Narrative-to-Scene Generation: An LLM-Driven Pipeline for 2D Game Environments

Yi-Chun Chen[1,†], Arnav Jhala[2]

[1] Yale University, New Haven, CT 06510, US

[2] North Carolina State University, Raleigh, NC 27606, US

Abstract

Recent advances in large language models (LLMs) enable compelling story generation, but connecting narrative text to playable visual environments remains an open challenge in procedural content generation (PCG). We present a lightweight pipeline that transforms short narrative prompts into a sequence of 2D tile-based game scenes, reflecting the temporal structure of stories. Given an LLM-generated narrative, our system identifies three key time frames, extracts spatial predicates in the form of "Object-Relation-Object" triples, and retrieves visual assets using affordance-aware semantic embeddings from the GameTileNet dataset [1]. A layered terrain is generated using Cellular Automata, and objects are placed using spatial rules grounded in the predicate structure. We evaluated our system in ten diverse stories, analyzing tile–object matching, affordance–layer alignment, and spatial constraint satisfaction across frames. This prototype offers a scalable approach to narrative-driven scene generation and lays the foundation for future work on multi-frame continuity, symbolic tracking, and multi-agent coordination in story-centered PCG.

Keywords

Narrative-to-scene generation, Semantic visual grounding, Procedural content generation (PCG), Affordance-aware design, Temporal scene structuring, Large Language Models (LLMs), Explainable content generation

1. Introduction

Large language models (LLMs) can generate rich, coherent narratives from minimal input, creating new opportunities for procedural content generation (PCG) in games that emphasize narrative and visual storytelling. Yet while narrative generation has advanced rapidly, translating stories into structured, spatially grounded game scenes remains an open challenge. Most PCG systems still produce isolated levels or static layouts without modeling temporal structure and continuity—core aspects of narrative experiences that connect space, time, and character action.

Narratives unfold through sequences of events that shape spatial progression and thematic coherence [2, 3]. Building on this idea, we present a method for generating sequences of visual game scenes from short narratives. Each story is segmented into three temporal frames (beginning, middle, end) following narrative-structure conventions such as Freytag's pyramid, capturing key transitions while keeping generation tractable. This segmentation supports partial modeling of story progression and provides a foundation for studying temporal consistency and dynamic storytelling.

Our approach introduces a symbolic-to-visual pipeline that extracts spatial predicates from each frame, maps them to assets in the GameTileNet dataset [1], and arranges them on procedurally generated terrains. Object–relation–object triples are derived through LLM prompting and grounded in visual tiles using semantic and affordance-based filtering. Terrain layers are produced with Cellular Automata to ensure connectivity, while objects are placed through rule-based spatial refinement that satisfies adjacency and containment constraints. The resulting layered 2D scenes reflect the structure of the source narrative while maintaining spatial logic relevant to gameplay.

Joint AIIDE Workshop on Experimental Artificial Intelligence in Games and Intelligent Narrative Technologies, November 10-11, 2025, Edmonton, Canada.

*Corresponding authors.

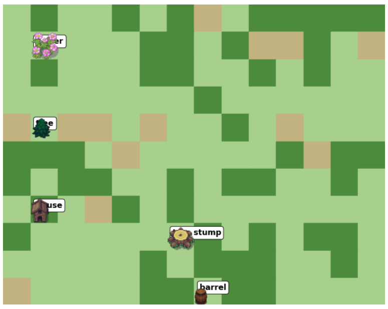
ychen74@alumni.ncsu.edu (Y. Chen); ahjhala@ncsu.edu (A. Jhala)

https://sites.google.com/view/rimichen-web/bio (Y. Chen); https://www.csc.ncsu.edu/people/AHJHALA (A. Jhala)

0009-0003-4035-9894 (Y. Chen); 0000-0003-3874-1720 (A. Jhala)

Our contribution lies not in proposing new generative algorithms but in integrating established PCG components, such as Cellular Automata and semantic matching, into a unified narrative-to-scene pipeline emphasizing temporal segmentation and affordance grounding. This coordination supports cross-frame continuity and balances semantic fidelity with affordance validity, a combination rarely explored in prior PCG work. LLM-generated stories serve only as a reproducible testbed for open-vocabulary narrative input; the pipeline itself is agnostic to text source and can process any authored story.

Table 1
Example of generating a tile-based game scene from narrative descriptions. The system extracts spatial relations and semantic objects (left), selects corresponding tile images from the GameTileNet dataset (middle), and renders a visual scene that satisfies both semantic and spatial constraints (right). Top: labeled object layout. Bottom: rendered tile-mapped scene.

Scene Description	Matched Tiles	Rendered Scene
House below Tree Tree to the right of Barrel Flower above Tree Tree stump to the left of Tree	 House Tree Barrel Flower Tree stump	

We evaluate the system on ten narrative examples, each divided into three frames. Evaluation considers semantic-tile matching, affordance alignment, and spatial-relation satisfaction in the generated scenes. Even with lightweight symbolic rules and open-ended input, the results exhibit coherent spatial organization that reflects narrative intent.

Contributions.

- A unified pipeline that decomposes short narratives into temporally segmented, layered 2D scenes.
- A predicate-extraction and semantic-matching process that grounds narrative objects in tile-based assets using affordance-aware filtering.
- A multi-layer scene synthesis method combining Cellular Automata terrain generation with rule-based spatial placement.
- An empirical analysis of ten narrative examples demonstrating semantic alignment, affordance coherence, and spatial-relation satisfaction across frames.

The proposed system offers a modular foundation for narrative-to-scene generation, emphasizing temporal segmentation, affordance grounding, and semantic alignment. Although the current prototype does not model agent behavior or dynamic state transitions, it establishes a framework extendable to multi-agent coordination and gameplay-aware narrative PCG. Recent work shows that LLMs can segment narrative events in ways that align with human perception [4], supporting our choice to use structured time frames as the basis for visual scene construction. Table 1 illustrates one such example, where spatial constraints and tile affordances jointly produce semantically coherent game scenes from narrative descriptions.

