G5038: Final Year Project

Week 3 Project Proposal

Title: AI-Powered Dynamic Narrative System for Role-Playing Games (RPGs)

Candidate Number: 281967 **Supervisor:** Dan Creed

Date:09 October 2025

Contents

1. Aims and Objectives

1.1 Aims Overarching

Aim: To investigate how far a bespoke, AI-driven 2D RPG narrative system can effectively generate a coherent, long-horizon thematic allegory, and to critically evaluate its architectural trade-offs against large-scale, theoretical "world model" approaches. This aim is built on three pillars:

Pillar 1: Technical Development. To develop an AI-driven 2D RPG narrative system. This involves designing and implementing a playable 2D RPG featuring dynamically generated storylines and NPC dialogue, where "dynamically" refers to the real-time generation of narrative content based on game state and player choice, rather than selecting from pre-written scripts. The focus is on enabling emergent, non-linear narratives while addressing long-horizon coherence.

Pillar 2: Thematic Execution. To craft a thematic allegory of university experiences. This involves creating a game world that allegorically explores a conflict between "positive energy" ideology and entropic decay, using core concepts like "The Void Entropy" and the "Positive Energy Curse" to guide AI narrative generation. This aim seeks to translate lived experiences into a compelling, darkly satirical game narrative.

Pillar 3: Critical Analysis. To produce a comprehensive comparative analysis report. This involves authoring a critical report comparing this project's bespoke AI narrative methodology with the described approaches of large-scale "world models" (e.g., Google Genie), emphasising architectural trade-offs, controllability, and practical contributions.

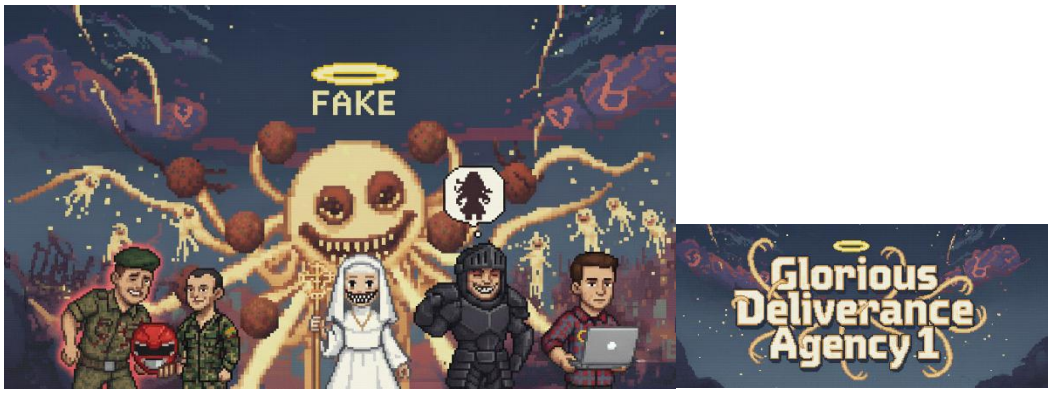
1.2 Objectives

Primary

1. **To build a Foundational 2D RPG framework in Godot 4.5**

   This objective is to build a lightweight and extensible project that serves as the stage for the AI narrative. It will include:

   - A player controller.
   - A dynamic UI system designed to visually communicate the story. This system's primary role is to bridge the gap between "infinite text" and "finite art" by binding AI-generated labels, names, and descriptions directly to generic on-screen assets. This will be complemented by simple VFX and SFX to create atmosphere and immerse the player in the AI-contextualised scene.
   - A minimal, narrative-focused inventory system. This system is not designed for traditional resource management (e.g., potions, gear). Its purpose is to store key symbolic items or concepts (e.g., a cryptic note, a cursed artefact) that can be referenced by the player or the AI to influence future narrative events and dialogue choices, making it a direct tool for interacting with the story..



2. **To implement an LLM integration & runtime pipeline**

   Implement a request/response pipeline that translates game state → structured prompt → LLM reply → in-game events/NPC lines. The initial provider will be Google Gemini, with a Settings page allowing users to supply their own API key and optionally select OpenRouter as an alternative provider. This pipeline will prioritise efficient token usage and responsive narrative generation.

