N2NHU LABS FOR APPLIED ARTIFICIAL INTELLIGENCE

Technical Disclosure and Prior Art Establishment

The World Knowledge Interview Architecture

Interview-Validated, Transformation-Based World Generation for MUD-Class Interactive Environments

Jim (J.P.) Ames

Founder, N2NHU Labs for Applied Artificial Intelligence

February 23, 2026

Abstract

This paper describes the World Knowledge Interview (WKI) architecture — a novel pattern for automated generation of high-fidelity, culturally coherent interactive game worlds from a single text identifier. The system conducts a structured multi-turn dialogue with a Large Language Model (LLM) to establish validated ground truth about a named world before any content generation begins. This validated WorldContext flows as a single source of truth into all downstream generation steps, eliminating the category of errors caused by independent, uncoordinated LLM inference across pipeline stages.

The architecture is implemented in the N2NHU Infinite Improbability Drive (IID), an open-source world generator for MUD-class game engines. The IID generates complete playable worlds — rooms, characters, objects, transformations, and pre-rendered images — from any named television show, film, or cultural property, in approximately three minutes on consumer hardware, with no hardcoded world-specific content in the codebase.

This document constitutes a public technical disclosure establishing prior art. The architecture is released under GPL3 and is intentionally unpatentable by any party.

1. Introduction

The generation of interactive game worlds has historically required substantial human authorship. MUD (Multi-User Dungeon) environments, text adventure games, and their descendants require rooms with navigable connections, characters with behaviors and statistics, objects with properties and interactions, and transformation rules governing how the world responds to player actions. Creating a complete world from scratch typically requires weeks of design work per property.

The emergence of Large Language Models capable of recalling detailed cultural knowledge about named properties — television shows, films, literary works — creates an opportunity to automate this authorship entirely. The challenge is not capability: LLMs demonstrably know the layout of the S.S. Minnow, the characters of Gilligan's Island, the props used in MASH 4077. The challenge is coherence: getting all generation steps to agree on the same world.

1.1 The Isolation Problem

Prior approaches to LLM-assisted game content generation treated each generation step as an independent inference. Room names were generated in one call. Characters in another. Objects in a third. Each call independently rediscovered the world — and each had independent failure modes. The room namer might correctly identify Gilligan's Island as a tropical island while the object generator hallucinated medieval weapons. The character generator might produce the correct cast while the setting classifier returned spy thriller imagery.

These failures are not random. They arise from a fundamental architectural flaw: no shared ground truth. When each step independently asks the LLM what world it is working in, each step can independently be wrong — and wrong in different directions.

1.2 The World Knowledge Interview Solution

The World Knowledge Interview (WKI) pattern solves this by inverting the architecture. Instead of each generation step performing its own world discovery, a single structured interview runs first. The interview establishes validated facts about the world through multi-turn dialogue. These facts are stored in a WorldContext dataclass. Every subsequent generation step receives this context as a parameter and uses it as its sole source of world identity.

> *The insight is that cultural knowledge about named properties already exists completely in LLM training data. The WKI does not generate this knowledge — it harvests it, validates it, and makes it available to the entire pipeline as a single coherent truth.*

2. Prior Art Context

The following approaches preceded this work and are distinct from the WKI architecture:

2.1 Procedural Generation

Roguelike dungeon generators (Rogue, 1980; NetHack, 1987; and descendants) generate world structure algorithmically. These systems produce structurally valid worlds but without cultural coherence — rooms and contents are random, not thematically grounded in a named property. No LLM inference is involved.

2.2 Real-Time Narrative LLMs

Systems such as AI Dungeon (Latitude, 2019) use LLMs for real-time narrative generation within a session. These systems do not produce persistent structured world output. They generate narrative text, not queryable game state. No structured INI, JSON, or equivalent output is produced for consumption by a separate game engine.

2.3 Manual LLM-Assisted Authorship

Authors have used general-purpose LLMs (ChatGPT, Claude, etc.) to assist in writing individual room descriptions, character backstories, and similar content. This is manual authorship with LLM assistance — not automated pipeline generation. Each query is independent and results are assembled manually.