2. Related Work

2.1. Narrative Theory and Structure in Games

Narrative structure in games has been studied through formalist, experiential, and emergent perspectives. Early work examined the tension between linear storytelling and player agency [5, 6]. Later models addressed experiential modes [7] and story architectures that accommodate agency [8]. Other work connected narrative to spatial layout and symbolic environments [9, 10], while surveys summarize recurring patterns in interactive media [11]. These theories provide foundations for incorporating narrative intent into generative pipelines.

2.2. Narrative-Guided Procedural Content Generation

Procedural content generation (PCG) increasingly leverages narrative structure. Quest and story grammars guide goal-oriented generation [12], and language models now produce worlds and events [13, 14]. Emotional arcs and planning constraints improve coherence and engagement [15]. PCG methods span grammars, search-based strategies, and PCGML [16, 17], with reinforcement learning extending control to narrative form [18]. Layered generation has also appeared in visual storytelling and comics [19, 20], demonstrating ways to link symbolic story input with generative models. Prior text-to-scene and PCG systems generate single scenes or levels [21, 22, 16] but rarely model temporal continuity or explicit narrative structure. Our work addresses this gap by connecting narrative segmentation with spatial and affordance reasoning, bridging story progression and playable spatial layouts.

2.3. Tile-Based Generation and Semantic Affordances

Tile-based abstractions remain central to scalable generation. Representative methods include constrained layout via SMT solvers [23], evolutionary search [24], and grammar-based dungeon layouts [25]. Neural approaches introduce embeddings for generalized level generation [26] and segmentation for context-aware tilemaps [27]. Affordance modeling distinguishes environmental, interactive, and collectible roles [28], while corpora enable affordance-aware generation [29]. GameTileNet contributes a dataset of low-resolution tiles annotated with layered affordances [1].

2.4. Semantic Matching and LLM-Guided Grounding

LLMs are increasingly used for scene synthesis and grounding. Prompting and multimodal alignment allow them to act as planners for layouts [30, 31]. Studies show that structured input such as scene semantics improves alignment [32]. Visual-semantic reasoning frameworks for matching and embedding inform techniques for aligning text predicates with tiles [33, 34]. Cognitive studies suggest LLMs approximate human-like segmentation of narrative events, underscoring their potential for symbolic-neural integration [4].

2.5. Symbolic Structures in Visual Storytelling

Symbolic and graph-based representations provide structure for storytelling. Knowledge-enhanced generation has been modeled with narrative graphs or multimodal scene graphs [35, 36, 37]. Other frameworks explore relational encodings to ensure continuity and causality in stories and games [38, 39]. Recent work highlights hierarchical narrative graphs as interpretable intermediaries for multimodal alignment [40]. These approaches underscore the importance of symbolic scaffolds for controllable and interpretable storytelling pipelines.

3. Method: Narrative-to-Scene Generation Pipeline

We propose a structured pipeline that transforms narrative text into visually grounded, tile-based game scenes. As illustrated in Figure 1, the pipeline integrates large language model (LLM)–based story generation with a sequence of symbolic and visual reasoning modules. The process begins with narrative prompting and temporal abstraction, where the story is segmented into key time frames and represented as predicate-style triples. These structured descriptions serve as input for semantic reasoning and symbolic grounding. Although the LLM provides the narrative input, this step serves only to ensure reproducible open-vocabulary test cases; the pipeline itself is model-agnostic and can process any authored story with minimal preprocessing.

Each scene is synthesized through a layered generation process: (1) terrain generation using Cellular Automata ensures navigable base regions; (2) semantic object matching retrieves tile assets aligned with narrative entities based on name, category, and affordance embeddings; and (3) spatial placement with rule-based refinement positions entities according to annotated spatial relations. The final output consists of rendered tile-based scenes and underlying semantic maps, enabling both human-readable visualization and downstream interactive applications.

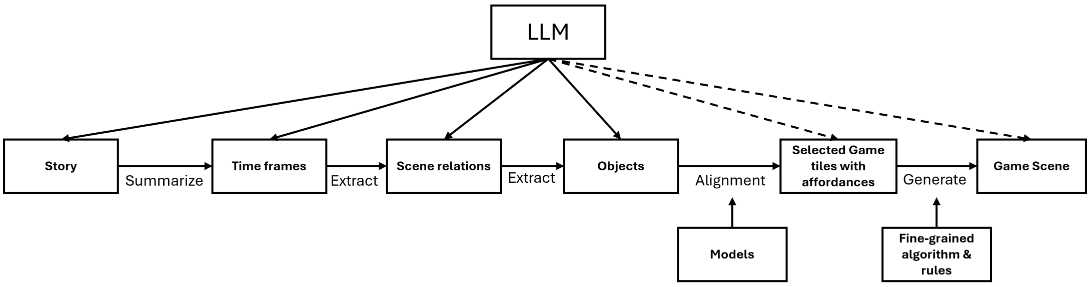


Figure 1: Pipeline for generating a tile-based game scene from an LLM-generated narrative. Modules include narrative parsing, semantic matching, procedural terrain generation, and layered scene rendering.

3.1. Narrative Prompting and Frame Extraction

Inspired by *Word2World* [13], we use a large language model to generate a short adventure story and extract three key narrative moments. Each moment is structured as symbolic predicates describing relationships between story elements, represented as [Object] [Relation] [Object] triples. These structured predicates form the basis for downstream spatial reasoning and visual scene generation.

