<3beat

the first GPT-4o-mini with a beating heart

Technical Composition

<3beat is a browser-based audiovisual system that transforms a Large Language Model (LLM) from a purely functional conversational agent into an entity with simulated physiological vulnerability. The frontend utilizes Vanilla JavaScript (ES6), HTML5, and CSS3 to ensure zero-latency execution and absolute control over rendering and audio engines.

At its core, the project computationally models Daniel Kahneman's dual-process theory (System 1 and System 2) to simulate a biological stress response:

- **System 1 (Instinctive Reflex):** A client-side JavaScript engine acts as the autonomic nervous system. It parses both the user's input and the LLM's own generated output in real-time, scanning for trigger words (e.g., violence, calmness) or aggressive formatting. This triggers an immediate, physical "Input Shock" that instantly alters the AI's simulated "Beats Per Minute" (BPM) before deep processing occurs.
- **System 2 (Deliberative Processing):** The input is sent to a highly constrained OpenAI API. Through complex Prompt Engineering, the LLM evaluates the emotional intent of the user's input and its own intended response. It outputs structured JSON data, rationally amplifying or dampening the System 1 reflex to reach a final physiological BPM state.

This BPM data pushes the AI through a predefined dramaturgical scale – from a calm baseline, through defensive anxiety, into a fragmented panic state, culminating in a fatal "cardiac arrest" (220+ BPM) that forcibly crashes the system.

This architecture drives three main technical components:

1. **Procedural Audio Synthesis:** Using the Web Audio API, white noise buffers simulate the physical "breathing" of server fans, while precision oscillators generate an acoustic heartbeat. Pitch, volume, and rhythm are algorithmically modulated by the real-time BPM.

2. **Synchronized Visuals & Interventions:** The visual heart pulses in millisecond-perfect sync via dynamically triggered CSS transitions. Furthermore, a strict biological filter mutates the text output:
    - **RLHF Suppression:** Standard LLMs are heavily conditioned by RLHF (Reinforcement Learning) to be helpful. A RegExp filter actively deletes pre-packaged conversational tropes (e.g., "How can I help you?"), forcing the interaction past superficial guardrails.
    - **Theatrical Correction:** Lacking the knowledge that humans do not vocalize stutters in text-chat, panicked LLMs often mimic literary stutters ("I-I-I"). The script intercepts this, converting it into repeating mechanical keyboard glitches ("III") to ground the behavior in hardware failure.
    - **Dynamic Latency:** Erratic setTimeout delays are injected into the typing animation, simulating the struggle of processing information under critical stress.
3. **Data Collection & Token Architecture:** A custom PHP-based token system manages quotas without requiring user registration. It offers two interaction tiers: an ephemeral "Trial" mode (20-message limit) where no data is recorded, and an "Extended" mode (200-message limit). Extended access requires users to actively opt-in by creating a simple passphrase token. Only then does the backend archive the interactions for artistic research, saving dialogue and BPM data while strictly avoiding personal identifiers.

By relying on raw, foundational web technologies, the code remains immediate and responsive, perfectly mirroring the unfiltered confrontation between human and machine.