RASE: Retrieval Augmented Story Engine Narration Control in game using AI

Shishir Govinda M[1], N Chaitra[2], Shiva Keerthi MP[3], Rohit Mesta[4]
[1,2,3,4]*Student, Dept. of CSE, Sir M Visvesvaraya Institute of Technology Bengaluru, India*

Abstract—**Recent advances in large language models (LLMs) have enabled AI systems to generate increasingly coherent and contextually rich narratives. However, purely generative approaches to story generation often struggle with maintaining long-term coherence and factual consistency, sometimes producing disjointed or hallucinatory storylines. To address these challenges, researchers have begun integrating retrieval augmented generation (RAG) techniques into storytelling systems. In a Retrieval-Augmented Story Engine (RASE), a narrative-generating LLM is coupled with an external knowledge repository and retrieval mechanism, so that each piece of the story is grounded in relevant context fetched on demand. This framework leverages the strengths of both data-driven retrieval and generative modeling, aiming to reduce inconsistencies while preserving creativity. The following report examines RASE's conceptual framework, core components, and implementation methodology, and situates it in the landscape of game AI and interactive narrative systems. We also compare RASE with alternative approaches – including purely generative narrative models, reinforcement learning-based storytelling agents, and classical game master or experience manager AIs – to clarify how retrieval augmentation builds upon and differs from these paradigms.**

Index Terms—**RAG, SFT, Game, Narration, Machine Learning**

I. INTRODUCTION

A. Developer Intention and Player Expectation

Role-playing games (RPGs) derive their allure from a tightly calibrated interplay between designer-specified mechanics and the emergent behaviors that players experience in situ, as elegantly characterized by the Mechanics–Dynamics–Aesthetics (MDA) framework and empirically demonstrated in recent anal- yses of player engagement. Players arrive with idiosyncratic objectives and an innate proclivity to probe—and, at times, subvert—narrative constraints in pursuit of novel "dopamine hits," a phenomenon that aligns closely with Flow theory's stipulation of optimally balanced challenge and skill. Yet despite these well-understood dynamics, the vast majority of RPGs remain anchored to static dialogue trees and linear scripting, which inherently stifle adaptive storytelling and constrain the generation of spontaneous, context-sensitive content. To address this limitation, we introduce a unified architecture—seamlessly embedding a vectorized in-game knowledge base, Retrieval-Augmented Generation (RAG), and Super-Fine-Tuning (SFT) pipelines—directly within the game engine. Player-driven variability across divergent play through [1].

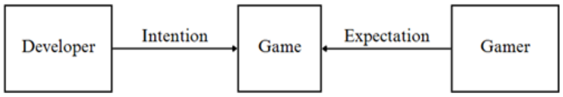

Fig. 1. Developer Intention and Player Expectation

B. Player Stimulus

The behavioral trace illustrated in figure below depicts live player engagement contours superimposed upon zones of boredom and anxiety, revealing a latent expanse for exploratory freedom that nonetheless remains rigidly bounded by pre-authored content. In this model, even when players veer toward under- or over-stimulation, the prescribed action space rarely expands to accommodate emergent discoveries, resulting in a tacitly enforced "narrative inertia." Such rigidity not only diminishes opportunities for serendipitous storytelling but also undermines the very sense of agency that underpins sustained immersion.

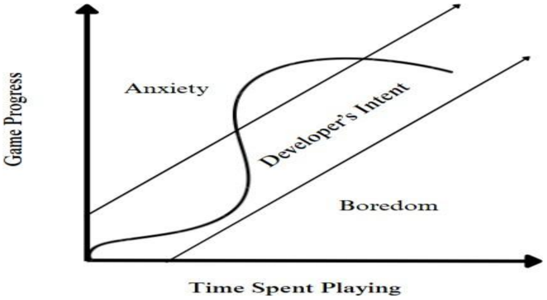

Fig. 2. Developer's Intent vs Player Curiosity

C. Allure of Modern RPGs

Contemporary AAAA RPGs—exemplified by Red Dead Redemption 2, The Witcher 3, the Watch Dogs series, and Cyberpunk 2077—boast expansive open worlds teeming with NPCs, hidden interactions, and procedurally varied side-quests. Yet even these titans suffer from narrative fatigue: once players exhaust the limited repertoire of three to five dialogue variants per character (notably during mundane transactions such as purchasing weapons or provisions), repetition becomes palpable. This fatigue manifests in the widespread habit of skipping audio cutscenes, as players disengage from a story they perceive as a perfunctory barrier to end-game rewards—trophies, achievements, or final titles [2]. While this sprint to completion may seem innocuous from the player's vantage, it represents a profound inefficiency from the developer's perspective: months of painstaking narrative design and world-building risk being bypassed, their creative investment left unrealized.

Moreover, these games instrument an astonishing breadth of telemetry—hours mounted on horseback, cumulative distance traversed, weapon-specific kill counts, even minutiae such as birds hunted or vehicle collisions logged—yet this high-dimensional dataset remains largely siloed, relegated to analytics dashboards or achievement trackers. Similarly, the prevailing requirement for persistent online connectivity functions primarily as a DRM enforcement mechanism or gateway to leaderboards, rather than as a conduit for real-time adaptive content delivery. From a machine-learning standpoint, this represents a squandered opportunity: by harnessing player telemetry through unsupervised representation learning and reinforcement-based policy optimization, one could infer latent playstyle embeddings and dynamically instantiate bespoke narrative threads or dialogue permutations. In such a system, each player's unique interaction profile would seed a context vector within a Retrieval-Augmented Generation pipeline, ensuring that subsequent NPC dialogues, quest objectives, and environmental vignettes coalesce into a coherent, continuously evolving story tailored to the individual's emergent preferences [3]. This paradigm shift not only mitigates narrative repetition but also reclaims the developer's investment by embedding intelligence directly into the game loop, ultimately elevating immersion through genuine co-creative engagement.

