Aryan Raghuvanshi

Aryan_Raghuvanshi_thesis

MDES Thesis/Dissertation

Document Details

Submission ID
trn:oid:::27535:141572293

Submission Date
Jun 3, 2026, 10:45 PM GMT+5:30

Download Date
Oct 19, 2026, 11:15 PM GMT+5:30

File Name
Aryan_Raghuvanshi_thesis.pdf

File Size
11.3 MB

21 Pages

4,359 Words

26,558 Characters

2% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **7** Not Cited or Quoted 2%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
0% Publications
1% Submitted works (Student Papers)

Match Groups

- **7** Not Cited or Quoted 2%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
0% Publications
1% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1 Student papers
Technological Institute of the Philippines on 2025-05-22 <1%

2 Student papers
Technological Institute of the Philippines on 2026-04-19 <1%

3 Internet
cdn-learn.adafruit.com <1%

4 Internet
dspace.daffodilvarsity.edu.bd:8080 <1%

5 Internet
www.coursehero.com <1%

4

Chapter 1

INTRODUCTION

1.1 Project Overview

Project Sentience is a short-form, narrative-driven psychological horror game that explores the vulnerability of the human subconscious. The project places the player in the perspective of a child suffering from extreme somnambulism (sleepwalking) shortly after relocating to a new, unfamiliar home. By stripping away traditional player agency and weaponizing the mundane environment of a residential house, the game aims to evoke a profound sense of disorientation and dread. The primary objective of this project is to investigate how interactive media can simulate the terrifying dissonance of waking up in a perilous situation with no memory of how one arrived there.

1.2 Core Narrative and Thematic Elements

The narrative is anchored in the protagonist's deeply unsettling transition into a new living space—a classic horror trope that establishes an immediate baseline of discomfort. The gameplay loop begins in a state of normalcy, establishing the geography of the house. However, once the player initiates the sleep state, the game transitions into a surreal, red-tinted distortion of reality.

The climax of the short-form narrative occurs when the protagonist regains consciousness outside the safety of their home, confronted by an uncanny, erratic antagonist. The narrative subverts expectations: rather than a standard monster, the threat is a deeply unnatural human figure whose sudden, violent reactions to the player trigger a desperate, randomized escape sequence. The story concludes with an ambiguous but terrifying environmental storytelling cue—scratch marks on the door lock—leaving the player to question the boundary between the sleepwalking nightmare and reality.

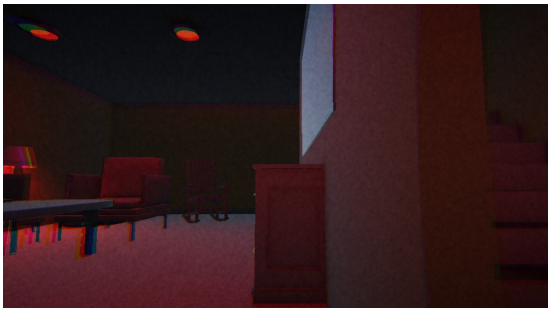


Figure 1.1: Two visual states

1.3 Game Design Concepts

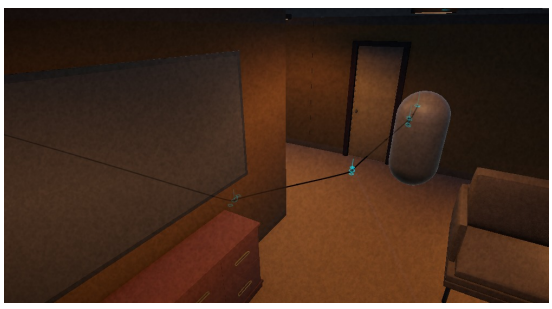
From a design perspective, Project Sentience utilizes the restriction of player control to build tension. The design relies on several core concepts:

- **Agency Deprivation:** During the sleepwalking sequence, the player's character is locked onto a predefined spline path. The player is forced to watch their character move through the distorted environment, simulating the exact feeling of being a passenger in one's own body during a sleepwalking episode.
- **Spatial Unfamiliarity:** The house is designed to feel slightly "off" even before the sleepwalking begins, leveraging the feeling of moving into a new home where the spatial mapping in the player's brain is not yet solidified."
- **Dynamic Pacing and Triggers:** The transition from passive observation to active panic is governed by spatial trigger colliders. The moment the player crosses a specific threshold near the antagonist, the pacing violently shifts from a slow, automated walk to a frantic chase.
- **Asymmetric Quick Time Events (QTEs):** To simulate the panic and clumsiness of running while half-awake, the escape sequence utilizes randomized input prompts. This prevents the player from relying on muscle memory, ensuring the tension remains high during the spline-based retreat back to the house.

