iOS Developer Portfolio – 3D/Game Blueprint (Vercel + R3F + Spline)

> Target stack: **Next.js (App Router)** + **Tailwind** + **Framer Motion** + **React Three Fiber/Drei** + **Spline embeds**; deploy on **Vercel**. Heavy assets on **Cloudflare R2**. Optional state/DB via **Supabase**. Dual mode: **Recruiter Mode** (fast, minimal) & **Game Mode** (3D experience).

---

1) Goals, Audience, and Success Metrics

- **Goals**: Showcase iOS projects and impact; wow with 3D/game UX without blocking quick evaluation.
- **Audience**: Recruiters, hiring managers, tech leads, clients.
- **Success Metrics**: Time-to-first-meaningful-info (<2s), CV downloads, contact submissions, recruiter dwell time, completion of “trophy” collection, return visits.

---

2) Information Architecture & Routes

- `/` Home: Hero + mode toggle; teaser to projects; social/CTA.
- `/projects` Grid + filters; supports tags (SwiftUI, CoreData, ARKit…).
- `/projects/[slug]` Case study with App Store-style gallery, metrics, tech stack, video demo, links.
- `/experience` Timeline (2D/2.5D) + achievements; switchable to **Game Timeline**.
- `/resume` Print CSS + PDF export; QR to contact; recruiter snapshot.
- `/contact` Form → email (Vercel Function / Resend) + Calendly embed.
- `/ar` Landing for AR/WebXR demos + QR codes to TestFlight/App Store.
- `/game` 3D world (R3F/Spline): office walk, portals to projects, trophy cabinet.
- `/privacy` and `/legal` (simple pages).

---

3) Dual Modes (Critical UX Feature)

- **Recruiter Mode**
- Minimal scripts; no 3D imports; static hero; instant LCP.
- Prominent CTAs: Resume, Projects, Contact.
- Accessible, print-ready resume.
- **Game Mode**
- Loads 3D modules lazily (dynamic import, `ssr:false`).
- 3D iPhone model on Home; portals to projects; trophy system.
- **Toggle Mechanics**
- Stored in `localStorage` + URL param `?mode=game|recruiter`.
- Server default to `recruiter`; client hydrates into game if chosen.

---

4) Visual Design System

- **Typography**: Inter/SF Pro Text (body), SF Pro Display (headings).
- **Color**: Dark-first palette; subtle neutrals; accent = iOS blue/purple gradient.
- **Components**: Card, Badge, Chip, CTA Button, iPhone Frame, Timeline Node, Achievement Trophy, Modal.
- **Motion**: Framer Motion for entrance and micro-interactions; GSAP for scroll/parallax on timeline; respect `prefers-reduced-motion`.
- **3D Aesthetic**: Low-poly to mid-poly; soft shadows; HDRI from Poly Haven; model compression (Draco + KTX2 textures).

---

5) Technical Stack & Key Dependencies

- **Next.js 15 App Router**, React 18.
- **Tailwind CSS** (dark/light support).
- **Framer Motion** (UI animations).
- **GSAP** (scroll-driven sections; optional).
- **React Three Fiber + Drei** (3D scenes), **three-stdlib**.
- **Spline** embeds for hero/inline 3D when faster than custom R3F.
- **Zustand** or **Context** for lightweight global state (mode, UI, trophies cache).
- **next-seo** for SEO; Open Graph image generation (Vercel OG).
- **Resend** (email) via Vercel Function.
- **Supabase** (optional): auth + DB for achievements persistence.
- **Cloudflare R2**: assets (GLB/GLTF, textures, videos) + public CDN URL.

---

6) Data Model (Content-first)

projects.json

```
[
  {
    "slug": "pulse-incident-ios",
    "title": "Pulse Incident Manager (iOS)",
    "tagline": "Offline-first incident response with SwiftUI & Core Data",
    "tags": ["Swift", "SwiftUI", "Combine", "Core Data", "WebSocket"],
    "impact": {"downloads": 52000, "crash_free": 99.8, "ttfb_improve": 0.45},
    "links": {
      "github": "https://github.com/you/pulse-ios",
      "testflight": "https://testflight.apple.com/join/xxxxx",
      "appstore": null
    },
    "media": {
      "images": ["https://r2.example.com/pulse/hero1.webp", "https://
r2.example.com/pulse/hero2.webp"],
      "video": "https://r2.example.com/pulse/demo.mp4",
      "model": "https://r2.example.com/models/iphone14.glb"
    },
    "summary": "Built real-time incident view with optimistic UI and offline cache.",
    "problem": "Ops teams missed critical updates during outages.",
    "solution": "Realtime WebSocket stream + background refresh with Core Data.",
    "role": "Lead iOS Engineer",
    "timeline": "2024-2025",
    "tech": ["Swift", "SwiftUI", "Combine", "Core Data", "URLSessionWebSocketTask", "XCTest"],
    "achievements": ["App Store ‘New Apps We Love’", "99.8% crash-free users"],
    "seo": {"og": "/og/pulse.png", "description": "SwiftUI offline-first incident app"}
  }
]
```

