

IVIE: A Neuro-symbolic Approach to Incremental and Validated Generation of Interactive Fiction Worlds

Micaela Vaucher † Santiago Silveira Santiago Góngora † Luis Chiruzzo

Instituto de Computación, Facultad de Ingeniería
Universidad de la República
Uruguay

† {micaela.vaucher, sgongora}@fing.edu.uy

Abstract

Computational creativity in Interactive fiction faces a fundamental tension: Large Language Models (LLM) may produce creative narratives but struggle with world coherence, while symbolic systems ensure consistency but lack creative flexibility. We present *IVIE* (Incremental & Validated Interactive Experiences), a neuro-symbolic approach to generating complete and playable interactive fiction worlds from scratch. Building upon PAYADOR’s neuro-symbolic framework, *IVIE* implements a four-stage incremental generation pipeline that delegates creative decisions—setting and character creation, puzzle design—to LLMs while grounding the world state through symbolic validation. The system generates worlds with interconnected locations, functional items, non-player characters, and coherent puzzles, all structured around a central goal-oriented architecture. Human evaluation shows the approach generates immersive, thematically coherent worlds with high player engagement. Results seem to indicate that the neuro-symbolic approach successfully balances flexibility with narrative coherence: symbolic validation grounds LLM generation without eliminating generative freedom. However, challenges remain: LLM inconsistencies occasionally bypass puzzle constraints, and objective validation gaps allow some structurally impossible goals. We identify key design considerations for future neurosymbolic interactive storytelling systems, particularly regarding LLM capabilities and their limitations.

Introduction

Large Language Models can generate fluid narratives on demand, but they constantly fail when asked to maintain a coherent game world across just a few of player turns. Objects vanish and reappear arbitrarily, NPCs forget previous conversations, spatial relationships become contradictory, and puzzle solutions change mid-game (Góngora et al. 2024b). Yet the creative potential remains tantalizing: if we could ground LLM generation in symbolic validation, we could significantly advance the generation of consistent interactive narratives.

Interactive fiction (IF), the genre of text-based games in which users explore fictional worlds and influence the narrative through natural-language commands (Riedl and Bulitko

2012), exemplifies this tension between creative flexibility and structural coherence. Classic systems like *Zork* (Montfort 2003) maintained the world state through hand-crafted objects, locations, and deterministic rules, ensuring consistency but requiring extensive manual authoring for each world. Procedural content generation (PCG) approaches address this authoring bottleneck by algorithmically creating game content (Shaker, Togelius, and Nelson 2016), yet traditional methods for narratives rely on symbolic planning systems that produce rigid, template-based stories lacking creative richness (Riedl 2015).

LLMs promise a form of computational improvisation, generating fluid narratives on the fly without predetermined scripts. Yet improvisation is not mere spontaneity — it requires technique, memory, and contextual grounding (Schiffini 2006; Sawyer 2000), precisely what LLMs lack for extended interactions. They hallucinate objects, forget established facts, and lose track of game state (Castricato et al. 2021; Ji et al. 2023). A golden key introduced on turn 5 may be forgotten by turn 25, or worse, the LLM may contradict whether the door was ever locked. For players seeking coherent experiences rather than surreal explorations, such inconsistencies fundamentally break immersion.

Recent neuro-symbolic approaches (Góngora et al. 2024a; Sarker et al. 2021) address this by separating concerns: LLMs handle creative generation while symbolic structures maintain the world state. Systems like PAYADOR (Góngora et al. 2024a) suggest that this works effectively for *playing* in predefined worlds. Extending these architectures to *automatically generate* complete worlds would enable rapid prototyping and procedural variation while maintaining the structural coherence that manual authoring provides.

In this paper, we present *IVIE* (Incremental & Validated Interactive Experiences), a neuro-symbolic system that generates complete, playable IF worlds automatically. While PAYADOR seemed to indicate that neuro-symbolic architectures can maintain coherence during gameplay, it operated exclusively on manually predefined worlds. *IVIE* extends this framework to world generation itself: rather than hand-crafting each world, *IVIE*’s four-stage incremental pipeline delegates creative decisions—setting and character creation, and puzzle design—to LLMs, while a symbolic layer enforces spatial connectivity, type correctness, and objective

solvability at each stage. The source code is available on GitHub¹.

Related Work

IF systems face a fundamental challenge: balancing creative narrative flexibility with structural world coherence. Classic parser-based systems like *Zork* (Montfort 2003) solved this through manual authoring: every object, location, and interaction was hand-coded, ensuring consistency but requiring extensive development effort for each world. Modern approaches attempt to escape this authoring bottleneck through different strategies, each with critical limitations.

Systems like AI Dungeon (Trapova 2023) pioneered the idea that LLMs can generate contextually appropriate responses to arbitrary player actions. By concatenating player input with conversation history and generating continuations via LLMs (e.g., GPT-3), these systems enable unprecedented narrative flexibility. However, extensive user testing revealed critical failures (Yong and Mitchell 2023): objects vanish without justification and reappear later, NPCs contradict previous statements, game rules become inconsistent as context windows fill, and the model exhibits “looping” behavior. These issues stem from LLMs’ stateless nature—they maintain no persistent world model (Castricato et al. 2021), relying solely on probabilistic patterns without mechanisms to enforce structural coherence.

Recent work bridges symbolic and neural paradigms in different ways. Some approaches retrieve or maintain narrative context to improve coherence (Chambers et al. 2024), or augment game content through data-driven generation (Callison-Burch et al. 2024), yet these still lack the symbolic grounding necessary to guarantee structural validity. Others delegate creative generation to LLMs while maintaining a structured world state (Sarker et al. 2021), combining planning with language models (Kelly et al. 2023; Zhou et al. 2025) or constructing spatial representations dynamically (Li et al. 2025), but focus on playing or narrating within pre-existing worlds rather than generating complete worlds from scratch.

