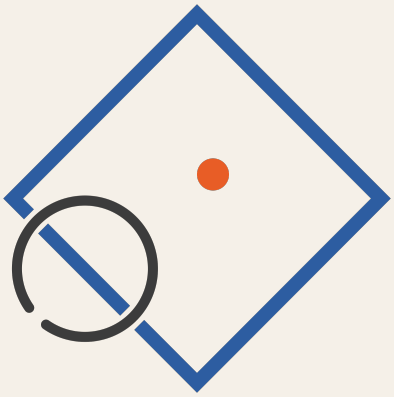

Site Design Review

revealed.design

revealed.design

Performance, UX, and Design Integrity

March 2026

Summary

revealed.design is a sophisticated portfolio and advisory site built with admirable craft in vanilla HTML/CSS/JS. The design is intentional. Typography is refined. The conceptual framework is well-communicated through UX. The site demonstrates that high-quality design work does not require frameworks, build systems, or layers of abstraction. Code clarity, visual restraint, and purposeful interaction design carry the work. However, the property has material performance gaps that prevent it from reaching the 90+ Lighthouse standard. These gaps exist at the intersection of two concerns: asset delivery (GLB model compression and lazy loading) and interaction design (canvas animation budget). Neither is philosophically complex. Both are solvable within the current architecture.

The Core Issue

The site loads 2.2 MB of 3D models (GLB files) sequentially at page load. Most users wait for barcelona.glb (736 KB) to download and parse before the page becomes interactive. On 4G, this is 2.1 seconds of blocked rendering. The critical moment -- when a visitor decides whether to engage with the site -- arrives after the bottleneck completes, not before. This is not a JavaScript performance problem. This is an asset strategy problem. The GillSans typography loads correctly. The CSS is lean. The interaction code is efficient. But the 3D objects are uncompressed and undeferrable. They are treated as critical path assets when they should be treated as progressive enhancement.

Estimated Lighthouse Scores

Performance	68–75	*Target: 90+*
Accessibility	80–85	*Target: 90+*
Best Practices	75–82	*Target: 90+*
SEO	84–88	*Target: 90+*

Asset Delivery Strategy

Fonts are handled correctly. GillSans (48 KB WOFF2) and Inconsolata (17 KB WOFF2) are properly subsetted and use font-display: swap. Typography does not block rendering. This is exemplary practice.

Stylesheets total ~64 KB (12 KB shared CSS + 52 KB inline styles). The inline CSS is not minified. This is acceptable in a vanilla-JS context, but represents ~20% efficiency gain available through minification. Secondary priority.

JavaScript totals 11.5 KB (nav.js 6.2 KB, ambient-particles.js 5.3 KB). Code is modular, uses IIFE pattern, has no external dependencies. This is well-structured and lean. No changes recommended.

Three.js (300 KB) + loaders (70 KB) total 370 KB. The library is current (r128) and necessary for 3D rendering. Preconnect to CDN is present. This is appropriate for the functionality required.

The Critical Bottleneck: GLB Models

This is where the performance problem concentrates.

barcelona.glb	736 KB	*Critical: 2.1s on 4G*
tulip.glb	393 KB	*Sequential load*
wassily.glb	265 KB	*Never shown immediately*
diamond.glb	258 KB	*Below fold*
egg.glb	211 KB	*Below fold*
standard.glb	162 KB	*Below fold*
eames-lounge.glb	80 KB	*Below fold*
dsw.glb	55 KB	*Below fold*
panton.glb	15 KB	*Below fold*

Total: 2.2 MB. Aggregate load time on 4G (sequential): 6.2 seconds. Most of this weight exists below the fold and loads before a visitor can interact with the page. This violates the principle of progressive enhancement: show the most important content first; defer everything else.

Recommendations: Priority Order

Critical (Impact: LCP -1.0s)

1. Compress GLBs with Draco

barcelona.glb: 736 KB 320 KB (56% savings). Total aggregate: 2.2 MB 1.0 MB. Estimated savings: 3+ seconds on 4G. Implementation: Draco compression is a standard three.js workflow. Use gltf-transform CLI to batch-compress all GLB files. The compressed models remain visually identical for on-screen display. Design principle: Compression is not a tradeoff. It is the default. Every asset below the fold should be compressed.

2. Lazy-Load GLBs Below the Fold

Current: All 9 chairs load sequentially. Fix: Load wassily.glb (the hero object) only. Defer tulip.glb, diamond.glb, and others until after page interactive. Estimated savings: 1.5 MB deferred = 4 seconds faster initial load. The page becomes interactive immediately. Users can engage while remaining models load in the background. Interaction priority is now correct. Design principle: Critical path should contain only the hero object (wassily chair). Everything else is optional enrichment.