achievements.json

```
[
  {"id":"wwdc23", "title":"WWDC Attendee 2023", "icon":"/icons/wwdc.svg", "type":"award"},
  {"id":"streak100", "title":"100-Day GitHub Streak", "icon":"/icons/streak.svg", "type":"streak"}
]
```

Supabase (optional) tables: - `users(id, email)` - `trophies(user_id, trophy_id, acquired_at)`

---

7) 3D & Game Features

7.1 3D iPhone Hero (Home)

- Load a compressed GLB iPhone model from R2.
- Raycast hotspots on screen → open modals for Projects/Skills/Contact.
- Idle animation and orbit controls (limited angle). Lazy-load under `Suspense`.

7.2 Mini 3D World (`/game`)

- Room: desk (project posters), hallway (timeline), trophy cabinet.
- **Portals**: Drei `PortalMaterial` to teleport into project scenes.
- **Collectables**: floating badges → on click, store in local state/Supabase.
- Performance: use baked lighting, instancing for props, limit draw calls.

7.3 Timeline (2.5D)

- Scroll or keyboard to move checkpoint to checkpoint.
- Each node expands with video, impact metrics, GitHub/Store links.
- GSAP `ScrollTrigger` or Framer Motion scroll; fallback static for reduced motion.

7.4 Spline usage

- For quick hero or decorative 3D when custom R3F isn't needed.
- Embed with `<iframe loading="lazy" />`; hand off interactions via postMessage if needed.

---

8) Performance Budget & Accessibility

- **Budgets**: Home < 200KB JS in Recruiter Mode; 3D route code-split; images WebP/AVIF; videos streamed (HLS or MP4 with `preload=metadata`).
- **Models**: < 2–5 MB per GLB; textures KTX2; Draco compression.
- **Preload**: critical fonts, hero image; defer non-critical scripts.
- **A11y**: keyboard navigation, ARIA labels, skip-links, sufficient contrast; `prefers-reduced-motion` respected everywhere.

---

9) SEO & Social

- `next-seo` configuration per route; canonical URLs.
- OG images via Vercel OG; per-project dynamic OG.
- Structured data (JSON-LD): `Person`, `Project`, `BreadcrumbList`.

---

10) Security & Privacy

- Headers via Next middleware: CSP, X-Frame-Options (relax for Spline), Referrer-Policy, Permissions-Policy.
- Rate-limit contact API; captcha (hCaptcha/Turnstile) if spam.
- Content only; no tracking cookies unless analytics enabled.

---

11) CI/CD & Environments

- GitHub → Vercel auto deploy previews per PR.
- **Env Vars**
- `NEXT_PUBLIC_MODE_DEFAULT=recruiter`
- `ASSETS_BASE=https://<r2-public-endpoint>`
- `RESEND_API_KEY=...`
- `SUPABASE_URL=...` (optional)

- `SUPABASE_ANON_KEY=...` (optional)

12) Implementation Plan (Milestones)

Milestone 1 — Foundation (1–2 days) - Next.js app, Tailwind, base layout, dark theme, SEO, OG. - Content JSON files; Projects grid; Project detail page.

Milestone 2 — Recruiter Mode (1–2 days) - Fast home hero, resume page (print CSS + PDF export), contact form (Resend).

Milestone 3 — 3D Hero (1–2 days) - R3F iPhone model; hotspots; lazy-load; performance pass.

Milestone 4 — Timeline (1–2 days) - 2.5D timeline with parallax; achievements chips.

Milestone 5 — Game World (3–5 days) - `/game` room with portals, trophy cabinet; collectables & local state; optional Supabase persistence.

Milestone 6 — Polish & Deploy (1–2 days) - Asset optimization, analytics, legal, lighthouse ≥ 90 in Recruiter Mode.

13) Component Inventory (Key)

- `ModeToggle`, `ThemeSwitch`, `Navbar`, `Footer`.
- `HeroRecruiter`, `Hero3D`.
- `ProjectCard`, `ProjectGallery`, `MetricBadge`.
- `Timeline`, `TimelineNode`.
- `Trophy`, `TrophyShelf`.
- `IPhoneModel`, `Portal`, `Collectable` (3D components).
- `ContactForm` (server action/function).