1.4 Implementation Snapshot

The technical implementation of Project Sentience focuses on seamless state transitions and environmental manipulation. The game relies heavily on spline-based locomotion systems to handle the forced movement during the somnambulistic phases. Post-processing volumes are utilized to dynamically shift the color grading and visual distortion, signaling the descent into the sleepwalking state.

Event-driven programming is heavily utilized; invisible trigger colliders dictate the pacing of the horror. The antagonist's behavioral state machine remains dormant until the player's proximity breaches a specific collider, which instantly fires the animation, audio, and pursuit logic. The escape mechanic is implemented via a dynamic UI system that listens for randomized keystrokes, gating the player's forward progress along the return spline until the correct input is registered.



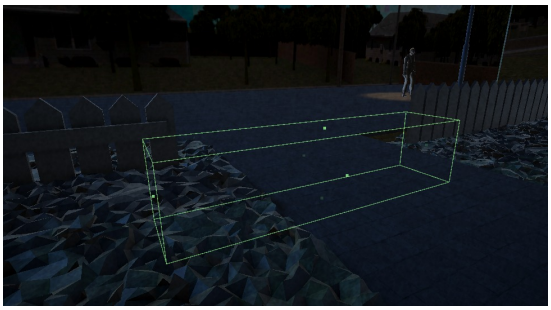


Figure 1.2: Editor view detailing the spline path network and chase-trigger volumes.

1.5 Proposed Future Improvements

While the current vertical slice of Project Sentience successfully demonstrates the core mechanics of forced movement and randomized escape, this dissertation proposes several avenues for future iteration to enhance the psychological impact:

1. **Expanded Procedural Distortion:** Implementing more dynamic environmental shifts during the sleepwalking phase, such as furniture changing scale or doors disappearing, to deepen the surrealism.

2. **Advanced Audio Integration:** Utilizing spatial audio and adaptive soundscapes that warp depending on the player's current state of consciousness.

3. **Refined Antagonist Animation:** Enhancing the erratic, unnatural movements of the antagonist using procedural animation blending to heighten the "uncanny valley" effect during the chase sequence.

4. **Branching Outcomes:** Introducing a penalty system during the spline-chase sequence where missed randomized inputs lead to unique, non-fatal psychological scares rather than a traditional binary fail state.

Chapter 2

LITERATURE REVIEW

2.1 The Psychology of Vulnerability and Somnambulism in Horror Narratives

Within psychological horror design, the subversion of altered states of consciousness—specifically sleep and parasomnias—serves as a powerful vehicle for generating narrative tension. Somnambulism (sleepwalking) inherently represents a profound dissolution of bodily autonomy, wherein the subject executes complex physical actions detached from conscious volition. In interactive media, adapting this state shifts the thematic focus from external monsters to internal vulnerability.

According to horror media theory, the most effective psychological dread arises not from the presence of a threat, but from the realization of defenselessness. By framing a narrative around a child protagonist experiencing severe parasomnia, design frameworks can leverage a naturally heightened state of vulnerability. Children possess limited physical agency and systemic understanding of their environments, making the domestic space both a sanctuary and an unpredictable labyrinth. Waking up mid-episode in an unfamiliar or exposed setting introduces acute disorientation, forcing the player to confront a fragmented reality where the boundaries of the safe, domestic sphere have been completely breached.

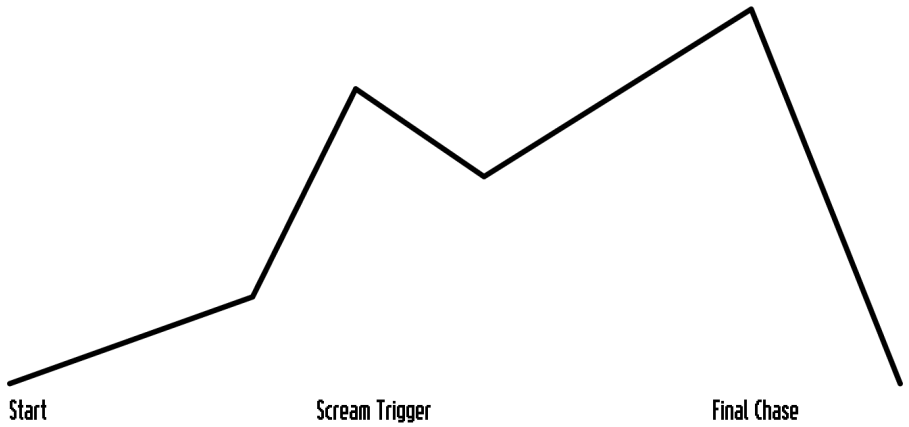


Figure 2.1: The graph illustrates the mapped tension spike during the transition from detection to mashing interactions.