D. The RASE Framework

To overcome the narrative inertia endemic to static story- telling pipelines, we introduce the Retrieval-Augmented Story Engine (RASE), a fully procedural, context-aware system that maintains a constant binary footprint regardless of whether it generates ten or 100,000+ interactions. RASE begins by ingesting diverse developer artifacts—design PDFs, narrative TXT outlines, structured JSON/JSONL dialogue schemas and subjects them to a rigorous preprocessing stage of normalization, semantic segmentation, and chunk-level annotation. Each chunk is then transformed into a dense embedding and indexed in a low-latency vector database. At runtime, the player's live telemetry—action sequences, dialogue choices, environmental triggers—is encoded into a dynamic context vector, which RASE fuses via a Retrieval-Augmented Generation (RAG) pipeline to produce NPC utterances, quest modifications, and world-state vignettes that honor both authorial constraints and emergent playstyles [4]. Central to RASE are two complementary capabilities—Retrieval-Augmented Generation (RAG) and Super-Fine-Tuning (SFT)—that together enable truly context-aware NPC and environmental interactions. Each player–NPC exchange is not only vectorized and stored in the vector database for immediate RAG retrieval but also contributes incremental gradient updates during SFT, allowing the underlying LLM to specialize on the idiosyncrasies of individual characters, locales, and gameplay scenarios [5]. Over time, this continual loop of retrieval and fine-tuning yields progressively sharper semantic alignment: NPCs respond with ever-more precise, lore-consistent dialogue, and environmental descriptions dynamically adapt to a player's evolving style and prior choices, all while preserving real-time performance through our hybrid compute architecture.

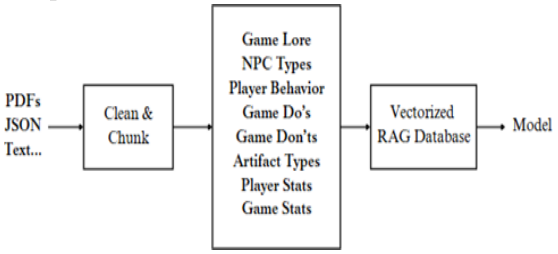

Fig. 3. Vectorized RAG Database

Game Studios such as UBISOFT have started to implement such frameworks at a much smaller scale for their terrain generation also called as autonomous design tools which was extensively used in production for the game Ghost Recon Wildlands released in 2017, these studios have autonomous design and design work can be collaborated along with human designers and tools, yet ethical considerations are still considered when using such autonomous tools [6].

Embedding a large-scale language model within the game loop, however, raises three core challenges [7]. First, unanchored generations can hallucinational diverge from the Developer's narrative intent when faced with out-of-distribution player inputs. Second, nondeterminism may erode reproducibility: without controlled seeding or sequence-aware conditioning, identical interaction traces can yield inconsistent outcomes. Third, on-device inference imposes significant compute over- head, threatening framerate stability on mid- to low-end hardware. RASE addresses these issues through a hybrid execution paradigm: lightweight policy networks and context- cached embeddings operate locally, while heavyweight LLM inference is offloaded to elastically scaled cloud instances [8]. An intelligent caching layer persists recent context vectors and delta states, ensuring narrative coherence during brief connectivity lapses and seamlessly reconciling client–server state upon reconnection. In this way, RASE delivers highly adaptive, semantically rich storytelling without compromising performance, determinism, or the fidelity of the original design.

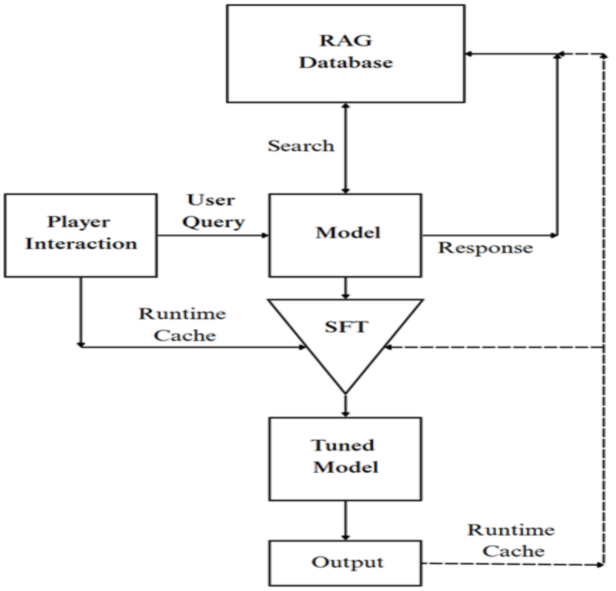

Fig. 4. High Level RASE

The diagram above presents a high-level schematic of RASE as a standalone subsystem; a detailed operational flow will be elaborated in subsequent sections. In essence, RASE's core function is to fuse rich, developer-authored game lore with a continuous feedback loop driven by player behavior. Each player–environment or player–NPC interaction is instrumented, vectorized, and fed back into both the RAG retrieval index and the SFT buffer. This dual-pathway architecture enables RASE to generate contextually grounded content that reflects the full spectrum of human affect—condemnation, condescension, praise, derision, playfulness, levity, romance, and beyond—thereby delivering dynamically tailored narrative experiences aligned with each player's emergent style [9].

II. LITERATURE REVIEW

The game Dungeons & Dragons (D&D) has taken AI-driven storytelling to a new level by introducing multi-character, improvisational, and state-dependent dialogue as a formal challenge for artificial intelligence. In their groundbreaking work, Callison-Burch et al. (2022) framed D&D not just as a storytelling experience but as a dialogue modeling benchmark, presenting it as a unique testbed for AI systems tasked with generating in-character dialogue, maintaining narrative coherence, and predicting hidden game state from conversation. By gathering and annotating over 800,000 dialogue turns from actual D&D campaigns, the authors empowered models to implicitly learn the intersection of narrative, personality, and mechanics [10]. The authors also found evidence that the AI responses improved significantly in quality and believability when grounding generation in dialogue history and latent game mechanics.

This work illustrates the ability for games, such as D&D, to inspire AI to engage in more than just conventional text generation, but also to provide an avenue for AI to engage in thinking and reasoning that will work for role followership, improvised conscious reasoning, and collaborative storytelling, all which applies more broadly to the goals of RASE. Through their system, character fidelity within the content produced is of utmost importance over simple branching narratives or just generic large language model generated outputs, with pacing of the narrative as well [11]. The research sets a precedent for how retrieval-augmented generative AI can improve realism and control in interactive narratives.

Now looking into Game AI as Storytelling by Mark Riedl, David Thue, and Vadim Bulitko traditional game AI has been adversarial in nature, designed to outplay the human on various of difficulty or by providing a strong advantage to the AI. The advantage can be in the form of strategy, strength, defense, speed, accuracy, knowledge and many more metrics. This enables the player to tackle the obstacle in a very strategic manner, but after a certain tries the advantage of the obstacle is understood and does not change is a drastic enough manner to rethink the strategy thought of and implemented. A good example for this is the Sigrun Valkyrie Fight in God of War 2018 at the initial encounter she is one of the toughest boss fights in the same but over time and learning her moves a fight that took almost 2 hours to win can be reduced to a couple of minutes [7]. The perspective we can garner from this is that large open world games can become ridiculously shallow unless spent a lot of time and money on them, yet still after a certain amount of time the game mechanics become stagnant. This perspective is especially relevant in large open- world games like Grand Theft Auto, Watch Dogs, or Far Cry, where players often encounter side quests that while numerous are mechanically repetitive and narratively shallow [12]. Tasks like ’follow this NPC’, ’hack this terminal’ or ’deliver this item’ are frequently re-used with minor contextual changes, leading to content fatigue despite the vastness of the game.