14) Skeleton Code & Examples

14.1 Dynamic import of 3D hero

```
// app/(home)/Hero3D.tsx
'use client'
import { Canvas } from '@react-three/fiber'
import { OrbitControls } from '@react-three/drei'
import { Suspense } from 'react'
import IPhone from '@/components/3d/IPhone'
```

```
export default function Hero3D(){
  return (
    <div className="h-[60vh]">
      <Suspense fallback={<div className="animate-pulse">Loading 3D…</div>}>
        <Canvas shadows camera={{ position: [0, 1.2, 3], fov: 45 }}>
          <ambientLight intensity={0.6} />
          <directionalLight position={[3,4,2]} castShadow />
          <IPhone />
          <OrbitControls enablePan={false} maxPolarAngle={Math.PI/2} />
        </Canvas>
      </Suspense>
    </div>
  )
}
```

14.2 Loading a GLB model

```
// components/3d/IPhone.tsx
'use client'
import { useGLTF } from '@react-three/drei'
import { useMemo } from 'react'

export default function IPhone(){
  const { scene } = useGLTF(process.env.NEXT_PUBLIC_ASSETS_BASE + '/models/
iphone14.glb')
  const model = useMemo(()=>scene.clone(), [scene])
  return <primitive object={model} position={[0,-0.6,0]} />
}
```

14.3 Spline embed

```
export const SplineHero = () => (
  <iframe
    src="https://my.spline.design/your-scene"
    className="w-full h-[60vh] rounded-2xl"
    loading="lazy"
    allow="xr-spatial-tracking; fullscreen"/>
)
```

14.4 Mode toggle (store)

```
// lib/store.ts
import { create } from 'zustand'
export const useMode = create<{mode:'recruiter'|'game'; set:(m:any)=>void}
```

```
>(set=>({
  mode: 'recruiter',
  set: (mode) => set({ mode })
}))
```

15) Deployment Steps (Vercel + R2)

1. **Repo**: Push Next.js project to GitHub.
2. **Vercel**: Import repo → set env vars (`NEXT_PUBLIC_ASSETS_BASE`, `RESEND_API_KEY`, optional Supabase).
3. **Cloudflare R2**: Create bucket → upload `/models`, `/videos`, `/images`; enable public access (or presigned URLs).
4. Replace asset URLs in JSON/config.
5. Test Preview Deploy → run Lighthouse → fix budgets.
6. Connect custom domain in Vercel; configure DNS; force HTTPS.

16) Testing & QA

- **Unit/Integration**: vitest/testing-library for components; basic store tests.
- **E2E**: Playwright → navigation, contact form, mode toggle, 3D load fallback.
- **Performance**: Lighthouse (mobile) ≥ 90 Recruiter Mode, ≥ 70 Game Mode; WebPageTest for filmstrip.
- **Accessibility**: axe-core, keyboard traps, focus order, labels.

17) Analytics & Monitoring

- Vercel Analytics (or Plausible).
- Track: mode toggle usage, project detail views, contact submissions, trophy collected events.
- Error monitoring (Sentry) for JS errors in 3D scenes.

18) Risk Register & Mitigations

- **3D too heavy** → strict budgets, lazy-load, Spline for light scenes.
- **SEO hit from JS** → Recruiter Mode is static-first, SSR content visible.
- **Spam on contact** → rate limit + captcha.
- **AR on iOS quirks** → provide video fallback and QR to TestFlight.

19) Budget & Ongoing Cost (typical)

- Vercel Free + R2 assets + domain ≈ **$1–$2/mo** (domain annually).
- Optional Supabase Pro/Analytics later if needed.

---

20) Acceptance Criteria (Go-Live)

- Recruiter can get resume and top 3 projects in **<10 seconds** from landing.
- Lighthouse mobile (Recruiter Mode) ≥ **90** across PWA/Perf/SEO/A11y.
- At least one **3D hero** interaction and one **Game World** room with 2 portals.
- Contact form delivers email successfully; QR codes resolve to working builds.

---

21) Content Checklist (You Provide)

- Headshot avatar, short bio.
- 3–6 curated projects: summaries, metrics, links, images/videos, repos.
- Achievements list (WWDC, hackathons, features).
- Contact email + preferred scheduler.
- Domain name to use.

---

22) Next Steps

1) Confirm content checklist items. 2) I'll generate a starter repo skeleton (pages, components, basic 3D hero, Spline embed, timeline stub, contact API) ready for Vercel. 3) Upload/replace assets in R2 and wire JSON content.