Feature Name: ***Fear Lockdown***

Quick Information

Owner:	*Karuna Ketan*
Programmer:	*Karuna Ketan*
Priority:	High ▾
Type of Feature:	Core Gameplay ▾
Progress:	Not Started ▾
Is a component:	No ▾ (Global System)

Overview

The "Fear Lockdown" is the signature mechanic for **Eclipse of Fear**. It's a system that permanently escalates all hostile forces and environmental dangers. This triggers every time the player completes one of the three main objectives (delivering Leave forms).

The player doesn't interact with this system directly. Instead, they suffer its consequences. The entire system is built to destroy the player's sense of relief after completing a quest. We replace that relief with a new, more intense wave of terror. This is the main engine that drives the game's core loop of escalating dread.

Intended Outcome

This feature's entire purpose is to mechanically reinforce our core design pillars: "Relentless Pressure" and "Escalating Dread."

We want the player to feel trapped. They have to complete their objectives to escape, but they'll learn fast that completing an objective is like striking a match in a gas-filled room. The emotional goal is to replace any feeling of "relief" with "panic" and "desperation." We will make the player dread their own success.

Art Design

The Lockdown is a distinct audio-visual event. It gives the player immediate and terrifying feedback for their action.

- **Lockdown 1 (1/3 Forms):** A 3-second, violent flicker of all lights in the building. A global audio stinger (loud, dissonant) plays, followed by the sound of multiple heavy doors slamming shut in the distance.
- **Lockdown 2 (2/3 Forms):** The player's flashlight beam flickers and narrows for 5 seconds. A brief, 1-second red "flash" fills the screen. This is accompanied by a more intense audio stinger and a closer, enraged roar from the Eclipse.
- **Lockdown 3 (3/3 Forms):** This is the climax. All standard lights in the building shut off for good, replaced by flashing, rotating red emergency lights. The world's colors desaturate, and a heavy vignette (darkness at the screen edges) appears. A building-wide alarm klaxon starts to sound.

Mechanics

The system has four distinct states or "Lockdown Stages." These are tied to the number of leave forms the player has delivered. The system globally modifies the parameters for the Eclipse AI, the Minion Spawner, and the environment itself.

Lockdown State Progression:

Lockdown Stage	**Trigger**	**The Eclipse (AI & Abilities)**	**The Minions (Spawning)**	**Environment & Audio**
State 0 (Start)	Game Start	**Passive Patrol:** Follows a simple, fixed path. Speed is equal to player walk. Chases only on direct line of sight.	**Inactive:** No Minions spawn.	**Quiet:** Low-level ambient building sounds. Tense, but not aggressive, music.
State 1 (1/3 Forms)	Player delivers 1st form	**Active Hunt:** Enters a dynamic, roaming state. Speed +15% (faster than player walk). Perception range +20%.	**Active:** Minions begin to spawn from predefined spectral spawn points in side rooms.	**Tense:** Audio "sting" & door slam event. Ambient music becomes more pulsing and ominous.
State 2 (2/3 Forms)	Player delivers 2nd form	**Aggressive Hunt:** Speed +20% (matches player sprint). Semi-omniscient (knows player's general quadrant). **Gains Ability: "Spectral Dash"** (a 10m teleport to close distance, 20s cooldown).	**Aggressive:** Spawn rate doubles. Minions can now spawn in main hallways, forcing conflict.	**Frantic:** Audio "sting" & roar event. Music adds frantic, percussive elements. Faint whispers begin.

State 3 (3/3 Forms)	Player delivers 3rd form	**Relentless Kill:** Speed +10% (faster than player sprint). **Fully Omniscient** (knows player's exact location). "Spectral Dash" cooldown reduced to 5s.	**Overwhelm:** Minions spawn constantly along the player's pre-defined escape route.	**Panic:** Audio "sting" & alarm event. All music replaced by a high-tempo "chase" track. Flashing red emergency lights activate.

How does the player interact with the feature? *The player never interacts with the feature directly. They only interact with its **trigger**.*

1. The player approaches a designated leave form delivery point (like a professor's desk).
2. The player initiates and completes the "Hold to Interact" prompt.
3. Once completed, the "Fear Lockdown" system triggers globally.
4. The player then experiences the A/V event and must survive the new, deadlier world state.

What are the steps involved to use this feature? *This section outlines the internal logic flow.*

1. A global integer, `LockdownStage`, starts at 0.
2. The Objective Interaction Point fires an event to the Game State (e.g., `OnObjectiveComplete`).
3. The Game State increments `LockdownStage` by 1.
4. The Game State broadcasts a global event, like `OnLockdownTriggered`, to all relevant systems.
5. All listening systems (Eclipse AI, Minion Spawner, Audio Manager, Lighting Manager) read the new `LockdownStage` variable and update their parameters.

