Documentación del Proyecto de Piano por Síntesis Modal

Fecha de generación: 2025-07-04 20:50:50

Este documento contiene una transcripción técnica y consolidación del desarrollo de un motor de piano basado en síntesis modal, incluyendo modelación de cuerdas, resonancias, soundboard, y ejecución MIDI en tiempo real.

# 1. Objetivo General

Desarrollar un motor de síntesis modal de piano, modelando cuerdas, resonancia simpática, tabla armónica, resonancia cruzada soundboard→cuerdas, y polifonía en tiempo real usando entrada MIDI.

# 2. Módulos creados

- realistic\_modal\_synthesis.py: modela cuerdas con inarmonía empírica.  
- sympathetic\_resonance.py: simula cuerdas libres que vibran por armónicos cercanos.  
- soundboard\_model.py: modela modos de la tabla armónica y su respuesta.  
- soundboard\_sympathetic.py: simula cuerdas libres activadas por la tabla.  
- midi\_interface.py: conexión en tiempo real con teclados MIDI.  
- piano\_engine.py: integración de todo lo anterior en modo offline polifónico.  
- realtime\_engine.py: arquitectura real-time con voces independientes por nota (NoteVoice + AudioEngine).

# 3. Avance Real-Time

Se construyó una arquitectura basada en `NoteVoice` (una voz por nota con buffer independiente) y `AudioEngine` (mezcla todas las voces en un `callback` de `sounddevice`). Esto permite ejecución polifónica sin recalcular toda la textura sonora en cada evento MIDI.

# 4. Integración MIDI

Con `midi\_interface.py` se recibe en tiempo real desde un teclado. Se ejecuta el motor de síntesis desde consola, y responde en milisegundos por cada evento de nota o pedal. Ejemplo:  
  
```  
from piano\_engine import PianoEngine  
from midi\_interface import MIDIEngineInterface  
engine = PianoEngine()  
midi = MIDIEngineInterface(engine)  
midi.start()  
```

# 5. Archivos exportados

- midi\_interface.py  
- piano\_engine.py  
- realtime\_engine.py  
- soundboard\_sympathetic.py  
Todos preparados para importarse desde un proyecto Python modular en carpeta `core/`.

# 6. Próximos pasos

- Agregar `SympatheticEngine` en tiempo real.  
- Integrar `SoundboardEngine` con respuestas tabla→cuerdas.  
- Control de sustain más fino (por nota, liberación condicional).  
- Optimizar uso de CPU para ejecución sostenida continua.