3. **To design and implement a multi-layer context management system for narrative coherence.**

   Adopt a three-layer approach:

   - Static Context ("world bible" of immutable lore and themes),
   - Short-Term Context (most recent scene/dialogue),
   - Long-Term Memory (periodic AI summaries of key plot points/decisions stored for future prompts). A "notes/to-do list" mechanism will record emergent facts to ensure consistency across generations, with a focus on minimising narrative drift.

4. **To Author the seed narrative and world setting**

   Provide initial backstories and the philosophical conflict for Glorious Deliverance Agency 1 (GDA1), including Void Entropy and the Positive Energy Curse as the canonical seeds that guide generation, ensuring a strong thematic foundation for the AI.

5. **To conduct a deep comparative analysis and system evaluation.**

   - This objective is twofold. First, to evaluate the developed system using a clear set of metrics, and second, to conceptually compare its architecture to theoretical models.
   - a) System Performance Evaluation: Collect development artefacts and generation samples throughout the project to inform a critical analysis of this bespoke pipeline. Planned measurements include:
     - Narrative consistency: Sample AI-generated passages and annotating contradictions vs. world rules (lower is better).
     - Player experience: Feedback on immersion, control, consistency, and pacing.
     - Latency & cost: Average response time per turn and API spend.

        - Ablations: Compare with/without the notes/summarisation mechanism; contrast cloud vs. local model performance.
    - b) Conceptual Architectural Comparison: Contrast this project's narrative-first architecture with the publicly described concepts of "world-model" systems like Google Genie. This analysis will focus on theoretical differences in scope, controllability, and practical feasibility for a narrative-focused project, rather than direct performance benchmarking.

Extensions (time-permitting)

- **Local LLM via Ollama** to allow an offline, self-contained build; optionally explore **fine-tuning** on a curated style corpus for a consistent authorial voice.
- **Architectural documentation website** (static site) to visualise data flow, modules, and context logic for assessors.
- **Supabase-backed online features** (login and cloud save) leveraging prior internship experience with that backend.
- **Prompt templates & control scripts** to steer narrative direction while working over a constrained asset set (see symbolic assets approach below).
- **Exploring AI-generated assets** (images, scripts) for dynamic content, acknowledging the current limitations and future potential.
- **Investigating methods for context summarisation** to optimise LLM token usage, potentially drawing inspiration from open-source projects to maintain narrative coherence with reduced computational cost.
- **Prioritising depth in AI story generation** as a core "AI-native" feature, ensuring a high-quality, focused implementation that contributes significantly to the project's critical analysis.
- **Researching fast AI models** (e.g., 8 billion parameters) with large context windows for efficient and responsive narrative generation, balancing performance with narrative complexity.
- **Integration of Kanban boards** or similar project management tools to visualise workflow and track progress, enhancing organisational efficiency and collaboration.

2. Background and Rationale

The project formalises a long-standing personal reflection into an allegorical, AI-native game. This means the Large Language Model (LLM) is not merely a tool for content generation but is central to the game's core experience, driving emergent narratives and dynamic NPC interactions. Structurally, play proceeds through looping missions where outward “success” paradoxically accelerates decline, producing a deliberately endless, bleakly comic arc consistent with the world’s rules. This establishes a fertile testbed for LLM-driven story generation to express irony, reversals, and psychological pressure.

A planned dimension of the written report is to contrast this system with world-model approaches such as Google Genie, which aim to generate entire interactive environments consistent with physics, an aspiration beyond the scope of this 2D

narrative-first project. This comparison frames the practical trade-offs between bespoke pipelines and end-to-end world models, forming a significant part of the critical analysis.

3. Relevance to Degree Course

This project serves as the capstone for the BSc Digital Media and Games Computing degree, synthesising the technical and critical skills developed throughout the course. The technical core 2D game in Godot with a bespoke AI narrative system builds on programming fundamentals and the practical experience gained from my successful Year 2 Software Engineering project. The integration of a Large Language Model (LLM) is a direct continuation of my proactive work with AI, evidenced by winning the "Most Creative Award" in the Year 1 Machine Challenge and implementing the Google Gemini API during a summer internship. Furthermore, the project's robust management and documentation structure applies key principles of architecture and version control learned from leading my Year 2 team to success. This project is the logical next step, applying these proven skills in a more complex and creative context.

4. Requirements and Resources

The project requires a combination of standard, readily available hardware, software, and services. No specialised departmental purchases are necessary.

