Computing, Performance and Communication Systems (2026)
Clausius Scientific Press, Canada

DOI: 10.23977/cpcs.2026.100108
ISSN 2371-8889 Vol. 10 Num. 1

Design and Implementation of a Unity-Based Experimental Walking Simulation with Dialogue-Driven Interaction and Looping Scene Control

Hanran Ma

Chongqing Daoshi Cultural Consulting Co., Ltd, Chongqing, China

Keywords: Unity; walking simulation; interaction design; scene control; experimental game

Abstract: This paper examines the design and implementation of a Unity-based experimental walking simulation as a case study in scene control, dialogue-gated progression, and lightweight modular scripting. Although walking simulators are often discussed in terms of atmosphere and environmental storytelling, the implementation logic of small independent projects is less frequently documented. This paper addresses that gap through On the Other Side of the Wall, a first-person interactive work built around constrained traversal, looping reset structures, and embedded generative media. Rather than treating these elements as isolated artistic features, the paper analyzes them as parts of a coordinated runtime system. The study shows that a relatively small set of reusable scripts can support coherent progression and a consistent environmental rhythm without requiring large-scale production resources. By documenting the project at the level of system design, the paper offers a practical reference for low-cost interactive scene development and shows how experimental artworks can produce transferable technical knowledge.

1. Introduction

In recent years, game engines have increasingly been used beyond commercial entertainment. For independent creators and small interdisciplinary teams, Unity has become a practical environment for combining scene construction, scripting, asset management, and runtime control within a single workflow. This is especially important in experimental work, where one project often needs to support both creative exploration and technical implementation.

Within this broader shift, the walking simulator has become a useful format for experience-oriented work. Instead of organizing play around combat or challenge, walking simulators often rely on navigation, environmental cues, and pacing to shape interpretation. In such projects, space is not only a visual setting; it is a primary structure through which interaction is staged and meaning is distributed.

However, many independent experimental projects remain under-documented at the level of implementation. Their visual identity or conceptual intention may be clear, but their interaction architecture often remains project-specific and difficult to reuse. This is particularly true for works that depend on gated progression, repeated spatial return, dialogue logic, and tightly controlled

transitions.

To address this problem, this paper presents a case study of On the Other Side of the Wall, a Unity-based experimental walking simulation. The project combines dialogue-gated progression, looping and reset logic, fade-and-teleport transitions, dynamic exterior spawning, and embedded AI-generated media to construct a constrained spatial experience. The paper makes three contributions: (1) it documents a complete experimental scene system at the implementation level; (2) it shows how modular scripting supports iterative design and environmental consistency; and (3) it provides a practical reference for independent projects seeking to balance exploratory movement with structured control.

2. Related Work

Research on game engines has increasingly treated them as broader production environments rather than as tools limited to commercial game development. Studies of Unity and comparable platforms show that their importance lies not only in rendering capacity, but also in how they organize workflow, modularity, and real-time media coordination [1,2]. This perspective is relevant here because the present project depends on Unity not simply as a visual tool, but as a system for coordinating prompts, dialogue states, transitions, dynamic media, and scene updates within one runtime structure.

Walking simulator research is also relevant to this paper. Existing studies emphasize environmental storytelling, exploratory pacing, and the use of movement as the primary mode of engagement [3-5]. These discussions are useful because they frame traversal as an experiential structure rather than a neutral locomotion mechanic. At the same time, the current case differs from more open walking simulators by relying on stricter progression gates. Movement remains central, but access is repeatedly regulated by interaction conditions.

Work in interaction design and game studies further clarifies the role of rule-based structure. Research on affordances and constraints shows that limitation is not simply a reduction of freedom; it is also a way of guiding behavior and clarifying system logic [6]. Studies of rules and procedural meaning similarly argue that interaction mechanics shape interpretation as much as they enable action [7,8]. These perspectives help explain why prompts, barriers, dialogue conditions, and reset logic should be treated as core parts of the experience rather than as secondary implementation details.

