LIMINAL STUDIO

Technical Design Document

[Type the document subtitle]

Yassir Essabbahy

Saad ElRhandouri

Mohamed Bairouk

Driss Moussaoui

1. Game Overview

Title: Recording 02:17
Genre: Psychological Horror
Perspective: First-Person (3D)
Platform: PC
Engine: Unity
Development Type: Team Project

Recording is a short, narrative-focused psychological horror experience. The game places the player inside fragmented memories of the protagonist, presented as physical rooms connected by a hallway. Each room represents an emotional or traumatic memory tied to the protagonist's father. The game avoids combat entirely and instead relies on atmosphere, sound design, subtle interaction, and visual distortion to create tension and unease.

The primary goal of the game is narrative discovery. The player must observe, interact, and reflect in order to understand what happened in the past and why these memories are resurfacing. The final scene resolves the mystery through a PC that contains notes and information left behind by the protagonist's father.

2. Vision & Design Pillars

2.1 Psychological Horror

Fear is created through discomfort rather than danger. The player is never attacked. Instead, stress is conveyed through audio distortion, ringing sounds, flickering lights, camera shake, and unsettling silence.

2.2 Environmental Storytelling

There are no traditional cutscenes. Story information is embedded directly into the environment through notes, images, computers, and room layout.

2.3 Player Interpretation

The narrative is intentionally indirect. The player pieces together meaning from fragments, encouraging personal interpretation.

2.4 Minimal Mechanics, Maximum Atmosphere

Gameplay mechanics are intentionally simple to avoid distraction from the atmosphere and story.

3. Core Game Loop (Expanded)

1. Player spawns in a hallway
2. Player moves forward and enters a memory room
3. Player observes environmental details
4. Player interacts with objects using `E`
5. Interaction triggers narrative elements and psychological effects
6. Stress and audio effects escalate
7. Door unlocks or next area becomes available

8. Player returns to hallway and continues forward
9. Loop repeats until final room
10. Player interacts with final PC
11. Narrative resolution occurs
12. Game ends

4. Player Systems

4.1 Player Movement System

Purpose: Enables grounded, realistic first-person movement.

Detailed Behavior:

- Movement is intentionally slow and controlled
- No sprinting or jumping to maintain tension
- Player always feels vulnerable and exposed

Scripts:

- `PlayerMovement`
  Handles translation of the player character. Reads input values, applies movement speed, checks for collisions, and ensures smooth movement.
- `PlayerControls`
  Acts as a central input handler. Collects input for movement, interaction, and UI control, then forwards it to the appropriate systems.

4.2 Camera & Look System

Purpose: Provides immersion and reinforces psychological effects.

Scripts:

- **`MouseLook`**
  Controls camera rotation based on mouse movement. Applies sensitivity scaling and clamps vertical rotation to prevent unnatural angles.
- **`CameraStressShake`**
  Adds procedural camera shake during stress moments. Intensity scales based on stress level.

5. Interaction System (Highly Detailed)

Purpose: Primary gameplay mechanic and narrative driver.

System Flow:

1. Camera casts a ray forward every frame
2. Ray checks for objects with interaction scripts
3. Interaction range is validated
4. UI prompt may appear
5. Player presses `E`
6. Object-specific logic executes
7. Optional: player movement disabled
8. Optional: stress/audio effects triggered

Script Responsibilities:

- **`PickUp`**
  Enables physical interaction with small objects. When

activated, the object may move toward the camera, rotate slowly, and be examined. Often used for notes or symbolic items.

- `PictureFrameInteractable`
  Handles framed memories. Opens an image viewer UI, displays memory-related imagery, and may trigger audio or stress escalation.
- `StoryItem`
  Flags an object as narratively important. Tracks whether the player has interacted with it. Used to prevent repeated triggers and to unlock progression logic.
- `InteractablePC`
  Specialized interaction logic for computers. Locks player movement, activates PC boot sequence, opens desktop UI, and allows access to files and notes.

6. Environment & Level Progression System

Purpose: Controls pacing, structure, and narrative order.

Design Philosophy:

- Linear progression ensures narrative clarity
- Backtracking is minimal
- Each room has a clear emotional purpose

Scripts:

- `Door`
  Controls basic door behavior including opening animations, sound playback, and interaction logic.

- `Door2`
  Extended door logic. Supports locked states, delayed opening, or story-based conditions.
- `DoorOpen`
  Automatically opens doors when the player enters a trigger zone. Used to control pacing.
- `InstanceTrigger`
  General-purpose trigger script. Activates events such as audio cues, lighting changes, stress effects, or progression flags.

7. Narrative Delivery System (Expanded)

Purpose: Deliver story information without exposition.

Narrative Techniques:

- Textual notes
- Old PC files
- Environmental symbolism
- Audio cues

Scripts:

- `NotesWindowController`
  Displays text-based notes. Handles UI scaling, scrolling, and closing behavior.
- `DesktopUIController`
  Controls the in-game computer interface. Handles file navigation, window management, and story triggers tied to files.

- `WindowEffects`
  Applies glitching, flickering, and distortion to UI windows.
- `WindowsErrorShutdown`
  Simulates system crashes and forced shutdowns to reinforce narrative tension.

8. Audio System (Expanded)

Purpose: Reinforces emotional and psychological states.

Audio Design Goals:

- Make silence uncomfortable
- Use sound sparingly but effectively

Scripts:

- `SoundTrigger`
  Plays specific sounds when entering trigger zones.
- `SoundAndRoomTrigger`
  Ensures sounds only play in correct rooms or contexts.
- `FootstepAudio`
  Plays footstep sounds based on movement state.
- `EarRingingTrigger`
  Triggers ringing audio to simulate psychological overload or trauma.

9. Stress & Psychological Effects System (Very Detailed)

Purpose: Externalizes the protagonist’s mental state.

Effects Include:

- Screen distortion
- Blur and vignette
- Camera shake
- Flickering lights

Scripts:

- **`StressEffectController`**
  Central controller managing all stress-related visual effects. Adjusts intensity dynamically.
- **`StressStartTrigger`**
  Activates stress effects when entering specific zones or interacting with key objects.
- **`StressStopTrigger`**
  Reduces or disables stress effects once conditions are met.
- **`LightFlicker`**
  Randomizes light intensity and timing to create unease.
- **`LightTrigger`**
  Enables or disables lighting effects based on player location.

10. UI & Menu System

Purpose: Controls navigation and player feedback.

Scripts:

- **`MainMenuController`**
  Handles main menu navigation and scene loading.
- **`PauseMenu`**
  Freezes gameplay and disables player input while paused.

- `CreditsTrapController`
  Displays credits in a way consistent with the horror tone.
- `UISounds`
  Plays audio feedback for UI interactions.
- `FPSCounter`
  Displays performance metrics for debugging.

11. Final Scene & PC System

Purpose: Deliver narrative resolution.

Flow:

1. Player reaches final room
2. PC interaction begins
3. Fake boot sequence plays
4. Desktop opens
5. Player reads notes about father
6. Stress peaks
7. Game ends

Scripts:

- `BootManager`
  Controls the sequence of PC boot screens.
- `FakeBootController`
  Simulates operating system startup for immersion.
- `DesktopUIController`
  Displays final narrative content.

12. Technical Constraints & Optimization

- No save/load system
- Linear level structure
- Trigger-based logic for performance
- Designed for low-end PCs

13. Risks & Challenges

- Communicating story clearly in short runtime
- Balancing subtle horror
- Ensuring player understanding without exposition

14. Completion Criteria

- All memory rooms visited
- Final PC interaction completed
- Narrative fully revealed
- Game exits cleanly