4.1 Hardware

- **Development PC Machine:**

4.2 Software & Tools

- **Game Engine:** Godot 4.5 (2D) for the core game build.
- **IDE:** Visual Studio Code.
- **Version Control:** Git for local source code management and GitHub for the remote repository.

4.3 APIs, Models, & Services

- **Primary LLM:** Google Gemini (via user-provided API key).
- **Alternative LLM Provider:** OpenRouter (optional, via user-provided API key).
- **Local LLM (Extension Goal):** Ollama for running local models (e.g., Llama, DeepSeek) for offline functionality.

4.4 Security, Privacy, & Backups

- **API Key Security:** Keys must not be committed to version control. They will be managed via user input in the game's settings screen.
- **Data Privacy & GDPR Compliance:** The system is designed to avoid transmitting sensitive personal data to external APIs. No user-identifiable information (e.g., names, emails, or personal details) will be collected or sent. Only anonymised game state data (e.g., scene descriptions, player choices, and

narrative prompts) will be shared with third-party APIs (Google Gemini or OpenRouter) for AI content generation. This adheres to GDPR principles of data minimisation and purpose limitation. Users will be informed via in-game notices about data processing for AI features, and consent will be implied through use of the settings screen. Data transmission will use secure HTTPS protocols, and no data will be stored by the APIs beyond the session unless required by the provider's terms.

- **Repository:** The project will be hosted in a private GitHub repository.
- **Backups:** Regular backups will be maintained through routine Git commits to the remote repository and weekly off-site archives (e.g., to OneDrive).
- **Contingency:** If cloud API access fails, the system is designed to fall back to a local LLM build (via Ollama).

5. System Design Overview

5.1 Runtime dataflow

The game state (scene, participants, flags, and player stats) is transformed into a structured prompt; the LLM replies with narrative acts or dialogue that the engine renders immediately. The notes register accumulates emergent facts and constraints so the LLM cannot contradict itself later. Periodic summarisation compresses long histories to fit context limits while preserving causally important beats.

5.2 Multi-layer context

- **Static context**: immutable “world bible” (Void Entropy; Positive Energy Curse; non-negotiable rules).
- **Short-term**: the current scene’s rolling transcript and salient events.
- **Long-term**: LLM-authored summaries of arcs and decisions, referenced when prompting.

5.3 Symbolic assets (bridging "infinite text" to "finite art")

Instead of chasing bespoke assets for every AI idea, the game uses a small, tagged library of generic props. The AI's role is to contextualise these symbols. For example, a generic 'monster' model can be dynamically labelled by the AI as a "Hope-Devouring Beast" via the UI, with its behaviour, weaknesses, and lore generated in real-time to fit the narrative context. This makes unlimited textual creativity playable within a finite art budget.

5.4 Systemic values: "Reality" vs "Positive Energy"

This is not just a theme, but a core gameplay mechanic designed to test the project's central hypothesis in a focused way.

Player choices directly affect two headline stats: Reality and Positive Energy.

These stats are fed back into the AI prompts, directly influencing the tone and direction of subsequent generations. For instance, a high 'Reality' score may cause NPCs to

become more cynical, while a high 'Positive Energy' score will accelerate the world's hidden decay.

This system will be tested through specific, repeatable narrative moments. A key example is the "Prayer System," where the player inputs "positive" prompts that the AI is instructed to interpret ironically, leading to predictably disastrous but thematically consistent outcomes. This provides a controlled environment to analyse the AI's ability to handle thematic constraints.

5.5 Player-facing configuration

A Settings page lets players paste a Gemini API key and optionally an OpenRouter key, selecting their provider at runtime.

6. Project Plan and Timeline

This section outlines the strategic plan for the project's execution throughout the academic year, emphasising an iterative development approach and a strong focus on the final report. The plan breaks down the work into distinct phases and tasks, providing a clear schedule and identifying key milestones. This structured approach is designed to ensure the timely completion of all primary objectives while allowing structured opportunities for implementing extension goals, with a particular emphasis on depth and critical analysis, especially from January onwards.

