Calamus Class Diagram v0.3

This document defines the core classes organized by the three-pillar architecture plus infrastructure.

Data Layer Classes

The contract between Sound Engine and Input Engine. All persistent data structures.

Project

Project
name: string tempo: float timeSignature: TimeSignature scale: Scale tracks: Track[] phrases: Phrase[] sounits: Sounit[] limits: Limits preferences: Preferences
save(path) load(path) export(options): AudioBuffer markDirty(phrase)
Root container for entire composition

Track

Track
name: string sounit: Sounit phrases: PhraseRef[] volume: float pan: float mute: bool color: Color
addPhrase(phrase, position) removePhrase(phraseRef)
1:1 with Sounit. Each track owns its sounit instance.

Phrase

Phrase
id: UUID notes: Note[] startTime: float duration: float isDirty: bool renderedBuffer: AudioBuffer?
addNote(note) removeNote(note) markDirty() getNotesInRange(start, end)

Unit of pre-rendering. Also used as test material in Sound Design.

Note

Note
id: UUID startTime: float duration: float pitch: Curve dynamics: Curve parameterCurves: Map<ParamId, Curve> visualShape: ShapeType
getPitchAt(time): float getDynamicsAt(time): float getParamAt(paramId, time): float
Compositional atom. What you see on canvas. Contains curves for all parameters.

Curve (abstract)

Curve
duration: float valueRange: Range
evaluate(t): float derivative(t): float integral(t0, t1): float
Base class for all curve types

HandDrawnCurve : Curve

HandDrawnCurve
points: CurvePoint[] thickness: float[] color: Color transparency: float[]
addPoint(point) simplify(tolerance) smooth(amount)
Captured from pen gesture. Thickness from pressure, transparency from speed.

MathCurve : Curve

MathCurve
function: MathFunction parameters: float[]
setParameter(index, value)
Mathematical function (sine, bezier, etc.)

EasingCurve : Curve

EasingCurve
easingFunction: EasingFunction startValue: float endValue: float mode: EasingMode
setEasing(function)

setRange(start, end)
Transition shaping (elastic, bounce, cubic, etc.)

EnvelopeCurve : Curve

EnvelopeCurve
segments: CurveSegment[] loopMode: LoopMode
addSegment(segment) removeSegment(index)
Arbitrary shape over note lifetime

CurvePoint

CurvePoint
time: float value: float pressure: float tiltX: float tiltY: float rotation: float resolution: float
Single point in hand-drawn curve with all pen dimensions

Scale

Scale
name: string degrees: ScaleDegree[] rootFrequency: float tuningSystem: TuningSystem
frequencyForDegree(degree, octave): float degreeForFrequency(freq): ScaleDegree
Defines tuning. Lines on staff = scale degrees.

ScaleDegree

ScaleDegree
index: int ratio: float cents: float harmonicFunction: HarmonicFunction color: Color
Single degree with ratio/cents and visual properties

Sounit

Sounit
id: UUID name: string color: Color containers: Container[] connections: Connection[] registerRange: Range

addContainer(container) connect(output, input, function, weight) disconnect(connection) render(note): AudioBuffer
Sound unit definition. Color = note blob color on canvas.

Connection

Connection
sourceContainer: Container sourcePort: Port destContainer: Container destPort: Port function: ConnectionFunction weight: float
Links container output to input with blend function

Sound Engine Classes

Audio generation, containers, physics, runtime state.

Container (abstract)

Container
id: UUID inputs: Port[] outputs: Port[] config: ContainerConfig
process(numSamples) reset() getInput(name): Port getOutput(name): Port
Base class. Subclasses implement specific DSP.

Container Subclasses

Essential:

- HarmonicGenerator — generates spectrum from pitch + DNA
- SpectrumToSignal — converts spectrum to audio signal

Shaping:

- RolloffProcessor — brightness/spectral slope
- FormantBody — resonant filtering (F1, F2)
- BreathTurbulence — voice/noise blend
- NoiseColorFilter — shapes noise spectrum

Modifiers:

- PhysicsSystem — mass/spring/damping
- EasingApplicator — shaped transitions
- EnvelopeEngine — parameter evolution
- DriftEngine — micro-detuning
- GateProcessor — note lifecycle

See Container Port Specification v1.1 for complete port definitions.

Port

Port
name: string dataType: DataType direction: Direction defaultValue: float currentValue: varies
read(): varies write(value) isConnected(): bool
DataType: Signal, Spectrum, Control, Trigger

Voice

Voice
noteRef: Note state: VoiceState age: int physicsState: PhysicsState[] containerStates: Map<Container, State>
trigger(note) release() process(numSamples): AudioBuffer isActive(): bool
Runtime instance. State: off/attack/sustain/release. Age for voice stealing.

SounitInstance

SounitInstance
sounit: Sounit voices: Voice[] maxPolyphony: int
allocateVoice(): Voice releaseVoice(voice) process(numSamples): AudioBuffer
Runtime instantiation of Sounit with voice pool

AudioEngine

AudioEngine
sampleRate: int bufferSize: int sounitInstances: SounitInstance[] renderedBuffers: Map<Phrase, AudioBuffer> commandQueue: LockFreeQueue statusQueue: LockFreeQueue
start() stop() processBlock(output, numSamples) queueCommand(cmd) setLivePhrase(phrase)
Runs on audio thread. Mixes pre-rendered + live.

RenderEngine

RenderEngine

dirtyQueue: Queue<Phrase> renderThread: Thread outputQueue: LockFreeQueue<RenderedPhrase>
start() stop() queuePhrase(phrase) renderPhrase(phrase): AudioBuffer
Background thread for pre-rendering dirty phrases

Input Engine Classes

Pen input, staff display, editing, transport.

