Conchordal Technical note (draft)

Koichi Takahashi

1. Introduction: The Bio-Acoustic Paradigm

Conchordal represents a fundamental divergence from established norms in generative music and computational audio. Where traditional systems rely on symbolic manipulation—operating on grids of quantized pitch (MIDI, Equal Temperament) and discretized time (BPM, measures)—Conchordal functions as a continuous, biologically grounded simulation of auditory perception. It posits that musical structure is not an artifact of abstract composition but an emergent property of acoustic survival.

This technical note serves as an exhaustive reference for the system's architecture, signal processing algorithms, and artificial life strategies. It details how Conchordal synthesizes the principles of psychoacoustics—specifically critical band theory, virtual pitch perception, and neural entrainment—with the dynamics of an autonomous ecosystem. In this environment, sound is treated as a living organism, an "Individual" possessing metabolism, sensory processing capabilities, and the autonomy to navigate a hostile spectral terrain.

The emergent behavior of the system is driven by a unified fitness function: the pursuit of Consonance. Agents within the Conchordal ecosystem do not follow a pre-written score. Instead, they continuously analyze their environment to maximize their "Spectral Comfort"—defined as the minimization of sensory roughness—and their "Harmonic Stability," or the maximization of virtual root strength. The result is a self-organizing soundscape where harmony, rhythm, and timbre evolve organically through the interactions of physical laws rather than deterministic sequencing.

This document explores the three foundational pillars of the Conchordal architecture:

1. **The Psychoacoustic Coordinate System**: The mathematical framework of `Log2Space` and ERB scales that replaces linear Hertz and integer MIDI notes.
2. **The Cognitive Landscape**: The real-time DSP pipeline that computes Roughness (R) and Harmonicity (H) fields from the raw audio stream.
3. **The Life Engine**: The agent-based model governing the metabolism, movement, and neural entrainment of the audio entities.

2. The Psychoacoustic Coordinate System

A critical innovation in Conchordal is the rejection of the linear frequency scale (f) for internal processing. Human auditory perception is inherently logarithmic; our perception of pitch interval is based on frequency ratios rather than differences. To model this accurately and efficiently, Conchordal establishes a custom coordinate system, `Log2Space`, which aligns the computational grid with the tonotopic map of the cochlea.

2.1 The Log2 Space Foundation

The `Log2Space` struct serves as the backbone for all spectral analysis, kernel convolution, and agent positioning within the system. It maps the physical frequency domain (f in Hz) to a perceptual logarithmic domain (l).

2.1.1 Mathematical Definition

The transformation from Hertz to the internal log-coordinate is defined as the base-2 logarithm of the frequency. This choice is deliberate: in base-2, an increment of 1.0 corresponds exactly to an octave, the most fundamental interval in pitch perception.

$$l(f) = \log_2(f)$$

The inverse transformation, used to derive synthesis parameters for the audio thread, is:

$$f(l) = 2^l$$

The coordinate space is discretized into a grid defined by a resolution parameter, `bins_per_oct` (B). This parameter determines the granularity of the simulation. A typical value of $B = 48$ or $B = 96$ provides sub-semitone resolution sufficient for continuous pitch gliding and microtonal inflection. The step size Δl is constant across the entire spectral range:

$$\Delta l = \frac{1}{B}$$

2.1.2 Grid Construction and Indexing

The `Log2Space` structure pre-calculates the center frequencies for all bins spanning the configured range $[f_{min}, f_{max}]$. The number of bins N is determined to ensure complete coverage:

$$N = \lfloor \frac{\log_2(f_{max}) - \log_2(f_{min})}{\Delta l} \rfloor + 1$$

The system maintains two parallel vectors for $O(1)$ access during DSP operations:

- `centers_log2`: The logarithmic coordinates $l_i = \log_2(f_{min}) + i \cdot \Delta l$.
- `centers_hz`: The pre-computed linear frequencies $f_i = 2^{l_i}$.

This pre-computation is vital for real-time performance, removing the need for costly `log2` and `pow` calls inside the inner loops of the spectral kernels. The method `index_of_freq(hz)` provides the quantization logic, mapping an arbitrary float frequency to the nearest bin index.

