Proceedings of the Twentieth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2024)

NarrativeGenie: Generating Narrative Beats and Dynamic Storytelling with Large Language Models

Vikram Kumaran, Jonathan Rowe, James Lester
North Carolina State University
{vkumara, jprowe, lester}@ncsu.edu

Abstract

Interactive narrative in games utilize a combination of dynamic adaptability and predefined story elements to support player agency and enhance player engagement. However, crafting such narratives requires significant manual authoring and coding effort to translate scripts to playable game levels. Advances in pretrained large language models (LLMs) have introduced the opportunity to procedurally generate narratives. This paper presents NarrativeGenie, a framework to generate narrative beats as a cohesive, partially ordered sequence of events that shapes narrative progressions from brief natural language instructions. By leveraging LLMs for reasoning and generation, NarrativeGenie, translates a designer's story overview into a partially ordered event graph to enable player-driven narrative beat sequencing. Our findings indicate that NarrativeGenie can provide an easy and effective way for designers to generate an interactive game episode with narrative events that align with the intended story arc while at the same time granting players agency in their game experience. We extend our framework to dynamically direct the narrative flow by adapting real-time narrative interactions based on the current game state and player actions. Results demonstrate that NarrativeGenie generates narratives that are coherent and aligned with the designer's vision.

Introduction

Interactive narrative has been integral to games since the advent of "choose your path" text adventures (Crowther, Woods, and Black 1977; Koenitz 2023) to provide players with agency and engagement. Interactive narratives have played a significant role in entertainment and educational games (Naul and Liu 2020; Koenitz 2023). Beyond manual authorship, planning-based methods used to procedurally generate these games focused on coherent event sequences, often failing to produce engaging stories (Young et al. 2013; Ramirez and Bulitko 2014; Porteous et al. 2021).The emergence of LLMs has revolutionized narrative content generation. Recently, LLMs have been used to generate scenes with branching dialogues for interactions between players and non-player characters (NPCs) (Kumaran et al. 2023; Gao and Emami 2023; Akoury, Salz, and Iyyer 2023). However, NPC interaction scenes that involve only clicking through linear dialogue lack player agency and feel more like narrated stories. On the other hand, long narrative arcs pose challenges due to the impracticality of predefining all outcomes. Our research addresses these challenges by creating a framework that generates playable 3D Unity-based game episodes that include player-triggered narrative beats and real-time narrative adaptations that respond to player choices, thereby offering engaging, interactive narrative experiences generated from natural language narrative arcs specified by designers.

Creating engaging game narratives has traditionally been resource-heavy and required significant expertise. Procedural content generation (PCG) has sought to ease this burden, though early planning-based methods mainly focused on linking event sequences by meeting local conditions, paying less attention to the overall narrative structure (Young et al. 2013; Ramirez and Bulitko 2014; Kreminski, Wardrip-Fruin, and Mateas 2020). Recent advances in large language models (LLMs) have revolutionized narrative generation by offering internalized world knowledge, instruction-following, and semantic event-tracking abilities (Guan et al. 2023; OpenAI 2023). These models generate coherent dialogue, structured outputs, and transform high-level outlines into interactive game scripts (Liu et al. 2023; Kumaran et al. 2023).

Language models have shown promise in co-creating dialogues, stories (Chung et al. 2022), and co-write screenplays (Mirowski et al. 2023). However, balancing the overall story with player agency in interactive narratives is crucial (Riedl and Bulitko 2013). We address this with a narrative generation framework for designers to provide high-level story guidance in natural language and then translating it into a partially ordered event graph using LLM reasoning capabilities. The events feature NPC interactions through dialogue, emotes, and gestures. Our framework also supports runtime adaptation of scenes based on player gameplay history to handle the combinatorial expansion of possible paths due to player decisions.

The primary contributions of our framework, NarrativeGenie, are the following:

- Simplifies and automates creating a 3D virtual interactive narrative game from a natural language narrative arc.
- The generated interactive narrative balances player

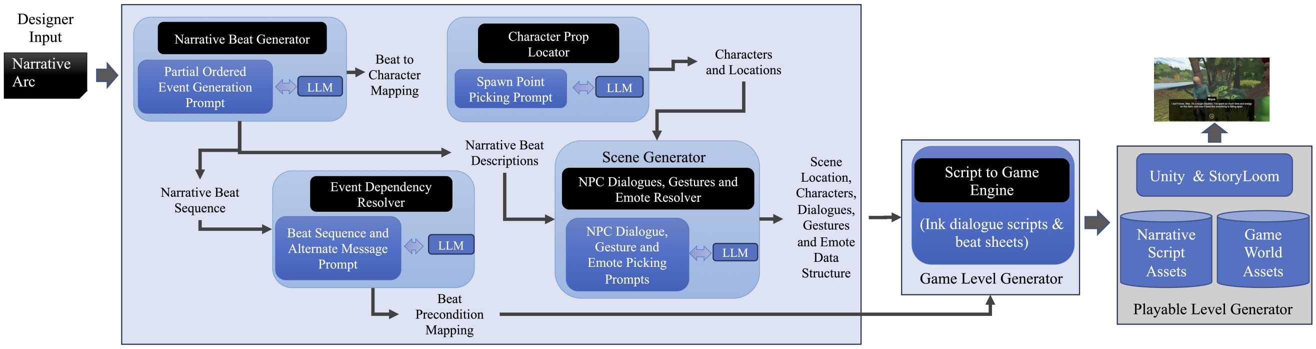

Designer Input
Narrative Arc
Narrative Beat Generator
Partial Ordered Event Generation Prompt
LLM
Beat to Character Mapping
Character Prop Locator
Spawn Point Picking Prompt
LLM
Characters and Locations
Narrative Beat Descriptions
Narrative Beat Sequence
Event Dependency Resolver
Beat Sequence and Alternate Message Prompt
LLM
Beat Precondition Mapping
Scene Generator
NPC Dialogues, Gestures and Emote Resolver
NPC Dialogue, Gesture and Emote Picking Prompts
LLM
Scene Location, Characters, Dialogues, Gestures and Emote Data Structure
Script to Game Engine
(Ink dialogue scripts & beat sheets)
Game Level Generator
Unity & StoryLoom
Narrative Script Assets
Game World Assets
Playable Level Generator

Figure 1: The NarrativeGenie interactive narrative generation framework.

agency and narrative progression with limited out-of-order interaction.

