StoryVerse: Towards Co-authoring Dynamic Plot with LLM-based Character Simulation via Narrative Planning

Yi Wang
Autodesk Research
San Francisco, USA
yi.wang@autodesk.com

Qian Zhou
Autodesk Research
Toronto, Canada
qian.zhou@autodesk.com

David Ledo
Autodesk Research
Toronto, Canada
david.ledo@autodesk.com

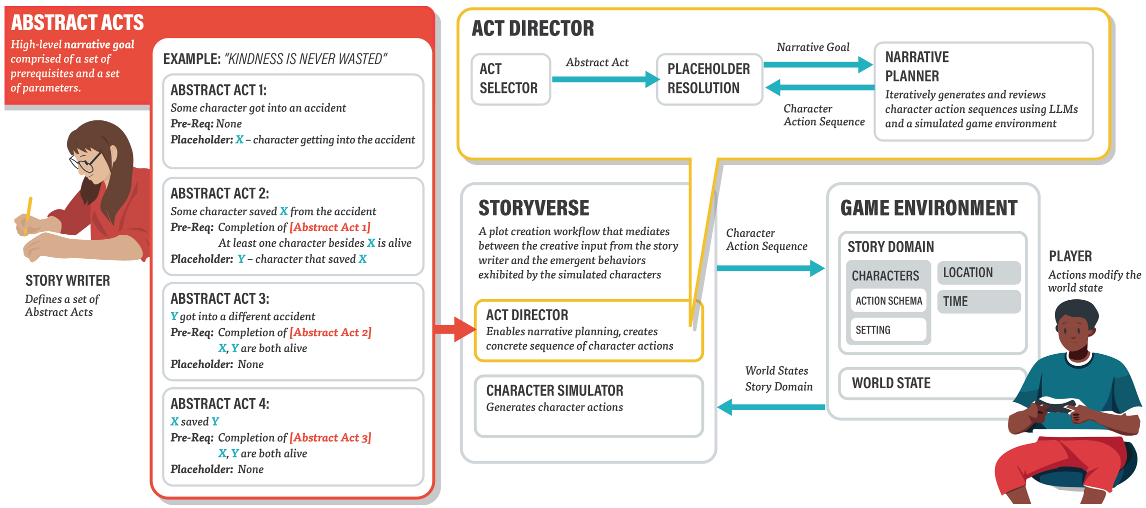

Figure 1: We present a novel plot creation workflow that mediates between a writer's authorial intent and the emergent behaviors from LLM-driven character simulation through an authorial structure called "abstract acts" as well as an LLM-based narrative planning process that iteratively generates and reviews character action sequences.

ABSTRACT

Automated plot generation for games enhances the player's experience by providing rich and immersive narrative experience. Recent advancements use Large Language Models (LLMs) to drive the behavior of virtual characters, allowing plots to emerge from interactions between characters and their environments. However, the emergent nature of such decentralized plot generation makes it difficult for authors to direct plot progression. We propose a novel plot creation workflow that mediates between a writer's authorial intent and the emergent behaviors from LLM-driven character simulations, through a novel authorial structure called "abstract acts". Writers create high-level plot outlines which are transformed into character actions via an LLM-based narrative planning process, based on the game world state. This results in narratives co-created by the author, the simulated characters, and the player. We present *StoryVerse* as a proof-of-concept system to demonstrate the workflow, and showcase its versatility across various stories and game environments.

CCS CONCEPTS

• **Computing methodologies → Natural language processing**; • **Applied computing → Computer games**; • **Software and its engineering → Interactive games**.

KEYWORDS

Narrative Planning, Character Simulation, Large Language Models, Video games, Generative AI

ACM Reference Format:
Yi Wang, Qian Zhou, and David Ledo. 2024. StoryVerse: Towards Co-authoring Dynamic Plot with LLM-based Character Simulation via Narrative Planning. In *Proceedings of the 19th International Conference on the Foundations of Digital Games (FDG 2024), May 21–24, 2024, Worcester, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3649921.3656987

FDG 2024, May 21–24, 2024, Worcester, MA, USA

ACM ISBN 979-8-4007-0955-5/24/05
https://doi.org/10.1145/3649921.3656987

1 INTRODUCTION

Automated plot generation for games enriches player experience by creating adaptive, immersive narratives. Traditional symbolic narrative planning [4, 7, 13, 17] achieves this by setting a narrative goal and generating "narrative plans" [4] or action sequences leading to this goal. However, this requires a hand-crafted knowledge base, limiting the complexity and scale of plots [10]. Large Language

