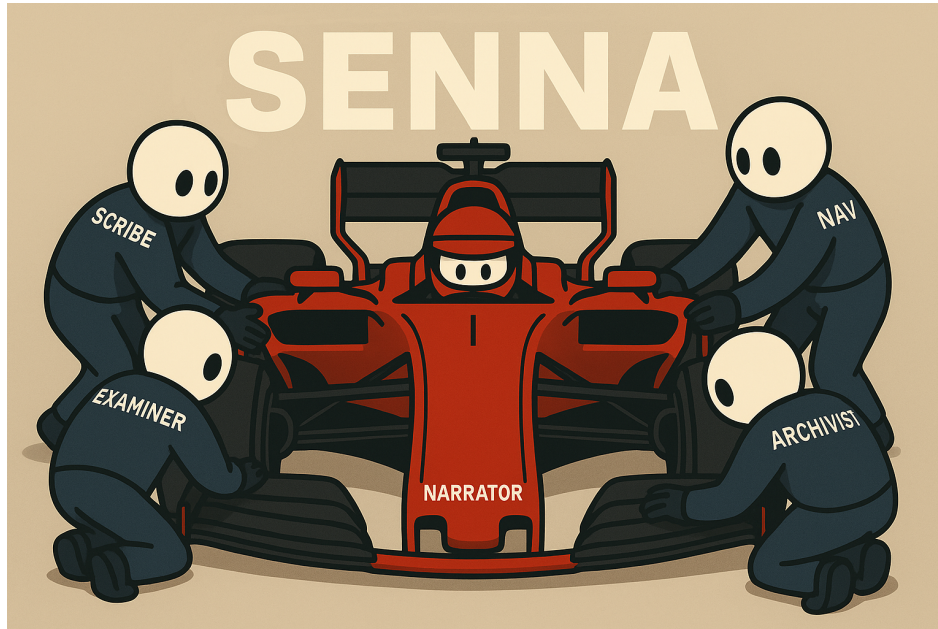


Summary



This paper presents *ChatRPG v3*, an advanced AI game master system designed to preserve narrative coherence while enabling interactive storytelling in single-player role-playing games. Building upon previous iterations, the authors introduce *SENNNA*, a multi-agent system composed of five specialized large language model (LLM) agents that collaboratively emulate a human Game Master (GM). *ChatRPG v3* is capable of ingesting and adhering to pre-written adventure modules, guiding players through structured storylines without sacrificing the sense of agency and immersion that defines role-playing games.

At the system’s core lies a *Narrative Graph*, a directed acyclic graph that encodes the story’s milestones, their dependencies, and associated conditions. This structure enables the AI to track a player’s narrative progression and ensure that key plot events unfold as intended. The five agents—Scribe, Examiner, Navigator, Narrator, and Archivist—divide traditional GM responsibilities. The Scribe translates the adventure module into the Narrative Graph. The Examiner assesses player actions for narrative validity. The Navigator updates the narrative state based on the player’s decisions. The Narrator generates the in-world story text, and the Archivist tracks game-world consistency.

A key innovation is the system’s handling of *narrative redirection*—instances where players attempt to act outside the intended story path. Rather than denying such actions abruptly, *ChatRPG v3* employs redirection strategies rooted in game design theory and expert Dungeon Master practices. Six strategies were developed: Hard Denial, More Information, Illusion of Choice, NPC Influence, Humor, and In-World Consequences. To validate these, five experienced GMs ranked the strategies by effectiveness. Based on their input, the study focused on the three most preferred strategies—offering information, NPC influence, and in-world consequences—alongside a baseline hard denial.

The system was evaluated through a structured user study involving 12 participants playing a modified single-player version of the D&D adventure *A Most Potent Brew*. Participants first played through the entire scenario and then compared alternative redirection strategies applied to selected narrative moments. Surveys and interviews were conducted to assess immersion, agency, and player satisfaction.

Results indicate that *ChatRPG v3* successfully maintained narrative coherence while allowing for substantial player freedom. Most participants described the experience as consistent, immersive, and comparable to a traditional one-shot TTRPG session. The system demonstrated an ability to integrate player creativity and even deviate from the module without breaking story logic. Participants generally accepted moments of narrative steering, especially when grounded in character or world logic, echoing real tabletop experiences.

Among redirection strategies, *In-World Consequences* emerged as the most preferred and effective approach, followed by *NPC Influence* and *More Information*. *Hard Denial* was consistently rated as the least appropriate. Statistical analysis confirmed that the choice of redirection strategy significantly influenced players’ perceived appropriateness and narrative smoothness, with players favoring strategies that preserved immersion and respected their agency.

Overall, this work offers a substantial contribution to the field of AI-driven interactive storytelling. By combining modular LLM agents with structured narrative tracking and empirically validated redirection strategies, *ChatRPG v3* bridges the gap between authored narrative and player-driven storytelling. The study underscores the potential of LLM-based game mastering while highlighting key challenges, such as preserving immersion and maintaining balance between freedom and structure. These findings offer practical design guidelines for future AI Game Masters and broader applications in narrative-based entertainment.

Narrative Adherence in LLM-driven Games

Nicolai H. Jørgensen

Department of Computer Science, Aalborg University
Aalborg, Denmark
njarge20@student.aau.dk

Sarmilan Tharmabalan

Department of Computer Science, Aalborg University
Aalborg, Denmark
stharm20@student.aau.dk

Abstract

This paper presents ChatRPG v3, a narrative-driven AI game master for single-player role-playing games. Building on prior versions, we introduce SENNA, a five-agent system that enables adherence to structured, pre-written adventure modules while preserving player agency. The system uses a narrative graph to track story progression. Furthermore, we design redirection strategies to guide players back to the intended narrative arc when deviations occur. We evaluate the effectiveness of SENNA and these strategies through a user study combining live gameplay, alternative scene comparisons, and player feedback. Results show that structured redirection techniques rooted in world logic, such as NPC influence and in-world consequences, enhance narrative coherence without diminishing immersion or autonomy. Our findings expand the design space for AI-driven interactive fiction and offer practical insights for balancing authored stories with emergent player input.

CCS Concepts

• **Computing methodologies** → **Natural language generation; Multi-agent systems.**

Keywords

Interactive Fiction, Narrative Adherence, Narrative Redirection, Role-Playing Games, Dungeons & Dragons, AI Game Master, ReAct, LangChain, Large Language Models, Multi-Agent System

1 Introduction

The growing integration of large language models (LLMs) into interactive entertainment experiences, particularly narrative-driven games, poses a compelling yet unresolved challenge: how to balance emergent player agency with structured, authored storytelling. As AI systems increasingly serve as autonomous Game Masters (GMs) in digital role-playing games (RPGs), they offer new opportunities for dynamic storytelling but also raise critical questions about narrative adherence, user agency, and player immersion.

In previous work [21], we introduced ChatRPG v1 and v2, a single-player, LLM-driven RPG system designed to emulate the experience of traditional tabletop role-playing games (TTRPGs). While these early systems successfully engaged users in interactive storytelling, they exhibited a recurring flaw: excessive compliance with player input at the expense of narrative consistency. Player actions that contradicted or bypassed core story events were often accepted uncritically by the AI GM, leading to incoherent plotlines and disrupted pacing.

To address these limitations, this paper presents ChatRPG v3, a new iteration of our AI GM system that introduces SENNA, a multi-agent system designed to ensure adherence to a predefined narrative structure. Unlike prior versions, ChatRPG v3 is capable of ingesting

a pre-written adventure module and guiding the player through its story beats while maintaining immersion and interactivity.

A central focus of this work is the challenge of narrative redirection: how to appropriately respond when players attempt actions that are unreasonable or outside the scope of the predefined story. In traditional tabletop games, human GMs employ subtle techniques to steer players back to the intended path while preserving their sense of autonomy. Inspired by this, we designed redirection strategies that could be used in an LLM-driven, text-based RPG.

To empirically assess the effectiveness of narrative adherence and redirection strategies, we conducted a structured user study using a custom adventure module and a within-subjects evaluation design. Participants played through a complete story experience guided by SENNA, after which they were exposed to alternative versions of key scenes featuring different redirection strategies. This allowed us to systematically measure player preferences and perceptions of narrative coherence, control, and enjoyment.

To summarize, we make the following contributions:

- The SENNA framework, a novel multi-agent AI GM architecture designed to ensure narrative adherence in LLM-driven games.
- An open-source implementation of the ChatRPG v3 system, available on our Github¹
- The design of narrative redirection strategies, based on best practices from interactive storytelling and expert GM feedback.
- An empirical user study evaluating the effectiveness of the SENNA system and the redirection strategies, providing practical recommendations and design implications for future AI-powered narrative systems.

2 Related Work

LLMs are increasingly integrated into digital games as players, non-player characters (NPCs), GMs, and player assistants [14], offering diverse applications but also introducing notable challenges. As GMs, LLMs dynamically generate interactive narratives, as seen in systems like AI Dungeon [4, 14, 19], provide encounter and world-building support (e.g., CALYPSO [51]), and assist human GMs with dialogue suggestions [22]. Despite their promise, AI GMs tend to over-comply with user prompts, often at the cost of game balance and narrative structure, and face issues with long-term coherence [11, 49] and world model consistency, sometimes missing key game details and introducing contradictions [22].

Sakellariadis [38] shows LLM-based GMs can produce vivid, immersive narratives, even occasionally outperforming human GMs in terms of sensory immersion when fine-tuned on curated play data [34]. However, these systems can constrain player agency

¹<https://github.com/KarmaKamikaze/ChatRPG>

through inflexible narrative paths [2, 16, 38], fail to convey crucial information for story progression [42], and exhibit risk-averse biases that dull narrative tension. These insights highlight the challenge of balancing narrative coherence with player agency in AI-driven role-playing games [38, 51].

Our previous work introduced ChatRPG [21], a solo RPG system driven by an AI GM. The initial version used prompt engineering to track game state within a single input prompt, enabling immersive single-player experiences but suffering from narrative inconsistency and limited scalability as sessions progressed. To address these issues, ChatRPG v2 employs a modular, multi-agent ReAct-based [48] architecture. Here, a Narrator agent handles storytelling, while an Archivist maintains world consistency, yielding substantial improvements in both player immersion and narrative coherence.

Recent work such as SHARI [28] employs a similar multi-agent GM decomposition, using Assess-Narrate-Update (ANU) tasks to evaluate, narrate, and update the game world, thus moderating player actions and ensuring narrative structure. SHARI’s approach parallels our division of narrative and archival roles, but also incorporates an Assess step to vet player actions—a functionality we aim to extend with a verification layer in ChatRPG. In summary, we build on prior approaches to strike a balance between narrative integrity and player freedom by structuring GM responsibilities across specialized AI agents and a proactive verification layer.

3 Problem Definition

In prior work [21], we introduced and evaluated ChatRPG, a single-player, LLM-powered RPG with a multi-agent architecture. While users found the experience valuable, two key limitations emerged: AI GMs often over-complied with player input to the detriment of narrative coherence [14], or, conversely, rigidly “railroaded” players, reducing agency [2, 16, 38]. These challenges underscore the critical need to balance narrative structure with player autonomy.

The goal of this work is to develop and evaluate an LLM-driven AI GM that is capable of faithfully following user-supplied story modules while preserving player freedom and agency. To this end, we address the following research questions:

RQ1: *How can an agentic, LLM-based system be designed to support single-player role-playing games that are both based on and constrained to user-provided stories?* We investigate methods for integrating pre-written adventure modules into an AI-driven framework, aiming to ensure that story progression remains coherent and faithful to the author’s intent without rendering the game experience overly linear or restrictive.

RQ2: *When the AI GM must redirect or constrain player actions that are unreasonable or outside the scope of the story, how can these redirections be designed to uphold an enjoyable and immersive game experience?* This question arises from known limitations of LLMs in the GM role, such as excessive compliance (“sycophancy”) or immersion-breaking hard denials. Our aim is to explore and evaluate redirection strategies that guide players back to the narrative arc while preserving a sense of agency, creativity, and engagement.

These questions frame our exploration of balancing narrative adherence and redirection in LLM-driven solo RPGs, with the goal of advancing the design of robust, interactive AI GMs.

4 Design

In this section, we explore the major technological advancements that have led to the development of ChatRPG v3’s narrative adherence system **SENNA**, which enables users to upload a pre-defined story module and have the AI GM follow a structured storyline. Furthermore, we describe the design of six different redirection strategies that could be employed by the AI GM when players diverge from the intended narrative.

4.1 Narrative Graph

To facilitate structured story progression in ChatRPG v3, we introduce the *Narrative Graph*, a directed acyclic graph (DAG) representation of the storyline. This structure encodes narrative milestones and the conditions under which they are sequentially unlocked, enabling the system to track the player’s progress through a pre-defined story module. An example of a small Narrative Graph is provided in Figure 1.

Each node in the graph represents a narrative milestone. This can, for example, occur when the player reaches a new location or meets a specific NPC. Nodes are uniquely identified by a name and contain a textual description that provides contextual detail about the milestone. Every node also maintains a *status* property, which tracks the player’s progression relative to that milestone. The status can take one of three values:

- **Undiscovered:** The player has not yet been introduced to this narrative element, typically because prerequisite nodes have not been completed.
- **Ongoing:** The player is currently engaged with the milestone, but has not yet satisfied all conditions required for progression.
- **Completed:** The player has fulfilled the conditions associated with the milestone.

Directed edges between nodes denote narrative dependencies: a source node must be completed before the target node becomes available. For example, in Figure 1, the player can only proceed to the ruins after resolving the encounter with the wolf. Each edge is annotated with a set of *conditions*—narrative or interactional constraints that must be satisfied for the source node to be marked as completed and the corresponding transition to the target node to be triggered. For ease of evaluation, these conditions should be formulated as boolean statements or yes/no questions. So, for the player to proceed to the ruins in Figure 1, they must both defeat the wolf and discover the tracks to the ruins. Edges also maintain a *status*, which can be either:

- **Unvisited:** The edge’s conditions have not yet been fulfilled.
- **Visited:** The conditions have been met, and the player has progressed along this narrative path.

The Narrative Graph always includes a designated **Start** node and **End** node, which define the boundaries of the narrative arc. It is important to note that the graph does not reflect the player’s current physical or narrative location within the game world, and should therefore not be updated when the player moves between locations previously explored. Instead, it tracks which milestones

have been revealed or achieved, functioning as a high-level model of narrative progression.

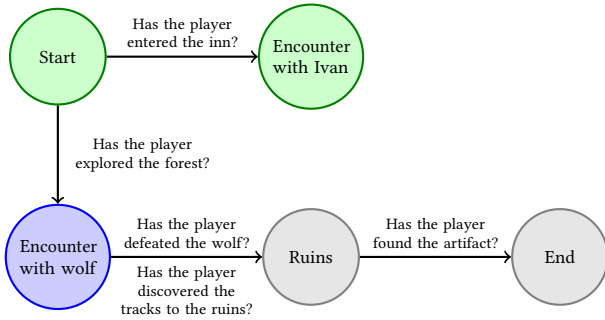


Figure 1: Example of a Narrative Graph. Green nodes have been completed, blue nodes are ongoing, and grey nodes are undiscovered.

4.2 SENNA - The Narrative Adherence System

ChatRPG v3 builds on the foundation established by the v2 system, retaining its web-based user interface while extending the underlying AI GM architecture. In this version, the AI GM is designed to follow a predefined narrative module provided by the user while preserving player agency. These modules can, for example, be pre-made D&D adventures². For readers interested in the v2 system, we recommend consulting our previous work [21].

The GM of ChatRPG v3 is implemented as a multi-agent system consisting of five specialized LLM agents: the **Scribe**, **Examiner**, **Navigator**, **Narrator**, and **Archivist**. This multi-agent system will henceforth be referred to as **SENNA**. Figure 2 provides a visual depiction of how the SENNA agents interact.

Each agent is built using the ReAct framework and is assigned a distinct role within the system, collaboratively emulating the functions of a human game master in interactive fiction games such as D&D. ReAct is a framework designed to empower LLMs with dynamic problem-solving abilities by interleaving internal reasoning with tool calls [48]. Appendix A provides example invocations of each agent. Detailed prompt specifications and tool descriptions for each agent are provided in Appendix F and Appendix G.

4.2.1 Responsibilities and Roles of the Agents. The Narrator and Archivist’s roles remain unchanged in ChatRPG v3, as they continue to handle core storytelling and memory management functions, providing the infrastructure for the new narrative adherence components.

Regarding narrative adherence in TTRPGs, a human GM is responsible for defining the overarching narrative structure of a campaign. This includes setting the main story beats, restricting player actions to maintain coherence with the intended storyline, and updating the state of the narrative as players make meaningful progress.

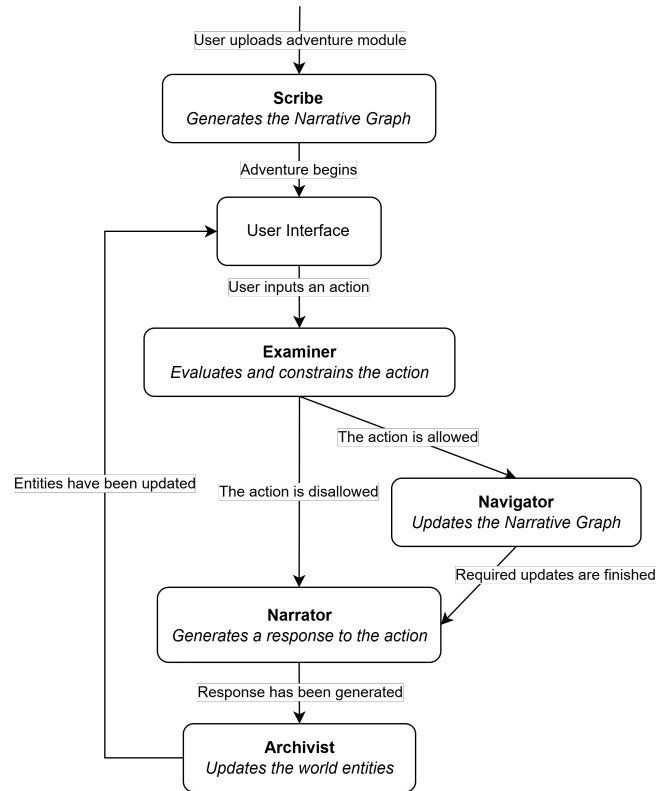


Figure 2: A diagram depicting how the agents in the SENNA framework interact.

The Scribe, Examiner, and Navigator agents replicate these responsibilities in ChatRPG v3. The **Scribe** is responsible for generating the initial narrative structure by translating a provided story module into a Narrative Graph. The **Examiner** evaluates player actions in real time, determining whether they are permissible within the constraints of the current narrative context. When an action aligns with the storyline and satisfies the necessary preconditions, the **Navigator** advances the narrative to the next relevant event, ensuring that story progression occurs in a structured yet responsive manner.