WacomInput

WacomInput
tabletId: string currentPressure: float currentTiltX: float currentTiltY: float currentRotation: float currentPosition: Point isDown: bool
poll(): InputState onPenDown(callback) onPenMove(callback) onPenUp(callback)
Six-dimensional continuous input

StaffCanvas

StaffCanvas
scale: Scale visibleRange: TimeRange zoom: float selectedNotes: Note[] activeSounit: Sounit ghostedSounits: Sounit[]
draw() screenToTime(x): float screenToPitch(y): float hitTest(point): Note? setZoom(level) scroll(delta)
Scale-degree lines, note blobs, selection

SounitSelector

SounitSelector
sounits: Sounit[] activeSounit: Sounit visibleSounits: Sounit[]
select(sounit) toggleVisibility(sounit) draw()
Left panel colored bars for sounit selection

Transport

Transport
playbackState: PlaybackState tempo: float
play() stop() rewind() forward() setPosition(time) toggleLoop()
Playback controls

PlaybackState

PlaybackState
nowPosition: float loopEnd: float? isLoopMode: bool isPlaying: bool currentPosition: float
enterLoopMode() exitLoopMode() setLoopEnd(time)
Now marker, loop region, playback position

NoteEditor

NoteEditor
selectedNotes: Note[] clipboard: Note[] undoStack: Command[] redoStack: Command[]
select(notes) move(delta) copy() paste() delete() split(time) stretch(factor) invert() mirror() undo() redo()
Editing operations on selected notes

GestureRecorder

GestureRecorder
isRecording: bool currentCurve: HandDrawnCurve captureMode: CaptureMode pitchMode: PitchMode
startRecording() stopRecording(): Note addPoint(inputState)
CaptureMode: drawing/realtime. PitchMode: discrete/continuous.

Exporter

Exporter
options: ExportOptions
exportMix(project, path) exportStems(project, directory) exportSelection(notes, path)
WAV export functionality

ExportOptions

ExportOptions
sampleRate: int bitDepth: int channels: int normalize: bool includeMetadata: bool
44100/48000/96000, 16/24/32 bit, mono/stereo

Infrastructure Classes

Preferences, limits, threading, libraries.

Limits

Limits
harmonics: Range f1Freq: Range f2Freq: Range rolloff: Range mass: Range driftRate: Range polyphony: Range attackTime: Range releaseTime: Range ...
get(paramName): Range set(paramName, range) reset(paramName) resetAll()
No hard-coded limits. All ranges configurable. Can be global or per-project.

Preferences

Preferences
limits: Limits audioDevice: string bufferSize: int defaultScale: Scale defaultTempo: float uiScale: float colorScheme: ColorScheme
save() load() reset()

Global application preferences

LockFreeQueue<T>

LockFreeQueue<T>
buffer: T[] readIndex: atomic<int> writeIndex: atomic<int> capacity: int
push(item): bool pop(): T? isEmpty(): bool isFull(): bool
SPSC queue for thread communication. Audio-safe.

AudioCommand

AudioCommand
type: CommandType payload: varies
Types: Play, Stop, SetPosition, SetLivePhrase, UpdateBuffer, SetVolume, SetPan, etc.

EasingLibrary

EasingLibrary
functions: Map<string, EasingFunction>
get(name): EasingFunction register(name, function) list(): string[]
Built-in: linear, quadratic, cubic, sine, elastic, bounce, back, spring, wobble

MathFunctionLibrary

MathFunctionLibrary
functions: Map<string, MathFunction>
get(name): MathFunction register(name, function) list(): string[]
Sine, triangle, sawtooth, bezier, polynomial, etc.

EnvelopeLibrary

EnvelopeLibrary
envelopes: Map<string, EnvelopeCurve>
get(name): EnvelopeCurve save(name, envelope) delete(name) list(): string[]
User-defined and preset envelope shapes

Enumerations

DataType: Signal, Spectrum, Control, Trigger

Direction: Input, Output

ConnectionFunction: Passthrough, Add, Multiply, Subtract, Replace, Modulate

VoiceState: Off, Attack, Sustain, Release

CaptureMode: Drawing (X=time), Recording (time scrolls)

PitchMode: Discrete (snap to scale), Continuous (literal pitch)

LoopMode: None, Loop, PingPong

EasingMode: In, Out, InOut

HarmonicFunction: Tonic, Fifth, Third, Other (for scale degree coloring)

CommandType: Play, Stop, Rewind, SetPosition, SetLivePhrase, UpdateBuffer, SetVolume, SetPan, SetMute

Key Relationships

- Project contains Tracks, Phrases, Sounits, Limits, Preferences
- Track owns 1 Sounit (1:1 relationship, independent copy)
- Track references multiple Phrases
- Phrase contains multiple Notes
- Note contains multiple Curves (pitch, dynamics, parameters)
- Sounit contains Containers connected by Connections
- SounitInstance manages multiple Voices at runtime
- Voice references a Note and has physics state
- AudioEngine receives commands via LockFreeQueue
- RenderEngine outputs rendered buffers via LockFreeQueue
- Limits can be global (in Preferences) or per-project (in Project)

Version History

v0.1 — Initial class definitions

v0.2 — Added PlaybackState, SounitSelector, Exporter, MathFunctionLibrary, SpectrumToSignal; clarified Voice.age and Voice.noteRef

v0.3 — Added Limits, Preferences, LockFreeQueue, AudioCommand, RenderEngine; documented threading model classes

December 2025