

Design Techniques for LLM-Powered Interactive Storytelling: A Case Study of the Dramamancer System

Tiffany Wang Yuqian Sun Yi Wang

Melissa Roemmele John Joon Young Chung Max Kreminski

Midjourney

{twang, ysun, ywang, mroemmele, jchung, mkreminski}@midjourney.com

1 Introduction

Interactive narrative is a game-like storytelling medium where stories envisioned by authors are realized dynamically through player choices. Traditionally, this medium has posed a challenge to authors in requiring them to anticipate the impact of many possible player choices on the resulting story. The rise of Large Language Models (LLMs) as powerful text generators has suggested a new paradigm for interactive narrative, which imagines that LLMs will eliminate authorial burden by readily instantiating story text that accommodates player agency within narrative boundaries defined by the author (e.g. Lu et al., 2025; Zhu et al., 2023). However, open questions remain about how to best leverage LLMs for this objective. In particular, as LLMs are increasingly noted for their story generation capabilities (e.g. Huot et al., 2025; Xie and Riedl, 2024; Yuan et al., 2022), what should the author, player, and LLM each be expected to contribute to the emerging story? To consider this, we look at the example of Dramamancer, a system that utilizes an LLM to transform author-created story schemas into interactive playthroughs driven by player input (Sun et al., 2025; Wang et al., 2025). We outline some design techniques exemplified by Dramamancer and identify some questions about how to evaluate the stories that emerge from it.

2 User Interface Design

The Dramamancer system involves two distinct user roles: an **author** who creates stories and a **player** who experiences them. In particular, the author establishes a **story schema**, and the player engages with an interactive **playthrough** that dynamically realizes this schema as a concrete story instance based on the player’s input. Below, we outline the interface for each of these user roles in terms of what is provided to the system.

Author Interface A **story schema** includes:

- **Style:** The style consists of instructions for the structure and presentation format of the playthrough (e.g. “*Write in Early Modern English with flowery metaphors*”).
- **Characters:** Each character is defined by a name and description (e.g., “*Sam: a young, novice reporter covering her first superhero competition*”). The first character in the list is designated as the **player character** represented by the player’s input during the playthrough.
- **Scenes:** Playthroughs are divided into one or more scenes. The author specifies the name of the scene, the characters present (a subset of the above list), a **setting** establishing the context (e.g. “*competition arena where superheroes face off*”), an **opening line** that is always presented verbatim to the player at the onset (e.g. “*you walk into the reporter’s corner on your first day — the superhero showdown is about to begin, and you want the story.*”), and one or more **events**.
- **Events:** Within a particular scene, **events** define what should happen in the story according to the player’s input. Events are structured as **storylets** (Kreminski and Wardrip-Fruin, 2018), each of which consists of a **condition** and an **outcome**. A condition is a statement that evaluates to true/false based on the playthrough state (e.g. “*Sam (player) asks what’s going on*”). A condition can be null and instead set to trigger after a certain number of lines have been generated in the scene. An outcome is a description of what should happen when the condition is satisfied (e.g. “*veteran reporter explains the competition while heroes argue on stage*”). An outcome can also be configured to end the scene, and additionally transition to a different scene.

Player Interface A **playthrough** unfolds line by line. A single **line** consists of actions and dialogue for one character, which can be accompanied by

third-person narration (e.g. “**The challenge has begun.** *Maria: (whispering) Which one catches your eye?*”) After certain lines, the playthrough pauses to elicit input from the player. They specify their input for the player character in the format of *(actions) dialogue*. The playthrough for a given scene continues as long as there is no triggered outcome indicating that the scene should end. Once the scene ends, the playthrough proceeds to the next scene if a transition is specified, otherwise the entire playthrough ends.

3 System Design

A Dramamancer playthrough is orchestrated by two LLM-based modules. The **instantiation** module produces the next line of the playthrough, while the **interpretation** module determines if the current playthrough leads to any of the event conditions being satisfied. Each of these modules is implemented with a single LLM interaction. Below we describe the design features of the prompts for these interactions, which rely on LLMs’ general ability to perform tasks in response to instructions and demonstrative examples.

