Strategic Analysis & Operational Framework: La Nuit de l'Info 2025

1. The Core Narrative: "David vs. Goliath"

The 2025 subject is explicitly political and narrative-driven. To win, your project must embody the spirit of the **NIRD** (*Numérique Inclusif, Responsable et Durable*) initiative.

- **The Enemy (Goliath):** The "Empire of Big Tech" (Microsoft/Windows). The urgency is driven by the **End of Life of Windows 10** (Oct 2025), which threatens to turn millions of functional school computers into e-waste due to planned obsolescence.[1]
- **The Hero (David):** The "Resistant Village" (Schools). Like the Gauls in Asterix, they resist by using **Linux** and **Open Source** tools to reclaim their digital sovereignty.[2]
- **Your Mission:** You are not just building a website; you are building the **recruitment and training center** for this resistance.

---

2. Technical Stack & Animation Strategy (The Winning Formula)

To balance the conflicting requirements of **"En trois dimensions"** (3D Challenge) and **"Green IT!"** (O2 Switch Challenge), we will use a specific high-performance stack.

2.1 The Animation Engine: Framer Motion + R3F

You asked about GSAP vs. Framer Motion. For **Next.js** in 2025, **Framer Motion** is the superior choice for this specific hackathon for three reasons:

1. **React Integration:** It handles "Exit Animations" (page transitions) natively with <AnimatePresence>, which is critical for the "smooth transition between pages" you requested.[3]
2. **Bundle Size:** It is generally tree-shakable and lighter than a full GSAP suite, aiding the EcoIndex score.[5]
3. **Development Speed:** In a 15-hour hackathon, its declarative syntax (<motion.div animate={{ x: 100 }} />) is faster to write than GSAP's imperative .to() chains.[6]

The Hybrid Architecture:

- **2D UI & Transitions: Framer Motion**. Use it for micro-interactions (buttons scaling on hover), staggered text reveals, and page wipes.[4]
- **3D Experience: React Three Fiber (R3F)**. This acts as the "background" world.[8]

2.2 The "Green IT" Optimization (Critical for O2 Switch Prize)

To win the **Green IT** challenge while having 3D, you must implement **On-Demand Rendering**.

- **The Trick:** Set <Canvas frameloop="demand">.
- **How it works:** The 3D scene **stops rendering** (0 FPS) when the user is not moving the camera. This drops GPU usage to near zero, saving the user's battery. Standard WebGL runs at 60FPS constantly, which kills "Green" scores.[10]

---

3. Visual Identity: The "NIRD" Aesthetic

Your design must reflect the actual NIRD resources to show the jury you did your homework.

- **OS:** The initiative uses **Linux NIRD** (based on Linux Mint Xfce). It uses the **Plank** dock (macOS style).[12]
- **Art Direction:** The visual identity is defined by the illustrator **Juliette Taka** (creator of the Debian "Emerald" theme).
- **Palette:** Use the **"Emerald"** palette: Deep Greens (#00997d), Slate Blues (#4a697d), and Soft Whites (#f4f4f9).[13]
- **Mascot:** The NIRD initiative often uses a **Pangolin** or similar animal in a resistance beret. Use this character as your "Guide" in the app.[15]

---

4. Product Requirement Document (PRD)

Project Title: Operation NIRD: The Sovereignty Protocol
Goal: Gamify the migration of a school from Windows to Linux.

4.1 Feature List & User Stories

Module 1: The Interactive Village (Home Page)

- **Tech:** React Three Fiber + ScrollControls.
- **Concept:** A low-poly 3D view of a school. As the user scrolls, the camera flies between the "Admin Block," "Classroom," and "Server Room."
- **Interaction:** The school starts "Grey and Polluted" (Windows). As the user scrolls and unlocks "Knowledge," trees pop up (Framer Motion spring animation) and the school turns Green (Linux).[16]

Module 2: "La Zerguèm" (Laser Game Challenge)

- **Partner:** Laser Game Evolution.
- **Concept:** A browser-based mini-game inside the 3D view.
- **Gameplay:** "Bugs" (representing trackers/bloatware) attack the server. The user controls the **NIRD Pangolin** holding a "Laser Blaster" (refurbishing tool).
- **Mechanic:** Click to shoot. Simple raycasting.

- **Win State:** "Server Cleaned. Linux Installed." Reward: A discount code for Laser Game Evolution.[17]

Module 3: "Chat'bruti" (Chatbot Challenge)

- **Partner:** Viveris.
- **Concept:** The "Bad Faith" Sales Bot.
- **Persona:** A "Clippy"-like assistant from the "Big Tech Empire" that tries to sell you useless licenses.
- **Gameplay:** The user must argue *against* the bot using NIRD arguments (e.g., "But Linux is free and respects privacy!").
- **Tech:** A simple decision tree disguised as a chat. If the user selects the right argument, the bot "glitches" and crashes.[19]

Module 4: The Hardware Clinic (Refurbishment)

- **Concept:** An interactive drag-and-drop game.
- **Action:** User drags a "USB Key (Linux NIRD)" onto an "Old Dusty Laptop."
- **Animation:** The laptop screen flashes, code scrolls (Matrix style), and it boots into the **Juliette Taka Emerald Desktop**.[12]

---

5. Detailed Implementation Plan (Timeline)

Phase 1: Setup (16h34 - 18h00)

- Initialize Next.js repo with TypeScript.
- Install framer-motion, @react-three/fiber, @react-three/drei.
- **Crucial:** Configure tailwind.config.js with the "Emerald" color palette immediately.