2.4 Academic Procedural Narrative

Academic work in procedural narrative generation (Riedl, Young, et al.) addresses story generation and plot coherence. This work does not address the specific problem of generating MUD-class world structure from named cultural properties, nor does it employ the multi-turn validation pattern described here.

2.5 Distinction of This Work

The WKI architecture is distinguished by the combination of: (1) multi-turn LLM interview for world knowledge extraction, (2) binary confirmation-based validation of extracted facts, (3) a single shared WorldContext consumed by all downstream generation steps, (4) complete structured output suitable for direct consumption by a game engine, and (5) no world-specific content hardcoded in the generating system. No prior work combines all five characteristics.

3. System Architecture

3.1 The Transformation Principle

The IID is built on a foundational architectural principle: every system operation is a matrix transformation. A show name is input state. A complete game world is output state. Between them lies a series of well-defined transformations, each consuming the output of the previous step. This mirrors the mathematical structure of linear algebra — and, notably, the operational structure of IBM 407 Accounting Machine plugboard programming from 1953, which the authors identified as an inadvertent historical parallel during development.

The practical consequence of this principle is that complex behaviors are implemented through configuration rather than code. Adding a new theme (Western, Superhero, Science Fiction) requires adding entries to a configuration map, not modifying core logic. The transformation engine remains constant; the transformation matrices change.

3.2 Pipeline Overview

The IID pipeline consists of five sequential stages:

- Stage 0 — World Knowledge Interview: Multi-turn LLM dialogue establishing validated WorldContext
- Stage 1 — World Structure Generation: Room graph, physics parameters, and configuration files
- Stage 2 — Room Description Enrichment: LLM-generated descriptions using WorldContext as anchor
- Stage 3 — Image Pre-generation: Stable Diffusion rendering of each room using setting-aware prompts
- Stage 4 — Object, Character, and Transformation Enrichment: Two-step generation using WorldContext
- Stage 5 — Matrix Validation: Cross-reference checking guaranteeing zero broken exits

3.3 The World Knowledge Interview (Stage 0)

The WKI is the novel architectural contribution of this work. It executes before any content generation and establishes the WorldContext that all subsequent stages consume.

3.3.1 Interview Structure

The interview consists of up to four turns:

- Turn 1 (Open Description): The LLM is asked to describe the named world in 2-3 sentences. The prompt requests specificity about genre, setting, and time period, and explicitly asks the LLM to acknowledge non-recognition rather than hallucinate.
- Turn 2 (Structured Extraction): The description from Turn 1 is fed back to the LLM with a structured extraction prompt requesting genre, setting, location, era, characters, and object categories in an exact key:value format.
- Turn 3 (Binary Validation): The extracted facts are presented to the LLM as a summary, and the LLM is asked a strict binary question: are these facts correct? The prompt requires a single word response: Yes or No.
- Turn 4 (Correction, conditional): If Turn 3 returns No, a correction prompt is issued requesting the corrected facts in the same structured format. The corrected facts are then re-validated in a second binary confirmation.

This structure is designed to catch the most common LLM failure mode in world generation: confident hallucination. An LLM that does not recognize a show name will often generate plausible-sounding but incorrect facts. The binary confirmation step surfaces this: an LLM that hallucinated in Turn 2 will often return No in Turn 3 when shown the summary of its own hallucination.

3.3.2 The Binary Confirmation Pattern

The choice of binary (Yes/No) rather than open-ended confirmation is architecturally significant. Open-ended prompts invite the LLM to second-guess correct answers — a phenomenon observed during development where correct settings were 'corrected' to incorrect alternatives by an over-helpful confirmation prompt. Binary prompts force a commitment without providing an opportunity to introduce new errors.

The Yes/No extractor strips all preamble before evaluating the response, taking only the first meaningful word. This handles the LLM's tendency to preface answers with affirmations ('Absolutely, yes...') that could confuse naive substring matching.

