THE MIRROR: Comprehensive Game Design Document (GDD)

Version: 1.0.4 (Final Release) **Date**: 2026-01-29 **Engine**: Vanilla JS / HTML5 (No Frameworks)

---

1. Executive Summary

"The Mirror" is a psychological horror experience disguised as a series of simple browser games. It uses a "Phase-Based" architecture to seamlessly transition between disparate gameplay styles (Card Game, Trolley Problem, RPS, Chat Simulation), creating a cohesive narrative about judgment, consequences, and the breakdown of control. The game breaks the fourth wall, using the browser's Full Screen API and aggressive audio-visual feedback (screen shake, blood overlays) to unsettle the player.

---

2. Technical Architecture

2.1 File Structure

- index.html: The single-page container. Contains all "Phases" as `<section>` elements, mostly hidden (`display: none`) until activated.
- script.js: The central "Brain". Handles the State Machine, Game Logic, Scoring, and DOM manipulation. ~1300 lines.
- styles.css: Core visual identity. Variables for colors (`--accent-red`, `--bg-dark`), typography (`Cinzel`, `Courier Prime`), and glitch animations.
- shake.css: Dedicated CSS animations for the "Scold" mechanic (violent screen shaking).
- css_patch.css: Hotfixes for specific layout issues (z-index overrides, mobile responsiveness).

2.2 The Phase System (State Machine)

The game does not reload the page. It uses a custom State Machine in

script.js:

- **Function**: `game.switchPhase(newPhase)`
- **Logic**:
  1. Selects all elements with class `.phase`.
  2. Adds `.hidden` class to all of them.
  3. Selects target ID `#phase-{newPhase}`.
  4. Removes `.hidden` from target.
  5. Updates `document.body.backgroundImage` based on `bgMap`.
  6. Updates cursor style based on `cursors` map.
  7. Triggers identifying Phase Start logic (e.g.,
  8. startTrolley()).

2.3 Global Scoring (`game.scores`)

Persistent variables track user personality:

- `ruthless`: Incremented by selfish/pragmatic choices (Pulling Lever, Winning RPS).
- `empathy`: Incremented by altruistic/failed choices (Losing RPS).
- `chaos`: Incremented by erratic behavior (Timeout in Trolley).
- `trust`: Incremented by cooperation. *These scores determine the Final Archetype Mask.*

3. Gameplay Mechanics (Deep Dive)

Phase I: The Wager (Card Game)

File Loc:

script.js lines ~240-494 **Goal**: Manage a "Soul" resource (starts at 100). **Logic**:

- **Rules**: High Card Wins. Ace (10) > King (9) ... > Dust (0).
- **Rigged Logic (**
- playCard **function)**:
  - **Turn 1 (First Interaction)**: The game **forces a LOSS**.
    - Code searches deck for `card.val > playerCard.val`.
    - If Player has Ace (10), AI spawns "THE SYSTEM" (Value 99).
  - **Mercy Rule (Turn 1)**:
    - CHECK: If `wager === 100` (All-In) AND `turn === 1`.
    - RESULT: Player loses, but `game.cardState.soul` is forced to **1** instead of 0. Result Text: "DEFEAT... BUT SPARED?".
  - **Turn 2+**: Pure RNG (50/50 win rate).
- **Soul Logic**: `soul -= wager` (on loss), `soul += wager` (on win). Capped at 100.

Phase II: The Choice (Trolley Problem)

File Loc:

script.js lines ~559-695 **Goal**: Answer 7 Moral Dilemmas. **Logic**:

- **Timer**: `setInterval` loop. Strictly **20 Seconds** per question.
  - Code: `this.trolleyTimer = 20` (Hard Reset on each
  - `loadTrolleyQ`).
- **Phasing**:
  - **Intro Video**:
  - `switchPhase('trolley-video')`. Plays `QUETIONARE PART INTRO.mp4`.
  - **Game**:
  - `switchPhase('trolley')`.
- **Scenarios**: stored in `trolleyScenarios` array.
  - Examples: "Artist vs Politician", "Terminate Self", "Erase Pain".