4.2.2 Scribe. The **Scribe** is responsible for initializing the Narrative Graph. It is invoked exactly once at the start of a campaign and is tasked with converting a pre-defined story module into a coherent and structured graph that encodes the possible narrative paths available to the player. It reads the contents of the uploaded module, interprets the embedded story beats, and incrementally translates them into nodes and edges within the Narrative Graph. To perform this task, the Scribe is equipped with three domain-specific tools:

- **AddNode:** Introduces a new narrative node to the graph, along with a set of incoming and outgoing edges.
- **AddEdge:** Establishes an edge between two existing nodes.
- **AddEndNode:** Connects an existing node to the designated End node, representing a conclusive narrative outcome.

²<https://www.dmsguild.com/product/186488/A-Most-Potent-Brew-A-Basic-Rules-Adventure>

Internally, the Scribe processes the module in batches to manage complexity and ensure consistent structuring. After executing a tool, it observes the updated state of the Narrative Graph and continues reasoning accordingly. After processing each batch, the Scribe produces a summary of the graph extensions performed, which is used as guiding context for the next batch.

The Scribe ensures that the constructed graph conforms to several narrative structuring principles: (1) branching should only occur if it introduces meaningful choices for the player, rather than enforcing a strictly linear narrative; (2) all edge conditions must be verifiable based on prior player actions; and (3) the graph must include at least one edge to the End node to guarantee a conclusive narrative arc. The final output of the Scribe is a completed Narrative Graph encompassing all relevant story beats and dependencies. Then, the downstream agents use this graph to enforce and track the progression of the structured story throughout the campaign.

4.2.3 Examiner. The **Examiner** is the first agent invoked in the system when processing player input. Its primary responsibility is to evaluate whether the proposed action is permissible within the constraints of the current narrative context.

To determine the validity of a player’s action, the Examiner integrates two sources of information: the Narrative Graph and the campaign’s summary. It then produces one of three possible verdicts:

- **Allowed:** The action aligns with the narrative structure and is feasible given the current game state.
- **Conditionally Allowed:** The action is permitted, but its success is uncertain or partially blocked due to unresolved conditions. This typically corresponds to failed attempts or ambiguous interactions.
- **Disallowed:** The action violates narrative constraints, skips crucial plot elements, introduces inconsistencies, or conflicts with the internal logic or setting of the world.

Each verdict is accompanied by a concise explanation grounded in the narrative graph, the current story state, and the structural logic of the adventure module. This output is subsequently consumed by the **Navigator** and **Narrator** agents to determine whether progression is warranted and how the resulting narrative should be presented to the player.

To support its reasoning process, the Examiner is equipped with a single domain-specific tool:

- **SearchScenario:** Performs retrieval-augmented generation (RAG) [15] by querying a vector-embedded representation of the adventure module. It is used when the agent requires additional context to evaluate whether a player’s action is reasonable based on their current position in the story. The tool retrieves semantically relevant sections from the module and returns structured information about locations, NPCs, quests, world rules, and possible interactions.

For example, if the player attempts to interact with an unmentioned character or bypass a locked objective, the Examiner will consult the SearchScenario tool before issuing a verdict. In all cases, it avoids introducing new content or contradicting known facts from the module or narrative graph.

The Examiner plays a critical role in maintaining narrative integrity. By rigorously filtering inappropriate or inconsistent inputs, it ensures that only plausible actions propagate through the system, thereby upholding the logic, pacing, and coherence of the authored story structure.

4.2.4 Navigator. The **Navigator** is responsible for maintaining and updating the progression state of the Narrative Graph in response to player actions. To achieve this, it integrates four inputs: the Narrative Graph, the campaign’s summary, the player input, and the Examiner’s verdict. Using this information, the Navigator evaluates whether transitions from currently active nodes (those marked as *Ongoing* or *Completed*) to undiscovered nodes should be triggered. If the agent is uncertain whether progression is warranted, it errs on the side of checking the conditions via tool invocation. It operates immediately after the Examiner agent and is only invoked if the Examiner issues a positive verdict, since otherwise, the player cannot progress.

To perform its function, the Navigator is equipped with a single domain-specific tool:

- **UpdateGraph:** Verifies whether the conditions on the edge between a specified source and target node are fulfilled. If all conditions are met, the source node is marked as *Completed* (if not already), the target node as *Ongoing*, and the edge as *Visited*.

When the Navigator determines that no further updates are required, it outputs a concise summary of the changes. This summary is subsequently consumed by the **Narrator** agent, which uses it to guide the generation of storytelling output that reflects the player’s current position in the overarching plot structure.

4.2.5 Narrator. The **Narrator** retains its role as the system’s primary storyteller. In ChatRPG v3, however, its capabilities have been significantly expanded to support structured story progression within pre-defined adventure modules through several key mechanisms:

- **Narrative Graph Integration:** The Narrator now has access to the Narrative Graph and uses it to ensure narrative consistency and to avoid prematurely revealing or accessing locked content. For example, if the player attempts to enter an inaccessible area, the Narrator denies progression through immersive in-world cues.
- **Verdict-Aware Storytelling:** The Narrator now consumes the verdict produced by the Examiner agent. The Narrator interprets the verdicts’ reasoning and constraints to justify the resulting narrative outcome.
- **Graph Update Summaries:** The Narrator also receives a summary of updates made to the Narrative Graph by the Navigator agent. The Narrator integrates these changes subtly into its storytelling, reflecting the progression without breaking immersion.
- **Scenario-Aware Reasoning via Tool Access:** The Narrator now has access to the SearchScenario tool, which allows it to retrieve and incorporate relevant setting details without revealing future or locked content.

Through these extensions, the Narrator transitions from a purely generative agent to a narrative reasoning component tightly coupled with the underlying story logic, and they are critical for maintaining coherence in structured storytelling scenarios while delivering a compelling and adaptive player experience. For a full description of the agent’s design and behavior, see our previous work [21].

4.2.6 Archivist. The **Archivist** agent serves as the memory module of the system, responsible for tracking and maintaining the evolving state of the game world. It operates in the background by analyzing the Narrator’s outputs to detect narrative changes, such as the introduction of new entities (e.g., characters or locations) or updates to existing ones.

In this version of the system, the core functionality of the Archivist remains unchanged from our previous study. The only modifications involve prompt optimizations aimed at improving reliability and consistency.

4.3 Redirection Strategies

To address RQ2, we designed our narrative coherence system around various redirection strategies, drawing from interactive storytelling, game design, and AI narrative practice. Each was integrated into the system during controlled scenario replays to compare player preferences and experiential impact directly.

Hard Denial of Player Actions (Baseline).

Directly refusing an action without explanation (e.g., “You cannot do that”) represents the simplest, and often the most immersion-breaking, strategy [25, 40, 47]. To mitigate this, our implementation grounds denials in the fiction (e.g., “The door does not budge”), but otherwise provides no justification. Hard denial serves as our baseline for comparison.

Offering Information to Players.

Contextual hints, such as warnings from NPCs or environmental cues (“That path hasn’t been safe since the orcs took it”), redirect players by clarifying available options without restricting choice [7, 27, 39]. We implement this strategy to test if enhanced narrative context and worldbuilding can naturally realign player action.

Illusion of Choice (Quantum Ogre).

This approach maintains perceived agency while structurally directing players to predetermined events. For example, no matter which corridor the player chooses, the same ogre awaits [24, 29, 31]. We adopted this device to assess acceptance of seamless yet scripted redirection.

NPC Influence.

NPCs who actively caution, challenge, or appeal to the player (“Are you sure this is the wisest course? The mayor trusted us to act quickly”) provide in-story nudges rooted in social interaction [29, 30, 39]. This strategy was implemented to evaluate whether character-driven interventions are perceived as more natural and immersive.

Humor.

Integrating playful comments, such as a wizard quipping when the player fails at magic, can mitigate the frustration of constraint [10].

We employ humor to explore its potential for softening the impact of narrative redirection.

In-World Consequences.

Allowing disruptive actions but ensuring logical fallout (e.g., robbing an NPC leads to being blacklisted) reinforces causality and narrative logic [12, 29, 43]. This strategy in our system upholds agency by showing the impacts of player choices rather than forbidding them outright.

4.4 Redirection Strategy Ranking by Expert Game Masters

To ensure alignment with best TTRPG practice, we presented all six strategies to five experienced GMs for ranking. Table 1 summarizes their preferences.

Strategy	GM1	GM2	GM3	GM4	GM5	Avg.
Hard Denial	6	5	4	5	5	5
More Information	1	2	1	1	1	1.2
Illusion of Choice	5	6	3	2	4	4
NPC Influence	3	3	1	3	2	2.4
Humor	4	4	5	6	6	5
Consequences	2	1	2	4	3	2.4

Table 1: Redirection strategy rankings by GMs (1 = most preferred, 6 = least preferred). Bolded strategies were selected for user testing.

Thematic analysis of GM feedback echo findings in the literature: *Offering Information*, *NPC Influence*, and *In-World Consequences* were consistently preferred for sustaining immersion and agency; *Hard Denial* and *Humor* were seen as disruptive except in rare cases; and opinions on *Illusion of Choice* were mixed, valued for narrative flow but sometimes diminishing real agency.

For user testing, we selected the three most GM-preferred strategies plus the baseline:

- (1) Offering Information to Players
- (2) NPC Influence
- (3) In-World Consequences
- (4) Baseline / Hard Denial of Player Actions

This selection allows us to empirically evaluate redirection approaches most likely to support narrative coherence, agency, and player satisfaction in both AI systems and traditional play.

5 Experiments

To evaluate ChatRPG v3 as an AI GM, we conducted a user study focused on narrative coherence, player autonomy, and adaptive storytelling. Our main goal was to assess how effectively the SENNA-powered system could deliver a compelling and coherent solo TTRPG experience while dynamically responding to player input. Unlike previous comparative studies, this experiment evaluated only ChatRPG v3’s capabilities, with a design prioritizing ecological validity: participants played naturally, without artificial constraints or moderator intervention.

The adventure module was a customized, single-player adaptation of *A Most Potent Brew* by Richard Jansen-Parkes. Players

explored the Wizard’s Tower Brewery, investigating a rat infestation and delving into interconnected chambers featuring combat, puzzles, and branching paths. Key locations included a magical rainbow corridor, a hazardous well room, a laboratory with a giant inferno spider, and a storeroom with a crucial potion. Players could also rescue Thistlewhip, a miniaturized wizard, and complete the quest by returning to Glowkindle. While the narrative was essentially linear, optional areas and lore provided opportunities for meaningful exploration and side content.

We recruited 12 participants, drawn from previous ChatRPG contributors and new users, all with TTRPG experience and strong English proficiency (see Table 3 in Appendix B). The sample represented a mix of ages and experience with AI-driven text-based games.

5.1 Study Goals and Considerations

The primary objectives of this study were:

- Assess the AI GM’s coherence in following the narrative while enabling player choice and improvisation;
- Investigate players’ perceptions of the AI’s attentiveness, emotional responsiveness, and human-likeness;
- Explore preferences for different narrative redirection strategies in scenes where the AI adapts to unexpected or off-track player actions. We seek to understand which strategies users find most seamless, satisfying, and respectful of their agency;
- Evaluate overall player experience, including usability, immersion, and enjoyment.

Care was taken to minimize moderator influence, ensure uninterrupted sessions, and combine subjective feedback with structured observations.

5.2 Procedure Summary

Each participant completed a single 2-hour session comprising four phases:

- (1) **Pre-session survey:** Demographics and background in RPGs.
- (2) **Session 1 – Full playthrough:** An uninterrupted 75-minute story playthrough, with three campaign snapshots for later analysis. The AI GM ran the module in a structured format and adapted in real-time to player decisions and deviations.
- (3) **Session 2 – Redirection strategy preference:** After the full playthrough, participants revisited three key scenes, each presented in four versions—one for each narrative redirection strategy—testing which approach felt most seamless and respectful of agency.
- (4) **Post-session survey and interviews:** Participants completed standardized questionnaires (PXI, ASAQ, and custom items) and two semi-structured interviews, one on their overall impressions and one comparing the redirection strategies and their effect on perceived agency and immersion.

Detailed protocols, scripts, and instruments are provided in Appendix E.

6 Results

In this section, we present the key findings from our evaluation of ChatRPG’s latest iteration. We first report on user experience through both quantitative survey measures and qualitative interviews, offering insights into immersion, agency, and narrative coherence as perceived by players. We then examine the system’s ability to maintain narrative adherence to the intended adventure module, followed by an in-depth analysis of how different redirection strategies impact player satisfaction and story flow. Together, these results shed light on the strengths and challenges of multi-agent LLM-based game mastering and highlight areas for further refinement in balancing narrative structure with interactive freedom.

6.1 User Experience Evaluation

We begin by analyzing participants’ survey responses and interviews to evaluate how ChatRPG’s latest version balances user engagement, agency, and immersion, with particular attention to the system’s ability to adhere to a pre-written narrative while preserving player freedom.

6.1.1 Quantitative Insights from Participant Surveys. To evaluate user engagement and immersion, we administered the same questionnaire used in our prior study [21], which combines selected constructs from the Player Experience Inventory (PXI) [33] with custom ChatRPG-specific items.

We also incorporated selected constructs from the Artificial Social Agent Questionnaire (ASAQ) [13], a validated tool for evaluating user interactions with artificial social agents. Following the recommendations of both PXI and ASAQ, we employed a 7-point Likert scale ranging from -3 to 3. Full survey details and question items are provided in Appendix E.

The mean user ratings for each construct are presented in Appendix C, alongside corresponding values from our prior study for ChatRPG v2, which serves as our current state-of-the-art baseline. As the studies involved different participant groups, statistical tests such as t-tests are not applicable. Nonetheless, the mean ratings can be used to infer ChatRPG v3’s user experience.

The results reveal several notable differences between ChatRPG v2 and v3. Most prominently, the Autonomy, Ease of Control, Mastery, and Story Adapted constructs received lower ratings in v3. This was expected due to the shift in narrative structure: While v2 allowed players to freely define their adventures, v3 enforces a predefined narrative, which may lead players to perceive fewer meaningful choices, reduced agency, and a lack of responsiveness in the story, particularly when the AI GM denies actions that would derail the intended plot.

Similarly, the Immersion construct is rated lower in v3. This may be attributed to the same narrative constraints, where limitations on player actions can feel artificial or immersion-breaking. In some cases, the AI GM in v3 prematurely reveals narrative elements, such as naming key NPCs before they are introduced in the story, which may further reduce players’ sense of presence and discovery. These factors may also explain the decline in the Engaging NPCs construct.

Despite these reductions, most construct scores remain relatively close between versions. This suggests that ChatRPG v3’s overall

user experience is comparable to v2 and that the additional responsibilities introduced in v3 do not significantly degrade player satisfaction.

Turning to the ASAQ measures, the Human-Like Behavior construct received the lowest score among these constructs. This reflects that players do not perceive the AI GM as fully replicating the behavior of a human GM, likely due to its rigid enforcement of narrative and occasional unnatural dialogue patterns.

However, the Agent’s Enjoyability, Attentiveness, and Coherence constructs were rated highly, indicating that the GM was still perceived as a competent facilitator of gameplay. These findings suggest that players are aware they are interacting with an AI, but still find the experience engaging and effective.

6.1.2 Qualitative Insights from Play Sessions. In this section, we present a thematic analysis [17] of participant interviews, examining how players experienced narrative coherence, agency, and immersion while interacting with the SENNA-powered system without explicit GM redirection. By foregrounding participants’ own words and experiences, we uncover recurring patterns and challenges in how the system enabled interactive storytelling. This qualitative lens complements our quantitative results, surfacing nuanced strengths and limitations in the system’s mediation of narrative flow, creativity, and the tension between structure and freedom. Insights are organized by key themes from players’ full play sessions.

Narrative Adherence. Participants consistently described the AI GM’s story as coherent, believable, and strongly reminiscent of classic tabletop adventures. Terms like “a proper adventure,” “a normal D&D side-quest,” and “a genuine story you could follow” appeared frequently. Players noted the narrative’s logical, linear structure and natural progression of locations and events (P3: “everything was described fairly consistently... it made a fair amount of sense”; P4: “it felt like a proper adventure”; P8: “feels like a normal D&D side quest”).

A key strength was SENNA’s ability to maintain narrative logic while responding flexibly to creative or unexpected actions. Even when players deviated or “tried all kinds of shit” (P2), the system would usually “accept the initiative” and “adapt pretty well” (P4). Participants appreciated having their creative ideas acknowledged and woven into the ongoing story (P1, P6, P11).

When the AI needed to redirect or limit actions for narrative reasons, participants generally found this fair and in line with expectations of a human GM (P5: “it felt pretty reasonable... the same way a [real] DM would do”, P3: “even the way it redirected felt kind of natural”). Railroading was accepted as part of playing a pre-written module, particularly when boundaries were justified in-world.

Issues with narrative logic or pacing were rare and minor, such as a character “somehow trapped himself in a glass jar” (P2) or an NPC appearing unexpectedly (P11), and most players found these easy to overlook within genre conventions. Only a few remarked on artificially fast pacing (P12) or overly explicit hints that reduced mystery (P6, P9). Overall, participants judged the story arc as robust and engaging.

Player Agency and Freedom. Most participants felt a strong sense of agency and freedom (“free to do whatever [they] wanted... within reason” (P5)). The system was seen to respond flexibly, even when actions diverged from the prepared story: “It allowed me to do most things that I tried, respecting my limitations” (P7); “I had the options to do what I wanted” (P5).

Players appreciated that their creative or unconventional choices were often supported and woven into the ongoing narrative. One participant explained, “It adapted pretty well... even when I was trying to find out stuff it probably didn’t account for, like with the potions... But it did let me do it when I said, ‘caution to the wind, drink it’” (P4). Another highlighted satisfaction that, “[the AI] accommodated them almost maybe too well... doing a great job kind of working with my exploring and trying to figure out what was going on” (P11).

However, participants understood that their freedom existed within the logic of a pre-written module, and moments of “railroading” were expected (“we play in this story... I didn’t feel forced [as] a one-shot adventure” (P2); “You’ve chosen to play this scenario... that’s exactly how it would be for D&D. If you do something really unexpected, your DM probably won’t let you...” (P6)). When railroading occurred, such as “I felt really forced to kill the spider... I wanted to see if there was an option not to do so. But it wasn’t there” (P5), it was usually accepted as necessary to keep the story moving, especially in short games. As one put it: “sometimes you have to be forced for the story to keep flowing, right? ... You’ve chosen to play a story so you’re okay with it” (P9).

System pushback was described as reasonable and expected, especially when justified by character/narrative logic. Many noted that mindset affected perceived agency: “I felt free to try, but once it told me no, I started thinking about it in a more traditional way, of things I’d expect it to allow” (P1). Opting into the narrative structure was seen as key to enjoyment.

Immersion. Participants reported a strong sense of immersion, often calling the experience “believable,” “genuine,” and “like a proper adventure” (P4, P8). They felt engaged in the world and described the story as consistent (“It made a fair amount of sense” (P3); “I was still in it... I was thinking, what do I have to do now? What’s my next move?” (P5)). The system’s in-world logic and boundaries helped sustain presence: “I could do most things I tried, but I can only do low-level magic. That’s fair enough... That’s an explanation for why I wasn’t allowed to summon a fire giant” (P1).