2.2 Constant-Q Bandwidth Characteristics

The `Log2Space` inherently enforces a Constant-Q (Constant Quality Factor) characteristic across the spectrum. In signal processing terms, Q is defined as the ratio of the center frequency to the bandwidth: $Q = f/\Delta f$.

In a linear system (like a standard FFT), Δf is constant, meaning Q increases with frequency. In `Log2Space`, the bandwidth Δf_i of the i-th bin scales proportionally with the center frequency f_i. This property mimics the frequency selectivity of the human auditory system, where the ear's ability to resolve frequencies diminishes (in absolute Hz terms) as frequency increases. This alignment allows Conchordal to allocate computational resources efficiently—using high temporal resolution at high frequencies and high spectral resolution at low frequencies—without manual multirate processing.

2.3 The Equivalent Rectangular Bandwidth (ERB) Scale

While `Log2Space` handles pitch relationships (octaves, harmonics), it does not perfectly model the critical bands of the ear, which are wider at low frequencies than a pure logarithmic mapping suggests. To accurately calculate sensory roughness (dissonance), Conchordal implements the Equivalent Rectangular Bandwidth (ERB) scale based on the Glasberg & Moore (1990) model.

The `core/erb.rs` module provides the transformation functions used by the Roughness Kernel. The conversion from frequency f (Hz) to ERB-rate units E is given by:

$$E(f) = 21.4 \log_{10}(0.00437 f + 1)$$

The bandwidth of a critical band at frequency f is:

$$BW_{ERB}(f) = 24.7(0.00437f + 1)$$

This scale is distinct from `Log2Space`. While `Log2Space` is the domain for pitch and harmonicity (where relationships are octave-invariant), the roughness calculation requires mapping spectral energy into the ERB domain to evaluate interference. The system effectively maintains a dual-view of the spectrum: one strictly logarithmic for harmonic templates, and one psychoacoustic for dissonance evaluation.

3. The Auditory Landscape: Analyzing the Environment

The "Landscape" is the central data structure in Conchordal. It acts as the shared environment for all agents, a dynamic scalar field representing the psychoacoustic "potential" of every frequency bin. Agents do not interact directly with each other; they interact with the Landscape, which aggregates the spectral energy of the entire population. This decouples the complexity of the simulation from the number of agents ($O(N)$ vs $O(N^2)$).

The Landscape is updated every audio frame (or block) by the Analysis Worker. It synthesizes three primary metrics:

- **Roughness (R)**: The sensory dissonance caused by rapid beating between proximal partials.
- **Harmonicity (H)**: The measure of virtual pitch strength and spectral periodicity.
- **Habituation (Ψ)**: A negative feedback loop representing auditory fatigue.

The net Consonance (C) at frequency f and time t is the fitness gradient the agents climb:

$$C(f,t) = H(f,t) - k_r \cdot R(f,t) - w_h \cdot \Psi(f,t)$$

3.1 Non-Stationary Gabor Transform (NSGT)

To populate the `Log2Space` with spectral data, Conchordal uses a custom implementation of the Non-Stationary Gabor Transform (NSGT). Unlike the Short-Time Fourier Transform (STFT), which uses a fixed window size, the NSGT varies the window length L inversely with frequency to maintain the Constant-Q property derived in Section 2.2.

3.1.1 Kernel-Based Spectral Analysis

The implementation in `core/nsgt_kernel.rs` utilizes a sparse kernel approach to perform this transform efficiently. For each log-frequency band k, a time-domain kernel h_k is precomputed.

This kernel combines a complex sinusoid at the band's center frequency f_k with a periodic Hann window w_k of length $L_k \approx Q \cdot f_s/f_k$.

$$h_k[n] = w_k[n] \cdot e^{-j2\pi f_k n/f_s}$$

These kernels are transformed into the frequency domain ($K_k[\nu]$) during initialization. To optimize performance, the system sparsifies these frequency kernels, storing only the bins with significant energy.

During runtime, the system performs a single FFT on the input audio buffer to obtain the spectrum $X[\nu]$. The complex coefficient C_k for band k is then computed via the inner product in the frequency domain:

$$C_k = \frac{1}{N_{fft}} \sum_{\nu} X[\nu] \cdot K_k^*[\nu]$$

This "one FFT, many kernels" approach allows Conchordal to generate a high-resolution, logarithmically spaced spectrum covering 20Hz to 20kHz without the computational overhead of calculating separate DFTs for each band or using recursive filter banks.