- The generated narrative allows player exploration and player-initiated NPC and game element interactions to progress the narrative.
- Adapts scenes dynamically to player choices during gameplay.
- Demonstrates effectiveness, validated through both human participant assessments and automated evaluations.

Related Work

The interactive narrative research community focuses on crafting engaging storylines and immersive experiences (Riedl and Bulitko 2013; Riedl and Young 2010; Kreminski, Wardrip-Fruin, and Mateas 2020; Stefnisson and Thue 2018). Interactive narrative frameworks foster player engagement by allowing exploration to control the narrative's progression. One technique to blend player input with narrative progression is using experience managers, which dynamically revise narrative elements to maintain the overall arc (Riedl and Bulitko 2013). Researchers often frame this task as an automated planning problem to balance player actions with authorial goals (Ramirez and Bulitko 2014; Riedl and Young 2010). Another approach is representing interactive narratives as story graphs, with story states connected through causal edges, allowing experience managers to track and control narrative progress (Riedl and Young 2010). Techniques like pruning are used on these graphs to control the story experience (Ware et al. 2022). Our framework employs a story-graph paradigm in the generation process to represent the partial order and relationships between narrative events. Our real-time adaptation of narrative interactions is generated by providing this graph and the player's gameplay history as context to the language model.

Researchers are exploring evolving stories through autonomous agents in a story world, followed by story sifting to identify compelling event patterns (Kreminski, Wardrip-Fruin, and Mateas 2020). The synergy of human gameplay with autonomous NPCs driven by LLMs has been used to create emergent narrative events in text-adventure games (Peng et al. 2024). Another approach involves players controlling NPC goals through conversational actions (Oliver and Mateas 2021). AI is often used during the authoring phase, suggesting narrative ideas, next steps for characters, or their final objective based on the current story context (Stefnisson and Thue 2018; Akoury et al. 2020; Kreminski et al. 2022; Martin, Harrison, and Riedl 2016).

Recent research has shown the effectiveness of neural network-based models in generating coherent narratives by linking high-level plot points with narrative events (Rashkin et al. 2020; Yao et al. 2019; Wang, Durrett, and Erk 2020). Generating events first, followed by sentences, has been shown to enhance story coherence (Ammanabrolu et al. 2020). Pre-trained language models have notably influenced narrative generation. Calderwood et al. (2022) finetuned a pre-trained language model on Twine stories to create a mixed-initiative tool for interactive stories. Author-provided control codes have been used to blend human and language model creativity in story generation (Lin and Riedl 2021). Professional writers have found these pre-trained models helpful for co-writing screenplays (Mirowski et al. 2023). Our framework adopts a similar hierarchical approach using LLMs to translate a narrative arc into event graphs. Subsequently, these events are translated into narrative interactions for players in virtual environments.

LLMs like OpenAI's GPT have shown remarkable human-level potential in text generation, understanding natural language semantics, and creating executable program snippets (OpenAI 2023; Chowdhery et al. 2023; Liu et al. 2023; Bubeck et al. 2023). LLMs are begining to have an impact on game design and development (Sweetser 2024), utilizing their vast pre-trained world knowledge and language generation capabilities to create autonomous NPCs that exhibit emergent behavior (Peng et al. 2024; Buongiorno et al. 2024). LLMs have been shown to generate dialogues that can pass as human-created content (Gao and Emami 2023) and produce code and dialogues for story progression (Akoury, Salz, and Iyyer 2023). They can also be trained to generate dialogues matching specific speaking styles (Latouche, Marcotte, and Swanson 2023). Kumaran et al. (2023) showcased a framework for automating the conversion of author instructions into playable scenes with generated dialogues in a Unity virtual environment. For immersive and engaging NPC interactions, dialogues must follow a plot sequence and adapt to player actions. Our framework enhances LLM-generated NPC interaction scenes by generating an overar-

Designer Narrative Arc input
The player is tasked with unraveling the mystery of abnormal behavior and death observed in Tilapia on a lush island. They interact with various characters, such as researchers and farmers, to gather clues and solve the mystery.

Table 1: Example designer input used to generate game episode.

ching narrative beat graph. This graph supports out-of-order interactions and enables real-time adaptive dialogues, ensuring a cohesive and flexible narrative experience in a virtual 3D environment.

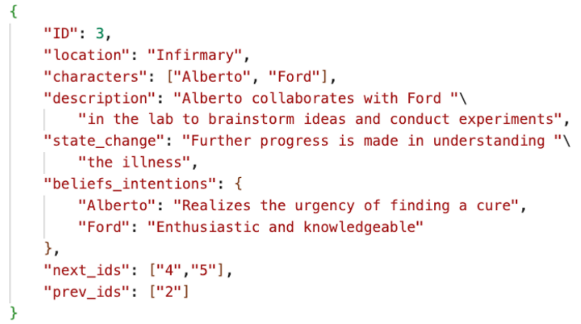
NarrativeGenie

Figure 1 presents the NarrativeGenie architecture that takes as input the designer's narrative arc, a natural language text typically ranging from 20 to 30 words. The *Narrative Beat Generator* then processes this narrative arc using LLMs to convert it into a series of partially ordered narrative events. Following this, the *Event Dependency Resolver* analyzes these events, pinpointing the necessary and sufficient conditions for the event to unfold successfully. Another critical component, the *Character Prop Locator*, examines the requisite characters and props for each narrative event, assessing whether they need to be relocated as a precondition for the event to happen. The *Scene Generator* enriches these narrative events with detailed interaction dialogue, gestures, emotes, and other elements, filling out the narrative. Finally, the *Game Level Generator* compiles these elements, drafting the Ink [1] script components required for the *Playable Level Generator*. This process ends in the creation of interactive playable game levels.

Narrative Beat Generator

The *Narrative Beat Generator* is designed to automate the deconstruction of a story into events within interactive narratives. Narrative beats are key events that drive a story forward, often marked by character development, plot advancement, or critical information revelation. The generator first interprets the designer provided input describing a narrative arc. An example input shown in Table 1. Based on the story details, using LLMs to match story details to available assets in the library, it selects a location, setting, and characters. The locations and characters in the library are given descriptions that the LLMs leverage to match with story events. The initial input acts as a blueprint, instructing the LLM to craft a short story centered around the main protagonist's experiences through these settings, each inhabited by one to two additional characters who are constant within their respective settings while the protagonist navigates between them.