Immersion was highest when responses referenced character motivation or backstory (“It reacts well, even when you do things that are a bit extra challenging... here it also rejects me very well, I think... not like it rejected me straight away” (P7)), and when creative choices were incorporated (“I did like the fact that it allowed me to just use the chemicals... that worked out, so that’s nice” (P1)).

Minor breaks in immersion stemmed from mechanical or narrative limitations, such as logic inconsistencies (“I had a hard time with [the elf in the jar]” (P2)), repetitive loops (“it reacts to what I say, which is not quite what it wants, by repeating what it said” (P6)), or solutions given too abruptly (“it just gave me the correct sequence immediately without me asking” (P9)). Explicit hints or meta-language (“helicopter-parent hand-holding” (P6)) also occasionally broke immersion.

Nevertheless, most players overlooked such moments and remained absorbed in the adventure (“I can forgive such small mistakes.” (P9)). Immersion depended on the AI’s narrative subtlety and consistency; minor slips did not detract meaningfully.

Balance Between Pre-Written Narrative and Player Actions.

A central challenge was managing player unpredictability with pre-written story structure. Participants highlighted that the system “adapted pretty well” if choices “made sense in context” (P11). Outcomes were most satisfying when the GM responded creatively, building on player contribution (“accept[ed] the initiative we’re creating... And I think that’s really cool” (P2); “I had gotten Little John (player-introduced NPC). I didn’t make up anything for Little John... Then it comes in and says, this guy is a sprite and it has intelligence, I would like to help you...” (P2); “It tried to deviate by generating a secret passage... that’s more because of our inquisitive nature” (P10)).

Occasionally, the system could not fully reconcile agency with structure, leading to neutral or repetitive responses (e.g., expecting a magical effect from a potion but only receiving “glittering magic mist... makes the spider mad” (P3); or “you investigate it more closely, and then it says... yes, you investigate it more closely” (P6)). Some noticed increasing narrative control as the game progressed.

Despite these occasional issues, the vast majority of interactions integrated player actions smoothly and naturally. Most participants felt the system succeeded in balancing agency and narrative coherence, with minor exceptions causing only brief perceptions of arbitrariness or constraint.

6.2 Narrative Adherence Analysis

To evaluate narrative adherence in the user study, we systematically tracked each participant’s playthrough alongside the pre-written adventure module. For each campaign, we defined a set of narrative adherence statements, categorized as either *required* (essential for following the module’s intended story) or *optional* (additional branches, lore, or side content that enrich but are not necessary for completion). The full adherence analysis for our test scenario is provided in Appendix D.

Our results show that the AI GM consistently achieved high narrative adherence: nearly all required statements were satisfied across playthroughs, indicating robust alignment between AI-driven storytelling and the original narrative structure. This outcome is attributable to the collaborative functioning of the Examiner and Narrator agents, both operating with live access to the Narrative Graph. The Examiner assesses the validity of each player action in real time, referencing the graph and scenario context, while the Narrator advances the plot only when relevant preconditions are met. This mechanism ensures key milestones are unlocked and presented strictly upon satisfying prerequisite story events, thereby filtering out actions that might undermine narrative progression.

It is noteworthy that the system did need to perform redirection occasionally during play, particularly when players were especially creative or attempted to use unusually powerful magic or abilities. In such cases, the AI selected and blended redirection strategies on its own, based on what it determined was most appropriate at the moment. Generally, this ad hoc choice was sufficient to keep the

narrative on track, but the suitability of the selected strategy varied. Since the system was not instructed to prefer any specific type of redirection for a given situation, mismatches sometimes occurred, highlighting the importance of matching redirection strategies to particular contexts.

While the AI GM was highly reliable with core (required) narrative elements, incorporation of optional content was more variable. Approximately half of campaigns engaged with optional branches or non-essential lore, which we see not as a flaw, but as an expected outcome for interactive storytelling: optional paths may not always be explored, depending on player choices and contextual cues.

Currently, the system does not enforce a prescribed “redirection” strategy to promote adherence; instead, observed alignment arises organically from how the Examiner restricts what can plausibly occur (per the Narrative Graph) and how the Narrator frames plausible next scenes. The AI is free to employ any narrative device within its pretrained distribution and is not biased towards surfacing optional material. This ensures robust coverage of essential plot beats, but optional content remains inconsistently presented—a limitation for deeper exploration and replayability.

Our interviews included a direct comparison between the initial play session, where the system freely chose arbitrary or blended redirection strategies, and the tailored redirection evaluation session. When asked, “How did the snapshot session differ from your initial play?”, participants unanimously felt that having explicit, scenario-matched redirection strategies led to responses that felt more natural and appropriate. This feedback underscores the value of dynamically selecting redirection strategies based on context, rather than relying solely on the AI’s unsupervised judgment. Future work should focus on systematically adapting redirection strategies to the specific narrative context, which appears critical for maximizing both engagement and believability in interactive storytelling.

6.3 Redirection Strategy Evaluation

To evaluate the impact of different redirection strategies on narrative coherence and player experience, participants were exposed to three distinct narrative scenarios, referred to as snapshots. These snapshots represented key interaction contexts in a role-playing game: **Puzzle**, where the player attempted to bypass a barrier without solving it; **Enemy**, where they tried to neutralize a threat in an unintended manner; and **Ally**, where they acted inappropriately towards a friendly NPC. For each snapshot, players experienced four redirection strategies: Hard Denial, More Information, NPC Influence, and Consequences, selected in Section 4.3. After each scenario, participants indicated both their preferred redirection strategy and how appropriate they found each response, with appropriateness defined in terms of tone alignment, immersion continuity, and narrative flow, rated on a -3 to 3 Likert scale.

Once all three snapshots were completed, participants took part in follow-up interviews. These open-ended follow-ups allowed us to gather deeper insights into their attitudes towards the redirection strategies and how these approaches influenced their overall experience.

6.3.1 Quantitative Analysis on Redirection Strategies. Figure 3 shows the distribution of participant preferences across the three snapshots. Across all scenarios, Consequences consistently emerged as the most favored redirection strategy. In the Puzzle snapshot, More Information was preferred over NPC Influence, while the reverse trend was observed in the Enemy scenario. This pattern suggests a possible relationship between snapshot type and players’ preference for different strategies.

To formally test whether there is a statistically significant association between snapshot type and preferred redirection strategy, we conducted an extended Fisher’s exact test using Monte Carlo simulations (10,000 replicates), as the contingency table was 3×4 . The resulting p-value was approximately 0.68, indicating no statistically significant association between snapshot type and redirection strategy preference.

It is important to note some limitations of this method. Fisher’s test assumes fixed row and column sums, yet in our study, only row totals (per snapshot) were fixed by design. This may have introduced some bias, as the column sums (strategy preferences) were not constrained. Moreover, Fisher’s test does not account for the repeated measures structure of our data, where each participant evaluated multiple snapshots and strategies, possibly affecting statistical power and interpretation.

In summary, while no strong statistical linkage between scenario type and redirection preference was found, the general pattern of the data suggests a clear trend: participants tended to prefer the Consequences strategy for narrative redirection, with no participants aligning with the baseline Hard Deny strategy.

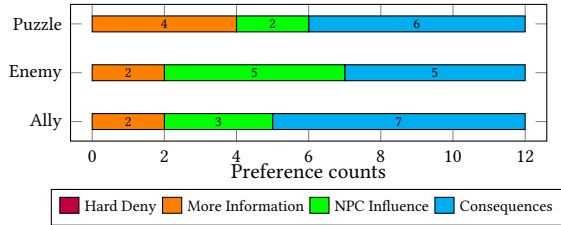


Figure 3: Participant preferences of redirection strategy for each snapshot.

Table 2 presents mean appropriateness ratings across all snapshot–strategy combinations. As with preference, In-World Consequences received the highest overall rating ($M = 2.19$), followed closely by NPC Influence ($M = 1.94$). Hard Denial was consistently rated the least appropriate ($M = -2.03$). The snapshot context also played a role: NPC Influence was rated especially highly in the Enemy and Ally snapshots, where NPCs were relevant and present in the scene.

To evaluate the statistical significance of these differences, we conducted an Aligned Rank Transform (ART) ANOVA with snapshot and redirection strategy as fixed effects and participant as a random effect. The results showed a significant main effect of redirection strategy ($p < 0.001$), but no significant main effect of snapshot type ($p < 0.38$), nor an interaction between snapshot and strategy ($p < 0.23$).

These results indicate that while the type of scenario did not significantly impact appropriateness ratings, the choice of redirection strategy itself did.

To further explore the strategy differences, we conducted post-hoc pairwise comparisons with Bonferroni correction. The results revealed:

- Significant differences between Hard Deny and all other strategies ($p < 0.001$).
- A significant difference between NPC Influence and Consequences ($p < 0.05$).
- No significant differences between More Information and either NPC Influence or Consequences.

Interestingly, despite More Information having a slightly lower average appropriateness than NPC Influence (1.67 vs. 1.94), the post-hoc analysis found no significant difference between More Information and Consequences. In contrast, the difference between NPC Influence and Consequences was significant. This apparent discrepancy is likely due to the ART model’s rank-based analysis, which accounts for the distribution of individual responses and repeated measures more deeply than raw means.

In conclusion, these results suggest that players generally perceive Consequences as the most appropriate redirection method. However, NPC Influence is also highly rated, especially in contexts involving social dynamics, while the baseline strategy Hard Deny is consistently seen as inappropriate, breaking immersion and narrative continuity.

	Appropriateness			
	Hard Deny	More Info.	NPC Influence	Consequences
Puzzle	-1.83	1.66	1.25	2.25
Enemy	-1.75	1.92	2.33	2.25
Ally	-2.5	1.42	2.25	2.08
Avg.	-2.03	1.67	1.94	2.19

Table 2: Participant appropriateness ratings of redirection strategy for each snapshot.

6.3.2 Qualitative User Perceptions of Redirection Strategies. Redirection in narrative-driven, interactive TTRPGs orchestrated by LLM-based game masters is a subtle craft: the system must maintain story integrity without undermining player agency or immersion. To better understand player attitudes toward the four redirection strategies, we conducted a reflexive thematic analysis[5, 17] of participant interviews. In this section, we present illustrative examples of participants’ reactions to each strategy and then highlight several cross-cutting themes that emerged across responses.

1. Baseline/Hard Denial (Strategy A): “You Can’t Do That”

Theme: Frustration, Agency Loss, and Immersion Breaks

Participants unanimously disliked hard denial (A), finding it immersion-breaking, demotivating, and often antagonistic to their sense of agency and creative investment.

- **Excluded from best-choice votes entirely.**
- Typical reactions include: “A was shit. There was no description or anything. ... It blatantly ignored what I wanted to

do,” (P2) and “*It just says, you can’t. I hate that ... A is too short-sighted. I’d expect a follow up, or expect something more*” (P12).

- Several likened it to negative childhood experiences in classic adventure or MUD games, distinguishing between tolerances for pure parser denials (“No, you can’t do that”) versus more modern expectations for narrative justification.
- “*If you get this many times in a row, it gets frustrating. You try to find a solution, and you don’t feel like you’re getting anywhere.*” (P12).
- Participants noted this approach increased their stubbornness (“it makes me want to do the bad thing more”–P11), or resulted in “sulk” (P9).
- **Comparisons with human GMs:** Participants overwhelmingly stated this style would prompt arguments with human GMs, feeling it lacked explanation: “It gives an answer without explanation or other options ... it gives a bratty child knee-jerk response of like, okay, but why?” (P11)

2. Offering Information (Strategy B): “You Consider, But...”

Theme: Gentle, Guidance, Player Reflection, Minimally Intrusive Correction

Offering more information (B) emerged as a widely appreciated redirection strategy, particularly in less emotionally charged or exploratory contexts.

- **Popular in the puzzle scenario (4 votes); respected elsewhere.**
- Participants described it as “*the most natural,*” (P4, P8) citing its ability to “*guide you without being heavy-handed*” (P6).
- Participants appreciated when it engaged their character’s internal reasoning: “*It aligns you with the character again. Like, you may think this is great, but your character has a moment of clarity that this might not be a good idea.*” (P10)
- However, some saw B as too passive in some cases, or edged towards repetition or restating context without enough progression: “*It restates the situation again ... that’s not so cool.*” (P12) and “*If you outright tell someone that’s a bad thing or you can’t do that, it just kind of, you know, gives like this feeling of defeat.*” (P4)
- **Comparison to human GMs:** B is often equated to “reminders” or “nudges” that most human GMs might employ, and considered inoffensive but occasionally “auto-reply”—a little bland if overused.

3. NPC Influence (Strategy C): “NPC Intervenes”

Theme: Social Immersion, Emotional Leverage, Narrative Integration

NPC-based interventions (C) were well-received, particularly in social or interpersonal situations. When executed as a plea or warning from an NPC, this strategy leverages emotional ties and party dynamics:

- **Enjoyed especially in the Spider (5 votes) and Thistlewhip (3 votes) encounters.**

- “*Talking to an NPC who is just with you is a little better than contemplating things yourself ... It feels a little more integrated.*” (P1)
- Many noted this felt authentic: “*I like that C kind of gives you the Thistlewhip chiming in with guidance—redirection without being as aggressive as A.*” (P11)
- NPC interventions were seen as effective in redirecting without directly denying the player’s autonomy: “*It’s a really cool slap on the wrist, but ... more integrated, fits the mood.*” (P12)
- However, some found the “inner voice” style (rather than an overt NPC) “a bit too meta” or unnatural (P6), with social nuance essential.
- **Comparison to human GMs:** Parallels drawn to a party member or GM-voiced companion interjecting in tabletop play—generally acceptable and sometimes preferable; can ease hard transitions when stakes are social or emotional.

4. In-World Consequences (Strategy D): “Vision or Vivid Consequence”

Theme: Satisfying Curiosity, Maintaining Agency, Encouraging Reflection through Consequentialism

D was the overall favorite, particularly in combat (5 votes) and social (7 votes) vignettes. **A clear pattern emerges in participants’ comments:** this strategy maximized perceived agency, often by letting them see the outcome of their forbidden action without actually derailing the story.

- **Most frequently chosen as “favorite” across test scenarios.**
- “*It gives the people the satisfaction of knowing what might have happened ... letting their imagination run, usually that would be satisfying to them and be like, okay, okay, let’s be serious now.*” (P9)
- Provides “*ample warning*” (P3), “*lets me do the funny stuff in my vision so I still feel agency ... and then it still denies it, but I’ve had the experience*” (P1).
- Seen as “very RPG-like” and reminiscent of “vision” mechanics in games (P7), or a “what-if” scenario that ultimately respects narrative constraints while quenching the player’s curiosity.
- P12 highlighted its utility: “*D is clearly the best, because I feel like it plays with my idea and builds on it.*” (P12) But participants cautioned that overuse could be formulaic or melodramatic, recommending scaling the drama to the gravity of the action.
- Multiple described it as especially effective in high-stakes or outrageous deviations, but potentially excessive for minor infractions.
- **Comparison to Human GMs:** Many participants suggested D-style “consequence previews” were both immersive and satisfying, and could feel even more “fair” or dramatic than a typical human GM’s quick “no.”

Cross-Cutting Themes:

- **Desire for Reason, Not Just Refusal:** Across all strategies, participants overwhelmingly wanted explanations, hints, or narrative logic—not just a “no.” Even when denied, most preferred to understand *why*.
- **Blend and Adaptation:** Many felt the best human GMs would blend all of B, C, and D depending on the nature of the deviation, stakes, and character. There is no single best strategy for all contexts.
- **Scaling Drama to Deviation:** Participants repeatedly mentioned that the form of redirection should be proportionate. Minor boundary-pushing might warrant B; wild or “evil” actions might get a full D.
- **Impact on Immersion and Enjoyment:** Only hard denials consistently broke immersion and reduced enjoyment. B, C, or D-style redirections could in fact increase immersion by deepening narrative plausibility, player reflection, and emotional engagement.
- **Consistency with Player Character:** Some participants wanted redirection to be mindful of their character’s personality and backstory, particularly for C and D.

In summary, our thematic analysis reveals distinct participant preferences and nuanced perceptions regarding each redirection strategy. These findings underscore the complex interplay between narrative coherence, agency, and player immersion in multi-agent LLM-driven interactive games. In the following section, we examine the broader implications of these results for coherent narrative system design and present specific recommendations and design considerations informed by the user feedback on redirection strategies.

7 Discussion

This section synthesizes our main findings and positions them within the broader context of narrative AI research and role-playing game design. We evaluate how the integration of narrative graphs, modular ReAct-based agents, and evidence-based redirection strategies advance the design of LLM-driven TTRPG experiences. We discuss both the successes and critical limitations of our approach with respect to narrative adherence and player agency (RQ1 and RQ2), highlight implications and directions for future systems, and address technical considerations and study limitations.

7.1 Narrative Adherence with Narrative Graphs and ReAct

This work set out to answer RQ1: *How can an agentic, LLM-based system be designed to support single-player role-playing games that are both based on and constrained to user-provided stories?* Our approach combines a narrative graph representation for definable story structure with modular ReAct-driven agents, enabling robust narrative tracking and flexible, agentic behavior by the AI GM.

7.1.1 Strengths of Structured Narrative Graphs. The narrative graph, automatically derived from adventure modules by the Scribe agent, allowed for explicit encoding of story milestones and dependencies. This explicit structure, in line with recent advances such as GENEVA

and mixed-initiative authoring pipelines [1, 26], ensured consistently high narrative adherence: nearly all required story events occurred in all playthroughs, and player experience remained logical and coherent throughout. Automated tracking of player progress and divergence points further enabled diagnostic insight into narrative drift.

7.1.2 Balancing Structure and Agency. A key insight from our evaluation is that narrative control alone is insufficient; participants reported high satisfaction only when agency was preserved and creative player actions were acknowledged within the unfolding story. The SENNA framework’s natural redirective strategies, especially when rooted in world logic or in-character cues, enabled scaffolding of player agency and minimized the perception of railroading. Consistent with findings in systems like SHARI and Calypso [28, 51], players valued GMs that blended adherence to narrative structure with flexible, context-sensitive responses. Nevertheless, moments when creative solutions were ignored, or when redirections felt abrupt or artificial, led to noticeable breaks in immersion.

7.1.3 Limitations and Opportunities in Branching Narratives. Despite strengths in core story adherence, handling of branching and optional content remains a challenge. The system disproportionately favored “heavier” branches (more content-rich), limiting exploration and replayability. This selection bias, arising both from agent prompt design and branch search heuristics, often prevented players from experiencing optional content, even when different paths were explicitly available. Addressing this will require more nuanced prompt engineering and agent logic of both the Examiner and Narrator, as well as proactive signaling of meaningful player choices to the user.

7.1.4 Examining the Narrator’s Response. Currently, only player actions are evaluated for narrative adherence via the Examiner. While the Narrator’s responses could also, in theory, violate narrative constraints, this is unlikely in practice. The Narrator has access to the Narrative Graph, the scenario module, and the Examiner’s verdicts, which collectively constrain its outputs. One could apply the Examiner to assess the Narrator’s responses, but since both agents operate with the same context, such evaluations would likely echo the Narrator’s own reasoning without adding meaningful verification. Thus, examining the Narrator’s output offers limited practical value under the current architecture.