2.2 Theoretical Frameworks of Agency Deprivation in Game Design

Ludenic agency—the capacity of a player to meaningfully interact with and alter the state of a digital environment—is traditionally considered the defining characteristic of video games. However, survival and psychological horror titles frequently establish dread by systematically restricting or completely removing this agency. This design methodology, known as *agency deprivation*, forces a cognitive shift in the player from an active agent of influence to a passive witness of an unfolding threat.

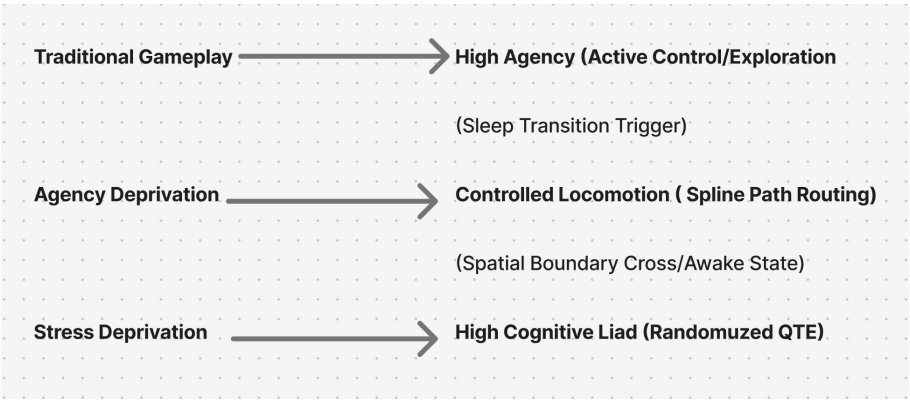


Figure 2.2

Forced movement along a predefined path, traditionally implemented via camera rails or algorithmic spline navigation, strips away the player's ability to retreat, hide, or optimize their spatial position. When a system dictates locomotion, the player's mechanical mastery over space is neutralized. This approach directly simulates the physiological sensation of sleep paralysis or somnambulistic dissociation, where the mind is aware of movement but entirely incapable of altering its trajectory. The deliberate bottlenecking of mechanical capability ensures that when control is abruptly restored, the sudden transition induces psychological friction and mechanical panic.

2.3 Spatial Unfamiliarity and the Architecture of the Domestic Uncanny

The effectiveness of environmental storytelling in residential horror relies heavily on Sigmund Freud's psychological concept of the *Unheimliche*, or the "uncanny"—literally translated as the "un-homely." This phenomenon describes an experience where something familiar and deeply rooted in domestic security suddenly becomes strange, alien, and hiddenly hostile.

In architectural level design, establishing spatial unfamiliarity within a domestic setting requires a careful subversion of layout expectations. When a narrative dictates that characters have recently relocated to a new residence, the structural layout is already alien to both the protagonist and the player. By manipulating this unfamiliar space through post-processing effects, such as a localized red color grading or geometric distortion, the environment mirrors the cognitive impairment of a dream state.

- **Chromatographic Shifting:** Utilizing highly saturated monochromatic hues (e.g., crimson or deep amber post-processing volumes) to signal a departure from objective reality into subjective subconscious terror.
- **Geometric Non-Euclidean Alterations:** Subtle shifts in doorway placements, elongated corridors, or shifting shadows that exploit the player's lack of spatial mapping in a new layout.

This spatial alienation transforms a house from a protective barrier against the outside world into an unpredictable, claustrophobic container of threat.

2.4 Interactive Tension: Mechanics of Chase Trajectories and Cognitive Stress Inputs

The transition from environmental suspense to an active chase sequence requires a mechanical trigger that bridges spatial geometry with system behavior. In modern engine architectures, this is achieved through the use of spatial trigger volumes and proximity colliders. When the player vector intersects a designated hidden volume, an automated sequence execution fires, transitioning the threat entity from a passive loop to an aggressive, animated pursuit state.