Following the initial story generation, the component's next task is to distill this story into a partially ordered sequence of narrative events or beats. By analyzing the generated story, the LLM outlines five to ten key events, each involving up to three characters. These events are not strictly linear but are designed to offer multiple pathways through the story, thus enabling a dynamic storytelling experience where the sequence of events can vary, giving players choice and variability in how the narrative unfolds. Each narrative beat generated by the LLM is returned as a structured Python object comprising several fields shown in the example (Figure 2). This multi-stage prompting approach is meant to facilitate the logical progression of the story and also enrich each narrative event by emphasizing the evolution of characters and the impact of their decisions or experiences. The culmination of this process is a partially ordered sequence that realizes the author's intended narrative arc. This sequence is a partially ordered set of events that the player can navigate in multiple ways. The output from the Narrative Beat Generator lays the foundation for the remaining components.



```
{
    "ID": 3,
    "location": "Infirmary",
    "characters": ["Alberto", "Ford"],
    "description": "Alberto collaborates with Ford "\
        "in the lab to brainstorm ideas and conduct experiments",
    "state_change": "Further progress is made in understanding "\
        "the illness",
    "beliefs_intentions": {
        "Alberto": "Realizes the urgency of finding a cure",
        "Ford": "Enthusiastic and knowledgeable"
    },
    "next_ids": ["4","5"],
    "prev_ids": ["2"]
}
```

Figure 2: Example of an individual generated narrative beat.

Event Dependency Resolver

The *Event Dependency Resolver* is a component within the framework designed to ensure a coherent progression through a story's events by managing dependencies between narrative beats. At its core, the *Event Dependency Resolver* analyzes narrative beats to delineate the necessary and sufficient conditions required for each beat's completion. This analysis involves identifying 'next_ids' and 'prev_ids' lists associated with each beat description, employing these lists to construct a directed graph that explains the requisite event dependencies. A beat can have multiple parents or a single parent. When they have multiple parents, the sufficiency condition is satisfied if any of its parent beats have completed. If a beat has a single parent, then it can progress if that parent beat has completed.

Another responsibility of the *Event Dependency Resolver* is to dynamically generate alternate NPC dialogues for scenarios where the player attempts to progress through the narrative out of order. This feature is achieved by prompting LLMs with the established narrative event order and the reasoned explanations behind this order. The LLMs then generate contextually appropriate dialogues for NPCs to guide the player back on track, ensuring the narrative unfolds coherently.

[1] Ink (https://www.inklestudios.com/ink), a narrative scripting language for games extended to support Unity functions.

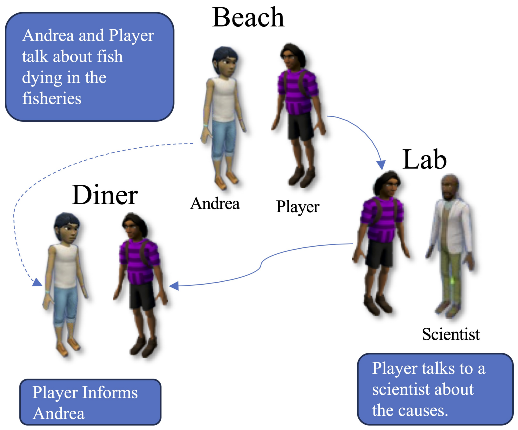

Beach
Andrea and Player talk about fish dying in the fisheries
Andrea
Player
Lab
Scientist
Player talks to a scientist about the causes.
Diner
Player Informs Andrea

Figure 3: Example of an NPC (Andrea) movement to match narrative context of each beat.

Finally, the *Event Dependency Resolver* generates a Python-based data structure that systematically tracks the conditions and alternate dialogues associated with each narrative event. This data structure supplements the original narrative beat description during Ink script generation.

Character and Prop Locator

The *Character and Prop Locator* validates that NPC and props are not static but dynamically positioned throughout the game environment to ensure seamless narrative passage. This component matches narrative elements with the game's physical environment, making narrative progression coherent. The alignment is achieved by interpreting the sequence of narrative beats, which detail the characters involved in each event and the settings in which these events occur. The component first addresses potential character sets and location mismatches to maintain consistency and avoid discrepancies between the game's narrative and available game assets in the visual representation. The framework restricts character and location setting choices to those available within the game's asset library by leveraging LLMs as zero-shot classifiers that map generated elements to asset library elements based on the context provided by each narrative beat.

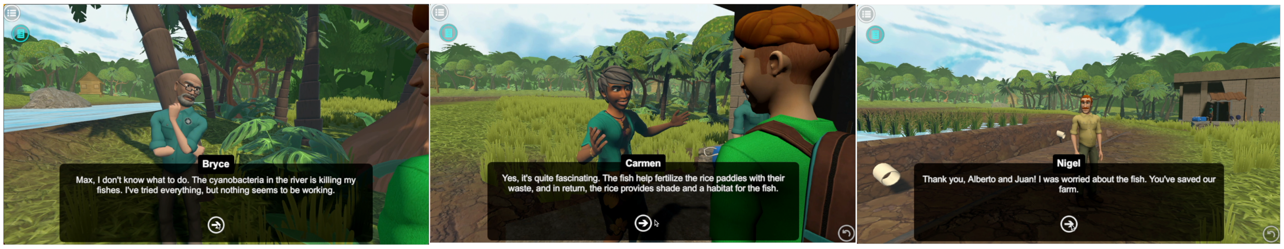
The *Character and Prop Locator* carefully manages the number of characters in each scene, ensuring that there is always at least one character within the scene besides the player. This management extends to aligning the number of characters with the spawn points available within the game's assets, thereby preventing overcrowding. Character movement is another aspect the *Character and Prop Locator* handles. Initially placing NPCs at specific spawn points based on their first interaction within the narrative and then adjusting their positions based on the player's interactions and the chronological order of narrative events, the system guarantees that characters move logically within the game world. This NPC location mapping includes moving characters to available empty slots to prevent an incoherent visual experience. Figure 3 shows an example scenario. All these operations, including character, location mappings, and movements, are recorded in a Python data object. This object generates the Ink script to control the game's mechanics. This Python object, along with the data structures generated by the *Event Dependency Resolver* and *Narrative Beat Generator*, is combined in subsequent components to build the interactive narrative game level.