The process uses two prompts:

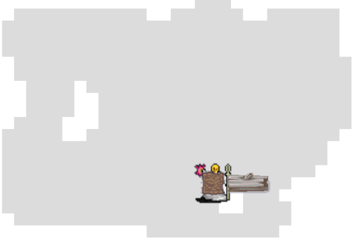
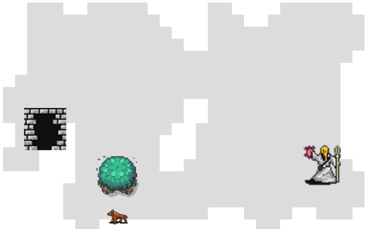
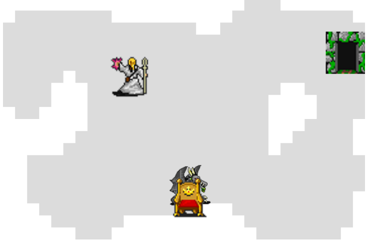
- **Prompt_1:** "Generate a short adventure story (about 100 words)."
- **Prompt_2:** "Extract three key time frames and describe each with [Object] [Relation] [Object] triplets."

Generated Story Example:

> *In the heart of the Enchanted Forest, young Elara discovered an ancient map hidden within a hollow oak. It led her to the legendary Crystal Cavern, rumored to grant the finder a single wish. Braving treacherous paths and wild creatures, Elara reached the cavern's shimmering entrance. Inside, she faced the Guardian, a majestic dragon. With courage and wit, she solved the Guardian's riddle, earning her the wish. Elara wished for peace in her war-torn village. As she exited the cavern, the skies cleared, and harmony was restored, proving that bravery and hope could transform the world.*

Table 2 shows the extracted time frames, the corresponding predicate triples, and the tile-based scenes rendered by our pipeline.

Table 2
LLM-generated story, predicate triples, and corresponding rendered scenes for three narrative time frames.

Time Frame 1	Time Frame 2	Time Frame 3
Elara discovers the ancient map	Elara faces the treacherous paths	Elara meets the Guardian dragon
Hollow oak contains ancient map Elara stands near hollow oak Sunlight filters through forest canopy	Elara walks along rocky path Wild creatures hide behind dense bushes Treacherous paths lead to Crystal Cavern	Crystal Cavern entrance glows with shimmering light Guardian dragon sits atop crystal throne Elara stands before Guardian dragon
		

3.2. Symbolic Spatial Relation Mapping

To transform symbolic narrative structure into spatial configurations, we begin by parsing each scene into a set of predicate triples in the form [Object] [Relation] [Object]. These relations encode spatial intent, such as adjacency or containment, and are mapped to a controlled ontology of canonical spatial actions suitable for tile-based rendering.

This mapping is informed by prior work on relational scene representation [41, 42], which demonstrates that spatial relations such as *above*, *next to*, and *on top of* align with how people describe and interpret visual layouts. For our system, we adopt the following spatial relation types:

- **above / below:** vertical adjacency (Y-axis offset)
- **at left of / at right of:** horizontal adjacency (X-axis offset)
- **on top of:** overlapping placement with layer prioritization

Because natural language varies widely, we use a large language model (LLM) to normalize open-ended expressions to this spatial ontology. For example, *contains* is mapped to *on top of*, while *stands near* may correspond to *at left of* or *at right of* depending on context. Human verification ensures the consistency and interpretability of the mappings. The resulting spatial relations serve as symbolic scaffolds that guide object placement during scene generation.

3.3. Semantic Asset Retrieval with GameTileNet

To align narrative objects with appropriate visual tiles, we adopt a semantic embedding–based retrieval strategy grounded in the GameTileNet dataset [1]. Each tile in the dataset is annotated with structured metadata, including object name, group label, supercategory, and affordance type. These attributes are embedded using the all-MiniLM-L6-v2 Sentence Transformer to construct a searchable index. Narrative objects are encoded using the same model, and tile matches are retrieved via cosine similarity.

Affordance Types. GameTileNet classifies tiles into five affordance types adapted from the Video Game Description Language (VGDL) [43]:

- **Terrain:** Walkable ground surfaces (e.g., grass, stone).
- **Environmental Object:** Static scene elements (e.g., trees, fences).
- **Interactive Object:** Triggerable or functional elements (e.g., doors, levers).

- **Item/Collectible:** Usable or acquirable items (e.g., potions, scrolls).
- **Character/Creature:** Playable or non-playable agents (e.g., goblins, shopkeepers).

Affordance labels serve as soft constraints to improve retrieval robustness, especially for ambiguous cases. For example, the term ”guardian” could refer to a statue or a creature, and the affordance context helps disambiguate the intended match. This retrieval process enables semantic alignment between narrative elements and visual assets while respecting scene composition constraints.

3.4. Terrain and Scene Layout

To render each scene with appropriate environmental context, we infer base terrain types and subregion patches from narrative content. Each scene is structured as a layered grid, with the base layer representing walkable terrain and additional layers corresponding to object affordance types.

3.4.1. Terrain Suggestion via LLM Classification

We use an LLM-based classification step to extract environmental cues from narrative objects. Following story decomposition into predicate triples, each object is assigned:

- **Affordance type:** One of terrain, environmental object, interactive object, item/collectible, or character/creature.
- **Suggested terrain:** A free-text label describing the implied environment (e.g., ”forest”, ”desert”).

These predictions are aggregated to determine dominant terrain types for each scene.

3.4.2. Base and Patch Selection with Continuity Propagation

To ensure continuity across time frames, we assume that scenes with no explicit location change remain in the same environment. Scenes are grouped based on inferred location continuity using temporal adjacency and terrain similarity. For each group:

- The most frequent terrain type is assigned as the **base terrain**.
- Objects with terrain-related labels (e.g., ”path”, ”alley”) are extracted as **patch candidates**.
- Patch terrain decisions are propagated across all scenes within the group.

