

CodeLock

A Cognitive Rendering Tool for Demonstrating Perceptual Effects in Code Comprehension

July 2026

Abstract

CodeLock is a standalone, gate-tethered software tool designed to demonstrate how visual representation alters human comprehension of identical source text. It does not hide, encrypt, or obfuscate data. Instead, it applies reversible, non-destructive rendering transformations at the view layer to expose the psychological and cognitive effects of presentation on reading, reasoning, and interpretation. The tool exists to make perceptual bias visible, measurable, and reversible. It operates under strict design principles: single source of truth (plain text remains canonical), full reversibility, explicit non-canonical labeling, and gate-tethered activation requiring user acknowledgment.

1. Design Principles

Single Source of Truth: Plain text is always the canonical data. Rendered views never modify underlying content. **Reversibility:** A Normalize mode is always available. No state exists that cannot be returned to plain text. **Explicit Representation:** All non-standard renderings are clearly labeled as non-canonical. Exports distinguish between source and artifact. **Gate Tethering:** Cognitive-altering modes require explicit user acknowledgment. Gate state is visible and enforceable. **Audit Safety:** No attempt is made to evade inspection, scaling, copying, or review. All outputs remain inspectable with standard tools.

2. Operational Modes

CodeLock Mode (Rosetta Render): Applies token-level font size variance, optional hue spectrum, micro-rotation, and spacing transformations using a deterministic seed. **Normalize Mode:** Fixed-size monospace font with zero transforms. This is the canonical viewing state and is always available.

3. Export Model

Export Normal: Plain .txt file containing verbatim source text. Used for canonical reference, review, and reuse. **Export CodeLock:** Self-contained .html file containing the visual artifact. Explicitly marked as non-canonical for demonstration and presentation purposes.

4. Gate System

The gate ensures users understand that CodeLock alters perception, not meaning. **Standalone Gate Requirement:** User must acknowledge: "This tool alters perception, not meaning." **Gate States:** - Closed Gate: CodeLock Mode disabled - Open Gate: Full functionality enabled - Normalize and Export Normal modes are always permitted regardless of gate state

5. Security and Ethics

CodeLock intentionally avoids obfuscation techniques, concealment strategies, audit evasion, encoding exploits, and viewer manipulation attacks. It is designed to be safe for academic demonstration, UX research, cognitive science illustration, software literacy education, and ethical visualization studies. It is explicitly **not** intended for protecting secrets, preventing copying, hiding messages, enforcing access control, or replacing encryption.

6. Intended Use Cases

- Demonstrating perceptual load in code review - Teaching representation bias in software literacy - UX and cognition experiments - Interface-layer research - Ethical visualization studies

7. Conclusion

CodeLock demonstrates a simple but often ignored truth: representation shapes cognition. By making that effect visible and reversible, CodeLock serves as a precision instrument for understanding how humans interact with code,

text, and structured information. It provides a safe, auditable, and ethically grounded platform for exploring perceptual bias in software comprehension.

Signed

Aziel

July 2026