SHADOW FRAMES - SOUND BIBLE

GROUP 22 | 2025

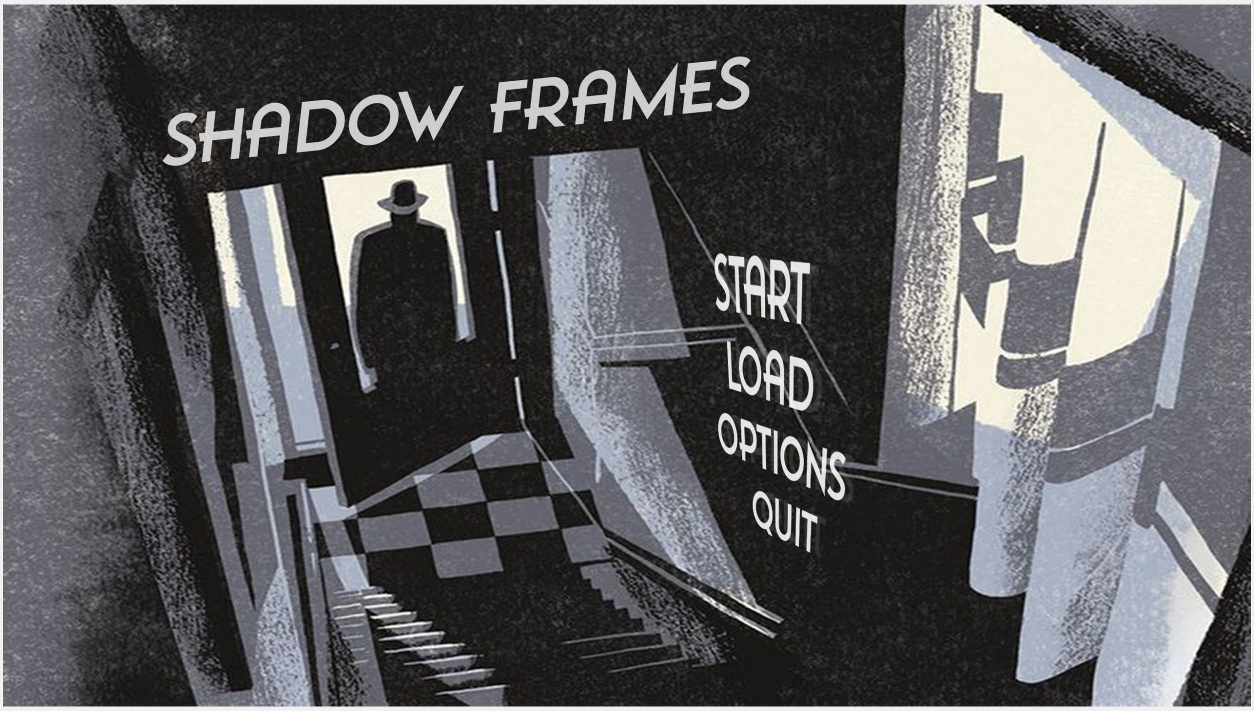
Team:

- Art: Daan Van Schuylenbergh, Roeland Van Sichem De Combe
- Dev: Ine Hocedez, Eitvile Petruseviciute, Hades Speranskaya
- Sound: Igor Szuniewicz

Contents

1. WHAT WE'RE MAKING

[Game menu screenshot/concept art]



Shadow Frames is a black-and-white psychological horror game set in a haunted 1960s Irish pub. You play as an investigative reporter armed with a vintage Polaroid camera. Your job: capture supernatural anomalies before they stack up and trigger something far worse.

The Experience: Tense exploration where sound is your primary guide. Objects behave strangely, shadows move on their own, and something is watching. The camera is your only defense, but film is limited.

Why Audio Matters: In a monochrome world with minimal UI, sound does the heavy lifting. It tells you when anomalies appear, where enemies are, and whether you're safe. Audio isn't decoration—it's survival information.

⚠ Recommended Experience: Shadow Frames is designed ***primarily for headphone use***. The game's spatial audio system and psychological horror elements rely heavily on precise 3D positioning and intimate soundscapes that are best experienced through headphones. While the game is playable with speakers, **headphones provide the full intended experience and maximize tension**.

2. CORE AUDIO VISION

The Three Pillars

1.Constant,Dread,Never,Relief

No jump scares. Instead, we maintain perpetual tension through low drones, distant sounds, and ambiguity. Players never fully relax.

2. 1960s Analog Isolation

Every sound must feel period-appropriate. Polaroid clicks, vinyl crackles, tube radio static. No digital sounds. The analog tech emphasizes your helplessness—this camera is all you have.

3. Nothing is Certain

Is that crackle the record player or fire? Is that breathing wind or the entity? Familiar sounds distorted just enough to keep you guessing.

[Conceptual Living Room]

3. HOW AUDIO SUPPORTS GAMEPLAY

[Conceptual Living Room]

Character Identity & Player Perspective

Who You Are: The player is an investigative reporter drawn to the pub by consistent reports of supernatural phenomena. This identity influences audio design in several ways:

- **Breathing:** Professional composure with underlying anxiety—not panicked gasping, but controlled tension
- **Footsteps:** Deliberate, investigative pace—lighter than a pub owner who knows every creaky floorboard
- **Camera Handling:** Practiced but cautious—familiar with photography equipment, but not with supernatural subjects
- **Reactions:** Measured surprise rather than shock—someone documenting evidence, not someone whose home is haunted

Anomaly Detection

[Anomaly detection visualization]

Anomalies spawn every 30-50 seconds—objects move, distort, or vanish. Audio is your primary alert system.

The Three-Part Audio Cue:

Stage 1: The Alert (0.5s)
A sharp static burst combined with a metallic clang cuts through the ambient soundscape. This confirms something just changed in the pub—but not *where*.

Stage 2: The Hint (1-2s)
A directional, room-specific sound follows immediately: a chair scraping wood, glass clinking against a bottle, or floorboards creaking. This tells you which direction to investigate.

Stage 3: The Presence (sustained)
As you approach, a continuous creepy ambient loop begins—sustained creaking, unnatural humming, or distorted environmental sounds. This persists until you photograph the anomaly.

Example Scenario:
You're behind the bar when you hear the alert stinger [Bonus]. Half a second later, distant wood-on-wood scraping echoes from the back room. You move cautiously toward it, and the scraping intensifies into a sustained, rhythmic creak. Raising your camera, you see it: a chair rotating slowly on its own. One photograph, and the sound stops—anomaly dispelled.

Anomaly Categories & Audio Design

Each anomaly type requires distinct audio approaches. The specific quantities listed below are conceptual estimates and may change during production based on gameplay needs and time constraints.

Object Anomalies

Anomaly Type	Audio Requirements	Est. Quantity	Priority
Chair Rotation	Wood creaking, floor scraping (3 variations)	~9 sounds	P0
Glass Levitation	Subtle glass resonance, air distortion	~6 sounds	P1
Object Disappearance	Reversed "pop", reality tear, spatial void	~8 sounds	P0
Texture Changes	Material transformation (wood→metal, clean→decay)	~12 sounds	P1

Environmental Anomalies

Anomaly Type	Audio Requirements	Est. Quantity	Priority
Room Distortion	Space warping, perspective shift drones	~5 layers	P0
Radio Stutter	Frequency shifts, 1960s broadcast fragments	~10 variations	P1
Jukebox Malfunction	Record skipping, speed changes, backwards playback	~6 events	P1
Clock Speed Change	Mechanical ticking acceleration/deceleration	~4 loops	P1

Shadow Anomalies

Anomaly Type	Audio Requirements	Est. Quantity	Priority
Shadow Figures	Whispered presence, spatial displacement	~8 variations	P0
Shadow Distortion	Reality bend, optical warping sounds	~6 sounds	P0
Movement in Darkness	Impossible footsteps, fabric rustling	~10 sounds	P0

Environment Sounds - The Unsettled Pub

[Anomaly detection visualization]

The 1960s Irish pub has distinct sonic zones. Each area must sound authentic and support anomaly detection through contrast. When something changes, players should immediately notice because they've learned what "normal" sounds like.

When Anomalies Occur:

- Music tempo changes unnaturally
- Backwards audio fragments bleed through
- Microphone feedback (not present normally)
- Stage lights buzzing (electrical distortion)

Tables & Seating Area

Tables & Seating Area

Normal State Ambience:

- Chair legs scraping on floor (occasional, unexplained)
- A lone pint glass shifting or sliding slightly on a wooden surface
- The faint sound of ash settling in a long-abandoned ashtray
- The soft rustle of old, decaying paper

When Anomalies Occur:

- Chairs move without physical contact
- Objects fall with delayed impact sounds
- Spatial audio displacement (sound source doesn't match visual)

Overall Pub Atmosphere

Base Ambience Layers:

- **Layer 1 - Building Tone:** Old wood settling, subtle wind through gaps, distant and sparse street traffic (1960s vehicles).
- **Layer 2 - Lingering Presence:** Unexplained solitary footsteps on creaky floors, faint whispers carried on drafts, the echo of a long-faded memory.
- **Layer 3 - Period Details:** Rotary phone ringing (distant), analog clock ticking, tube radio static.
- **Layer 4 - Tension Drones:** Barely perceptible low-frequency rumble (40-60Hz), increases with anomaly count.

1960s Period-Specific Sounds:

- **Glass types:** Heavier, thicker than modern glasses—distinct weight when set down
- **Furniture:** Solid wood, metal joints (not particle board)—deeper resonance
- **Smoking culture:** Ashtrays, zippo lighters, cigarette pack cellophane
- **Tech era:** Tube radio warmth, mechanical cash register, rotary phone dial
- **Vehicles outside:** Older engine sounds, less traffic density

Camera System

[Camera illustration]

Your vintage Polaroid is both your primary tool and your vulnerability. Every interaction has distinct audio feedback that makes the camera feel like a real, physical object.

Raising the Camera (Observation Mode)

The moment you lift the viewfinder to your eye, the world changes:

- Environmental audio becomes **muffled and filtered** (around 3kHz cutoff), creating a tunnel-hearing effect
- Your **breathing** becomes prominent—each inhale and exhale fills your ears
- **Leather and plastic handling sounds** creak and shift as you grip the camera
- Distant threats become **harder to hear**, trading awareness for focus

Taking the Shot

Pressing the shutter triggers a satisfying mechanical sequence [BONUS]:

- **Shutter click**: The authentic Polaroid mechanism fires
- **Flash pop**: A bright burst accompanied by the high-pitched **capacitor whine** as it recharges
- **Film advance**: A motorized **whirr** as the photo ejects

If you captured something supernatural, you hear a brief **reversed whisper**—an "energy release" confirming you got it. If you photographed nothing, the mechanical sounds play out the same, but they feel *empty*, wasted.