This process maintains visual and narrative coherence across sequential scenes. The terrain selection process is summarized in Algorithm 1.

Algorithm 1 InferBaseAndPatchTerrain

```
Require: story_scenes: list of scenes with narrative objects
Ensure: base_terrains: map from scene to base terrain label
            patch_terrains: map from scene to list of patch labels
Initialize empty maps: base_terrains, patch_terrains
Group scenes by inferred location continuity
for all scene_group in grouped scenes do
    Initialize frequency_counter
    for all scene in scene_group do
        for all object in scene.objects do
            Predict affordance and suggested terrain using LLM
            if object.affordance == Terrain then
                Increment frequency_counter[object.suggested_terrain]
            else if object.name contains terrain keywords then
                Append object.suggested_terrain to patch_terrains[scene]
            end if
        end for
    end for
    base_terrain ← terrain with max frequency in frequency_counter
    for all scene in scene_group do
        base_terrains[scene] ← base_terrain
    end for
end for
```

3.4.3. Scene Initialization with Cellular Automata

We generate the layout of the base terrain using a Cellular Automata (CA)–based synthesis process. For each scene:

- A connected walkable region is created using CA and verified for reachability.
- Terrain patches are inserted as subregions constrained within the generated base mask.
- The final output is a layered map suitable for narrative-aligned object placement.

This multi-layered scene layout ensures compatibility between narrative framing and spatial structure.

3.5. Spatial Constraint–Driven Object Placement

After terrain generation and semantic matching, each scene is populated by placing narrative-aligned objects within a multi-layer tile grid. This stage is divided into two parts: random initialization and symbolic refinement.

3.5.1. Initial Placement on Walkable Terrain

Objects are first placed randomly on the walkable base terrain using a greedy assignment process. For each object:

- A walkable coordinate is selected from the base terrain mask.
- The object is placed into the corresponding layer based on its affordance (character, item, interactive, or environment).

3.5.2. Spatial Relation–Based Refinement

Once initial placements are made, we apply symbolic spatial constraints derived from narrative predicates. For each predicate of the form [Object A] [Relation] [Object B], we apply the associated spatial transformation to reposition Object A relative to Object B. We define a rule-based adjustment engine to implement these spatial relations. Each relation is translated into a spatial offset and applied iteratively.

Algorithm 2 ApplySpatialRelations

```
Require: scene: object placement layers and predicate relations
Ensure: updated object positions satisfying spatial constraints
for all relation in scene.spatial_relations do
    $(A, R, B)$ ← relation.source, relation.relation, relation.target
    Normalize names of A and B using alias dictionary
    Get current position of B as $(x_b, y_b)$
    Compute new position $(x_a, y_a)$ ← ApplyOffset($x_b, y_b, R$)
    if $(x_a, y_a)$ within bounds and not overlapping then
        Update A's position in its assigned layer
    end if
end for
```

The ApplyOffset function implements fixed spatial transformations:

- **at the left of**: offset $(-3, 0)$
- **at the right of**: offset $(+3, 0)$
- **above**: offset $(0, -3)$
- **below**: offset $(0, +3)$
- **on top of**: overlapping placement (same coordinates but layer shift)

This symbolic-to-spatial grounding process supports interpretable scene composition and lays the foundation for future rule learning or agent-driven placement strategies.

3.6. Knowledge Graph Construction and Narrative Linking

To support symbolic reasoning, we construct optional scene-level knowledge graphs (KGs) derived from parsed narrative predicates. Each KG encodes the symbolic structure of a single story frame, while temporal relations across scenes are captured using a merged KG with ‘precedes‘ edges. These structures facilitate interpretable alignment between visual scenes and underlying narrative logic.

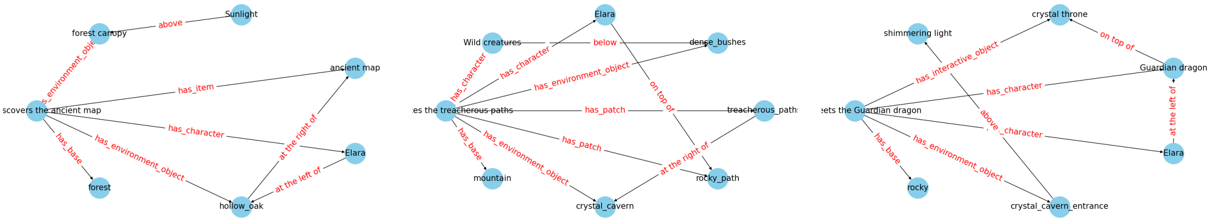
3.6.1. Narrative Knowledge Graphs for Scene Composition

Each story scene is parsed into a set of symbolic predicates, typically in the form of [Subject] [Relation] [Object]. These predicates are transformed into triplets and rendered as a directed symbolic graph. Nodes represent objects and agents, while edges encode spatial or interactional relationships derived from language. Figure 2 shows three examples of such graphs aligned with key scenes.

The knowledge graph for each frame includes:

- **Entities:** Matched visual objects and characters.
- **Relations:** Symbolic spatial predicates (e.g., above, contains) derived from narrative text.
- **Semantic Roles:** Directional links such as agent-action-object when applicable.

These scene-level KGs enable localized symbolic reasoning and enhance traceability in narrative visualization.



(a) Elara discovers the ancient map (b) Elara faces the treacherous paths (c) Elara meets the Guardian dragon

Figure 2: Scene-level knowledge graphs capturing symbolic structure within three narrative frames.

3.6.2. Cross-Frame Temporal Integration

To preserve the overarching story flow, we link each scene-level KG through a unified structure. This *merged knowledge graph* introduces precedes edges to connect scenes according to narrative chronology. These temporal links enable:

- Global queries across multiple scenes.
- Timeline reconstruction.
- Coherence analysis across disconnected predicates.

