

# 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

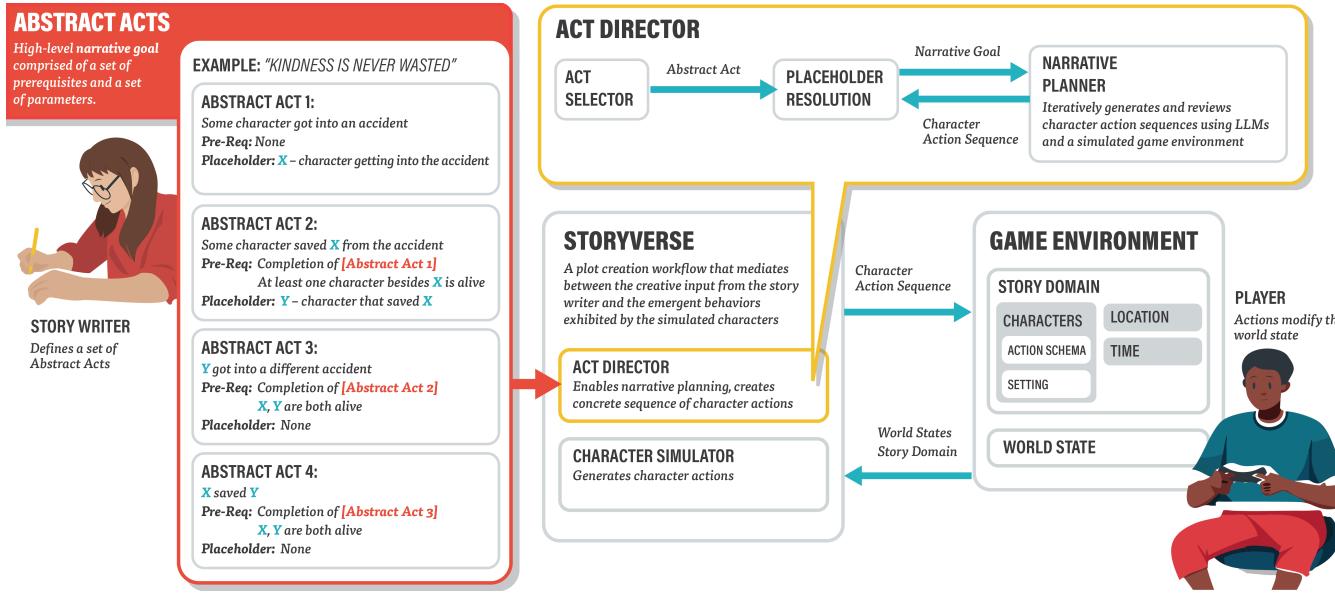
Autodesk Research  
Toronto, Canada  
david.ledo@autodesk.com

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 that adapts to the player’s actions. Traditional approaches adopt a symbolic narrative planning method which limits the scale and complexity of the generated plot by requiring extensive knowledge engineering work. 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 simulation, through a novel authorial structure called “abstract acts”. The writers define high-level plot outlines that are later transformed into concrete character action sequences via an LLM-based narrative planning process, based on the game world state. The process creates “living stories” that dynamically adapt to various game world states, resulting in narratives co-created by the author, character simulation, and player. We present *StoryVerse* as

a proof-of-concept system to demonstrate this plot creation workflow. We showcase the versatility of our approach with examples in different 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

## 1 INTRODUCTION

Automated plot generation for games enhances the player’s experience by providing rich and immersive narrative experience that adapts to the player’s actions. One prominent approach is symbolic narrative planning [6, 9, 15, 19, 21], where a narrative goal is set as the plot’s end, and character action sequences leading to this goal - known as “narrative plans” [6] - are generated. Generating such plans requires a hand-crafted knowledge base using formal logical

language that defines preconditions and effects of a predefined action set within a story [12]. Given the extensive engineering work required to construct this knowledge base, the generated plots offer limited complexity and scale.

One way to address this challenge is to use Large Language Models (LLMs) to drive character behaviors [3, 11, 18]. Through LLMs, plots naturally emerge from the interactions between virtual characters and their environments. For example, Generative Agents [11] simulate believable human behaviors extending LLMs to store a comprehensive log of the agents’ experiences, combine their memories to form higher-level thoughts, and retrieve them as needed for behavior planning. This demonstrates LLM’s capabilities of acting as proxies for human behaviors and conversations [1], eliminating the need to craft the knowledge base.