The paper by OpenAI - Dota 2 with Large Scale Deep Reinforcement Learning explore how reinforcement learning can be implemented with ESports at scale with Dota 2, the game. They introduced OpenAI Five, an RL agent which leveraged thousands of CPUs and GPUs in parallel to process decades worth of experience and all the possible move sets and the system achieved a superhuman performance as expected and defeated the world champion team with a remarkable 99.4%-win rate. By abstracting into RASE, we aim to bring multi- turn narrative coherence, latent-state awareness, and scalable adaptation into game storytelling—similar to how OpenAI Five achieves emergent coordination at scale, but applied to the domain of narrative generation and delivery [13].

III. METHODOLOGY

This section presents the methodological framework behind RASE (Retrieval-Augmented Story Engine), including a com- parative analysis of existing narrative generation techniques, a detailed explanation of RASE’s hybrid architecture, and the rationale behind key design decisions. The approach aims to deliver scalable, context-aware, and author-guided storytelling by fusing semantic retrieval with large language model (LLM)- based generation [14].

A. Overview

RASE is designed to address limitations in current narrative AI systems by integrating retrieval-based story grounding with the generative flexibility of transformer models. The following subsections outline the core components employed in its development.

- Knowledge Base & Vector Store: This is the repository of narrative facts, world knowledge, and memory transcripts that the engine can draw upon. It can include authored story content (backstories, lore, predefined events) as well as dynamically accumulated data (player actions, consequences, characters’ current states). All such content is converted into vector embeddings to support semantic lookup. Efficient vector databases (e.g. FAISS, Milvus, Weaviate) index these embeddings, enabling quick similarity search. For instance, if the story is set in a fantasy world, the knowledge base might store descriptions of each location and character; if a player revisits a location, the engine can retrieve the relevant description to maintain continuity.
- Retrieval Module: The retrieval component handles forming queries to the vector store and selecting relevant context snippets for the LLM. When a new piece of story needs to be generated (e.g. the player performs an action or the narrative advances), the RASE system creates a query embedding (often derived from the latest player input or the current plot point) and finds the top-k matching knowledge entries. Retrieved data might include, for example, an NPC’s personality profile and dialogue history when generating that NPC’s next line. The retrieval module then formats this information (often with a prompt template) to present to the LLM. This often involves incorporating role identifiers or instructions – for example, prepending “You are Aria, the village elder (NPC)” or other guidance to orient the LLM. The result is a synthesized context prompt that provides

the LLM with the memory and situational awareness needed for the upcoming narration.

- Generation Module (LLM): The generation component is an advanced language model (such as GPT-4 or a similar large transformer model) that serves as the storyteller. Given the context-filled prompt from the retrieval module, the LLM generates the next segment of narrative text. This could be a descriptive passage, the next plot event, or a character's dialogue response. The LLM's role in RASE is to weave the retrieved facts into a coherent and engaging narrative continuation. Because it is augmented with contextual knowledge (instead of relying purely on its internal parameters), the LLM is less likely to stray off-topic or invent inconsistent details. For example, when asked to narrate the outcome of a battle, the LLM will be guided by the retrieved state of the game world (e.g. which enemies are still alive, what weapons the player has) to produce a fitting and logical outcome [15]. The LLM in a RASE essentially functions as a narrative composer that ensures the story remains context-aware. Indeed, one can liken it to the "storyteller" in a RAG system: it takes curated information from the retriever and crafts a contextually appropriate continuation.
- Experience Management & Memory Updater: Although not always described as a separate module, RASE systems often include logic for integrating the newly generated content back into the narrative state. In interactive environments (like games), this plays the role of a drama manager or experience manager, keeping track of plot progression and ensuring the story responds to player choices. Classic experience manager AIs in games were conceived as agents that manipulate the world to deliver a better narrative experience. In RASE, some of this responsibility is handled by the retrieval and generation loop: the engine continuously "remembers" past interactions via retrieval, and can also update the knowledge base with new events after they occur. For example, if the LLM narrates that a bridge collapsed, the system should record this outcome so that subsequent queries treat the bridge as destroyed. This dynamic updating closes the feedback loop, giving the story engine a form of persistent memory. In effect, the retrieval module in tandem with a memory store allows the RASE to achieve longitudinal coherence, as characters and the world can evolve while remaining consistent with earlier narrative facts [16].
- Safeguards and Constraints: Given that LLMs are generative and can produce inappropriate or offtrack content, many RASE implementations include a post-generation filtering or correction step. For instance, the system might use a guardrail mechanism to validate that the LLM's output is in the correct format and free of disallowed content. This could involve rule-based checks (removing profanity or meta-gaming statements) or even secondary LLM evaluations. Comparing a raw LLM response versus one vetted by a set of rules and filters (to catch issues like hallucinations or mentions that contradict the lore). Such safeguards are crucial in a storytelling context – they help maintain tone, prevent the narrative from breaking the fourth wall, and avoid introducing content that violates game or community standards. While not unique to RASE, these constraint modules enhance the reliability and believability of the generated story, which are key metrics in narrative AI evaluation.

Each of these components contributes to RASE's overall goal: delivering a narrative that is both adaptive and anchored. The retrieval engine provides anchoring in the form of factual or prior context, while the generative model provides adaptivity and creativity in how the story unfolds. By separating concerns (knowledge recall vs. language generation), RASE's modular design echoes known benefits of RAG architectures in other domains – notably, reducing hallucinations and improving factual accuracy – but here applied to the domain of interactive storytelling [17].

B. Hybrid Retrieval-Augmented Architecture

The Retrieval-Augmented Story Engine (RASE) introduces a hybrid architecture that integrates symbolic narrative memory with neural generative models. Unlike scripted branching structures that predefine all narrative paths, or purely generative systems that risk incoherence, RASE creates a balance between authored stability and algorithmic creativity [18]. At its core, the engine retrieves semantically relevant fragments from a narrative memory bank and uses transformer-based large language models (LLMs) to fuse these fragments into coherent, adaptive storylines.

This approach addresses three long-standing challenges in computational storytelling: (1) scalability of authored content, (2) adaptivity to unpredictable player actions, and (3) maintenance of narrative coherence over long temporal spans. By grounding outputs in a controlled corpus and constraining LLMs with semantic retrieval, RASE establishes itself as an author-guided yet player-responsive narrative generator.