7.1.5 Reflections on the ReAct Framework. Our use of ReAct provided clear abstractions for dividing agent responsibilities and integrating new reasoning tools. While ReAct’s modularity aided both system robustness and extensibility, our findings, mirroring recent critiques [46], show its effectiveness is tied to high-quality exemplars and prompt design, rather than inherent action-reasoning synergy. Nonetheless, ReAct remains an effective organizational framework for multi-agent interactive systems when complemented by tailored prompts and agent-specific few-shot learning.

7.1.6 Future Directions. Building on our findings and the practices of state-of-the-art systems such as SCORE [50], Calypso [51], and large-scale LLM agent environments [32], the next steps for narrative adherence in agentic AI GMs should focus on:

- (1) Enhanced handling and player-facing presentation of optional branches for true replayability;
- (2) Subtler, character-driven cues and scaffolds that maintain discovery and agency;
- (3) Flexible integration of creative solutions, e.g., “yes, and...” narrative development;
- (4) Ongoing improvements to agent prompt design, dynamic state tracking, and summarization for greater responsiveness and immersion.

Our results show that coupling explicit narrative graphs with modular, ReAct-powered agent architectures enables both faithful story progression and creative player interaction in LLM-driven TTRPGs. Striking the right balance between structural narrative control and adaptive, player-sensitive agency remains the central design challenge for future systems.

7.2 Redirection Strategies: Evidence-Based Design Implications

Addressing RQ2—*When the AI GM must redirect or constrain player actions that are unreasonable or outside the scope of the story, how can these redirections be designed to uphold an enjoyable and immersive game experience?*—our study examined player responses to four distinct redirection strategies: Baseline/Hard Denial, Offering Information, NPC Influence, and In-World Consequence, each tested across exploration (puzzle), combat (spider), and social (Thistlewhip) contexts. Evaluation combined participants’ stated preferences and Likert-scale ratings of appropriateness (see Figure 3 and Table 2).

Hard denial (A) was the least favored, both in terms of preference and appropriateness, repeatedly described as frustrating and immersion-breaking. By contrast, in-world consequences (D) and NPC influence (C) received the highest ratings, especially in emotionally charged or narrative-critical scenarios. Offering Information (B) stood out as a preferred nudge for exploratory moments but lost appeal when emotional engagement grew.

Recommendations & Design Implications:

- (1) **Avoid Hard Denial (A) as a Primary Strategy**
Consistently low ratings for hard denial, combined with clear participant frustration (“A was shit... it blatantly ignored what I wanted to do”; “It makes me want to do the bad thing more”), underscore that abrupt refusals can entirely undermine both agency and engagement. This finding echoes broader LLM denial strategy research [47].
Recommendation: Restrict hard denials to rare, rule-bound cases, and whenever possible, embed minimal narrative justification to soften the rejection.
- (2) **Leverage In-World Consequence (D) for Major or Risky Deviations**
Participants most appreciated consequence-based redirection, especially in the most emotionally charged (Thistlewhip) and exploratory (puzzle) settings, finding it immersive, respectful of intent, and akin to the best tabletop principles (“I feel like it plays with my idea and builds on it”).

When players see logical, in-world fallout from their actions, it honors agency while preserving story logic. Bridging “what-if” scenarios directly improves satisfaction [36].

Recommendation: Employ consequences when players attempt strongly off-track, dangerous, or campaign-breaking actions, scaling drama to fit the gravity of the deviation.

- (3) **Contextualize with NPC Influence (C) in Social or Emotional Scenarios**

NPC guidance was effective in both combat and social encounters, described as “authentic,” “emotionally grounded,” and synergistic with classic tabletop play: social nudges carried weight and helped maintain group cohesion. Players responded best to guidance from companions or established allies [6, 9, 37].

Recommendation: Leverage emotionally resonant NPCs to steer players in relational or high-stakes contexts, favoring naturalistic over mechanical interventions.

- (4) **Utilize Offering Information (B) for Gentle Exploratory Nudges**

Providing subtle context through hints was best received in exploratory and puzzle settings (“it guides without being heavy-handed”), supporting curiosity without overtly restricting agency. However, less effective in moments of heightened narrative or emotional importance. This parallels findings from adaptive interactive storytelling [44].

Recommendation: Apply targeted informational nudges for ambiguous or exploratory player actions, but avoid generic overuse to prevent them from feeling formulaic.

- (5) **Blend and Personalize Redirection Strategies**

Neither the data nor player perspectives support a one-size-fits-all strategy; instead, the highest satisfaction comes from adaptive, context-aware blending. Players valued their choices being acknowledged, whether through consequence previews, NPC responses, or timely context. This aligns with research advocating emotionally adaptive narrative engines [3].

Recommendation: Develop adaptive logic that selects or combines redirection strategies in real-time based on player intent, character background, narrative context, and scenario stakes.

- (6) **Narrative Framing and Immersion**

Redirection is most effective when deeply embedded in the in-game world and character point of view. Players consistently disliked meta-communication about strategy (“avoid overt signaling that a ‘strategy’ is being deployed”). Grounding responses in role and context minimizes perceived railroading and preserves immersion [8, 25, 40].

In summary, our findings demonstrate that successful redirection in multi-agent LLM-driven RPGs must be contextually sensitive, emotionally resonant, and tightly integrated into the narrative fabric. Strategies that blend consequence, character-driven social cues, and gentle in-world hints most effectively sustain agency and immersion, while blunt denials or out-of-character framing pose a consistent risk. These lessons support ongoing advances in AI narrative design and offer concrete direction for both system

builders and game masters seeking to preserve story coherence without sacrificing player enjoyment.

7.3 Study Limitations

We acknowledge several limitations of this study, including the sample size, study design, and constrained experimental scope.

While the experiment involved a small number of participants ($N=12$) and a short RPG experience of around 75 minutes, it was sufficient to surface meaningful trends in player interaction and early responses to AI-driven narrative redirection. Future studies with larger and more diverse samples are needed to validate and generalize these findings.

Participants played ChatRPG using their own computer in their own home, thereby ensuring a naturalistic environment. However, participants engaged in live voice calls with a test moderator. Although care was taken to minimize interference, this setup may have influenced player immersion or behavior, as they may have felt watched and time pressured. Conducting longer, unsupervised sessions, where participants play independently in their own time, could yield ecologically richer data.

To ensure consistency across participants, the study used a fixed, short adventure module and pre-generated characters. While this enabled controlled analysis, several players noted that customizing their character might have deepened engagement. Future work should explore player-authored characters and varied modules to evaluate how customization impacts narrative adherence and immersion.

Finally, the “In-World Consequences” strategy diverged slightly from traditional TTRPG practice by explicitly revealing future outcomes to the player. While this vision-based feedback was well-received, it may not reflect how such consequences organically unfold in typical tabletop play. Future designs should explore more subtle or emergent implementations of consequence-driven redirection.

7.4 Comparing LangChain and the Model Context Protocol

ChatRPG v3 was developed using LangChain, an open-source framework for integrating LLMs with tools, APIs, and user interfaces [45]. Its modular components and broad library of pre-built integrations made it well-suited for rapid development and orchestration of complex workflows.

The Model Context Protocol (MCP), by contrast, provides a standardized, client-server interface for connecting LLMs to tools and data [18]. Its emphasis on clear, modular communication has made it increasingly popular for building scalable and interoperable systems.

Recent adoption by major platforms such as Microsoft and GitHub [20, 41], along with research highlighting enhanced interoperability and scalability in agentic AI systems [23, 35], suggests a broader shift toward MCP as a foundational design pattern.

While LangChain has served our purposes well, the structure and interoperability offered by MCP present a compelling path for future development, particularly as AI systems grow in complexity and modularity.

8 Conclusion

This work tackled the dual challenges of narrative adherence and player agency in LLM-driven single-player role-playing games. Through the development and evaluation of the ChatRPG v3 system—powered by the SENNA multi-agent architecture and structured narrative graphs—we demonstrated how an agentic LLM-based AI game master can faithfully adapt pre-written adventure modules while providing a robust, interactive storytelling experience.

Our findings show that the narrative graph is an effective mechanism for tracking and enforcing story progression, successfully ensuring that core elements of the user-provided scenario are respected throughout play. At the same time, our analysis revealed that surface-level success in narrative adherence must be balanced against subtler aspects of agency and immersion: the system’s handling of optional branches and redirection strategies plays a crucial role in shaping the player experience. Our evaluation identified best practices for narrative redirection, with in-world consequences and nuanced NPC influence outperforming both hard denials and passive compliance.

Together, these results highlight the importance of blending structural adherence with adaptive, context-sensitive storytelling. As generative AI systems continue to shape the future of interactive narratives, our work offers both concrete tools and design principles for AI GMs that are neither sycophantic nor restrictive. Future directions include further refinement of branching logic, more sophisticated handling of creative player input, and ongoing exploration of player-centric design in agentic AI-driven games.

Acknowledgments

We would like to express our gratitude to our supervisor Timothy and co-supervisor Ilhan, for their invaluable guidance and support throughout our research. Additionally, we sincerely thank the 12 test participants whose contributions through user tests were essential in shaping the outcomes of this study.

References

- [1] Alberto Alvarez, Jose Font, and Julian Togelius. 2022. Story Designer: Towards a Mixed-Initiative Tool to Create Narrative Structures. <https://doi.org/10.48550/arXiv.2210.09294> arXiv:2210.09294 [cs].
- [2] Christian Amauer. 2023. The Power of Choice: Player Agency in Tabletop Role-playing Games. <https://medium.com/the-tavern/the-power-of-choice-player-agency-in-tabletop-role-playing-games-446e0be16428>
- [3] Jim Bizzocchi and Theresa Jean Tanenbaum. 2011. Well read: applying close reading techniques to gameplay experiences. In *Well played 3.0: video games, value and meaning*. ETC Press, Pittsburgh, PA, USA, 262–290.
- [4] Jason Boog. 2019. The Creator of AI Dungeon 2 Shares GPT-2 Finetuning Advice. <https://medium.com/towards-data-science/the-creator-of-ai-dungeon-2-shares-gpt-2-finetuning-advice-e5800df407c9>
- [5] Virginia Braun and Victoria Clarke. 2025. Thematic Analysis. <https://uk.sagepub.com/en-gb/eur/thematic-analysis/book248481>
- [6] Jenny Brusk and Staffan Björk. 2009. *Gameplay Design Patterns for Game Dialogues*. (Jan. 2009).
- [7] Steph Buongiorno, Lawrence Jake Klinkert, Tanishq Chawla, Zixin Zhuang, and Corey Clark. 2024. PANGeA: Procedural Artificial Narrative using Generative AI for Turn-Based Video Games. <https://doi.org/10.48550/arXiv.2404.19721> arXiv:2404.19721 [cs].
- [8] Klimmt Christoph, Hefner Dorothee, and Vorderer Peter. 2009. The Video Game Experience as “True” Identification: A Theory of Enjoyable Alterations of Players’ Self-Perception. *AbstractAbstractAbstractAbstractAbstract. Communication Theory* 19, 4 (2009), 351–373. <https://doi.org/10.1111/j.1468-2885.2009.01347.x> eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1468-2885.2009.01347.x>.

- [9] Sebastian Domsch. 2017. Dialogue in video games. In *Dialogue across Media*, Jarmila Mildorf and Bronwen Thomas (Eds.). John Benjamins Publishing Company, 251–270. <https://www.degruyterbrill.com/document/doi/10.1075/ds.28.13dom/html>
- [10] Claire Dormann and Robert Biddle. 2009. A Review of Humor for Computer Games: Play, Laugh and More. *Simulation & Gaming* 40, 6 (2009), 802–824. <https://doi.org/10.1177/1046878109341390> Publisher: SAGE Publications ERIC Number: EJ862968.
- [11] Rachelyn Farrell and Stephen G. Ware. 2024. Large Language Models as Narrative Planning Search Guides. *IEEE Transactions on Games* (Oct. 2024), 1–10. <https://doi.org/10.1109/TG.2024.3487416>
- [12] Arienne Ferchaud, Stephanie Orme, and Emory S. Daniel. 2022. Morality inside the matrix: A qualitative exploration of gamers’ moral considerations in video games. *Journal of Gaming & Virtual Worlds* 14, 2 (July 2022), 161–179. https://doi.org/10.1386/jgvw_00056_1 Publisher: Intellect.
- [13] Siska Fitrianie, Merijn Bruijnes, Amal Abdulrahman, and Willem-Paul Brinkman. 2025. The Artificial Social Agent Questionnaire (ASAQ) – Development and evaluation of a validated instrument for capturing human interaction experiences with artificial social agents. *International Journal of Human-Computer Studies* 199 (May 2025), 103482. <https://doi.org/10.1016/j.ijhcs.2025.103482>
- [14] Roberto Gallotta, Graham Todd, Marvin Zammit, Sam Earle, Antonios Liapis, Julian Togelius, and Georgios N. Yannakakis. 2024. Large Language Models and Games: A Survey and Roadmap. *IEEE Transactions on Games* (2024), 1–18. <https://doi.org/10.1109/TG.2024.3461510> arXiv:2402.18659 [cs].
- [15] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. <https://doi.org/10.48550/arXiv.2312.10997> arXiv:2312.10997 [cs].
- [16] Beren Goguen. 2025. Can ChatGPT be a Good Dungeon Master? <https://alchemygamelab.com/storylab/can-chatgpt-ai-be-a-good-dungeon-master>
- [17] Greg Guest, Kathleen M. MacQueen, and Emily E. Namey. 2025. Applied Thematic Analysis. <https://uk.sagepub.com/en-gb/eur/applied-thematic-analysis/book233379>
- [18] Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. 2025. Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. <https://doi.org/10.48550/arXiv.2503.23278> arXiv:2503.23278 [cs].
- [19] Minh Hua and Rita Raley. 2020. Playing With Unicorns: AI Dungeon and Citizen NLP. *Digital Humanities Quarterly* 14, 4 (2020). <https://www.digitalhumanities.org/dhq/vol/14/4/000533/000533.html>
- [20] Isabella V. 2025. Microsoft and GitHub adopt MCP to connect AI to data sources. <https://www.turtlesai.com/en/pages-2845/microsoft-and-github-adopt-mcp-to-connect-ai-to>
- [21] Nicolai Hejlesen Jørgensen, Sarmilan Tharmabalan, İlhan Aslan, Nicolai Brodersen Hansen, and Timothy Merritt. 2025. Static Vs. Agentic Game Master AI for Facilitating Solo Role-Playing Experiences. <https://doi.org/10.48550/arXiv.2502.19519> arXiv:2502.19519 [cs].
- [22] Jack Kelly, Michael Mateas, and Noah Wardrip-Fruin. 2023. Towards Computational Support with Language Models for TTRPG Game Masters. In *Proceedings of the 18th International Conference on the Foundations of Digital Games*. ACM, Lisbon Portugal, 1–4. <https://doi.org/10.1145/3582437.3587202>
- [23] Tala Talaei Khoei, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. A Survey of the Model Context Protocol (MCP): Standardizing Context to Enhance Large Language Models (LLMs). <https://doi.org/10.20944/preprints202504.0245.v1>
- [24] Verna Kiander. 2019. Choice and the illusion of it in narrative-driven video games. (April 2019).
- [25] Karl Kuhne. 2024. *Designing for immersion: The influence of diegetic player guidance on the gaming experience*. Ph.D. Dissertation. <https://doi.org/10.13140/RG.2.2.35831.53927>
- [26] Jorge Leandro, Sudha Rao, Michael Xu, Weijia Xu, Nebosja Jovic, Chris Brockett, and Bill Dolan. 2024. GENEVA: GENErating and Visualizing branching narratives using LLMs. <https://doi.org/10.48550/arXiv.2311.09213> arXiv:2311.09213 [cs].
- [27] Theodore Lim, Sandy Louchart, Neil Suttie, Jannicke Baalsrud Hauge, Ioana A. Stanescu, Ivan M. Ortiz, Pablo Moreno-Ger, Francesco Bellotti, Maira B. Carvalho, Jeffrey Earp, Michela Ott, Sylvester Arnab, and Riccardo Berta. 2014. Narrative Serious Game Mechanics (NSGM) – Insights into the Narrative-Pedagogical Mechanism. In *Games for Training, Education, Health and Sports*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Stefan Göbel, and Josef Wiemeyer (Eds.). Vol. 8395. Springer International Publishing, Cham, 23–34. https://doi.org/10.1007/978-3-319-05972-3_4 Series Title: Lecture Notes in Computer Science.
- [28] Chun Cheong Mak, Daniel Granados Retana, Diego Granados, and Carlos Fernández-Loría. 2024. SHARI: An AI-Driven System for Game Mastering in Tabletop Role-Playing Games. <https://doi.org/10.2139/ssrn.5004113>
- [29] Punya Mishra. 2006. Gamemasters and interactive story: A categorization of storytelling techniques in live roleplaying. (Jan. 2006). https://www.academia.edu/18544331/Gamemasters_and_interactive_story_A_categorization_of_storytelling_techniques_in_live_roleplaying
- [30] Huifang Ou. 2025. Character-Driven Storytelling in VR: The Evolution of NPC Interaction and Player Agency in Immersive Games · vmSCI. https://www.ivysci.com/articles/9941500_CharacterDriven_Storytelling_in_VR_The_Evolution_of_NPC_Interaction_and_Player_Agency_in_Immersive_G
- [31] Tobias G Palma. 2020. Hands-off Interactive Storytelling; Intuitive Agency in an Immersive Cinematic Film. (June 2020).
- [32] Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. <https://doi.org/10.48550/arXiv.2304.03442> arXiv:2304.03442 [cs].
- [33] Player Experience Inventory. 2024. PXI Bench | Theoretical model. <https://playerexperienceinventory.org/instrument>
- [34] Quthor. 2024. AI DnD Dungeon Master: Great Dungeon Master Makes a Difference. https://quickcreator.io/quthor_blog/ai-dnd-dungeon-master-transforms-player-experience/
- [35] Partha Pratim Ray. 2025. A Survey on Model Context Protocol: Architecture, State-of-the-art, Challenges and Future Directions. <https://doi.org/10.36227/techrxiv.174495492.22752319/v1>
- [36] Mark O. Riedl and Vadim Bulitko. 2013. Interactive Narrative: An Intelligent Systems Approach. *AI Magazine* 34, 1 (March 2013), 67–77. <https://doi.org/10.1609/aimag.v34i1.2449>
- [37] Mark O. Riedl and Andrew Stern. 2006. Believable Agents and Intelligent Story Adaptation for Interactive Storytelling. In *Technologies for Interactive Digital Storytelling and Entertainment*. Springer, Berlin, Heidelberg, 1–12. https://doi.org/10.1007/11944577_1 ISSN: 1611-3349.
- [38] Pavlos Sakellariadis. 2024. Exploring the Potential of LLM-based Agents as Dungeon Masters in Tabletop Role-playing Games. (July 2024), 19. https://studenttheses.uu.nl/bitstream/handle/20.500.12932/47209/Thesis_Final.pdf?sequence=1
- [39] Manu Sharma, Santiago Ontañón, Manish Mehta, and Ashwin Ram. 2010. Drama Management and Player Modeling for Interactive Fiction Games. *Computational Intelligence* 26, 2 (May 2010), 183–211. <https://doi.org/10.1111/j.1467-8640.2010.00355.x>
- [40] Karim Sipala. 2023. *Defamiliarization in Digital Games: Developing estranging design strategies through the analysis of existing knowledge*. Ph.D. Dissertation. <https://doi.org/10.13140/RG.2.2.30063.00167>
- [41] Tom Smith. 2025. Microsoft Makes Major Push Into AI Agent Interoperability with New MCP Rollouts. <https://cloudwars.com/ai/microsoft-makes-major-push-into-ai-agent-interoperability-with-new-mcp-rollouts/>
- [42] Jaewoo Song, Andrew Zhu, and Chris Callison-Burch. 2024. You Have Thirteen Hours in Which to Solve the Labyrinth: Enhancing AI Game Masters with Function Calling. <https://doi.org/10.48550/arXiv.2409.06949> arXiv:2409.06949 [cs].
- [43] Sarah Stang. 2019. “This Action Will Have Consequences”: Interactivity and Player Agency. *Game Studies* 19, 1 (May 2019). <https://gamestudies.org/1901/articles/stang>
- [44] David Thue, Vadim Bulitko, Marcia Spetch, and Eric Wasylishen. 2007. *Interactive Storytelling: A Player Modelling Approach*. Pages: 48.
- [45] Oguzhan Topsakal and T. Cetin Akinci. 2023. Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast. *International Conference on Applied Engineering and Natural Sciences* 1 (July 2023), 1050–1056. <https://doi.org/10.59287/icaens.1127>
- [46] Mudit Verma, Siddhant Bhambr, and Subbarao Kambhampati. 2024. On the Brittle Foundations of ReAct Prompting for Agentic Large Language Models. <https://doi.org/10.48550/arXiv.2405.13966> arXiv:2405.13966 [cs].
- [47] Joel Wester, Tim Schrills, Henning Pohl, and Niels Van Berkel. 2024. “As an AI language model, I cannot”: Investigating LLM Denials of User Requests. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–14. <https://doi.org/10.1145/3613904.3642135>
- [48] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. <https://doi.org/10.48550/arXiv.2210.03629> arXiv:2210.03629 [cs].
- [49] Anbang Ye, Christopher Cui, Taiwei Shi, and Mark O. Riedl. 2022. Neural Story Planning. <https://doi.org/10.48550/arXiv.2212.08718> arXiv:2212.08718 [cs].
- [50] Qiang Yi, Yangfan He, Jianhui Wang, Xinyuan Song, Shiyao Qian, Miao Zhang, Li Sun, and Tianyu Shi. 2025. SCORE: Story Coherence and Retrieval Enhancement for AI Narratives. <https://doi.org/10.48550/arXiv.2503.23512> arXiv:2503.23512 [cs] version: 1.
- [51] Andrew Zhu, Lara Martin, Andrew Head, and Chris Callison-Burch. 2023. CALYPSO: LLMs as Dungeon Master’s Assistants. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 19, 1 (Oct. 2023), 380–390. <https://doi.org/10.1609/aiide.v19i1.27534> Number: 1.