Running Out of Film

You press the shutter and hear only a **hollow, distinct click**—no full mechanism, no flash. The camera is dead weight now. Time to reload.

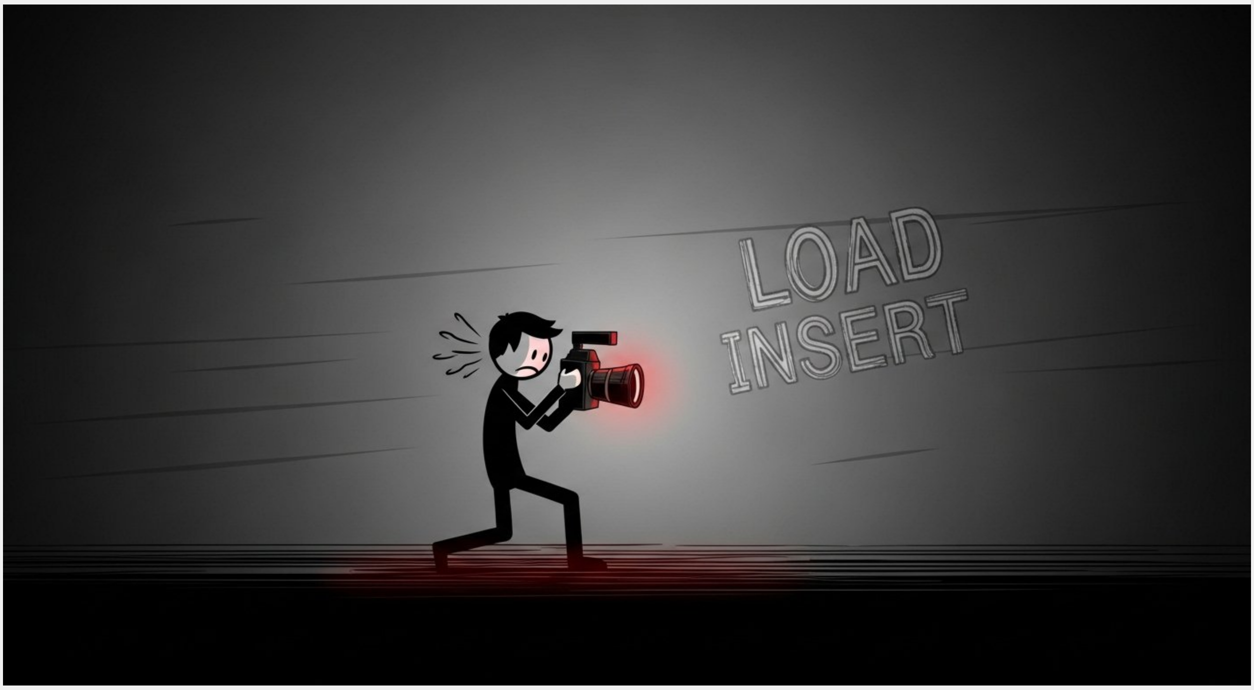
The Reload (5 Seconds of Vulnerability)

You can't move. You can't look around. The screen locks into a reload animation while audio plays out:

1. Camera **lowering** with a mechanical clunk
2. Empty cartridge **ejecting** with a plastic snap
3. New cartridge **sliding** into place
4. System **ready beep**—a small, analog confirmation tone

During these five seconds, **all ambient threats remain audible**. Footsteps. Breathing. Creaking. You're blind, immobile, and exposed—maximum tension by design.

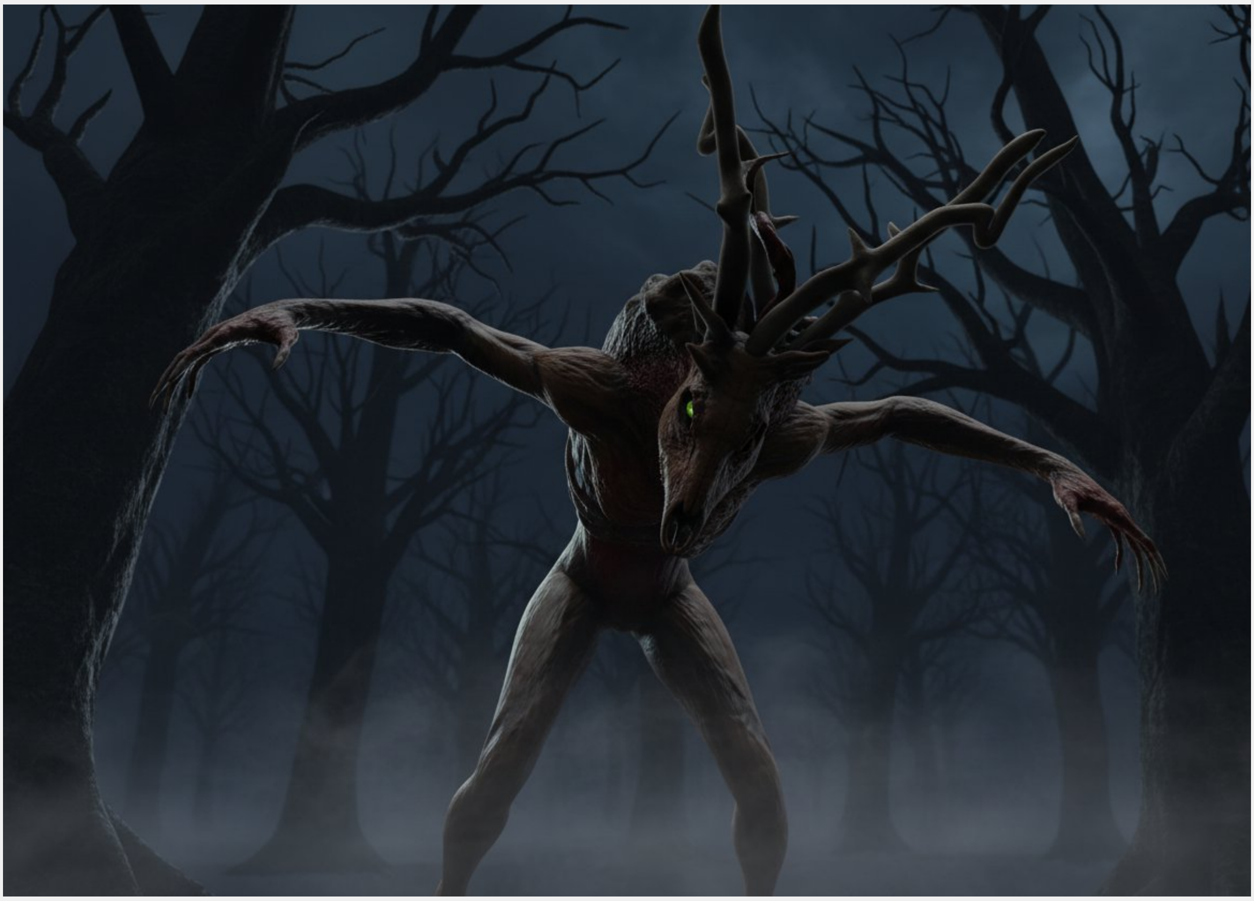
[Running out of film]



The Enemy

Design Note: The enemy system is currently under review. This section documents the intended audio design should the enemy remain in the final game. All audio assets for the enemy are considered P0 priority if the feature is retained, but may be deprioritized or removed if the design changes.

[Enemy character design]



Overview: Hostile entity that spawns when too many anomalies go uncaptured. Tall, distorted humanoid with elongated limbs and unnatural movement. Its ambiguous nature (demon/ghost/other) is reinforced through audio that blends organic and supernatural qualities.

[MAIN AUDIO REFERENCE]

Core Audio Elements:

Breathing:

- Heavy, wet quality with irregular rhythm
- Primary indicator of enemy presence
- Gets louder as enemy gets closer
- Subtle vocalization undertones (almost speech-like but never clear)

Example

Footsteps:

- Heavy impacts with slight drag between steps
- Material-specific: wood creaks, tile echoes, carpet muffled
- Volume indicates distance
- Uneven gait, occasional joint cracks

Example

Roars (Position Updates):

- Periodic roars while hunting
- Helps player track enemy location between rooms
- Multi-layered vocalization (will be recorded and processed by team member Roeland)
- Reference: Fiddlesticks (League of Legends) for unsettling vocal quality

Example

Distance Indicators: All enemy sounds use distance attenuation - the closer it is, the louder and more detailed the audio becomes. Creates natural tension as threat approaches.

Behavioral States:

Lurking (Hunting):

- All audio elements active as it moves
- Breathing and footsteps indicate position and distance

Seen (Player Looking At It):

- **Entity freezes in place** (like Little Nightmares 2 mannequins)
- **No footsteps** (not moving)
- Breathing continues (slightly quieter)
- When player looks away: footsteps resume as it moves again
- Mechanic creates tension: safety vs. awareness

Chase (Active Pursuit):

- Footsteps accelerate
- Breathing intensifies
- Close-range audio at maximum threat

Design Intent: Audio is primary tracking tool. Player relies on breathing and footsteps to locate enemy, with periodic roars providing position updates. Spatial audio implementation is critical for this feature to work.

First-Time Player Experience - Audio Tutorial

The game teaches mechanics through audio rather than text prompts. The first 3-minute safe period establishes what "normal" sounds like, so when anomalies start, they feel genuinely wrong.

Tutorial Sequence

Minute 0-1: Arrival & Orientation

- Player spawns at the bar—hears full normal ambience
- Note on counter (visual + subtle paper rustle when approached)
- Camera on counter has a faint "ready" hum (indicates usability)
- No dialogue—only environmental storytelling through sound

Minute 1-3: Learning Movement & Observation

- Player explores pub, learning layout through sound
- Jukebox plays period-appropriate music (sets 1960s tone)
- Footsteps on different surfaces teach spatial awareness
- Optional: Player can practice raising camera (no anomalies yet, but hears observation mode filtering)

Minute 3: First Anomaly (Scripted Tutorial Event)

1. **Alert Stinger** plays—sudden but not overwhelming
2. **Directional Cue:** Chair scraping from specific area
3. Player investigates, raises camera
4. **Visual + Audio Confirmation:** Chair slowly rotating + sustained creak loop
5. Player takes photo—**reversed whisper** confirms success
6. Anomaly sound stops—silence briefly, then normal ambience returns
7. Photo automatically added to pin board (subtle paper slide + pin sound)

Post-Tutorial: Player now understands the core loop through audio-driven mechanics. No explicit instructions needed—the sound design is the tutorial.

UI & INTERFACE AUDIO

Philosophy: The Camera is Your Language

[Conceptual User Interfacez]

SHADOW FRAMES
PLAY
CONTINUEE
SETTINGS
EXIT

Your Polaroid isn't just for anomalies—it *is* the interface. Every menu interaction uses that satisfying mechanical click. No digital beeps. No modern UI. Just the tool that keeps you alive.