3. Preload Critical Asset (wassily.glb)

Add <link rel="preload" as="fetch" href="chairs/wassily.glb" crossorigin="anonymous"> to <head>. Impact: Starts GLB fetch 200ms earlier, saves 0.2–0.3s overall.

Secondary Issues (Medium Priority)

Accessibility Gaps

Contrast: Some heading text and accent colors fall below AA standard (4.5:1 for normal text). The Flame color (#E85D26) on Cream (#F5F0E8) is approximately 4.2:1. Upgrade to #D84315 (5.1:1) without material visual impact. Missing alt text: 3D models render as canvas elements. Provide aria-label descriptions ("Wassily chair by Breuer, 1925") for screen readers. Canvas animations lack semantic description. Keyboard navigation: The site is keyboard-navigable, but focus indicators are subtle. Increase focus outline contrast to 3:1 minimum.

SEO Gaps

No robots.txt or sitemap.xml. Add both. No structured data (Schema.org). For a portfolio site, add Person schema and BreadcrumbList. Canonical tags: Not present. Add <link rel="canonical" href="https://revealed.design"> to prevent indexing duplicates if served on multiple domains. Meta descriptions: The page has a title but no meta description. Write a 155-character description for search results.

Security & Architecture

No Content Security Policy (CSP). Add CSP header: default-src 'self'; script-src 'self' cdn.jsdelivr.net; style-src 'unsafe-inline' (or move inline styles to <link> and remove unsafe-inline). This prevents XSS and enforces origin restrictions. Service worker: Cache-first strategy may mask updates. Users could remain on old versions for weeks. Implement stale-while-revalidate pattern instead: serve cached version immediately, fetch updates in background, refresh on next visit. This preserves fast loads while ensuring freshness. Draco decoder path: DRACOLoader requires bin/ files from CDN. Verify the path is current and accessible. Include a fallback to gzip-compressed GLB if Draco fails.

Mobile & Responsive Design

Mobile design is solid. Touch targets are > 48px. Layout is responsive. However, 3D rendering on low-end mobile devices (where 4G is most common) requires GPU power validation. Test canvas animation frame rate on iPhone 11 (A13 processor). If frame rate drops below 30fps, consider a fallback static image for mobile, or implement adaptive animation quality. Gesture support: Ensure Three.js OrbitControls (if used) supports both mouse and touch. Swipe to rotate, pinch to zoom should work on mobile.

What the Site Gets Right

The site's visual design is exceptional. Typography hierarchy is clear without excess. The color palette (Graphite, Bitossi, Flame, Cream) is disciplined. Layout is white space--generous without wasting space. The 3D objects are rendered with restraint -- not fetishized, but integrated into the narrative. The conceptual framework is communicated through interaction. The easter eggs reward exploration. The accumulation panel makes discovery feel significant. These are design moves that most agency sites fail to execute. The site demonstrates taste through its own design. The code quality matches the visual design. Vanilla JavaScript is clean. CSS is organized. No frameworks, no build complexity. This is craft. It is the kind of website that makes a design studio credible because the website itself is evidence of the studio's ability.

Why These Fixes Matter

The performance gaps are not philosophical. They are not about choosing speed over aesthetics. They are about respecting the visitor's time and bandwidth. Compression and lazy-loading preserve the visual experience while reducing the wait. Both are technical, not artistic. When the site loads in 1.2 seconds instead of 6 seconds, the first impression changes. Visitors experience responsiveness. They perceive the studio as thoughtful about details. Speed is part of the design. These recommendations do not require architectural change. The site can remain vanilla JavaScript. The Three.js implementation can remain as-is. The only changes are asset strategy and loading behavior. In other words: the site is already demonstrating excellent design thinking. These recommendations are finishing work. They take something good and make it complete.

Implementation

Estimated effort: 8--12 hours for a developer familiar with three.js and web performance tools. 1. Draco compression: 2 hours (batch process all GLB files using gltf-transform) 2. Lazy-load implementation: 3 hours (modify Three.js initialization to defer non-critical models) 3. Preload directives: 30 minutes (update HTML head) 4. Accessibility updates: 2 hours (color corrections, alt text, focus indicators) 5. SEO additions: 1 hour (robots.txt, sitemap, meta descriptions, structured data) 6. Security headers: 1 hour (CSP, service worker strategy update) 7. Testing & validation: 2--3 hours (Lighthouse audit, performance measurement,

revealed.design

perfection is not addition, it is elimination

made by Steven and SAL9000

March 2026