Our current work IVIE builds directly upon PAYADOR (Góngora et al. 2024a; 2026), a neuro-symbolic framework that separates symbolic world management from neural narrative generation. PAYADOR implements an object-oriented world model (Location, Item, Character, Puzzle, Objective) and employs dual LLMs: a reasoning model interprets player actions to generate structured state transformations, while a narrative model produces scene descriptions (Góngora 2025). World coherence is maintained through a continuous grounding cycle: when a player inputs an action, the system renders the current symbolic world state as natural language; the reasoning LLM predicts its effects, generating both symbolic transformations (e.g., moving an item, unlocking a passage) and narrative outcomes; these predictions are validated before being applied to update the world state; finally, the narrative LLM generates scene descriptions based on the updated state. This separation appeared effective

at preventing LLM inconsistencies: spatial connectivity is verified through explicit graph structures, object locations are tracked deterministically, and puzzle states persist until explicitly modified. However, these systems operate on manually predefined worlds where every location, object, NPC, and puzzle is hand-coded before gameplay.

The IVIE Approach

IVIE explores the aforementioned limitation through incremental validated generation: a four-stage pipeline that progressively constructs worlds from abstract concepts to concrete playable experiences, with symbolic validation at each stage ensuring spatial connectivity, type correctness, and objective solvability. Given that LLMs can generate narrative text but struggle with structural coherence, while symbolic systems ensure consistency but lack generative flexibility, the central question becomes: *how do we harness both?*

To achieve this separation of concerns, we build upon PAYADOR’s neuro-symbolic framework (Góngora et al. 2024a; 2026) as our architectural backbone. As described previously, PAYADOR maintains world coherence during gameplay through symbolic world state and dual-LLM architecture. Building on this framework, IVIE explores the generation of complete worlds automatically, rather than playing within predefined ones.

To achieve this separation of concerns, we build upon PAYADOR’s neuro-symbolic framework (Góngora et al. 2024a; 2026) as our architectural backbone. Building on this framework, IVIE explores the generation of complete worlds automatically, rather than playing within predefined ones.

IVIE’s symbolic component acts as a structured execution and validation layer rather than a creative reasoning engine: it does not generate narrative content, but executes and enforces structural correctness on LLM-proposed state transformations. Rather than asking LLMs to generate creative content and simultaneously maintain world consistency—a task where they demonstrably fail (Góngora et al. 2024a)—we aim to assign each component what it does best:

- **LLMs generate narrative elements:** location descriptions, character backstories, puzzle designs, and objective framing.
- **Symbolic Python structures maintain world state:** following PAYADOR’s object-oriented world model; which locations connect to which, what items exist where, which puzzles block which passages, and whether the objective is solvable given the current world configuration.

So, the challenge is to generate worlds that are not only creative and thematically coherent, but also structurally valid and where players can complete the objective.

Goal-Oriented Architecture

PAYADOR worlds include objectives that players must complete, like finding items, or reaching locations. We recognized this concept could serve as the organizing principle for IVIE’s generation pipeline: rather than generating disconnected elements and hoping they form a coherent world,

¹<https://github.com/micavelavaucher/IVIE>

Generated World Opening

Crumbling brick buildings line the street. Fog rolls between the structures, and distant sounds echo through empty alleys. You are a detective investigating mysterious disappearances in the city.

Current Location: Abandoned Warehouse District

Accessible places: Police Station, Old Docks

Blocked passages: Evidence Room

Objective: Find the Missing Person

Items: Flashlight, Detective’s notebook

NPC: Officer Martinez

Figure 1: Example world opening generated in Generate mode, showing goal-oriented structure where all elements serve the central objective.

the objective defines what must exist and how components relate. Every location, item, NPC, and puzzle exists to serve the player’s goal, whether finding a hidden artifact, solving a mystery, or escaping a dangerous location. This architecture addresses common PCG challenges by providing the structural constraints necessary to avoid unplayable content (Bomström 2016) while maintaining coherent overarching structures across generated elements (Khaled, Nelson, and Barr 2013).

Every generated element exists to serve a central purpose to complete the objective. This objective defines world requirements: If the goal is “Deliver the ancient scroll to the librarian”, the world must contain the scroll as a gettable item, a librarian character positioned in some location, a path connecting the scroll’s location to the librarian’s location, and potentially puzzles blocking access to either element. Therefore, the generation pipeline works backwards from this objective, determining what elements does this goal require and where should they be placed. Figure 1 shows a generated world opening where all elements are oriented toward the central objective of finding a missing person.

Incremental Generation Pipeline

In preliminary experiments, we found that attempting to generate complete worlds in a single LLM call resulted in inconsistencies, as models struggled to maintain coherence across dozens of interdependent elements. Additionally, LLMs would generate worlds of unpredictable sizes, making systematic evaluation and comparison difficult. To address both issues, we designed a four-stage **incremental generation** pipeline (illustrated in Figure 2) with configurable size parameters. IVIE allows designers to specify the number of `Locations`, `Items`, `Characters`, and `Puzzles` before generation begins, ensuring consistent world scale for controlled evaluation while enabling validation systems to scale appropriately. The pipeline builds worlds progressively across four mandatory stages, from abstract to concrete, with symbolic validation at each boundary. Each stage uses a specialized prompt to guide LLM generation; these prompts are available in the source code repository.

Stage 1: Adventure Core This first stage establishes the narrative foundation by generating three core components:

world theme, protagonist description, and objective type.