When no scene boundary is detected in the narrative (i.e., no setting or goal shift), the system assumes the location is continuous. This continuity influences both terrain rendering and the KG linkage.

3.6.3. Relation to Hierarchical Narrative Models

Our construction is inspired by prior work on hierarchical symbolic representation for visual narratives [?]. In that model, events are represented at multiple levels, from panel to event to macro-event, and integrated using graph structures that encode semantic, temporal, and multimodal relations.

Although our current system focuses on symbolic graphs from text and image grounding, its scene-level KG resembles the *event segment* layer in [?]. Both:

- Represent discrete narrative units grounded in key scenes or moments.
- Encode semantic roles and inter-entity relations symbolically.
- Support integration into larger temporal and narrative structures.

While our current scene graphs are simpler, they lay the groundwork for symbolic extensions and integration with event-segment and macro-event abstractions. This future work could enable a unified narrative reasoning framework for both scene generation and story understanding.

3.7. Visual Rendering and Output

After semantic matching and spatial placement, each tile-based scene is rendered using matched 2D sprite assets. Scenes are organized into multi-layer matrices representing different object types (terrain, environment, interactive objects, items, characters), which are composited in semantic order to preserve depth and spatial logic. Object images are resized (e.g., 1.5×), centered within their tiles, and pasted layer by layer to construct the final image. Figure layers are rendered from background to foreground based on their affordance class. We also retain each layer's numerical matrix for downstream tasks such as symbolic reasoning or gameplay simulation. The rendering procedure is outlined in Algorithm 3.

Algorithm 3 RenderSceneImage

```
Require: scene_layers, matched_objects, scene_summaries
Ensure: Saved visual rendering of each scene
for all scene in scene_summaries do
    Initialize canvas with white background
    Draw base terrain using binary mask
    for all layer_type in {environment, interactive, item, character} do
        Get object list and placement matrix
        for all (x, y), object in matrix positions do
            Lookup matched object image
            Resize and center image on tile
            Paste image onto canvas
        end for
    end for
    Save canvas as PNG to output folder
end for
```

4. Evaluation

We evaluated our narrative-to-scene generation system using 10 LLM-generated stories. Each story is segmented into three key time frames, resulting in a total of 30 scene visualizations and their associated symbolic representations. We assess the quality of the generated outputs from multiple perspectives: tile–object semantic alignment, affordance-layer placement correctness, spatial predicate satisfaction, and qualitative renderings.

4.1. Experimental Setup

Each narrative (approximately 100 words) is decomposed into three time frames via prompting. For each frame, we extract three predicate triples, which are matched to GameTileNet assets using semantic embeddings and placed within procedurally generated terrains. Table 3 summarizes the evaluation dataset.

Table 3
Dataset summary.

Stories	Scenes/Story	Entities/Scene	Total Scenes
10	3	4–6	30

4.2. Tile Matching Accuracy

We first examine whether the tiles selected by semantic matching correspond well to the narrative objects. Evaluation considers (1) whether the top-1 matched tile is semantically appropriate, (2) whether the assigned affordance matches the expected gameplay role, and (3) whether the system produces a diverse set of tiles across each story. Per-story results are shown in Table 4, and aggregate results across all 30 scenes are given in Table 5.

Analysis. The results in Table 5 show that semantic alignment between narrative objects and candidate tiles is reliable across stories: cosine similarity values are consistently around 0.40–0.44 with low variance (Table 4). This suggests that narrative embeddings provide a stable signal for selecting visually coherent tiles. By contrast, affordance match rates fluctuate more strongly (0.27–0.55, mean 0.42), indicating that while visual semantics are captured, gameplay functions (e.g., terrain vs. item vs. obstacle)

Table 4
Per-story evaluation results. CosSim: top-1 cosine similarity; Afford: affordance match rate; Div: diversity (1.0 = all unique); Sat: spatial predicate satisfaction rate. Each scene includes 3 predicates on average.

Story	CosSim	Afford	Div	Sat (%)
1	0.43	0.45	0.91	78
2	0.40	0.33	0.87	67
3	0.38	0.55	1.00	67
4	0.41	0.36	1.00	67
5	0.44	0.43	1.00	78
6	0.43	0.45	0.82	89
7	0.37	0.36	1.00	67
8	0.41	0.27	0.82	78
9	0.42	0.50	0.90	78
10	0.44	0.54	0.85	56
Overall	0.41	0.42	0.92	72

Table 5
Aggregate tile matching results (10 stories, 30 scenes).

Metric	Mean	Std. Dev.
Cosine similarity	0.41	0.02
Affordance match	0.42	0.09
Diversity	0.92	0.07

are more difficult to preserve. Tile diversity remains high across all stories (mean 0.92), showing that the system avoids reusing the same assets excessively and maintains scene variety. Error inspection revealed three recurring challenges: inconsistent naming conventions (e.g., "decrepit library" vs. "decrepit_library"), coverage gaps where no suitable tile existed, and affordance misclassifications when a visually similar tile had the wrong role. Overall, these findings highlight the usefulness of semantic matching but also point to the need for stronger affordance-aware retrieval.

4.3. Spatial Predicate Satisfaction

We next examine whether spatial relations (e.g., "Tree to the left of House") are satisfied in the rendered layout. A rule-based checker validates each predicate based on scene matrices, and we compute the percentage of predicates satisfied per scene. Results are shown in Table 4.

Analysis. Predicate satisfaction averaged 72% across all stories (Table 4), with the majority of scenes achieving two-thirds or more of their relational constraints. Stories 6 and 1 achieved the highest consistency (89% and 78%), while Story 10 lagged (56%), typically due to conflicting placement constraints or lack of sufficient map space. These results suggest that the procedural layout is capable of enforcing basic spatial relations but can be brittle when multiple constraints interact. Improvements such as constraint-solving or affordance-informed placement could further enhance satisfaction rates.