6.1 Phased Development Plan

The project is structured into four main phases, spanning the Autumn and Spring terms: Phase 1: Foundation and Prototyping (Autumn Term: Weeks 1-5): Focuses on establishing the project's technical groundwork, including research, planning, and developing the core 2D RPG framework. Phase 2: Core AI Implementation and Reporting (Autumn Term: Weeks 6-9): This phase tackles the central technical challenge of integrating the LLM and developing the multi-layered context management system. It culminates with the submission of the Interim Report. Phase 3: Narrative Integration and System Refinement (Autumn Term: Weeks 10-12 & Spring Term): Involves authoring the seed narrative, integrating it with the AI system, and conducting extensive testing and iteration to ensure narrative coherence and player engagement. The game build is expected to be largely complete by the end of January, allowing for focused report writing. Phase 4: Extensions, Finalisation, and Dissertation (Spring Term): Dedicated to implementing selected extension objectives, finalising the project deliverables, and writing the comprehensive final report and critical analysis, which constitutes 80% of the overall project mark.

6.2 Detailed Project Timeline (Gantt Chart)

Autumn Term

Week	Key Tasks & Milestones	Deliverables	Status
1-2	Project Inception & Planning: Background reading on generative agents and narrative systems. Initial proposal drafting and supervisor consultation.	Project Proposal Document: Confirmed Supervision.	Completed

3	Technical Scaffolding: Set up Git repository and version control. Establish the Godot project. Begin development of the foundational 2D RPG framework.	Private GitHub Repository; Basic window and game loop.	To Do
4	RPG Framework Development: Implement core RPG mechanics: tile-based map rendering and a basic character controller for player movement.	Playable character on a static map.	To Do
5	UI & Inventory Systems: Design and implement the UI for displaying dialogue and a simple inventory system. This is a critical dependency for displaying AI-generated content.	Functional UI text box; Basic inventory logic.	To Do
6	LLM Integration & Pipeline: Establish the communication pipeline with the Google Gemini API. Implement functions for sending game state and receiving responses.	Successful API call from the game engine that prints a response to the console.	To Do
7	Interim Report & Context System Design: Write and submit the Interim Report. Design the architecture for the multi-layered context management system (Static, Short-Term, Long-Term). The report will also provide a detailed rationale for key technical decisions (choice of Godot, Gemini API) and the importance of the narrative theme, laying the critical groundwork for the final dissertation.	Interim Report Submission: Architectural diagrams for the context system.	To Do
8	Context System Implementation (Layers 1 & 2): Implement the Static Context ("world bible") and Short-Term Context (scene memory) modules.	The system can feed the LLM with lore and recent conversation history.	To Do
9	Core Challenge - Long-Term Memory: Develop the Long-Term Memory module, focusing on the mechanism for prompting the LLM to create and store summaries of key plot points.	A functional logging and summarisation system that condenses game events.	To Do
10	Narrative Seeding: Author the core narrative premise, character backstories, and the "world bible" content (e.g., "The Void Entropy," "Positive Energy Curse").	Detailed Design Document containing the initial narrative setup.	To Do
11	Integration and End-to-End Testing: Integrate the narrative content with the complete AI context system. Conduct the first end-to-end tests of the dynamic narrative generation.	A playable prototype where NPC dialogue is fully generated by the AI based on all context layers.	To Do

12	Autumn Term Review & Bug Fixing: Consolidate progress, perform extensive bug fixing, and refine the AI's prompting strategy based on test results. Plan detailed tasks for the Spring Term.	Stable prototype build; A clear plan for Spring Term extensions.	To Do
Spring Term	(Provisional Plan)		
Term Period	Key Tasks & Milestones	Deliverables	
Weeks 1-4	Extension Goal 1 (Local LLM): Investigate and implement a local LLM (e.g., via Ollama) as an alternative to the cloud API.	A fully self-contained version of the application.	
Weeks 5-7	Extension Goal 2 & 3 (Fine-Tuning & Documentation): If the local LLM is successful, fine-tune the model. Create the architectural documentation website.	A fine-tuned model with a more distinct voice; A public-facing documentation site.	
Weeks 8-12	Final Report and Dissertation: Write the final dissertation, including the critical analysis comparing this project's methodology to models like Google Genie. Finalise all code and prepare for submission.	Final Dissertation Submission: Complete, documented source code.	

6.3 Weekly Work Allocation and Time Management

To ensure consistent progress and successful completion of the project milestones outlined above, I have integrated dedicated project work sessions directly into my weekly academic timetable. This structured approach allocates specific blocks of time for different project activities, ensuring a balance between development, research, and documentation. Project management will be further supported by tools such as Kanban boards to visualise workflow and track progress.