3.3.3 WorldContext Dataclass

The interview produces a WorldContext dataclass with the following fields:

```
world_name:         str   # e.g. 'GILLIGAN'S ISLAND'
description:        str   # e.g. '1960s castaways stranded on a tropical island'
genre:              str   # e.g. 'SITCOM'
setting:            str   # e.g. 'TROPICAL ISLAND'
location:           str   # e.g. 'SOUTH PACIFIC'
time_period:        str   # e.g. '1960s'
characters:         List[str]  # e.g. ['Gilligan', 'The Skipper', 'The 
Professor'...]
object_categories: List[str]  # e.g. ['coconut shells', 'fishing nets', 'bamboo 
tools'...]
confidence:         str   # 'high' / 'medium' / 'low'
method:             str   # 'llm_interview' / 'llm_interview_corrected'
```

Confidence is set to 'high' when Turn 3 returns Yes on first attempt, 'medium' when correction was required or validation was ambiguous, and 'low' when the LLM indicated non-recognition. All confidence levels produce usable output; confidence affects only whether downstream steps apply additional validation safeguards.

3.4 Downstream Context Propagation

The WorldContext flows into every subsequent generation step as a named parameter:

- Room Namer: Uses ctx.setting and ctx.location to anchor room name generation, bypassing independent setting discovery entirely when context is available.
- Theme Classifier: Uses ctx.genre to select the appropriate theme (SITCOM, CRIME_SPY, SCIFI, WESTERN, etc.), which determines combat parameters, physics defaults, and Stable Diffusion style prompts.
- Object Enricher: Uses ctx.object_categories as the anchor for Step 1 object name discovery, replacing generic category inference with interview-validated categories.
- Sprite Enricher: Uses ctx.characters as the anchor for Step 1 character name discovery, dramatically reducing hallucination by providing a validated cast list.

- Setting Suffix Mapper: Uses ctx.setting to select the appropriate Stable Diffusion visual style override, ensuring room images match the show's visual identity.

3.5 Two-Step Object and Character Generation

Object and character generation uses a two-step pattern within Stage 4:

- Step 1 (Name Discovery): The LLM is asked to list names appropriate to the world, anchored by ctx.object_categories or ctx.characters. Output is a CSV list of names.
- Step 2 (Structured Generation): The validated name list from Step 1 is fed back to the LLM with a structured format prompt requesting pipe-delimited output with all required fields.

This separation is architecturally important. Step 1 asks an easy question (what are some X in this world?) and gets reliable answers. Step 2 asks a formatting question (give me this data in this exact structure) and gets reliable structure. Combining both questions in a single prompt produces unreliable answers to both.

A critical implementation detail: the Step 2 structured prompt uses maxsplit on the pipe delimiter, so the description field absorbs all remaining content including embedded commas and pipes. Without maxsplit, comma-rich descriptions corrupt the field mapping.

3.6 Character Hallucination Challenge

Character generation includes a Step 1b validation pass that addresses LLM hallucination of non-existent cast members. After Step 1 produces a character list, Step 1b presents each name to the LLM with a binary challenge: is this character a real, recurring cast member of the named show?

In testing, this step removed hallucinated characters including Josh Randall (Wanted: Dead or Alive, not Alias Smith and Jones), Doc Holliday (historical figure), and DI Waters (British police rank, wrong country) from their respective worlds. The challenge has a configurable acceptance threshold: any result with at least one verified name proceeds rather than being discarded entirely.

3.7 Fallback Architecture

The system implements a graceful degradation chain rather than hard failure at any step:

- Step 2 structured parse succeeds: Full rich objects with descriptions (optimal path)
- Step 2 parse fails, Step 1 names available: Minimal valid objects built directly from Step 1 names (name-driven fallback)
- No names and no parse: Empty output with warning (honest failure, no corrupted data)

The name-driven fallback is architecturally significant: it preserves the world-correct names from Step 1 rather than substituting generic templates. A Step 2 failure on Gilligan's Island produces objects named 'Skipper's Cap' and 'Volleyball Net' with minimal metadata — not 'Combat Weapon' and 'Explosive Device' from a generic military template.

4. The Cultural Knowledge Harvest Principle

The WKI architecture rests on an observation about the nature of LLM training data that has significant implications beyond game generation.

Large language models trained on internet-scale corpora have absorbed the content of decades of human cultural production: television scripts, fan wikis, episode guides, critical reviews, academic analyses. This content is not merely stored — it is encoded in the model's weights as structured knowledge, queryable through appropriately constructed prompts.

The IID does not generate game content. It harvests latent cultural knowledge that already exists in the LLM and transforms it into structured game data. The Skipper's Cap, the Professor's bamboo gadgets, the Minnow's wreckage — these exist in Llama 3's training data as thoroughly as they existed in CBS's production bible. The WKI is the extraction mechanism.