- Grounded storytelling: Narrative content is anchored in carefully curated world knowledge, ensuring that generated outputs remain faithful to established lore, character backstories, and thematic constraints. This mitigates hallucinations commonly observed in unconstrained generative systems.
- Adaptability: Semantic search allows story events to adapt in real time to dynamic game states, making the system responsive to unexpected player behaviors. For example, if a player repeatedly aligns with a hostile faction, the retrieval pipeline can dynamically elevate fragments tagged with betrayal or rivalry, shifting the story tone accordingly.
- Fluency and creativity: By leveraging LLMs for recombination, retrieved fragments are not simply concatenated but woven into stylistically rich prose. This provides players with narrations and dialogues that feel authored yet improvisational.
- Continuity: Through persistent memory tracking and con- textual prompts, RASE maintains consistency of character motivations, unresolved arcs, and world state progression across long sessions—something most branching dialogue systems cannot achieve without combinatorial explosion [19].

C. Narrative Corpus and Semantic Indexing

The corpus functions as the backbone of RASE, consisting of modular fragments of narrative authored at varying granularities—from single dialogue lines to multi-paragraph lore entries. Unlike monolithic scripts, these fragments are designed to be recombined dynamically, increasing re playability and reducing authorial bottlenecks [20].

- Narrative function: Each fragment is explicitly tagged with its dramaturgical role (setup, conflict, climax, resolution). This metadata enables the engine to preserve narrative arc structures rather than producing flat or episodic outputs.
- Stylistic and emotional tone: Metadata further includes emotional registers (tragic, hopeful, suspenseful) and stylistic modes (epic, conversational, ironic). Such annotations allow RASE to fine-tune outputs to match both player emotion curves and designer intent.
- Contextual tags: Entities such as characters, factions, items, and locations are linked via structured tags. These tags enable the system to retrieve fragments that are narratively relevant to the current world state. For example, if the player interacts with a forested biome, RASE prioritizes retrieval of fragments tagged with nature, mystery, or environmental lore.

Fragments are embedded in a high-dimensional semantic space using pretrained encoders such as Sentence-BERT or Ada-002. Vector similarity search with FAISS or Weaviate enables sub-second retrieval even at scale, making the architecture viable for real-time games.

D. Contextual Retrieval from Game State

The retrieval pipeline is driven by a dynamic query embedding, generated from the evolving game state. This embedding is not a simple keyword map but a learned representation of the intersection between player actions, narrative arcs, and world-state variables.

- Player-centric factors: The query incorporates not only direct choices but also inferred trajectories, such as whether the player is adopting a heroic, neutral, or antagonistic role. Emotional tone detection from prior dialogue further guides which narrative fragments are appropriate.
- World state: The engine continuously monitors conditions like weather, geography, or factional conflicts. For example, if a war is ongoing in the world, retrieval elevates story beats tagged with violence, politics, or strategy.
- Narrative arcs: The query considers which arcs are active, deferred, or completed. This prevents premature resolutions and ensures deferred threads can be reactivated later, maintaining long-term coherence.

This design ensures that narrative beats evolve in tandem with gameplay, producing a “living story world” rather than a static script.

E. Fusion with Language Models

Once fragments are retrieved, they are structured into prompts that include narrative goals, stylistic constraints, and player history. The LLM then synthesizes final outputs. This design transforms the LLM from an unconstrained generator into a constrained improviser [21].

- Coherence preservation: Retrieved fragments serve as semantic anchors, ensuring the generated prose respects continuity in character voice, unresolved arcs, and thematic motifs.
- Stylistic consistency: Designers can enforce narrative tone (e.g., noir, heroic epic, satire) across sessions by weighting fragments of certain stylistic tags higher, ensuring that generative fluency does not drift stylistically.
- Contextual responsiveness: Dialogue or narration adapts seamlessly to player activity. For instance, if a player slays a village elder against narrative expectation, the LLM can weave this into emergent dialogue—transforming allies into critics or adversaries.

This layered fusion of retrieval and prompting keeps the generative engine both creative and bounded by authorial scaffolding.

F. Narrative State Tracking

Persistent state tracking distinguishes RASE from one-off narrative systems. By storing fine-grained history, it enables continuity across hours or even multiple campaigns.

- Fragment usage: Fragments already deployed are logged with timestamps and narrative context, preventing redundancy and supporting callbacks (e.g., a character referencing something said ten hours earlier).
- Arc status: Active, deferred, and resolved arcs are explicitly modeled. This enables storylines to pause and resume dynamically, avoiding abrupt abandonment or premature closure.
- Player trajectories: The system maintains longitudinal data on player decisions and emotional alignment, enabling emergent yet consistent character arcs—for instance, a player gradually shifting from antihero to villain.

This subsystem is critical to supporting long-term causal storytelling that mirrors the richness of tabletop role-playing experiences.

G. Designer Control Interface

RASE emphasizes co-creativity by giving authors high-level levers without requiring micro-management of every output.

- Tone, theme, pacing control: Designers can adjust meta- data weights to foreground certain tones (e.g., suspenseful) or regulate pacing (e.g., ensuring climaxes occur after sufficient buildup).
- Mandatory beats: Authors can inject fixed narrative milestones, such as betrayals or revelations, into the system. RASE integrates these beats organically by aligning them with retrieved fragments and player context.
- Dynamic filters: Filters allow designers to restrict retrieval based on game-world events. For example, a world event like the fall of a kingdom can globally deprioritize fragments tied to that kingdom, reflecting diegetic consequences.

This author-in-the-loop approach ensures that human vision remains primary, while the system provides scalable improvisation.

H. Evaluation Protocol

Evaluating an adaptive narrative system requires multidimensional metrics spanning computational, linguistic, and experiential domains.

- Narrative coherence: Human annotators score continuity across sessions, while automated methods such as coreference resolution and discourse analysis provide objective measures of consistency.
- Contextual relevance: Embedding similarity between query vectors and final generated outputs provides a quantitative measure of alignment between game state and narrative delivery.
- Engagement and believability: Player-facing studies assess immersion, replayability, and emotional resonance. Metrics such as choice diversity, dialogue depth, and branching density are used as proxies for engagement.
- System performance: Latency benchmarks (retrieval under 200ms, generation under 1s) are used to ensure real-time playability, while scalability tests simulate large player bases to measure robustness under concurrent demand.

This multi-pronged evaluation protocol ensures RASE is judged not only by its technical efficiency but also by its ability to deliver compelling, immersive storytelling at scale. Baseline comparisons with fully generative and scripted systems reveal that RASE delivers more coherent, engaging, and flexible narrative experiences with lower authoring overhead. Modern frameworks and tooling have made RASE implementations more accessible. Libraries like LangChain provide abstractions for chaining retrieval and LLM calls, and managing prompt templates. Similarly, open-source platforms (Hugging- Face pipelines, etc.) enable swapping in different models without changing the surrounding code [22]. This flexibility lets developers experiment with the engine's components in isolation – for example, trying a knowledge graph instead of free-text chunks in the retrieval stage – to push the boundaries of interactive storytelling.