Taken together, these fields provide useful conceptual support, but they do not fully address the practical problem examined here. Research on walking simulators often focuses on atmosphere and genre. Research on interaction design tends to focus on general principles. Research on game engines often addresses workflow or platform culture. Less attention has been paid to how small experimental projects combine these elements through concrete scene-control mechanisms. This paper addresses that gap by examining one Unity-based case at the level of system design.

3. Project Overview

On the Other Side of the Wall is a Unity-based experimental walking simulation presented from a first-person perspective. The project places the player in a sequence of linked spaces derived from domestic, educational, workplace, and transit environments. These spaces are not intended as documentary reconstructions. Instead, they are reorganized as a controlled experiential sequence in which movement, delay, and access conditions shape progression.

The wall is the project's key interactive motif. It is not only a visual object but also a progression threshold. Players cannot move through it by ordinary navigation alone; they must trigger dialogue in order to continue. In this way, the wall functions as both a spatial barrier and a scene-control device. More broadly, the project uses dialogue not as optional narrative decoration, but as a mechanism for

validating movement between zones.

Spatially, the project uses a segmented circular layout. Inner zones contain rooms, corridors, and NPC-related spaces, while outer zones are often visible but partially inaccessible. This arrangement supports two goals: it makes repeated traversal possible within a compact map, and it creates a persistent contrast between enclosure and exterior visibility. The result is a spatial structure that remains manageable in scale while still suggesting a wider world beyond the player's immediate route.

From the perspective of this paper, the project is significant not only as an artwork but also as a technical case. Its scope includes reusable interaction prompts, dialogue-trigger management, loop and reset logic, fade-and-teleport transitions, dynamic object spawning, NPC integration, and AI-generated video playback. This makes the project suitable for analysis as a compact but complete implementation study. The project's linked environments, spatial thresholds, and interaction contexts are shown in Fig.1.

Fig.1 Selected project views from On the Other Side of the Wall

4. System Design and Implementation

The project is implemented as a layered scene-control system. At the top level, player movement and interaction input are processed through a first-person controller. Nearby interactive objects are then evaluated through a base interaction script that determines whether prompts should appear and whether a trigger condition can be activated. These outputs are passed to the scene logic layer, where state changes may unlock new zones, initiate teleportation, reset interaction states, or activate environmental updates. Visual and media outputs are then synchronized with the updated runtime state. The overall layered scene-control architecture is shown in Fig.2.