Scene Generator

Given the scene location, the characters, and their motivation and goal of a scene, theSCENE GENERATOR uses LLMs to generate scenes with dialogues between the characters, along with NPC gestures and emotes that match each NPC utterance. Each narrative beat, as shown in Figure 2, contains all the necessary information to generate a scene. The framework iteratively invokes *Scene Generator* for each narrative beat to flesh out the details of the interaction. The prompts provided to the LLMs ensure narrative coherence and consistency across the overall storyline and between narrative beats. This is achieved by including the overarching plot and a high-level summary of preceding and subsequent events within the LLM context, which helps generate scene dialogues that are contextually relevant and seamlessly integrated into the narrative. Each scene is built as a separate Ink script with the dialogue flow. The *Playable Game Generator* supports interaction triggers enabled in-game entities to start the scene when the player starts a conversation with the corresponding NPC or interacts with a game object. The *Scene Generator* also sets or modifies global variables that record the game state as interactions are triggered. These variables are checked against criteria established by the *Event Dependency Resolver* generated data structure, ensuring that the necessary and sufficient conditions are met before triggering narrative beats. This mechanism streamlines the narrative flow and enhances the game's dynamics, allowing for an engaging player experience.

Playable Game Level Generator

The *Playable Game Generator* is a realization of the StoryLoom architecture (Mott et al. 2019) and uses a textual representation of in-game interactions as a script. The tool combines two key asset types to render the game. The first set of assets are game world assets corresponding to physical entities in the game. It could be the location layout, the characters present, and props like books, food, and other items with which the players can interact. The other types of assets are the narrative scripts that represent the flow of conversation in a scene and mappings of the game interactions that trigger narrative beats recorded in spreadsheets, called beat sheets. The *scene scripts* are the dialogue graph of character conversations, narrations, and instructions on character gestures, emotes, movement, and other aspects necessary to run the scene. Our extended version of the Ink narrative scripting language supports text-based dialogue trees by adding the ability to call custom-defined Unity functions. These functions enable the framework to place characters and props in the environment, control the camera, trigger the movement of game objects, and change the game state, among other actions, primarily through the ink script. The *beat sheet* is



Bryce

Max, I don't know what to do. The cyanobacteria in the river is killing my fishes. I've tried everything, but nothing seems to be working.

Carmen

Yes, it's quite fascinating. The fish help fertilize the rice paddies with their waste, and in return, the rice provides shade and a habitat for the fish.

Nigel

Thank you, Alberto and Juan! I was worried about the fish. You've saved our farm.

Figure 4: Screenshots from game episodes generated by NARRATIVEGENIE.

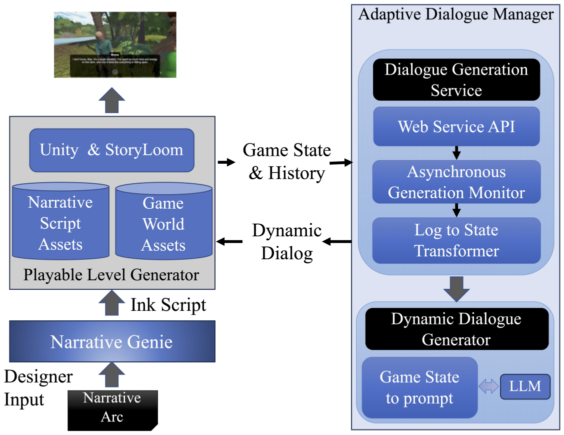


Figure 5: NARRATIVEGENIE dynamic narrative interaction generation framework architecture.

represented as a spreadsheet mapping the game element that triggers the event, the preconditions to trigger the corresponding *scene script*, which then describes the next steps along with the game elements to be rendered for the scene. The framework uses 3D character models and physical objects from an asset library created by artists and game designers. Transactional events are conveyed through dialogue rather than visuals. State changes at the beat level are managed by Unity state variables in the Ink script (Figure 2). The entire game-level generation process is automated, requiring no manual input beyond selecting library assets.

The narrative beats and corresponding scene details generated by the previous components are translated into the Ink script by the *Script to Game Engine* for each scene and mapped in the beat spreadsheet with the corresponding trigger and trigger rule. The *scene scripts* and *beat sheet* are rendered by the *Playable Game Generator* as a 3D game episode in unity. Figure 4 shows a sequence of screenshots from one such generated game episode.

NARRATIVEGENIE Adaptive Dialogue Manager

When creating branching narratives, it is difficult to accommodate all possible combinations of player actions at design time. We extend NARRATIVEGENIE to support dynamic adaptation of narrative interactions using the *Adaptive Dialogue Manager* as shown in Figure 5.

Adaptive Dialogue Manager

In the previous section, we discussed the design-time generation of narrative structures and beats. Here, we extend this framework (Figure 5) to include dynamic NPC interactions at explicitly designated points in the narrative. These interactions are generated in real-time using the *Adaptive Dialogue Manager*, which is integrated with the game engine via a REST interface. When dynamic NPC interaction is indicated in the Ink script, the game engine calls the *Dialogue Generation Service*, sending a detailed gameplay history log. This service then uses LLMs to craft a context-aware prompt, generating appropriate NPC dialogues, gestures, and emotes based on the current game state.

Given that this generation process is time-consuming, it is implemented asynchronously to maintain the interactivity of the game experience. This design choice ensures that the narrative pacing can accommodate the delay inherent in generating NPC interactions, thus preventing any disruption to the player's immersion. Integrating LLMs by the *Adaptive Dialogue Manager* component allows the framework to adapt NPC interactions in real-time based on the player's journey through the game thus far. The LLM's prompt incorporates the original story prompt crafted by the designer and the current game state. This dual context enables the LLM to generate responses that are not only coherent but also aligned with designer goals, guiding the narrative forward in a manner that feels both natural and engaging to the player.

Evaluation

Our evaluation methodology employs automated and human evaluation techniques to assess the generated game episodes. Automated evaluation, while efficient, poses significant challenges in accurately gauging semantic understanding, coherence, and cohesion across extended text passages. This limitation underscores the necessity of human evaluation. We evaluate the generation capabilities of NARRATIVEGENIE and the dynamic real-time narrative adaptations of the NARRATIVEGENIE *Adaptive Dialogue Manager* independently.

NARRATIVEGENIE Evaluation Approach

Our automated evaluation of NARRATIVEGENIE utilizes a dataset comprising ten diverse prompts, each yielding five partially ordered narrative beats, to analyze the generated text's quality.