The following schedule details the planned allocation of time for this project during the Autumn Term:

- **Monday (11:00 - 13:00): Research and Planning.** This two-hour block, following my morning lab, will be dedicated to background reading, reviewing academic papers, designing system architecture, and planning the tasks for the week ahead.
- **Tuesday (10:00 - 13:00): Core Development Session 1.** This significant three-hour morning session will be the primary slot for hands-on coding. Activities will include developing the Godot RPG framework, implementing the AI communication pipeline, and testing new features.
- **Wednesday (13:00 - 15:00): Documentation and Supervisor Meetings** Following my morning lab, this session will be used for updating the project log,

writing reports (such as the Interim Report), preparing for and attending meetings with my supervisor, and managing the project's version control.

- **Thursday (10:00 - 12:00): Core Development Session 2.** This two-hour session will be used for continued development, focusing on tasks like API integration, UI implementation, iterative testing, and debugging.
- **Friday (11:00 - 14:00): Flexible Development and Integration.** This three-hour block before the weekly project lecture provides a flexible window for integrating different components of the system, tackling complex bugs, or getting a head start on the following week's development goals.
- **Weekend (Flexible):** Approximately 3-5 hours will be reserved for overflow tasks, exploring extension goals, or conducting more in-depth playtesting sessions that require longer, uninterrupted periods. Summary of Weekly Time Commitment:
- Scheduled Project Work: 12 hours
- Mandatory Lecture: 1 hour
- Flexible Weekend Work: 3-5 hours
- Total Estimated Project Time Per Week: 16-18 hours

7. Risks and Mitigations

- API access/quotas: Keep a local-model path (Ollama); cache summaries; design prompts to be compact.
- Narrative drift/contradictions: Enforce static-lore constraints; maintain long-term notes; gate key decisions through control scripts.
- Asset mismatch: Rely on symbolic assets + VFX/SFX/UI polish to keep scenes legible and atmospheric.
- Data loss: Follow the backup routine (GitHub + OneDrive) and ensure regular commits to the private GitHub repository.
- Overly ambitious scope: Focus on depth over breadth for core AI narrative features; prioritise the 80% report weighting by consolidating the game build by late January to allow ample time for critical analysis and documentation.

8. Expected Deliverables

- **Playable Godot 4.5 prototype** demonstrating end-to-end AI-driven dialogue and story progression with multi-layer context.
- **Source code repository** with documentation and architecture diagrams (plus optional static **documentation website**).
- **Final dissertation** including the **critical comparison** with world-model approaches like Genie.
- **Recorded gameplay video** showcasing key AI-driven narrative moments for presentation and demonstration.

9. List of Other Students Doing Related Projects

Not applicable

10. Interim Log

Date	Activity	Notes
12 Sep 2025	Initial email inquiry to supervisor.	Sent email to Dr Dan Creed expressing interest in the project and proposing a 2D implementation.
12 Sep 2025	Supervisor's response received.	Dr Creed confirmed willingness to supervise and suggested submitting it as a personal proposal.
15 Sep 2025	Supervisor confirmation on selection.	Received email from Dr Creed confirming project selection process.
22 Sep 2025	Project acceptance notification.	Received official acceptance email from Sussex Projects.
Week 1	Initial planning & background reading.	Began research on generative agents and interactive storytelling. Started drafting this proposal.
03 Oct 2025	First Supervisor Meeting.	Discussed the proposal draft and project scope with the supervisor. Key takeaways: focus on depth over breadth by refining the project to one core aim and four measurable objectives. The supervisor emphasised that while the game is important, the final report is worth 80% of the mark and should be the focus, especially from January onwards. Agreed to refine the aims and objectives for the next meeting.
Week 3	Proposal Submission	Finalise and submit the project proposal document via the Final Year Project Canvas site. This document outlines the project's refined aims, objectives, scope, and initial plan, incorporating feedback from the first supervisor meeting.
Week 7	Interim Report Submission	Submit the formally assessed Interim Report (3,000-5,000 words) via Canvas. This report will expand on the proposal, including a detailed introduction, professional/ethical considerations (BCS Code of Conduct, GDPR), requirements analysis, and an updated project plan and log. The supervisor emphasised that this report is a critical component, laying the groundwork for the final dissertation.