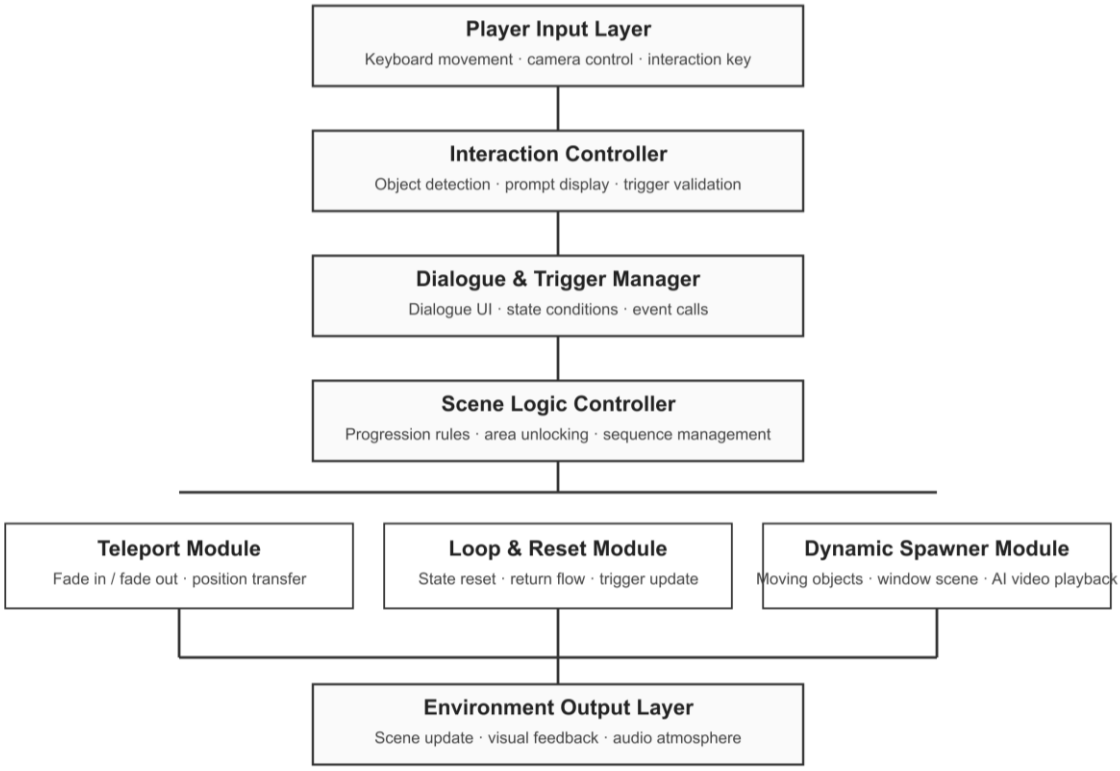

Fig.2 Overall system framework

A generalized interaction grammar is one of the project's key design strengths. Instead of building a separate logic block for each object, the project uses a detect-prompt-trigger pattern across multiple scene elements. This reduces code duplication and makes the experience easier to read. Players repeatedly encounter the same basic interaction sequence—approach, prompt, input, response—even when the objects involved are different. As a result, uncertainty is concentrated at the level of progression rather than at the level of interface comprehension.

Dialogue functions as a core progression mechanism. It is not treated as optional narrative decoration. Certain areas and transitions only become available after dialogue conditions are completed, which means that interaction is repeatedly formalized through confirmation and gating. The implementation notes also indicate that interaction is temporarily disabled while dialogue is active in order to avoid repeated triggering and potential state inconsistency. This small design decision is technically important because it shows how progression stability depends on controlling concurrent inputs.

Looping and reset logic form another major component of the system. A reset script returns the player to an earlier position and recalibrates specific dialogue or interaction states in order to produce the effect of recurring progression. This serves both design and implementation goals. It allows repeated traversal without requiring a large number of separate scenes, and it ensures that prior interactions do not permanently block later sequences. Repetition therefore becomes both an experiential feature and a reliability strategy.

Scene transitions are handled through a fade-and-teleport system. Instead of abrupt relocation, the screen darkens, the player is repositioned, and the scene returns gradually. This approach is lightweight from a technical standpoint, but it is also important for pacing. It turns what could feel like a disruptive jump into a staged transition and helps preserve continuity across segmented spaces. The corresponding dialogue-gated scene progression flow is shown in Fig.3.