Models (LLMs) offer a solution by simulating character behaviors, allowing plots to emerge naturally from character-environment interactions [2, 9, 15]. Generative Agents [9], for example, simulate human behaviors using LLMs, eliminating the need for crafting a knowledge base. They store agent experiences, form higher-level thoughts from these memories, and retrieve them for behavior planning, showcasing LLMs' potential as proxies for human behaviors. Plot generation through LLM-controlled characters creates *emergent narratives* [1]. However, this type of plot generation *"struggles with 'herding cats' when the narrative is required to contain certain content"* [16]. It is unclear how a story writer might control and direct the plot progression, especially considering how it depends on the collective behavior of multiple characters involved. In contrast, traditional symbolic planning allows writers to set constraints on behaviors and centralize decision-making [16]. The question remains how to combine the advantages of both, crafting an authorial structure compatible with decentralized, LLM-based character simulations.

In this work, we present *StoryVerse*, a system that uses LLM-based narrative planning and character simulation to generate dynamic stories through a novel authorial structure called "abstract acts". *StoryVerse* introduces a plot creation workflow that mediates between a writer's authorial intent and the emergent behaviors from LLM-driven character simulations, given a game environment. Instead of directly specifying characters' actions to interfere with the character simulation, the writer defines high-level plot outlines as abstract acts, which are later "instantiated" into character actions through an *Act Director* component. When instantiating an abstract act, the *Act Director* takes into account the state of the game world, the previous acts, and character actions resulting from an ongoing LLM-driven *Character Simulator* component. The resulting character actions, which can be executed in the game environment to update the world state, thus informing subsequent character simulation and the following acts. This allows writers to create "living stories" that dynamically adapt to various game world states, influenced by the writer's authorial intents, character simulation results, and players' actions. To demonstrate the feasibility of the proposed workflow, we present a proof-of-concept prototype integrating an LLM-based character simulator similar to *AgentVerse* [2], and a novel LLM-based narrative planner, tightly coupled by the game environment and the state of the game world. We showcase *StoryVerse* with examples across various stories and game environments.

2 STORYVERSE: DYNAMIC PLOT CREATION

Our work is built on top of character simulations in game environments where the characters' behaviors are driven by LLMs [9]. The goal is to provide a novel plot creation workflow that allows a story writer to direct the progression of the plot while leaving room for the details to emerge from the character simulation as well as the interaction between the player and characters.

2.1 Plot Creation Workflow

Our goal is to enable writers to create abstract, high-level plot outlines that can be automatically "instantiated" into specific in-game character actions, based on the game world state. The system integrates an *Act Director*, a *Character Simulator*, and a *Game Environment* (Figure 1). The *Act Director* processes writer input and uses an LLM-based planner to generate character action sequences. The *Character Simulator* generates default character actions in the absence of writer input. The *Game Environment* includes characters, locations, and action schemas, referred to as the *Story Domain* [10] (Figure 2). It also maintains the *World State*, holding relevant game mechanic values (e.g., location, strength and health points). We introduce "abstract acts" for writers to define high-level plot outlines.

During play-time, the system checks for any eligible abstract acts. If present, the *Act Director* processes these acts and generates character actions. If not, the *Character Simulator* takes over. After the character actions are generated, the *Game Environment* executes the actions and updates the world states. The player can interfere by changing world states, for example, saving or killing a character. The above process repeats with updated world states.

This resulting story is shaped by the writer, the simulated characters, and the player. While the writer sets abstract acts outlining high-level conflicts and turning points, the structure leaves room for details to emerge from character simulation and player interactions.

2.2 Abstract Acts

Inspired by traditional screenplay writing [6], we use "abstract acts" to structure character actions toward a (possibly abstract) narrative goal, separating the story from specific world state details. This abstraction allows the final story to be collaboratively created by the author, the player, and the LLM-driven character simulation. By detaching the story from specific world states, it becomes more adaptable to inputs from multiple sources, enabling dynamic storytelling.

An *abstract act* is defined by the following elements: (1) **A high-level narrative goal** representing a dramatic conflict or a turning point in the story (e.g., *"a character gets into a life-threatening accident"*). (2) **A set of prerequisites** connected by logical conjunction and/or disjunction. Each prerequisite can be: (i) A statement about the current world state that can be evaluated (as true or false) by the LLM or the game environment (e.g., ' *"John is loved by all characters"* or *"there are at least 5 characters alive"*), (ii) A statement about the player's action, (e.g., *"The player executes* `eat(X)` *action with* `X` *being a poisonous food item"*), and (iii) The fulfilment or failure of another abstract act, used to specify dependencies between abstract acts. (3) **A set of placeholders**, which refer to specific content in the "instantiation" of current act, to carry over to subsequent acts. Each placeholder is a pair of code-name and descriptions (e.g., *"X - the character who got into the life-threatening accident"*).

