systems* 37 (2024), 133571–133599.

[16] Florian Leiser, Sven Eckhardt, Valentin Leuthe, Merlin Knaeble, Alexander Maedche, Gerhard Schwabe, and Ali Sunyaev. 2024. Hill: A hallucination identifier for large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–13.

[17] Joanne Leong, David Ledo, Thomas Driscoll, Tovi Grossman, George Fitzmaurice, and Fraser Anderson. 2025. Paratrouper: Exploratory Creation of Character Cast Visuals Using Generative AI. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–20.

[18] James R Lewis. 2018. The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction* 34, 7 (2018), 577–590.

[19] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.

[20] Zhuoran Lu, Qian Zhou, and Yi Wang. 2025. WhatELSE: Shaping narrative spaces at configurable level of abstraction for AI-bridged interactive storytelling. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–18.

[21] OpenAI. 2023. DALL-E 3. https://openai.com/dall-e-3.

[22] OpenAI. 2023. GPT-4 Technical Report. *arXiv* abs/2303.08774 (2023). doi:10.48550/arXiv.2303.08774

[23] OpenAI. 2025. Introducing GPT-5. https://openai.com/gpt-5.

[24] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*. 1–22.

[25] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).

[26] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, et al. 2023. Dreambooth3d: Subject-driven text-to-3d generation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2349–2359.

[27] Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922* (2023).

[28] Mohi Reza, Nathan M Laundry, Ilya Musabirov, Peter Dushniku, Zhi Yuan "Michael" Yu, Kashish Mittal, Tovi Grossman, Michael Liut, Anastasia Kuzminykh, and Joseph Jay Williams. 2024. Abscribe: Rapid exploration & organization of multiple writing variations in human-ai co-writing tasks using large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–18.

[29] Noor Shaker, Julian Togelius, and Mark J Nelson. 2016. Procedural content generation in games. (2016).

[30] Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. 2024. On the self-verification limitations of large language models on reasoning and planning tasks. *arXiv preprint arXiv:2402.08115* (2024).

[31] Yuqian Sun, Zhouyi Li, Ke Fang, Chang Hee Lee, and Ali Asadipour. 2023. Language as reality: a co-creative storytelling game experience in 1001 nights using generative AI. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 19. 425–434.

[32] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. 2024. Promptcharm: Text-to-image generation through multi-modal prompting and refinement. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–21.

[33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[34] Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2022. Large language models are better reasoners with self-verification. *arXiv preprint arXiv:2212.09561* (2022).

[35] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. 2024. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191* (2024).

[36] Xingchen Zeng, Ziyao Gao, Yilin Ye, and Wei Zeng. 2024. IntentTuner: an interactive framework for integrating human intentions in fine-tuning text-to-image generative models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–18.

[37] Andrew Zhu, Lara Martin, Andrew Head, and Chris Callison-Burch. 2023. CALYPSO: LLMs as Dungeon Master's Assistants. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 19. 380–390.