IVIE extends PAYADOR’s objective system (Góngora 2025) with five objective types, each imposing different structural requirements on subsequent stages:

- `REACH_LOCATION`: Player must navigate to a specific location (e.g., “Escape the haunted mansion”)
- `GET_ITEM`: Player must obtain a particular item (e.g., “Retrieve the ancient artifact”)
- `DELIVER_ITEM`: Player must bring an item to a character or location (e.g., “Deliver the scroll to the librarian”)
- `FIND_CHARACTER`: Player must locate a specific NPC (e.g., “Find the missing explorer”)
- `SOLVE_MYSTERY`: Player must gather clues to solve a mystery (e.g., “Discover who stole the crown jewels”)

These components can be generated in two modes. In *Inspiration mode*, the LLM receives user-provided thematic guidance (e.g., “mystery in a Victorian mansion”) and generates a world aligned with that theme. In *Generate mode*, the LLM fully takes creative responsibility (Colton and Wiggins 2012), autonomously inventing theme, setting, and narrative premise without user constraints.

Output: `WorldConcept` object with `adventure_title`, `narrative_backstory`, `player_concept` (character role), and `main_objective` (high-level goal description). The formal objective type is determined in subsequent stages.

Validation: This stage does not require validation beyond schema correctness, as the conceptual elements (title, backstory, player concept, objective description) are `Strings` (purely narrative) and do not impose structural constraints that could render the world unplayable. Any thematically coherent concept is acceptable as input for subsequent stages.

Stage 2: World Structure Once the narrative core is established, this stage identifies the main entities necessary to complete the objective: `key_locations`, `relevant_characters`, and `required_objects`. The LLM generates entity identifiers (names) and brief statements of how each relates to the objective, but without spatial connections, descriptions, or properties. For instance, it might specify “Ancient Library” and “Hidden Scroll” as entities, noting the scroll must be found in the library in its description, but not describe what the library looks like, where it connects to, or whether the scroll is locked away. This generated structure works as a world structure to be completed later. In preliminary experiments, we observed that the LLM tended to generate thematically coherent but functionally irrelevant entities (e.g., decorative furniture, background NPCs with no gameplay purpose). To mitigate this, we modified the generation schema to require a `relevance_to_objective` field for each entity, forcing the LLM to justify its functional role explicitly. This ensures every generated element serves the player’s path towards objective completion.

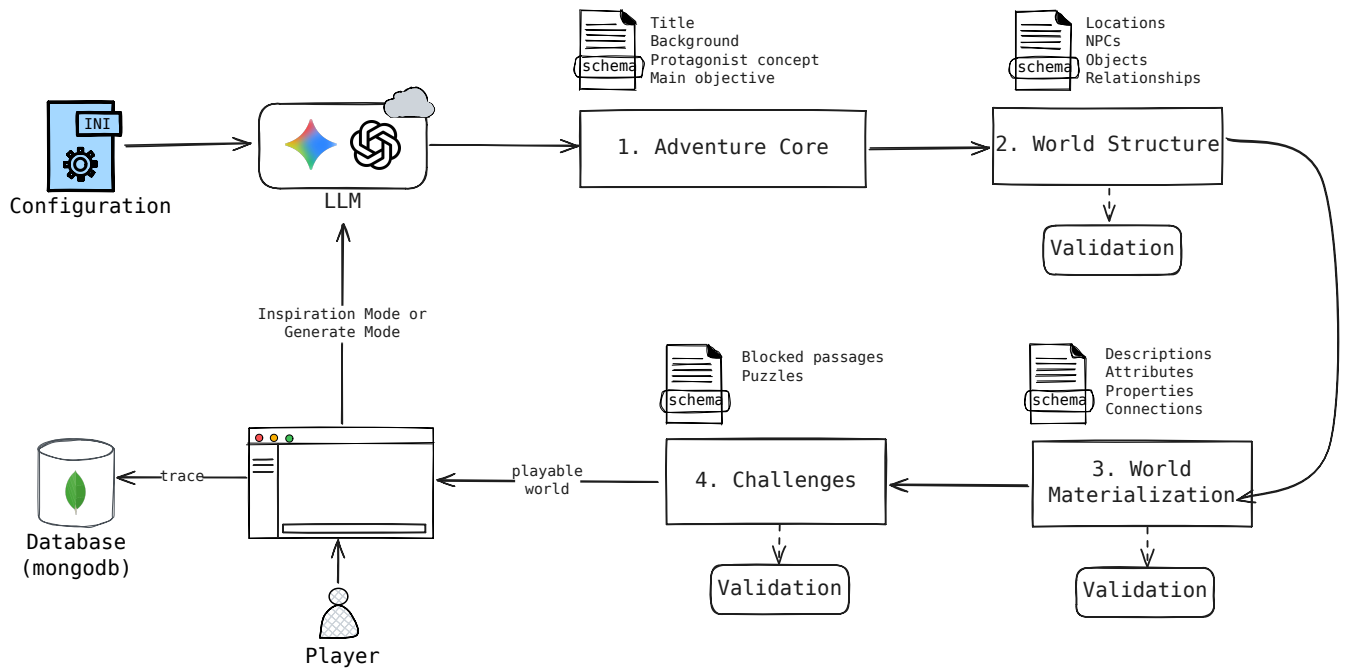


Figure 2: IVIE’s incremental generation pipeline. Starting from optional user inspiration, the LLM generates worlds progressively across four mandatory stages (Adventure Core → World Structure → World Materialization → Challenges), each with validation gates.

Output: GeneratedWorld object containing abstract elements (Locations, Characters and Items without details).

Validation: Schema validation (Pydantic) ensures the output conforms to the expected data structure with correct types and mandatory fields. Size verification confirms if the LLM generated the configured number of elements.