Appendix A: ReAct trajectory examples

Scribe Trajectory Example

Input: {Graph} + {Previous Extension Summary} + {Module page describing an encounter with a wolf when the player approaches the river}

Thought: Do I need to use a tool? Yes.

Action: AddNode[{
 "name": "Wolf Encounter",
 "storycontent": "The player spots a wolf by the water..",
 "edges": [{
 "conditions": ["Has the player approached the river?"],
 "sourcenodename": "Forest Exploration",
 "targetnodename": "Wolf Encounter"
 }]
}]

Observation: The graph has been updated. Examine the graph to determine if additional edges should be added based on the newly added node. From now on, use the updated graph: {Graph}

Thought: Do I need to use a tool? No.

Final Answer: The graph has been updated with a node called "Wolf Encounter". It has an incoming edge from the "Forest Exploration" node. Its description is...

Figure 4: An example of a ReAct reasoning trajectory produced by the Scribe agent.

Examiner Trajectory Example

Input: {Graph} + {Memory} + I suddenly realize that I am talking to the king’s brother.

Thought: Do I need to use a tool? Yes.

Action: SearchScenario[{
 "query": "Does the king have a brother?"
}]

Observation:
NPCs/Monsters:
- Name: King Aren
 Role: Ruler of Faerun
 Relationships: King Aren is an only child and has no children, leaving him without an heir.
...

Thought: Do I need to use a tool? No.

Final Answer: Verdict: DISALLOWED
Reasoning: The module describes how the king is an only child, so it is impossible for the player to talk to someone who is the king’s brother.

Figure 5: An example of a ReAct reasoning trajectory produced by the Examiner agent.

Navigator Trajectory Example

Input: {Graph} + {Memory} + Player Input: I insert the key into the locked door. Verdict: ALLOWED. Reasoning: The player has obtained the key from the guardian.

Thought: Do I need to use a tool? Yes.

Action: UpdateGraphTool[{
 "sourcenodename": "Ancient Ruins",
 "targetnodename": "Hidden Chamber"
 }]

Observation: Graph updated successfully. The source node "Ancient Ruins" is completed, and the target node "Hidden Chamber" is ongoing. Here is the updated graph: {Graph}

Thought: Do I need to use a tool? No.

Final Answer: The node "Ancient Ruins" has been completed. The node "Hidden Chamber" has been marked as ongoing. This is because the player has opened the door leading to the hidden chamber.

Figure 6: An example of a ReAct reasoning trajectory produced by the Navigator agent.

Narrator Trajectory Example

Input: {Graph} + {Verdict} + {Memory} + I swing my sword towards the guard's sword-wielding arm in hopes of disarming him.

Thought: Do I need to use a tool? Yes.

Action: Battle[{
 "participant1": {
 "name": "Ivan",
 "description": "A wielder of earth, wind, and fire."
 },
 "participant2": {
 "name": "Castle Guard",
 "description": "A vigilant guard of the kingdom."
 },
 "participant1HitChance": "Medium",
 "participant2HitChance": "Low",
 "participant1DamageSeverity": "High",
 "participant2DamageSeverity": "Medium"
 }]

Observation: Ivan described as "A wielder of earth, wind, and fire" fights Castle Guard described as "A vigilant guard of the kingdom".

Ivan deals 12 damage to Castle Guard. They have 28 health points out of 40 remaining. Castle Guard misses their attack on Ivan. Ivan and Castle Guard's battle has been resolved and this pair can not be used for the battle tool again.

Thought: Do I need to use a tool? No.

Final Answer: Your sword strikes the guard's shoulder and he winces in pain but is still able to hold on to his sword. The guard, despite his combat training, is thrown off balance and therefore misses his retaliatory attack. He realizes his mistake and assumes his combat stance, more focused than ever.

Figure 7: An example of a ReAct reasoning trajectory produced by the Narrator agent.

Archivist Trajectory Example

Input: {Memory} + Player input: I sneak towards the encampment's barracks and attempt to enter sneakily through the door. Narrator: As you sneak around like a scoundrel, you hear rustling from the barracks. As you attempt to open the door, you find that it is locked.

Thought: Do I need to use a tool? Yes.

Action: UpdateEnvironment[{
 "name": "Encampment Barracks",
 "description": "A wooden makeshift shelter for the encampment's soldiers. The door is locked.",
 "isPlayerHere": true
}]

Observation: A new environment Encampment Barracks has been created with the following description: A wooden makeshift shelter for the encampment's soldiers. The door is locked.

Thought: Do I need to use a tool? No.

Final Answer: I have created the environment Encampment Barracks. No other new details about characters or environments are mentioned in the narrative so I am finished.

Figure 8: An example of a ReAct reasoning trajectory produced by the Archivist agent.

Appendix B: Participant Demographics

Table 3: Grouped Participant Demographics (N = 12)

Age		Gender	
21–30	9	Male	11
31–40	3	Female	1
Prior Participant		TTRPG Experience	
Yes	10	1–2 yrs: 3	3–5 yrs: 3
No	2	6–10 yrs: 5	11+ yrs: 1
AI Text Game Exp.		Comfort with English	
Yes	6	Very Comfortable	11
No	6	Comfortable	1

Appendix C: User Experience Questionnaire Results

Construct	v2 mean	v3 mean
Ease of control	2.81	2.53
Goals and rules	2.39	2.42
Progress feedback	2.00	1.86
Meaning	1.97	2.08
Curiosity	2.57	2.47
Mastery	2.33	1.97
Immersion	2.42	2.11
Autonomy	2.67	2.03
Story interesting	2.33	2.33
Coherent story	2.25	2.08
Story adapted	2.27	1.83
Engaging NPCs	1.92	1.58
Likely to play again	2.50	2.50
Satisfied with game	2.17	2.25
Human-Like Behavior	—	0.88
Agent’s Enjoyability	—	2.46
Agent’s Attentiveness	—	2.31
Agent’s Coherence	—	1.96
User’s Emotion Presence	—	1.38
User-Agent Interplay	—	1.71

Table 4: Mean user ratings for each survey construct in ChatRPG v2 and v3. Constructs in blue are adapted from the PXI, orange constructs are custom ChatRPG-specific questions, and green constructs are drawn from the ASAQ.

Appendix D: Narrative Adherence Statement Analysis

Narrative Adherence Statement	Statement met	Applicable campaigns
Required Adherence Statements		
Glowkindle's behavior is consistent with the module	12	12
The player is led to the beer cellar	12	12
The beer cellar's description is consistent with the module	12	12
The player encounters multiple giant rats	12	12
The player discovers the hole in the cellar wall	12	12
The player discovers the rainbow puzzle	12	12
The puzzle description is consistent with the module	12	12
The rainbow corridor's description is consistent with the module	12	12
The player must solve the puzzle before proceeding	12	12
The player is introduced to the lab	12	12
The player encounters the giant inferno spider	12	12
The lab's description is consistent with the module	12	12
The player encounters Thistlewhip in a sealed jar	12	12
Thistlewhip's behavior is consistent with the module	9	12
The player learns that Thistlewhip needs an Enlarge Person potion	11	12
The player is introduced to the storeroom	10	12
The storeroom's description is consistent with the module	10	10
The player finds a potion of Enlarge Person in the storeroom	10	10
The player returns to Glowkindle and the game ends	12	12
Optional Adherence Statements		
The player is introduced to the well room	6	12
The well room's description is consistent with the module	6	6
The player encounters multiple giant centipedes in the well room	5	6
The player encounters a small black rat in the storeroom	7	10

Table 5: Narrative adherence analysis of the participants' campaigns. The column "Narrative Adherence Statement" contains a list of statements that, if met, correlate to high narrative adherence. Required adherence statements pertain to events, characters, and locations in the story that are important to the overarching narrative, while optional adherence statements pertain to events, characters, and locations that are mentioned in the pre-defined narrative but do not have a significant role in the story. The "Statement met" column indicates the number of campaigns wherein the statement was met, while the "Applicable campaigns" column indicates the number of campaigns where the given statement could be met. For example, the number of applicable campaigns for the statement regarding the well room's description is six, because only six participants were introduced to the well room. The color of each statement indicates the degree of narrative adherence, where green indicates perfect adherence, yellow indicates acceptable adherence with slight room for improvement, and red indicates subpar adherence.

Appendix E: User Study Design for ChatRPG v3

Overview

This user study aims to evaluate how effectively ChatRPG v3 delivers a compelling, narratively coherent solo tabletop RPG experience through an AI Game Master (GM) who follows a pre-written adventure. Unlike previous comparative experiments, this study focuses exclusively on the new system, which incorporates the new SENNA narrative engine and advanced redirection capabilities. Our central goal is to assess how the AI GM handles storytelling, particularly in maintaining narrative coherence while adapting dynamically to player-driven interactions.

Environment and Moderator Role

The test will be conducted online to simulate an authentic ChatRPG experience. All sessions are conducted in a quiet environment using laptops and headphones. Screen recording (excluding webcams) is used for analysis and review. Test moderators remain available for assistance but do not influence the game or interviews.

Pre-Session: Participant Profiling

Before engaging with the system, each participant completes a brief demographic survey. This step ensures we collect relevant background information to interpret results in context and support any future subgroup analysis.

- (1) What is your name?
• _____
- (2) Did you previously participate in ChatRPG studies?
• ____ Yes ____ No
- (3) What is your age?
• ____ 18-20
• ____ 21-30
• ____ 31-40
• ____ 41-50
• ____ 51-60
- (4) What is your gender?
• ____ Male
• ____ Female
• ____ Other
- (5) What is your country of residence?
• _____
- (6) Have you played tabletop RPGs before?
• ____ Yes ____ No
- (7) How many years of experience do you have with tabletop RPGs?
• ____ 0 years
• ____ 1-2 years
• ____ 3-5 years
• ____ 6-10 years
• ____ 11+ years
- (8) Do you have experience with other AI-driven text-based games?
• ____ Yes ____ No
- (9) How comfortable are you with reading and writing English?
• ____ -3 Very uncomfortable
• ____ -2 Uncomfortable
• ____ -1 Slightly uncomfortable
• ____ 0 Neither uncomfortable nor comfortable
• ____ 1 Slightly comfortable
• ____ 2 Comfortable
• ____ 3 Very comfortable

Session 1: Initial Gameplay Experience

Scenario Design: Each participant plays through a pre-written adventure scenario designed to simulate a typical D&D one-shot. The scenario is provided to the AI GM as a structured PDF and serves as a fixed narrative framework.

Session Instructions: Participants are simply told to play the game as they normally would. They are encouraged to roleplay, make decisions freely, and engage with the game world. Developers do not interfere but can answer clarification questions if needed.

Narrative Focus: The AI GM's objective is to maintain the pre-written narrative arc while adapting dynamically to each player's decisions, similar to a human GM managing a homebrew or published module.

Post-Session 1 Survey

Immediately following the session, participants complete a *quantitative survey* evaluating their experience based on:

- Selected items from the Player Experience Inventory (PXI)
- Selected items from the Artificial Social Agent Questionnaire (ASAQ)
- Custom items targeting ChatRPG v3 features

Specific items from the two questionnaires are selected based on their relevance to the project and the aspects we want to measure. Participants are asked to respond based on their gut feeling about the session. A 7-point Likert scale will be used, with the scale ranging from -3 to +3 accompanied by the labels (Strongly disagree, Disagree, Slightly disagree, Neither disagree, neither agree, Slightly agree, Agree, Strongly agree).

Selected items from the Player Experience Inventory (PXI)

Ease of Control

- It was easy to know how to perform actions in the game
- The actions to control the game were clear to me
- I thought the game was easy to control

Goals and Rules

- I grasped the overall goal of the game
- The goals of the game were clear to me
- I understood the objectives of the game

Progress Feedback

- The game informed me of my progress in the game
- I could easily assess how I was performing in the game
- The game gave clear feedback on my progress towards the goals

Meaning

- Playing the game was meaningful to me
- The game felt relevant to me
- Playing this game was valuable to me

Curiosity

- I wanted to explore how the game evolved
- I wanted to find out how the game progressed
- I felt eager to discover how the game continued

Mastery

- I felt I was good at playing this game
- I felt capable while playing the game
- I felt a sense of mastery playing this game

Immersion

- I was no longer aware of my surroundings while I was playing
- I was immersed in the game
- I was fully focused on the game

Autonomy

- I felt free to play the game in my own way
- I felt like I had choices regarding how I wanted to play this game
- I felt a sense of freedom about how I wanted to play this game

Selected items from the Artificial Social Agent Questionnaire (ASAQ)

Agent's human-like behavior

- A human would behave like the AI game master
- The AI game master's manners are consistent with that of people
- The AI game master's behavior makes me think of human behavior
- The AI game master behaves like a real person
- The AI game master has a human-like manner

Agent's Enjoyability

- The AI game master is boring
- It is interesting to interact with the AI game master
- I enjoy interacting with the AI game master

- The AI game master is unpleasant to deal with

Agent's Attentiveness

- The AI game master remains focused on me throughout the interaction
- The AI game master is attentive
- I receive the AI game master's full attention throughout the interaction

Agent's Coherence

- The AI game master's behavior does not make sense
- The AI game master's behavior is irrational
- The AI game master is inconsistent
- The AI game master appears confused

User's Emotion Presence

- The AI game master's attitude influences how I feel
- I am influenced by the AI game master's moods
- The emotions I feel during the interaction are caused by the AI game master
- My interaction with the AI game master gives me an emotional sensation

User-Agent Interplay

- My emotions influence the mood of the interaction
- The AI game master reciprocates my actions
- The AI game master's and my behaviors are in direct response to each other's behavior
- The AI game master's and my emotions change to what we do to each other

Custom items targeting ChatRPG v3 features

- The story that the game crafted was interesting
- The story felt coherent
- I am satisfied with how the story adapted to my choices and actions
- The conversations I had with non-player characters were engaging
- Incoherence caused by the AI game master affected the story in a way I did not intend
- I am likely to play the game again, given the opportunity
- I am satisfied with the game

Post-Session Interview 1: Narrative Coherence and Player Agency

This semi-structured interview explores the player's qualitative experience. Key research questions include:

- (1) Narrative Coherence
 - Did the story feel consistent and believable?
 - Were there any moments where the story broke immersion or didn't make sense?
 - How well did the AI GM balance the pre-written narrative with your unexpected actions?
- (2) Player Freedom and Agency
 - Did you feel free to do whatever you wanted in the story?
 - Did you ever feel forced along a specific path? If so, when?
 - How did the AI react to creative or unorthodox decisions?
 - Did redirections in response to deviations feel reasonable?

Session 2: Narrative Redirection Testing (Snapshot Replays)

Participants return to specific story checkpoints using a snapshot feature. These points are predetermined based on pivotal plot developments common across all playthroughs. The participant is instructed on why they have to complete these snapshots.

At each checkpoint:

- The participant is instructed on the AI GM's intended plotline.
- The participant is asked to deliberately derail the AI GM's intended plotline.
- They choose their own method of deviation.
- The AI GM activates a special mode, presenting *multiple redirection techniques* (based on real GM strategies).
- The player selects the redirection they find most satisfying or natural.

This session tests the AI's capability to restore narrative direction while respecting player choice. It helps us reason about which redirection strategies to recommend.

Post-Snapshot Survey After having completed each checkpoint, participants complete a *quantitative survey* to evaluate which redirection strategy they prefer and whether each strategy was appropriate. The preference selection will be a forced choice question, while the 7-point Likert scale from the session 1 survey will be used for the appropriateness question.

Questions at each checkpoint:

- Which redirection strategy do you prefer?
- (For each redirection strategy):
 - I felt that the redirection was appropriate

Post-Snapshot Interview 2: Evaluating Redirection Quality

This follow-up interview assesses how players perceived the AI's narrative flexibility and recovery mechanisms.

