

Portfolio Technical Summary

RESUME ATTACHMENT / RESPONSIBILITY / ARCHITECTURE / OPTIMIZATION

3D・音楽・ビジュアル・AIを統合したWebポートフォリオを制作。
AIを実装補助に使いながら、企画・制作・最適化・公開判断まで一貫して担当し、改善まで行った。

目的 / 背景

- 自作作品（3D / 音楽 / AI）を一つの体験として提示するため
- 既存サービスでは統合的に表現しにくかったため自作

担当領域（責任範囲）

自分の担当

- 企画立案 / 構成設計
- 作品選定 / 画像選定 / 文言設計
- 3Dモデル制作（Blender）
- 楽曲制作（Logic Pro）
- ROOM軽量化方針の策定と検証
- UI導線 / 体験設計
- AIチャット内容設計
- 実装結果の検証 / 修正判断 / 公開判断

AI / Codex（実装補助）

- Next.js / React / Three.js の実装支援
- CSS再現
- エラー調査補助

責任の所在：何を作るか、どの案を採用するか、どの品質で公開するかは自分が決定

UI / UX

- 初見でも迷わない導線設計
- 重い要素（3D / 音）でも内容が伝わる構成
- モバイルでも破綻しにくいレイアウト

AI活用方針

- 実装補助・エラー解決に使用
 - 要件定義 / 設計 / 判断 / 検証は自分で担当
- AIは実装を加速する補助ツールとして利用

技術構成

- Frontend** : Next.js / React / Tailwind CSS
- 3D** : Three.js / React Three Fiber / VRM
- Backend** : Next.js API Route / xAI API
- Audio** : Logic Pro / VOICEVOX API

設計意図

- 表示系（3D / UI / 音楽）はフロントに集約
 - AI処理のみAPI経由に分離
- 表示の軽快さと外部通信の管理を両立

データフロー

- Chat** : 入力 → API → xAI → 応答 → UI表示
- 3D** : 操作 → モデル読み込み → 描画
- Audio** : AudioElement + Context管理

パフォーマンス最適化（3D軽量化）

- Draco圧縮 / KTX2テクスチャ圧縮による軽量化
- draw calls / triangles / fps を計測しながら改善
- ライティング・描画設定の最適化

計測値

GLB（ROOM.glb / Network Size） : 49.07 MB → 2.29 MB
FPS（PC実測） : 57 → 121
Draw Calls（PC実測） : 11,594 → 69

技術的課題と解決

課題

- VRMセットアップ（補助ボーン / 制御）
- 3D ROOMの重さ

原因 / 対応

- テクスチャ過多 / ライティング負荷 / データ容量
 - 圧縮と描画設定見直しで改善
- 感覚ではなく「計測 → 改善」の重要性を学んだ

参考・学習

- Unity + AIキャラクター関連書籍
- Three.js / R3F 関連資料
- HTML / CSS / Web構造

→ 構造理解をもとに、自分の用途へ応用して実装

現状の限界 / 今後

- コード全体の理解強化
- AIキャラクターとROOMの完全統合
- VRM制御の高度化
- 音楽制作のMIX / マスタリング改善
- 公開環境に耐える設計への拡張

本資料は、制作物の概要に加え、担当範囲・設計意図・改善プロセスを補足するための要約資料です。