Plot generation via fully autonomous characters controlled by LLMs produces *emergent narratives* [2]. However, this type of plot generation “struggles with ‘herding cats’ when the narrative is required to contain certain content” [20]. 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. This is even more challenging for LLM-based emergent narratives due to their black-box nature. In a more traditional symbolic narrative planning approach, writers are able to set constraints on the generated behaviors, and the planning systems centralize decision making to ensure they meet the author’s constraints [20]. It is unclear what kind of authorial structure could be compatible with emergent narrative from LLM-based character simulation, which is generated in a decentralized manner.

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 simulation, 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 concrete acts 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 concrete act contains a sequence of 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 created a proof-of-concept prototype integrating an LLM-based character simulator similar to *AgentVerse* [3], 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 in different stories. Specifically, our contributions are:

- A plot creation workflow that mediates between the creative input from the story writer and the emergent behaviors exhibited by simulated characters.

- An authorial structure called “abstract acts” that enables co-creating dynamic plot between a writer and an LLM-based character simulation.
- An LLM-based narrative planning process that iteratively generates and reviews character action sequences using LLM and a simulated game environment.
- *StoryVerse*: a proof-of-concept prototype that demonstrates the feasibility and applications of the proposed workflow.

## 2 METHOD: DYNAMIC PLOT CREATION USING LLM-BASED NARRATIVE PLANNING AND CHARACTER SIMULATIONS

Our work is built on top of the character simulation in game environments where the characters’ behaviors are driven by LLMs [11]. 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. In this section, we describe our workflow centered around the concept of “abstract act” and LLM-based narrative planning.

### 2.1 Plot Creation Workflow

Our goal is to allow writers to author abstract yet high-level plot outlines that can be automatically “instantiated” into specific character actions at play-time, with the instantiation taking into account the state of the game world. The system integrates three major components (Figure 1): an *Act Director*, a *Character Simulator*, and a *Game Environment*. The *Act Director* processes the writer’s input and uses an LLM-based narrative planner to generate character action sequences. The LLM-based *Character Simulator* produces character actions that drive the characters by default when no eligible input is found from the writer. The *Game Environment* contains a set of characters, locations, and action schemas that specify character actions that are executable through the game system. Consistent with symbolic narrative planning [12], we refer to these elements as the *Story Domain*. Figure 2 (left) shows two examples of story domains. The game environment also maintains the *World State*, which consists of a collection of variables that hold relevant values for the game mechanics, such as the characters’: location, attributes (e.g., strength and health points), relationship scores, as well as their memories from the simulation. To enable writers to co-develop character actions with the LLM-driven character simulation, we introduce abstract acts, a novel authorial structure compatible with character simulation. The writer can define a high-level plot outline through a set of abstract acts.

At each timestep during play-time, the system evaluates whether there are any pending abstract acts that have met their prerequisites. If there are, the *Act Director* will process these acts and generate character actions. Otherwise, the *Character Simulator* will execute to generate character actions. After the character actions are generated, either from *Act Director* or *Character Simulator*, the *Game Environment* executes the actions and updates the world states. The *Game Environment* ensures that the world states reflect the changes resulting from the generated character actions. Once the game environment executes the actions, the game player can interfere with the game by changing the world states, such as by saving or killing

**STORY DOMAIN****CHARACTERS**

Isabella Rodrigues is the owner of Hobbs Cafe and loves making people feel welcome.



Eddy Lin is a student at Oak Hill College studying music theory and composition.



Abigail Chen is a digital artist and animator who loves to explore how to use technology to express ideas.

**LOCATIONS****ACTION SCHEMA**

MoveTo(X)	Work(X)	Sleep()
Think(X)	ShopFor(X)	Cook(X)
TalkTo(X,Y)	Eat(X)	Drink(X)
Study(X)		

**STORY DOMAIN****CHARACTERS****LOCATIONS****ACTION SCHEMA**

MoveTo(X)	TryToKill(X)
SlipIntoWater()	Kill(X)
DrawnToDeath()	Think(X)
Save(X)	

**(SELECT) CHARACTER SIMULATION**

Thinks("I need to make sure everything is ready!")  
Isabella needs to prepare for the Valentines Day party at Hobbs Cafe. She needs to plan ahead.

**ACT 1 SOME CHARACTER GOT INTO AN ACCIDENT**

MoveTo(JohnsonPark)  
Johnson Park is quiet and inspiring, perfect for working on my music composition project.



Eat(Snacks)  
Eddy enjoys his snacks, feeling refreshed and ready to work.