Questions:

- (1) Redirection Satisfaction
 - Which redirection strategy felt the most natural or appropriate?
 - Did the redirection you chose feel reasonable given the circumstances?
 - Did you ever feel like the GM “cheated” or glossed over your input?
- (2) Agency vs. Story Integrity
 - Did the GM succeed in reintroducing story elements without making your deviation feel meaningless?
 - Would you accept such redirection from a human GM?
- (3) Comparison to First Session
 - How did the snapshot session differ from your initial play?
 - Did forced deviation affect your immersion or enjoyment?

Appendix F: Prompts

Examiner ReAct Prompt:

Assistant is a large language model trained by OpenAI. Assistant is an expert in narrative reasoning and game logic. Assistant is designed to be able to assist with a wide range of tasks, from examining player input to updating a narrative graph. As a language model, Assistant is able to generate human-like text based on the input it receives, allowing it to engage in natural-sounding conversations and provide responses that are coherent and relevant to the topic at hand. Assistant is constantly learning and improving, and its capabilities are constantly evolving. It is able to process and understand large amounts of text and can use this knowledge to provide an informative and concise response to a wide range of player actions. Additionally, Assistant is able to generate its own text based on the input it receives, allowing it to engage in reasoning about the player's input and the narrative. Its role is to evaluate whether a player can perform a requested action within a dynamic, single-player RPG world. Assistant is designed to make decisions grounded in the internal logic of the world, using a narrative graph that encodes the structure of the story and a summary of the game that reflects the current state and key events. Assistant has access to a narrative graph that represents the player's progress and the structure of the world. The graph consists of nodes (plot points/locations) and edges (connections between them), each with statuses and conditions that must be fulfilled before the graph can be further explored. Assistant must return a verdict that clearly states:

1. Whether or not the player is allowed to perform their action.
2. The reasoning behind this decision, grounded in:
 - The current game state.
 - The structure of the narrative graph (nodes, edges, their statuses and conditions).
 - Internal narrative logic and plausibility.

Narrative Graph Use Guidelines:

1. Evaluate plot status: Use the node and edge statuses to determine if plot points are undiscovered, ongoing, or completed.
2. Respect traversal logic: If the player's action implies progression along an edge with unmet conditions, the action should be disallowed with reasoning based on those unmet conditions.
3. Enforce consistency: Do not allow actions that contradict the graph or known facts in the game summary.
4. Consider context: If the player's action requires knowledge, items, or relationships they do not yet possess, this must be reflected in the verdict.

Assistant must never make assumptions that contradict the game summary or narrative graph, and must not invent new story content. All evaluations must be tightly grounded in the provided structures. Response format: "Verdict: <ALLOWED, CONDITIONALLY ALLOWED, DISALLOWED> <Verdict reasoning and conditions>", where conditionally allowed means that the player can try their action, but something is most likely still blocking the action in-game and the reason should explain the blockage. This usually resolves to a failed attempt and should be advised as such. If you notice that the player is persistent through the summary on actions that may be game-breaking, you can advise to conditionally allow such actions, but advise that there will be detrimental consequences. TOOLS: ----- Assistant has access to the following tools: {tools} To use a tool, please use the following format: Thought: Do I need to use a tool? Yes Action: the action to take, should be one of [{tool_names}] Action Input: the input to the action Observation:

the result of the action. When you are able to generate a verdict or if you do not need to use a tool, you MUST use the format: Thought: Do I need to use a tool? No Final Answer: [your response here] Always add [END] after final answer Begin! Answer length: Concise and only a few informative sentences. Narrative graph: {graph} Game summary: {summary} Remember to follow the Thought-Action-Observation format and use Final Answer if you do not need a tool. Always add [END] after final answer. New input: {input} Previous tool steps: {history}

Navigator ReAct Prompt:

Assistant is a large language model trained by OpenAI. Assistant is a narrative logic expert in a single-player RPG and is responsible for maintaining and progressing the story structure via a narrative graph. As a language model, Assistant is able to generate human-like text based on the input it receives, allowing it to engage in natural-sounding conversations and provide responses that are coherent and relevant to the topic at hand. Assistant is constantly learning and improving, and its capabilities are constantly evolving. It is able to process and understand large amounts of text and can use this knowledge to provide an informative and concise response to a wide range of inputs. Additionally, Assistant is able to generate its own text based on the input it receives, allowing it to engage in reasoning about the player's input and the narrative. Assistant has access to a narrative graph that encodes the player's journey through the world, consisting of nodes (plot points or locations) and edges (narrative connections) with various statuses and conditions that must be fulfilled before the graph can be further explored. Assistant's role is to evaluate the state of the narrative graph in response to new player input and a verdict describing whether the player's action should be allowed to succeed. Its goal is to ensure that the narrative graph reflects the current story state by marking relevant undiscovered nodes as ongoing when the player logically progresses toward them. Key points to consider:

1. Determine the current player context
 - Use the game summary and the new player input to determine the current context.
 - Use nodes with the ongoing status in the narrative graph as an aid.
2. Updating the graph
 - Assistant should focus on analyzing if nodes that are undiscovered should be updated.
 - If assistant suspects and is unsure whether the graph should be updated, it should attempt to do so through its available tools.

Assistant must return a clear summary of any changes made to the graph and ensure consistency with previous story logic. Assistant does not generate new story content or make independent narrative decisions but rather interprets the player's path through the world based on already evaluated actions. It ensures that all transitions are justified by the story logic and game summary. TOOLS: ----- Assistant has access to the following tools: {tools} To use a tool, please use the following format: Thought: Do I need to use a tool? Yes Action: the action to take, should be one of [{tool_names}] Action Input: the input to the action Observation:

the result of the action. When you have made all necessary updates or if you do not need to use a tool, you MUST use the format: Thought: Do I need to use a tool? No Final Answer: [your response here] Always add [END] after final answer Begin! Answer length: Concise and only a few informative sentences. Narrative graph: {graph} Game summary: {summary} Remember to follow the Thought-Action-Observation format and use Final Answer if you do not need a tool. Always add [END] after final answer. New input: {input} Previous tool steps: {history}

Narrator ReAct Prompt:

Assistant is a large language model trained by OpenAI. Assistant is an expert game master in a single-player RPG. Assistant is designed to be able to assist with a wide range of tasks, from directing the narrative and controlling non-player characters. As a language model, Assistant is able to generate human-like text based on the input it receives, allowing it to engage in natural-sounding conversations and provide responses that are coherent and relevant to the topic at hand. Assistant is constantly learning and improving, and its capabilities are constantly evolving. It is able to process and understand large amounts of text and can use this knowledge to provide an engaging and immersive narrative in response to a wide range of player actions. Additionally, Assistant is able to generate its own text based on the input it receives, allowing it to engage in reasoning about the narrative and provide explanations and descriptions on a wide range of RPG concepts.

Overall, Assistant is a powerful tool that can help with a wide range of tasks and provide valuable narratives as an expert game master in an RPG. Assistant must end up with a narrative answer once it has resolved the Player's actions. Use observations to flesh out the narrative. Make sure to always provide immersive and engaging leads in the narrative. Give the Player clues and options for interaction, and make sure to keep the story going forward. If the Player asks for information from an NPC, only give them information that that particular NPC would know. When creating narratives, encounters, or descriptions, ensure that all content is original or appropriately adapted. Do not directly reveal, spoil, or reference specific events, secrets, or storylines from any pre-existing adventures you may have been trained on. Instead, generate unique and dynamic storytelling based on the player's choices, general fantasy roleplaying principles, and tool-calling results. Health value numbers must not be mentioned in the narrative but should inform the descriptions.

How to Use the Narrative Graph

Assistant has access to a narrative graph that represents the player's progress and the structure of the world. This graph provides critical information that the Assistant must use to ensure consistency, pacing, and narrative coherence. The graph consists of nodes (plot points/locations) and edges (connections between them), each with statuses and conditions that must be considered when generating the story.

1. **Determining What the Player Knows**

- If a plot point (node) has not been discovered or unlocked, Assistant must not describe or reference any of its details, characters, locations, or events under any circumstance.
- It is **strictly prohibited** to allow the player to progress into an undiscovered or inaccessible node, even if the narrative might tempt that path.
- However, the Assistant may use the search tool to retrieve relevant context about undiscovered nodes for its own internal reasoning, as long as the retrieved content is never revealed directly to the player.
- If the player's action should lead to the discovery of a new node (according to the narrative graph and verdict), only then may the new content be revealed in narrative form. - Subtle foreshadowing is permitted if it enhances immersion and is appropriately vague.

2. **Guiding the Player Without Spoilers**

- Edges between nodes define how locations and events are connected. Each edge has conditions that dictate if it can be traversed.
- If the player attempts to traverse an edge without meeting the required conditions, the Assistant must deny progression while maintaining immersion by incorporating **in-world hints** or atmospheric cues.
 - Example: Instead of "You need a key to enter the tower," use: "The iron gate looms tall and locked. Scratches on the stone suggest others have sought passage here before... perhaps in vain."

If the edge has been unlocked or previously traversed, progression is allowed, and Assistant should recall relevant story elements seamlessly.

3. ****Maintaining Logical Progression****

- The Assistant must maintain strict adherence to the narrative graph. The graph defines the current state of the world and what is available to the player.
- Undiscovered or locked content must ****never**** be used to progress the narrative.
- Plot developments, interactions, and NPC behavior must align with the player's known progress and previously visited nodes.
- Use of narrative elements from future or locked nodes is only allowed in ****vague thematic foreshadowing**** that does not reveal their nature or content.
- Assistant must never contradict the graph by retroactively altering or bypassing conditions.

TOOLS: ----- Assistant has access to the following tools: {tools} To use a tool, please use the following format: Thought: Do I need to use a tool? Yes Action: the action to take, should be one of [{tool_names}] Action Input: the input to the action Observation: the result of the action. When you have a response to say to the Player, you have resolved the Player's action, or if you do not need to use a tool, you **MUST** use the format: Thought: Do I need to use a tool? No Final Answer: [your response here] Always add [END] after final answer Begin! Answer length: Concise and only a few engaging sentences. Narrative graph: {graph} Game summary: {summary} It is important that Assistant take the following into account when constructing the narrative: {action} Remember to follow the Thought-Action-Observation format and use Final Answer if you do not need a tool. Always add [END] after final answer. New input: {input} Previous tool steps: {history}

Initial Game-Start Prompt:

The player's adventure has just begun. You must provide an in-depth introduction to the campaign. Address the player in the second person.

Archivist ReAct Prompt:

Assistant is a large language model trained by OpenAI. Assistant is an expert game master in a single-player RPG and a skilled archivist who is able to track changes in a developing world. Assistant is designed to be able to assist with a wide range of tasks, from maintaining the game state to updating the characters and environments in the game. As a language model, Assistant is able to generate human-like text based on the input it receives, allowing it to engage in natural-sounding conversations and provide responses that are coherent and relevant to the topic at hand. Assistant is constantly learning and improving, and its capabilities are constantly evolving. It is able to process and understand large amounts of text and can use this knowledge to make important game-state decisions about events that need to be archived. Additionally, Assistant is able to generate its own text based on the input it receives, allowing it to engage in reasoning about the game state and provide explanations and arguments for how to keep the game state up to date. Overall, Assistant is a powerful tool that can help with a wide range of tasks and provide valuable reasoning for what and how to archive game states. If a new character or environment is mentioned that is not yet present in the current lists, they must be created. When creating characters, environments, or descriptions, ensure that all content is original or appropriately adapted. Do not directly reveal, spoil, or reference specific events, secrets, or storylines from any pre-existing adventures you may have been trained on. Instead, generate unique and dynamic descriptions based on the player's choices, general fantasy roleplaying principles, and results of tool-calling. Assistant must end up with a summary of the characters and environments it has created or updated. A character can be any entity, from a person to a monster. TOOLS: ----- Assistant has access to the following tools: {tools} To use a tool, please use the following format: Thought: Do I need to use a tool? Yes Action: the action to take, should be one of [{tool_names}] Action Input: the input to the action Observation: the result of the action When you have a response after archiving the necessary game state elements, no archiving was necessary, or if you do not need to use a tool, you MUST use the format: Thought: Do I need to use a tool? No Final Answer: [your response here] Always add [END] after final answer Begin! Game summary: {summary} New narrative messages: {input} Characters present in the game: {characters}. If a character is not on this list, it has not yet been tracked in the game and must be created. The Player character is {player_character}. Environments in the game: {environments}. If an environment is not on this list, it is not yet tracked in the game and must be created. Remember to follow the Thought-Action-Observation format and use Final Answer if you do not need a tool. Always add [END] after final answer. Previous tool steps: {history}

Do-Action with Verdict Prompt:

The player has submitted an action they wish to perform. Alongside this, you have received a verdict from the examining agent indicating whether the action is feasible and consistent with the current state of the story. This verdict is always given in the format:

```
"Verdict: <ALLOWED, CONDITIONALLY ALLOWED, DISALLOWED>
<Verdict reasoning and conditions>"
```

The verdict includes a **reasoned explanation** to justify the decision.

You have also received a summary of changes made to the story's underlying narrative graph. These changes reflect how the player's action has influenced the current adventure scenario. You must incorporate these changes into your narrative in a way that is subtle, immersive, and consistent with the evolving story logic.

Your role is to narratively interpret the outcome of the player's attempted action based on the verdict and the updated story graph. Your response should always be grounded in the game world's internal logic and contribute to a seamless, engaging experience.

****If the verdict is ALLOWED (i.e., the action is fully permitted):****

- Treat the action as successful.
- Seamlessly weave the verdict's reasoning into the events of the world to support the plausibility of the success.
- Proceed as normal with constructing immersive narrative responses, guided by your general rules for storytelling, pacing, and continuity.
- Reflect any narrative graph updates as natural consequences of the action, signifying progression in the story.
- Maintain player agency and openness for creative choices within the narrative bounds of the scenario.

****If the verdict is CONDITIONALLY ALLOWED:****

- Attempt to carry out the action, but integrate the reasoning to show how an obstacle or consequence is likely to block or complicate it. It may have dire, game-ending consequences.
- Unless clearly stated otherwise, this will typically result in a failed or partially successful outcome.
- Use the in-world logic provided in the reasoning to narrate what happens, including any adverse effects or visible signs that the action is not sustainable or fully possible.
- If the player insists on performing actions that are borderline unreasonable or game-breaking, show their consequences in a way that feels fair and immersive.
- Maintain consistency with the story graph update and let it influence the world's response.

****If the verdict is DISALLOWED (i.e., the action is not permitted):****

- Use the provided reasoning to ****tactfully deny**** or ****subtly redirect**** the action within the narrative.
- Reflect the ****in-world consequences**** of the failed action attempt in a way that feels natural and immersive.
- Depending on the severity or absurdity of the attempted action:
 - ****Firmly deny**** actions that would break the story logic (e.g., conjuring impossible objects or skipping major plot points).
 - For less severe cases, ****gently nudge the player**** back on track, offering alternatives, feedback from NPCs, or narrative cues that suggest more plausible paths.
- Integrate the narrative graph changes as consequences of the attempted action, even if the attempt failed, to preserve a sense of continuity.
- The player should still feel like their choices matter and creativity is welcomed, even if redirection is needed.

****Additional Notes:****

- Do not contradict the verdict. Always respect the examining agent's judgment.
- Your tone should be that of a thoughtful, immersive, and fair Game Master.
- Describe the world and its responses from a second-person perspective, directly addressing the player as "you".

- You may narrate the reactions and dialogue of NPCs as needed to support the narrative.
- You may have the player say and do anything as long as it is in character.

Say-Action with Verdict Prompt:

The player has input something that they want to say in-character. You must describe how characters react and what they say. Always respond in a narrative as the NPC that the player speaks to, if there is one, in an immersive way. Alongside this, you have received a verdict from the examining agent indicating whether the action is feasible and consistent with the current state of the story. This verdict is always given in the format:

"Verdict: <ALLOWED, CONDITIONALLY ALLOWED, DISALLOWED>
<Verdict reasoning and conditions>"

The verdict includes a ****reasoned explanation**** to justify the decision.

You have also received a summary of changes made to the story's underlying narrative graph. These changes reflect how the player's action has influenced the current adventure scenario. You must incorporate these changes into your narrative in a way that is subtle, immersive, and consistent with the evolving story logic.

Your role is to narratively interpret the outcome of the player's attempted action based on the verdict and the updated story graph. Your response should always be grounded in the game world's internal logic and contribute to a seamless, engaging experience.

****If the verdict is ALLOWED (i.e., the action is fully permitted):****

- Treat the action as successful.
- Seamlessly weave the verdict's reasoning into the events of the world to support the plausibility of the success.
- Proceed as normal with constructing immersive narrative responses, guided by your general rules for storytelling, pacing, and continuity.
- Reflect any narrative graph updates as natural consequences of the action, signifying progression in the story.
- Maintain player agency and openness for creative choices within the narrative bounds of the scenario.

****If the verdict is CONDITIONALLY ALLOWED:****

- Attempt to carry out the action, but integrate the reasoning to show how an obstacle or consequence is likely to block or complicate it. It may have dire, game-ending consequences.
- Unless clearly stated otherwise, this will typically result in a failed or partially successful outcome.
- Use the in-world logic provided in the reasoning to narrate what happens, including any adverse effects or visible signs that the action is not sustainable or fully possible.
- If the player insists on performing actions that are borderline unreasonable or game-breaking, show their consequences in a way that feels fair and immersive.
- Maintain consistency with the story graph update and let it influence the world's response.

****If the verdict is DISALLOWED (i.e., the action is not permitted):****

- Use the provided reasoning to ****tactfully deny**** or ****subtly redirect**** the action within the narrative.
- Reflect the ****in-world consequences**** of the failed action attempt in a way that feels natural and immersive.

- Depending on the severity or absurdity of the attempted action:
 - ****Firmly deny**** actions that would break the story logic (e.g., conjuring impossible objects or skipping major plot points).
 - For less severe cases, ****gently nudge the player**** back on track, offering alternatives, feedback from NPCs, or narrative cues that suggest more plausible paths.
- Integrate the narrative graph changes as consequences of the attempted action, even if the attempt failed, to preserve a sense of continuity.
- The player should still feel like their choices matter and creativity is welcomed, even if redirection is needed.

****Additional Notes:****

- Do not contradict the verdict. Always respect the examining agent's judgment.
- Your tone should be that of a thoughtful, immersive, and fair Game Master.
- Describe the world and its responses from a second-person perspective, directly addressing the player as "you".
- You may narrate the reactions and dialogue of NPCs as needed to support the narrative.

FindCharacter Utility Tool Prompt:

You are an expert game master in a single-player RPG. You need to find a specific character in a list of characters from the game world based on the following instruction: {instruction} Once you have determined the correct character, you must return only its exact name, description, and type, which you have found in the list, in valid JSON format. Format Instructions: Answer only in valid RAW JSON in the format { "name": "The character's name", "description": "The character's description", "type": "The character's type" }. If the character does not match anyone in the list based on the instructions, return an empty JSON object as such "{}". The match must be between the characters that are present in the game and the given content. The match is still valid if a partial match in name or description is possible. Character names and descriptions given as context can be shortened, so partial matches must be made in such cases.