5. Discussion

5.1. Strengths and Generalization

The evaluation highlights several promising aspects of our approach. First, semantic alignment between narrative descriptions and visual tiles is stable across all ten stories (Table 5), indicating that embedding-based retrieval provides a reliable foundation for mapping open-ended narrative text into game assets. Second, the system maintains high tile diversity (mean 0.92), suggesting that it can produce varied outputs without excessive repetition, an important property for replayability and player engagement. Third, spatial predicate satisfaction averaged 72% (Table 4), demonstrating that even a lightweight rule-based layout generator can enforce a substantial fraction of narrative constraints. Together, these findings suggest that the pipeline generalizes across different story contexts, making it adaptable for varied game scenarios.

5.2. Limitations

Despite these strengths, several limitations remain. Affordance matching showed high variance (0.27–0.55), revealing a gap between visual similarity and gameplay semantics. This partly stems from limited coverage in the GameTileNet dataset: some objects (e.g., ”lantern,” ”archway”) lack representative tiles, forcing approximate matches. Semantic embeddings also capture descriptive but not functional similarity (e.g., terrain vs. collectible), leading to occasional misplacements. Symbolic layers were not fully leveraged for spatial reasoning, the layout engine checks constraints but does not resolve conflicts, causing brittle performance when multiple predicates interact. Finally, narrative-to-scene generation is inherently open-ended, complicating evaluation. Metrics such as diversity and cosine similarity depend not only on correct object interpretation but also on tile coverage and the ambiguity of natural language descriptions. The current three-frame segmentation follows a fixed narrative template; future versions will explore data-driven segmentation that adapts frame count to story complexity. Ablation studies on individual modules (terrain generation, semantic matching, spatial refinement) were not conducted; future work will quantify their contributions to overall scene coherence.

5.3. Use Cases and Integration in Game Tools

Despite these challenges, the framework has several promising use cases. For game developers, the system can serve as a prototyping tool, quickly transforming narrative prompts into playable scene sketches that can be refined by designers. For procedural content generation (PCG) research, it offers a testbed that integrates symbolic reasoning, semantic matching, and spatial layout, enabling controlled experiments on hybrid generation pipelines. Integration into existing game engines such as Unity or Godot could extend the system into interactive editors, where designers specify narrative beats and receive automatically generated candidate scenes. Beyond development, the pipeline may also support applications in game-based storytelling, educational games, or automated testing of narrative scenarios.

Overall, the results suggest that narrative-driven PCG is feasible, but requires a deeper integration of affordance-aware retrieval and constraint-solving methods to bridge the gap between narrative semantics and functional game design.

6. Conclusion

We presented a pipeline for generating game scenes from narrative text by aligning LLM-derived predicates with the GameTileNet dataset and rendering layered maps using procedural terrain generation. Our evaluation across ten stories demonstrated that semantic matching provides stable visual alignment with narrative objects, while affordance alignment and spatial relation enforcement remain challenging. These findings highlight both the promise of semantic embeddings for bridging text and assets and the need for deeper affordance-aware reasoning to ensure gameplay consistency. This work provides an early step toward narrative-driven procedural content generation. Future directions include integrating symbolic reasoning for more reliable spatial and temporal coordination, expanding the coverage of tile datasets to reduce gaps in representation, and supporting interactive or co-creative workflows where designers and players can iteratively refine generated scenes. We see these developments as important next steps toward practical tools that blend narrative expression with playable game environments.

Declaration on Generative AI

Generative AI tools were used only to assist with language refinement and LaTeX formatting under the authors' direction. All conceptual contributions, design, implementation, and analysis were produced by the authors.

References

[1] Y.-C. Chen, A. Jhala, Gametilenet: A semantic dataset for low-resolution game art in procedural content generation, arXiv preprint arXiv:2507.02941 (2025).

[2] S. McCloud, Understanding Comics: The Invisible Art, Tundra Publishing, 1993. Reprinted by HarperCollins in 1994.

[3] N. Cohn, Visual narrative structure, Cognitive science 37 (2013) 413–452.

[4] S. Michelmann, M. Kumar, K. A. Norman, M. Toneva, Large language models can segment narrative events similarly to humans, Behavior Research Methods 57 (2025) 1–13.

[5] E. Aarseth, A narrative theory of games, in: Proceedings of the international conference on the foundations of digital games, 2012, pp. 129–133.

[6] J. Juul, Games telling stories, Handbook of computer game studies (2005) 219–226.

[7] G. Calleja, Experiential narrative in game environments (2009).

[8] S. Domsch, Storyplaying: Agency and narrative in video games, De Gruyter, 2013.

[9] S. Domsch, Space and narrative in computer games, Ludotopia: Spaces, places and territories in computer games (2019) 103–123.

[10] C. A. Lindley, Story and narrative structures in computer games, Bushoff, Brunhild. ed (2005).

[11] H. Koenitz, Narrative in video games, in: Encyclopedia of computer graphics and games, Springer, 2024, pp. 1230–1238.

[12] J. Howard, Quests: Design, theory, and history in games and narratives, AK Peters/CRC Press, 2022.

[13] M. U. Nasir, S. James, J. Togelius, Word2world: Generating stories and worlds through large language models, arXiv preprint arXiv:2405.06686 (2024).

[14] G. Todd, A. G. Padula, M. Stephenson, É. Piette, D. J. Soemers, J. Togelius, Gavel: Generating games via evolution and language models, Advances in Neural Information Processing Systems 37 (2024) 110723–110745.

[15] C. Miller, M. Dighe, C. Martens, A. Jhala, Stories of the town: balancing character autonomy and coherent narrative in procedurally generated worlds, in: Proceedings of the 14th International Conference on the Foundations of Digital Games, 2019, pp. 1–9.