Phase 2: The Core (18h00 - 22h00)

- **Dev A (3D):** Build the low-poly school scene. *Constraint:* Keep poly count under 10k. Use .glb format compressed with gltf-pipeline.[20]
- **Dev B (UI):** Build the layout with Framer Motion page transitions. Implement the "Chat'bruti" logic.
- **Dev C (Content):** Write the "Resistance" narrative. Download NIRD assets (logos, Linux screenshots).

Phase 3: The Magic (22h00 - 02h00)

- **Integration:** Merge the 3D scene into the Home Page.
- **Optimization:** Enable <Canvas frameloop="demand">. Check EcoIndex score.
- **Gamification:** Hook up the "Laser Game" click events.

Phase 4: Polish (02h00 - 06h00)

- **Sound:** Add "satisfying" UI sounds (clicks, success chimes) using use-sound.
- **Easter Eggs:** Add a Konami code that triggers a "Linux Tux" parade.
- **Submission:** Generate the video demo (essential for the jury).

6. Technical Checklist for "Green IT" Jury

- [ ] **Server-Side Generation (SSG):** Use Next.js output: 'export' if possible for static hosting.
- [ ] **Lazy Loading:** Use next/dynamic to load the heavy 3D canvas only when visible.
- [ ] **Image Formats:** Convert all textures/screenshots to .webp or .avif.
- [ ] **Font Subsetting:** Use only woff2 with required glyphs.

This plan gives you a **professional, immersive** experience that hits every major challenge criteria while solving the technical paradox of "Green 3D".

Works cited

1. Démarche NIRD, accessed December 4, 2025, https://nird.forge.apps.education.fr/
2. La Nuit de l'Info 2025 - Sujet.pdf
3. The Future of Framer Motion in Web Development and UI Animation - Sparks by Codezela, accessed December 4, 2025, https://sparks.codezela.com/framer-motion-future/
4. Motion — JavaScript & React animation library, accessed December 4, 2025, https://motion.dev/
5. Web Animation for Your React App: Framer Motion vs GSAP - Semaphore CI, accessed December 4, 2025, https://semaphore.io/blog/react-framer-motion-gsap
6. Interactive UI Animations with GSAP & Framer Motion | by Toukir Ahamed Pigeon - Medium, accessed December 4, 2025, https://medium.com/@toukir.ahamed.pigeon/interactive-ui-animations-with-gsap-framer-motion-f2765ae8a051
7. Page Transitions with React Router and Framer Motion - YouTube, accessed December 4, 2025, https://www.youtube.com/watch?v=3QrkCmsfewM
8. Why I Prefer React Three Fiber Over Vanilla Three.js | by Harry Hao - Medium, accessed December 4, 2025, https://medium.com/@koler778/why-i-prefer-react-three-fiber-over-vanilla-three-js-28025cb324ff
9. React Three Fiber: Introduction, accessed December 4, 2025, https://r3f.docs.pmnd.rs/
10. Scaling performance - React Three Fiber, accessed December 4, 2025, https://r3f.docs.pmnd.rs/advanced/scaling-performance
11. React-Three-Fiber: Enhancing Scene Quality with Drei + Performance Tips - Medium, accessed December 4, 2025, https://medium.com/@ertugrulyaman99/react-three-fiber-enhancing-scene-qual

ity-with-drei-performance-tips-976ba3fba67a
12. Le choix Linux - Démarche NIRD, accessed December 4, 2025, https://nird.forge.apps.education.fr/linux/
13. DebianArt/Themes/Emerald - Debian Wiki, accessed December 4, 2025, https://wiki.debian.org/DebianArt/Themes/Emerald
14. MEET THE TEAM, accessed December 4, 2025, http://103.203.175.90:81/fdScript/RootOfEBooks/E%20Book%20collection%20-%202023%20-%20C/ENGG%20TECHNICAL%20MAGAZINES/Linux_Format_UK_Summer_2023.pdf
15. jop (@jop@mamot.fr) - Mamot - Le Mastodon de La Quadrature du Net, accessed December 4, 2025, https://mamot.fr/@jop
16. Shader transitions - Wawa Sensei, accessed December 4, 2025, https://wawasensei.dev/courses/react-three-fiber/lessons/shader-transitions
17. Défis 2025 - Nuit de l'informatique, accessed December 4, 2025, https://www.nuitdelinfo.com/inscription/defis/liste
18. lasergame-evolution - La Nuit de l'Info, accessed December 4, 2025, https://www.nuitdelinfo.com/inscription/defis/509
19. viveris - La Nuit de l'Info, accessed December 4, 2025, https://www.nuitdelinfo.com/inscription/defis/494
20. Building Efficient Three.js Scenes: Optimize Performance While Maintaining Quality, accessed December 4, 2025, https://tympanus.net/codrops/2025/02/11/building-efficient-three-js-scenes-optimize-performance-while-maintaining-quality/
21. Web 3D in 2025 Pixel Streaming vs WebGL Threejs Analysis - Rave Space, accessed December 4, 2025, https://ravespace.io/blog/pixel-streaming-vs-webgl-three-js-2025-s-scientific-verdict-on-3d-web-technologies