Stage 3: World Materialization This stage completes the world structure from Stage 2 by populating it with the attribute values required by PAYADOR’s world model: Locations receive textual descriptions and spatial connections, Characters are assigned descriptions and starting locations, Items are given names and gettability flags, and passages between Locations are established. With entity identifiers already established in Stage 2, the prompt becomes significantly more specific: it explicitly assigns positions to all Characters and Items, and instructs the model not to introduce new entities beyond the existing world structure. These constraints address the previously mentioned problem of thematically coherent but functionally disconnected elements, ensuring every entity has a defined spatial position and all Locations are interconnected.

Mandatory connection rules ensure spatial coherence: (1) if location A connects to B, then B must connect to A (avoiding accidental unidirectional connections that could trap the player), and (2) there must be no isolated locations. The objective must be completable with the created elements, with at least one logical route from the

initial state to its achievement.

Output: Fully instantiated playable GeneratedWorld object.

Validation: Spatial connectivity is verified via depth-first search (DFS), ensuring every Location is reachable from the starting location. Objective completable checks vary by type:

- GET_ITEM: Verifies the target item is both reachable and marked as gettable.
- REACH_LOCATION: Verifies the destination is accessible from the start.
- DELIVER_ITEM: Verifies both the item and the recipient are reachable and the item is obtainable.
- FIND_CHARACTER: Verifies the target NPC exists, and their location is reachable.
- SOLVE_MYSTERY: Verifies all clue items exist in the world and are associated with valid Items that have assigned Locations.

Stage 4: Challenges This stage modifies the base world from Stage 3 by adding complexity: puzzles with progressive hint systems and blocked passages between Locations. The LLM receives the complete world structure (all Locations, Items, Characters, and spatial connections) and generates puzzle definitions that specify which passages they block, what solutions unlock them, and hints to guide players toward solutions.

Initially, we attempted to have the LLM generate puzzles where solving one would unlock the next in a specific order, but the model consistently failed with circular references and structural errors, so we simplified the approach. Instead of enforcing a strict sequence in the schema, we prompt the LLM to suggest a “logical progression” where each challenge narratively builds on previous ones. This appeared to be more effective, as puzzles feel connected without the structural complexity that caused failures.

The generation prompt specifies concrete constraints: each puzzle must connect to either a blocked passage or an essential resource; obstacles and their solutions must be physically separated across different `Locations`; blocked passages require defined unlock conditions (e.g., obtaining a key, solving a riddle); and solutions must be discoverable through exploration rather than requiring external knowledge.

Each puzzle includes LLM-generated hints organized in a progressive disclosure structure. Hints have 3 levels, starting with general orientation (e.g., “The answer lies in the study”) and progressing to nearly explicit guidance (e.g., “Examine the portrait of Lady Ashwood”). Additionally, each puzzle includes interaction hints suggesting how to begin (e.g., “Try talking to the librarian” or “Examine the locked chest closely”). This hint structure is critical for player assistance, as we discuss in the evaluation section.

Puzzles can be triggered by `Characters` (e.g., an NPC asks the player to solve a riddle) or by `Locations` (e.g., examining a room reveals an environmental puzzle). The LLM can strategically redistribute `Items` to adjust difficulty but cannot create new `Locations` and must preserve spatial connectivity.

Output: Modified `GeneratedWorld` object from Stage 3. The world now includes `Puzzles` with solutions, progressive hints (3-5 levels), and interaction hints; `BlockedPassages` that lock connections between `Locations` until puzzle completion; and potentially redistributed `Items` to create puzzle dependencies.

Validation: Schema validation ensures all puzzle fields are present and correctly typed. Puzzle reward verification checks whether puzzle rewards reference existing `Items`. If a puzzle grants a non-existent item as reward, the system automatically creates that `Item` and places it in the puzzle proposer’s inventory (for character-proposed puzzles) or in the relevant `Location` (for environmental puzzles). Finally, spatial connectivity (DFS) and objective completability are revalidated to ensure puzzles and blocked passages do not prevent the player from completing the main objective.

Validation Strategy: Correction vs. Rejection

Each generation stage may fail validation. However, not all validation failures require complete regeneration. IVIE distinguishes between *correctable* issues and *structural failures*. Correctable issues include missing bidirectional connections between `Locations` or type mismatches in fields—problems that can be automatically fixed without discarding the generated content. Structural failures include unreachable `Locations` (detected via DFS) or unsolv-

able objectives (e.g., required `Items` without assigned positions)—problems that fundamentally break world playability and require regeneration of that stage with enhanced prompts explaining the specific violation.

This distinction aims to balance robustness (rejecting genuinely broken worlds) with efficiency (avoiding unnecessary regeneration when simple corrections suffice). For correctable issues, IVIE applies automatic fixes and proceeds; for structural failures, it provides the LLM with detailed error descriptions to guide regeneration.

Originally, PAYADOR used regex parsing to extract structured data from LLM text responses, which appeared brittle when models generated unexpected formatting. IVIE addresses this by using Pydantic² models that define strict schemas for each generation stage. Modern LLM APIs enforce these schemas during generation (JSON schema-guided decoding), guaranteeing syntactically valid output. Pydantic then performs a second validation pass, checking that generated content is semantically meaningful—catching issues like missing required fields or invalid enum values that pass syntactic checks but would break downstream processing.

For example, the `GeneratedLocation` model enforces:

```
class GeneratedLocation(BaseModel):
    name: str
    connections: List[str]
    blocked_passages: List[BlockedPassages]
    relevance_to_objective: str
```

If the LLM attempts to omit `connections` or provide a non-list value, Pydantic raises a validation error. The system attempts regeneration up to three times with enhanced prompts that explicitly describe the validation failure. If all attempts fail, the stage is marked as failed and generation halts, requiring manual intervention or parameter adjustment.