Instantiation The instantiation prompt facilitates the LLM to generate the next line based on all previously generated lines in the playthrough, conditioned on the style, scene characters, and scene setting. The prompt specifies some of the following expectations for what the LLM will return:

- The output consists of the line shown to the player as well also other information used to configure the playthrough. This includes a *pause* variable specifying (as a boolean true/false) if the playthrough should pause after that line in order to elicit input from the player. This is expected to be true when there has been meaningful progression in the playthrough since the last time the player contributed input.
- In generating a line, the LLM implicitly decides which character is contributing that line. The line should only pertain to non-player characters, since actions/dialogue for the player character should come only from the player input.
- Regarding events whose conditions have been satisfied, the prompt lists the outcomes of these events, and instructs the LLM to incorporate them organically into the playthrough. This influences the next line to address outcomes that are not yet reflected in the previously generated lines, while also allowing for the possibility of

using multiple lines to convey an outcome.

- The line should be highly responsive to the most recent input from the player character, regardless of whether the player’s input causes any event conditions to be satisfied.

Interpretation The interpretation module runs after every player input. The prompt facilitates the LLM to determine if a current playthrough satisfies any of the conditions for events in the ongoing scene. The LLM returns a list of satisfied conditions, which enables the outcomes for these events to become available to the instantiation module.

4 Evaluation Considerations

The quality of a Dramamancer playthrough is mediated by the performance of the above LLM modules. There are two dimensions to evaluating this quality, corresponding to the perspectives of the author and the player.

Author Perspective Author-side evaluation primarily concerns the alignment between the story schema and the playthrough. Some of the specific variables relevant to assessing this are **style adherence** (does the playthrough consistently follow the style?), **character distinctiveness** (does the action/dialogue attributed to each non-player character reflect how they are described?), **scene awareness** (within a given scene, does the playthrough accurately convey the setting?), **event detection accuracy** (do the scene events get triggered appropriately based on player input?), and **outcome realization** (are event outcomes expressed suitably based on how they are described?).

Player Perspective Variables relevant to player-side evaluation include **responsiveness** (does the player feel like their input is directly influencing the playthrough?), **timing** (does the playthrough elicit input from the player at the times it is most impactful?), **reflection** (does the player feel like they are making meaningful decisions when providing input?), and **engagement** (does the playthrough motivate the player to stay engaged with the story?).

5 Summary

We examine the Dramamancer system as an example of how to utilize LLMs for bridging authorial intent and player agency in interactive narrative. In presenting this case study, we hope to engage further discussion of how evolving LLM capabilities can shape the future of storytelling.

References

- Fantine Huot, Reinald Kim Amplayo, Jennimaria Palomaki, Alice Shoshana Jakobovits, Elizabeth Clark, and Mirella Lapata. 2025. Agents’ room: Narrative generation through multi-step collaboration. In *The Thirteenth International Conference on Learning Representations*.
- Max Kreminski and Noah Wardrip-Fruin. 2018. Sketching a map of the storylets design space. In *Interactive Storytelling: 11th International Conference on Interactive Digital Storytelling, ICIDS 2018, Dublin, Ireland, December 5–8, 2018, Proceedings*, page 160–164, Berlin, Heidelberg. Springer-Verlag.
- Zhuoran Lu, Qian Zhou, and Yi Wang. 2025. Whatelse: Shaping narrative spaces at configurable level of abstraction for ai-bridged interactive storytelling. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI ’25, New York, NY, USA. Association for Computing Machinery.
- Yuqian Sun, Phoebe J Wang, John Joon Young Chung, Melissa Roemmele, Taewook Kim, and Max Kreminski. 2025. Drama llama: An llm-powered storylets framework for authorable responsiveness in interactive narrative. *arXiv preprint arXiv:2501.09099*.
- Tiffany Wang, Yuqian Sun, Yi Wang, Melissa Roemmele, John Joon Young Chung, and Max Kreminski. 2025. Dramamancer: Interactive narratives with large language model-powered storylets. UIST ’25, New York, NY, USA. Association for Computing Machinery.
- Kaige Xie and Mark Riedl. 2024. Creating suspenseful stories: Iterative planning with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2391–2407, St. Julian’s, Malta. Association for Computational Linguistics.
- Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story writing with large language models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, IUI ’22, page 841–852, New York, NY, USA. Association for Computing Machinery.
- 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):380–390.