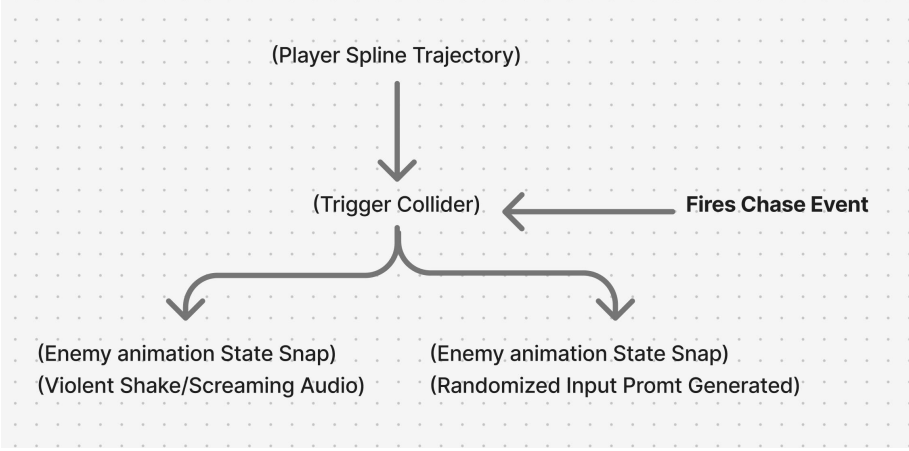


Figure 2.3

To prevent the player from relying on established muscle memory during these high-stress sequences, contemporary game design introduces asymmetric or randomized input mechanics, often stylized as Quick Time Events (QTEs). Traditional survival horror chases often rely on standard sprint mechanics; however, randomizing input configurations on every distinct attempt introduces severe cognitive load.

- **Muscle Memory Disruption:** Forcing the player to read, interpret, and execute an entirely unexpected keystroke format breaks the automated instinctual responses built over hours of standard gameplay.
- **Simulated Panic:** The delay caused by the brain processing an unpredictable visual prompt and translating it to physical finger placement accurately mirrors the real-world physiological panic, clumsiness, and motor-skill degradation experienced during a sudden nocturnal emergency.

By tying forward progress along a return spline path directly to these accurate, randomized keystrokes, the system creates a mechanical choke point where failure feels imminent, culminating in a sudden, high-tension resolution.

Chapter 3

METHODOLOGY

3.1 Introduction to the Design Pipeline

The development of *Project Sentience* followed an iterative, design-centric methodology, prioritizing player psychology and atmospheric storytelling over complex combat mechanics. The project was developed utilizing the Unity 6 engine, chosen for its robust handling of first-person controllers, advanced lighting pipelines, and modular event systems. The methodology is divided into four primary design phases: environmental construction, locomotion and spline integration, event-driven scripting (colliders), and the dynamic input interface for the escape sequence.

3.2 Environmental Construction and Visual Design

The foundation of the project relies on establishing a believable, albeit unfamiliar, domestic space. The level design phase focused on creating a standard residential layout that feels claustrophobic. To achieve the psychological dissonance required for the sleepwalking sequence, a dual-state environmental design approach was employed.

Instead of building two separate levels, the transition between the waking state and the sleepwalking state is handled via dynamic post-processing and lighting alterations.

- **The Waking State:** Utilizes standard global illumination, warm color temperatures, and clear visibility to establish baseline safety.
- **The Somnambulistic State:** Triggered upon the player interacting with the bed. This state dynamically activates a global post-processing volume that applies a heavy red-tinted color grade, chromatic aberration, and localized vignette. This visual design choice immediately signals to the player that objective reality has been compromised.

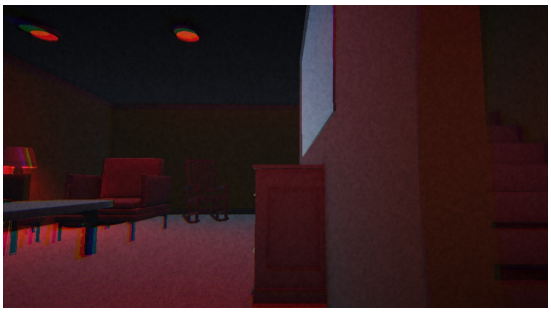


Figure 3.1: Two visual states

3.3 Locomotion Systems and Agency Deprivation

To accurately simulate the physical dissociation of sleepwalking, standard first-person controller inputs are forcefully disabled during the initial phase of the nightmare sequence. The methodology for this relies on mathematical splines—a series of connected coordinates that plot a smooth curve through the 3D environment.

1. **Path Generation:** A spline path was drawn from the protagonist's bed, navigating down the stairs, and exiting through the front door, terminating at the streetlamp.
2. **Controller Override:** Upon waking in the dream state, the standard first-person controller script is bypassed. The camera and character transform are rigidly parented to a spline follower script.
3. **Pacing:** The movement speed along this spline is intentionally kept slow and steady, preventing the player from rushing through the environment and forcing them to absorb the uncanny visual changes.