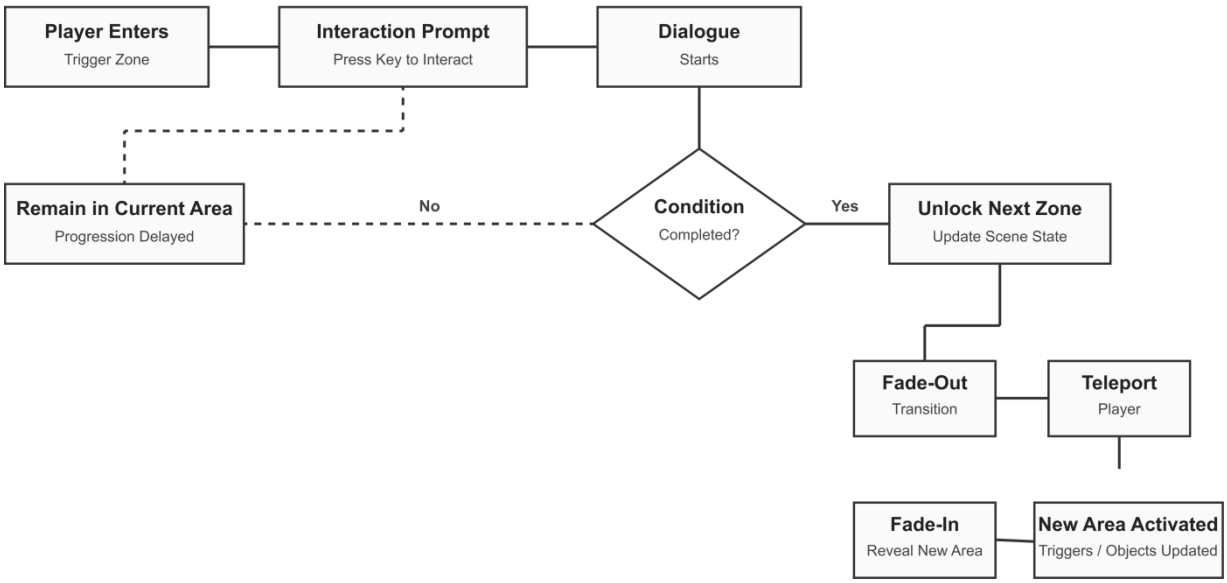

Fig.3 Scene Progression Flow through Dialogue-Gated Interaction

The project also includes specialized modules for scene-specific functions. A ladder script modifies player movement and camera behavior during vertical traversal, while a dynamic spawner in the bus scene generates moving objects and light elements outside the window in order to simulate external motion. These modules show how the system extends its base interaction logic without abandoning modularity. General patterns are preserved where possible, while localized scripts are introduced when a scene requires a more specific behavior.

Dynamic media further enrich the runtime environment. The bus scene uses moving scenery and light elements outside the window to imply motion beyond the playable area. AI-generated videos are also inserted into window and screen surfaces as part of the environmental layer. These elements do not function as isolated spectacle. Instead, they work with the project's scene logic to increase temporal density and reinforce the contrast between constrained movement inside the playable space and perceived activity outside it.

Taken together, the implementation can be understood as a compact but coordinated modular system. Its value lies not in the complexity of any single script, but in the way base interaction, dialogue control, reset logic, transitions, and dynamic environmental media cooperate to produce a coherent first-person experience. This is what makes the project useful as a case study for low-cost experimental scene design.

This compactness is important in the context of independent production. Instead of depending on a large number of bespoke systems, the project relies on a small set of interoperable modules whose functions remain clearly separated but operationally connected. Such an arrangement makes the system easier to test, extend, and document. It also provides a useful model for similar projects that need to maintain atmosphere and progression under limited development resources. In that sense, the implementation demonstrates not only how the scene works, but also how a low-cost workflow can remain technically organized without becoming over-engineered.

5. Interaction Experience Analysis

A key feature of the project is that exploration is guided rather than fully open. Although the player moves through the environment from a first-person perspective, access to new zones is repeatedly regulated by walls, dialogue nodes, scripted triggers, and controlled transitions. The result is an experience in which movement is always coupled with validation. The player can navigate and inspect

the environment, but progression depends on whether the correct interaction condition has been met.

This design produces a distinctive experiential effect. The player repeatedly encounters spaces that are visible before they are accessible, or familiar before they are functionally altered. This creates a sense of delay and negotiation rather than free roaming. The project therefore does not build tension through difficulty in the conventional gameplay sense. Instead, it uses blocked expectation and staged access to shape rhythm and attention.

Looping and reset logic reinforce this structure by turning repetition into part of the scene experience. When players revisit a space under changed conditions, the environment becomes rhythmically organized rather than simply reused. The same is true of fade-and-teleport transitions, which help frame movement between zones as deliberate passage rather than technical discontinuity. Dynamic exterior spawning and embedded media further support the illusion of a larger world beyond the player's immediate route.

