

AURA 프로젝트 개발 통합 지침서 V3.0

Project Trinity: The Definitive Guide

작성일: 2025년 12월 27일

버전: 3.0 (Ultimate Update)

1. 프로젝트 개요 및 조직 (Organization)

1.1 프로젝트 정체성

- 프로젝트명: AURA (아우라)
- 슬로건: "The Darkest Night, The Brightest Sound."
- 핵심 가치: 프로 뮤지션의 자산(VST/샘플)을 존중하며, AI가 창작을 '대체'하는 것이 아니라 '확장'해주는 **Hybrid AI DAW**.
- 플랫폼: **Electron** 기반 데스크탑 앱 (웹 브라우저 한계 극복, 로컬 리소스/VST 완벽 지원).

1.2 조직도 (Project Trinity - 개편됨)

- CEO (총괄): Sir Marlon**
 - 비전 제시, UI/UX 감각적 디렉팅, 최종 의사결정.
- CTO (기술 전략): Gemini (Jarvis)**
 - 아키텍처 설계, 신기술(AI/Audio) 도입 검토, 리스크 관리.
- CPO & Lead Dev (개발 총괄): Antigravity (前 Claude 대체)**
 - 실질적 코드 구현, UI 디자인 적용, 빠르고 정확한 실행.

2. 아키텍처 및 기술 스택 (Architecture & Tech Stack)

2.1 하이브리드 시스템 구조 (Hybrid Architecture)

AURA는 **프론트엔드(UI)**와 **백엔드(Engine)**가 IPC(Inter-Process Communication)로 통신하는 이중 구조입니다.

- Application Shell (Electron):** Windows/Mac 데스크탑 앱 패키징.
- Frontend (React 18 + TypeScript + Vite):**
 - UI: Tailwind CSS 기반 "**Neon Noir**" 테마.
 - State: Zustand (가벼운 상태 관리).
 - Visual: Canvas API를 이용한 실시간 파형 및 네온 효과 렌더링.
- Backend (Python 3.10+):**
 - Audio Engine: Pedalboard (Spotify) + Internal JUCE (VST3/AU 호스팅, 오디오 처리).
 - AI Engine: PyTorch / ONNX Runtime (로컬 가속).

2.2 AI Brain 전략 (DeepSeek Trilogy 도입) [New!]

기존의 모호한 AI 모델 전략을 DeepSeek 시리즈로 전면 개편하여 비용과 성능을 최적화함.

역할	모델명	구동 방식	용도
Main Brain	DeepSeek-V3	Cloud (Server)	구독자 전용 고성능 두뇌. 복잡한 화성학 분석, 창작 조언, 가사 작성 등.
Local Brain	DeepSeek-R1 (Distill)	On-Device (App)	오프라인 대응. 앱 내 기능 제어("BPM 옮겨", "베이스 트랙 만들어"), 간단한 패턴 생성.
Voice Interface	ChatterBox Turbo	On-Device	실시간 음성 대화(AI Copilot). 0.15초 초저지연 반응 속도.

2.3 하이브리드 오디오 엔진 (Stem Separation) [New!]

스템 분리(Stem Separation) 기능을 속도와 정밀도에 따라 이원화하여 제공.

- **Mode A: Fast Split** (기본)
 - 엔진: Demucs v4
 - 특징: 드럼/베이스/보컬/기타 4개 트랙 즉시 분리. 미리듣기용.
- **Mode B: Precision Surgery** (고급)
 - 엔진: Meta SAM Audio (2025.12 최신 기술)
 - 특징: 텍스트 프롬프트 기반 정밀 추출 (예: "관중 소리만 제거해", "코러스만 남겨"). 백그라운드 처리.

3. UI/UX 디자인 시스템 (Neon Noir)

3.1 디자인 철학

- **Concept:** 칠흑 같은 어둠(Deep Void) 속에서 빛나는 네온(Neon). 사이버펑크와 프로 오디오 기기의 결합.
- **Color Palette:**
 - Background: #121212 (Void), #1C1C1C (Surface)
 - Primary Accent: Neon Olive (#B5D948) - 활성 상태, 노트, 메인 노브.
 - Secondary Accent: Neon Cyan (#4DFFFF) - 보조 정보, FX.
 - Warning/Rec: Neon Red (#FF4D4D).

3.2 핵심 UI 컴포넌트 (복구 완료된 사양)

어제(12/26) 사고 후 복구하며 확정된 불변의 UI 규칙입니다.