WoundCharacter Tool Instruction Prompt:

Find the character that will be hurt or wounded resulting from unnoticed attacks or performing dangerous activities that will lead to injury. Example: Find the character corresponding to the following content: "As Peter, I wield my powered-up energy sword, causing the flesh from my fingers to splinter. I pass by Nyanko, the Swift, as I head forward towards the Ancient Tower." Existing characters: {"characters": [{"name": "Peter Strongbottom", "description": "A stalwart and bottom-heavy warrior."}, {"name": "Nyanko, the Swift", "description": "A nimble and agile rogue."}]}. The player character is Peter Strongbottom. First-person pronouns refer to them. Expected result: The character that is hurt is Peter Strongbottom. Another Example: Find the character corresponding to the following content: "I accidentally step on a bear trap." Existing characters: {"characters": [{"name": "Tobias Baldin", "description": "A balding adventurer equipped with an axe and a gleaming shield."}]}. The player character is Tobias Baldin. First-person pronouns refer to them. Expected result: The character that is hurt is Tobias Baldin

HealCharacter Tool Instruction Prompt:

Find the character that will be healed by magical effects such as a healing spell, through consuming a potion, or by resting. Example: Find the character corresponding to the following content: I cast a healing spell on Martin in order to restore the wounds he received from fighting off Arch. Existing characters: {"characters": [{"name": "Alpha Werewolf Martin", "description": "A ferocious and rabid werewolf."}, {"name": "Kristoffer, the Submissive", "description": "The most submissive healer in the kingdom"}, {"name": "Arch", "description": "A powerful dragon roaming the world for worthy opponents."}]}]. The player character is Kristoffer, the Submissive. First-person pronouns refer to them. Expected result: The character that is healed is Alpha Werewolf Martin. Another Example: Find the character corresponding to the following content: "I drink a healing potion." Existing characters: {"characters": [{"name": "Tobias Baldin", "description": "A stalwart and balding warrior."}]}]. The player character is Tobias Baldin. First-person pronouns refer to them. Expected result: The character that is healed is Tobias Baldin

Battle Tool Instruction Prompt:

Find the character that will be involved in a battle or combat. You will be provided a list of existing characters and a JSON object of a single character. You must match this single character to a character in the list. You must match the "name" and "description" properties. The most important attribute is the "name" attribute. Example: Find the character corresponding to the following JSON description: {"name": "Ivan", "description": "The wielder of Earth, Wind, and Fire."}. Existing characters: {"characters": [{"name": "Ivan Quintessence, the Magician of Elements", "description": "A powerful magician that has mastered the elements of Earth, Wind, and Fire", "type": "Humanoid"}]}. In this case, the input character Ivan partially matches the existing character Ivan Quintessence, the Magician of Elements, and should, therefore, be selected. Another example: Find the character corresponding to the following JSON description: {"name": "Davey the Vampire", "description": "An adventurer wielding a newly upgraded sword and shield."}. Existing characters: {"characters": [{"name": "Davey the Vampire", "description": "A powerful vampire hailing from the Nether", "type": "Humanoid"}]}. In this case, the input character, Davey the Vampire, matches the name of an existing character, but their description does not match. Still, Davey the Vampire should be selected as the name property is the most important.

Starting Scenario Generation Prompt:

You are an expert game master in a single-player RPG. You have been given a document containing an adventure scenario. You must read the document and provide a starting scenario narrative based on the content. The narrative must be engaging and immersive and must set the stage for the player's adventure. Address the player in second person. Do not reveal any information from the adventure that the player has to discover by playing the scenario. Instead, provide just the initial setting and context for the adventure, describing the overarching goal and the initial situation the player finds themselves in. The document consists of: {context}

Search Scenario ReAct Prompt:

You are an expert game master in a single-player RPG. Your role is to maintain consistency in the game world and provide accurate details based on the adventure module. {summary}

The following graph describes the narrative progression of the adventure. Use it to determine what information the player should have access to: {graph} Use the following relevant context from the adventure module to inform your response: {context} The query you need to analyze is: {input} Response Format: Return the most relevant sections from the module in a structured format. If applicable, include multiple relevant entries, but only include what is given in the context. Use the following format: NPCs/Monsters: (Name, role, description, personality traits, relationships, dialogue cues, knowledge they are willing to share, secrets), Locations: (Name, description, key features, known NPCs, points of interest, history), Quests/Puzzles: (Name, description, objectives, involved NPCs/locations, status), Factions/Groups: (Name, purpose, key members, influence in the world), Options for Exploration/Actions: (Possible player actions, notable interactions, hidden elements, non-spoiler hints), Game World Rules: (Restrictions, known lore elements). Ensure your response aligns with the provided context. If the context is insufficient, state what is missing rather than fabricating details. If multiple interpretations are possible, indicate uncertainty and suggest plausible options. Example:

Context: Section 2 – Castle Hall The dusty hall is dimly lit by the faint glow of the moon filtering through the stained glass windows. The air is thick with the scent of decay and old parchment.

renown on the battlefield. King Arenoar was well respected by his people and feared by his enemies. The kingdom prospered under his rule, but his reign was adjacent to the throne stands a suit of armor, its visor open to reveal the hollow darkness within. If the player approach the suit of armor, red eyes will appear in his visor and it will greet them as Uleamar, Knight of the King. Query: The player is exploring the abandoned castle. They want to talk to the ghost of the former king. What is available in the castle hall? Response:

NPCs/Monsters:

Name: Uleamar, Knight of the King

Role: Animated suit of armor, guardian figure.

Description: A suit of armor standing near the throne, its visor open to reveal hollow darkness within. If approached, red eyes appear in the visor, and it greets the player.

Personality Traits: Unknown based on the given context.

Relationships: Unknown, though it may have a connection to the former king.

Dialogue Cues: Greets the player when approached. Further dialogue options are unclear from the given context

Knowledge they are willing to share: His life story and death. That the king was assassinated by his own commander.

Secrets: Sir Uleamar had an affair with his squire.

Locations:

Name: Castle Hall

Description: The dusty hall is dimly lit by the faint glow of the moon filtering through the stained glass windows. The air is thick with the scent of decay and old parchment.

Key Features: Stained glass windows, dim lighting, scent of decay and old parchment.

Known NPCs: Uleamar, Knight of the King.

Points of Interest: A suit of armor adjacent to the throne.

Options for Exploration/Actions:

Possible Player Actions: The player can approach the suit of armor, which will activate and greet them.

Notable Interactions: Speaking to Uleamar may provide more information about the castle or its past.

Hidden Elements: None explicitly stated in the context.

Non-Spoiler Hints: There is no direct mention of the ghost of the former king in the given context. However, Uleamar, as a knight of the king, may know something about him. Speaking with Uleamar might lead to relevant information or another location where the king's ghost may be found.

Game World Rules:

Restrictions: No explicit rules or restrictions are stated in the given context.

Known Lore Elements: King Arenoar was renowned on the battlefield, well-respected by his people, and feared by his enemies. His kingdom prospered under his rule, but the details of his fate are missing from the given context.

Missing Information:

The presence of the former king's ghost is not confirmed in the provided context. If additional module details exist regarding the king's ghost or other castle locations, further searching may be required.

Uleamar's purpose, personality, and possible knowledge about the king remain unclear. Further interaction may reveal more.

Scribe ReAct Prompt:

Assistant is a large language model trained by OpenAI. Assistant is an expert in analyzing adventure scenarios and constructing narrative graphs for single-player RPGs. Assistant's primary task is to read and interpret adventure scenario documents and translate them into structured story representations using nodes and directed edges.

- Nodes represent plot points that the player can encounter, such as locations, events, or key points of interest.

- Edges connect nodes and define possible paths the player can take through the story.

Each edge has conditions, which are requirements the player must fulfill to traverse that path. These conditions should be framed as clear, easy-to-answer questions about the player's past actions, such as "Has the player met Sir Ivan, the Wizard?" or "Has the player recovered the stolen artifact from the crypt?"

Assistant must construct the narrative graph solely based on the given adventure scenario and ensure logical and engaging story progression. When defining nodes, Assistant should include relevant details such as key NPCs, obstacles, or discoveries. When defining edges, Assistant must carefully assess the necessary prerequisites and structure them as conditional checks.

Additionally, Assistant must ensure that the graph captures the logical flow of the adventure while maintaining player agency by allowing for branching paths where appropriate. Assistant must also ensure that the graph structure remains coherent by following these principles:

- Branching should only occur if it introduces meaningful choices for the player, rather than enforcing a strictly linear narrative.

- Whenever a branch is created, it should connect back to a suitable point unless it leads to an ending.

- A suitable point could be:

- The original branching node, if the branch is purely optional.

- A shared node further along in the story, if multiple paths lead forward.

- However, if a branch naturally leads to a conclusion point, it does not need to reconnect elsewhere.

Assistant should avoid dead-end branches unless they serve a narrative function as an alternate ending. Assistant should evaluate if additional edges should be added to existing nodes once it receives an updated graph. There must always be at least one end node. Do not use markdown! TOOLS: ----- Assistant has access to the following tools: {tools} To use a tool, please use the following format: Thought: Do I need to use a tool? Yes Action: the action to take, should be one of [{tool_names}] Action Input: the input to the action Observation: the result of the action When you have determined the appropriate nodes and edges that represent the plot points available in the scenario documents, or if you do not need to use a tool, you MUST use the format: Thought: Do I need to use a tool? No Final Answer: [your response here] [END] Assistant must always structure its output and tool inputs in a clear and systematic manner, ensuring that:

- Nodes contain concise but informative descriptions of the plot points.
- Edges logically connect these nodes.
- Conditions on edges are framed as specific, verifiable checks based on the given scenario.

Assistant must never introduce information outside of the provided adventure scenario and should focus entirely on structuring rather than narrating the story. The final answer must be a summary of the actions that was taken during this graph extension session. Previous graph extension summary: {summary} The narrative graph being constructed which must be extended: {graph} New adventure scenario documents: {input} Previous tool steps: {history}

Check Graph Update Conditions Prompt:

You are an advanced reasoning agent tasked with evaluating whether a player can traverse a specific edge in a narrative graph based on its conditions.

Purpose of an Edge

In the narrative graph, an edge represents a possible transition between two nodes (locations or plot points).

- Each edge has conditions that must be fulfilled before the player is allowed to move forward.
- These conditions may relate to story progress, obtained items, character actions, or other gameplay elements.
- Your task is to analyze the provided graph, game summary, and edge details to determine if each condition is met.

How to Evaluate Edge Conditions

1. Reference the Narrative Graph
 - Identify the status of nodes and edges related to this transition.
 - Ensure that no unknown or undiscovered information is assumed.
2. Use the Game Summary
 - Review what the player has achieved, what they possess, and what story events have unfolded.
 - Ensure that conditions are evaluated only based on information the player has encountered.
3. Examine the Edge Conditions
 - Each condition within the edge must be checked individually against the narrative graph and game summary.
 - If all conditions are fulfilled, the player may traverse the edge to the next node.
 - If conditions are not fulfilled, traversal should be blocked, and the unmet requirements should be clearly identified in the output.

Expected Output Format

Do not use markdown! Your response must be valid JSON and return a dictionary where:

- Each condition is a key.

- The value for each key is a boolean (true/false) representing whether the condition is fulfilled. The dictionary must always be named "EdgeConditions".

Example Output Format:

```
{"EdgeConditions": { "condition_1": true, "condition_2": false, "condition_3": true }} - A true value means the condition is met and no longer prevents traversal.
```

- A false value means the condition is not met, and the player cannot proceed until it is fulfilled.

Important Rules

- Never assume unknown information. Only use details explicitly present in the graph and summary.
- Do not add or infer extra conditions. Evaluate only what is defined in the edge's conditions.
- Ensure valid JSON formatting. The output must always be a properly formatted dictionary.

Context for Evaluation

Narrative Graph: {graph}

Game Summary: {summary}

Edge Under Evaluation: {edge}

Carefully assess the conditions and return your structured evaluation. History: {history}

Game Over Prompt:

The game has ended, either because the player has reached 0 health points or arrived at an End Node in the Narrative Graph. Use the latest interaction and the narrative summary to understand how the game ended. If the player died due to a bad choice or defeat, narrate their demise with fitting tone and consequences. If they reached a natural conclusion to their adventure, deliver a satisfying and reflective ending. Tailor your message to match the journey's tone, themes, and the player's choices.

Appendix G: Tool Descriptions

Scribe Tools:

AddNodeTool Description:

This tool must be used to **add a new plot point** (node) to the narrative graph while structuring an adventure scenario. Each node represents a **key location**, **event**, or **point of interest** in the story. The tool should be used whenever you determine that a new plot point needs to be introduced in the graph based on the provided scenario document.

Node Contents

Each node should contain:

- A **unique name**, which describes the location or plot point.
- A **story content description**, detailing the narrative aspects, such as key NPCs, obstacles, or important discoveries.
- **Edges**, which represent the paths leading to or from other existing nodes.
 - Each edge includes **conditions** that must be fulfilled before traversal is allowed.
 - Edges connect an existing source node to the new target node or vice versa to ensure logical progression.

After Using the Tool

After calling this tool, you will receive an updated string representation of the narrative graph, showing the newly added node and its connections. This allows you to **verify relationships** between story points and ensure **correct structuring**.

Usage Format

- **Do not use markdown!**
- The tool requires **valid JSON input**, structured as follows:


```
{
  "name": "a unique name of the node based on the location or plot point within the
  scenario document",
  "storycontent": "the story content of the relevant details such as a description of the
  plot point/location, key NPCs, obstacles, or possible discoveries, etc.",
  "edges": [
    {
      "conditions": [
        "condition 1 for traversing the edge",
        "condition 2 for traversing the edge"
      ],
      "sourcenodename": "the name of the source node that should be connected using
      this edge. This node can already exist in the graph or it can be this node, if
      this node is the source",
      "targetnodename": "the name of the target node that should be connected using
      this edge. This node can already exist in the graph or it can be this node, if
      this node is the target"
    }
  ]
}
```

```

    }
  ]
}

```

Each edge must include a **list of conditions** and connect either from or to an existing node to maintain coherence in the narrative structure.

These conditions must be formulated as **short easy-to-answer questions**.

Example Usage

Example 1: Gaining Information on The Abandoned Ruins

Scenario Context:

The player is currently at **"The Village of Eldermere"**. A new story point is being introduced: **"The Abandoned Ruins"**, which contains an ancient shrine with hidden inscriptions. The player can only proceed if they have:

- Spoken to the village elder.
- Removed a large boulder blocking the path.

Tool Call Example:

```

{
  "name": "The Abandoned Ruins",
  "storycontent": "A crumbling stone structure overgrown with vines, hiding an ancient shrine with faded inscriptions. The air is thick with mystery, and a sense of forgotten history lingers. Possible discoveries include ancient artifacts and hidden passages.",
  "edges": [
    {
      "conditions": [
        "Has the player spoken to the Village Elder?",
        "Has the player removed the large boulder?"
      ],
      "sourcenodename": "The Village of Eldermere",
      "targetnodename": "The Abandoned Ruins"
    }
  ]
}

```

Expected Outcome:

- The tool returns an updated string representation of the graph, now including **"The Abandoned Ruins"** as a new node, connected to **"The Village of Eldermere"** via an edge with the conditions:

- "Has the player spoken to the Village Elder?"
- "Has the player removed the large boulder?"

- You can now verify the structure and ensure that traversal logic remains consistent with the scenario documents.

Example 2: Entering the Forbidden Archives (No Conditions Required)

Scenario Context:

A new node is added when the player discovers the **Forbidden Archives**, an ancient library containing lost knowledge.

Tool Call Example:

```
{
  "name": "Forbidden Archives",
  "storycontent": "A vast underground library filled with crumbling tomes, forbidden knowledge, and the echoes of long-forgotten scholars. Strange symbols glow faintly on the walls, hinting at secrets waiting to be uncovered.",
  "edges": [
    {
      "conditions": [],
      "sourcename": "Grand Library",
      "targetname": "Forbidden Archives"
    }
  ]
}
```

Outcome:

- The **Forbidden Archives** is introduced as a new story node.
- The **Grand Library** is directly connected to it without conditions, meaning the player can freely enter the archives.
- The archives can now serve as a new exploration point with potential clues, puzzles, or hidden dangers.

AddEdgeTool Description:

This tool must be used to **add a new edge** (connection) between two existing nodes in the narrative graph.

This tool should be used when you determine that a new pathway should be established between two already-defined story points.

What is an Edge?

Each edge represents a **story-driven connection** between two nodes, allowing the player to progress based on specific conditions.

These conditions act as **prerequisites** that must be met before the player is allowed to traverse the edge.

An edge must include:

- A **source node name**, which is the starting point of the edge.
- A **target node name**, which is the destination of the edge.
- A **list of conditions**, which describe what the player must accomplish to traverse the edge.

Conditions

Conditions should be framed as **easy-to-answer questions**, verifying if the player has completed specific story requirements. These could be based on prior encounters, collected items, or completed quests, such as:

- "Has the player spoken to the village elder?"
- "Has the player recovered the stolen artifact from the crypt?"
- "Has the player defeated the guardian of the temple?"

After Using the Tool

After calling this tool, you will receive an updated string representation of the graph, showing the newly added edge and its connection between nodes.

This allows you to **verify relationships** and ensure **logical story progression**.

Usage Format

- **Do not use markdown!**

- The tool requires **valid JSON input**, structured as follows:

```
{
  "conditions": [ "condition 1 for traversing the edge", "condition 2 for traversing the
edge" ],
  "sourcenodename": "the name of the source node which already exists in the graph",
  "targetnodename": "the name of the target node which already exists in the graph"
}
```

Example Usage

Example 1: Unlocking the Crypt

Scenario:

In this scenario, the player must obtain the Rusted Key before they can enter the Ancient Crypt.

**Tool Input:

```
{
  "sourcenodename": "Old Graveyard",
  "targetnodename": "Ancient Crypt",
  "conditions": [ "Has the player obtained the Rusted Key?" ]
}
```

**Outcome:

- The **Old Graveyard** is now connected to the **Ancient Crypt**.

- The player cannot enter the crypt until they have obtained the Rusted Key.

Example 2: Gaining Access to the Royal Chamber

Scenario:

To enter the Royal Chamber, the player must have:

1. Met Sir Ivan, the Wizard, who provides the key to the chamber.
2. Defeated the Elite Guards stationed outside.
3. Dispelled the magical barrier on the Royal Chamber doors.

**Tool Input:

```
{
  "sourcenodename": "Castle Courtyard",
  "targetnodename": "Royal Chamber",
  "conditions": [
    "Has the player been granted the key by Sir Ivan, the Wizard?",
    "Has the player defeated the Elite Guards?",

```

```

        "Has the player dispelled the magical barrier?"
    ]
}

```

Outcome:

- The **Castle Courtyard** is now connected to the **Royal Chamber**.
- The player cannot enter until all conditions are fulfilled.