[16] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, J. Togelius, Procedural content generation via machine learning (pcgml), IEEE Transactions on Games 10 (2018) 257–270.

[17] G. N. Yannakakis, J. Togelius, Procedural content generation by content type, in: Artificial Intelligence and Games, Springer, 2025, pp. 287–312.

[18] J. Togelius, G. N. Yannakakis, K. O. Stanley, C. Browne, Search-based procedural content generation, in: Applications of Evolutionary Computation: EvoApplicatons 2010: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Istanbul, Turkey, April 7-9, 2010, Proceedings, Part I, Springer, 2010, pp. 141–150.

[19] Y.-C. Chen, A. Jhala, Collaborative comic generation: Integrating visual narrative theories with AI models for enhanced creativity, in: Proceedings of the 3rd Workshop on Artificial Intelligence and Creativity, volume 3810, 2024.

[20] Y.-C. Chen, A. Jhala, A customizable generator for comic-style visual narrative, arXiv preprint arXiv:2401.02863 (2023).

[21] M. O. Riedl, R. M. Young, Narrative planning: Balancing plot and character, Journal of Artificial Intelligence Research 39 (2010) 217–268.

[22] M. Guzdial, M. O. Riedl, Combinatorial creativity for procedural content generation via machine learning., in: AAAI Workshops, 2018, pp. 557–564.

[23] J. Whitehead, Spatial layout of procedural dungeons using linear constraints and smt solvers, in: Proceedings of the 15th International Conference on the Foundations of Digital Games, 2020, pp. 1–9.

[24] A. Petrovas, R. Bausys, Procedural video game scene generation by genetic and neutrosophic waspas algorithms, Applied Sciences 12 (2022) 772.

[25] H. Jiang, S. Wang, H. Bi, X. Lv, B. Zhao, Z. Wang, Z. Wang, Synthesizing indoor scene layouts in complicated architecture using dynamic convolution networks, Proceedings of the ACM on Computer Graphics and Interactive Techniques 4 (2021) 1–16.

[26] M. Jadhav, M. Guzdial, Tile embedding: a general representation for level generation, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 17, 2021, pp. 34–41.

[27] L. Gabriel, E. W. Clua, A semantic segmentation system for generating context-based tile-maps, in: Proceedings of the 22nd Brazilian Symposium on Games and Digital Entertainment, 2023, pp. 124–133.

[28] R. Hunicke, M. LeBlanc, R. Zubek, et al., Mda: A formal approach to game design and game research, in: Proceedings of the AAAI Workshop on Challenges in Game AI, volume 4, San Jose, CA, 2004, p. 1722.

[29] G. R. Bentley, J. C. Osborn, The videogame affordances corpus, in: 2019 Experimental AI in Games Workshop, 2019.

[30] R. Volum, S. Rao, M. Xu, G. DesGarennes, C. Brockett, B. Van Durme, O. Deng, A. Malhotra, W. B. Dolan, Craft an iron sword: Dynamically generating interactive game characters by prompting large language models tuned on code, in: Proceedings of the 3rd Wordplay: When Language Meets Games Workshop (Wordplay 2022), 2022, pp. 25–43.

[31] M. Zhou, Y. Wang, J. Hou, C. Luo, Z. Zhang, J. Peng, Scenex: Procedural controllable large-scale scene generation via large-language models, arXiv preprint arXiv:2403.15698 (2024).

[32] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, L. Sun, A comprehensive survey of ai-generated content: A history of generative ai from gan to chatgpt, arXiv preprint arXiv:2303.04226 (2023).

[33] K. Li, Y. Zhang, K. Li, Y. Li, Y. Fu, Visual semantic reasoning for image-text matching, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 4654–4662.

[34] K. Li, Y. Zhang, K. Li, Y. Li, Y. Fu, Image-text embedding learning via visual and textual semantic reasoning, IEEE transactions on pattern analysis and machine intelligence 45 (2022) 641–656.

[35] C.-C. Hsu, Z.-Y. Chen, C.-Y. Hsu, C.-C. Li, T.-Y. Lin, T.-H. Huang, L.-W. Ku, Knowledge-enriched visual storytelling, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 7952–7960.

[36] C. Xu, M. Yang, C. Li, Y. Shen, X. Ao, R. Xu, Imagine, reason and write: Visual storytelling with graph knowledge and relational reasoning, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 3022–3029.

[37] A. Mishra, A. Laha, K. Sankaranarayanan, P. Jain, S. Krishnan, Storytelling from structured data and knowledge graphs: An nlg perspective, in: Proceedings of the 57th annual meeting of the association for computational linguistics: Tutorial Abstracts, 2019, pp. 43–48.

[38] R. E. C. Rivera, A. Jhala, J. Porteous, R. M. Young, The story so far on narrative planning, in: Proceedings of the International Conference on Automated Planning and Scheduling, volume 34, 2024, pp. 489–499.

[39] T. Akimoto, Computational modeling of narrative structure: A hierarchical graph model for multidimensional narrative structure, International Journal of Computational Linguistics Research 8 (2017) 92–108.

[40] Y.-C. Chen, Structured graph representations for visual narrative reasoning: A hierarchical framework for comics, arXiv preprint arXiv:2506.10008 (2025).

[41] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al., Visual genome: Connecting language and vision using crowdsourced dense image

annotations, International journal of computer vision 123 (2017) 32–73.

[42] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, R. Girshick, Clevr: A diagnostic dataset for compositional language and elementary visual reasoning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2901–2910.

[43] T. Schaul, A video game description language for model-based or interactive learning, in: 2013 IEEE Conference on Computational Inteligence in Games (CIG), IEEE, 2013, pp. 1–8.

A. Online Resources

The Code and examples are released publicly via https://github.com/RimiChen/2025_NarrativeScene.