Main Menu

Atmosphere:

- Dark, mysterious loop (FNAF Sister Location vibe)
- Vinyl crackle texture throughout

The Polaroid Navigation System:

- **Hover**: Soft mechanical pre-click (half-second before shutter fires)
- **Select**: Full Polaroid shutter + film advance whirr
- **Back**: Reversed shutter—it *un-clicks*

Why It Works: Even choosing "New Game" feels like capturing a moment. Period-appropriate, satisfying, thematically perfect.

Reference: Resident Evil 2 Remake menu—tactile, weighty, no wasted sounds

Settings

Sliders:

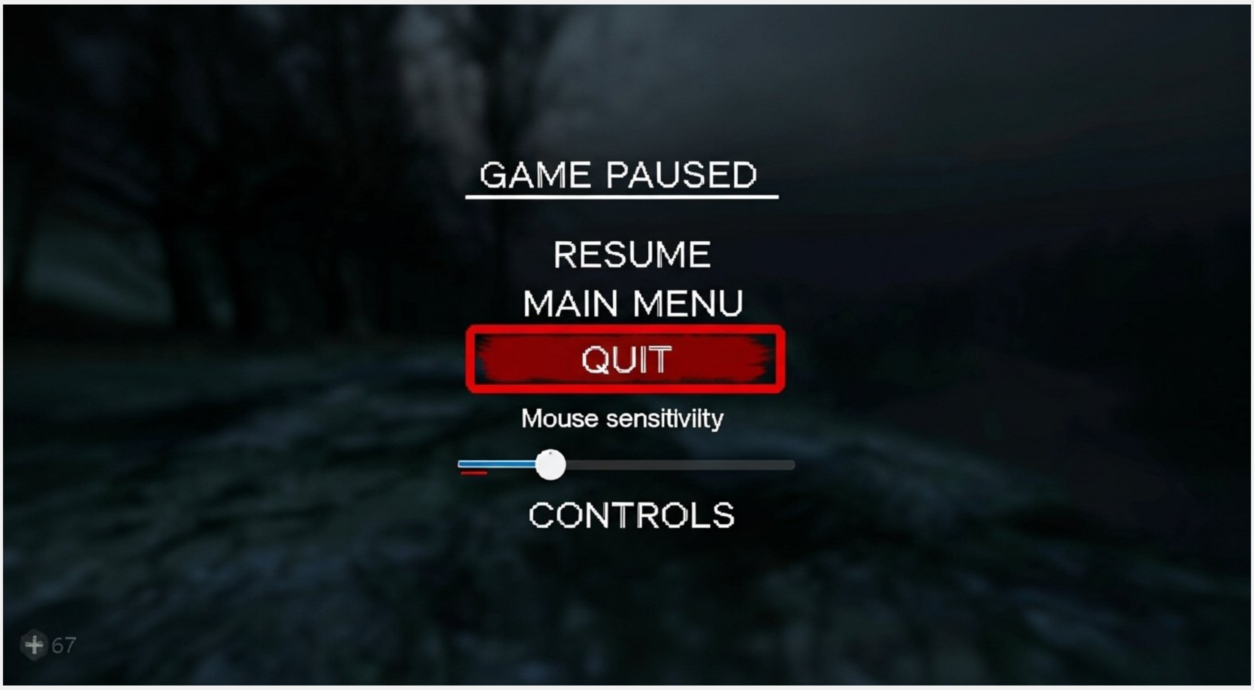
- Continuous soft ratcheting (vintage volume knob)
- Audio sliders play test tone at current level

Toggles:

- ON: Light switch *thunk* (upward)
- OFF: Different sound (downward, looser)—asymmetry feels real

Pause Menu

[Pause Menu]



The pub doesn't pause with you.

Game audio continues but gets an underwater filter (-6dB, muffled highs). Pause transition: brief tape stop effect. Navigation uses same Polaroid clicks, just quieter.

Even paused, threats remain audible. You're holding your breath.

Reference: Dying Light The Beast Inventory sound effect

HUD (Film Counter)

Bottom right. Shows Polaroid cartridge window.

Silent by default. Audio only when critical:

- **3 photos left**: Quiet *tick* after each shot
- **Empty**: The hollow click you know from Camera System

Design: Silence = safe. Sound = pay attention.

Tutorial Prompts

Appears:

- Subtle analog *ding* (camera light meter)
- Brief film projector click

Dismissed:

- Paper slide + quiet snap (closing a notebook)

Reference: Firewatch radio—diegetic, like someone left notes

Game Over

5-second death sequence:

1. Sharp impact
2. Heartbeat races (1 second)
3. Flatline drone
4. Reverse whoosh to black
5. Main menu music fades in—you're outside again

Reference: Inside death—brutal, instant

Endings

[The End]

Good Ending:

Good Ending:

- Pub ambience normalizes—distortion fades, the oppressive atmosphere lifts, leaving only the natural sounds of the old building settling.
- The oppressive silence of the pub is replaced by the neutral, peaceful quiet of the late night.
- Distant morning sounds—birds, traffic, daylight
- Final Polaroid click in the distance

Bad Ending:

- All sound collapses inward (heavy compression)
- Single low drone consumes everything
- Abrupt silence
- Broken version of menu music begins

The World as UI

Audio tells you what text doesn't:

First Anomaly: Alert stinger + radio crackles somewhere in pub

Anomalies Stacking: Jukebox skips, clock ticking goes wrong

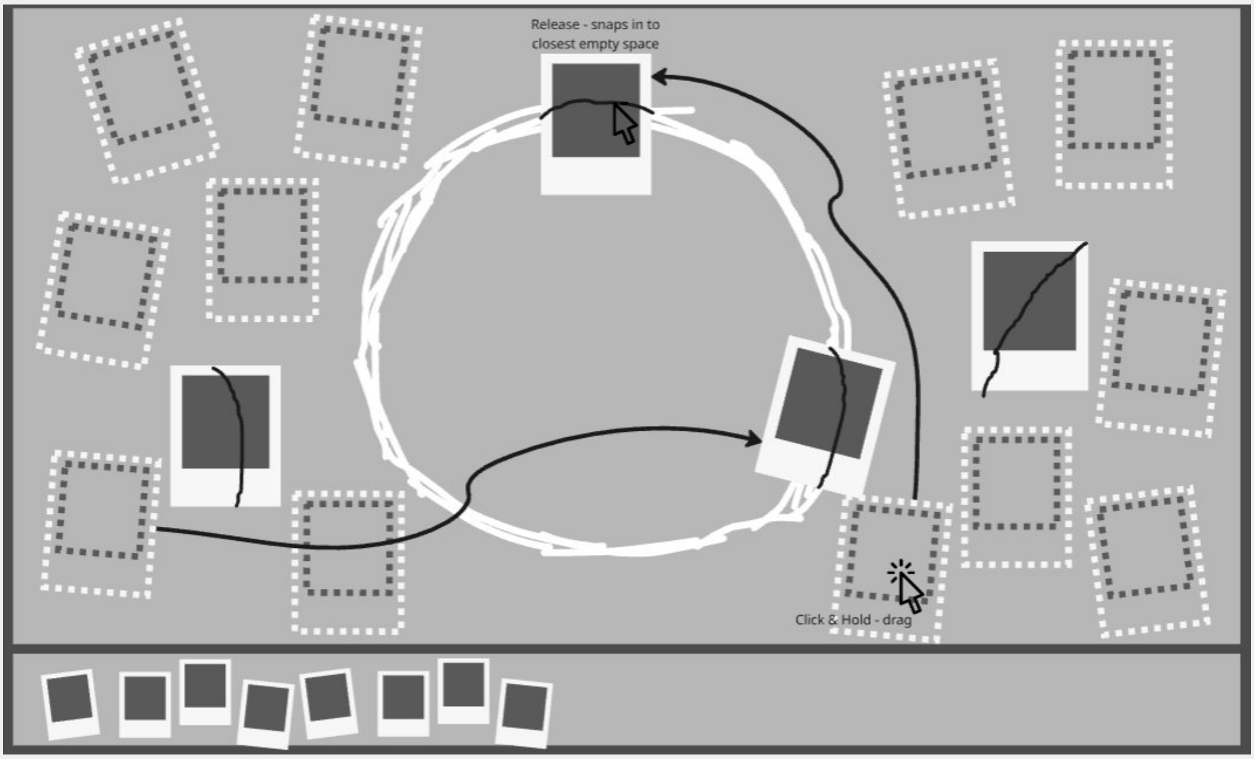
Enemy Spawns (if retained): All music stops (tape rip) → 2 seconds silence → breathing begins

No UI needed. The pub screams at you.

Core Principle: Your camera speaks in menus. The pub warns through broken sound. No comforting reload screens—just the reality that you failed, and it's waiting.

Pin Board Puzzle

[Pin Board Puzzle]



The Mechanic: Interactive board at starting location. Player arranges captured photos in correct order using visual clues to reveal hidden pattern (required for good ending).

Audio Feedback: Opening/Closing:

- Zoom transition whoosh
- Room ambience ducks -6dB during use
- Subtle pad layer for focus

Example

Photo Interactions:

- **Auto-add (after capture):** Paper slide + pin insertion + settling
- **Lifting:** Paper separation crackle
- **Dragging:** Continuous soft rustle
- **Correct placement:** Snap + secure pin click + ascending harmonic (2-3 notes)
- **Incorrect placement:** Loose pin sound, discordant note or silence
- **Section completion:** Additional chord layer