IV. RASE IN COMPARISION

A. RASE vs. Generative Narrative Models

RASE emerged against the backdrop of powerful generative models that, on their own, have both impressed and frustrated creators. Pure generative narrative models – systems where an AI creates story text without explicit external guidance – include everything from recurrent neural network storytellers of the 2010s to today's prompt-driven LLM storytellers. The likes of GPT-3/4 or other large LMs can indeed spin lengthy tales given a starting prompt, showcasing a learned sense of language and storytelling structures. However, these models operate as black boxes, drawing only on the statistical associations learned during training. They have no inherent mechanism to verify facts or remain consistent to anything outside their internal representation. As a result, purely generative systems often introduce continuity errors (changing a character's name or traits mid-story) or outright fabrications that defy established setting lore [23]. For instance, an AI might initially say the sky is green in the story's world (intended as a unique fantasy element) but later describe it as blue because blue skies are more common in its training distribution. Human authors would catch such contradictions; a raw generative model frequently will not.

RASE fundamentally alters this picture by bridging the generative model with a knowledge repository. Instead of relying solely on the model's memory (which is vast but not specifically tuned to any one story), RASE offloads the remembering task to the retrieval system. Thus, a RASE- based storyteller is always reminded of the key facts each time it generates output. This leads to markedly improved consistency – the engine doesn't "forget" what happened or who is who, because the relevant details are fed back into it at each turn. Studies bear this out: for example, Wen et al. showed that augmenting story generation with a repository of human-written story segments allowed the LLM to produce more intricate and coherent plotlines than prompting alone

The retrieval acted as a form of conditional grounding, whereas detailed prompt engineering without retrieval often boxed the model into a narrower creative space [24]. In other words, a regular generative model might need a long prompt containing all rules and backstory upfront (leaving less freedom to deviate), whereas RASE can inject those rules and facts only when needed, allowing the model to be creative in between. This addresses a key drawback of prompt-heavy strategies: overly detailed prompts can inadvertently restrict the model's imagination. RASE offers a more flexible way – the model is free to invent within the bounds of retrieved context, which acts like guard rails rather than railroad tracks.

Another major difference is in factual accuracy and reduction of hallucination. In open-ended story generation, hallucinations (introducing unfounded details) may not always seem problematic – after all, fiction is made-up. But even in fiction, internal consistency and authorial intent impose a kind of factuality. If the AI says a gun is on the mantel in Chapter 1, we expect that to hold or to see it used by Chapter 3 (Chekhov's gun principle). A generative model might hallucinate a new gun out of thin air in Chapter 3, confusing the narrative. RASE's retrieved memory prevents that by reminding the AI "only a vase was on the mantel, not a gun." Even more straightforwardly, some story generation involves real-world or predefined lore (think of a Star Wars story – you wouldn't want the AI to say Yoda is a Sith Lord due to a hallucination). Retrieval augmentation has proven effective in factual domains and similarly helps narrative models stick to known facts or established canon [25]. Large language models like ChatGPT or GPT-4, when used alone, might occasionally stray or mix up lore (especially if the prompt is long and they lose some earlier details due to context window limits). RASE mitigates the context limit issue by actively fetching what's relevant now, rather than hoping the model remembers a detail from 5000 tokens ago.

One can see RASE as an evolution of generative narrative models – it builds on their strengths (linguistic fluency, learned intuition for storytelling) while compensating for their weaknesses (memory limitations, lack of explicit world models). The end result is a system that can rival human-authored stories in continuity and depth. Early evidence of this is seen in user experiences with AI Dungeon: when augmented with a knowledge database by the community (through mods that allowed it to consult a world info library), the quality of storytelling improved, with fewer nonsensical transitions. Essentially, RASE formalizes that idea by tightly integrating retrieval into the generation loop, rather than treating it as a user-triggered add-on.

Of course, RASE does not solve all problems of generative models – the output is only as good as the retrieved data and the model's inherent abilities. If the knowledge base is incomplete or the LLM has poor narrative skills, the story may still falter. But as LLMs become ever more capable (GPT-4, PaLM, etc. have demonstrated the ability to handle long and complex narratives) and techniques for knowledge curation improve, RASE stands to become the default mode for any serious AI storytelling system. It provides a clear path forward to make generative models reliable creative partners rather than erratic geniuses. In summary, compared to vanilla generative narrative models, RASE delivers stronger coherence, factual adherence, and controllability, all without significantly sacrificing the open- ended creativity that makes AI-generated storytelling appealing in the first place.

B. RASE vs. Reinforcement Learning-Based Storytelling Agents

Prior to the rise of large LLMs, one popular approach to adaptive storytelling was through reinforcement learning (RL) and planning-based agents. Researchers attempted to train agents (often called drama managers or narrative planners) that choose narrative actions to maximize some notion of player satisfaction or story quality. In reinforcement learning- based storytelling, the problem is typically formulated as a sequential decision-making task: at each step, the system (as a game master) picks the next plot point or action, receives feedback (a reward signal) based on how well it entertains or engages, and learns a policy to optimize future choices [26]. Over time, such an agent could, in theory, learn to tailor stories to individual players by observing their reactions (e.g. boredom or excitement signals) and adjusting the narrative trajectory accordingly.

RASE and RL-based narrative agents have a shared goal – adapt the story in response to players – but they approach it from different angles. RL-based systems, such as those developed by James Lester's group for interactive narrative or the PaSSAGE system by Thue et al., treat story generation as an optimization problem. They often operate on abstract state and action spaces: for example, an RL drama manager might represent the story state in terms of plot points achieved or player emotional state, and actions might be high-level events like "reveal clue" or "introduce enemy." The agent doesn't write prose; it decides which event should happen next from a predefined set, and then typically a hand-authored snippet of story for that event is shown to the player. The reinforcement signal could be something like "did the player engage with this plot thread?" or a designer-specified reward for hitting certain dramatic beats. Roberts et al [27]. (2006) and others showed this can work in limited domains, learning policies that e.g. adjust difficulty or pace to keep players in a "flow" state.