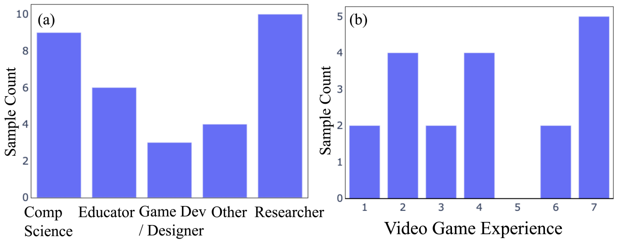


Figure 6: (a) Background of human study participants. (b) Game-playing experience among human study participants.

Along with automated evaluation, human evaluators examine the end-to-end generation process by actively playing through the generated game levels. The end-to-end review offers insights into the overall player experience and narrative alignment to instruction. This dual approach comprehensively assesses the generated content's quality and the framework's generative potential.

NarrativeGenie Automated Evaluation

We evaluate our system's creativity by comparing generated beat text using ROUGE-L scores (Lin 2004).We generated multiple narrative beat sets from designer-provided narrative arcs and analyzed the ROUGE-L score distributions. One distribution compared beat pairs from the same narrative arc, while the other compared beat pairs from different narrative arcs. We expect the beats from the same narrative arc to be more alike than those from different arcs. We use an independent t-test to evaluate the separation between the distributions (Student 1908).

To evaluate how well the system aligns with designer inputs, we use Sentence-BERT (Reimers and Gurevych 2019) to create sentence embeddings from both the designer's narrative arc and the generated texts. By comparing cosine similarity distributions between narrative-arc-to-generated text and narrative-arc-to-unrelated text, we assess the alignment of generated texts with their input narrative arc.

NarrativeGenie Human Evaluation

A purposive sampling method was used for the study recruitment, leveraging the researcher's professional and academic network. 19 participants (aged 25 to 53) with diverse experience levels in playing and developing games evaluated NarrativeGenie. Participants were asked to categorize themselves as Computer Scientists/Software Engineers, Educators, Game Developers or Designers, Researchers, or Other, with the option to select multiple categories. Figure 6(a) displays the histogram of participant categories. Additionally, participants rated their video game experience on a scale of 1 to 7, as shown in the histogram in Figure 6(b). Each participant engaged with the framework during a 1-hour session, starting with an introduction to the framework and instructions on providing an author story summary for game-level generation. After recording their story prompt using the online web interface, participants waited 5-10 minutes for the framework to generate the game episode. They then played the episode and answered five questions about their experience on a 1 to 7 Likert scale and had the option to give qualitative feedback.

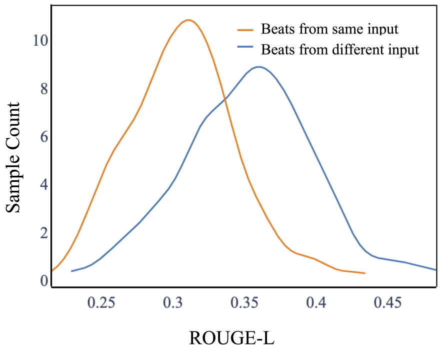


Figure 7: ROUGE-L distribution narrative beats comparisons between those generated from the same and unrelated designer inputs.

The participants were asked to answer the following questions adapted from the User Experience Questionnaire (UEQ) (Schrepp et al., 2017). The UEQ is a validated instrument for assessing the psychometric aspects of a user's experience with a product.

- How would you rate the ease of generating new narratives using the tool (Ease-of-Use)? From 'Complicated' to 'Easy'.
- How engaging and unexpected did you find the generated game (Creativity)? From 'Dull' to 'Creative'.
- To what extent did the interaction in the game accurately represent and respond to your intent (Adaptability)? From 'Unsatisfactory' to 'Satisfactory'.
- How would you evaluate the game's characters and storyline in terms of their believability, coherence, and engagement (Dependability)? From 'Not Interesting' to 'Interesting'.
- Overall, how satisfied are you with the generated game episode (Satisfaction)? From 'Not Satisfied' to 'Satisfied'.

Adaptive Dialogue Manager Evaluation Approach

Adaptive Dialogue Manager is evaluated by asking the participant to play a game episode that dynamically adapts based on gameplay. The participant is introduced to this setting with the mission of helping to solve the mystery behind the theft. The episode's narrative is structured around a fixed set of narrative beats and a predefined story arc, ensuring a consistent storyline to minimize variability introduced by an open-ended generated narrative. This narrative scaffold introduces players to three pivotal characters, each playing a role in the unfolding mystery. Kim provides the context regarding the stolen artifact, including details about the ensemble of characters. Through interactive dialogue, Elise provides all pertinent clues about the case, functioning as a dynamic repository of information regarding the suspect

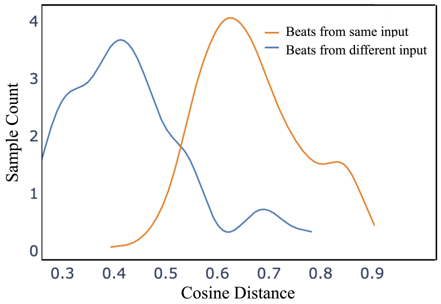


Figure 8: Cosine distance between designer input string embedding and generated narrative beat embedding.

details, potential motives, and other clues. The game's design incorporates up to 16 clues, categorized across multiple dimensions, including deliberately placed red herrings, to challenge the player's deductive reasoning. As players navigate these clues, they are encouraged to formulate hypotheses regarding the perpetrator's identity. Through the *Adaptive Dialogue Manager* intervention, Elise provides timely hints that nudge the player towards asking more pointed questions. Upon deciding on a suspect, the narrative arc progresses to an interaction with the chief investigator, Robert, who summarizes the gathered evidence. This synthesized summary is not static but dynamically generated in real-time by the *Adaptive Dialogue Manager* based on gameplay. Our evaluation methodology employs automated and human evaluation techniques to assess the generated real-time narrative adaptations of the NARRATIVEGENIE *Adaptive Dialogue Manager*.