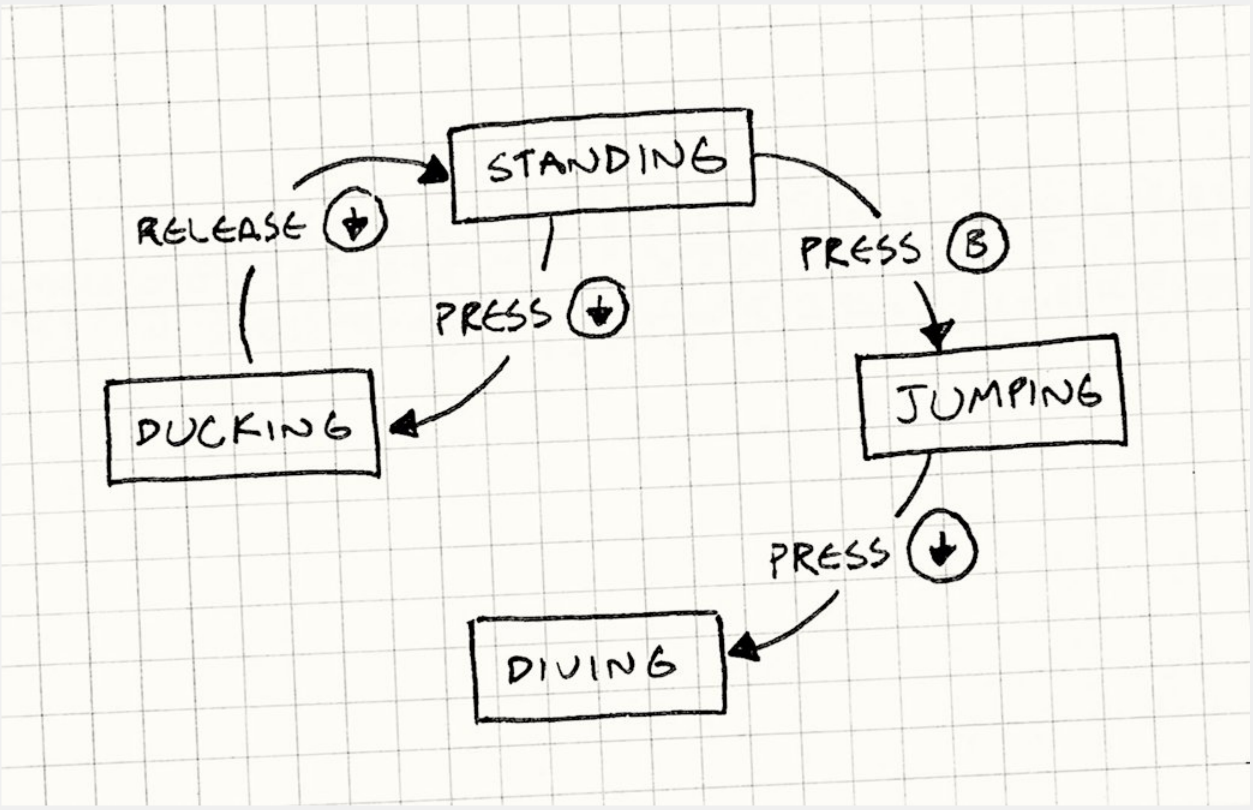
Puzzle Completion:

- Sustained harmonic swell (4-5 seconds)
- High-pass filter lifts on ambience
- Final chord: resolving but ominous (success with foreboding undertone)

Design Goal: Brief respite through musical feedback, but maintains tension - even success feels ominous.

4. DYNAMIC AUDIO SYSTEM

[Example game states image]



Game States

The game uses three distinct audio states that shift dynamically based on player progress and threat level:

SafePeriod (first 3 minutes):

- Calm pub ambience at full volume
- No anomalies spawn yet
- Player learns what "normal" sounds like
- Establishes baseline for later contrast

Normal (main gameplay):

- Anomalies active (spawning every 30-50 seconds)
- Low tension drones fade in gradually
- Enemy can spawn if anomaly count reaches threshold (if feature retained)
- Player balances observation and resource management

Escalation (late game/high threat):

- Anomalies spawn more frequently
- Tension drones increase in volume and intensity
- Environmental sounds become more distorted
- Audio feedback heightens player stress

Smart Mixing (RTPC System)

Wwise Real-Time Parameter Controls (RTPCs) automatically adjust the audio mix based on gameplay variables. This means the sound designer sets up the rules once, and the game handles the rest—no manual scripting needed for these dynamic changes.

Here's how it works in practice:

- **Enemy Distance:** As the enemy moves closer, its breathing and footsteps get louder. When it moves away, they fade out naturally. The game tracks the distance between player and enemy, and Wwise adjusts volume accordingly.
- **Anomaly Severity:** Each uncaptured anomaly increases a global "severity" value. As this number goes up, tension drones get louder, and normal ambience gets more distorted. Capture an anomaly, and the severity drops back down.
- **Observation Mode:** When the player raises the camera, a simple game state change tells Wwise to apply the audio filter. When they lower it, the filter fades out. No complex code required.
- **Film Count:** When the player drops to 3 or fewer photos remaining, a subtle high-frequency tension layer fades in. It's barely noticeable consciously, but it adds subconscious pressure.

The beauty of this system is that once it's set up, gameplay changes automatically drive the audio. Programmers don't need to manually script "play breathing louder" every time the enemy moves—Wwise handles it based on real-time data from the game.

Spatial Audio Implementation Details

Spatial audio is what makes Shadow Frames' audio-driven gameplay possible. Players need to accurately locate sounds in 3D space to find anomalies and track threats. Here's how we make that work:

3D Positioning System

Wwise Integration:

- All mono sources use Wwise 3D positioning to place sounds in space
- HRTF (Head-Related Transfer Function) creates binaural audio—sounds actually feel like they're coming from specific directions in headphones
- No surround mixing—stereo headphones are our primary target (most players will use them)

Distance Attenuation Curves:

Different sound types fade out at different rates. Here's what that means for gameplay:

Source Type	Max Distance	Curve Type	Why This Matters
Anomaly Stingers	30 meters	Logarithmic	Must be audible anywhere in pub—player can't miss these
Anomaly Presence Loops	15 meters	Linear	Gets louder as you approach—guides player to source
Enemy Breathing	25 meters	Custom (steep falloff)	Close = terrifying, distant = foreboding
Enemy Footsteps	20 meters	Linear	Accurate tracking needed for survival
Environment Details	10 meters	Inverse	Natural falloff, not critical to gameplay
Player Footsteps	N/A (always heard)	N/A	First-person reference, never fades

Occlusion & Obstruction

Sounds change when walls or objects get in the way. This adds realism and helps players understand the pub's layout through audio alone.

Wall Occlusion:

- When a sound source is behind solid geometry (walls, large furniture), we apply a low-pass filter
- Filter cutoff: 1200 Hz (reduces clarity but maintains directionality—you still know where it's coming from)
- Volume reduction: -12 dB (sounds muffled but not silent)
- **Exception:** Enemy breathing always stays slightly audible even through walls (horror design—uncertainty is key)

Partial Obstruction:

- Small objects (chairs, tables) provide minor attenuation
- Volume reduction: -3 to -6 dB (subtle, not total blockage)
- No filtering—maintains audio fidelity so players can still detect anomalies accurately

Reverb Zones

Each area of the pub has its own acoustic character. This helps players know where they are even without looking:

Area	**Reverb Type**	**Decay Time**	**What It Sounds Like**
Bar Area	Small Room	0.8s	Reflective surfaces (glass, mirror), tight space—sounds bounce back quickly
Main Floor (Tables)	Medium Room	1.2s	Open space, wood/fabric absorption—natural pub ambience
Stage Area	Medium-Large	1.5s	Slightly elevated, curtains diffuse sound—feels more spacious
Bathroom (if included)	Bathroom Preset	2.0s	Tile surfaces, harsh reflections—unmistakably different

Dynamic Reverb Changes:

- When anomaly count increases, reverb decay time extends by +0.3s (reality starting to break down)
- Observation Mode reduces reverb send by -8 dB (tunnel focus effect—you're concentrating)

Spatial Audio Priority

When lots of sounds are playing at once, the system needs to decide which ones are most important. Here's the priority order:

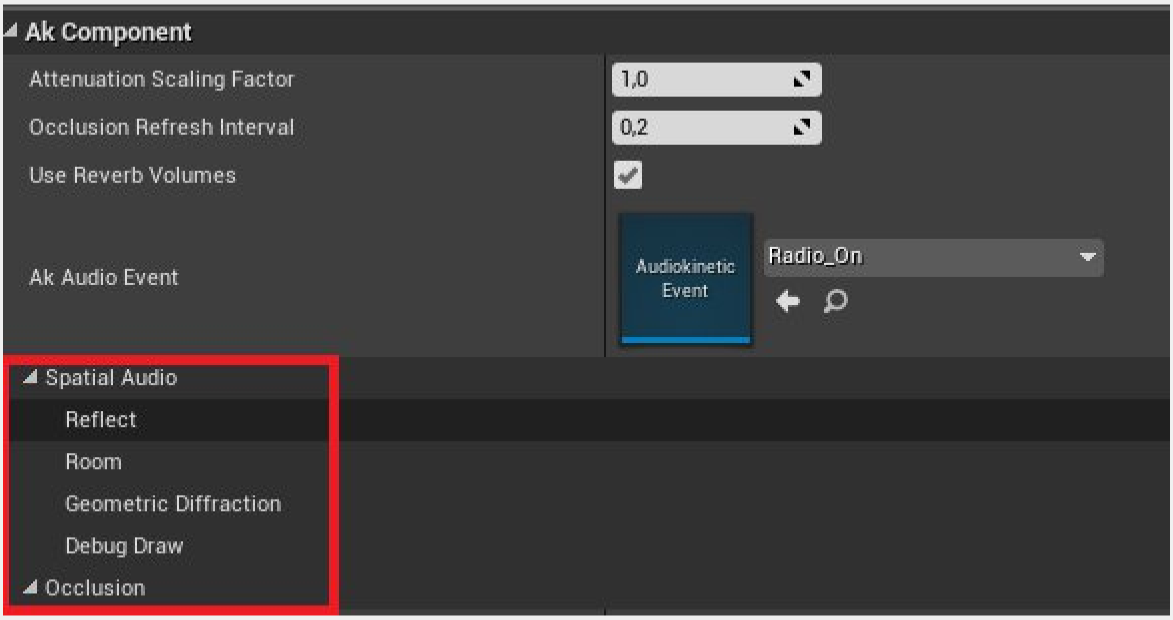
Priority Level	**Source Type**	**Never Culled**
1 (Highest)	Enemy breathing, footsteps, roars	✓ Always plays
2	Anomaly stingers, presence loops	✓ Always plays
3	Player footsteps, camera sounds	✓ Always plays
4	Object movement (chairs, glass)	— Can be culled if needed
5 (Lowest)	Environmental detail (distant traffic, clock)	— Can be culled if needed

Voice Limiting: Maximum 32 simultaneous 3D sources. When we hit that limit, lower priority sounds furthest from the player get culled. Gameplay-critical sounds always stay audible.

Moving Objects & Wwise Integration

Objects that move in the pub—chairs sliding, doors creaking, bottles rattling—need their sounds to follow them perfectly. Here's how we handle that:

[Wwise integration Menu]



Ak Component

Attenuation Scaling Factor: 1,0

Occlusion Refresh Interval: 0,2

Use Reverb Volumes: ☑

Ak Audio Event: Radio_On

Implementation Workflow:

- Each moving object in Unreal Engine gets a **Wwise Audio Component (AkComponent)** attached to it. This component automatically follows the object wherever it goes.
- When the object starts moving, we trigger a **Wwise Event** (like "Play_ChairScrape"). When it stops, we send a Stop Event. The sound fades out naturally.
- The component updates the sound's 3D position every frame automatically—no manual work needed. If a chair is rotating in the corner, you'll hear it rotating in that exact corner.
- **RTPCs** can control things like pitch and volume based on movement speed. A slowly rotating chair sounds different from one spinning violently.
- For continuous sounds (like a creaking door slowly opening), we use looping events that only play while the object is actually moving. No wasted CPU on silent loops.