3.4 Event-Driven Triggers and Antagonist Behavior

The pivot from psychological suspense to active terror is managed through spatial trigger volumes (invisible colliders). The antagonist—a hairless, uncanny human figure—is initially spawned in a dormant, idle animation state beneath the streetlamp.

The methodology for the scare relies on proximity rather than time. An invisible box collider is placed across the player's spline path, acting as a tripwire. When the player's character intersects this collider, an event fires that simultaneously:

- Snaps the antagonist's rigging to gaze directly at the player.
- Triggers a violent, shaking animation protocol.
- Emits a high-pitched, spatialized audio scream.
- Switches the antagonist's state machine from 'Idle' to 'Pursuit'.

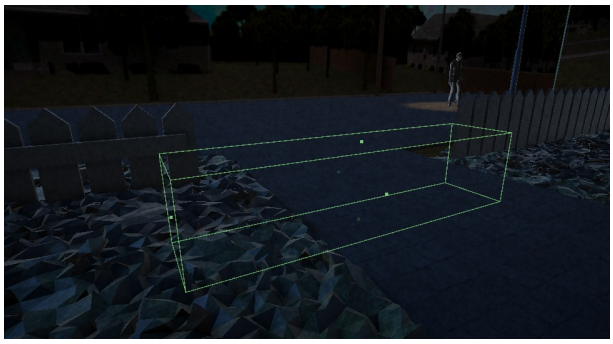


Figure 3.2: Spatial trigger volume utilized to initiate the antagonist's behavioral state shift.

3.5 Dynamic UI and Randomized Escape Mechanics

The final phase of the methodology focuses on the escape sequence. Once the pursuit is triggered, the player regains forward-movement agency, but it is heavily gated by cognitive stress mechanics.

Instead of allowing the player to simply hold a 'Sprint' key, the system demands active focus. A dynamic User Interface (UI) script randomly selects an input key (e.g., 'W', 'Space', 'Shift', 'E') and displays it on-screen.

- **Success Condition:** The player must press the exact displayed key to advance a set distance along the return spline path toward the house.
- **Iteration:** Upon a successful press, the UI instantly generates a new, randomized prompt.

This design choice ensures that the player cannot rely on muscle memory. The mechanical friction of reading the prompt and finding the correct key accurately simulates the panic and lack of motor control experienced during a real-life night terror. The sequence concludes when the player successfully crosses the threshold of the front door, triggering a final collider that slams the door shut, terminates the antagonist entity, and fades the screen to black for the narrative text resolution.

Chapter 4

RESULTS and DISCUSSION

4.1 System Performance and Implementation Results

The primary objective of *Project Sentience* was to successfully simulate the disorientation and vulnerability of a somnambulistic state through restrictive game mechanics. The integration of spline-based locomotion within the Unity engine performed exactly as intended during the waking-to-sleep transition. By overriding the standard first-person controller and locking the player to a predetermined mathematical path, the system successfully enforced a state of agency deprivation.

The spatial trigger volumes effectively managed the transition between the passive observation state and the active panic state. Upon crossing the designated collider threshold, the system reliably snapped the antagonist character into its pursuit state, triggering the corresponding visual and auditory feedback without any noticeable latency or systemic errors.

4.2 Visual Design and Cinematic Storytelling Effectiveness

A major success of the project lies in its visual design and environmental execution. The transition from the safety of the waking world to the nightmare state relied heavily on advanced lighting pipelines and post-processing volumes. The stark red-tinted color grading and environmental distortion proved highly effective in establishing immediate visual dissonance.

Instead of relying on dialogue or explicit exposition, the project successfully utilized cinematic storytelling. The environment itself communicated the shift in reality. The stark contrast between the warm, natural lighting of the initial house exploration and the harsh, unnatural red tones of the sleepwalking sequence effectively manipulated the player's emotional baseline. This visual design choice reinforced the psychological themes of the uncanny, making the mundane domestic space feel actively hostile.

4.3 Analysis of Cognitive Load via QTE Mechanics

The escape sequence served as the mechanical climax of the project, designed to induce panic through cognitive overload. Testing the dynamic User Interface (UI) revealed that forcing players to execute randomized inputs (rather than holding a standard sprint key) successfully disrupted established muscle memory.

During the spline-based retreat, players could not predict which key would be required next. The fraction of a second required to read the randomized on-screen prompt, cognitively process it, and physically locate the key accurately simulated the physiological clumsiness of a night terror. The mechanical friction deliberately designed into this sequence proved that stripping away predictable first-person controls and replacing them with asymmetric inputs drastically heightens player anxiety, fulfilling the core hypothesis of the project.