The narrative goal and prerequisites may include placeholders for story specifics that will be determined by the instantiation of a previous act. For example, *"a character saves X's life"*, with *"X"* being a placeholder for the character who *"got into a life-threatening accident"* in a previous act (which is unknown at the time of authoring). Incorporating placeholders into the narrative goals and prerequisites allows the story to adapt and respond to the choices and actions made by players as well as the interventions from character simulation. In addition, it allows the author to enforces coherence and define continuations between acts without knowing the specifics of each act.

We use a story about *"Kindness is never wasted"* to showcase the usage of abstract acts extracted from the Ant & Dove story from Aesop Fable following prior work on narrative planning [3]. As shown in Figure 1, the writer creates the four abstract acts with high-level

Figure 2: Two example story domains - The Ville (top) and Ant & Dove (bottom) - together with instantiated versions of the four abstract acts from Figure 1. Note that the text for narrations, dialogs, and monologues is all generated by LLMs.

narrative goals such as life-threatening and life-saving events to depict the turning points as well as the relationship between characters. Note that the execution order does not depend on the creation order. The prerequisites indicate the chronological relationship between acts and enforce consistency among them. This approach allows the writer to create branching stories, where the path to take is determined by both the player's actions and the character simulation.

2.3 Act Director

The Act Director uses writer-specified abstract acts and LLMs for narrative planning, creating a sequence of character actions, or a *(narrative) plan*, to achieve the goals outlined in the abstract acts. During each play-time timestep, the Act Selector checks for any pending abstract acts that have met their prerequisites for execution. Once an act is selected, placeholders in the narrative goal are replaced with the referred content from previously executed acts. For example, a narrative goal *"Y saved X from the accident"* might become *"The dove saved the ant from the accident"*.

Following [14], we design an LLM-based iterative narrative planning process involving plan generation and plan revision for better quality of the narrative plans (Figure 3). The plan generator takes a narrative goal, the world state, and the story domain as the input and is prompted to generate a plan. The generated narrative plans are in the form of a list of tuples containing an action following the action schema in the story domain, the subject of the action, and the thought behind the action. Once generated, the plan reviewer provides feedback regarding the quality and feasibility of the action sequence to improve it. The feedback has three parts: (1) **Overall Coherency Evaluation**: Feedback is obtained by prompting an LLM to comment on the overall coherency of the generated plot and make suggestions for improvement. (2) **Game Environment Evaluation**: Similar to [14], feedback comes from executing the generated

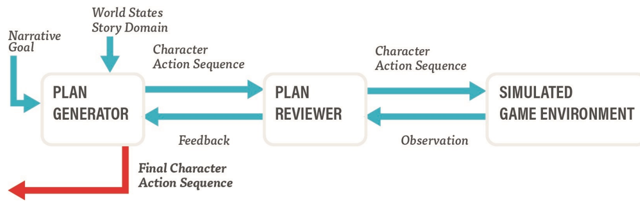

Figure 3: LLM-based narrative planning that iteratively generates and reviews the plan.

action sequence in a simulated game environment, and reporting observations on the success/failure of the execution. (3) **Character Simulation Evaluation**: For every action in the sequence, we prompt an LLM to play the role of the subject of the action. Given the current world state including the character's memory, we ask the LLM if the motivation for the character to perform the action has been established. We include the explanation to this question in the feedback, if the motivation has not been established.

The plan generator then revises the action sequence based on the feedback. This process repeats until the plan is executable or for a user-specified number of iterations. Once the final character action sequence is generated, it is executed by the game environment to update the world state. It's also sent to the Placeholder Resolution component, which uses an LLM to identify specific content referred to by each placeholder in the act, storing the placeholder-content mapping for future act execution. For example, if placeholder X is defined to refer to "*the character who got into an accident*", then Placeholder Resolution will ask an LLM, "*what is the character who got into an accident*", and then store the answer with placeholder X.

2.4 Implementation and Examples

We implemented a minimal version of *AgentVerse* [2] for character simulation. Each character has a textual description and structured memory, updated through action execution. An action from LLM is executed every timestep. We also implemented a game environment proxy that simply maintains and updates world state variables upon the execution of character actions. We utilized gpt-4-0125-preview as the LLM, with a maximum of 2 narrative plan revisions ($maxStep$).