> *The folders of completed worlds produced by the IID are not generated content. They are crystallized cultural memory — decades of human creative output transformed into interactive form by a pipe that asks the right questions in the right order.*

This principle has a direct parallel in the history of computing. The IBM 407 Accounting Machine (1953) did not create financial data — it processed existing ledger data through plugboard-configured transformation matrices. The insight that complex transformations could be implemented through configuration rather than hardwired logic was foundational to enterprise computing. The IID rediscovers this principle applied to cultural data and LLM inference.

5. Legal Architecture and Licensing

5.1 The Tool Provider Distinction

The IID is a tool for generating game worlds. It does not distribute game worlds. It does not contain named-property content. The codebase contains no character names, no show titles used as data, no world-specific examples. A forensic examination of the repository finds only: a prompt template asking the LLM to describe a user-supplied name, a structured extraction mechanism, and a pipeline for transforming extracted facts into game data.

The legal position of the IID is analogous to a printing press. The press manufacturer is not responsible for what is printed. The LLM provider is responsible for the content of its training data and the fidelity of its recall. The user who enters a show name and generates a world bears responsibility for how that world is used.

5.2 GPL3 and Intentional Unpatentability

The IID is released under the GNU General Public License version 3 (GPL3). This license, combined with the public disclosure in this document, establishes prior art that renders the described architecture unpatentable by any party.

This is a deliberate choice. The World Knowledge Interview pattern, the binary confirmation loop, the WorldContext propagation architecture, and the two-step name/structure generation separation are all described here in sufficient detail that no subsequent patent claim can establish novelty. Any patent application filed after the date of this document's publication will fail the novelty requirement.

The authors make this choice in the tradition of foundational open technologies: the World Wide Web (Berners-Lee, 1989, no patent filed), the Linux kernel (Torvalds, 1991, GPL), and the academic tradition of publishing discoveries for the benefit of all rather than enclosing them for commercial advantage.

5.3 The Area 51 World

The one original game world distributed with the IID — Area 51 — is original creative work by the authors and is not derived from any named third-party property. Its inclusion demonstrates the engine's capability without creating any third-party IP exposure for the project.

6. Empirical Results

6.1 Validated World Generations

At the time of this publication, the IID has successfully generated complete worlds for the following properties, with outputs manually reviewed for thematic correctness:

- Gilligan's Island — tropical island setting, correct cast (Gilligan, Skipper, Professor, Mary Ann, Ginger, Mr. and Mrs. Howell), correct objects (coconut items, island props), correct Stable Diffusion visual style
- MASH 4077 — Korean War military hospital setting, correct cast, surgical and military objects, correct visual style
- Alias Smith and Jones — Arizona Territory frontier setting, correct cast, Western objects, frontier visual style
- Superfriends — Hall of Justice setting, correct DC heroes, superhero objects, primary-color heroic visual style
- Star Trek — Federation starship setting, correct bridge crew, Starfleet objects, sci-fi visual style
- I Love Lucy — Manhattan apartment setting, correct cast, 1950s domestic objects, period visual style
- Gomer Pyle USMC — Marine base setting, correct cast with hallucinations removed, military objects
- Planet of the Apes — post-apocalyptic future setting, ape society objects and locations
- Adam-12 — LAPD patrol car and precinct setting, correct cast, police equipment
- Laugh-In — comedy variety show setting
- The Brady Bunch — suburban home setting

6.2 Performance Characteristics

Complete world generation (10 rooms, 8 characters, 12 objects, pre-rendered images) completes in approximately 3 minutes on consumer hardware with local LLM inference (Llama 3 8B via GPT4All) and local Stable Diffusion. The bottleneck is image generation; text generation for all components completes in under 60 seconds.

6.3 Failure Mode Analysis

Systematic testing identified and resolved the following failure modes during development:

- Genre hallucination: Independent setting inference producing incorrect genre (ADAM 12 generating gothic horror). Resolved by WKI architecture.
- Character hallucination: LLMs adding plausible but non-existent cast members. Resolved by Step 1b binary challenge.