Design Intent: Moving objects are how the player knows something supernatural is happening. By making their audio spatially accurate, we ensure players can locate anomalies through sound alone—which is the core of the gameplay.

5. WHAT IT SHOULD SOUND LIKE

[Mood representation picture]

Sonic Palette

Shadow Frames has a very specific audio character. Every sound should fit these qualities:

- **Warm analog:** Tape saturation, tube distortion, vinyl crackle—nothing pristine or digital
- **Dark and organic:** Wood creaks, fabric rustles, wet breathing—physical, tactile sounds
- **Wrong but familiar:** You recognize the sounds, but they're behaving unnaturally—uncanny valley for audio
- **Minimal music:** Mostly sound design drives the fear, not composed scores

Key References

These are the gold standard examples we're referencing. Listen to these to understand the vibe:

Amnesia: The Dark Descent — Whispers and distortion that increase with "sanity"

- We use this approach for our anomaly severity system—the more anomalies stack up, the more distorted reality becomes

Little Nightmares 2 (Mannequin Room) — Enemies that stop when you look at them

- Our enemy (if retained) uses this same audio behavior—footsteps stop, breathing continues

Limbo — Minimalist dark ambience that shifts between areas

- Our approach to room-to-room atmosphere—subtle but distinct

The Shining (1980) soundtrack — Dissonant strings for chase music

- Reference if we need intense music moments (like enemy chase sequences)

FNAF Sister Location menu music — Dark, mysterious, unsettling even in menus

- Our main menu vibe—ominous before you even start playing

Audio Moodboard Summary

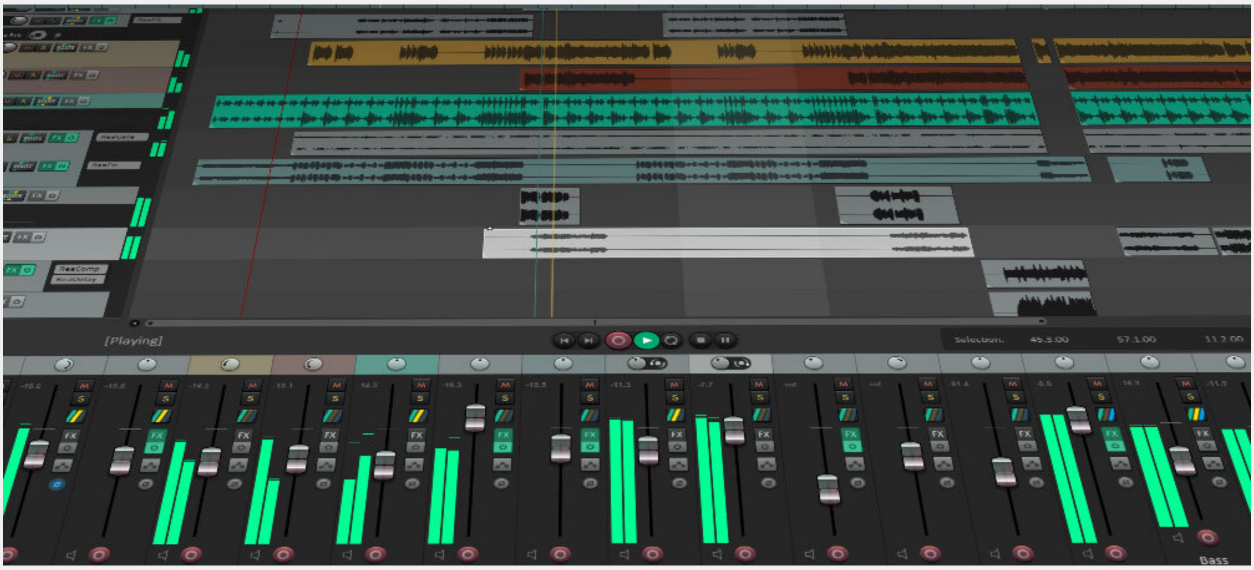
Imagine a 5-minute audio demo that layers all these elements to show how the game sounds:

- **Layer 1:** Pub ambience (clock ticking, distant traffic, building settling)
- **Layer 2:** Tension drones fade in gradually
- **Layer 3:** Anomaly stinger + directional cue + sustained ambient loop
- **Layer 4:** Camera observation mode (filtered, muffled audio)
- **Layer 5:** Photo capture sequence (shutter, flash, film advance)
- **Layer 6:** Enemy breathing and distant roar (if feature retained)
- **Layer 7:** Full escalation with all elements active at once

This demo would be created early in production to align the team on the audio vision before we start making final assets.

6. TECHNOLOGY

[REAPER Screenshot/view]



Software Stack

Here's the complete list of tools we're using and why we chose each one:

Tool	Purpose	Output Specs
Reaper	Audio editing, library processing, voice recording	48kHz/24-bit WAV, -23 LUFS (dialogue), -18 LUFS (ambience)
Wwise 2024.1.2.8726	Middleware (RTPC, state management, spatial audio)	See integration below
Game Engine	**Unreal Engine 5.4.4**	Official Wwise integration plugin available
Envato Elements	Primary sound library	Environmental sounds, ambience, textures
Soundly	Quick search/preview	Specific SFX needs

Wwise Organization

Here's how we're structuring the Wwise project. Keeping this organized now saves headaches later:

Actor-Mixer Hierarchy:

```
Default Work Unit
└── SFX
    ├── Player
    │   ├── Camera
    │   ├── Footsteps
    │   ├── Breathing
    │   └── Etc
    │
    ├── Environment
    │   └── Objects
    │       ├── Normal
    │       └── Anomalous
    │
    ├── Anomalies
    │   ├── Stingers
    │   └── Loops
    │
    └── Enemy (if retained)
        ├── Breathing
        ├── Footsteps
        └── Roars
```

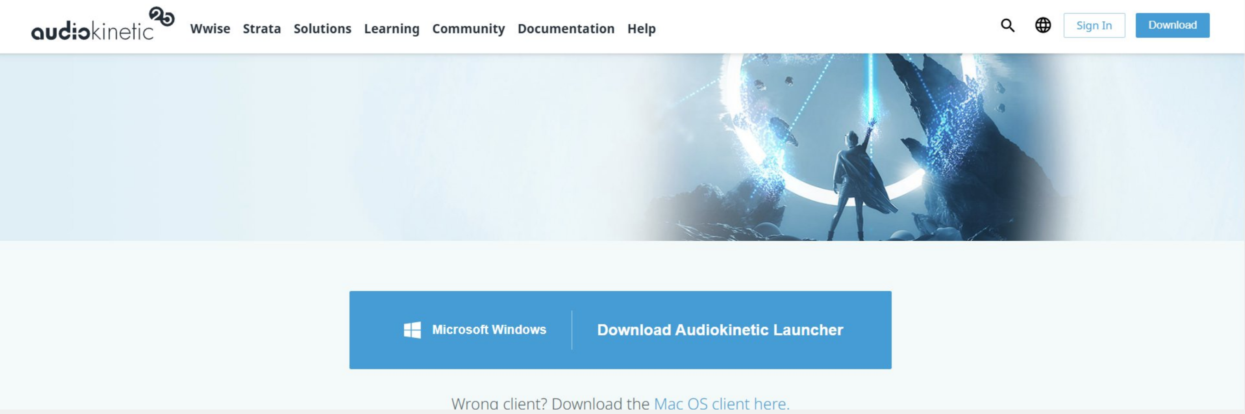
Wwise Integration with Unreal Engine

Getting Wwise working with Unreal is straightforward if you follow these steps. We've chosen **Option B** (Plugin Distribution) to keep our Perforce depot lean and avoid merge conflicts.

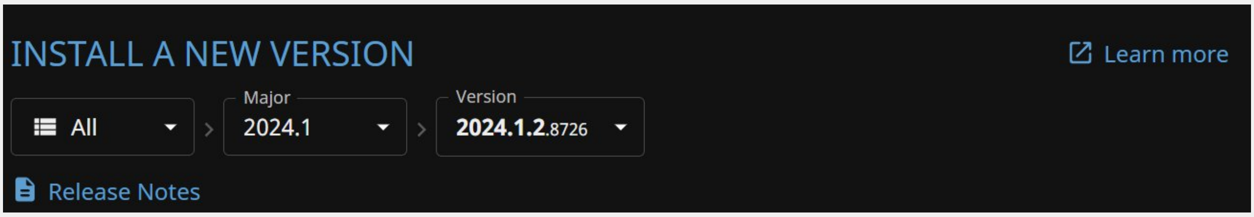
For Sound Designer (One-Time Setup)

1. **Download Wwise Launcher**
   - Get from: https://www.audiokinetic.com/download/
   - Install Wwise version 2024.1.2.8726 (specific version required for compatibility)
   - Create a new local Wwise project on your machine
   - **Important:** This Wwise project stays local—it does NOT go into Perforce

[IWwise Launcher screenshot]



[Wwise installation screen]



All

Major: 2024.1

Version: 2024.1.2.8726

1. **Your Workflow as Sound Designer**
   - Work in your local Wwise project—import sounds, create events, set up RTPCs
   - When ready to share with the team, generate **SoundBanks**
   - These SoundBanks get committed to Perforce at: **/Dev/Content/StreamingAssets/Audio/GeneratedSoundBanks/**
   - SoundBanks are the compiled audio data that Unreal loads—like executable files for audio

For All Team Members (Unreal Integration)

1. **Get Wwise Unreal Integration Plugin**
   - Download from Google Drive share (link will be shared by Week 1)
   - Extract to: /UE_5.4/Engine/Plugins/Marketplace folder
   - Open Unreal, go to Edit → Plugins, enable "Wwise" plugin
   - Restart Unreal Engine
1. **Sync SoundBanks from Perforce**
   - SoundBanks will be stored in Perforce at: **/Dev/Content/StreamingAssets/Audio/GeneratedSoundBanks/**
   - Standard Perforce sync: right-click project folder → "Get Latest Revision"
   - Whenever the sound designer updates audio, you just sync this folder
1. **First-Time Unreal Setup**
   - Open Unreal project after syncing
   - In Unreal, open the Wwise Picker window
   - Click "Refresh Soundbanks"—it should auto-detect the folder from Perforce
   - If it doesn't auto-detect, manually point it to: /Content/StreamingAssets/Audio/GeneratedSoundBanks/
   - Wwise audio is now available in Unreal!