We present two examples to demonstrate our plot creation workflow using two story domains: *The Ant & Dove* [3] and *The Ville* [9]. Both examples are driven by a set of abstract acts based on the theme of "Kindness is never wasted" (Figure 1). Figure 2 lists the story domains and generated plots. The generated plot can adapt to different possible world states. For example, in *Ant & Dove*, if the ant is dead, the Act Director can adapt the story and let the hunter and dove save each other.

3 DISCUSSION AND CONCLUSION

Like other work utilizing LLMs for story generation, the quality of the generated plots is limited by the LLMs' known challenge of preserving long-term dependency and coherence [8]. It is worth investigating how to incorporate existing solutions such as utilizing Retrieval-Augmented Generation [5] and adopting a hierarchical generation approach [8] to mitigate this issue. The frequent call to LLMs also raises concerns on latency during gameplay.

Riedl and Bulitko [11] have proposed a taxonomy of interactive narrative systems in terms of three dimensions - authorial intent, character autonomy, and player modeling. *StoryVerse* situates at hybrid authorial intent with a relatively strong level of virtual character autonomy, which has been identified as a gap in [11]. The Act Director serves as an autonomous surrogate for the writer, which has a similar role as the experience manager in [12]. However, an LLM-based approach enables a more flexible and generalizable representation of author's constraints on the plot search space and reduces the required knowledge engineering expertise for specifying authorial intents compared to symbolic planning based approaches.

In this work, we proposed a novel plot creation workflow intended to balance authorial intent and emergent behaviors for a game with LLM-driven virtual characters, demonstrated by our prototype system *StoryVerse*. The workflow allows stories to be co-created by the author, the character simulation and the player's actions asynchronously. Our next steps include systematic evaluations in terms of standard quantitative metrics for story generation, as well as the feasibility and potential of the workflow in a real-world setting.

REFERENCES

[1] Ruth Aylett. 1999. Narrative in virtual environments-towards emergent narrative. In *Proceedings of the AAAI fall symposium on narrative intelligence*. USA, 83–86.

[2] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848* (2023).

[3] Arthur C Graesser, Scott P Robertson, and Patricia A Anderson. 1981. Incorporating inferences in narrative representations: A study of how and why. *Cognitive Psychology* 13, 1 (1981), 1–26.

[4] Michael Lebowitz. 1985. Story-telling as planning and learning. *Poetics* 14, 6 (1985), 483–502.

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

[6] Robert McKee. 1997. Story: Substance, structure, style and the principles of screenwriting. 1997. *Kent, Great Britain: Methuen* (1997).

[7] James R Meehan. 1977. TALE-SPIN, An Interactive Program that Writes Stories.. In *Ijcai*, Vol. 77. 91–98.

[8] Piotr Mirowski, Kory W Mathewson, Jaylen Pittman, and Richard Evans. 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.

[9] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–22.

[10] Steven Poulakos, Mubbasir Kapadia, Guido M Maiga, Fabio Zünd, Markus Gross, and Robert W Sumner. 2016. Evaluating accessible graphical interfaces for building story worlds. In *Interactive Storytelling: 9th International Conference on Interactive Digital Storytelling, ICIDS 2016, Proceedings 9*. Springer, 184–196.

[11] Mark Owen Riedl and Vadim Bulitko. 2013. Interactive narrative: An intelligent systems approach. *Ai Magazine* 34, 1 (2013), 67–67.

[12] Mark O Riedl, Andrew Stern, Don Dini, and Jason Alderman. 2008. Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning* 4, 2 (2008), 23–42.

[13] Mark O Riedl and Robert Michael Young. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39 (2010), 217–268.

[14] Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. 2023. Describe, Explain, Plan and Select: Interactive Planning with Large Language Models Enables Open-World Multi-Task Agents. *arXiv preprint arXiv:2302.01560* (2023).

[15] Zhilin Wang, Yu Ying Chiu, and Yu Cheung Chiu. 2023. Humanoid Agents: Platform for Simulating Human-like Generative Agents. arXiv:2310.05418 [cs.CL]

[16] Stephen G Ware and Cory Siler. 2021. Sabre: A narrative planner supporting intention and deep theory of mind. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 17. 99–106.

[17] R Michael Young, Stephen G Ware, Brad A Cassell, and Justus Robertson. 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 (2013), 41–64.