11. Bibliography

AI-Driven Story Generation Technologies

- AlHussain, A. I. & Azmi, A. M. (2021). Automatic story generation: a survey of approaches. ACM Computing Surveys, 54(5), Article 103 (May 2021), 38 pages.Available at: DOI: 10.1145/3453156 Available at: https://www.researchgate.net/publication/351858590_Automatic_Story_Generation_A_Survey_of_Approaches (Accessed: 07 October 2025).

- Riedl, M. (2021). An Introduction to AI Story Generation. Medium (Mark Riedl's Blog), 4 January. Available at: https://mark-riedl.medium.com/an-introduction-to-ai-story-generation-7f99a450f615 (Accessed: 07 October 2025).

Role-Playing Game Design and Narrative Systems

- Balint, J. T. & Bidarra, R. (2023). Procedural Generation of Narrative Worlds. IEEE Transactions on Games, 15(2), pp. 262–272. Available at: DOI:10.1109/TG.2022.3216582 Available at: https://dl.acm.org/doi/10.1609/aiide.v20i1.31876 (Accessed: 07 October 2025).

- Short, E. (2019). Storylets: You Want Them. Emily Short's Interactive Storytelling (blog), 29 November. Available at: https://emshort.blog/2019/11/29/storylets-you-want-them/ (Accessed: 07 October 2025).

Large Language Models and Their Integration in Games

- Buongiorno, S., Klinkert, L., Zhuang, Z., Chawla, T. & Clark, C. (2024). PANGeA: Procedural Artificial Narrative using Generative AI for Turn-Based Role-Playing Video Games. In: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2024). Available at: https://www.researchgate.net/figure/The-two-main-phases-of-the-narrative-world-generation-pipeline-a-a-Narrative-and_fig1_364425120 (Accessed: 07 October 2025).

- Jeong, J. & Lee, T. Y. (2025). LIGS: Developing an LLM-infused Game System for Emergent Narrative. In: CHI EA 2025 – Extended Abstracts of the 2025 CHI Conference on Human Factors in Computing Systems, Article 369. ACM. Available at: https://pure.kaist.ac.kr/en/publications/ligs-developing-an-llm-infused-game-system-for-emergent-narrative (Accessed: 07 October 2025).

- Boog, J. (2019). How the Creator of AI Dungeon 2 Used GPT-2 to Create Neverending Adventure Games. Towards Data Science (Medium), 9 December. Available at: https://medium.com/data-science/the-creator-of-ai-dungeon-2-shares-gpt-2-finetuning-advice-e5800df407c9 (Accessed: 07 October 2025).

- Juliani, A. (2025). One Trillion and One Nights: An experiment using LLMs to procedurally generate browser-based JRPGs. Medium, 22 January. Available at: https://awjuliani.medium.com/one-trillion-and-one-nights-e215d82f53e2 (Accessed: 07 October 2025).

Techniques for Narrative Coherence and Long-Horizon Storytelling

- Yi, Q., He, Y., Wang, J., et al. (2025). SCORE: Story Coherence and Retrieval Enhancement for AI Narratives. arXiv preprint (submitted March 2025). Available at: arXiv:2503.23512. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5243040 (Accessed: 07 October 2025).

Bespoke Narrative Pipelines vs. World-Model Approaches

- OpenAI (2023). GPT-4 Technical Report. OpenAI (Tech. Report, 15 March 2023). Available at: arXiv:2303.08774 Available at: https://arxiv.org/html/2303.08774v6 (Accessed: 07 October 2025).
- Pichai, S. & Hassabis, D. (2023). Introducing Gemini: our largest and most capable AI model. Google Keyword Blog, 6 December. Available at: https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/ (Accessed: 07 October 2025).
- Parker-Holder, J. & Fruchter, S. (2025). Genie 3: A new frontier for world models. Google DeepMind Blog, 5 August. Available at: https://deepmind.google/discover/blog/genie-3-a-new-frontier-for-world-models/

  (Accessed: 07 October 2025).
- O'Brien, L. (2024). How Ubisoft's New Generative AI Prototype Changes the Narrative for NPCs. Ubisoft News (online), 19 March. Available at: https://news.ubisoft.com/en-gb/article/5qXdxhshJBXoanFZApdG3L/how-ubisofts-new-generative-ai-prototype-changes-the-narrative-for-npcs (Accessed: 07 October 2025).