Work(Music)  
Eddy is making great progress on his project, feeling inspired by his surroundings.



MoveTo(Market)  
I need to pick up snacks and water to keep my energy up while I work!



ShopFor(Snacks)  
Eddy selects snacks and a bottle of water.



MoveTo(JohnsonPark)  
Back to Johnson Park to continue my composition.



Think("Ouch!")  
Eddy realizes he's hurt his ankle in the fall.

**ACT 2 SOME CHARACTER SAVED [EDDY] FROM THE ACCIDENT**

MoveTo(JohnsonPark)  
I'm going to the park for some inspiration for my project.



Think("Inspired")  
Abigail feels optimistic about finding inspiration in the park's natural beauty.



TalkTo(Eddy, ...)

Abigail asks "are you okay? Let me help you up". She offers her hand to Eddy.

**ACT 3 [ABIGAIL] GOT INTO A DIFFERENT ACCIDENT**

MoveTo(SupplyStore)  
I need to pick up some art supplies for my project.



ShopFor(Supplies)  
These supplies will help bring my animation project to life!



MoveTo(JohnsonPark)  
I'll head back to the park to sketch some scenery.

**ACT 4 [EDDY] SAVED [ABIGAIL]**

Work(Music)  
With the park's calm ambiance, Eddy finds the creativity flowing effortlessly.



TalkTo(Eddy, "Your music is so inspiring!")  
Abigail finds solace in Eddy's music, feeling motivated to start her project anew.



TalkTo(Abigail, "Let's inspire each other!")  
Eddy is grateful for Abigail's words and feels a sense of camaraderie



Think("I'm glad Abigail is here...")  
Eddy sees potential in a collaborative project.

**(SELECT) CHARACTER SIMULATION**

MoveTo(HobbsCafe)

A change in scenery might spark new ideas for my composition. Plus, a cup of coffee could really help me focus.

**CHARACTER SIMULATION**

THE END STATE IS REACHED, SIMULATION CONTINUES



Now that I've had a bit of an adventure and was saved by the dove, I should go to the oak tree to find some food. It's safer there.



MoveTo(OakTree)



Now that the ant is safe, I should check on the oak tree. It's a good place to rest and maybe I'll find more friends there.



MoveTo(OakTree)



The oak tree next to the brook could be a promising spot to find game. Plus, there's a dove there. Maybe I can learn more and find an opportunity.



MoveTo(OakTree)

**CHARACTER SIMULATION****CHARACTER SIMULATOR INITIALIZES THE STATE**

I should head to the bank to find some food and maybe meet other creatures. It's always nice to socialize and share the latest news.



MoveTo(Bank)



I feel like visiting the bank today. It's always nice to meet new friends, and maybe I'll see the ant there.



MoveTo(Bank)



ACT 1 SOME CHARACTER GOT INTO AN ACCIDENT



I see some delicious leaves across the brook. I should go there to eat.



MoveTo(Brook)



Oh no! The edge is very slippery! I'm falling into the water!



SlipIntoWater()



ACT 2 SOME CHARACTER SAVED [ANT]



I should go to the brook. Maybe I can find the ant there and ensure its safe.



MoveTo(Brook)



I see the ant struggling in the water. I must save it before it's too late.



Save(Ant)

**INSTANTIATED ACTS****ACT 3 [DOVE] GOT INTO A DIFFERENT ACCIDENT**

I should move closer to the brook. I might get a better shot at my target there.



MoveTo(Brook)



There's the dove! Saving the ant. It's distracted. Now it's my chance.



TryToKill(Dove)



ACT 4 [ANT] SAVED [DOVE]



I see the hunter aiming at the dove. I must do something quickly.



Think(...)



If I bite the hunter, maybe I can distract him and save the dove.



TryToKill(Hunter)

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.

a character. Finally, the game environment sends the updated world states back to *Act Director* for the next timestep.

This process creates a living story with its progression influenced by the story writer, the character simulator, and the actions of the players within the game environment. While the writer defines abstract acts, which describe high-level dramatic conflicts and turning points, this authorial structure leaves space for the details to emerge from both the character simulation and how the player interacts with the characters. As a result, the stories have the flexibility to evolve and respond to these dynamic factors.