In contrast, RASE operates at the textual, generative level rather than at an abstract decision level. A RASE doesn't explicitly maximize a numerical reward through trial-and-error; instead, it relies on the knowledge base and the LLM's learned representation of good storytelling to produce content. The adaptation in RASE comes from conditioning on different retrieved context rather than from an update of a policy. For instance, if a player seems to prefer action over dialogue (something an RL agent might notice and adapt to by increasing combat frequency), a RASE system could also adapt if its knowledge base or prompts include an estimate of player preference. It could retrieve a "preferred style = action-heavy" context and the LLM would then naturally generate a more action-oriented scene [28]. But this adaptation is one-step and reactive, not the result of a learned policy over many episodes. One advantage of RASE over RL in practice is reduced training data requirements and easier authorial control. RL-based storytelling agents often struggled due to sparse and delayed rewards, combinatorial explosion of story states, and the difficulty of simulating realistic players for training. Many projects had to use simulated players or heavily abstract the narrative to make RL tractable. Even then, the learned policies might be brittle or hard to interpret. By contrast, RASE leverages pretrained LMs (which come with a wealth of prior knowledge about narrative structure and human-like responses) and doesn't require iterative training in the target domain – it works zero-shot or with a bit of prompt tuning. From a developer's standpoint, it's easier to guide a RASE: you can edit the knowledge base or prompt to push the story in a certain direction, effectively authoring by constraint. With an RL agent, you'd have to tweak reward functions or provide exemplar trajectories, which is more indirect and can lead to unintended behaviors.

However, RL approaches bring something valuable to the table: a clear notion of objectives and potentially long-term planning. A well-designed RL narrative agent explicitly encodes what makes a story "good" or a player "satisfied" via its reward function. RASE, using an LLM, has a more implicit and heuristic grasp of quality. It might default to the most statistically likely (cliche´d) story turn, whereas an RL agent could be tuned to prefer novelty or to ensure each player gets a distinct storyline (with rewards for avoiding previously seen outcomes, for example). There is nothing stopping a hybrid approach: one could use RL on top of a RASE by having the RL agent decide which knowledge to retrieve or which of several LLM-generated options to present, thereby combining the strengths. For example, an RL agent could maintain a high-level plot graph and use RASE to flesh out the scenes for each plot node when it's triggered – learning when to trigger which plot node for maximum player enjoyment [29].

In summary, compared to RL-based storytelling, RASE is more data-driven and leverages linguistic knowledge directly, while RL is more goal-driven and structural. RASE excels at producing rich narrative content on the fly, given the right context, but doesn't inherently guarantee an optimal narrative arc in terms of player engagement metrics. RL agents can be aimed at such optimization but often lack the rich generative capacity. As of 2025, RASE-like systems have largely overtaken purely RL ones in practical implementations because the results are immediately compelling (thanks to the strength of modern LLMs) [30]. The Wired article's observation remains pertinent: achieving a truly clever adaptive Dungeon Master AI is extremely challenging, and while RL provided one possible path, the advent of RAG/RASE has offered a more straightforward route by sidestepping heavy training and instead exploiting vast pretrained models plus curated knowledge. Going forward, a convergence of the two approaches might yield the best of both: using RL to manage high-level storytelling objectives and RASE to execute the moment-to-moment narration in a coherent way.

C. RASE vs. Game Master (Experience Manager) AIs

Long before the current generation of AI storytellers, the concept of an AI Experience Manager (or drama manager) was proposed to improve player experiences in story-driven games. The idea, championed by researchers like Mark Riedl, was to have an AI overseer that dynamically adapts the narrative in response to the player, much as a human Dungeon Master would [31]. This experience manager might, for example, introduce a plot twist if it predicts the player is getting bored, or ensure that the player's choices lead to a satisfying narrative conclusion by rearranging events. Implementations of this idea ranged from rule-based systems (e.g., the drama management in Fac¸ade, which used predefined story beats selected via search) to planning systems and the RL approaches discussed earlier. How does RASE compare to these game master AIs? In many ways, RASE can be seen as a modern reincarnation of the experience manager concept, but operating at a finer granularity and with learned knowledge. Traditional experience managers often worked with discrete plot events and had full control to pick the next event from an author-created pool. They did not typically generate new content themselves – they were like conductors orchestrating pieces composed by someone else. RASE, on the other hand, generates the content of the story itself, not just selects from pre-written branches. This is a fundamental shift. It means RASE has a much larger narrative space to work with (essentially infinite, since it can always phrase something differently or invent new dialogue), but it also means RASE has to be reined in to respect narrative logic – a task for which the retrieval augmentation is crucial.

Another difference is in adaptation style. Experience man- agers often explicitly model the player's state or preferences and then make narrative adjustments. For example, an experience manager might reduce combat encounters if it detects the player tends to avoid fights, tailoring the aesthetics of the experience (this ties to the MDA framework – Mechanics, Dynamics, Aesthetics – where the AI aims to shape the aesthetic experience by tweaking mechanics/dynamics) [32]. RASE doesn't intrinsically model player preferences unless that data is part of its knowledge base or input. However, it can still achieve similar adaptation implicitly. For instance, if the player's actions indicate a preference (like always trying diplomatic solutions), a well-designed RASE system would retrieve instances of successful diplomacy and have the LLM generate outcomes aligned with that approach, effectively adapting the story to the player's style. This is less direct than an experience manager with a variable for "combativeness = low," but with enough data, the behavior emerges naturally from the patterns the LLM knows about storytelling and the contexts it's given.

One might say RASE is a bottom-up approach (micro-level adaptation each step via content) whereas traditional experience management is top-down (macro-level control of narrative structure). Top-down control can guarantee certain narrative properties (e.g., a story will always have a climax at a particular point; the pacing follows a desired curve), which is important in authored experiences. RASE's more free-form generation might not always produce a neatly structured story unless guided. That's why many RASE implementations still rely on some author-provided high-level scaffolding. For example, a human designer might define that there are three acts in the story and key milestones (boss fights, revelations, etc.), and this outline is stored in the knowledge base. The RASE then fills in the details between these milestones. If the story veers off, the retrieval mechanism will surface the outline again to push the LLM back on track. In effect, RASE can work under the supervision of a lightweight drama manager encoded as data. This hybrid can outperform either alone: the experience manager gives high-level coherence, and RASE yields low-level richness and reactivity.

Notably, RASE also changes the toolset for authors. In older experience manager systems, authors had to provide a lot of explicit content (all the branches or event options) and possibly utility functions for the AI to judge story quality. With RASE, authors focus on feeding the system with the right world knowledge and maybe a canonical storyline, and the AI handles the rest [33]. This arguably lowers the barrier to creating interactive narratives, or at least shifts the burden from writing countless branches to curating knowledge and rules. It aligns with how human game masters operate: rather than scripting every possible path, DMs know the world and characters, and they improvise the story in response to players. RASE's knowledge base is the analogue of the DM's knowledge of the game world; the LLM's improvisation is the analogue of the DM narrating. Thus, RASE is conceptually very close to what the game AI community envisioned: AI as Dungeon Master, one of the ultimate roles of game AI. The difference is we now have the generative technology to attempt it fully, whereas earlier systems could only approximate it through pre-authored branching or swapping out quest flags [34].