Adaptive Dialogue Manager Automated Evaluation

Our study analyzes 30 gameplay samples featuring hint and summary generation to evaluate our automated system. Our setup involves simulating a gameplay history where the player has encountered 2 to 3 clues, followed by the provision of a hint, and culminating in a summary and resolution of the mystery. First, we assess the relevance of the hints provided to the player by categorizing all the clues within the game as either 'relevant' or 'irrelevant', with the former being defined as clues that contribute to solving the mystery. We postulate that hints will exhibit a closer semantic proximity to relevant clues, measured through cosine distance between sentence embeddings. Second, to determine whether the hints offer new information to the player, we examine the cosine distance between clues already discovered by the player and those that remain unknown but marked as relevant. Our hypothesis suggests that the hint should be semantically further from known clues and closer to unknown but relevant clues. Lastly, we evaluate the coherence of the summarization with the player's gameplay history by comparing the cosine distance between the summary text and already discovered and undiscovered clues.

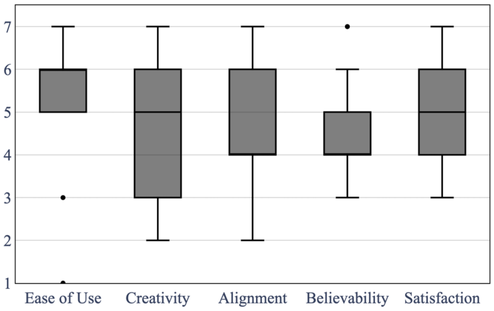


Figure 9: Score distribution for human evaluation of NARRATIVEGENIE.

Adaptive Dialogue Manager Human Evaluation

The 19 participants played and completed the mystery game episode and responded to four questions on a Likert scale (1-7). Additionally, the survey allowed for qualitative feedback, enabling participants to elaborate on their ratings. The participants answered the following questions in the post-survey:

- How engaging and unexpected did you find the mystery game (Engaging)?
- Were the hints from Elise and the overall summarization from Robert coherent (Coherence)?
- How would you evaluate the game's characters and storyline in terms of their believability, coherence, and engagement (Believability)?
- Overall, how satisfied are you with the generated game episode (Satisfaction)?

Results and Discussion

We evaluate the capabilities of NARRATIVEGENIE to generate novel game interactions that align with the author's intent and the *Adaptive Dialogue Manager*'s ability to generate interactions in real-time based on a player's gameplay history.

Variability and Creativity

We use the ROUGE-L score to analyze the similarity between generated narrative beats. ROUGE-L evaluates the longest common subsequence between texts, making it suitable for assessing generated content similarity. Instead of a reference based comparison, we use ROUGE-L to do a relative comparison between narrative beats from identical and different input prompts. Figure 7 illustrates the distribution of ROUGE-L scores, showing higher similarity between narrative beats derived from the same prompts than those from unrelated ones.

We generate five narrative beats for ten prompts and calculate the similarity distance to build the distribution. Our findings confirm that narrative beats from the same prompts show higher similarity, reflected by higher ROUGE-L scores, than those from different prompts. This difference is statistically significant ($p < 0.001$) via an independent t-test. We use a parametric test due to the sample size (hundreds), continuous variables, and near-normal distribution.

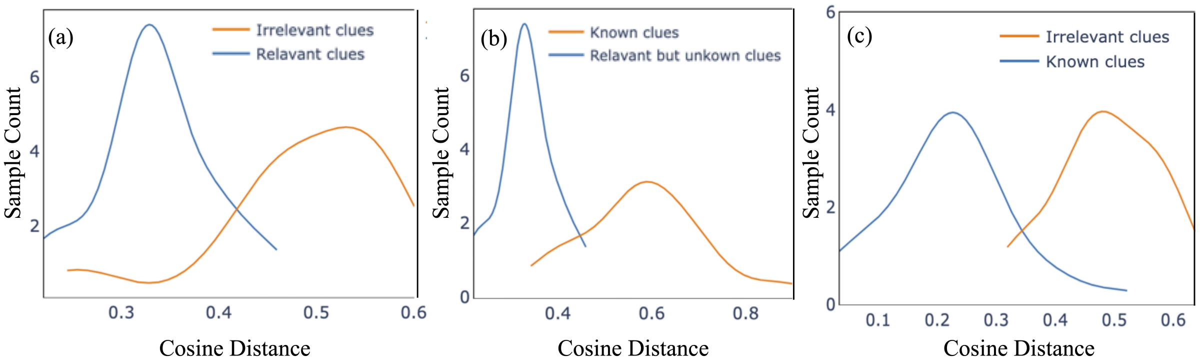


Figure 10: (a) Cosine distances: Hint texts vs. 'Relevant'/'Irrelevant' clues. (b) Cosine distances: Hint texts vs. 'Known'/'Undiscovered' clues. (c) Summary dialogues vs 'Known'/'Unknown' clues.

This result highlights the effectiveness of the script generation process. Notably, the high standard deviation within scores for same-prompt (0.046) is slightly greater than for different-prompt (0.038). Overlapping ROUGE-L score distributions suggest that while the generation process produces similar beats from identical prompts, it also creates diverse narratives, adding novelty and diversity, which will be further substantiated by human evaluations.

Alignment with Designer Intent

We evaluated the alignment between designers' narrative intentions, as expressed in their story summaries, and the narrative beats produced by our framework. This involved assessing the semantic coherence between the prompts and their narrative outcomes. Alignment measured through cosine similarity between designer prompt and generated script sentence embedding supports our hypothesis. Designer's initial input shows higher semantic similarity with the corresponding narrative beats than those generated from unrelated inputs. Figure 8 shows the cosine distance distribution for designer input to generated beats versus input to unrelated beats. These distributions reveal a significant difference (p-value <0.001) in cosine distance between an input and its corresponding generated beats compared to unrelated beats. This result confirms our framework's effectiveness in preserving the designer's intent in the narrative output.

Interestingly, Figure 8 reveals a small peak indicating higher cosine distance between some prompts and their generated beats. This variance was traced back to one particularly vague prompt ("A competition to create the most innovative scientific project brings out unexpected talents"), which led to a broad range of narrative beats. This outlier highlights the challenge of generating tightly aligned content from non-specific inputs. Conclusively, our automated evaluation metrics show that the framework can generate narrative beats that align with the designers' original intent, ensuring the content reflects the initial creative vision. In a later section, we will further investigate this through a human-centric evaluation.