3.1.2 Real-Time Temporal Smoothing

The raw spectral coefficients C_k exhibit high variance due to the stochastic nature of the audio input (especially with noise-based agents). To create a stable gradient for agents to follow, the `RtNsgtKernelLog2` struct wraps the NSGT with a temporal smoothing layer.

It implements a per-band leaky integrator (exponential smoothing). Crucially, the time constant τ is frequency-dependent. Low frequencies, which evolve slowly, are smoothed with a longer τ, while high frequencies, which carry transient details, have a shorter τ.

$$y_k[t] = (1 - \alpha_k) \cdot |C_k[t]| + \alpha_k \cdot y_k[t-1]$$

where the smoothing factor α_k is derived from the frame interval Δt:

$$\alpha_k = e^{-\Delta t/\tau(f_k)}$$

This models the "integration time" of the ear, ensuring that the Landscape reflects a psychoacoustic percept rather than instantaneous signal power.

3.2 Roughness (R) Calculation: The Plomp-Levelt Model

Roughness is the sensation of "harshness" or "buzzing" caused by the interference of spectral components that fall within the same critical band but are not sufficiently close to be perceived as a single tone (beating). Conchordal implements a variation of the Plomp-Levelt model via convolution in the ERB domain.

3.2.1 The Interference Kernel

The core of the calculation is the Roughness Kernel, defined in `core/roughness_kernel.rs`. This kernel $K_{rough}(\Delta z)$ models the interference curve between two partials separated by Δz ERB. The curve creates a penalty that rises rapidly as partials separate, peaks at approximately 0.25 ERB (maximum roughness), and then decays as they separate further.

The implementation uses a parameterized function `eval_kernel_delta_erb` to generate this shape:

$$g(\Delta z) = e^{-\frac{\Delta z^2}{2\sigma^2}} \cdot (1 - e^{-(\frac{\Delta z}{\sigma_{suppress}})^p})$$

The second term is a suppression factor that ensures the kernel goes to zero as $\Delta z \to 0$, preventing a single pure tone from generating self-roughness.

3.2.2 Convolutional Approach

Calculating roughness pairwise for all spectral bins (N^2 complexity) is computationally prohibitive for real-time applications. Conchordal solves this by treating the Roughness calculation as a linear convolution.

1. **Mapping**: The log-spaced amplitude spectrum from the NSGT is mapped (or interpolated) onto a linear ERB grid.
2. **Convolution**: This density $A(z)$ is convolved with the pre-calculated roughness kernel K_{rough}.

$$R(z) = (A * K_{rough})(z) = \int A(z - \tau) K_{rough}(\tau) d\tau$$

The result $R(z)$ represents the "Roughness Potential" at frequency z. If an agent were to place a tone at z, it would interact with the existing spectral energy to generate roughness proportional to $R(z)$. Agents seeking consonance actively avoid peaks in this field.

3.3 Harmonicity (H): The Sibling Projection Algorithm

While Roughness drives agents away from dissonance (segregation), Harmonicity (H) drives them toward fusion—the creation of coherent chords and timbres. Conchordal introduces a novel algorithm termed "Sibling Projection" to compute this field. This algorithm approximates the brain's mechanism of "Common Root" detection (Virtual Pitch) entirely in the frequency domain.

3.3.1 Concept: Virtual Roots

The algorithm posits that any spectral peak at frequency f implies the potential existence of a fundamental frequency (root) at its subharmonics ($f/2, f/3, f/4 \ldots$). If multiple spectral peaks share a common subharmonic, that subharmonic represents a strong "Virtual Root".

3.3.2 The Two-Pass Projection

The algorithm operates on the `Log2Space` spectrum in two passes, utilizing the integer properties of the logarithmic grid:

1. **Downward Projection (Root Search)**: The current spectral envelope is "smeared" downward. For every bin i with energy, the algorithm adds energy to bins $i - \log_2(k)$ for integers \$k ∈ {1, 2, …, N\$.

$$Roots[i] = \sum_k A[i + \log_2(k)] \cdot w_k$$

   Here, w_k is a weighting factor that decays with harmonic index k (e.g., $k^{-\rho}$), reflecting that lower harmonics imply their roots more strongly than higher ones. The result `Roots` describes the strength of the virtual pitch at every frequency.