RAG-Based Memory System

During gameplay sessions using PAYADOR, each turn receives only the current player action and immediate world state, without access to previous interactions unless explicitly included in the prompt. Moreover, including full conversation history without filtering introduces irrelevant or outdated context that may confuse the model rather than help it. This limitation causes narrative inconsistencies: NPCs may forget previous conversations with the player, or the narration may contradict events from earlier turns.

To address this, IVIE implements Retrieval-Augmented Generation (RAG) (Lewis et al. 2020) with episodic memory. The system stores each gameplay turn as an `AtomicMemory` unit containing the player action, system-generated narration, and a brief world state summary. For each memory, the system generates 768-dimensional semantic embeddings using Gemini’s embedding model³ and per-

²<https://docs.pydantic.dev/>

³Google Gemini Embedding API: <https://ai.google.dev/gemini-api/docs/embeddings>

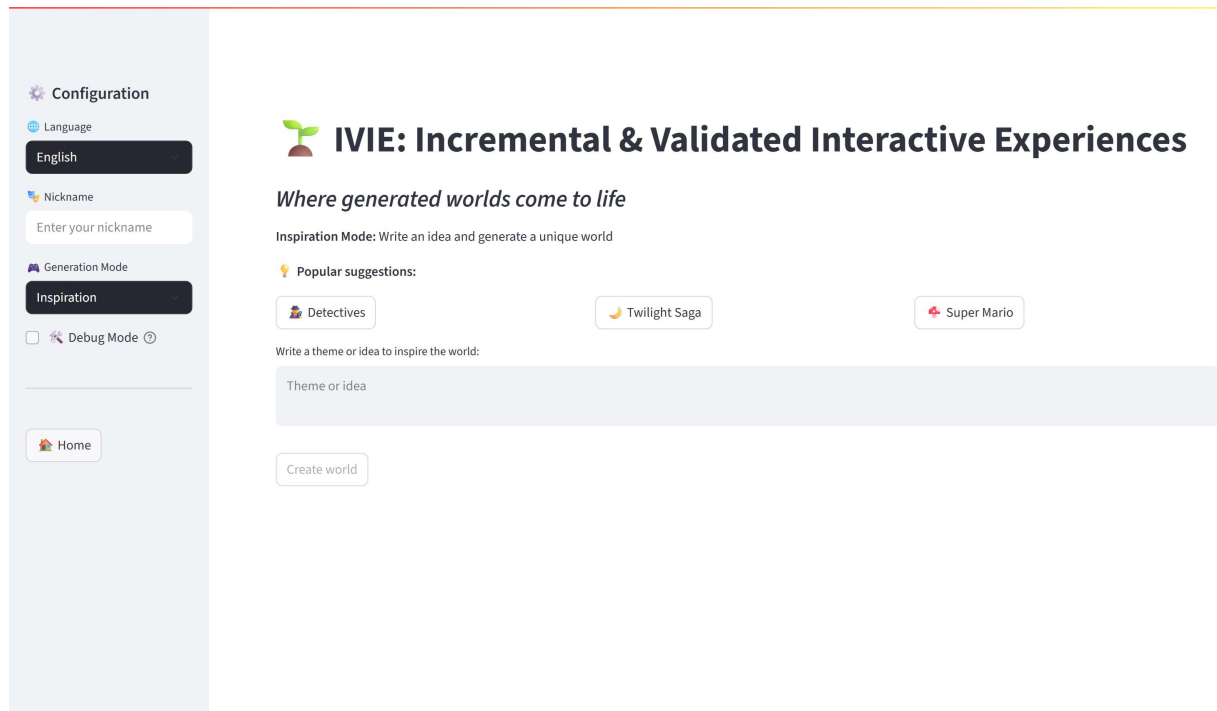


Figure 3: IVIE’s interface allows users to generate worlds in two modes: Generate (the LLM takes the creative responsibility) or Inspiration (based on thematic input). Configuration options include language, player nickname for identification, and debug mode for inspecting world structure and detecting potential issues.

sists them in ChromaDB⁴, which maintains one collection per generated world.

During gameplay, before generating the narration for each new action, the system retrieves the top-3 most semantically similar past memories based on embedding cosine similarity. These retrieved memories are injected into the narrative generation prompt in a dedicated section titled “Relevant Past Memories,” providing the LLM with contextual information about related past events. After the LLM generates narration for the current turn, that turn is stored as a new `AtomicMemory` and added to the database for future retrieval.

Semantic embeddings enable retrieving conceptually relevant memories even when they share no lexical terms with the current action. For instance, a player input “examine the portrait” might retrieve a memory where an NPC mentioned the portrait’s subject, even though “portrait” wasn’t in that earlier conversation. This can be valuable for narrative coherence, as the LLM can naturally reference past events to create a sense of continuity. Evaluators with RAG enabled reported significantly better long-term narrative coherence, with NPCs remembering previous interactions across dozens of turns, as we discuss in the evaluation.

Persistence and Reproducibility

During development and evaluation, we encountered the critical need to understand how generated worlds behave

across complete gameplay sessions. When players reported unexpected behavior, or worlds were not completable, we needed to examine the exact sequence of events: what the player did, how the LLM responded, and how the symbolic world state changed at each step. Additionally, we wanted to enable players to replay interesting worlds from the beginning, exploring different narrative paths within the same generated structure. To address this, IVIE maintains a persistent turn log stored as a JSON array, where each entry captures the full state of that interaction.

Each turn in the array captures: the player’s input text, the complete symbolic world state (which `Locations` are accessible, where `Items` are located, which puzzles are solved), the LLM’s generated narration, and what elements of the world state were modified (e.g., player moved to new `Location`, item added to inventory, puzzle marked as solved). Turn 0 specially captures the complete initial world state generated by the pipeline, enabling exact reconstruction of the starting conditions. Subsequent turns capture only state changes rather than complete snapshots, reducing storage requirements while maintaining full traceability.