Human Evaluation of NarrativeGenie

Figure 9 presents results from a human study, showing scores from 19 participants across five metrics: Ease-of-Use, Creativity, Alignment, Believability, and Overall Satisfaction. Ease-of-Use had the highest average rating (mean = 5.42), indicating the framework was user-friendly. However, outliers affected this average due to a code bug requiring users to regenerate episodes. Creativity and Alignment with the designer's intention averaged 4.63, indicating moderate originality and adherence to inputs. Believability scored slightly lower (mean = 4.58), suggesting participant skepticism.

The Creativity metric showed high score variance (standard deviation = 1.7), indicating subjective perceptions based on varying user experiences or expectations. Maximum scores consistently reached 7 across all metrics, showing some users perceived the system to excel in all evaluated aspects. Overall, the dataset presents a generally favorable view of the system's usability and effectiveness, with areas for improvement in perceived Creativity, Alignment with input, and Believability.

After reviewing the reasons for low scores (<4.0), Creativity was rated poorly twice due to monotony and shallow character interactions. Despite using designer-specified topics, the narrative was criticized for repetitive and vague dialogue, lacking engaging elements, resulting in redundant tasks with no satisfying conclusion. One case of low Alignment with designer intent highlighted a mismatch between user expectations and the story output, suggesting that open-ended prompts may lead to misaligned narratives. Low Believability scores were due to repetitive dialogue and a disconnect between visuals and dialogues, likely caused by the LLM generating scenes independently without a complete dialogue history.

The small sample size limits robust statistical conclusions at the subgroup level, but some trends are evident. Participants with significant gaming experience rated Ease-of-Use higher (mean = 5.818) than those with less experience (mean = 4.875), indicating that familiarity with games enhances understanding of game development and appreciation for automation frameworks. Conversely, the most experienced

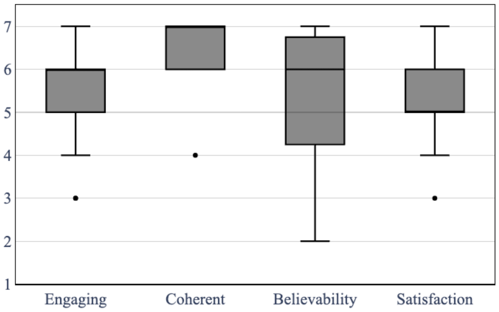


Figure 11: Score distribution for each of the human evaluation questions for *Adaptive Dialogue Manager*.

gamers rated Creativity lower (mean = 4.091) than less experienced participants (mean = 5.375), suggesting greater exposure may lead to more critical assessments. Similarly, computer scientists rated Ease-of-Use higher (mean = 5.67) than non-computer scientists (mean = 5.2), possibly reflecting their understanding of game development. A larger sample size might confirm these findings, emphasizing the need to consider designers' backgrounds in future framework implementations.

Adaptive Dialogue Manager Hint and Summary

To evaluate the *Adaptive Dialogue Manager*, we analyzed 30 mystery episode gameplay sessions to generate a distribution of cosine distances between generated hint texts and clues classified as 'Relevant' and 'Irrelevant'. The results, depicted in Figure 10(a), demonstrate a clear distinction in the distribution of distances. Specifically, the distance between the hint text and 'Irrelevant' clues is significantly (p <0.001) higher than between the hint text and 'Relevant' clues, aligning with our hypothesis that hints presented are relevant to solving the mystery. Continuing with the same 30 gameplay samples, we generated a distribution of cosine distances between the hint text and clues discovered by the player ('Known clues') versus clues relevant to solving the puzzle but not yet revealed to the player. Figure 10(b) visually illustrates the hint text is closer (p <0.001) to 'Relevant' but undiscovered clues. This result indicates that the hints provide new, helpful information to assist players in solving the mystery. Further analysis involved examining whether the generated summary dialogues, which help resolve the mystery, reference clues observed by the player during the game. We applied the same cosine distance methodology to assess the relationships. Figure 10(c) displays the distance distributions between the summary dialogues and 'Known clues' versus other clues. A statistically significant separation (p <0.001) reveals that the summary is more closely aligned with 'Known clues', indicating its coherence and that it incorporates facts from the player's actions within the game.

Human Evaluation of Interaction Intervention

Figure 11 shows the scores from the human evaluation of the *Adaptive Dialogue Manager*. The mean scores suggest that coherence was rated highest among the four metrics, with a mean score of 6.53. This indicates that participants found the narrative threads generated by the framework to be logical and consistent. The low standard deviation (0.772) supports this high mean, indicating a tight clustering of responses around the mean and showcasing a consensus on the framework's ability to maintain narrative coherence despite interactions generated on the fly based on gameplay history. Many participants specifically appreciated Robert's summary, feeling it effectively consolidated all the plot points.

Engagement and satisfaction had similar mean scores of 5.37 and 5.53, indicating participants found the narrative game engaging and satisfying but saw room for improvement. Those who rated these aspects lower wanted more diverse characters and a more challenging way to obtain clues beyond interacting with Elise. They also mentioned that some hints were too obvious, sometimes making the game feel too easy.

Believability scored slightly lower, with a mean of 5.42 and the highest standard deviation (1.465) among the metrics. This higher variance in the narrative's believability score may be attributed to instances where some participants quickly uncovered the correct clues, rendering the game too simple and predictable.

Conclusion

NARRATIVEGENIE harnesses the power of LLMs to automate and streamline the creation of engaging, interactive player-driven narratives. By transforming high-level story outlines into partially ordered event graphs, NARRATIVEGENIE maintains narrative coherence while allowing for player agency through interactive gameplay. It supports dynamic runtime adaptation, ensuring player choices influence the unfolding narrative, thereby enhancing the player experience. Empirical evaluations using automated and human metrics demonstrate that NARRATIVEGENIE is easy to use, with the generated narratives aligning with designers' natural language instructions, effectively balancing narrative structure and player agency.

Several directions for future work are promising. First, it will be instructive to explore approaches for enabling designers to provide guidance during the intermediate stages of narrative generation, improving creativity and alignment. Second, investigating customization of subject matter, plot details, and character inputs rather than requiring a single comprehensive prompt offers another important direction for future work. This could enhance the believability and engagement of generated stories. Finally, exploring an expanded range of real-time narrative interventions could enable more complex experience management and create a richer, more immersive player experience.

Acknowledgments

This work is supported by the National Science Foundation under award DRL-2112635. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Akoury, N.; Salz, R.; and Iyyer, M. 2023. Towards Grounded Dialogue Generation in Video Game Environments. In *Creative AI Across Modalities Workshop, AAAI*.