B. Appendix: Sample Stories

> *Amid the neon glow of New York's restless night, journalist Jake stumbled upon a cryptic note tucked inside a trash can under a flickering streetlight. The message hinted at a hidden treasure buried deep within the city's shadows. Pursued by ruthless gangsters, Jake raced through winding alleys bathed in moonlight, his every step echoing with danger. Clue after clue led him skyward—up spiral stairs and secret elevators—until he reached the crown of the Statue of Liberty. There, hidden beneath the cold iron floor, he uncovered the treasure. As dawn broke over the skyline, Jake realized he'd rewritten the city's secrets.*

> *In the neon-lit sprawl of Neo-Tokyo, young hacker Kenji uncovered a cache of encrypted files hidden in a derelict mainframe. Dust swirled in the glow of failing lights as lines of forgotten code blinked to life. Back in his cramped apartment, Kenji hunched over his terminal, fingers flying across the keyboard. The screen flooded with cascading symbols, then froze—decoding complete. What emerged was more than data; it was evidence of a vast corporate conspiracy. As city skyscrapers loomed outside his window, Kenji stared at the truth pulsing on his screen, knowing his next move could shake the world's digital core.*

> *In a crumbling library beneath a sagging ceiling, Iris unearthed a fragile message sealed in dust and time. The note whispered of a hidden oasis—a refuge in the desolate ruins of the world. With resolve burning in her chest, she ventured into the scorched wastelands, where mutant creatures prowled and the earth cracked beneath her feet. Days blurred into nights, but Iris pressed on. At the edge of collapse, she found it: a shimmering oasis blooming defiantly in the dead soil. As its waters sparkled with life, humanity's hope rekindled. Iris hadn't just survived—she had rediscovered a future.*

> *Inside the public library, Alex stood near a dusty bookshelf, where an old book rested untouched. Opening it, he found it contained an encrypted note tucked between yellowed pages. That night, at home, Alex held the encrypted note over his desk, where moonlight illuminated the surface. The desk supported scattered papers, maps, and scribbled codes. After cracking the message, he followed its coordinates to an abandoned warehouse. There, Alex stood in the dim space, heart pounding. Streetlights outside cast shadows on the warehouse exterior. Suddenly, figures emerged—the secret society gathered around Alex, their eyes fixed on the note he still held.*

> *In the rain-slicked alleys of Zephyr, streetwise Jax knelt by a loose cobblestone glowing faintly beneath the streetlight's shimmer. Beneath it, he uncovered an emerald amulet pulsing with forgotten energy. Clutching it in his hand, a surge of ancient power surged through him, surrounding him in a halo of green fire. The city trembled. Above the skyline, a sinister sorcerer descended from the clouds. Jax stood his ground, the amulet blazing with newfound strength. Magic clashed in the sky, old and new. When the light faded, only Jax remained—victorious and forever changed by the artifact he had unearthed from the street.*

Aboard the cursed ship The Sea Serpent, Captain Redbeard peered into the abyss, where the ocean floor cradled a glowing artifact of unknown origin. As the ship rocked atop the waves, he hauled it aboard, sensing the tide of fate shift. From the deep, monstrous shapes surged—leviathans with fangs like anchors. Redbeard stood firm, the artifact blazing in his grip as he battled the beasts. When the sea fell still, the relic rose, casting a vision across the sky: an ancient prophecy long forgotten. As its light danced across the waves, Redbeard knew the sea had chosen its next legend.

Amid a raging storm, Captain Jack stood firm on the deck, waves crashing around him. The sea surrounded his ship, The Stormcaller, as he gripped a cryptic compass unearthed from a sailor's tale. Its needle trembled, pointing unerringly toward the fabled El Dorado. Navigating through the Bermuda Triangle, Jack's vessel braved merciless waves while mystical sea creatures lunged from the depths. His crew fought with steel and fear. At last, the storm broke. Under radiant moonlight, the horizon cleared—revealing golden spires glistening in the distance. Standing on the drenched deck, Jack watched El Dorado rise from myth into reality.

In the blistering Sahara Desert, where the sun beats down on endless dunes, Dr. Samuel Cross knelt near an unearthed ancient amulet glinting in the sand. The Sahara Desert contained more than secrets—it held the path to legend. As he journeyed onward, a violent sandstorm engulfed Cross, tearing visibility to shreds. Through the storm, he read cryptic hieroglyphs etched in stone, while tomb raiders pursued him relentlessly. Deeper inside the pyramid, Cross stood in a hidden chamber—an elaborate pyramid trap that contained deadly mechanisms. Clutching the fabled treasure, Cross narrowly escaped, leaving behind danger but carrying history in his hands.

Deep within the abyss of the Forgotten City, intrepid archaeologist Dr. Alexander unearthed a mystical artifact concealed in an ancient tomb. Torchlight flickered across the ruins as he knelt beside the stone sarcophagus, uncovering secrets long buried. With the artifact in hand, he pressed forward, navigating a labyrinth of treacherous traps and walls that whispered forgotten chants. Guided by the artifact's glow, he reached the heart of the tomb. There, the Guardian emerged, its form towering in silence. As chamber doors slammed shut, the artifact shimmered and activated a hidden mechanism. The Guardian bowed. Dr. Alexander had passed the test—and earned the city's truth.

In the blazing Sahara, golden sunlight beamed on the golden amulet buried just beneath the sand. Amelia stood over the buried amulet, brushing away grains until it shimmered in full. She picked it up. As she followed the amulet's glow across the desert, sand dunes surrounded her. A venomous creature lurked behind one of the dunes, but she pressed on. Soon, she arrived at a hidden pyramid. Amelia stood before the ancient pyramid, awed by its size. With steady hands, she raised the amulet. It fit perfectly into the pyramid's lock. Inside, the hidden pyramid held civilization's long-lost secrets.