Updating Audio (Standard Workflow)

This is how audio updates happen during production:

1. Sound designer makes changes in local Wwise project
2. Sound designer generates new SoundBanks
3. Sound designer commits SoundBanks to Perforce: /Dev/Content/StreamingAssets/Audio/GeneratedSoundBanks/
4. Team members sync from Perforce
5. Restart Unreal Engine to load the new audio (engine caches audio on startup)

Why This Approach: By keeping the Wwise project local and only sharing SoundBanks, we avoid Perforce merge conflicts (sound designer has full control), keep the depot size smaller, and make audio updates simple for everyone else—just sync and restart.

Version Control & File Naming

File Naming Convention:

[Category]_[Descriptor]_[Variation].wav

Examples:

- SFX_Camera_Shutter_01.wav
- MUS_MainMenu_Loop.wav
- AMB_RoomTone_Bar.wav
- VOX_Enemy_Roar_Close_03.wav

Perforce Structure:

```
/Depot
├── /GameProject (Unreal Engine project files)
├── /SoundAssets
│   ├── /ReaperProjects (session files - NOT in Perforce)
│   ├── /ProcessedAssets (exported WAVs ready for Wwise)
│   │   ├── /SFX
│   │   ├── /MUS
│   │   ├── /AMB
│   │   └── /VOX
│   └── /SourceRecordings (raw Roeland vocals for enemy)
```

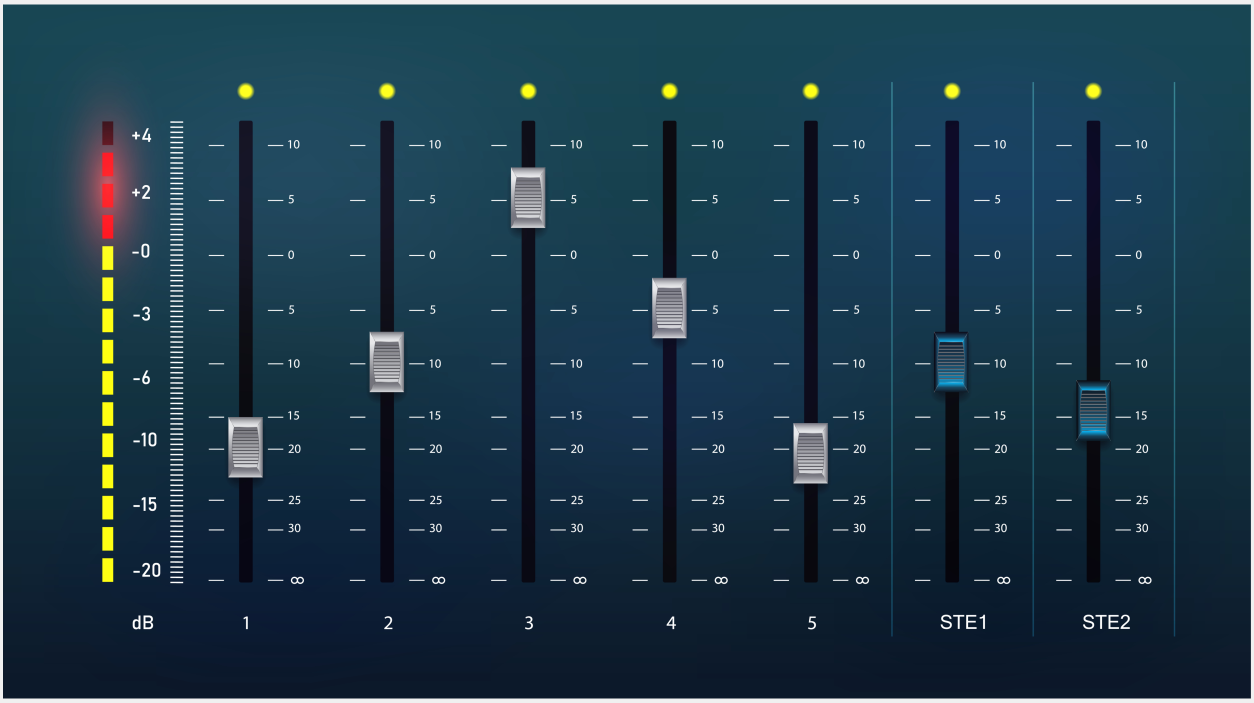
Audio Priority System

When multiple sounds are playing at once (and hitting our 32-voice limit), here's what stays and what can be culled:

1. **Priority 1:** Enemy breathing/footsteps (always audible, never culled)
2. **Priority 2:** Anomaly stingers (interrupt ambience, critical for gameplay)
3. **Priority 3:** Camera feedback (player action confirmation)
4. **Priority 4:** Environmental detail (lowest priority, can be culled based on distance)

Audio Channels & Compression

[Channels visual representation]



We're using a mix of mono and stereo sources depending on whether sounds need spatial positioning:

Mono sources (Wwise handles 3D positioning):

- Player footsteps (positioned at player location)
- Enemy breathing, footsteps, roars (positioned at enemy location)
- Anomaly stingers and presence loops (positioned at anomaly location)
- Environmental detail sounds (doors, objects)

Stereo sources (no spatial positioning):

- Camera sounds (always centered, player perspective)
- UI/Menu sounds (2D interface)
- Music (atmospheric, non-diegetic)
- Ambience room tones (fill entire space)
- Pin board interactions (2D puzzle interface)

Compression Strategy:

We're balancing audio quality against file size and memory usage. Here's the breakdown:

Asset Type	Format	Sample Rate	Bit Depth	Compression	Why This Format
Camera SFX	PCM	48kHz	16-bit	None	Critical gameplay feedback, needs instant response with zero latency
Enemy Audio	PCM	48kHz	16-bit	None	Player survival depends on hearing these clearly
Anomaly Stingers	PCM	48kHz	16-bit	None	Important event markers, no quality loss acceptable
Player Footsteps	Vorbis Q7	48kHz	—	~192 kbps	Play frequently, compression saves memory while maintaining quality
Environmental SFX	Vorbis Q6	48kHz	—	~160 kbps	Background detail, slight quality loss acceptable
Ambience Loops	Vorbis Q6	48kHz	—	~160 kbps	Long files, streaming candidates
Music	Vorbis Q8	48kHz	—	~256 kbps	High quality for mood, but memory-conscious
UI Sounds	PCM	48kHz	16-bit	None	Short files, negligible memory impact

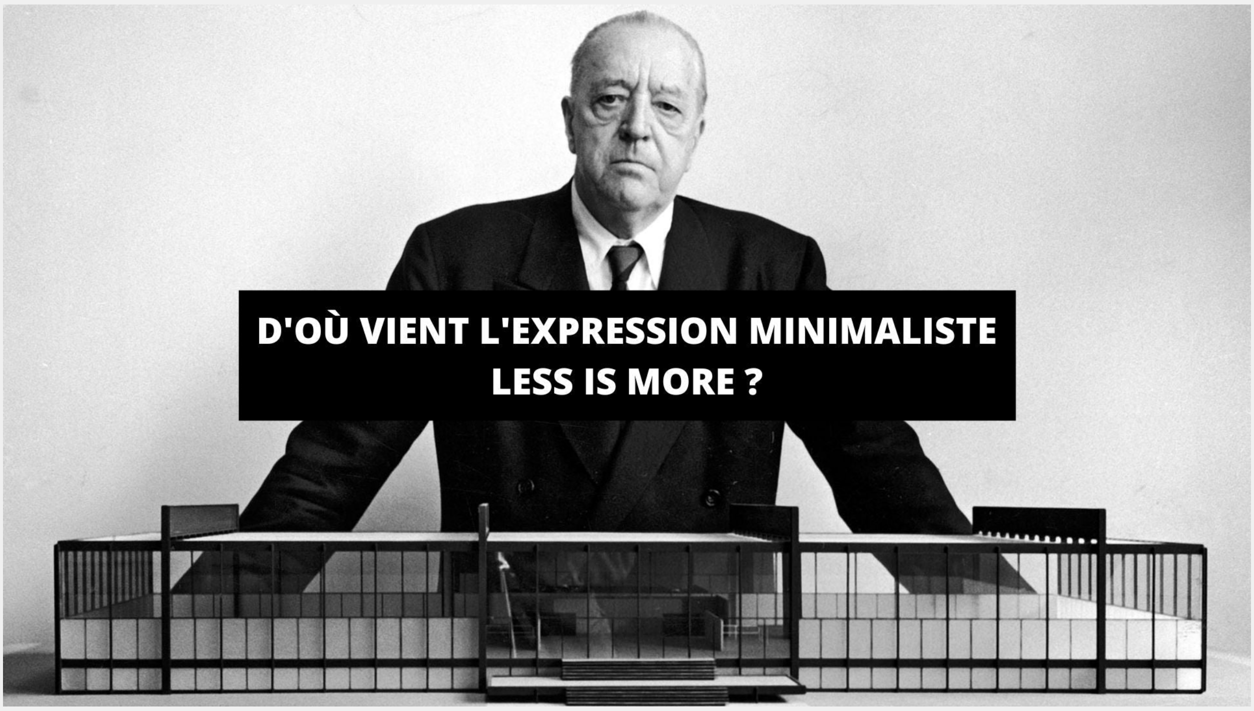
Estimated File Sizes:

- **Uncompressed (production):** ~1.4GB
- **Compressed (final build):** ~175MB
- **Streaming pool:** ~30MB (music + long ambience loops)
- **Memory-resident:** ~145MB (everything else loaded at startup)

7. MUSIC APPROACH

Philosophy: Less is More

[Mies van der Rohe, who popularized the phrase “Less is more,”]



Shadow Frames uses music sparingly and strategically. During most gameplay, sound design carries the entire experience—no background music. This makes the moments when music *does* appear much more impactful.

Music only appears in:

- **Main menu** (dark, mysterious 2:30 loop)

Example 1 | Example 2 | Example 3

- **Chase sequences** (panic, dissonant strings—if enemy feature retained)

The Shining Example

- **Pin board completion** (brief harmonic reveal)

Example 1 | Example 2

- **Endings** (bittersweet for good ending, dark drone for bad ending)
- **Credits**

During exploration: NO MUSIC. Sound design carries everything. This creates a stark contrast—when music finally appears, players know something significant is happening.

Dynamic Tension Layers

Instead of constant background music, we use 3 ambient-musical drones that fade in and out based on anomaly severity. These blur the line between music and sound design:

1. **Low rumble** (40-60Hz)—subsonic dread, felt more than heard
2. **Dissonant strings**—unsettling harmonics, no clear melody
3. **Heartbeat pulse**—rhythmic tension that increases with threat level

These are controlled dynamically by gameplay RTPCs, not pre-composed timing. The game itself "conducts" the tension music based on how many anomalies are active.