Akoury, N.; Wang, S.; Whiting, J.; Hood, S.; Peng, N.; and Iyyer, M. 2020. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, 6470–6484.

Ammanabrolu, P.; Tien, E.; Cheung, W.; Luo, Z.; Ma, W.; Martin, L. J.; and Riedl, M. O. 2020. Story realization: Expanding plot events into sentences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7375–7382.

Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S.; Nori, H.; Palangi, H.; Ribeiro, M. T.; and Zhang, Y. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. arXiv:2303.12712.

Buongiorno, S.; Klinkert, L. J.; Chawla, T.; Zhuang, Z.; and Clark, C. 2024. PANGeA: Procedural Artificial Narrative using Generative AI for Turn-Based Video Games. *arXiv preprint arXiv:2404.19721*.

Calderwood, A.; Wardrip-Fruin, N.; and Mateas, M. 2022. Spinning Coherent Interactive Fiction through Foundation Model Prompts. In *Proceedings of the 13th International Conference on Computational Creativity, Bozen-Bolzano, Italy, June 27 - July 1, 2022*, 44–53. Association for Computational Creativity (ACC).

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

Chung, J. J. Y.; Kim, W.; Yoo, K. M.; Lee, H.; Adar, E.; and Chang, M. 2022. TaleBrush: Sketching stories with generative pretrained language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–19.

Crowther, W.; Woods, D.; and Black, K. 1977. Colossal Cave Adventure [Video game]. *PDP-10*.

Gao, Q. C.; and Emami, A. 2023. The Turing Quest: Can Transformers Make Good NPCs? In Padmakumar, V.; Vallejo, G.; and Fu, Y., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, 93–103. Association for Computational Linguistics.

Guan, L.; Valmeekam, K.; Sreedharan, S.; and Kambhampati, S. 2023. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, 36: 79081–79094.

Koenitz, H. 2023. *Understanding interactive digital narrative: immersive expressions for a complex time*. Routledge.

Kreminski, M.; Dickinson, M.; Wardrip-Fruin, N.; and Mateas, M. 2022. Loose Ends: a mixed-initiative creative interface for playful storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 18, 120–128.

Kreminski, M.; Wardrip-Fruin, N.; and Mateas, M. 2020. Toward Example-Driven Program Synthesis of Story Sifting Patterns. In *AIIDE Workshops*.

Kumaran, V.; Rowe, J.; Mott, B.; and Lester, J. 2023. SCENECRAFT: automating interactive narrative scene generation in digital games with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 19, 86–96.

Latouche, G. L.; Marcotte, L.; and Swanson, B. 2023. Generating Video Game Scripts with Style. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Lin, Z.; and Riedl, M. O. 2021. Plug-and-blend: a framework for plug-and-play controllable story generation with sketches. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 17, 58–65.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35.

Martin, L. J.; Harrison, B.; and Riedl, M. O. 2016. Improvisational computational storytelling in open worlds. In *Interactive Storytelling: 9th International Conference on Interactive Digital Storytelling, ICIDS 2016, Los Angeles, CA, USA, November 15–18, 2016, Proceedings 9*, 73–84. Springer.

Mirowski, P.; Mathewson, K. W.; Pittman, J.; and Evans, R. 2023. Co-Writing Screenplays and Theatre Scripts with Language Models: Evaluation by Industry Professionals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–34.

Mott, B. W.; Taylor, R. G.; Lee, S. Y.; Rowe, J. P.; Saleh, A.; Glazewski, K. D.; Hmelo-Silver, C. E.; and Lester, J. C. 2019. Designing and developing interactive narratives for collaborative problem-based learning. In *Interactive Storytelling: 12th International Conference on Interactive Digital Storytelling, ICIDS 2019, Little Cottonwood Canyon, UT, USA, November 19–22, 2019, Proceedings 12*, 86–100. Springer.

Naul, E.; and Liu, M. 2020. Why story matters: A review of narrative in serious games. *Journal of Educational Computing Research*, 58(3): 687–707.

Oliver, E.; and Mateas, M. 2021. Crosston tavern: modulating autonomous characters behaviour through player-NPC conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 17, 179–186.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Peng, X.; Quaye, J.; Rao, S.; Xu, W.; Botchway, P.; Brockett, C.; Jojic, N.; DesGarennes, G.; Lobb, K.; Xu, M.; Leandro, J.; Jin, C.; and Dolan, B. 2024. Player-Driven Emergence in LLM-Driven Game Narrative. In *Proceedings of the 2024 IEEE Conference on Games*.

Porteous, J.; Ferreira, J. F.; Lindsay, A.; and Cavazza, M. 2021. Automated narrative planning model extension. *Autonomous Agents and Multi-Agent Systems*, 35(2): 19.

Ramirez, A.; and Bulitko, V. 2014. Automated planning and player modeling for interactive storytelling. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(4): 375–386.

Rashkin, H.; Celikyilmaz, A.; Choi, Y.; and Gao, J. 2020. PlotMachines: Outline-Conditioned Generation with Dynamic Plot State Tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4274–4295.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992.

Riedl, M. O.; and Bulitko, V. 2013. Interactive narrative: An intelligent systems approach. *Ai Magazine*, 34(1): 67–67.

Riedl, M. O.; and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39: 217–268.

Stefnisson, I.; and Thue, D. 2018. Mimisbrunnur: AI-assisted authoring for interactive storytelling. In *Proceedings of the AAAI Conference on artificial Intelligence and Interactive Digital entertainment*, volume 14, 236–242.

Student. 1908. The probable error of a mean. *Biometrika*, 1–25.

Sweetser, P. 2024. Large language models and video games: A preliminary scoping review. *Proceedings of the 6th ACM Conference on Conversational User Interfaces*.

Wang, S.; Durrett, G.; and Erk, K. 2020. Narrative interpolation for generating and understanding stories. *arXiv preprint arXiv:2008.07466*.

Ware, S.; Garcia, E. T.; Fisher, M.; Shirvani, A.; and Farrell, R. 2022. Multiagent Narrative Experience Management as Story Graph Pruning. *IEEE Transactions on Games*, 15(3): 378–387.

Yao, L.; Peng, N.; Weischedel, R.; Knight, K.; Zhao, D.; and Yan, R. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7378–7385.

Young, R. M.; Ware, S. G.; Cassell, B. A.; and Robertson, J. 2013. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative*, 37(1-2): 41–64.