4.4 Narrative Resolution and Environmental Ambiguity

The conclusion of the game successfully integrates the narrative systems with environmental storytelling to leave a lasting psychological impact. Mechanically, the climax resolves the moment the player crosses the final threshold; the door slams shut, the antagonist abruptly disappears, and the player loses consciousness, cleanly terminating the interactive sequence.

The choice to transition to a text-based narrative update for the epilogue served as a vital grounding mechanism. The on-screen text informs the player that the protagonist's mother found her sleeping near the front door and carried her to bed. This initially suggests the entire sequence was merely a vivid night terror. However, the final narrative revelation—the discovery of physical scratch marks outside the door lock, indicating a forceful entry attempt—shatters this assumption.

This concluding twist effectively blurs the line between the subjective psychological nightmare and objective physical danger. By utilizing environmental evidence (the scratch marks) to validate the sleepwalking hallucination, the narrative successfully denies the player the comfort of a standard "it was all a dream" resolution, solidifying the overarching theme of domestic vulnerability.

Figure 4.1: (Concept)Environmental storytelling utilized in the epilogue to blur the boundaries between somnambulistic hallucination and physical reality.

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

Project Sentience successfully demonstrates how interactive media can leverage the psychological phenomenon of somnambulism to create a profound and unsettling horror experience. By shifting the focus away from traditional combat and resource management toward agency deprivation and cognitive overload, the project establishes a unique framework for narrative-driven psychological horror.

The methodology of utilizing spline-based locomotion to simulate the physical dissociation of sleepwalking proved highly effective. Stripping the player of their standard navigational agency and forcing them to act as a passive observer in a distorted, red-tinted environment successfully established a baseline of severe vulnerability. Furthermore, the deliberate architectural and visual manipulation of the domestic space reinforced the concept of the uncanny, transforming a universally safe environment—a new home—into a container of unpredictable threat.

The climax of the interactive sequence validated the project's core hypothesis regarding cognitive stress. The implementation of a dynamic, randomized User Interface (UI) for the escape sequence successfully disrupted player muscle memory. By requiring active, unpredictable keystrokes to progress along the return spline, the system accurately mirrored the motor-skill degradation and mental friction associated with real-world night terrors.

Finally, the narrative resolution—punctuated by the environmental storytelling of the scratch marks on the door—ensured the psychological tension extended beyond the gameplay loop. By blurring the boundaries between a subjective nightmare and objective reality, *Project Sentience* succeeds as a compact, highly impactful exploration of domestic vulnerability and subconscious dread.

5.2 Future Scope

While the current vertical slice effectively proves the core design concepts, there is significant potential to expand the mechanics and narrative scope of *Project Sentience*. Future iterations of this project will focus on deepening the systemic complexity and the atmospheric immersion through the following proposed avenues:

5.2.1 Procedural Environmental Distortion

Currently, the transition into the sleepwalking state relies heavily on post-processing volumes (color grading and chromatic aberration). Future development could implement procedural geometric distortion. As the player progresses further along the sleepwalking spline, the architecture of the house could subtly shift—hallways elongating, doorways vanishing, or furniture altering in scale—to dynamically increase spatial disorientation without requiring entirely new static level builds.

5.2.2 Advanced Spatial Audio and Psychoacoustics

Audio design is a critical component of psychological horror that can be vastly expanded. Future iterations should implement dynamic, spatialized audio systems that react to the player's progression. This includes integrating warped, muffled soundscapes during the passive sleepwalking phase to simulate being underwater or half-asleep, contrasted by sharp, hyper-realistic, binaural audio cues (such as a heartbeat or the antagonist's erratic footsteps) during the active pursuit phase.

5.2.3 Branching Outcomes and Non-Lethal Fail States

The current escape sequence relies on a pass/fail mechanical choke point. To enhance the narrative depth, a branching penalty system could be introduced. Instead of a traditional "Game Over" upon failing a randomized UI prompt, the failure could trigger alternative psychological scares. For example, stumbling might cause the player to wake up in a completely different, even more vulnerable area of the house, adding a layer of roguelike unpredictability to the parasomnia episodes.

5.2.4 Virtual Reality (VR) Integration

The core mechanics of *Project Sentience*—agency deprivation, spatial unfamiliarity, and restricted locomotion—make it a prime candidate for Virtual Reality (VR) adaptation. Porting the project to a VR environment would drastically amplify the claustrophobia and the physiological terror of the antagonist's proximity. In VR, the forced spline movement would generate an even more visceral sensation of sleep paralysis, pushing the boundaries of immersive psychological horror.

Figure 5.1: VR Concept(AI generated image)

Appendix A

CORE SYSTEM SCRIPTS (UNITY C#)

This appendix contains the primary C# scripts developed for *Project Sentience* to handle agency deprivation, dynamic UI inputs, and environmental triggers.