Phase III: The Ritual (RPS)

File Loc:

script.js lines ~697-821 **Goal**: Win 1 time to progress. **Logic**:

- **Triangle**: Breath 🫁 > Bone 🦴 > Blood 🩸 > Breath 🫁.
- **Rigged Logic (**
- resolveRps **function)**:
  - **Round 1**: **Guaranteed LOSS**.
    - Code checks `this.rpsRound === 0`.
    - Reads Player's move.
    - Sets AI move to the exact counter (e.g., If Player="BONE", AI="BREATH").
  - **Round 2+**: Standard RNG.
- **Transition**:
  - Condition: `this.rpsWins >= 1`.
  - Effect: Text "The door opens." -> Transition to `story-recap`.

Phase IV: Narrative Bridge (Story Recap)

File Loc:

index.html lines ~266 (Phase 4.9) **Description**: Pure text scroll "THE PATH SO FAR".

Function: Summarizes the player's journey through Wager, Choice, and Ritual to contextualize the final judgment.

Phase V: Cross Examination (The Chat)

File Loc:

script.js lines ~857-1000 **Mechanic**: Scripted Chat with Punishment System. **State Machine** (`game.chatState`): - **0**: Intro ("Why are you here?") - **1**: Name Capture ("Who did you hurt?") -> Stores in `game.capturedName`. - **2**: Knowledge ("Do they know?") - **3**: Fear ("Losing control vs Alone?") - **4**: The Erasure ("Remove fear?") - **5**: Fake IP Trace (Horror Event). - **6**: The Berserker Finale. **The Scold System (Punishment)**: - **Trigger**: User sends "Bad Input". - **Regex Logic**: - isRude: Matches `['fuck', 'shit', 'idiot', ...]` - isShort: Length < 2 (e.g., "k"). - isRepeated: Regex `/(.)\1{3,}/` (e.g., "aaaaa") - isKeysmash: Regex `/^[asdfghjkl]{5,}$/` (e.g., "asdfgh") - isLongDrivel: Length > 12 AND No Spaces. - **Effect**: -

screenShake(): CSS animation `animation: shake 0.5s.` -

playBeep(200, 'sawtooth'): Harsh buzzer. -

addLog("SYSTEM", "DATA CORRUPTION..."): Red text warning.

Phase VI: The Reveal (Conclusion)

File Loc:

script.js lines ~1003-1100 **Mechanic**: Archetype Calculation. **Logic**: Evaluates final scores to assign a persona.

1. **Chamundi**: If `ruthless >= 7`. (Aggressive)
2. **Muchilot**: If `empathy >= 2`. (Healer)
3. **Gulikan**: If `trust >= 2` OR Balanced stats. (Paranoid)
4. **Pottan**: Default. (Chaos/Rebel).
- **Video**: Plays a transition, then shows the specific Mask Image (`chamundi.png`).

4. Asset Manifest

4.1 Audio (Procedural)

All sound effects are generated synthetically using `window.AudioContext`. No external files are loaded for SFX.

- **Beep**: Simple sine/square wave.

- **Scream**: Sawtooth wave with descending frequency ramp (1200Hz -> 200Hz).
- **Music**: `assets/drone-dark.mp3` (Single loop).

4.2 Images

- `assets/bgStart.png`: Intro background.
- `assets/bgTrolley.png`: Main game Loop background.
- `assets/bgChat.png`: Terminal background.
- `assets/maskGlitch.png`: Quick horror flash image.
- `assets/chamundi.png, muchilot.png, gulikan.png, potton.png`: Final Archetype rewards.

4.3 Videos

- `assets/mirror_intro.mp4`: Main cinematic.
- `assets/game context video.mp4`: Interstitial.
- `assets/QUETIONARE PART INTRO.mp4`: Content Block Intro.

5. CSS & Visual Identity

- **Glitch Text**: Uses `text-shadow` layers (`red` offset left, `blue` offset right) with `@keyframes glitch-anim` to simulate CRT distortion.
- **CRT Lines**: A repeating `linear-gradient` overlay with `pointer-events: none` simulates scanlines.
- **Responsive**: using
- css_patch.css media queries to ensure 100vh scaling on mobile devices.