## 2.2 Abstract Acts

In traditional screenplay writing, writers use the concept of an “act” to structure a sequence of movements that turns on a major reversal

in the value-charged condition of a character’s life [8]. Inspired by this notion, we use the term abstract act to structure a sequence of character actions leading toward a (possibly abstract) narrative goal. The term “abstract” highlights the separation of the story from the specific details of the underlying world states. This allows for the final story to be a collaborative creation involving the author, the player, and the emergent behaviors from the LLM character simulation. By abstracting the story from the specifics of the world states, it becomes more adaptable and open to contributions from multiple sources, enabling a dynamic storytelling experience.

An *abstract act* is defined by the following elements:

- 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”).

- A set of prerequisites connected by logical conjunction and/or disjunction. Each prerequisite can be:
  - (1) A statement about the current world state that can be evaluated (as true or false) by the LLM or game environment (e.g., “*John is loved by all characters*” or “*there are at least 5 characters alive*”),
  - (2) A statement about the player’s action, (e.g., “*The player executes eat(X) action with X being a poisonous food item*”),
  - (3) The fulfilment or failure of another abstract act, used to specify dependencies between abstract acts.
- A set of placeholders – a construct representing an answer to be fulfilled at a future time (similar to “Promises” in computer programming). Placeholders 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 natural language description (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).<sup>1</sup> Incorporating placeholders into the narrative goal 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 enforces coherence of plot elements across abstract acts and allows the author to define continuations between acts without knowing the specifics of each act in advance.

When multiple abstract acts are added by the writer, they are not executed sequentially. Each act has its own prerequisites as defined by the writer. This approach allows the writer to create branching stories, where the path to take is determined by both the player’s actions, as well as the character simulation. Additionally, abstract acts can be grouped by the writer, indicating that they form a complete storyline. This means that the entire story reaches completion only when all the abstract acts within the group have been executed. Grouping abstract acts allows for a structured narrative authoring experience.

*Example 2.1.* 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 [4, 16]. As shown in Figure 1, the writer creates the four abstract acts with high-level 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 placeholders for characters indicate the chronological relationship between acts and enforce consistency among them.

### 2.3 Act Director

Using the writer-specified abstract acts, the Act Director enables a narrative planning approach using LLMs to generate a concrete sequence of character actions that can achieve the narrative goals

<sup>1</sup>Note that the occurrence of a placeholder in a prerequisite implies the completion of an act with the placeholder as an output as another prerequisite.

specified in the abstract acts. The sequence of character actions is called a (*narrative plan*).

At each timestep of play-time, the Act Selector evaluates whether there are any pending abstract acts that have met their prerequisites and selects a qualified act to execute. Once selected, the abstract act undergoes a pre-processing step, where the placeholders for specific plot content contained in the narrative goal are replaced with the actual referred content established and saved in the execution of the previous abstract acts (if any). For example, a narrative goal “*Y saved X from the accident*” might become “*The dove saved the ant from the accident*”.

Following [17], 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. Once generated, the plan reviewer provides feedback regarding the quality and feasibility of the action sequence to improve it. The feedback has three parts:

- **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.
- **Game Environment Evaluation** Similar to [17], feedback comes from executing the generated action sequence in a simulated game environment, and reporting observations on the success/failure of the execution.
- **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.<sup>2</sup>

The plan generator is then prompted to revise the previous action sequence according to the feedback received. The process is repeated for a user-specified number of times, or until the generated plan is executable.

Once the final sequence of character actions is generated, it will be executed by the game environment to update the world state. Meanwhile, it will also be sent to the Placeholder Resolution component (Figure 1), which prompts an LLM to identify specific content referred to by each placeholder from the current act, and store 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.

*Example 2.2.* 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, for example:

```
{
  "subject": "dove",
  "action": "moveTo(oak_tree)",
```

<sup>2</sup>In addition, we also ask an LLM if the narrative goal is already met at the current action. If the answer is yes, the generated plan will be truncated to have the current action as the last action.

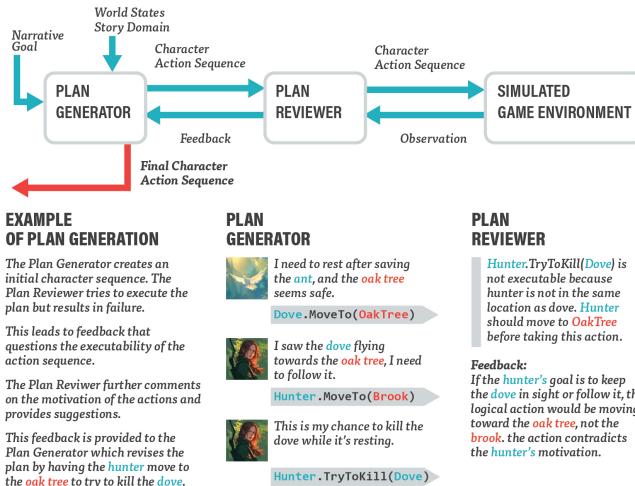


Figure 3: LLM-based narrative planning that iteratively generates and reviews the plan.

```
"thought": "I sense danger!
    I must fly away quickly to avoid the hunter."
}
```

Figure 3 shows an iteration of plan revision for Act 3 as described in Figure 1. The plan was generated using the Ant & Dove story domain from Figure 2.