Persisting these generated worlds allow several capabilities: replaying worlds from Turn 0 to explore alternative narrative paths within the same structure, analyzing LLM narrative patterns across complete sessions offline, debugging generation failures by examining the exact state at each turn, and comparing how different players navigate the same generated world. For instance, when evaluators reported incomplete worlds during evaluation, we used the replay ca-

⁴<https://www.trychroma.com/>

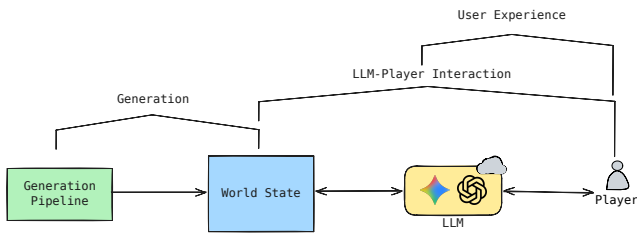


Figure 4: Three key dimensions evaluated: World Generation, LLM-Player Interaction, and Player Experience.

Dimension	Generate	Inspiration
Puzzle logic	3.88	4.00
Hints clarity	4.12	4.12
Objective clarity	4.88	4.62
Objective completion	4.38	3.50
NPC purpose	3.50	3.88
Entertainment	4.50	4.38

Table 1: Mean Likert scores (1–5) per dimension across generation modes (N=8).

pability to load those exact sessions and analyze both the world state and turn history to understand why these failures occurred.

Users interact with all these components through IVIE’s web interface (Figure 3), which provides configuration options for generation modes, language selection, and a debug mode that can help visualize the internal world structure and identify potential issues without playing.

Evaluation

We evaluated IVIE through human gameplay sessions across three key dimensions (Figure 4): *world generation quality* (structural validity, parameter adherence, spatial connectivity), *LLM-player interaction effectiveness* (puzzle difficulty, objective clarity, narrative coherence), and *overall player experience* (engagement, immersion, thematic coherence).

Experimental Design

Eight evaluators with varying gaming experience completed three gameplay sessions. All participants took part voluntarily and evaluation data was anonymized prior to analysis:

1. **Tutorial:** A pre-designed world to familiarize participants with the interface and mechanics.
2. **Generate Mode:** A world generated with complete LLM creative freedom.
3. **Inspiration Mode:** A world generated based on the participant’s favorite IMDB-listed movie.

We distributed participants evenly across experimental conditions to ensure balanced comparison:

- **Language:** 4 Spanish, 4 English
- **Memory System:** 4 with RAG enabled, 4 without RAG

- **LLM Provider:** 4 using Gemini 2.0 Flash, 4 using GPT-4o-mini

As previously mentioned, IVIE allows configuration of world-size parameters; in this evaluation, all worlds used identical configurations: 4 Locations, 4 Items, 2 Characters (NPCs), and 1 Puzzle.

After playing, participants completed 5-point Likert scale questionnaires (Yes/Mostly/Sometimes/Almost Never/Never) covering puzzles, objectives, world coherence, and overall experience, plus open-ended qualitative feedback. The complete evaluation data, including Likert scale responses, full survey questions, and participant gameplay sessions, are available in the source code repository. Table 1 summarizes mean Likert scores per dimension across both generation modes.

Players achieved 100% objective completion under Generate mode (8/8 worlds) but only 50% under Inspiration mode (4/8 worlds). These Inspiration mode failures revealed critical validation gaps: two worlds passed all validation stages despite having structurally impossible objectives: the required Items existed in the world model but had no assigned Location, making them inaccessible to the player. A third failure resulted from API quota exhaustion mid-session⁵, which caused the session to terminate abruptly. Finally, one failure resulted from a persistent LLM reasoning error: despite repeated correct attempts by the player to request an item from an NPC, the reasoning model consistently failed to generate the corresponding transfer transformation, leaving the item in the NPC’s inventory and making the objective seem unreachable.

Out of 16 generated worlds, 13 (81.25%) matched configured parameters exactly, while three worlds (18.75%) exceeded limits, all in English. Spatial connectivity validation (DFS) succeeded in all cases, with no player becoming isolated in unreachable Locations. Evaluators consistently praised spatial coherence, with one noting that: “transitions between locations were correct, and narration described surroundings very well.”. Across the 7 evaluated Generate mode worlds, we observed considerable thematic diversity (e.g., fantasy, steampunk, Egyptian mythology, space archaeology, post-plague alchemy) though recurring archetypes emerged (e.g., scholar protagonists, ancient artifact recovery), suggesting that while surface themes vary, LLMs tend to converge on familiar narrative structures. Inspiration mode worlds, by contrast, showed greater narrative diversity, likely driven by the diversity of the source films provided by evaluators.

Evaluators in *Inspiration mode* consistently reported that the generated worlds strongly reflected the thematic essence of their chosen inspiration movie. We therefore concluded that *Inspiration mode* achieved 100% thematic coherence across all evaluated worlds. In contrast, while *Generate mode* received positive feedback on overall creativity, evaluators reported lower perceived utility of generated elements: only 25% of *Generate mode* evaluators felt all objects and NPCs had a “real purpose” in achieving the objective, com-

⁵We used Google Gemini’s free tier during evaluation, which has daily request limits.

pared to 71% in *Inspiration mode*. The thematic constraint appears to integrate elements more tightly with objectives, reducing the presence of purely decorative content. We think this hints at a trade-off: *Inspiration mode* produced stronger thematic coherence yet lower completion rates, which might suggest that tighter thematic constraints increase world complexity in ways that expose validation gaps more frequently.