2. **Upward Projection (Harmonic Resonance)**: The system then projects the `Roots` spectrum back upwards. If a strong root exists at f_r, it implies stability for all its natural harmonics ($f_r, 2f_r, 3f_r \ldots$).

$$H[i] = \sum_m Roots[i - \log_2(m)] \cdot w_m$$

Emergent Tonal Stability: Consider an environment with a single tone at 200 Hz.

- **Step 1 (Down)**: It projects roots at 100 Hz ($f/2$), 66.6 Hz ($f/3$), 50 Hz ($f/4$), etc.
- **Step 2 (Up)**: The 100 Hz root projects stability to 100, 200, 300, 400, 500… Hz.
  - 300 Hz is the Perfect 5th of the 100 Hz root.
  - 500 Hz is the Major 3rd of the 100 Hz root.

Thus, without any hardcoded knowledge of Western music theory, the system naturally generates stability peaks at the Major 3rd and Perfect 5th relationships, simply as a consequence of the physics of the harmonic series. An agent at 200 Hz creates a "gravity well" at 300 Hz and 500 Hz, inviting other agents to form a major triad.

3.3.3 Mirror Dualism: Overtone vs. Undertone

The implementation in `core/harmonicity_kernel.rs` includes a profound parameter: `mirror_weight` (α). This parameter blends two distinct projection paths:

- **Path A (Overtone/Major)**: The standard "Down-then-Up" projection described above. It creates gravity based on the Overtone Series, favoring Major tonalities.
- **Path B (Undertone/Minor)**: An inverted "Up-then-Down" projection. It finds common overtones and projects undertones. This is the theoretical dual of Path A and favors Minor or Phrygian tonalities (the Undertone Series).

$$H_{final} = (1 - \alpha)H_{overtone} + \alpha H_{undertone}$$

By modulating `mirror_weight`, a user can continuously morph the fundamental physics of the universe from Major-centric to Minor-centric, observing how the ecosystem reorganizes itself in response.

3.4 Habituation: The Boredom Mechanism

To prevent the system from finding a single optimal chord (e.g., a perfect unison or octave) and staying there forever, Conchordal implements a physiological model of habituation (auditory fatigue).

A Habituation field tracks the history of spectral energy. It is essentially a very slow leaky integrator of the amplitude spectrum (time constant τ_{hab} typically 5–10 seconds).

$$\Psi[t] = (1 - \beta) \cdot A[t] + \beta \cdot \Psi[t - 1]$$

This field Ψ is subtracted from the final Consonance C. If an agent lingers at a frequency f for too long, $\Psi(f)$ increases, $C(f)$ decreases, and the once-comfortable spot becomes "boring" (low fitness). This forces the agent to migrate to a new, fresh frequency, driving the perpetual evolution of the music.

4. The Life Engine: Agents and Autonomy

The "Life Engine" is the agent-based simulation layer that runs atop the DSP landscape. It manages the population of "Individuals," handling their lifecycle, sensory processing, and actuation (audio synthesis).

4.1 The Individual Architecture

The `Individual` struct (`life/individual.rs`) is the atomic unit of the ecosystem. It is composed of two distinct modules, separating the biological "Brain" from the physical "Body".

4.1.1 The SoundBody (Actuator)

The `SoundBody` trait defines the sound generation capabilities of an agent. It is responsible for rendering the waveform and projecting its spectral footprint back to the system (for the Landscape update).

- `SineBody`: Synthesizes a pure sine tone.
- `HarmonicBody`: Synthesizes a complex tone consisting of a fundamental and a series of partials. This body introduces the concept of a `TimbreGenotype`, which encodes parameters such as:
  - `stiffness`: The inharmonicity coefficient (stretching the partial series).
  - `brightness`: The spectral slope (decay of higher partials).
  - `damping`: Frequency-dependent decay rates.
  - `mode`: Harmonic (integer multiples) vs. Metallic (non-integer ratios).

The `HarmonicBody` allows for the evolution of timbre. An agent with high stiffness might find survival difficult in a purely harmonic landscape, forcing it to seek out unique "spectral niches" where its inharmonic partials do not clash with the population.