Does this feature interact with anything other feature? Yes. This is a "hub" feature. It directs several other key systems.

1. **Eclipse AI:** The Behavior Tree must read `LockdownStage` to adjust its speed, logic, and ability use.
2. **Minion Spawner:** Reads `LockdownStage` to change its spawn rate and available locations.
3. **Objective System:** This is what triggers the feature when an objective is completed.
4. **End-Game State:** The 3rd Lockdown (State 3) triggers the final "Escape" objective.

What are the sub features?

1. **Global State Manager:** The core `LockdownStage` variable that tracks game progression.
2. **A/V Event System:** The scripted audio-visual "sting" for immediate player feedback.
3. **AI Parameter Scaling:** The system that feeds new values (like speed and aggression) to the Eclipse.
4. **Spawner Parameter Scaling:** The system that increases Minion spawns.

What you can control as a Designer in Engine

To allow for proper balancing and tuning, we must expose the following variables in a central Blueprint (like the GameState). This lets the designer adjust them easily.

- **For State 1:**
  - (float) `State1_EclipseSpeed`
  - (float) `State1_EclipsePerceptionRange`
  - (float) `State1_MinionSpawnRate`
- **For State 2:**
  - (float) `State2_EclipseSpeed`
  - (float) `State2_EclipseDashCooldown`
  - (float) `State2_MinionSpawnRate`
  - (bool) `State2_bCanMinionsSpawnInHallways`
- **For State 3:**
  - (float) `State3_EclipseSpeed`
  - (float) `State3_EclipseDashCooldown`
  - (float) `State3_MinionSpawnRate`
- **Event Assets (per stage):**
  - (SoundCue) `Stage1_Stinger_Audio`
  - (SoundCue) `Stage2_Stinger_Audio`
  - (SoundCue) `Stage3_Stinger_Audio`
  - (MusicCue) `Stage1_Ambient_Music`
  - (MusicCue) `Stage2_Ambient_Music`
  - (MusicCue) `Stage3_Chase_Music`

Timeline & Resources

Estimated Time to Implement and expected Due date:

Estimated Time: 1.5 Sprints (3 weeks)

Breakdown:

- **Sprint 1 (Weeks 1-2):** Core programming (Global state, AI/Spawner hooks), Audio (stingers, new tracks).
- **Sprint 2 (Weeks 2-3, partial):** VFX (light flickers, red lights), Implementation, and initial balance/testing.

Team Members Needed and Impact:

Role	Impact Level	How many	Notes
Game Designer	Mid	1	Tuning the difficulty curve of the 3 stages will be critical.
Animator	Low	0	Relies on existing Eclipse/Minion animations.
2D Artist	Low	0	
3D Artist	Low	0	Relies on existing environment assets.
Animation	Low	0	
Programmer	High	1	Needs to build the core system and hook it into AI, Spawners, Audio, and Lighting.
Narrative Designer	Low	0	Logic is driven by objectives, not narrative beats.
UI/UX	Low	1	Only for the objective text updates.
Audio	High	1	The entire soundscape changes. New stingers and music are required.
VFX Artist	Mid	1	Needed for the "event" stingers (light flickers, red lights).

Dependencies:

- The base Eclipse AI (including its Behavior Tree and "Spectral Dash" ability) must be complete first.
- The base Minion Spawner system must be functional.
- The Objective Interaction system must be in place.
- The Global Game State must be set up.

Features dependant on this feature:

- The End-Game Sequence. It relies on State 3 being triggered to unlock the exit.
- Player Progression. This *is* the core progression and difficulty-scaling system.

Programming Notes

Provide any programming notes, screenshots, explanations, etc here using the table below.

Date	**Name**	**Notes**
NA	Programmer	This whole system should be managed from the `GameState`. The `LockdownStage` variable needs to be replicated so all other systems (AI, Audio, VFX) can access it reliably. We should create a single, reusable function (like `OnLockdownTriggered(stage)`) to fire all the "Event" feedback. This will keep it clean.

Known Bugs

Use the table below to provide any information on known bugs.

Date	**Name**	**Description**	**Priority**	**Status**
N/A	N/A	Feature is "In Design." No known bugs yet.	- ▾	Not Started ▾

Additional Notes

Provide any additional notes, screenshots, explanations, etc here use the table below.

Date	**Name**	**Notes**
NA	Karuna Ketan	This feature is the lynchpin of the entire game loop. Getting it right is critical to delivering the "escalating dread" I'm aiming for. The audio-visual "sting" for each event needs to be impactful and jarring. It has to sell the consequence of the player's action properly.