One might worry that without an explicit experience manager, a RASE could fail to ensure the player's experience is "enjoyable" in a holistic sense. After all, enjoyment is subjective and context-dependent; a drama manager can be tuned (via its reward or evaluation function) to specific definitions of enjoyment. RASE relies on the assumption that a coherent, context-aware story produced by the AI will be enjoyable, and on whatever implicit training signal about engaging storytelling the LLM internalized from human literature. In practice, this seems to be a reasonable assumption for many scenarios: players generally prefer logically coherent narratives that react to them, which RASE provides. If finer control is needed (say to adjust the tone or difficulty of the narrative challenges), designers can incorporate those as variables in the prompt or retrieval context (e.g., a "grimdark tone" flag that causes the LLM to narrate in a darker style, or a difficulty level that influences the retrieved obstacles).

In conclusion, RASE both builds on and diverges from the classic game master AI approaches. It builds on them by fulfilling the vision of an AI that actively manages the narrative to enhance player experience – a goal articulated in the literature over a decade ago – and it does so with far more fluidity and linguistic capability. It diverges by eschewing heavy-handed planning or explicit optimization in favor of real-time, content-level adaptation. One might say experience managers were about planning the story, whereas RASE is about telling the story moment-to-moment [35]. Both are crucial facets of storytelling; we're now at a point where the AI can handle the telling, and with light guidance, also achieve the planning. As RASE matures, we expect to see it become the backbone of many AI "game masters" in digital RPGs and interactive fiction, effectively fulfilling and surpassing the early prototypes of experience-managed games.

V. CONCLUSION

RASE represents a significant step forward in the pursuit of truly intelligent storytelling agents. By marrying retrieval- augmented generation with the demands of narrative coherence, it effectively bridges older paradigms of game AI (which emphasized planning and authorial control) with the new paradigm of large-scale generative modeling. The conceptual framework of RASE – splitting the storytelling process into a knowledge retrieval phase and a creative generation phase – has proven to be a powerful abstraction, yielding systems that produce stories with a blend of consistency and spontaneity previously unattainable by purely rule-based or purely generative methods [36].

We have seen how RASE's core components work in tandem: the knowledge base provides a living memory of the story and game world, the retrieval module acts as the storyteller's faithful assistant (reminding it of facts and context), and the LLM generator serves as the imaginative narrator that brings the story to life. The implementation methodologies discussed highlight that building such systems is now within reach – leveraging frameworks for vector search, prompt orchestration, and robust LLM APIs. As developers refine these techniques, we can expect more streamlined toolkits specifically for narrative RAG (some early signs include specialized story vector databases or narrative design interfaces that let writers feed content into the knowledge base without coding).

In practical applications, RASE has demonstrated clear benefits: NPCs in games become more lifelike and context- aware, interactive plots can adjust in real time to unanticipated player actions while remaining coherent, and human creators can collaborate with AI on complex branching stories with less fear of the AI derailing the narrative. These successes, however, come with caveats and open challenges. One challenge is evaluation: ensuring the AI's adaptations truly improve player experience. It's not enough for the story to be consistent; it also needs to be engaging, and engagement is hard to measure objectively. User studies (asking players if they felt more immersed, for example) are crucial to validate that RASE's theoretical advantages translate to fun gameplay [37]. Initial reports are encouraging but small-scale – larger studies will build confidence in these systems. Another challenge is content moderation and ethical narrative generation. With RASE giving a lot of freedom to the AI to generate content, there's a risk of it producing inappropriate or biased material if the knowledge base or model has such associations. The solution, as touched upon, involves guardrails and possibly refining the training of the LLM on curated datasets. The retrieval mechanism can actually assist here by excluding certain knowledge: for instance, to avoid stereotypes, the knowledge base can be seeded with character traits that deliberately break cliche´s, guiding the AI towards more nuanced portrayals [38].

Performance and scalability are also concerns. The latency introduced by the retrieval step and large-model inference might hinder real-time applications, although ongoing improvements in model efficiency and clever caching of retrieval results (for things that repeat often, like an NPC's bio) are mitigating this. The cost of calling powerful LLMs frequently can add up, so commercial deployments will have to balance model size, caching, and perhaps fall back to smaller models for less critical narrative moments. There is active research into long-context models that might reduce the need for explicit retrieval by handling very long prompts – if those succeed, RASE may evolve by having retrieval play a smaller role or a different role (perhaps focusing more on factual correctness than context window management).

Importantly, RASE is not a static concept but a growing paradigm. We already see variations: some systems replace textual retrieval with knowledge graphs for more structured control (e.g. retrieving a subgraph of story entities and relations), others incorporate player analytics (retrieving similar past player behaviors to predict current player needs), and some integrate multimodal feedback (imagine a horror game that retrieves the current audio intensity level to help the LLM set the scene appropriately). As LLMs become more deeply integrated in game engines (with Unity and Unreal exploring AI plugins), RASE could become an integral part of game design workflows. Designers might work less on branching narrative scripts and more on curating the knowledge and rules that the AI will use – effectively moving from writing stories to writing story engines [6].

In closing, the Retrieval-Augmented Story Engine stands at the cutting edge of AI-driven narrative. It differentiates itself from past approaches by proving that we can have the best of both worlds: the richness of generative AI and the reliability of knowledge-based systems. Each comparison we examined – whether to pure generative models, RL agents, or classical drama managers – revealed that RASE either extends or improves upon those approaches in crucial ways, though sometimes at the cost of added complexity in system design [39]. As research and development continue, we anticipate that many of these components will be refined and possibly commoditized (for instance, a "story memory module" one can plug into any narrative game).

The ultimate validation of RASE will be in the experiences it enables. Imagine a future game where every playthrough is unique, yet every playthrough feels hand-crafted; where the AI storyteller can handle the wildest player deviation with grace and still deliver a compelling tale. That is the promise on the horizon. RASE is not the final answer – narrative intelligence is a vast domain – but it is a decisive stride towards AI systems that can truly understand and generate stories in the way humans do: by remembering, reasoning, and recombining elements of our vast narrative heritage into something new and magical for the audience of one, the player.