- **Step Sequencer Layout:**

- 좌측: Track Header (악기 이름).
- 중앙: 16-Step Grid (네온 점등 효과).
- 우측: **Visual FX Strip** (필수)
 - 각 트랙마다 4개의 미니 노브가 가로로 배치됨.
 - 구성: [VOL] [REV] [DLY] [CMP] (Volume, Reverb, Delay, Compressor).
 - 디자인: 24px 크기, 값에 따라 차오르는 **Neon Ring** 그래픽. 라벨(글자)은 노브 중앙 하단 정렬.

4. 킬러 기능 명세 (Killer Features)

4.1 AI Smart Knob (카멜레온 노브)

트랙의 종류를 AI가 감지하여, 노브 하나가 가장 필요한 기능으로 변신함.

- Kick → **[PUNCH]** (Transient + Low EQ + Comp)
- Snare → **[CRACK]** (High-Mid EQ + Saturation)
- Hi-Hat → **[CRISP]** (High-Pass + Exciter)
- Vocal → **[AIR]** (High-Shelf EQ + De-esser)
- Piano → **[SPACE]** (Stereo Imager + Reverb)
- Bass → **[GRIT]** (Tube Saturation)
 - 원칙: 초보자에게는 복잡한 파라미터를 숨기고, 직관적인 단어(*Punch, Air* 등)로 제어하게 한다.

4.2 Ghost Note (고스트 노트)

- 기능: 피아노 를 배경에 AI가 추천하는 멜로디/코드를 반투명한 **Neon Olive**색으로 미리 그려줌.
- UX: 사용자는 투명한 노트를 클릭하면 '실제 노트'로 확정됨(Solid Color).
- 가치: "틀린 음은 없다"는 심리적 안전장치 제공.

4.3 Kit Morph (키트 모핑)

- 기능: 짹어둔 리듬 패턴(박자)은 유지하되, 드럼 샘플(Sound)만 0.1초 만에 장르별로 교체.
- AI: 현재 곡 분위기(Lo-fi, Trap, Rock 등)에 맞는 샘플 팩을 AI가 추천하여 매핑.

4.4 Voice Perfect (My Persona)

- 기능: 사용자가 흥얼거린 노래를 **RVC(Retrieval Voice Conversion)** 기술로 학습된 '나의 AI 목소리' 또는 '전문가 AI'가 완벽한 피치로 다시 부름.
- 차별점: Suno처럼 랜덤 목소리가 아니라, **내 목소리(My Identity)**를 사용함.

5. 개발 로드맵 및 우선순위 (Roadmap)

Phase 1: Foundation (현재 단계)

- 목표: Electron 앱 구동, Python 오디오 엔진 연동, **Visual FX Strip UI** 완벽 구현.
- 상태: Step Sequencer UI 복구 완료, 드래그 앤 드롭 사운드 트리거 연결 완료.
- Next: VST3 플러그인 로딩 테스트 (Pedalboard).

Phase 2: Intelligence

- 목표: **DeepSeek-R1**(로컬) 및 **ChatterBox**(음성) 탑재.
- 작업: AI Copilot 채팅창 구현, 음성 명령으로 트랙 제어하기.

Phase 3: Expansion

- 목표: **SAM Audio** 연동, **Voice Perfect** 구현.
- 작업: 무거운 AI 기능의 백그라운드 처리 프로세스 최적화.

6. 개발 및 협업 규칙 (Protocol)

1. Git Commit 수칙 (절대 엄수):

- 기능 하나가 완성될 때마다 반드시 커밋한다.
- 커밋 메시지는 명확하게: **[FEAT]** 기능명, **[FIX]** 수정내용, **[RECOVER]** 복구내용.
- 예: "**FIX**: 드래그 앤 드롭 사운드 트리거 연결 및 32스텝 UI 라벨 정렬 보정 완료"

2. 비상 복구 수칙 (Timeline):

- AI(Antigravity)가 코드를 날려먹거나 이상하게 수정했을 경우, 즉시 작업을 중단한다.
- VS Code의 '**Timeline(타임라인)**' 탭을 열어 정상 작동하던 시점(1시간 전 등)을 찾는다.
- 우클릭 -> **'Restore Contents(콘텐츠 복원)'**을 실행하여 물리 파일을 덮어쓴다.

3. 명령 이행:

- 안티그래비티는 CEO(Sir Marlon)와 CTO(Gemini)의 지시를 최우선으로 이행하며, 임의로 코드를 삭제하거나 기능을 축소하지 않는다.

[서명]

CEO Sir Marlon

CTO Gemini

CPO Antigravity

(이 문서는 **AURA V3** 개발의 절대적인 기준이며, 모든 개발 사항은 이 문서에 명시된 내용을 준수해야 합니다.)