4.1.2 The NeuralCore (Controller)

The `NeuralCore` trait defines the agent's behavior and internal state. It processes sensory input (C, R, Ψ) and determines the agent's output state (`ArticulationSignal`).

- `KuramotoCore`: This is the most advanced brain type, implementing the Kuramoto model for phase synchronization. It couples the agent's internal clock to the global `NeuralRhythms` (see Section 5).
- `DroneCore`: A simplified brain for ambient textures, focusing on slow drifts and long sustain.
- `SequencedCore`: A deterministic brain that follows a pre-programmed envelope, useful for "machine" elements or rhythmic seeds.

4.2 Lifecycle and Metabolism

Agents in Conchordal are governed by energy dynamics modeled on biological metabolism. The `LifecycleConfig` defines two modes of existence:

- **Decay**: The agent is born with a fixed `initial_energy` pool. It expends this energy over time (half-life) and dies when it reaches zero. This models transient sounds like plucks or percussion.
- **Sustain**: The agent has a `metabolism_rate` (energy loss per second) but can gain energy via `breath_gain`.
  - **Breath Gain**: This is the critical feedback loop. The rate at which an agent regains energy is a function of its current Consonance.
  - **Survival**: An agent in a dissonant (low C) region "starves"—its energy depletes, its amplitude fades, and it eventually dies. An agent in a consonant (high C) region "feeds"—it maintains or gains energy, allowing it to sing louder and live longer.

This mechanic creates a Darwinian pressure: **Survival of the Consonant**. The musical structure emerges because only the agents that find harmonic relationships survive to be heard.

4.3 Organic Movement Logic

Agents are not static; they move through frequency space to improve their fitness. The `update_organic_movement` method implements a gradient ascent algorithm with biological constraints.

1. **Sensory Polling**: The agent samples the Landscape at its current frequency f, and at nearby points $f \pm \delta$.
2. **Gradient Analysis**: It determines the direction of increasing consonance (C).
3. **Commitment vs. Drift**:
   - The `commitment` parameter determines the agent's "stubbornness." A high commitment makes the agent resist moving, effectively anchoring the harmony. A low commitment makes the agent fluid, sliding easily into new chords.
   - A `PinkNoise` generator adds sway (random drift). This stochasticity prevents the system from getting stuck in local optima (static chords) and gives the movement an organic, living quality.

The movement is continuous in `Log2Space`, creating portamento / glissando effects as agents slide from a dissonant interval (e.g., a tritone) into a resolution (e.g., a perfect fifth).

5. Temporal Dynamics: Neural Rhythms

Conchordal eschews the concept of a master clock or metronome. Instead, time is structured by a continuous modulation field inspired by Neural Oscillations (brainwaves). This is the "Time" equivalent of the "Space" landscape.

5.1 The Modulation Bank

The `NeuralRhythms` struct manages a bank of resonating filters tuned to physiological frequency bands:

- **Delta (0.5–4 Hz)**: The macroscopic "pulse" of the ecosystem. Agents locked to this band play long, phrase-level notes.
- **Theta (4–8 Hz)**: The "articulation" rate. Governs syllabic rhythms and medium-speed motifs.
- **Alpha (8–12 Hz)**: The "texture" rate. Used for tremolo, vibrato, and shimmering effects.
- **Beta (15–30 Hz)**: The "tension" rate. High-speed flutters associated with dissonance or excitement.

5.2 Vitality and Self-Oscillation

Each band is implemented as a Resonator, a damped harmonic oscillator. A key parameter is `vitality`.

- `Vitality = 0`: The resonator acts as a passive filter. It only rings when excited by an event (e.g., a loud agent spawning) and then decays.
- `Vitality > 0`: The resonator has active gain. It can self-oscillate, maintaining a rhythmic cycle even in the absence of input.

This creates a two-way interaction: The global rhythm drives the agents (entrainment), but the agents also drive the global rhythm (excitation). A loud "kick" agent spawning in the Delta band will "ring" the Delta resonator, causing other agents coupled to that band to synchronize.

5.3 Kuramoto Entrainment

The `KuramotoCore` brain implements the Kuramoto model of coupled oscillators.