- Embedded delimiter corruption: Object names containing commas or quotes corrupting pipe-delimited field parsing. Resolved by maxsplit and name sanitization.
- Preamble ID leakage: LLM conversational preamble being parsed as object or character ID. Resolved by ID length and word-count guards.
- Generic template fallback: Parse failures falling back to military-themed generic content. Resolved by name-driven minimal fallback.
- T3 validation ambiguity: Verbose LLM responses to binary questions causing false negative classification. Resolved by first-word extraction with punctuation stripping.

7. Future Directions

The WKI architecture described in this paper is a foundation for several natural extensions:

- Enterprise domain application: The same interview-validate-transform pattern applies to enterprise data domains. A compliance rule set, a network topology, a business process — all can be described through multi-turn LLM interview and transformed into structured operational configuration. The authors have demonstrated this with a quantum-inspired UTM firewall system and an MSP customer service portal.
- Multi-model validation: Running the WKI interview across multiple LLMs and taking the consensus WorldContext would further reduce hallucination rates for obscure properties.
- Dynamic world evolution: The transformation architecture supports world state changes as additional matrix operations. A world that evolves in response to player actions is an extension of the same mathematical structure.
- Visual coherence through single pane of glass: AI-generated room images already demonstrate the principle that world state can be expressed visually. Dashboard systems where operational state is represented through AI-generated imagery are a direct extension.

8. Conclusion

The World Knowledge Interview architecture solves the world generation coherence problem by establishing validated ground truth before any content generation begins. The insight that LLMs contain structured cultural knowledge that can be harvested through appropriately designed multi-turn dialogue — rather than generated anew in each pipeline step — is the core contribution.

The result is a system that transforms a single show name into a complete, playable, thematically coherent interactive world in three minutes, with no world-specific knowledge hardcoded in the generating system. The codebase is forensically clean of named-property content. The legal architecture is clear. The prior art is established.

The architecture is given to the world under GPL3. It is intentionally unpatentable. It is a gift in the tradition of foundational open technologies — a shovel for whoever wishes to dig.

> *Everything is just transformations. The show name is input state. The playable world is output state. Between them lies a pipe that asks the right questions.*

Jim (J.P.) Ames
N2NHU Labs for Applied Artificial Intelligence
February 23, 2026

Appendix A: Novel Patterns Summary

The following patterns are identified as novel contributions of this work for prior art purposes:

Pattern 1: World Knowledge Interview (WKI)

A multi-turn LLM dialogue for extracting and validating cultural knowledge about a named property before any downstream content generation. Characterized by: open description turn, structured extraction turn, binary validation turn, and conditional correction turn.

Pattern 2: Binary Confirmation Loop

A Yes/No validation mechanism applied to LLM-extracted facts, using first-word extraction to handle verbose responses. The binary structure prevents the LLM from introducing new errors during the validation step.

Pattern 3: Single-Source WorldContext Propagation

A validated WorldContext dataclass produced by the WKI and consumed by all downstream pipeline steps as a named parameter, eliminating independent world discovery and its associated failure modes.

Pattern 4: Two-Step Name/Structure Generation

Separation of content generation into a name discovery step (easy, high reliability) and a structured format step (requires exact output format), with maxsplit protection on the delimiter to prevent description field corruption.

Pattern 5: Binary Character Hallucination Challenge

A per-character binary validation step that asks the LLM whether each generated name is a real, recurring cast member of the named property, with hallucinated names removed before structured generation.

Pattern 6: Name-Driven Minimal Fallback

When structured parsing fails, construction of minimal valid game objects from Step 1 names rather than generic templates, preserving world-correct naming through the fallback path.

Appendix B: Repository and Timestamp

Repository:	https://github.com/jimpames/n2nhu-infinite-improbabilty-drive-for-universal-game-engine
License:	GNU General Public License v3 (GPL3)
Publication Date:	February 23, 2026

Author: Jim (J.P.) Ames, N2NHU Labs for Applied Artificial Intelligence

Video Demonstration: https://www.youtube.com/watch?v=YmiGD3yX81g

Prior Art Status: Established by public disclosure under GPL3. Unpatentable by any party.

This document is itself part of the prior art record. Its existence, combined with the timestamped GitHub repository and YouTube demonstration, establishes unambiguous prior art for all patterns described herein.