A.1 Spline Locomotion Override

The following script manages the transition from standard first-person control to the locked spline path during the sleepwalking sequence.

```
using UnityEngine;
using UnityEngine.Splines;

public class SplineEscape : MonoBehaviour
{
    [Header("Spline Setup")]
    public SplineContainer escapePath;    // Drag your Spline
       ↪ object here
    public GameObject pressSpacePromptUI;     // A UI Text object
       ↪  telling them to press Space

    [Header("Chase Mechanics")]
    public float momentumPerPress = 0.05f; // How much forward
       ↪ progress one 'E' press gives
    public float momentumDecay = 0.5f;     // How fast they slow
       ↪ down if they stop mashing
    public float maxSpeed = 0.3f;          // Cap the speed so
       ↪ they don't zoom off instantly

    private Rigidbody rb;
    private MonoBehaviour normalMovementScript; // Replace '
       ↪ MonoBehaviour' with your actual movement script class
       ↪ name if you want to disable it via code

    private bool isChasing = false;
    private float splineProgress = 0f;     // Goes from 0 (start)
       ↪  to 1 (end of spline)
    private float currentMomentum = 0f;

    void Start()
    {
```

```
        rb = GetComponent<Rigidbody>();
        if (pressSpacePromptUI != null) pressSpacePromptUI.
            ↪ SetActive(false);
    }

    // Called by the ScareTrigger to begin the sequence
    public void StartChaseSequence()
    {
        isChasing = true;
        splineProgress = 0f;
        currentMomentum = 0f;

        // Turn off physics so the spline controls movement
            ↪ entirely
        if (rb != null) rb.isKinematic = true;

        // Show the prompt!
        if (pressSpacePromptUI != null) pressSpacePromptUI.
            ↪ SetActive(true);

        // Snap to the very beginning of the spline
        UpdatePositionOnSpline(0f);
    }

    void Update()
    {
        if (!isChasing) return;

        // 1. Check for player mashing the Space key
        if (Input.GetKeyDown(KeyCode.Space))
        {
            currentMomentum += momentumPerPress;
        }

        // 2. Decay momentum over time so they are forced to keep
            ↪  mashing
        currentMomentum -= momentumDecay * Time.deltaTime;
        currentMomentum = Mathf.Clamp(currentMomentum, 0f,
            ↪ maxSpeed); // Prevent going backwards or too fast

        // 3. Move along the spline if they have momentum
        if (currentMomentum > 0f)
        {
            // Move our progress forward based on momentum
            splineProgress += currentMomentum * Time.deltaTime;
            splineProgress = Mathf.Clamp01(splineProgress); //
                ↪ Keep it between 0 and 1

            UpdatePositionOnSpline(splineProgress);

            // Check if we reached the end of the spline
```

```
            if (splineProgress >= 1f)
            {
                EndChase();
            }
        }
    }

    private void UpdatePositionOnSpline(float t)
    {
        // Get the exact position and rotation on the spline at
           ↪ percentage 't'
        Vector3 newPos = escapePath.EvaluatePosition(t);
        Vector3 forwardDir = escapePath.EvaluateTangent(t);
        Vector3 upDir = escapePath.EvaluateUpVector(t);

        transform.position = newPos;

        if (forwardDir != Vector3.zero)
        {
            // Rotate the player to face down the spline
            transform.rotation = Quaternion.LookRotation(
               ↪ forwardDir, upDir);
        }
    }

    private void EndChase()
    {
        isChasing = false;
        if (pressSpacePromptUI != null) pressSpacePromptUI.
           ↪ SetActive(false);
        if (rb != null) rb.isKinematic = false; // Turn physics
           ↪ back on

        Debug.Log("Player survived the chase!");
        // Add code here to transition to the next state, load a
           ↪ level, or open a door
    }
}
```