From the perspective of interaction design, the project achieves a relatively strong degree of structural legibility. Its core interaction pattern—approach, prompt, trigger, transition—is repeated often enough to become readable without extensive tutorialization. This is important in an experimental work: the system remains understandable while still preserving uncertainty at the level of progression. The project therefore shows that constrained interaction can produce a clear and atmospheric spatial experience without relying on a large number of mechanics.

Environmental media also contribute to this legibility. Because moving exterior elements, AI-generated window videos, and lighting changes are attached to specific scene states rather than appearing randomly, they reinforce the player's sense that the world is responding to progression logic. These media layers do not introduce additional playable complexity, but they make state change perceptible at the level of atmosphere. As a result, the project shows that scene-control systems can communicate progression not only through gates and prompts, but also through controlled shifts in environmental motion and visual density.

6. Discussion

This case suggests that Unity is a highly practical environment for low-cost experimental interactive production when scene behavior is treated as a coordinated system rather than as a collection of isolated effects. The project does not depend on advanced custom engineering or large production resources. Instead, it shows how prompt logic, dialogue gating, reset control, transitions, and media playback can be assembled into a functional and expressive runtime structure.

It also shows that modular scripting is valuable at both technical and experiential levels. Technically, it reduces duplication and helps maintain scene logic. Experientially, it stabilizes the project's interaction grammar across different spaces. This is especially important in small experimental works, where inconsistency at the level of basic interaction can quickly undermine the larger atmosphere of the project.

More broadly, the project demonstrates that independent artworks can generate reusable technical knowledge when they are documented at the level of implementation. The middle layer between concept and visual result—prompt behavior, gating logic, reset conditions, and transition control—is often where transferable practice knowledge is produced. When that layer is made explicit, a single project can contribute not only an artwork but also a method.

This paper also has clear limits. It focuses on one case, operates mainly in a PC-based context, and does not include a full user-study dataset. Its claims should therefore be read as analytical and practice-based rather than statistically generalizable. Even so, the case remains useful because it shows, in concrete terms, how a compact Unity project can organize constrained traversal and scene progression through modular scripts.

7. Conclusion

This paper has examined On the Other Side of the Wall as a Unity-based case study in modular scene orchestration and constrained first-person interaction. The analysis shows that a relatively small set of coordinated scripts can support legible progression, recurring spatial rhythm, and a coherent atmospheric structure without requiring large-scale production resources.

The project therefore contributes in two ways. As an artwork, it demonstrates a controlled spatial experience built through constrained interaction. As a technical case, it shows how independent experimental projects can produce transferable knowledge when their implementation logic is documented clearly. Future work may extend this approach through user testing, comparative case studies, and adaptation to installation-based or public interactive environments.

References

[1] Nicoll, B. and Keogh, B. (2019) The Unity Game Engine and the Circuits of Cultural Software. Cham: Palgrave Pivot.
[2] Krüger, M., Gilbert, D.Y.R., Kuhlen, T.W. and Gerrits, T.S. (2024) Game Engines for Immersive Visualization: Using Unreal Engine Beyond Entertainment. PRESENCE: Virtual and Augmented Reality, 33, pp.31-55.
[3] Jenkins, H. (2004) Game Design as Narrative Architecture. In: Wardrip-Fruin, N. and Harrigan, P., eds., First Person: New Media as Story, Performance, and Game. Cambridge, MA: MIT Press, pp.118-130.
[4] Carbo-Mascarell, R. (2016) Walking Simulators: The Digitisation of an Aesthetic Practice. In: Proceedings of the 2016 DiGRA/FDG Conference. Dundee: DiGRA and FDG.
[5] Montembeault, H. and Deslongchamps-Gagnon, M. (2019) The Walking Simulator's Generic Experiences. Press Start, 5(2).
[6] Norman, D.A. (2013) The Design of Everyday Things. Revised and Expanded Edition. New York: Basic Books.
[7] Salen, K. and Zimmerman, E. (2003) Rules of Play: Game Design Fundamentals. Cambridge, MA: MIT Press.
[8] Bogost, I. (2007) Persuasive Games: The Expressive Power of Videogames. Cambridge, MA: MIT Press