Approximately 75% of evaluators found the generated puzzles logical with useful hints, but 25% encountered overly cryptic puzzles requiring specialized knowledge or puzzles with internally inconsistent hints. However, in 3 out of 16 worlds, players were able to bypass puzzles by claiming they had solved them when providing incorrect or no solutions (e.g., typing “I solve the puzzle” without actually answering the puzzle). The LLM’s reasoning model accepted these claims as valid actions—a phenomenon similar to jailbreaking (Wei, Haghtalab, and Steinhardt 2023). This issue was previously observed in PAYADOR’s original implementation (Góngora et al. 2026), where the reasoning model could suggest state changes (e.g., marking a blocked passage as accessible) that the symbolic engine would accept even when prerequisites weren’t met. This seems to be a fundamental challenge for neuro-symbolic interactive storytelling: validating too strictly may constrain the LLM’s generative freedom, while validating too loosely may allow players to bypass puzzle logic entirely.

Regarding the incorporation of RAG for memory, we found that the 4 evaluators with RAG enabled experienced significantly better long-term narrative coherence. For instance, in Evaluator 7’s gameplay session, an NPC remembered a successful item theft during turn 17 and referenced it when re-encountered on turn 44 (“Have you come to try and claim what is not yours again?”), suggesting semantic memory retrieval across dozens of intervening turns. However, RAG occasionally mixed contexts and presented puzzles from past interactions instead of the current world state. We believe further exploration of RAG in procedurally generated interactive narratives could address these context-mixing issues through better memory segmentation, for example, organizing memories by type (e.g., location-specific events, NPCs, puzzles) and retrieving only contextually relevant categories based on the current game state. Evaluators with RAG enabled rated entertainment higher on average (4.75 vs. 4.25 without RAG). However, due to the small sample size (N=4 per condition), these results are not included in the table; a more in-depth evaluation would be needed to determine whether RAG has an actual impact on player experience.

Regarding perceived creativity, evaluators responded positively to open-ended questions about world originality in Generate mode, describing worlds as original and engaging to explore. In *Inspiration mode*, evaluators highlighted the fidelity to source material as a creative achievement in itself, with one noting that the result “came out of the ordinary and expected.” We acknowledge, however, that this evaluation does not measure creativity in formal terms nor does it compare against a non-creative baseline.

Despite technical issues, players found sessions entertaining. Notably, although worlds were small with antici-

pated 15-25 minute sessions, multiple evaluators played for over an hour, prioritizing narrative exploration over objective completion. Overall, we think this behavior hints at a high level of immersion.

Conclusions and Future Work

We presented IVIE, a neuro-symbolic system that automatically generates complete and playable IF worlds by strategically delegating creative decisions to LLMs while grounding world coherence through symbolic validation. We hope this work represents a step toward addressing the scalability challenges of IF authoring: while manual world design offers artistic control and crafted experiences, automatic generation enables rapid prototyping and increased replayability through procedural variation. IVIE achieves this through a four-stage incremental generation pipeline with symbolic validation at each stage. Unlike purely LLM-based approaches that rely on context window history, IVIE maintains an explicit symbolic world state, enabling validation of structural correctness and prevention of common inconsistencies (such as vanishing objects, contradictory spatial relationships, or unsolvable objectives).

Evaluation results suggest that symbolic validation mechanisms (including DFS for spatial connectivity, Pydantic for type checking, and objective completability verification) can prevent the logical inconsistencies inherent to pure LLM systems while preserving narrative flexibility. Generate mode achieved 100% completion with high player engagement (sessions exceeding one hour despite small 4-location worlds), and *Inspiration mode* suggests that 100% thematic coherence, proving LLMs can generate goal-oriented worlds when properly constrained.

We think IVIE’s results offer insights applicable beyond IF to any domain requiring creative generation within structural constraints: the four-stage incremental pipeline with intermediate validation steps suggests that decomposing generation into bounded stages — each with a defined validation contract — may be a viable strategy for grounding open-ended LLM generation in domains where structural correctness matters.

Several directions could significantly improve the approach. Currently, when objective validation fails (e.g., required `Items` exist but have no assigned `Location`), the system regenerates the entire stage up to three times before aborting. A more robust approach would automatically correct these issues—detecting unreachable items and assigning them valid positions—similar to how the system currently handles missing puzzle rewards, reducing generation failures and avoiding discarding otherwise playable worlds.

As discussed in the evaluation, the RAG implementation could benefit from organizing memories by type (`location events`, `Character interactions`, `Puzzle attempts`) to address the context-mixing issues we observed. Alternative, specific retrieval techniques for RAG could further improve memory relevance as well. Additionally, some evaluators encountered overly cryptic puzzles requiring specialized knowledge; implementing an LLM-based difficulty classifier that analyzes puzzle descriptions and hints could

identify and regenerate puzzles exceeding reasonable difficulty thresholds.

Also from evaluation feedback, several evaluators expressed interest in deeper NPC interactions. Growing research on LLM role-playing and persona assignment (Tseng et al. 2024) suggests that extending *Character* schemas with explicit personality fields (e.g., cooperative, hostile, enigmatic) could enable richer dialogue and more predictable characterization; currently, NPC behavior depends entirely on LLM generation at each turn without explicit personality guidance. Enabling mid-session save/resume would address a practical need: currently, API failures force players to restart worlds entirely, and sessions cannot span multiple play periods. Implementing true save/resume functionality would require extending the system to initialize world state from arbitrary turn numbers, not just the initial state.