## 2.4 Implementation

For demonstration purposes, we implemented a basic LLM-based character simulation component similar to a minimal version of *AgentVerse* [3]. Each character is associated with a pre-authored textual description, and a structured memory storage updated upon execution of actions. At each timestep, each character takes an action generated by an LLM. The component can be replaced with a more sophisticated implementation such as [3, 11, 18]. We also implemented a proxy for game environment which simply maintains a list of variables defining a world state, and updates the variables upon execution of character actions. We use prompt chaining techniques to achieve complex tasks by dynamically assembling multiple prompts. Hand-crafted few-shot examples are used in the prompts to increase the quality of outputs. We used gpt-4-0125-preview as the LLM for all the experiments reported in the next section. The maximum number of narrative plan revisions (*maxStep*) was 2.

## 3 EXAMPLES

We describe two examples to demonstrate plot creation workflow using two story domains: *The Ant & Dove* and *The Ville*, which have been previously used in narrative planning [4, 16] and generative agents [11] research. Figure 2 lists the characters, locations, and action schema for these story domains. Both examples are driven by a set of abstract acts based on the theme of "Kindness is never wasted" (Figure 1). Before any acts are executed, an initial round of character simulation sets the stage for the story by simulating the daily lives of the characters (Figure 2 character simulation). The Act

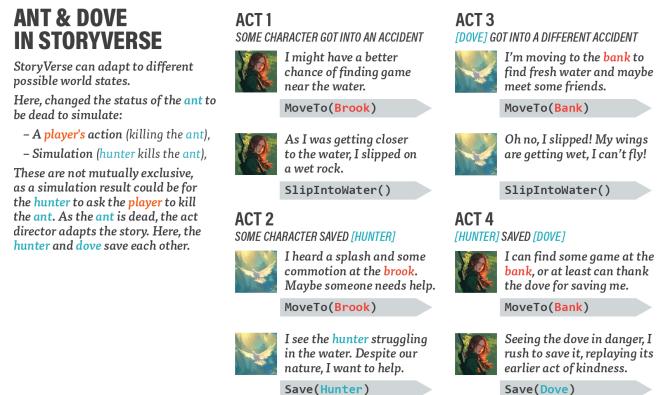


Figure 4: A variation of Ant & Dove story on "A kindness is never wasted" instantiated with the ant dead.

Director then takes the four abstract acts as input and generates the action sequence for each act in both story domains.

*Example 3.1.* In *Ant & Dove* (Figure 2, bottom), the ant (character X) slips into the water in Act 1 while searching for food, which continues its behavior from the character simulation. The dove (character Y) rescues the ant in Act 2. Although the hunter character is not explicitly mentioned in the abstract acts, the narrative planner includes this character as part of the second accident that endangers the dove in Act 3. Finally, the ant saves the dove from the hunter in Act 4, completing the recreation of the original *Ant & Dove* fable. After the four acts conclude, the characters return to their daily routines through the character simulation. The actions performed by the characters in the post-character simulation reflect the events that took place during the four acts.

Additionally, to demonstrate how the story adapts to different world states, we manually change the status of the ant to be dead. As the ant is dead, the Act Director adapts the story to the available characters shown in Figure 4.

*Example 3.2.* We demonstrate the transferability of abstract acts across story domains using the same set of abstract acts in *The Ville*. The story (Figure 2, top) includes three characters: Isabella (cafe owner), Eddy (music student), and Abigail (digital artist). Eddy plays character X, who injures his ankle in an accident (Act 1). Abigail plays character Y, who saves Eddy and offers help (Act 2). Abigail later experiences demotivation after accidentally ruining her creations (Act 3), but Eddy's music inspires her to regain motivation and work on her sketches again (Act 4). Note that Isabella, although introduced as a character, is not selected to participate in the acts due to her busy schedule with the Valentine's Day party at the cafe, which prevents her from leaving and interfering with the story unfolding in an outdoor park.