A.2 Dynamic Randomized QTE Interface

The following script handles the generation of randomized keystrokes during the escape sequence and validates player input to advance the character along the return spline.

```
using UnityEngine;
using TMPro;
using UnityEngine.Events;

public class RandomKeyPrompt : MonoBehaviour
{
    [Header("UI Elements")]
```

```
public GameObject promptCanvasPanel; // The UI panel
    ↪ containing the text
public TextMeshProUGUI promptText;   // The text showing the
    ↪ letter

[Header("Key Settings")]
// List of possible keys to randomize. You can add or remove
    ↪ keys in the Inspector.
public KeyCode[] possibleKeys = { KeyCode.W, KeyCode.A,
    ↪ KeyCode.S, KeyCode.D, KeyCode.E, KeyCode.Space };

[Header("Events")]
// What happens when they press the right key (e.g., play
    ↪ scream audio, man runs)
public UnityEvent onSuccess;

private KeyCode currentTargetKey;
private bool isWaitingForInput = false;

void Start()
{
    // Ensure the prompt is hidden when the game starts
    if (promptCanvasPanel != null)
    {
        promptCanvasPanel.SetActive(false);
    }
}

void Update()
{
    if (isWaitingForInput)
    {
        // Check if the player presses the currently assigned
            ↪  random key
        if (Input.GetKeyDown(currentTargetKey))
        {
            SuccessfulPress();
        }
        // Optional: You could add an "else if (Input.
            ↪ anyKeyDown)" here to penalize wrong presses.
    }
}

// Call this method via your window Trigger Box when the
    ↪ stare-down begins
public void StartRandomPrompt()
{
    // 1. Pick a random key from the array
    int randomIndex = Random.Range(0, possibleKeys.Length);
    currentTargetKey = possibleKeys[randomIndex];
```

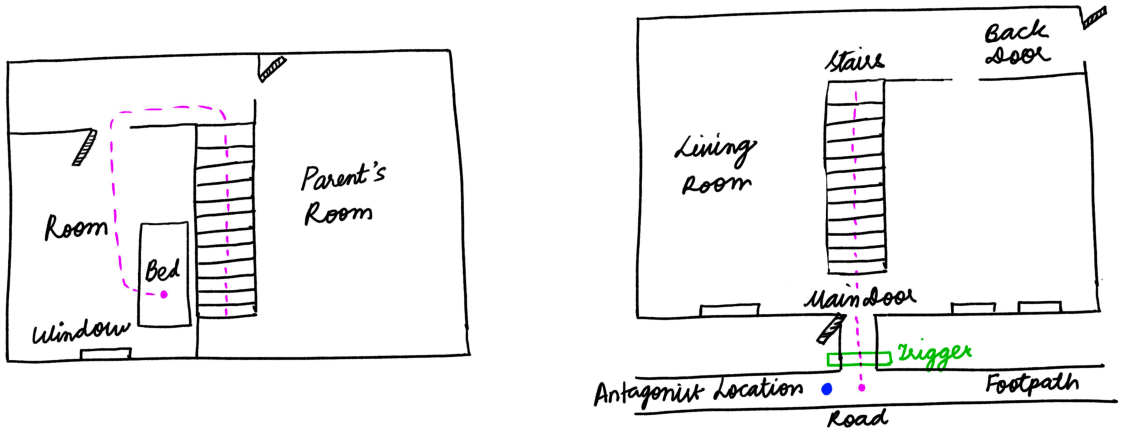
```
        // 2. Update the UI Text to clearly show the required key
        if (promptText != null)
        {
            // Formats the text to look like: PRESS [E] TO SCREAM
               ↪ !
            promptText.text = "PRESS [" + currentTargetKey.
               ↪ ToString() + "] TO SCREAM!";
        }

        // 3. Show the UI Panel
        if (promptCanvasPanel != null)
        {
            promptCanvasPanel.SetActive(true);
        }

        // 4. Start listening for input in the Update loop
        isWaitingForInput = true;
    }

    private void SuccessfulPress()
    {
        // 1. Stop listening for input
        isWaitingForInput = false;

        // 2. Hide the UI Panel instantly
        if (promptCanvasPanel != null)
        {
            promptCanvasPanel.SetActive(false);
        }

        // 3. Trigger the climax events!
        onSuccess.Invoke();
    }
}
```

Appendix B

LEVEL DESIGN AND ARCHITECTURAL BLUEPRINTS

This section contains the top-down architectural layouts utilized to construct the domestic space and plan the spatial triggers.

B.1 First Floor and Chase Trajectory



(a) First Floor (b) Ground Floor

Figure B.1: Floor architecture and Spline path.

Appendix C

USER PLAYTESTING QUESTIONNAIRE

To validate the psychological impact and the cognitive load of the randomized escape sequence, the following questionnaire was distributed to playtesters after their initial playthrough.

C.1 Post-Game Survey Questions

1. On a scale of 1 to 10, how disorienting was the visual transition from the normal house lighting to the red-tinted sleepwalking state?
2. Did you feel a sense of panic when your standard first-person controls were disabled? (Yes / No / Somewhat)
3. During the chase sequence, did the randomized on-screen key prompts cause you to make physical input mistakes? (Yes / No)
4. How did the environmental storytelling (the scratch marks on the door) affect your interpretation of the game's ending?