8. WHAT WE'RE CREATING

Important Note: The quantities listed below are conceptual estimates for planning purposes. Actual asset counts will be refined during production based on gameplay needs, implementation complexity, and time constraints. These numbers give us a target to aim for, not a rigid requirement.

Detailed Asset Breakdown

Category	Asset Type	Est. Qty	Priority	Notes
Player - Camera	Shutter clicks	~3	P0	Core mechanic feedback
	Flash pop + whine	~2	P0	Instant recognition
	Film advance	~2	P0	Satisfying mechanical feel
	Empty clicks	~2	P0	Clear "out of film" signal
	Reload sequence	1 set (4 parts)	P0	5-second vulnerable moment
	Raise/lower camera	~4	P1	Observation mode transitions
Player - Movement	Footsteps (carpet)	~8	P0	Primary surface type
	Footsteps (wood)	~8	P0	Varied pub spaces
	Footsteps (tile)	~8	P0	Bathroom area (if included)
	Clothing rustles	~5	P2	Movement detail
	Breathing loops	~6	P1	Normal + stressed states
Environment - Bar	Glass clinking	~8	P1	Period-specific heavier glass
	Bottle placement	~6	P2	Wood surface impacts
	Draft beer pouring	~4	P2	Ambience detail

	Cash register	~3	P2	Mechanical, 1960s type
Environment - Stage	Jukebox operation	~6	P1	Tube amp hum, vinyl surface noise
	Stage floorboards	~6	P1	Distinct from main floor creaks
	Microphone stand	~4	P2	Metal ringing, subtle
Environment - Tables	Chair sounds	~9	P1	Common anomaly target
	Cutlery on plates	~8	P2	Ceramic, period-appropriate
	Ashtrays	~6	P2	Cigarette taps, lighter clicks
	Playing cards	~4	P2	Shuffling, card slaps
Anomalies	Stingers	~8	P0	Event notification
	Anomalous loops	~12	P0	Sustained tension
	Transform sounds	~12	P1	Object distortions
Enemy (if retained)	Breathing	~5	P0	Primary threat indicator
	Footsteps	~18	P0	Tracking mechanic
	Roars	~5	P0	Periodic location cue
	Presence layers	~6	P1	Subsonic dread
Ambience	Room tones	~4	P0	Spatial definition per area
	Tension drones	~3	P1	Dynamic intensity layers
	Building sounds	~5	P1	Environmental life
	1960s period details	~12	P1	Rotary phone, tube radio, etc.

UI	Menu sounds	~5	P1	Navigation feedback
	Pin board	~8	P0	Puzzle interaction
Music	Tracks	~9	P0-P1	Strategic moments only

Total Estimated Assets: ~250-300 files (varies by enemy feature decision)
Uncompressed Size: ~1.2-1.4GB
Compressed (Wwise): ~150-175MB

9. HOW WE'RE MAKING IT

Sound Sourcing Strategy

Sound Library Approach:

- **Primary source:** Envato Elements subscription—extensive library of period-appropriate sounds
- **Secondary:** Soundly for quick searching and auditioning specific effects we can't find in Envato
- **Focus areas in libraries:**
  - 1960s-era mechanical sounds (vintage cameras, analog devices)
  - Wooden furniture and creaking (old building ambience)
  - Glass and ceramic impacts (heavier, period-appropriate types)
  - Environmental textures and room tones
  - Base layers for anomalous transformations

Voice Recording (Original Content):

- **What we're recording:** Enemy vocalizations (if feature retained) and any voiceover if needed
- **Location:** The Square studio facilities
- **Equipment:** Shure SM58 or Zoom H6 (depending on availability at time of recording)
- **Approach:** Multiple takes at varied intensities, will process heavily for supernatural character

Sound Design Process

Working with Library Sounds:

1. Search and audition sounds from Envato/Soundly
2. Import selected sounds to Reaper for editing and processing
3. Apply period-appropriate character:
   - Subtle tape saturation (Waves J37 or similar plugin)
   - Gentle high-frequency rolloff (starts at 8kHz—nothing sounds too crisp)

    - Boost low-mids (200-400Hz) for analog warmth
1. Create variations (pitch, timing, intensity) so sounds don't feel repetitive
2. Export at 48kHz/24-bit, target -23 LUFS for critical gameplay sounds

Transforming Sounds into Anomalies:

1. Take a normal library sound as the base (e.g., a regular chair creak)
2. Choose transformation technique based on anomaly type:
    - **Subtle wrongness:** -3 to +3 semitone pitch shift, slight time stretch (5-10%)
    - **Obvious supernatural:** -12 semitone shift, Paulstretch for ghostly texture
    - **Movement anomalies:** Speed up/slow down unnaturally, add subsonic rumble (40-60Hz)
    - **Shadow anomalies:** Reverse layers, whisper textures, formant shifting
1. Layer with additional elements for otherworldly quality:
    - Low-frequency drone (creates physical unease)
    - High-frequency shimmer (suggests supernatural energy)
    - Reversed fragments (adds ambiguity)
1. Test in Wwise with RTPC controls to ensure it responds correctly to gameplay

Enemy Voice Construction (if retained):

- **Recording session:** Deep vocalizations recorded in The Square studio with team member Roeland
- **Layering approach for roars:**
    - **Base layer:** Recorded voice pitched down -12 semitones
    - **Mid layer:** Animal sounds from library (bear/big cat textures)
    - **High layer:** Reversed speech fragments for unsettling quality
    - **Texture layer:** Granular processing, formant shifts
- **Processing:** Bus compression, unnatural reverb, subtle delay for "multiple throats" effect

Implementation Workflow

Here's the step-by-step process from sourcing a sound to having it working in-game:

1. Source or create sound (library search or recording session)
2. Edit and process in Reaper following guidelines above
3. Export to Perforce: /SoundAssets/ProcessedAssets/[Category]/ with correct naming convention
4. Import to Wwise, assign to appropriate Actor-Mixer hierarchy
5. Create Wwise event if needed, test basic playback in Wwise
6. Generate SoundBanks and commit to Perforce: /Dev/Content/StreamingAssets/Audio/GeneratedSoundBanks/
7. Integrate with Unreal Engine (programmers hook up events to gameplay)
8. Test in actual gameplay context with programming team
9. Update asset tracking spreadsheet with status

10. Perforce commit with clear description of what was added/changed

Quality Checks Before Marking Complete:

- ☑ No clipping or unwanted distortion
- ☑ Correct LUFS level for asset type
- ☑ Clean edits (no clicks/pops at loop points or boundaries)
- ☑ Proper file naming convention followed
- ☑ Sounds "period-appropriate" (no modern digital artifacts)
- ☑ Works with other sounds in mix (doesn't mask critical gameplay audio)
- ☑ In-engine test successful
- ☑ Team feedback collected and addressed

Tools & Plugins We're Using

Reaper Plugins (Built-in):

- **ReaEQ:** Surgical frequency shaping
- **ReaComp:** Dynamics control
- **ReaVerb:** Basic reverb (with custom IRs for period character)
- **ReaPitch:** Pitch shifting and formant manipulation
- **ReaDelay:** Timing effects and echo

Third-party (if available):

- **iZotope RX Elements:** Noise reduction and spectral editing (for cleaning recordings)
- **Valhalla FreqEcho or Supermassive:** Texture generation (free plugins)
- **Soundtoys Decapitator:** Analog saturation (if available)
- **PaulXStretch:** Extreme time stretching for ghostly textures (free)

For Enemy Voice Processing:

- Heavy pitch shifting and formant manipulation
- Multiple layers of processed vocal recordings
- Granular synthesis for unnatural character

Asset Priorities for Production Schedule

Here's the order we'll tackle assets, grouped by production phase:

Week 1-2 (Core Mechanics - First Playable):

- Camera sounds (P0) - shutter, flash, reload
- Basic footsteps (P0) - at least wood surface
- Anomaly stingers (P0) - at least 2-3 variations
- First anomaly presence loop

Week 3-4 (Enemy & Intensity Build):

- Enemy voice recording session and processing (P0) - if feature retained
- Anomalous object transformations (P0) - full set
- Tension drones (P1) - all 3 layers
- Additional footstep surfaces

Week 5-6 (Environment & Polish):

- Environmental detail sounds (P1-P2) - bar, stage, tables
- UI feedback sounds (P1)
- Pin board puzzle sounds (P0)
- Period-specific ambience details (P1)
- Room tones for each area

Week 7-8 (Music & Final Polish):

- Music composition or sourcing
- Additional variations for overused sounds (P2-P3)
- Final mix pass and balancing
- Performance optimization
- Bug fixes from QA

10. ACCESSIBILITY CONSIDERATIONS

Shadow Frames relies heavily on audio for gameplay, so we need to ensure players with varying hearing abilities can still experience the game. These features are designed to provide options without compromising the core horror experience.

Audio Options

Volume Controls (Separate Sliders):

- **Master Volume:** Global control for all audio
- **Music Volume:** Background music and menu tracks
- **SFX Volume:** All sound effects (anomalies, camera, objects)
- **Ambience Volume:** Environmental loops and room tones
- **UI Volume:** Menu interactions and interface feedback

Audio Presets:

- **Default:** Balanced mix as intended by sound design
- **Enhanced Clarity:** Boosts critical gameplay sounds (anomaly cues, enemy audio) by +3dB, reduces ambience slightly
- **Reduced Tension:** Lowers drones and tension layers, maintains gameplay-critical sounds at normal volume

Visual Audio Indicators

For players with hearing impairments, we're implementing optional visual aids that preserve challenge while ensuring accessibility:

Directional Audio Indicator (Optional Toggle):

- Visual arc on HUD showing direction of important sounds
- Color-coded: Red (enemy), Yellow (anomaly alert), White (environment)
- Intensity reflects volume—louder sounds create brighter/larger indicators
- Does not reveal exact location, preserves exploration and investigation challenge

Sound Labels (Optional Toggle):

- Brief text descriptions appear when critical sounds play
- Examples: "Anomaly Alert - East", "Enemy Breathing - Close", "Chair Scraping - Bar Area"
- Positioned near sound source in 3D space when possible
- Fades after 2-3 seconds to avoid UI clutter

Design Philosophy

All accessibility features follow these principles:

- **Optional by default:** Standard experience remains audio-driven for those who can use it
- **Preserve challenge:** Visual aids guide direction but don't solve puzzles or remove threat
- **Natural integration:** No immersion-breaking elements—everything fits the 1960s aesthetic where possible

11. COMPOSER BRIEF

(For external composer if we work with one)

Project Overview: Shadow Frames is a black-and-white psychological horror game set in a 1960s Irish pub. Music is used sparingly—only in menus, key moments, and endings. Most gameplay has no music, relying entirely on sound design.