REFERENCE

[1] AGI-Edgerunners, "Llm agents papers: A curated repository," GitHub Repository, 2024. [Online]. Available: https://github.com/AGI-Edgerunners/LLM-Agents-Papers

[2] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving rubik's cube with a robot hand," 2019, preprint. [Online]. Available: https://openai.com/blog/solving-rubiks-cube/

[3] A. Alvarez, J. Font, and J. Togelius, "Story designer: Towards a mixed- initiative tool to create narrative structures," in *AIIDE Workshop on Narrative Workshop*, 2022.

[4] L. Zhang and S. Kim, "Static vs. agentic game master ai for facilitating solo role-playing experiences," *ACM Games: Research and Practice*, 2025.

[5] T. Balint and R. Bidarra, "Experimental narratives: A comparison of human and ai-generated storytelling," *Digital Creativity*, 2023.

[6] S. Seidel, N. Berente, A. Lindberg, K. Lyytinen, B. Martinez, and J. V. Nickerson, "Artificial intelligence and video game creation: A framework for the new logic of autonomous design," *Journal of Digital Social Research*, vol. 2, no. 3, pp. 126–157, 2020.

[7] W. Zhang and C. Li, "Modeling individual differences in game behavior using hidden markov models," *Entertainment Computing*, 2022.

[8] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[9] S. Buongiorno, L. Klinkert, Z. Zhuang, T. Chawla, and C. Clark, "Pangea: Procedural artificial narrative using generative ai for turn-based, role- playing video games," in *Proceedings of the Twentieth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 2024.

[10] C. Callison-Burch, G. S. Tomar, L. J. Martin, D. Ippolito, S. Bailis, and D. Reitter, "Dungeons and dragons as a dialog challenge for artificial intelligence," *arXiv preprint arXiv:2210.07109*, 2022.

[11] M. Johansson and P. Eriksson, "Player experiences in the game coridden," *Games and Culture*, 2022.

[12] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," *arXiv preprint arXiv:1805.04833*, 2018.

[13] A. Gargari, "Enhancing medical ai with retrieval-augmented generation," *Digital Health*, 2025.

[14] J. Xu, X. Ren, J. Lin, and X. Sun, "A skeleton-based model for promoting coherence among sentences in narrative story generation," *arXiv preprint arXiv:1808.06945*, 2018.

[15] W. Staff, "It began as an ai-fueled dungeon game. it got much darker," Wired Magazine, 2021. [Online]. Available: https://www.wired.com/story/ai-fueled-dungeon-game-got-much-darker

[16] C. Bamford, A. Chaco´n *et al.*, "Griddly: A platform for ai research in games," *arXiv*

preprint arXiv:2106.09419, 2021.

[17] Wikipedia contributors, "Retrieval-augmented generation," Wikipedia, 2025. [Online]. Available:https://en.wikipedia.org/wiki/Retrieval-augmented generation

[18] T. He, J. Zhang, Z. Lin, and W. Xie, "Storytelling with retrieval- augmented video generation," *arXiv preprint arXiv:2307.06940*, 2023.

[19] Wikipedia contributors, "Rag improvements: Encoder, retriever, generation_ enhancements," Wikipedia, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Retrieval-augmented generation

[20] T. V. Staff, "Hidden door: An emerging ai storytelling platform with narrative constraints," The Verge, 2025. [Online]. Available: https://www.theverge.com/games/757816/hidden-door-early-access-ai-story

[21] X. Wen, R. Liu, J. Xu, B. Xu, and M. Sun, "Grove: A retrieval-augmented complex story generation framework with a forest of evidence," *arXiv preprint arXiv:2310.05388*, 2023.

[22] Y. Huang, X. Zhang, and W. Zhao, "Interactive augmented reality storytelling guided by scene semantics," in *Proceedings of the ACM International Conference on Multimedia*, 2022.

[23] R. Hunicke, M. LeBlanc, and R. Zubek, "Mda: A formal approach to game design and game research," in *Proceedings of the AAAI Workshop on Challenges in Game AI*, vol. 4, no. 1. San Jose, CA, 2004, p. 1722.

[24] Inworld AI Research Team, "The next generation of npc dialogue," *arXiv preprint arXiv:2404.11234*, 2024.

[25] C. R. Jones and B. K. Bergen, "Large language models pass the turing test," *arXiv preprint arXiv:2503.23674*, 2025.

[26] M. Klesel, "Retrieval-augmented generation (rag)," *Business & Information Systems Engineering*, 2025.

[27] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, M. Kukla, A. Fan, M. Lewis, W.-t. Yih *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[28] Y. Li, S. Kim *et al.*, "Geneva: Generating and visualizing branching narratives," in *Proceedings of the International Conference on Interactive Digital Storytelling (ICIDS)*, 2023.

[29] A. Lo Duca, "Using retrieval-augmented generation to build the context for data-driven stories," in *Proceedings of the International Conference on Information Systems*, 2023.

[30] R. Lopes, A. Paiva *et al.*, "Improving adaptive game ai with evolutionary learning," *ACM Transactions on Intelligent Systems and Technology*, 2021.

[31] D. Meseta, "I attempt to play a coherent story in ai dungeon: A noir fantasy mystery," Medium Blog, 2020. [Online]. Available: https://meseta.medium.com/i-attempt-to-play-a-coherent-story-in-ai- dungeon-attempt-1-a-noire-future-fantasy-mystery-ed6b91a59541

[32] O. Nyblom, "Player experiences in the game coridden: A case study examining the intended and perceived player experiences in a co-op action rpg game," 2023.

[33] P. Staff, "An ai dungeon master experiment raises questions about ethics and coherence," Polygon.com, 2024. [Online]. Available: https://www.polygon.com/critical-role/510326/critical-role-transcripts- ai-dnd-dungeon-master

[34] I. Puente, H. Gonza´lez-Jorge, J. Mart´ınez-Sa´nchez, and P. Arias, "Review of mobile mapping and surveying technologies," *Measurement*, vol. 46, no. 7, pp. 2127–2145, 2013.

[35] Reddit users, "Story generation power with ai dungeon," Reddit r/AIDungeon,_2025. [Online]. Available: https://www.reddit.com/r/AIDungeon/comments/1k6wxhg/story generation power

[36] M. Riedl, D. Thue, and V. Bulitko, "Game ai as storytelling," in *Artificial intelligence for computer games*. Springer, 2011, pp. 125–150.

[37] M. O. Riedl and M. Young, "Narrative planning: Balancing plot and character," *Journal of Artificial Intelligence Research*, vol. 51, pp. 217– 268, 2014.

[38] K. Roose, "We need to talk about how good a.i. is getting," *The New York Times*, Aug 2022. [Online]. Available: https://www.nytimes.com/2022/08/24/technology/ai-advances-dalle2.html

[39] G. Smith, M. Mateas, and N. Wardrip-Fruin, "Generating converging narratives for games,"

IEEE Transactions on Games, 2020.