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N} \sum_{j=1}^{N} \sin(\theta_j - \theta_i)$$

In Conchordal, the "coupling" K is to the global `NeuralRhythms` rather than directly to every other

agent (Mean Field approximation).

- **Sensitivity**: Each agent has a sensitivity profile determining which bands (Delta, Theta, etc.) it listens to.
- **Phase Locking**: Agents adjust their internal articulation phase to match the phase of the resonator.

This results in emergent synchronization. Agents spawned at random times will gradually align their attacks to the beat of the Delta or Theta bands, creating coherent rhythmic patterns without a central sequencer.

6. System Architecture and Implementation Details

Conchordal is implemented in Rust to satisfy the stringent requirements of real-time audio (latency < 10ms) alongside heavy numerical analysis (NSGT/Convolution). The architecture uses a concurrent, lock-free design pattern.

6.1 Threading Model

The application creates three primary thread contexts:

1. **Audio Thread (Real-Time Priority)**:
   - Managed by `cpal` in `audio/output.rs`.
   - **Constraint**: Must never block. No Mutexes, no memory allocation.
   - **Responsibility**: Iterates through the `Population`, calling `render_wave` on every active agent, mixing the output, and pushing to the hardware buffer. It reads from a read-only snapshot of the Landscape.
2. **Analysis Worker (Background Priority)**:
   - Defined in `life/analysis_worker.rs`.
   - **Responsibility**: Performs the heavy DSP. It receives copies of the audio buffer via a ring buffer. It runs the NSGT, Roughness convolution, and Harmonicity projection.
   - **Update Cycle**: When analysis is complete, it constructs a new `LandscapeFrame` and sends it to the Audio thread via a lock-free Single-Producer Single-Consumer (SPSC) channel.
3. **App/GUI Thread (Main)**:
   - Runs the `egui` visualizer and the `Rhai` scripting engine.
   - **Responsibility**: Handles user input, visualizing the Landscape (`ui/plots.rs`), and executing the Scenario script. It sends command actions (e.g., `SpawnAgent`, `SetGlobalParameter`) to the `Population`.

6.2 Data Flow and Double Buffering

To maintain data consistency without locking the audio thread, Conchordal uses a double-buffering strategy for the Landscape.

1. The Analysis Worker builds a new Landscape in the background.
2. The `Population` holds the "current" Landscape.
3. When a new frame arrives from the worker, the `Population` atomically swaps the pointer (or replaces the data structure) at the start of a processing block. This ensures that the audio thread always sees a consistent snapshot of the physics, even if the analysis lags slightly behind real-time.

6.3 The Conductor: Scripting with Rhai

The Conductor module acts as the interface between the human artist and the ecosystem. It embeds the Rhai scripting language, exposing a high-level API for controlling the simulation.

The `ScriptHost` struct maps internal Rust functions to Rhai commands:

- `spawn(tag, method, life,...)`: Maps to `Action::SpawnAgents`. Allows defining probabilistic spawn clouds (e.g., "Spawn 5 agents in the 200-400Hz range using Harmonicity density").
- `set_harmonicity(map)`: Maps to `Action::SetHarmonicity`. Allows real-time modulation of physics parameters like `mirror_weight`.
- `wait(seconds)`: A non-blocking wait that yields control back to the event loop, allowing the script to govern the timeline.

Scenario Parsing: Scenarios are loaded from `.rhai` or `.json5` files. This separation allows users to compose the "Macro-Structure" (the narrative arc, the changing laws of physics) while the "Micro-Structure" (the specific notes and rhythms) emerges from the agents' adaptation to those changes.

7. Case Studies: Analysis of Emergent Behavior

The following examples, derived from the `samples/` directory, illustrate how specific parameter configurations lead to complex musical behaviors.

7.1 Case Study: Self-Organizing Rhythm (`self_organizing_rhythm.rhai`)

This script demonstrates the emergent quantization of time.

1. **Phase 1 (The Seed)**: A single, high-energy agent "Kick" is spawned at 60 Hz. Its periodic articulation excites the Delta band resonator in the `NeuralRhythms`.
2. **Phase 2 (The Swarm)**: A cloud of agents is spawned with random phases.
3. **Emergence**: Because the agents have `KuramotoCore` brains coupled to the Delta band, they sense the rhythm established by the Kick. Over a period of seconds, their phases drift and lock into alignment with the Kick. The result is a synchronized pulse that was not explicitly programmed into the swarm—it arose from the physics of the coupled oscillators.