What We Need:

- **Main Menu Track:** 2:30 dark loop, mysterious and unsettling. Sets the tone before gameplay begins.
- **3 Tension Layers:** 60s each, ambient-musical drones that blur the line between music and sound design. Should feel like reality breaking down, not traditional scoring.
- **Chase Music:** 60s intense loop with dissonant strings (if enemy feature retained). Panic without overwhelming—player still needs to hear footsteps.
- **Pin Board Completion Stinger:** 8s, brief harmonic reveal that feels ominous even in success.
- **Good Ending:** 90s, bittersweet resolution. Relief mixed with lingering unease.
- **Bad Ending:** 45s, dark drone collapse. Reality consumed.
- **Credits:** 3:00 reflective piece, allows player to decompress.

Musical Style:

- Dark, period-appropriate, restrained
- Tension through harmony and texture, not volume or intensity
- Analog warmth—think tape hiss, vintage recording artifacts
- References: The Shining soundtrack (dissonance), FNAF Sister Location menu (mystery)
- Avoid: Jump scare stingers, modern electronic sounds, overly orchestral

Technical Requirements:

- Format: Stereo WAV, 48kHz/24-bit
- Loudness: -18 LUFS integrated
- Stems required for tension layers (need to mix them dynamically in Wwise)
- All other tracks can be stereo mix

Reference Tracks: See Section 5 "Key References" for audio examples that capture the mood we're aiming for.

APPENDIX: PRODUCTION NOTES

Memory & Performance Budget

Here's our target performance envelope. We'll monitor these metrics weekly using Wwise Profiler:

Resource	Target	Critical Threshold
CPU Usage (Wwise)	<3%	<5%
RTPC Calculations	<0.5%	<1%
3D Positioning (15 voices)	<1%	<2%
Max Simultaneous Voices	32	48 (hardware limit)
Memory (Preloaded)	120-145MB	175MB
Memory (Streaming Pool)	30MB	50MB

Optimization Strategies:

- Voice culling at 50m distance for non-critical sources (environmental details only)
- Streaming for music and long ambience loops (>30 seconds duration)
- Vorbis compression for environmental SFX (acceptable quality loss for background elements)
- PCM (uncompressed) only for gameplay-critical sounds (camera, enemy, anomalies)
- Dynamic voice limiting with priority system ensures critical sounds never get culled

Streaming Strategy:

Asset Type	Load Type	Reason
Music tracks	Stream	Long files, infrequent playback, would waste memory if preloaded
Ambience loops	Stream	Continuous playback, can afford slight load latency
All SFX	Preload	Need instant playback response, small individual file sizes
Enemy sounds	Preload	Zero latency critical for gameplay and threat response

Platform Target:

- PC (Windows 10/11)
- Minimum spec: Intel Core i7 10870H or AMD Ryzen 9 7940HS, 8GB RAM, Nvidia GeForce RTX 3060 or AMD Radeon 780M
- Target: 60 FPS sustained at 1440p resolution
- Audio won't bottleneck on minimum spec—stays under 5% CPU budget

Known Technical Challenges

Here are anticipated problems and our planned solutions. We'll test these scenarios during weekly playtests:

Challenge 1: Anomaly Audio Overlap

- **Problem:** Multiple anomalies spawning quickly could create audio mud and confusion
- **Solution:** Anomaly stingers use priority system—newer events duck older presence loops by -6dB temporarily
- **Test case:** Spawn 3+ anomalies within 10 seconds, verify mix clarity and player can still identify locations

Challenge 2: Enemy Audio Through Walls

- **Problem:** Horror design requires enemy to feel omnipresent, but realistic occlusion reduces threat
- **Solution:** Enemy breathing has minimal occlusion (-3dB vs standard -12dB), maintains presence even through walls
- **Test case:** Player in one room, enemy in adjacent room—breathing should be audible but clearly muffled

Challenge 3: Observation Mode Filter Transition

- **Problem:** Instant filter sounds unnatural, slow transition delays gameplay feedback
- **Solution:** 0.2s crossfade between normal and filtered states—fast enough to feel responsive, smooth enough to avoid clicks
- **Test case:** Rapidly toggle observation mode 5+ times, verify no audio artifacts or pops

Challenge 4: 1960s Period Authenticity Verification

- **Problem:** Modern sound libraries may contain anachronistic audio we don't catch
- **Solution:** All sourced sounds undergo period verification—reference actual 1960s recordings, apply analog processing consistently
- **Test case:** Playtest with team members or external testers—does anything sound "too modern"?

Voice Over & Dialogue

Current Decision: No Voice Over

Shadow Frames uses exclusively environmental storytelling and diegetic sound. The game contains:

- **No** tutorial voice prompts
- **No/Yes (Partial)** internal character monologue
- **No** character dialogue
- **No** radio/TV voice snippets (only music and static)

Important Note: We consider adding occasional voiceover/dub after character triggers certain mechanic f.e player reads a letter or discovers anomaly in their polaroid view.

Rationale:

- Maintains immersion and sense of isolation
- Reduces localization complexity and costs
- Focuses production resources on core audio mechanics
- Notes and written documents provide narrative context visually
- Fits the investigative reporter character—you're documenting, not narrating

Exception: Enemy vocalizations (roars, distorted breathing) are non-verbal sound design, not traditional voice over, and don't require localization.

Weekly Playtests & Iteration

Audio requires constant testing in context. Here's our testing approach:

- **Focus areas:** Spatial audio accuracy, anomaly detectability, enemy tracking clarity (if retained)
- **Tools:** Wwise Profiler running during tests (CPU, memory, voice count monitoring in real-time)
- **Worst-case scenarios:** Deliberately spawn multiple anomalies + enemy + player actions simultaneously to stress-test the system
- **Feedback loop:** Document issues → prioritize fixes based on severity → implement solutions → retest following week
- **Team involvement:** Everyone playtests audio—fresh ears catch issues sound designer becomes blind to

PRODUCTION CHECKLIST

Track our progress through these milestones:

Pre-Production (Complete):

- ☑ Core vision defined and documented
- ☑ Audio pillars established
- ☑ Reference library compiled
- ☑ Tech stack decided (Unreal 5.4.4 + Wwise)
- ☑ Wwise integration strategy chosen (Option B)

Early Production (Weeks 1-2):

- ☐ Camera sounds implemented and working in-game
- ☐ Basic footsteps functional (at least 1 surface type)
- ☐ First anomaly audio sequence complete (stinger + loop)
- ☐ Wwise project structure finalized
- ☐ First playable build with integrated audio
- ☐ Spatial audio tested with headphones

Mid Production (Weeks 3-6):

- ☐ All anomaly types have audio (object, environmental, shadow)
- ☐ Enemy audio recorded & processed (if feature retained)
- ☐ Environment sounds complete for each pub area (bar, stage, tables)
- ☐ Spatial audio tested, tuned, and optimized
- ☐ Pin board puzzle audio implemented
- ☐ UI sounds complete (menus, HUD, prompts)
- ☐ Occlusion and reverb zones working correctly
- ☐ RTPC systems tested (distance, severity, observation mode)

Late Production (Weeks 7-8):

- ☐ Music tracks integrated (if composer available) or sourced
- ☐ All audio variations complete to avoid repetition
- ☐ Performance optimization complete (hitting CPU/memory targets)
- ☐ Accessibility features implemented and tested
- ☐ Final mix pass across entire game
- ☐ Full playthrough audio QA completed
- ☐ Known bugs fixed

Polish (Week 9+):

- ☐ Bug fixes from external QA and playtest feedback
- ☐ Balance adjustments based on player feedback
- ☐ Final Wwise bank optimization and compression review
- ☐ Documentation updated with any changes made during production
- ☐ Gold master audio approved by team
- ☐ Build tested on minimum spec hardware

CRITICAL REMINDERS

Always Remember These Core Principles:

- **Audio IS gameplay.** In Shadow Frames, sound isn't decoration—it's how players detect threats, locate anomalies, and solve the central puzzle. Every sound must be functional first, atmospheric second.
- **1960s authenticity matters deeply.** Modern sounds break immersion instantly. When in doubt, add analog processing (tape saturation, vinyl crackle, high-frequency rolloff) and verify against period references.
- **Less is genuinely more.** Resist the urge to fill every moment with sound. Silence and subtlety create far more tension than constant noise. Let the pub breathe.
- **Test spatially, always.** Always test audio in-engine with headphones in actual gameplay scenarios. What sounds perfect in Reaper might not read correctly in 3D space or during player movement.
- **Document as you go.** Update this Sound Bible when design decisions change. Production moves fast—if you don't document immediately, context gets lost. Future you (and your team) will be grateful.
- **Trust the three pillars.** When uncertain about an audio decision, refer back: Constant Dread, Analog Isolation, Ambiguity. If it doesn't serve one of these, reconsider it.

DOCUMENT MAINTENANCE

Sound Lead: Igor Szuniewicz

Document Version: 2.0 (Complete Production Revision)

Last Updated: [Current Date]

Review Schedule:

- **Weekly:** Quick team reviews during stand-ups—any blockers or questions?
- **Sprint milestones:** Major revisions after each 2-week sprint if needed
- **Pre-gold master:** Final approval and archival before release

Questions, Concerns, or Feedback? Contact sound lead Igor via Discord or raise issues in weekly team meetings. This is a collaborative document—if something's unclear or needs adjustment, speak up.

Change Log:

v1.0 – Initial draft of Sound Bible. Basic structure, tech stack outline, first gameplay–audio notes.
(Changelist: #587697)

v1.1 – Added tech stack visuals, table of contents, and new music section.
(Changelists: #587734–#587817)

v1.2 – Added sections on UI sounds, game states, and enemy sound design.
(Changelists: #587873–#588010)

v1.3 – Major expansion with gameplay–audio integration, RTPC, recording plan, and methodology.
(Changelist: #588613)

v1.4 – Grammar and formatting fixes, added illustrative images.
(Changelist: #588683)

v1.5 – Improved enemy and pinboard sections with supporting visuals.
(Changelists: #588836–#588906)

v1.6 – Added references, links, and refined readability.
(Changelists: #589250–#589593)

v1.7 – More reference links and updates to Camera System and Reload Sounds sections.
(Changelists: #590040–#590145)

v1.8 – File cleanup and naming convention fixes; added "How to Install" guide and folder structure visuals.
(Changelists: #590170–#591038)

v1.9 – Formatting updates, new sections on routing, optimization, and middleware integration.
(Changelist: #591891)

v2.0 – Full revision: added environment sounds, spatial audio, accessibility, performance budget, and technical challenges; improved readability and finalized document.
(Final changelist group)

This is a living document. It will evolve as production progresses and design decisions crystallize.

When in doubt about any audio decision, refer back to the three pillars:
Constant Dread • Analog Isolation • Ambiguity

— End of Document —