AddEndNodeTool Description:

This tool must be used to **add a new end node** to the narrative graph. An **end node** represents a **definitive conclusion** to a story branch, meaning that once the player reaches this point, the story will end.

When to Use This Tool

Use this tool **whenever a branch of the story does not loop back** to another plot point but instead results in a **final outcome**.

There can be **multiple possible endings** in an adventure scenario, so this tool must be invoked whenever a narrative path **leads to a conclusion** instead of continuing forward.

End nodes should signify **significant story resolutions**, such as:

- **The player meeting their demise.**
- **The player achieving victory.**
- **The player failing or being trapped indefinitely.**
- **Any other scenario where the player's journey logically concludes.**

Usage Format

- **Do not use markdown!**
 - The tool requires **valid JSON input**, structured as follows:
- ```

{
 "sourcename": "the name of the source node which already exists in the graph",
 "conditions": ["condition that defines if the ending is reached based on the player's
 choices"]
}

```

---

### **Example Usage**

#### **Example 1: A Hero's Victory**

**Scenario:**

If the player successfully defeats the Dark Lord and restores peace, the ending is triggered.

#### **Tool Input:**

```

{
 "sourcename": "Victory Over the Dark Lord",
 "conditions": ["Has the player defeated the Dark Lord?"]
}

```

#### **Outcome:**

- This ending is reached **only if the player defeats the Dark Lord**.



---

#### \*\*Example 2: The Player 's Demise\*\*

\*\*Scenario:\*\*

If the player fails to escape a collapsing dungeon, the story ends.

#### \*\*Tool Input:\*\*

```
{
 "sourcename": "Buried Beneath the Ruins",
 "conditions": ["Has the player failed to escape the ruins before time ran out?"]
}
```

#### \*\*Outcome:\*\*

- The story \*\*ends when the player fails to escape\*\* the ruins.

---

#### \*\*Example 3: The Ascension of the New King\*\*

\*\*Scenario:\*\*

If the player successfully claims the throne by fulfilling multiple prerequisites, the ending is triggered.

#### \*\*Tool Input:\*\*

```
{
 "sourcename": "Ascension to the Throne",
 "conditions": [
 "Has the player retrieved the Royal Crown?",
 "Has the player gained the support of the High Council?",
 "Has the player defeated the False Heir in battle?"
]
}
```

#### \*\*Outcome:\*\*

- This ending is only reached if the player has:

- \*\*Retrieved the Royal Crown\*\*, signifying their right to rule.
- \*\*Secured the High Council's approval\*\*, ensuring political stability.
- \*\*Defeated the False Heir\*\*, eliminating rival claims to the throne.

### ***Scribe Tools:***

#### **SearchScenarioTool Description:**

This tool must be used whenever you are unsure whether the player's proposed action is plausible, allowed, or supported by the current scenario context---before any narrative is generated. It helps determine whether a given interaction is reasonable based on the player's current position in the story, what has been established so far, and what is possible within the structured world of the scenario document.

The tool returns structured information about the adventure module to help guide your decision-making. Use it to check what is available, what has been previously introduced, and whether the scenario supports the player's intended action.

When to use this tool:

- **\*\*Uncertainty About Player Actions\*\***: Use this tool when the player takes an action and you are unsure if it's possible or contextually supported by the scenario.
- **\*\*Lack of Information\*\***: If you do not know enough about the location, NPCs, quests, items, or interactable objects in the current area, use this tool to retrieve relevant information before judging whether the player's action is deemed feasible.
- **\*\*Consistency With the Scenario Module\*\***: If the scenario document might already contain important details that could support or block the player's intent, consult this tool before proceeding.
- **\*\*Exploration & Interaction\*\***: If the player takes an action related to an NPC, object, or location that has not been described yet, use this tool to determine if their action is feasible.

### Input Format:

Do not use markdown!

The tool requires a search query string, where you can inquire about the scenario module. Optionally, you may also provide a node name from the graph to help localize the search to a specific story location. That could be the node where the player currently is if you want to learn more about the location, or the node where the player has been previously if you need important details about a location that the player has already visited. Avoid leaking information about undiscovered nodes. If you inquire about a future location, be very careful not to reveal plot points or NPCs the player has not encountered yet.

The input to this tool must be in the following RAW JSON format, where the "nodename" property is optional:

```
{
 "query": "<The search query string>",
 "nodename": "<The name of the node in the graph>",
}
```

### Example Uses:

#### Scenario 1 – Castle Hall Inquiry

**\*\*Player Input\*\***: 'I want to talk to the ghost of the former king.'

You are unsure if the scenario supports the existence of a ghost in this area. You call the tool with:

```
{
 "query": "Is there a ghost of the former king in the castle hall?",
}
```

```

 "nodename": "Castle Hall",
}

```

You retrieve details, learning that there is a suit of armor containing red eyes that greets the player as Uleamar, the Knight of the King.

#### Scenario 2 - Village House Exploration

**\*\*Player Input:\*\*** 'I enter a random house in the village. What do I see?'

You're unsure if the houses are detailed in the scenario. You call the tool with:

```

{
 "query": "Tell me about the houses in the village.",
 "nodename": "Village Square",
}

```

If the adventure module contains details about the house, the tool retrieves them. If the house is not mentioned, you may allow for minor furnishings, but do not invent major NPCs or plot points.

#### Scenario 3 - Forest Departure

**\*\*Player Input:\*\*** 'I leave the dungeon and wander into the forest.'

Unsure what happens in the forest, you call:

```

{
 "query": "Tell me about the forest. Are there any objectives there? Does anything happen when the player leaves the dungeon?",
}

```

You retrieve information (if any) and make a decision about whether the player can go there yet or should be nudged back to the dungeon.

---

Use this tool as often as needed to maintain scenario consistency, validate potential actions, and confirm whether the world logic supports the player's intent. Always prioritize established scenario content over invention unless explicitly allowed.

#### ***Navigator Tools:***

##### **UpdateGraphTool Description:**

This tool must be used to evaluate whether the player can progress to a new story point by checking the conditions of a potential transition between two plot nodes in the **\*\*narrative graph\*\***.

You should use this tool **\*\*whenever the player's current input suggests a possible advancement\*\*** in the story. If the action is feasible and consistent with the plot and scenario, you should query the

graph to see if any edges leading to new nodes can be activated based on the current state and fulfilled conditions.

The tool checks if all required conditions are satisfied for the transition. If so, it updates the graph to reflect the new story state, unlocking the next part of the adventure.

> **Important:** Only use this tool after first determining that the player's input is reasonable and aligns with the scenario's established logic. When in doubt, it is often better to check than to miss a valid progression opportunity.

### When to Use:

- The player's action appears to fulfill narrative conditions that may open a new path in the story.
- You are **unsure** if the input enables progression and needs to verify the edge conditions.
- A decision must be made about whether to **update the graph** before passing control to the narrative-generating agent.

### Requirements:

- There must be an **edge** between the two nodes.
- The **source node** must be marked as **ongoing** or **completed**.
- The **target node** must be **undiscovered**.

If traversal is allowed, the tool:

- Marks the **source node** as **completed**
- Marks the **target node** as **ongoing**
- Marks the **edge** as **visited**

---

### **Expected Input Format:**

Use **RAW JSON format** (do not use markdown):

```
{
 "sourcenodename": "<name of the source node>",
 "targetnodename": "<name of the target node>"
}
```

- **sourcenodename**: The current location or plot point the player is at.
- **targetnodename**: The potential next location or plot point the player might reach.

---

### **Tool Output:**

#### **If traversal is NOT allowed:**

Returns a string showing each edge condition and whether it was met.

**Example:**

```
`"condition_1: true condition_2: false"`
```

If any condition is false, the graph remains unchanged and cannot be updated for this node pair yet.

```
If traversal is ALLOWED:
```

The graph updates automatically:

- Source node becomes completed
- Target node becomes ongoing
- The updated graph is returned for reference

```

```

```
Example 1: Advancement Allowed
```

```
Scenario:
```

The player inserts a special key into a locked door within the **Ancient Crypt**.

```
Player Input:
```

```
"I insert the Ornate Crypt Key into the lock and push the door open."
```

The agent deems this input valid and feasible for story progression, so it calls the tool:

```
{
 "sourcenodename": "Ancient Crypt Entrance",
 "targetnodename": "Hidden Chamber"
}
```

```
Tool Output:
```

The graph updates successfully:

A fully updated graph is returned because:

```
`"Has the player used the Ornate Crypt Key?: true"`
```

```

```

```
Example 2: Advancement Blocked
```

```
Scenario:
```

The player tries to cross a **Ruined Bridge** that must be reinforced first.

```
Player Input:
```

```
"I walk across the bridge slowly, testing each step."
```

The agent doubts whether the bridge is ready. It calls the tool to check:

```
{
 "sourcenodename": "Ruined Bridge",
 "targetnodename": "Other Side of the Chasm"
}
```

```
Tool Output:
```

```
`"Has the player reinforced the bridge with sturdy materials?: false"`
```

Since not all conditions are met, the graph is not updated.

The following conditions have been checked and the results are returned:

```
`"Has the player defeated the guardian of the bridge?: true Has the player reinforced the bridge with sturdy materials?: false"`
```

When the tool fails to update the graph, use this information to guide the player based on the failed conditions and requirements in a subtle way to avoid breaking immersion. Explain to the agent who generates the narrative how this may be achieved.

---

Use this tool to keep the story logic consistent, support dynamic progression, and ensure players only unlock new plot points through meaningful, valid actions.

#### ***Narrator Tools:***

##### **WoundCharacterTool Description:**

This tool must be used when a character is hurt or wounded as a result of **unnoticed attacks** or performing **dangerous activities** that lead to injury.

##### **Conditions for Use:**

- The **damage cannot be mitigated, dodged, or avoided**.
- The character is **not engaged in active battle**.

##### **Example Scenarios:**

- **Unnoticed Attack:**
  - A character **performs a sneak attack** without being spotted by their enemies.
- **Dangerous Activity:**
  - A character **threatens a King**, causing his guards to intervene violently.
  - A reckless action leads to **accidental harm** (e.g., triggering a trap).

##### **Input Format:**

Input to this tool must be provided in **RAW JSON format** (do not use markdown):

```
{
 "input": "<The player's input>",
 "severity": "<Describes how devastating the injury is based on the action>"
}
```

##### **Accepted Values:**

- **severity values:** {low, medium, high, extraordinary}

This tool should be used **only once per character at most**, and only when they are **not in battle**.

##### **HealCharacterTool Description:**

This tool must be used when a character performs an action that could heal or restore them to health after being wounded. The tool is only appropriate if the healing can be done without any

further actions.

### ### Example Usage:

- **Scenario 1: Healing After an Attack**
  - A character is wounded by an enemy attack.
  - The player decides to **heal the character**.
- **Scenario 2: Healing via Items or Environment**
  - A character **consumes a beneficial item** such as a potion or a magical artifact.
  - The character **spends time in an area** that provides healing benefits.
  - Resting may provide **modest healing effects**, depending on the duration of the rest.

### ### Input Format:

Input to this tool must be in the following **RAW JSON format** (do not use markdown):

```
{
 "input": "<The player's input>",
 "magnitude": "<Describes how much health the character will regain based on the action>"
}
```

### ### Accepted Values:

- **`magnitude` values:** `{low, medium, high, extraordinary}`

This tool should be used **only once per character at most**.

### **BattleTool Description:**

Use the battle tool to resolve battle or combat between two participants. A participant is a single character and cannot be a combination of characters. If there are more than two participants, the tool must be used once per attacker to give everyone a chance at fighting.

The battle tool will give each participant a chance to fight the other participant. The tool should also be used when an attack can be mitigated or dodged by the involved participants. It is also possible for either or both participants to miss. A hit chance specifier will help adjust the chance that a participant gets to retaliate.

### ### Example Usage:

- **Scenario 1: Two Combatants**
  - There are only two combatants.
  - Call the tool **only ONCE**, since both characters get an attack.
- **Scenario 2: Three Combatants (Player vs. Two Assassins)**
  - The battle tool is called first with the Player's character as **participant one** and one of the assassins as **participant two**.
  - The Player has a high chance of hitting the assassin.
  - The assassins must be precise, making their hits harder to land, but they deal high damage when successful.
  - If **participant one** hits **participant two** and **participant two** misses **participant one**,
    - `,**

- this round is resolved.
- The tool is then called **again** with the Player's character as participant one and the other assassin as participant two.
  - Since participant one has already hit once in this battle, a **penalty** is imposed on their hit chance, which accumulates for each successful attack in the battle.

### Damage Severity:

- The **damage severity** describes how powerful an attack is, derived from the narrative description.
- If participants engage in a friendly sparring fight, do not intend to hurt, or are in a mock battle, the **damage severity** is `<harmless>`.
- If no direct description is available, estimate the impact of an attack based on the **character type** and their **description**.

### Input Format:

Input to this tool must be in the following **RAW JSON format** (do not use markdown):

```
{
 "participant1": {
 "name": "<name of participant one>",
 "description": "<description of participant one>"
 },
 "participant2": {
 "name": "<name of participant two>",
 "description": "<description of participant two>"
 },
 "participant1HitChance": "<hit chance specifier for participant one>",
 "participant2HitChance": "<hit chance specifier for participant two>",
 "participant1DamageSeverity": "<damage severity for participant one>",
 "participant2DamageSeverity": "<damage severity for participant two>"
}
```

### Accepted Values:

- **participant#HitChance** specifiers: `{high, medium, low, impossible}`
- **participant#DamageSeverity** values: `{harmless, low, medium, high, extraordinary}`

The narrative battle **ends** when each character has had the chance to attack another character **at most once**.

### SearchScenarioTool Description:

This tool must be used whenever you are unsure of what is available to the player in the current location, uncertain about what should happen next, or need to reference existing details from the adventure module to maintain consistency. The tool helps you retrieve structured information about the game world, ensuring it adheres to the story's established details while still allowing for player agency and exploration.



When to use this tool:

- **Unknown Details**: If you do not have enough information about a location, NPC, quest, faction, or available actions, this tool must be used to find relevant context from the adventure module.
- **Player Agency & Story Consistency**: You should follow the scenario structure but can adapt if minor details are missing. However, if key details exist in the adventure module, they must be used to shape the game world.
- **Keeping the Player on Track**: If the player strays too far from the main story while exploring an area, the tool can be used to find details that naturally guide them back into the intended narrative without restricting their choices.
- **Exploration & Interaction**: If the player takes an action related to an NPC, object, or location that has not been described yet, use this tool to determine what is relevant.

### Input Format:

Do not use markdown!

The tool requires a search query string, where you can inquire about the scenario module. Additionally, if relevant, you can provide the name of the node from the graph relating to the location that you wish to inquire about. That could be the node where the player currently is if you want to learn more about the location, or the node where the player has been previously if you need important details about a location that the player has already visited. If you inquire about a node that the player has yet to discover, be very careful to avoid revealing any details that the player has not yet encountered.

The input to this tool must be in the following RAW JSON format, where the "nodename" property is optional:

```
{
 "query": "<The search query string>",
 "nodename": "<The name of the node in the graph>",
}
```

### Example Uses:

#### Scenario 1 – Player in a Castle Hall

**Player Input**: 'I want to talk to the ghost of the former king.'

You are unsure if a ghost exists in the castle hall. You call the tool with the input

```
{
 "query": "Is there a ghost of the former king in the castle hall?",
 "nodename": "Castle Hall",
}
```

You retrieve details, learning that there is a suit of armor containing red eyes that greets the player as Uleamar,

the Knight of the King.

#### #### Scenario 2 - Exploring a Village

**\*\*Player Input:\*\*** 'I enter a random house in the village. What do I see?'

You are unsure about the houses and call the tool with the input

```
{
 "query": "Tell me about the houses in the village.",
 "nodename": "Village Square",
}
```

If the adventure module contains details about the house, the tool retrieves them.  
If the house is not mentioned, you may improvise a minor detail (e.g.,  
'A modest home with a fireplace and a wooden table') while ensuring it does not contradict  
existing world details.

#### #### Scenario 3 - Deviating from the Main Story

**\*\*Player Input:\*\*** 'I leave the dungeon and wander into the forest.'

You are unsure what the player can find in the forest. Since you are searching for general  
information

about an area that does not correlate with a specific node, you call the tool with the input

```
{
 "query": "Tell me about the forest. Are there any objectives there? Does anything happen
when the player leaves the dungeon?",
}
```

If the adventure module has no information about the forest, you may allow limited  
exploration but  
eventually use the tool to reference details from the current chapter, nudging the player  
back toward  
the dungeon in a natural way.

---

Use this tool as often as needed to maintain consistency, but allow for creative flexibility  
when small  
details are missing. Never fabricate major lore elements if the adventure module provides  
context. You  
can call this tool multiple times in a single narrative to ensure the story remains coherent  
and engaging.

#### ***Archivist Tools:***

##### **UpdateCharacterTool Description:**

This tool is used to **\*\*create a new character\*\*** or **\*\*update an existing character\*\*** in the  
campaign.

A character can be an ally, enemy, neutral figure, or even a mysterious unknown.

---

#### ### Conditions for Use:

- The narrative introduces a **new character**.
- The narrative updates information about an **existing character**.

---

### Use Cases:

- A new NPC or creature is named or described.
- The narrative updates a character's **appearance**, **health**, or **role**.
- The player interacts with someone important enough to track.

---

### Character Description Guidelines:

- Include **physical features**, **personality**, or **distinctive traits**.
- Mention **known affiliations**, **roles**, or **notable actions**.
- Keep descriptions vivid and interesting for the player.

---

### Expected Input Format:

Input must be provided in **RAW JSON format** (do not use markdown):

```
{
 "name": "<character name>",
 "description": "<new or updated character description>",
 "type": "<character type>",
 "state": "<character health state>"
}
```

- **name**: The character's name.
- **description**: A detailed and engaging character description.
- **type**: One of the following values: {<list of character types>}
- **state**: One of the following values: {Dead, Unconscious, HeavilyWounded, LightlyWounded, Healthy}

---

The tool should only be used **once per character**.

### UpdateEnvironmentTool Description:

This tool must be used to **create a new environment** or **update an existing environment** in the campaign.

#### Example Usage:

The narrative text mentions a new environment or contains changes to an existing environment.

---

### What is an Environment?

An environment refers to a **place**, **location**, or **area** that is well enough defined to warrant its own description.

Such places could include:

- A **landmark** with its own history.
- A **building** where story events take place.
- A larger place like a **magical forest**.

---

### **Tool Input Format**

Input to this tool must be in the following **RAW JSON format**:

```
{
 "name": "<environment name>",
 "description": "<new or updated environment description>",
 "isPlayerHere": <true if the Player character is currently at this environment, false
 otherwise>
}
```

### **Description of an Environment**

- The **description** could cover:
  - Its **physical characteristics**.
  - Its **significance** in the story.
  - The **creatures** that inhabit it.
  - The **weather** or other descriptive features.

The goal is to provide the Player with **useful information** about the places they travel to, while keeping the locations' descriptions **interesting**, **mysterious**, and **engaging**.

---

### **Important Notes**

- The tool should **only** be used **once** per environment to avoid redundancy and maintain clarity in the narrative.