7.2 Case Study: Mirror Dualism (`mirror_dualism.rhai`)

This script explores the structural role of the `mirror_weight` parameter.

1. **Setup**: An anchor drone is established at C4 (261.63 Hz).
2. **State A (Major)**: `set_harmonicity(#{ mirror: 0.0 })`. The system uses the Common Root projection (Overtone Series). Agents seeking consonance cluster around E4 and G4, forming a C Major triad.
3. **State B (Minor)**: `set_harmonicity(#{ mirror: 1.0 })`. The system switches to Common Overtone projection (Undertone Series). The "gravity" of the landscape inverts. Agents now find stability at Ab3 and F3 (intervals of a minor sixth and perfect fourth relative to C), creating a Phrygian/Minor texture. This demonstrates that "Tonality" in Conchordal is a manipulable environmental variable, akin to temperature or gravity.

7.3 Case Study: Drift and Flow (`drift_and_flow.rhai`)

This script validates the gradient ascent movement logic.

1. **Action**: A strongly dissonant agent (C#3) is placed next to a strong anchor (C2).
2. **Observation**: The C#3 agent immediately begins to slide in frequency. It is "pulled" by the Harmonicity gradient. It does not jump; it glides continuously (portamento) until it lands in a nearby harmonic "well" (likely E3 or G3).
3. **Dynamics**: If Habituation is enabled, the agent will settle at E3 for a few seconds, then "get bored" (local consonance drops), and slide away again to find a new stable interval. This results in an endless, non-repeating melody generated by simple physical rules of attraction and repulsion.

8. Conclusion

Conchordal successfully establishes a proof-of-concept for Bio-Mimetic Computational Audio. By replacing the rigid abstractions of music theory (notes, grids, BPM) with continuous physiolog-

ical models (`Log2Space`, ERB bands, neural oscillation), it creates a system where music is not constructed, but grown.

The technical architecture—anchored by the `Log2Space` coordinate system and the "Sibling Projection" algorithm—provides a robust mathematical foundation for this new paradigm. The use of Rust ensures that these complex biological simulations can run in real-time, bridging the gap between ALife research and performative musical instruments.

Future development of Conchordal will focus on spatialization (extending the landscape to 3D space) and evolutionary genetics (allowing successful agents to pass on their `TimbreGenotype`), further deepening the analogy between sound and life.

Appendix A: Key System Parameters

Parameter	Module	Unit	Description
`bins_per_oct`	`Log2Space`	Int	Resolution of the frequency grid (typ. 48-96).
`sigma_cents`	`Harmonicity`	Cents	Width of harmonic peaks. Lower = stricter intonation.
`mirror_weight`	`Harmonicity`	0.0-1.0	Balance between Overtone (Major) and Undertone (Minor) gravity.
`habituation_tau`	`Landscape`	Seconds	Time constant for auditory fatigue/boredom.
`roughness_k`	`Landscape`	Float	Weight of dissonance penalty in fitness function.
`vitality`	`NeuralRhythms`	0.0-1.0	Self-oscillation energy of the rhythm section.
`commitment`	`Individual`	0.0-1.0	Resistance to movement/change of an agent.

Appendix B: Mathematical Summary

Consonance Fitness Function:

$$C(f,t) = \underbrace{H(f,t)}\_\text{Harmonicity} - k \cdot \underbrace{R(f,t)}\_\text{Roughness} - \underbrace{\int_{-\infty}^{t} E(f,\tau) e^{-(t-\tau)/\tau_{hab}} d\tau}\_\text{Habituation}$$

Harmonicity Projection (Sibling Algorithm):

$$H[i] = (1-\alpha) \sum_m \left(\sum_k A[i + \log_2(k)] \right) [i - \log_2(m)] + \alpha \sum_m \left(\sum_k A[i - \log_2(k)] \right) [i + \log_2(m)]$$

Roughness Convolution:

$$R(z) = \int A(\tau) \cdot K_{plomp}(|z - \tau|_{ERB}) d\tau$$