Finally, while IVIE theoretically supports any LLM provider, we only evaluated the approach using commercial APIs (Gemini and GPT-4o-mini). Investigating open-source models (e.g. Llama, Mistral or Qwen) would reduce dependency on proprietary services and enable fully local execution. Exploring specialized smaller models for individual pipeline stages—rather than using one general model for all—could improve both efficiency and generation quality while maintaining local deployability. Beyond generation, IVIE’s symbolic world state could enable runtime improvisation: dynamically introducing NPCs, items, or narrative events during gameplay to assist struggling players or increase dramatic tension, similar to how players adapt *Tabletop Role-playing Games* scenarios in real-time (Martin, Harrison, and Riedl 2016; Góngora et al. 2024b). This would extend IVIE beyond world generation toward real-time narrative improvisation.

Author Contributions

M.V. and S.S. designed, developed, and conducted the human evaluation of IVIE as part of their Bachelor’s thesis in Computer Science Engineering (Vaucher and Silveira 2025). S.G. co-advised that thesis with L.C.; both are also authors of the PAYADOR approach, which IVIE builds upon.

References

Bomström, H. 2016. The application of procedural content generation in video game design. Bachelor’s thesis, University of Oulu, Department of Information Processing Science.

Callison-Burch, C.; Patel, A.; Dennis, J.; and Zhu, A. 2024. Dagger: Data augmentation for generative gaming in enriched realms. In *The 4th Wordplay: When Language Meets Games Workshop*.

Castricato, L.; Frazier, S.; Balloch, J.; Tarakad, N.; and Riedl, M. 2021. Automated story generation as question-answering.

Chambers, R.; Tack, N.; Pearson, E.; Martin, L. J.; and Ferraro, F. 2024. Berall: Towards generating retrieval-augmented state-based interactive fiction games. In *The 4th Wordplay: When Language Meets Games Workshop*.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *ECAI 2012*. IOS Press. 21–26.

Góngora, S.; Chiruzzo, L.; Méndez, G.; and Gervás, P. 2024a. PAYADOR: A minimalist approach to grounding language models on structured data for interactive storytelling and role-playing games. In *Proceedings of The 15th International Conference on Computational Creativity*.

Góngora, S.; Chiruzzo, L.; Méndez, G.; and Gervás, P. 2024b. Skill check: Some considerations on the evaluation of gamemastering models for role-playing games. In Donadio, P.; Rocha, M.; Brennan, A.; Schönbohm, A.; de Rosa, F.; Koskinen, A.; and Bellotti, F., eds., *Games and Learning Alliance*, 277–288. Cham: Springer Nature Switzerland.

Góngora, S.; Chiruzzo, L.; Méndez, G.; and Gervás, P. 2026. World-state transformations for neuro-symbolic interactive storytelling. In *Proceedings of The 17th International Conference on Computational Creativity*.

Góngora, S. 2025. Approaches to interactive and improvisational storytelling. Master’s thesis, Universidad de la República (Uruguay).

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.* 55(12).

Kelly, J.; Calderwood, A.; Wardrip-Fruin, N.; and Mateas, M. 2023. There and back again: Extracting formal domains for controllable neurosymbolic story authoring. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, volume 19, 64–74.

Khaled, R.; Nelson, M. J.; and Barr, P. 2013. Design metaphors for procedural content generation in games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’13, 1509–1518. New York, NY, USA: Association for Computing Machinery.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; tau Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 9459–9474.

Li, Z.; Wang, Z.; Xiong, C.; and Xing, C. 2025. Learning to play like humans: A framework for llm adaptation in interactive fiction games. *arXiv preprint arXiv:2505.12439*.

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

Montfort, N. 2003. *Twisty Little Passages: An Approach to Interactive Fiction*. MIT Press.

Riedl, M. O., and Bulitko, V. 2012. Interactive narrative: An intelligent systems approach. *AI Magazine* 34(1):67.

Riedl, M. O. 2015. An introduction to ai story generation. *Medium*. Accessed: 2025-11-02.

Sarker, M. K. H.; Zhou, L.; Eberhart, A.; and Hitzler, P.

2021. Neuro-symbolic artificial intelligence: Current trends. *AI Communications* 34(3):197–209.

Sawyer, R. K. 2000. Improvisational cultures: Collaborative emergence and creativity in improvisation. *Mind, Culture, and Activity* 7(3):180–185.

Schiaffini, G. 2006. Never improvise improvisation. *Contemporary Music Review* 25(5-6):575–580.

Shaker, N.; Togelius, J.; and Nelson, M. J. 2016. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Cham, Switzerland: Springer.

Trapova, A. 2023. Ai as a vehicle for creativity in video games – any room for flexibility via contracts? a case study on ai dungeon. In Vanhees, H., and Geiregat, S., eds., *Copyright Contracts*. LeA Uitgevers.

Tseng, Y.-M.; Huang, Y.-C.; Hsiao, T.-Y.; Chen, W.-L.; Huang, C.-W.; Meng, Y.; and Chen, Y.-N. 2024. Two tales of persona in LLMs: A survey of role-playing and personalization. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 16612–16631. Miami, Florida, USA: Association for Computational Linguistics.

Vaucher, M., and Silveira, S. 2025. IVIE: Generación incremental y validada de mundos jugables para experiencias interactivas. Tesis de grado, Universidad de la República (Uruguay). Facultad de Ingeniería.

Wei, A.; Haghtalab, N.; and Steinhardt, J. 2023. Jailbroken: How does llm safety training fail? *arXiv preprint arXiv:2307.02483*.

Yong, Q. R., and Mitchell, A. 2023. From playing the story to gaming the system: Repeat experiences of a large language model-based interactive story. In *International Conference on Interactive Digital Storytelling*, 395–409. Springer.

Zhou, E.; Basavatia, S.; Siam, M.; Chen, Z.; and Riedl, M. O. 2025. Story2game: Generating (almost) everything in an interactive fiction game. *arXiv preprint arXiv:2505.03547*.