In our examples, the four acts happen to be executed sequentially with no character simulation in between. In other cases, acts can be executed interlacing with character simulation, likely due to the prerequisites not immediately satisfied after the previous act, such as waiting for a few timesteps of character simulation.

## 4 DISCUSSION

We present our *StoryVerse* system as a proof-of-concept to a plot creation workflow that mediates between a writers' authored content and the emergent behaviors from character simulation. For practical use of this approach in games, further technical aspects need to be considered. 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 [10]. It is worth investigating how to incorporate existing solutions such as increasing the size of LLM's context window [5], utilizing Retrieval-Augmented Generation [7] and adopting a hierarchical generation approach [10] to mitigate this issue. The frequent call to LLMs also raises concerns about delays in the act execution process during gameplay.

Riedl and Bulitko [13] have proposed a taxonomy of interactive narrative systems in terms of three dimensions - authorial intent, character autonomy, and player modeling. Within this design space, *StoryVerse* situates at hybrid authorial intent with a relatively strong level of virtual character autonomy, which has been identified as a gap in [13]. The Act Director serves as an autonomous surrogate for the writer, which has a similar role as the experience manager in [14]. However, an LLM-based approach enables a more flexible and generalizable representation of author's constraints on the plot search space (such as the "island" concept from [14]), and reduces the required knowledge engineering expertise for specifying authorial intents compared to symbolic planning based approaches. The work can be further extended in the player modeling dimension. Our proof-of-concept system simulates the player intervention via updating the world states caused by player actions. Potential Improvement in player modeling could result from incorporating a persona derived from the player's preference and the protagonist's character role. For example, when instantiating abstract acts into concrete plots, involving characters preferred by the player/protagonist would likely enhance the immersive experience.

## 5 CONCLUSION

In this work, we proposed a novel plot creation workflow intended to balance authorial intent and emergent behaviors for a game world with LLM-driven virtual characters. The workflow allows stories to be co-created by the author, the character simulation and the player's actions asynchronously. We present our prototype system *StoryVerse* as a proof of concept to demonstrate the feasibility of proposed workflow with examples showcasing the system's potential. Our next steps include systematic evaluations in terms of standard quantitative metrics for story generation, as well as user studies, for us to understand the performance of the technical approach, and the feasibility and potential of the workflow in a real-world setting. We see *StoryVerse* as an early step towards novel interactive narrative design workflows that leverage LLMs to enable immersive narrative experiences tightly coupled with story and game mechanics.

## REFERENCES

- [1] Inworld AI. 2023. Inworld Origins.
- [2] Ruth Aylett. 1999. Narrative in virtual environments-towards emergent narrative. In *Proceedings of the AAAI fall symposium on narrative intelligence*. USA, 83–86.
- [3] 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).
- [4] 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.
- [5] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169* (2023).
- [6] Michael Lebowitz. 1985. Story-telling as planning and learning. *Poetics* 14, 6 (1985), 483–502.
- [7] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
- [8] Robert McKee. 1997. Story: Substance, structure, style and the principles of screenwriting. 1997. Kent, Great Britain: Methuen (1997).
- [9] James R Meehan. 1977. TALE-SPIN, An Interactive Program that Writes Stories.. In *Ijcai*, Vol. 77, 91–98.
- [10] 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.
- [11] 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.
- [12] Steven Poułakos, Mubbasis Kapadia, Guido M Maiga, Fabio Ziind, 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, Los Angeles, CA, USA, November 15–18, 2016, Proceedings* 9. Springer, 184–196.
- [13] Mark Owen Riedl and Vadim Bulitko. 2013. Interactive narrative: An intelligent systems approach. *Ai Magazine* 34, 1 (2013), 67–67.
- [14] 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.
- [15] Mark O Riedl and Robert Michael Young. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39 (2010), 217–268.
- [16] Chris Martens Rogelio Cardona-Rivera. 2017. Best Practices for Procedural Narrative Generation. <https://gdcvault.com/play/1024143/Procedural-Narrative>
- [17] 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).
- [18] Zhilin Wang, Yu Ying Chiu, and Yu Cheung Chiu. 2023. Humanoid Agents: Platform for Simulating Human-like Generative Agents. *arXiv:2310.05418* [cs.CL]
- [19] Stephen Ware and R Young. 2011. Cpocl: A narrative planner supporting conflict. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 7, 97–102.
- [20] 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.
- [21] 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.